PARALLAX SEMICONDUCTOR
www.parallaxsemiconductor.com
sales@parallaxsemiconductor.com
support@parallaxsemiconductor.com
phone: 916-632-4664 • fax:916-624-8003

Application Note AN007

Soft Loading an Application Image into the Propeller P8X32A via XBee Transceivers

Abstract: There are times when it is extremely convenient to load a new version of firmware without having to be physically present at the device. This article describes one method of loading an application image into the Propeller chip's main RAM or EEPROM using XBee wireless transceivers.

Once the system is set up and primed, the resulting operation is similar to in system programming (ISP) with the added benefit of wireless serial communication of user data. Illustrations describe the operation of this loader in detail, and step-by-step instructions show how to connect an XBee transceiver to a project and program it wirelessly.

Introduction

Whether an engineer is in the development cycle or supporting a deployed product, the ability to load and debug new firmware onto a project is paramount. However, when a product has to remain outside for safety reasons or is installed in a customer's attic, it is not so convenient to debug a software problem.

Pairing a Propeller P8X32A with Digi International's XBee wireless transceiver easily solves the problem of remote loading and debugging new firmware. The solution presented in this article takes advantage of the Propeller's multi-core architecture to give it the ability to wirelessly program itself and utilize the XBee as a wireless serial cable.

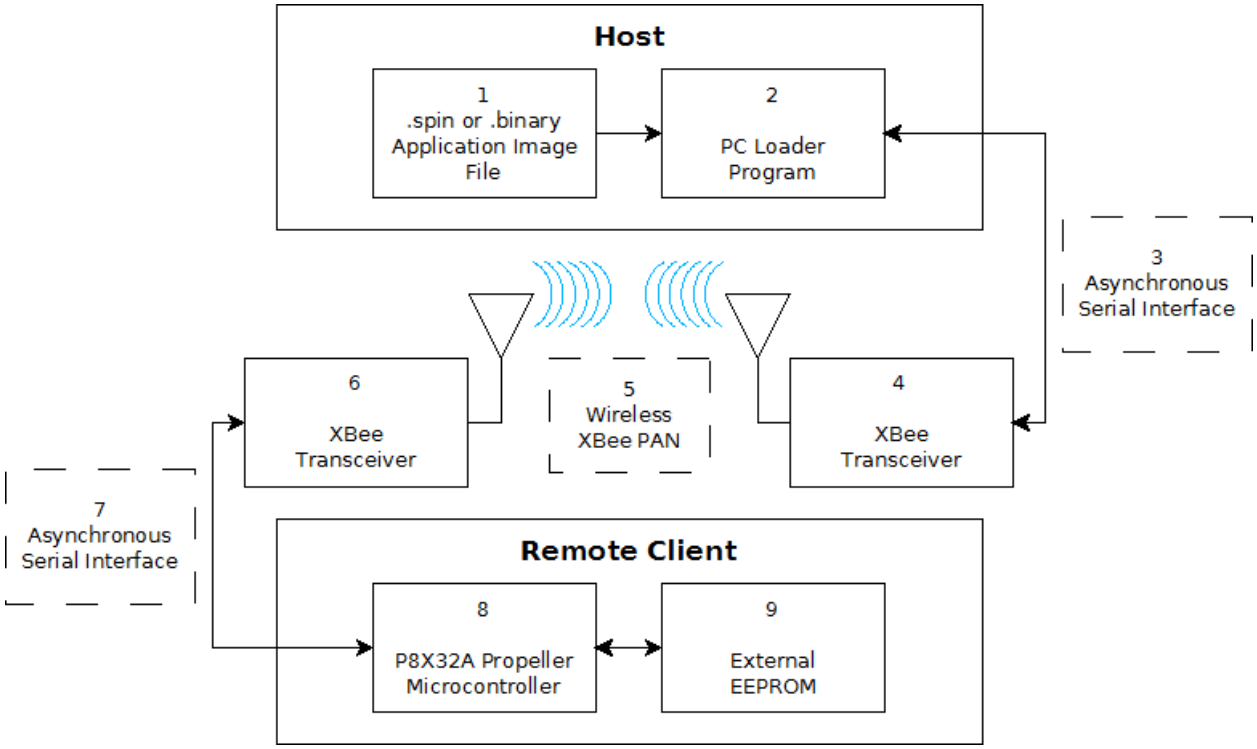
Achieving this functionality requires only an available cog and two I/O pins on the Propeller. This buys the same flexibility of wireless programming and debugging as a physically connection to the device.

