



# UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO  
INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

---

## ESTUDIO DE RENDIMIENTO DE REDES BLUETOOTH LOW ENERGY (BLE) DE TIPO MESH

---

**Autor**

Juan Pablo Cano López

**Directores**

Francisco Javier Romero Maldonado

Victor Toral López

---

**ETSIIT**

Escuela Técnica Superior  
de Ingenierías Informática  
y de Telecomunicación

---



Granada, 27 de Marzo de 2023









# **Estudio de rendimiento de redes bluetooth low energy (BLE) de tipo mesh**

---

## **Autor**

Juan Pablo Cano López

## **Directores**

Francisco Javier Romero Maldonado

Victor Toral López



# **Estudio de rendimiento de redes bluetooth low energy(BLE) de tipo mesh**

Juan Pablo Cano López

Conceptos Clave: Tecnología BLE, iBeacon, Intervalo de Advertising, Scanning Window, Scanning Interval, Proporción de Tramas Recibidas.

## **Resumen**

La tecnología BLE, Bluetooth Low Energy es un estándar que engloba a una serie de dispositivos Bluetooth con una serie de características comunes, siendo su ventaja principal su bajo consumo, sin una disminución del rango de comunicación frente a las tecnologías anteriores.

Estas características permiten que el uso de estos dispositivos esté experimentando un florecimiento en su uso, tanto es así que los sistemas operativos móviles como iOS, Android, Windows Phone y Blackberry, así como los de computadores Windows 8, Windows 10, Linux y MacOS sean compatibles de forma nativa con esta tecnología.

Existen numerosas aplicaciones para esta tecnología, entre ellas, destaca sobre todo dentro del campo del internet de las cosas. Las casas domóticas pueden estar operadas de manera eficiente a través de una red de dispositivos de este tipo, no suponiendo apenas carga para el consumo eléctrico, debido a su gran eficiencia. Por otro lado, también se ha utilizado para la identificación de personal en lugares de trabajo, a través de tarjetas o accesorios.

El trabajo consiste en estudiar en cierta profundidad los cambios en la proporción de tramas recibidas en diversos escenarios, variando entre otros parámetros básicos de los iBeacons, como lo son el intervalo de advertising, la scanning window y el scanning interval.

Además interesa, en particular, determinar si es necesario retransmitir las tramas más de una vez, en escenarios con retransmisores, para asegurar una recepción aceptable, ya que esto se propone como una posible mejora al modelo broadcast tradicional (retransmisión global con cierto número de saltos límite).





# **Bluetooth low energy( BLE) mesh networks: Efficiency Study**

Juan Pablo Cano López

Keywords: BLE Technology, iBeacon, Advertising Interval, Scanning Window, Scanning Interval, Information Frames Received Ratio.

## **Summary**

BLE Technology, Bluetooth Low Energy is a standard that encompasses a group of Bluetooth devices Bluetooth with a number of similar features, being their main advantage, the energy consumption efficiency, without losing communication range, compared to previous technologies .

These features allow the usage of these devices to have flourished, a proof of this is that mobile operating systems such as iOS, Android, Windows Phone and Blackberry as well as PC OS like Windows 8, Windows 10, Linux y MacOS support this technology natively.

There are a myriad of applications using this technology, being one of its spotlights in the IoT(Internet of Things) field. In home automation, the house appliances can be managed using a BLE network, not implying a noticeable increase in electrical consumption, due to its high efficiency. On the other hand, it has been employed in personal identification inside working environments, in the form of cards or accessories.

This project consists in studying deeply the changes in the information frames received ratio in diverse cases, modifying these devices' basic parameters as the advertising interval, the scanning window and the scanning interval.

Moreover, it is particularly interesting to measure, if it is necessary to rebroadcast information frames more than once, to ensure a reasonable reception ratio, since this would be an improvement to the traditional broadcast model (in which every message is broadcasted with a certain hop limit).



---

Yo, **Juan Pablo Cano López**, alumno de la titulación, **Ingeniería de Telecomunicaciones de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación** de la Universidad de Granada, con DNI:75579036R, autorizo la ubicación de la siguiente copia de mi trabajo de fin de grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Juan Pablo Cano López

Granada a 7 de Septiembre de 2023



---

**D. Francisco Javier Romero Maldonado** Profesor del Área de la tecnología electrónica del Departamento de Electrónica y Tecnología de Computadores de la universidad de Granada

**D. Victor Toral López** Profesor del Área de la tecnología electrónica de Departamento de Electrónica y Tecnología de Computadores de la universidad de Granada

**Informan:**

Que el presente trabajo, titulado **Estudio de Rendimiento de Redes Low Energy de tipo Mesh**, ha sido realizado bajo su supervisión por **Juan Pablo Cano López**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 7 de Septiembre de 2023.

**Los directores:**

**Francisco Javier Romero Maldonado**

**Victor Toral López**



# **Agradecimientos**

Agradezco a los tutores del trabajo la atención dedicada, en especial a D.Francisco Javier Romero Maldonado, el cuál ha dedicado parte del mes de Agosto, a tutorías con el objetivo de que el proyecto pudiera tener éxito y sin las que, probablemente, no se podría presentar este estudio. Además debo agradecer a la empresa eesy-innovation Spain SL por proporcionar gratuitamente dos de los dispositivos dongles utilizados. A parte de ello me gustaría agradecer a mi familia el gran apoyo mostrado durante el desarrollo del proyecto.





# Índice de Contenidos

<b>1-Motivación y Finalidad</b>	/ Página: 20
1.1-Introducción	/ Página: 20
1.2 Finalidad	/ Página: 20
<b>2-Marco Teórico</b>	/ Página: 22
2.1-¿Qué es un dispositivo BLE?	/ Página: 22
2.2-¿Qué es un iBeacon?	/ Página: 22
2.3-Eschema de comunicación	/ Página: 23
2.3.1-Intervalo de Advertising	/ Página: 24
2.3.2-Scanning Window	/ Página: 24
2.3.3-Scanning Interval	/ Página: 25
<b>3-Desarrollo del Proyecto</b>	/ Página: 26
3.1-Software y Hardware Utilizado	/ Página: 26
3.1.1-CY5677 CYSMART	/ Página: 26
3.1.2-PSoC Creator	/ Página: 26
3.1.3-NRF Connect	/ Página: 26
3.1.4-Hterm.exe	/ Página: 27
3.2-Configuración de los Dispositivos	/ Página: 28
3.2.1-Otros	/ Página: 28
3.2.2-Advertiser	/ Página: 31
3.2.3-Scanner	/ Página: 36
3.2.4-ReTransmisor	/ Página: 39
3.3-Explicación del Código	/ Página: 42
3.3.1-Advertiser	/ Página: 42
3.3.2-Scanner	/ Página: 44
3.3.3-ReTransmisor	/ Página: 46
<b>4-Desarrollo del Estudio</b>	/ Página: 50
4.1-Escenarios de Estudio	/ Página: 50
4.1.1-Escenarios en función de los parámetros básicos	/ Página: 50
4.1.2-Escenarios sin retransmisores	/ Página: 53
4.1.3-Escenarios con retransmisores	/ Página: 54
4.2-Resultados del Estudio	/ Página: 55
4.2.1-Conclusiones Generales	/ Página: 55
4.2.2-Conclusiones en cuanto a Tramas Recibidas	/ Página: 57
<b>5-Conclusiones del Proyecto</b>	/ Página: 65
<b>6-Bibliografía, Webgrafía y Otras Referencias</b>	/ Página: 66



## Índice de Figuras, Tablas y Ecuaciones

<b>Figura 1-</b> Imagen de la red cedida por el profesor Francisco Javier Romero Maldonado .....	/ Página: 22
<b>Figura 2-</b> Estructura del iBeacon .....	/ Página: 23
<b>Figura 3-</b> Intervalo de Advertising, Scanning Window, Scanning Interval representation .....	/ Página: 24
<b>Figura 4-</b> Dispositivo CY5677 CYSMART Dongle .....	/ Página: 26
<b>Figura 5-</b> iBeacon propio en NRF Connect .....	/ Página: 27
<b>Figura 6-</b> Tramas Recibidas en hterm.exe .....	/ Página: 27
<b>Figura 7-</b> Tramas escaneadas en hterm.exe .....	/ Página: 28
<b>Figura 8-</b> Icono UART en PSoC .....	/ Página: 28
<b>Figura 9-</b> Configuración UART .....	/ Página: 29
<b>Figura 10-</b> Icono LED de Control .....	/ Página: 29
<b>Figura 11-</b> Configuración LED de control .....	/ Página: 30
<b>Figura 12-</b> Icono BUTTON .....	/ Página: 30
<b>Figura 13-</b> Ventana de configuración BUTTON .....	/ Página: 31
<b>Figura 14-</b> Dispositivo BLE colocado en el esquemático .....	/ Página: 32
<b>Figura 15-</b> Pestaña general dispositivo BLE .....	/ Página: 32
<b>Figura 16-</b> Pestaña GAP Settings->General dispositivo BLE .....	/ Página: 33
<b>Figura 17-</b> Pestaña GAP Settings->Advertising Settings en un anunciador .....	/ Página: 34
<b>Figura 18-</b> Pestaña GAP Settings-> Advertisement packet del anunciador .....	/ Página: 35
<b>Figura 19-</b> Asignación de pines Advertiser .....	/ Página: 36
<b>Figura 20-</b> Esquemático Advertiser .....	/ Página: 36
<b>Figura 21-</b> Ventana General de configuración del escáner .....	/ Página: 37
<b>Figura 22-</b> GAP Scan Settings .....	/ Página: 37
<b>Figura 23-</b> Esquemático final Escáner .....	/ Página: 38
<b>Figura 24-</b> Asignación de pines Escáner .....	/ Página: 39
<b>Figura 25-</b> Ventana general configuración retransmisor .....	/ Página: 39
<b>Figura 26-</b> Ventana de configuración GAP Settings-> General Retransmisor .....	/ Página: 40
<b>Figura 27-</b> Ventana de configuración Advertisement settings en un retransmisor .....	/ Página: 41
<b>Figura 28-</b> Ventana de configuración Scan settings en un retransmisor .....	/ Página: 41
<b>Figura 29-</b> Advertiser Código 1 .....	/ Página: 42
<b>Figura 30-</b> Advertiser Código 2 .....	/ Página: 43
<b>Figura 31-</b> Advertiser Código 3 .....	/ Página: 43
<b>Figura 32-</b> Advertiser Código 4 .....	/ Página: 44
<b>Figura 33-</b> Scanner Código 1 .....	/ Página: 44
<b>Figura 34-</b> Scanner Código 2 .....	/ Página: 45
<b>Figura 35-</b> Scanner Código 3 .....	/ Página: 45
<b>Figura 36-</b> Scanner Código 4 .....	/ Página: 46
<b>Figura 37-</b> Scanner Código 5 .....	/ Página: 46
<b>Figura 38-</b> Retransmisor Código 1 .....	/ Página: 47
<b>Figura 39-</b> Retransmisor Código 2 .....	/ Página: 47
<b>Figura 40-</b> Retransmisor Código 3 .....	/ Página: 48
<b>Figura 41-</b> Retransmisor Código 4 .....	/ Página: 48
<b>Figura 42-</b> Retransmisor Código 5 .....	/ Página: 49
<b>Figura 43-</b> Escenario con topología básica .....	/ Página: 53
<b>Figura 44-</b> Escenarios con dos anunciadores .....	/ Página: 53
<b>Figura 45-</b> Escenarios con 3 anunciadores .....	/ Página: 54
<b>Figura 46-</b> Escenario con anunciante, re-transmisor y escáner .....	/ Página: 54
<b>Figura 47-</b> Escenario con 2 retransmisores en serie .....	/ Página: 55

<b>Figura 48-</b> Tiempo de Transmisión /Advertising Interval .....	/ Página: 55
<b>Figura 49-</b> Gráficos Comparativos Estimación vs Paquetes Recibidos .....	/ Página: 58
<b>Figura 50-</b> Gráficos Comparativos SW/SI vs Diferencia en la Estimación para 500ms de AI .....	/ Página: 63
<b>Tab1a 1-</b> Advertising Interval 100 ms, Scan Window y Scan Interval iguales .....	/ Página: 51
<b>Tab1a 2-</b> Advertising Interval 100ms, Scan Window y Scan Interval distintos .....	/ Página: 51
<b>Tab1a 3-</b> Advertising Interval 500ms, Scan Window y Scan Interval iguales .....	/ Página: 51
<b>Tab1a 4-</b> Advertising Interval 500ms, Scan Window y Scan Interval distintos .....	/ Página: 52
<b>Tab1a 5-</b> Advertising Interval 1000ms, Scan Window y Scan Interval iguales .....	/ Página: 52
<b>Tab1a 6-</b> Advertising Interval 1000ms, Scan Window y Scan Interval distintos .....	/ Página: 52
<b>Tab1a 7-</b> Tiempo de Transmisión/ Advertising Interval .....	/ Página: 56
<b>Tab1a 8-</b> Comparativa Medidas Paquetes Recibidos con diferentes tramas o todas iguales .....	/ Página: 56
<b>Tab1a 9-</b> Comparativa Estimación vs Medida 1 Anunciador .....	/ Página: 59
<b>Tab1a 10-</b> Comparativa Estimación vs Medida 3 Anunciadores .....	/ Página: 59
<b>Tab1a 11-</b> Comparativa Paquetes Recibidos 1 retransmisor vs 2 retransmisores en cadena .....	/ Página: 60
<b>Tab1a 12-</b> Comparativa Escenarios con Retransmisores variaciones SW/SI SC .....	/ Página: 61
<b>Tab1a 13-</b> Comparativa Escenarios con Retransmisores variaciones SW/SI RT .....	/ Página: 62
<b>Tab1a 14-</b> Comparativa Diferencias Modelo y Medida Retransmisores cambios en SW/SI RT ...	/ Página: 63
<b>Ecuación 1-</b> Estimación de los paquetes recibidos a la salida para escenarios sin retransmisores ...	/ Página: 57
<b>Ecuación 2-</b> Ecuación 2- Modelo de Estimación para cambios en SW/SI en retransmisores .....	/ Página: 62

# Capítulo 1

## Motivación y Objetivos

### 1.1-Introducción

La tecnología Bluetooth ha sufrido una evolución de la que poca gente fuera de la industria es consciente, sus capacidades generales, tanto de rango de cobertura, como de velocidad de transmisión como compatibilidad e integración en dispositivos ha crecido considerablemente.

En particular el estándar BLE (Bluetooth Low Energy) encapsula a un grupo de dispositivos con buenas capacidades en cuanto a la transmisión y recepción y excelente eficiencia energética. Es por ello que su uso se vuelve más común y, a su vez, diverso conforme el tiempo transcurre.

Escenarios donde este tipo de tecnología se ha utilizado ha sido en aplicaciones IoT, redes de electrodomésticos con capacidades de BLE integradas pueden ser gestionadas de manera sencilla, sin suponer un sobre coste y posibilitando una mayor automatización gracias al intercambio de información entre todos ellos. [\[1\]](#)

A su vez, se han realizado controles de identificación o localización en ropa, pulseras, y cualquier tipo de accesorio con esta tecnología incorporada, además de esto existen módulos con capacidades BLE que incorporan sensores, de manera que los escenarios de uso se transforman, pudiendo extender redes conformadas por estos dispositivos con el objetivo de controlar cualquier tipo de magnitud, como la temperatura de las zonas abarcadas.

Un ejemplo concreto de esto último es la tesis doctoral de Carlos Andrés Cifuentes García, con master en ingeniería biomédica de la facultad de ingeniería de la Universidad Nacional de Entre Ríos, Argentina, redactada en Mayo de 2011 y titulada *Diseño e Implementación de una red inalámbrica de sensores aplicados a la instrumentación biomédica-RIB*. Se trata de una aplicación de una red de sensores con capacidad de transmisión de información al medio inalámbrico compuesta por los siguientes: un micrófono inalámbrico, un sensor GPS, un sensor de capacidad pulmonar, un termómetro, un pulso oxímetro, un sensor de presión, un sensor para la medición de actividad muscular y un podómetro. [\[2\]](#)

Esta red trabaja de forma coordinada para llevar a cabo un control completo y continuo de parámetros médicos de los pacientes. Se realizó utilizando tecnología Zigbee, no BLE, pero podría haberse realizado con esta última.

### 1.2 Finalidad

La comunicación en redes Bluetooth tradicional suele seguir el modelo de conexión central-peripheral entre dispositivos y el envío de información con modelo broadcast, es decir, los anunciadores emiten tramas con un tiempo de vida, comúnmente representado por un número de saltos, esperando que lleguen al objetivo final. La simpleza de este modelo de transmisión posee ciertas ventajas, pero a su vez, surgen desventajas, como la congestión de la red para un número elevado de dispositivos.

