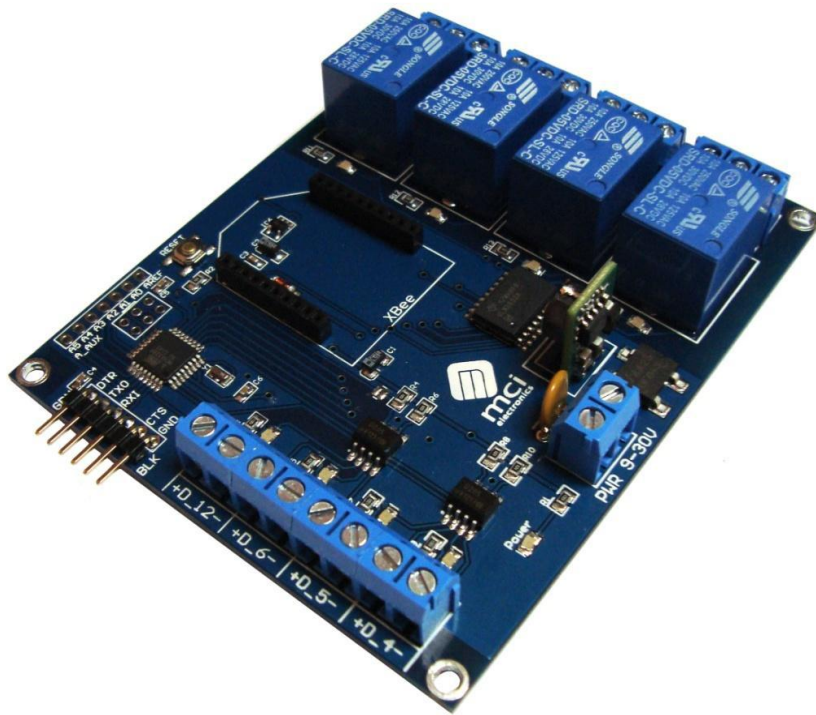


# Manual de Usuario

[Xbee IO Pro]

Rev. [1.0]

MCI-MA-1026



**INGENIERÍA MCI LTDA.**

Luis Thayer Ojeda 0115. Of. 402 - Providencia, Santiago, Chile - Tel. +56 2 3339579 - info@olimex.cl

[www.olimex.cl](http://www.olimex.cl)





MCI Ltda.

Luis Thayer Ojeda 0115. Oficina 402  
Santiago, Chile

[www.olimex.cl](http://www.olimex.cl)

Tel: +56 2 3339579

Fax: +56 2 3350589

® MCI Ltda. 2012

Desarrollado por Cesar Muñoz para MCI

**Atención:** Cambios y modificaciones hechas en el dispositivo, no autorizados expresamente por MCI, anularán su garantía.

Código Manual: MCI-MA-1026

## Descripción

La tarjeta Xbee IO Pro nos permite manejar de forma inalámbrica las entradas y salidas presentes en el módulo, ya sea por medio de un PC o de algún microcontrolador usando un módem Xbee. Como la tarjeta Xbee IO PRO es “Arduino Compatible” (con el modelo Arduino Pro Mini 5V 16 MHz), nos permite usar toda la potencia y versatilidad de la plataforma Arduino. La asignación de pines está descrita en la documentación de la tarjeta Xbee IO Pro.

Por medio de esta plataforma se puede monitorear entradas de sensores digitales y análogos, controlar actuadores por medio de relés, enviar o recibir datos por medio de una interfaz serial y comunicar todo por la red Zigbee usando los módulos Xbee Serie 1 (para comunicación punto a punto), o mediante una red distribuida (mesh) usando los módulos Xbee Serie 2.

Entre las ventajas que presenta esta tarjeta podemos mencionar: la capacidad de usar las librerías Arduino, el monitoreo en tiempo real de variables por medio de una red inalámbrica, el control automático de actuadores usando el propio microcontrolador presente en la tarjeta o por medio de un PC en forma remota, a través de un terminal serial o por algún otro medio, como puede ser una página Web montada en el PC.

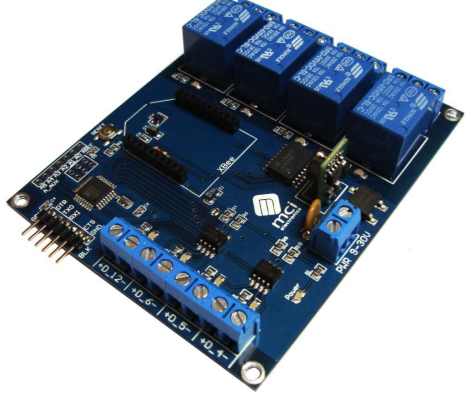
Haciendo uso de los pines del Módulo FTDI, se puede tener una interfaz serie de uso general que es muy simple de controlar por medio de la librería “Serial” presente en la plataforma de software, como veremos en nuestros ejemplos podemos destinar esta interfaz para leer y escribir datos desde y hacia otros dispositivos en forma conjunta o distribuida.

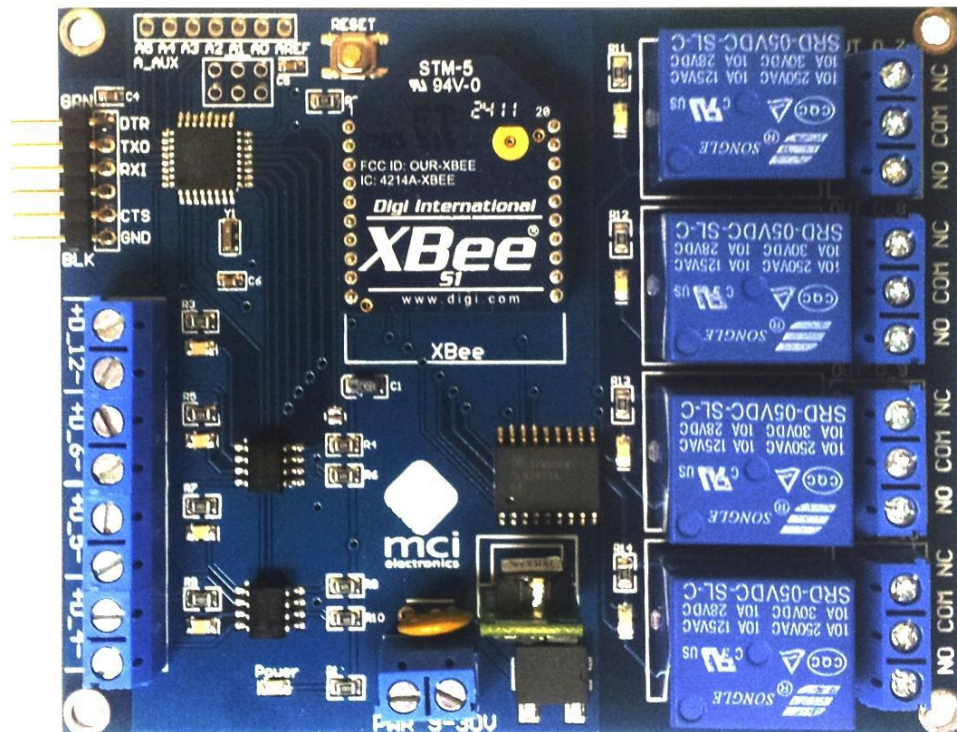
Su alimentación es otra de las ventajas que presenta esta tarjeta, y algo que se incorpora en gran parte de los módulos de [MCI](#), pues permite alimentar el sistema con una entrada de 9-30 [VAC/VDC], permitiéndonos usar casi cualquier fuente que encontremos en el mercado o en variados sistemas industriales.

Las limitaciones de esta tarjeta son la poca disponibilidad de pines de uso general, sin embargo este módulo está diseñado para realizar tareas de monitoreo y control, permitiendo acoplar variados sensores (digitales y/o análogos) y realizar tareas de control en forma automática.

# MANUAL

Este modelo cuenta con

4 entradas optoacopladas	
4 salidas con relés de 10A	
6 entradas analógicas	
Comunicación inalámbrica	
Comunicación I2C	
Alimentación 9-30 VDC	
Socket para XBee	



## Recursos Necesarios

[Xbee IO Pro](#)

[Xbee Explorer USB](#)

[Módulo FTDI 5\[V\]](#)

2 [Xbee](#) "Serie 1" o "Serie 2"

Algunos Sensores que deseemos probar<sup>1</sup> (análogos y/o digitales)

[Contacto Magnético](#)

[Botón Pulsador](#)

[Sensor Presencia \(PIR\)](#)

[Sensor Proximidad](#) (medidor de distancia)

[Sensor de Temperatura ZX](#)

[Sensor de Luz](#)

Otros componentes<sup>2</sup>

[LCD](#) con [interfaz serializada](#)

[Lector RFID](#) y [RFID USB Reader](#)

Programas

[Arduino](#)

[XCTU](#)<sup>3</sup> ([www.digi.com](http://www.digi.com) ->Support -> XCTU)

[Driver FTDI](#)<sup>4</sup>

[Putty](#) o [Hércules](#) (para manejar el puerto serie)

PC O Notebook

Es importante tener presente que a través de los relés podemos controlar también elementos de bajo Voltaje, siendo este dispositivo una manera excelente de obtener una aislación completa desde un circuito hacia un actuador u otro circuito, permitiéndonos evitar contaminaciones electromagnéticas (EMI/EMC).

---

<sup>1</sup> Los sensores son opcionales, aunque necesarios para una experiencia más completa sobre las funcionalidades que nos entrega la tarjeta.

<sup>2</sup> Al igual que los sensores, estos componentes son opcionales.

<sup>3</sup> Es conveniente asegurarse que la versión que vayamos a instalar sea una reciente o actualizada.

<sup>4</sup> Solo para Windows 7, las versiones anteriores de Windows pueden encontrar el driver de manera automática en Windows Update.