Soft Loading Background and Theory

The term "soft load" puts emphasis on software performing the steps necessary to load a new program into the Propeller. When a compiled binary, also called the application image, is ready for loading on the Propeller, a boot loader routine has to run on the microcontroller in order to fetch the application image and copy it into the Propeller's main RAM for general execution. Normally at startup, the regular boot loader residing the Propeller's ROM takes care of this functionality. However, in the case of a soft loader, special user-written software executes during run time to mimic the operation of the ROM boot loader.

Figure 1 is an illustration of the high-level structure of the major components involved in this implementation of this soft loader. Refer to each component's section number for more detail about that particular operation.

Figure 1: High-level Block Diagram of the Major Components



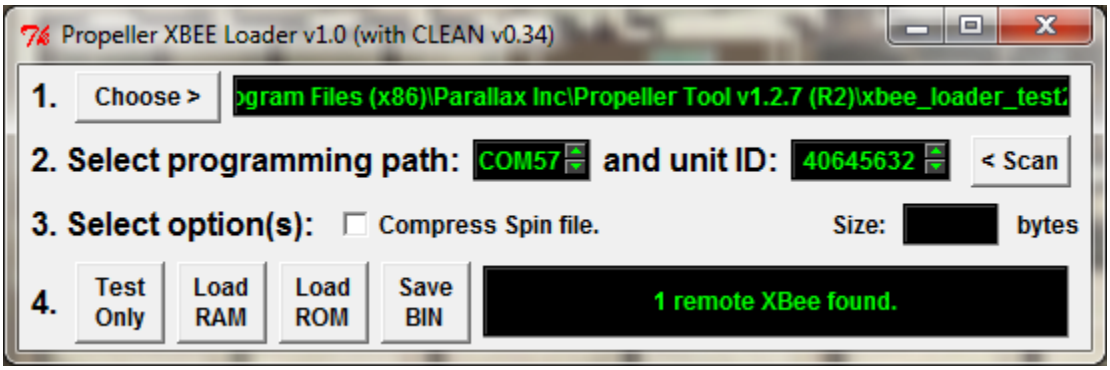
1 – Application Image File (.spin or .binary)

The Propeller XBee Loader Program (discussed in section 2) requires either a .spin source code file or a compiled .binary to load onto the remote P8X32A Propeller microcontroller. Either type of file can be loaded, though the .spin file will be compiled before it is loaded. As long as the .spin source file is syntactically correct and can be correctly compiled, any editor can be used to author the spin source code including the Propeller Tool[1], Brad's Spin Tool[2], or Notepad.

2 – Propeller XBee Loader Program for the PC

The Propeller XBee Loader program, a third party application written by Phil Pilgrim of Bueno Systems, loads an application image onto a remote Propeller project. The Propeller XBee Loader program requires both the Propeller Tool and Propellent[3] to load an application image. It is because of these supporting software requirements that the Propeller XBee Loader program can only be run on a computer running Microsoft Windows XP or newer. Figure 2 is a screen capture of the Loader's user interface.

Figure 2: Propeller XBee Loader Program



Upon startup, the Propeller XBee Loader program searches for an active serial port that is likely to be interfaced to an XBee. If it finds an XBee attached, the program will then use the XBee to search for other XBee modules that have associated with the host's XBee. Once the network discovery has completed and one or more associated XBee modules have been found, you have the option to select which remote XBee to attempt a soft load to. Available remote XBee modules are enumerated by the lower 32-bit address of the XBee's full 64-bit address.

