



Universidad de Jaén

Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

DESARROLLO DE UNA PLATAFORMA BÁSICA DE COMUNICACIÓN BASADA EN BLE (BLUETOOTH LOW ENERGY)

Alumno: Antonio Francisco Moreno García

Tutor: Prof. D. Jesús de la Casa Cárdenas

Depto.: Ingeniería Electrónica y Automática

Junio, 2020

RESUMEN

Este proyecto consiste en el desarrollo de una plataforma básica de comunicación basada en una tecnología de baja potencia, de área personal inalámbrica, como es *Bluetooth Low Energy* (BLE, de aquí en adelante).

Para la realización de este proyecto se ha utilizado el módulo CY8CKIT-042-BLE, el cual pertenece a una familia de microcontroladores desarrollados por la empresa Cypress cuya denominación comercial es PSoC (*Programable System on Chip*). Este módulo está conectado a un pequeño circuito que incluye tanto el sensor DHT22, como el sensor de proximidad. La programación de dicho módulo, donde se implementa el funcionamiento de los sensores, se desarrolla a través del software PSoC Creator.

Por último, se realiza una aplicación móvil a través del IDE Android Studio, donde se utilizan los lenguajes de programación XML y JavaScript. Con esta aplicación se establece la conexión con el microcontrolador de PSoC y se obtiene información de las medidas realizadas.

ÍNDICE DE ILUSTRACIONES Y TABLAS

Tabla 1. Recursos utilizados	13
Ilustración 1. Dispositivos con bluetooth, BLE o ambas.	15
Ilustración 2. Canales físicos tecnología BLE	15
Ilustración 3. Historia del BLE	16
Ilustración 4. Arquitectura BLE	17
Ilustración 5. Internet of Things	18
Ilustración 6. Smart Agriculture Xtreme.	19
Ilustración 7. Niveles de control para el cultivo	19
Ilustración 8. Diagrama conectividad Smart Agriculture Xtreme.....	20
Ilustración 9. de Libelium-SmartEye Agriculture Solution Kit.....	20
Ilustración 10. MySignals	21
Ilustración 11. Diagrama conectividad MySignals.....	22
Ilustración 12. Kits MySignals Libelium	22
Ilustración 13. Conexión maestro-esclavo	25
Ilustración 14. Módulo I2C.....	26
Ilustración 15. Conexión maestro-esclavo	26
Ilustración 16. Sensor DHT22	27
Tabla 2. Patillaje sensor DHT22.....	27
Ilustración 17. Sensor proximidad	28
Ilustración 18. Flujo operativo sensor proximidad.....	28
Ilustración 19. Especificación pin para proximidad	29
Ilustración 20. Pantalla de cristal líquido	29
Ilustración 21. Diseño conexiones.....	30
Ilustración 22. Pines SDA y SCL.....	30
Ilustración 23. Montaje en placa de prototipos (1)	31
Ilustración 24. Montaje en placa de prototipos (2)	31
Ilustración 25. Configuración del sensor DHT22.....	32
Ilustración 26. Configuración del sensor de proximidad	33
Ilustración 27. Configuración del protocolo I2C	33
Ilustración 28. Configuración de pines (1).....	34
Ilustración 29. Configuración de pines (2).....	34
Ilustración 30. Archivos autogenerados al construir el proyecto.....	35
Ilustración 31. Ficheros desarrollados en PSoC Creator.....	35
Ilustración 32. Componentes gráficos utilizados para el desarrollo del proyecto (PSoC Creator) ..	36
Ilustración 33. Configuración del perfil BLE	36
Ilustración 34. Configuración de servicios y características	37
Ilustración 35. Petición para activar Bluetooth	40
Ilustración 36. Pantalla principal de la aplicación.....	40
Ilustración 37. Pantalla secundaria aplicación	41
Ilustración 38. Medida humedad	42
Ilustración 39. Medida temperatura.....	43
Ilustración 40. Medida proximidad.....	44
Ilustración 41. BLE Pioneer kit	47
Tabla 3. Componentes BLE Pioneer kit	47
Ilustración 42. Mapa de pines placa BLE Pioneer kit.....	48
Ilustración 43. Módulo CY8CKIT-143A	49
Tabla 4. Especificaciones CY8C4248LQI-BL583	50
Ilustración 44. Hoja de características DHT22.....	51
Ilustración 45. Características eléctricas DHT22	52

<i>Ilustración 46. Cypress.....</i>	<i>53</i>
<i>Ilustración 47. Microcontrolador CY8C4248LQI-BL583.....</i>	<i>55</i>
<i>Ilustración 48. Módulo PSoC 4 BLE.....</i>	<i>56</i>
<i>Ilustración 49. Partes del software PSoC Creator.....</i>	<i>57</i>
<i>Tabla 5. Comparación de aspectos técnicos entre bluetooth clásico y BLE</i>	<i>59</i>
<i>Tabla 6. Canales BLE</i>	<i>60</i>
<i>Ilustración 50. Canales IEEE 802.11 EEUU</i>	<i>61</i>
<i>Ilustración 51. Estructura máquina de estado capa de enlace</i>	<i>62</i>
<i>Ilustración 52. Comunicación esclavo-maestro.....</i>	<i>63</i>
<i>Ilustración 53. Estructura paquetes capa de enlace</i>	<i>63</i>
<i>Ilustración 54. Estructura modo prueba directa</i>	<i>64</i>
<i>Ilustración 55. UART, USB y SDIO</i>	<i>65</i>
<i>Ilustración 56. Envío de comandos del L2CAP.....</i>	<i>66</i>
<i>Ilustración 57. Estructura paquete de respuesta del ATT.....</i>	<i>66</i>
<i>Ilustración 58. Estructura perfil del GATT</i>	<i>68</i>
<i>Ilustración 59. UUID base del bluetooth.....</i>	<i>68</i>
<i>Tabla 7. Operaciones sobre una característica</i>	<i>70</i>
<i>Tabla 8. Modos y procedimientos del GAP.....</i>	<i>71</i>

ÍNDICE

RESUMEN	3
ÍNDICE DE ILUSTRACIONES Y TABLAS.....	5
ÍNDICE	7
AGRADECIMIENTOS	9
1. INTRODUCCIÓN	10
1.1. PROPÓSITO.....	10
1.2. MOTIVACIÓN Y OBJETIVOS	10
1.3. METODOLOGÍA	11
1.4. ORGANIZACIÓN	12
1.5. RECURSOS UTILIZADOS.....	12
2. <i>BLUETOOTH LOW ENERGY</i>	14
2.1. ¿QUÉ ES <i>BLUETOOTH LOW ENERGY</i> ?.....	14
2.2. HISTORIA DEL <i>BLUETOOTH LOW ENERGY</i>	16
2.3. ARQUITECTURA DEL BLE.....	17
3. SISTEMAS DE MEDIDA CON CONEXIÓN INALÁMBRICA. ESTADO DEL ARTE.....	18
4. DESARROLLO DEL TRABAJO.....	23
4.1. INTRODUCCIÓN	23
4.2. PRERREQUISITOS	23
4.3. DESARROLLO	24
4.3.1. Tecnología BLE	24
4.3.2. Protocolo I2C	25
4.3.3. Análisis de los sensores y la pantalla LCD.....	26
4.3.4. Desarrollo del circuito en la placa de prototipos	29
4.3.5. Desarrollo del circuito en PSoC Creator.....	32
4.3.6. Configuración del módulo BLE.....	36
4.3.7. Desarrollo de la aplicación móvil.....	39
5. CONCLUSIONES Y LÍNEAS FUTURAS.....	45
ANEXO 1. BLE PIONEER KIT	47
ANEXO 2. SENSOR DE TEMPERATURA Y HUMEDAD DHT22	51
ANEXO 3. CYPRESS.....	53

PSOC	53
<i>PSoC 4 BLE.....</i>	<i>54</i>
<i>PSoC Creator</i>	<i>56</i>
ANEXO 4. <i>BLUETOOTH LOW ENERGY</i>	58
CARACTERÍSTICAS DEL BLE.....	58
CAPAS DE LA ARQUITECTURA BLE.....	60
ANEXO 5. VÍDEOS DE FUNCIONAMIENTO. APLICACIÓN MÓVIL Y PANTALLA LCD.....	72
ANEXO 6. CÓDIGO DE PSOC CREATOR Y ANDROID STUDIO....	73
6. APÉNDICE	74
7. ACRÓNIMOS	76
8. BIBLIOGRAFÍA.....	77

AGRADECIMIENTOS

Durante los años que ha durado mi etapa de formación académica, han sido muchas personas las que se han cruzado en mi camino, a los cuales estaré siempre agradecido por los momentos compartidos.

En primer lugar, quiero agradecer a todos los familiares que me han estado apoyando en los momentos buenos y no tan buenos, a través de los cuales está casi obligado a pasar un estudiante de Ingeniería.

Agradecer también a los compañeros que esta etapa ha puesto a mi lado y que hoy día ya se consideran amigos y, por supuesto, a todos los profesores que han hecho posible que este momento llegara gracias a los conocimientos y lecciones que han aportado hacia mi persona.

Por último, destacar a uno de estos profesores, D. Jesús de la Casa Cárdenas, tutor y guía de este Trabajo de Fin de Grado, que ha hecho posible que este proyecto sea realizado con éxito.

1. INTRODUCCIÓN

1.1. PROPÓSITO

El propósito de este Trabajo de Fin de Grado es el análisis de la tecnología *Bluetooth Low Energy* (BLE), mediante el diseño e implementación de una plataforma básica y una aplicación móvil para realizar una conexión entre ambos mediante la tecnología BLE, a través de la cual se transmiten datos de temperatura y humedad obtenidos por un sensor DHT22, y proximidad, dato obtenido a través de un sensor táctil capacitivo que incorpora el kit utilizado (*BLE Pioneer kit*, del que se hablará más adelante). Estos datos son mostrados tanto en un dispositivo móvil, a través de la aplicación móvil implementada, como en una pantalla de cristal líquido.

1.2. MOTIVACIÓN Y OBJETIVOS

Es una realidad que la tecnología está en pleno crecimiento en la actualidad y junto con esta van creciendo las necesidades de la sociedad y, por tanto, se incrementa la necesidad de crear nuevas técnicas que satisfagan estas necesidades.

La motivación principal para este proyecto, es establecer una conexión, a través de la tecnología inalámbrica de baja potencia BLE, entre un dispositivo móvil y la placa base del *BLE Pioneer kit*, a través de la cual se obtienen los datos de temperatura, humedad y proximidad, mediante dos sensores básicos que van conectados a dicha placa. Para un desarrollo exitoso de este Trabajo de Fin de Grado, se plantearon los siguientes objetivos:

- **Estudio de la tecnología BLE y su utilización en la comunicación con sensores básicos:** Estudiar y comprender en qué consiste esta nueva tecnología de comunicación entre dispositivos, para posteriormente utilizar el conocimiento adquirido en este análisis.
- **Diseño y desarrollo de una plataforma básica:** Diseño y desarrollo de una plataforma básica que incluye un sensor de proximidad y el sensor DHT22, que mide temperatura y humedad, para posteriormente enviar dichos datos a través del BLE.

- **Diseño y desarrollo de una aplicación basada en tecnología BLE para la comunicación inalámbrica entre la plataforma y un dispositivo móvil:** Diseño y desarrollo de una aplicación móvil para la comunicación, a través de la tecnología *Bluetooth Low Energy*, con la plataforma que incluye los sensores previamente mencionados.
- **Evaluación, mediante experimentación, de la plataforma y la aplicación desarrollada:** Establecer la conexión para comprobar el correcto funcionamiento de esta mediante la recepción de los datos medidos por los sensores.

1.3. METODOLOGÍA

Pasos a seguir para llegar a una solución exitosa:

- Estudio y análisis de la tecnología *Bluetooth Low Energy*.
- Estado del arte de los sistemas de medida inalámbricos.
- Estudio y selección de los sensores básicos que se van a utilizar.
- Estudio de sistemas de desarrollo de dispositivos *Bluetooth Low Energy*.
- Diseño y desarrollo de una plataforma de bajo coste.
- Diseño y desarrollo de una aplicación móvil.
- Experimentación y evaluación de la plataforma desarrollada, a través de la aplicación móvil.
- Conclusiones obtenidas y líneas de futuro.

1.4. ORGANIZACIÓN

Este proyecto se dividirá en cinco partes:

1ª Parte: Consta de una breve introducción, junto con el propósito del proyecto, los objetivos impuestos previamente y la metodología que se va a seguir para el desarrollo de este.

2ª Parte: Se analizan todos los conceptos necesarios para entender la tecnología *Bluetooth Low Energy*.

3ª Parte: Presentación de la empresa *Cypress* y sus herramientas que se han utilizado en este proyecto.

4ª Parte: Explicación de cómo ha sido el desarrollo del proyecto, terminando con una conclusión de los resultados obtenidos y con la explicación individual de todos los ficheros utilizados para la realización de este.

5ª Parte: Anexos, donde se incluye documentación adicional relacionada con el desarrollo del proyecto. Por último, para concluir la memoria se menciona la bibliografía consultada para la realización de esta.

1.5. RECURSOS UTILIZADOS

 CY8CKIT-042-BLE <i>Bluetooth Low Energy Pioneer Kit</i>	 Cable para placa de prototipo macho-macho y macho-hembra	 Sensor de temperatura y humedad DHT22	 Resistencia de 10 kΩ
---	---	--	---

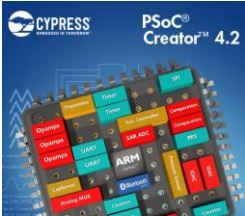
 Dispositivo móvil	 Pantalla de cristal líquido con protocolo I2C	 Ordenador portátil
 Software PSoC Creator 4.2	 Software CySmart	 Software Android Studio

Tabla 1. Recursos utilizados

2. **BLUETOOTH LOW ENERGY**

2.1. ¿QUÉ ES **BLUETOOTH LOW ENERGY**?

La tecnología *bluetooth* nació en el año 1994, con la idea principal de sustituir la tecnología de cableado de corta distancia y hoy en día se puede encontrar en lugares públicos como, por ejemplo, balizas electrónicas que guían a bomberos por los edificios o en dispositivos médicos portátiles encargados de enviar datos del paciente a las computadoras del médico.

La versión *Bluetooth 4.0* incorporó una nueva tecnología, *Bluetooth Low Energy*, aunque puede ser también referida como *Bluetooth Smart* o *Bluetooth 4.0*. Esta tecnología comenzó destinada a aplicaciones como:

- ❖ Seguridad y sensores de proximidad.
- ❖ Sector automovilístico.
- ❖ Contadores inteligentes dentro del sector energético.
- ❖ Dispositivos para salud, hogar y bienestar.

El uso de esta en dispositivos es debido a su mayor ahorro de energía frente a la tecnología *Wireless Fidelity (WIFI)*, aunque WIFI se sigue utilizando ya que tiene algunas características mejores que BLE, entre las que destaca el alcance.

A causa de su ahorro energético, su uso habitual se encuentra en aplicaciones en las que el consumo de energía es un elemento principal, como por ejemplo en dispositivos que trabajen con batería o dispositivos diseñados para usos largos. También es utilizada para aplicaciones en las que se encuentran sensores que tienen que transferir cantidades pequeñas de datos con poca frecuencia o en dispositivos *wearables*, es decir, dispositivos que con solo llevarlos puestos son capaces de recoger y transmitir datos del cuerpo o del estilo de vida de una persona. Un ejemplo de dispositivo *wearable* es la pulsera de actividad, capaz de informar de la frecuencia cardíaca o el número de pasos realizados, entre otros datos.

Ambas tecnologías, *bluetooth* clásico y BLE, son compatibles, pero no se puede establecer una comunicación directa entre ellas. Es por esto que los teléfonos inteligentes y algunos ordenadores ya traen incorporadas ambas tecnologías.

El *bluetooth* clásico es utilizado para la transmisión de audio (auriculares, altavoz) o transmisión de archivos, teniendo mayor velocidad de datos que BLE (3 Mbps frente a 1 Mbps del BLE), a pesar de esta ventaja la tecnología de *bluetooth* no es óptima para baja potencia. La tecnología BLE sin embargo se utiliza para sensores de datos o control de dispositivos, además de estar destinada, como su nombre indica, a la baja potencia. También se puede encontrar en dispositivos como relojes o pulseras inteligentes, o casas domóticas (Ilustración 1).

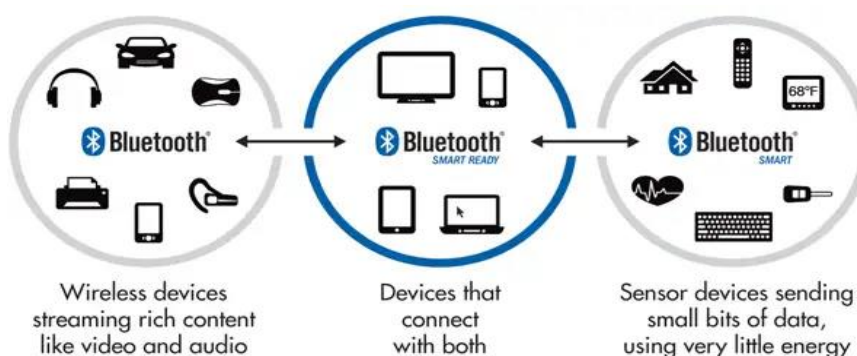


Ilustración 1. Dispositivos con bluetooth, BLE o ambas.

La tecnología BLE crece año tras año de forma agigantada. Su estructura se compone de 40 canales físicos en la banda de frecuencia de 2.4 GHz, con una separación entre estos de 2 MHz. De los 40 canales, 3 son utilizados para publicidad y el resto para datos, siendo estos los dos tipos de transmisión de BLE (Ilustración 2). Todos los dispositivos periféricos comienzan con el modo publicitario, que permite transmitir información que define las intenciones que el dispositivo tiene.

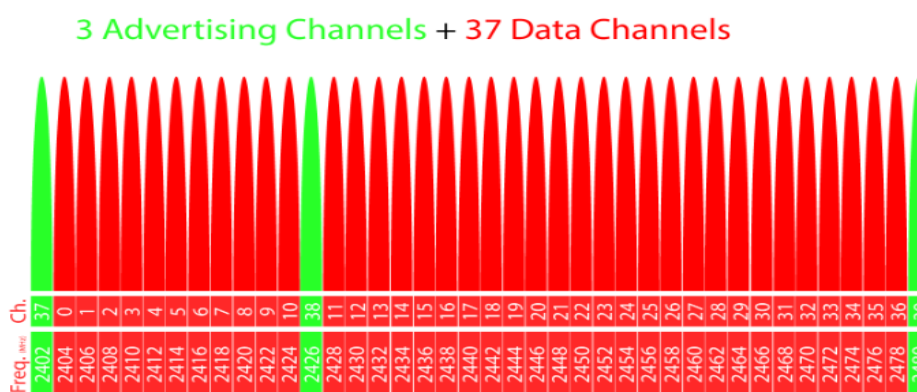


Ilustración 2. Canales físicos tecnología BLE

2.2. HISTORIA DEL *BLUETOOTH LOW ENERGY*

A continuación, se va a hacer un recorrido por la historia del BLE (Ilustración 3).

