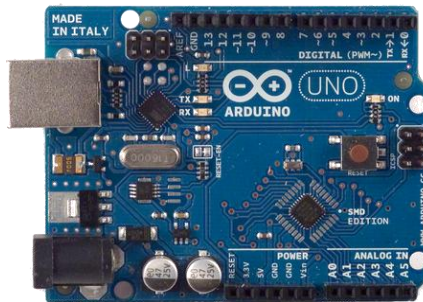


# Arduino + XBee

**Implementación de Sistemas de Trasmisión de Datos y Sensores en Redes Inalámbricas con XBee integrado en la “Plataforma Open Hardware” Arduino**



+



Ver. 1.0

José Manuel Ruiz Gutiérrez

**Serie:** Arduino Comunicación



## Índice

1. Objetivo de este trabajo.
2. Una introducción a XBee
3. Ejemplos de Montajes con los módulos XBee conectados autónomamente sin necesidad de Arduino.
  - 3.2 Comunicación entre dos Ordenadores mediante dos módulos XBee 802.15.4 (serie 1)
  - 3.3 Comunicación entre dos Ordenadores mediante dos módulos XBee 802.15.4 (serie 1)
  - 3.4 Configuración de un botón como Entrada Digital
  - 3.5 Salida Digital con un diodo LED
4. Trabajo de los módulos XBee con Arduino
  - 4.1. Programación de los Módulos XBee
  - 4.2 Test de una comunicación PC ->XBee
  - 4.3 Timbre inalámbrico - XBee Serie 1
5. Documentación Software y Bibliografía



Este trabajo está bajo licencia [Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by-nc-nd/3.0/)  
14 Noviembre de 2012 Versión de Documento: Versión. 1  
Blog de referencia: <http://josemanuelruizgutierrez.blogspot.com>  
[j.m.r.gutierrez@gmail.com](mailto:j.m.r.gutierrez@gmail.com)

# 1. Objetivo de este trabajo.

Con este trabajo se pretende facilitar un “primer contacto” de la tecnología XBee para la comunicación vía radio integrada en la Plataforma Open Hardware Arduino.

No es necesario explicar las grandes ventajas que ofrece el protocolo XBee y su tecnología en el ámbito de las redes de sensores y el procesamiento de datos.

La “comunidad” de usuarios e investigadores de Arduino ha recogido esta tecnología y ha diseñado Shields que permiten unir varios Arduinos en una red inalámbrica.

Pretendo con este trabajo facilitar una guía de iniciación a quienes deseen experimentar con esta tecnología.

El manual que sigue aborda una explicación básica del sistema, los métodos y herramientas para la programación de las unidades y unos cuantos ejemplos de aplicaciones, tanto con módulos aislados XBee como con módulos integrados en el Shield para Arduino.

Como con el resto de mis trabajos deseo que este documento sea de ayuda a quienes lo utilicen. Agradezco a las personas e instituciones que me precedieron en la elaboración de documentación sus aportaciones y animo a quien realice algún trabajo en este campo a que lo comparta con todos los que formamos esta enorme comunidad de usuarios de Arduino.

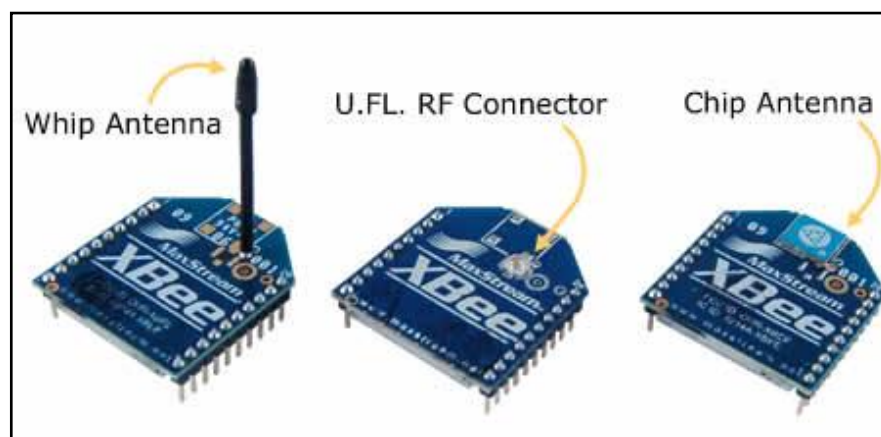
## 2. Una introducción a XBee

Los módulos XBee son dispositivos que integran un transmisor - receptor de ZigBee y un procesador en un mismo módulo, lo que le permite a los usuarios desarrollar aplicaciones de manera rápida y sencilla.

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estandar de comunicaciones para redes inalámbricas IEEE 802.15.4. Creado por Zigbee Alliance, una organización, teóricamente sin ánimo de lucro, de más de 200 grandes empresas (destacan Mitsubishi, Honeywell, Philips, Motorola, Invensys, ...) , muchas de ellas fabricantes de semiconductores.

Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticos.

Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth no utiliza FHSS (Frequency hopping), 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.



Los módulos Xbee son económicos, potentes y fáciles de utilizar. Algunas de sus principales características son:

- Buen Alcance: hasta 100 metros en línea vista para los módulos Xbee y hasta 1 milla (1.6 Km) para los módulos Xbee Pro.
- 9 pines entradas/salidas con entradas analógicas y digitales.

- Bajo consumo <50mA cuando están en funcionamiento y <10uA cuando están en modo sleep.
- Interfaz serial.
- 65,000 direcciones para cada uno de los 16 canales disponibles. Se pueden tener muchos de estos dispositivos en una misma red.
- Fáciles de integrar.

Existen 2 series de estos módulos. La serie 1 y la serie 2 o también conocida como 2.5. Los módulos de la Serie 1 y la Serie 2 tienen el mismo pin-out, sin embargo, NO son compatibles entre sí ya que utilizan distintos chipset y trabajan con protocolos diferentes.

La serie 1 está basada en el chipset Freescale y está pensado para ser utilizado en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están basados en el chipset de Ember y están diseñados para ser utilizados en aplicaciones que requieren repetidores o una red mesh. Ambos módulos pueden ser utilizados en los modos AT y API.

### **Características técnicas:**

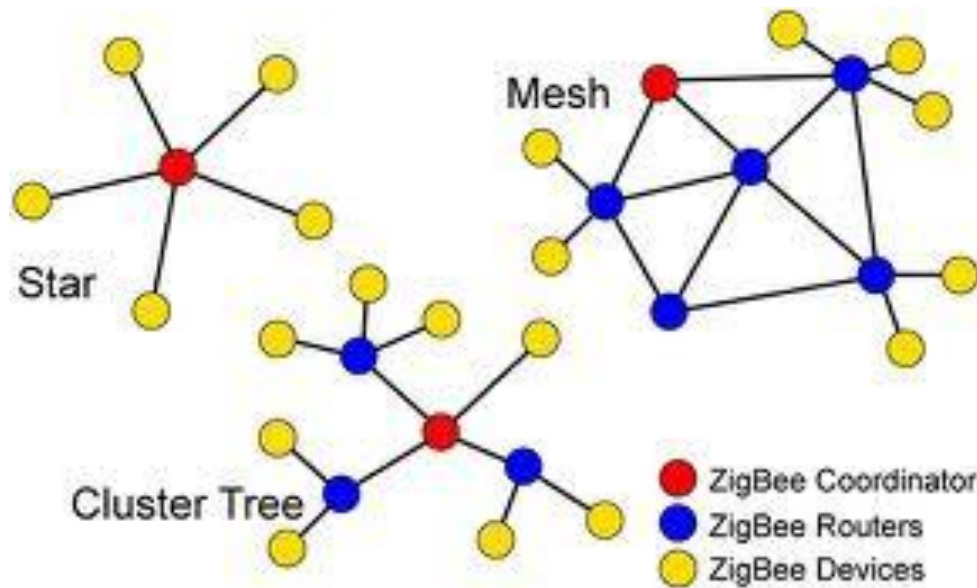
- ISM 2.4 GHz operating frequency
- 1 mW (0 dBm) power output (up to 100m range)
- RPSMA Connector, U.FL. Connector, Chip or Whip antenna options
- Industrial temperature rating (-40° C to 85° C)
- Approved for use in the United States, Canada, Australia and Europe
- Advanced networking & low-power modes supported

Para más información visiten la pagina <http://www.xbee.cl/> donde encontraran más información.

### **Arquitectura Básica de una Red XBee.**

Una red Zigbee 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:



- Punto a punto
- Estrella
- Malla
- Árbol
- Mixtas

Una vez decidida la que red con la que trabajar se deben escoger los módulos XBee. Tendremos dos posibilidades: usar todo en Serie 1 o todo en Serie 2. Recordemos que estos módulos también los encuentras en la versión PRO que te dan una mayor distancia:

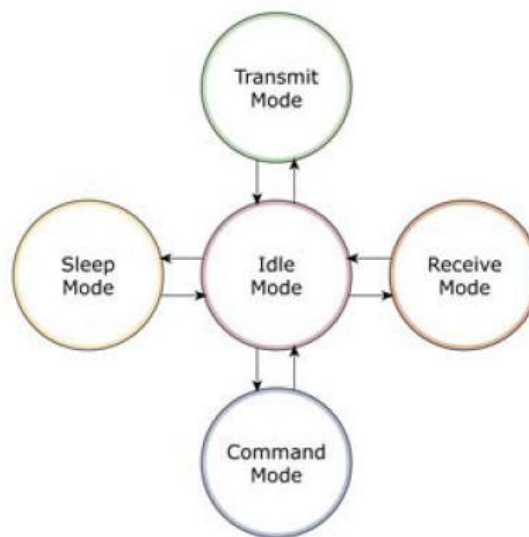
**El Coordinador:** Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones (como hablábamos antes) y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el Coordinador puede formar una red, permitiendo 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.

### Modos de Operación de un Elemento XBee

Los módulos pueden ser configurados para trabajar en distintos modos, tantos como 4.



### Modos RECIBIR/TRANSMITIR.

Se encuentra en estos modos cuando al módulo le llega algún paquete RF a través de la antena(modos Receive) o cuando se manda información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmit).

La información transmitida puede ser Directa o Indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita.

Además es posible enviar información por dos modos. Unicast y Broadcast. Por el primero, la comunicación es desde un punto a otro, y es el único modo que permite respuesta de quien recibe el paquete RF, es decir, quien recibe debe enviar un **ACK** (paquete llamado así, y que indica que recibió el paquete, el usuario no puede verlo, es interno de los módulos) a la dirección de origen. Quien envió el paquete, espera recibir un **ACK**, en caso de que no le llegue, reenviará el paquete hasta 3 veces o hasta que reciba el **ACK**. En el modo Broadcast la comunicación es entre un nodo y a todos los nodos de la red. En este modo, no hay confirmación por **ACK**.

### Modo de Bajo Consumo (Sleep Mode).

El modo de sueño hace posible que el módulo RF entre en un modo de bajo consumo de energía cuando no se encuentra en uso.

La programación en este modo se puede mirar en el manual técnico y la dejamos por escaparse al objetivo de este manual.

### Modo de Comando.

Este modo permite ingresar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otras cosas. Para poder ingresar los comandos AT es

necesario utilizar el Hyperterminal de Windows, el programa X-CTU<sup>3</sup> o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma.

### **Idle**

Cuando el módulo no se está en ninguno de los otros modos, se encuentra en éste. Es decir, si no está ni transmitiendo ni recibiendo, ni ahorrando energía ni en el modo de comandos, entonces se dice que se encuentra en un estado al que se le llama IDLE.



### 3. Ejemplos de Montajes con los módulos XBee conectados autónomamente sin necesidad de Arduino.

Presentamos a continuación una serie de aplicaciones prácticas orientadas a la utilización de módulos XBee que se comunican sin necesidad de utilizar Arduino ni ninguna otra tarjeta.

Los módulos poseen una serie de entradas y salidas analógicas y digitales que pueden ser utilizadas directamente para hacer control. Esta es una posibilidad muy interesante, aplicable a pequeños automatismos gobernados por vía radio.

Los ejemplos han sido sacados en su totalidad de la pagina Web <http://examples.digi.com/> Digi XBee® Examples & Guides y sus autores son los relacionados en cada una de las aplicaciones.

#### 3.1 Comunicación entre dos Ordenadores mediante dos módulos XBee 802.15.4 (serie 1)

(Basado en el el Ejemplo realizado por *Rob Faludi with Matt Richardson*)

En esta aplicación vamos a realizar la conexión vía radio entre dos módulos conectados a dos PC. Se trata de enviar y recibir textos que se enviaran desde uno u otro PC a modo de un chat (bidireccional). Para ello utilizaremos los siguientes dispositivos:

- 2 XBee 802.15.4 Radios (Series 1)
- 2 [XBee Explorer USB Boards from Sparkfun Electronics](#) (part number WRL-08687)
- 2 Ordenadores corriendo en cada uno de ellos un programa de Comunicación con el puerto serie.



