

VSC-X: Using ZigBee to Implement a Virtual Serial Cable for Remote Programming of Mobile Radios

John A. Hansen, Ph.D., W2FS
Department of Computer and Information Sciences
225 Fenton Hall
State University of New York at Fredonia
Fredonia, NY 14063
john@coastalchip.com

Abstract

XBee-Pro is a relatively high power ZigBee-based module. This paper describes how to use these modules to create a virtual serial cable, allowing users to program their mobile radios remotely.

Introduction

With hundreds if not thousands of memories available, mobile radios have become increasingly capable and increasingly complex to program. This has naturally given rise to the development of computer software to allow users to manage memory contents and program the radios. However, mobile radios are most often located in cars and trucks where users generally don't have computers. As a result, in order to program the radios it is necessary to bring the radio to the computer or bring a computer to the radio. This causes many users to make changes to their radios either infrequently, or never.

The solution to transmitting data over a distance that is too long to run a cable is, of course, radio. For some time I've been looking for an inexpensive mechanism for creating a "virtual serial cable" that would allow users to plug a dongle into their computer and another dongle into their mobile radio and have the radio programmed wirelessly.¹ The design characteristics necessary to make such a system feasible include:

- A data throughput rate that is compatible with most radio programming software (at least 9600 baud).
- Error free data transmission.
- Transmitter power that is sufficient to communicate from inside a house, at least 50-100 feet to a vehicle parked outside. The signal must be strong enough to pass through both the house and the car (which is mostly metal).
- Some rudimentary form of data security so it's not possible for someone to pull up across the street from your car and reprogram your radio.
- Compatibility with computer USB ports (not serial!)
- Compatibility with radios that use both RS-232 level signals and TTL level signals.
- Reasonable cost (no more than about \$100 for the link).

Introducing ZigBee

¹ Hansen, John A., The Nordic nRF2401 Single Chip Data Transceiver: High Speed, Short Range Data Communication At An Extremely Low Cost. *Proceedings of the 26th ARRL and TAPR Digital Communications Conference September 28-30, 2007* (Hartford: ARRL, 2007).

ZigBee (formally IEEE 802.15.4) is a data communications protocol designed to create *ad hoc* wireless mesh networks using relatively short range radios. ZigBee supports multicasting and wireless routing of packets through a mesh network of ZigBee nodes. Most of the radios that support ZigBee are rather low power, often 1 to 2 milliwatts (0 to 3 dBm). The application envisioned here does not use most of ZigBee's wonderful networking characteristics because it only requires simple point to point data transmission. It is not hard to imagine some really interesting (non-amateur radio) applications for ZigBee involving collection of data from groups of sensors distributed over a significant area. However, from the standpoint of this project, ZigBee is mostly interesting because of the availability of the XBee-Pro radio.

XBee-Pro

XBee radio/processor modules were originally developed by Maxstream, which was later acquired by Digi International, Inc. Digi describes the XBee series of modules as “designed to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks”.² It's not exactly clear what the words “within the ZigBee protocol” actually mean. These radios do not appear to be compatible with other ZigBee products. For the purposes of this project, that really doesn't matter, but it does mean that you can't reasonably expect to pick up an XBee-Pro module and have it inter-operate with other ZigBee products. Like most other things these days, the modules operate in the 2.4 GHz ISM band (900 MHz versions are also available at a higher price).

The beauty of the XBee-Pro modules is that they have an adjustable output power of up to 50mw (17 dBm). This is substantially higher than any of the small, low cost radios that I've previously come across. There is, by the way, also a standard XBee module (not Pro). It has an output power of 1 mw.