The Loader's Choose button selects which Propeller application image to load into the target Propeller. The programming path field sets which of the host's COM ports has an XBee attached. The unit ID field selects the remote XBee target for the soft load attempt from the list of associated XBee modules found; the Scan button will refresh this list.

The option to "Compress Spin file" attempts to reduce the size of the downloaded application image. It makes copies of all referenced objects and methods, then recompiles the application image with any unused method calls and objects removed.

The four buttons at the bottom of the user interface select which operation the application performs. "Test Only" attempts to compile the .spin program specified by the "Choose" option. This functionality allows for comparison of compressed vs. un-compressed application image sizes. The "Load RAM" and "Load ROM" buttons initiate a soft load to the remote target's Main RAM or external EEPROM respectively. The "Save BIN" button saves a possibly compressed binary version of the application image.

3 – Asynchronous Serial Interface with the Host

The Propeller XBee Loader program needs to be able to communicate with an XBee module connected to the host computer. The simplest way to connect the XBee module to a serial port on a PC is with an XBee USB Adapter Board[4]. There are other means; the XBee's DOUT signal line connects to the serial port's RX line, while the DIN signal line connects to the serial port's TX line. Be sure to observe proper signaling voltage level buffering between the PC and XBee's 3.3 V operating voltage. If using an external power supply to power the XBee, make sure that the XBee and host computer share a common ground reference.

4 – Host's XBee Transceiver

The Propeller XBee Loader application communicates with a connected XBee module via an asynchronous serial interface (a serial port) operating at 9600 baud. The USB Adapter

Board supplies the XBee with proper power levels and brings the signaling lines straight to the XBee—no additional hardware is required to connect the XBee to the computer. The XBee requires 3.3 V and a maximum current supply of approximately 215 mA, depending on the type of XBee module used. Both the 802.15.4 and the ZB XBee modules are able to perform a wireless soft load, but be aware that the ZB and 802.15.4 XBee modules are not compatible with each other—use at least two of the same type to establish a functioning network. Also, ZB XBee modules may require additional configuration; see the device's datasheet[5]. Supplementary drivers beyond what is required by the serial port connection are not needed to allow the operating system to communicate with the XBee.

For a general introduction to using XBee modules, including examples with the P8X32A, see Getting Started with XBee RF Modules.[6]

5 – Wireless XBee PAN

When configured properly, two or more XBee modules form a personal area network (PAN) capable of bidirectional traffic between two nodes in the network. An XBee network implements a layer of communication protocol that ensures packet delivery and error correction. For this article, think of the XBee PAN as a wireless serial cable.

6 – Remote Client's XBee Transceiver

The remote client's XBee transceiver is set to join with the host XBee's PAN and Channel. This allows the client XBee to associate with the host. For the most part, default settings on the XBee will be appropriate for most users.

7 – Asynchronous Serial Interface with the Microcontroller

Make sure that the client XBee's baud rate is set to 9600. This should match the baud rate specified in the Propeller application when starting the xbee_loader object. Default XBee settings will take care of the baud rate.

8 – Propeller P8X32A Microcontroller

Since the Propeller itself does the work of the boot loader during run time, the bulk of the "action" takes place here. See Figure 4 for a flow chart of the operations performed by the xbee_loader object's code to load a valid application image.

The method of soft loading implemented here is quite robust. Once started, the xbee_loader object, which runs on the Propeller, acts like a low-speed wireless serial cable operating at 9600 baud until the remote device receives a very specific string of bytes.

This specific string drops the xbee_loader object into its command mode. At this point, the xbee_loader object stops acting as a serial pass through, stops all other running cogs, and waits to receive command bytes. These command bytes instruct the soft loader routines to perform either a RAM load or EEPROM load. Once the host issues a command and the xbee_loader object receives and verifies that command, then the host sends a byte count of the new application image to the remote unit so a checksum can take place after the firmware transfer completes. If everything checks out, the xbee_loader object instructs the Propeller to begin executing the new application image.

