XBee-PRO 900 DigiMesh RF Module

User Guide

XBee-PRO 900 DigiMesh RF Module User Guide

(Part number 90000903 G)

Revision	Date	Description
A	August 2008	Baseline release of the document.
B	September 2008	Updated.
C	September 2009	Updated to support DigiMesh 900 firmware.
D	December 2010	Firmware updates.
E	July 2011	Reduced this manual to the DigiMesh version, moved multipoint information to its own manual.
F	July 2013	0x90 frame, first byte of 64- bit address was missing.
G	March 2015	Updated the template. Updated the warranty information. Added language on RF optimization services.

Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. Digi provides this document "as is," without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

Trademarks and copyright

Digi, Digi International, and the Digi logo are trademarks or registered trademarks in the United States and other countries worldwide. All other trademarks mentioned in this document are the property of their respective owners.

Customer support

Telephone (8:00 am — 5:00 pm CST):

US & Canada: 866.765.9885

Worldwide: 801.765.9885

Online: www.digi.com/support/eservice

Mail:

Digi International
11001 Bren Road East
Minnetonka, MN 55343
USA

Warranty

View the product's warranty online: http://www.digi.com/howtobuy/terms

Related publications

The following table lists the related publications for the XBee-PRO® 900 DigiMesh RF Module. These publications are available on Digi's website.

Publication Name	Publication Number
XBee & XBee-PRO DigiMesh Development Kit Getting Started Guide	90001110

Contents

RF Modules

RF Module operation

Advanced application features

Sleep Mode

XBee-PRO 900 DigiMesh

Command Reference Tables

API operation

Definitions

Agency certifications

RF Modules

The XBee-PRO® 900 DigiMesh RF Module was engineered to support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices.

The modules operate within the ISM 900 MHz frequency band.

Easily build networks up to 128 nodes using the XBee-PRO 900 DigiMesh RF Module. For larger networks up to 1,000+ nodes, we offer RF optimization services to assist with proper network configuration. Contact Digi Technical Support for more details.

Key features

High performance, low cost	Low power
• Indoor/urban: up to 450 ft (140 m) • Outdoor line-of-sight: up to 1.8 miles (3 km) • Transmit power output: 50 mW (+17 dBm) EIRP • Receiver sensitivity: -100 dBm • RF data rate: 156 kb/s	• TX Current: 210 mA (@3.3 V) • RX Current: 80 mA (@3.3 • Asynchronous. sleep current: 48 µA (typical @3.3 V) Synchronous sleep current: 60 µA (typical @3.3 V)
Advanced networking and security	**Easy to use**
Retries and acknowledgments Supports point-to-point, point-to-multipoint and peer-to-peer topologies	No configuration necessary for out-of box RF communications AT and API Command Modes for configuring module parameters Small form factor Extensive command set Free XCTU software (testing and configuration software)

Worldwide acceptance

FCC Approval (USA) Refer to the Government Agency Certification webpage. Systems that contain XBee-PRO 900 DigiMesh inherit Digi Certifications.

Industrial, Scientific and Medical (ISM) **900 frequency band**.

Manufactured under **ISO 9001:2000** registered standards

XBee-PRO RF Modules are optimized for use in **US** and **Canada**. Refer to the Government Agency Certification webpage for a complete list of agency approvals.

Specifications

Table 1: Specifications of the XBee-PRO 900 DigiMesh

Specification	XBee-PRO
Performance	
Indoor / urban range	up to 450 ft (140 m)
Outdoor RF line-of-sight range	up to 1.8 miles (3 km) up to 6 miles (10 km) w/high gain antenna
Transmit power output	+17 dBm (50 mW)
Interface data rate	Up to 230 kb/s software selectable
RF data rate	156.25 kb/s
Data throughput	Up to 87000 b/s
Receiver sensitivity	-100 dBm (1% packet error rate)
Power Requirements	
Supply voltage	3.0 to 3.6 VDC
Transmit current	210 mA (@ 3.3 V)
Idle / receive current	80 mA (@ 3.3 V)
Sleep current (asynchronous)	48 µA typical @ 3.3 V
Sleep current (synchronous)	60 µA typical @ 3.3 V
General	
Operating frequency band	Standard variant: 902-928 MHz (ISM) International variant: 916-928 MHz
Dimensions	0.960" x 1.297" (2.438 cm x 3.294 cm)
Operating temperature	-40 to 85° C (industrial), 0 to 95% non-condensing
Antenna options	1/4 wave wire antenna, RPSMA RF connector, U.Fl RF connector
Digital I/O	13 I/O lines
ADC	Six 10-bit analog inputs
Networking and security	
Supported network topologies	Mesh, Point-to-point, point-to-multipoint, peer-to-peer
Number of channels (software selectable)	Eight hopping patterns on 12 channels (standard variant), four hopping patterns on five channels (international variant)
Addressing options	PAN ID, Channel and 64-bit addresses
Encryption	128 bit AES

Table 1: Specifications of the XBee-PRO 900 DigiMesh

Specification	XBee-PRO
Agency approvals	
United States (FCC Part 15.247)	MCQ-XBEE09P
Industry Canada (IC)	1846A-XBEE09P
Europe (CE)	N/A
RoHS	Lead-free and RoHS compliant
Australia	C-tick

Mechanical drawings of the XBee-PRO 900 DigiMesh RF Module

Figure 1: Mechanical drawings of the XBee-PRO 900 DigiMesh RF Module RF Module (antenna options not shown)

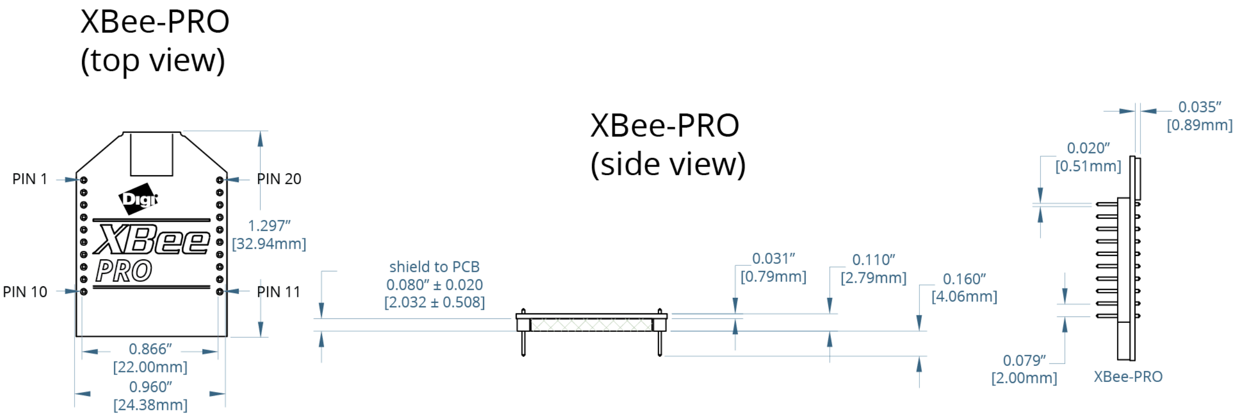
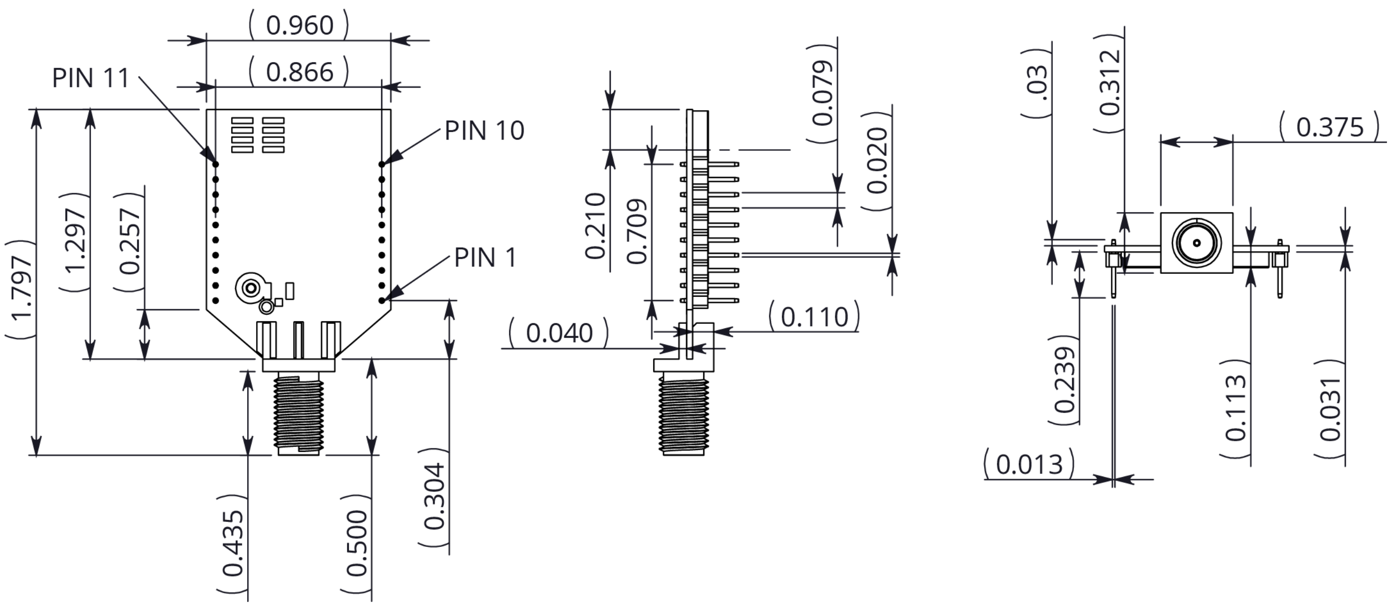


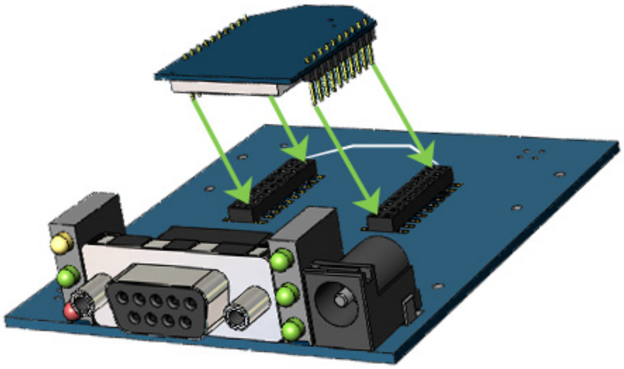
Figure 2: Mechanical drawings for the RPSMA variant



Mounting considerations

The XBee-PRO RF Module (through-hole) was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

Figure 3: XBee-PRO 900 DigiMesh mounting to an RS-232 interface board.



The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

- Through-hole single-row receptacles - Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles - Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles - Samtec P/N: SMM-110-02-SM-S

We also recommend printing an outline of the module on the board to indicate the orientation the module should be mounted.

Pin signals

Table 2: Pin assignments for the XBee-PRO 900 DigiMesh. Low-asserted signals are distinguished with a horizontal line above signal name.

