ZigBee/XBee

[10 respuestas](https://aprendiendoarduino.wordpress.com/2016/11/16/zigbeexbee/#comments)

**ZigBee**

**IEEE 802.15.4** es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (low-rate wireless personal area network, LR-WPAN). El grupo de trabajo [IEEE 802.15](https://es.wikipedia.org/wiki/IEEE_802.15) es el responsable de su desarrollo. También es la base sobre la que se define la especificación de [ZigBee](https://es.wikipedia.org/wiki/ZigBee), cuyo propósito es ofrecer una solución completa para este tipo de redes construyendo los niveles superiores de la pila de protocolos que el estándar no cubre.

IEEE802.15.4:

* <https://es.wikipedia.org/wiki/IEEE_802.15.4>
* <https://en.wikipedia.org/wiki/IEEE_802.15.4>
* <http://www.ieee802.org/15/pub/TG4.html>
* <https://es.wikipedia.org/wiki/IEEE_802.15>

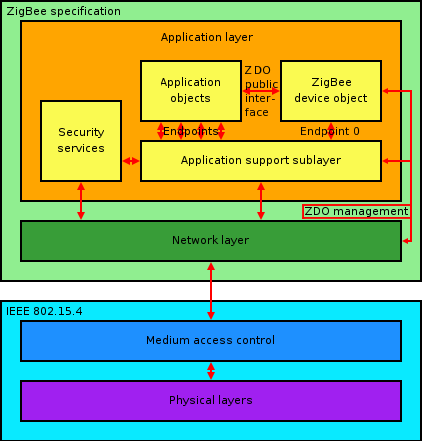
El estandar 802.15.4 simplificado: <http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf>

Bluetooth y RFID se encuentran en el grupo de trabajo 802.15.

ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar [IEEE 802.15.4](https://es.wikipedia.org/wiki/IEEE_802.15.4) de redes inalámbricas de área personal (wireless personal area network, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

En principio, el ámbito donde se prevé que esta tecnología cobre más fuerza es en domótica, como puede verse en los documentos de la ZigBee Alliance. La razón de ello son diversas características que lo diferencian de otras tecnologías:

* Su bajo consumo.
* Su topología de red en malla.
* Su fácil integración (se pueden fabricar nodos con muy poca electrónica).



ZigBee utiliza la [banda ISM](https://es.wikipedia.org/wiki/Banda_ISM) para usos industriales, científicos y médicos; en concreto, 868 MHz en Europa, 915 en Estados Unidos y 2,4 GHz en todo el mundo. Sin embargo, a la hora de diseñar dispositivos, las empresas optarán prácticamente siempre por la banda de 2,4 GHz, por ser libre en todo el mundo. El desarrollo de la tecnología se centra en la sencillez y el bajo costo más que otras redes inalámbricas semejantes de la familia [WPAN](https://es.wikipedia.org/wiki/WPAN), como por ejemplo Bluetooth.

Zigbee:

* <https://es.wikipedia.org/wiki/ZigBee>
* <https://en.wikipedia.org/wiki/ZigBee>
* Zigbee Alliance: <http://www.zigbee.org/>

Tabla comparativa de módulos 802.1.4 <https://en.wikipedia.org/wiki/Comparison_of_802.15.4_radio_modules>

El estándar Zigbee se utiliza primordialmente para aplicaciones domóticas donde es mínima la capacidad de transferencia de información y el costo y consumo tienen un papel fundamental. ZigBee se utiliza para controlar la calefacción, iluminación, sistema de seguridad, etc. de cualquier edificio inteligente. Se espera que ZigBee se aplique para industrias, juguetes, periféricos de PC, componentes electrónicos, sistemas de control automático, medicina, etc. pero en este momento su principal aplicación es en los sistemas de domótica y de automatización.

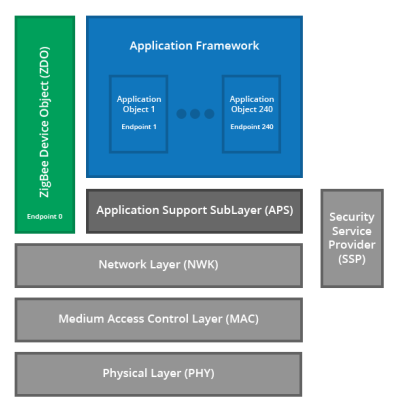
Zigbee Soporta tres tipos de topologías de red:

* Star o Estrella: presenta larga vida útil como consecuencia del bajo consumo que requiere.
* Mesh o Malla: en la cual existen múltiples rutas para alcanzar un destino, obteniéndose alta confiabilidad.
* Cluster Tree o Racimo de Árbol: es una topología del tipo Mesh-Star que encierra los beneficios de ambas.

Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth no utiliza FHSS (Frequency hooping), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de emisión del dispositivo así como el tipo de antenas utilizadas (cerámicas, dipolos, …) El alcance normal con antena dipolo en visión directa suele ser aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1mW de potencia) de 100m y en interiores de unos 30m. La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps. Por último decir que una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos. La realidad es menor, siendo, de todas formas, de miles de equipos.

El estándar Zigbee permite la interoperabilidad entre diferentes dispositivos de de diferentes fabricantes. Por ejemplo, nos permitiría controlar luces, sistemas de climatización, riego automático, etc.. de diferentes marcas usando un mismo protocolo con productos certificados Zigbee.

Para lograr esta interoperabilidad entre dispositivos, el protocolo Zigbee está organizado en capas que separan los componentes y las funciones en módulos independientes.

[](https://aprendiendoarduino.files.wordpress.com/2016/11/zigbee.png)

Las capas PHY y MAC son las especificadas por el estándar IEEE 802.15.4 y las capa NWK está especificada por el estándar Zigbee que maneja la estructura de red, enrutamiento y seguridad.

La capa de aplicación es la que habilita la interoperabilidad y está compuesta por:

* APS: ofrece un interfaz entre la red y la capa de aplicación, define los mensajes estandarizados que permite la comunicación entre dispositivos de diferentes fabricantes.
* Application Framework: es el entorno donde se almacenan las aplicaciones en los dispositivos ZigBee.
* ZDO (ZigBee Device Object): ofrece funcionalidades de descubrimiento de dispositivos y gestión avanzada de red.

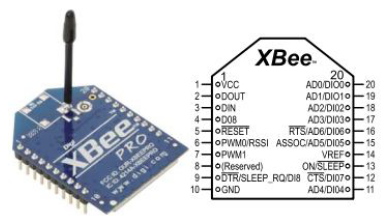
Más información: <http://www.electrocomponentes.com/articulos/diciembre06/zigbee.html>

**XBee**

**XBee**es el nombre comercial del Digi de una familia de módulos de comunicación por radio y están basados en el estándar zigbee, pero digi tiene muchos Xbee y algunos son zigbee estándar y otros son propietarios o modificaciones del estándar. Existen muchos módulos Xbee basados en el estándar [IEEE 802.15.4](https://en.wikipedia.org/wiki/IEEE_802.15.4)

Los módulos Xbee han sido diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible. Por lo que básicamente XBee es propiedad de Digi basado en el protocolo Zigbee. En términos simples, los XBee son módulos inalámbricos fáciles de usar.

Xbee es el nombre comercial de la familia de módulos de radio con un factor de forma compatible de [Digi International](https://en.wikipedia.org/wiki/Digi_International)



Los módulos de radio XBee pueden ser usados con un número mínimo de conexiones: Power (3.3V), GND y TX/RX de la UART, junto con otras conexiones recomendables como reset y sleep. La mayoría de los módulos XBee tienen otras conexiones omo [flow control](https://en.wikipedia.org/wiki/Flow_control_(data)), [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O), [analog-to-digital converter](https://en.wikipedia.org/wiki/Analog-to-digital_converter) (A/D) and indicator

Los módulos XBee funcionan a 3.3V y los pines no son tolerantes a 5V. Desde Arduino podemos alimentar un módulo XBee, pero la comunicación serie en Arduino es a 5V y en el módulo XBee es a 3.3V.