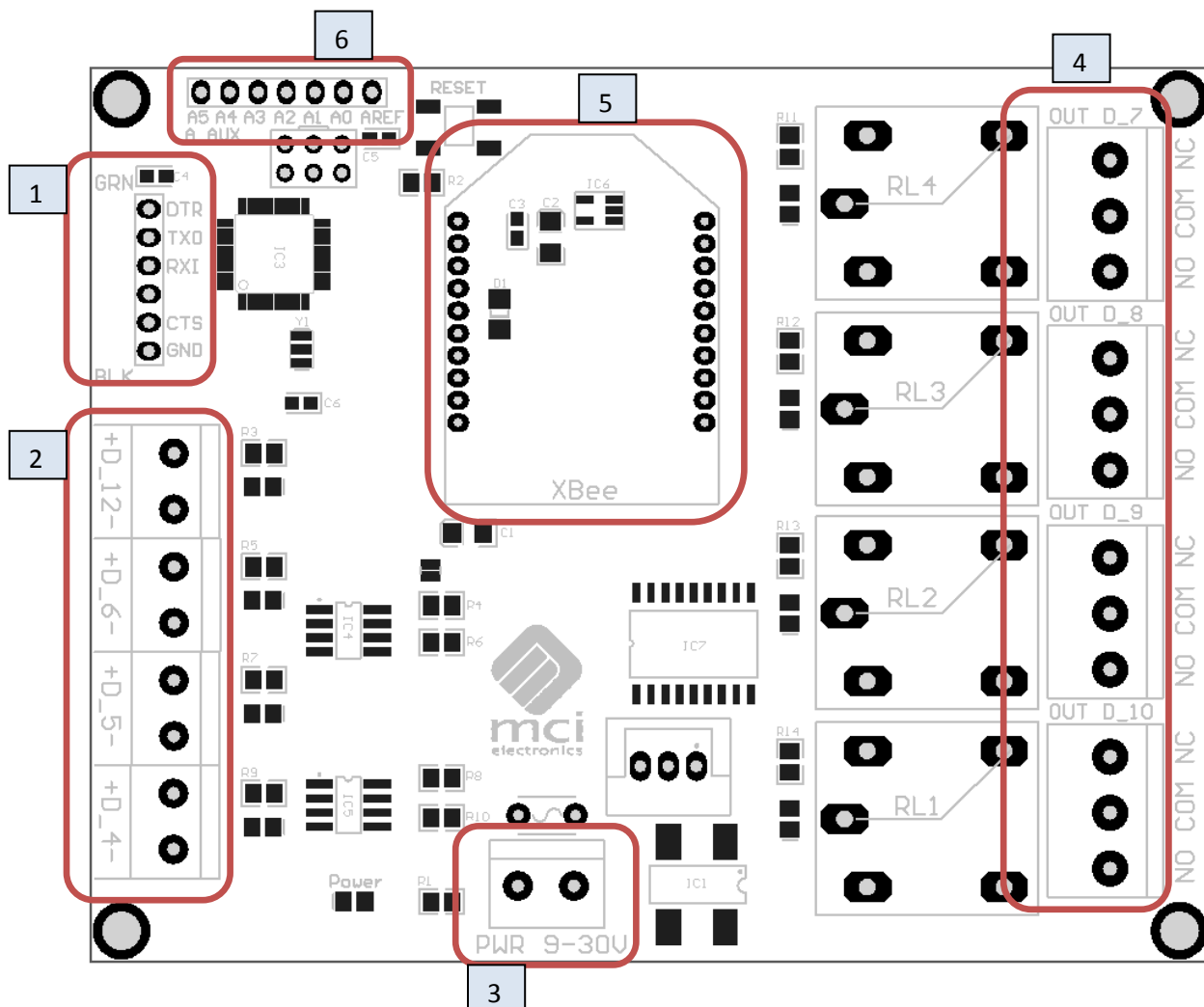
## Configuraciones y Esquema Eléctrico

En esta sección describiremos como realizar el conexionado completo de los sensores y actuadores que utilizaremos para nuestro proyecto, para poder explorar las potencialidades de la tarjeta [Xbee IO Pro](#) y cómo hacer para controlarla en forma remota por medio de una interfaz serie en nuestro PC.

### Conexionado

El siguiente diagrama describe nuestra tarjeta [Xbee IO Pro](#) de manera esquemática, en ella indicaremos donde debemos conectar los sensores para operar nuestro ejemplo. Como indicamos anteriormente, contamos con varios sensores y dispositivos para su uso:

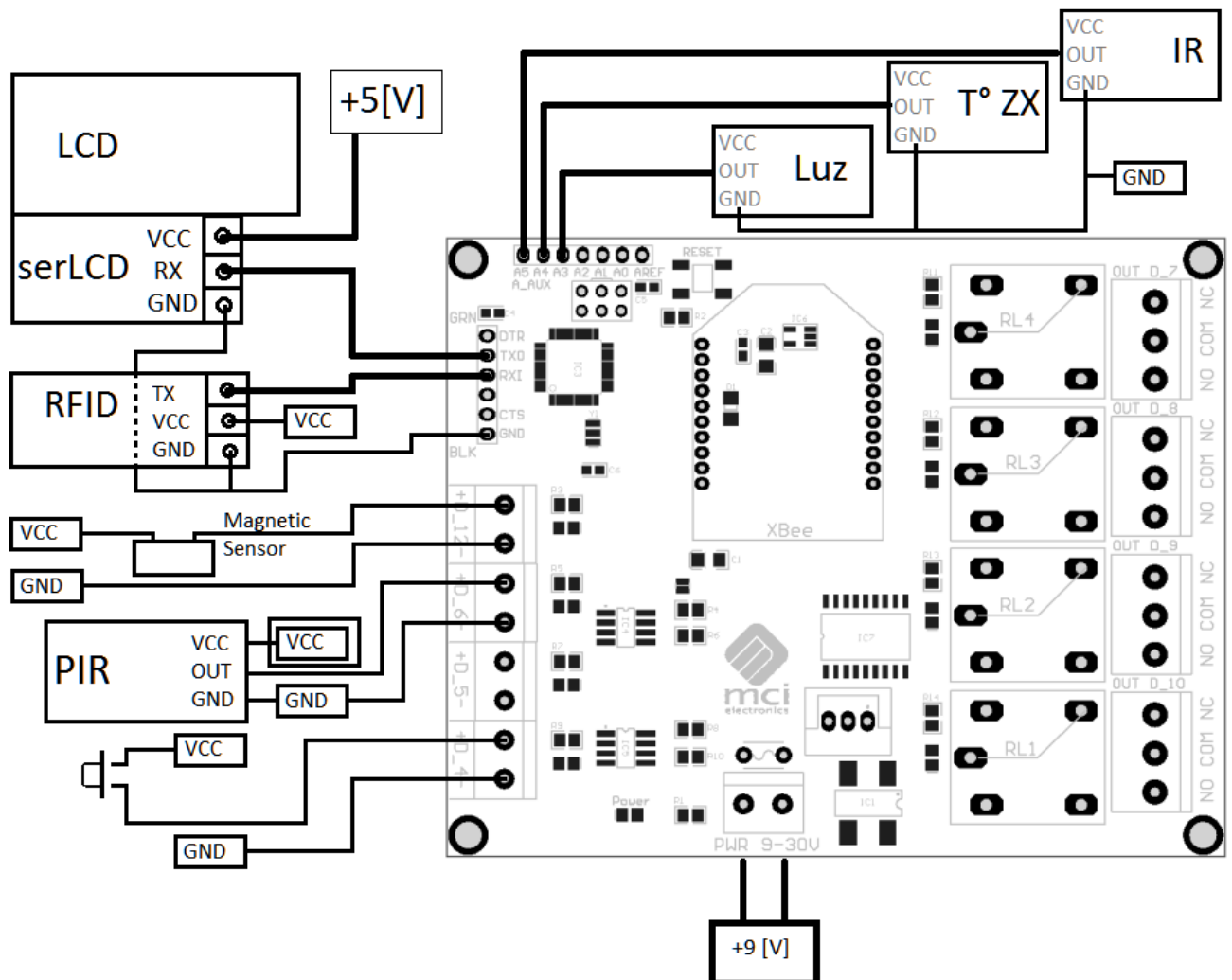
Un Módulo de Comunicación Xbee, tres sensores análogos, tres sensores digitales, un LCD, un RFID que son los que debemos conectar a nuestra tarjeta.



# MANUAL

En cada uno de los sectores marcados, encontramos:

N°	Descripción
1	Interfaz Módulo FTDI / Serial
2	Entradas Digitales Optoacopladas
3	Power
4	Salidas con Relé
5	Conector Módulo Xbee
6	Salidas Análogas



## Procedimiento de Armado

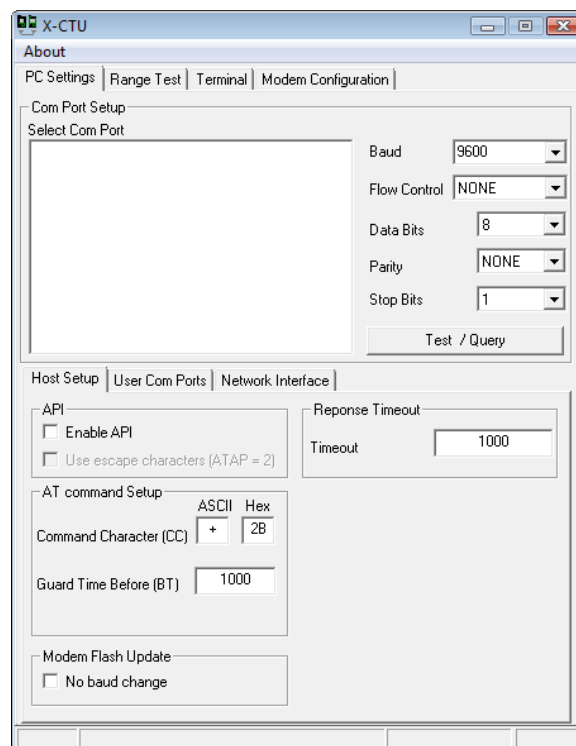
### *Procedimiento de Ejecución del experimento*

Habiendo realizado el procedimiento de la sección anterior, debemos realizar la configuración de nuestros dispositivos de comunicación Módulos Xbee Serie 1 o Serie 2. Según sea la elección que se haya hecho se debemos proseguir en la sección que corresponda.