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / $\overline{\text{CONFIG}}$	Input	UART Data In
4	DIO12	Either	Digital I/O 12
5	$\overline{\text{RESET}}$	Input / open drain output	Module Reset (reset pulse must be at least 100 us. This must be driven as an open drain/collector. The module will drive this line low when a reset occurs. This line should never be driven high.)
6	PWM0 / RSSI / DIO10	Either	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM / DIO11	Either	PWM Output 1 / Digital I/O 11
8	[reserved]	-	Do not connect

Table 2: Pin assignments for the XBee-PRO 900 DigiMesh. Low-asserted signals are distinguished with a horizontal line above signal name.

Pin #	Name	Direction	Description
9	$\overline{\text{DTR}}$ / SLEEP_RQ/ DIO8	Either	Pin Sleep Control Line or Digital IO 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	$\overline{\text{CTS}}$ / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / $\overline{\text{SLEEP}}$	Output	Module Status Indicator or Digital I/O 9
14	VREF	-	This line must be connected if analog IO sampling is desired. Must be between 2.6 V and Vcc.
15	Associate / DIO5 / AD5	Either	Associated Indicator, Digital I/O 5
16	$\overline{\text{RTS}}$ / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Either	Analog Input 0, Digital IO 0, or Commissioning Button

Design notes

The XBee modules do not specifically require any external circuitry or specific connections for proper operation. However, there are some general design guidelines that are recommended for help in troubleshooting and building a robust design.

Power supply design

Poor power supply can lead to poor radio performance especially if the supply voltage is not kept within tolerance or is excessively noisy. To help reduce noise a 1.0 uF and 8.2pF capacitor are recommended to be placed as near to pin1 on the PCB as possible. If using a switching regulator for your power supply, switching frequencies above 500kHz are preferred. Power supply ripple should be limited to a maximum 100mV peak to peak. To ensure proper power up, Vcc SLOPE must be superior or equal to 6V/ms.

Recommended pin connections

The only required pin connections are VCC, GND, DOUT and DIN. To support serial firmware updates, VCC, GND, DOUT, DIN, RTS, and DTR should be connected.

All unused pins should be left disconnected. All inputs on the radio can be pulled high with internal pull-up resistors using the PR software command. No specific treatment is needed for unused outputs.

Other pins may be connected to external circuitry for convenience of operation including the Associate LED pin (pin 15) and the commissioning button pin (pin 20). The Associate LED pin will flash differently depending on the state of the module, and a pushbutton attached to pin 20 can enable various deployment and troubleshooting functions without having to send UART commands.

The combined source and sink capabilities of the module are limited to 120mA for all pins on the module. Module pins 11 and 15 can source/sink a maximum of 2mA; pins 9, 6, and 13 can source/ sink a maximum of 16mA; and all other pins can source/sink a maximum of 8mA.

If analog sampling is desired the VRef pin (pin 14) should be attached to a voltage reference.

Board layout

XBee modules are designed to be self sufficient and have minimal sensitivity to nearby processors, crystals or other PCB components. As with all PCB designs, Power and Ground traces should be thicker than signal- traces and able to comfortably support the maximum current specifications. No other special PCB design considerations are required for integrating XBee radios except in the antenna section.

Antenna performance

Antenna location is an important consideration for optimal performance. In general, antennas radiate and receive best perpendicular to the direction they point. Thus a vertical antenna's radiation pattern is strongest across the horizon. Metal objects near the antenna may impede the radiation pattern. Metal objects between the transmitter and receiver can block the radiation path or reduce the transmission distance, so antennas should be positioned away from them when possible. Some objects that are often overlooked are metal poles, metal studs or beams in structures, concrete (it is usually reinforced with metal rods), vehicles, elevators, ventilation ducts, refrigerators, microwave ovens, batteries, and tall electrolytic capacitors. If the XBee is to be placed inside a metal enclosure, an external antenna should be used.

Electrical characteristics

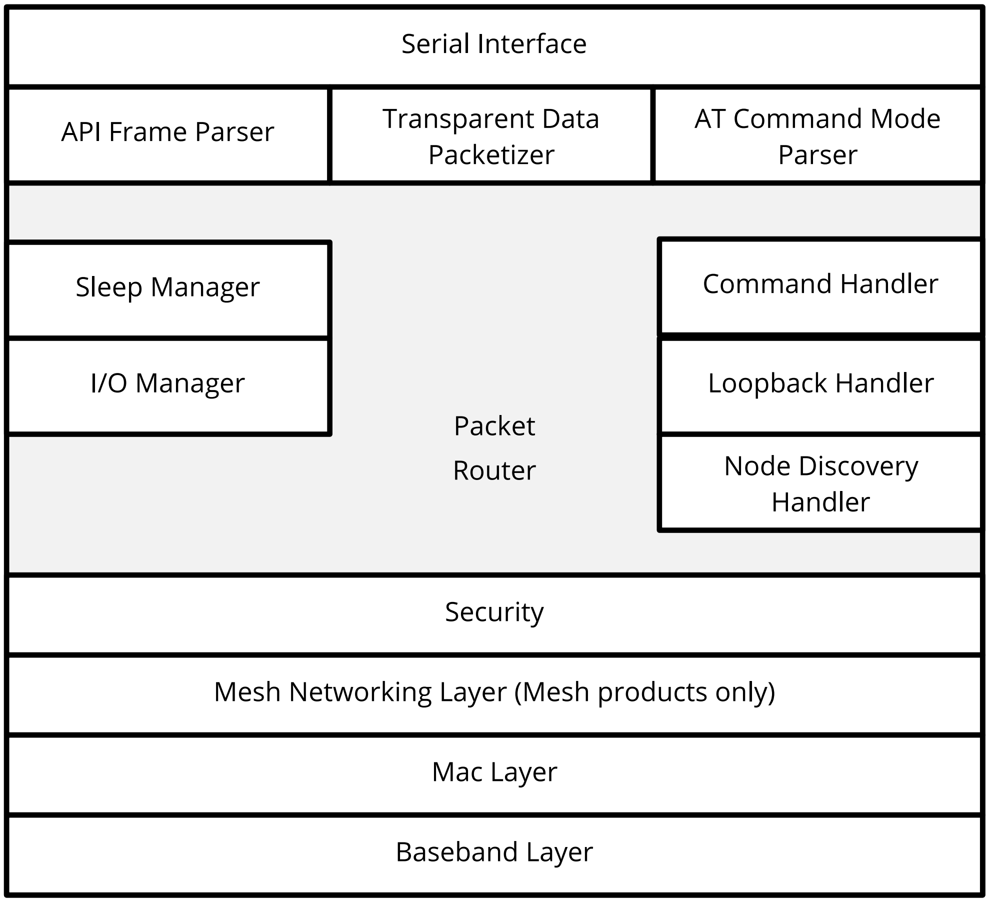
Table 3: DC characteristics of the XBee-PRO (VCC =3.0-3.6VDC).

Symbol	Parameter	Condition	Min	Typical	Max	Units
VIL	Input low voltage	All digital inputs	-	-	0.2 VCC	V
V_{IH}	Input high voltage	All digital inputs	0.8 * VCC	-	-	V
V_{OL}	Output low voltage	I_{OL} = 2 mA, VCC >= 3.0 V	-	-	0.18 VCC	V
V_{OH}	Output high voltage	I_{OH} = 2 mA, VCC >= 3.0 V	0.82 *VCC	-	-	V
II_{IN}	Input leakage current	V_{IN} = VCC or GND, all inputs, per pin	-	-	0.5	µA

RF Module operation

Overview

The XBee module provides a serial interface to an RF link. The XBee module can convert serial data to RF data that can be sent to any device in an RF network. In addition to RF data communication devices, the XBee module provides a software interface for interacting with a variety of peripheral functions, including I/O sampling, commissioning and management devices. The following diagram illustrates the functionality of the XBee module.



Serial Communications

The XBee /XBee-PRO ZNet 2.5s interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device; for example: Through a Digi proprietary RS-232 or USB interface board.

UART data flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

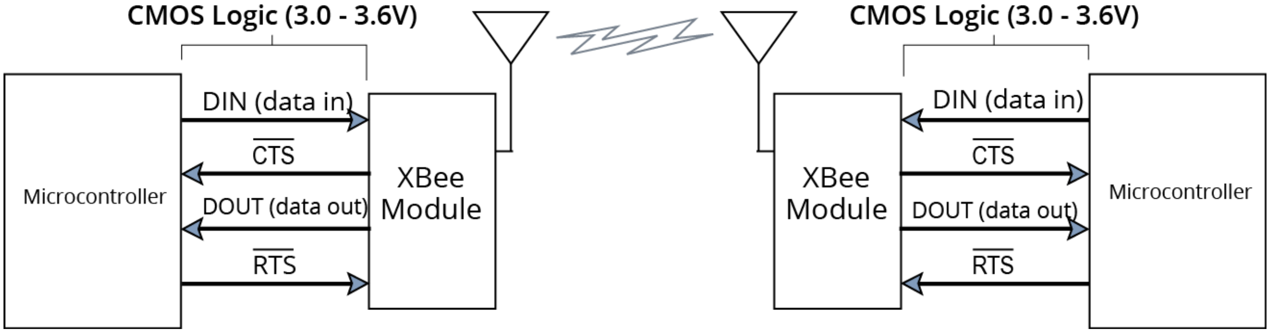
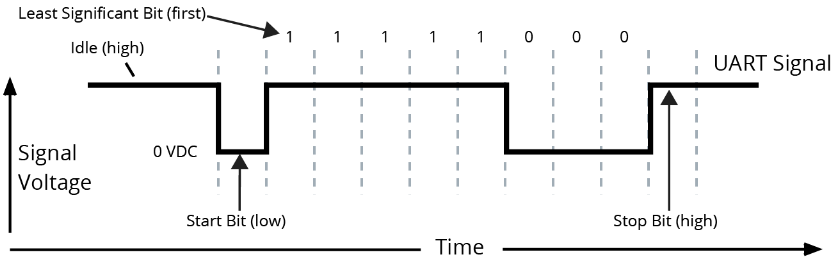


Figure 4: System data flow diagram in a UART-interfaced environment. Low-asserted signals are distinguished with horizontal line over signal name.