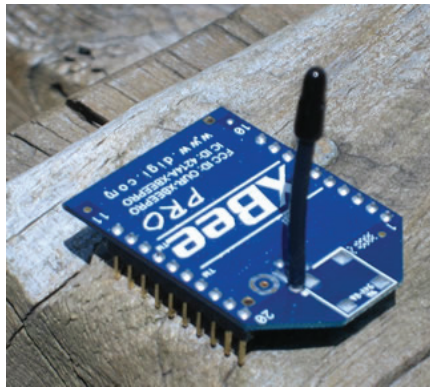


Figure 1: The XBee-Pro Module (wire antenna version).

Figure 1 shows the XBee-Pro module. These modules are available in the three antenna configurations. Figure 1 shows the module with the wire whip antenna built in. A second configuration uses a “chip antenna” which is fully contained on the module itself. The range of this module is somewhat less than the module shown in Figure 1 because of the limited antenna. For longer distance applications there is another version that includes an external antenna connector, allowing the user to connect whatever antenna they would like. The quantity 1 prices of these modules are all \$32 from suppliers like Mouser Electronics.³ The lower power modules cost approximately \$20. Note that

² Digi International, Inc. *XBee/XBee-Pro ZB OEM RF Modules* (Minnetonka, MN: 2008). p.5

³ www.mouser.com

modules come with pins, so they are designed to be plugged in. Unfortunately, the pin spacing is 2 mm, not the standard .1 inch. As a result, if you use a socket it is somewhat more expensive than would be case with standard .1 inch spacing. On the bright side, only four pins from the module (power, ground, TXData and RXData) are absolutely necessary to make it work.

The data interface to the modules is asynchronous serial TTL level data. As a result, it is relatively easy to interface them to either a PC (assuming you provide a USB or RS232 interface) or to a radio (with no interface for radios that support TTL level signals or with an RS232 level converting interface for the rest). In either case it's not really necessary to provide any "intelligence" in the circuit beyond that which is supplied by the XBee module itself. The XBee module also requires a 3.3 volt power supply of approximately 300 ma.

Casual experimentation showed that the XBee-Pro modules could easily handle 9600 bps data over fairly decent distances. I plugged one into the computer on the second floor of my house and was able to walk 3 houses down the street with no loss of signal. I brought the unit back to my driveway, put it in the trunk of my car and *closed the trunk* and still had almost perfect copy. With the radio on the floor of my car under the driver's seat and the car parked in the driveway, I had no trouble reprogramming the radio from inside the house. Digi's documentation says the units have an outdoors range of up to a mile, but I tend to discount manufacture range claims (perhaps they were using the model that could connect to an external antenna). Of course, your mileage may vary.

VSC-X: Constructing a Virtual Serial Cable

I wanted to create modules that would provide all of the circuitry necessary to interface the XBee modules to a computer and to a remote radio. I've called the resulting adapter pair VSC-X, meant to describe an eXtended Virtual Serial Cable. On the PC side, this involves creating a USB interface.

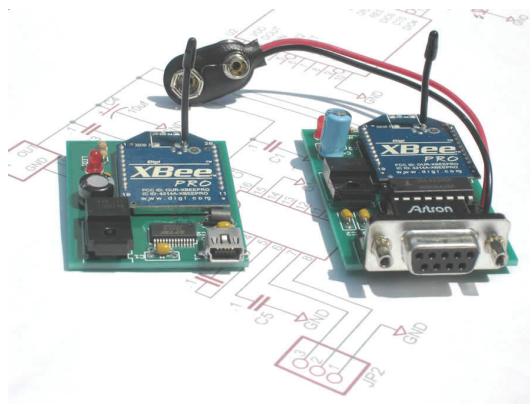


Figure 2: The VSC-X USB and Serial Adapters

Power for the VSC-X USB module is drawn directly from the USB port on the computer, but an external 3.3 volt regulator is required to step down the 5 volt USB power supply to 3.3 volts. The USB interface to the computer is provided by a standard FTDI FT232RL USB interface chip. The resulting circuit is shown in Figure 3. Note: C4 must be at least 35 volts.

