Introduction

As researchers and teachers at Southern Illinois University Carbondale (SIUC) in Electronics Systems Technologies (EST), we have used the Series 1 XBee extensively in the classroom and in research endeavors. Through national and international collaborations, we have developed systems using the Series 1 XBee in real-time biological monitoring, citrus vibration monitoring during harvest and other monitoring and control systems—such as robotics! With the Series 2 modules we have collaborated with the USDA on field irrigation monitoring and control. We hope you find them as useful and fun as we have.

This tutorial is designed for Series 1 modules. What about Series 2?

All of our projects have been designed using Series 1 XBee modules. All examples in this tutorial are designed for Series 1. Series 2 XBee modules are not a drop-in replacement for Series 1 due to code incompatibility and X-CTU configuration requirements. So you will need to make some significant changes in the XBee setup and microcontroller code to perform these activities with Series 2 modules.

For example, the following pertains to Series 2 modules:

- An XBee must have the AT Coordinator firmware downloaded to it using Digi's X-CTU software and the XBee USB Adapter Board (Part number 32400).
- XBee modules used as Endpoints need AT Endpoint firmware downloaded using X-CTU software and the XBee USB Adapter.
- If a Router is needed, the XBee needs to have AT Router firmware downloaded to it using X-CTU software and the USB Adapter.
- Unlike the Series 1 XBee, the addresses of XBee modules are NOT configurable by the user. They are assigned by the Coordinator, so it makes duplex communication more troublesome.
- For remote Endpoints to send data to the base Coordinator, simply use a destination address (DL) of 0.
- The best suggestion for Coordinator to remote Endpoint communications is to use the broadcast address (DL) of $FFFF from the Coordinator to send data to ALL Endpoints.
- If you decide to use API firmware versions, few to none of the API code examples in Chapter 6 or the XBee_Object.spin operations will be functional.

For these reasons we have chosen to use Series 1 XBee Modules in this tutorial. Series 1 XBee modules support networking and have proven capable for the vast majority of our projects. If mesh networking with routing is needed, some of the principles in this tutorial may be used, but the code will need to be modified to meet addressing needs. You can learn great information from this tutorial to use with Series 2, but it is written around the Series 1 XBee Modules.

XBee Modules Avaliable from Parallax Inc.	
Series 1 Modules	**Series 2 Modules**
32404 XB24-ACI-001 XBee 2mW Chip Ant 32405 XB24-AWI-001 XBee 2mW Wire Ant 32406 XBP24-ACI-001 XBee 60mW Chip A 32407 XBP24-AWI-001 XBee 60mW Wire A	32408 XBP24-Z7WIT-004XBEE PRO WIRANT 32409 XB24-Z7WIT-004 XBEE 1MW WIRANT

1:Networking & XBee Overview

The XBee RF Modem from Digi International is a wireless transceiver. The XBee uses a fully implemented protocol for data communications providing features needed for robust network communications in a wireless sensor network (WSN). Features such as addressing, acknowledgements and retries help ensure safe delivery of data to the intended node. The XBee also has additional features beyond data communications for use in monitoring and control. This chapter will discuss some essential elements of network communications and provide an overview of the XBee module.

Multi-Node Network Issues

Many wireless modules for microcontrollers simply send data and receive data on the provided frequency. It is up the end user and his application code to deal with issues such as media access rules, data delivery verification, error checking and, in multi-node networks, which node will accept and use the data. Devices using networking protocol can ease the work of the programmer by handling these tasks.

In discussing the problems associated with networking, consider humans and the problems and resolutions in ensuring the proper flow of communications. Certain rules of communications or protocols are used in ensuring our message flows properly from sender to receiver across the medium such as the air gap when talking in person.

Media Access - Can I talk now?

In conversations between two or more people, it is important that two people don't talk at once or the message from either or both may not get through as their words collide and create confusion to the listener. In networking terms this is media access – how each party gains access to get their words out to be accepted and understood. People in an informal group tend to wait for an opening before beginning to speak. What happens when two see an opening and both begin to speak at once? They note the problem, both back off and after a little negotiating one begins to speak, allowing that person use of the medium to get their thoughts out.

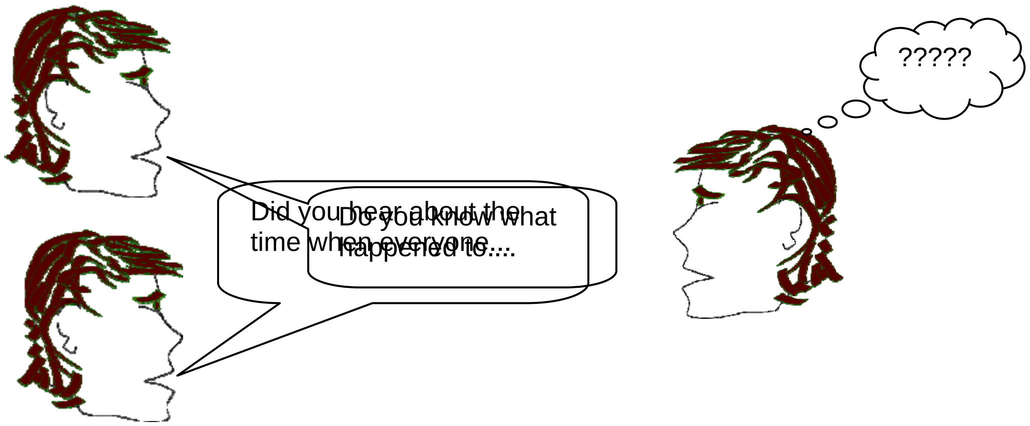
?????
Did you hear about the time when everyone...
Do you know what happened to....