Execution begins in one of two ways, depending on whether RAM or EEPROM has been loaded on the remote unit. If the new application image has been loaded into external EEPROM, the Propeller simply reboots itself and boots from EEPROM normally. On the other

hand, if the application image has been loaded into the Propeller's main RAM, then the Propeller cannot reboot as the newly loaded information in RAM would be lost. Instead of rebooting, the Propeller changes its system clock settings to the new application image's clock settings. Next, the xbee_loader object instructs the Propeller to launch cog 0, executing the first Spin instruction. At this point, execution begins as normal.

Once in command mode, if the loader receives invalid command bytes, it will prompt the host computer for correct command bytes. The loader will not leave command mode until it executes a valid command. However, this does not necessarily mean that a valid application image has been accepted. In the case that the downloaded application image is corrupted and the checksum fails, the loader will attempt to recover gracefully. The recovery operation consists of re-loading the original application image from the external EEPROM. The checksum verification happens before the loader copies a verified image to the external EEPROM, thus ensuring program integrity.

The "xbee_loader.spin" object provides serial pass through functionality while not in command mode, operating at 9600 baud. The serial transmit and receive routines can be accessed by calling the TX or RX methods respectively. As expected, the TX method requires a single argument – the byte to transmit. Similarly, the RX method returns the next byte that is waiting in the receive buffer and blocks if no byte is in the buffer.

9 – External EEPROM

Though technically not required to perform a soft load into Main RAM, in practice an EEPROM is almost always a necessity when the Propeller boots from a power cycle or reset. The exception here is if an external host is ready to load an application image into the Propeller on startup, similar to how a host PC would load the Propeller's RAM. In order to soft load to the EEPROM, the EEPROM must be present in its normal configuration: I^2C communication lines connected to pins P28 and P29.

Performing a Soft Load with the XBee Loader Program

This section gives instructions on how to integrate the XBee soft loader presented here with a remote Propeller project. In order to use this soft loader, the target project must first be "primed" by wiring an XBee module to your project and programming the Propeller's EEPROM with a program that includes the started xbee_loader.spin object. Note that, in general, a soft loader requires its code to be running in a cog on the Propeller when a soft load is attempted. This implies that you must continue to include and start the xbee_loader object in the target Propeller application to have the ability to wirelessly load a new application image.

Minimum Hardware and Software Requirements

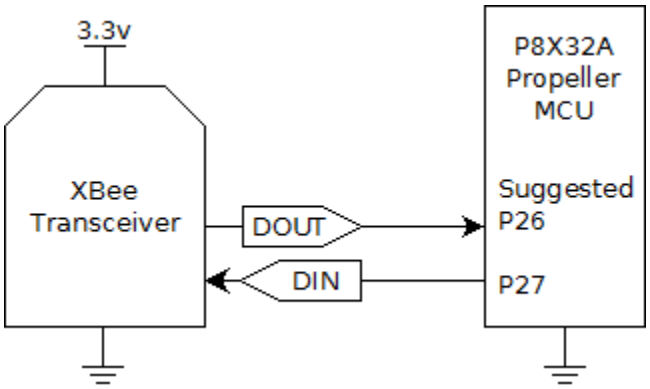
- Hardware:
  - Propeller project with two free I/O pins (other than pins P28 – P31)
  - XBee USB Adapter Board[4] or other serial connection
  - XBee Adapter Board[7]
  - XBee modules, either 802.15.4[8] or ZB[9] modules. Note: 802.15.4 XBee modules and ZB modules are not compatible with each other
  - Jumper wires if wiring up on a breadboard

- Software:
  - Microsoft Windows XP or newer
  - Propeller XBee Loader program for the PC
  - Propeller Tool ver. 1.2.7 or newer
  - Propellent 1.3 or newer
  - xbee_loader.spin object included and started in the remote Propeller application's source code
  - One available cog in the remote Propeller project