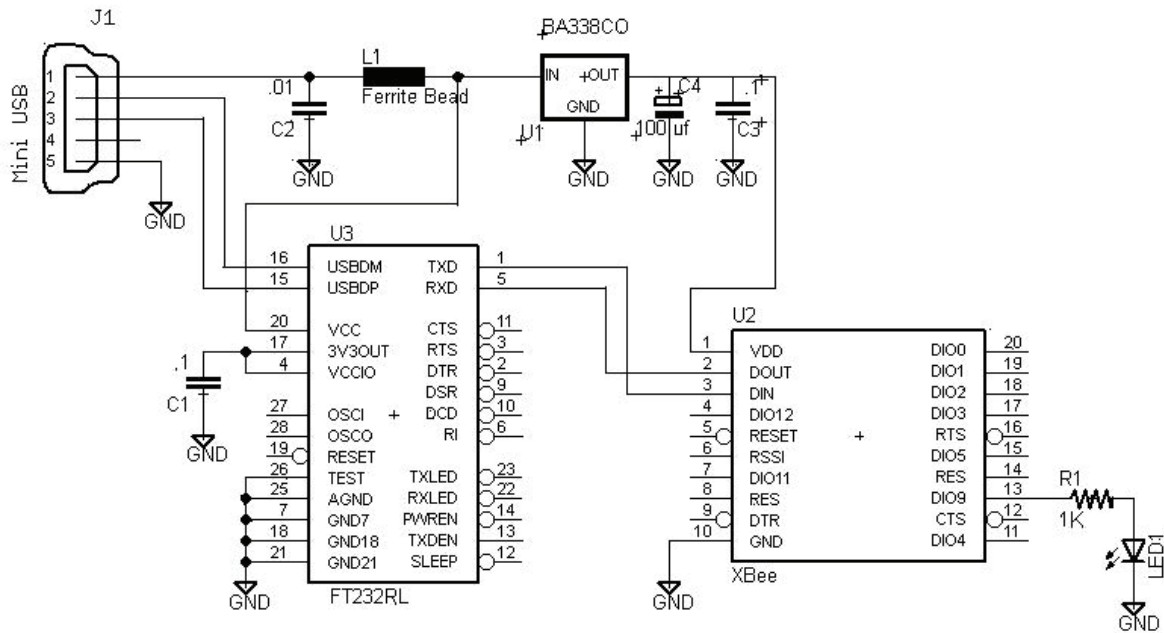


Figure 3: VSC-X USB Adapter Circuit

To use the VSC-X USB adapter, you first must load a driver on your computer. Drivers are available for Windows and Mac OS operating systems on the FTDI driver website:

<http://www.ftdichip.com/Drivers/VCP.htm>

If you are using a Windows based computer the easiest way to load the driver is by utilizing the “setup executable” that is available on the above page. If you are using Linux, the driver is already built into the kernel. After the driver is installed, when you plug the VSC-X adapter into your computer (using a mini USB cable such as those that usually come with digital cameras) the adapter will appear as a serial port on your computer. If you prefer that it is a different serial port (for example, if the number is too high to be recognized by your radio programming software), you can change this from the Windows device manager.

A second board was developed to support the serial port connection to the mobile radio. The VSC-X serial adapter requires 5 to 15 volts DC at 300 ma. Typically this would be provided either by the radio’s power supply or with a 9 volt battery. Be very careful if you power it from the radio’s power supply because if the ground connection from the adapter to the power supply comes loose, the full 300 ma will flow through the ground circuit on the radio’s computer connector. The VSC-X adapter lowers the input voltage to 3.3 volts to power the XBee. The XBee also uses 3.3 volt signal levels. As a result, I used the 3 volt version of the usual MAX 232 chip (the ICL3232). The resulting circuit is shown in Figure 4. Note: C4 must be at least 35 volts.