Modulo XBee Serie 1

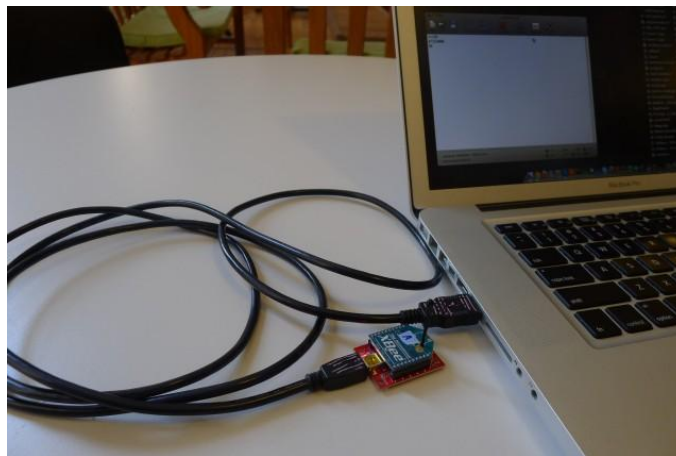


Modulo XBee Explorer USB

En la figura vemos un esquema del principio de funcionamiento de la aplicación.



Procederemos a programar cada uno de los módulos XBee haciendo uso de la tarjeta XBee Explorer USB (es sencillamente una tarjeta que permite comunicar al módulo XBee con el ordenador a través del puerto serie).

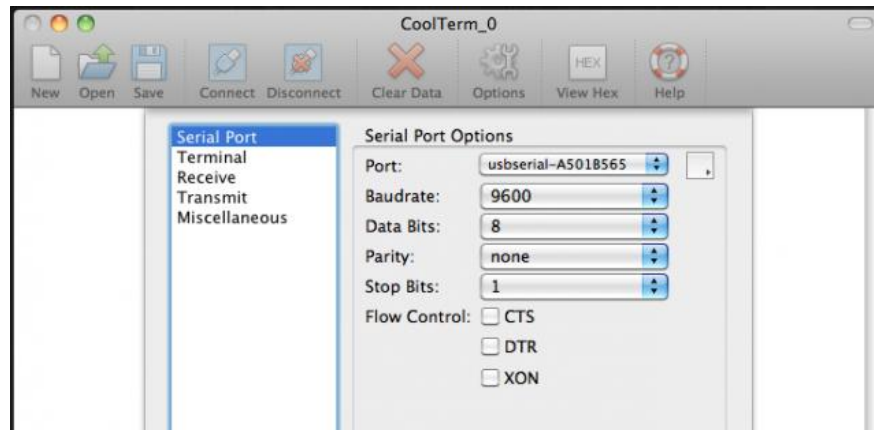


Marcamos el primer modulo XBee con el que vamos a trabajar con la letra “A.”

Insertamos el módulo XBee, sobre el módulo XBee Explorer, y conectmos al Ordenador mediante el cable USB.

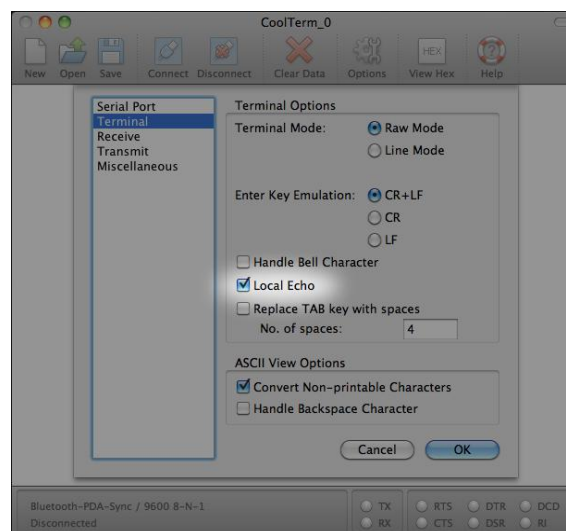
Téngase en cuenta que deberms instalar los drivers necesarios para que el ordenador reconozca el módulo XBee Explores en nuestro caso los drivers FTDI [Pulsar aqui para descargarlos de Internet](#).

Se recomienda el software CoolTerm para realizar la programación del modulo. Este se encuentra en [CoolTerm by Roger Meier](#). Este software es recomendable porque trabaja con todos los sistemas operativos Windows, Mac, and Linux y además es libre



Procedemos a configurar el software CoolTerm.

- Abrimos el software CoolTerm y seleccionamos “Options”
- Seleccionamos el puerto serie por el que se ha conectado el módulo XBee. Si no estamos seguros de qué puerto es, miramos la lista de puertos disponibles y luego desconectamos el Explorador de XBee de su equipo. Hacemos clic en "Volver a explorar puertos serie" y comprobamos qué puerto serial se incluye (ese será el nuestro).
- Debemos asegurarnos de que la velocidad de transmisión se establece en 9600. El número de bits de los datos es de 8. Se ajusta la paridad a "ninguno". Se ajusta el número de bits de parada en 1.
- En la lista de la parte izquierda de la ventana de opciones, hacemos clic en "Terminal". Asegúrese de que esta opción "Eco local" está activada. Esto le permitirá ver lo que está escribiendo en el terminal.



- Hacemos Click en OK para guardar la configuración y cerramos la ventana de Opciones.
- Hacemos Click en “Connect” en la barra de menus.
- Debemos ver el estado “Connected” en la barra de estado.
- Escribimos “+++” para entrar en el modo Comando. El sistema nos devuelve “OK.”

**Nota:** Por defecto el modulo XBee se desconecta si deja de recibir comandos en 10 segundos.

### Comandos para configuración del módulo

Función	Comando	Parámetro
PAN ID	ATID	3001 (cualquier dirección desde 0 a FFFE)
MY Address	ATMY	1
Dirección de destino (parte alta)	ATDH	0 (indica una dirección de 16-bits)
Dirección de destino (parte baja)	ATDL	2

Mirar en el manual de XBee para ampliar la información de los comandos [página de referencia para ver la lista completa de comandos AT de configuración de módulo XBee](#).

Escribir cada comando seguido de sus parámetros y pulsar ENTER.

**PAN ID:** PAN es el número de la Red de Área Personal. Se trata de un identificador único para la red. Solo los XBees asignados a un PAN ID pueden comunicarse entre sí. Esto permite configurar redes separadas en el mismo lugar.

**MY Address:** Esta es la dirección de origen de un XBee, es una dirección única para cada unidad en particular.

**Dirección de destino (parte alta):** Representa la primera mitad de la dirección que queremos habilitar. En los módulos XBee puede tener una dirección de 64 bits, por lo que esta es la parte más alta de 32-bit de ese número de direcciones. Puesto que no necesitamos tantas direcciones, vamos a ponerlo a 0 y sólo usar la parte alta.

**Dirección de destino (parte baja):** Esta es la dirección que usaremos para localizar el XBee. Asegúrese de que coincida con el ajuste de la ATMY XBee con el que quiere hablar.

Los ajustes no se guardan hasta que no escriba el comando ATWR para guardar la configuración.

Así es como se verá la sesión en el terminal, a partir de la "+ + +" entraremos en modo comando.+++

```

OK
ATID 3001
OK
ATMY 1
OK
ATDH 0
OK
ATDL 2
OK
ATID
3001
ATMY

```

```
1
ATDH
0
ATDL
2
ATWR
OK
```

**Nota:** Debemos obtener una respuesta OK después de emitir cada comando para establecer los parámetros, y otra respuesta OK al escribir los cambios en el firmware. Si usted no recibe una respuesta OK, lo más probable es que tomó más de diez segundos para el comando y el módulo dejó de estar en modo comando. El otro error común es no escribir el comando ATWR para guardar los cambios, si no lo hace su configuración se pierde al desconectar el módulo.

Haga clic en desconectar CoolTerm y quite el módulo XBee

### Configuración de la segunda unidad de Radio

- Marcar la segunda unidad XBee con una “B.”
- Conectar el módulo XBee sobre el módulo XBee Explorer y hacer clic en “Connect” en el programa CoolTerm.
- A continuación seguir los mismos pasos que anteriormente hemos realizado con el módulo “A”. En este caso hacer ATMY 2 y el valor de ATDL 1. Ver la siguiente tabla:

Función	Comando	Parámetro
PAN ID	ATID	3001 (cualquier dirección desde 0 a FFFE)
MY Address	ATMY	2
Dirección de destino (parte alta)	ATDH	0 (indica una dirección de 16-bits)
Dirección de destino (parte baja)	ATDL	1

- Recordar que hay que escribir ATWR y presionar enter para salvar la configuración.

A continuación se muestran los códigos tal como se deben escribir y aparece en el terminal:

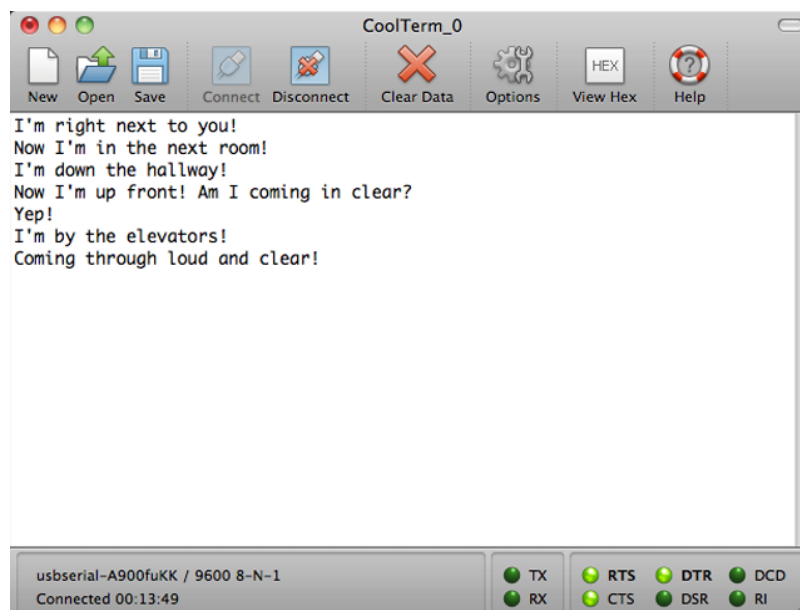
```
+++
OK
ATID 3001
OK
ATMY 2
OK
ATDH 0
OK
ATDL 1
```

```
OK
ATID
3001
ATMY
2
ATDH
0
ATDL
1
ATWR
OK
```

## Funcionamiento

Para probar que funcionan la aplicación procederemos de la siguiente manera:

- Conectar XBee “A” a la tarjeta XBee Explorer y conectar a distintos PC en la misma habitación. Cargar el programa CoolTerm y conectar este siguiendo los pasos que ya comentamos anteriormente.
- Si usted todavía está en el modo de comandos en cada módulo, puede escribir ATCN para salir del modo de comando sin tener que cuidarse de los 10 segundos de tiempo que tiene para escribir en el terminal sin que se desconecte el módulo.
- Estamos en condiciones de realiza las pruebas de comunicación de mensajes entre ambos módulos de radio XBee. Si todo está configurado correctamente, el texto que se escribe en el programa del terminal serie en el primer equipo será transmitida al segundo ordenador y aparecerá en su pantalla de la terminal de serie también.



## Más funciones

- Si separamos los módulos se perderá la comunicación.

- El protocolo 802.15.4 no sólo permite la comunicación punto a punto, como hemos tratado en esta guía, también permite comunicarse punto-a-multipunto. Si tiene más de dos XBees y los puso a todos a la misma PAN ID (con ATID) y luego ponga la dirección de destino bajo de la emisora a FFFF (con ATDL FFFF). Ahora, al escribir en el terminal del XBee emisor, debemos ver el texto en todas las otras estaciones terminales.

## Más información

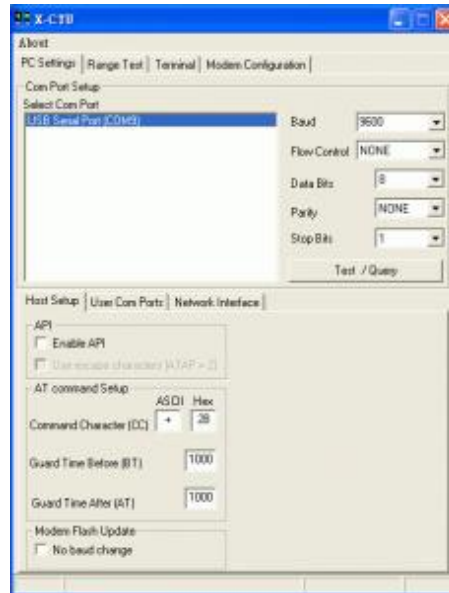
### Solución de problemas.