La estructura de red que motivó el proyecto es la siguiente:

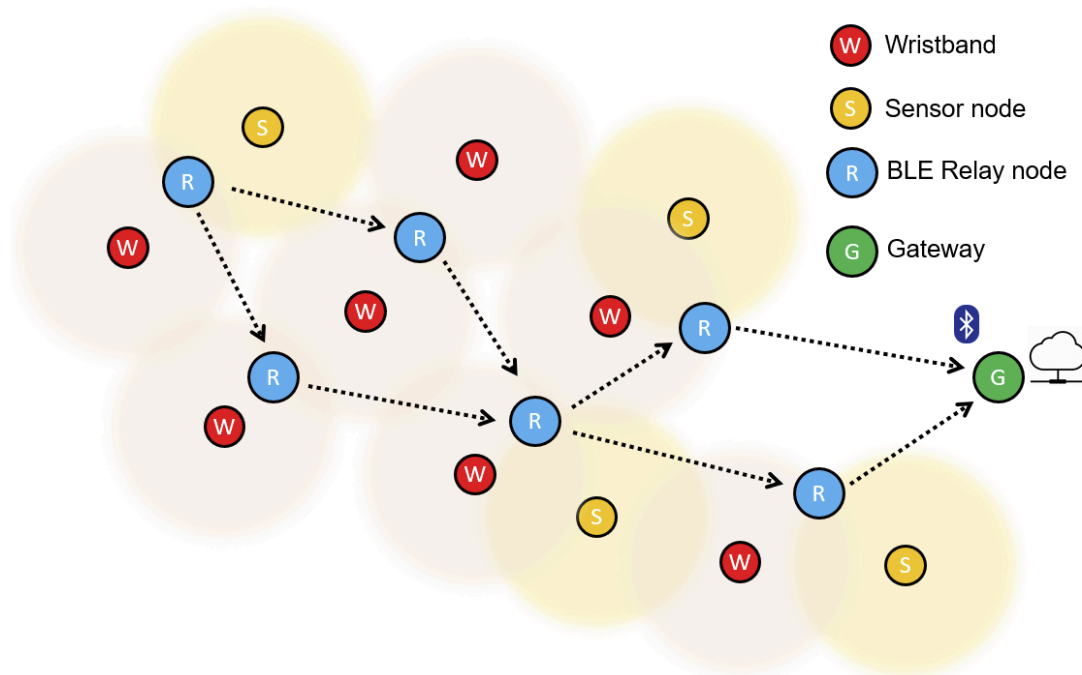


Figura 1-Imagen de la red cedida por el profesor Francisco Javier Romero Maldonado

En ella está representado como  $W$  bandas para la muñeca, como  $S$  nodos sensores, como  $R$  nodos relay y como  $G$  un Gateway final. Las bandas de la muñeca y los nodos sensores son dispositivos BLE anunciadores, los relays son en realidad dispositivos configurados como retransmisores y  $G$  sería un escáner final BLE.

La particularidad de la red es que propone un sistema de comunicación direccional, a través de retransmisores en cadena, oponiéndose al diseño tradicional broadcast.

El objetivo del proyecto es probar si este tipo de configuración supone un beneficio al modelo tradicional o, por el contrario, la eficiencia del sistema se reduce, debido al uso de los retransmisores.

## Capítulo 2

### Marco Teórico

#### 2.1-¿Qué es un dispositivo BLE?

Un dispositivo BLE es aquél que cumple los estándares de la tecnología BLE.

Esta tecnología fue creada y explotada comercialmente por el grupo Bluetooth SIG (Bluetooth Special Interest Group), una organización que agrupa a compañías de telecomunicaciones, informática, automovilismo, música, textil, automatización industrial y tecnologías de red. Los miembros del SIG se encargan de dirigir la evolución de la tecnología Bluetooth, además de fomentar su uso, incluyendo en los productos que comercializan compatibilidad con ella. [\[3\]](#)[\[4\]](#)

Los miembros corporativos más activos y con más influencia en el mundo de las telecomunicaciones del SIG son los siguientes: Ericsson, Intel Corporation, Lenovo, Microsoft, Motorola, Nokia y Toshiba.

En cuanto a las características comunes de la tecnología BLE, permiten la comunicación entre dispositivos Bluetooth operando en la banda de 2,4GHz con una tasa de transferencia de 1Mbps en la capa física, que se verá algo reducida en escenarios reales. [\[5\]](#)[\[6\]](#)

#### 2.2-¿Qué es un iBeacon?

iBeacon es un protocolo desarrollado por Apple sobre la base de la tecnología BLE, utilizando este protocolo smartphones compatibles pueden localizarse entre ellos a distancias de hasta 30 metros. En este protocolo se utilizan paquetes de advertising, cuya estructura de trama se detalla más adelante para el envío de información.

Apple lo ha utilizado enfocándose en el marketing y el aumento de las ventas. Ejemplos de reales donde se ha aplicado son los siguientes: Transmisión de información específica en exposiciones cuando el usuario se localiza en un lugar concreto a través del teléfono, pagos a través del teléfono móvil en conciertos, envío de ofertas especiales de tiendas o grandes almacenes a sus clientes, envío de ofertas en restaurantes a personas cercanas, ayudar en la navegación en ferias comerciales y finalmente localización de personas.

En particular la estructura de la trama de información es lo más relevante en cuanto al desarrollo del proyecto.

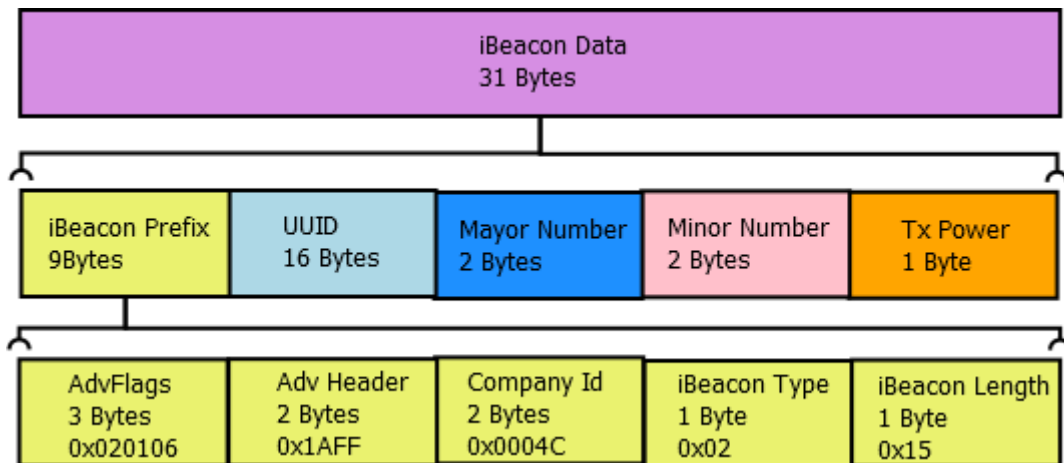


Figura 2- Estructura del iBeacon

Es una trama de 31 Bytes. Los 9 primeros forman el prefijo del iBeacon e indican, los 3 primeros, flags de advertising, los 2 siguientes la cabecera de advertising, los 2 posteriores la identificación de la compañía, a continuación, en un byte, el tipo de iBeacon utilizado y finalmente, el el último byte la longitud de la trama.

Seguidamente se incluyen 16 Bytes que identifican a dispositivos de la red. Posteriormente en los 4 bytes siguientes dos números de 1 a 65535 sin signo que pretenden identificar la trama en un grupo común, como las que pertenecen a una habitación o un piso de un edificio. Finalmente el último byte indica la potencia de transmisión con la que se emitirá la trama.

Además del iBeacon existen otros formatos para la trama de anuncio compatibles con la tecnología BLE, más concretamente, la especificación AltBeacon surge como solución ante la no existencia previa de una especificación abierta e interoperable para beacons de proximidad. Es de libre uso para todo el mundo, sin la necesidad de pago de tasas y sin favorecer específicamente a ciertas empresas frente a otras. [7]

Además de AltBeacon existe otra especificación alternativa para el formato de la trama de anuncio, Eddystone. Esta última fue desarrollada por Google en Julio de 2015, sin embargo a partir de Diciembre de 2018 Google detuvo el envío de tramas con este formato, debido a, entre otras razones, su bajo uso, una mala experiencia de los usuarios y la detección de fallos de seguridad, por los cuales conversaciones podían ser ilegalmente grabadas combinando la utilización de balizas Eddystone en colaboración con Nearby Messages API, una aplicación con modelo publicador-suscriptor, que permite el envío de pequeñas cantidades de información a dispositivos iOS y Android conectados a internet.

## 2.3-Eschema de comunicación

Ahora que ya se ha explicado que es un iBeacon y la estructura de la trama se deben introducir los conceptos de tiempo de advertising, scanning window y scanning interval, que son los parámetros principales que se modifican en cada uno de los diferentes escenarios de estudio.

Para comprender estos conceptos de manera intuitiva se incluye el siguiente gráfico:



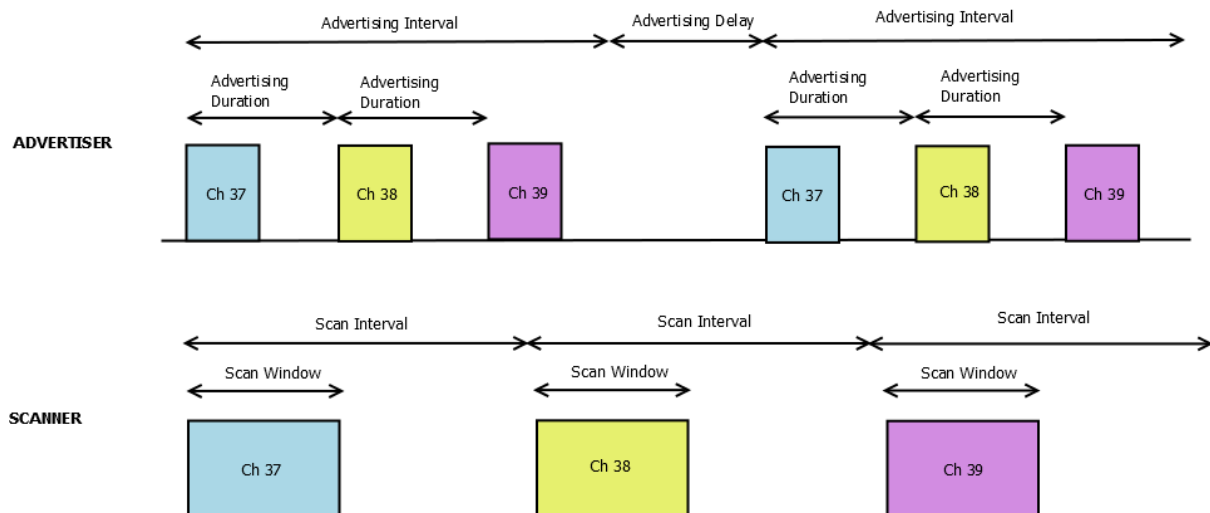


Figura 3- Intervalo de Advertising, Scanning Window, Scanning Interval representation

Los dispositivos BLE cuentan con 40 canales físicos que operan a 2,4 GHz en la banda ISM, de 2400 MHz a 2483,5 MHz, asignando a cada canal una anchura de 2MHz. Dentro de estos canales existen 37 canales de datos y 3 canales de advertising, en concreto, el canal 37, 38 y 39, que son, precisamente, los que aparecen en la figura.

Estos tres canales se encuentran en frecuencias de 2402MHz, el canal 37, 2426 MHz, el canal 38 y 2480 MHz, el canal 39, notablemente separados entre ellos respecto de canales adyacentes. Esto es así para reducir la posibilidad de interferencias con dispositivos que operan en el mismo espectro, como redes WAN, y otros aparatos de la misma índole, por ejemplo, aquellos BLE encontrados en la misma red. [8]

En el gráfico se puede observar como se hace uso de estos tres canales tanto para el advertising como para el escaneo. A continuación se procede a explicar el resto de conceptos clave que aparecen en la figura.

### 2.3.1-Intervalo de Advertising

Los dispositivos BLE transmiten paquetes en 3 canales, el 37, el 38 y el 39. Los paquetes transmitidos son los mismos en los tres canales, redundancia que se introduce con el objetivo de facilitar que las tramas no se pierdan.

El *intervalo de advertising* es el tiempo en el que el advertiser transmite un paquete en los tres canales (3 paquetes en total). Cuanto mayor sea más se tardará en transmitir un cierto número fijo de paquetes, lo cual se hace muy notable para un número de paquetes considerable.

Además de este concepto surge otro, la *duración del advertising*, que representa el tiempo en el que el transmisor anuncia el paquete en un solo canal, antes de cambiar al siguiente y el *advertising delay*, que es un tiempo que transcurre tras la finalización de cada intervalo de advertising, normalmente debido al tiempo necesario para que el BLE procese los eventos.

### 2.3.2-Scanning Window

Los dispositivos BLE escanean paquetes también en estos mismos tres canales, anteriormente mencionados, el 37, 38 y 39. Se reciben tramas de información de todos los dispositivos en el

rango de cobertura, por lo que se debe hacer realizar un filtrado, buscando solo las tramas de los dispositivos pertenecientes a nuestra red.

La *scanning window* es el tiempo donde el dispositivo detectará tramas en un mismo canal, antes de hacerlo en el canal siguiente, entre los 3 canales utilizados. Cuanto mayor sea más tramas se recibirán, sin embargo aparece la siguiente problemática, el dispositivo no puede realizar ninguna otra acción durante el tiempo de ventana de escaneo, por ello, se debe alcanzar un compromiso entre los paquetes que es aceptable recibir, dado un número fijo de paquetes transmitidos.

### 2.3.3-Scanning Interval

El *scanning interval* es el tiempo completo que comprende el tiempo de la ventana de escaneo más un tiempo donde no está activa y el dispositivo puede realizar otras acciones. Cuanto mayor sea la diferencia entre la ventana de escaneo y el scanning interval, mayor será la proporción de tramas perdidas.

## Capítulo 3

### Desarrollo del Proyecto

#### 3.1-Software y Hardware Utilizado

##### 3.1.1-CY5677 CYSMART

Son los dispositivos físicos utilizados en el proyecto, configurados como anunciadores, escáner o retransmisores. Se trata de un dispositivo puente, que ofrece acceso a tecnología BLE a través de una conexión USB. Se programa utilizando el software PSoC Creator desarrollado por la empresa Cypress Semiconductor Corporation. [\[9\]](#)

Es una solución de chip único que ofrece un front end analógico programable, periféricos digitales programables, tecnología de sensores de tacto CapSense y recursos radio compatibles con la tecnología Bluetooth BLE 4.2.

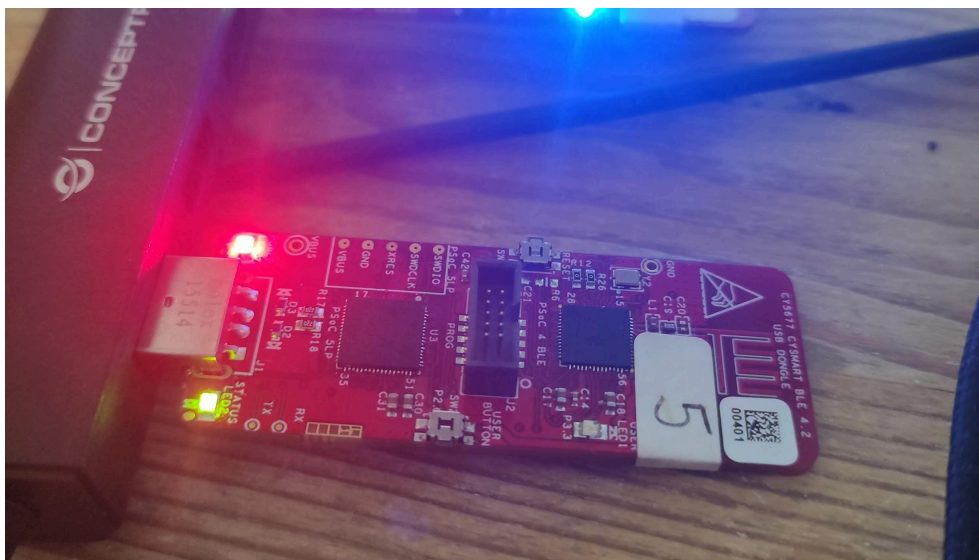


Figura 4- Dispositivo CY5677 CYSMART Dongle

##### 3.1.2-PSoC Creator

PSoC es la denominación comercial de una familia de microcontroladores programables diseñados y comercializados por la empresa Cypress Semiconductor desde 2002, con la particularidad de incorporar en un solo chip un sistema programable completo.

PSoC Creator es un software de uso gratuito compatible con sistemas operativos Windows que permite el desarrollo de proyectos con tecnología PSoC. Es un entorno integrado de diseño que permite edición simultánea de hardware y firmware y la compilación y el depurado de código desarrollado para este tipo de dispositivos. [\[10\]](#)

Se puede descargar gratuitamente en la página de la empresa Infineon Technologies AG, empresa que adquirió a la corporación Cypress Semiconductor, desarrolladora del programa.

##### 3.1.3-NRF Connect

Es un software diseñado por Nordic Semiconductor ASA, una empresa Noruega especializada en el desarrollo de tecnología de comunicación inalámbrica de muy baja potencia y de software para el desarrollo de productos IoT.

NRF Connect es una herramienta que permite el escaneo de dispositivos Bluetooth de bajo consumo desde sistemas operativos móviles y de computadoras. La visualización de algunas características de la comunicación es posible a través de este software.

Se utilizó al inicio del proyecto para visualizar si se conseguía una transmisión satisfactoria de las tramas por parte del anunciador en formato iBeacon. Se incluye una imagen de prueba.

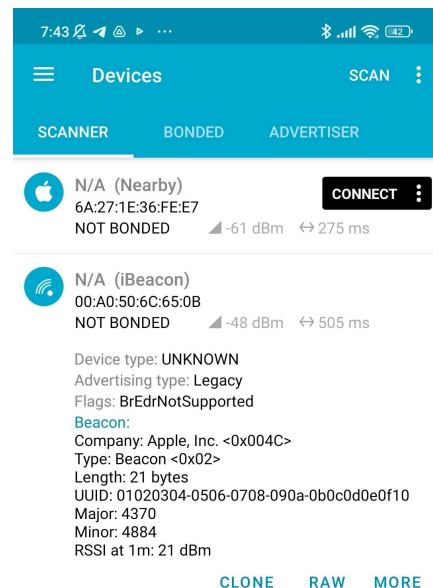


Figura 5-iBeacon propio en NRF Connect

### 3.1.4- HTERM

Es un software creado por Tobias Hammer que permite realizar comunicaciones serie en sistemas operativos Windows y Linux en ejecución. Se ha hecho uso de esta herramienta a lo largo del desarrollo completo del proyecto para posibilitar la visualización de datos. Se incluye una captura donde se incluye la visualización de las tramas recibidas finales y otra de las tramas escaneadas a continuación.