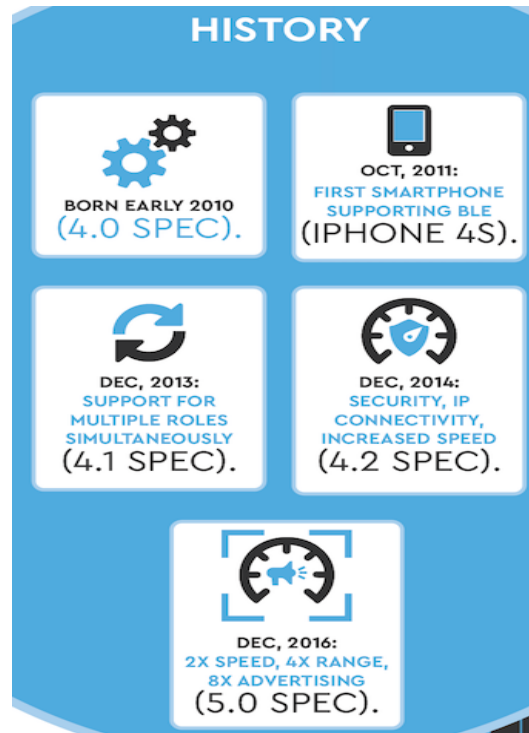


Ilustración 3. Historia del BLE

La tecnología BLE nació en el año 2010 con la llegada de la versión *Bluetooth* 4.0.

En octubre de 2011 apareció el primer dispositivo que soportaba esta tecnología, un dispositivo móvil, el iPhone 4S perteneciente a la compañía Apple.

En diciembre de 2013, llega al mercado el *Bluetooth* 4.1, tecnología que mejora la versión 4.0, corrigiendo las interferencias que se encuentren y además permitiendo la conexión simultánea, es decir, que un dispositivo pueda ser utilizado tanto de emisor como de receptor de datos al mismo tiempo.

Un año después, en diciembre de 2014, aparece el *Bluetooth* 4.2, que permite que un dispositivo se conecte a internet, a través de la conectividad de *bluetooth* sobre IPv6, además de aportar a los emparejamientos entre dispositivos una mayor seguridad y velocidad de

transmisión de datos. Con este nivel de *bluetooth* es que con el que se trabaja en este proyecto.

En diciembre de 2016, llega el *Bluetooth* 5.0, aumentando al doble la velocidad de transmisión de datos, cuatro veces el alcance y aportando una mayor capacidad para la emisión de mensajes, además de tener mayor interoperabilidad y coexistencia con otras tecnologías inalámbricas de conexión.

Por último, a principios del año 2019, en enero, llega al mercado el *Bluetooth* 5.1, cuya principal incorporación es que permite conocer la localización de otro dispositivo que esté conectado a la misma tecnología.

2.3. ARQUITECTURA DEL BLE

Esta tecnología está formada por una pila de protocolos compuesta por diferentes capas que interactúan entre sí. Cada capa se encarga de realizar una función. Se pueden diferenciar tres partes diferentes en la pila de protocolos que forma el diseño del sistema BLE (Ilustración 4):

- ❖ Controlador
- ❖ Host
- ❖ Aplicación

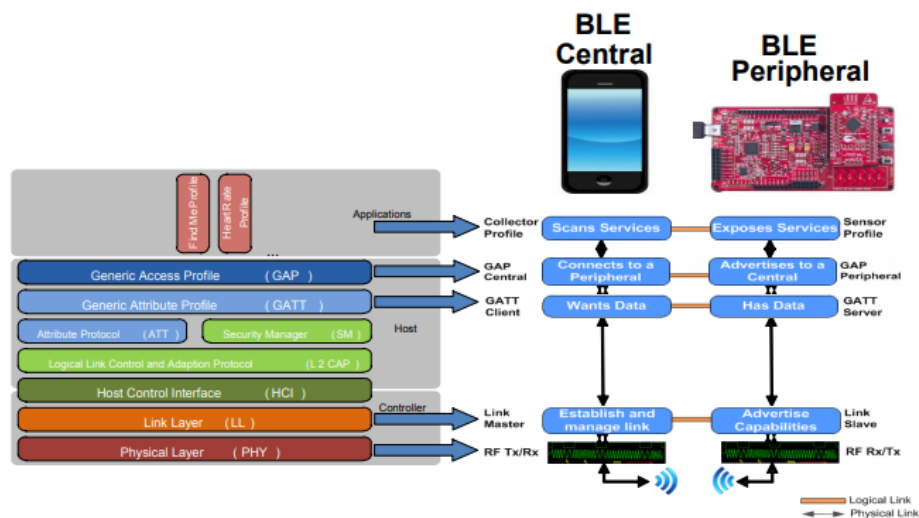


Ilustración 4. Arquitectura BLE

En el Anexo 5 se puede encontrar información más extendida de todas las capas, donde es definido el funcionamiento de cada una de ellas.

Un ejemplo de estos sistemas es *Smart Agriculture Xtreme* (Ilustración 6), sistema que permite monitorear desde las características del clima hasta el seguimiento del crecimiento de las plantas. Permite la conexión de más de 35 sensores que miden diferentes magnitudes como, por ejemplo, temperatura del suelo, aire o plantas, humedad del suelo o aire, viento y presión atmosférica o contenido del agua de riego, entre otras magnitudes.



Ilustración 6. Smart Agriculture Xtreme.

Algunos de los beneficios más importantes que aporta este sistema son:

- ❖ Conocer la cantidad de cada producto que necesita el cultivo en cada momento (Ilustración 7).
- ❖ Decisión del momento específico, tanto para aportar algún producto al cultivo, como para la recogida.
- ❖ Alarmas que avisen de algún peligro para el cultivo o si este ha sufrido algún daño.



Ilustración 7. Niveles de control para el cultivo

La instalación de este sistema de medidas, además de eficiencia puede aportar también comodidad al agricultor, ya que, por ejemplo, se puede instalar un sistema de riego autónomo controlado por un sensor de humedad del suelo.

La comunicación inalámbrica de dicho sistema se realiza a través de *WIFI* y permite tanto recibir los datos de las magnitudes obtenidas, como reprogramar cualquier sensor (Ilustración 8).



Ilustración 8. Diagrama conectividad Smart Agriculture Xtreme

Smart Agriculture Xtreme pertenece a *Libelium*, empresa que nació en 2006 y que hoy día se considera un referente mundial en IoT, cuya sede se encuentra en Zaragoza. Dentro de esta empresa se pueden encontrar varios sistemas de medida inalámbricos para la agricultura, siendo *Smart Agriculture Xtreme* uno de los kits más completos, cuyo precio ronda los 7.350€ en el mercado. Los precios de los demás sistemas se sitúan entre los 1.500€ y los 6.000€, a excepción de *Libelium-SmartEye Agriculture Solution Kit* (Ilustración 9), con características muy parecidas al *Smart Agriculture Xtreme* y cuyo precio se sitúa alrededor de los 7.600€



Ilustración 9. de Libelium-SmartEye Agriculture Solution Kit

En el ámbito de la medicina, IoT está previsto que crezca un 26% en el mercado en 2021. Actualmente, una de las aplicaciones que más está siendo investigada en este ámbito está orientada a las personas mayores, donde se busca la monitorización continua de la salud para poder tener un control total de esta sin que dichas personas tengan que acudir a una consulta. Diagnósticos más exactos, acompañados de tratamientos personalizados para cada paciente, son dos de las cosas que permite conseguir los sistemas de medida en este ámbito.

MySignals (Ilustración 10), es el sistema de medida con conexión inalámbrica más completo dentro de este ámbito, ya que está preparado para obtener 20 parámetros diferentes relacionados con la salud de una persona. A continuación, se nombran los diferentes parámetros:

- Pulso
- Presión arterial
- Niveles de glucosa
- Oxígeno en sangre
- Frecuencia respiratoria
- Electrocardiograma
- Electromiografía muscular
- Capacidad pulmonar
- Ronquidos
- Posición del paciente
- Parámetros de la escala corporal (peso, grasa corporal...)



Ilustración 10. MySignals

MySignals ofrece dos opciones diferentes de conexión, *WIFI* y *Bluetooth Low Energy 4.0*. BLE 4.0 se utiliza para enviar los datos al teléfono móvil, mientras que el *WIFI*

o 4G, se utiliza para subir los datos a la nube, bien directamente desde la central que recoge los diferentes parámetros, o bien desde el dispositivo móvil una vez que se reciban los datos en este (Ilustración 11).



Ilustración 11. Diagrama conectividad MySignals

MySignals, pertenece también a la empresa *Libelium* y su precio en el mercado es de 1.100€. Dentro de *Libelium*, existen 3 kits más de *MySignals* (Ilustración 12) que, junto al comentado anteriormente, todos tienen características más o menos similares y un precio que oscila entre los 995€ y 1.100€.

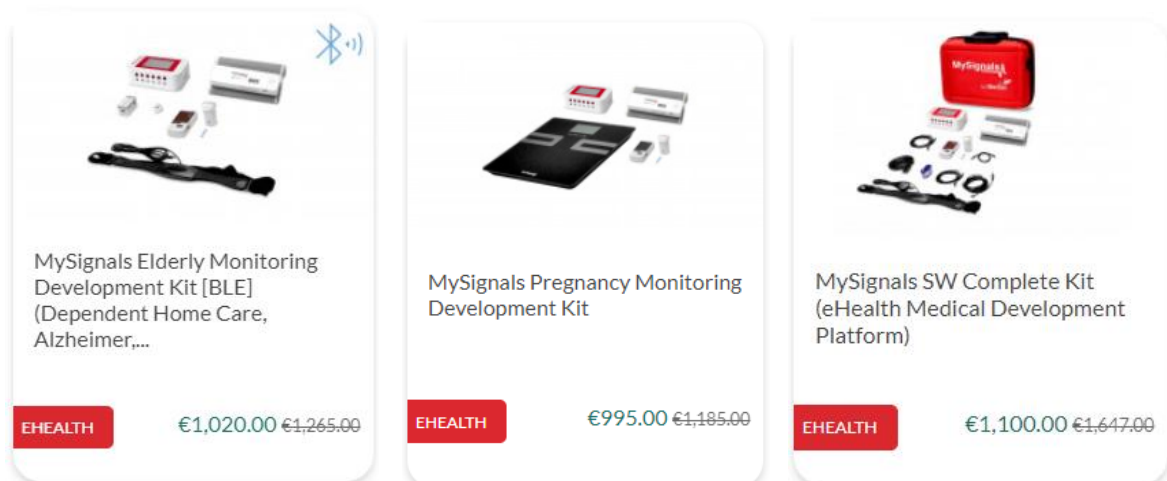


Ilustración 12. Kits MySignals Libelium

Por último, cabe destacar algunos de los siete premios mundiales que ha recibido la plataforma *MySignals* desde su nacimiento en 2016.

- Premio eHealth 2016
- Convocatoria de soluciones finalista (Congreso mundial *Smart City Expo*)
- Premio de aplicación *God Level* de la *Best of Sensor Expo* 2017
- Premios de salud digital otoño 2017

4. DESARROLLO DEL TRABAJO

4.1. INTRODUCCIÓN

Una vez se han explicado todas las partes que se necesitan conocer antes de la producción de este proyecto, se va a proceder a explicar cómo se ha realizado este hasta conseguir los objetivos propuestos.

Para llegar a obtener los objetivos finales propuestos, se ha dividido el proyecto en 5 partes diferentes, comprobando, siempre que es posible, que se está avanzando de forma correcta, para de esta forma tener la posibilidad de detectar errores antes de seguir avanzando de forma errónea. En los próximos apartados se entra más en detalle sobre cada paso seguido.

- ❖ **Parte 1**: Análisis y estudio de la tecnología BLE y el protocolo I2C.
- ❖ **Parte 2**: Diseño y montaje del circuito que incluye los sensores básicos en una placa de prototipos, así como el diseño de este en el programa PSoC Creator. Implementación del código relacionado con los sensores básicos.
- ❖ **Parte 3**: Añadir componente y código relacionado con la conexión BLE al diseño anterior.
- ❖ **Parte 4**: Diseño y desarrollo de la aplicación mediante el programa Android Studio.
- ❖ **Parte 5**: Experimentación con el prototipo y la aplicación implementada, donde se comprueba su correcto funcionamiento.
- ❖ **Parte 6**: Ejecución de una memoria donde se describan los conceptos necesarios para la realización de este proyecto y donde se explique el proceso seguido para el desarrollo de dicho proyecto.

4.2. PRERREQUISITOS

Se van a mencionar las herramientas que se han utilizado para realizar este proyecto:

- ❖ Hardware:
 - PSoC 4 BLE Pioneer Kit.
 - Ordenador con Windows 8.1.
 - Teléfono móvil con Android 9.

- Placa de prototipos y cables para esta placa.
- Pantalla de cristal líquido de 16 caracteres x 2 filas con comunicación mediante I2C
- Sensor DHT22 de temperatura y humedad
- Resistencia de 10 k Ω .

❖ Software:

- PSoC Creator 4.2.
- CySmart Android.
- Android Studio.

4.3. DESARROLLO

4.3.1. Tecnología BLE

Antes de comenzar a desarrollar este proyecto se ha analizado y estudiado todo lo necesario para poder trabajar de forma correcta con la tecnología BLE. La mayor parte de la información obtenida sobre BLE es adquirida del libro *Bluetooth Low Energy: The Developer's Handbook*. Una de las partes que podemos encontrar en este libro es la arquitectura del BLE, de donde se obtiene que existen varias capas que pueden adoptar diferentes roles a lo hora de establecer una conexión. A continuación, se detalla cada uno de los roles que se van a adoptar en cada una de las capas en la conexión establecida en este trabajo.

En la capa de enlace, localizada en el controlador, se encuentran 5 estados diferentes, que dependen del estado en el que se encuentre la conexión. Además, existen dos roles distintos, maestro y esclavo. Sólo puede haber un maestro en la conexión, aunque existe la posibilidad de que exista más de un esclavo (Ilustración 13), el cual, no es el caso de este trabajo. El dispositivo móvil es el maestro de esta conexión, pudiendo establecer conexión con otros dispositivos *bluetooth*, mientras que la tarjeta PSoC es el esclavo, esperando peticiones de conexión.

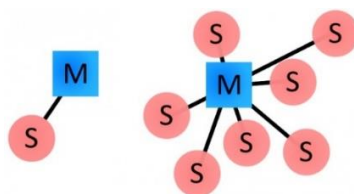


Ilustración 13. Conexión maestro-esclavo

En la capa SMP, localizada en el *host*, también se puede diferenciar entre dos roles, iniciador, que se corresponde con el que actúa como central (capa GAP), en este caso el dispositivo móvil, y el respondedor, que se corresponde con el periférico (capa GAP), es decir, la tarjeta PSoC.

En la capa GATT, perteneciente a la parte del *host*, se encuentran dos roles diferentes, servidor y cliente. En este trabajo el teléfono móvil actúa como cliente, mientras que la placa PSoC actúa como servidor, siendo la encargada de que el cliente tenga los datos a su disposición.

En el caso de la capa GAP, también en el *host* al igual que las dos capas anteriormente mencionadas, caben destacar dos roles, periférico y central. La placa PSoC actúa como periférico, enviando paquetes publicitarios al dispositivo que actúa como central, que en este caso es el dispositivo móvil, para que este inicie la conexión.

4.3.2. Protocolo I2C

I2C (Ilustración 14), basado en circuito integrado, es un protocolo que permite que uno o varios microcontroladores tengan el control de una red de circuitos integrados utilizando un software muy sencillo y solamente dos pines de entrada/salida, que son SDA y SCL. SDA, es el pin por donde se transfieren los datos y va conectado al pin 3.4 de la placa, mientras que SCL se utiliza para sincronizar el sistema a través de pulsos de reloj y va conectado al pin 3.5. Además de estos dos pines, tiene otros dos para VCC y GND, que van conectados al pin VCC +5V y al pin GND de la placa, respectivamente.

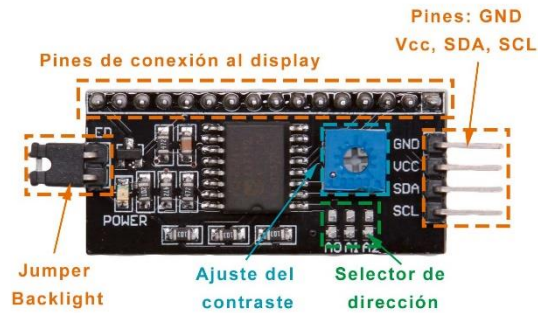


Ilustración 14. Módulo I2C

I2C permite realizar una múltiple conexión de dispositivos que actúen como esclavos y sean controlados por un único maestro que sería el encargado del inicio de todas las comunicaciones. Cada esclavo estaría conectado al maestro a través de dos pines, SDA y SCL (Ilustración 15). El dispositivo que actúa como maestro en este trabajo es la placa PSoC, y como esclavo actúa la pantalla de cristal líquido. Cada uno de los dispositivos que actúen como esclavos tendrán un número de identificación único para que cuando el maestro envíe la información, ésta la reciba el esclavo correspondiente. En el caso de este trabajo la dirección del único esclavo que se tiene es 0x27. Para el uso de I2C con PSoC Creator existen librerías que facilitan el uso de este protocolo.

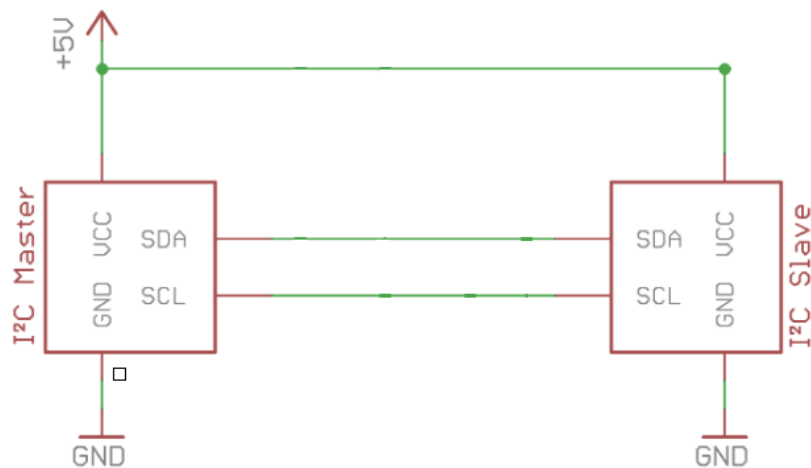


Ilustración 15. Conexión maestro-esclavo

4.3.3. Análisis de los sensores y la pantalla LCD

A continuación, se realiza un análisis para conocer los sensores que se han utilizado en este trabajo y la pantalla LCD.