Es posible usar un módulo adaptador de 5V a 3.3V como este <https://www.parallax.com/sites/default/files/downloads/32401-XBee-5V-3.3V-Adapter-v1.2.pdf>

XBee:

* <https://en.wikipedia.org/wiki/XBee>
* <http://www.digi.com/lp/xbee>
* <http://examples.digi.com/>

La familia de módulos de radio XBee es:

* XBee 802.15.4 — The initial [point-to-point topology](https://en.wikipedia.org/wiki/Network_topology" \l "Point-to-point) or [star topology](https://en.wikipedia.org/wiki/Network_topology" \l "Star) module running the [IEEE 802.15.4](https://en.wikipedia.org/wiki/IEEE_802.15.4) protocol
* XBee-PRO 802.15.4 — A higher power, longer range version of the XBee 802.15.4
* XBee DigiMesh 2.4 — A 2.4 GHz XBee module that uses DigiMesh, a sleeping mesh networking protocol developed by Digi International
* XBee-PRO DigiMesh 2.4 — A higher power, longer range version of the XBee DigiMesh 2.4
* XBee ZB — An XBee module that incorporates the [ZigBee](https://en.wikipedia.org/wiki/ZigBee) PRO mesh networking protocol
* XBee-PRO ZB — A higher power, longer range version of the XBee ZB
* XBee ZB SMT — A surface mount XBee running the ZigBee protocol
* XBee-PRO ZB SMT — A higher power, longer range version of the XBee ZB SMT
* XBee SE — An XBee ZB module that incorporates the security cluster for the [ZigBee Smart Energy](https://en.wikipedia.org/w/index.php?title=ZigBee_Smart_Energy&action=edit&redlink=1) public profile
* XBee-PRO SE — A higher power, longer range version of the XBee SE
* XBee-PRO 900HP — A 900 MHz XBee-PRO module with up to 28 mile range with high-gain antenna that supports DigiMesh networking protocol
* XBee-PRO 900 (Legacy) — A [900 MHz](https://en.wikipedia.org/wiki/33-centimeter_band) proprietary point-to-point and star topology module, not recommended for new design
* XBee-PRO XSC (S3B) — A 900 MHz module compatible over the air with the Digi 9XStream radios
* XBee-PRO DigiMesh 900 (Legacy) — A 900 MHz module that uses DigiMesh, not recommended for new design (see XBee-PRO 900HP for new designs)
* XBee-PRO 868 — An [868 MHz](https://en.wikipedia.org/wiki/ZigBee#Radio_hardware) 500 mW long-range module that supports proprietary point-to-point and star, for use in Europe
* XBee 865/868LP — An 868 MHz XBee module that uses DigiMesh, available in Surface Mount form-factor (also configurable to [865 MHz](https://en.wikipedia.org/w/index.php?title=865_MHz&action=edit&redlink=1) for use in India)
* XBee ZigBee (S2C) — Incorporates an upgrade to the transceiver chip, replacing the [Silicon Labs](https://en.wikipedia.org/wiki/Silicon_Labs) EM250 with the Silicon Labs EM357, effectively adding more RAM, more flash, faster clock speed and lowering the current draw.[[11]](https://en.wikipedia.org/wiki/XBee#cite_note-11)
* XBee-PRO ZigBee (S2C) — A higher power, longer range version of the XBee ZigBee (S2C)

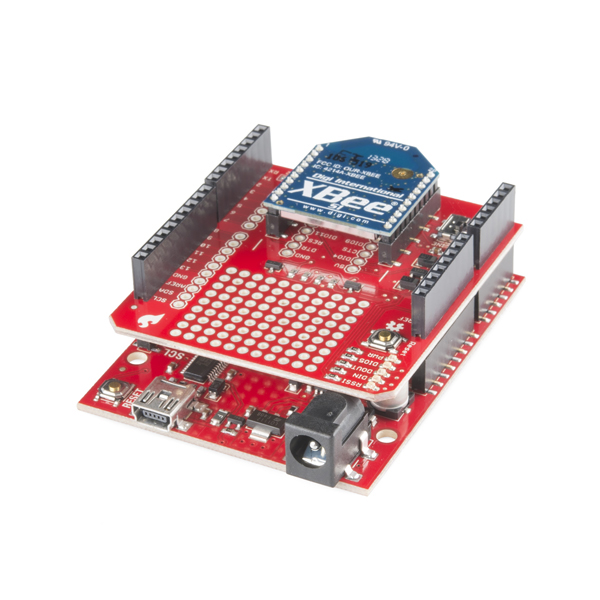
Nuevos módulos de XBEE SMD para Europa: <http://www.digi.com/pdf/ds_xbee868lp.pdf>. Son los 868LP operan en la frecuencia 863-870 MHz utilizando 20 canales para ofrecer más referencia e inmunidad en 868 MHz.

Kits de aprendizaje de XBee de Digi:

* XBee Arduino compatible Coding Platform: <https://docs.digi.com/display/XBeeArduinoCodingPlatform/XBee+Arduino+Compatible+Coding+Platform>
  + Comprarlo: <http://www.digikey.es/product-detail/en/digi-international/XKB2-AT-WWG/602-1550-ND/5271212>
* Wireless Connectivity Kit: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm>
  + Comprarlo: <http://www.digikey.es/product-detail/en/digi-international/XKB2-AT-WWC/602-1551-ND/5305247>

Los módulos más sencillos de Xbee son los serie 1 (también llamada 802.15.4) que no soportan mesh e implementan el estándar 802.15.4. Son los más fáciles de usar y los más recomendados para empezar a trabajar. Más información: <http://www.digi.com/pdf/ds_xbeemultipointmodules.pdf> y ejemplo de uso: <http://examples.digi.com/get-started/basic-xbee-802-15-4-chat/>

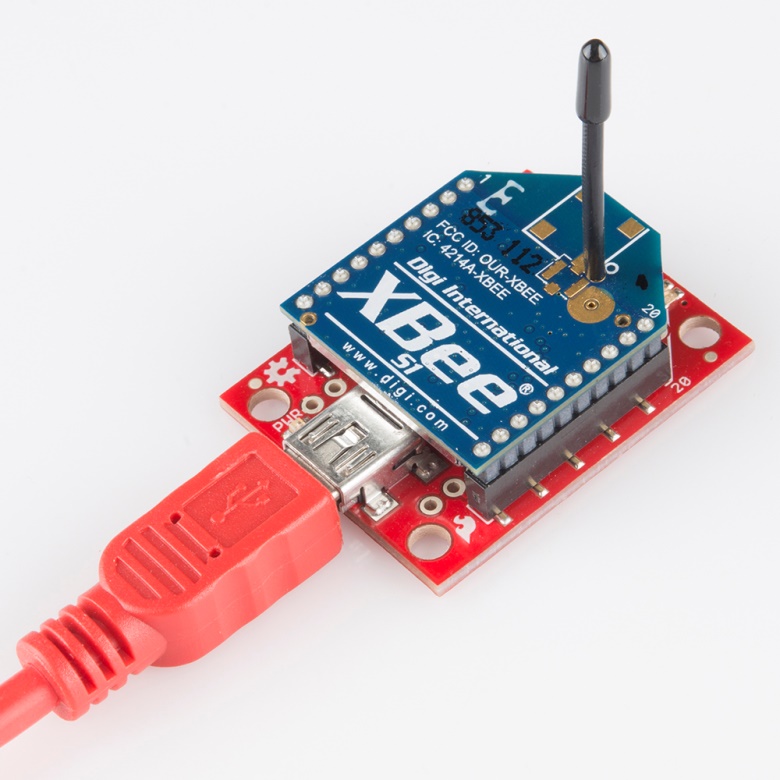
Para usar un módulo XBee con Arduino es necesario un Shield o un adaptador para conectar el puerto serie de XBee con el de Arduino.



Para usar o configurar un módulo XBee con Ordenador es necesario un adaptador que generalmente tiene un chip FTDI que hace de pasarela entre el puerto serie y el USB.

XBee explorer:

* <http://tienda.bricogeek.com/modulos-radiofrecuencia/156-xbee-explorer-usb.html>
* <http://www.electan.com/xbee-explorer-usb-p-3121.html>
* Dongle <https://www.sparkfun.com/products/11697>
* Versión económica <http://es.aliexpress.com/item/FT232RL-USB-to-Serial-Port-Bee-Adapter-Board-Foca-Compatible-with-XBee-USB-Adapter/1959031547.html>
* Otra versión económica <http://www.dx.com/p/arduino-ft232rl-xbee-usb-to-serial-adapter-v1-2-board-module-blue-140960#.V5-vxfmLTcs>
* Hazlo tu mismo <http://www.kobakant.at/DIY/?p=204>