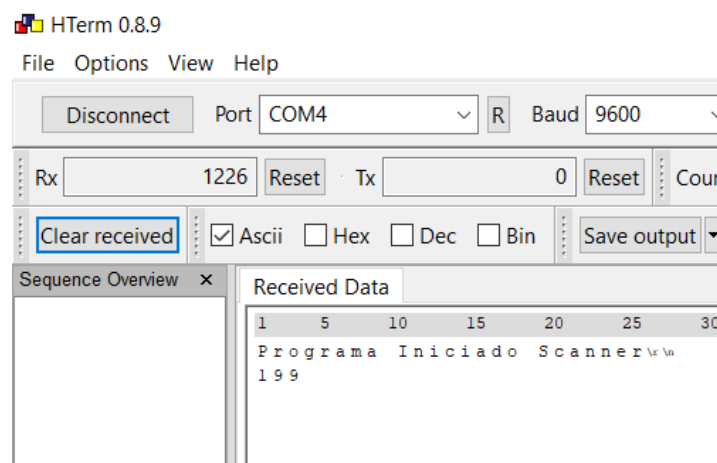


Figura 6-Tramas Recibidas en hterm.exe

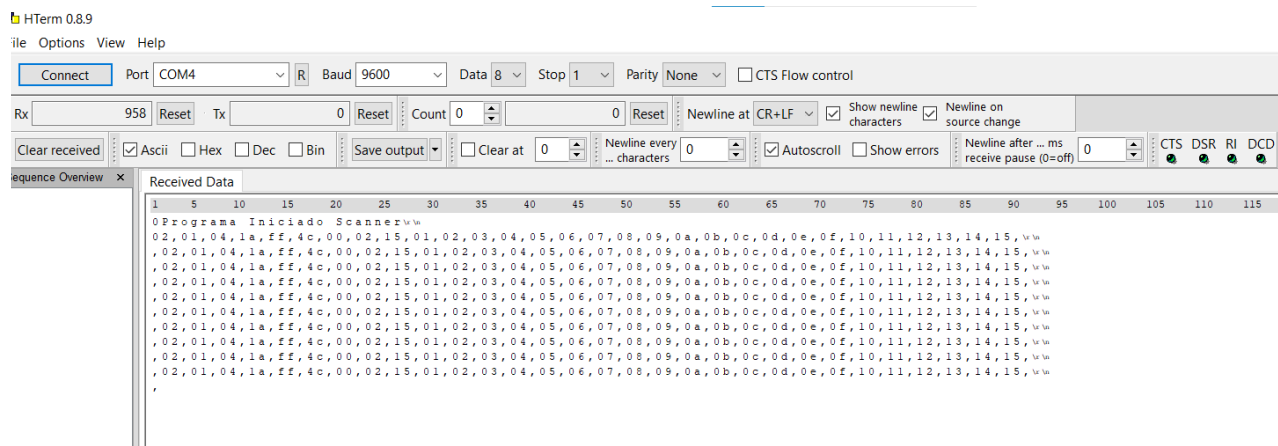


Figura 7-Tramas escaneadas en hterm.exe

## 3.2-Configuración de los Dispositivos

Antes de que los programas puedan ser ejecutados se necesita realizar una configuración correcta de los dispositivos, además, es dentro de esta configuración donde se pueden modificar ciertos parámetros vitales en el estudio posterior.

Se dispone de 4 CY5677 CYSMART™, 4 instrumentos que adoptarán roles diferenciados según su configuración, por lo que es de gran importancia que esta última sea correcta, ya que, de no ser así la red no se comportará de la forma esperada.

Es destacable que, en cuanto a la configuración, para cada uno de los dispositivos se realiza un proyecto distinto dentro del PSoC Creator, a pesar de tener una funcionamiento combinado como red, los cuatro dispositivos configurados. [\[11\]](#)[\[12\]](#)

### 3.2.1-Otros

UART(Universal Asynchronous Receiver/Transmitter). Todos los proyectos realizados utilizan una para la transmisión de información hacia y desde los dispositivos y a su vez, tener la posibilidad de visualización de esta.

Se comienza la configuración colocando el dispositivo en el esquemático, llamado TopDesing.cysch. Este dispositivo puede encontrarse en la sección Cypress-> Communications-> UART[v2.50].

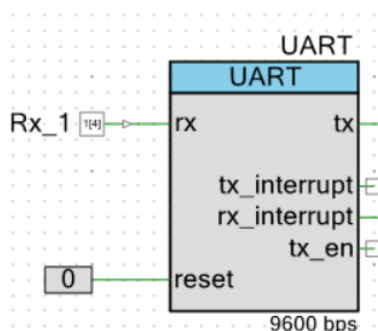


Figura 8-Icono UART en PSoC

La imagen superior muestra el icono con el que aparecerá al colocarla en el diseño. Seguidamente, si se clica en ella dos veces se puede acceder a la sección de configuración, donde lo destacable para el proyecto aparece en la figura inferior a este párrafo.

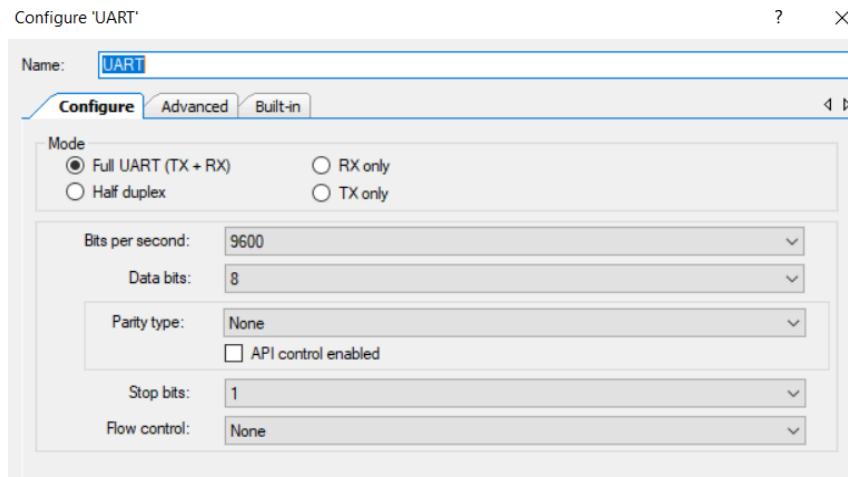


Figura 9-Configuración UART

En esta ventana emergente, dentro de la sección “Configure” se selecciona un modo de recepción y transmisión, que puede ser transmisión solo (TX only), recepción solo (RX only), Full UART (TX+RX) transmisión y recepción simultánea y Half duplex, transmisión y recepción pero solo en uno de estos modos para un mismo instante de tiempo.

En los proyectos realizados que intervienen en el estudio final solo se utiliza para mostrar información que recibe, no se necesita transmitir información a los dispositivos, pero se han configurado como Full UART porque no se ha detectado que haya supuesto una disminución en el rendimiento de los dispositivos.

Bits per second indica la velocidad de transmisión y data bits los bits que se considera un símbolo, a partir de estos dos parámetros se obtienen los baudios, (símbolos/s) a los que funciona la UART, información que se debe tener en cuenta para una visualización correcta del contenido transmitido a través de la UART. Con esto termina la configuración de la UART en el diseño.

*LED de control:* Se ha hecho uso de este componente para cronometrar los tiempos de transmisión, además de realizar otro tipo de pruebas de control durante el desarrollo de los proyectos. Cabe destacar que, a pesar de colocar el elemento en el esquemático, no es un dispositivo físico a parte de los BLE dongles, sino que se encuentra integrado en ellos.

Para utilizarlo, en primer lugar debe colocar en el esquemático y puede ser encontrado en la sección Cypress-> Ports and Pins->Digital Output Pin. En la imagen a continuación se puede visualizar este componente en el diseño.



Figura 10-Icono LED de Control

Para acceder a los parámetros que dictaminan su comportamiento se debe clicar en el componente. Una vez hecho esto se mostrará la siguiente ventana:

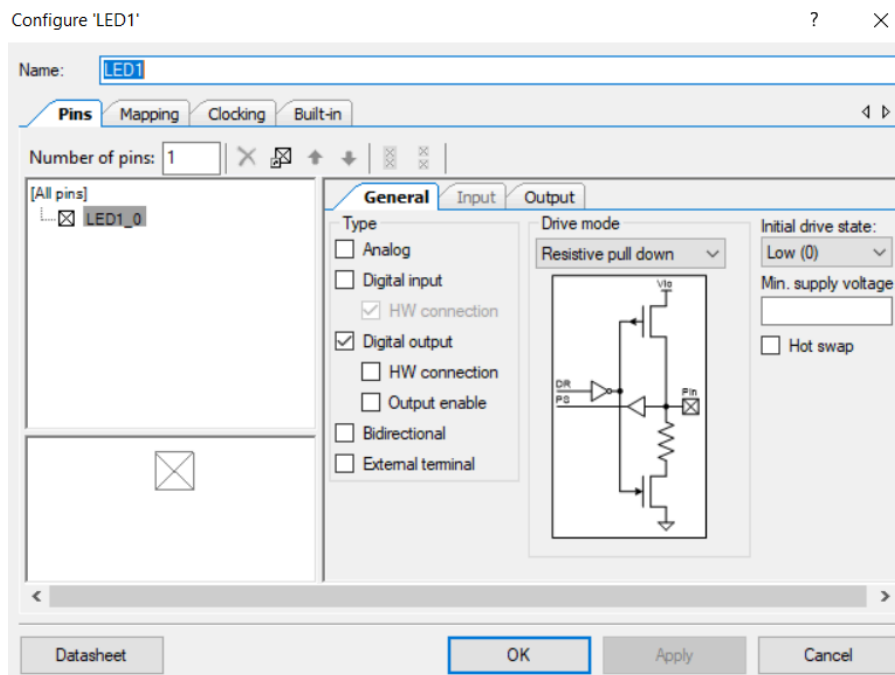


Figura 11-Configuración LED de control

Existen numerosas opciones de configuración, pero solo se van a comentar las relevantes para los proyectos realizados.

Entre estas, el drive mode indica la estructura del circuito interno que se formará entre el dispositivo BLE y el LED. Resistive pull down es una de ellas, en la cual se procura que la caída de voltaje en el diodo en su estado por defecto sea cercana a 0V. Además Initial drive state: Low(0) indica que el diodo debe comenzar apagado por defecto, siendo la opción High(1), la contraria, comenzaría encendido si se seleccionase esta última opción. En los programas realizados se comienza con el diodo apagado, así que es importante seleccionar esta opción en el diseño. Con esto se termina la configuración del diodo de control.

*Botón de usuario (BUTTON):* Los dongles BLE utilizados cuentan con dos botones en sus laterales, uno de ellos es el botón de reset, para comenzar de nuevo la ejecución de un programa desde el inicio y el otro el botón de usuario. Es a este último al que se refiere la sección actual. Este botón ha sido utilizado en el escáner, para pedirle al dispositivo que transmita por la UART un contador que indicará el número de tramas recibidas para el ensayo actual, si se ha realizado correctamente.

Este botón se encuentra en la misma sección que el diodo LED anterior, Cypress->Ports and Pins-> Digital Input Pin. Su ventana de configuración e icono es muy similar al caso previo.



Figura 12-Icono BUTTON

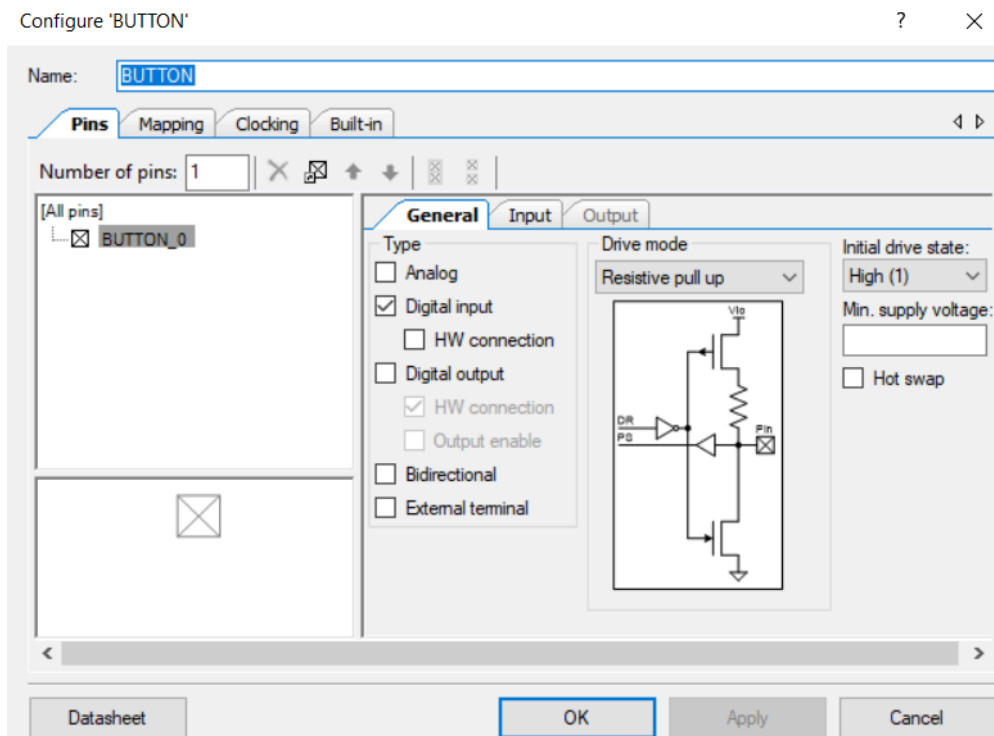


Figura 13-Ventana de configuración BUTTON

Lo significativo en el caso del botón es su configuración como resistive pull up y su estado inicial alto. Esto es porque el botón sin pulsar se considera activado y es en el momento de pulsar cuando su estado cambia a Low(0). La configuración es así y no la contraria con el objetivo que, durante la ejecución del programa no se deba mantener pulsado el botón, sino que pueda ser pulsado solo al final, en el momento de querer obtener el dato.

### 3.2.2-Advertiser

Aquél o aquellos dispositivos que funcionan como *Advertisers/Anunciadores* emiten tramas al medio inalámbrico, normalmente de manera broadcast, es decir, que se envían tramas con un cierto tiempo de vida limitado, usualmente por cierto número de saltos en la red, de manera indiscriminada. Se espera que lleguen al destino correcto y que sean descartadas en aquellos lugares que no sean objetivo de la comunicación.

Los anunciadores pueden contener sensores y transmitir los datos medidos en las tramas anunciadas, la funcionalidad de la red suele estar determinada por la información que estos dispositivos envían.

Para configurar cualquier dispositivo BLE se comienza por colocarlo en el esquemático, dentro del programa PSoC Creator. Se puede encontrar en la sección Cypress-> Communications -> Bluetooth Low Energy(BLE)[v3.66]



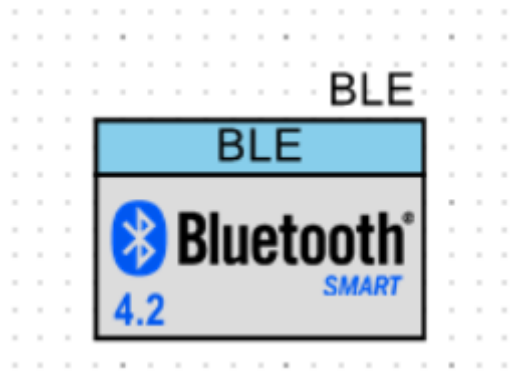


Figura 14-Dispositivo BLE colocado en el esquema

El siguiente paso, en el caso de un anunciador es indicarle este rol al dispositivo. Para ello se realiza un doble click en el icono mostrado y en la ventana emergente, dentro de la pestaña general, se selecciona la opción de broadcaster.

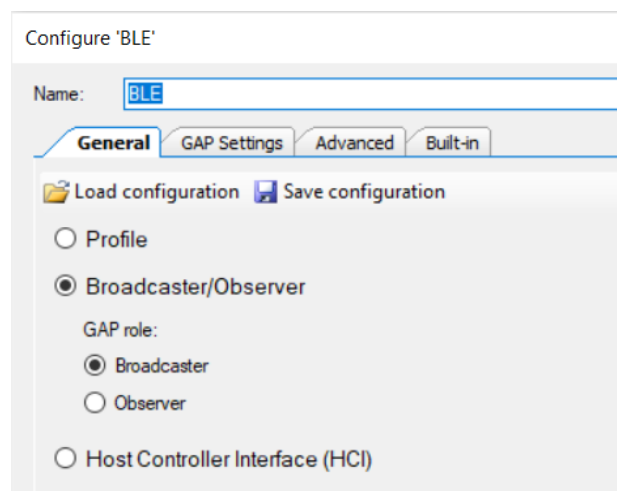


Figura 15-Pestaña general dispositivo BLE

Tras esto se le habrá indicado al dispositivo el rol que debe adoptar y finalizado el primer paso de la configuración.

Seguidamente se navega a la pestaña de GAP Settings. No es necesario modificar las opciones en la pestaña general, dentro de GAP Settings, pero se describirá brevemente su funcionalidad, por su interés para otro tipo de configuraciones.

Configure 'BLE'

Name: BLE

General GAP Settings Advanced Built-in

General

- Broadcaster role
  - Advertisement settings
    - Advertisement packet
    - Scan response packet

Restore Defaults

Device address

Public address (Company ID - Company assigned): 00A050-XXXXXX

☒ Silicon generated "Company assigned" part of device address

*You can use the user configuration section of the supervisory flash to store the public device address for mass production.*

Device name: PruebaBLE

Appearance: Unknown

Attribute MTU size (bytes): 23

Link layer max TX payload size (bytes): 27

Link layer max RX payload size (bytes): 27

Adv/Scan TX power level (dBm): 0

Connection TX power level (dBm): 0

Datasheet OK Apply

Figura 16-Pestaña GAP Settings->General dispositivo BLE

- *Device name*: Permite asignar un nombre personalizado al dispositivo configurado.
- *Appearance*: Permite personalizar la apariencia que tiene el dispositivo configurado cuando es escaneado por otros, esta apariencia, puede ser vista, por ejemplo, con la aplicación nRF Connect y algunas de las opciones son, por ejemplo, medidor de tensión sanguínea de brazo o muñeca, velocímetro o lector de código de barras.
- *Link layer max TX y RX*: Es el tamaño de trama máximo, max TX se refiere a tramas enviadas y RX a tramas recibidas.
- *Adv/Scan TX power level(dbM)*: Indica una potencia de amplificación medida en dBm, decibelios en referencia a milivatios para la transmisión o el escaneo de tramas.
- *Public address*: Es la dirección pública del dispositivo, compuesta por un identificador de la compañía fijo y otro, personalizable o, al tener el campo Silicon generated marcado, que tomará los valores asignados en fábrica por defecto.