- Si todo funciona a la perfección la primera vez, eso es genial! Sin embargo, la experiencia demuestra que a veces toma varios intentos para hacerlo bien todo. Usted acaba de establecer un sistema bastante complejo. No te desespere si tu chat no funciona de inmediato. En casi todos los casos, no hay nada mal en cualquiera de su hardware o incluso en la mayor parte de su configuración. Sólo se necesita un parámetro incorrecto a lanzar una llave en las obras. Aprender a encontrar esa llave y arreglarlo es una habilidad esencial, así que aquí están algunos consejos acerca de lo que debemos probar si las cosas no funcionan al principio.
- Comience con las cosas simples. Asegúrese de que sus radios están correctamente asentados en las tarjetas adaptadoras y que todos los cables USB están conectados en la forma en que debe ser. Compruebe que cada radio está respondiendo correctamente en la ventana de terminal al tratar de usar + + + para ponerlo en el modo de comando. Si usted no recibe una espalda bien, compruebe la selección de puerto, velocidad de transmisión, y las otras opciones hasta que encuentre la razón de la radio no se comunica correctamente.
- Si ambos equipos están respondiendo, utilice los comandos AT para comprobar la configuración. Los problemas más comunes son: no utilizar el mismo ID PAN en ambos radios, no se establece la dirección de destino en cada radio con el número de mi dirección de la otra radio, y no guardar la configuración adecuada. Si los ajustes de todos parece ser correcta, asegúrese de que usted tiene el firmware coordinador en una radio y el firmware del router en la otra radio.
- Puede utilizar el comando ATVR para mostrar el firmware que está en uso. Si usted ve la versión antigua, vuelva a X-CTU y cargue el firmware apropiado.
- A veces las radios estarán perfectamente configurado y conectado, pero uno de sus radios accidentalmente estar hablando con otra persona. Esto sucede a menudo en situaciones de aula, donde muchas personas están utilizando el mismo ID PAN en la misma habitación. Trate de usar diferentes identificadores del PAN para cada par.
- A veces, estableciendo dos radios de nuevo a los valores de fábrica y volver a configurar los enjuagará un ambiente malo que había sobrado de una instalación

anterior, o un no reconocido error tipográfico. El comando ATRE acabará con configuración personalizada de su radio y el firmware quedará fijado limpiamente a los valores predeterminados de fábrica. Siga con el comando ATWR a escribir esos valores por defecto para el firmware, y luego volver a los pasos de configuración y trate de poner en la configuración de nuevo.



### 3.2 Comunicación entre dos Ordenadores mediante dos módulos XBee 802.15.4 (serie 1)



Cuando se trabaja con XBee, puede que tenga que actualizar o cambiar el firmware de vez en cuando. Por ejemplo, si desea cambiar un módulo ZigBee de router a coordinador o cambiar entre los modos de Programas y AT, tendrás que cargar el firmware apropiado para la radio.

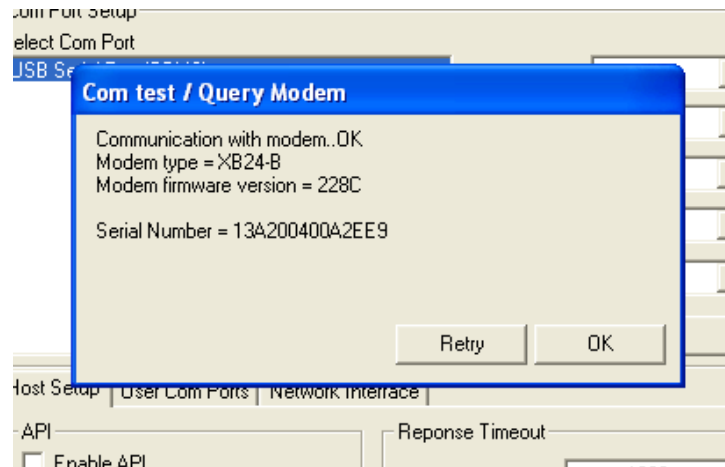
**Para ello, vamos a utilizar X-CTU.**

X-CTU es una aplicación basada en Windows proporciona Digi. Este programa está diseñado para interactuar con los archivos de firmware que se encuentran en los productos de Digi RF y proporcionar un fácil de usar interfaz gráfica de usuario para ellos. X-CTU está diseñado para funcionar con todos los equipos basados en Windows que ejecutan Microsoft Windows 98 SE o superior. Para los que no son usuarios de Windows, puede probar uno de los productos Moltosenso's para actualizar el firmware de su radio. Otra opción es utilizar Wine o una máquina virtual como VirtualBox para ejecutar X-CTU en ordenadores que no tengan Windows.

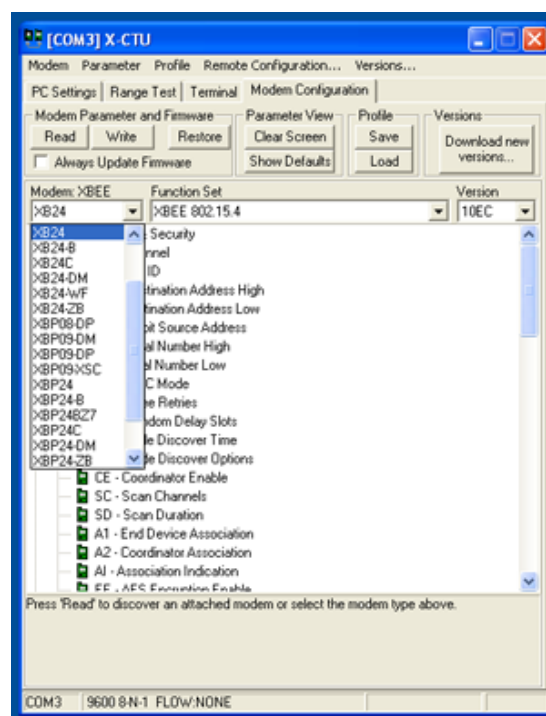
- Descargar e instalar X-CTU de Digi.com. El enlace de descarga se puede encontrar bajo el título "Diagnóstico, Servicios Públicos y MIB".
- Conecte la radio a un explorador XBee o similar y conectar el explorador de su ordenador.

Nota: Para que en su explorador XBee aparezca el modulo conectado como un puerto COM, tendrá que instalar los drivers FTDI para su sistema.

- Abrir X-CTU. Seleccione un puerto COM y haga clic en "Consulta de prueba /" para ver si el X-CTU se puede comunicar con la radio.



Usted debe ver una ventana que dice que la comunicación con el módem estaba bien y alguna información adicional acerca del módulo XBee. Si no es así, pruebe con otro puerto COM y solucionar la conexión serial con el módulo. Puede probar si funciona con el software CoolTerm.



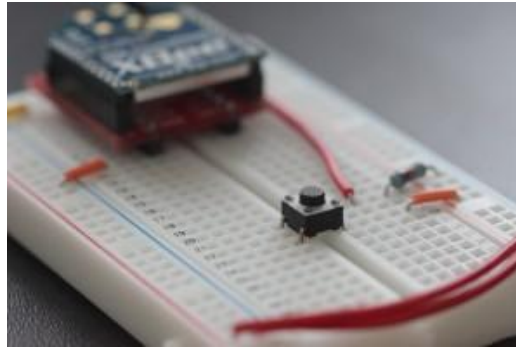
Si la prueba funciona, haga clic en la pestaña "Modem Configuration" en la parte superior y luego, en l lista "Módem:" Seleccione XB24 para 802.15.4 (Serie 1) y los módulos XB24-ZB para ZigBee (Serie 2).

- Seleccione la configuración adecuada en la lista "Function set" .
- Haga clic en el botón "Write" en la parte superior de la ventana para cargar el firmware.

- A medida que se carga el firmware, el estado del proceso se muestra en la parte inferior de la ventana.
- Si desea actualizar cualquier otra configuración, haga clic en "Read" para cargar la configuración actual del módem.
- Actualice las opciones de la lista y haga clic en "Write" para cargar esa configuración al módem

### 3.3 Configuración de un botón como Entrada Digital

(Basado en el trabajo escrito por Matt Richardson en <http://examples.digi.com/> Digi XBee® Examples & Guides)



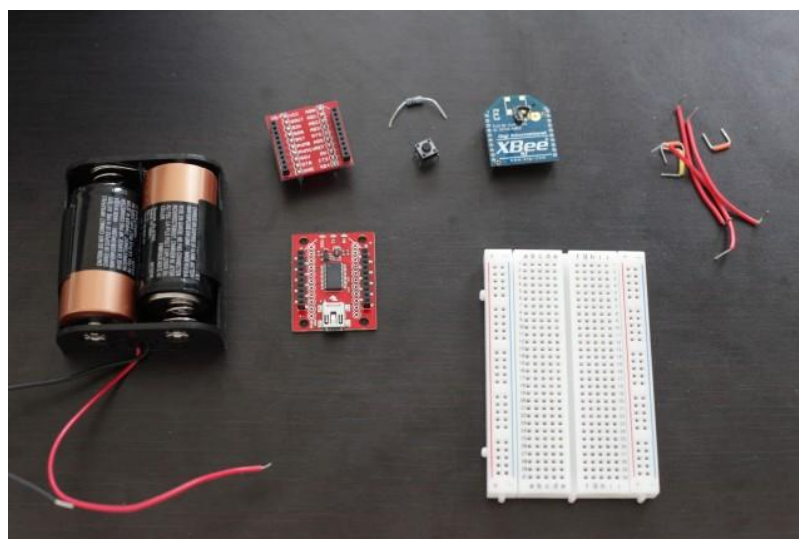
Con esta aplicación se trata de programar un módulo XBee con una de sus entradas como entrada digital a la que conectaremos un botón. Este montaje permite enviar una señal de activación/desactivación a una red de elementos XBee.

#### Introducción.

Como entrada digital, lo más sencillo que su botón básico

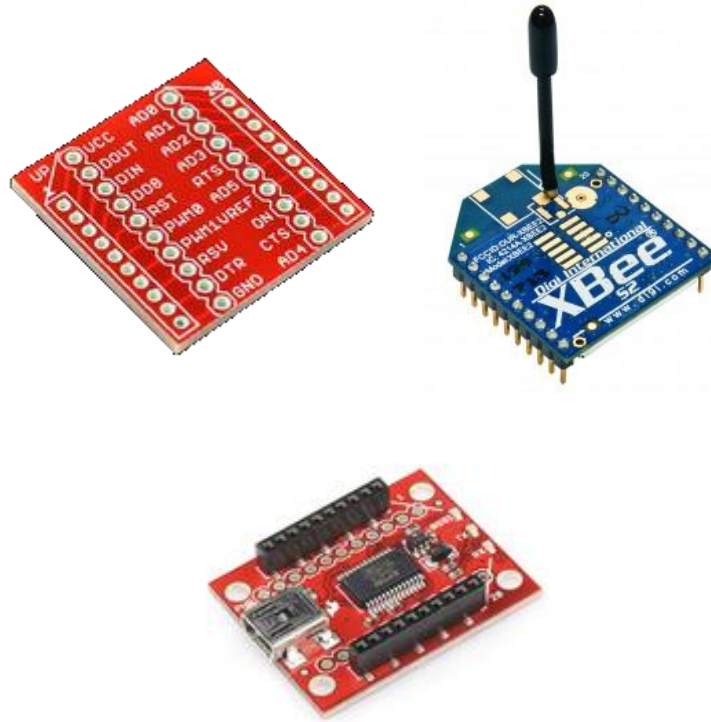
En el módulo XBee se pueden configurar para hasta 8 terminales de entrada digital que pueden controlar directamente los pines de salida en un segundo XBee que actuaría como receptor. En este tutorial, vamos a hacer un sencillo proyecto de electrónica digital inalámbrica.

#### Materiales necesarios



- XBee 802.15.4 Radio
- Pulsador insertable en pcboard
- Resistencia de 10K.

- Breadboard
- Cables de conexión
- [XBee Breakout Board](#) – para insertar en la board.
- [XBee Explorer USB](#) – para programar el elemento XBee.
- Una batería de DC 2.8 – 3.3V – Pueden usarse dos pilas tipo C .