Figure 1-1: When Words Collide

In more formal settings, other means may be used to control who speaks when. In the classroom, the teacher normally has exclusive access to the medium. By looking around the classroom, the teacher informally polls the students to see if anyone requires access. Should a student raise a hand, the teacher will allow that student to speak. In the classic novel *Lord of the Flies*, to ensure each castaway boy had a chance to speak in council, a conch shell was passed around. Whichever child held the conch had exclusive speaking rights. This ensured each had a chance to speak without others talking over him.

In networking, different network protocols handle access to media in different ways, but are similar to human communications in many ways. Ethernet (IEEE 802.3) over a shared wire uses CSMA/CD much like informal human conversations. All nodes (network devices) have an equal right to send data on the network (Multiple Access - MA). First, a node listens to see if the medium (CAT 5 cable) is in use. If it is, it will wait for an opening. Once it sees an opening, it attempts to transmit its data (Carrier Sense – CS). While sending data, the node also monitors what is on the medium and if they sense another node transmitted at the same time, the collision of data is detected and they both stop transmitting, wait a unique amount of time, and try again (Collision Detection – CD).

The WiFi protocol (IEEE 802.11) uses CSMA/CA for access – since a node cannot transmit and listen at the same time, they use extra measures to ensure collisions do not occur (Collision Avoidance – CA). Collision avoidance also helps to alleviate the situation where 2 wireless transmitters are within range of each other, but a third wireless transmitter is outside the range of the first transmitter. The transmitter in the middle hears transmissions from both neighbors, but the transmitters on the ends do not hear each other. They are "hidden" from each other. This is called, in networking, the “Hidden Node Problem.”

The Token Ring protocol (IEEE 802.5) ensures access and prevents collisions by passing a token from node to node in a ring fashion, allowing each to have exclusive access to the medium in-turn much like passing the conch shell. The USB protocol uses polling—the USB host (normally the PC) polls each USB device to see if they require access to communicate and provides them a time-slot to communicate with the host.

Addressing – Are you talking to me?

When talking in groups, informally or in more formal situations, such as the classroom, sometimes our words are meant for a specific person and sometimes they are intended for the entire group. When speaking to an entire group, our mannerisms and mode of speech imply the words are meant for the group. To speak to someone individually, we address that person directly using their name to identify the intended recipient. Others may hear the message, but understand they are not meant to respond or perhaps use the information.

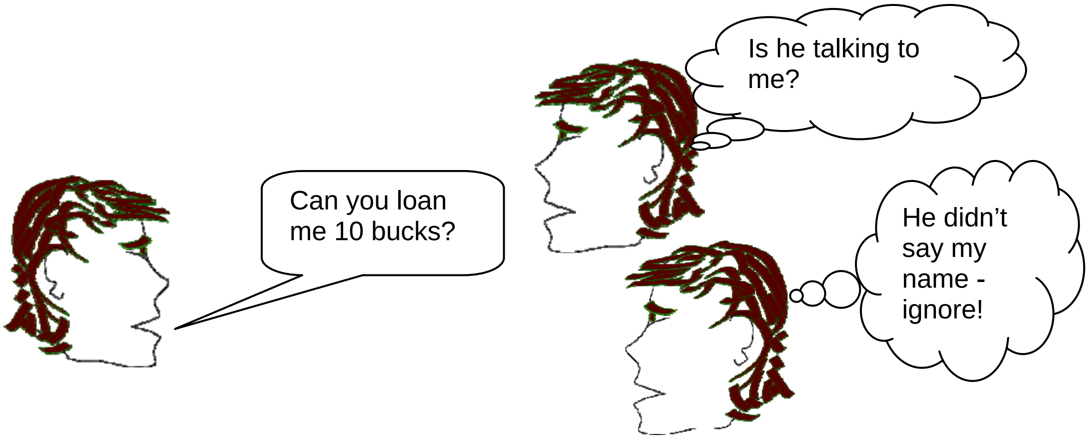

Figure 1-2: Was That Message Meant for Me?