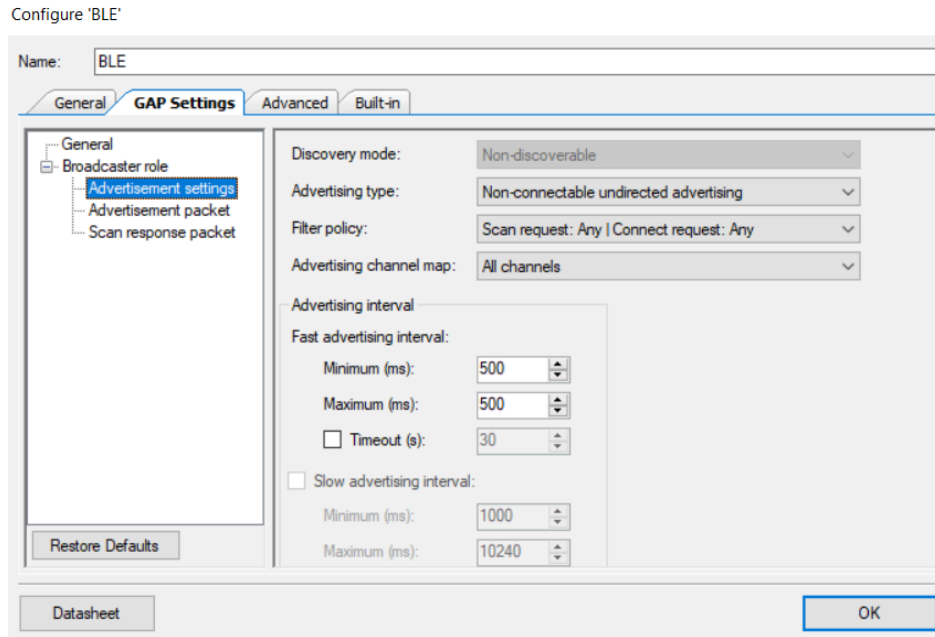


Figura 17-Pestaña GAP Settings->Advertising Settings en un anunciador

El segundo paso de la configuración es modificar ciertas opciones encontradas en esta pestaña.

- *Discovery mode*: Non-discoverable es una opción no modificable que previene que el dispositivo sea listado a través de una búsqueda de dispositivos bluetooth cercanos por dispositivos que no conozcan o puedan descubrir su dirección.
- *Advertising type*: Existen dos configuraciones, “Non-connectable undirected advertising” y “Scannable undirected advertising”. La primera de ellas no permite que otros dispositivos hagan peticiones de conexión con el anunciador configurado de esta manera y la segunda si lo hace.
- *Filter policy*: Esto es, la política de filtrado, en concreto, que tipo de peticiones se permiten y cuáles se desestiman directamente, cuando se reciben peticiones de escaneo o de conexión. Existen 4 opciones, que son la combinatoria entre cualquier dispositivo y aquellos pertenecientes a una lista de dispositivos especial, llamada la white list, que agrupa dispositivos reconocidos por la red, para los dos tipos de peticiones anteriormente mencionados. La White List se debe configurar manualmente si se hace uso de ella, pero no es recomendable, ya que supone problemas de escalabilidad, al tener que incluir cada dispositivo aceptable por la red en esta.
- *Advertising channel map*: Es particularmente interesante, ya que permite manualmente seleccionar y delimitar los canales donde se realizarán los anuncios de tramas, entre el 38, 39 y 37, los utilizados para esta funcionalidad. Para todas las configuraciones se ha seleccionado la opción “All channels”, es decir, que se emitirán anuncios en cada uno de estos tres canales, ya que esto minimiza la posibilidad de que ocurran pérdidas de tramas.

- *Advertising Interval*: Opción de configuración principal en la realización del estudio, se configura un tiempo de escaneo constante para todos los ensayos realizados, es decir, con valores mínimos y máximos en ms iguales, sin incluir timeout. El timeout es una mecánica por la cuál, si se detectan pérdidas de tramas, tras cierto tiempo, el dispositivo reduce el intervalo de advertising, lo cuál reduce la velocidad a la que se realiza la transmisión y proporciona tiempo a la red para recuperarse, de manera similar a como funciona la ventana de recepción en el protocolo TCP. En los campos Minimum(ms) y Maximum(ms) del advertising interval se podría personalizar el nuevo intervalo de advertising, mayor que al anterior, tras haberse producido un timeout.

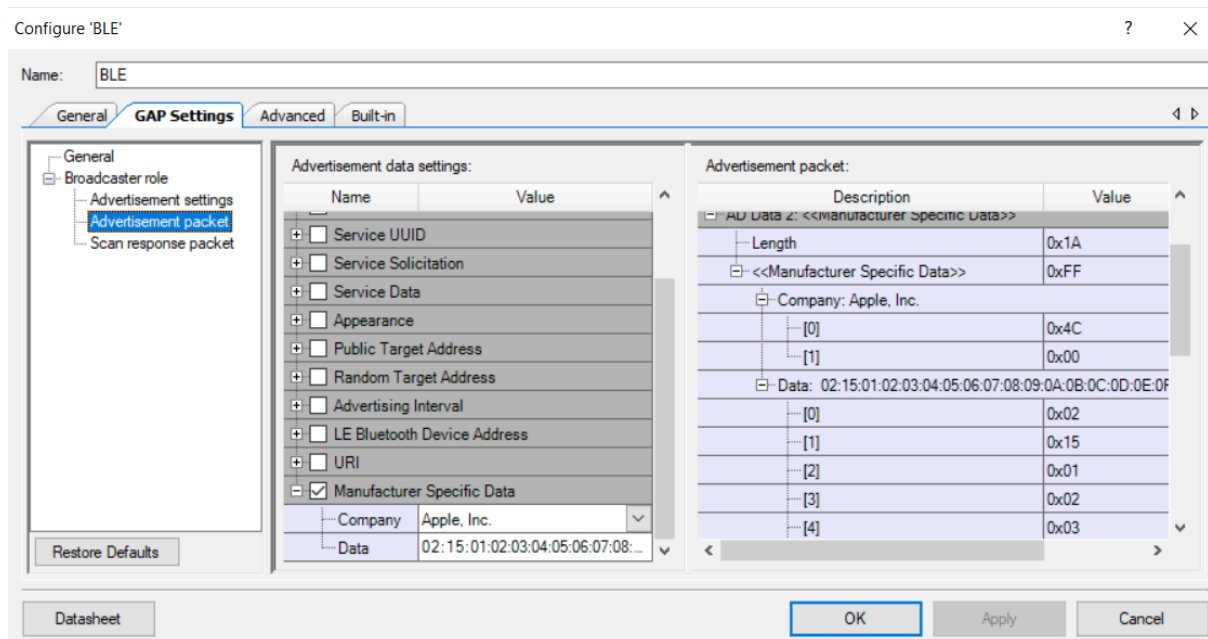


Figura 18- Pestaña GAP Settings-> Advertisement packet del anunciador

Tras haber modificado las opciones anteriores de manera pertinente, se debe indicar el contenido de la trama enviada por el anunciador. Esto se realizará en la ventana observada en la figura superior a este párrafo, el Advertisement packet o paquete anunciado.

Dentro del campo de compañía se debe seleccionar Apple, Inc de entre las diversas opciones y luego escribir una trama compatible con la estructura seguida por los iBeacons. Es decir, que Data debe comenzar por el campo 02:15 en hexadecimal y con un tamaño no superior al máximo de bytes que sea posible emitir. En este caso, el anunciador está configurado para enviar, lo considerado en el código utilizado como la TRAMA\_X.

Así concluye el tercer paso y el último en cuanto a la configuración del anunciador/es en la pestaña, “TopDesing”.

Una vez que se compila el código escrito en la función main, para terminar la configuración del dispositivo se debe realizar una asignación de puertos de los elementos en el esquemático a entidades de la realidad.

Así pues se debe abrir la ventana Design Wide Resources-> Pins, la cuál mostrará lo siguiente:



Figura 19- Asignación de pines Advertiser

En el caso del anunciador solo se deben asignar puertos al led de control y a la UART. En cuanto al led de control se asigna al puerto P3[3] del BLE, ya que este conecta con el led encontrado en el dispositivo. RX\_1 y TX\_1 son puertos asociados a la UART, RX para la recepción de datos y TX para la transmisión.

Para conocer los puertos correctos a donde conectar cada uno de los elementos del esquemático se utiliza el esquemático proporcionado por la empresa Infineon para los dongles utilizados, que se encuentra entre los enlaces indicados en la última sección del trabajo. [\[13\]](#)

El esquemático final del proyecto del anunciador es el siguiente:

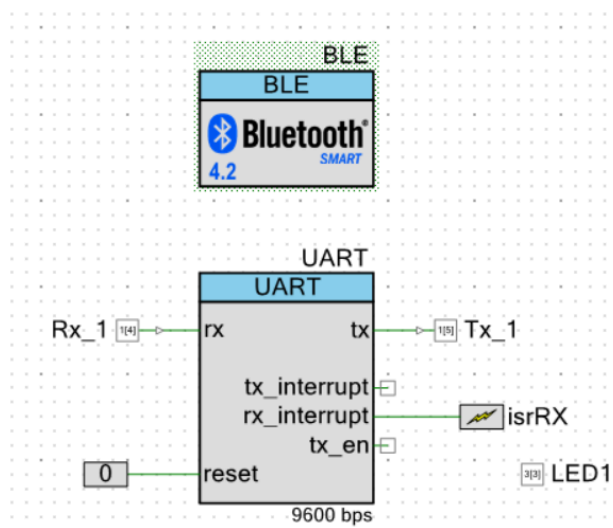


Figura 20-Esquemático Advertiser

### 3.2.3-Scanner

El escáner es un dispositivo que rastrea tramas de dispositivos que las anuncien, durante el tiempo de la ventana de escaneo. Es indispensable en el funcionamiento de la red, porque sin este componente, en una red de anunciadores, no se podría obtener la información transmitida por estos, la red perdería su funcionalidad.

La configuración es similar a la de los anunciadores, siendo el primer paso colocar un dispositivo BLE en el esquemático, de forma análoga a cuando se hizo con el anunciador. Los cambios con el caso anterior surgen dentro de la configuración.

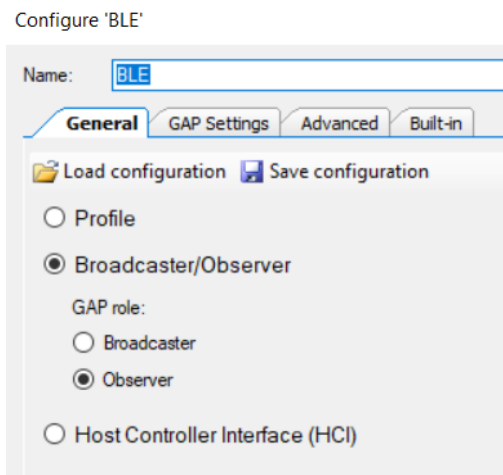


Figura 21-Ventana General de configuración del escáner

En la ventana general se debe seleccionar el rol, Observer, observador como único cambio y la ventana General, dentro de la sección GAP Settings es idéntica a la de los anunciadores.

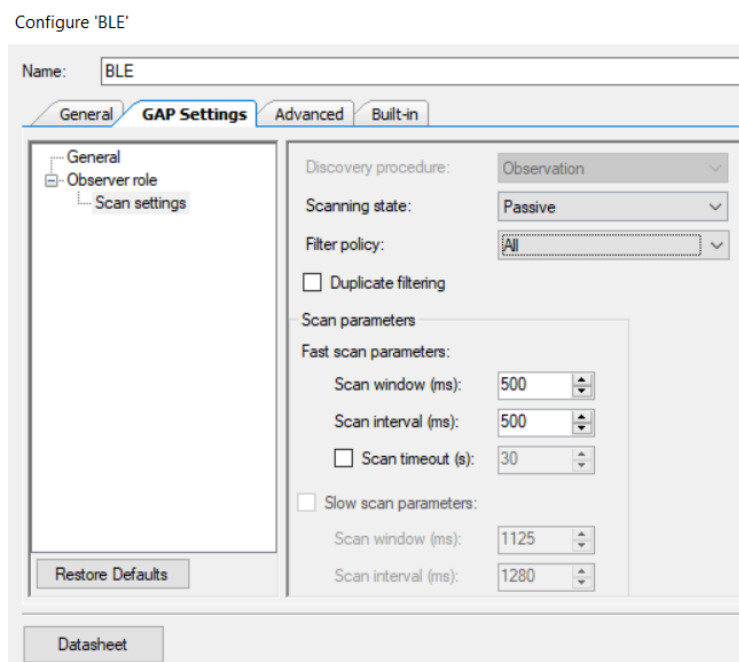


Figura 22-GAP Scan Settings

Sin embargo GAP Settings-> Scan settings si contiene parámetros de configuración con mayor interés.

Scanning state: Se configura si se quiere que el dispositivo realice un escaneo pasivo o activo. El escaneo activo implica una comunicación dirigida con los dispositivos escaneados, se utiliza para recabar información detallada o realizar pruebas sobre los dispositivos escaneados. El escaneo pasivo se realiza con el único objetivo de recibir las tramas recibidas,

no es de interés generar un feedback en respuesta a la información recibida, como en el escaneo activo.

Filter policy: Es la política de filtrado seguida por el escáner, en este caso solo existen dos opciones, aceptar todos los dispositivos como objetivos del escaneo o solo los encontrados en la white list. Se selecciona la primera opción, ya que no se ha creado esta lista para los experimentos efectuados.

En la sección “Scan parameters” se encuentran los parámetros de configuración de los que depende gran parte del estudio realizado, la ventana de escaneo y el intervalo de escaneo. El tamaño de la ventana de escaneo frente al intervalo de escaneo es la principal modificación para una cantidad considerable de las medidas efectuadas.

De manera similar al caso de los anunciadores, se puede escoger un tiempo de timeout, tras el cual, si se han detectado errores en la recepción de tramas, el sistema utilizará una ventana mayor, con el objetivo de poder escanear con más fiabilidad. El objetivo de esto es minimizar los errores que podrían producirse debido a un tamaño de ventana insuficiente. No se ha utilizado en los experimentos realizados puesto que no se ha detectado una mejora del rendimiento para valores de ventana mayores y un mismo advertising interval.

Tras realizar las modificaciones pertinentes y compilar el programa principal del escáner se debe realizar la asignación de pines, con la única diferencia frente al caso anterior en que se deberá asignar el puerto correcto al botón de control.

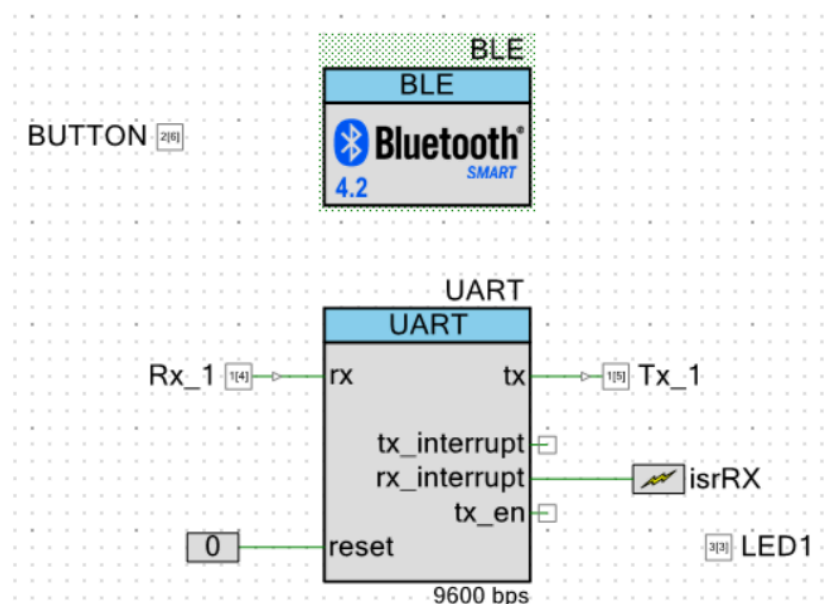


Figura 23- Esquemático final Escáner

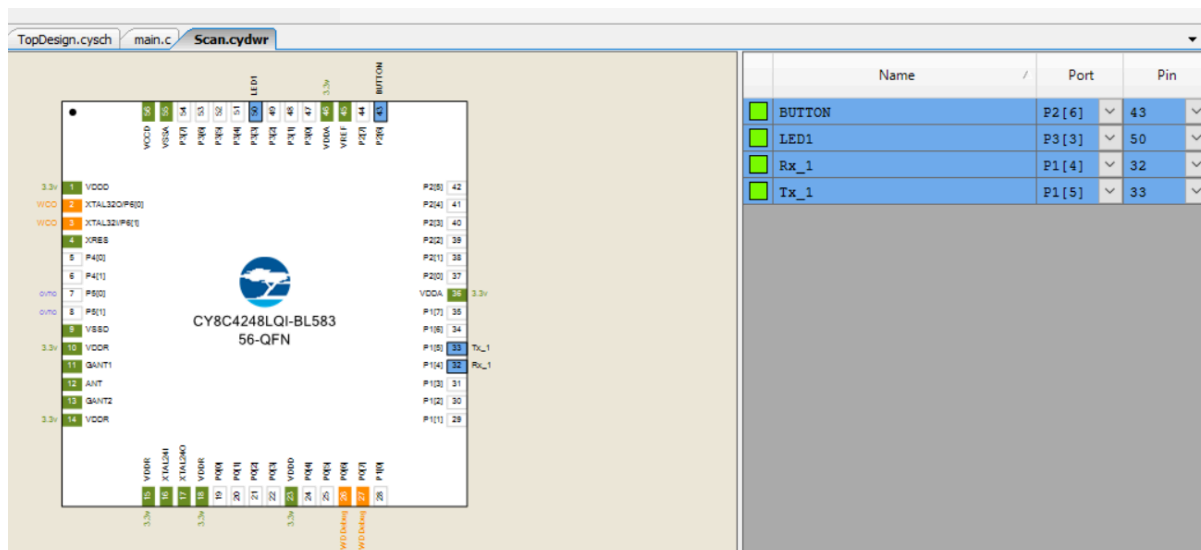


Figura 24-Asignación de pines Escáner

De nuevo, para poder llevar a cabo este proceso se necesita consultar el esquemático proporcionado por la empresa Infineon.