*Módulo XBee Breakout Board*

*Módulo XBee Serie 1*

*Módulo XBee Explorer USB*

### Configuración del módulo XBee



Diagrama de pines del módulo Xbee. Vista Superior.

Aquí vemos el patillaje de l modulo XBee que tenemos que tener en cuenta para realizar el cableado.

En el apartado anterior de este manual hemos descrito como realizar la programación del modulo de radio XBee.

En la siguiente tabla tenemos los valores de programación.

Función	Comando	Parámetro
Reset	ATRE	N/A (resets the radio to its factory settings)
PAN ID	ATID	3001 (cualquier dirección desde 0 hastaFFE )
Mi dirección (MY Address)	ATMY	1
Dirección de destino (Parte alta)	ATDH	0 (indica una dirección de 16-bit)
Dirección de destino (Parte baja)	ATDL	2 (la dirección del receptor)
Configuración de Pin 0 I/O	ATD0	3 (entrada digital)
Velocidad del transmisión	ATIR	14 (20 ms, 14 in hexadecimal = 20 in decimal)
Pull-up resistors	ATPR	0 (desactiva todas las resistencias internas en todos los pins)
Graba la configuración en XBee	ATWR	N/A (graba configuración en la memoria flash)

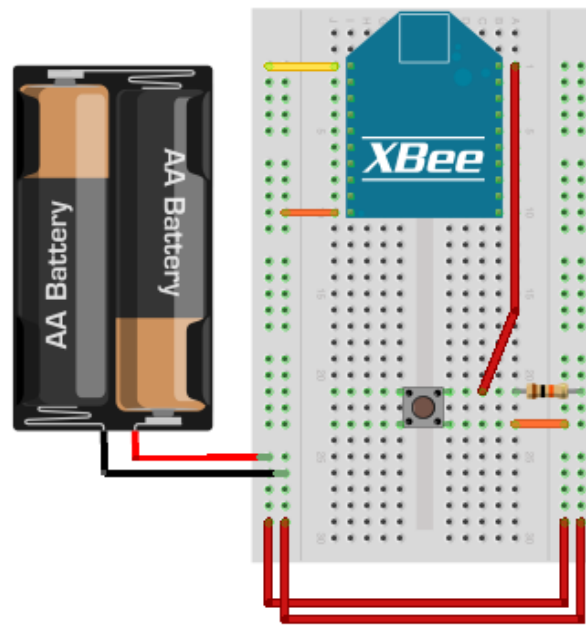
- Esto es lo que escribiremos en la ventana “Terminal” del software de comunicación del puerto USB:

```
+++
OK
ATRE
```

```

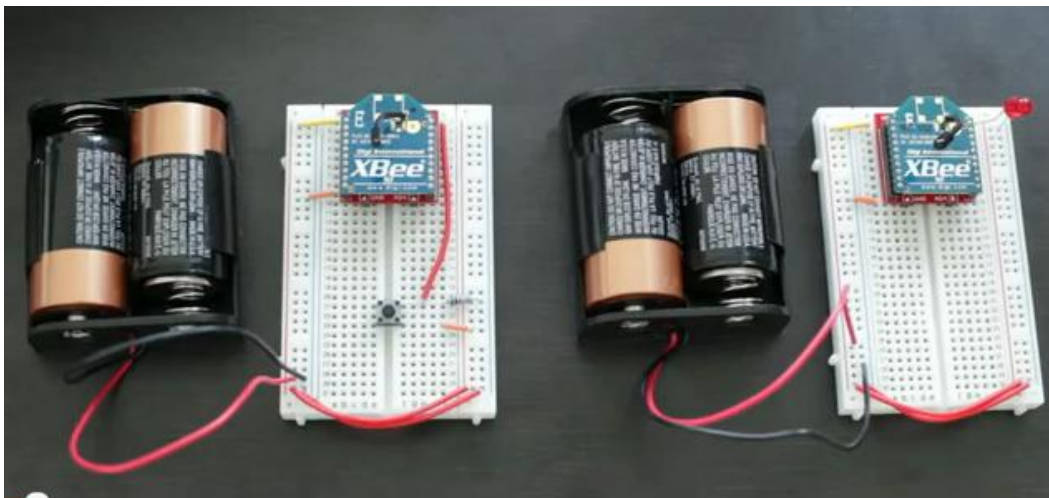
OK
ATID 3001
OK
ATMY 1
OK
ATDH 0
OK
ATDL 2
OK
ATD0 3
OK
ATIR 14
OK
ATPR 0
OK
ATID
3001
ATMY
1
ATDH
0
ATDL
2
ATD0
3
ATIR
14
ATPR
0
ATWR
OK
    
```

**Nota:** Recuerde que hay que finalizar con el comando ATWR.



### Probando el montaje

Para probar el montaje bastara conectar el modulo receptor que se explica en el apartado correspondiente



Este sería el montaje de pruebas.

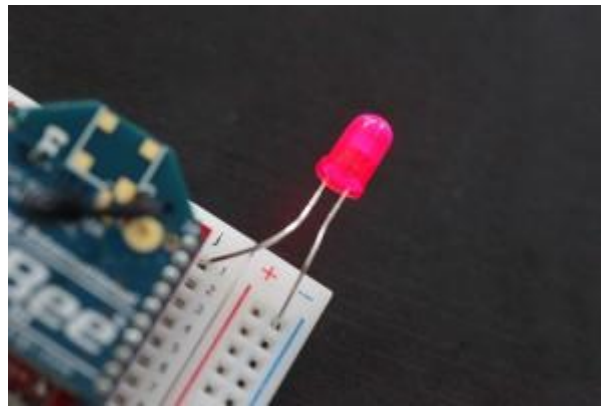


## 3.4 Salida Digital con un diodo LED

(Basado en el trabajo de Matt Richardson publicado en en <http://examples.digi.com/> **Digi XBee® Examples & Guides**)

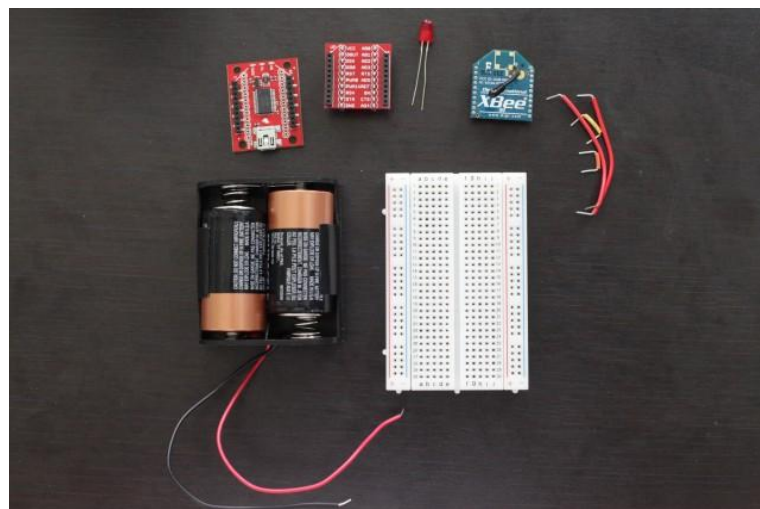
### Introducción:

Se trata de realizar la programación de un módulo XBee que permita controlar el estado de una salida digital conectada en el PIN correspondiente.



Iluminar un LED es una de las primeras cosas que muchas personas hacen cuando empiezan a utilizar un circuito electrónico. El módulo XBee se puede configurar para un máximo de 8 pines de salida digitales que pueden ser controladas directamente por los pines de entrada desde un segundo módulo.

### Materiales necesarios



- Un módulo XBee 802.15.4
- Un LED
- Una protoboard
- Cables

- Un modulo [XBee Breakout Board](#) para ponerlo de base en la protoboar.
- Un módulo de conexión USB con el Ordenador [XBee Explorer USB](#).
- Una fuente de alimentación a base de pilas que suministren una DC entre 2.8 – 3.3V – Podemos usar dos pilas del tipo C.

### Configuración del módulo XBee

Para la programación del módulo procederemos como lo hicimos en el anterior ejemplo. En esta caso los códigos a escribir serian los siguientes.

Función	Comando	Parámetro
Reset	ATRE	N/A (resetea el módulo a sus parámetros de fábrica)
PAN ID	ATID	3001 (cualquier dirección desde 0 a FFFE )
Mi dirección	ATMY	2
Configuración del PIN 0 I/O	ATD0	5 (digital output, HIGH on startup)
I/O Dirección del modulo del que recibiremos las ordenes	ATIA	1 (address of the transmitting radio)
Grabar en memoria la configuración	ATWR	N/A (save the settings to flash memory)

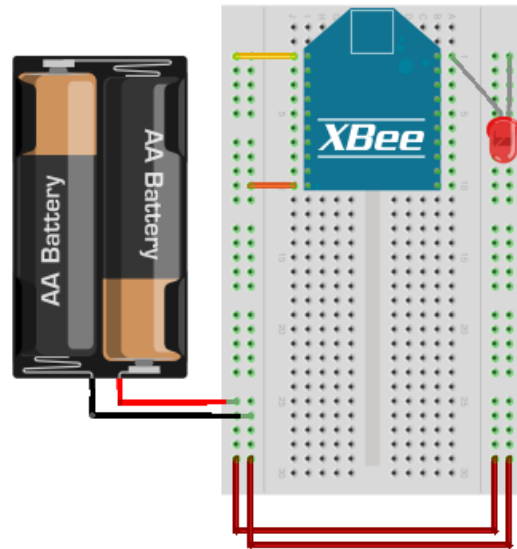
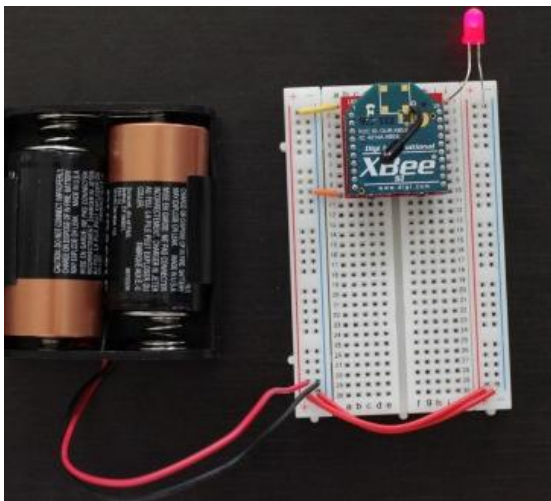
En el terminal se escribirá lo siguiente:

```

+++
OK
ATRE
OK
ATID 3001
OK
ATMY 2
OK
ATD0 5
OK
ATIA 1
OK
ATID
3001
ATMY
2
ATD0
5
ATIA
1
ATWR
OK

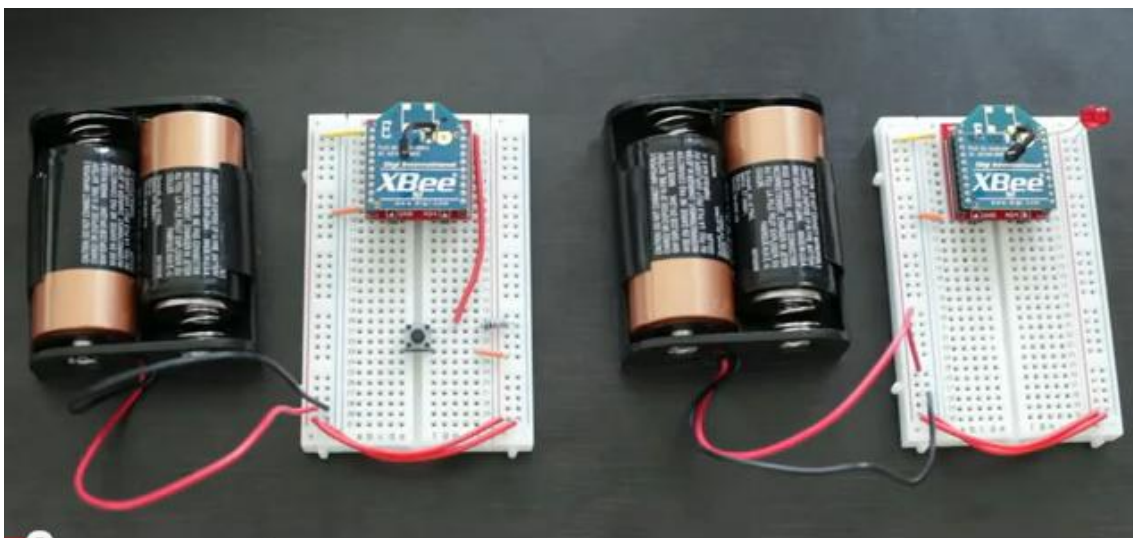
```

**Nota:** Recordar escribir ATWR para grabar la configuración



### Probando el montaje

Para realizar las pruebas de funcionamiento basta que recurrimos al módulo de entrada de señal mediante pulsador explicado anteriormente.