Networking provides the same features—sometimes a message sent is meant for all to accept and sometimes a message is meant for only a specific receiver. While we use names in identification, in networks address numbers are assigned to various devices allowing data sent to be used by only one device (point to point) or by an entire group of devices (broadcast message or point to multi-point). In many cases all nodes receive the data sent message, but if the data does not contain their address they are programmed to disregard the message—only the device with the correct destination address uses the message. Networks also have specialized addresses to indicate that all the nodes should accept and use the message—the broadcast address.

The source address of the received message can be important too. You are in a group, hear your name and a question, but in order to respond properly you need to know who sent the message. In networking, part of the data sent includes the originator or source address of who sent the message in order to acknowledge and reply accordingly.

Error Checking & Acknowledgments – Did you understand me?

It's usually important that the words we speak are acknowledged and understood. Whether in informal setting where we are conveying some events in our lives or in more formal settings where instructions and orders are being conveyed, it can be important that we know the recipient received and understood the message. Some form of acknowledgment lets us know through verbal or non-verbal feedback they are receiving the message we convey. Some situations require direct acknowledgement, such as military orders, other situation where less important information is being sent may not require direct acknowledgement, such as excusing yourself from a group. In some circumstance it can be critically important that the message is not only received, but correctly understood. In the military, orders are often repeated back to ensure the party received the message without error. Many times, we respond in informal situations to ensure we understood the message—"Let's meet at 7:00 in the library." "Library at 7:00? Okay."

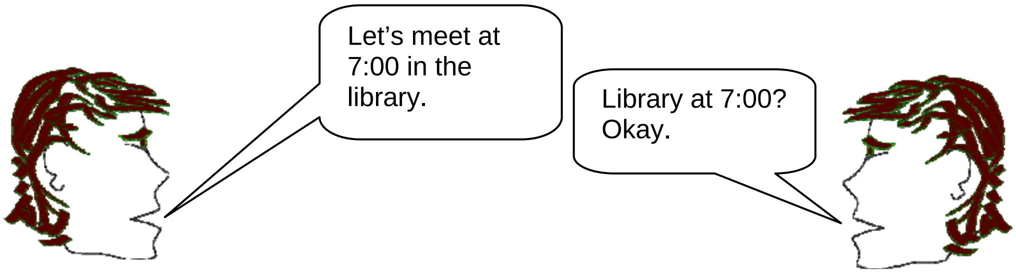

Figure 1-3: Did You Understand Me? Yes!

When passing data in a network, it is important to verify the message was received and the data contained no errors. A simple form of error checking is to add up all the bytes values it receives and add that value to the data sent (checksum value). On reception, the same math is performed on the received data and that value is checked against the received checksum value. If there is a mismatch, the data contains errors and is not acknowledged. If the message is not received or contains errors, the receiving node does not acknowledge the reception and the sending node retries the transmission.

Encapsulation – Pack it up and ship it!

Just as when we want to send a letter, we don't simply write the letter and drop it in the mailbox hoping it makes it to the recipient. The letter needs to be placed in an envelope and addressed to ensure that it is received and read by the correct person. The return address allows the recipient to know who it was from (and for return by the post office if required). Data on network transfers messages in a similar manner. The message (the data we are sending) is packaged with other data the protocol requires as shown in Figure 1-4. This added data typically includes: Source Address, Destination Address, Error Checking values, and other pertinent information needed by the protocol. Our message in encompassed with the other data to help ensure proper delivery to intended node.

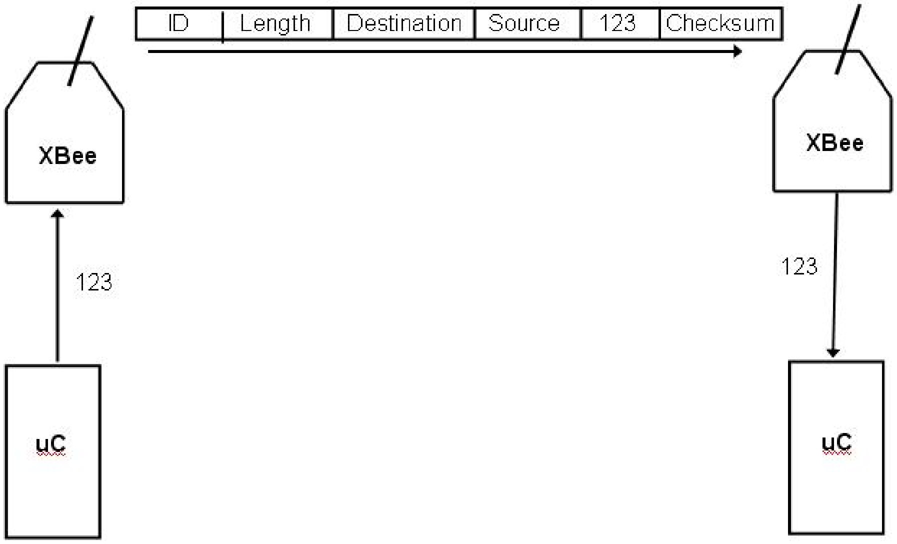

Figure 1-4: Packaging Data for Delivery

Application—Got the message, now what?