Serial data

Data enters the module UART through the DIN (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

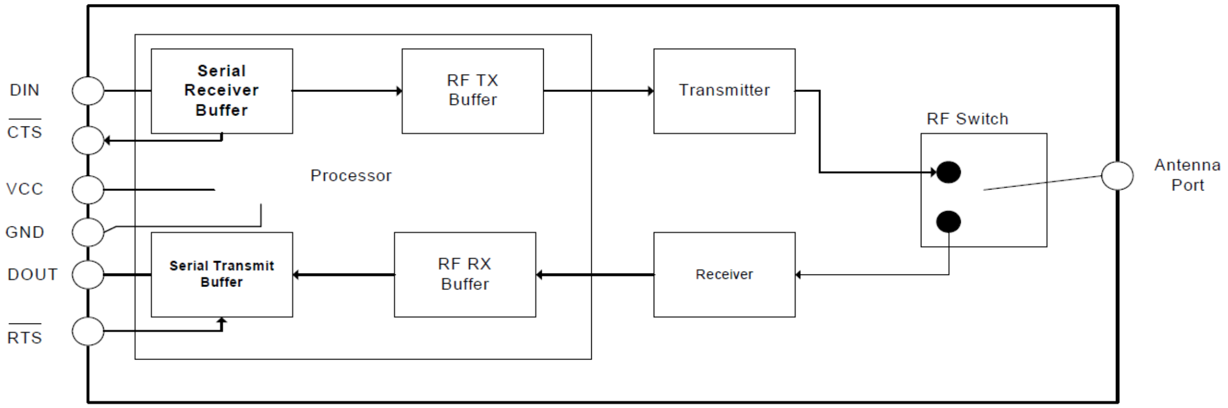


The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

Serial buffers

The XBee-PRO modules maintain buffers to collect received serial and RF data, which is illustrated in the figure below. The serial receive buffer collects incoming serial characters and holds them until they can be processed. The serial transmit buffer collects data that is received via the RF link that will be transmitted out the UART.

Figure 5: Internal data flow diagram



Serial receive buffer

When serial data enters the RF module through the DIN Pin (pin 3), the data is stored in the serial receive buffer until it can be processed. Under certain conditions, the module may not be able to process data in the serial receive buffer immediately. If large amounts of serial data are sent to the module, CTS flow control may be required to avoid overflowing the serial receive buffer.

Cases in which the serial receive buffer may become full and possibly overflow:

1. If the module is receiving a continuous stream of RF data, the data in the serial receive buffer will not be transmitted until the module is no longer receiving RF data.
2. If the module is transmitting an RF data packet, the module may need to discover the destination address or establish a route to the destination. After transmitting the data, the module may need to retransmit the data if an acknowledgment is not received, or if the transmission is a broadcast. These issues could delay the processing of data in the serial receive buffer.

Serial transmit buffer

When RF data is received, the data is moved into the serial transmit buffer and sent out the UART. If the serial transmit buffer becomes full enough such that all data in a received RF packet won't fit in the serial transmit buffer, the entire RF data packet is dropped.

Cases in which the serial transmit buffer may become full resulting in dropped RF packets:

1. If the RF data rate is set higher than the interface data rate of the module, the module could receive data faster than it can send the data to the host. Even occasional transmissions from a large number of modules can quickly add up and overflow the transmit buffer.
2. If the host does not allow the module to transmit data out from the serial transmit buffer because of being held off by hardware flow control.

Serial flow control

The RTS and CTS module pins can be used to provide RTS and/or CTS flow control. CTS flow control provides an indication to the host to stop sending serial data to the module. RTS flow control allows the host to signal the module to not send data in the serial transmit buffer out the UART. RTS and CTS flow control are enabled using the D6 and D7 commands.

$\overline{\text{CTS}}$ flow control

If $\overline{\text{CTS}}$ flow control is enabled (D7 command), when the serial receive buffer is filled with FT bytes, the module de-asserts $\overline{\text{CTS}}$ (sets it high) to signal to the host device to stop sending serial data. $\overline{\text{CTS}}$ is re-asserted when less than FT - 16 bytes are in the UART receive buffer.

$\overline{\text{RTS}}$ flow control

If $\overline{\text{RTS}}$ flow control is enabled (D6 command), data in the serial transmit buffer will not be sent out the DOUT pin as long as $\overline{\text{RTS}}$ is de-asserted (set high). The host device should not de-assert $\overline{\text{RTS}}$ for long periods of time to avoid filling the serial transmit buffer. If an RF data packet is received, and the serial transmit buffer does not have enough space for all of the data bytes, the entire RF data packet will be discarded.

Serial interface protocols

The XBee modules support both transparent and Application Programming Interface (API) serial interfaces.

Transparent operation

When operating in transparent mode, the modules act as a serial line replacement. All UART data received through the DIN pin is queued up for RF transmission. When RF data is received, the data is sent out through the DOUT pin. The module configuration parameters are configured using the AT command mode interface.

Data is buffered in the serial receive buffer until one of the following causes the data to be packetized and transmitted:

- No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.
- The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the serial receive buffer before the sequence is transmitted.
- The maximum number of characters that will fit in an RF packet is received.

API operation

API operation is an alternative to transparent operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit data frames (received through the DIN pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive data frames (sent out the DOUT pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, sync status, and so forth

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, and payload information from received data packets.

The API operation option facilitates many operations such as the examples cited below:

- Transmitting data to multiple destinations without entering Command Mode
- Receive success/failure status of each transmitted RF packet
- Identify the source address of each received packet

Comparing Transparent and API Operation

The following table compares the advantages of transparent and API modes of operation:

Transparent Operation Features	
Simple Interface	All received serial data is transmitted unless the module is in command mode.
Easy to support	It is easier for an application to support transparent operation and command mode
API Operation Features	
Easy to manage data transmissions to multiple destinations	Transmitting RF data to multiple remotes only requires changing the address in the API frame. This process is much faster than in transparent operation where the application must enter AT command mode, change the address, exit command mode, and then transmit data. Each API transmission can return a transmit status frame indicating the success or reason for failure.
Received data frames indicate the sender's address	All received RF data API frames indicate the source address.
Advanced addressing support	API transmit and receive frames can expose addressing fields including source and destination endpoints, cluster ID and profile ID.
Advanced networking diagnostics	API frames can provide indication of IO samples from remote devices, and node identification messages.
Remote Configuration	Set / read configuration commands can be sent to remote devices to configure them as needed using the API.

Generally, we recommend using API firmware when a device:

- sends RF data to multiple destinations
- sends remote configuration commands to manage devices in the network
- receives IO samples from remote devices
- receives RF data packets from multiple devices, and the application needs to know which device sent which packet.

If the above conditions do not apply (i.e. a sensor node, router, or a simple application), then AT firmware might be suitable. It is acceptable to use a mixture of devices running API and AT firmware in a network.

To implement API operations, refer to API operation on page 67.

Modes of operation

Idle Mode

When not receiving or transmitting data, the RF module is in Idle Mode. During Idle Mode, the RF module is also checking for valid RF data. The module shifts into the other modes of operation under the following conditions:

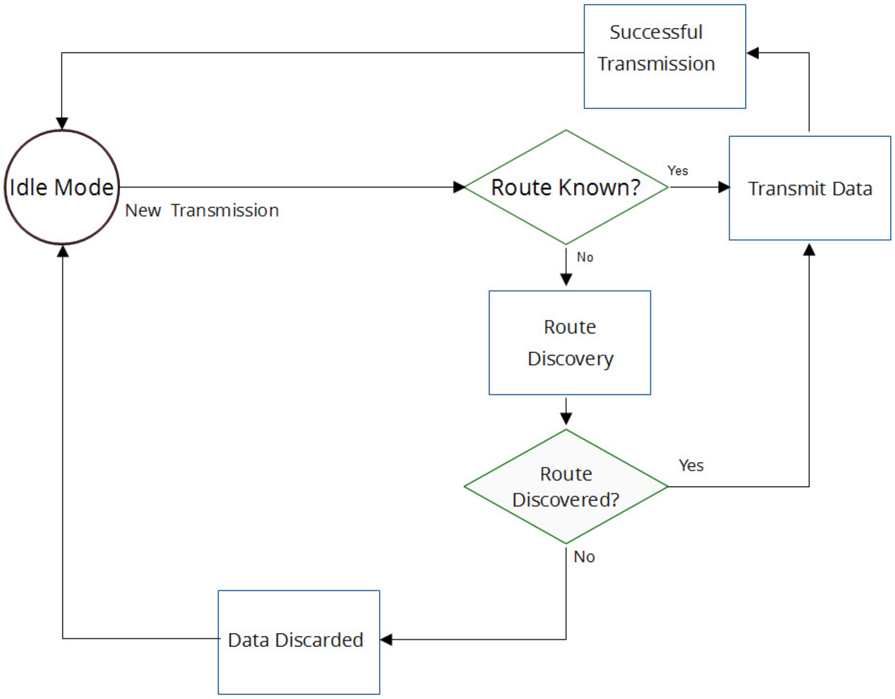
- Transmit Mode (serial data in the serial receive buffer is ready to be packetized)
- Receive Mode (valid RF data is received through the antenna)
- Command Mode (Command Mode sequence is issued)
- Sleep Mode (a device is configured for sleep)

Transmit Mode

When serial data is received and is ready for packetization, the RF module will exit Idle Mode and attempt to transmit the data. The destination address determines which node(s) will receive the data.

For mesh firmware, if a route is not known, route discovery will take place for the purpose of establishing a route to the destination node. If a module with a matching network address is not discovered, the packet is discarded. The data will be transmitted once a route is established. Route discovery will be attempted only once per packet.

Figure 6: Transmit Mode sequence



When data is transmitted from one node to another, a network-level acknowledgment is transmitted back across the established route to the source node. This acknowledgment packet indicates to the source node that the data packet was received by the destination node. If a network acknowledgment is not received, the source node will re-transmit the data.

See Data transmission and routing on page 48 for more information.

Receive Mode

If a valid RF packet is received, the data is transferred to the serial transmit buffer.

Command Mode

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming serial characters are interpreted as commands. Refer to the API operation on page 67 for an alternate means of configuring modules.

AT Command Mode

Entering AT Command Mode

Send the 3-character command sequence "+++" and observe guard times before and after the command characters. [Refer to the "Default AT Command Mode Sequence" below.]

Default AT Command Mode Sequence (for transition to Command Mode):

- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters ("+++") within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

Once the AT command mode sequence has been issued, the module sends an "OK\r" out the DOUT pin. The "OK\r" characters can be delayed if the module has not finished transmitting received serial data.

When command mode has been entered, the command mode timer is started (CT command), and the module is able to receive AT commands on the DIN pin.

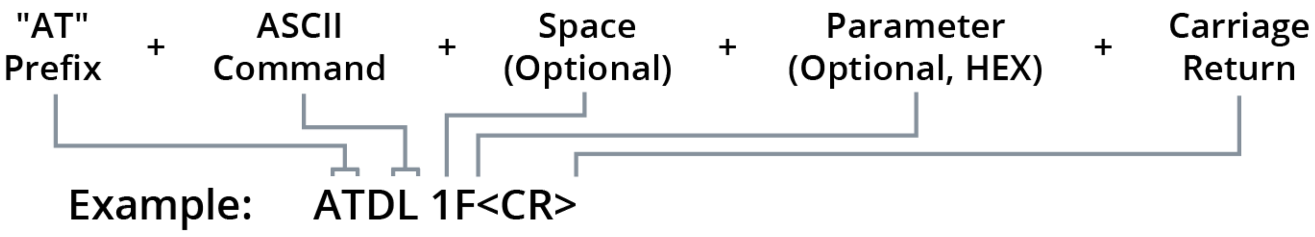
All of the parameter values in the sequence can be modified to reflect user preferences.

