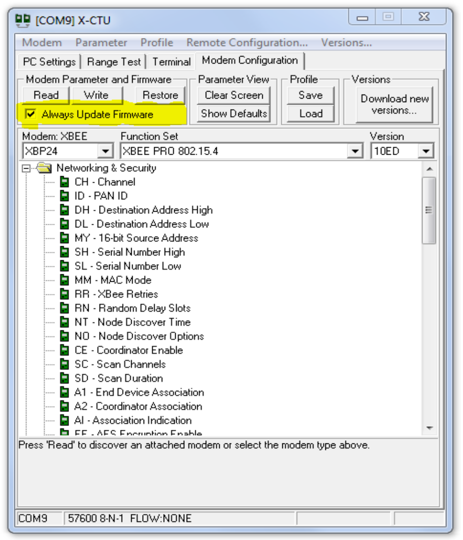
## 3. XBee radio setup - continued

The XBee's out-of-the box are already configured with a default set of parameters and firmware. There are minimal changes required to get the transmitter and receiver parameterized to work. This example configures the two devices as point to point END devices without the need for a COORDINATOR unit. This is OK for a pair of radios but if you have a network of radios (i.e. more than 2) then this setup will require an alternative setup. At this stage it is best to mark the two XBees - one as a TX and one as RX so that you can easily identify them during swap out. Below is a table showing the parameters that require changing from the default configuration.

| Parameter | FIO Remote side RX | Computer side TX |
|---|---|---|
| BD | 6 (57600) | 6 (57600) |
| ID | user's preference (same as computer side) | user's preference (0000 to FFFF) |
| MY | user's preference (e.g 0001) | user's preference (e.g 0000) |
| DL | Computer side MY value | FFFF or Fio's MY value |
| D3 | 5 | 3 |
| IC | Not set | 8 |
| RR | Not set | 3 |
| IU | 0 | Not set |
| IA | FFFF | Not set |
| R0 | 10 | 10 |
| | | |

It is assumed that the reader is familiar with the X-CTU software and is capable of establishing a connection to XBees such that the codes can be read by uploaded to the package. The parameterizing of XBees is the same irrespective of which version of XCTU is being used. (i.e. version 5 or 6). For the purpose of ease screenshots relate to Version 5 of XCTU.
It is also a good idea to update the firmware on the XBees to the latest version. This is achieved by ticking the check box in the modem configuration screen. At the time of writing the latest firmware version for the XB24 PRO is 10ED. Now, select your Tx XBee, insert into a USB breakout, select the correct Comm port in the XCTU application and default baud rate of 9600. Test to confirm correct communication by trying the Test/Query function. This should report a confirmation screen if all is OK. Tab to the Modem Configuration screen and select your type of XBee from the drop down list. Select a corresponding Function Set and firmware version. In my case for the XBee24 Pro, the type is XBP24, XBEE PRO 802.15.4 and 10ED. The screen should be populated with an empty parameter list. At this stage select "Show Defaults". This will populate the parameters with default values. Change the deault parameters so that they correspond to the valus shown in the table on the left. For the Tx, choose the list in the right hand column under "Computer side TX". Now flash this parameter set to the XBee by selecting "Write". The parameters and firmware will uploaded to the radio.

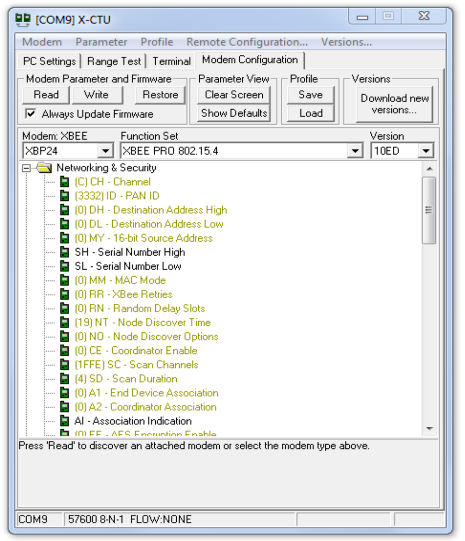

[COM9] X-CTU
Modem Parameter Profile Remote Configuration... Versions...
PC Settings | Range Test | Terminal | Modem Configuration
Modem Parameter and Firmware
Read Write Restore
Parameter View
Clear Screen Show Defaults
Profile
Save Load
Versions
Download new versions...