In networking terms, the packaging of the data, its point-to-point transmission, error checking and acknowledgement is handled by the Data Link protocol. This includes Ethernet, Token Ring and WiFi protocols. These protocols ensure the intended node receives the data error-free. It is up to the application to define what the message is and what to do with it once received. What message is sent and how it will be used is the task of the application engineer—those of us designing systems with nodes that send data between points. How will the software and hardware will interact with one another across the network?

Through this tutorial we look at how the XBee performs the link for our data, how the XBee can be configured as needed by our applications, and how to code our controllers to send and receive data for specific applications.

XBee Overview

XBee Benefits

In modern wireless protocols, such as WiFi (IEEE 802.11) and Bluetooth (IEEE 802.15.1), the protocol helps ensure data arrives at the correct destination without errors as discussed. The protocol greatly reduces the work of the programmer in ensuring data delivery. Some key features of protocols in ensuring data delivery and integrity include:

- Media Access: A means to ensure two network nodes do not transmit at the same time causing data collisions and errors in transmission.
- Addressing: A means to ensure only the intended node uses the received data, allowing data to be sent from one point to another point. Or, point to multi-point by sending a broadcast meant for all nodes on the network.
- Error Detection: A means to verify data received at the node correctly.
- Acknowledgements & Retries: A means to inform the transmitting node that the data was delivered successfully. Lacking this, several retries may be performed in an effort to deliver the data.

The XBee utilizes the IEEE 802.15.4 protocol which implements all of the above features. This protocol is known as a Low-Rate, Wireless Personal Area Network (LR-WPAN), it provides up to 250Kbps of data throughput between nodes on a CSMA/CA network. While not intended for large volumes of data, such as image files, it provides a means of moving data quickly between nodes for use in monitoring and control systems commonly referred to as a Wireless Sensor Network (WSN).

In comparison to Bluetooth (IEEE 802.15.1), the LR-WPAN is designed as a much simpler protocol with lower data transfer rates (250 Kbps compared to 1 Mbps). Bluetooth was designed as a replacement for peripheral cables and is used in communications between handheld devices, such as phones, requiring access security and high rates of data transfer.

- The XBee, using the IEEE 802.15.4 protocol, incorporates the following for communications and control on the WSN.
- Clear Channel Assessment (CCA): Before transmitting, an XBee node listens to see if the selected frequency is busy.
- Addressing: The XBee has both a fixed 64-bit serial number (MAC address) which may be used for addressing, or the 16-bit assignable address which we will use allowing over 64,000 addresses on our network.
- Error Checking and Acknowledgements: The XBee uses a checksum to help ensure received data contains no errors. Acknowledgements are sent to the transmitting node to indicate proper reception. Up to 3 retries are performed by default if acknowledgements are not received.

Communication Modes

The XBee supports both an AT and API (Application Programming Interface) mode for sending and receiving data at your controller. Both have their advantages.

AT Mode

In AT Mode, also called Transparent Mode, just the message data itself is sent and received. The protocol link between the two is transparent to the end user and it appears to be a nearly direct serial link between the nodes as illustrated previously in Figure 1-4. This mode allows simple transmission and reception of serial data. AT Commands are used to configure the XBee, such as to send the data to a node with an address of 5: ATDL 5

AT Commands and examples will be explored in more depth later, but it requires placing the XBee into command mode, sending AT codes for configuration and exiting the command mode.

Note that even though the transmission and reception is the raw data, the message itself is passed between nodes encapsulated with needed information such as addressing and error checking bytes.

API Mode

In API Mode, the programmer packages the data with needed information, such as destination address, type of packet, and checksum value. Also, the receiving node accepts the data with information such as source address, type of packet, signal strength, and checksum value. The advantages are the user can build a packet that includes important data, such as destination address, and that the receiving node can pull from the packet information such as source address of the data. While more programming intensive, it allows the user greater flexibility and increased reliability in some cases.

Note that both sides do not need to be in the same mode. Data may be sent in API mode and received in AT mode or vice-versa. The mode defines the communications link between the PC/controller and the XBee modem. Data between XBees is always sent using the IEEE 802.15.4 LR-WPAN protocol. Examples using AT and API modes will be explored in later chapters.

XBee Module Styles

The XBee module comes in several versions but all have similar pinouts as shown in Figure 1-5. Differences between XBee versions include the power output, antennae style, operating frequency and networking abilities. The XBee is a 20-pin DIP module with pin spacing of 2 mm (0.079 in) as opposed to typical pin spacing of 2.54 mm (0.1 in). The XBee is available in two major versions and variants of those versions. This tutorial uses the XBee and XBee-Pro 802.15.4 (Series 1) versions exclusively.