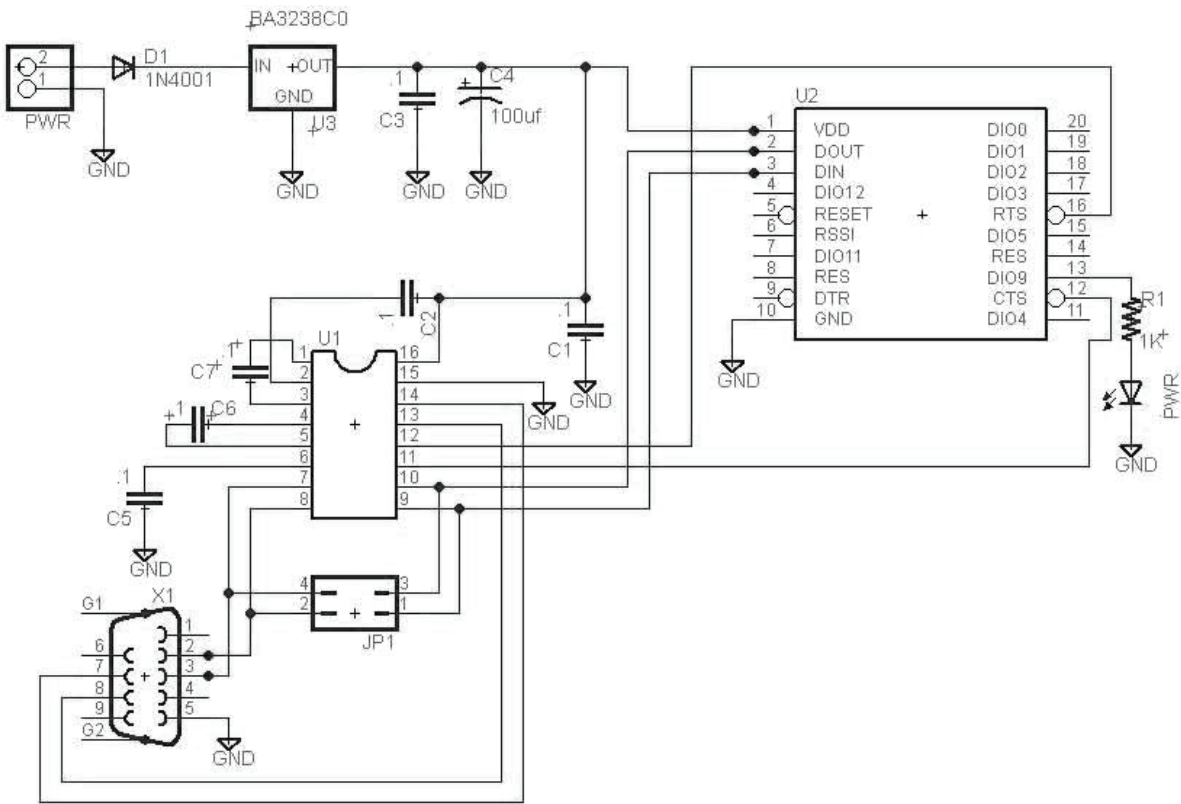


Figure 4: VSC-X Serial Port Adapter Circuit

The VSC-X serial adapter utilizes a female DB-9 connector. Since radios that use RS-232 level signals (like the Kenwood D700) generally have male DB-9 connectors, the adapter will plug directly into the radio (no cable required!). While the Kenwood D710 also uses RS-232 level signals, it does not have a DB-9 connector. As a result, an adapter cable is necessary. You can either use the PG-5G cable available from Kenwood, or you can construct your own using the diagram in Figure 5:

