**Estado del Arte**

## Introducción

En este capítulo se va a realizar una revisión de los diferentes kits de desarrollo existentes en el mercado, los cuales están formados por el hardware necesario para realizar el diseño de un *Gadget* para un *Smartphone/Tablet*. Además, también será objeto de este capítulo la descripción de los módulos software de dichos kits.

Otros elementos tratados en el capítulo son los sistemas operativos (S.O) utilizados en los *Smartphones/Tablets*, además de los diferentes medios disponibles para la comunicación entre los kits de desarrollo y los dispositivos electrónicos anteriormente mencionados.

A continuación se hará una breve descripción de *Smartphone*, *Tablet* y *Gadget*.

**Sistemas Operativos para Smartphones**

**iOS**

iOS [8] (anteriormente denominado iPhone OS) es un sistema operativo móvil de Apple. Fue originalmente desarrollado para el *Smartphone* iPhone, siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV.

La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gestos multitáctiles (*multitouch*). Los elementos de control son deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz.

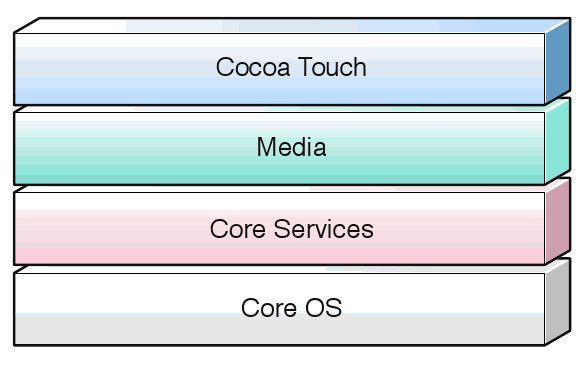


Ilustración 2‑18: Cuatro capas de iOS.

iOS se deriva de Mac OS X, que a su vez está basado en Darwin BSD (plataforma de código abierto), que es un sistema operativo tipo Unix. iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de “Servicios Personales”, la capa de “Medios” y la capa de “*Cocoa Tocuh*”. La versión actual del sistema operativo (iOS 6.0) ocupa más o menos 770 MB, variando por modelo.

**Android**

Android es un sistema operativo móvil basado en Linux, que junto con aplicaciones *middleware* (*software* que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones) está enfocado para ser utilizado en dispositivos móviles como *Smartphones*, *Tablets*, Google TV y otros dispositivos.

Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Pero en realidad es el principal producto de la *Open Handset Alliance*, un conglomerado de fabricantes y desarrolladores de *hardware*, *software* y operadores de servicios.

El anuncio del sistema Android se realizó el 5 de Noviembre de 2007 junto con la creación de la *Open Handset Alliance*. Google libero la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto.

El sistema operativo se halla en una zona de memoria de sólo lectura por dos motivos: para evitar que el usuario lo dañe sin querer y para que se sea fiel a las pequeñas modificaciones y aplicaciones integradas que los fabricantes suelen incluir en sus modelos.

Al ser un sistema operativo de código fuente abierto, Android permite toda clase de modificaciones. Además de las ROM oficiales, es muy habitual encontrar ROM hechas por grupos de voluntarios que toman el código base y le añaden o quitan características o interfaces de usuario. Incluso uno mismo puede crear la suya.

Para actualizar un móvil Android se tienen varias opciones, siempre dependiendo de la operadora y sobre todo del fabricante de nuestro dispositivo. Algunos permiten actualizar por medio de la conexión USB entre el móvil y el PC, y otros directamente en el dispositivo descargando un archivo a la microSD y encendiendo el móvil. Sin embargo, lo más normal es que se actualice un Android por OTA (*Over The Air*) o inalámbricamente, sin tener que conectar el móvil por cable ni para descargar un archivo ni para actualizar mediante un programa.



Ilustración 2‑16: Arquitectura de Android.