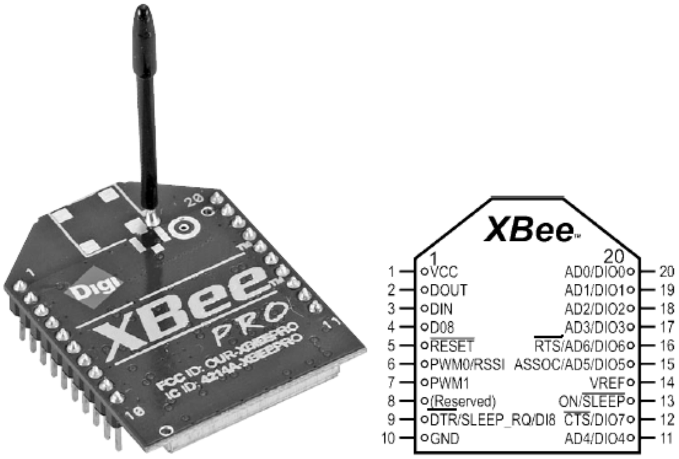

Figure 1-5: XBee-Pro Module and Pinouts

Don’t be alarmed by the number of pins—typically very few pins are used in our examples. Figure 1-6 illustrates a typical 3.3 V controller connection to the XBee. Table 1-1 is a brief discussion of the pins and their function on the XBee.

WAIT!

PLEASE SEE CHAPTER 2 BEFORE CONNECTING A 5V CONTROLLER, SUCH AS THE BASIC STAMP.

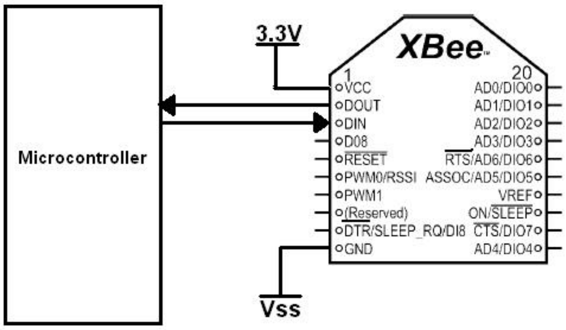

Figure 1-6: Typical Microcontroller Interfacing to the XBee

Table 1-1: XBee pin assignments

Pin	Name	Type	Function
1	VCC	P	2.8V – 3.4V
2	DOUT	O	Serial data output from XBee (received data)
3	DIN	I	Serial data input to XBee (data to transmit)
4	DO8	O	Digital data output 8
5	RESET	I	Reset module (low)
6	PWM0/ RSSI	O O	Pulse Width Modulated output Received Signal Strength Indication as PWM signal
7	PWM1	O	Pulse Width Modulated output
8	(Reserved)		
9	DTR SLEEP_RQ DI8	I I I	Data Terminal Ready handshaking for firmware updates (low) Sleep Request – A high places XBee in sleep mode when configured Digital Output 8
10	GND	G	Ground (Vss)
11	AD4 DIO4	A IO	Analog to Digital Input 4 Digital Input/Output 4
12	CTS DIO7	O IO	Clear to Send output for controller handshaking (low) Digital Input/Output 7
13	ON/SLEEP	O	Digital output, status indication: High = Awake, Low = Sleep
14	VREF	A	Analog to Digital reference voltage
15	ASSOC AD5 DIO5	O A IO	Associated indication when joining a network Analog to Digital Input 5 Digital Input/Output 5

16	RST AD6 IO6	I A IO	Ready to Send Handshaking input (Low) Analog to Digital Input 6 Digital Input/Output 6
17-20	AD3-AD0 DIO3-DIO0	A IO	Analog to Digital Input 3 to 0 Digital Input/Output 3 to 0

Pin Type: P = Power, G = Ground, I = Input, O = Output, A = Analog Input

A short discussion of pin groups will provide better understanding of use and features of the module:

- DOUT and DIN: These are the pins through which serial data is received by our controller or PC (DOUT) and sent to the XBee (Din). This data may be either for transmission between XBees or for setting and reading configuration information of the XBee. The default data rate is 9600 baud (bps) using asynchronous serial communications.
- RESET: A momentary low on this pin will reset the XBee to the saved configuration settings.
- CTS/RTS/DTR: (Marty, I still have trouble with this particular paragraph)These are used for handshaking between the XBee and your controller or the PC. Using RTS, the XBee will not send data out to DOUT unless the line is asserted (held low). Using CTS, the XBee can signal our controller when it is ready to receive more data. For the XBee, CTS is seldom needed since the XBee sends data much more quickly than it accepts data. DTR is typically used in communications with the PC when downloading new firmware to the XBee.
- DIO0–DIO7/D08: These are used as standard 3.3V digital inputs and outputs. The XBee can be controlled to set the state of the pins. They can also be used in “line passing” to control the outputs on one XBee with the inputs of another XBee.
- AD0 to AD6: These are 10-bit Analog to Digital Converter (ADC) inputs to the XBee. While we cannot directly read these values, they can be sent to another XBee to be read as an API packet by the controller. Some can also be used in “line passing” to control a PWM output on another module.
- RSSI: The XBee can report the strength of the received RF signal as PWM output on this pin. This value can also be retrieved using AT commands or as part of a packet in API mode.
- PWM0/1: These pins can be set for 10-bit pulse width modulated output, which can be used directly or filtered for analog output. They can also be controlled using “line passing” by the analog input on another XBee.
- ASSOC: When configured, the XBee can be set to join an existing network and assigned certain parameters. In this tutorial we will manually configure the XBee in the network instead of joining networks.