Una vez realizado lo anterior, continuaremos con la programación de la tarjeta Xbee IO Pro, por medio de la plataforma Arduino.

### *Instalar XCTU*

La instalación de la plataforma [XCTU](#) es bastante simple, si bien puede tomar una cantidad de tiempo considerable dado que el programa bajará los firmwares actualizados para los diferentes módems que puede manejar. Una vez realizado el proceso de instalación nos encontraremos con la siguiente interfaz.



El manejo del programa es muy simple, si bien nos presenta muchas opciones solo debemos usar una cantidad muy limitada de ellas.

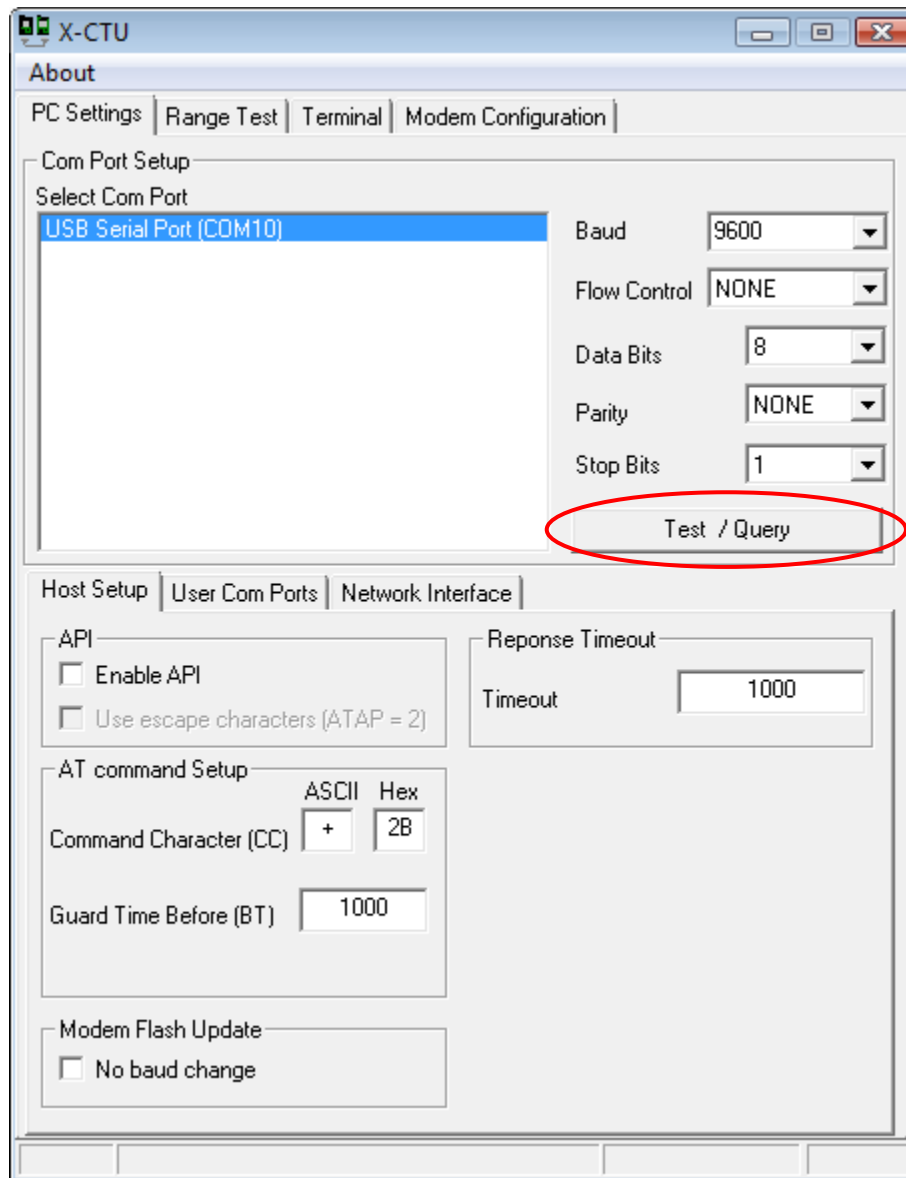


### ***Instalar Xbee Explorer USB***

Antes de continuar, debemos tener instalado el [Xbee Explorer USB](#), el cual nos permitirá configurar y operar el módem Xbee desde nuestro PC. Simplemente debemos poner el modem "Xbee" en el "Xbee Explorer USB" y enchufar el puerto USB a nuestro computador, para las versiones de Windows Vista y anteriores, los drivers serán instalados de manera automática. Para Windows 7 debemos realizar el proceso de instalación de los drivers de manera manual desde el administrador de dispositivos.

## ***Configurar los módulos Xbee***

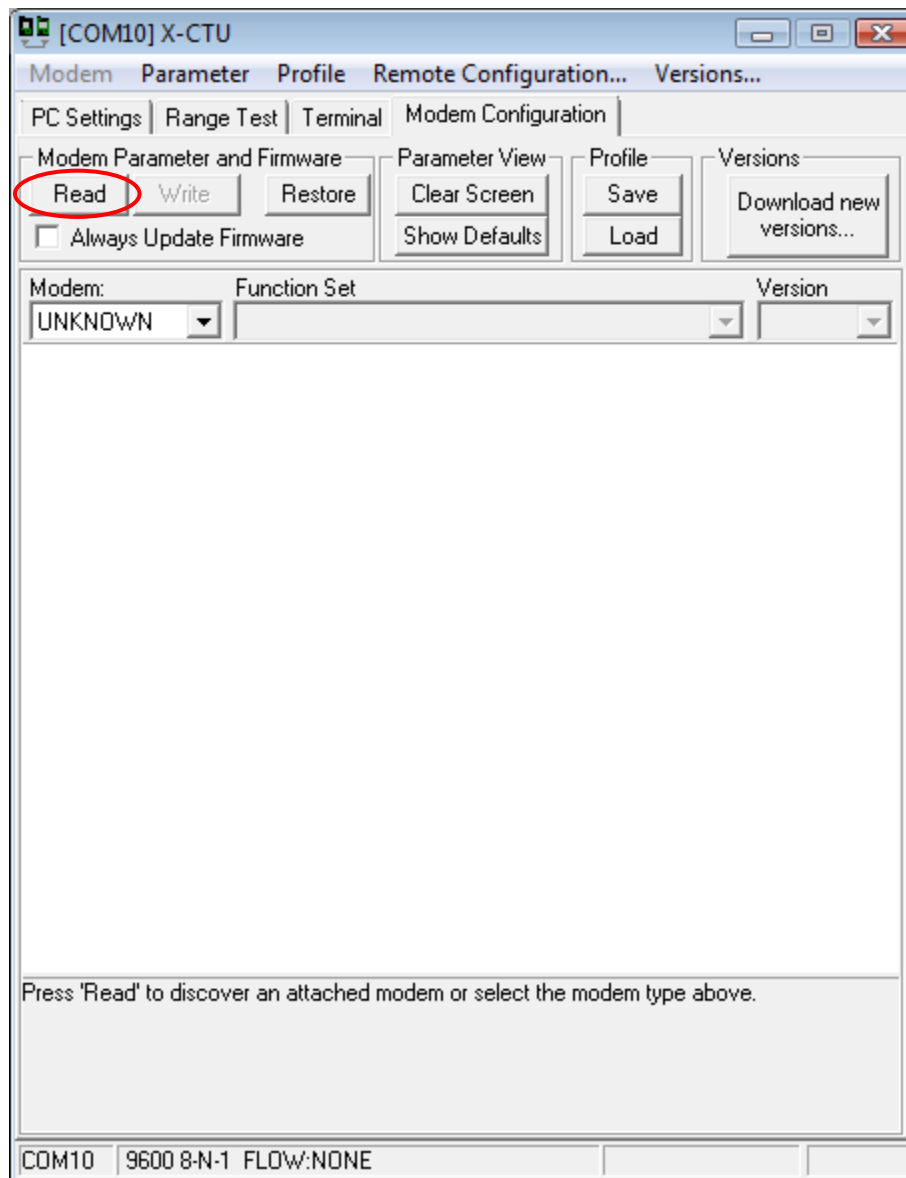
Una vez realizado el proceso de instalación del "Xbee Explorer USB" deberemos abrir nuevamente el XCTU y veremos como el puerto que acabamos de instalar aparece en la lista de puertos COM. Para asegurarnos que todo está operativo, podemos presionar el botón "Test/Query" y se desplegará una pequeña ventana con la información de nuestro módulo Xbee.