Sensor de temperatura y humedad DHT22 (Ilustración 16). Este sensor mide magnitudes de temperatura que se encuentren entre -40 y 125 grados centígrados, con una precisión de 0.5 grados, y magnitudes de humedad de valores entre 0 y 100%, con una precisión de 3%.



Ilustración 16. Sensor DHT22

Es un módulo muy barato ya que se puede llegar a encontrar por menos de 2€ en el mercado. El patillaje de este módulo es muy sencillo ya que solo está compuesto por 4 patillas (Tabla 2).

PIN	FUNCIÓN
1	VCC- Fuente de alimentación
2	Datos
3	Nulo
4	GND- Tierra

Tabla 2. Patillaje sensor DHT22

Para su correcto funcionamiento necesita del acompañamiento de una resistencia de 10 kΩ entre sus patillas 1 (VCC) y 2 (Datos). Una vez conectada la resistencia entre estas patillas, la patilla 2 (Datos) se une con el pin 3.0 de la placa PSoC, mientras que el pin de VCC se conecta al pin VCC +5V, además la patilla 4 (GND) se conecta con el pin GND de la placa. Esta conexión se puede observar de manera gráfica en el esquema de la ilustración 21.

El sensor capacitivo (*CapSense*) del BLE Pioneer kit, permite la configuración de varios sensores táctiles o de proximidad. A través de este, se ha desarrollado un sensor de proximidad (Ilustración 17), como tal actúa un cable de proximidad, que incorpora en *Pioneer kit*, que es capaz de medir niveles de proximidad gracias a la configuración del componente gráfico del *CapSense* en PSoC Creator.



Ilustración 17. Sensor proximidad

A continuación, se muestra un esquema del funcionamiento del sensor de proximidad en este trabajo (Ilustración 18).

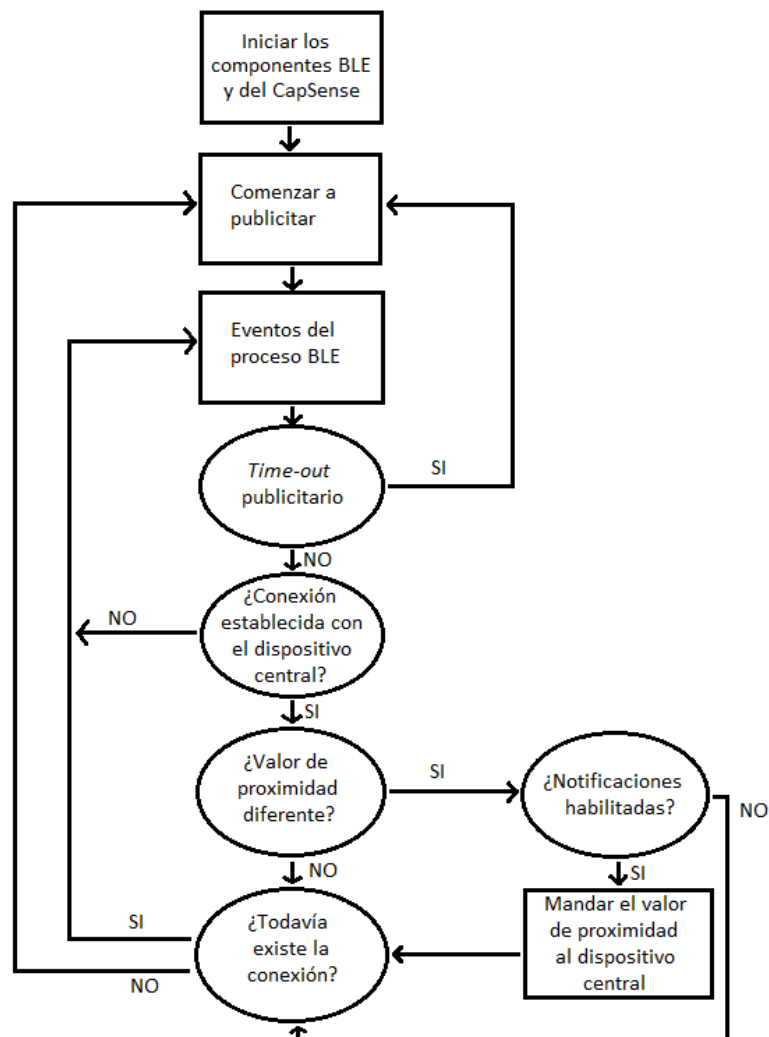


Ilustración 18. Flujo operativo sensor proximidad

El pin de la placa PSoC del Pioneer kit utilizado para el sensor de proximidad es el 2.0 (Ilustración 19), esta información se obtiene del documento *Bluetooth Low Energy: Pioneer Kit Guide*.

Proximity header Pin Map					
Pin #	Description	Pioneer series kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	PROXIMITY	—	P2[0]	P2[0]	P3[7]
2		—	—	—	P3[6]

Ilustración 19. Especificación pin para proximidad

Aparte de estos dos sensores, se ha utilizado una pantalla de cristal líquido (Ilustración 20), donde se muestran los datos que son leídos por ambos sensores. Esta pantalla tiene unas dimensiones de 2 filas con 16 caracteres en cada una de ellas. Además, esta pantalla lleva incorporado el protocolo I2C, explicado en el apartado anterior (Apartado 4.3.2. Protocolo I2C).

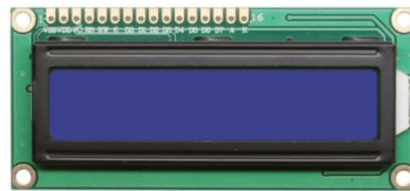


Ilustración 20. Pantalla de cristal líquido

4.3.4. Desarrollo del circuito en la placa de prototipos

Una vez se conoce el funcionamiento de los dispositivos, se procede a implementar el circuito físico de este trabajo en la placa de prototipos. Para esta construcción se han necesitado los siguientes componentes:

- ❖ Placa de prototipos.
- ❖ Sensor de temperatura y humedad DHT22.
- ❖ Resistencia de 10 k Ω .
- ❖ Pantalla de cristal líquido (*Liquid Crystal Display, LCD*).
- ❖ *BLE Pioneer kit* (Placa base, cable de proximidad, cable USB Estándar A-Mini B).
- ❖ Cables dupont macho-macho y macho-hembra.

Antes de comenzar con las conexiones físicas, se hace un diseño previo (Ilustración 21), para que a la hora de montarlo en la placa de prototipos no haya ningún problema. La elaboración del siguiente diseño se ha realizado consultando el documento *Bluetooth Low Energy: Pioneer Kit Guide* para los pines tanto del sensor de proximidad como de la pantalla LCD (Ilustración 22) y la hoja de características del sensor DHT22 para los pines de dicho sensor.

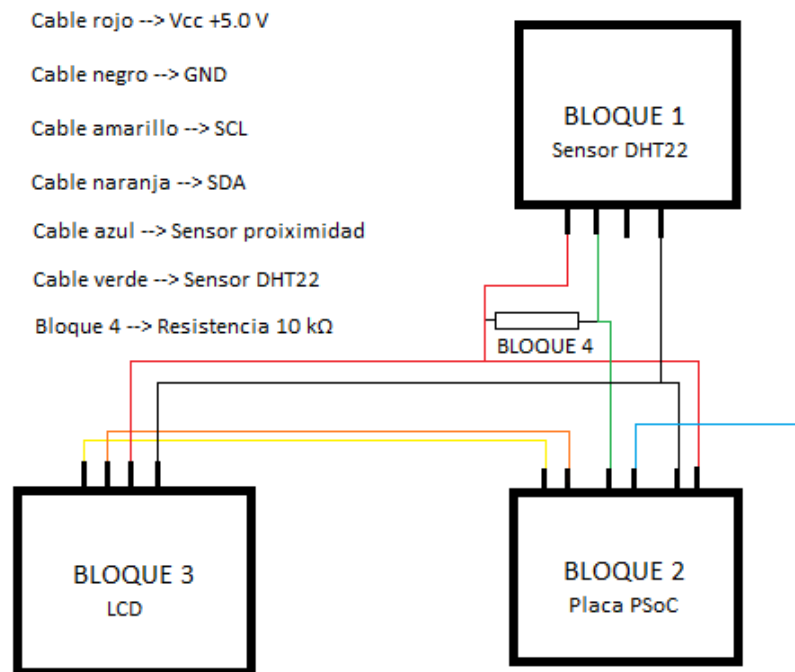


Ilustración 21. Diseño conexiones

Pin #	Arduino Pin	Pioneer Series Kits			
		CY8CKIT-042	CY8CKIT-040	CY8CKIT-042-BLE	CY8CKIT-044
1	D8	P2[6]	P1[4]	P0[5]	P0[2]
2	D9	P3[6]	P1[5]	P0[4]	P0[3]
3	D10	P3[4]	P1[6]	P0[2]	P2[7]
4	D11	P3[0]	P1[1]*	P0[0]	P6[0]
5	D12	P3[1]	P3[1]	P0[1]	P6[1]
6	D13	P0[6]	P1[7]	P0[3]	P6[2]
7	GND	GND	GND	GND	GND
8	AREF	P1[7]	NC	VREF	P1[7]
9	SDA	P4[1]	P1[3]	P3[4]	P4[1]
10	SCL	P4[0]	P1[2]	P3[5]	P4[0]

Ilustración 22. Pines SDA y SCL

Una vez realizado el diseño, se comprueba que no hay ningún error y que se puede montar en la placa sin riesgo de que ningún componente sea dañado (Ilustración 23 y 24).

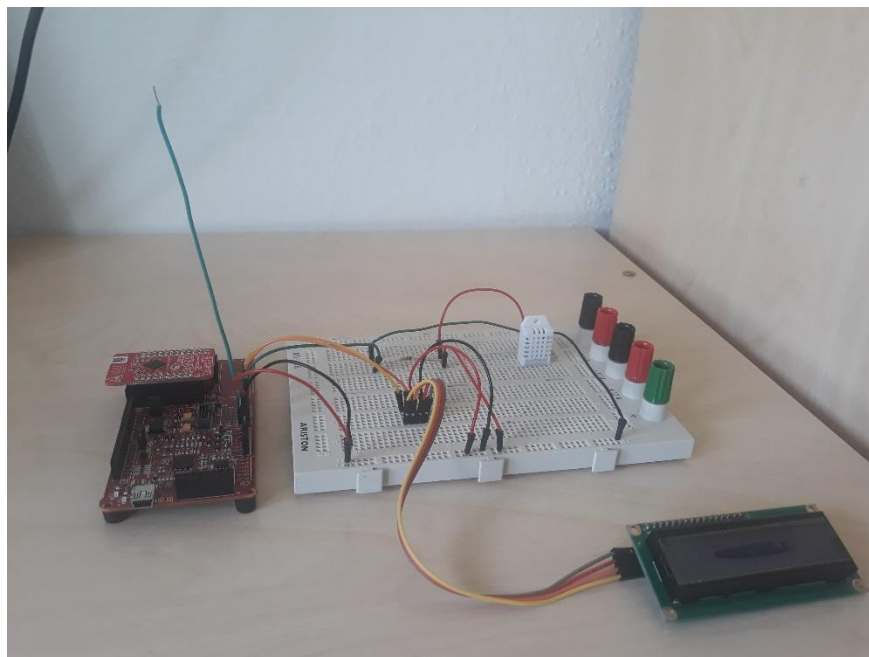


Ilustración 23. Montaje en placa de prototipos (1)

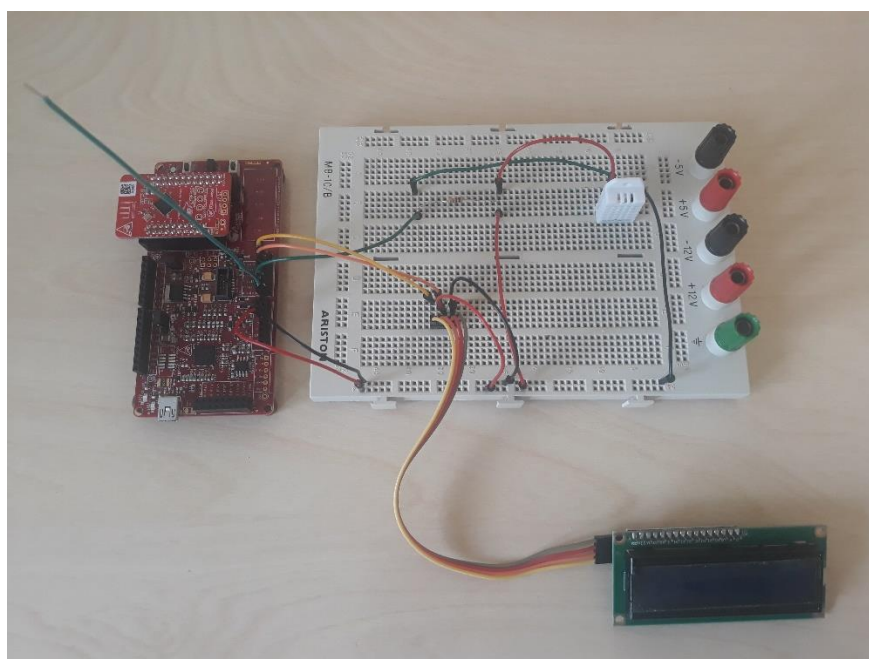


Ilustración 24. Montaje en placa de prototipos (2)

4.3.5. Desarrollo del circuito en PSoC Creator

Una vez se tienen las conexiones físicas tanto del sensor DHT22, como del sensor de proximidad y de la pantalla LCD, el siguiente procedimiento es el diseño de estas conexiones en el software PSoC Creator.

Antes de comenzar a trabajar con este programa, se realiza un análisis de su funcionamiento siguiendo el documento *PSoC Creator: User Guide* que la propia empresa de Cypress, de la cual se habla en el Anexo 4, proporciona en su web oficial. Cuando el funcionamiento básico del programa es entendido, se comienza a desarrollar el diseño.

Los pasos seguidos para construirlo han sido los siguientes:

- ❖ Añadir los componentes necesarios y realizar la configuración de todos estos.
 - Se añade un pin bidireccional que representa al sensor de temperatura y humedad (Ilustración 25).

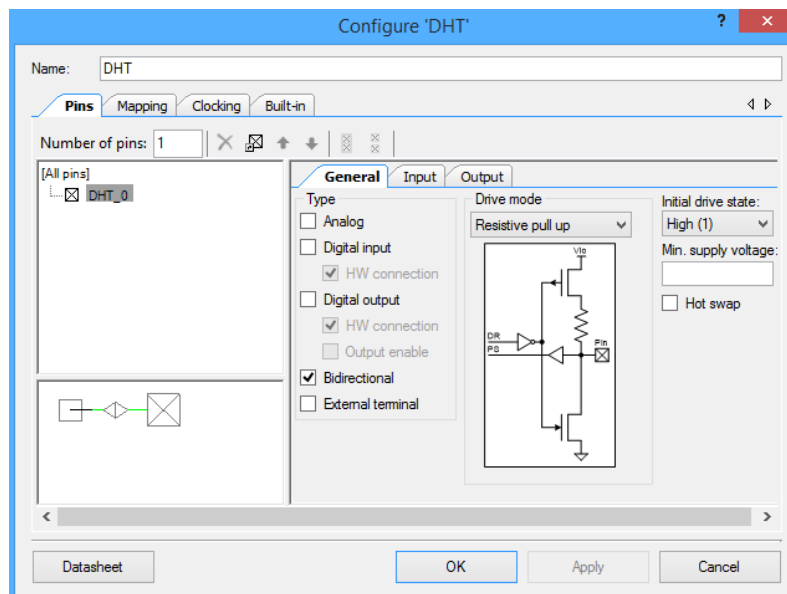


Ilustración 25. Configuración del sensor DHT22

- Se añade el componente CapSense CSD, el cual se configura para que actúe como sensor de proximidad (Ilustración 26).

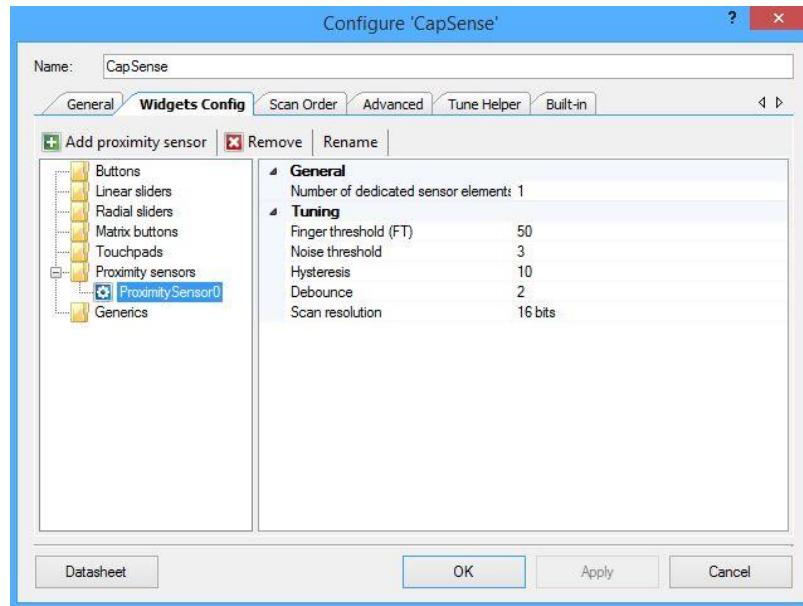


Ilustración 26. Configuración del sensor de proximidad

- Se añade el componente del protocolo I2C, seleccionando el modo maestro y configurando de forma manual el factor de sobremuestreo (Ilustración 27).

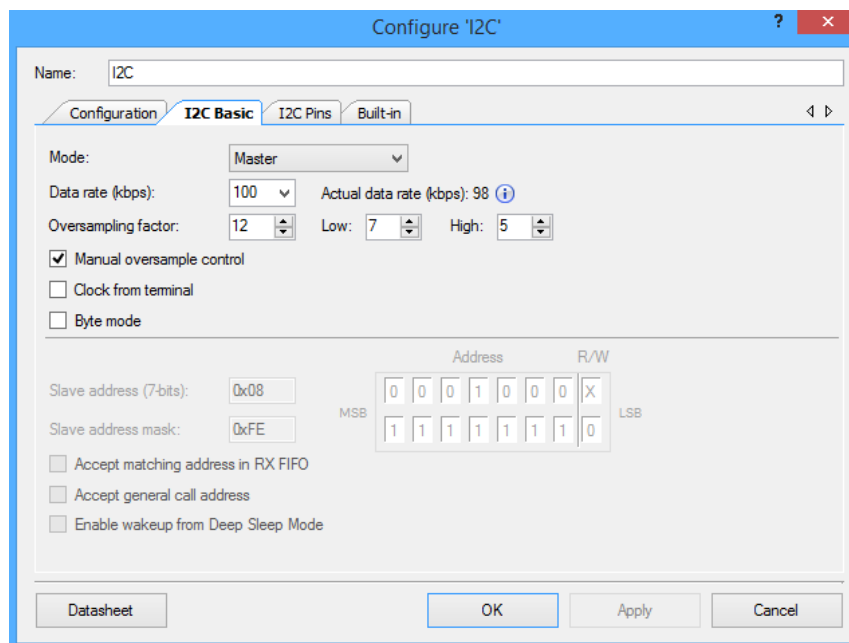


Ilustración 27. Configuración del protocolo I2C

- ❖ Asignar los pines a los componentes previamente incorporados (Ilustración 28 y 29).