### 3.2.4-ReTransmisor

Un retransmisor es un dispositivo capaz de transmitir y escanear tramas. Su función en el proyecto es escanear una trama concreta, anunciada por un dispositivo y mandar una trama al siguiente retransmisor o el escáner cuando esto suceda.

La configuración empieza colocando un dispositivo BLE en el esquemático, de la misma forma que en casos anteriores. Posteriormente se clic en el dispositivo para iniciar la configuración, que si implica particularidades frente a los casos anteriores.

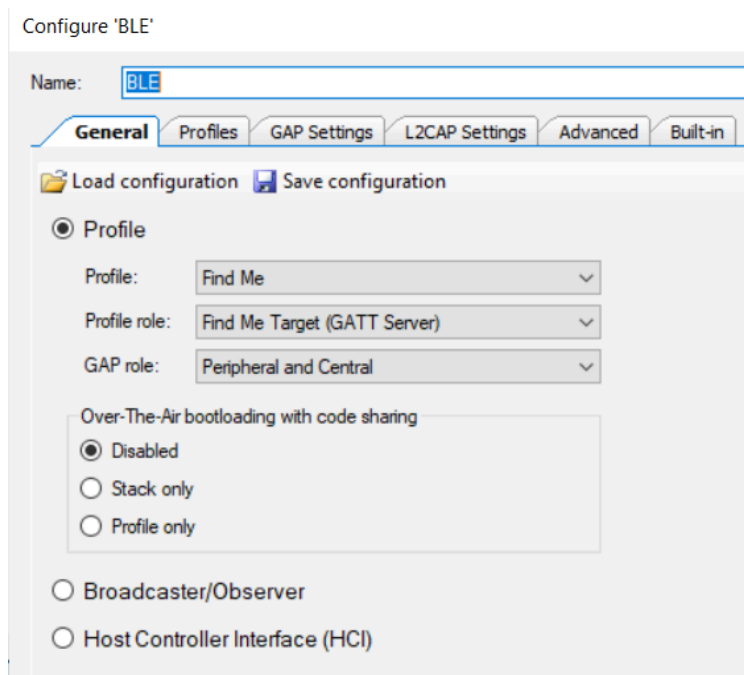


Figura 25-Ventana general configuración retransmisor



En este caso no se debe seleccionar las opciones de broadcaster o observer, ya que el mismo dispositivo debe ser capaz de escanear y anunciar tramas, sino la opción perfil. Dentro de estos perfiles existen numerosas opciones, sin embargo, para los objetivos del estudio se obtiene el mismo resultado seleccionando cualquiera de ellas, siempre que se pueda seleccionar un GAP Role mixto, “Peripheral and Central”.

La razón por la que el rol del perfil debe ser mixto es porque esto es lo que permite que el dispositivo pueda realizar escaneos y anuncios de tramas dentro de su código programable.

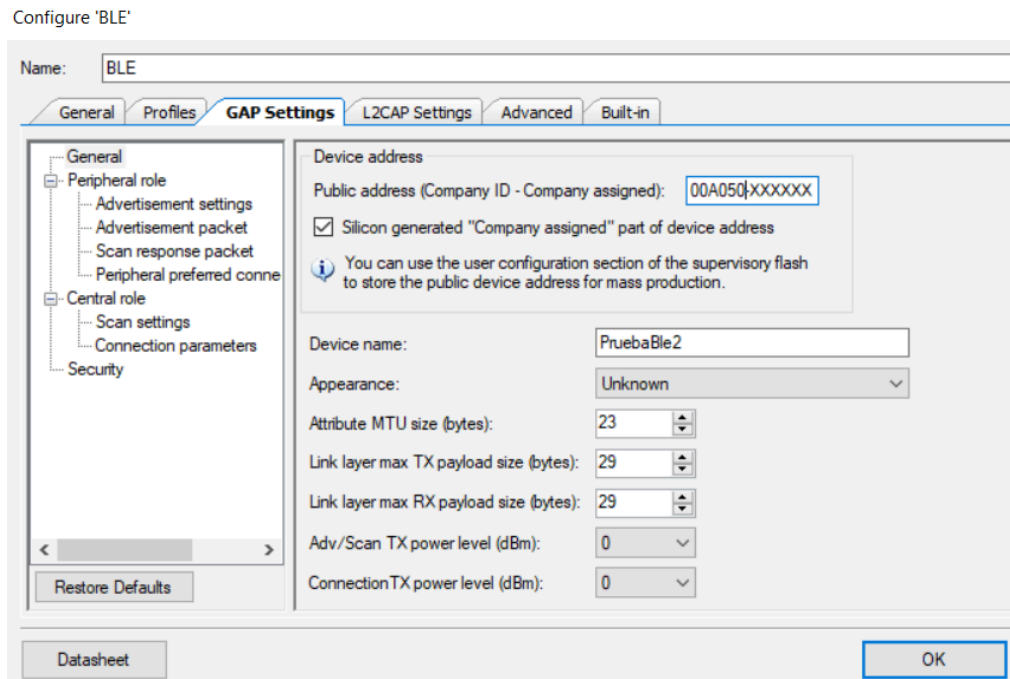


Figura 26-Ventana de configuración GAP Settings-> General Retransmisor

En el caso de retransmisores, en la pestaña de GAP Settings aparecerán opciones muy similares a los casos anteriores combinadas. La configuración de la trama transmitida, se realizará dentro de la sección, Advertisement packet, igual que para el anunciador, a su vez, la de los parámetros de escaneo se hará en la sección Scan settings.

En cuanto la configuración del intervalo de advertising, se realizará en Advertisement settings. En el estudio se mantiene constante, en un valor bajo, porque interesa que las tramas se retransmitan de manera rápida. Se muestra una imagen de esto último a continuación.

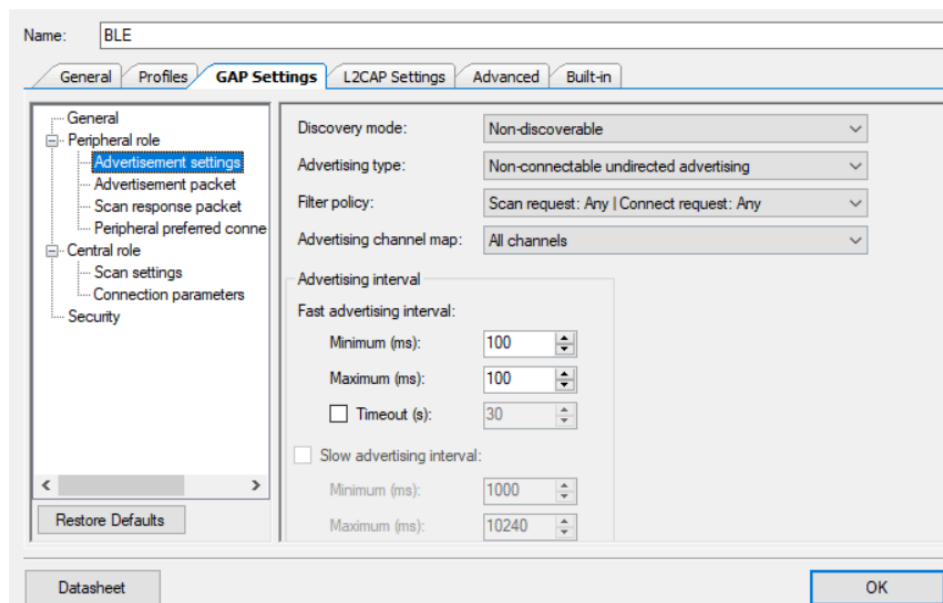


Figura 27-Ventana de configuración Advertisement settings en un retransmisor

Por otro lado, respecto a la ventana de escaneo y el intervalo de escaneo de los retransmisores, se realizará en la sección, Scan settings. La modificación de estas dos variables y los efectos sobre las tramas recibidas en el escáner final ha sido objetivo principal del estudio realizado, por ello, también se incluye una captura de pantalla de esta pestaña a continuación.

Configure 'BLE'

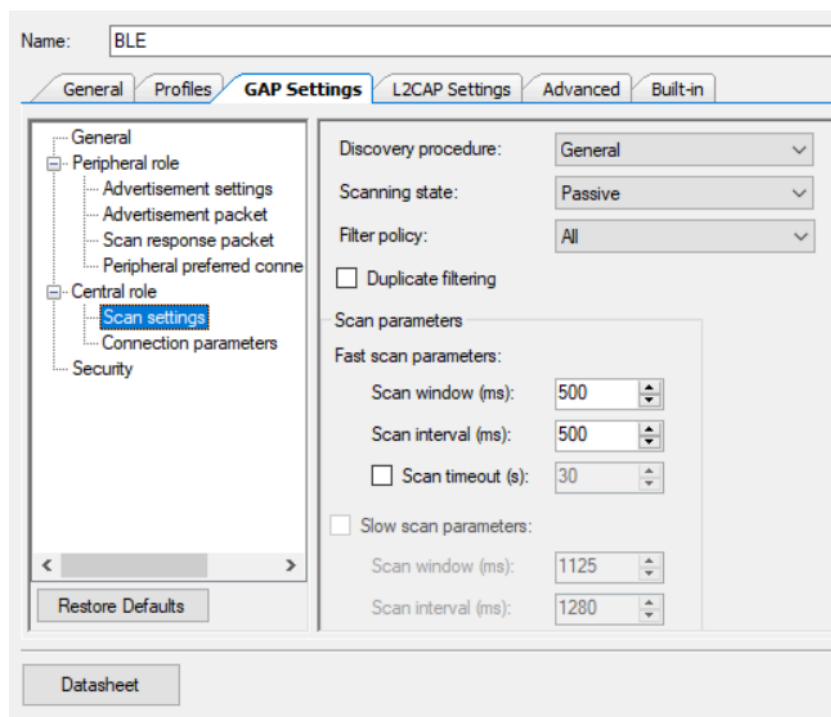


Figura 28-Ventana de configuración Scan settings en un retransmisor

Existe una particularidad frente a la misma pestaña cuando el dispositivo se configura como observer. El Discovery procedure cambia a general, lo cual permite que el dispositivo, funcionando como periférico, pueda ser descubierto por un dispositivo funcionando como

central. También existe la opción Limited para este campo, que solo permite que el dispositivo sea descubierto cuando el dispositivo central también sigue el modo Limited, lo cual implica ciertas restricciones. En los casos de estudio se mantiene siempre como General.

El diseño final posee la misma estructura visual y la asignación de pines que la realizada en el caso del scanner, a pesar de que en los ensayos finales el botón de control no se ha utilizado, pero si como medida de control durante el desarrollo de los retransmisores.

### 3.3-Explicación del Código

Con el objetivo de que el estudio pueda ser replicado y ampliado por otras personas en las que haya suscitado curiosidad o un interés académico se incluye a continuación una explicación breve del código utilizado para cada uno de los tres arquetipos de programa para los dispositivos.

El código está escrito en C, ya que este lenguaje de programación permite un control preciso de la memoria utilizada por el programa, hecho que resulta sustancial cuando se trabaja con dispositivos como los utilizados, con baja capacidad de memoria.

Esto es así por la utilización de punteros cuando se trabaja con datos de tamaño considerable, lo cual implica que estos datos no se copian cada vez que son utilizados por funciones, sino que se opera con ellos a partir de las posiciones de memoria que ocupan.

El PSoC Creator tiene integrado un editor y un compilador de C que incluye gran cantidad de funciones y definiciones utilizadas para la operación de los aparatos empleados no incluidas en C básico, por lo que utilizar este software es especialmente conveniente si se trabaja con los dispositivos utilizados.

#### 3.3.1-Advertiser

Todos los programas creados comienzan con 3 “include”, el primero incluye el ámbito del proyecto y es necesario para trabajar en conjunto con las otras partes del proyecto. Los dos “include” siguientes contienen funciones que permiten mostrar a través de la UART fácilmente ciertas variables.

En el caso del anunciador se comienza declarando tres variables globales, un contador que controla el envío de tramas, un vector de caracteres utilizado para mostrar información por la UART y una variable definida dentro de las librerías de PSoC Creator, utilizada para el control de la función CyBle\_GappSetNumOfAdvPkts.

```
#include "project.h"
#include <stdio.h>
#include <stdlib.h>

uint16_t contador=0; //Contador utilizado para controlar cuando el dispositivo detiene el envio de tramas
char PrintBuffer[255]={0}; //String utilizado para mostrar información por la UART
extern CYBLE_GAPP_DISC_MODE_INFO_T cyBle_discoveryModeInfo; //Variable definida externamente utilizada como dato
//en la orden CyBle_GappSetNumOfAdvPkts
```

Figura 29-Advertiser Código 1

Seguidamente aparece la función void AppCallback que gestiona los eventos sucedidos y la respuesta ante estos.

En concreto, cuando se enciende el dispositivo se activa el evento CYBLE\_EVT\_STACK\_ON. En el caso del anunciador, cuando esto sucede se determina una

cantidad fija de paquetes que se van a enviar, se realiza una espera de 5 segundos, para poder reiniciar el escaner y los retransmisores y se comienza la transmisión de tramas. Lo descrito se muestra en la imagen a continuación.

```
void AppCallBack(uint32 event, void *eventParam)
{
    (void) eventParam;

    switch (event)
    {
        //Evento que sucede al iniciar el dispositivo
        case CYBLE_EVT_STACK_ON:

            //Se fija una cantidad total de 600 paquetes (200 paquetes en cada uno de los canales)
            CyBle_GappSetNumOfAdvPkts(&cyBle_discoveryModeInfo, 600);
            //Espera de 5s antes de iniciar a transmitir
            CyDelay(5000);
            //Encender led para indicar que se comienza a transmitir tramas
            LED1_Write(1);
            //Comenzar a enviar tramas de anuncio
            CyBle_GappStartAdvertisement(CYBLE_ADVERTISING_FAST);

            break;
```

Figura 30-Advertiser Código 2

El siguiente evento es CYBLE\_EVT\_GAPP\_ADVERTISEMENT\_START\_STOP. Sucede cuando el advertising comienza o cesa, es decir, se ejecuta el código incluido en esta sección una vez al inicio del programa y una segunda vez cuando la función que limita el número de paquetes mandados toma efecto. El resultado es que el contador dentro de esta sección tomará el valor 2 cuando esto último suceda.

Una vez que esto pase se quiere detener definitivamente el anuncio de tramas, por lo que se añade la condición de que si el contador toma el valor 2, se ejecute la función que detiene el advertising y además se apague el diodo de control, el cual indica la finalización de la transmisión. En la siguiente imagen se puede observar lo expuesto.

```
case CYBLE_EVT_GAPP_ADVERTISEMENT_START_STOP:

    contador++;

    //Sección que sirve para visualizar el contador, tendrá 2, como valor al finalizar el programa

    //snprintf(PrintBuffer, 255, "%u", contador);
    //UART_PutString(PrintBuffer);
    //UART_PutString("\r\n");

    //Fin sección de visualización del contador

    //Cuando el contador toma el valor 2 el dispositivo cesa de enviar tramas de anuncio y apaga el led de control
    if(contador==2){
        //Detener anuncios
        CyBle_GappStopAdvertisement();
        //Apagar led de control
        LED1_Write(0);
    }

    break;
```

Figura 31-Advertiser Código 3

La función de callback termina con el caso de default, donde no se toman acciones en respuesta.

Seguidamente comienza la función main en la que se permiten interrupciones globales, se inicia el dispositivo BLE y la UART y finalmente el dispositivo entra en un bucle infinito donde procesa los eventos sucedidos con una espera corta de 50 ms, en otras palabras, el dispositivo ejecuta las acciones descritas en la función de callback según los eventos que sucedan. La figura incluida a continuación muestra lo presentado anteriormente.

```

    default:
        break;
    }
}

int main(void)
{
    CyGlobalIntEnable; /* Enable global interrupts. */
    CyBle_Start(AppCallback); // Iniciamos el dispositivo a través de la función de callback
    UART_Start(); // Iniciamos la UART para poder visualizar datos

    for(;;)
    {
        //El dispositivo procesa las órdenes enviadas
        CyBle_ProcessEvents();
        //Espera de un tiempo pequeño debe ser la mitad del advertising interval o menos
        CyDelay(50);
    }
}

/*FINAL*/

```

Figura 32-Advertiser Código 4

### 3.3.2-Scanner

El programa del escáner inicia con una declaración de variables distinta a la del anunciador. En este caso se declara otro contador, pero se usará para contabilizar las tramas recibidas. El databaseacon declarado tiene como objetivo poder mostrar por la UART ciertas variables.