Note Failure to enter AT Command Mode is most commonly due to baud rate mismatch. By default, the BD (Baud Rate) parameter = 3 (9600 b/s).

Sending AT commands

Send AT commands and parameters using the syntax shown below.

Figure 7: Syntax for sending AT commands



To read a parameter value stored in the RF module's register, omit the parameter field.

The preceding example would change the RF module Destination Address (Low) to "0x1F". To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

For modified parameter values to persist in the module's registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset.

Command response

When a command is sent to the module, the module will parse and execute the command. Upon successful execution of a command, the module returns an "OK" message. If execution of a command results in an error, the module returns an "ERROR" message.

Applying command changes

Any changes made to the configuration command registers through AT commands will not take effect until the changes are applied. For example, sending the BD command to change the baud rate will not change the actual baud rate until changes are applied. Changes can be applied in one of the following ways:

- The AC (Apply Changes) command is issued.
- AT command mode is exited.

Exiting AT Command Mode

1. Send the ATCN (Exit Command Mode) command (followed by a carriage return).

   [OR]

2. If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the RF module automatically returns to Idle Mode.

For an example of programming the RF module using AT Commands and descriptions of each configurable parameter, refer to Command Reference Tables on page 52.

Sleep Mode

Sleep modes allow the RF module to enter states of low power consumption when not in use. The XBee /XBee-PRO ZNet 2.5s support both pin sleep (sleep mode entered on pin transition) and cyclic sleep (module sleeps for a fixed time). The XBee DigiMesh modules support a network synchronized sleep to conserve power. Sleep modes on page 37 discusses the sleep modes in detail.

Note For applications that need to ensure the lowest possible sleep current, inputs should never be left floating. Use internal or external pull-up or pull-down resistors, or set the unused I/O lines to outputs. For minimum sleep current, you can leave the I/O settings at default (disabled) with the exception of D9. D9 is not disabled by default and must be disabled (D9=0) to ensure minimum sleep current.

Advanced application features

Remote configuration commands

A module in API mode has provisions to send configuration commands to remote devices using the Remote Command Request API frame; see API operation on page 67. This API frame can be used to send commands to a remote module to read or set command parameters.

Sending a remote command

To send a remote command, the Remote Command Request frame should be populated with the 64-bit address of the remote device, the correct command options value, and the command and parameter data (optional). If a command response is desired, the Frame ID should be set to a non-zero value. Only unicasts of remote commands are supported. Remote commands cannot be broadcast.

Applying changes on remote

When remote commands are used to change command parameter settings on a remote device, parameter changes do not take effect until the changes are applied. For example, changing the BD parameter will not change the actual serial interface rate on the remote until the changes are applied. Changes can be applied using remote commands in one of three ways:

- Set the apply changes option bit in the API frame
- Issue an AC command to the remote device
- Issue a WR + FR command to the remote device to save changes and reset the device.

Remote command responses

If the remote device receives a remote command request transmission, and the API frame ID is non-zero, the remote will send a remote command response transmission back to the device that sent the remote command. When a remote command response transmission is received, a device sends a remote command response API frame out its UART. The remote command response indicates the status of the command (success, or reason for failure), and in the case of a command query, it will include the register value.

The device that sends a remote command will not receive a remote command response frame if:

- The destination device could not be reached
- The frame ID in the remote command request is set to 0.

Network commissioning and diagnostics

Network commissioning is the process whereby devices in a network are discovered and configured for operation. The XBee modules include several features to support device discovery and configuration. In addition to configuring devices, a strategy must be developed to place devices to ensure reliable routes.

To accommodate these requirements, the XBee modules include various features to aid in device placement, configuration, and network diagnostics.

Device configuration

XBee modules can be configured locally through serial commands (AT or API), or remotely through remote API commands. API devices can send configuration commands to set or read the configuration settings of any device in the network.

Network link establishment and maintenance

Building aggregate routes

In many applications it is necessary for many or all of the nodes in the network to transmit data to a central aggregator node. In a new DigiMesh network the overhead of these nodes discovering routes to the aggregator node can be extensive and taxing on the network. To eliminate this overhead the AG command can be used to automatically build routes to an aggregate node in a DigiMesh network.

To send a unicast, modules configured for transparent mode (AP=0) must set their DH/DL registers to the MAC address of the node to which they need to transmit to. In networks of transparent mode modules which transmit to an aggregator node it is necessary to set every module's DH/DL registers to the MAC address of the aggregator node. This can be a tedious process.

Upon deploying a DigiMesh network the AG command can be issued on the desired aggregator node to cause all nodes in the network to build routes to the aggregator node. The command can optionally be used to automatically update the DH/DL registers to match the MAC address of the aggregator node. The AG command requires a 64-bit parameter. The parameter indicates the current value of the DH/DL registers on a module which should be replaced by the 64-bit address of the node sending the AG broadcast. If it is not desirable to update the DH/DL of the module receiving the AG broadcast then the invalid address of 0xFFFE can be used. API enabled modules will output an Aggregator Update API frame if they update their DH/DL address (see the API section of this manual for a description of the frame). All modules which receive an AG broadcast will update their routing table information to build a route to the sending module, regardless of whether or not their DH/DL address is updated. This routing information will be used for future transmissions of DigiMesh unicasts.

Example 1: To update the DH/DL registers of all modules in the network to be equal to the MAC address of an aggregator node with a MAC address of 0x0013a2004052c507 after network deployment the following technique could be employed:

- Deploy all modules in the network with the default DH/DL of 0xFFFF.
- Issue an ATAGFFFF command on the aggregator node.

Following the preceding sequence would result in all of the nodes in the network which received the AG broadcast to have a DH of 0x0013a200 and a DL of 0x4052c507. These nodes would have automatically built a route to the aggregator.

Example 2: To cause all nodes in the network to build routes to an aggregator node with a MAC address of 0x0013a2004052c507 without affecting the DH/DL of any nodes in the network the ATAGFFFE command should be issued on the aggregator node. This will cause an AG broadcast to be sent to all nodes in the network. All of the nodes will update their internal routing table information to contain a route to the aggregator node. None of the nodes will update their DH/DL registers (because none of the registers are set to an address of 0xFFFE).

Node replacement

The AG command can also be used to update the routing table and DH/DL registers in the network after a module is replaced. The DH/DL registers of nodes in the network can also be updated. To update only the routing table information without affecting the DH/DL registers then the process of Example 2 above can be used. To update the DH/DL registers of the network then the method of Example 3 below can be used.

Example 3: The module with serial number 0x0013a2004052c507 was being used as a network aggregator. It was replaced with a module with serial number 0x0013a200f5e4d3b2. The AG0013a2004052c507 command should be issued on the new module. This will cause all modules which had a DH/DL register setting of 0x0013a2004052c507 to update their DH/DL register setting to the MAC address of the sending module (0x0013a200f5e4d3b2).

Device placement

For a network installation to be successful, the installer must be able to determine where to place individual XBee devices to establish reliable links throughout the network.

Link testing

A good way to measure the performance of a network is to send unicast data through the network from one device to another to determine the success rate of many transmissions. To simplify link testing, the modules support a loopback cluster ID (0x12) on the data endpoint (0xE8). Any data sent to this cluster ID on the data endpoint will be transmitted back to the sender.

The configuration steps to send data to the loopback cluster ID depend on the AP setting:

AT Configuration (AP=0)

To send data to the loopback cluster ID on the data endpoint of a remote device, set the CI command value to 0x12. The SE and DE commands should be set to 0xE8 (default value). The DH and DL commands should be set to the 64-bit address of the remote. After exiting command mode, any received serial characters will be transmitted to the remote device, and returned to the sender.

API Configuration (AP=1 or AP=2)

Send an Explicit Addressing Command API frame (0x11) using 0x12 as the cluster ID, 0xC105 as the profile ID and 0xE8 as the source and destination endpoint. Data packets received by the remote will be echoed back to the sender.

RSSI indicators

It is possible to measure the received signal strength on a device using the DB command. DB returns the RSSI value (measured in –dBm) of the last received packet. The dB value only indicates the received signal strength of the last hop. If a transmission spans multiple hops, the dB value provides no indication of the overall transmission path, or the quality of the worst link – it only indicates the quality of the last link and should be used sparingly.

The DB value can be determined in hardware using the RSSI/PWM module pin (pin 6). If the RSSI PWM functionality is enabled (P0 command), when the module receives data, the RSSI PWM is set to a value based on the RSSI of the received packet. Again, this value only indicates the quality of the last hop. This pin could potentially be connected to an LED to indicate if the link is stable or not.

Note XBee-PRO 900 modules only report RSSI values near the sensitivity level of the radio.

Network discovery

The network discovery command can be used to discover all Digi modules that have joined a network. Issuing the ND command sends a broadcast network discovery command throughout the network. All devices that receive the command will send a response that includes the device's addressing information, node identifier string (see NI command), and other relevant information. This command is useful for generating a list of all module addresses in a network.

When a device receives the network discovery command, it waits a random time before sending its own response. The maximum time delay is set on the ND sender with the NT command. The ND originator includes its NT setting in the transmission to provide a delay window for all devices in the network. Large networks may need to increase NT to improve network discovery reliability.

The default NT value is 0x82 (13 seconds).

Neighbor polling

The neighbor poll command can be used to discover the modules which are immediate neighbors (within RF range) of a particular node. This command is useful in determining network topology and determining possible routes. The command is issued using the FN command. The FN command can be initiated locally on a node using AT command mode or by using a local AT command request frame. The command can also be initiated remotely by sending the target node an FN command using a remote AT command request API frame.

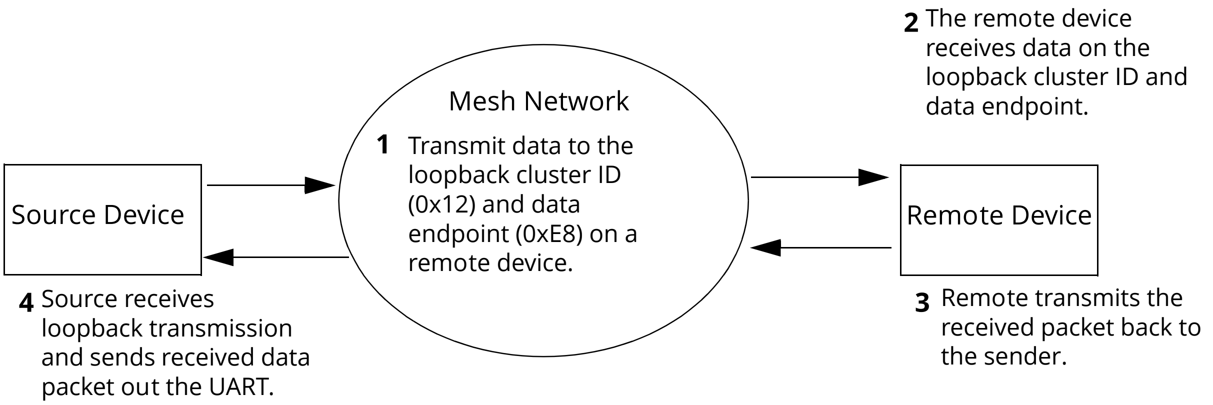
A node which executes an FN command will send a broadcast to all of its immediate neighbors. All radios which receive this broadcast will send an RF packet to the node that initiated the FN command. In the case where the command is initiated remotely this means that the responses are sent directly to the node which sent the FN command to the target node. The response packet is output on the initiating radio in the same format as a network discovery frame.