XBee & XBee Pro, 802.15.4 (Series 1)

The 802.15.4 style of XBee (commonly called Series 1) allows point-to-point networking, shown in Figure 1-7, and point-to-multipoint (one node to all nodes) networking. They use the IEEE 802.15.4 data link protocol to move data directly between 2 or more devices. All nodes on the network use the same firmware version though settings on various nodes may vary. The XBee and XBee-Pro are nearly identical in operation and function with the biggest differences being size and power. While the pinouts are the same, the casing of the XBee Pro is slightly longer. The power ratings and distances, along with other specifications, are listed in Table 1-2.

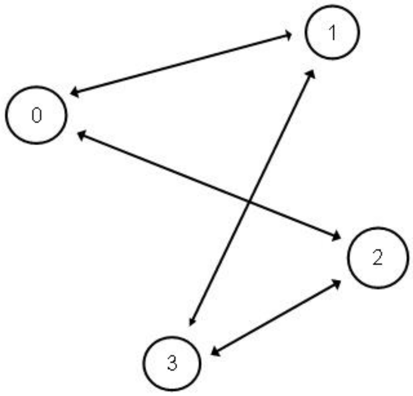

Figure 1-7: Point-to-Point Networking

Table 1-2: XBee and XBee-Pro Specifications

Specification	XBee	XBee-Pro
Supply Voltage	2.8 VDC – 3.4 VDC	2.8 VDC – 3.4 VDC
RF Power	0 dBm, 1 mW	18 dBm, 63 mW
Outdoor Distance (LOS)	300 ft (90 m)	1 mile (1.6 km)
Indoor Distance	100 ft (30 m)	300 ft (90 m)
Current Draw – Receive	45 mA	50 mA
Current Draw – Transmit	50 mA	215 mA
Current Draw – Sleep	< 10 μA	<10 μA
RF Data Throughput	250 kbps	250 kbps
Operating Frequency	2.4 GHz, 16 Channels	2.4 GHz, 12 Channels
Receiver Sensitivity	-92 dBm	-100 dBm

Don’t get too excited about the distances.

RF transmissions have many factors including line-of-sight (LOS) issues. We will discuss RF issues shortly.

XBee-Pro 900 and 868

These 900 MHz (U.S.) and 868 MHz (Europe) modems operate at lower frequencies with slower data throughput (156 kbps) but much longer range (6 miles or 10 km). Please visit www.digi.com for more information on these modules.

XBee ZB and XBee ZB-Pro (Series 2)

The XBee ZB modules extend the range of the network through routing. They form self-establishing, self-healing networks for moving data across the network as shown in Figure 1-8. While these modems use the IEEE-802.15.4 protocol for point-to-point communications, higher protocols are placed on top of these for network routing. There are two styles of mesh networking protocols available: ZigBee and DigiMesh. Devices can be programmed with either protocol in either AT or

API versions. Depending on the function of the device, the correct firmware version must be loaded. In a mesh network, such as ZigBee, devices have one of three different duties:

- Coordinator: Establishes and maintains the network by assigning addresses to joining (associated) devices and assisting with route building.
- Routers: move data between nodes that cannot communicate directly.
- Endpoints: The node that collects data and controls devices on the network and is typically connected to our controllers, sensors and other devices for network interfacing.