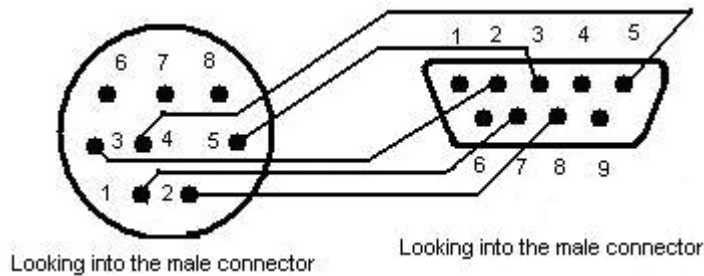


Figure 5: Hook-up Diagram for Connecting VSC-X to a Kenwood D710

If the radio you are using requires TTL level signals rather than RS232 level signals (for example, the Yaesu FT-857 or most ICOM radios), it is necessary to remove the ICL3232 from the circuit and jumper the inputs and outputs of the chip's socket. This can be done by placing jumpers across pins 1 and 2 and across pins 3 and 4 of JP1.

Since virtually all radios that use TTL level signals do not have a DB-9 connector, it will be necessary to construct a cable that connects the TXData, RXData and ground lines. For example, on the 800 series Yaesu radios, the connection to the computer is made through an 8 pin mini-din connector, wired as shown in Figure 6.

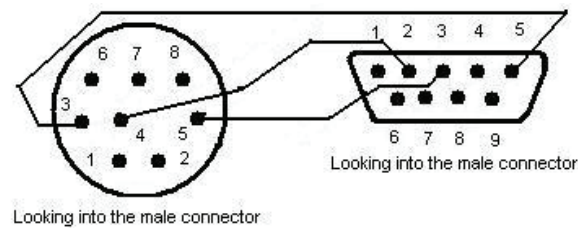


Figure 6: Connection Diagram for Yaesu 800 Series Radios (FT817, FT857, FT897, etc)

In order to connect VSC-X to an ICOM radio with a CI-V interface (706, 756, 703, etc.), you will need an adapter using a 1/8 inch phone plug using the following wiring diagram.

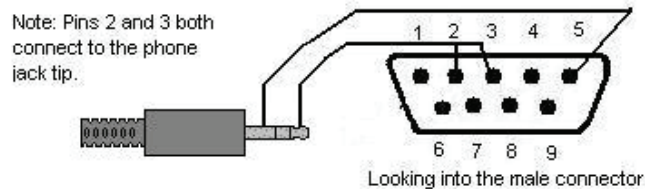


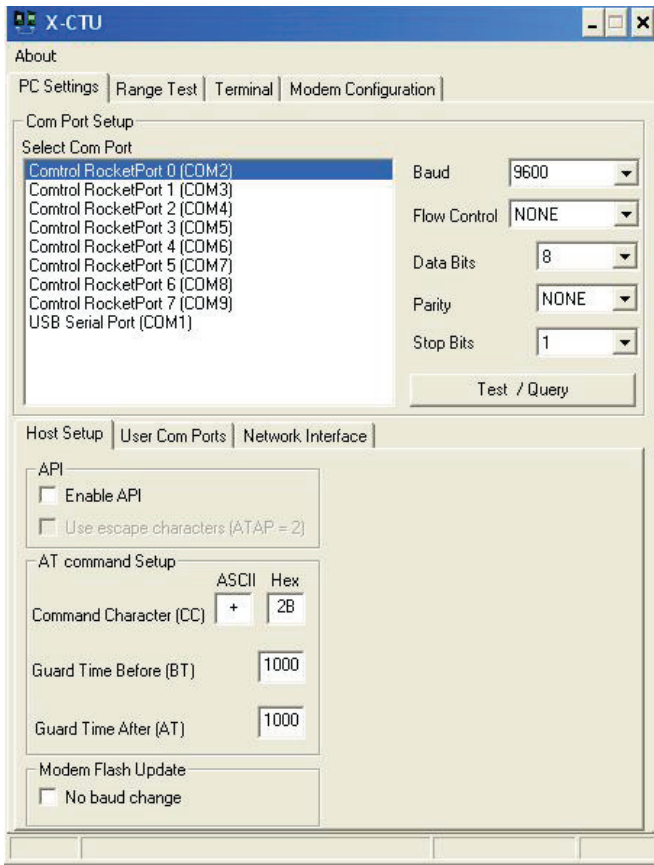
Figure 7: Connection Diagram for ICOM radios that use a CI-V style interface.