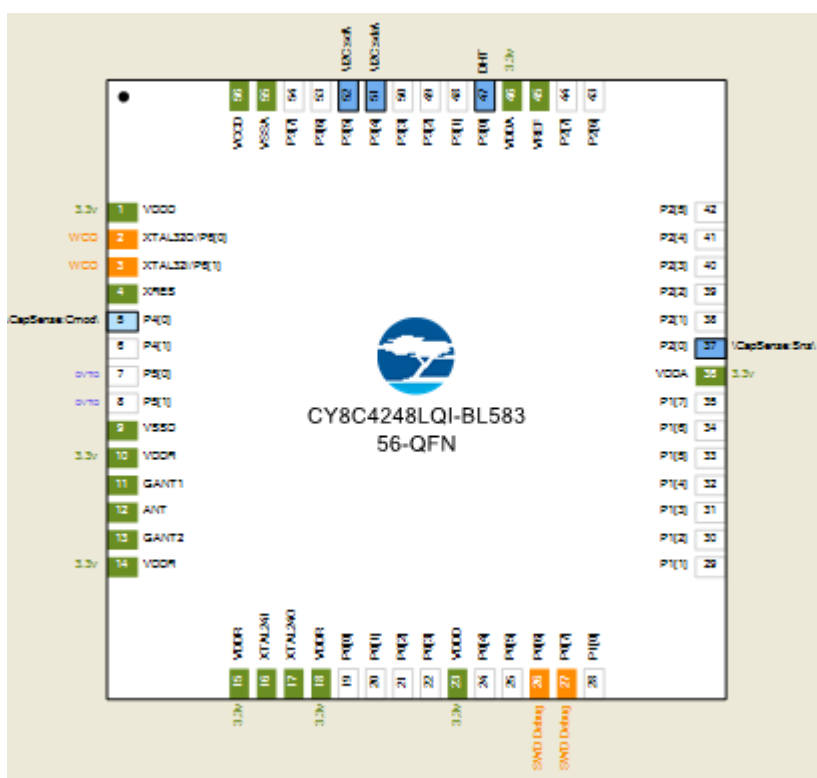


Ilustración 28. Configuración de pines (1)

	Name	Port	Pin	Lock
☐	\CapSense:Cmod\ (Cmod)	P4[0]	5	☐
☐	\CapSense:Sns\ (ProximitySensor0_0__PROX)	P2[0]	37	☒
☐	\I2C:scl\	P3[5]	52	☒
☐	\I2C:sda\	P3[4]	51	☒
☐	DHT	P3[0]	47	☒

Ilustración 29. Configuración de pines (2)

- ❖ Una vez realizada la configuración de los componentes y la asignación de pines, se construye el proyecto y se generan los ficheros necesarios para el correcto funcionamiento de los componentes (Ilustración 30).

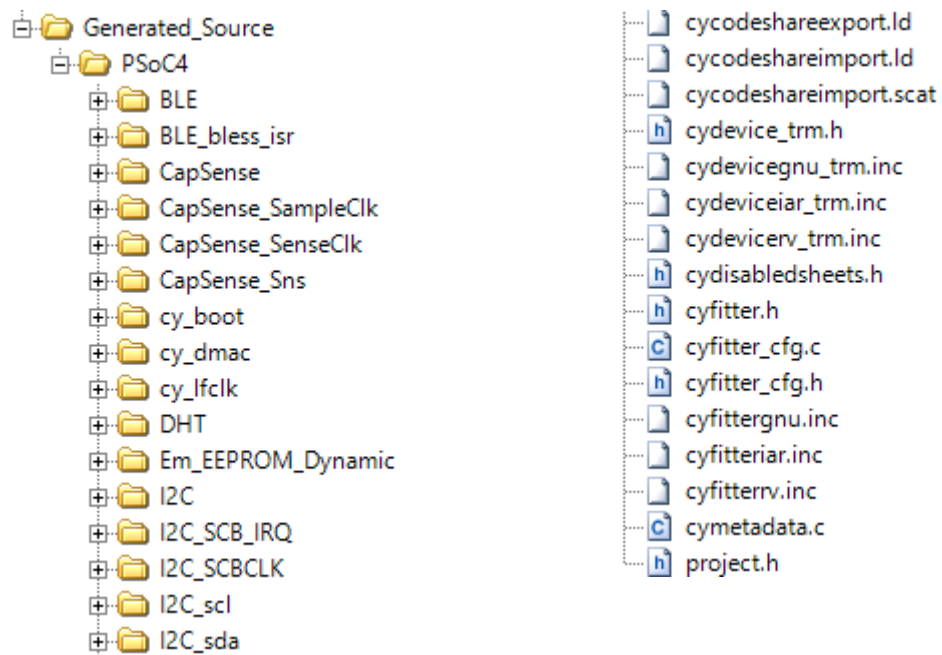


Ilustración 30. Archivos autogenerados al construir el proyecto

- ❖ Desarrollar el código para el funcionamiento de los sensores y la pantalla de cristal líquido. Para el correcto funcionamiento de la LCD se utiliza la librería LiquidCrystal_I2C (Ilustración 31).

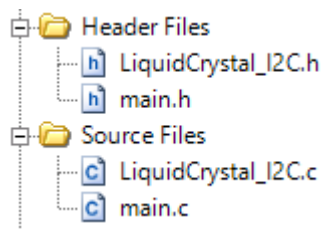


Ilustración 31. Ficheros desarrollados en PSoC Creator

- ❖ Por último, antes de pasar a trabajar con la parte del BLE, se comprueba el correcto funcionamiento de lo desarrollado hasta el momento, es decir, que en la pantalla se muestren correctamente los valores de los sensores.

4.3.6. Configuración del módulo BLE

Lo siguiente es añadir la tecnología BLE al diseño previamente implementado. En primer lugar, en la parte del diseño de PSoC Creator se debe añadir el componente del BLE. Una de las grandes ventajas de la empresa Cypress, la cual se presenta en el Anexo 4, es la simplificación de estándares, en relación a este trabajo, del estándar de BLE de aproximadamente 400 páginas a un componente gráfico visual (Ilustración 32) donde se pueden seleccionar qué parámetros y cómo se quieren enviar.

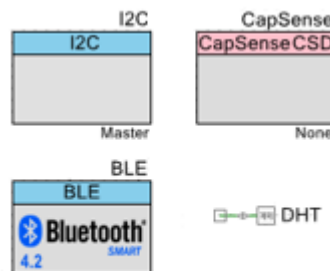


Ilustración 32. Componentes gráficos utilizados para el desarrollo del proyecto (PSoC Creator)

Tras añadir dicho componente, se realiza su configuración (Ilustración 33). Para la realización de dicha configuración se ha seguido el documento *Creating a BLE Custom Profile*, proporcionado por la empresa Cypress. Lo primero que se debe configurar es el tipo de perfil que se quiere utilizar. En segundo lugar, el rol del perfil, es decir, si la tarjeta PSoC va a actuar como servidor o como cliente y, por último, el rol del GAP, si la tarjeta PSoC va a actuar como periférico o central. Estos roles son explicados en el apartado de Arquitectura del BLE.

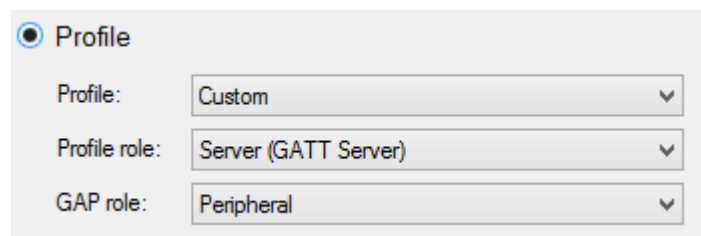


Ilustración 33. Configuración del perfil BLE

Después, se incorporan los servicios y las características con las que va a contar dicha plataforma BLE, además de configurar los ajustes del GAP (Ilustración 34).

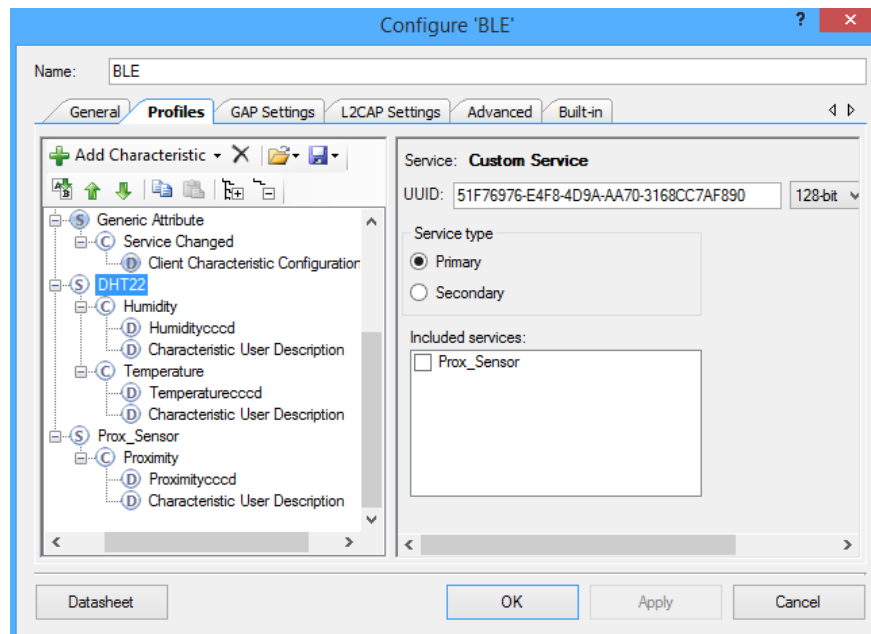


Ilustración 34. Configuración de servicios y características

Cada servicio tiene una UUID diferente y cada característica tiene una UUID relacionada a la de su servicio, pero a la vez diferente de todas las demás.

En este proyecto las UUID, todas de 128 bits, que se han utilizado son:

- ❖ **DHT22:** 51F76976-E4F8-4D9A-AA70-3168CC7AF890
 - **Humedad:** 51F76976-E4F8-4D9A-AA70-3168CC7AF891
 - **Humiditycccd:** 00002902-0000-1000-8000-00805F9B34FB
 - **Temperatura:** 51F76976-E4F8-4D9A-AA70-3168CC7AF892
 - **Temperaturecccd:** 00002902-0000-1000-8001-0805F9B34FB0
- ❖ **Prox_Sensor:** 1C9C78F8-DD31-4408-8CA4-766C8BE7C260
 - **Proximity:** 1C9C78F8-DD31-4408-8CA4-766C8BE7C261
 - **Proximitycccd:** 00002902-0000-1000-8002-00805F9B34FB

Cuando se haya configurado el componente BLE, se procede a la incorporación del código para que se pueda establecer la conexión entre un dispositivo, teléfono móvil en el caso de este proyecto, y la placa base del *BLE Pioneer Kit*. Mediante esta conexión el dispositivo móvil recibe los datos que los sensores detecten.

Para el correcto funcionamiento del BLE se añaden dos archivos a los previamente implementados, “BLE_APP.c” y “BLE_APP.h”. Estos dos archivos definen los eventos del BLE, es decir, que sucede cuando se establece la conexión, cuando se habilita la notificación de una característica o cuando se finaliza la conexión.

Además de estos dos archivos también se modifica el archivo principal, main.c, donde se indican los datos que se deben de enviar a través de la conexión BLE y bajo qué condiciones se tienen que enviar estos datos.

Las condiciones impuestas para la transmisión de los datos es que el cliente active las notificaciones de la característica perteneciente al dato que quiere recibir. Mientras que no se activen estas notificaciones, el cliente no recibe ningún dato a pesar de que los sensores están realizando medidas.

Por otro lado, para reforzar el bajo consumo utilizado para establecer la conexión se ha impuesto también que, aunque el cliente tenga las notificaciones activadas, no se enviará información, si el valor leído por el sensor no difiere del valor que previamente ha sido enviado. De esta forma se evita que se esté continuamente enviando datos a través de la conexión BLE de forma innecesaria.

Para comprobar que los datos leídos se envían a través del BLE, se ha utilizado la aplicación CySmart, que pertenece a la empresa Cypress, comprobando así que existen los dos servicios creados, sensor DHT22 y sensor de proximidad, y que dentro de estos existen las 3 características desarrolladas, humedad, temperatura y proximidad. Además, en cada característica se comprueba que se cumplen las restricciones impuestas de que no se recibe datos hasta que no se activan las notificaciones de esta. Por último, se comprueba que solo hay transmisión de datos cuando el valor leído por el sensor varía del valor previo. Para saber que los datos recibidos en la aplicación móvil son los valores reales que miden los sensores, se realiza una comparación con los valores que se muestran en la pantalla de cristal líquido.

4.3.7. Desarrollo de la aplicación móvil

Una vez se termina de crear la plataforma BLE en PSoC Creator, se comienza a desarrollar la aplicación para Android con la que el cliente va a establecer la conexión BLE. Antes de comenzar a desarrollar la aplicación, se realiza un estudio previo de cómo funciona Android Studio consultando el libro *Android Studio IDE Quick Reference: A Pocket Guide to Android Studio Development*. Una vez se entiende el funcionamiento del *software*, se pasa a desarrollar la aplicación.

En primer lugar, se desarrolla el código que define la parte visible de la aplicación, es decir, la estética que va a tener cuando el cliente la abra en su dispositivo. Esta parte se programa con lenguaje XML.

Una vez se define la estética de la aplicación, se desarrolla la parte que hace posible el correcto funcionamiento de esta. Para el desarrollo de esta parte se utiliza el lenguaje de programación JavaScript.

Al igual que en la aplicación CySmart, al abrir la aplicación móvil, se comprueba que el *bluetooth* del dispositivo móvil está conectado. Si es así, la aplicación comienza un escaneo de dispositivos BLE automáticamente. Si se detecta que está desconectado, salta un cuadro informativo que te sugiere que lo habilites (Ilustración 35), en el momento en el que aceptas, comienza el escaneo. En la parte superior derecha de la pantalla principal hay un botón para comenzar el escaneo, por si en una primera instancia no se detecta el dispositivo buscado. Una vez que aparezca el nombre y la dirección del dispositivo buscado en la lista de dispositivos encontrados, se pulsa sobre este para establecer la conexión (Ilustración 36).

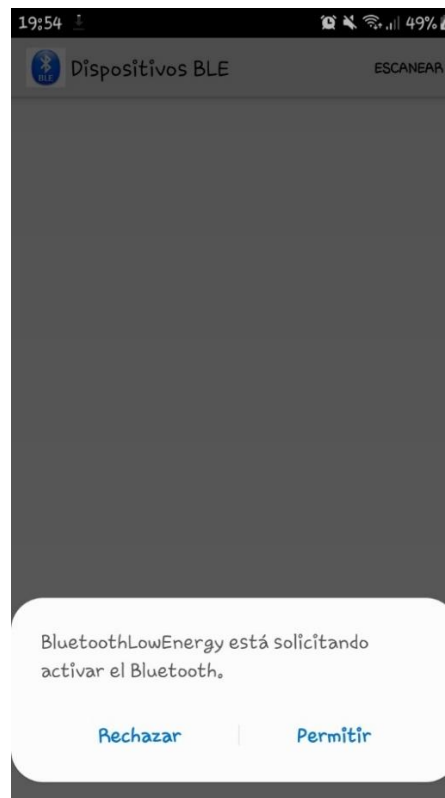


Ilustración 35. Petición para activar Bluetooth

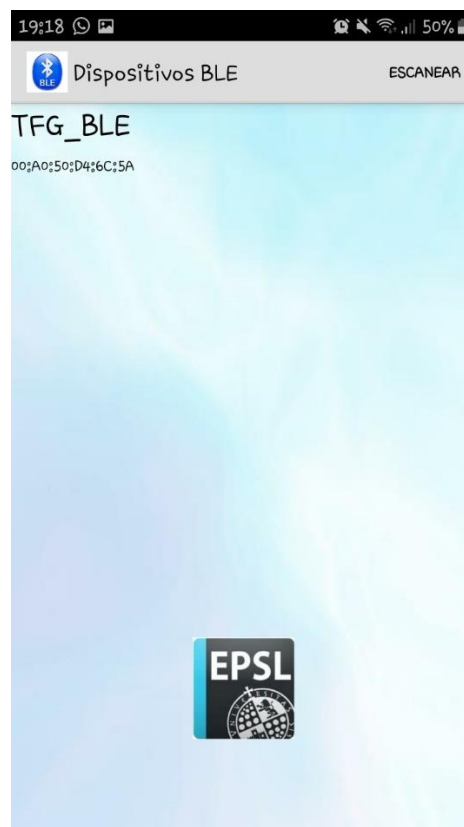


Ilustración 36. Pantalla principal de la aplicación

Una vez se establece la conexión se muestran, en una segunda pantalla (Ilustración 37), los servicios disponibles, así como un cuadro de texto con la fecha y hora actual y otro cuadro de texto donde se muestran los datos.

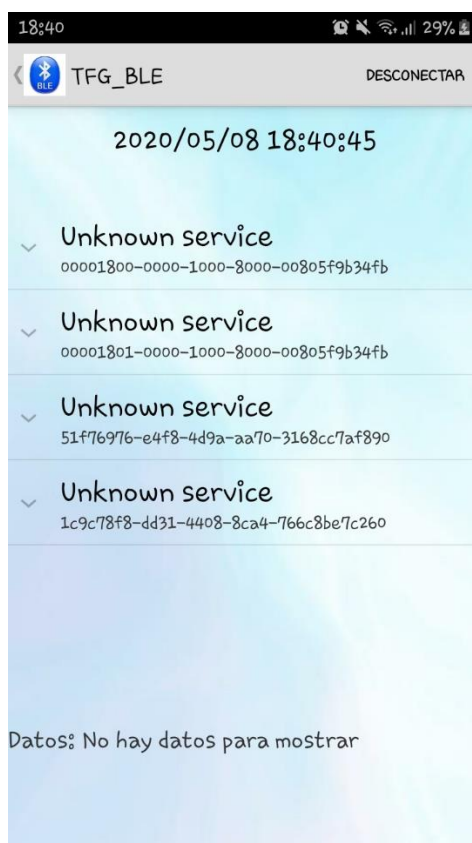


Ilustración 37. Pantalla secundaria aplicación

En este trabajo hay un total de 4 servicios, los dos primeros, muestran información sobre el dispositivo, mientras que los dos siguientes son los servicios del sensor DHT22 y sensor de proximidad, respectivamente. Dentro de los dos últimos servicios se puede descubrir las características que tiene cada uno, para habilitar las notificaciones de cualquier característica solamente hay que pulsar sobre ella, si se pulsa una vez se deshabilitan las notificaciones de la característica que haya activada, en caso de pulsar sobre la característica que tiene las notificaciones activadas, no sucede nada. Cuando se pulsa la segunda vez se activan las notificaciones de esa característica y se obtienen los datos de esta. A continuación, se muestran imágenes de los datos obtenidos tanto en la aplicación, como en la pantalla (Ilustración 38, 39 y 40).