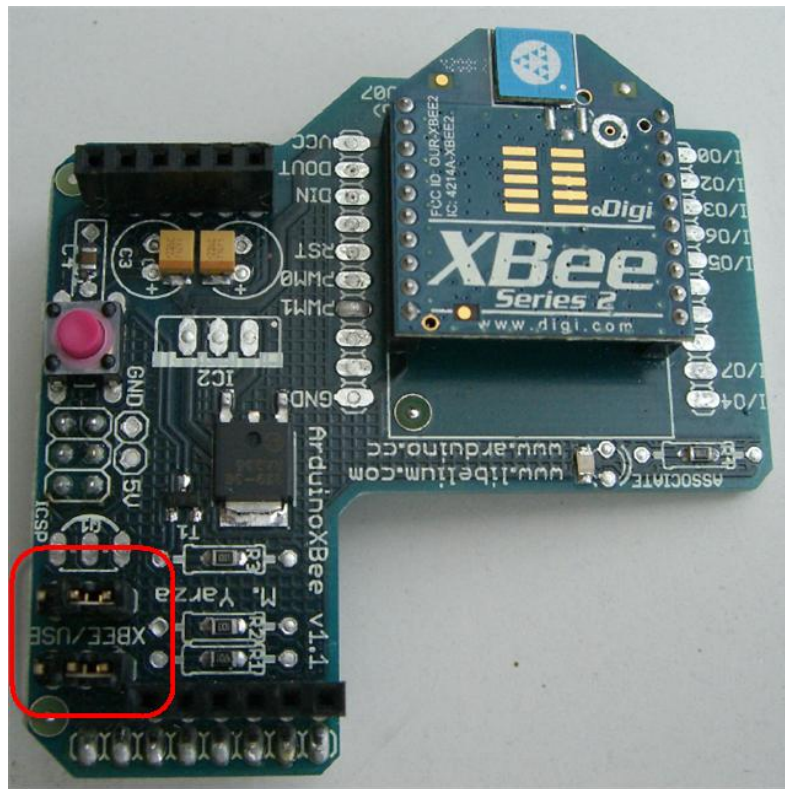
## 4. Trabajo de los módulos XBee con Arduino

### 5.1. Programación de los Módulos XBee

#### Configuración de los jumpers del Shield

El shield de Xbee que usaremos en estas prácticas posee dos jumpers para seleccionar el modo de trabajo del módulo etiquetados con XBee y USB.

Cuando vamos a programar el módulo, previamente debemos colocar los dos jumpers en el modo XBee y levantar de la tarjeta Arduino el microprocesador ya que en este caso lo que hacemos es directamente conectar el modulo a un puerto del PC desde el que realizaremos la conexión y la programación del módulo.



Con los jumpers en la posición **XBee** (e.g. en los dos pines más cercanos al interior de la placa), el pin DOUT de el módulo Xbee está conectado al pin RX del microcontrolador; y el pin DIN está conectado a TX. Notar que los pines RX y TX del microcontrolador están todavía conectados a los pines TX y RX (respectivamente) del chip FTDI - los datos enviados desde el microcontrolador serán transmitidos al ordenador vía USB y a la vez enviados de forma inalámbrica por el módulo Xbee. El microcontrolador, sin embargo, solo será capaz de recibir datos desde el módulo Xbee, no desde el USB del ordenador.

Con los jumpers en la posición **USB** (e.g. en los dos pines más cercanos al borde de la placa), el pin DOUT del módulo Xbee está conectado al pin RX del pin *del chip FTDI*, y el

DIN del módulo Xbee está conectado al pin TX del el chip FTDI. Esto significa que el módulo Xbee puede comunicarse directamente con el ordenador - sin embargo, *esto solo funciona si el microcontrolador ha sido quitado de la placa Arduino*. Si el microcontrolador se deja en la placa Arduino, solo será capaz de comunicarse con el ordenador vía USB, pero ni el ordenador ni el microcontrolador podrán comunicarse con el módulo Xbee. Para cargar un programa en Arduino deberemos tener quitado el shield

## Direccionamiento del modulo XBee

Hay múltiples parámetros que necesitan ser configurados correctamente para que dos módulos puedan comunicarse entre ellos (de todos modos con la configuración por defecto, todos los módulos deberían ser capaces de hablar unos con otros). Necesitan estar en la misma red, definida por el parámetro **ID** (ver "Configuración" a continuación para más detalles sobre los parámetros). Los módulos necesitan estar en el mismo canal, definido por el parámetro **CH**. Finalmente, la dirección de destino de un módulo (parámetros **DH** y **DL**) determina que módulo en esa red y canal recibirá los datos transmitidos. Esto puede suceder de las siguientes formas:

- Si el **DH** de un módulo es 0 y su **DL** es menor de 0xFFFF (e.g. 16 bits), los datos transmitidos por ese módulo serán recibidos por cualquier módulo cuyos 16 bits de dirección del parámetro **MY** sea igual al **DL**.
- Si el **DH** es 0 y el **DL** es igual a 0xFFFF, las transmisiones del módulo serán recibidas por todos los módulos.
- Si el **DH** no es cero o el **DL** es mayor de 0xFFFF, la transmisión solo será recibida por el módulo cuyo número de serie sea igual a la dirección de destino del módulo transmisor (e.g. cuyos **SH** es igual al **DH** del módulo transmisor y cuyo **SL** sea igual a su **DL**).

De nuevo, esta correspondencia de direcciones solo sucederá entre módulos en la misma red y canal. Si dos módulos están en diferentes redes o canales, no podrán comunicarse sea cual sea sus direcciones.

## Configuración.

Volvemos de nuevo a explicar la configuración del modulo XBee, pero esta vez lo haremos con el software CX-CTU

Aquí hay algunos de los parámetros más útiles para configurar tu módulo Xbee. Para unas instrucciones paso-a-paso sobre leerlas y escribirlas, ver la [guía para la Xbee shield](#). Asegúrate de poner AT delante de cada nombre de parámetro cuando envíes un comando a el módulo (e.g. para leer el parámetro ID, deberías enviar ATID). Describimos a continuación los comandos más usuales.

Comando	Descripción	Valores válidos	Valor por defecto
ID	El ID de la red del módulo Xbee.	0 - 0xFFFF	3332
MY	La dirección de 16-bit del módulo.	0 - 0xFFFF	0
DH y DL	La dirección de destino para las comunicaciones inalámbricas(DH son los 32	0 - 0xFFFFFFFF (para ambos DH y y DL)	0 (para ambos DH

	bits superiores, DL son los 32 inferiores).	DL)	
BD	La velocidad de transmisión usada para las comunicaciones con el Arduino o el ordenador.	0 (1200 bps) 1 (2400 bps) 2 (4800 bps) 3 (9600 bps) 4 (19200 bps) 5 (38400 bps) 6 (57600 bps) 7 (115200 bps)	3 (9600 bps)

Nota: a pesar de que los valores por defecto y válidos de arriba están escritos con el prefijo "0x" (para indicar que son números decimales), el módulo no incluye el "0x" cuando reporta el valor de un parámetro, y tu debes omitirlo cuando establezcas valores.

Aquí hay unos cuantos comandos útiles más para configurar el módulo Xbee (necesitarás anteponer AT a estos también).

#### *Comando Descripción*

RE	Restaura los valores por defecto (notar que como el parámetro cambia, esto no es permanente a no ser que esté seguido por el comando WR).
WR	Escribe un nuevo valor para un parámetro configurado a la memoria no volátil (larga-duración). De otro modo, solo durarían hasta que el módulo sea desconectado de la corriente.
CN	Salte del modo de comandos. (Si no mandas ningún comando a él módulo durante unos pocos segundos, el modo de comandos saldrá tras un tiempo incluso sin el comando CN).

Para más detalles sobre configurar el módulo Xbee, ver el [manual de producto](#) de MaxStream.

### **Modo de configuración.**

Para realizar la configuración del modulo XBee se puede utilizar el Hiperterminal de Windows, el software X-CTU sugerido por el propio fabricante o el software UTF-8 TeraTerm Pro.

Para entran en el modo de configuración con el Hiperterminal o con UTF-8 debemos proceder de la siguiente manera:

Necesitamos enviar tres signos '+': +++ y no enviar ningún otro carácter al módulo durante un segundo antes y un segundo después. Esto incluye caracteres especiales como retorno de carro o nueva línea. Por lo tanto, si estas intentando configurar el modulo desde el ordenador, asegúrate de que tu terminal está configurado para enviar los caracteres tal como



los escribes, sin esperar a que pulses intro. Si no, enviará los tres signos '+' seguidos de nueva línea (de modo que no esperará un segundo después de enviar +++). Si consigues entrar en el modo de configuración, el módulo enviará de vuelta 'OK', seguido de retorno de carro.

<b>Envía comando</b>	<b>Espera respuesta</b>
+++	OK<CR>

Una vez en el modo de configuración, podemos enviar comandos AT al módulo. Las cadenas de comandos AT tienen la forma ATXX (donde XX es el nombre del parámetro). Para leer el valor actual de un parámetro, envía la cadena de comando AT seguido de retorno de carro (<CR>). Para escribir un nuevo valor del parámetro, envía la cadena de comando AT, seguido del nuevo valor (sin espacios o nueva línea por medio), seguido de retorno de carro. Por ejemplo, para leer el identificador (ID) de red del módulo (que determina con qué otros módulos XBee se comunicará), usa el comando **ATID**

<b>Envía comando</b>	<b>Espera respuesta</b>
ATID<enter>	3332<CR>

**Para cambiar el valor del ID de red del módulo:**

Envía comando	Espera respuesta
ATID3331<enter>	OK<CR>

**Ahora, comprueba que el parámetro se ha configurado:**

Envía comando	Espera respuesta
ATID<enter>	3331<CR>

A no ser que le digas al módulo que escriba los cambios a la memoria no volátil, esos cambios sólo tendrán efecto hasta que el módulo pierda la alimentación. Para guardar los cambios de forma permanente (hasta que los modifiques otra vez) usa el comando **ATWR**:

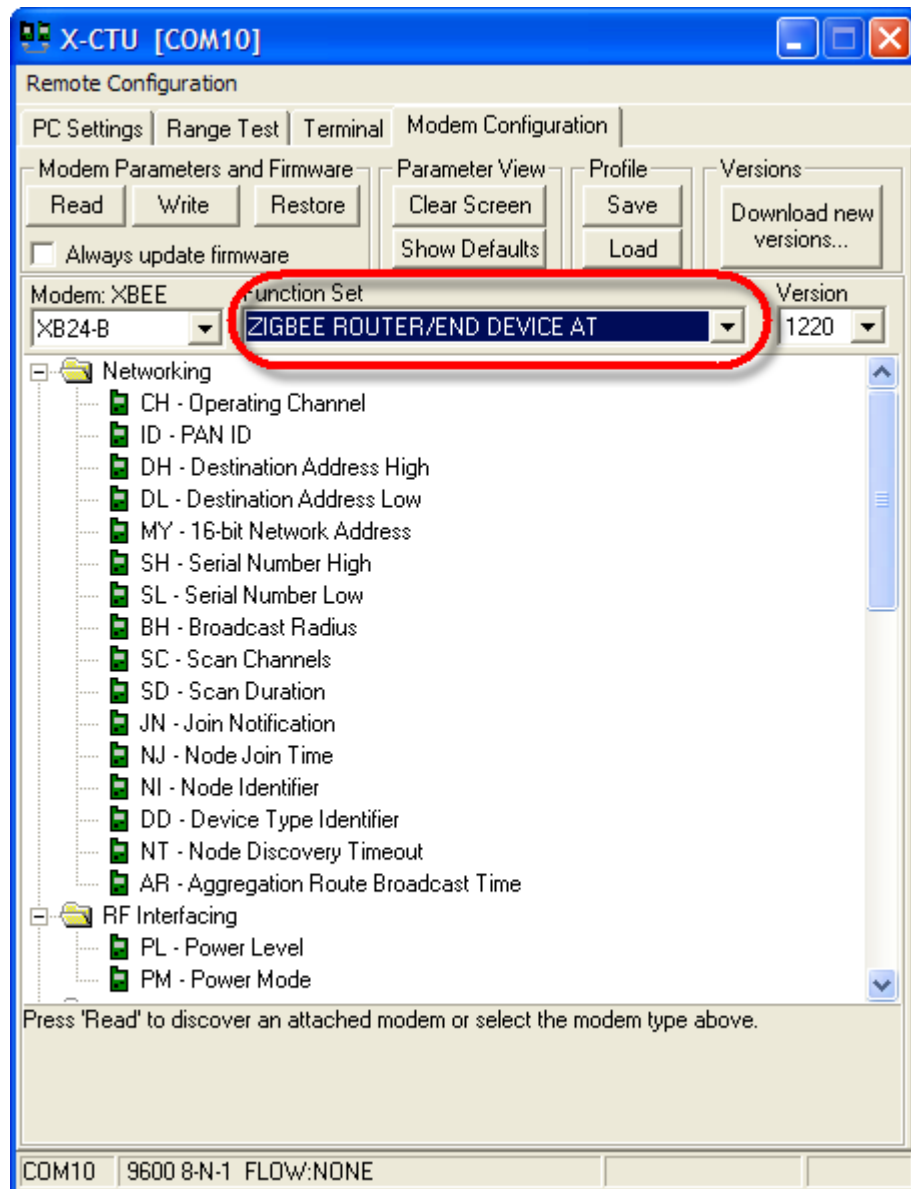
<b>Envía comando</b>	<b>Espera respuesta</b>
ATWR<enter>	OK<CR>