La arquitectura de Android está distribuida en diferentes capas, las cuales se describen a continuación:

*Applications* (Aplicaciones): Las aplicaciones bases incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

*Application Framework* (Marco de trabajo de aplicaciones): Los desarrolladores tienen acceso completo a los mismos APIs del *framework* usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del *framework*). Este mecanismo permite que los componentes sean reemplazados por el usuario.

*Libraries* (Bibliotecas): Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del *framework* de aplicaciones de Android. Las bibliotecas escritas en lenguaje C/C++ incluyen un administrador de pantalla táctil (*surface manager*), un *framework Open Core* (para el aprovechamiento de las capacidades multimedia), una base de datos relacional *SQLite*, una API gráfica *OpenGL ES 2.0 3D*, un motor de renderizado *WebKit*, un motor gráfico SGL, el protocolo de comunicación segura SSL y una biblioteca estándar de C, llamada “*Bionic*” y desarrollada por Google específicamente para Android a partir de bibliotecas estándar “*libc*” de BSD.

*Android Runtime* (Funcionalidad en tiempo de ejecución): Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual *Dalvik*. *Dalvik* ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. *Dalvik* ejecuta archivos en el formato *Dalvik Executable* (.dex), el cual está optimizado para memoria mínima.

*Linux Kernel* (Núcleo Linux): Android dispone de un núcleo basado en *Linux* para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores, y también actúa como una capa de abstracción entre el *hardware* y el resto de la pila de *software*.

Los desarrolladores informaron de la dificultad de mantener aplicaciones para versiones diferentes de Android, debido a problemas de compatibilidad entre la versión 1.5 y 1.6, especialmente, por diferencias de resolución entre los distintos teléfonos Android.

Estos problemas se hicieron patentes durante el concurso ADC2. Posteriormente, el rápido aumento de modelos de teléfonos basados en Android con diferentes capacidades de *hardware* complicaba el desarrollo de aplicaciones para todos los modelos de teléfonos Android.

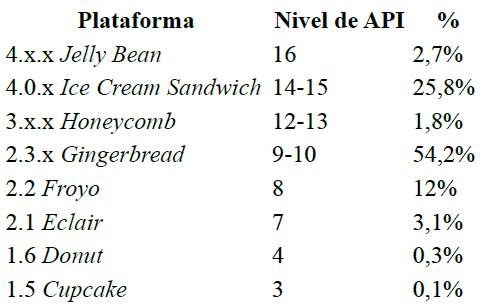


Ilustración 2‑17: Versiones Android.

Sin embargo, la situación parece haber mejorado “en parte”, porque aún más del 54 % de teléfonos con Android usan versiones 2.3.X (salida en 2010). El principal problema de esto es a la hora de desarrollar aplicaciones nuevas con las APIs más recientes del SDK de Android, se tiene que usar una versión de API compatible con 2.3, así las nuevas funcionalidades de APIs superiores, como NFC a partir de la versión Android ICS 4.0, no se pueden usar si no se implementan a mano por no poder usar la API que hace uso de su funcionalidad.

**Windows Phone**

**Symbian Os ¿?**

Symbian es un sistema operativo que fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encuentran: Nokia, Sony Ericsson, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. Sus orígenes provienen de su antepasado EPOC32, utilizado en PDA´s.

El objetivo de Symbian fue crear un sistema operativo para terminales móviles que pudiera competir con el de Palm o el Windows Mobile 6.X de Microsoft y ahora con el SO Android de Google Inc, iOS de Apple Inc y BlackBerry OS de RIM.