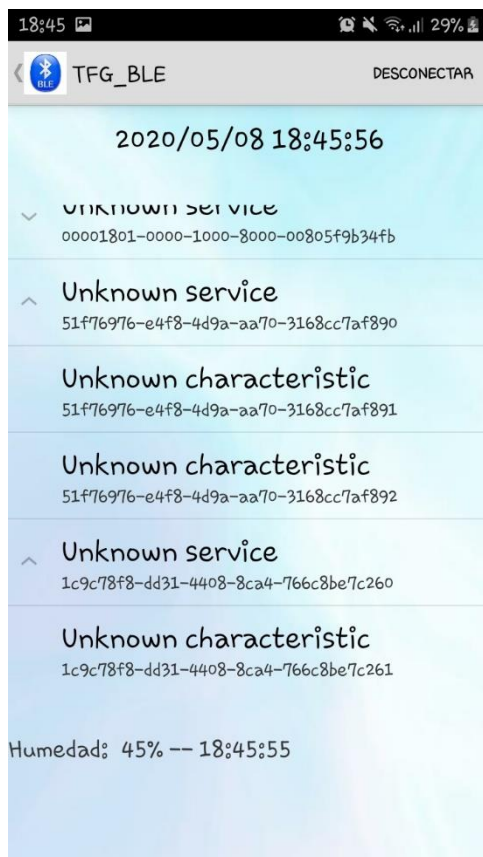


Ilustración 38. Medida humedad

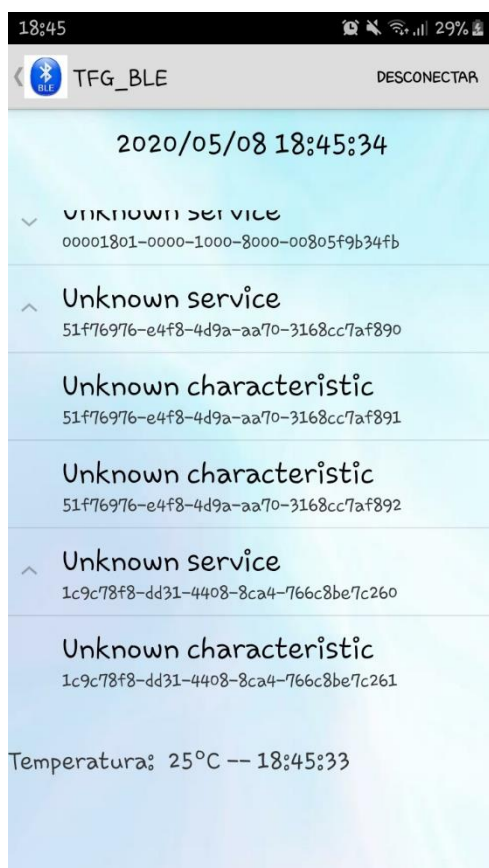


Ilustración 39. Medida temperatura

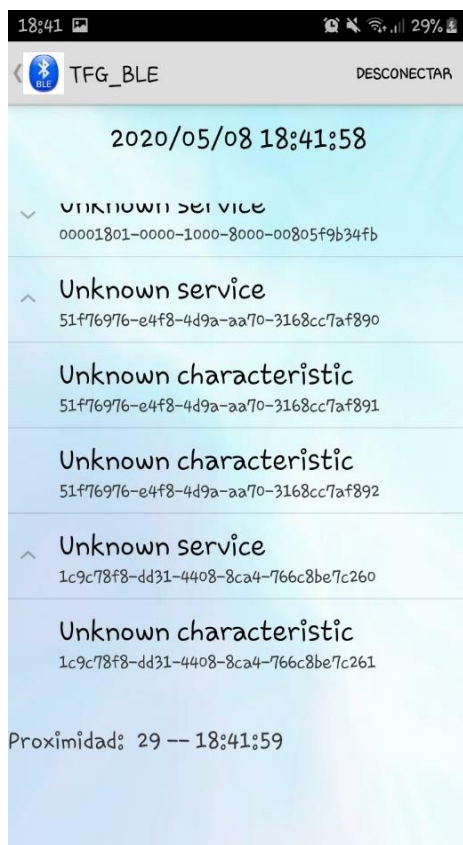


Ilustración 40. Medida proximidad

5. CONCLUSIONES Y LÍNEAS FUTURAS

Las conclusiones a las que se ha llegado después de haber finalizado dicho proyecto con éxito han sido valoradas positivamente, primero por ver que los resultados obtenidos han sido satisfactorios, llegando a cumplir todos y cada uno de los objetivos propuestos para este trabajo, y, en segundo lugar, por todos los conocimientos nuevos adquiridos para poder haber llegado a obtener dichos resultados.

En primer lugar, se ha obtenido nuevos conocimientos referidos a la tecnología *Bluetooth Low Energy*, una tecnología de la que nunca había oído hablar y, por lo tanto, nunca había trabajado con ella y he de decir que me ha encantado poder manejarla e investigar sobre ella, puesto que me ha parecido una tecnología bastante completa y útil, sobre todo por su bajo consumo de energía, siendo esta la principal característica de esta. Además, es una tecnología que en la actualidad ya es una realidad, puesto que, aunque ya existe un amplio repertorio de dispositivos que interactúan entre sí a través de esta, el crecimiento de estos dispositivos año tras año es bastante grande.

En segundo lugar, los nuevos conocimientos adquiridos relacionados con la empresa Cypress, más en concreto con algunos productos suyos como son las diferentes familias de microcontroladores, especialmente la familia que se utiliza en el desarrollo de este proyecto que ha sido PSoC 4, más concretamente el módulo PSoC 4200 BLE. A parte de los productos, también se han adquirido nuevos conceptos referidos a programas o aplicaciones de Cypress como son PSoC Creator y CySmart. PSoC Creator utilizado para el diseño de la plataforma BLE, en el cual gracias a los diferentes componentes gráficos que incorporan, se puede realizar una programación más sencilla y eficaz que en otros desarrolladores, y la aplicación CySmart que es de gran ayuda a la hora de realizar un proyecto de este tipo ya que en esta se puede comprobar el correcto funcionamiento del diseño que se va desarrollando en PSoC Creator.

En tercer lugar, se han adquirido también conocimientos sobre Android como han sido las herramientas de desarrollador (SDK) o la construcción de un archivo APK para poder ser utilizado en el dispositivo, además de la implementación del código relacionado con la aplicación, utilizando el lenguaje de programación JavaScript, en un IDE tan popular como Android Studio, IDE oficial para programar en Android.

A través de estos conocimientos necesarios para el desarrollo de este proyecto, siendo la mayoría nuevos para mí, se ha conseguido finalidades necesarias como la creación de una aplicación móvil personalizada para lo que el proyecto requería o la construcción, acompañado de un correcto funcionamiento de una plataforma. Por este motivo estoy muy satisfecho con el trabajo que se ha realizado, ya que he podido superar las adversidades encontradas durante el camino, que no han sido pocas, para poder llegar a cumplir con éxito los objetivos propuestos.

A continuación, se comentan las posibles líneas futuras de este proyecto. En primer lugar, la utilización de sensores más complejos o la creación de una plataforma con un grupo de sensores, todos ellos orientados al mismo ámbito. Un ejemplo de esta idea podría ser un conjunto de sensores relacionados con las constantes vitales de una persona, donde se podrían incluir sensores de pulso, presión arterial, frecuencia respiratoria o temperatura corporal.

Cada uno de los valores obtenidos es representado por una característica a la hora de configurar el BLE, y cada característica tiene diferentes propiedades para trabajar con ella. Otra de las aplicaciones futuras a este proyecto es la utilización de otras propiedades, ya que en este proyecto las dos que se han utilizado son leer y notificar. En la parte del anexo 4, en la página 70, se pueden observar el resto de propiedades disponibles y que en un futuro proyecto se podrían incluir.

Por último, otra aplicación futura sería mostrar en la aplicación móvil todos los datos, que lean los sensores que formen la plataforma, al mismo tiempo. Así, se puede tener un mayor control de todos los valores.

ANEXO 1. BLE PIONEER KIT

Componentes del BLE Pioneer Kit (Ilustración 41 y

Tabla 3)

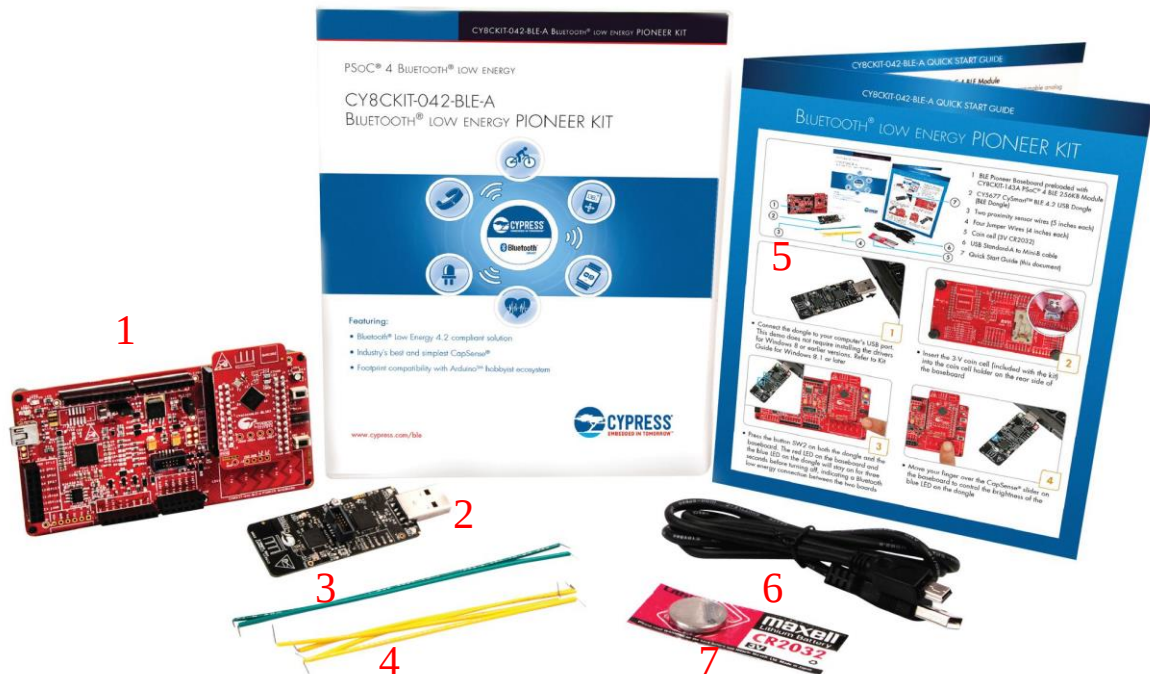


Ilustración 41. BLE Pioneer kit

1	Placa base del BLE Pioneer kit, con módulo CY8CKIT-143A PSOC 4 BLE 256 KB
2	Dongle BLE. USB dongle CY5677 CySmart BLE 4.2
3	2 cable de sensor de proximidad (5 pulgadas cada uno)
4	4 cables de puente (4 pulgadas cada uno)
5	Guía de comienzo rápida
6	Cable USB estándar-A a mini-B
7	Pila de moneda (3V CR2032)

Tabla 3. Componentes BLE Pioneer kit

Pines placa BLE Pioneer kit (Ilustración 42)

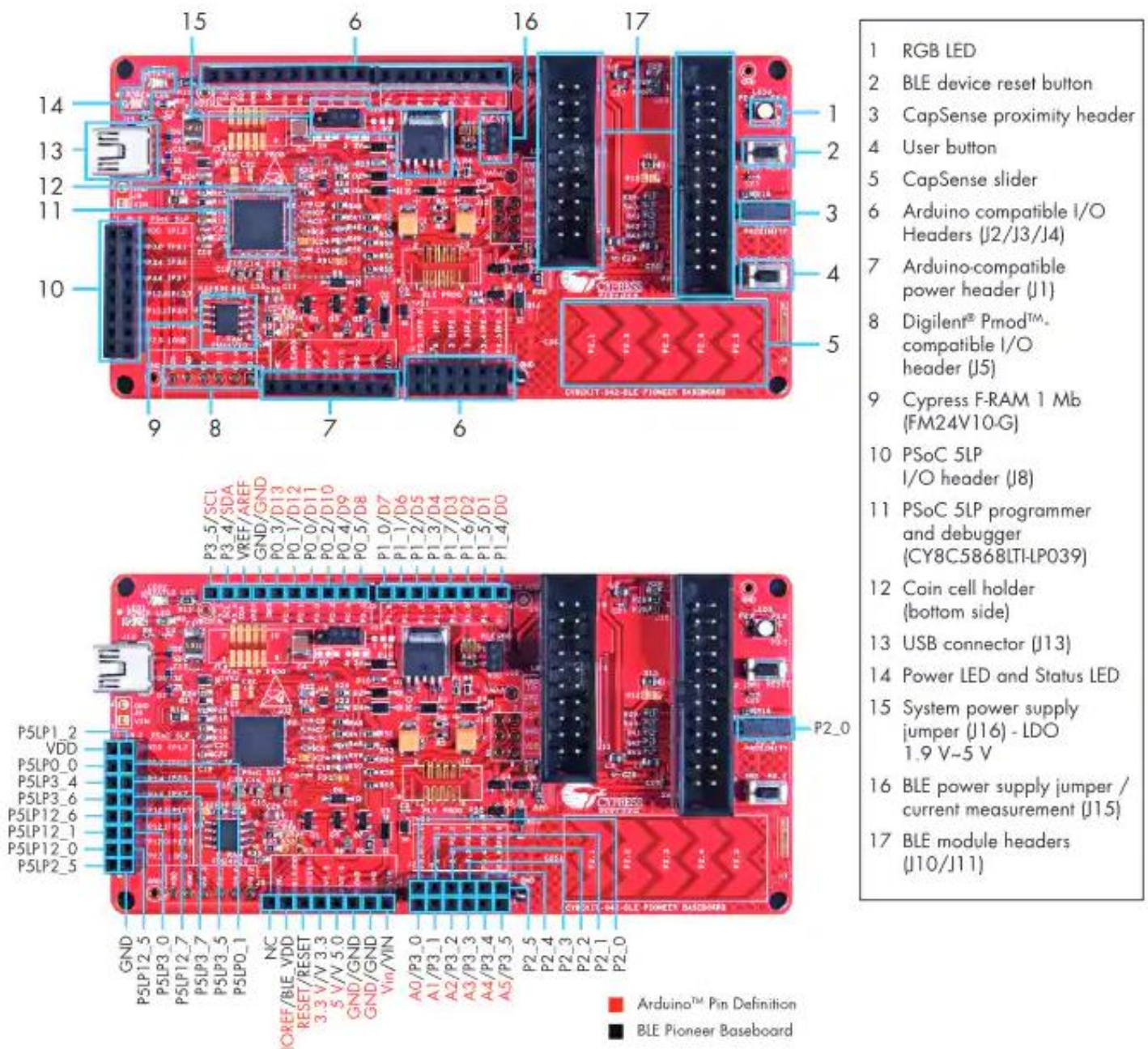


Ilustración 42. Mapa de pines placa BLE Pioneer kit

Presentación módulo CY8CKIT-143A PSoC 4 BLE

256 KB (Ilustración 43)

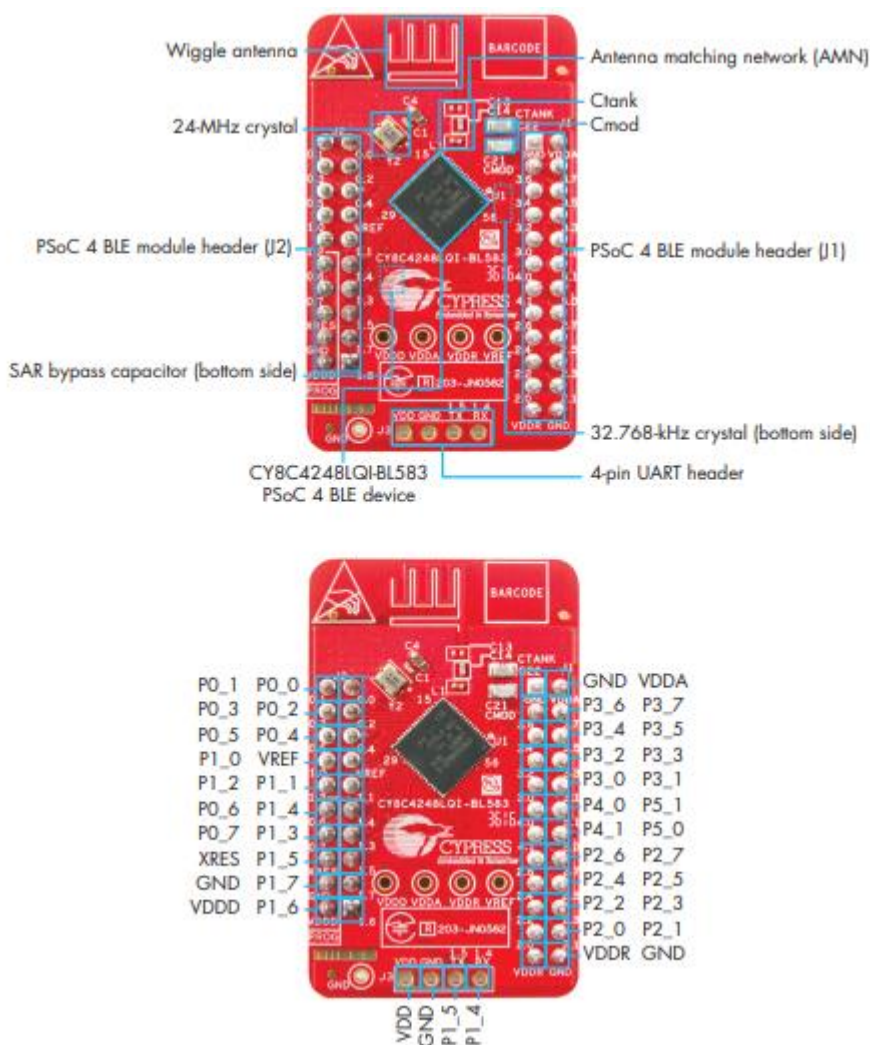


Ilustración 43. Módulo CY8CKIT-143A

Especificaciones del microcontrolador

CY8C4248LQI-BL583 (Tabla 4)

CY8C4248LQI-BL583	
Núcleo CPU	ARM Cortex M0
CapSense	Si
ADC dedicado	SAR (1, 12-bit @ 1 msp/s)
EEPROM (KB)	0
Flash (KB)	256
Conductor directo LCD	Si
Frecuencia máxima para trabajar (MHz)	48
Temperatura máxima para trabajar (°C)	85
Temperatura mínima para trabajar (°C)	-40
Voltaje máxima para trabajar (V)	5.5
Voltaje mínimo para trabajar (V)	1.9
Número controladores CAN	0
Número bloques de tiempo continuo	2
Número canales DMA	8
Número de comparadores	2
Número de bloques de tiempo/contadores/PWM	4
Número de GPIOs	36
Número de amplificadores operacionales	4
Número de SIO	0
Número de bloques de comunicación serial	2
Número de salidas/entradas USB	0
Número de bloques analógicos universales	0
Número de bloques digitales universales	4
PLL	No
SRAM (KB)	32
Serie	PSoC 4200BL
Tipo USB	Ninguno

Tabla 4. Especificaciones CY8C4248LQI-BL583

ANEXO 2. SENSOR DE TEMPERATURA Y HUMEDAD DHT22

Hoja de características del sensor DHT22

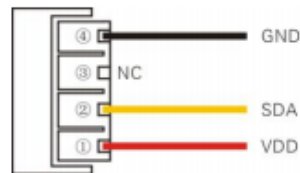
(Ilustración 44 y 45)

、 The definition of single-bus interface

4.1 AM2302 Pin assignments

Table 1: AM2302 Pin assignments

Pin	Name	Description
①	VDD	Power (3.3V–5.5V)
②	SDA	Serial data, bidirectional port
③	NC	Empty
④	GND	Ground



PIC1: AM2302 Pin Assignment

4.2 Power supply pins (VDD GND)

AM2302 supply voltage range 3.3V – 5.5V, recommended supply voltage is 5V.