Link reliability

For a mesh network installation to be successful, the installer must be able to determine where to place individual XBee devices to establish reliable links throughout the mesh network.

Network link testing

A good way to measure the performance of a mesh network is to send unicast data through the network from one device to another to determine the success rate of many transmissions. To simplify link testing, the modules support a loopback cluster ID (0x12) on the data endpoint (0xE8). Any data sent to this cluster ID on the data endpoint will be transmitted back to the sender. This is shown in the figure below:


Source Device
Mesh Network
1 Transmit data to the loopback cluster ID (0x12) and data endpoint (0xE8) on a remote device.
2 The remote device receives data on the loopback cluster ID and data endpoint.
Remote Device
3 Remote transmits the received packet back to the sender.
4 Source receives loopback transmission and sends received data packet out the UART.

Demonstration of how the loopback cluster ID and data endpoint can be used to measure the link quality in a mesh network.

The configuration steps to send data to the loopback cluster ID depend on the AP setting:

AT Configuration (AP=0)

To send data to the loopback cluster ID on the data endpoint of a remote device, set the CI command value to 0x12. The SE and DE commands should be set to 0xE8 (default value). The DH and DL commands should be set to the address of the remote. After exiting command mode, any received serial characters will be transmitted to the remote device, and returned to the sender.

API Configuration (AP=1 or AP=2)

Send an Explicit TX Request API frame (0x11) using 0x12 as the cluster ID and 0xE8 as the source and destination endpoint. Data packets received by the remote will be echoed back to the sender.

Link testing between adjacent devices

It is often advantageous to test the quality of a link between two adjacent nodes in a network. The Test Link Request Cluster ID can be used to send a number of test packets between any two nodes in a network.

A link test can be initiated using an Explicit TX Request frame. The command frame should be addressed to the Test Link Request Cluster ID (0x0014) on destination endpoint 0xE6 on the radio which should execute the test link. The Explicit TX Request frame should contain a 12 byte payload with the following format:

Number of Bytes	Field Name	Description
8	Destination address	The address with which the radio should test its link
2	Payload size	The size of the test packet. The maximum payload size the radio can support can be queried with the NP command.
2	Iterations	The number of packets which should be sent. This should be a number between 1 and 4000.

After completing the transmissions of the test link packets the executing radio will send the following data packet to the requesting radio's Test Link Result Cluster (0x0094) on endpoint (0xE6). If the requesting radio is configured to operate in API mode then the following information will be output as an Explicit RX Indicator API Frame:

Number of Bytes	Field Name	Description
8	Destination address	The address with which the radio tested its link.
2	Payload size	The size of the test packet that was sent to test the link.
2	Iterations	The number of packets which were sent.
2	Success	The number of packets successfully acknowledged.
2	Retries	The total number of MAC retries used to transfer all the packets.
1	Result	0x00 - command was successful. 0x03 - invalid parameter used.
1	RR	The maximum number of MAC retries allowed.
1	maxRSSI	The strongest RSSI reading observed during the test.
1	minRSSI	The weakest RSSI reading observed during the test.
1	avgRSSI	The average RSSI reading observed during the test.

Example:

Suppose that the link between radio A (SH/SL = 0x0013a20040521234) and radio B (SH/ SL=0x0013a2004052abcd) is to be tested by transmitting 1000 40 byte packets. The following API packet should be sent to the serial interface of the radio on which the results should be out- put, radio C. Note that radio C can be the same radio as radio A or B (whitespace used to delineate fields, bold text is the payload portion of the packet):

7E 0020 11 01 0013A20040521234 FFFE E6 E6 0014 C105 00 00 **0013A2004052ABCD 0028 03E8** EB

And the following is a possible packet that could be returned:

7E 0027 91 0013A20040521234 FFFE E6 E6 0094 C105 00 **0013A2004052ABCD 0028 03E8 03E7 0064 00 0A 50 53 52** 9F

(999 out of 1000 packets successful, 100 retries used, RR = 10, maxRSSI = -80 dBm, minRSSI = -83 dBm, avgRSSI = -82 dBm)

If the result field is not equal to zero then an error has occurred. The other fields in the packet should be ignored. If the Success field is equal to zero then the RSSI fields should be ignored.

Trace routing

In many applications it is useful to determine the route which a DigiMesh unicast takes to its destination. This information is especially useful when setting up a network or diagnosing problems within a network. The Trace Route API option of Tx Request Packets (see the API section of this manual for a description of the API frames) causes routing information packets to be transmitted to the originator of a DigiMesh unicast by the intermediate nodes.

When a unicast is sent with the Trace Route API option enabled, the unicast is sent to its destination radios which forward the unicast to its eventual destination will transmit a Route Information (RI) packet back along the route to the unicast originator. A full description of Route Information API packets can be found in the API section of this manual. In general they contain addressing information for the unicast and the intermediate hop for which the trace route packet was generated, RSSI information, and other link quality information.

Example:

Suppose that a data packet with trace route enabled was successfully unicast from radio A to radio E, through radios B, C, and D. The following sequence would occur:

- After the successful MAC transmission of the data packet from A to B, A would output a RI Packet indicating that the transmission of the data packet from A to E was successfully for- warded one hop from A to B.
- After the successful MAC transmission of the data packet from B to C, B would transmit a RI Packet to A. A would output this RI packet out its serial interface upon reception.
- After the successful MAC transmission of the data packet from C to D, C would transmit a RI Packet to A (through B). A would output this RI packet out its serial interface upon reception.
- After the successful MAC transmission of the data packet from D to E, D would transmit a RI Packet to A (through C and B). A would output this RI packet out its serial interface upon reception.

It is important to note that Route Information packets are not guaranteed to arrive in the same order as the unicast packet took. It is also possible for the transmission of Route Information packets on a weak route to fail before arriving at the unicast originator. Because of the large number of Route Information packets which can be generated by a unicast with Trace Route enabled it is suggested that the Trace Route option only be used for occasional diagnostic purposes and not for normal operations.

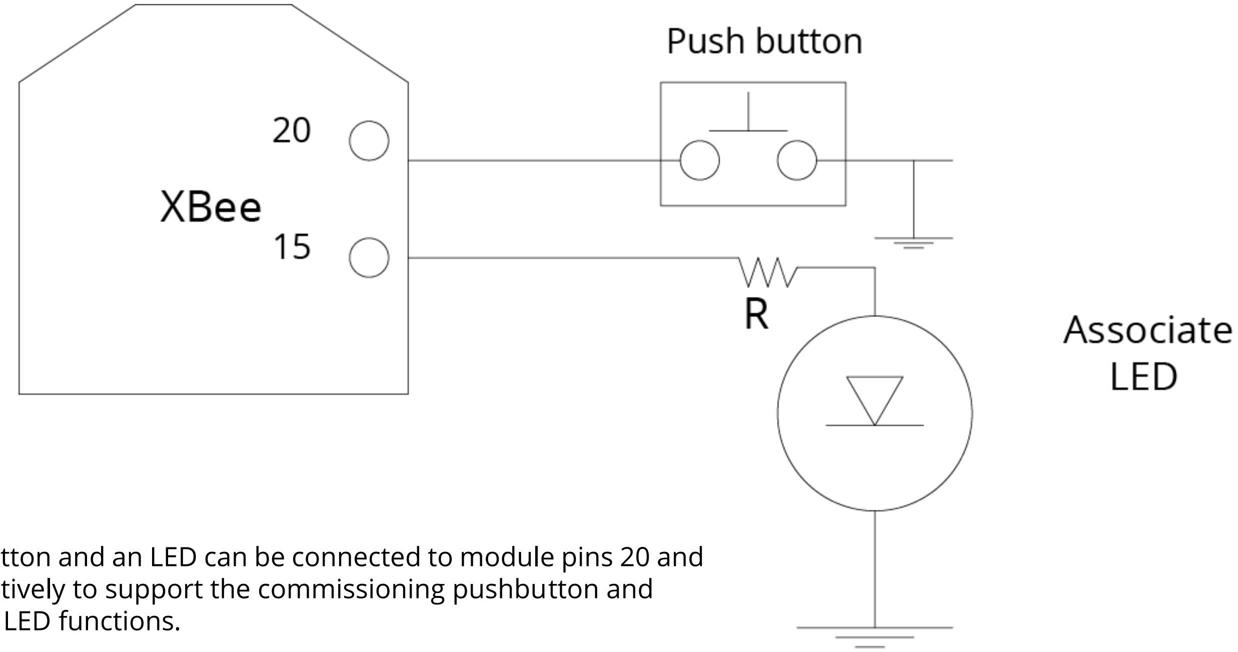
NACK messages

The NACK API option of Tx Request Packets (see the API section of this manual for a description of the API frames) provides the option to have a Route Information packet generated and sent to the originator of a unicast when a MAC acknowledgment failure occurs on one of the hops to the destination. This information is useful because it allows marginal links to be identified and repaired.

Commissioning pushbutton and associate LED

The XBee modules support a set of commissioning and LED behaviors to aid in device deployment and commissioning. These include the commissioning push button definitions and associate LED behaviors. These features can be supported in hardware as shown below.

Figure 8: Commissioning pushbutton and associate LED functionalities



A pushbutton and an LED can be connected to module pins 20 and 15 respectively to support the commissioning pushbutton and associate LED functions.

Commissioning pushbutton

The commissioning pushbutton definitions provide a variety of simple functions to aid in deploying devices in a network. The commissioning button functionality on pin 20 is enabled by setting the D0 command to 1 (enabled by default).

Table 4:

Button Presses	Sleep Configuration	Action
1	Not configured for sleep	Immediately sends a Node Identification broadcast transmission. All devices that receive this transmission will blink their Associate LED rapidly for 1 second. All API devices that receive this transmission will send a Node Identification frame out their UART (API ID 0x95).
1	Configured for asynchronous sleep	Wakes the module for 30 seconds. Immediately sends a Node Identification broadcast transmission. All devices that receive this transmission will blink their Associate LED rapidly for 1 second. All API devices that receive this transmission will send a Node Identification frame out their UART (API ID 0x95).
1	Configured for synchronous sleep	Wakes the module for 30 seconds (or until the entire module goes to sleep). Queues a Node Identification broadcast transmission to be sent at the beginning of the next network wake cycle. All devices that receive this transmission will blink their Associate LEDs rapidly for 1 second. All API devices that receive this transmission will send a Node Identification frame out their UART (API ID 0x95).
2	Not configured for synchronous sleep	No effect.

Table 4:

Button Presses	Sleep Configuration	Action
2	Configured for synchronous sleep	Causes a node that has sleeping router nomination enabled to nominate itself as the network sleep coordinator. See Sleep on page 63.
4	Any	Issues an ATRE to restore module parameters to default values.