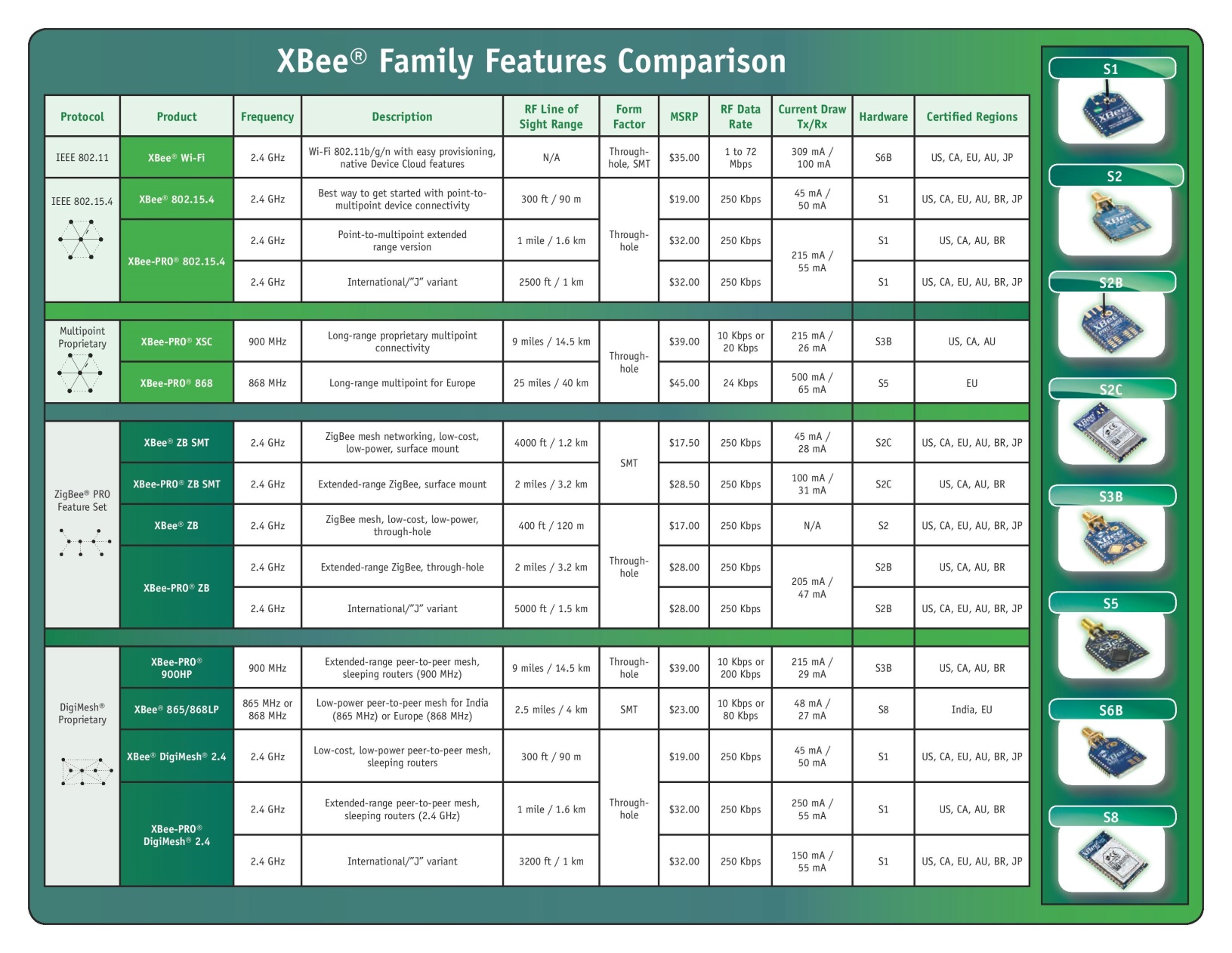
Guía de uso de XBee Shield: <https://learn.sparkfun.com/tutorials/xbee-shield-hookup-guide>

Buena página para saber más de XBee: <http://xbee.cl/>

**Guía de compra de XBee**

Digi dispone de una amplia variedad de módulos con diferentes características para diferentes funciones/aplicaciones. A la hora de elegir, esta tabla de comparación puede ser útil: <http://www.digi.com/pdf/chart_xbee_rf_features.pdf>

Y en este enlace también explica perfectamente como elegir nuestro módulo XBee: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/XBee+buying+guide>



Los parámetros que afectan a la hora de elegir un módulo en función de la aplicación son:

* El país de uso, las frecuencias están reguladas en cada país y debe tenerse en cuenta.
* Alcance de la señal. Para un mayor rango será necesario seleccionar una antena o usar un módulo XBee-PRO
* Consumo del módulo
* Topología de la red de nuestra aplicación.

Guia de compra de módulos XBee de sparkfun: <https://www.sparkfun.com/pages/xbee_guide>

Resellers de Digi para compras on-line:

* [Adafruit](http://www.adafruit.com/search?q=xbee&b=1)
* [Fry’s](http://www.frys.com/search?search_type=regular&sqxts=1&cat=&query_string=xbee)
* [Maker Shed](http://www.makershed.com/pages/search-results?q=xbee&p=1)
* [Microcenter](http://www.microcenter.com/search/search_results.aspx?Ntt=xbee)
* [Parallax](https://www.parallax.com/search?search_api_views_fulltext=xbee)
* [RobotShop](http://www.robotshop.com/en/catalogsearch/result?q=xbee&order=stats_sales_order_count&dir=desc)
* [Seeed Studio](http://www.seeedstudio.com/depot/s/xbee.html?search_in_description=0)
* [Solarbotics](https://solarbotics.com/search/xbee/)
* [Sparkfun](https://www.sparkfun.com/search/results?term=xbee)
* [TrossenRobotics](http://www.trossenrobotics.com/store/Search.aspx?SearchTerms=xbee)

Para cualquier duda o ayuda, se puede recurrir al foro de digi en <http://www.digi.com/support/forum/>

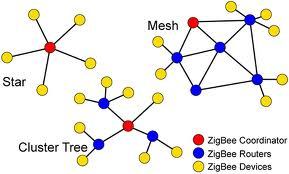
**Arquitectura Básica de una Red XBee**

Una red XBee la forman básicamente 3 tipos de elementos. Un único dispositivo Coordinador, dispositivos Routers y dispositivos finales (end points). Los módulos XBee son versátiles a la hora de establecer diversas topologías de red, dependiendo la serie de XBee que escojamos pueden crearse redes

**El Coordinador**: Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones (como hablábamos antes) y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el Coordinador puede formar una red, permitiendo unirse a él a dispositivos Routers y End Points. Una vez formada la red, el Coordinador hace las funciones de Router, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información.

**Los Routers**: Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para enrutar un paquete de información. Lógicamente un router debe unirse a una red Zigbee antes de poder actuar como Router retransmitiendo paquetes de otros routers o de End points.

**End Device:** Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un Router, es decir, no puede enviar información directamente a otro end device. Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.



Los módulos XBee son versátiles a la hora de establecer diversas topologías de red, dependiendo la serie de XBee que escojas puedes crear redes:

* Punto a punto
* Estrella
* Malla
* Árbol
* Mixtas

**Módulos XBee**

Los módulos XBee de [digi](http://www.digi.com/)son pequeños módulos RF (radio frecuencia) que transmiten y reciben datos sobre el aire usando señales de radio. La capacidad inalámbrica es esencial cuando se quieren instalar sensores en lugares donde no hay cables.

Los módulos XBee son altamente configurables y soportan múltiples protocolos para permitir usarlo tanto en un enlace punto a punto como en un diseño complejo con muchos dispositivos en una red mesh.

Algunos ejemplos de uso de los módulos XBee:

* Controlar un robot remotamente
* Añadir inteligencia a una casa edificio sin tener que realizar cableados
* Para aplicaciones industriales, por ejemplo para monitorizar temperatura, presión o máquinas complejas.

Más información en: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#concepts/c_introduction_to_xbee_modules.htm%3FTocPath%3D_____3>

**Las series**

**XBee Series 1** (también llamados XBee 802.15.4)  – Son la serie más fácil para trabajar, no necesitan ser configurados, pero incluso así se pueden obtener beneficios de estos módulos. Debido a que son fáciles para trabajar, son los más recomendables especialmente si se está empezando. Para comunicaciones Punto-a-Punto, estos módulos trabajan tan bien como los de la Serie 2, pero sin todo el trabajo de pre configuración previo. El hardware de las Series 1 y las Series 2/2.5/ZB NO SON COMPATIBLES. No trates de mezclarlos, no funcionará.

Guia de usuario serie 1:

<http://www.hmangas.com/Electronica/Datasheets/Shield%20XBee%20Arduino/XBee-Guia_Usuario.pdf>

**XBee Znet 2.5 (Formalmente Series 2) Retirado** – Los módulos Serie 2 deben ser configurados antes de ser usados. Pueden funcionar en modo Transparente o por medio de comandos API, pero todo esto depende de que firmware se configure en los módulos. También pueden funcionar en una red mesh. Son más difíciles que usar que los de la Serie 1. No existe una forma en que estos módulos sean compatibles con los de la Serie 1. Los módulos Znet 2.5 ya no se venden, pero han sido reemplazados con módulos ZB más compatibles. [Datasheet](http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-2.5-Manual.pdf)

<https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-2.5-Manual.pdf>

**XBee ZB (el actual módulo Series2)** – Básicamente es el módulo Znet 2.5, pero con un nuevo firmware. Esto significa que también funcionan en modo transparente o por medio de comandos API. También funcionan en redes mesh. Estos a menudo son llamados módulos de Serie 2, por lo que si escuchas a alguien hablar sobre esta serie, probablemente estén hablando de estos módulos. Puede que no sea el término correcto, pero se hace distinción de estos con los módulos de la Serie 1, los cuales son los más populares.

**XBee 2B (el imás actual módulo Series2)** – Son nuevos módulos que poseen mejoras en el hardware respecto de los de la Serie 2, básicamente son los mismo que los anteriores pero con un firmware más nuevo, mejorando por ejemplo el uso de la potencia. Funcionan con el Firmware del módulo ZB, pero debido al cambio de hardware, ya no pueden funcionar con el firmware del módulo Znet 2.5. Por lo que ten cuidado si agregas uno de estos módulos a una red ya existente que utilice módulos Znet 2.5. Actualmente algunas tarjetas son 2B y otras son ZB.

Básicamente el problema es mezclarlos.

**900 MHz vs 2.4GHz** – La mayoría de los módulos XBee operan a 2.4 GHz , pero hay unos pocos que operan a 900 MHz. Básicamente los de 900 MHz pueden llegar muy lejos con una antena de alta ganancia (hasta casi 24 Km). Además a menor frecuencia, la señal posee mayor penetración. Otro punto importante es que los módulos de 900 MHz no están permitidos en algunos países,  Digi tiene versiones de 868 MHz que sí está permitido en la mayoría de los países.