## MANUAL

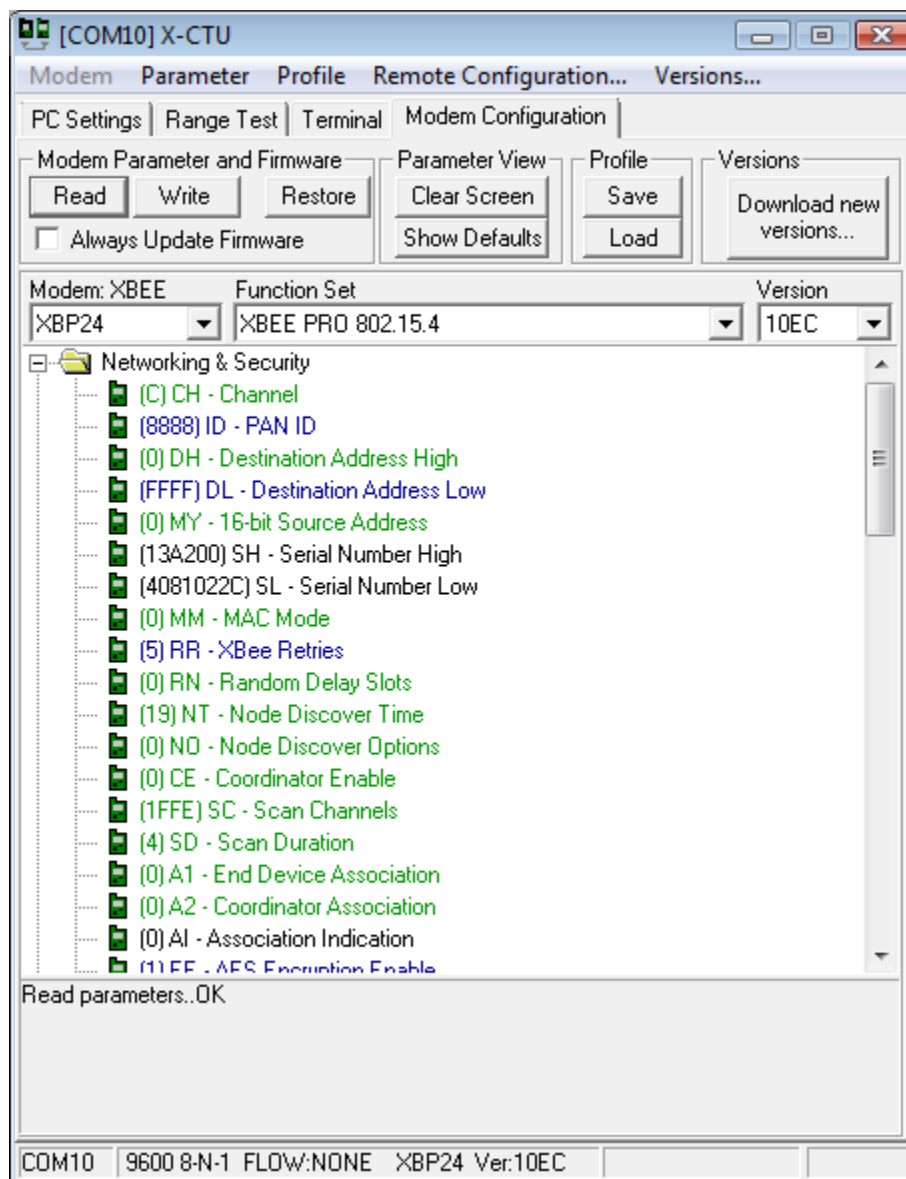
Para configurar el módulo Xbee debemos acceder a la pestaña “Modem Configuration” y presionar el botón “Read” con lo cual se leerán los parámetros del modem que tenemos conectado.

Ahora, debemos seguir en la sección correspondiente al tipo de módulo que hayamos adquirido (Serie 1 o Serie 2).



## ***Cómo configurar la red Xbee Serie 1***

En la imagen podemos encontrar la configuración básica de un módulo [Xbee Serie 1](#).




# MANUAL

En las siguientes tablas, podemos encontrar la descripción de los campos y los valores que debemos asignar a cada uno de ellos para configurar los [Xbee Serie 1](#) en “modo transparente” o en una conexión punto a punto.

Indicador	Nombre
DH	Destination Address High
DL	Destination Address Low
MY	16-bit Source Address
ID	PAN ID
SH	Serial Number High
SL	Serial Number Low
CE	Coordinator Enable

A continuación se muestra un ejemplo de cómo se deben de configurar los dos módulos:

XBee A Valores	XBee B Valores
DH 13A200	DH 13A200
DL 4076E267	DL 4076E26E
MY AAAA	MY AAAA
SH 13A200 (viene por defecto)	SH 13A200 (viene por defecto)
SL 4076E26E (viene por defecto)	SL 4076E267 (viene por defecto)
CE 1 -Coordinator	CE 0 -End Device Serie 1 Pro



XBee A
XBee B

Existen otras formas de configurar una red Xbee, la anterior es una forma fácil que solo involucra dos módems; aunque en todas ellas debemos asegurarnos que se comparta el mismo identificador de la red (PAN ID), pues este valor nos permite diferenciar redes que se encuentren cercanas y descarta cualquier recepción que no corresponda con el valor PAN ID de nuestro modem.

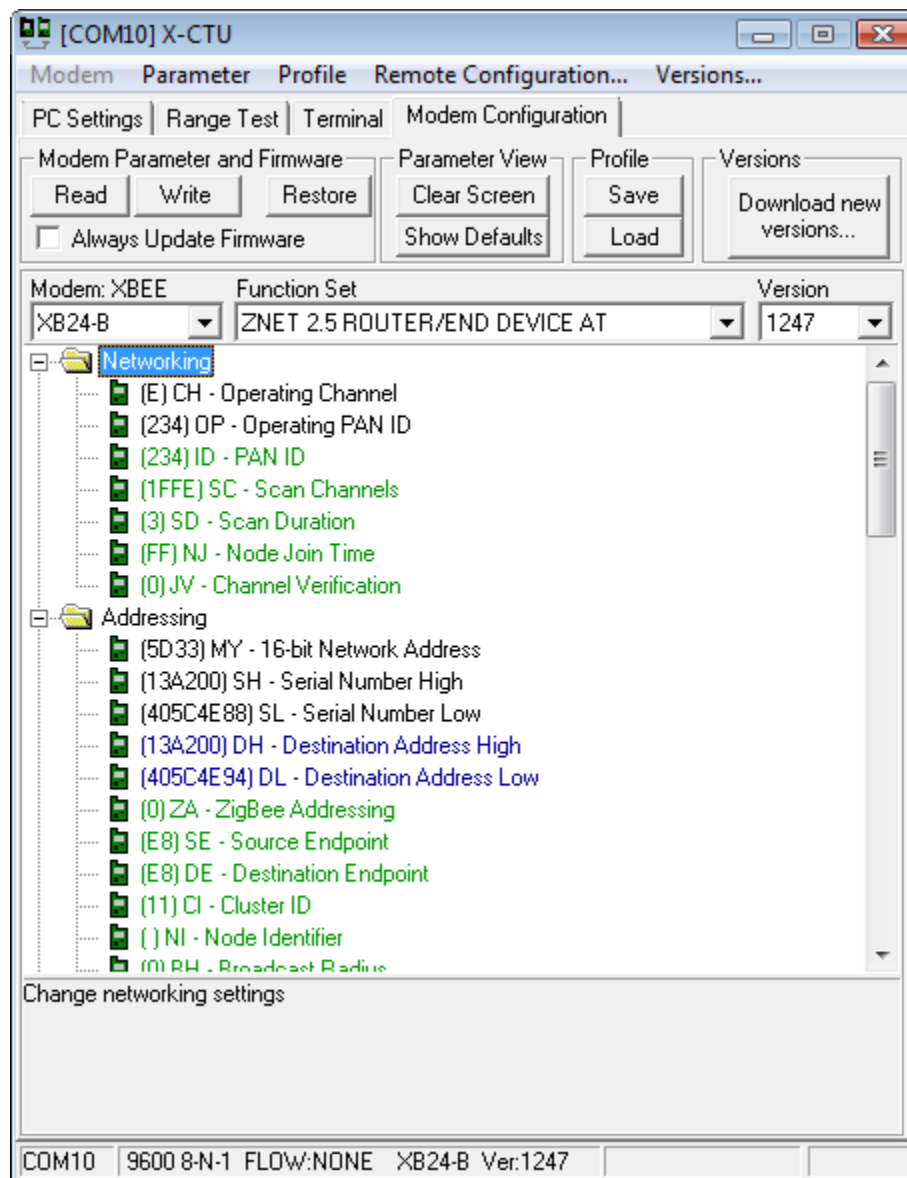
Si deseamos conectar 3 nodos, podemos optar por una forma fácil donde todos se escuchen entre ellos, algo que es similar a una red tipo bus. Para ello, optaremos por hacer un broadcast desde cada uno de los Xbee Serie 1 configurándolos de la siguiente forma:

<b>XBee A</b>	<b>XBee B</b>	<b>Xbee C (o cualquier otro)</b>
DH 0	DH 0	DH 0
DL FFFF	DL FFFF	DL FFFF
MY 0	MY 0	MY 0
PAN ID 8888	PAN ID 8888	PAN ID 8888
SH (viene por defecto)	SH (viene por defecto)	SH (viene por defecto)
SL (viene por defecto)	SL (viene por defecto)	SL (viene por defecto)
CE 1 -Coordinator	CE 0 -End Device	CE 0 -End Device

Para usar esta red, debemos hacer un pequeño protocolo por software, que envíe un identificador desde cada módem y que se tomen acciones por cada uno de estos identificadores (y aquellos que no sean dirigidos al módem involucrado deben ser descartados).

## ***Cómo configurar la red Xbee Serie 2***

En la imagen podemos encontrar la configuración básica de un módulo [Xbee Serie 2](#), no hay muchas diferencias en su forma básica de configuración con la Serie 1, si bien en esta versión encontraremos más campos y una distribución un poco diferente de los mismos.



En las siguientes tablas, podemos encontrar la descripción de los campos y los valores que debemos asignar a cada uno para configurar los Xbee Serie 2 en “modo transparente” o en una conexión **punto a punto**

Indicador	Nombre
<b>DH</b>	Destination Address High
<b>DL</b>	Destination Address Low
<b>MY</b>	16-bit Network Address
<b>SH</b>	Serial Number High
<b>SL</b>	Serial Number Low
<b>PAN ID</b>	Operating PAN ID

A continuación se muestra un ejemplo de cómo se deben de configurar las dos módulos

XBee A Valores	XBee B Valores
<b>DH 13A200</b>	DH 13A200
<b>DL 4076E267</b>	DL 4076E26E
<b>MY AAAA</b>	MY AAAA
<b>PAN ID 234</b>	PAN ID 234
<b>SH 13A200 (viene por defecto)</b>	SH 13A200 (viene por defecto)
<b>SL 4076E26E (viene por defecto)</b>	SL 4076E267 (viene por defecto)

La ventaja de los módems [Xbee Serie 2](#), es que se pueden configurar redes más completas (tipo mesh). Esto nos brinda la ventaja de poder tener redes que abarquen zonas geográficas mucho más amplias que con la serie 1, aunque estas redes son mucho más complejas.