Step-by-step Instructions

1. Download and install the following software:
   a. Propeller Tool 1.2.7 or newer
   b. Propellent 1.3 or newer
   c. Propeller XBee Loader 1.0 or newer
2. Connect the XBee to the remote Propeller-based project. The XBee's DOUT and DIN lines can be connected to any I/O pin on the Propeller, except pins P28 – P31 (which are reserved for dedicated host communication and external EEPROM signals). The code example in Figure 4 defines the XBee's DOUT pin as connected to Propeller pin P26 and the DIN pin to Propeller pin P27. See Figure 3 for a visual representation of the required connections.
3. Connect the host's XBee to an available serial port on the computer.
4. Configure the XBee modules to associate with each other. Digi's X-CTU software[10] is a convenient way to perform the necessary configuration, among other methods. The XBee's default configuration is sufficient for this process.
5. Run the Propeller XBee Loader program.
   a. Select the desired .spin or .binary file to be soft loaded to your remote project.
   b. The program should automatically detect an attached XBee. However, you have the option to select the desired COM port.
   c. The program will attempt to discover associated XBee modules within range of the host. To trigger this discovery process manually, click the "< Scan" button.
   d. Select the Spin file compression preference. For most cases, compression is not necessary.
   e. Click the button that corresponds to the desired load operation.
   f. Locate the Propeller Tool or the Propellent program, if prompted.
6. The new application image should begin to be loaded to the remote project.
7. The Propeller XBee Loader program will report success of failure upon completion of the download process:
   a. If the soft load was successful, the new application image will automatically begin to execute on the remote Propeller project.
   b. If the application image download failed, the xbee_loader object will restore the original application image wait in command mode for another valid command.

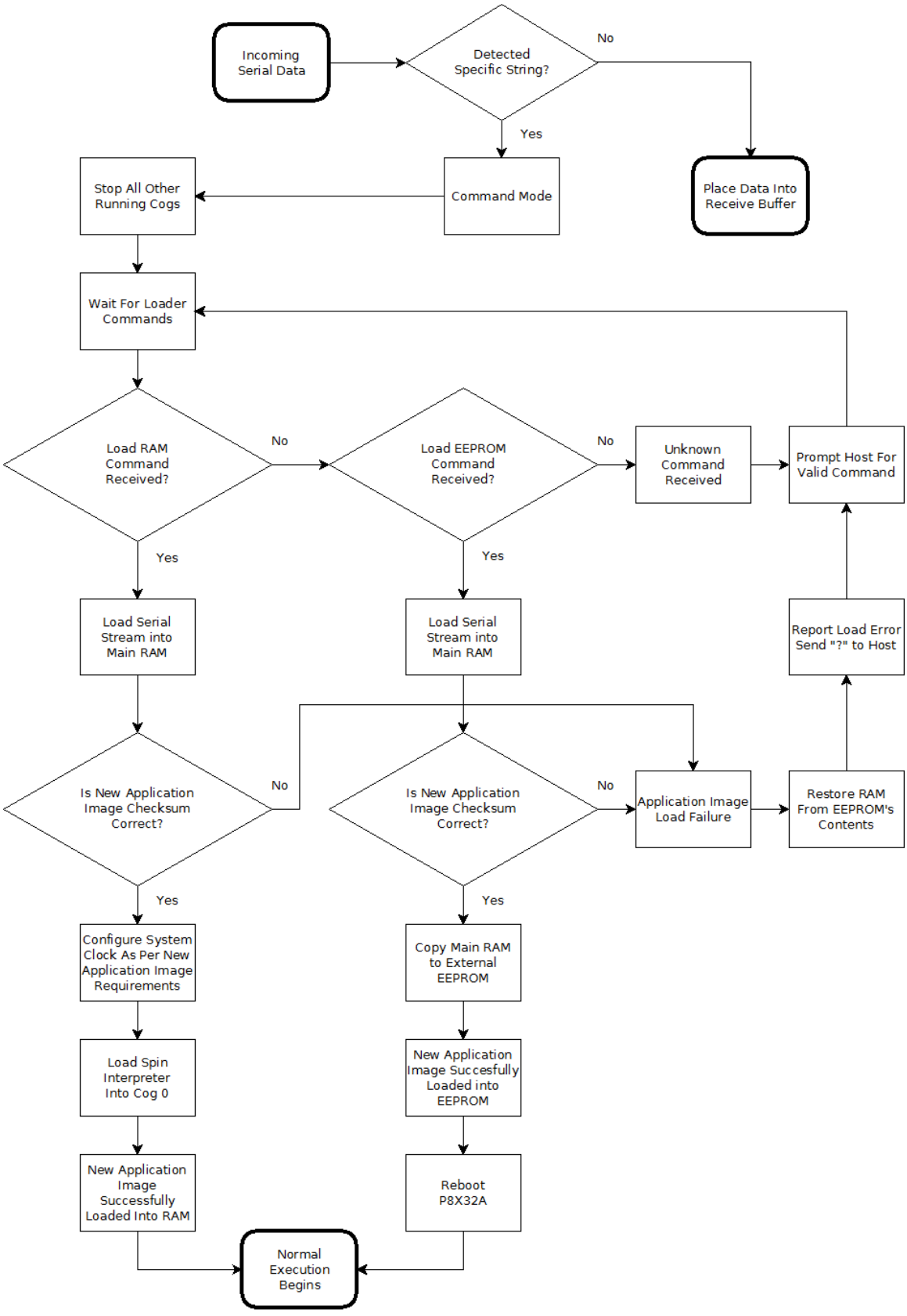
Figure 3: Recommended Connections between the XBee and the P8X32A