☑ Always Update Firmware

Modem: XBEE: XBP24
Function Set: XBEE PRO 802.15.4
Version: 10ED

```
Networking & Security
    CH - Channel
    ID - PAN ID
    DH - Destination Address High
    DL - Destination Address Low
    MY - 16-bit Source Address
    SH - Serial Number High
    SL - Serial Number Low
    MM - MAC Mode
    RR - XBee Retries
    RN - Random Delay Slots
    NT - Node Discover Time
    NO - Node Discover Options
    CE - Coordinator Enable
    SC - Scan Channels
    SD - Scan Duration
    A1 - End Device Association
    A2 - Coordinator Association
    AI - Association Indication
    EE - AES Encryption Enable
Press 'Read' to discover an attached modem or select the modem type above.
COM9   57600 8-N-1  FLOW:NONE
```

XCTU empty parameter screen


[COM9] X-CTU
Modem Parameter Profile Remote Configuration... Versions...
PC Settings | Range Test | Terminal | Modem Configuration
Modem Parameter and Firmware
Read Write Restore
Parameter View
Clear Screen Show Defaults
Profile
Save Load
Versions
Download new versions...

☑ Always Update Firmware

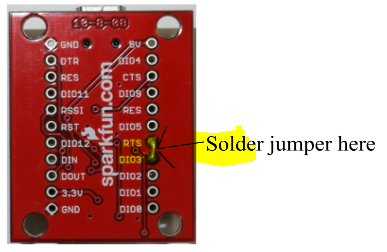
Modem: XBEE: XBP24
Function Set: XBEE PRO 802.15.4
Version: 10ED

```
Networking & Security
    (C) CH - Channel
    (3332) ID - PAN ID
    (0) DH - Destination Address High
    (0) DL - Destination Address Low
    (0) MY - 16-bit Source Address
    SH - Serial Number High
    SL - Serial Number Low
    (0) MM - MAC Mode
    (0) RR - XBee Retries
    (0) RN - Random Delay Slots
    (19) NT - Node Discover Time
    (0) NO - Node Discover Options
    (0) CE - Coordinator Enable
    (1FFE) SC - Scan Channels
    (4) SD - Scan Duration
    (0) A1 - End Device Association
    (0) A2 - Coordinator Association
    AI - Association Indication
    (0) EE - AES Encryption Enable
Press 'Read' to discover an attached modem or select the modem type above.
COM9   57600 8-N-1  FLOW:NONE
```

XCTU default parameter screen

The ID (Pan ID) can be left at the default of 3332. This will be the same ID for both the Tx and Rx radios.
Now, repeat the exercise with the RX radio plugged into the breakout board. This time, replace the default parameters with those shown in the left hand column of the table (FIO Remote Side RX).
Its worth double checking these parameters by reading back each XBee after you have uploaded to confirm that the parameters have been flashed correctly.

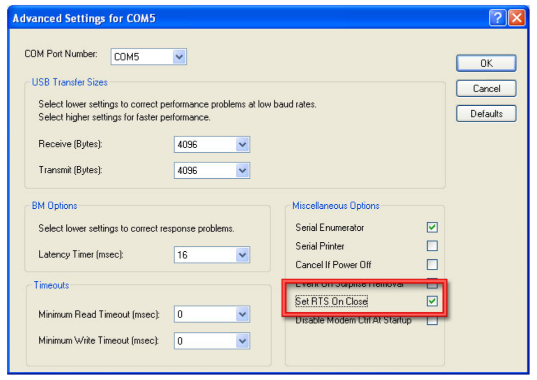
At this point you should remove the RX XBee and insert it into the Arduino FIO receptacle.
The TX Xbee needs a small modification to the breakout board to assert the RTS on the XBee.
A small jumper needs inserting between pins 16 (RTS) and 17 (DI03). See photo.
After the modification, insert the TX Xbee into the board and connect the board to an available USB slot on your PC/Laptop.