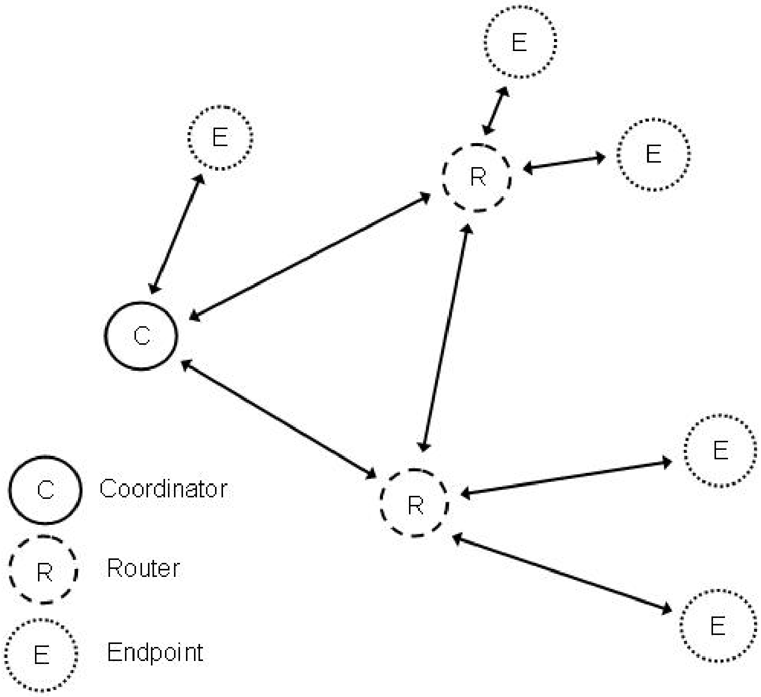

Figure 1-8: Nodes in a Mesh Network

RF Transmissions

While distance specifications, such as 1 mile, sound great, there are many factors that can affect the transmission distance including absorption, reflection and scattering of waves, line-of-sight issues, antennae style and frequency. The sensitivity of the receiver allows it receive and use signals as low at -100dBm, which is 0.1 picowatts. The RF signal uses Direct-Sequence Spread Spectrum, spreading the signal out over the frequency spectrum greatly improving the signal-to-noise ratio (SNR). The XBee reports the RSSI level (Receiver Signal Strength Indication) in several ways so that you may monitor the strength of your signal:

- The PWM0/RSSI output produces a pulse width providing RSSI level indication.
- In Command Mode, the unit may polled by using the ATDB command.
- When receiving in API mode, a byte in the packet contains the RSSI level.

We will explore monitoring the strength of your signal later in the tutorial.

Absorption and Reflection

Radio frequency waves can be absorbed and reflect causing loss of power through objects and over a distance. Metal is a great reflector of RF energy, but almost any surface can cause a reflection of waves causing loss of signal strength and sometimes interference with other waves. Absorption is also a major factor in power loss. Absorption can be also be a major factor, and sometimes useful—A microwave oven heats food by the water absorbing the energy and transferring the RF energy to heat. In fact, microwave ovens operate around 2.4 GHz, showing how susceptible to absorption RF energy at this frequency is. Across a distance water molecules in the air will cause absorption of our signal as will foliage of trees and other plants. While our signal may not be affected by a normal rainfall, heavy downpours can affect the signal when rain is dense enough. Indoors, walls and other objects will cause absorption and reflection as well, limiting the transmission distance to penetrating just a few walls.

Line-of-Sight Issues

Line-of-Sight (LOS) can help increase reliability of the signal. If there are no obstructions, such as trees, building and walls in the way absorbing or reflecting the signal, the power at the receiver will be greater. This is easy to plan for. A more difficult aspect is reflection and deconstructive interference from wave reflecting due to nearby objects or the ground. Even with line-of-sight, reflections from the ground or other surfaces will cause waves to reflect, join with other waves at the receiver and cause errors in the reception. The XBee’s antennas are omni-directional meaning that the radio frequencies are radiated in all directions. A few go straight to the receiver and others go in variety of directions. While the XBee-Pro may send a transmission of 60 milliwatts, only a very small amount of this power is directed at the receiver. Reflections of some of the other waves will arrive slower and out of phase at the receiver interfering with the quality of the signal as shown in Figure 1-9.

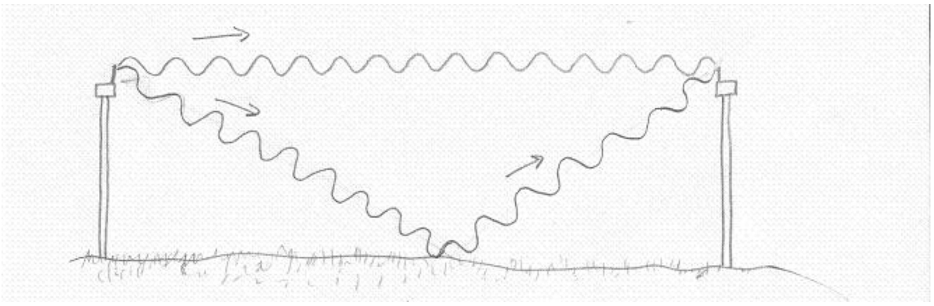

Figure 1-9: Reflected waves causing problems at receiver

For best transmission, the area needed between the transmitter and receiver can be pictured as a football shaped zone, called the Fresnel Zone. The further the separation distance between the nodes, the wider the radius of this zone needs to be as shown in Figure 1-10.