Documentación módulos XBee:

* Módulos: <http://www.digi.com/products/xbee-rf-solutions/modules>
* Series 1 (XBee 802.15.4): <http://www.digi.com/products/xbee-rf-solutions/modules/xbee-series1-module>
* XBee ZB (Series 2): <http://www.digi.com/products/xbee-rf-solutions/rf-modules/xbee-zigbee>
* XBee digimesh: <http://www.digi.com/products/xbee-rf-solutions/modules/xbee-digimesh-2-4>
* Diferencias entre serie 1 y serie 2: <http://knowledge.digi.com/articles/Knowledge_Base_Article/The-Major-Differences-in-the-XBee-Series-1-vs-the-XBee-Series-2>

Más información <http://xbee.cl/tutorial-xbee/>

Regulaciones en paises: <http://www.digi.com/resources/certifications>

Toda la documentación de digi: <http://www.digi.com/resources/documentation/digihelp/default.htm>

**XCTU**

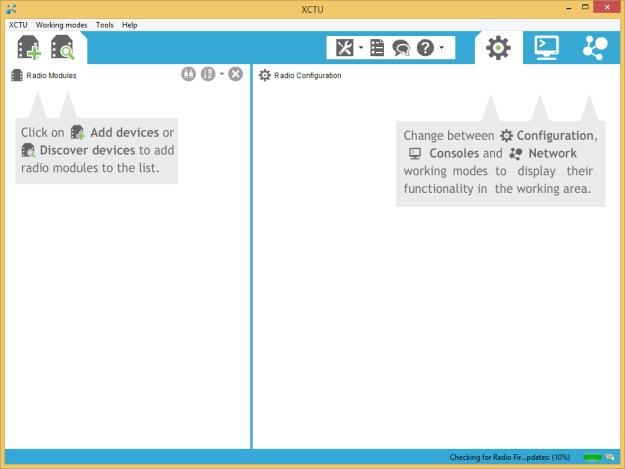
Para configurar y usar los módulos XBee es necesario descargar e instalar XBee Configuration and Test Utility (XCTU) que es un software multiplataforma que permite interactuar con los módulos mediante un interfaz gráfico. Esta aplicación incluye herramientas que hacen muy sencillo configurar y probar los módulos XBee.

XCTU nos sirve para configurar, inicializar, actualizar firmware y testear los módulos XBee, comunicándose por puerto serie a los módulos. Una ventaja de este software es que puedes ver rápidamente un resumen de todos los parámetros del módulo y una descripción de ellos.

Más información y descarga de XCTU: <http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>

XCTU User Guide: <http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm>

Drivers para los módulos: <http://www.digi.com/support/productdetail?pid=3257&type=drivers>

[](https://aprendiendoarduino.files.wordpress.com/2016/06/xctu.png)

Para empezar a usarlo es tan simple como instalar el software XCTU, conectar el módulo RF a la placa de desarrollo, que tiene un módulo FTDI que saca un interfaz USB y luego enchufar el USB a nuestro PC. Para usarlo con USB es necesario poner el jumper en la posición UART en lugar de la de loopback.

Más información de la placa de desarrollo de XBee: <https://docs.digi.com/display/XBeeHardware/XBee+Grove+Development+Board>

Tutorial XCTU: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/XCTU+walkthrough>

Cómo configurar los módulos: <http://docs.digi.com/display/XCTU/Configure+your+modules>

**Primera Aplicación. Chat Básico**

El primer ejemplo es un chat básico para transmitir en tiempo real mensajes por el aire con los módulos XBee.

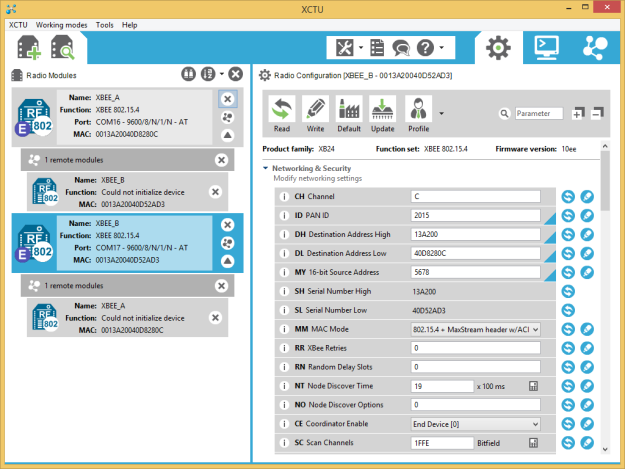


**Ejercicio completo**: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_basic_chat.htm%3FTocPath%3DBasic%2520chat%7C_____0>

Para configurar un módulo XBee con el XCTU, lo primero es poner el modo configuración y descubrir los módulos, seleccionando el puerto COM del USB al que he conectado la placa de desarrollo.

Para poder ver el módulo remoto es necesario configurar los parámetros DH y DL de la MAC del módulo remoto.

**NOTA**: Si se conectan los módulos en las placas de desarrollo y se alimenta mediante batería, los módulos trabajan de forma autónoma, por ejemplo recogiendo datos de un sensor y mandandolo al nodo central

[](https://aprendiendoarduino.files.wordpress.com/2016/06/xctu2.png)

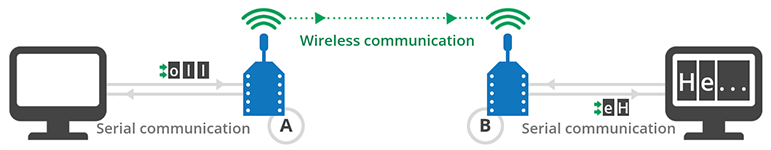
**Cómo se comunican los dispositivos XBee**

Los dispositivos XBee se comunican entre ellos a través del aire enviando y recibiendo mensajes. Estos dispositivos no pueden gestionar los datos enviados o recibidos, sin embargo pueden comunicarse con otros dispositivos a través del interfaz serie.

Los dispositivos XBee transmiten al aire los datos que llegan del puerto y transmiten al puerto serie cualquier dato que llega por el aire. Los microcontroladores o los PCs pueden controlar que envían los dispositivos XBee y gestionan los mensajes inalámbricos entrantes.

Por lo tanto tenemos dos tipos de comunicación en los dispositivos XBee:

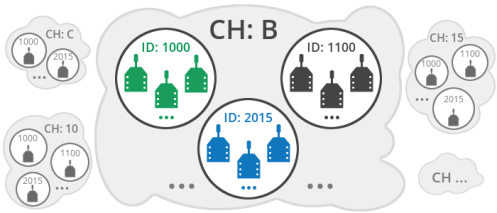
* Comunicación inalámbrica: es la comunicación entre los módulos XBee, estos módulos deben ser parte de la misma red y usar la misma frecuencia de radio.
* Comunicación serie: es la comunicación entre el módulo XBee y el microcontrolador o el PC a través de un puerto serie.



En la comunicación inalámbrica los módulos transmiten y reciben información a través de la modulación de las ondas electromagnéticas. Para que se realice la transmisión ambos módulos deben estar en la misma frecuencia y en la misma red. Esto se determina por dos parámetros:

* **Channel (CH)** es la frecuencia usada para comunicar, es decir, el canal dentro de la red.
* **Personal Area  Network Identifier (ID)** es un identificador único que establece que los módulos están en la misma red.

Un módulo XBee solo recibirá y transmitirá datos a otros XBee dentro de las misma red (mismo ID) y usando el mismo canal (mismo CH).



**Direccionamiento**

Los módulos de XBee tienen varias direcciones diferentes, cada una con un propósito.

|  |  |  |
| --- | --- | --- |
| Type | Example | Unique |
| 64-bit | 0013A20012345678 | Always |
| 16-bit | 1234 | Yes, but only within a network |
| Node identifier | Bob’s module | Uniqueness not guaranteed |

Cada módulo de XBee tiene una dirección única de 64 bits, esta dirección se llama MAC y es análogo a la MAC de las tarjetas de red o wifi. El valor de 64 bits está compuesto por los parámetros Serial Number High (SH) y Serial Number Low (SL), que aparecen impresos en la parte trasera del módulo. El valor SH es generalmente el mismo para todos los módulos XBee (0013A200) e identifica los módulos de Digi. La dirección 000000000000FFFF está reservada para mandar un mensaje de broadcast.

Una dirección de 16 bit puede ser asignada a cada módulo XBee y no es única. Este valor puede leerse o escribirse a través del parámetro MY. Si el valor es FFFF se deshabilita la recepción de paquetes con direcciones de 16 bit.

El identificador de nodo es un string corto que permite identificar fácilmente un módulo con un nombre. Para leer o escribir el node identifier se hace a través del parámetro NI.

**Comunicación serie**

Un módulo XBee puede funcionar como un módulo independiente o conectado a un microcontrolador o PC. Cuando opera como módulo independiente simplemente manda datos al nodo central de los sensores o dispositivos conectados a los puertos del módulo. Cuando está conectado a un microcontrolador o un PC, el módulo XBee usa la comunicación serie. En el caso de Arduino, este manda a través del puerto serie datos al módulo XBee para que sean transmitidos por el aire.