Button presses may be simulated in software using the ATCB command. ATCB should be issued with a parameter set to the number of button presses to execute (i.e. sending ATCB1 will execute the action(s) associated with a single button press).

The node identification frame is similar to the node discovery response frame – it contains the device's address, node identifier string (NI command), and other relevant data. All API devices that receive the node identification frame send it out their UART as an API Node Identification Indicator frame (0x95).

Associate LED

The Associate pin (pin 15) can provide indication of the device's network status and diagnostics information. To take advantage of these indications, an LED can be connected to the Associate pin as shown in the figure above. The Associate LED functionality is enabled by setting the D5 command to 1 (enabled by default). If enabled, the Associate pin is configured as an output and will behave as described in the following sections.

The Associate pin indicates the synchronization status of a sleep compatible node. On a non-sleep compatible node the pin functions as a power indicator. The following table describes this functionality.

The LT command can be used to override the blink rate of the Associate pin. When set to 0, the device uses the default blink time (500 ms for sleep coordinator, 250 ms otherwise).

Sleep mode	LED Status	Meaning
0	On, blinking	The device is powered and operating properly.
1, 4, 5	Off	The device is in a low power mode.
1, 4, 5	On, blinking	The device is powered, awake and is operating properly.
7	On, solid	The network is asleep or the device has not synchronized with the network or has lost synchronization with the network.
7, 8	On, slow blinking (500 ms blink time)	The device is acting as the network sleep coordinator and is operating properly.
7, 8	On, fast blinking (250 ms blink time)	The device is properly synchronized with the network.
8	Off	The device is in a low power mode.
8	On, solid	The device has not synchronized or has lost synchronization with the network.

Diagnostics support

The Associate pin works with the commissioning pushbutton to provide additional diagnostics behaviors to aid in deploying and testing a network. If the commissioning push button is pressed once the device transmits a broadcast node identification packet at the beginning of the next wake cycle if sleep compatible, or immediately if not sleep compatible. If the Associate LED functionality is enabled (D5 command), a device that receive this transmission will blink its Associate pin rapidly for one second.

I/O line monitoring

I/O samples

The XBee modules support both analog input and digital IO line modes on several configurable pins.

Queried sampling

Parameters for the pin configuration commands typically include the following:

Pin Command Parameter	Description
0	Unmonitored digital input.
1	Reserved for pin-specific alternate functionalities.
2	Analog input (A/D pins) or PWM output (PWM pins).
3	Digital input, monitored.
4	Digital output, default low.
5	Digital output, default high.
6-9	Alternate functionalities, where applicable.

Setting the configuration command that corresponds to a particular pin will configure the pin:

Module Pin Names	Module Pin Number	Configuration Command
CD / DIO12	4	P2
PWM0 / RSSI / DIO10	6	P0
PWM1 / DIO11	7	P1
DTR / SLEEP_RQ / DIO8	9	D8
AD4 / DIO4	11	D4
CTS / DIO7	12	D7
ON_SLEEP / DIO9	13	D9
ASSOC / AD5 / DIO5	15	D5
RTS / DIO6	16	D6
AD3 / DIO3	17	D3
AD2 / DIO2	18	D2
AD1 / DIO1	19	D1
AD0 / DIO0 / Commissioning Button	20	D0

See the command table for more information. Pullup resistors for each digital input can be enabled using the PR command.

1	Sample Sets	Number of sample sets in the packet. Always set to 1.
2	Digital Channel Mask	Indicates which digital IO lines have sampling enabled. Each bit corresponds to one digital IO line on the module. bit 0 = AD0/DIO0 bit 1 = AD1/DIO1 bit 2 = AD2/DIO2 bit 3 = AD3/DIO3 bit 4 = DIO4 bit 5 = ASSOC/DIO5 bit 6 = RTS/DIO6 bit 7 = CTS/GPIO7 bit 8 = DTR / SLEEP_RQ / DIO8 bit 9 = ON_SLEEP / DIO9 bit 10 = RSSI/DIO10 bit 11 = PWM/DIO11 bit 12 = CD/DIO12 For example, a digital channel mask of 0x002F means DIO0,1,2,3, and 5 are enabled as digital IO.

1	Analog Channel Mask	Indicates which lines have analog inputs enabled for sampling. Each bit in the analog channel mask corresponds to one analog input channel. bit 0 = AD0/DIO0 bit 1 = AD1/DIO1 bit 2 = AD2/DIO2 bit 3 = AD3/DIO3 bit 4 = AD4/DIO4 bit 5 = ASSOC/AD5/DIO5
Variable	Sampled Data Set	If any digital IO lines are enabled, the first two bytes of the data set indicate the state of all enabled digital IO. Only digital channels that are enabled in the Digital Channel Mask bytes have any meaning in the sample set. If no digital IO are enabled on the device, these 2 bytes will be omitted. Following the digital IO data (if any), each enabled analog channel will return 2 bytes. The data starts with AIN0 and continues sequentially for each enabled analog input channel up to AIN5.

If the IS command is issued from AT command mode then a carriage return delimited list will be returned containing the above-listed fields. If the command is issued via an API frame then the module will return an AT command response API frame with the IO data included in the command data portion of the packet.

Example	Sample AT Response
0x01\r	[1 sample set]
0x0C0C\r	[Digital Inputs: DIO 2, 3, 10, 11 enabled]
0x03\r	[Analog Inputs: A/D 0, 1 enabled]
0x0408\r	[Digital input states: DIO 3, 10 high, DIO 2, 11 low]
0x03D0\r	[Analog input ADIO 0= 0x3D0]
0x0124\r	[Analog input ADIO 1=0x120]

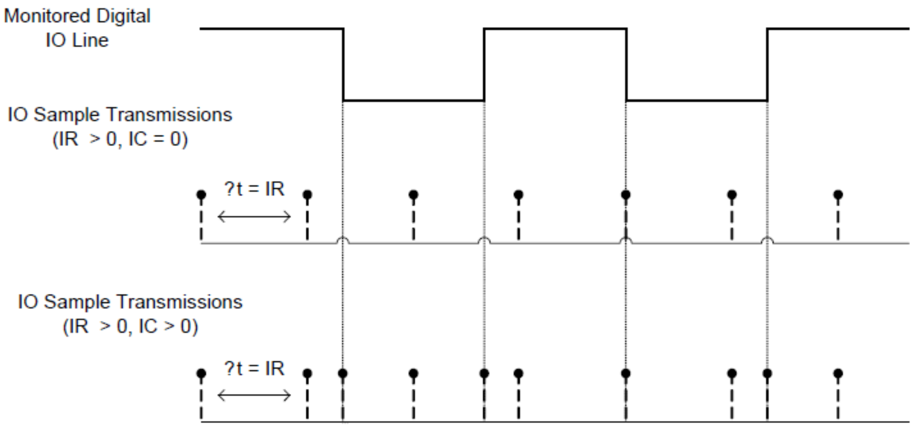
Periodic I/O sampling

Periodic sampling allows an XBee-PRO module to take an I/O sample and transmit it to a remote device at a periodic rate. The periodic sample rate is set by the IR command. If IR is set to 0, periodic sampling is disabled. For all other values of IR, data will be sampled after IR milliseconds have elapsed and transmitted to a remote device. The DH and DL commands determine the destination address of the IO samples. Only devices with API mode enabled will send IO data samples out their UART. Devices not in API mode will discard received IO data samples.

A module with sleep enabled will transmit periodic I/O samples at the IR rate until the ST time expires and the device can resume sleeping. See Sleep Mode on page 37 for more information on sleep.

Digital I/O change detection

Modules can be configured to transmit a data sample immediately whenever a monitored digital I/O pin changes state. The IC command is a bitmask that can be used to set which digital I/O lines should be monitored for a state change. If one or more bits in IC is set, an I/O sample will be transmitted as soon as a state change is observed in one of the monitored digital I/O lines. The figure below shows how edge detection can work with periodic sampling.



Enabling Edge Detection will force an immediate sample of all monitored digital IO lines if any digital IO lines change state.

Sleep Mode

A number of low-power modes exist to enable modules to operate for extended periods of time on battery power. These sleep modes are enabled with the SM command. The sleep modes are characterized as either asynchronous (SM = 1, 4, 5) or synchronous (SM = 7,8). Asynchronous sleeping modes should not be used in a synchronous sleeping network, and vice versa.

Asynchronous sleep modes can be used to control the sleep state on a module by module basis. Modules operating in an asynchronous sleep mode should not be used to route data. Digi strongly encourages users to set asynchronous sleeping modules as end-devices using the CE command. This will prevent the node from attempting to route data.

The synchronous sleep feature of DigiMesh makes it possible for all nodes in the network to synchronize their sleep and wake times. All synchronized cyclic sleep nodes enter and exit a low power state at the same time. This forms a cyclic sleeping network. Nodes synchronize by receiving a special RF packet called a sync message which is sent by a node acting as a sleep coordinator. A node in the network can become a coordinator through a process called nomination. The sleep coordinator will send one sync message at the beginning of each wake period. The sync message is sent as a broadcast and repeated by every node in the network. The sleep and wake times for the entire network can be changed by locally changing the settings on an individual node. The network will use the most recently set sleep settings.

Sleep modes

Normal Mode (SM=0)

Normal mode is the default for a newly powered-on node. In this mode, a node will not sleep. Normal mode nodes should be mains-powered.

A normal mode module will synchronize to a sleeping network, but will not observe synchronization data routing rules (it will route data at any time, regardless of the wake state of the network). When synchronized, a normal node will relay sync messages generated by sleep- compatible nodes but will not generate sync messages. Once a normal node has synchronized with a sleeping network, it can be put into a sleep-compatible sleep mode at any time.

Asynchronous Pin Sleep Mode (SM=1)

Pin sleep allows the module to sleep and wake according to the state of the Sleep_RQ pin (pin 9). Pin sleep mode is enabled by setting the SM command to 1. When Sleep_RQ is asserted (high), the module will finish any transmit or receive operations and enter a low-power state. The module will wake from pin sleep when the Sleep_RQ pin is de-asserted (low).

Asynchronous Cyclic Sleep Mode (SM=4)

Cyclic sleep allows the module to sleep for a specified time and wake for a short time to poll. Cyclic sleep mode is enabled by setting the SM command to 4. In cyclic sleep, the module sleeps for a specified time. If the XBee receives serial or RF data while awake, it will then extend the time before it returns to sleep by the amount specified by the ST command. Otherwise, it will enter sleep mode immediately. The On_SLEEP line is asserted (high) when the module wakes, and is de-asserted (low) when the module sleeps. If hardware flow control is enabled (D7 command), the CTS pin will assert (low) when the module wakes and can receive serial data, and de-assert (high) when the module sleeps.

Asynchronous Cyclic Sleep with Pin Wake Up Mode (SM=5)

(SM=5) is similar to both the (SM=1) and (SM=4) modes. When the SLEEP_REQUEST pin is asserted, the module will enter a cyclic sleep mode similar to (SM=4). When the SLEEP_REQUEST pin is de-asserted, the module will immediately wake up. The module will not sleep when the SLEEP_REQUEST pin is de-asserted.