4.3 Serial data (SDA)

SDA pin is tri structure for reading, writing sensor data. Specific communication timing, see the detailed description of the communication protocol.

、 Sensor performance

5.1 Relative humidity

Table 2: AM2302 Relative humidity performance table

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		%RH
Range		0		99.9	%RH
Accuracy ^[1]	25°C		± 2		%RH
Repeatability			± 0.3		%RH
Exchange		Completely interchangeable			
Response ^[2]	1/e(63%)		<5		S
Sluggish			<0.3		%RH
Drift ^[3]	Typical		<0.5		%RH/yr

5.2 Temperature

Table 3: AM2302 Relative temperature performance

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		°C
n			16		bit
Accuracy			± 0.5	± 1	°C
Range		–40		80	°C
Repeat			± 0.2		°C
Exchange		Completely interchangeable			
Response	1/e(63%)		<10		S
Drift			± 0.3		°C/yr

Ilustración 44. Hoja de características DHT22

Electrical Characteristics:

Items	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	6	V
Current supply	Measuring	1		1.5	mA
	Stand-by	40	Null	50	uA
Collecting period	Second		2		Second

Ilustración 45. Características eléctricas DHT22

ANEXO 3. CYPRESS

Cypress (Ilustración 46) es una empresa americana cuya oficina central está en San José, California y fue una de las primeras empresas en trabajar con la tecnología BLE.



Ilustración 46. Cypress

Un grupo de ingenieros empezaron a desarrollar un día la tecnología PSoC, el responsable de la empresa Cypress se interesó por esta tecnología, hasta llegar a invertir en ella para que esta se incubara dentro de sus filas. Cuando esta tecnología avanzó en su desarrollo, Cypress realiza un *spin off* y la adquiere para convertirla en una unidad de negocios, cuya sede principal está en Washington. Es una de las empresas que resisten en silicon valley.

Hasta el año 2015 la empresa Cypress contaban con el control de PSoC y Capsense. En ese mismo año se unió a la empresa Spansion, la cual desarrollaba y fabricaba memorias flash y microcontroladores, entre otros productos, y adquirió en 2013 el negocio de microcontroladores y sistemas analógicos que incluía en sus filas la empresa japonesa Fujitsu. Con esta unión Cypress se convierte en una de las empresas más grandes a nivel mundial de semiconductores y microcontroladores.

Por último, en junio de 2019, la empresa *Infineon* con sede en Munich, catalogada como la segunda fabricante de *chips* en Europa, firma un acuerdo de compra de la empresa Cypress.

PSOC

La característica principal que destaca de PSoC es que se pueden generar diferentes funciones a través de hardware con un código, en lenguaje de programación C, mínimo, que aporta una solución más eficaz que en el caso de hacerse a través de software. Debido al poco código necesario se gana espacio-tiempo y se pueden desarrollar muchas funciones sin

demandar al procesamiento del CPU, dejando libre este para las cosas de mayor importancia. A nivel hardware se pueden hacer implementaciones que de otra manera serían complicadas. PSoC es la culpable de que Cypress sea el líder mundial de sensores capacitores, esto se debe a las propiedades de este y a su arquitectura que es diferente a cualquier microcontrolador.

Dentro de PSoC se pueden encontrar diferentes familias. PSoC 1 fue el primer sistema analógico programable que tenía un rendimiento óptimo, ya que varias empresas habían realizado anteriormente algún intento, pero estos eran demasiado restringidos y lentos. Posteriormente PSoC lanza las familias PSoC 3, PSoC 4 y PSoC 5, tanto la familia 3, como la 5 se tardaron muchos años en desarrollarlas, cuidando todos los detalles al máximo y cuando estas llegaron al mercado sus precios eran muy altos debido a que eran muy adelantadas para lo que se podía encontrar en el mercado en ese momento. Con PSoC 4 se fue buscando que fuera adaptable para las necesidades de todo el mundo, sacando así al mercado varias familias dentro de la familia de PSoC 4 como son la 4000, 4100, 4200, L, M o BLE. Gracias a esta gran variedad PSoC permite cubrir la necesidad de cualquier cliente. Por último, PSoC ha lanzado la familia de PSoC 6, mejorando algunos aspectos de las familias anteriores, en los cuales no se va a entrar en detalle puesto que no es la familia utilizada para el desarrollo de este proyecto.

A la hora de programar, cada USB tiene sus propios registros, porque es un pequeño coprocesador. Esta es la parte en la que más diferencia se genera entre PSoC y otro microcontrolador. La otra parte es la analógica, que se encuentra muy por encima de los otros microcontroladores que se pueden encontrar en el mercado.

Algunas características de este módulo son:

PSoC 4 BLE

Este dispositivo que proviene de Cypress, es un sistema de chip programable integrado que incorpora una radio BLE, además de periféricos tanto analógicos como digitales y un microcontrolador Arm Cortex-M0, CY8C4248LQI-BL583 (Ilustración 47), mencionado y descrito en el apartado 3, en un solo chip. Los periféricos analógicos y digitales aportan flexibilidad al diseño a través del software PSoC Creator, programa utilizado para el desarrollo de aplicaciones BLE utilizando el dispositivo PSoC 4.

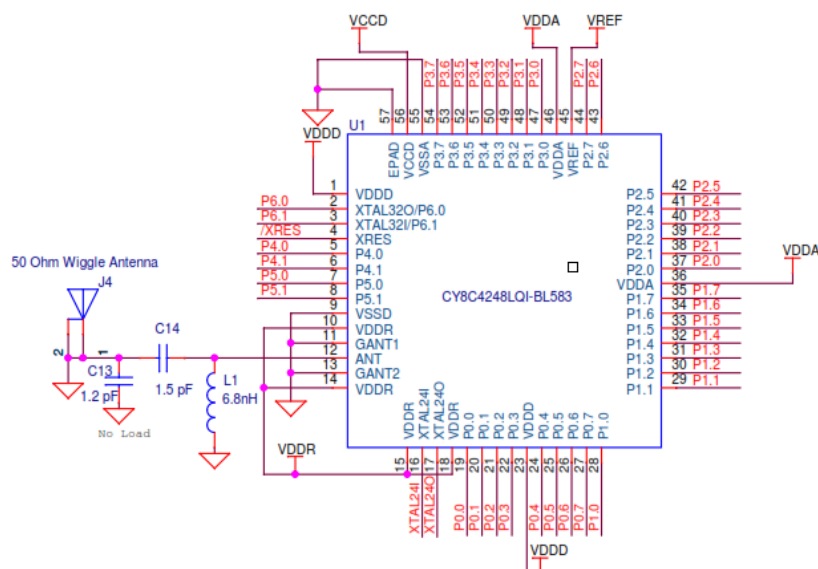


Ilustración 47. Microcontrolador CY8C4248LQI-BL583

Como se ha mencionado en la introducción de Cypress, esta empresa aporta una interfaz gráfica de usuario que permite una fácil configuración de los componentes minimizando la complejidad del protocolo.

Características del módulo PSoC 4 BLE (Ilustración 48):

- ❖ Tiene una CPU Arm Cortex-M0 alcanzando una frecuencia máxima de operación de 48 MHz y tiene hasta 256 KB de memoria flash que incluye un acelerador de lectura y 32 KB de SRAM.
- ❖ Controlador *CapSense* con *SmartSense* con capacidad de panel táctil.
- ❖ Interfaz analógica programable (AFE) formada por cuatro amplificadores operacionales que pueden ser configurados como comparadores, filtros o PGA entre otros.
- ❖ Cuatro bloques digitales universales que hacen referencia a los periféricos digitales. Cuatro bloques TCPWM compuestos por un temporizador de 16 bits, contador o PWM y dos bloques de comunicación serie configurables.
- ❖ Radio BLE de 2.4 GHz para la conexión *Bluetooth Smart* con *Bluetooth 4.2* y un controlador compatible con este.
- ❖ Sus dimensiones son muy pequeñas, 10mm x 10mm x 1.8mm.
- ❖ Módulos de BLE destinados a aplicaciones de conectividad y compatibles con el IDE PSoC Creator.

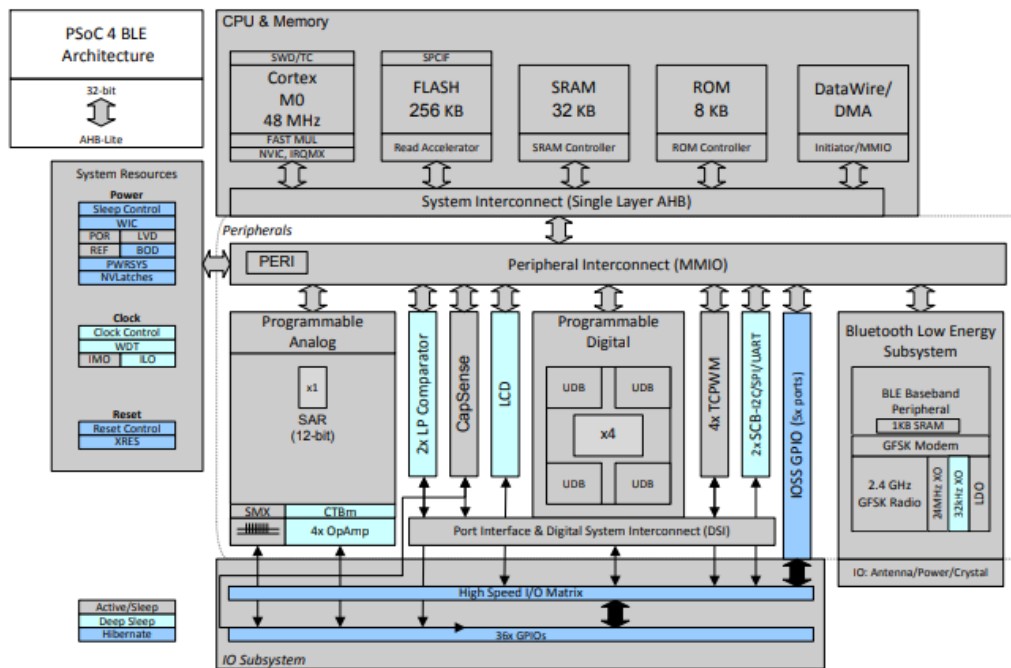


Ilustración 48. Módulo PSoC 4 BLE

PSoC Creator

PSoC Creator es un entorno de diseño integrado (IDE). Este es el programa para realizar diseños de hardware y software para la familia de PSoC 4, con componente de PSoC que ayudan a crear diseños de hardware por el método de esquemas clásicos.

Dentro de este software hay que diferenciar dos contenedores diferentes donde se almacenará información:

- ❖ **Espacio de trabajo (Workspace, en inglés):** Es el contenedor de mayor nivel y en su interior es capaz de almacenar varios proyectos juntos. El formato en el que se guarda este contenedor es .cywrk.
- ❖ **Proyecto (Project, en inglés):** Un proyecto almacena en su interior los diferentes archivos hexadecimales, el diseño, el esquema del proyecto y los ficheros con los códigos fuente. Un proyecto siempre va a formar parte de un espacio de trabajo. Además, dentro de un proyecto se pueden encontrar dos tipos diferentes:

- **Proyecto de diseño:** Se compone de un esquema formado por componentes que posteriormente son configurados. Además, se realiza la configuración de los recursos del diseño y se escriben los códigos en C para su aplicación.

Una vez realizado todo lo anterior se construye el proyecto, creando así los archivos hexadecimales y archivos del proyecto, por último, se ejecuta para programar el dispositivo BLE.

- **Proyecto de biblioteca:** Es el conjunto de uno o varios componentes y su código asociado. Estos proyectos se podrán guardar en las bibliotecas para posteriormente poder utilizarlo en algún diseño.

Un proyecto es una mezcla de los dos tipos de proyectos que se acaban de definir. Los proyectos van a estar guardados con una extensión .cyprj como por ejemplo NombreProyecto.cyprj, en la que los archivos del proyecto deben estar siempre ubicados en un directorio llamado NombreDirectorio.cydsn (diseño) o NombreDirectorio.cylib (biblioteca). Por último, se diferencian las distintas partes de las que consta PSoC Creator (Ilustración 49):

1. Esquema diseñado con los componentes de PSoC Creator.
2. Ficheros para la programación en c del firmware.
3. Configuración personalizada de cada componente.
4. Acceso a las hojas de características de cada componente.
5. Listado de todos los componentes de los que se dispone.

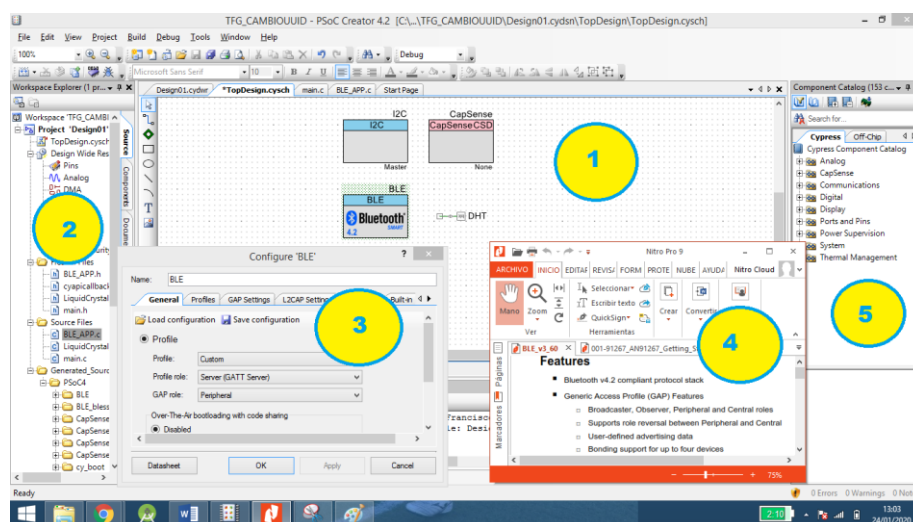


Ilustración 49. Partes del software PSoC Creator

ANEXO 4. *BLUETOOTH LOW ENERGY*

CARACTERÍSTICAS DEL BLE

La versión *Bluetooth* 4.2 tiene una velocidad de datos de 1 Mbps, la cual hace que el rendimiento de datos sea limitado. BLE no está diseñado para largo alcance, en caso de utilizar esta función la velocidad disminuye hasta los 125-500 Mbps.

La tecnología BLE consta de importantes beneficios y ventajas en comparación con otras tecnologías similares. Es una tecnología optimizada, cuyo consumo de energía es menor que en las demás tecnologías, entre otras cosas, gracias a que permite el envío de pequeñas cantidades de datos a bajas velocidades de transferencia. Además, no existen costos para acceder a documentación específica oficial, al contrario que en la mayoría de las tecnologías con la que compete en el mercado.

En cuanto a los módulos o *chips* se refiere, se observa que los que pertenecen a la tecnología BLE tienen un menor precio en el mercado.

Una de las ventajas más importantes del BLE es su existencia en la mayoría de los teléfonos inteligentes que hoy en día se encuentran en el mercado, característica que no albergan otros competidores como ZigBee, Z-Wave y Thread.

En cuanto a aspectos técnicos del BLE se refiere, el espectro de frecuencia que ocupa está entre 2,400 y 2,4835 GHz, con un alcance que puede superar los 100 metros y con una corriente pico de 15 mA de un conjunto de chips durante la transmisión de radio. A continuación, se puede observar una comparación entre el *bluetooth* clásico y el BLE (Tabla 5).

Specifications	Classic Bluetooth	Bluetooth Low Energy (V 4.2)	Bluetooth 5
Range	100 m	Greater than 100 m	Greater than 400 m
Data Rate	1-3 Mbps	1 Mbps	2 Mbps
Application Throughput	0.7 -2.1 Mbps	0.27 Mbps	—
Frequency	2.4 GHz	2.4 GHz	—
Security	56/128-bit	128-bit AES with Counter Mode CBC-MAC	—
Robustness	Adaptive fast frequency hopping, FEC, fast ASK	24-bit CRC, 32-bit Message Integrity Check	—
Latency	100 ms	6 ms	—
Time Lag	100 ms	3 ms	—
Voice Capable	Yes	No	—
Network Topology	Star	Star	—
Power Consumption	1 W	0.01 to 0.5 W	—
Peak Current Consumption	less than 30 mA	less than 15 mA	—

Comparison of Bluetooth standards

everythingRF

Tabla 5. Comparación de aspectos técnicos entre bluetooth clásico y BLE

El BLE puede ser utilizado para la configuración de un dispositivo antes de que este establezca la conexión inalámbrica principal, para esta actividad los dispositivos WIFI ya están incorporando la tecnología BLE para realizar esta configuración previa.

Un dispositivo móvil puede ser utilizado como interfaz para mostrar mayor cantidad de datos, que en otros dispositivos de baja potencia no pueden ser mostrados debido al pequeño tamaño de su pantalla.

Por último, BLE es utilizado para la transmisión de pequeñas cantidades de datos que provienen de un sensor.

CAPAS DE LA ARQUITECTURA BLE

Física

Incorpora todo el circuito de comunicaciones encargado de la modulación y demodulación de señales analógicas, para después realizar su transformación a símbolos de tipo digitales. Se pueden diferenciar dos tipos de canales, de datos y de publicidad (Tabla 6).