**Para resetear el módulo a los valores de fábrica, usa el comando ATRE:**

Envía comando	Espera respuesta
ATRE<enter>	OK<CR>

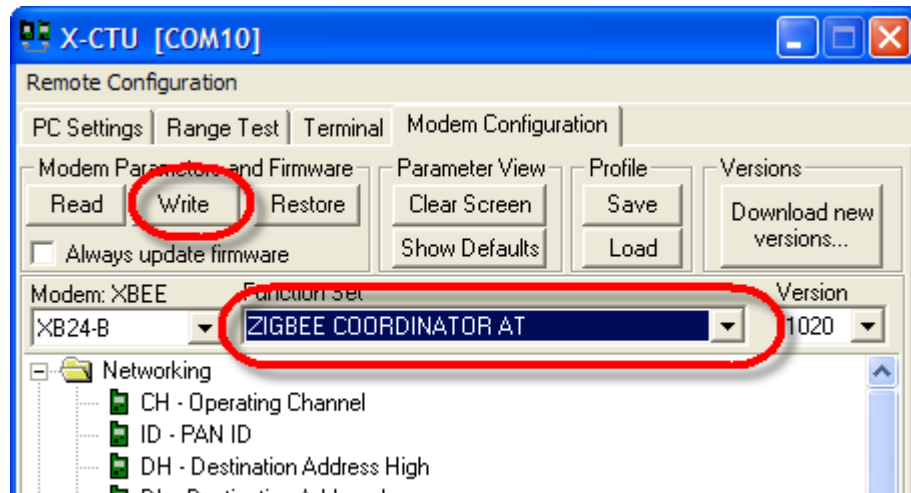
Ten en cuenta que al igual que los demás comandos, el reset no será permanente a no ser que después envíes el comando **ATWR** que lo que hace es grabar la configuración actual de modo permanente.

Otra forma de realizar la configuración más cómoda es mediante el software X-CTU al cargar este software procederíamos de la siguiente manera:



- Nos iríamos al modo “Modem Configuration”
- Pulsaríamos el botón “Read” para leer y detectar el módulo XBee y una vez detectado nos aparecería una ventana en la que se muestran todos los parámetros del módulo y pinchando en cada uno lo podremos modificar.
- Una vez realizada la configuración pulsamos el botón “Write”.
- Podríamos también enviar comandos mediante la opción “Terminal”





### Conexión del módulo XBee sin necesidad de la tarjeta shield

A pesar de que el módulo XBee tiene muchos pines es cierto que para realizar una conexión de este con Arduino solo bastan dos terminales de comunicación y los dos de alimentación. Por este motivo es sencillo realizar la conexión del módulo prescindiendo del shield.

Como los terminales del módulo están separados dos mm se puede conectar sin problema a una protoboard y realizar el cableado con Arduino. Solo debemos tener la precaución de alimentar con 3.3 v. la unidad. Esta tensión está presente en la tarjeta Arduino.

La siguiente tabla muestra las conexiones entre pines de Arduino y XBee y la imagen de abajo les muestra en una placa XBee ruptura.

### La asignación de clavijas entre Arduino XBee y

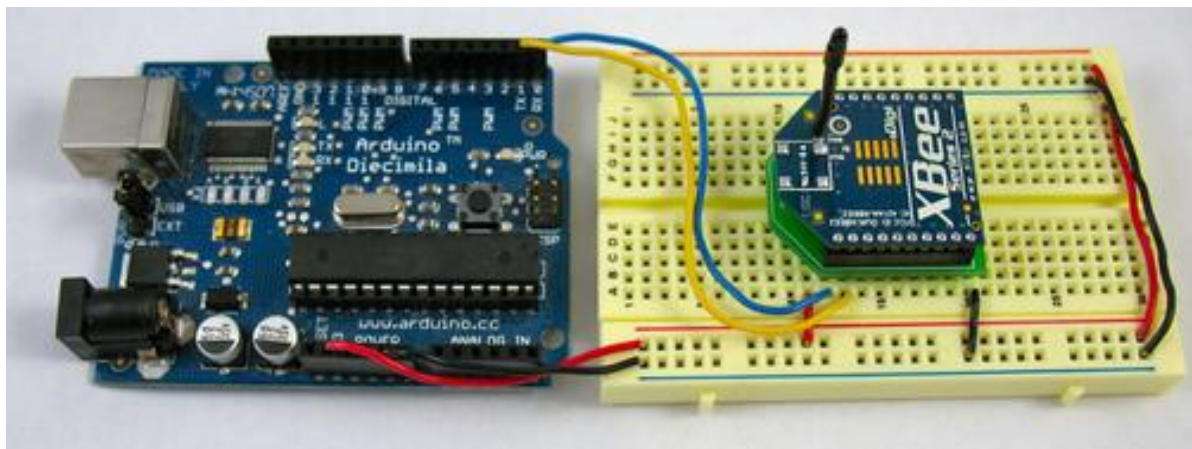
XBee	Arduino
VCC V o 3,3	3V3
TX o DOUT	0 RX 0
RX o DIN	TX o 1
GND	GND



Diagrama de pines del módulo Xbee. Vista Superior.



## Pines utilizados para una conexión básica con Arduino



Arduino conectado a una radio XBee, utilizando una placa sin soldadura y alambre de conexión

## Como utilizar entradas/salidas directamente del módulo XBee.

Cada XBee ZB tiene la capacidad de recoger directamente los datos de un sensor y transmitirlos sin necesidad de Arduino. También se pueden utilizar algunas salidas exteriores en el módulo. Por ejemplo, es posible enviar información digital directamente a un módulo de radio XBee para que se active una salida.

## Características de las E/S XBee.

La Serie 2 XBee hardware ofrece varias funciones flexibles para proyectos que necesitan de entrada simple y salida. Hay 10 pines que pueden configurarse como entradas digitales para sensores interruptores y otras cosas que funcionan como interruptores, o como salidas digitales para el control de LEDs y motores pequeños directamente. Grandes cargas, incluidos las que funcionan con corriente alterna, puede funcionar con estas salidas digitales a través de un relé. Los primeros cuatro pines pueden ser configurados como entradas analógicas para la detección de una enorme variedad de fenómenos cuyos valores varíen dentro de un rango, tales como la luz, la temperatura, la fuerza, la aceleración, la humedad, los niveles de gas, etc.

Sólo hay 10 pins totales por lo que si usted tiene las 10 entradas digitales configuradas, se encuentra fuera de los pines y no se puede utilizar cualquier entrada o salida digital analógica. Afortunadamente, los pines se pueden utilizar en una mezcla. Por ejemplo, tres entradas analógicas, cuatro entradas digitales y tres salidas digitales estaría bien. La única cosa a tener en cuenta es que muchos de los 10 pines configurables se utilizan para otras funciones opcionales. Estos otros deberes son importantes en muchas aplicaciones, pero han sido cuidadosamente seleccionados para que sean los que no suelen ser necesarios en los proyectos de teledetección y actuación.

#### **Nombres de los pines de entrada / salida con los números físicos, comandos y otras funciones**

<b>Pin nombre</b>	<b>Pin físico #</b>	<b>Comando AT</b>	<b>Otras funciones</b>
DIO0, AD0	20	D0	Entrada analógica, puesta Button
ESD1, AD1	19	D1	Entrada analógica
ESD2, AD2	18	D2	Entrada analógica
ESD3, AD3	17	D3	Entrada analógica
ESD4	11	D4	
DIØ5	15	D5	Asociación indicador
DIO6	16	D6	RTS
DIO7	12	D7	CTS
(DIO8)	9	Ninguno	Pin de control del sueño, DTR
(DIO9)	13	Ninguno	On / Sleep indicador
ESD10	6	P0	Indicador de intensidad de señal recibida (RSSI)
DIO11	7	P1	
DIO12	4	P2	

## 4.2 Test de una comunicación PC ->XBee

Se trata de realizar una prueba de dialogo entre dos unidades XBee , una remota conectada a una tarjeta Arduino Uno a la que llamaremos terminal **Receptor** y otra conectada directamente al PC a través de uno de sus puerto USB a la que llamaremos **Emisor**. Ambos módulos estarán unidos en una comunicación dúplex es decir bidireccional.

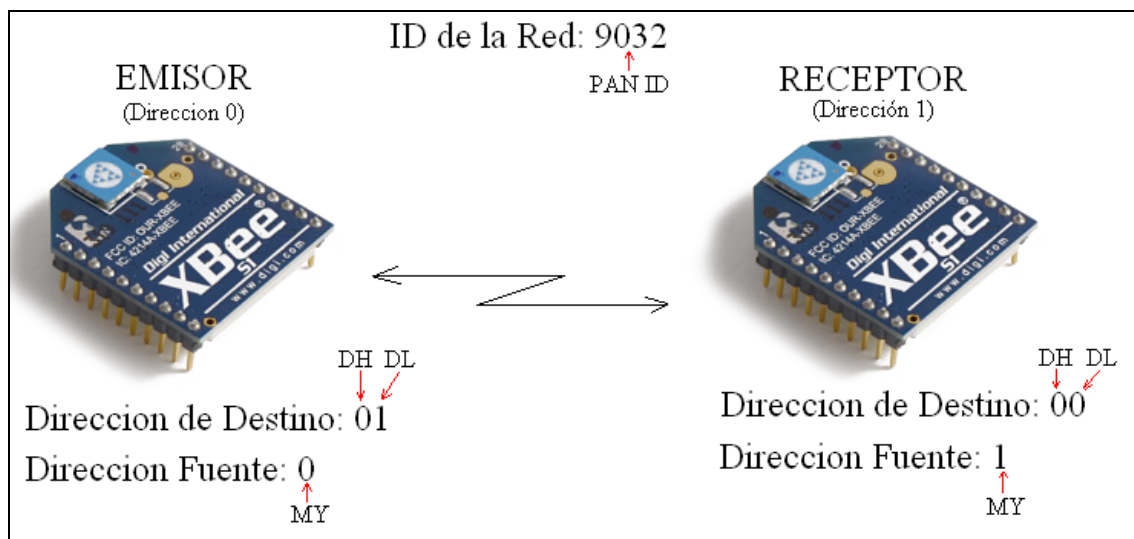
El módulo Receptor se alimentara con una pila y la unidad XBee trabajara gobernada por la tarjeta Arduino (switches del shield colocados en modo XBEE)

El módulo Emisor tendrá su XBee trabajando en modo XBee directamente (los switches del shield estarán en modo XBee), por lo que directamente utilizando el software X-CTU en modo monitor podremos leer y escribir directamente en la unidad XBee, por lo tanto emitiremos y recibiremos datos que podremos ver en el “terminal”.