*Modified XBee Explorer USB*

## 4. Computer/ Laptop settings

For correct synchronising, there needs to be a configuration change if you are running Windows. It is necessary to change the properties of the USB port you intend using for the communications.
This is done via the Windows Device Manager. In Device Manager, navigate to "Ports (Comm & LPT)".
Select the port that you will be using for your XBee. Double click to open the port properies. Tab to Port Settings and choose the Advanced tab. Now check the box labelled "Set RTS On Close" as shown below.



COM Port Number: COM5

Select lower settings to correct performance problems at low baud rates.
Select higher settings for faster performance.

Receive (Bytes): 4096
Transmit (Bytes): 4096

Select lower settings to correct response problems.

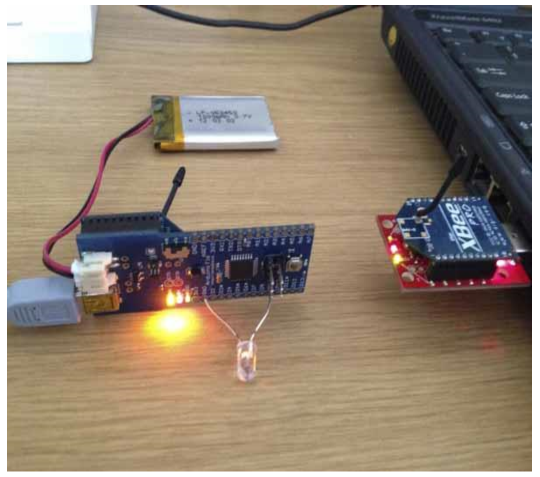
Latency Timer (msec): 16

Minimum Read Timeout (msec): 0
Minimum Write Timeout (msec): 0

Serial Enumerator ☑
Serial Printer ☐
Cancel If Power Off ☐
Event On Surprise Removal ☐
Set RTS On Close ☑
Disable Modem Ctrl At Startup ☐

The serial port is now initialised and ready. Insert the TX XBee and confirm that the board is active. This can be seen as brief activity on the Tx and Rx LEDs on the breakout board. Insert the RX XBee into the FIo if not already done. Connect the FIO to a mini USB power supply or battery. Make sure the ON/OFF slider switch on the FIO is in the ON position. Confirm that the board is opeartional via the ON LED.
At this point the radios are ready to start transmitting and receiving. The next step is to load an example sketch up using the Arduino IDE on the computer/laptop. For this exercise, use the basic BLINK sketch. Load this sketch up, select the FIO from the Boards drop down menu under Tools menu. Select the correct port under the Serial Port tab. Now, choose 'Arduino as ISP' as the programmer.

Now is a good time to compile the BLINK sketch to make sure the IDE is ready to start the upload.



If the upload is clicked, a couple of things are going to happen. The Arduino compiler will start to recompile and the transmitting XBee will spring into life. The RSSI LED will illuminate and the TX and RX LEDS on the breakout board will show comms activity.
On the receiving end, the RSSI LED on the FIO will illuminate. If all is well, within a few seconds the onborad LED (LED13) on the FIO will start blinking.
In my own example, I modified the sketch to use digital pin 9 and used a 5mm LED as shown in the photo.

The other (yellow) LED that is illuminated is the charge indication for the battery pack.

If the uplaod was unsuccessful (indicated by the ubiquitous "stk500_getsync() error in the Arduino IDE), recheck the XBees configuration, baud rates and make sure the TX and RX radios are used in their correct places.
Also, the small jumper on the TX breakout board - did you remeber to put it in?

This concludes the wireless programming exercise. It really comes into its own if a remote FIO is used in an inaccessible place and you need to modify your sketch without having access. Its also a great introduction to XBees and wireless radios.
In the next feature we will put the radios to good use by implementing a MODBUS communication link wirelessly. Modbus is an industrial communications protocol widely used in machine and automation control systems.

## Software Required for this system setup:

1. X-CTU ver. 6.1.0 available from http://www.digi.com

2. Arduino IDE available from http://www.arduino.cc

3. The "Blink.ino" Arduino example file. (Included with IDE)

4. The Rx and Tx XBee config files (*.pro), included in the GITHUB repository...
https://github.com/aetcnc/Arduino_FIO_XBee_Programming)

| | |
|---|---|
| Arduino FIO Wireless programming<br>Prepared By: AC Date: 2412/2013 | Affordable•Engineering•Technologies |
| Arduino Fio Wireless Programming - Sheet 2/2 | |