Frecuencia (MHz)	Número de canal	Tipo	Frecuencia (MHz)	Número de canal	Tipo
2402	37	Publicidad	2442	18	Datos
2404	0	Datos	2444	19	Datos
2406	1	Datos	2446	20	Datos
2408	2	Datos	2448	21	Datos
2410	3	Datos	2450	22	Datos
2412	4	Datos	2452	23	Datos
2414	5	Datos	2454	24	Datos
2416	6	Datos	2456	25	Datos
2418	7	Datos	2458	26	Datos
2420	8	Datos	2460	27	Datos
2422	9	Datos	2462	28	Datos
2424	10	Datos	2464	29	Datos
2426	38	Publicidad	2466	30	Datos
2428	11	Datos	2468	31	Datos
2430	12	Datos	2470	32	Datos
2432	13	Datos	2472	33	Datos
2434	14	Datos	2474	34	Datos
2436	15	Datos	2476	35	Datos
2438	16	Datos	2478	36	Datos
2440	17	Datos	2480	39	Publicidad

Tabla 6. Canales BLE

La colocación de los canales publicitarios, fue realizada de tal manera para que la superposición con los canales de la IEEE 802.11 EEUU fuese mínima (Ilustración 50).

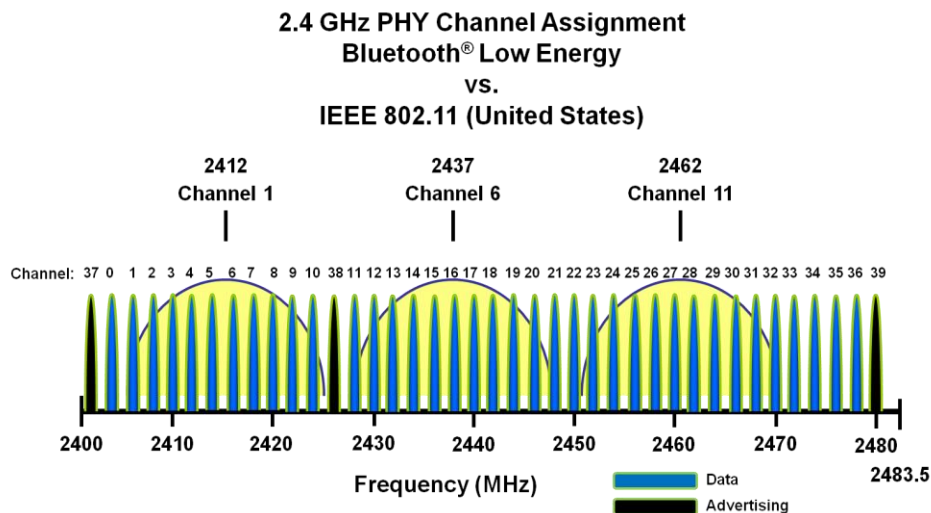


Ilustración 50. Canales IEEE 802.11 EEUU

Enlace

La capa de enlace interactúa de forma directa con la capa física y es la encargada de cumplir los requisitos temporales que se definan en las especificaciones.

Esta capa define la posibilidad de transmitir información entre dos dispositivos, pudiendo actuar éste como maestro, esclavo o ambas. Para realizar esta acción se definen otros procedimientos como son el rastreo de otros dispositivos, establecer y administrar la conexión con el otro dispositivo y una vez se establece dicha conexión el envío de datos a través de esta. Estos procedimientos son debilitados debido a las interferencias o ruidos que se producen en la banda de ISM para los sistemas de comunicación inalámbrica.

La máquina de estado de esta capa (Ilustración 51) explica cómo se realiza la conexión entre dos dispositivos.

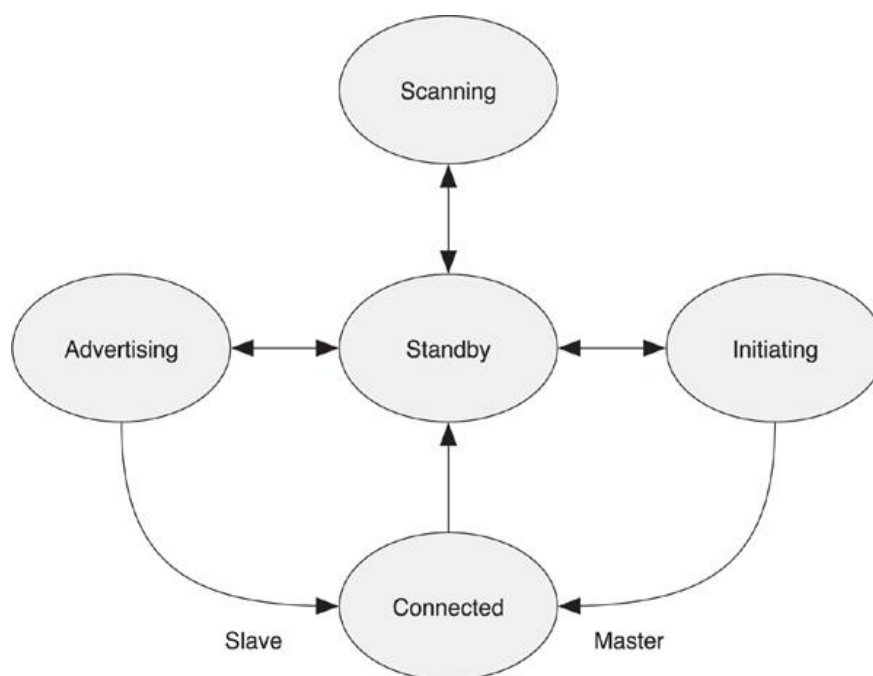


Ilustración 51. Estructura máquina de estado capa de enlace

- ❖ **Estado de espera (Standby):** Estado en el que permanecen las capas de enlace desde su puesta en funcionamiento hasta que las capas del controlador le envían otra orden.
- ❖ **Estado publicitario (Advertising):** Es el estado que permite que los paquetes publicitarios sean transmitidos por la capa de enlace, además de responder a peticiones de escaneo de otros dispositivos.
- ❖ **Estado de escaneo (Scanning):** Mediante este estado se reciben los paquetes provenientes de canales de publicidad, de esta forma se permite conocer qué otros dispositivos se encuentran enviando paquetes de publicidad dentro del área local. Se puede diferenciar dos tipos de escaneo, pasivo, que solamente permite el envío de paquetes de publicidad o activo que obtiene datos de respuestas de escaneo enviadas por otros dispositivos publicitarios.
- ❖ **Estado iniciador (Initiating):** Estado en el que debe estar situada la capa de enlace para poder establecer una conexión con otro dispositivo.
- ❖ **Estado de conexión (Connected):** En este estado se realizan envíos y recibos de paquetes, entre los dispositivos que haya conectados, a través de los canales

destinados a la transmisión de datos. Se encuentran dos subestados diferentes (Ilustración 52), maestro, que es el dispositivo que inicia la conexión, transmite paquetes a otros dispositivos y da permiso a los dispositivos que actúan como esclavos a transmitir paquetes, o esclavo, que previamente ha enviado un paquete de publicidad a otro dispositivo (maestro) para que este inicie la conexión.

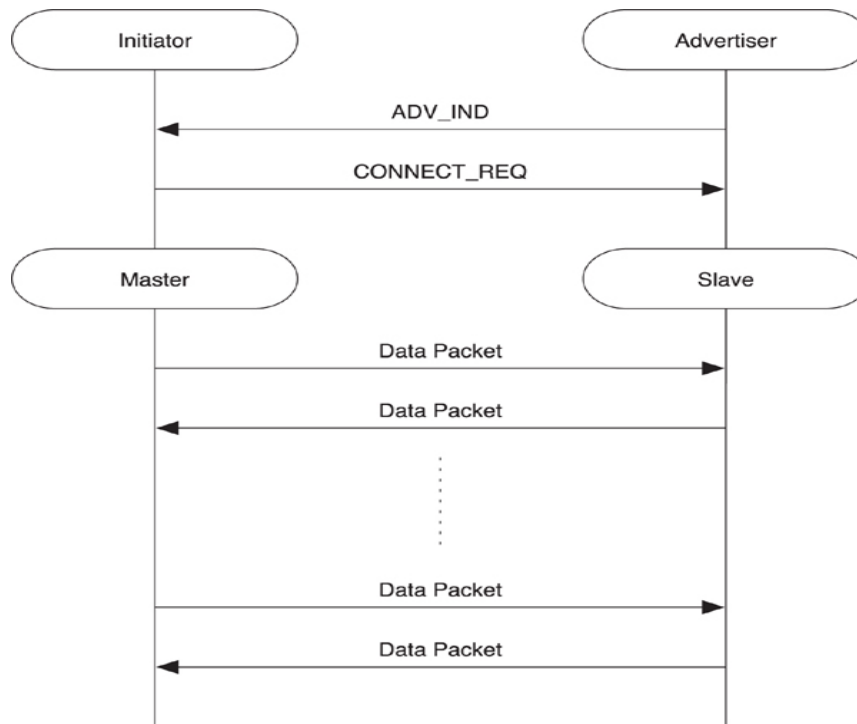


Ilustración 52. Comunicación esclavo-maestro

Se ha estado hablando de paquetes, pero no se habían definido todavía, un paquete es la pieza principal de la capa de enlace. Se define como un dato que incorpora una etiqueta, donde especifica su emisor y su o sus receptores, que será transmitido por un dispositivo para posteriormente ser recibido por uno o varios dispositivos. A continuación, se puede observar la estructura que tienen todos los paquetes utilizados en la capa de enlace (Ilustración 53).

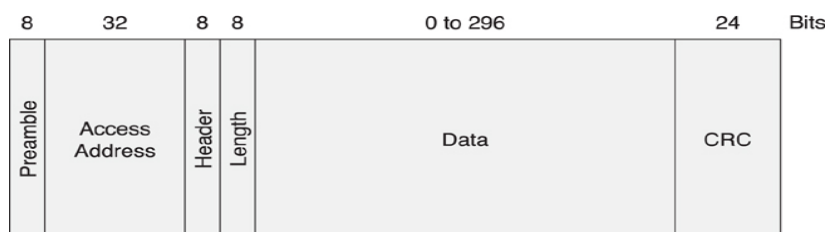


Ilustración 53. Estructura paquetes capa de enlace

Cabe destacar que de la parte de la dirección de acceso (*Access Address*) se puede extraer la información de cuál es el destinatario al que va dirigido y que la información principal se encuentra en la parte de datos (*Data*).

Modo de prueba directa

El modo de prueba directa da solución al problema de la calibración del sistema inalámbrico. Para la solución de este problema se definen procedimientos estándar y una interfaz de hardware que controle dicho protocolo. Son necesarios tres dispositivos para que actúe el modo de prueba directa (Ilustración 54):

- ❖ **Dispositivo bajo prueba (DUT)**: Representa al producto final que se prueba y debe incorporar una antena y un transmisor receptor asíncrono universal (UART) o una interfaz de *host*/controlador (HCI) para el probador superior.
- ❖ **Probador superior (UT)**: Guía al dispositivo que hay bajo prueba sobre la interfaz UART o HCI.
- ❖ **Probador inferior (LT)**: Dispositivo encargado de la transmisión y recepción de paquetes, estableciendo una conexión a través de una antena que incorpora y conectando con la antena que incorpora el dispositivo al que se está realizando la prueba.

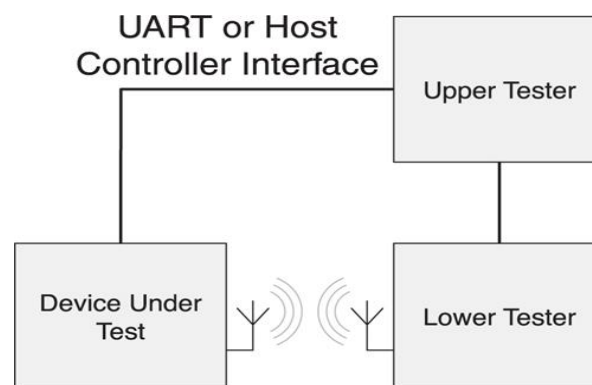


Ilustración 54. Estructura modo prueba directa

Interfaz host/controlador (HCI)

Sus funciones más características son el envío de comandos al controlador, el envío de eventos de copia de seguridad que circulan desde el controlador al *host* y el envío y recibimiento de datos de un dispositivo al que esté conectado. Se pueden diferenciar dos partes en la interfaz, una es la parte lógica que define los formatos de los paquetes y la otra es la física que indica como debe ser el transporte de los paquetes. Dentro de la física se pueden encontrar 3 tipos importantes, UART, USB y SDIO (Ilustración 55).

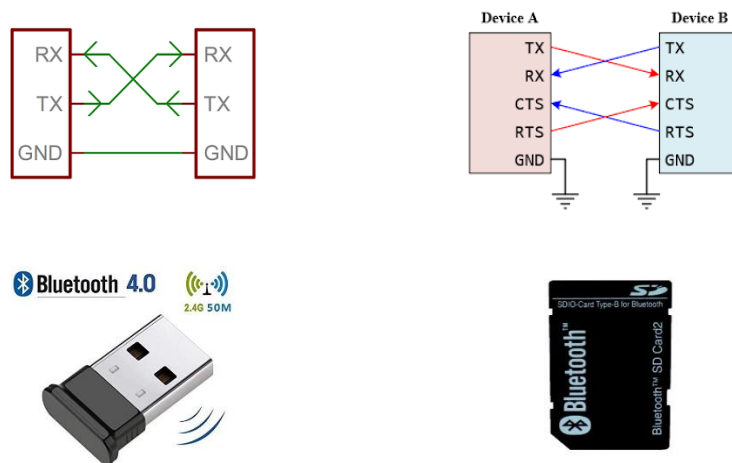


Ilustración 55. UART, USB y SDIO

Control de enlace lógico y protocolo de adaptación (L2CAP)

Esta es la capa de multiplexación para BLE. Realiza envíos de comandos a otras capas a través de canales bidireccionales (Ilustración 56). También se encarga de la fragmentación y recombinación de paquetes.

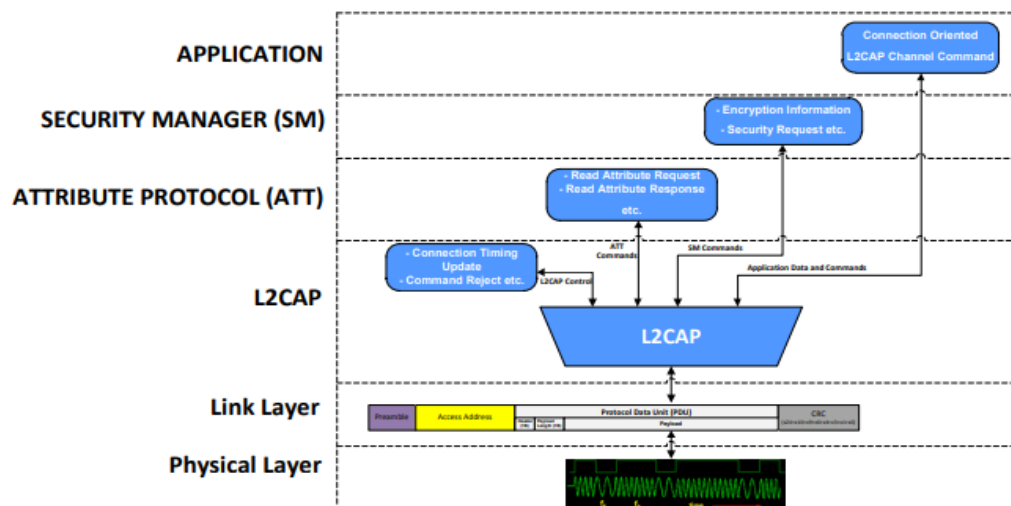


Ilustración 56. Envío de comandos del L2CAP

Protocolo de atributo (ATT)

Es un protocolo basado en cliente/servidor que define la estructura de los datos y la forma en que estos son presentados a un cliente. Dichos datos están estructurados como atributos. De forma ajena a que un dispositivo se comporte como maestro o esclavo, puede desempeñar la función de cliente, servidor o ambos. Cada dispositivo que conforme una conexión BLE actúa con una de estas funciones.

Cuando un cliente desea realizar la escritura o lectura de un valor de un atributo hacia un servidor, se transmite a éste una petición a través del manejador perteneciente a ese atributo. Hasta que una petición no obtenga respuesta, recibida o procesada, no se podrá enviar otra solicitud. En respuesta a esta petición se puede responder con un paquete que englobe el valor del atributo (Ilustración 57) o con una aceptación de la petición enviada previamente. A través de la escritura el servidor espera el aporte de datos desde el dispositivo cliente.

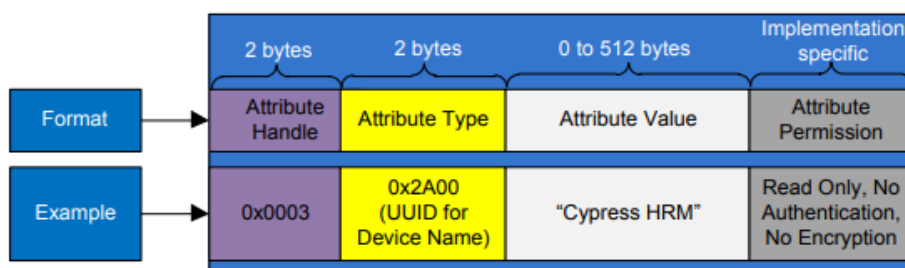


Ilustración 57. Estructura paquete de respuesta del ATT

Protocolo de gestión de seguridad (SMP)

SMP es la capa que se encarga de la seguridad de BLE. Esta capa permite el intercambio y la generación de claves de seguridad que posteriormente proporcionan seguridad ocultando la dirección pública en la comunicación mediante enlace cifrado. De esta forma evitará que dispositivos no deseados encuentren este dispositivo. En el SMP se pueden diferenciar dos roles diferentes:

- ❖ **Iniciador:** Perteneciente a la central GAP, y también al dispositivo que actúe como maestro en la capa de enlace.
- ❖ **Respondedor:** Perteneciente al periférico GAP, y también al dispositivo que actúe como esclavo en la capa de enlace.

Atributo Genérico (GATT)

El perfil de atributo genérico se encarga de procedimientos y formatos utilizados para la transferencia de datos, también de detallar cómo debe realizarse el intercambio de datos pertenecientes al perfil y al usuario mediante la conexión BLE.

Como es lógico GATT utiliza el ATT para el intercambio de datos cuando se establece una conexión de dispositivos. Además, estos son encapsulados en servicios que están formados por características que incluyen una serie de atributos, cada uno de ellos desempeñando una función diferente. Un perfil está formado por uno o varios servicios (Ilustración 58). Para un correcto funcionamiento de los perfiles de BLE, estos deben cumplir con el marco que GATT aporta para los perfiles basados en él. El formateo, empaquetado y envío de los elementos de los datos relevantes se deben de realizar con las reglas del GATT.