El módulo XBee hace de interface con el microcontrolador a través de la UART (puerto serie asíncrono). Para más información ver las guías de usuario:

* XBee/XBee-PRO S1 802.15.4 <http://ftp1.digi.com/support/documentation/90000982.pdf>
* XBee/XBee-PRO S2C 802.15.4 <http://ftp1.digi.com/support/documentation/90001500.pdf>

Un arduino o el dispositivo externo conectado a uno módulo XBee mediante puerto serie puede tener varios modos de operación en función de cómo se comunican por el puerto serie. Los módulos XBee soportan dos modos de operación:

* Modo Transparente (Aplicación transparente). La radio pasa la información tal cual la recibe por el puerto serie. Este modo tiene funcionalidades limitadas pero es la forma más sencilla de comenzar.
* Modo API (Aplicación de programación). En este caso un protocolo determina la forma en que los datos son intercambiados. Este modo permite hacer una red de comunicaciones más grande.

Ventajas y desventajas de un modo u otro: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#concepts/c_xbee_comparing_at_api_modes.htm%3FTocPath%3DHow%2520XBee%2520devices%2520work%7CWireless%2520communication%7CSerial%2520communication%7C_____2>

Más información en: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_how_xbees_work.htm%3FTocPath%3DHow%2520XBee%2520devices%2520work%7C_____0>

**Modo Transparente**

En modo transparente el módulo XBee funciona de forma que todo lo recibido por el puerto serie es enviado inmediatamente al aire y todo lo recibido por la radio es se manda tal cual al puerto serie.

En modo transparente para comunicar dos módulos, es necesario configurar la dirección del destino en el módulo que envía datos. Esta dirección se programa en los parámetros:  Destination Address High (DH) y Destination Address Low (DL).

El modo transparente tiene muchas limitaciones, por ejemplo al trabajar con varios módulos es necesario configurar la dirección de destino antes de mandar un mensaje. Sin embargo, el modo transparente es perfecto cuando hay comunicación punto a punto entre solo dos elementos.

En modo transparente podemos usar el modo comando que es un estado en que los caracteres enviados al módulo XBee son interpretados como comandos en lugar de transmitirlos vía radio. Para cambiar a modo comando hay que mandar el string “+++”, cuando el módulo recibe un segundo de silencio seguido del string “+++” (sin /n o /r) y otro segundo de silencio, deja de mandar datos por radio y comienza a aceptar comandos locales. Al entrar en modo comando si transcurren 10 segundo sin recibir datos automáticamente sale de modo comando y vuelve a modo transparente. Para salir del modo configuración usar ATCN.

El propósito del modo comando es leer o escribir la configuración local del módulo XBee. Para ello se usando los comandos AT (<https://es.wikipedia.org/wiki/Conjunto_de_comandos_Hayes>) . Un comando AT comienza con las letras “AT” seguida de dos caracteres que identifican el comando a ejecutar y algunos otros parámetros opcionales.

Por lo tanto desde un Arduino sería muy sencillo configurar un módulo XBee en lugar de usar el interfaz gráfico XCTU.

Más información en: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_xbee_transparent_mode.htm%3FTocPath%3DXBee%2520transparent%2520mode%7C_____0>

**Modo API**

El modo API ofrece un interfaz estructurado donde los datos son comunicados a través del interfaz serie en paquetes organizados y en un determinado orden. Esto permite establecer una comunicación compleja entre módulos sin tener que definir un protocolo propio.

Por defectos los módulos XBee trabajan en modo transparente, pero esto tiene unas limitaciones:

* Si un módulo necesita transmitir mensajes a módulos diferentes debe cambiar la configuración para establecer el nuevo destino.
* Para leer y escribir la configuración del módulo hay que entrar primero en modo comando.
* En modo transparente un módulo XBee no puede distinguir el origen de la comuinicación que recibe.

Para resolver estas limitaciones XBee da la alternativa del modo API (Application Programming Interface).



Ventajas del modo API:

* Configurar módulos locales y remotos en la red y sin necesidad de entrar en modo comando.
* Comunicar con uno o varios destinatarios
* Identificar el módulo que ha mandado la comunicación
* Recibir el estado de la transmisión de los paquetes
* Obtener la fuerza de la señal de los paquetes recibidos
* Hacer gestión y diagnóstico de la red.
* Hacer funciones avanzadas domo actualización de firmware remota.

**Estructura de la Trama**

En modo API los datos enviados están estructurados en una trama. Se envían a través de interfaz serie de XBee y contiene el mensaje inalámbrico e información adicional como la calidad de la señal.

La estructura de la trama es:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Start  delimiter | Length | | Frame data | | | | | | | | Checksum |
| Frame  type | Data | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | … | n | n+1 |
| 0x7F | MSB | LSB | API  frame  type | Frame-type-specific data | | | | | | | Single  byte |

API frame type determina el tipo de trama e indica cómo está organizada la información de los datos enviados. Los tipos de tramas soportadas dependen del tipo de módulo de Xbee que usemos. Más información para lo módulos usados en página 85 de <http://ftp1.digi.com/support/documentation/90000982.pdf>

Checksum ayuda a comprobar la integridad de los datos mandados. Las tramas enviadas por puerto serie con checksum incorrecto nunca serán procesadas por el módulos y serán ignoradas.

Para configurar el modo AP, se modifica el parámetro AP. Permite seleccionar entre los dos modos API soportados y el modo por defecto que es el transparente.

|  |  |  |
| --- | --- | --- |
| Mode | AP value | Description |
| Transparent | 0 | API modes are disabled and the module operates in transparent mode |
| API 1 | 1 | API mode without escaped characters |
| API 2 | 2 | API mode with escaped characters |

La diferencia ente API 1 y API 2 es que las tramas usan caracteres de escape. Puesto que la transmisión por el aire solo es del payload, dos módulos con diferente configuración API 1 y API 2 pueden comunicarse sin problemas.

El modo AP non-escaped (API 1) se basa únicamente en el delimitador de inicio y la longitud de los bytes para diferenciar las tramas, si los bytes en un paquete se pierden, la cuenta de bytes será incorrecta y el siguiente paquete (trama) también se perderá. API escaped (API 2) implica secuencias de caracteres de escape en una trama para mejorar la fiabilidad especialmente en entornos ruidosos.

La estructura de la trama es básicamente la misma en ambos modos API pero en API 2, todos los bytes excepto el delimitador de inicio deben estar escapados si es necesario. Los siguientes bytes de datos deben ser escapados en modpo API 2:

* 0x7E: Start delimiter
* 0x7D: Escape character
* 0x11: XON
* 0x13: XOFF

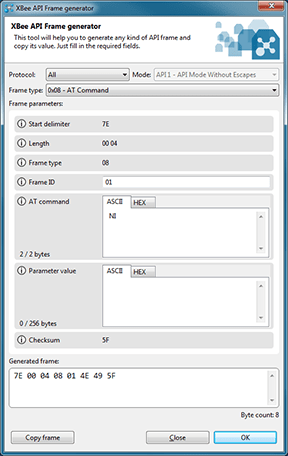
El modo API 2 garantiza que todos los bytes 0x7E recibidos son delimitadores de inicio, este carácter no puede ser parte de cualquier otro campo de la trama (longitud, datos o checksum), puesto que debe estar escapado. Para escapar un carácter, insertar 0x7D (carácter de escape) y añadirlo con el byte a ser escapado (XOR con 0x20). En modo API 2 la longitud no incluye los caracteres de escape y el checksum es calculado con los datos no escapados.

**Configurar un módulo local XBee en modo API**

Para consultar o modificar un valor de configuración de un módulo local (módulo conectado directamente al puerto serie), es necesario usar los comandos AT. Son los mismos comandos usados en modo transparente pero incluyendolo en una trama de tipo comando AT (0x08 o 0x09) y la respuesta vuelve en una trama de tipo respuesta AT (0x88).

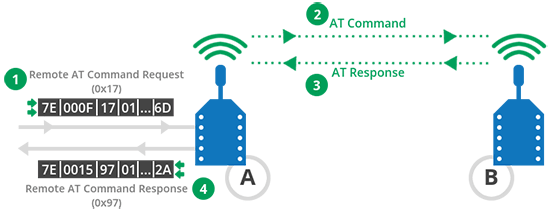
**Ejemplo**: Configurar un módulo local en modo API. <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_sending_frames.htm>

Cuando un módulo está en modo API, al usarlo con el SW XCTU y entrar en la consola, la vista es diferente a si está en modo transparente. Además desde consola hay una utilidad para generar una trama con el frame generator.



**Configuración Remota de un módulo XBee**

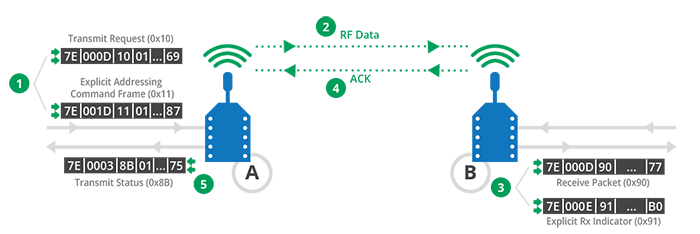
Trabajar en modo API nos permite configurar un módulo remoto al que estamos conectados. Cualquier comando AT usado en local se puede usar en un módulo remoto. Para ello se manda una Remote AT Command Request (0x17) al módulo remoto.