Después de esto se declara como constante una variable que contiene la longitud de la trama y otras tres, con el contenido de las tres tramas distintas usadas a lo largo del proyecto. (Nota: en la imagen no aparecen completas porque la declaración tiene gran extensión)

```

uint16_t contador=0; //Variable que se utiliza para contar las tramas recibidas
char databaseacon[255]={0}; //Variable que se utiliza para copiar la trama recibida y poder mostrarla por pantalla

//Variables conceptuales, juntas componen la trama, el encabezado fijo + la UUID
//const uint8_t UUID_BEACON[16] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10};
//const uint8_t HEADER_BEACON[9] = {0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15};

const uint8_t BEACON_LENHT=30; //Longitud de la trama 25+5 Bytes no pertenecientes a la UUID

//TRAMA_X, TRAMA_Y, TRAMA_Z Se definen las tres siempre, por simpleza en los cambios, son las tramas recibidas y enviada.
const uint8_t HEADER_AND_UUID_BEACONX[25]={0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10};
const uint8_t HEADER_AND_UUID_BEACONY[25]={0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15, 0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E, 0x2F, 0x30};
const uint8_t HEADER_AND_UUID_BEACONZ[25]={0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15, 0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E, 0x3F, 0x40};

```

Figura 33-Scanner Código 1

A continuación comienza la función void AppCallback del escáner. Antes del inicio del dispositivo se declara la variable puntero llamada advReport y se inicia a NULL. Esta variable es de un tipo definido internamente en las librerías del programa PSoC Creator, CYBLE\_GAPC\_ADV\_REPORT\_T este tipo es una estructura de datos que agrupa información obtenida en el escaneo de dispositivos. Cuando el dispositivo se inicia, como respuesta comienza el escaneo, pero no se ejecuta ninguna función adicional.

```

void AppCallBack(uint32 event, void *eventParam)
{
    (void) eventParam;

    switch (event)
    {

        //Puntero del tipo CYBLE_GAPC_ADV_REPORT_T llamado advReport iniciado a NULL
        CYBLE_GAPC_ADV_REPORT_T *advReport = NULL;

        //Evento que sucede al iniciar el dispositivo
        case CYBLE_EVT_STACK_ON:
            //Comenzar a escanear
            CyBle_GapcStartScan(CYBLE_SCANNING_FAST);
            break;
    }
}

```

Figura 34-Scanner Código 2

CYBLE\_EVT\_GAPC\_SCAN\_PROGRESS\_RESULT es el siguiente evento para el que se debe programar una respuesta. Este evento se produce cuando el escáner detecta una trama de otro dispositivo.

La meta principal cuando se detecta una trama es ser capaz de detectar rápidamente si no es de interés para optimizar la eficiencia del programa. Por ello es que primero se utiliza un condicional que termina el evento si las tramas detectadas no tienen la longitud de un iBeacon y en segundo lugar se comprueba si la trama recibida es la que se espera que transmita el anunciador o retransmisor pertinente y si no lo es, el programa termina el evento. Finalmente si estas condiciones no se han cumplido y por tanto, la trama es una de las que se querían detectar se aumenta el contador que contabiliza las tramas recibidas.

```

/*Cada vez que se escanea un dispositivo sucede este evento, donde se puede operar con los resultados*/
case CYBLE_EVT_GAPC_SCAN_PROGRESS_RESULT:

    //Cunto más tiempo se pierda, menor será la eficiencia, por ello se intenta detectar que no es un ibeacon

    //Variable que se utiliza para poder acceder a datos obtenidos en el escaneo
    advReport = (CYBLE_GAPC_ADV_REPORT_T*)eventParam;

    SECCIÓN 1: DEBE COMENTARSE PARA VER TODAS LAS TRAMAS QUE LLEGAN

    /*Si las tramas escaneadas no son de la longitud correcta se termina el evento*/
    if (advReport->dataLen!=BEACON_LENGTH){
        break;
    }

    /*Se comprueba si la trama es de nuestra red y si no lo es, se termina el evento*/
    if (memcmp (&HEADER_AND_UUID_BEACON[0], &advReport->data[0], 25)!=0){
        break;
    }

    // FIN SECCIÓN 1

    //Contador utilizado para contabilizar las tramas recibidas, solo aumenta cuando la trama recibida está
    //dentro de las buscadas
    contador++;
}

```

Figura 35-Scanner Código 3

De forma análoga al caso de programación del anunciador, no se programa una respuesta para el evento default. En este caso, tampoco se programa una respuesta para el evento producido al iniciar o detener el escaneo. También existe una sección dedicada a la visualización en la UART de las tramas recibidas pero no se ejecuta en los experimentos por el decrecimiento en la eficiencia del programa.

```

        //Evento que sucede al iniciarse o detenerse el escaneo
        case CYBLE_EVT_GAPC_SCAN_START_STOP :
            break;

        default:
            break;
    }
}

```

Figura 36-Scanner Código 4

Finalmente se programa el cuerpo de la función main. Se comienza declarando un array de caracteres utilizado para poder enseñar la información contenida en el contador a través de la UART. Luego se permiten interrupciones globales, se inicia el dispositivo BLE Dongle, se inicia la UART y se escribe por la UART “Programa Iniciado Scanner”, esto último se hace para asegurarse de que en la experimentación se está conectando con el dispositivo correcto.

A partir de entonces el programa entra en un bucle infinito donde se procesan los eventos y del cuál puede salir al pulsarse el botón, imprimir por pantalla, a través de la UART el valor del contador, es decir, de las tramas recibidas y finalizar el programa tras ello.

```

int main(void)
{
    char PrintBuffer[255]={0}; //Variable que utilizamos para que la UART pueda mostrar el valor del contador
    CyGlobalIntEnable; /* Enable global interrupts. */

    CyBle_Start(AppCallBack); //Se inicia el dispositivo a través de la función de callback, que gestiona los eventos
    UART_Start(); //Se inicia la UART, para poder mostrar datos por pantalla
    UART_PutString("Programa Iniciado Scanner\r\n");

    for(;;)
    {
        CyBle_ProcessEvents(); // Se procesan los eventos
        // Si se pulsa el botón de usuario se muestra a través de la UART el valor del contador y termina el programa
        if(BUTTON_Read()==0) {
            //Lineas necesarias para mostrar el valor del contador
            snprintf(PrintBuffer, 255, "%u", contador);
            UART_PutString(PrintBuffer);

            //Terminar el programa
            return(0);
        }
    }
}
/*FINAL*/

```

Figura 37-Scanner Código 5

### 3.3.3-Retransmisor

La programación del retransmisor comienza con una declaración de variables, la mayoría idénticas a las del escáner, siendo una mención destacable que el contador, en este caso no se utiliza para contabilizar las tramas recibidas sino para detener el anuncio de tramas y la variable CYBLE\_GAPC\_DISC\_INFO\_T scaninfo. Esta última es utilizada por la función que limita los paquetes enviados, como en el caso del anunciador, pero también para la configuración del escaneo de tramas.



```

5 uint16_t contador=0; //Variable que se utiliza para controlar cuando el dispositivo deja de anunciar información
7 char databeacon[255]={0}; //Variable que se utiliza para copiar la trama recibida y poder mostrarla por pantalla
3
3 //Variables conceptuales, juntas componen la trama, el encabezado fijo + la UUID
3 //const uint8_t UUID_BEACON[16] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0
3 //const uint8_t HEADER_BEACON[9] = {0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15};
3
3 const uint8_t BEACON_LENGTH=30; //Longitud de la trama 25+5 bytes no pertenecientes a la UUID
3
3 //TRAMA_X, TRAMA_Y, TRAMA_Z Se definen las tres siempre, por simpleza en los cambios, son las tramas recibidas y enviada
5 const uint8_t HEADER_AND_UUID_BEACONX[25]={0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15, 0x01, 0x02, 0x03, 0x04,
7 const uint8_t HEADER_AND_UUID_BEACONY[25]={0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15, 0x20, 0x02, 0x03, 0x04,
3 const uint8_t HEADER_AND_UUID_BEACONZ[25]={0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15, 0x30, 0x02, 0x03, 0x04,
3 CYBLE_GAPC_DISC_INFO_T scaninfo;

```

Figura 38-Retransmisor Código 1

Seguidamente comienza la declaración de la función void AppCallback. Se declara el puntero llamado advReport de nuevo, con la misma funcionalidad y estructura al inicio.

Posteriormente se indica al dispositivo que debe comenzar el descubrimiento de tramas con la función CyBle\_GapcStartDiscovery(&scaninfo) al iniciarse el dispositivo. Esta función es distinta a la utilizada para el escaneo de tramas cuando se configura el dispositivo como observador. Es necesario hacerlo de esta forma cuando se utilizan perfiles, de otra manera, no se reciben las tramas buscadas. Esta función toma como dato &scaninfo, variable que se configura dentro del main con los parámetros deseados para la realización de este descubrimiento de tramas.

```

void AppCallback(uint32 event, void *eventParam)
{
    (void) eventParam;

    switch (event)
    {
        //Puntero del tipo CYBLE_GAPC_ADV_REPORT_T llamado advReport iniciado a NULL
        CYBLE_GAPC_ADV_REPORT_T *advReport = NULL;

        //Evento que sucede al iniciar el dispositivo
        case CYBLE_EVT_STACK_ON:
            //Comenzar a escanear
            CyBle_GapcStartDiscovery(&scaninfo);
            break;
    }
}

```

Figura 39-Retransmisor Código 2

El siguiente evento a programar es el ocurrido cuando se detecta una trama. El comienzo es similar al del escáner básico pero en lugar de aumentarse el contador cuando se detecta una trama esperada se detiene el escaneo, se realiza una espera mientras se hace efectiva esta detención del escaneo, se establece el límite de transmisión de tramas a 3 paquetes (Se recibe 1 trama porque se emite 1 de estos paquetes por cada uno de los canales de anuncio), se comienza la transmisión de tramas y se enciende el diodo de control cuando se hacen medidas del retardo de transmisión. En la imagen siguiente se muestra este último proceso.



```

// FIN SECCIÓN 1

//Se detiene el escaneo de tramas si se detecta una de las buscadas
CyBle_GapcStopDiscovery();
//Espera a terminar de escanear
while(CyBle_GetState()==CYBLE_STATE_SCANNING);
//Se configura para enviar un solo paquete (3, uno en cada canal)
CyBle_GappSetNumOfAdvPkts(&cyBle_discoveryModeInfo,3);
//Se inicia el anuncio del paquete hacia el siguiente dispositivo

CyBle_GappStartAdvertisement(CYBLE_ADVERTISING_FAST);
//LED1_Write(1);Para visualizar el retardo

```

Figura 40-Retransmisor Código 3

Sin embargo, todavía no se ha configurado cuando se detiene el envío de tramas y comienza de nuevo el escaneo de tramas. CYBLE\_EVT\_GAPP\_ADVERTISEMENT\_START\_STOP es el evento que se utiliza para este propósito. La primera vez que se comienza a anunciar tramas se aumenta el valor del contador a 1, seguidamente, cuando la función que limita la transmisión de paquetes toma efecto, el valor del contador incrementa a 2 y se cumple la condición escrita en esta sección del programa. Tras cumplirse la condición se detiene el anuncio de tramas, se realiza una espera hasta que esta detención tiene efecto y seguidamente se comienza a escanear tramas de nuevo. Cuando se mide el retardo se apaga el led de control tras ejecutarse la parada en el anuncio de tramas.

```

//Evento que sucede al iniciarse o detenerse el anuncio de tramas
case CYBLE_EVT_GAPP_ADVERTISEMENT_START_STOP :
    //Aumenta el contador en 1
    contador++;
    //Cuando el contador es 2 se detiene el anuncio de tramas, se reinicia el contador
    //y se comienza a escanear de nuevo
    if(contador==2){
        CyBle_GappStopAdvertisement();
        //LED1_Write(0); Para visualizar el retardo
        while(CyBle_GetState()==CYBLE_STATE_ADVERTISING);
        contador=0;
        CyBle_GapcStartDiscovery(&scaninfo);
    }
    break;

```

Figura 41-Retransmisor Código 4

Después de esto se indica al dispositivo que no realice acciones durante el estado por defecto y comienza la función main.

La función empieza permitiendo las interrupciones globales y con la configuración de la variable scaninfo. Entre los parámetros configurados merecen especial atención scanItv y scanWindow, estos parámetros indican a los retransmisores la duración del intervalo de escaneo y el tamaño de ventana de escaneo que utilizarán los dispositivos cuando funcionen como observadores. Para los diferentes escenarios con retransmisores era objetivo principal del estudio estudiar como las variaciones en estos parámetros afectarían a las tramas recibidas. Es importante mencionar que se modifican en esta sección del código, no dentro de la configuración del diseño para que tomen efecto en los experimentos realizados.

Seguidamente se inicia el dispositivo, la UART y se escribe a través de la UART “Programa Iniciado Re-Transmisor” y finalmente el programa entra en un bucle infinito donde se procesan los eventos ocurridos.

```

        default:
            break;
    }
}

int main(void)
{
    CyGlobalIntEnable; /* Enable global interrupts. */
    //Configuración de scaninfo con los parámetros deseados, para poder detectar los dispositivos anunciantes
    scaninfo.discProcedure=CYBLE_GAPC_OBSER_PROCEDURE;//Procedimiento de escaneo, como observador
    scaninfo.scanType=CYBLE_GAPC_PASSIVE_SCANNING;//Tipo de escaneo, pasivo
    scaninfo.scanIntv=1000;//Intervalo de escaneo
    scaninfo.scanWindow=500;//Ventana de escaneo
    scaninfo.ownAddrType=CYBLE_GAP_ADDR_TYPE_PUBLIC;//Tipo de dirección propia, pública
    scaninfo.scanFilterPolicy=CYBLE_GAPC_ADV_ACCEPT_ALL_PKT;//Politica de filtrado, aceptar todo
    scaninfo.scanTo=0;
    scaninfo.filterDuplicates=CYBLE_GAPC_FILTER_DUP_DISABLE;//Deshabilitar el filtrado de duplicados

    CyBle_Start(AppCallback);//Se inicia el dispositivo a través de la función de callback, que gestiona los eventos
    UART_Start();//Se inicia la UART para poder mostrar contenido en pantalla
    UART_PutString("Programa Iniciado Re-transmisor1\r\n");//Se escribe este mensaje en la UART, al iniciar el programa

    for(;;)
    {
        CyBle_ProcessEvents();// Se procesan los eventos
    }
}
/*FINAL*/

```

Figura 42-Retransmisor Código 5

# Capítulo 4

## Desarrollo del Estudio

### 4.1-Escenarios de Estudio

El estudio pretende mostrar las diferencias en la proporción de tramas recibidas para diferentes escenarios y con distintos parámetros para el advertising interval, scanning interval y scanning window. [\[14\]](#)

En los casos de estudio que se describen a continuación, existen tres tramas distintas, todas siguiendo el formato iBeacon. Estas tramas son fijas, determinadas por el usuario que programa los dispositivos para las redes descritas, aunque como se explicó anteriormente, suelen contener datos medibles por los dispositivos BLE a través de sensores, como por ejemplo la temperatura, codificados dentro, además de los identificadores pertinentes al dispositivo en concreto que transmite.

Para los casos de interés en este proyecto son las siguientes:

TRAMA\_X:

Cabecera: 0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15 (Fija iBeacon)

UUID:0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10

TRAMA\_Y:

Cabecera: 0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15 (Fija iBeacon)

UUID:0x20, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x20, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x20

TRAMA\_Z:

Cabecera: 0x02, 0x01, 0x04, 0x1A, 0xFF, 0x4C, 0x00, 0x02, 0x15 (Fija iBeacon)

UUID:0x30, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x30, 0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x30

Se han introducido tres cambios, en diferentes partes de la trama, para poder asegurar que no hay error en la comparación de las tramas.

En concreto, se van a estudiar los siguientes escenarios principales, respecto a los conceptos anteriormente mencionados.

#### 4.1.1-Escenarios en función de los parámetros básicos

Para un intervalo de advertising de 100 ms, se considera un scan interval y una scan window del mismo tamaño de 30, 100 y 200 ms, siendo estos valores más pequeños y más grandes que el advertising interval.

ADVERTISING INTERVAL (ms)	Scan Interval (ms)	Scan Windows (ms)
100	100	100
100	30	30
100	200	200

Tabla 1-Advertising Interval 100 ms, Scan Window y Scan Interval iguales

Además también se considera, para este mismo advertising interval, tamaños distintos de la ventana de escaneo y el intervalo de escaneo, en concreto, de la mitad, un cuarto y un octavo del scan interval más grande y de la mitad del scan interval más pequeño, de los escogidos anteriormente.

ADVERTISING INTERVAL (ms)	Scan Interval (ms)	Scan Windows (ms)
100	200	100
100	30	15
100	200	50
100	200	25

Tabla 2-Advertising Interval 100ms, Scan Window y Scan Interval distintos

Para un intervalo de advertising de 500 ms, se considera un scan interval y una scan window del mismo tamaño de 150, 500 y 1000 ms, siendo estos valores más pequeños y más grandes que el advertising interval.

ADVERTISING INTERVAL (ms)	Scan Interval (ms)	Scan Windows (ms)
500	500	500
500	150	150
500	1000	1000

Tabla 3-Advertising Interval 500ms, Scan Window y Scan Interval iguales

Además también se considera, para este mismo advertising interval, tamaños distintos de la ventana de escaneo y el intervalo de escaneo, en concreto, de la mitad, un cuarto y un octavo del scan interval más grande y de la mitad del scan interval más pequeño, de los escogidos anteriormente, igual que en el caso anterior.

ADVERTISING INTERVAL (ms)	Scan Interval (ms)	Scan Windows (ms)
500	1000	500
500	1000	250
500	1000	125
500	150	75

Tabla 4-Advertising Interval 500ms, Scan Window y Scan Interval distintos

Para un intervalo de advertising de 1000 ms, se considera un scan interval y una scan window del mismo tamaño de 300, 1000 y 2000 ms, siendo estos valores más pequeños y más grandes que el advertising interval, de nuevo.

ADVERTISING INTERVAL (ms)	Scan Interval (ms)	Scan Windows (ms)
1000	1000	1000
1000	300	300
1000	2000	2000

Tabla 5-Advertising Interval 1000ms, Scan Window y Scan Interval iguales

Para este último intervalo de advertising, solo se considera una scan interval de la mitad del valor más alto del scan interval y de la scan window, por la poca variabilidad de los resultados y la extensa cantidad de tiempo que consume realizar el ensayo.

ADVERTISING INTERVAL (ms)	Scan Interval (ms)	Scan Windows (ms)
1000	2000	1000
1000	300	150

Tabla 6-Advertising Interval 1000ms, Scan Window y Scan Interval distintos

También cabe mencionar que para todos los casos se transmiten 600 paquetes desde el advertiser, por lo cuál se espera, que en condiciones perfectas, se reciban 200 paquetes en el escáner.