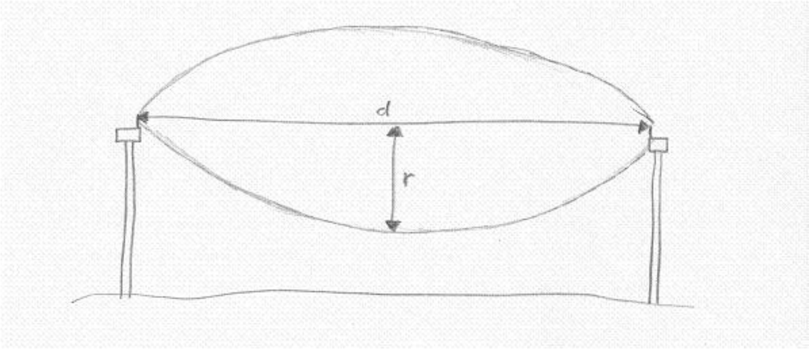

Figure 1-10: Fresnel Zone between nodes

The radius of this zone can be calculated using the following equations depending on your units of choice:

$$r_m = 17.32\sqrt{\frac{d_{Km}}{4 f_{Ghz}}} \quad \text{or} \quad r_{ft} = 72.05\sqrt{\frac{d_{Miles}}{4 f_{Ghz}}}$$

For a distance of 300 feet or 0.058 miles (300 ft / 5280 ft), the radius of the zone is approximately 5.6 feet for good signal quality. There are higher order zones further out, but clearance in this zone will typically allow sufficient quality at the receiver.

But RF can be a tricky thing. In testing, we have had two units 50 feet apart and 5 feet off the ground and experienced low signal strength. Moving one node a foot back or forward provided excellent transmission. At the initial position a reflection from a higher order zone was just right in damaging the signal at the receiver. A small shift in position cleared up the problem. Of course reflections occur not only off the ground, but any surface in the area.

Operating Frequency

As was shown in Table 1-2, the XBee has multiple frequency channels it can operate on. This allows multiple networks in the same area to operate on unique frequencies limiting contention of nodes for the same frequency. It also allows the user to select a frequency that may have less noise on it than other channels from other sources in the 2.4 GHz spectrum such as WiFi networks, cordless phones and microwave ovens. Using AT commands, the selected operating channel may be selected. Chapter 3 will provide examples showing channel selection and monitoring.

Antennae Style

The XBee comes with a variety of antennae styles providing differing amounts of gain and use. Two of the most popular versions are the chip antennae and whip antennae styles shown in Figure 1-11. The whip style has about 20% better range than the chip style. Other styles include U.FL and RPSMA connectors for attaching higher-gain antennas.

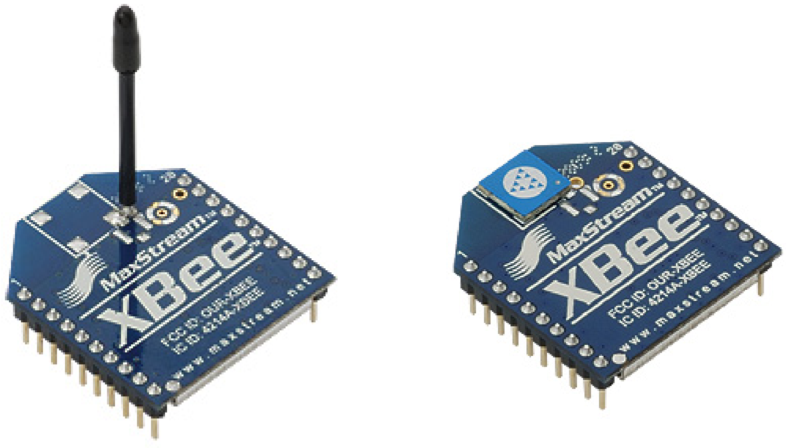

Figure 1-11: Whip (left) and Chip (right) Antennae Modules

Other XBee Features

Besides simply passing data between nodes for microcontrollers, the XBee has other features that can aid in process monitoring and control. We will explore some of these in the course of the tutorial.

Digital & Analog I/O

As discussed in the XBee pinout section, the modules have digital I/O, analog input and PWM output that may be used in a variety of ways:

- Direct digital output control: Through AT commands, the I/O may be set to digital output and controlled as high or low. For example, the command "ATD0 4" would set D0 to be a low output and "ATD0 5" would be a high output.
- Digital and Analog input for transmission: The inputs may also be set to be digital inputs or 10-bit analog to digital input. Using sampling, the values of the inputs are sent as data to a receiving XBee in API mode where the ADC data is extracted.
- PWM/analog output: The 10-bit PWM value of an output may be set and optionally filtered for analog output of the unit.
- Line Passing: The digital inputs can control a digital output on a remote unit, and analog input can control a PWM output on a remote unit.

Remote Configuration

Using AT commands, the configuration of a remote module may be changed by sending the command in an API packet. This could be used to control remote digital or PWM outputs, or change other parameters of the remote XBee.

Summary

The XBee is a feature-rich RF module for use on a wireless sensor network. The IEEE 802.15.4 protocol greatly reduces the work of the programming in ensuring data communications. The XBee has many other features for use in a WSN beyond its networking ability. Now that you have a better understanding about the XBee's features and uses, we will look at means of interfacing the RF modem to your microcontroller and showing examples of use.