La respuesta del módulo remoto al comando AT es Remote AT Command Response (0x97) con el resultado del comando procesado por el módulo remoto. Si el frame ID del  Remote AT Command es0, esta respuesta no es enviada.

**Transmitir y recibir datos inalámbricamente en modo API**

Esta imagen resume el intercambio de información en los puestos serie de los módulos.



1. Se manda por el puerto serie una Petición de transmisión (0x10) o un Explicit Addressing Command Frame (0x11) al módulo XBee A.
2. El módulo XBee A transmite por el aire los datos de la trama al módulo configurado como destino, en este caso el módulo XBee B
3. El módulo remoto B recibe los datos por el aire y manda por su puerto serie una trama de paquete recibido (0x90) o Explicit Rx Indicator (0x91), dependiendo del valor de API Options (AO). Esta trama contiene los datos recibidos por el aire y la dirección de origen del módulo que la ha transmitido, en este caso XBee A.
4. El módulo remoto B transmite un paquete de wireless acknowledge con el estado al XBee A
5. El módulo XBee A manda un paquete Transmit Status (0x8B) a través de su puerto serie con el estado de las transmisión.

**Ejemplo**: Transmitir y recibir datos en modo API <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_transmit_and_receive_data_with_xctu.htm> (El tipo de trama es un Tx Request 0x00 del protocolo 802.15.4 y no como pone en la práctica un Transmit Request 0x10 del protocolo ZigBee)

Más información en <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_api_mode.htm>

**Librerías Xbee**

Para poder manejar el modo API disponemos de librerías que nos ayudan a generar la trama a enviar y a interpretar la trama recibida. Estas librerías nos ayudan a interpretar las tramas de API.

* **XBee mbed Library** para dispositvos ARM mbed <https://www.mbed.com/en/>. Más detalles en: <https://developer.mbed.org/teams/Digi-International-Inc/code/XBeeLib/>.
* **Digi XBee Ansi C Library** es una librería en ANSI C para comunicar con módulos XBee en modo API. Más detalles en: <https://github.com/digidotcom/xbee_ansic_library/>.
* **XBee-arduino** es una librería de Arduino para comunicar con módulos XBee en modo API. Más información en: <https://github.com/andrewrapp/xbee-arduino>
* **XBee Java Library** librería desarrollada en Java que permite interactuar como módulos XBee en modo API. Más información en: [XBee Java Library documentation](https://docs.digi.com/display/XBJLIB).

**Seguridad y cifrado**

Los módulos XBee pueden ser configurados para comunicación segura mediante claves de cifrado.

Los datos son cifrados antes de ser enviados y luego en el receptor son descifrados para poder usarlos. Activar la seguridad en una red XBee puede producir un ligero incremento en latencia y tamaño de los paquetes enviados.

Para habilitar la comunicación segura se deben configurar los siguientes parámetros con el mismo valor en todos los dispositivos de la red:

* EE = 1: AES Encryption Enable
* KY: AES Encryption key que debe ser un string de 32 caracteres hexadecimales. Una vez configurado este valor es imposible recuperarlo.

El cifrado es de 128 bits AES.

**Ahorro de energía**

Los módulos XBee tienen capacidades de ahorro de energía. los módulos se pueden poner en estado sleep y apenas consumir energía, pudiendo llegar a una duración de batería de varios años.

El protocolo 802.15.4 contiene cuatro conductas básicas para el modo sleep que se puede dividor en dos categorías: pin-controlled sleep mode y cyclic sleep mode. Por defectos el modo sleep está siempre deshabilitado.

Los parámetros básicos para configurar en modo sleep son:

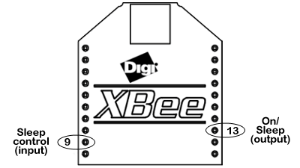
* SM (Sleep Mode)
* ST (Time before sleep)
* SP (Cyclic sleep period)

**Modo pin-controlled sleep**: este modo es controlado por el Sleep\_RQ (pin 9) de forma que cuando es puesto a HiGH (3.3 V) entra en modo sleep. (SM = 1)

**Modo Cyclic Sleep mode**: el módulo se despierta y vuelve a modo sleep en con una programación fija basada en el tiempo. Con SM = 4 se activa el modo y con SM = 5 además de activar el modo cíclico, permite activar a través del pin 9, siendo una mezcla de ambos modos. En estos dos modos se debe configurar los parámetros ST y SP.

Para comunicación bidireccional en modos sleep, los protocolos ZigBee o DigiMesh pueden dar funcionalidades más apropiadas.

Los pines de los módulos relacionados con el modo sleep son el pin 9 que pone el módulo en modo sleep cuando está a HIGH (3.3V) y el pin 13 es una salida que se pone a HIGH cuando está despierto o a LOW cuando está como sleep, este pin se puede conectar a un led o a una entrada del microcontrolador.



Para más información ver guía de usuario del correspondiente módulo: <http://ftp1.digi.com/support/documentation/90000982.pdf>

Más información en: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_explore_low_power.htm>

Ejercicio: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#tasks/t_ex_enable_sleep.htm>

**Comandos AT**

Para entrar en modo comando a un XBee hay que poner la cadena “+++” y esperar a que nos devuelva un “OK” el módulo. El propósito es leer o cambiar la configuración del módulo XBee.

Comandos AT:

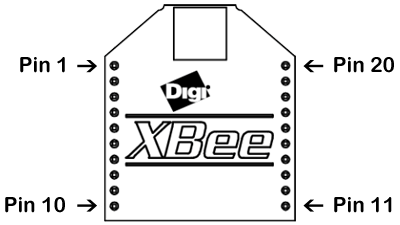
* AT: comprueba la conexión con el módulo
* ATCN: se sale del módulo de comandos.
* ATWR: Escribe la configuración actual a la memoria no volátil y persiste después de iniciar de nuevo el módulo.

Todos los comandos para la serie 1 pueden verse en la guia de usuario: <http://ftp1.digi.com/support/documentation/90000982.pdf>

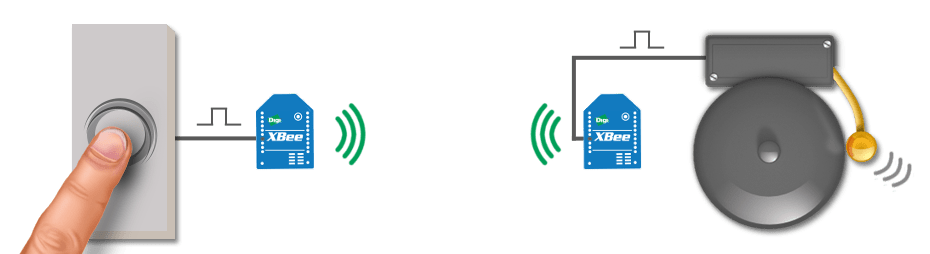
Más información en: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#concepts/c_at_commands.htm%3FTocPath%3DXBee%2520transparent%2520mode%7CCommand%2520mode%7C_____1>

**Pin Pairing**

Todos los módulos XBee tienen un conjunto de pines que pueden usarse como entradas y salidas sin necesidad de conectar un microcontrolador externo.



Pin pairing se refiere a la comunicación directa de los pines entre dos módulos XBee. Esto enlaza virtualmente uno de los pines del XBee directamente con a los pines de otro XBee, por ejemplo para un timbre inalámbrico.



Los módulos de la serie 1 tienen 9 pines de I(O, seis de los cuales pueden leer valores analógicos.

**Ejercicio completo** en: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#reference/r_pin_pairing.htm%3FTocPath%3DXBee%2520transparent%2520mode%7CPin%2520pairing%7C_____0>

Aclaraciones:

* Al poner el parámetro MY a FFFF habilita la recepción de paquetes de direcciones 64 bits deshabilitando la recepción de paquetes de direcciones de 16 bit.
* Señalar que el parámetro I/O Input Address (IA) enlaza una salida de XBee a una dirección de 64 bit específica. Esto significa que las salidas digitales de un XBee configuradas con el parámetro IA sólo aceptarán cambios de las peticiones que lleguen de la dirección configurada. De ahí la necesidad de configurar este parámetro en el XBee\_B.
* En el ejercicio no explica bien porque poner el parámetro IC a 10. La explicación completa está en: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/tasks/t_obtain_data_from_sensor.htm>

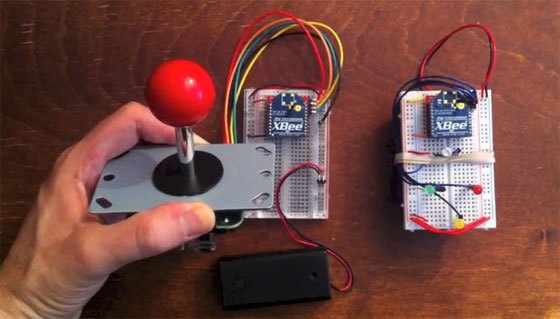
**NOTA**IC = 0x10 = B10000

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DIO12 | DIO11 | DIO10 | DIO9 | DIO8 | DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

**Ejercicio**: cómo obtener datos de un sensor conectado a un módulo XBee: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/tasks/t_obtain_data_from_sensor.htm>

**Ejemplo Pin Pairing para manejar un joystick**

Para manejar un joystick y controlar remotamente un elemento inalámbricamente podemos usar el pin pairing de forma sencilla.