En ciertos casos de interés se han realizado medidas extra, para producir estadísticas más fiables, pero el proceso general, en este aspecto, ha sido siempre el descrito anteriormente. En

concreto, en escenarios con retransmisores se ha optado por modificar el tamaño de ventana de escaneo frente al intervalo de escaneo para la mayoría de casos salvo 1, por cada Advertising Interval. Se ha elegido esta opción, ya que, reducir la ventana de escaneo del escáner final tiene el mismo comportamiento que para escenarios sin retransmisores.

#### 4.1.2-Escenarios sin retransmisores

Los parámetros que se modifican en la sección anterior se refieren principalmente a los diferentes tiempos básicos que intervienen en el funcionamiento de los dispositivos BLE, en esta sección se explican diferentes topologías de red, las cuales se estudiarán, a su vez, para cada uno de los casos anteriores.

También se quiere determinar si diferencias en las tramas de información, suponen diferencias en la proporción de tramas recibidas, se espera que no suceda.

##### Escenario Básico

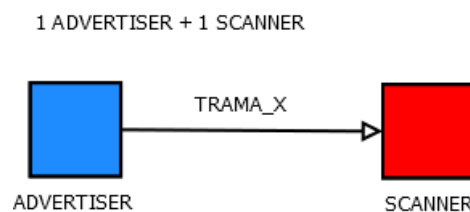


Figura 43- Escenario con topología básica

Este es el escenario más simple, un advertiser transmite tramas y un escáner las detecta. Se cuantifica la proporción de tramas recibidas frente a las transmitidas.

##### Escenarios con dos advertisers

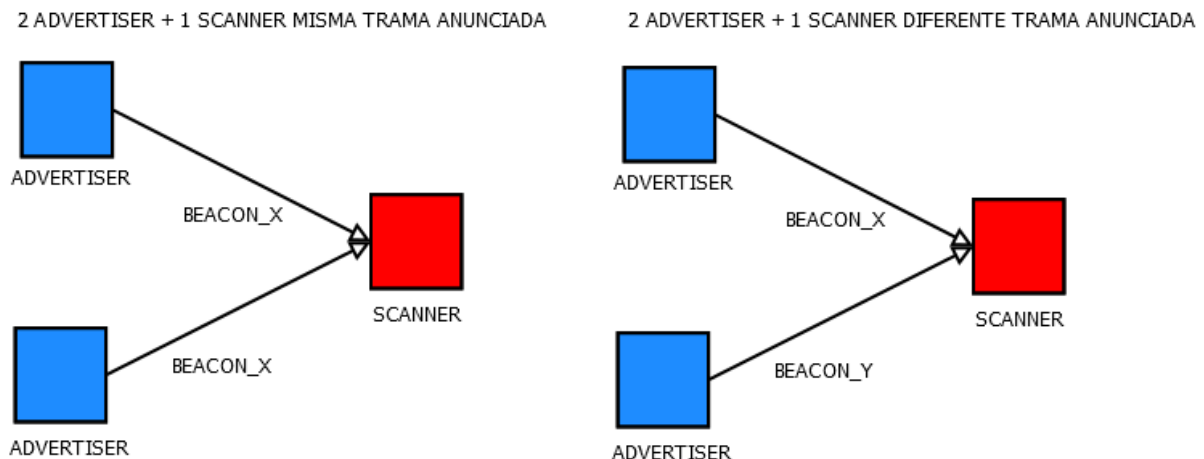


Figura 44-Escenarios con dos anunciantes

Este es el segundo escenario de estudio, donde surge la cuestión mencionada anteriormente, sufrirá cambios en la eficiencia que el escáner detecte ambas tramas, siendo estas diferentes o no supondrá diferencia.

##### Escenarios con 3 advertisers

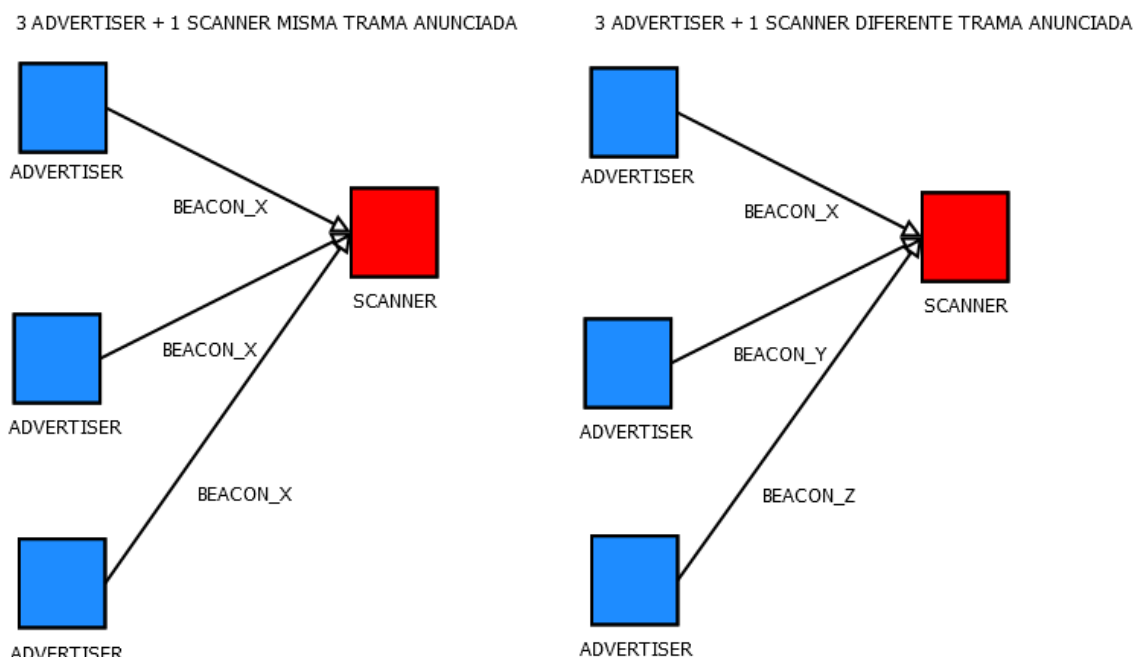


Figura 45-Escenarios con 3 anunciantes

En este caso se presenta la misma problemática que en el caso anterior en cuanto a si se producen variaciones en el número de tramas recibidas al transmitir tramas diferentes. Se espera que ocurra el mismo fenómeno que anteriormente. Además al introducir otro advertiser puede que las colisiones impliquen una pérdida en las tramas recibidas.

#### 4.1.3-Escenarios con retransmisores

En esta sección se presentan nuevas topologías, utilizando un retransmisor, es decir, un dispositivo que durante un tiempo escanea tramas de un anunciante y seguidamente deja esta actividad para transmitir tramas a un escáner final.

Escenario con 1 re-transmisor y 1 anunciante

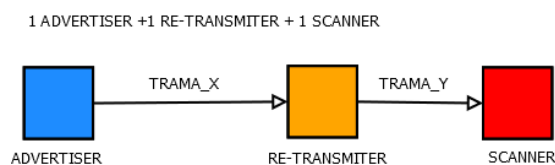


Figura 46-Escenario con anunciante, re-transmisor y escáner

Este es el escenario más básico utilizando los tres elementos, la trama\_x debe ser distinta de la trama\_y porque de otra manera el escáner la detectará las tramas emitidas por el anunciador directamente, además de las retransmitidas, si el escáner está en el rango de cobertura del anunciador y no se obtendrá un resultado interesante para el estudio.

Escenarios con 2 re-transmisores: 2 re-transmisores en cadena (serie).

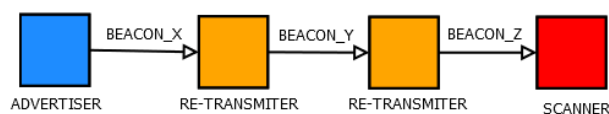


Figura 47-Escenario con 2 retransmisores en serie

En este caso se debe recibir un número igual al de paquetes transmitidos, ya que, el segundo re-transmisor, solo retransmite, si la trama recibida es la que el primer re-transmisor anuncia, que, a su vez, solo envía su trama, cuando recibe la esperada por el anunciador.

Es particularmente interesante en este escenario conocer las diferencias que se producen frente a colocar un re-transmisor único en medio.

## 4.2-Resultados del Estudio

### 4.2.1-Conclusiones Generales

Respecto al *tiempo de transmisión* se ha utilizado el LED integrado en el dispositivo BLE como forma de control, se ha cronometrado el tiempo en segundos en realizar la transmisión completa. Esto puede hacerse debido a que se configura el LED para permanecer encendido durante la transmisión de tramas e inmediatamente se apaga al finalizar la transmisión.

Es destacable que encender estos LED en escenarios con re-transmisores reduce la eficiencia del programa, por lo que siempre se comenta el código respectivo a esto para los ensayos de medidas de tramas. También es importante mencionar que las acciones que realice el anunciador y sobre todo los retransmisores, su eficiencia en el código, puede influir en el tiempo de transmisión.

Dicho esto, para los escenarios con re-transmisores, no se ha detectado ningún retardo apreciable para la percepción humana en cuanto al tiempo de transmisión, frente a los casos sin retransmisores.

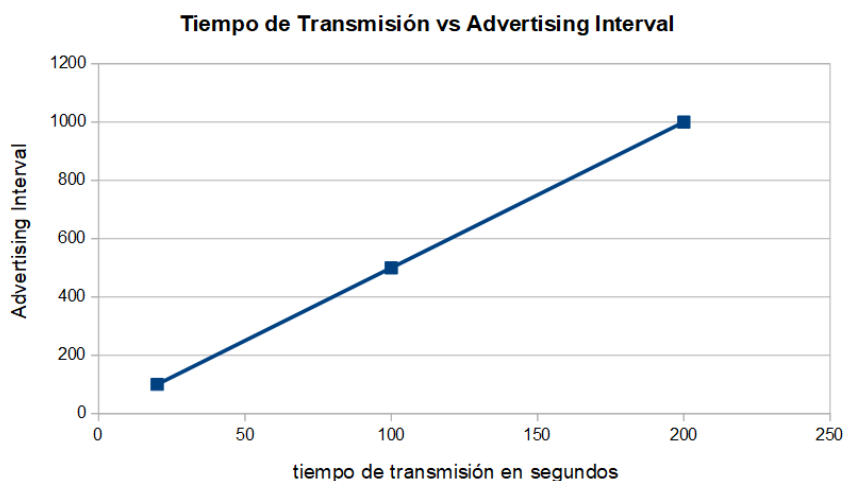


Figura 48-Tiempo de Transmisión /Advertising Interval



Para todos los casos de estudio el gráfico superior se mantiene sin cambios, es decir, el tiempo de transmisión es directamente proporcional a el advertising interval y en concreto sigue una proporción de  $\frac{1}{3}$  del advertising interval en ms.

Otra forma de mostrar las medidas es la siguiente tabla:

T(s)Transmission	ADVERTISING INTERVAL (ms)
20	100
100	500
200	1000

T(s)/AI
0.2
0.2
0.2

Tabla 7-Tiempo de Transmisión/ Advertising Interval

Respecto a los *escenarios sin retransmisores* podemos obtener varias conclusiones generales. En cuanto a si existe diferencia entre utilizar un escáner que acepte más de una trama distinta, se incluyen las siguientes medidas que demuestran que esto parece suceder.

Avg Rx Packets	Avg Rx Packets	Diferencia (F-E)	Diferencia/Paquetes Enviados
395	398	3	0.0075
197	196	-1	-0.0025
97	98	1	0.0025
397	394	-3	-0.0075
TRAMA_X+TRAMA_Y	TRAMA_X+ TRAMA_X		

Avg Rx Packets	Avg Rx Packets	Diferencia (F-E)	Diferencia/Paquetes Enviados
596	598	2	0.0033
292	297	5	0.0083
145	148	3	0.005
595	595	0	0
TRAMA_X+TRAMA_Y+TRAMA_Z	TRAMA_X+TRAMA_X+TRAMA_X		

Tabla 8-Comparativa Medidas Paquetes Recibidos con diferentes tramas o todas iguales

Estas medidas de los paquetes recibidos medios han sido realizadas para el mismo tamaño de ventana de escaneo, intervalo de escaneo y tiempo de advertising, siendo la diferencia entre los casos, que para el primero, se estudia el escenario con 2 anunciadores transmitiendo a la vez, transmitiendo todos diferentes tramas en la primera columna y todas iguales en la segunda columna. El segundo caso es el realizado con 3 anunciadores e igual configuración.

Se puede observar que la diferencia entre los casos en los que todas las tramas transmitidas son iguales y cuando esto no sucede es mínima o prácticamente nula, al compararla con el número de paquetes recibidos.

Por tanto la conclusión es que es independiente, al menos, con la configuración realizada, si los anunciadores anuncian varias tramas distintas y los escáneres las reciben a si anuncian las mismas.

#### 4.2.2 Conclusiones en cuanto a los paquetes recibidos

Esta sección es la más relevante en el trabajo, puesto que la eficiencia, en este caso, se considera como el número de tramas recibidas frente a las transmitidas.

Para los tres escenarios sin retransmisores se puede obtener una fórmula que puede predecir con gran fiabilidad los paquetes que se recibirán y es la siguiente.

$$(SW/SI) * PaqueteTX$$

Ecuación 1- Estimación de los paquetes recibidos a la salida para escenarios sin retransmisores

Siendo SW, Scanning Window, el tamaño de la ventana de escaneo, SI, Scanning Interval, el tamaño del intervalo de escaneo y PaquetesTX, el número de paquetes mandados.

Esta fórmula tiene una particularidad y es que no depende del intervalo de advertising, uno de los parámetros principales configurables que ha sido mencionado durante todo el trabajo. Las comprobaciones empíricas indican una independencia de este valor en cuanto a la salida cuando no se toman valores límite. Es decir, el escáner y los retransmisores no sufren una

reducción de la eficiencia de escaneo en función del canal donde los anunciantes emiten la trama o lo rápido que se cambia entre estos fuera de valores límite.

Para demostrar la calidad de la estimación en estos escenarios se incluyen las siguientes medidas y gráficas:

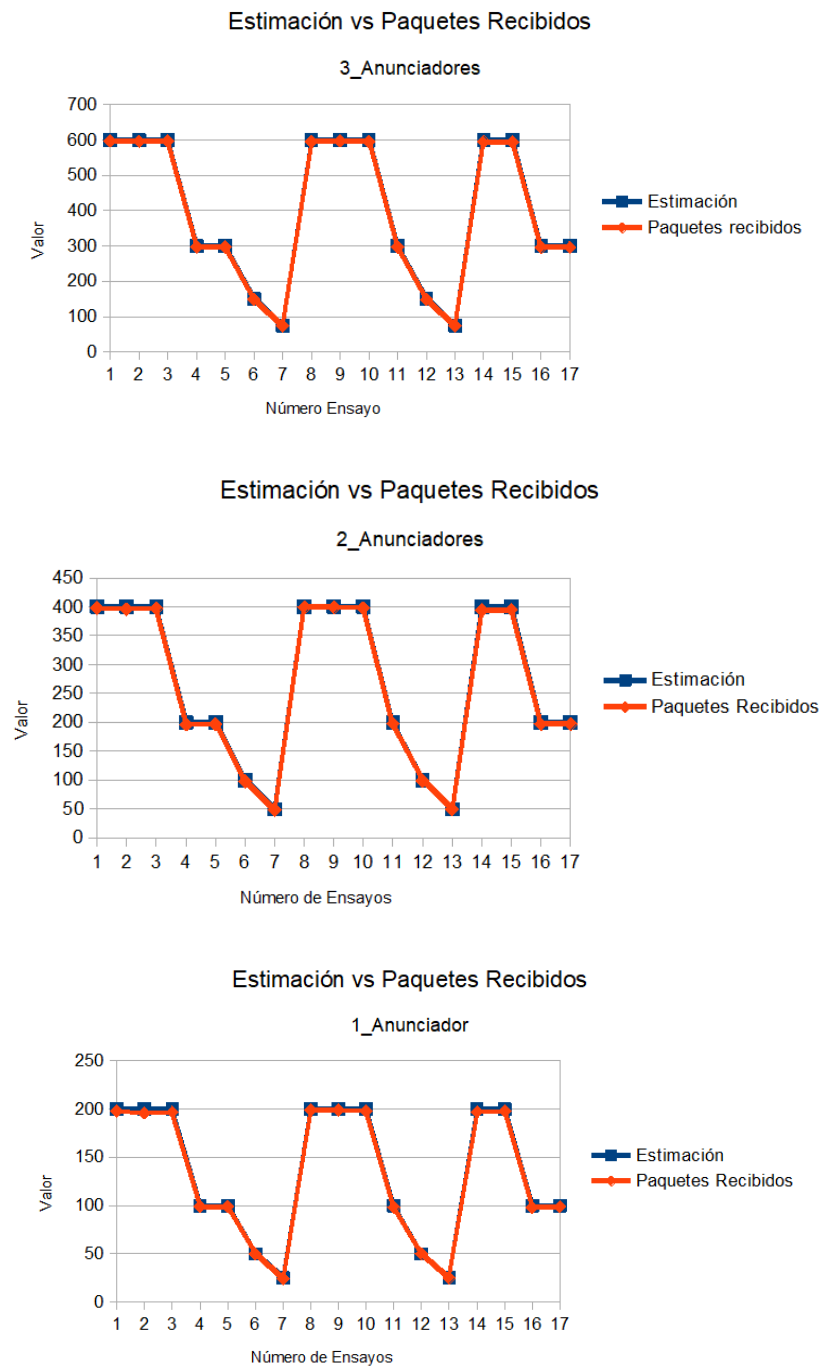


Figura 49-Gráficos Comparativos Estimación vs Paquetes Recibidos

Estos gráficos muestran en una línea azul los valores de la estimación para 17 diferentes escenarios con los mismos parámetros y los valores de los paquetes recibidos finales.

Se puede observar que ambas líneas se solapan prácticamente en su totalidad para los 3 casos de estudio sin retransmisores y los 17 ensayos distintos. Una de las hojas de cálculo del anexo muestran en mayor detalle las tablas completas, pero mostrar el gráfico es más orientativo.

