

# The Unofficial XBee FAQ (version 1.2)

johnf

March 14, 2011

## 1. What's the difference between Series 1 (Freescale) and Series 2 (Ember) XBees, and how can I tell what I've got?

The two types use different hardware, and each has a different set of firmware options. You can't use firmware intended for one type on a module of the other type.

To tell what sort of chip you have, look on the underside of the module for the part number, which begins with either XB24- or XBP24-.

If the part number begins with XBP24- then it's the Pro version. If the part number continues as XB24-A or XBP24-A then it's a Freescale (Series 1) module that originally shipped with 802.15.4 firmware. If the part number continues as XB24-DM or XBP24-DM then it's a Freescale module that originally shipped with DigiMesh 2.4 firmware. If the part number continues as XB24-B or XBP24-B then it's an Ember (Series 2) module.

Also note: if the part number (XB[P]24...) ends with a J then you have the international version of the chip, with a reduced RF power output.

If you have a Freescale XBee you can load it with either the 802.15.4 firmware or the DigiMesh firmware. Looking back at earlier posts in the forum will reveal that there used to be a further option to load Zigbee firmware. That has been discontinued: see the knowledge base article "XBee/XBee-PRO Series 1 ZigBee Module Firmware Revision History" at <http://www.digi.com/support/kbase/kbaseresultdet1.jsp?id=2182>

Some earlier modules can't run the DigiMesh firmware. You can load it onto them, but it won't function correctly. For details, see the knowledge base article "Series 1 XBee / XBee-PRO 802.15.4 and DigiMesh 2.4 Compatibility Factsheet" at <http://www.digi.com/support/kbase/kbaseresultdet1.jsp?id=3150>

The 802.15.4 firmware is simple to use, but it does not provide routing capabilities. Therefore, all communication in a network must take place directly between the sending and receiving node, which puts a limit on the physical size of the network. If you're just starting out with XBees and want to play around with them to get experience you might consider 802.15.4 to be a good starting point.

The DigiMesh firmware includes routing capabilities, but at the moment it lacks some of the functions that are present on 802.15.4. (Note: that comment applies to version 8003. There is a now a newer version out, which means this FAQ needs updating.)

If you have an Ember XBee you can load it with either ZNet 2.5 or ZB (Zigbee) firmware. The ZB firmware is a later development than the ZNet 2.5, and it uses version 3.x.x of the EmberZNet stack (the firmware component that controls radio transmission), whereas the ZNet 2.5 uses version 2.5.x. Digi continue to support both versions, but for new development you'll probably want to go with ZB. Both versions allow for routing, and both versions work with a variety of Digi sensors and other devices.

Whichever type of firmware you use (802.15.4, DigiMesh, ZNet 2.5 or ZB), all nodes in a network must use the same type. You can't mix-and-match. It's also a good idea to make sure that all nodes are running the same version of the firmware.

For a more detailed discussion on the differences between Freescale (series 1) and Ember (series 2), try this knowledge base article: "The Major Differences in the XBee Series 1 vs. the XBee Series 2" at <http://www.digi.com/support/kbase/kbaseresultdet1.jsp?id=2213>

## 2. Can I create a network with a mixture of Freescale modules and Ember modules?

No. Sorry, but the Freescale and Ember devices are not compatible with each other.

### **3. I can set XBee parameters with AT or API commands but after a power cycle the settings are lost. This doesn't happen with X-CTU. What's wrong?**

If you want settings to be preserved, you have to write them to non-volatile memory with the WR command. When you use X-CTU's menu options it does this for you automatically.

### **4. I'm trying to send a perfectly good API packet, but I get either no response or an error status. Is this a bug?**

Well, it could be a bug but it probably isn't. If you're composing API packets by hand, it can be an error-prone process. And if you've written a program to compose them, it's always possible that an obscure bug is still in there.

There is a program called packet-check, which is a Tcl script contributed by johnf . It's a command line program that lets you enter the hex codes for a packet, and then decodes the packet checking for errors. You'll find the program if you look in the pinned posts in the 802.15.4 forum.

You don't have to use the packet-check program. If you post the packet and its response in the forum, with luck someone will contribute a suggestion.

One thing that has bitten people in the past is that when writing code to generate API packets, it's a bad idea to use signed 8-bit variables to hold the packet bytes. The outcome is the appearance of spurious FF bytes in the generated packet, and when that happens nothing works. This comment applies particularly to C/C++ and Java. In C or C++ you can use 8-bit variables if you declare them as `unsigned char` . In Java there are no unsigned data types, so the best solution is usually to use the `short` data type (16 bits).