Acá se entrega una pequeña descripción de cómo configurar una red [Xbee Serie 2](#) con más de dos dispositivos:

#### **Punto a multipunto:**

Una conexión punto a multipunto es óptima cuando se requiere enviar información de manera controlada a uno o varios nodos a la misma vez, lo cual lo diferencia de una broadcast.

Solo basta con utilizar el método de una red punto a punto para la transmisión a un nodo, para realizar la transmisión a varios nodos a la vez solo basta con utilizar como dirección destino 0xFFFF.



Es importante tener en cuenta que se debe configurar el módulo Xbee en la misma dirección PAN y el mismo canal de comunicación: recuerde que la Xbee tiene 16 canales de comunicación.

### **Red Broadcast**

Esta red se diferencia de las demás en que todos los nodos tienen la dirección broadcast configurada lo que implica que los datos son recibidos hacia y por todos los nodos de la red. La dirección broadcast es DH=0x00000000 DL=0x0000FFFF.

### ***Lector RFID***

Para hacer uso del [Lector RFID](#) es necesario contar con el [RFID USB Reader](#), el cual hace mucho más fácil operar con el dispositivo. Si bien no es indispensable, sí es muy útil contar con él, dado que nos permite un acceso fácil a los pines además de incluir un buzzer que nos indica cuando se realiza cada lectura!

Al conectar el lector a la interfaz de comunicación USB podemos acceder al lector RFID por medio de una conexión serial: simplemente debemos identificar el puerto serie asociado a la interfaz rfid-usb y pasar la tarjeta por sobre el Sensor, esto enviará un String que corresponde al ID de la tarjeta leída.

Como NOTA debemos considerar que el lector RFID solo envía datos por la puerta serial, y no hace uso de la parte de recepción, por ende, si usamos una interfaz SoftwareSerial en arduino, podemos asignar solo el pin de Recepción (RX) y el pin de Transmisión lo podemos asignar el “pin 0” (no asignado), por ejemplo:

```
#include <SoftwareSerial.h>
#define pinrx 7
SoftwareSerial rfid(pinrx,0);
```

Recuerda modificar el pinrx por el que desees usar en el módulo SoftwareSerial.

### **Reemplazar el valor del ID para el lector RFID**

Para operar el lector RFID desde el PC, solo debemos conectar el [RFID USB Reader](#) con el [Lector RFID](#) insertado: esto instalará un puerto serie en nuestro equipo. Por medio de algún terminal serial (se recomienda [Putty](#)) podremos leer el ID de alguna tarjeta: simplemente debemos acercar una al lector y esto nos entregará su código por la interfaz serial. Debemos anotar este número para incluirlo en el código y lograr que la tarjeta elegida pueda activar un relé.

## ***Sensores Análogos***

Entre los sensores que escogimos para nuestro proyecto, tenemos:

- [Sensor Proximidad](#) (medidor de distancia)
- [Sensor de Temperatura ZX](#)
- [Sensor de Luz](#)

La ventaja de los sensores análogos es que entregan una salida de nivel de voltaje variable asociada a la medición que hacen, esto nos permite operarlos de manera muy transparente!

Para nuestro caso, podremos observar en el código que la sección que hace lectura de estos valores es bastante similar lo cual nos permite adaptar nuestro sistema a cualquier sensor que opere de manera similar sin tener que realizar cambios asociados.

## ***Sensores/Entradas Digitales***

Entre los sensores digitales que fueron elegidos, tenemos algunos que operan de manera un poco más diversa que el caso de los sensores análogos. Sin embargo, dado que tenemos una entrada optoacoplada, esto nos permite operar estos sensores de manera transparente.

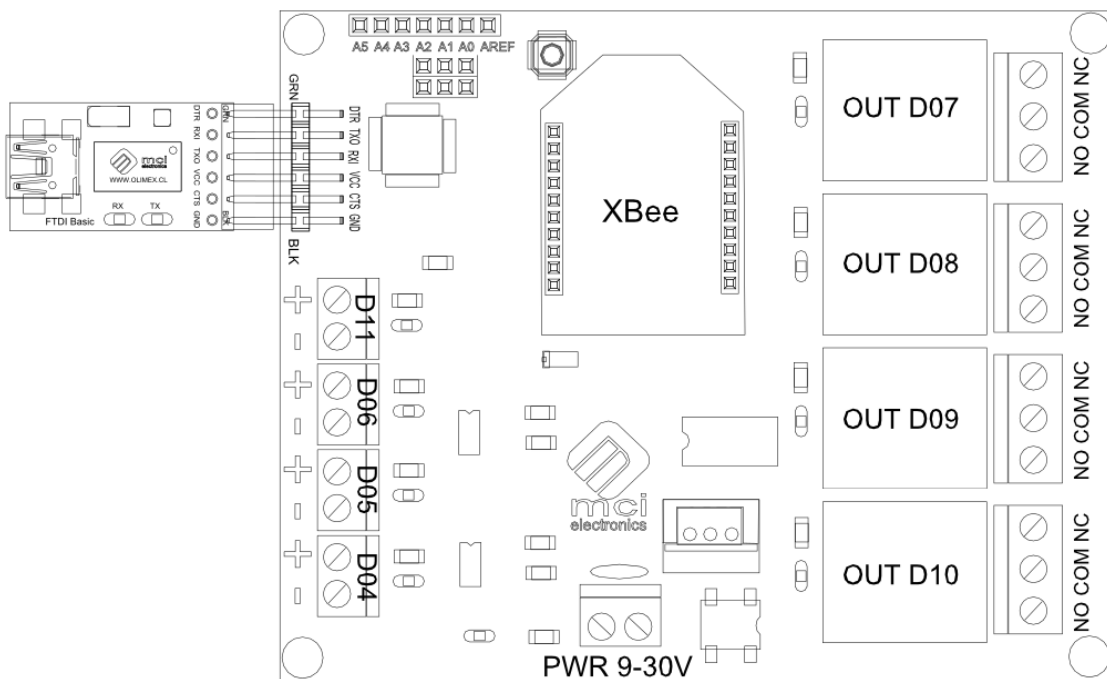
- [Contacto Magnético](#)
- [Botón Pulsador](#)
- [Sensor Presencia \(PIR\)](#)

Para cada sensor debemos configurar el circuito que lo alimenta, según podemos encontrar en la sección de conexionado. Con esto, tendremos todos nuestros sensores operando en forma de un “switch” para ser leído por las entradas de la tarjeta Xbee IO Pro.

Además, las entradas optoacopladas nos permiten operar cada sensor de manera independiente y sin tierras comunes, para voltajes que pueden llegar hasta 24 [VDC], con esto podemos evitar contaminación electromagnética entre los sensores o proveniente de la instalación misma del sensor que queremos controlar.

## Cómo programar el módulo Xbee IO Pro

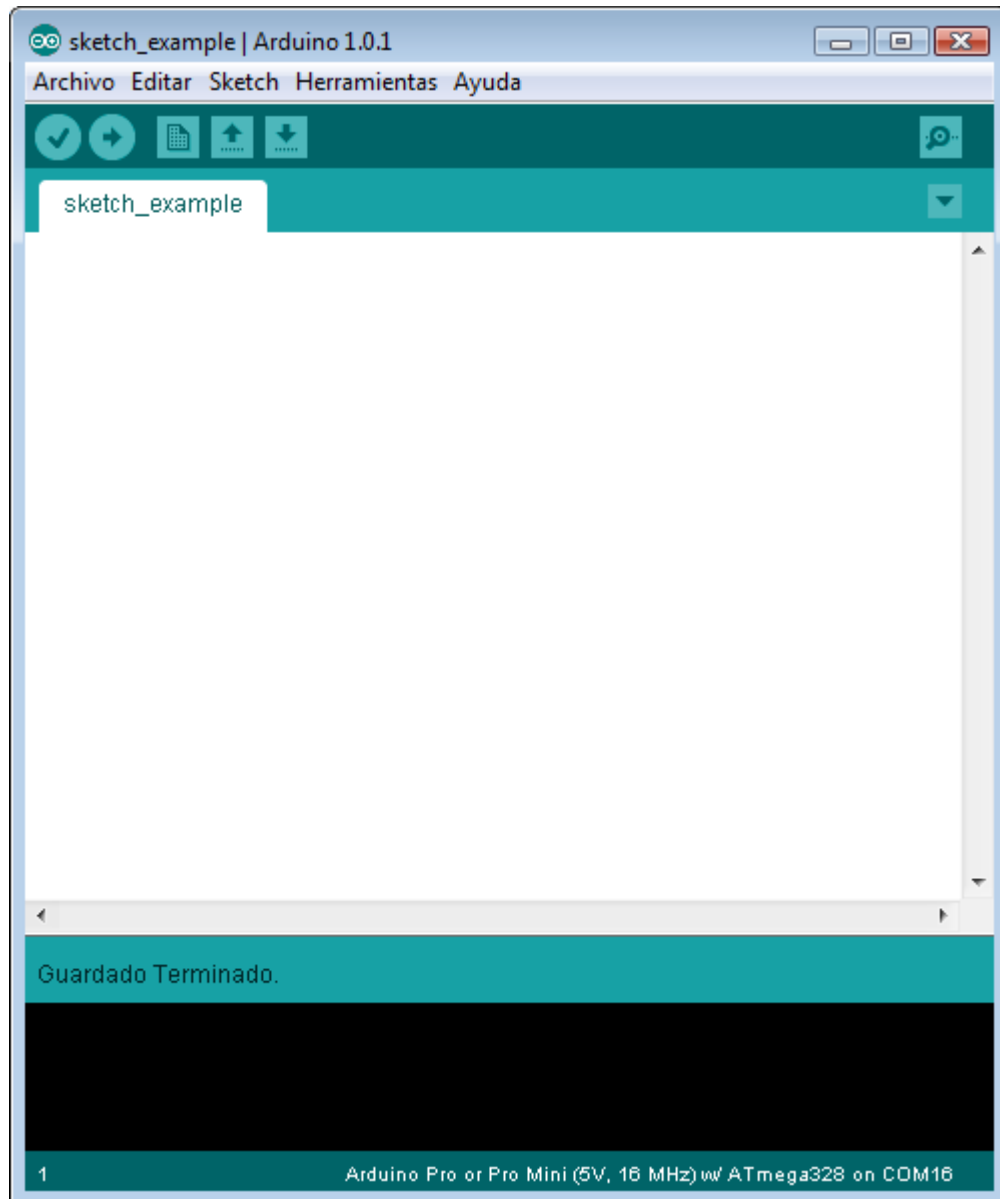
Para programar la nuestra plataforma debemos contar con el módulo FTDI conectado al PC, la tarjeta [Xbee IO Pro](#) y una fuente de alimentación para energizar la tarjeta. Esto último como se ha mencionado anteriormente, lo podemos solucionar con cualquier fuente DC o AC, de 9 a 30 [V] (dado que existe un puente rectificador en la entrada que nos permite adaptarnos a estas dos opciones).