En 2003 Motorola vendió el 13 % de su participación a Nokia, lo cual hizo que se quedara con el 32,3 % de la compañía. Más tarde, sin embargo, después de no tener el éxito esperado con sus terminales “*Linux-Like*”, volvió al mundo de Symbian comprándole el 50 % de las participaciones a Sony Ericsson. El 24 de Junio de 2008, Nokia decidió comprar Symbian, adquiriendo el 52 % restante de las acciones de la compañía. El objetivo era establecer la Fundación Symbian y convertir este sistema operativo en una plataforma abierta. Entre 2009 y 2010 Nokia decide transferir el soporte y desarrollo del sistema operativo Symbian a la consultora *Accenture*, terminando la operación a finales de Septiembre de 2011, cuando se terminó el desarrollo de la nueva versión Symbian Belle, convirtiéndose en la última versión de Symbian en la que Nokia participó de forma exclusiva. En Octubre de 2011 se confirma de forma oficial que Symbian tendrá soporte, sólo, hasta el año 2016, por no poder seguir siendo un competidor para la nueva versión de *Smartphones* con sistemas operativos de última generación como Androis, iOS o Windows Phone.

La interfaz gráfica por defecto es S60: La plataforma S60 es una plataforma para terminales móviles que utilicen el sistema operativo Symbian OS. Está desarrollada principalmente por Nokia y licenciada por ellos a otros fabricantes.

Symbian^3 se considera una nueva generación del sistema operativo Symbian y se usa en los *Smarpthones* de nueva generación de Nokia. Ésta tiene una gran compatibilidad de *hardware* y soporte para gráficos acelerados con la aceleración de *hardware* en 2D y 3D, soporte para HDMI, hasta 3 pantallas de inicio personalizables con *widgets*, mejoras estéticas notables gracias a la aceleración de gráficos y muchas mejoras generales en estabilidad, entre ellas la consistencia. Ésta es la primera versión de código abierto de Symbian, la cual se presentó un mes después de haber liberado el código fuente de todo el sistema.

**Blacberry ¿?**

BlackBerry Os es un sistema operativo móvil, de código cerrado, desarrollado por Research In Motion (RIM) para sus propios dispositivos BlackBerry.

El sistema permite multitarea y tiene soporte para diferentes métodos de entrada adoptados por RIM para su uso en computadoras de mano, particularmente la trackwhell, trackball, touchpad y pantallas táctiles.

El SO BlackBerry está claramente orientado a su uso profesional como gestor de correo electrónico y agenda. Desde la cuarta versión se puede sincronizar el dispositivo con el correo electrónico, el calendario, tareas, notas y contactos de Microsoft Exchange Server. Al igual que en el SO Symbian, desarrolladores independientes también pueden crear programas para BlackBerry pero en el caso de querer tener acceso a ciertas funcionalidades restringidas necesitan ser firmados digitalmente para poder ser asociados a una cuenta de desarrolladores de RIM.

## Protocolos inalámbricos

### Introudccion a las Redes WPAN

Las redes inalámbricas de área personal WPAN por sus siglas en inglés Wireless Personal Area Network son redes que comúnmente cubren distancias del orden de los 10 metros como máximo, normalmente utilizadas para conectar varios dispositivos portátiles personales sin la necesidad de utilizar cables. Esta comunicación de dispositivos peer-to-peer normalmente no requiere de altos índices de transmisión de datos. La tecnología inalámbrica Bluetooth, por ejemplo, tiene un índice nominal de 10 metros con índices de datos de hasta 1Mbps. El tipo de ámbito y los relativos bajos índices de datos tienen como resultado un bajo consumo de energía haciendo a la tecnología WPAN adecuada para el uso con dispositivos móviles pequeños, que funcionan con baterías, tales como teléfonos celulares, asistentes personales PDAs o cámaras digitales. [1]

**Bluetooth**

**Bluetooth Low Energy**

Bluetooth Low Energy (BLE, también comercializado como Bluetooth Smart) comenzó como parte de la especificación Bluetooth 4.0.