X-CTU: Software for Range Testing, Reconfiguration, and Diagnosis

The range tests described above were conducted using a program called X-CTU, which is available from Digi's website free of charge:

<http://www.digi.com/support/kbase/kbaseresultdetl.jsp?kb=125>

To do range testing, you can use this software on two different computers and connect the VSC-X USB module to one computer and the serial module to the other. Ideally one of these computers would be a portable laptop that would allow you to walk around. When you connect the serial module to a computer rather than a radio, it is necessary to use a null modem cable to make the connection. Then run the X-CTU software on both computers and you will see the screen in Figure 8. All of the serial ports on your computer should be visible including the one that has the USB module connected to it (COM1 in Figure 8).



I Figure 8: Digi's X-CTU Program

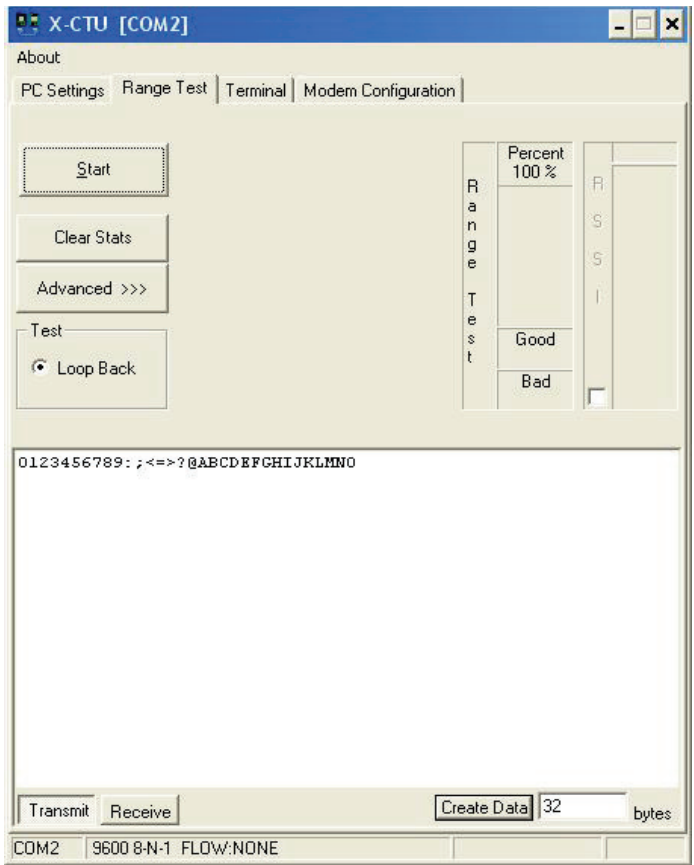


Figure 9: The X-CTU Range Test

Select the port that you have connected your module to and press the Test/Query button. You should see a confirmation that the module has been found. Note the default data communication rate is 9600 baud. Note that this is the rate at which data is transmitted between the module and your PC, not the rate that data is transmitted over the air. You can alter this value, but if data arrives faster at the module than it can be processed over the radio link, the relatively small buffer in the module will overflow and errors will result. I have not found data rates higher than 9600 baud to provide reliable communications for downloading data to mobile radios.

To do a range test, select the range test tab from the X-CTU software on both computers. You will then see the screen in Figure 9. Push the “Transmit” button on one of the copies of X-CTU and the “Receive” button on the other. Then press the Start button on both computers. If the modules are communicating, you should see a rising count of “Good” packets on both screens. You can then walk around with one of the computer/modules until you begin to see “Bad” packets. This will give you an idea of how far the modules can communicate.