**PARA PROGRAMAR LA TARJETA XBEE IO PRO SE DEBE CONECTAR UNA FUENTE DE ALIMENTACIÓN EXTERNA**

## ***Instalar plataforma Arduino***

La instalación de la plataforma Arduino es muy sencilla, simplemente debemos bajarla desde el sitio de Arduino ([acá](#)) y descomprimirla en una carpeta. Luego ejecuta “Arduino.exe” y ya está operativo!

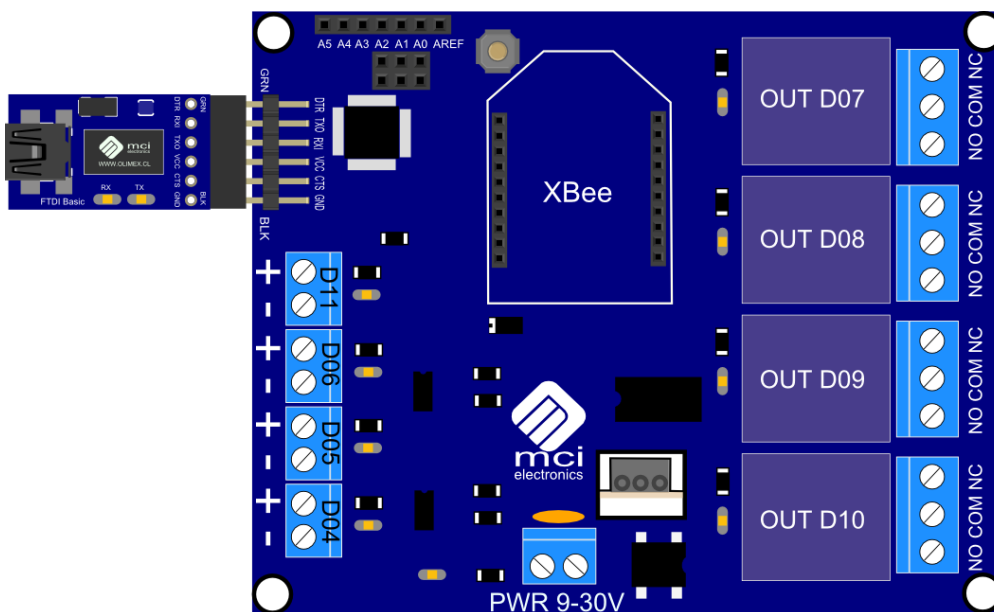


## ***Bajar el Código fuente***

El código fuente de nuestro ejemplo lo puedes obtener desde el sitio de la tarjeta [Xbee IO Pro](#). Debemos descomprimirlo en un lugar de nuestra elección para acceder a él en la siguiente sección.

## ***Verificar las llaves del FTDI***

Si observamos el módulo FTDI y el conector dispuesto para ello en la tarjeta [Xbee IO Pro](#), veremos que hay unos indicadores (o llaves) en los bordes del conector del módulo FTDI (conector hembra) con las etiquetas GRN y BLK. En el conector de la tarjeta [Xbee IO Pro](#) encontraremos los mismos indicadores, cuando enchufemos el módulo FTDI a la tarjeta debemos asegurarnos que se encuentren acoplados de manera que cada uno haga match con su par homónimo (BLK-BLK y GRN-GRN).



## ***Compilar y Cargar***

Lo primero, es identificar el modelo de la tarjeta o su compatibilidad, que para nuestro caso es compatible con “Arduino Pro” o “Pro Mini”, 5[V] / 16 [MHz].

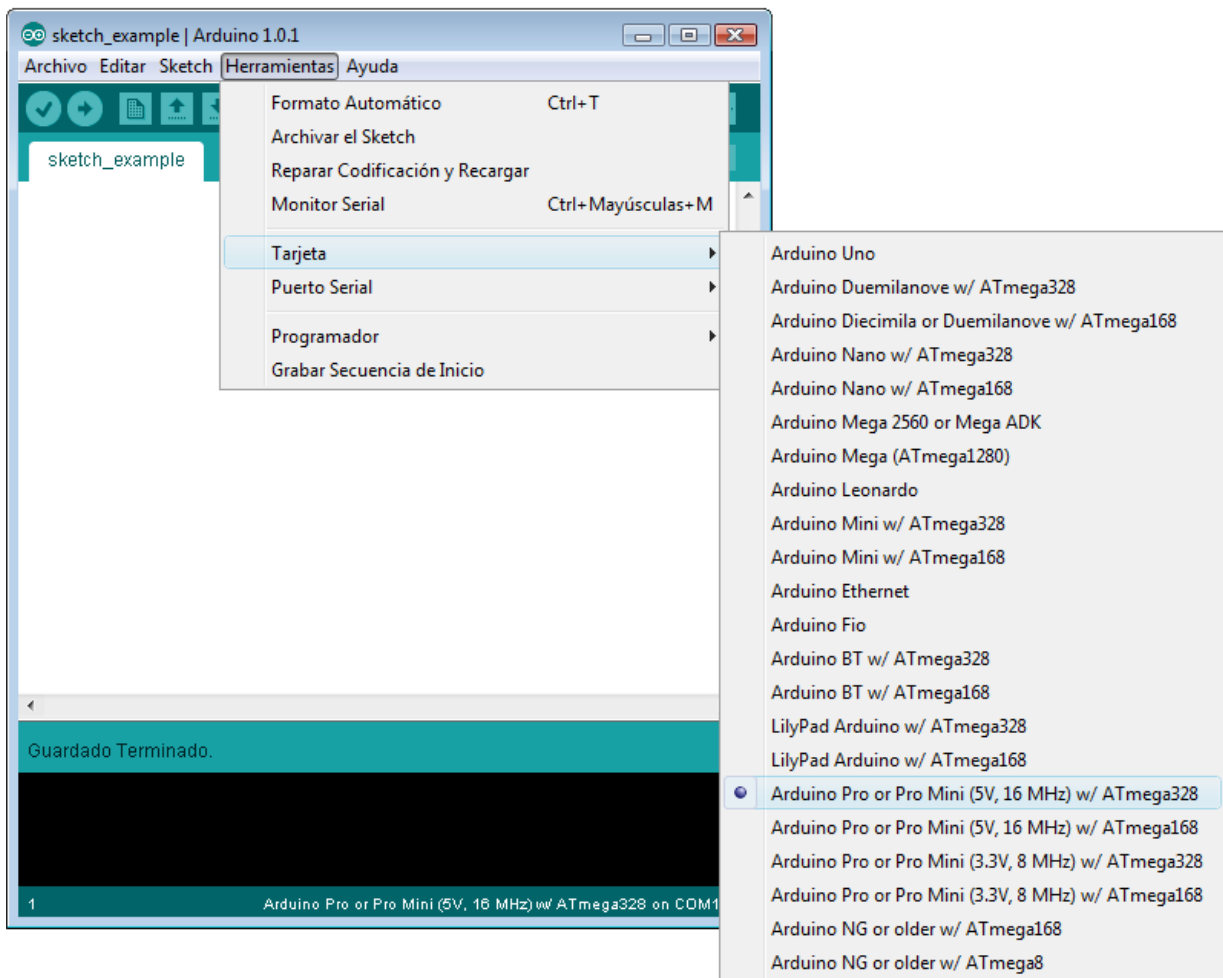
En el caso de Arduino el código fuente está escrito en lenguaje C++, pero de manera simplificada pues solo se deben escribir dos funciones principales - Setup() y loop() - y si lo deseamos, podemos definir algunas funciones auxiliares, estos códigos se llaman “Sketch”.