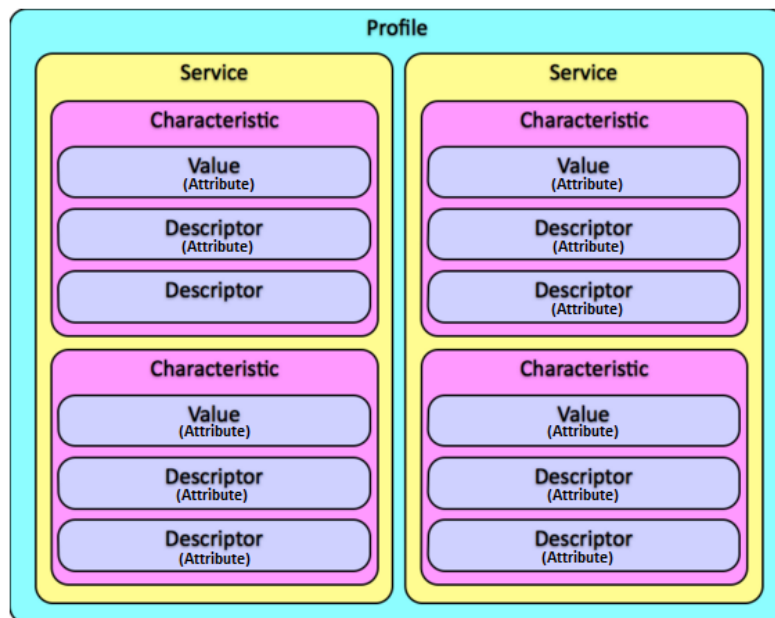


Ilustración 58. Estructura perfil del GATT

A continuación, repasamos diferentes aspectos que pueden ser encontrados dentro del GATT:

- ❖ **Roles**: En GATT existen dos roles diferentes que un dispositivo puede desempeñar, el primero es como cliente, que hasta que no conozca la naturaleza de los atributos de un servicio no podrá ni leer ni escribir y, por otro lado, tenemos el servidor, que es el encargado de que los datos estén disponibles para que el cliente disponga de estos. Un servidos GATT es imprescindible en un dispositivo BLE.
- ❖ **Identificador Universalmente Único (UUID)**: Es un número único que consta de 128 bits pero que se puede resumir en un número de 16 o 32 bits. Para volver a reconstruir los 128 bits se debe seguir la UUID base del *bluetooth* (Ilustración 59). Donde la UUID abreviada se incluye en “xxxxxxx” rellenando con ceros a la izquierda en el caso de que la UUID sea de 16 bits.

xxxxxxxx -icssoft1000-8000-00805F9B34FB

Ilustración 59. UUID base del bluetooth

- ❖ **Atributos:** La información se debe de organizar en atributos para que un cliente pueda interactuar con un servidor y así poder modificarlos. Cada atributo contiene tanto información de datos como información propia. El contenido real de un atributo va incluido en su valor, debido a la especificación, su longitud se limita a 512 bytes. Los permisos indican las operaciones ATT que pueden ser ejecutadas en cada atributo. Se pueden encontrar diferentes tipos de permisos:
 - **De acceso:** Determina si un cliente tiene la posibilidad de leer o escribir en ese atributo.
 - **Cifrado:** Nivel de cifrado requerido para poder acceder al atributo. Se puede acceder con una conexión sin cifrar o encriptada, y si la conexión es encriptada con o sin la necesidad de autenticación de sus claves.
 - **Autorización:** Un atributo puede elegir también la necesidad de un permiso de usuario o no para poder acceder a este.
- ❖ **Jerarquía de datos:** Cada conjunto de atributos es agrupado en servicios que pueden o no, tener características, que además pueden incluir descriptores dentro de ellas. Este tipo de jerarquías se utilizan normalmente para dispositivos BLE. No puede haber atributos aislados de esta jerarquía, ya que no podría haber intercambio de datos a través de BLE.

Dentro de las propiedades características se encuentran los procedimientos y operaciones que se pueden aplicar a una característica (Tabla 7). El cliente puede conocer las operaciones que se le permite realizar en la característica a través de una lectura de estas propiedades. Caben destacar el caso de dos operaciones como son la de notificar y la de indicar cuyo inicio proviene del servidor, pero con una activación previa del cliente mediante un descriptor característico.

Operaciones	Ubicación	Descripción
Transmitir	Propiedades	Permite la colocación en paquetes publicitarios del valor característico.
Leer	Propiedades	Permite la lectura al cliente.
Escribe sin respuesta	Propiedades	Permite la escritura al cliente.
Escribir	Propiedades	Permite la escritura/respuesta al cliente.
Notificar	Propiedades	Permite la notificación del valor de controlador al servidor.
Indicar	Propiedades	Permite la indicación/confirmación del valor de manejo.

Tabla 7. Operaciones sobre una característica

Perfil de Acceso Genérico (GAP)

Es un elemento principal para que se pueda establecer una conexión entre dos dispositivos a través de *bluetooth*. Hay un marco a seguir para toda implementación de BLE que permite la realización de las operaciones principales a través de un estándar mundialmente extendido. Al igual que en el GATT, se pueden diferenciar varios aspectos del GAP:

- ❖ **Roles**: Un conjunto de roles permiten la comunicación entre dos dispositivos. Se pueden diferenciar cuatro roles diferentes en el GAP a la hora de que un dispositivo se incorpore a una red BLE. Locutor, para aplicaciones que solo transmitan de forma regular paquetes publicitarios con datos. Observador, para aplicaciones que solo reciben datos de dispositivos de transmisión. Central, puede establecer múltiples conexiones siendo él mismo quien las inicie, tiene la función de maestro. Periférico, envía paquetes publicitarios a la central para que esta establezca una conexión con él, tiene la función de esclavo.
- ❖ **Modos y Procedimientos**: Un modo es referido a un estado que permite al dispositivo alcanzar un objetivo, de forma más explícita, permite la realización de un procedimiento particular al dispositivo con el que tenga una conexión establecida. Con el mismo fin de alcanzar un objetivo se define la palabra procedimiento, que es una secuencia de acciones que permite al dispositivo llegar al objetivo. Aquellos

modos donde no existe un procedimiento natural están marcados con N/A. A continuación, se puede observar algunos modos/procedimientos que realizan dos dispositivos, uno actuando como periférico y otro como central (Tabla 8).

MODO	ROL/ES APLICABLE/S	PROCEDIMIENTO/S DE PAR/ES APLICABLE/S (CENTRAL)
No reconocible	Periférico	N/A
Limitada reconocible	Periférico	Descubrimiento limitado
General reconocible	Periférico	Descubrimiento general
No conectable	Periférico	N/A
Cualquier conectable	Periférico	Cualquier establecimiento de conexión

Tabla 8. Modos y procedimientos del GAP

❖ **Seguridad**: Desde la aparición de BLE el objetivo principal que ha perseguido esta tecnología ha sido el poder transmitir de forma segura los datos de un usuario. Existen características destinadas a seguridad que no están relacionadas con los modos y procedimientos particulares, cuya implementación es libre de uso para realizar un aumento de la protección de los datos. La seguridad y privacidad de la tecnología inalámbrica BLE depende de dos partes:

- **Gerente de seguridad y su protocolo (SM Y SMP)**: SM realiza intercambios de protocolos que junto a algoritmos criptográficos ofrecen seguridad en el intercambio de datos entre dispositivos.
- **Seguridad de GAP**: GAP se basa en modos y procedimientos destinados a la seguridad que permite el intercambio de datos secretos a través de la conexión de un dispositivo similar. Los modos de seguridad de una conexión definen la seguridad actual que tiene esta y se puede basar en cifrado o en firmas de datos.

ANEXO 5. VÍDEOS DE FUNCIONAMIENTO. APLICACIÓN MÓVIL Y PANTALLA LCD

En los siguientes enlaces se puede observar el funcionamiento tanto de la aplicación móvil, como de la pantalla LCD. En ambos videos se muestran los datos medidos por los sensores y, por tanto, certifica que la conexión establecida mediante Bluetooth Low Energy funciona correctamente.

Pantalla LCD

<https://youtu.be/4LQAoJu66sw>

Aplicación móvil

<https://youtu.be/KXFJndvPNfw>

ANEXO 6. CÓDIGO DE PSOC CREATOR Y ANDROID STUDIO

Todo el código que se ha implementado para el desarrollo de este proyecto se puede encontrar en el siguiente enlace de GitHub.

https://github.com/afmgepsl2019/TFG_BluetoothLowEnergy

6. APÉNDICE

Comentario sobre los archivos de los que consta dicho proyecto. Como se menciona en apartados anteriores en este proyecto se diferencian archivos de dos programas diferentes, PSoC Creator y Android Studio.

❖ PSoC Creator

- ***BLE_APP.c*** → Código donde se describen los diferentes eventos del BLE.
- ***BLE_APP.h*** → Declaración de las variables y de la función *BLECallBack* utilizada para los eventos del BLE en el archivo *BLE_APP.c*.
- ***LiquidCrystal_I2C.c*** → Código desarrollado de las funciones de la pantalla de cristal líquido.
- ***LiquidCrystal_I2C.h*** → Declaración de variables y funciones utilizadas en *LiquidCrystal_I2C.c*.
- ***main.c*** → Archivo principal que contiene todo el código de desarrollo del proyecto.
- ***main.h*** → Declaración de variables *TRUE* y *FALSE* utilizadas en el *main.c*.

❖ Android Studio

- ***actionbar_indeterminate_progress.xml*** → Se define la barra de acciones que va situada en lo alto de la pantalla, donde se muestran las diferentes acciones de escaneo o conexión.
- ***listitem_device.xml*** → Correspondiente a la pantalla principal de la aplicación. En ella solamente aparece un listado de los dispositivos BLE encontrados y el símbolo de la Escuela Politécnica Superior de Linares.
- ***gatt_service_characteristics.xml*** → Correspondiente a la segunda pantalla de la aplicación. En este archivo se desarrolla la estética de dicha pantalla, donde hay una lista con los servicios y características de los que dispone la conexión BLE, y dos cuadros de texto, uno para la hora y fecha actual y otro para mostrar los datos.
- ***main.xml*** → Se definen los botones de escanear y parar el escaneo, utilizados para el escaneo de dispositivos BLE.
- ***gatt_service.xml*** → Se definen los botones de conectar y desconectar, utilizados para habilitar o deshabilitar la conexión de dispositivos BLE.

- ***string.xml*** → Se define el diferente texto que se mostrará en la aplicación.
- ***AndroidManifest.xml*** → En este archivo se definen los permisos necesarios, el nombre y el icono de la aplicación y los archivos que actúan como actividades y servicios.
- ***BluetoothLeService.java*** → Define todos los procesos relacionados con el BLE como, por ejemplo, habilitar las notificaciones o leer los datos de las características, entre otros.
- ***DeviceScanActivity.java*** → Define la parte del escaneo de dispositivos BLE.
- ***DeviceControlActivity.java*** → Define la parte de manejo de todos los datos mostrados en la aplicación.
- ***SampleGattAttributes.java*** → Define las UUID de los servicios y características definidas.

7. ACRÓNIMOS

BLE	<i>Bluetooth Low Energy</i>
WiFi	<i>Wireless Fidelity</i>
PSoC	<i>Programmable System on Chips</i>
XML	<i>eXtensible Markup Language</i>
ISM	<i>Medical, Scientific and Industrial</i>
DUT	<i>Device Under Test</i>
UT	<i>Upper Tester</i>
LT	<i>Lower Tester</i>
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
HCI	<i>Host-Controller Interface</i>
USB	<i>Universal Serial Bus</i>
SDIO	<i>Secure Digital Input-Output</i>
ATT	<i>ATtribute protocol</i>
SMP	<i>Security Management Protocol</i>
GATT	<i>Generic ATtribute</i>
GAP	<i>Generic Access Profile</i>
AD	<i>Advertising Data</i>
UUID	<i>Universally Unique IDentifier</i>
SRAM	<i>Static Random Access Memory</i>
PGA	<i>Pin Grid Array</i>
PWM	<i>Pulse-Width Modulation</i>
TCPWM	<i>Timer-Counter PWM</i>
GPIO	<i>General Purpose Input/Output</i>
LCD	<i>Liquid Crystal Display</i>
I2C	<i>Inter-Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
GND	<i>Ground</i>
VCC	<i>DC Voltage</i>
SDA	<i>System Data</i>
SCL	<i>System Clock</i>

8. BIBLIOGRAFÍA

- [1] Kevin Townsend, Carles Cufi, Akiba, Robert Davidson, “Getting started with Bluetooth Low Energy”. O'Reilly Media, Inc. Mayo 2014.

- [2] Madhur Bhargava, “IoT Projects with Bluetooth Low Energy”. O'Reilly Media, Inc. Agosto 2017.

- [3] Muhammad Usama bin Aftab, “Building Bluetooth Low Energy Systems”. O'Reilly Media, Inc. Abril 2017.

- [4] Reza Mahmoudi and Krzysztof Iniewski, “Low Power Emerging Wireless Technologies”. O'Reilly Media, Inc. Julio 2017.

- [5] Robin Heydon, “Bluetooth Low Energy: The Developer’s Handbook”. O'Reilly Media, Inc. Octubre 2012.

- [6] Jeremy Blum, “Exploring Arduino, 2nd Edition”. O'Reilly Media, Inc. Noviembre 2019.

- [7] Ted Hagos, “Android Studio IDE Quick Reference: A Pocket Guide to Android Studio Development”. O'Reilly Media, Inc. Julio 2019.

- [8] Hernández, R. [PSoC Projects]. Introducción al mundo de PSoC – una opción al microcontrolador tradicional. [10 enero 2020]. Disponible en:
<https://www.youtube.com/watch?v=Jrm6ZdAwpn8>

- [9] Bermúdez, B. A. [PSoC Projects]. Unboxing: CY8CKIT – 942 BLE PSoC 4 Pioneer Kit BLE. [18 enero 2020]. Disponible en:
https://www.youtube.com/watch?v=BsY46q_CPjE

- [10] Afaneh, M. 22 febrero 2017. 5 Essential Tools for Every Bluetooth Low Energy Developer. Bluetooth. [25 enero 2020]. Disponible en:

<https://www.bluetooth.com/blog/five-essential-tools-for-every-bluetooth-low-energy-developer/>

[11] Katsandres, J. 15 febrero 2017. Bluetooth Low Energy – It Starts with Advertising. [25 enero 2020]. Disponible en:
<https://www.bluetooth.com/blog/bluetooth-low-energy-it-starts-with-advertising/>

[12] Android Studio. [17 noviembre 2019]. Disponible en:
<https://developer.android.com/studio/>

[13] Cypress. PSoC 4: 4200 BLE Family Datasheet. Programmable System-on-chip (PSoC). [20 enero 2020]. Disponible en:
<https://www.cypress.com/file/416486/download>

[14] Cypress. 32-bit Arm Cortex-M0 PSoC 4200 Programmable Digital MCU's. [20 enero 2020]. Disponible en: <https://www.cypress.com/products/32-bit-arm-cortex-m0-psoc-4200-programmable-digital-blocks>

[15] Cypress. Internet of Things (IoT). [21 enero 2020]. Disponible en:
<https://www.cypress.com/solutions/internet-things-iot>

[16] Cypress. BLE & Bluetooth. [22 enero 2020]. Disponible en:
<https://www.cypress.com/products/ble-bluetooth>

[17] Cypress. MCU & PSoC. [5 diciembre 2019]. Disponible en:
<https://community.cypress.com/community/MCU>

[18] Editorial Team – everything's RF. 23 febrero 2017. ¿Cuál es la diferencia entre Bluetooth 5, Bluetooth 4.2 y Bluetooth 2.0? [13 enero 2020]. Disponible en:
<https://www.everythingrf.com/community/what-is-the-difference-between-bluetooth-5-0-bluetooth-low-energy-bluetooth-v4-2-and-classic-bluetooth>

[19] Developers. Descripción general del Bluetooth de bajo consumo. [8 enero 2020]. Disponible en: <https://developer.android.com/guide/topics/connectivity/bluetooth-ble?hl=es-419#java>

[20] Smartlighting. 2 mayo 2019. Bluetooth Low Energy – Introducción a la tecnología. [25 enero 2020]. Disponible en: <https://smart-lighting.es/bluetooth-low-energy-introduccion-la-tecnologia/>

[21] Microchip Developer Help. Actualizado 16 enero 2020. Introducción a Bluetooth Low Energy. [20 enero 2020]. Disponible en: <https://microchipdeveloper.com/wireless:ble-introduction#top-of-page>

[22] Bensky, A. 2019. Redes inalámbricas de área personal. [26 enero 2020]. Disponible en: <https://www.sciencedirect.com/topics/computer-science/bluetooth-low-energy>

[23] Cypress. Como crear aplicaciones de Android para interactuar con Cypress BLE. [20 noviembre 2019]. Disponible en: <https://www.cypress.com/training/psoc-video-tutorial-series-how-create-android-apps-interact-cypress-ble>

[24] Llamas, L. 29 marzo 2016. Medir temperatura y humedad con Arduino y sensor DHT11-DHT22. [20 noviembre 2019]. Disponible en: <https://www.luisllamas.es/arduino-dht11-dht22/>

[25] Cypress. *Creating a BLE Custom Profile*. [21 noviembre 2019]. Disponible en: <https://www.cypress.com/documentation/application-notes/an91162-creating-ble-custom-profile>

[26] Departamento de Informática, Escuela Técnica Superior de Ingeniería, Universidad de Valencia Grupo de Investigación de Ingeniería Agromótica y del Mar (UPCT). 19 febrero 2018. Sensores de bajo coste aplicados al control de los cultivos. [7 mayo 2020]. Disponible en: <https://www.interempresas.net/Horticola/Articulos/208300-Sensores-de-bajo-coste-aplicados-al-control-de-los-cultivos.html>

[27] Infineon. Cypress is now Infineon. [18 mayo 2020]. Disponible en:
<https://www.infineon.com/cms/en/about-infineon/company/cypress-acquisition/>

[28] MySignals. MySignals cambia el futuro de aplicaciones médicas y de eSalud. [26 mayo 2020]. Disponible en: <http://www.my-signals.com/>

[29] Libelium Comunicaciones Distribuidas S.L. Mayo 2019. MySignals SW: eHealth and medical IoT Development Platform Technical Guide [26 mayo 2020]. Disponible en:
http://www.libelium.com/downloads/documentation/mysignals_technical_guide.pdf

[30] The IoT Marketplace. [25 mayo 2020]. Disponible en: <https://www.the-iot-marketplace.com/solutions>

[31] Libelium. Modelos. [25 mayo 2020]. Disponible en:
<http://www.libelium.com/products/plug-sense/models/#smart-agriculture-xtreme>

[32] Libelium Comunicaciones Distribuidas S.L. Abril 2019. Smart Agriculture Xtreme 1.0: Technical Guide [25 mayo 2020]. Disponible en:
http://www.libelium.com/downloads/documentation/smart_agriculture-xtreme_technical_guide.pdf