### **5. I'm sending data between XBees in 50-byte (or whatever) chunks, and I've noticed that each chunk is being sent as a number of small RF packets. Can I fix this?**

The XBee will transmit a packet either when its buffer is full enough or when a timeout value between bytes received is exceeded. The timeout is controlled by the RO parameter and it defaults to quite a low value (three character periods). What's probably happening is that the original data stream is being sent with occasional short pauses, and these pauses are triggering the sending of the packet.

So experiment with higher values of the RO parameter, and see whether it helps.

### **6. I'm designing a circuit board for my XBee. What serial lines need to be connected? Any other tips?**

For normal use, including the setting of parameters, you need connections only for VCC, GND, DOUT and DIN. If you want to be able to update the firmware on the XBee while it's still in the board, you also need RTS and DTR. For hardware flow control you also need RTS and CTS.

The DTR line is connected to the sleep pin. It is held low throughout the reprogramming process, so you could do that with a switch on the board. Alternatively, if you will not be using the module in sleep mode, you could tie the sleep pin permanently to ground.

A wise precaution if you want the ability to update the firmware is to make sure you provide a reset switch or equivalent function on the board. See question 11 for the reason.

### **7. I've built a circuit board for my XBee but the chip doesn't work properly. Any ideas?**

It could be any number of things, but a good place to start is to check the power supply you're using. If there's too much ripple on it the XBee will have problems. Maybe try a larger capacitor on the regulator input or output. I tend to use 47 $\mu$ F at the regulator input and 2.2 $\mu$ F at the output, but your mileage may vary. The manufacturer's data sheet for the regulator will probably give you some suggested values.

### **8. I'm writing software to control my XBees. Is there any sample code out there that I can use?**

There aren't many people who have posted sample code, but a notable exception is at <http://code.google.com/p/xbec-api/> where there is a very impressive Java implementation of an API to the XBees. It covers both Series 1 and Series 2 types, and there's quite a bit of useful general XBee information there too.

Do you know of any other such resources? If so please tell me and I'll add them to this FAQ.

## **9. At 115,200 baud, data transmission becomes unreliable. What can I do?**

This is a problem involving the serial link, and the short answer is to configure the host PC to use two stop bits.

There is a much longer answer, in a topic entitled "Using the XBee at 115,200 Baud" by johnf . It was posted in the 802.15.4 forum. The long answer goes into the detail of how baud rates are derived from clock speeds. If your XBee is connected to an ordinary PC, you probably don't need to read it: just use the short answer. If you have a more exotic setup, or you're simply interested in the background information, you may find the long answer useful.

In a nutshell, the problem arises because of the way UARTs work. The UART is driven by a clock signal whose frequency is 16 times the desired baud rate. In turn, this clock is obtained by dividing the system clock by an integral value. If the system clock is an exact multiple of (16 times the baud rate) then a completely accurate baud rate is obtained, and this will be the case on a PC host. If the system clock is not an exact multiple, the baud rate obtained will be inaccurate to some degree. The XBee, along with many microcontroller circuits, uses a 16MHz clock. At 115,200 baud, there is no possible value for the integral divisor that will give a baud rate close enough for reliable operation when connected to a PC unless the two-stop-bit solution is also employed.

## **10. X-CTU says it can't communicate with my module. What can I do?**

This can happen if you've set the module into a sleep mode with no way of waking it. Other parameter setting problems may also be causes.

Whatever the cause, using X-CTU to reflash the firmware should restore the module to life.

## **11. I want to update the firmware on my XBee but X-CTU is giving error messages. How can I recover from this?**

Here are a few ideas that people have in the past found helpful.

1. Are you using the latest version of X-CTU? If not, download and install the latest version from the Digi site.
2. Check that the version of X-CTU you're running really is the version you think it is (use the About... menu option). There has been one report that after uninstalling an old version it was necessary to delete the program folder and reboot before installing the new version. Without the folder deletion and the reboot the new version was installed, but the old one was still there and was the one that ran.
3. In X-CTU go to PC Settings and make sure it's set for 9600 8N1 with hardware flow control. Also make sure that the Enable API box is NOT checked.
4. (Note: this one is a lightly edited quote from Digi Support.) Look at the bottom of the RF module and see which model you have. Then in X-CTU select the correct modem model and click show defaults. You should click on Download New Versions so you have the latest firmware version for your RF modem, and also make sure you have Update Firmware checked.
5. If you can communicate with the module from X-CTU or from a terminal program, try issuing the ATRE command to reset all parameters to their default values. Then if you're using a terminal or the X-CTU terminal window, issue ATWR to save the default values to non-volatile memory. Then try again with the reprogramming.
6. If you get an error message from X-CTU when you click the Write button, then hold down the Reset switch on the development board while you click the Write button again. Keep the Reset switch down for another second, then release it.
7. If that doesn't do the trick, try the Write command again and press the Reset switch when one of the LEDs on the development board goes solid.
8. Still no luck? Try this suggestion from Digi:
  - (a) Download and install latest X-CTU from the downloads section.
  - (b) Power off your module.