## MANUAL

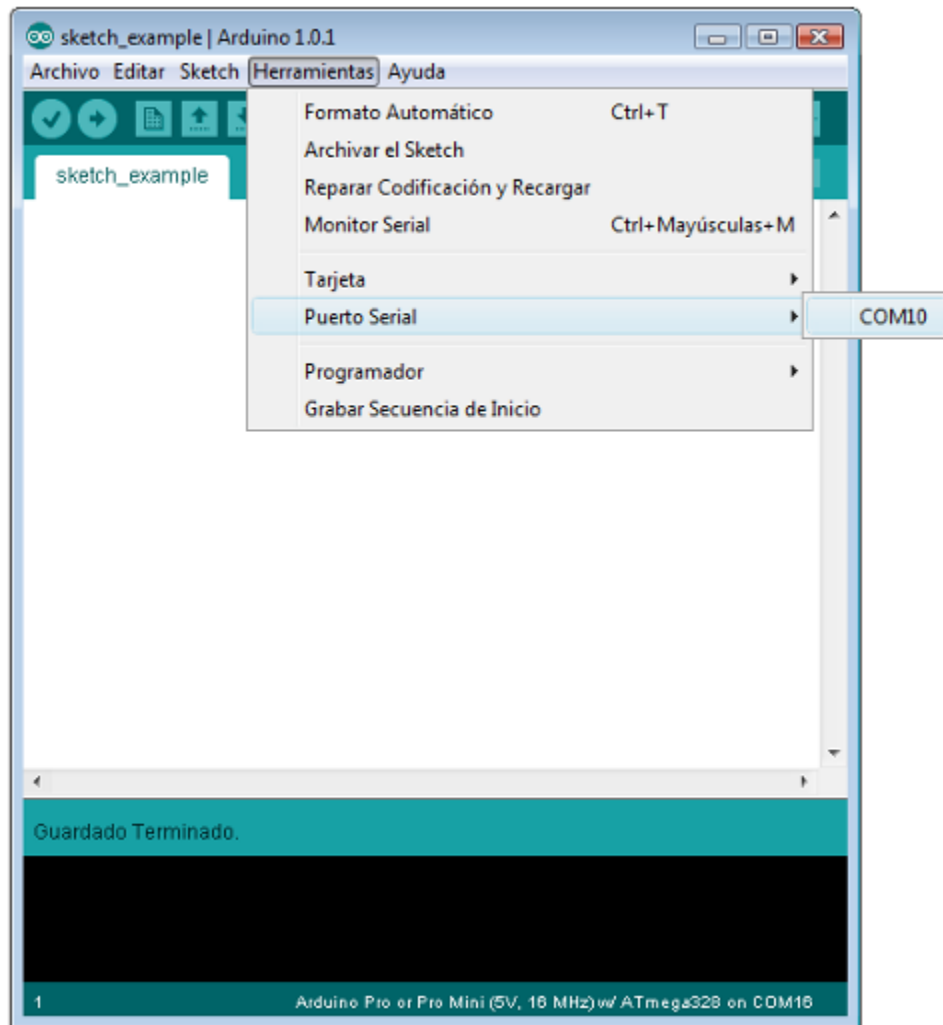
En la página de [Arduino](#) podemos encontrar mucha información de cómo utilizar la plataforma, además de una extensa documentación sobre el uso de las librerías que incorpora la plataforma y que la convierten en una de las más populares para desarrollo.

Para cargar el código en la tarjeta [Xbee IO Pro](#) debemos seguir el siguiente procedimiento (mismo proceso a seguir para cargar cualquier código).

Seleccionar el Microcontrolador compatible (Arduino Pro Mini 5v 16 MHz)

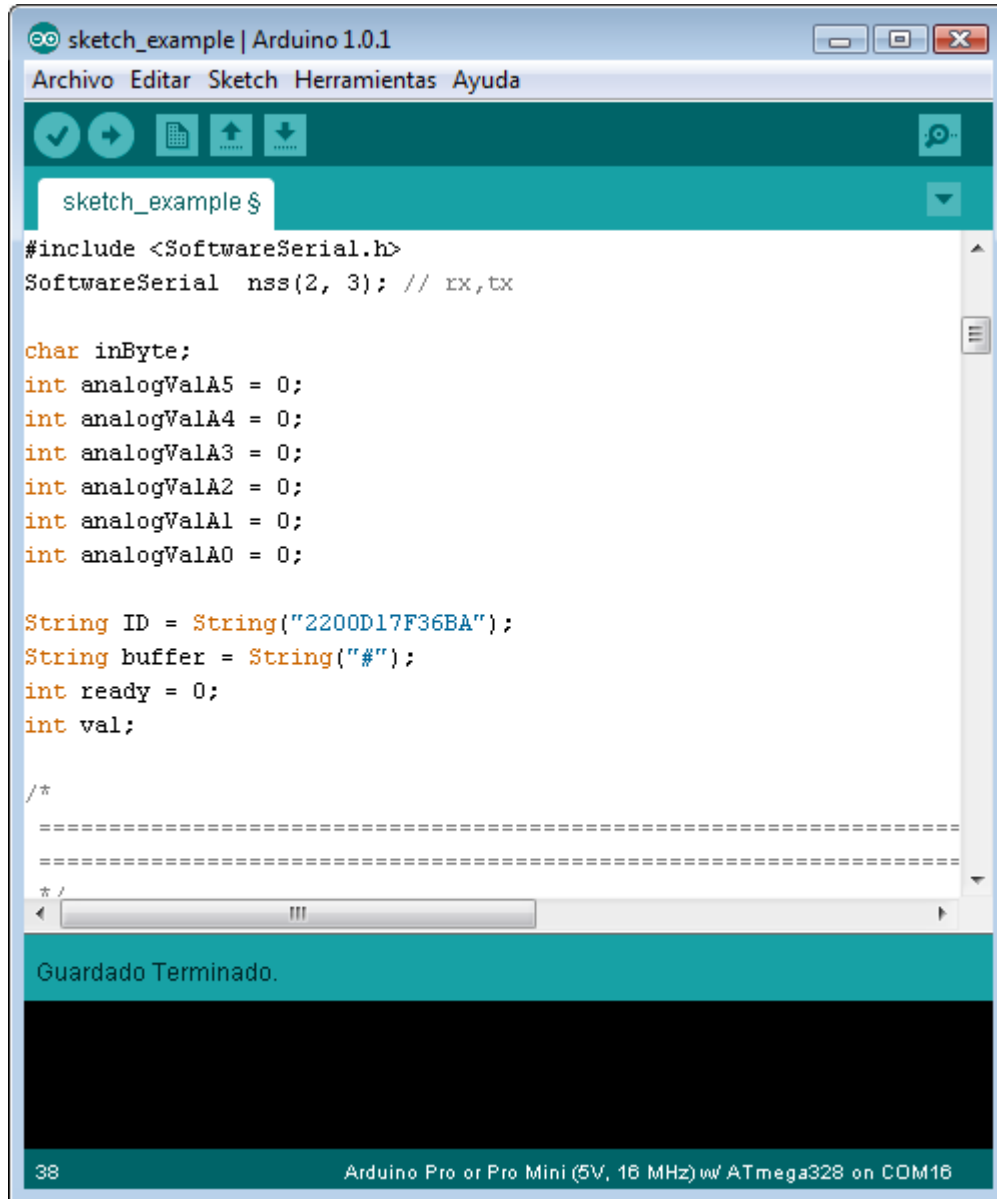


Seleccionar el puerto asignado al Módulo FTDI (esto varía según la propia instalación, el puerto indicado es solo referencial).



Se puede observar en la parte inferior de la ventana el modelo de tarjeta elegida y el puerto en el cual se encuentra conectado el módulo FTDI (esto es muy útil cuando estamos trabajando con diversos modelos de tarjetas).

Una vez hecho esto, abrimos el código que bajamos en la sección previa y deberá verse algo así:



```
sketch_example | Arduino 1.0.1
Archivo  Editor  Sketch  Herramientas  Ayuda

sketch_example $

#include <SoftwareSerial.h>
SoftwareSerial nss(2, 3); // rx,tx

char inByte;
int analogValA5 = 0;
int analogValA4 = 0;
int analogValA3 = 0;
int analogValA2 = 0;
int analogValA1 = 0;
int analogValA0 = 0;

String ID = String("2200D17F36BA");
String buffer = String("#");
int ready = 0;
int val;

/*
=====
=====
=====
*/
```

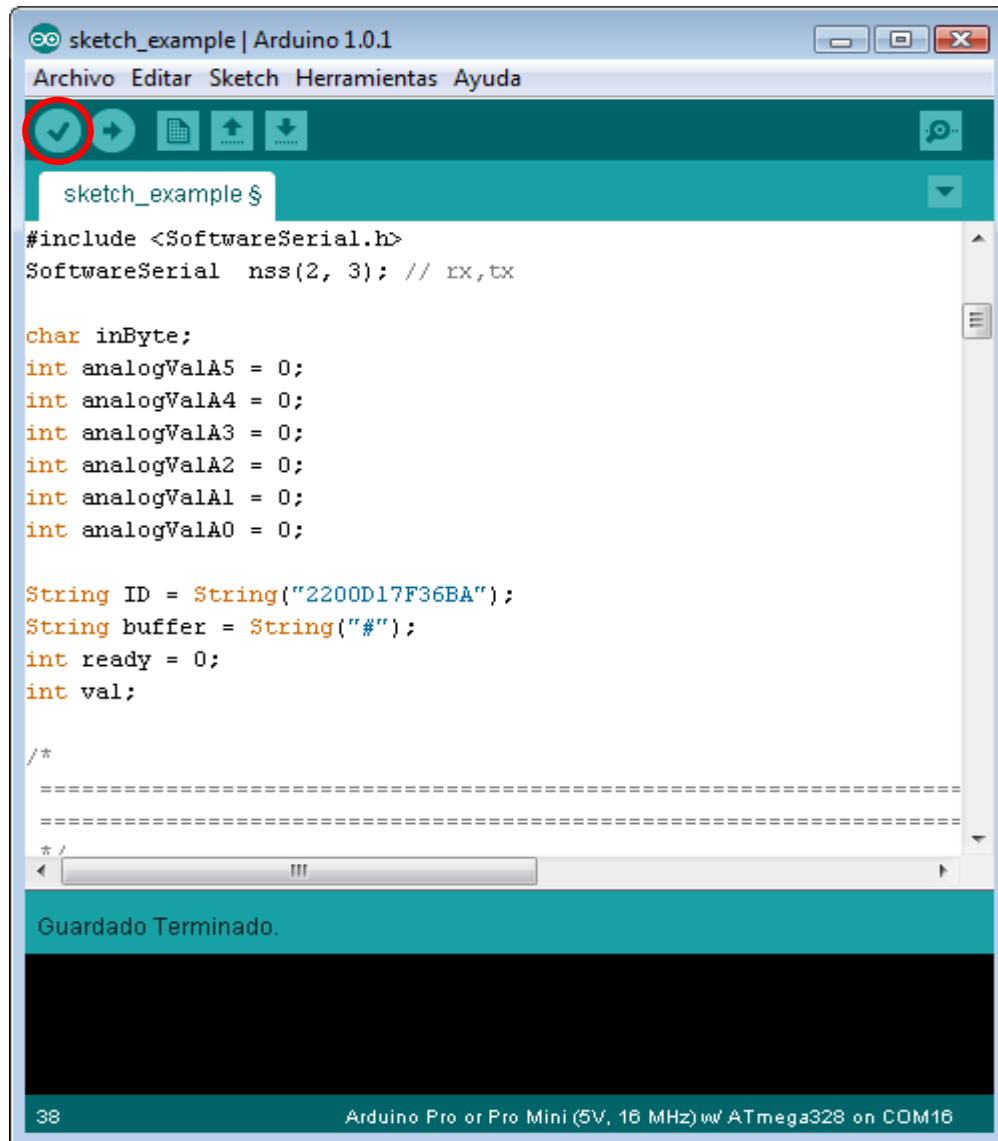
Guardado Terminado.