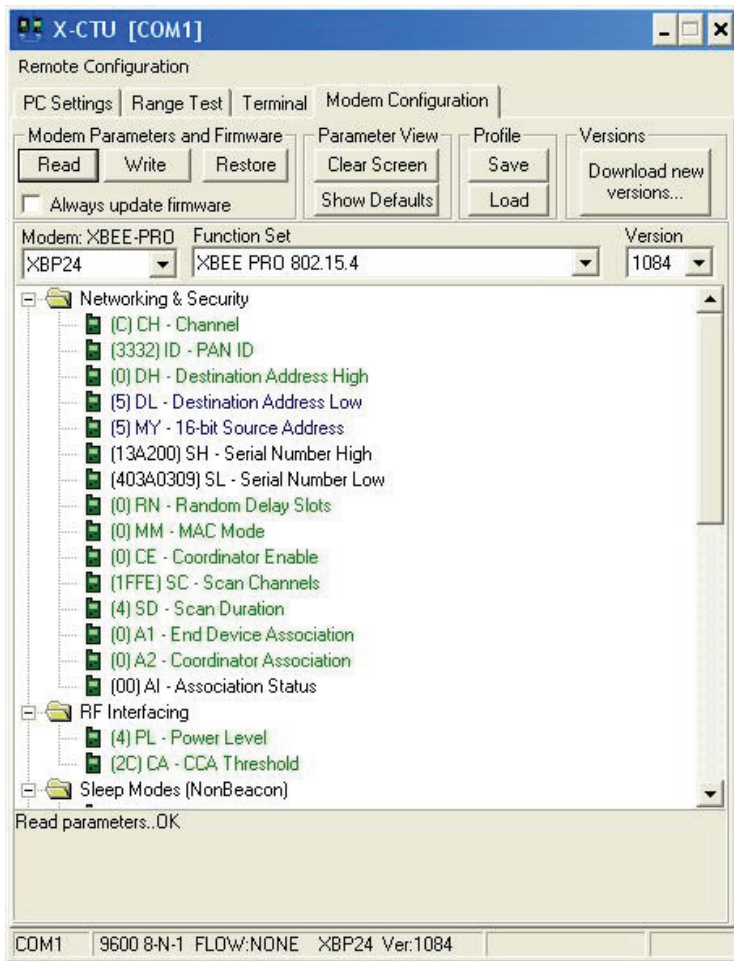


Figure 10: Modem Configuration Parameters

computer. The destination address is the address that this module expects to communicate with. If you click on these items you will be able to edit them. By altering these values you can specify your own “private” addresses for each of the modules connected to your radio(s). Note that there are a number of other parameters here that are of interest, including the channel the radio operates on within the ISM band and the power level. Pressing the “Write” button then causes the new values to be written into your XBee-Pro. Note that the values in the XBee-Pro will not be changed (despite the fact that changes appear on the screen) until you press the “Write” button.

Conclusion

XBee-Pro modules can serve as the basis for the creation of a virtual serial cable. My original intent was to create a mechanism to remotely program radios. However, I have been advised by users that the range of applications is actually significantly wider than I original intended. One user is using VSC-X to provide communication with his remote weather station that had originally connected by a serial cable. Another is using it to communicate with tower mounted peripherals for his station. Virtually anything that was once connected by a serial cable can be controlled by VSC-X.

Information on obtaining VSC-X modules can be found at www.vsc-x.com.

The X-CTU software also allows you to reconfigure the modules to enhance data security. Given the relatively short range of these modules and the non-critical nature of the data being communicated, you may not be very concerned about the security of your data transmissions. However, if you are, or if you wish to use a single computer to communicate with more than one radio (and those radios are left on at the same time), you can alter the addresses of the modules to something other than the default value (which is 0). To do this press the “Modem Configuration” tab on the X-CTU software. Then press the “Read” button in the “Modem Parameters and Firmware” section. The screen in Figure 10 should appear.

Note the values that say “Destination Address High”, “Destination Address Low” and “16 bit Source Address”. The default setting for all of these is 0. The source address refers to the address of the module currently attached to your