Originalmente diseñado por Nokia como *Wibree* antes de ser adoptado por el *Bluetooth Special Interest Group* (GSIP, en español Grupo de Especial Interés en Bluetooth), los desarrolladores no estaban intentando sugerir otra solución inalámbrica capaz de resolver todos los posibles problemas de las conexiones inalámbricas. Desde el principio el foco se centró en diseñar un estándar de radio con el menor consumo de energía posible, específicamente optimizado para el bajo coste, bajo ancho de banda, baja potencia, y baja complejidad.

Estas metas de diseño son evidentes en las especificaciones fundamentales*,* las cuales contemplan hacer del BLE un estándar de baja potencia, diseñado para que se pueda implementar actualmente por los fabricantes de componentes y que se integre fácilmente en las aplicaciones actuales. Posiblemente sea el primer estándar ampliamente adoptado que realmente proclama que puede ser usado durante un largo periodo de tiempo desde una batería tipo *botón.*

Para un estándar relativamente joven (fue introducido en 2010), BLE ha tenido un tasa de adopción muy rápida, y el número de productos diseñados que ya incluyen BLE lo sitúa en la cabeza junto a otras tecnologías inalámbricas con mayor tiempo de lanzamiento. Compara do con otras tecnologías inalámbricas el crecimiento del BLE es relativamente sencillo de explicar: BLE está ligado a la computación móvil, *tablets* y *smartphones*, y por tanto a su alto crecimiento. Una temprana y activa adopción por parte de algunas compañías importantes dentro de la industria móvil como *Apple* o *Samsung* abrió las puertas para una amplia implementación de BLE. *Apple*, en particular, ha hecho un esfuerzo significativo produciendo una pila BLE fiable y publicando líneas generales de diseño en torno al BLE. Esto empujó a los fabricantes de componentes a asignar sus limitados recursos a la tecnología que mayor tasa de éxito y rentabilidad supusieran a la larga, y el sello de aprobación de *Apple* a esta tecnología fue un gran argumento a su favor.

Mientras los mercados de *tablets* y *smartphones* van incrementando su madurez y los costes y los márgenes van decreciendo, la necesidad de conectividad con el mundo exterior en estos dispositivos ha crecido exponencialmente, y esto ofrece a los vendedores de periféricos una oportunidad única de proveer innovadoras soluciones a problemas de los que las personas no se habían dado cuenta hasta ahora.

Todos esos beneficios han convergido en torno al BLE, y se ha abierto una gran oportunidad para los pequeños desarrolladores de productos al ganar acceso un potencial mercado masivo con productos creativos, innovadores, destinados a tareas específicas y a un modesto presupuesto de diseño. Actualmente se puede compra sistemas SoC (System On a Chip, un microcontrolador con todo el sistema incorportado) a precios y volúmenes por debajo de los que permiten otras tecnologías como WiFi, GSM, ZigBee… y BLE permite diseñar productos viables que a día de hoy pueden comunicarse con cualquier plataforma móvil moderna usando herramientas y estándares fácilmente accesibles.

Una diferencia fundamental con el *Bluetooth* clásico, el cual está enfocado en un conjunto de situaciones, es que BLE está concebido para permitir que cualquiera pueda hacer un intercambio de datos desde un accesorio, sin tener gran conocimiento acerca de las capas inferiores de la tecnología. Los fabricantes de *smartphones* también proveen una flexible y relativamente de bajo nivel APIs para dar a los desarrolladores de aplicaciones móviles la libertad de usar el *framework* BLE.

Con un modelo de datos relativamente sencillo de entender, costes de licencia no intrusivos, sin tasas para el acceso a las *core specs*, y un una pila de protocolo simple en general, queda expuesto porqué BLE es uno de los estándares más elegido en la actualidad.

La especificación

En Junio de 2010, *Bluetooth SIG* introdujo el protocolo BLE con la versión 4.0 de las *Bluetooth Core Specification* (Especificaciones fundamentales Bluetooth).