38 Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328 on COM16

Este es un buen momento para cambiar el ID leído desde nuestro lector de tarjetas RFID!



Una vez realizado lo anterior, debemos compilar nuestro proyecto: para ello presionamos el botón de la parte superior izquierda, marcado en rojo en la siguiente imagen.



Una vez que hayamos completado el proceso de compilación, debemos cargar nuestro proyecto a la tarjeta [Xbee IO Pro](#). Junto al botón compilador (al lado derecho) encontraremos el botón Cargar.

Si todo está ok, comprobamos que la asignación de las llaves del módulo FTDI estén correctamente ubicadas (BLK-BLK / GRN-GRN) y presionamos Cargar. Una vez terminado el proceso, el programa empezará a correr automáticamente.

## **Cómo operar las entradas/salidas por medio del PC (Vía Puerto Serie)**

### ***XBee Explorer USB***

Como ya se han configurado anteriormente los módulos Xbee, en este punto ya debemos estar familiarizados con el manejo del Xbee Explorer USB. Lo que haremos ahora será usar el Explorer como una interfaz o puerto Serie en forma directa, sin usar el programa XCTU. Para esto debemos identificar el puerto serie asociado a nuestro Xbee Explorer USB (Puertos COM en Windows y Puertos ttyUSB en Linux), una vez identificado procedemos a abrir el puerto con algún programa que nos permita manipular el puerto. Para simplicidad se recomienda usar [Putty](#) el cual está para variadas plataformas y protocolos.

Por defecto los valores a usar son 9600 baudios, para el puerto que nos corresponda.

¡Y listo! ya estamos conectados al modem Xbee y podemos enviar y recibir datos de otros nodos según sea la configuración de nuestra red. Si así lo queremos, podemos usar esta comunicación serial por medio de un programa en C, Java, .Net, o algún lenguaje scripting como Bash, Python o incluso por medio de una Página Web usando CGI (por ejemplo).

Una vez conectados podemos empezar a interactuar con la tarjeta, la interfaz de comunicación con la plataforma es bastante simple, para poder encender los relés en forma remota se debe enviar '1', '2', '3' y/o '4' para activarlos/desactivarlos, por cada dato bien recibido la tarjeta enviará un "OK" como respuesta.

Si se activa alguna de las entradas digitales, el sistema enviará el identificador asociado a esa entrada "D4", "D5", "D6" y/o "D11", los cuales serán recibidos en el PC

Si alguno de los sensores análogos supera el umbral asignado, se enviará un identificador con la entrada análoga asociada y su valor "A5-234", "A4-400", ..., "A0-101". Por defecto en el código solo está activada la entrada "A5".

Adicionalmente estos valores serán enviados al LCD y se actualizarán en cada lectura sucesiva.

El Lector RFID por su parte, opera en forma autónoma y nos permite controlar el encendido de un relé en forma directa usando la tarjeta Xbee IO PRO. Para su uso solamente debemos pasar la tarjeta por el lector RFID y si corresponde con el ID asignado previamente se activará el RELE4.

## Descripción del Protocolo de Comunicación

Acá encontramos la descripción del protocolo de comunicación implementado en el código y que debe estar operando en la tarjeta en estos momentos. Como podemos observar, la parte del RFID no está en el protocolo de comunicación, pues es controlado de manera autónoma por el microcontrolador!

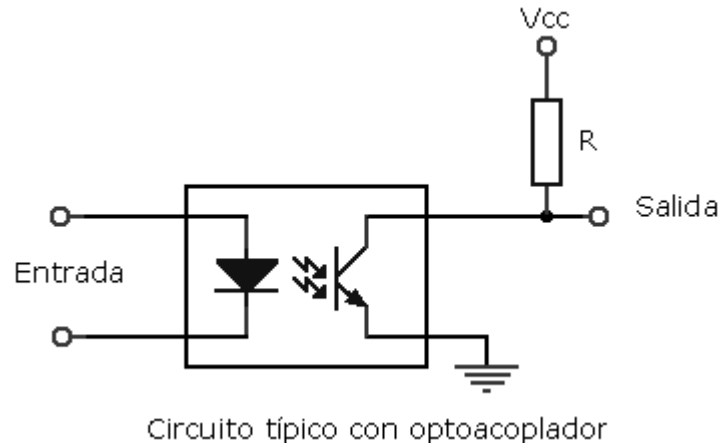
PC / USB Explorer		
Send	Receive	Action
1	Ok	Cambia estado Relé 1
2	Ok	Cambia estado Relé 2
3	Ok	Cambia estado Relé 3
4	Ok	Cambia estado Relé 4
-	D4	Se activó la entrada D4 en la Tarjeta Xbee IO Pro
-	D5	Se activó la entrada D4 en la Tarjeta Xbee IO Pro
-	D6	Se activó la entrada D4 en la Tarjeta Xbee IO Pro
-	D11	Se activó la entrada D4 en la Tarjeta Xbee IO Pro

Xbee IO Pro		
Receive	Send	Action
-	D4	Se activó la entrada Digital D4
-	D5	Se activó la entrada Digital D5
-	D6	Se activó la entrada Digital D6
-	D11	Se activó la entrada Digital D11
1	Ok	Cambia estado Relé 1
2	Ok	Cambia estado Relé 2
3	Ok	Cambia estado Relé 3
4	Ok	Cambia estado Relé 4

Y algo importante: TODAS ESTAS FUNCIONES PUEDEN OPERAR EN FORMA CONJUNTA!!!!!!

### ***Extra 1: Uso de los Optoacopladores***

Un optoacoplador es un aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor mediante la luz emitida por un diodo LED que satura un fototransistor. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotorreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado y se suelen utilizar para aislar eléctricamente dos dispositivos o para proteger uno de ellos (generalmente se protege el microcontrolador). Además, como funciona de aislador óptico, podemos no conectar la tierra (GND) del circuito que alimenta al led con la tierra (GND) de la tarjeta, consiguiendo con esto un funcionamiento completamente independiente.



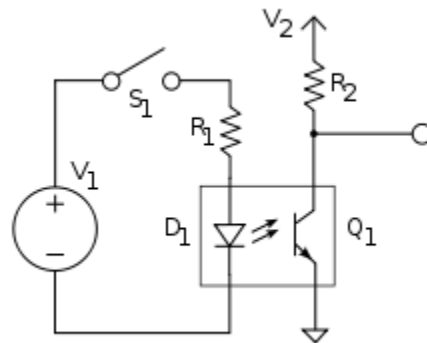
**LOS OPTOACOPLOADORES SON ELEMENTOS ACTIVOS Y PARA SU OPERACIÓN DEBEMOS ENERGIZAR EL LADO DEL LED CON UN MAXIMO DE 24 [VDC].**

En nuestro caso, para que las entradas optoacopladas funcionen correctamente debemos proveer la energía necesaria para estimular el diodo de entrada que realizará la activación de nuestra salida.

En general, podemos usar la salida de los sensores para activar el led y con esto notificar a la plataforma Arduino que la entrada ha sido activada.

## MANUAL

Como podemos observar, al activar el circuito del led (S1 cerrado) el transistor interno del optoacoplador quedará conduciendo, y esto hará que la salida quede en CERO lógico. Por el contrario, cuando el circuito del led esta desactivado (S1 abierto) el transistor se encontrará cortado y la salida del optoacoplador estará en UNO lógico.



Lo anterior explica por qué el manejo del optoacoplador debe hacerse considerando lógica negada, lo cual es manejable por software como podemos observar en los códigos de ejemplo.

## ***Extra 2: Programar los módulos Xbee sin un Xbee Explorer USB***

Si no contamos con un [Xbee Explorer USB](#), existe una forma de programar los módulos Xbee Serie1 o Serie2, usando solamente la tarjeta [Xbee IO Pro](#) y el programa XCTU que hemos mencionado anteriormente.

Para ello debemos cargar el código que encontraremos en la página de la tarjeta [Xbee IO Pro](#) en la sección de códigos (Xbee\_IO\_PRO\_Explorer.zip), para lo cual podemos seguir el mismo procedimiento que describimos anteriormente para cargar un código en la tarjeta. Se recomienda usar una velocidad de conexión de "9600 baudios", por lo cual debemos revisar que el código sea consistente con este valor. Se recomienda esta velocidad para disminuir la posibilidad de errores producto de la transferencia de datos.

Una vez que hayamos cargado el código, usamos el programa XCTU y asignamos el Puerto que corresponda al módulo FTDI para la comunicación, también a '9600 baudios'. Podemos seguir el procedimiento que se describe en este mismo documento para programar los módulos Xbee.

**ES IMPORTANTE RESALTAR QUE ESTE ES UN MÉTODO ALTERNATIVO, Y QUE SIEMPRE ES RECOMENDABLE USAR UN XBEE EXPLORER USB**