Lo primero que haremos será programar los módulos Receptor y Emisor de acuerdo a la siguiente tabla:

	Xbee-Envia	Xbee-Recibe
PAN ID	9032	9032
DH	0	0
DL	1	0
MY	0	1

En la tabla no olvidemos que:



**PAN ID** Representa el código nidentificativo de la red que pretendemos montar

**DH y DL** Forman la Dirección de Destino (DH parte alta y DL parta baja) del Modulo (01 para el Emisor y 00 para el receptor)

**MY** Representa la dirección con la que se comunica el módulo (El Emisor se comunica con el elemento 0 que es el Receptor y el Receptor se comunica con el elemento 1 que es el Emisor)

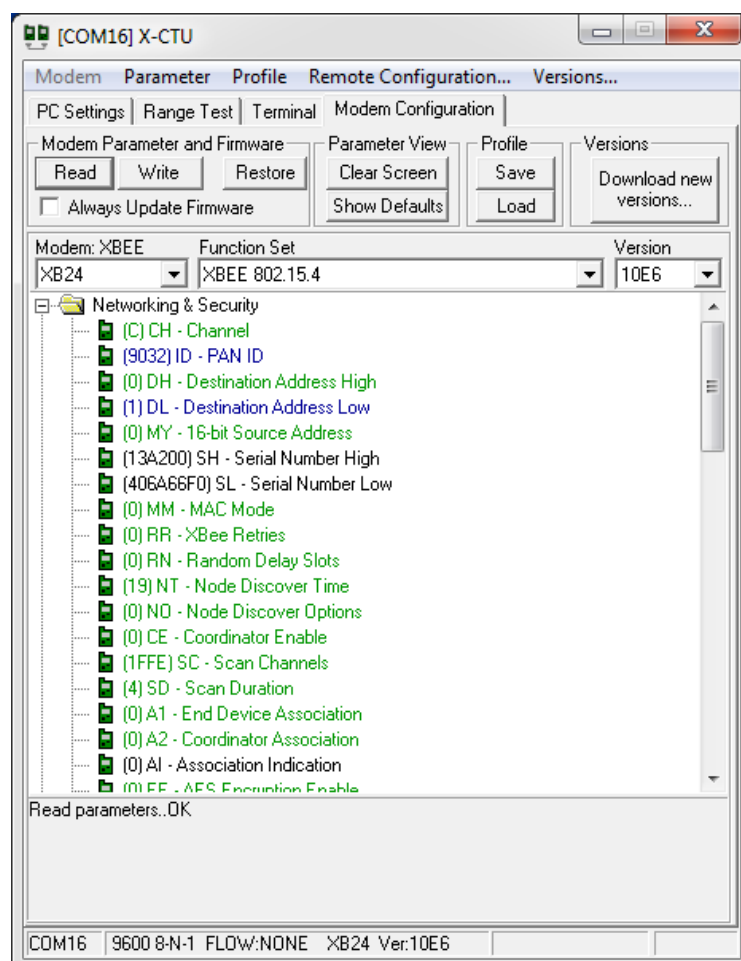
## Programación del módulo XBee Emisor.

Se conecta el módulo sobre el Shield y este sobre la tarjeta Arduino Uno teniendo la precaución de:

- Sacar del zócalo el Microcontrolador de la tarjeta Arduino
- Colocando los puentes del shield en modo USB

Se abre la aplicación software X-CTU y se selecciona la opción “Modem Configuration”. Después se debe pulsar el botón “Read” para leer el tipo de dispositivo conectado y una vez leído aparecerá la lista de los parámetros del módulo. De los parámetros mostrados se seleccionan:

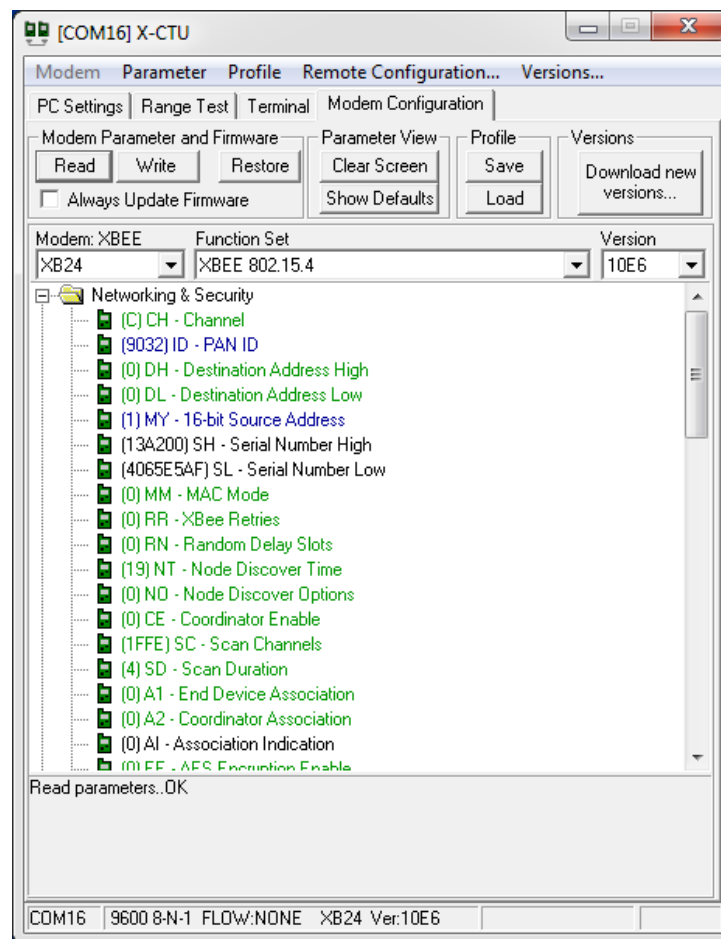
<b>PAN ID</b>	Se escribe <b>9032</b>
<b>DH Destination Address High</b>	Se escribe <b>0</b>
<b>DL Destination Address Low</b>	Se escribe <b>1</b>
<b>MY 16-bit Source Address</b>	Se escribe <b>0</b>
<b>BD Interface Rate</b>	Se escribe <b>9200</b> (baudios)



## Programación del módulo XBee Receptor.

Se procede de la misma manera que con el módulo Emisor pero en este caso los datos serán:

<b>PAN ID</b>	Se escribe <b>9032</b>
<b>DH Destination Address High</b>	Se escribe <b>0</b>
<b>DL Destination Address Low</b>	Se escribe <b>0</b>
<b>MY 16-bit Source Address</b>	Se escribe <b>1</b>
<b>BD Interface Rate date</b>	Se escribe <b>9200</b> (baudios)

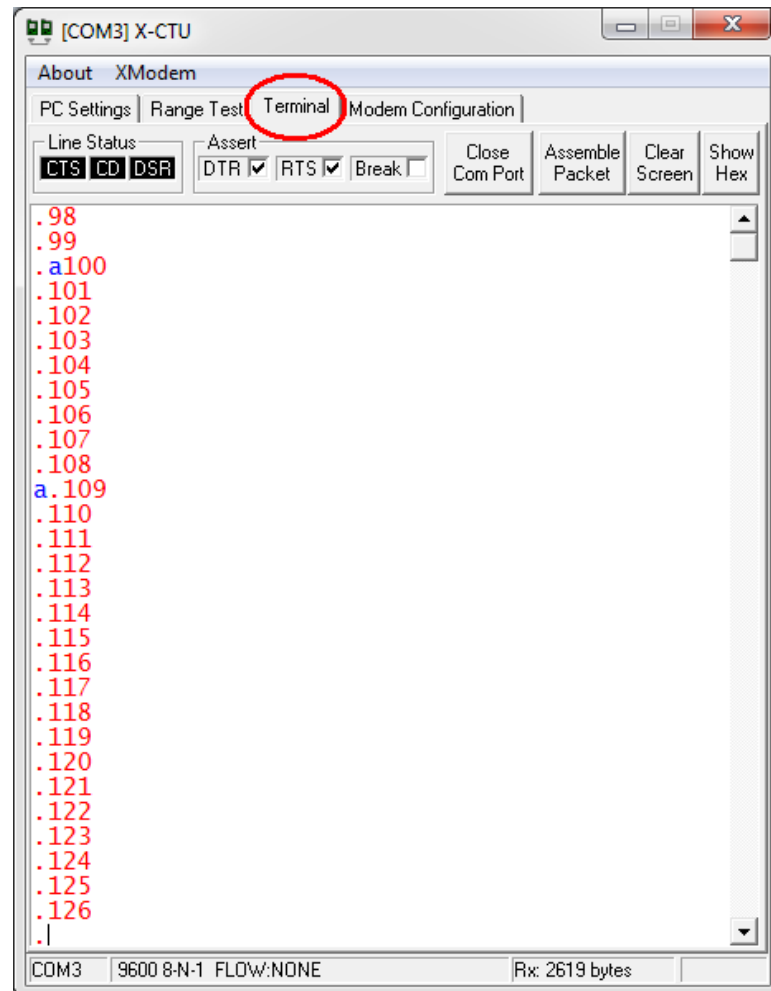


## El código a escribir.

Una vez programados los módulos procederemos a escribir un programa que vamos a colocar en el modulo Remoto y que básicamente lo que hará este programa es : Por un lado generar un numero que se ira incrementando y lo enviará a través del puerto al XBee que hemos llamado Emisor que estará conectado en el PC trabajando en modo XBee sobre el Shield pero con la tarjeta Arduino sin microcontrolador. Mediante el software C-CTU trabajando en el modo “Monitor” podamos leer este valor que se ira incrementando.

Por otro lado colocaremos un diodo LED en el PIN 13 del modulo Receptor y lo activaremos y desactivaremos escribiendo en el PC (a través del puerto conectado al termino XBee Emisor) una “a” que al llegar al módulo remoto Receptor activara y desactivara el LED en modo conmutador.

En la figura vemos el terminal leyendo y escribiendo la letra “a” de control del LED



A continuación vemos el programa que debemos cargar en el Arduino Uno que se coloca con el modulo distante Receptor.

/\*

-----  
*Escritura serial inalámbrica - XBee Serie 1*  
 -----

*Consiste en escribir por la pantalla del computador (consola serial) una letra predeterminada, la primera vez que se escriba esta un LED se enciende, si se vuelve a escribir por segunda vez el LED se apaga, además el modulo XBee (E) ubicado en el computador esta recibiendo en tiempo real el incremento de un contador ubicado en el modulo XBee (R). Esta comunicación serial se establece de manera inalámbrica por medio de dos módulos XBee - Serie 1*

*Cosas de Mecatronica y Tienda de Robotica*

```

*/

//-----
//Declara puertos de entradas y salidas y variables
//-----
int led = 13;      //Pin donde se encuentra el LED, salida
char leer;        //Variable donde se almacena la letra
boolean prendido=false; //Estado LED la primera vez, apagado
int contador = 0;  //Variable del contador

//-----
//Función principal
//-----
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  Serial.begin(9600); //Inicia comunicación serial
  pinMode(led, OUTPUT); //Configurar el LED como una salida
}

//-----
//Funcion ciclica
//-----
void loop() // Esta funcion se mantiene ejecutando
{          // cuando este energizado el Arduino

  contador++;          //Incrementa el contador en una unidad
  Serial.println(contador); //Imprime por consola el valor del contador
  delay(100);          //Retardo

  //Guardar en una variable el valor de la consola serial
  leer=Serial.read();

  // Si es la letra 'a' y ademas el LED esta apagado
  if ( (leer=='a') && (prendido==false) )
  {
    digitalWrite(led,HIGH); // Enciende el LED
    prendido=true;          // Actualiza el estado del LED
  }
  // Si es la letra 'a' y ademas el LED esta encendido
  else if ( (leer=='a') && (prendido==true) )
  {
    digitalWrite(led,LOW); // Apaga el LED
    prendido=false;        // Actualiza el estado del LED
  }
}

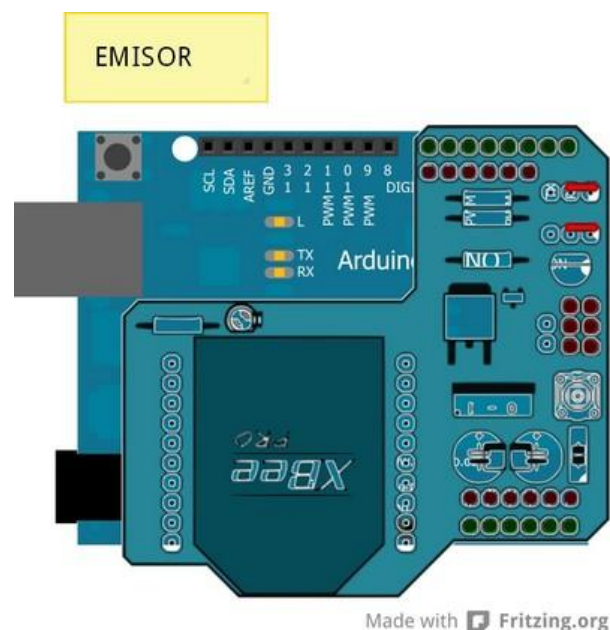
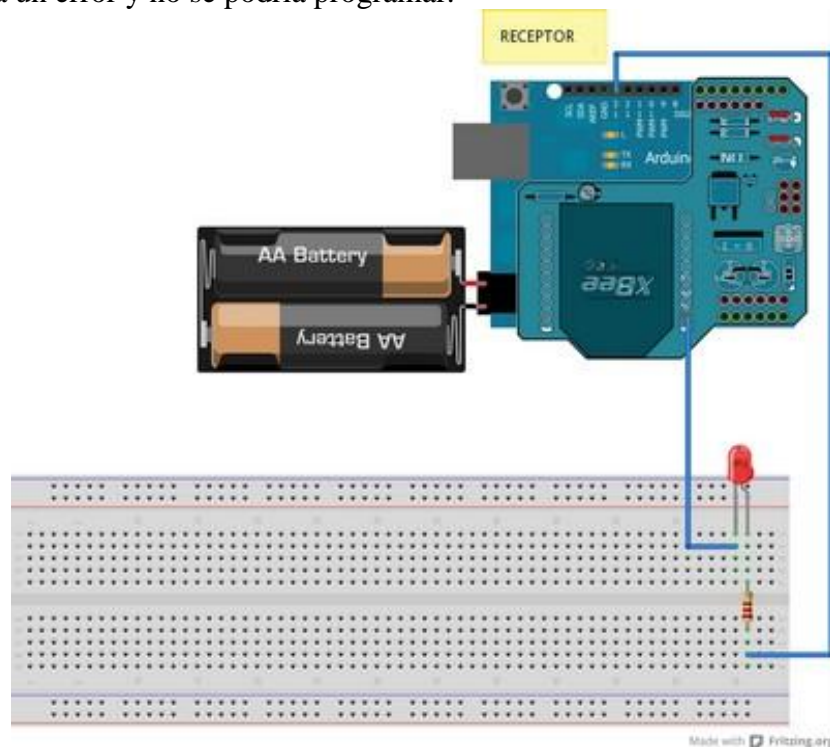
```



```
}
```

```
//Fin programa
```