A pesar de ello, si se considera interesante mostrar algunos de los valores de las tabla para mostrar las diferencias entre la estimación y la medida final. Esto se hará a continuación:

SW/SI SC *TxPackets	Diferencia(Estimación- Medida)	Diferencia/TX
200	2	0.01
200	4	0.02
200	3	0.015
100	1	0.005
100	1	0.005
50	0	0
25	1	0.005

Tabla 9-Comparativa Estimación vs Medida 1 Anunciador

Esta tabla pertenece al escenario con un solo anunciador. Se puede observar numéricamente como la diferencia de la estimación y la medida se mantiene mínima y más al compararse con el número de los paquetes enviados, la última columna.

SW/SI SC *TxPackets	Diferencia(Estimación- Medida)	Diferencia/TX
600	2	0.0033
600	4	0.0067
600	2	0.0033
300	3	0.005
300	2	0.0033
150	1	0.0017
75	2	0.0033

Tabla 10-Comparativa Estimación vs Medida 3 Anunciadores

Esta tabla pertenece al escenario de tres anunciadores, la conclusión se mantiene, pero la diferencia frente a los paquetes transmitidos es aún menor que anteriormente, ya que en este caso se transmiten 600 paquetes y no 200.

En un punto a parte, por lo comentado anteriormente se puede intuir que los escenarios sin retransmisores han tenido un comportamiento similar. Así ha sucedido, las medidas han resultado extremadamente similares y esto plantea una pregunta ¿Este comportamiento es extrapolable para cualquier número de anunciadores?

Para hallar respuesta a esta pregunta y debido a que sólo se contaba con 4 dispositivos no se pudo comprobar en la red propia, pero el trabajo *Optimizing Scan Times of BLE Scanning Systems*[1] realizado por el departamento de ingeniería eléctrica de la universidad de Mississippi es capaz de proporcionar esta respuesta buscada. [15]

En su estudio se describe como la probabilidad de colisión entre tramas existe y en concreto estudia este fenómeno para 100, 200, 400, 800 y 1000 dispositivos, con diferente tamaño de memoria. Para estas dimensiones de red si existirá una diferencia entre las tramas recibidas y las calculadas por el modelo, dadas las pérdidas producidas por colisiones, a pesar de que sean prácticamente inexistentes en el escenario con 3 anunciadores y menos probables aún en el escenario con 2 anunciadores y 1 solo.

Se muestra una comparativa entre los dos escenarios de estudio, aquél con un solo retransmisor entre el escáner y el anunciador y el que cuenta con dos retransmisores en cadena.

Avg Rx Packets 1RT	Avg Rx Packets 2RT	Diferencia Paquetes	Diferencia/Paquetes Enviados
196	194	2	0.01
97	95	2	0.01
60	57	3	0.015
39	37	2	0.01
15	11	4	0.02
98	98	0	0
199	195	4	0.02
96	96	0	0
66	64	2	0.01
46	46	0	0

35	36	-1	-0.005
97	99	-2	-0.01
195	194	1	0.005
189	193	-4	-0.02
65	64	1	0.005
49	42	7	0.035
99	98	1	0.005

Tabla 11-Comparativa Paquetes Recibidos 1 retransmisor vs 2 retransmisores en cadena

Esta tabla muestra las diferencias en las medidas de los paquetes medios recibidos para ambos escenarios configurados con los mismos parámetros de advertising interval, scanning window y scanning interval, incluyendo los de los retransmisores.

Se podría imaginar que al encontrarse dos dispositivos retransmisores entre transmisor y receptor, los cambios en el tamaño de la ventana de escaneo frente al intervalo de transmisión de los retransmisores, perjudicasen a la eficiencia de la red en mayor medida que si solo se encontrase uno de estos, pero esto no sucede.

Como se puede ver en la tabla la diferencia entre ambos escenarios es mínima para los mismos parámetros de configuración, lo cual es muy positivo, porque significa que una red que cuente con cadenas más largas de retransmisores no perderá eficiencia extremadamente rápido.

En cuanto a las tramas recibidas, los escenarios con retransmisores se comportan de manera similar a los escenarios sin retransmisores si hay variaciones en la ventana del escáner final frente a un intervalo de escaneo fijo. La siguiente tabla muestra escenarios que representan lo comentado:

	RT	RT	SC	SC	
ADVERTISING INTERVAL (ms)	Scan Interval (ms)	Scan Windows (ms)	Scan Interval (ms)	Scan Windows (ms)	RX Paquetes
100	100	100	100	100	194
100	200	200	100	50	98
500	500	500	500	500	195
500	1000	1000	500	250	99

1000	1000	1000	1000	1000	194
1000	2000	2000	1000	500	98

Tabla 12-Comparativa Escenarios con Retransmisores variaciones SW/SI SC

Sin embargo, los casos únicos surgen al variar la ventana de escaneo frente al intervalo de escaneo de los retransmisores. La siguiente tabla muestra todos los casos relevantes medidos junto con los parámetros de configuración para el caso de retransmisores en cadena.

ADVERTISING INTERVAL (ms)	Scan Interval (ms)	Scan Windows (ms)	Scan Interval (ms)	Scan Windows (ms)	RX Paquetes	
100	200	150	100	100	95	2 RT
100	200	100	100	100	57	
100	200	50	100	100	37	
100	200	20	100	100	11	
500	1000	750	500	500	96	
500	1000	500	500	500	64	
500	1000	250	500	500	46	
500	1000	100	500	500	36	
1000	2000	1000	1000	1000	64	
1000	2000	500	1000	1000	42	

Tabla 13-Comparativa Escenarios con Retransmisores variaciones SW/SI RT

Con afán de intentar realizar una estimación de los paquetes recibidos finales, se crea la siguiente fórmula, replicando la metodología para los cambios en la configuración del escáner.

$$SW/SI (Retransmisor) * SW/SI(Scanner) * TXPaquetes$$

Ecuación 2- Modelo de Estimación para cambios en SW/SI en retransmisores

No obstante, este modelo no estima correctamente el resultado final, para mostrar la diferencia de las medidas empíricas frente a los valores estimados se incluye la siguiente tabla como ejemplo.

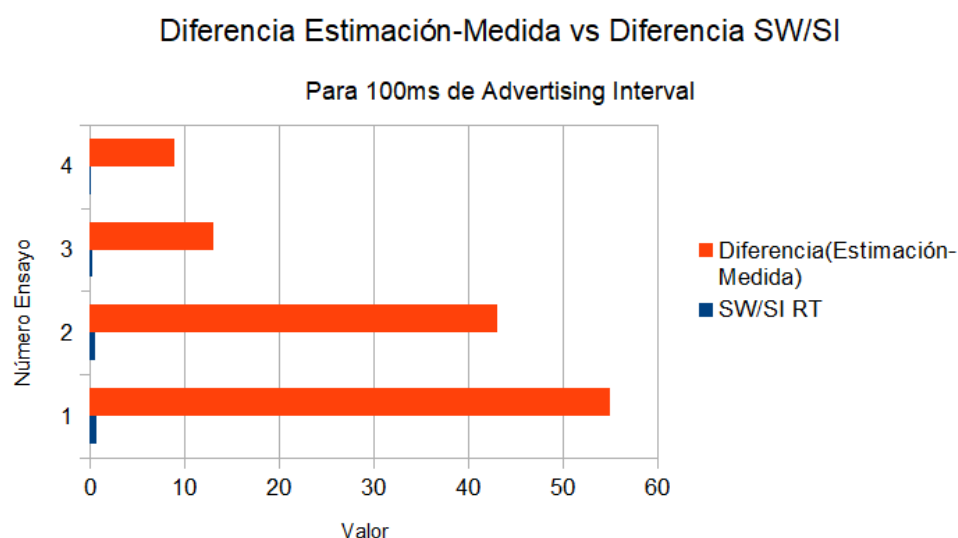
### ESCENARIO 1 RE-TRANSMISOR

ADVERTISING INTERVAL(ms)	SW/SI RT	Diferencia(Estimación- Medida)	Diferencia/TX
100	1	6	0.03
100	0.75	55	0.275
100	0.5	43	0.215
100	0.25	13	0.065
100	0.1	9	0.045

Tabla 14-Comparativa Diferencias Modelo y Medida Retransmisores cambios en SW/SI RT

Es común que la medida de las tramas recibidas resulte menor que las esperadas por el modelo.

Se propuso tratar de buscar si existía una tendencia fija de crecimiento o decrecimiento de la diferencia Estimación-Medida, frente a la diferencia entre SW/SI de los retransmisores. Los siguientes gráficos tratan de mostrar justo esto:





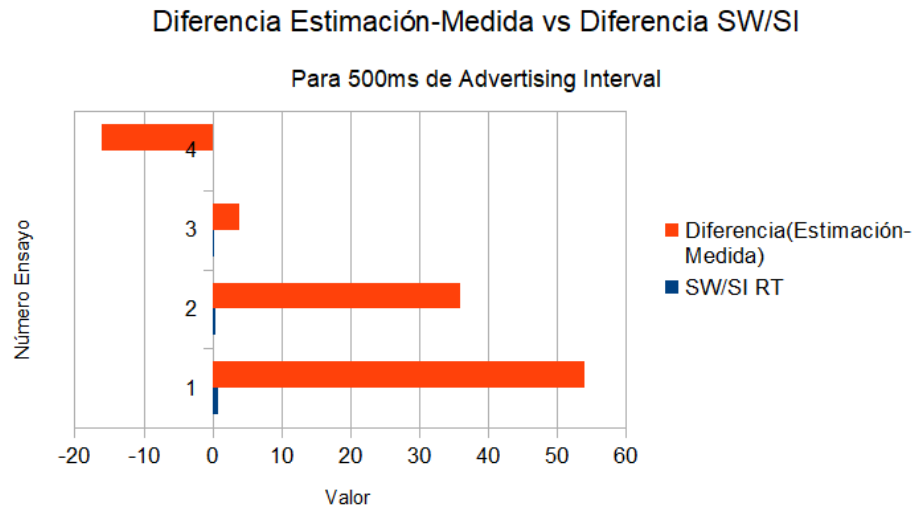


Figura 50-Gráficos Comparativos SW/SI vs Diferencia en la Estimación para 500ms de AI

Si bien puede parecer que para 100ms podría existir esa proporción, esto es una ilusión falsa, ya que en el caso de 500ms, se puede percibir claramente que esto no sucede, además tampoco tiene sentido pensar que existiría tal proporción si en los casos sin retransmisores no existe dependencia entre estas dos magnitudes, la diferencia en la estimación y los cambios en la ventana de escaneo del escáner.

## Capítulo 5

### Conclusiones del Proyecto y Trabajo Futuro

Las topologías de redes Bluetooth, como la que inspiró el proyecto en las que los mensajes no se transmiten de manera broadcast sino direccionalmente a través de retransmisores hasta los dispositivos de escaneo finales son muy interesantes.

Esta estructura posee grandes ventajas, como la reducción de la congestión de la red, ya que elimina gran cantidad de tramas sin uso de la red, al no reenviar tramas a destinos fuera de la ruta pre-establecida. Además, existen una gran serie de protocolos existentes que serían aplicables en las partes direccionales de la red, tanto de enrutamiento automático como protocolos de seguridad, control de tráfico etc. En redes grandes (más de 100 anunciadores) se presenta como una solución interesante a los problemas creados por las colisiones.

Sin embargo, esta topología no permite que los dispositivos transmisores realicen tareas simultáneas sin perjudicar a la eficiencia de manera grave y poco estimable. Además no presenta grandes ventajas en redes pequeñas (menos de 100 anunciadores), donde las colisiones tienen baja probabilidad de ocurrir.

Respecto a trabajo futuro, los escenarios con retransmisores en paralelo pueden resultar interesantes, donde se multiplique el número de tramas retransmitidas, para los anunciadores deseados. Esto añadiría redundancia, implicaría un mayor número de tramas recibidas, sin perder la direccionalidad de la red, no se producirían envíos a destinos fuera de la ruta. No se obtuvieron los resultados esperados cuando se intentó realizar este escenario, por lo que sería interesante seguir explorando el campo.

Por otro lado, tampoco existen estudios para redes BLE con cantidades considerables de retransmisores en cadena, sería de especial interés realizar un estudio mayor para conocer bien las limitaciones a gran escala de la topología de red propuesta.

Además de todo ello, se debería realizar un estudio complementario a este sobre las redes con modelo broadcast y tiempos de vida de los mensajes determinados por el número de saltos. Se procedería de manera análoga midiendo las tramas llegadas a los destinos finales para diversos tamaños de ventana y con el número de saltos como nuevo parámetro a optimizar.

De esta manera, utilizando ambos estudios y realizados en mayor profundidad, se podría elaborar un informe con un contexto general de las redes Bluetooth Low Energy, tal que proporcionase conclusiones y comparativas entre la utilización de ambos modelos de comunicación.

## Capítulo 6

### Bibliografía, Webgrafía y Otras Referencias

[1] Dixys L.Hernández Rojas. Tiago M.Fernández Caramés, Paula Fraga Lamas, Carlos J Escudero, “Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing Through BLE Beacons in IoT Telemetry Applications”, Universidad Técnica de Machala, Ecuador, 27 de Diciembre de 2017

<https://www.mdpi.com/1424-8220/18/1/57>

[2] Carlos Andrés García Cifuentes, “Diseño e Implementación de una red inalámbrica de sensores aplicados a la instrumentación biomédica-RIB”, Universidad Nacional de Entre Rios, Argentina, Mayo 2011.

[https://www.researchgate.net/publication/268445331\\_Design\\_and\\_Implementation\\_of\\_a\\_Wireless\\_Sensor\\_Network\\_Applied\\_in\\_Biomedical\\_Instrumentation\\_In\\_Spanish](https://www.researchgate.net/publication/268445331_Design_and_Implementation_of_a_Wireless_Sensor_Network_Applied_in_Biomedical_Instrumentation_In_Spanish)

[3] “Bluetooth Special Interest Group”, (19/2/2023), Wikipedia. Consulta: 06/09/2023, [Web Online]

[https://es.wikipedia.org/wiki/Bluetooth\\_Special\\_Interest\\_Group](https://es.wikipedia.org/wiki/Bluetooth_Special_Interest_Group)

[4] “Bluetooth de baja energía”, (30/11/2021), Wikipedia. Consulta: 06/09/2023, [Web Online]

[https://es.wikipedia.org/wiki/Bluetooth\\_de\\_baja\\_energ%C3%ADa](https://es.wikipedia.org/wiki/Bluetooth_de_baja_energ%C3%ADa)

[5] Novida, “BLE (Bluetooth Low Energy) ¿Qué es y cómo usarlo en IoT?” [Web Online]

<https://www.novida.com/es/blog/ble/>

[6] IDcapt, “La tecnología BLE”.Consulta:06/09/2023, [Web Online]

<https://www.idcapt.com/es/tecnologia-ble/>

[7] Altbeacon.org, “Altbeacon The Open Proximity Beacon Specification”, Consulta: 07/09/2023, [Web Online]

<https://altbeacon.org/>

[8] The Mathworks Inc, “Bluetooth LE Channel Selection Algorithms”. Consulta:06/09/2023, [Web Online]

<https://es.mathworks.com/help/bluetooth/ug/bluetooth-le-channel-selection-algorithms.html>

[9] Cypress Semiconductor Corporation, “CY5677 CYSMART™, BLE 4.2 USB DONGLE”, [Documento pdf Online]

[https://www.infineon.com/dgdl/Infineon-Quick\\_Start\\_Guide-UserManual-v01\\_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0efc004c11f6](https://www.infineon.com/dgdl/Infineon-Quick_Start_Guide-UserManual-v01_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0efc004c11f6)

[10] Infineon Technologies AG, “PSoC™ Creator”. Consulta: 1/07/2023 [Web Online]

<https://www.infineon.com/cms/en/design-support/tools/sdk/psoc-software/psoc-creator/>

[11] Cypress Semiconductor Corporation, “PSoC User Guide” [Documento pdf Online]

[https://www.infineon.com/dgdl/Infineon-PSoC\\_Creator\\_User\\_Guide-UserManual-v15\\_00-EN](https://www.infineon.com/dgdl/Infineon-PSoC_Creator_User_Guide-UserManual-v15_00-EN)

[12] Uday Argawal, Ajay Sahu, “PSoC 4 BLE, Designing BLE Applications”

[Documento pdf Online]

[https://www.infineon.com/dgdl/Infineon-AN91184\\_PSoC\\_4\\_BLE\\_Designing\\_BLE\\_Applications-ApplicationNotes-v07\\_00-EN.pdf?fileId=8ac78c8c7cdc391c017d07363095595a&utm\\_source=cypress&utm\\_medium=referral&utm\\_campaign=202110\\_globe\\_en\\_all\\_integration-application\\_note](https://www.infineon.com/dgdl/Infineon-AN91184_PSoC_4_BLE_Designing_BLE_Applications-ApplicationNotes-v07_00-EN.pdf?fileId=8ac78c8c7cdc391c017d07363095595a&utm_source=cypress&utm_medium=referral&utm_campaign=202110_globe_en_all_integration-application_note)

[13] Cypress Semiconductor Corporation, “BLE dongle schematic”, [Documento pdf online]

[https://www.infineon.com/dgdl/Infineon-Bluetooth\\_Low\\_Energy\\_Dongle\\_Schematic-PCBDesignData-v01\\_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0efc9a3912d4](https://www.infineon.com/dgdl/Infineon-Bluetooth_Low_Energy_Dongle_Schematic-PCBDesignData-v01_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0efc9a3912d4)

[14] Ting-Chao Hou, Kuo Chang Huang, “Accurate Analysis on Bluetooth Low Energy Neighbor Discovery”, National Chung Cheng University, Chiayi, Taiwan

<https://www.scirp.org/journal/paperinformation.aspx?paperid=106311>

[15] John N Daigle, George A. Humphrey II, Henry C. Lena V, Avijit Sarker, “Optimizing Scan Times of BLE Scanning Systems”, University of Mississippi, Oxford, MS, USA

<https://par.nsf.gov/servlets/purl/10275622>