La primera y mayor actualización, Bluetooth 4.1, fue lanzada en Diciembre de 2013 y es la actual referencia para el desarrollo de productos BLE. Aunque los bloques básicos de construcción, procedimientos, y conceptos permanecen intactos, esta actualización también introdujo múltiples cambios y mejoras para suavizar la experiencia del usuario.

Como en todas las especificaciones Bluetooth, la 4.1 es retrocompatible con la 4.0, asegurando la correcta interoperabilidad entre los dispositivos que implementan las diferentes versiones de la especificación. También permite a los desarrolladores actualizar y clasificar productos en cualquiera de las versiones, aunque la adopción rápida de las nuevas actualizaciones es recomendable.

The specifications allow developers to release and qualify products against either of the

versions (until deprecated), although the rapid adoption of new specification releases

and the fact that the 4.1 version standardizes several common practices among devices

makes it recommendable to target the latest available one.

Unless otherwise noted, this book uses the Bluetooth 4.1 specification as reference.

Wherever necessary, and especially when mentioning a noteworthy change or addition,

we will clarify when the previous 4.0 specification does not cover a particular area.

To obtain the latest adopted version of the Bluetooth specification, see the Bluetooth

SIG’s Specification Adopted Documents page.

**ZigBee**

**Conclusión**

¿Por qué BLE?

Permite crear Profiles propios.

**Módulos**

En la mayoría de los casos que implican un uso del protocolo BLE, el dispositivo con el que interactuar actúa como *periférico* en lugar de como *central* (este rol lo asumen los *smartphones, tablets* u ordenadores portátiles). En la actualidad existen diversos módulos *SoC* (*System-On –Chip*) especialmente diseñados para aplicaciones basadas en BLE (barato y bajo consumo de energía). Algunos de ellos son:

Texas Instruments CC2540 y CC2541



El CC2540 y su versión mejorada, el CC2541, han sido muy populares durante varios años. Ambos dispositivos incluyen un procesador 8051 que permite ejecutar aplicaciones sin un microcontrolador externo, con lo cual el coste del sistema disminuye. Aunque esto también tiene un inconveniente común en muchas soluciones integradas y es que la aplicación está atada a este procesador, y en algunos casos los productos requieren más desempeño.

Algunas de las especificaciones del componente son:

Sistema completo que integra BLE, microcontrolador y gestión de periféricos.

Microcontrolador 8051 con 256kB o 128kB de memoria flash y 8kB de RAM.

Empaquetamiento 6x6mm QFN.

Máxima potencia de salida de 0dBm.

Consumo de corriente RX de 17,9mA a 14,7mA; y TX 18,2mA a 14,3mA.

Sensibilidad del receptor -94dBm a 1Mbps.

Además el dispositivo se completa con diversos periféricos para dar cobertura a todo tipo de productos.

Un aspecto importante de los dispositivos CC2540 y 2541 es el kit de herramientas de desarrollo. La mayoría de los chips actuales de BLE son pequeños dispositivos de radio con software especializado, por lo tanto es este firmware el que gestiona algunas de las funcionalidades de bajo nivel del BLE. Texas Instruments provee con frecuencia nuevas y mejoradas versiones de firmware. El desarrollo las aplicaciones está ligado al entorno de desarrollo IAR Workbench 8051.

El dispositivo CC2541 además de radio BLE puede ejecutar otros protocolos propietarios que el CC2540 no es capaz de ejecutar. Otras diferencias entre ambos dispositivos son:

La corriente Rx en el modo de menor consumo se reduce de 19,6mA a 17,9mA.

CC2541 no tiene interfaz USB.

CC2541 tiene interfaz I2C.

CC2541 puede transmitir a una potencia de 0dBm, frente a los 4dBm del CC2540.

[More Information on the CC2541](http://www.ti.com/product/CC2541)

Nordic Semiconductor nRF8001



El nRF8001 es un dispositivo independiente que requiere el uso de un microcontrolador externo, el chip se encarga de las tareas de protocolo de bajo nivel. Se controla a través del envío de comandos, datos y de la recepción de eventos por parte del dispositivo. Se conecta al microcontrolador generalmente por SPI.