Example Code

```
{
File:           XBee_Loader_Test.spin
Version:        1.0
Date:           February 10, 2011

Description:
Sample code that implements wireless soft loading functionality
for the P8X32A Propeller microcontroller.  The code is a minimal
implementation which starts the xbee_loader object in an available
cog.  After the xbee_loader object is started, the method
"YourCodeHere" is called which blinks an LED at 1 Hz forever.
}

CON
  'set up the system clock
  _clkmode = xtal1 + pll16x
  _clkfreq = 80_000_000

  'xbee_loader definitions
  DOUT_PIN  = 26
  DIN_PIN   = 27
  XBEE_BAUD = 9_600

  'other constant definitions
  LED_PIN   = 0

OBJ
  'import xbee_loader object
  soft_loader   : "xbee_loader.spin"

PUB go
  'start xbee_loader object
  soft_loader.start(DOUT_PIN, DIN_PIN, XBEE_BAUD)

  'call to the "YourCodeHere" method
  YourCodeHere

PUB YourCodeHere
  'set pin to output
  dira[LED_PIN]~~

  'main loop, repeat forever.  Toggles an LED.
  repeat
    !outa[LED_PIN]
    waitcnt(cnt+clkfreq/2
```

Figure 4: Operation of the soft loader running in a cog on the Propeller P8X32A



Resources

The zip archive download at www.parallaxsemiconductor.com/an007 contains the following files:

xbee_loader_test.spin
xbee_loader.spin
Propeller XBee Loader utility

References

1. Propeller Tool: www.parallaxsemiconductor.com/software
2. Brad's Spin Tool: http://www.fnarfbargle.com/bst.html
3. Propellent software: www.parallaxsemiconductor.com/software
4. XBee USB Adapter Board; Parallax #32400; www.parallax.com/go/xbee
5. XBee/XBee-PRO ZB RF Modules; http://ftp1.digi.com/support/documentation/90000976_G.pdf
6. Getting Started with XBee RF Modules pdf; www.parallax.com/go/xbee
7. XBee Adapter Board; Parallax #32403, www.parallax.com
8. XBee 802.15.4 modules; Parallax #32404 – #32407; www.parallax.com
9. XBee ZB modules; Parallax #32408 and #32409; www.parallax.com
10. Digi International's X-CTU configuration tool; http://www.digi.com/support/kbase/kbaseresultdetl.jsp?kb=125

Revision History

Version 1.0: original document