- (c) Put reset to ground, or activate reset switch in your test board.
  - (d) While powered off (yes!) start the update on X-CTU and wait until it complains that a reset is needed.
  - (e) Power on your board, and after a few seconds, release reset.
  - (f) Update should start.
9. And here's a recipe that was posted in the Zigbee Pro forum by NATP who says it has worked for him:
- (a) Using the USB board (XBIB-U-Dev rev 3), X-CTU 5.2.5.0 with Windows XP, SP3
  - (b) Close X-CTU, disconnect the board
  - (c) Power on the board
  - (d) Open X-CTU
  - (e) On the PC-Settings tab:
    - Set Baud Rate to 38400
    - Select "No baud Change"
  - (f) Click Test/Query - this will fail with the "action required" dialog... - close it when it comes up (do not follow the instructions on the dialog)
  - (g) Click "OK" to the "cannot find modem" dialog
  - (h) On the PC-Settings tab:
    - Set Baud Rate to 9600
    - Un-select "No baud Change"
  - (i) Click Test/Query - Success!
  - (j) On Modem Configuration Tab – click "read" – Success!
10. Still need more? Here's one posted by shahrj1988 in response to a request for help. The original poster replied that it had worked.
- (a) Make sure the XBee module is NOT plugged into the interface board.
  - (b) Connect the interface board to the computer via appropriate cable (USB or Serial) (If using Serial, connect power as well).
  - (c) Open the X-CTU program and select the proper COM port.
  - (d) The first screen you see contains the PC Settings tab. On this tab please change the baud rate to 38400 and put a check in the 'No Baud Change' check box.
  - (e) Navigate to the Modem Configuration tab.
  - (f) Select the proper modem type, function set and firmware version from the 3 drop down menus located about 1/3 of the way from the top of the window.
  - (g) Click on the Show Defaults button and put a check in the 'Always update firmware' check box.
  - (h) Click on Write. (Yes, the module is NOT plugged in) An Action Required box will appear. This is good. When this happens, hold in the config/reset switch on the interface board and CAREFULLY plug in the module and then release the config/reset button.
  - (i) You should now see that box disappear and a progress bar will appear along the bottom of the window. You will see the bar progress and then the box will appear again. This is good. Just hold in the reset switch on the interface board for about 1-2 seconds and then release. The box will disappear and the status at the bottom of the window will show that the write was complete.
  - (j) Go back to the PC Settings tab and change the baud rate back to 9600 and uncheck the 'No baud change' check box.
  - (k) Click the 'Test/Query' button. You should get a clean bill of health.
11. If none of the above worked, you can either post the problem to the forum or send a support request to Digi. When you do find a method that works for you, it would be a kindness if you would post it to the forum for the benefit of others.

## **12. I think I just killed my XBee. How can I tell whether it's really dead?**

You could try to reprogram the firmware, using the suggestions in question 11 if necessary.

If you have an oscilloscope or other suitable measuring equipment, try connecting that to the data output of the development board while you do the reprogramming exercise. That should tell you whether the XBee is responding at all.

Or try this. Remove the XBee from the development board and then try to reprogram something that isn't there. Yes, this is a serious suggestion. What you will learn is what error message you get from X-CTU when it receives no responses at all. If you get the same message when the XBee is inserted in the board, then fear the worst.

## **13. I want to post a question in the forum. What information should I include?**

You should certainly include the firmware version that you're using. And be warned: if your firmware is not a recent version, you're likely to get a reply suggesting that you upgrade and try again. So if you have a problem, upgrading to see whether it goes away is a useful step before you post.

Depending on the problem, it may also help if you give a list of the non-default parameter settings on your module(s), or a list of all settings if you're not sure which are different from the defaults. Using X-CTU it's possible to generate a .pro file giving details of all settings. Attaching that file to your post is one way of providing the information.

And if your question concerns an API packet you're having trouble with, please post the exact packet you sent, plus the response packet (if any).

All these details will help people to help you.