**NOTAS IMPORTANTES:** No debemos olvidar conectar adecuadamente los puentes en las tarjetas shield: En el modulo Emisor (PC) en la posición USB sin el chip microcontrolador en la tarjeta Arduino. En el módulo Recetor se deben conectar los puentes en modo XBEE. Cuando se descargue el programa sobre la tarjeta Arduino Remota se debe quitar el shield por que si no daría un error y no se podría programar.



## 4.3 Timbre inalámbrico - XBee Serie 1

Con este montaje vamos a probar los módulos XBee dialogando. Se trata de colocar en uno de ellos al que llamaremos EMISOR un pulsador conectado en el terminal digital PIN 2 de la tarjeta Arduino UNO que enviara una orden al terminal RECEPTOR en el que colocaremos un zumbador en el terminal digital PIN 5 y un LED en el terminal digital PIN 13 de la tarjeta Arduino UNO.

### Funcionamiento de la aplicación:

Al pulsar sobre el pulsador se enviara una orden a la estación remota RECEPTOR en la que se encenderá el diodo LED y a la vez zona el zumbador.

En la estación de Emisión (E) tenemos un pulsador para poder activar vía inalámbrica el buzzer y el LED ubicados en la estación de Recepción (R). Luego de este tutorial ya tendrás conocimientos para poder elaborar tus propios proyectos inalámbricos bajo el envío de diversas tramas de comunicación.

### Programación de las unidades XBee.

La programación de las unidades XBee es exactamente la misma que hemos realizado para la anterior práctica.

#### Programación del módulo XBee Emisor.

<b>PAN ID</b>	Se escribe <b>9032</b>
<b>DH Destination Address High</b>	Se escribe <b>0</b>
<b>DL Destination Address Low</b>	Se escribe <b>1</b>
<b>MY 16-bit Source Address</b>	Se escribe <b>0</b>
<b>BD Interface Rate date</b>	Se escribe <b>9200</b> (baudios)

#### Programación del módulo XBee Receptor.

Se procede de la misma manera que con el módulo Emisor pero en este caso los datos serán

<b>PAN ID</b>	Se escribe <b>9032</b>
<b>DH Destination Address High</b>	Se escribe <b>0</b>
<b>DL Destination Address Low</b>	Se escribe <b>0</b>
<b>MY 16-bit Source Address</b>	Se escribe <b>1</b>
<b>BD Interface Rate date</b>	Se escribe <b>9200</b> (baudios)

### Materiales que utilizaremos

Dos módulos XBee 1mW Serie1

Dos tarjetas Arduino Uno  
 Dos tarjetas Shield XBee  
 Un diodo led  
 Un zumbador  
 Un pulsador  
 Una resistencia de 10k  
 Una resistencia de 220 ohmios  
 Una protoboard

La distancia entre la Estación de Emisión (E) y la Estación de Recepción (R) debe ser 30m si se encuentra en interiores y si se encuentran en línea abierta la máxima distancia es de 90m.

## Programa

Vamos a desarrollar el código tanto para la Estación de Emisión (E) como para la Estación de Recepción (R)

A continuación se encuentra el programa desarrollado

### Estación de Emisión (E)

/\*

-----  
*Timbre inalámbrico - Estación Emisión (E)*  
 -----

*Consiste en escribir por la pantalla del computador (consola serial) una letra predeterminada, en este caso la letra 'D', cada vez que se presione el pulsador se va a enviar la letra, como el Arduino tiene conectado un módulo XBee esta letra se enviar de manera inalámbrica.*

*Cosas de Mecatronica y Tienda de Robotica*

\*/

//-----

//Declara puertos de entradas y salidas y variables

//-----

int boton = 2; //Pin donde se encuentra el pulsador, entrada

//-----

//Funcion principal

//-----

void setup() // Se ejecuta cada vez que el Arduino se inicia

{

Serial.begin(9600); //Inicia comunicación serial

pinMode(boton,INPUT); //Configura el pulsador como una entrada

```

}

//-----
//Funcion ciclica
//-----
void loop() // Esta funcion se mantiene ejecutando
{          // cuando este energizado el Arduino

    //Pregunta si el pulsador esta oprimido
    if(digitalRead(boton)==HIGH){
        Serial.print('D'); //Imprime vía serial la letra 'D'
        delay(10);         //Retardo para no congestionar la escritura serial
    }

}

//Fin programa

```

## Estación de Recepción (R)

```

/*
-----
Timbre inalámbrico - Estación Recepción (R)
-----

Este programa se mantiene recibiendo datos por el puerto serie,
si recibe la letra esperada en este caso la 'D', el programa
activa el buzzer (alarma sonora) y activa el LED (alarma visual)

Cosas de Mecatronica y Tienda de Robotica

*/

//-----
//Declara puertos de entradas y salidas y variables
//-----
int buzzer=5; //Pin donde se encuentra el buzzer, salida
int led=13;   //Pin donde se encuentra el led, salida

//-----
//Funcion principal
//-----
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
    Serial.begin(9600); //Inicia comunicación serial
    pinMode(buzzer,OUTPUT); //Configura el buzzer como una salida
    pinMode(led,OUTPUT);   //Configura el led como una salida
}

```

```

}

//-----
//Funcion ciclicla
//-----
void loop() // Esta funcion se mantiene ejecutando
{          // cuando este energizado el Arduino

    // Pregunta si hay algún dato en el puerto serial
    if(Serial.available()>0){

        // Pregunta si el dato recibido es la letra 'D'
        if(Serial.read() == 'D'){

            digitalWrite(buzzer,HIGH); //Activa buzzer
            delay(10);                  //Retardo
            digitalWrite(buzzer,LOW);  //Apaga buzzer

            digitalWrite(led,HIGH);    //Activa led
            delay(10);                  //Retardo
            digitalWrite(led,LOW);     //Apaga led

        } //Corchete If lectura letra 'D'

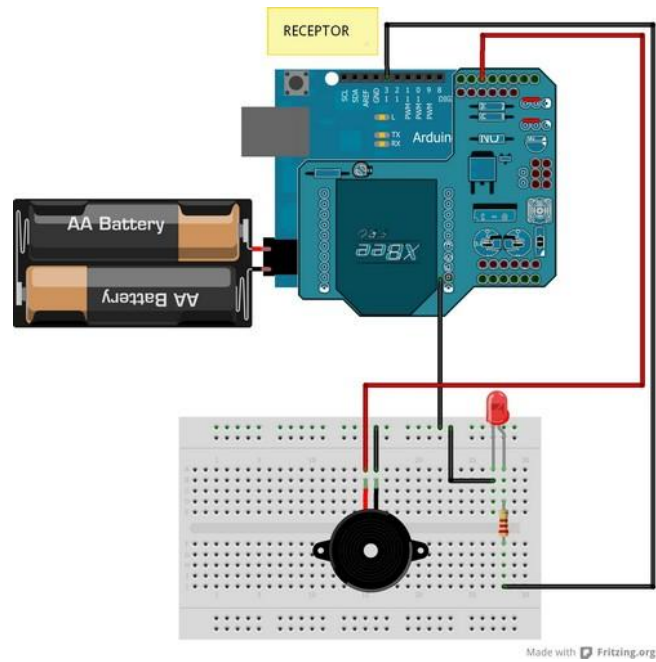
    } //Corchete If lectura puerto serie

}

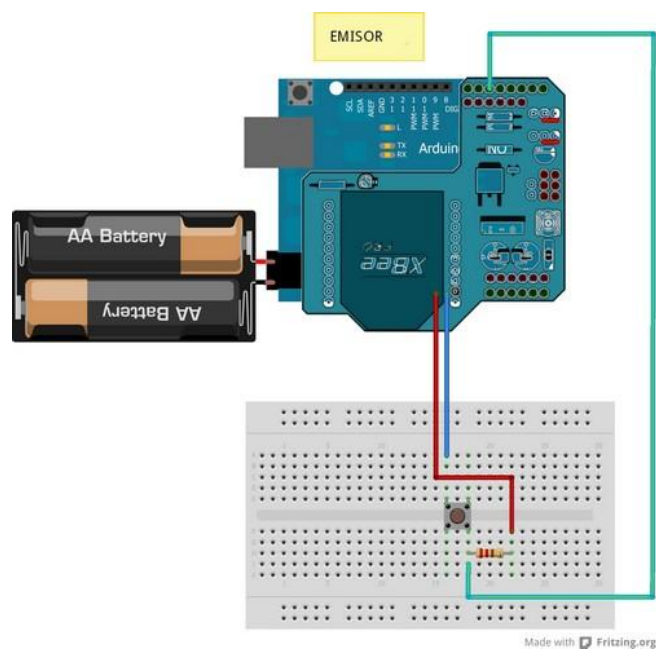
//Fin programa

```

## Montajes de los circuitos



Receptor



Emisor

## 6. Documentación Software y Bibliografía

### Páginas WEB

Guia de Usuario de XBee

[http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia\\_Usuario.pdf](http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf)

Blog de Electrónica:

<http://www.blogelectronica.com/author/admin/>

Lugar oficial de la plataforma Arduino:

<http://www.arduino.cc/es/Main/Software>

XBEE CL Módulos de Transmisión Inalámbrica:

<http://www.xbee.cl/index.html>

Digi XBee® Examples & Guides.

<http://examples.digi.com/>

Tienda Sparkfun

<https://www.sparkfun.com/>

Arduino XBee Shield

<http://arduino.cc/es/Guide/ArduinoXbeeShield>

Manual de usuario de XBee en español

[http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia\\_Usuario.pdf](http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf)

Cosas de mecatronica

<http://www.cosasdemecatronica.com/tutoriales>

Cooking Hacks (Distribución componentes)

<http://www.cooking-hacks.com/index.php/shop/arduino/wireless.html>

Robotica Educativa

<http://www.aprenderrobotica.com/>

XBee Tutorial

<https://sites.google.com/site/xbeetutorial/home>

Digi. Learning Center

<http://www.digi.com/learningcenter/>

Primeros pasos en XBee, usando arduino

<http://www.internetdelascosas.cl/primeros-pasos-en-xbee-usando-arduino/>

XBee Radios

<http://www.ladyada.net/make/xbee/>

### Libros

- Margolis, Michael. “**Arduino CookBook**”, 2da edición, O’Reilly. Link [aquí](#)
- Faludi, Robert. “**Building Wireless Sensor Networks: A practical guide to the ZigBee Mesh Networking Protocol**”. Ed, O’Reilly [aquí](#)

### Software:

Software [XCTU](#) Download Site

Software Teraterm <http://tssh2.sourceforge.jp/>

Drivers FTDI [Pulsar aquí para descargarlos de Internet.](#)

Software CoolTerm: [CoolTerm by Roger Meier](#)

Terminal. <https://sites.google.com/site/terminalbpp/>

ZidBee Operator <http://www.zigbeeoperator.com/>

Iron Moltosenso Network Manager <http://www.moltosenso.com/>