Este dispositivo aporta el beneficio de adaptar el microcontrolador, por el que se puede elegir este según la experiencia del desarrollador, o según los requerimientos de la aplicación. Sin embargo usar un microcontrolador externo viene ligado a un mayor coste, además existe una mayor carga de trabajo y responsabilidad en el desarrollador debido a la necesidad de tratar con drivers de bajo nivel para la comunicación entre chip y microcontrolador.

Algunas de las especificaciones del chip son:

Versión Bluetooth soportada BLE v4.0 Single Mode.

Integra el protocolo BLE completo.

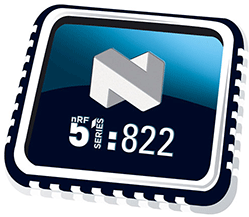
Requiere un microcontrolador externo.

Empaquetamiento 6x6mm QFN.

Consumo de corriente RX de 8mA y TX 9,7mA.

Otra característica a tener en cuenta es que no es posible enviar paquetes iBeacon debido a limitaciones del fabricante.

Nordic Semiconductor nRF51822



Nordic Semiconductor lanzó el nRF51822 después del nRF80001 como versión SoC para comunicaciones BLE. De muy pequeño tamaño, incorpora un microprocesador Cortex-M0. El uso de este procesador implica que hay diversas opciones de compiladores para el desarrollo de aplicaciones. Algunos de los valores que distinguen al nRF51822 respecto al nRF80001 son:

Versión de Bluetooth soportada BLE v4.1 single Mode.

ARM® Cortex™-M0 integrado en el chip, con 256kB o 128kB de memoria flash y 16kB o 32kB de memoria RAM.

Disponible en empaquetamientos QFN 6x6mm y WLCSP 3.5x3.8mm.

Voltaje de alimentación de 1.8v-3.6V.

Consumo de corriente RX de 12,7mA y TX 14,6mA.

Además del protocolo, los chips de Nordic Semiconductor pueden soportas más protocolos. Soportan protocolos propietarios en la banda de 2.4GHz, con tres diferentes tasas de datos por encima de los 2Mbps. También son compatibles con la gama de dispositivos nRF24L de Nordic que son ampliamente usados en muchos productos.

Más información sobre el nRF51822

[Dialog Semiconductor DA14580](http://www.dialog-semiconductor.com/products/bluetooth-smart/smartbond-da14580)

El chip DA14580 no tiene memoria flash integrada, lo que disminuye el consumo de energía y abarata su coste ya que la memoria flash requiere circuitos especializados y voltajes más altos. Algunas de sus características son:

Versión de Bluetooth soportado BLE v4.1 Single Mode.

Incorpora un microprocesador ARM® Cortex™-M0 que se ejecuta a 16MHz, con 32kB de memoria PROM (memoria de sólo lectura programable una sola vez).

Disponible en empaquetamientos QFN 6x6mm, 5x5mm y WLCSP 2.5x2.5mm.

Voltaje de alimentación de0.9V-3.6V

Impedancia de salida para RF de 50 Ohm

10-bit ADC

Consumo de corriente en TX y RX de 4,9mA.

En las memorias tipo PROM (One-Time Programmable Memory), el valor de cada bit depende del estado de un fusible que puede ser quemado una sola vez. Por esto la memoria puede ser programada una sola vez. Para el desarrollo, la aplicación y el protocolo se cargan en la memoria vía JTAG. El dispositivo puede conectarse a una memoria externa vía SPI o I2C para cargar el firmware. Aunque no es tan flexible como tener una memoria flash en el chip, puede ser usada para actualizaciones de firmware.

Puesto que la salida para RF ya está ajustada a 50 Ohms, no son necesarios componentes externos.