Synchronous Sleep Support Mode (SM=7)

A node in synchronous sleep support mode will synchronize itself with a sleeping network but will not itself sleep. At any time, the node will respond to new nodes which are attempting to join the sleeping network with a sync message. A sleep support node will only transmit normal data when the other nodes in the sleeping network are awake. Sleep support nodes are especially useful when used as preferred sleep coordinator nodes and as aids in adding new nodes to a sleeping network.

Note Because sleep support nodes do not sleep, they should be mains powered.

Synchronous Cyclic Sleep Mode (SM=8)

A node in synchronous cyclic sleep mode sleeps for a programmed time, wakes in unison with other nodes, exchanges data and sync messages, and then returns to sleep. While asleep, it cannot receive RF messages or read commands from the UART port. Generally, sleep and wake times are specified by the SP and ST respectively of the network's sleep coordinator. These parameters are only used at start up until the node is synchronized with the network. When a module has synchronized with the network, its sleep and wake times can be queried with the OS and OW commands respectively. If D9 = 1 (ON_SLEEP enabled) on a cyclic sleep node, the ON_SLEEP line will assert when the module is awake and de-assert when the module is asleep. CTS is also de-asserted while asleep (D7 = 1). A newly-powered unsynchronized sleeping node will poll for a synchronized message and then sleep for the period specified by SP, repeating this cycle until it becomes synchronized by receiving a sync message. Once a sync message is received, the node will synchronize itself with the network.

Note All nodes in a synchronous sleep network should be configured to operate in either Synchronous Sleep Support Mode or Synchronous Cyclic Sleep Mode. Asynchronous sleeping nodes are not compatible with synchronous sleep nodes.

Asynchronous sleep operation

Wake timer

In cyclic sleep mode (SM=4 or SM=5), if serial or RF data is received, the module will start a sleep timer (time until sleep). Any data received serially or by RF link will reset the timer. The timer duration can be set using the ST command. The module returns to sleep when the sleep timer expires.

Sleeping routers

The Sleeping Router feature of DigiMesh makes it possible for all nodes in the network to synchronize their sleep and wake times. All synchronized cyclic sleep nodes enter and exit a low power state at the same time. This forms a cyclic sleeping network. Nodes synchronize by receiving a special RF packet called a sync message which is sent by a node acting as a sleep coordinator. A node in the network can become a sleep coordinator through a process called nomination. The sleep coordinator will send one sync message at the beginning of each wake period. The sync message is sent as a broadcast and repeated by every node in the network. The sleep and wake times for the entire network can be changed by locally changing the settings on an individual node. The network will use the most recently set sleep settings.

Operation

One node in a sleeping network acts as the sleeping coordinator. The process by which a node becomes a sleep coordinator is described later in this document. During normal operations, at the beginning of a wake cycle the sleep coordinator will send a sync message as a broadcast to all nodes in the network. This message contains synchronization information and the wake and sleep times for the current cycle. All cyclic sleep nodes receiving a sync message will remain awake for the wake time and then sleep for the sleep period specified.

The sleep coordinator will send one sync message at the beginning of each cycle with the currently configured wake and sleep times. All router nodes which receive this sync message will relay the message to the rest of the network. If the sleep coordinator does not hear a re-broadcast of the sync message by one of its immediate neighbors then it will re-send the message one additional time. It should be noted that if SP or ST are changed, the network will not apply the new settings until the beginning of the next wake time. See the Changing Sleep Parameters section below for more information.

A sleeping router network is robust enough that an individual node can go several cycles without receiving a sync message (due to RF interference, for example). As a node misses sync messages, the time available for transmitting messages in the wake time is reduced to maintain synchronization accuracy. By default, a module will also reduce its active sleep time progressively as sync messages are missed.

Synchronization messages

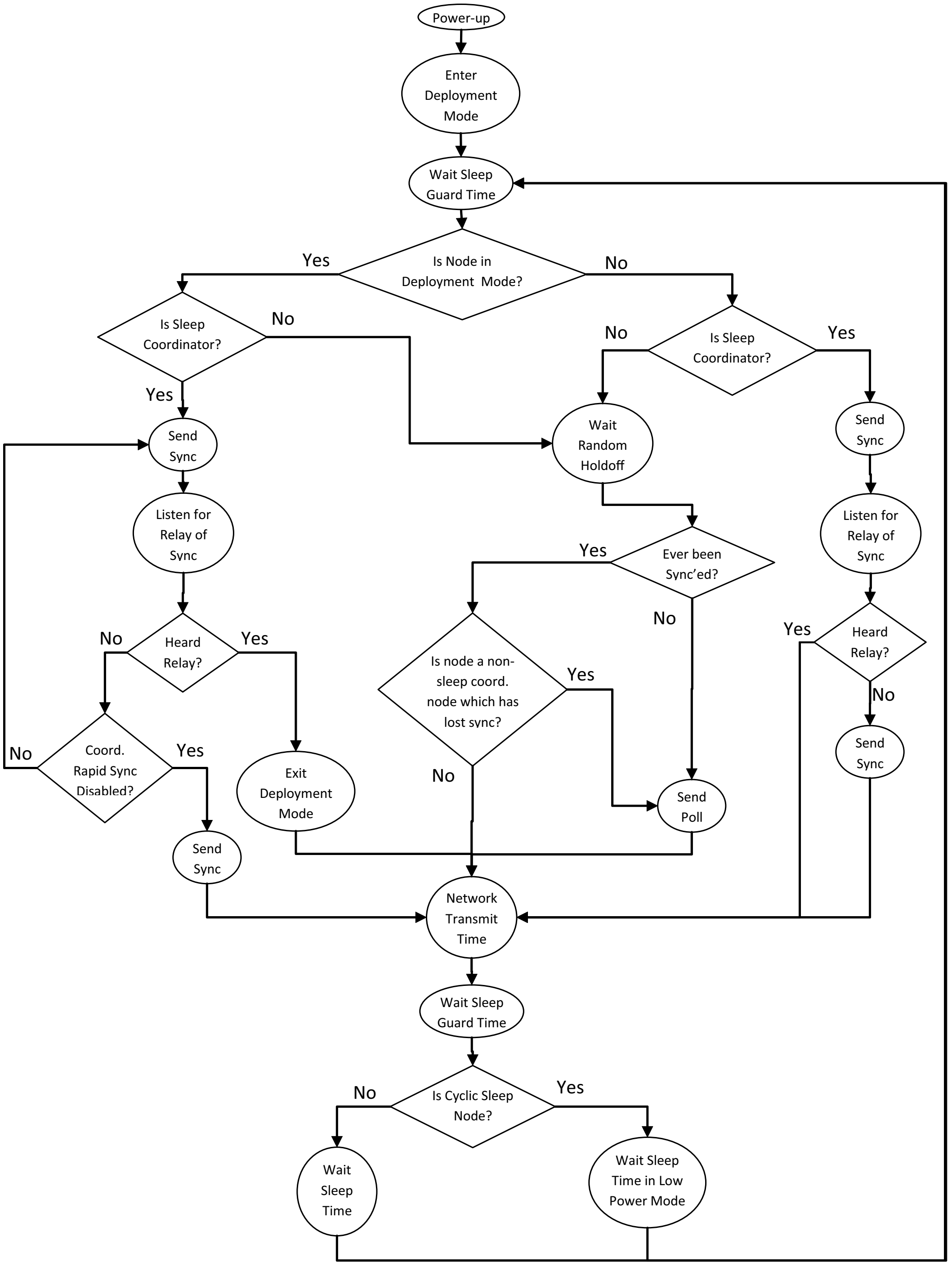
A sleep coordinator will regularly send sync messages to keep the network in sync. Nodes which have not been synchronized or, in some cases, which have lost sync will also send messages requesting sync information.

Deployment mode is used by sleep compatible nodes when they are first powered up and the sync message has not been relayed. A sleep coordinator in deployment mode will rapidly send sync messages until it receives a relay of one of those messages. This allows a network to be deployed more effectively and allows a sleep coordinator which is accidentally or intentionally reset to rapidly re-synchronize with the rest of the network. If a node which has exited deployment mode receives a sync message from a sleep coordinator which is in deployment mode, the sync will be rejected and a corrective sync will be sent to the sleep coordinator. Deployment mode can be disabled using the sleep options command (SO).

A sleep coordinator which is not in deployment mode or which has had deployment mode disabled will send a sync message at the beginning of the wake cycle. The sleep coordinator will then listen for a neighboring node to relay the sync. If the relay is not heard, the sync coordinator will send the sync one additional time.

A node which is not acting as a sleep coordinator which has never been synchronized will send a message requesting sync information at the beginning of its wake cycle. Synchronized nodes which receive one of these messages will respond with a synchronization packet. Nodes which are configured as non-sleep coordinators (using the SO command) which have gone six or more cycles without hearing a sync will also send a message requesting sync at the beginning of their wake period.

The following diagram illustrates the synchronization behavior of sleep compatible modules:



Becoming a Sleep Coordinator

This section describes the four ways in which a node an become a sleep coordinator.

Preferred Sleep Coordinator option

A node can be specified to always act as a sleep coordinator. This is done by setting the preferred sleep coordinator bit (bit 0) in the sleep operations parameter (SO) to 1. A node with the sleep coordinator bit set will always send a sync message at the beginning of a wake cycle. For this reason, it is imperative that no more than one node in the network has this bit set. Although it is not necessary to specify a preferred sleep coordinator, it is often useful to select a node for this purpose to improve network performance. A node which is centrally located in the network can serve as a good sleep coordinator to minimize the number of hops a sync message must take to get across the network. A sleep support node and/or a node which is mains powered may be a good candidate.

The preferred sleep coordinator bit should be used with caution. The advantages of using the option become weaknesses when used on a node that is not positioned or configured properly. The preferred sleep coordinator option can also be used when setting up a network for the first time. When starting a network, a node can be configured as a sleep coordinator so it will begin sending sleep messages. After the network is set up, the preferred sleep coordinator bit can be disabled.

Nomination and election

Nomination is an optional process that can occur on a node in the event that contact with the network sleep coordinator is lost. By default, this behavior is disabled. This behavior can be enabled with the sleep options command (SO). This process will automatically occur in the event that contact with the previous sleep coordinator is lost. Any sleep compatible node which has this behavior enabled is eligible to become the sleep coordinator for the network. If a sleep compatible node has missed three or more sync messages and is not configured as a non-sleep coordinator (presumably because the sleep coordinator has been disabled) it may become a sleep coordinator. Depending on the platform and other configured options, such a node will eventually nominate itself after a number of cycles without a sync. A nominated node will begin acting as the new network sleep coordinator. It is possible for multiple nodes to nominate themselves as the sleep coordinator. If this occurs, an election will take place to establish seniority among the multiple sleep coordinators. Seniority is determined by four factors (in order of priority):

1. Newer sleep parameters: a node using newer sleep parameters (SP/ST) is considered senior to a node using older sleep parameters. See Changing sleep parameters on page 42.
2. Preferred Sleep Coordinator: a node acting as a preferred sleep coordinator is senior to other nodes.
3. Sleep Support Node: sleep support nodes are senior to cyclic sleep nodes. This behavior can be modified using the SO parameter.
4. Serial number: in the event that the above factors do not resolve seniority, the node with the higher serial number is considered senior.