La configuración de los módulos XBee usando direcciones de 16 bits para simplificar quedaría:

|  |  |  |  |
| --- | --- | --- | --- |
| Param | Joystick | Follower | Effect |
| CH | C | C | Define la frecuencia a usar. Debe ser la misma en ambos módulos. |
| ID | 2015 | 2015 | Define la red a la que se conecta la radio. Debe ser la misma en ambos módulos. |
| DH | 0 | — | DH y DL forma la dirección de destino. Es donde las notificaciones son mandadas cuando cambia el valor del pin. Se configura como dirección corta de 16 bits. |
| DL | 5678 | — |
| MY | 1234 | 5678 | Define la dirección corta de 16 bits. |
| D1 | DI [3] | DO Low [4] | Configura el pin DIO1/AD1 en el joystick como entrada y en el follower como salida. Lo que ocurre en el joystick como entrada se transmite al follower como salida. |
| D2 | DI [3] | DO Low [4] | Configura el pin DIO2/AD2 en el joystick como entrada y en el follower como salida. |
| D3 | DI [3] | DO Low [4] | Configura el pin DIO3/AD3 en el joystick como entrada y en el follower como salida. |
| D4 | DI [3] | DO Low [4] | Configura el pin DIO4/AD4 en el joystick como entrada y en el follower como salida. |
| IC | 1E | — | Configura en el joystick que pines monitorizar para mandar la señal al detectar un cambio. 00011110 (binario) = 1E (hexadecimal) |
| IA | — | 1234 | Defines the address of the transmitting module (leader) to pair for I/O. |

Los pines I/O de Xbee son: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#reference/r_xbee_s3b_io_pins.htm>

Con esta simple configuración ya puedo manejar remotamente con un joystick unas luces o incluso un coche con las conexiones adecuadas.

**Modelos de Comunicación**

Existen dos tipos de comunicación y en ambos casos es bidireccional:

* Punto a punto
* Punto a multipunto

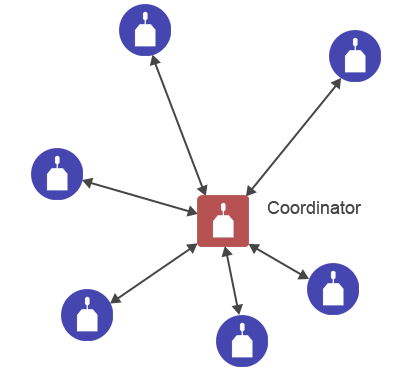
**Punto a punto**

Para que se establezca la comunicación los módulos deben estar en el mismo canal (CH) y en la misma Network ID (ID), además para iniciar la comunicación es necesario saber la dirección MAC de 64-bit del destinatario. La forma en que se comunican dos módulos punto a punto depende si están en modo API o transparente.

Ejemplo de comunicación punto a punto en modo API: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_ex_chat.htm>

**Punto a multipunto**

En este modelo un módulo puede comunicarse con un módulo o múltiples módulos que estén en la misma red. Esta comunicación implica un nodo central coordinador con varios nodos remotos (end devices) conectándose al nodo central.



En el protocolo 802.15.4 los módulos XBee tienen dos roles:

* **Coordinator**, es el nodo central de la red. Inicia la red y permite a otros dispositivos conectarse, puede seleccionar la frecuencia del canal y hace la sincronización de la red. Para configurar un nodo como coordinador hay que cambiar el parámetro CE (Coordinator Enable) a 1.
* **End device**, es un nodo remoto de la red. Puede comunicarse con el coordinador y con otros end devices de la red. Se puede poner en modo sleep.

Las redes mesh solo son soportadas en protocolo ZigBee o Digimesh.

Es posible hacer broadcast, es decir, mandar el mismo datos a todos los nodos posible de la red. Para mandar un broadcast, la dircción de destino debe ser 000000000000FFFF.

En el coordinador, cambiado el parámetro A2 es posible asociar end devices sin que conozcan el canal (CH) y el ID de red (ID) siendo inicializados por el coordinador.

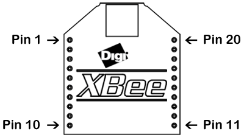
Más información: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#concepts/c_comm_models.htm>

**I/O en módulos XBee**

Todos los módulos XBee tienen un conjunto de pines que pueden usarse para conectar sensores o actuadores y configurarlos. A diferencia de un Arduino, estos módulos no pueden ejecutar un programa para interaccionar, sino que es necesario un elemento externo como un Arduino para programarlos o cambiar su comportamiento.

Con estos pines se puede recoger el dato de un sensor y mandarlo a otro módulo o encender una luz al estar conectado a un relé cuando recibe la señal sobre un pin configurado como salida.

Pines I/O para módulos XBee 802.15.4

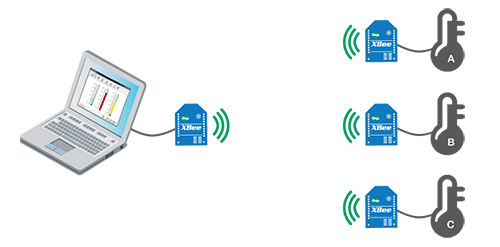


|  |  |  |
| --- | --- | --- |
| Pin name | Physical pin # | Parameter |
| DIO0, AD0 | 20 | D0 |
| DIO1, AD1 | 19 | D1 |
| DIO2, AD2 | 18 | D2 |
| DIO3, AD3 | 17 | D3 |
| DIO4, AD4 | 11 | D4 |
| DIO5, AD5 | 15 | D5 |
| DIO6 | 16 | D6 |
| DIO7 | 12 | D7 |
| DI8 | 9 | D8 |
| PWM0 | 6 | P0 |
| PWM1 | 7 | P1 |

(D = digital; I = input; O = output; AD = analog input; PWM = pulse-width modulation)

**Leer Sensores**

Los módulos XBee se pueden usar leer datos de una red de sensores. Para recibir los datos es necesario configurar los módulos remotos para escuchar en el pin donde se conecta el sensor y mandar los datos al módulo principal que estará conectado a una MCU o un ordenador.



En función del tipo de sensor o actuador que conectemos al pin del módulo deberemos configurarlo adecuadamente.

Hay dos formas de obtener datos de un sensor:

* Haciendo una consulta para que lea todos los pines habilitados como entradas. Queried sampling (IS)
* Transmitir automáticamente los datos del sensor de forma periódica o cuando un pin digital cambia. El parámetro IR configura la frecuencia con que se mandan los datos leídos de los sensores. El parámetro IC configura que pines monitorizar para detección de cambio, cuando los pines monitorizados detectan un cambio inmediatamente se leen los sensores y se mandan los datos.

Para seleccionar qué pines monitorizar, se asigna un valor binario a IC basado en este patrón:

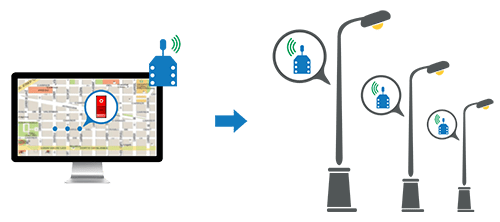
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DIO12 | DIO11 | DIO10 | DIO9 | DIO8 | DIO7 | DIO6 | DIO5 | DIO4 | DIO3 | DIO2 | DIO1 | DIO0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Ejercicios en:

* <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_receive_digital.htm>
* <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_receive_analog.htm>

**Controlar Dispositivos**

Un módulo XBee es capaz de recibir comandos y poner una salida digital o analógica a un valor al ocurrir un determinado evento, sin el uso de un microcontrolador externo.



Los módulos XBee 802.15.4 tienen 8 salidas digitales (D0 a D7) y pueden configurarse con una resistencia de pull up o pull down. Los módulos también tienen salida analógica mediante PWM, en el caso de los módulos XBee 802.15.4 tienen 2 salidas analógicas (P0 y P1)

Recordar que los módulos XBee van a 3.3 V.

Para mandar una actuación a un módulo hay que mandar un comando AT configurado de la manera correcta para actuar sobre un pin, una vez se ha configurado anteriormente como salida.

Ejercicios en:

* <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_control_devices.htm>
* <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#tasks/t_ex_send_analog_actuations.htm>

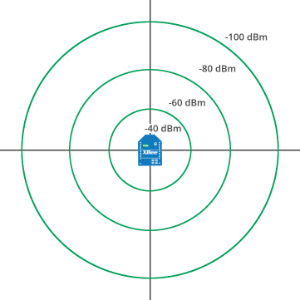
Más información: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_inputs_and_outputs.htm>

**Señal y Rangos de Frecuencia**

La distancia de alcance de la señal de los módulos XBee está afectada por diversos factores:

* Algunos materiales pueden reflejar las ondas de radio provocando interferencias. En particular materiales metálicos.
* Las ondas de radio pueden ser absorbidas por objetos en su camino
* Las antenas pueden ajustarse para incrementar la distancia.
* La línea de visión puede ayudar a incrementar la fiabilidad de la señal.