Al ser un componente con el consumo de corriente más bajo actualmente en el mercado, es un opción atractiva para productos que necesiten consumir poca corriente durante muchos años.

Cypress Semiconductor PSoC 4 BLE / PRoC BLE

Cypress ha anunciado recientemente sus dispositivos BLE. Han integrado el BLE a su familia de dispositivos PSoC 4. Si un sistema puede hacer uso de los periféricos y de los bloques flexibles analógico/digitales de la arquitectura PSoC, entonces añadir BLE es sencillo. PRoC es otro dispositivo con menos funcionalidad que tiene radio BT LE. Algunas de las características son:

Versión de Bluetooth soportada BLE v4.1 Single Mode

Microprocesador ARM® Cortex™-M0 48MHz integrado en el chip con 127kB de memoria Flash y 16kB de RAM

Disponible en empaquetamientos QFN 6x6mm, 5x5mm y WLCSP 2.5x2.5mm

Disponibe de bloques flexibles digitales y analógicos PSoC

Voltaje de alimentación de 1.9V-5.5V

Consumo de corriente RX de 15,6mA y TX 16,4mA.

El alto uso de corriente de estos dispositivos no los predispone para aplicaciones de baja potencia donde el consumo es crítico, pero la alta frecuencia del reloj y a flexibilidad de sus partes hace que se les pueda añadir conectividad a productos ya creados sin comprometer el mismo.

[PSoC 4 BLE Information](http://www.cypress.com/psoc4ble/?source=BLE)

[PRoC BLE Information](http://www.cypress.com/PRoCBLE/?source=BLE)

Texas Instruments CC256X

El CC256x es el único dispositivo capaz de soportar el llamado Bluetooth clásico en una configuración Dual Mode. Esto permite trasmitir audio y soportar las funcionalidades típicas del Bluetooth. El CC256x está disponible para cualquiera, mientras que otras soluciones Dual Mode de otros comerciantes no son accesibles para pequeños fabricantes. Una característica importante del CC256x es que requiere un microcontrolador externo para ejecutar el protocolo Bluetooth.

Algunas de las principales características son:

Versión de Bluetooth soportada BLE v4.0 y Bluetooth clásico – Dual Mode.

Dual Mode aplicado en el nivel HCI.

Necesita de un microcontrolador externo ejecutando el protocolo.

Empaquetamiento 8.10×7.83mm

[More information on the CC256x](http://processors.wiki.ti.com/index.php/CC256x)

Texas Instruments CC2640 y CC2650

Normalmente, el microprocesador del dispositivo comparte tiempo y recursos entre la aplicación y el protocolo de comunicación BLE. En el caso de la familia CC26XX, existe un microprocesador cortex-M0 ejecutando las capas más bajas del protocolo BLE, incluyendo la capa física y de enlace. A este dispositivo se ha añadido un CORTEX-M3 que ejecuta la aplicación, las capas superiores del protocolo y algo de la capa de enlace.

Un microprocesador Cortex-M3 es más potente y puede ejecutar mejor algoritmos complejos, usados por ejemplo en monitores de fitness, controles industriales y automatización del hogar, por lo que se puede evitar en algunos casos incorporar un procesador externo.

Respecto a su predecesor, el CC2540/2541, el cuál su consumo de energía era demasiado alto, ha disminuido considerablemente este consumo. El CC26xx también tiene un controlador específico de periféricos diseñado para descargar de procesamiento el microprocesador y por lo tanto reduce el consumo.

Algunas de las especificaciones del dispositivo son:

Microprocesador ARM® Cortex™-M3 48-MHz con 128KB de memoria flash y 20KB de SRAM.

Voltaje de alimentación de 1,8V-3,8V.

Versión Bluetooth BL v4.1 Single Mode.

Sensibilidad -97dBm Rx.

Controlador de periféricos independiente.

Empaquetamientos disponibles QFN 7x7mm y WLCSP 3,9x3,5mm.