RSSI (Received Signal Strength Indicator) es el indicador de la cantidad de energía presente en una señal de radio. Midiendo la fuerza de la señal en la antena receptora es una forma de determinar la calidad del enlace de comunicación. Su valor se mide en [dBm](https://es.wikipedia.org/wiki/DBm)cuanto mayor sea el valor negativo la señal es más débil. Por lo tanto -50dBm es mejor que -60 dBm.



El pin 6 de los módulos XBee puede ser configurado como salida PWM con el valor de RSSI, para ello configurar P0 como RSSI [1]. También en el parámetro DB se guarda el valor de RSSI del último paquete recibido expresado en valor decimal.

Además del valor de RSSI, es conveniente medir la fiabilidad del enlace en porcentaje de paquetes perdidos, puesto que en entornos con interferencias el valor de RSSI puede ser alto pero el enlace será malo.

**Antenas**

A la de lograr un determinado alcance de un módulo, es importante la antena, para ello los módulos RF XBee disponen de distintos tipos de antenas:

* PCB antenna: está formada en la propia PCB con trazo conductor.



* Cable de antena integrada: es un pequeño cable de 80mm colocado perpendicularmente a la PCB y soldado directamente a la PCB



* Whip antenna: una antena sólida pero flexible que sobresale 25mm del módulo. Esta antena aumenta el rango sobre la antena del chip pero en el exterior.



* U.FL antenna: es un pequeño conector para una antena externa. Es una buena opción si el módulo está en una caja y queremos poner la antena en el exterior.



* RP-SMA antenna: es un conector mayor para una antena externa.

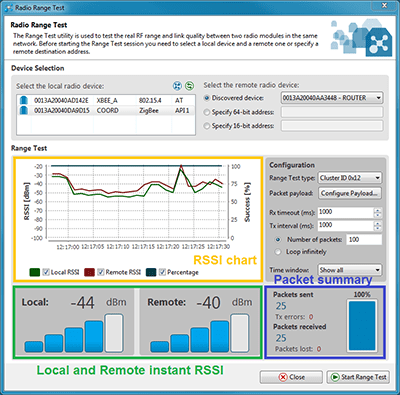


**Prueba de Rango**

Una prueba de rango nos sirve para medir la calidad del enlace entre dos módulos XBee. XCTU permite hacer una prueba de rango que supone enviar paquetes desde un módulo local a uno remoto esperando el echo, contabilizando el número de paquetes enviados y recibidos y midiendo el RSSI.



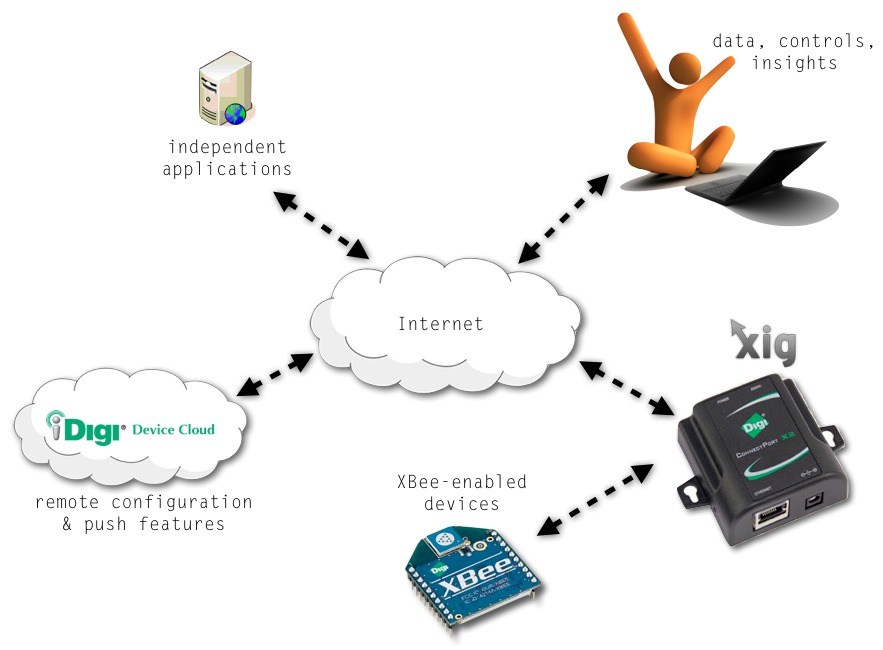
Como hacer una prueba de rango: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#tasks/t_try_range_test.htm>



Más información: <http://www.digi.com/resources/documentation/Digidocs/90001456-13/Default.htm#containers/cont_signal_strength.htm>

**XBee Gateways**

Hemos visto cómo hacer una red inalámbrica con los módulos XBee y como hay un intercambio de información entre los módulo. Ahora bien, si esos datos los queremos sacar fuera de esa red inalámbrica necesitaremos un gateway que nos saque esos datos a una red ethernet (Internet/Intranet) o a un bus de campo standard donde poder almacenar y mostrar los datos y estados de la red Xbee.



**XBee to IP Gateway**

Este gateway habilita la conexión remota, configuración y gestión de las redes Xbee con redes IP. Todos los datos Xbee enviados al gateway están automáticamente disponibles para aplicaciones on line via Device Cloud. Este gateway puede ejecutar aplicaciones python que comuniquen y gestionen la red XBee.

Más información: <http://www.digi.com/products/xbee-rf-solutions/gateways/xbee-gateway>

**XBee RF Modems**

Los módems XBee RF se usan para comunicar con sistemas que usen RS-232, RS-485 o interfaz USB.

Más información: <http://www.digi.com/products/xbee-rf-solutions/modems/xbee-pro-900hp-rf-modems>

**Interacción Software – Xbee**

Todo el código está disponible en:

* <https://github.com/digidotcom/XBeeArduinoCodingPlatform>
* Descarga de la última versión <https://github.com/digidotcom/XBeeArduinoCodingPlatform/releases>

Los proyectos propuestos en este tutorial para interacción de XBee con software, en este caso con processing son:

* Follow the Leader: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Follow+the+Leader>
* Car Race Game: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Car+Race+Game>
* Mechanical Sketching Toy: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Mechanical+Sketching+Toy>
* Scalable Car Race Game: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Scalable+Car+Race+Game>
* XBee Game Controller: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/XBee+Game+Controller>

Librería XBee Java: <https://github.com/digidotcom/XBeeJavaLibrary>

Instalar la librería XBee java <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Installing+the+xbjlib+Processing+library>para que una aplicación de processing pueda comunicarse con un modulo XBee

**XBee y Arduino**

Instalar librería XBee Arduino <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Installing+the+xbee-arduino+library>

Librería: <https://github.com/andrewrapp/xbee-arduino>

Trabajando con Arduino: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Working+with+Arduino>

Primer proyecto <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Stop-It%21+LED+Game>

Todo el código está disponible en:

* <https://github.com/digidotcom/XBeeArduinoCodingPlatform>
* Descarga de la última versión <https://github.com/digidotcom/XBeeArduinoCodingPlatform/releases>

Los proyectos propuestos en este tutorial para uso de XBee con Arduino son:

* LED Game: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Stop-It%21+LED+Game>
* Piedra, Papel y Tijera: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Rock+Paper+Scissors>
* Quiz Show: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Quiz+Show>
* Bomb Defuser: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Bomb+Defuser>
* Treasure Hut: <http://docs.digi.com/display/XBeeArduinoCodingPlatform/Treasure+Hunt>

**HW para Arduino**

Los dos kits lleva los módulos Serie 1:

* Data sheet del kit: <http://media.digikey.com/pdf/Data%20Sheets/Digi%20International%20PDFs/XKB2-AT-WWC.pdf>
* Datasheet del módulo XBee S1 802.15.4 <http://www.digi.com/pdf/ds_xbeemultipointmodules.pdf>
* User Guide del módulo XBee S1 802.15.4: <http://ftp1.digi.com/support/documentation/90000982.pdf>

XBee Shields:

* <https://www.arduino.cc/en/Guide/ArduinoWirelessShield>
* <https://www.arduino.cc/en/Guide/ArduinoWirelessShieldS2>
* <https://learn.sparkfun.com/tutorials/xbee-shield-hookup-guide>
* <http://www.seeedstudio.com/wiki/XBee_Shield_V2.0>

El Xbee Explorer:

* <https://www.sparkfun.com/products/11812>

Poner en marcha un Xbee shield:

* <https://www.arduino.cc/en/Guide/ArduinoWirelessShieldS2>
* <https://www.sparkfun.com/products/12847>
* <https://learn.sparkfun.com/tutorials/xbee-shield-hookup-guide>

XBee en la industria: <http://www.digi.com/industries>

* Sistema de gestión de alumbramiento público: <http://www.digi.com/customersuccesses/xbee-enables-street-light-management-system>
* Monitorización de depósitos: <http://www.digi.com/customersuccesses/wireless-tank-monitoring-with-1844myfuels>

Referencia: <https://aprendiendoarduino.wordpress.com/2016/11/16/zigbeexbee/>

Armando Mtz. ITNL