Commissioning button

The commissioning button can be used to select a module to act as the sleep coordinator. If the commissioning button functionality has been enabled, a node can be immediately nominated as a sleep coordinator by pressing the commissioning button twice or by issuing the CB2 command. A node nominated in this manner is still subject to the election process described above. A node configured as a non-sleep coordinator will ignore commissioning button nomination requests.

Changing sleep parameters

Any sleep compatible node in the network which does not have the non-sleep coordinator sleep option set can be used to make changes to the network's sleep and wake times. If a node's SP and/or ST are changed to values different from those that the network is using, that node will become the sleep coordinator. That node will begin sending sync messages with the new sleep parameters at the beginning of the next wake cycle.

Note For normal operations, a module will use the sleep and wake parameters it gets from the sleep sync message, not the ones specified in its SP and ST parameters. The SP and ST parameters are not updated with the values of the sync message. The current network sleep and wake times used by the node can be queried using the OS and OW commands.

Note Changing network parameters can cause a node to become a sleep coordinator and change the sleep settings of the network. The following commands can cause this to occur: NH, NN, NQ, and MR. In most applications, these network parameters should only be configured during deployment.

Sleep guard times

To compensate for variations in the timekeeping hardware of the various modules in a sleeping router network, sleep guard times are allocated at the beginning and end of the wake time. The size of the sleep guard time varies based on the sleep and wake times selected and the number of cycles that have elapsed since the last sync message was received. The sleep guard time guarantees that a destination radio will be awake when a transmission is sent. As more and more consecutive sync messages are missed, the sleep guard time increases in duration and decreases the available transmission time.

Auto-early wake-up sleep option

Similarly to the sleep guard time, the auto early wake-up option decreases the sleep period based on the number of sync messages missed. This option comes at the expense of battery life. Auto- early wake-up sleep can be disabled using the sleep options (SO) command.

Configuration

Selecting sleep parameters

Choosing proper sleep parameters is vital to creating a robust sleep-enabled network with a desirable battery life. To select sleep parameters that will be good for most applications, follow these steps:

1. **Choose NN and NH.** Based on the placement of the nodes in your network, select appropriate values for the Network Hops (NH) and Network Delay Slots (NN) parameters.

Note The default values of NH and NN have been optimized to work for the majority of deployments. In most cases, we suggest that these parameters not be modified from their default values. Decreasing these parameters for small networks can improve battery life, but care should be taken so that the values are not made too small.

2. Calculate the sync message propagation time (SMPT). This is the maximum amount of time it takes for a sleep synchronization message to propagate to every node in the network. This number can be estimated with the following formula:

   SMPT = NN * NH * (MT + 1) 18ms

3. Select the desired duty cycle. The ratio of sleep time to wake time is the factor that has the greatest effect on the RF module's power consumption. Battery life can be estimated based on the following factors: sleep period, wake time, sleep current, RX current, TX current, and battery capacity.

4. Choose the sleep period and wake time. The wake time needs to be long enough to transmit the desired data as well as the sync message. The ST parameter will automatically adjust upwards to its minimum value when other AT commands are changed that will affect it (SP, NN, and NH). Use a value larger than this minimum. If a module misses successive sync messages, it reduces its available transmit time to compensate for possible clock drift. Budget a large enough ST time to allow for a few sync messages to be missed and still have time for normal data transmissions.

Starting a sleeping network

By default, all new nodes operate in normal (non-sleep) mode. To start a sleeping network, follow these steps:

1. Enable the preferred sleep coordinator option on one of the nodes, and set its SM to a sleep compatible mode (7 or 8) with its SP and ST set to a quick cycle time. The purpose of a quick cycle time is to allow commands to be sent quickly through the network during commissioning.

2. Next, power on the new nodes within range of the sleep coordinator. The nodes will quickly receive a sync message and synchronize themselves to the short cycle SP and ST.

3. Configure the new nodes in their desired sleep mode as cyclic sleeping nodes or sleep support nodes.

4. Set the SP and ST values on the sleep coordinator to the desired values for the deployed network.

5. Wait a cycle for the sleeping nodes to sync themselves to the new SP and ST values.

6. Disable the preferred sleep coordinator option bit on the sleep coordinator (unless a preferred sleep coordinator is desired).

7. Deploy the nodes to their positions.

Alternatively, nodes can be set up with their sleep pre-configured and written to flash (using the WR command) prior to deployment. If this is the case, the commissioning button and associate LED can be used to aid in deployment:

1. If a preferred sleep coordinator is going to be used in the network, deploy it first. If there will be no preferred sleep coordinator, select a node for deployment, power it on and press the commissioning button twice. This will cause the node to begin emitting sync messages.

   Verify that the first node is emitting sync messages by watching its associate LED. A slow blink indicates that the node is acting as a sleep coordinator.

2. Power on nodes in range of the sleep coordinator or other nodes which have synchronized with the network. If the synchronized node is asleep, it can be woken by pressing the commissioning button once.
3. Wait a cycle for the new node to sync itself.
4. Verify that the node syncs with the network. The associate LED will blink when the module is awake and synchronized.
5. Continue this process until all nodes have been deployed.

Adding a new node to an existing network

To add a new node to the network, the node must receive a sync message from a node already in the network. On power-up, an unsynchronized sleep compatible node will periodically send a broadcast requesting a sync message and then sleep for its SP period. Any node in the network that receives this message will respond with a sync. Because the network can be asleep for extended periods of time, and as such cannot respond to requests for sync messages, there are methods that can be used to sync a new node while the network is asleep.

1. Power the new node on within range of a sleep support node. Sleep support nodes are always awake and will be able to respond to sync requests promptly.
2. A sleeping cyclic sleep node in the network can be woken by the commissioning button. Place the new node in range of the existing cyclic sleep node and wake the existing node by holding down the commissioning button for 2 seconds, or until the node wakes. The existing node stays awake for 30 seconds and will respond to sync requests while it is awake.

If you do not use one of these two methods, you must wait for the network to wake up before adding the new node. The new node should be placed in range of the network with a sleep/wake cycle that is shorter than the wake period of the network. The new node will periodically send sync requests until the network wakes up and it receives a sync message.

Changing sleep parameters

Changes to the sleep and wake cycle of the network can be made by selecting any node in the network and changing the SP and/or ST of the node to values different than those the network is currently using. If using a preferred sleep coordinator or if it is known which node is acting as the sleep coordinator, it is suggested that this node be used to make changes to network settings. If the network sleep coordinator is not known, any node that does not have the non-sleep coordinator sleep option bit set (see the SO command) can be used.

When changes are made to a node's sleep parameters, that node will become the network's sleep coordinator (unless it has the non-sleep coordinator option selected) and will send a sync message with the new sleep settings to the entire network at the beginning of the next wake cycle. The network will immediately begin using the new sleep parameters after this sync is sent.

Changing sleep parameters increases the chances that nodes will lose sync. If a node does not receive the sync message with the new sleep settings, it will continue to operate on its old settings. To minimize the risk of a node losing sync and to facilitate the re-syncing of a node that does lose sync, the following precautions can be taken:

1. Whenever possible, avoid changing sleep parameters.
2. Enable the missed sync early wake up sleep option (SO). This command is used to tell a node to wake up progressively earlier based on the number of cycles it has gone without receiving a sync.

This will increase the probability that the un-synced node will be awake when the network wakes up and sends the sync message.

Note Using this sleep option increases reliability but may decrease battery life. Nodes using this sleep option which miss sync messages will have an increased wake time and decreased sleep time during cycles in which the sync message is missed. This will reduce battery conservation.

3. When changing between two sets of sleep settings, choose settings so that the wake periods of the two sleep settings will happen at the same time. In other words, try to satisfy the following equation: (SP1 + ST1) = N * (SP2 + ST2), where SP1/ST1 and SP2/ST2 are the desired sleep settings and N is an integer.

Rejoining nodes that have lost sync

Mesh networks get their robustness from taking advantage of routing redundancies which may be available in a network. It is recommended to architect the network with redundant mesh nodes to increase robustness. If a scenario exists such that the only route connecting a subnet to the rest of the network depends on a single node, and that node fails -- or the wireless link fails due to changing environmental conditions (catastrophic failure condition), then multiple subnets may arise while using the same wake and sleep intervals. When this occurs the first task is to repair, replace, and strengthen the weak link with new and/or redundant modules to fix the problem and prevent it from occurring in the future.

When the default DigiMesh sleep parameters are used, separated subnets will not drift out of phase with each other. Subnets can drift out of phase with each other if the network is configured in one of the following ways:

- If multiple modules in the network have had the non-sleep coordinator sleep option bit disabled and are thus eligible to be nominated as a sleep coordinator.
- If the modules in the network are not using the auto early wake-up sleep option.

If a network has multiple subnets that have drifted out of phase with each other, get the subnets back in phase with the following steps:

1. Place a sleep support node in range of both subnets.
2. Select a node in the subnet that you want the other subnet to sync up with. Use this node to slightly change the sleep cycle settings of the network (increment ST, for example).
3. Wait for the subnet’s next wake cycle. During this cycle, the node selected to change the sleep cycle parameters will send the new settings to the entire subnet it is in range of, including the sleep support node which is in range of the other subnet.
4. Wait for the out of sync subnet to wake up and send a sync. When the sleep support node receives this sync, it will reject it and send a sync to the subnet with the new sleep settings.
5. The subnets will now be in sync. The sleep support node can be removed. If desired, the sleep cycle settings can be changed back to what they were.

In the case that only a few nodes need to be replaced, this method can also be used:

1. Reset the out of sync node and set its sleep mode to cyclic sleep (SM = 8). Set it up to have a short sleep cycle.
2. Place the node in range of a sleep support node or wake a sleeping node with the commissioning button.

3. The out of sync node will receive a sync from the node which is synchronized to the network and sync to the network sleep settings.

Diagnostics

The following are useful in some applications when managing a sleeping router network:

- Query current sleep cycle: the OS and OW command can be used to query the current operational sleep and wake times a module is currently using.
- Sleep Status: the SS command can be used to query useful information regarding the sleep status of the module. This command can be used to query if the node is currently acting as a network sleep coordinator, as well as other useful diagnostics.
- Missed Sync Messages Command: the MS command can be used to query the number of cycles that have elapsed since the module last received a sync message.
- Sleep Status API messages: when enabled with the SO command, a module configured in API mode will output modem status frames immediately after a module wakes up and just prior to a module going to sleep.

XBee-PRO 900 DigiMesh

DigiMesh networking

Mesh networking allows messages to be routed through several different nodes to a final destination. DigiMesh firmware allows system integrators to bolster their networks with the self-healing attributes of mesh networking. In the event that one RF connection between nodes is lost (due to power-loss, environmental obstructions, etc.) critical data can still reach its destination due to the mesh networking capabilities embedded inside the modules.

XBee-PRO 900 modules support a point-multipoint firmware variant and a DigiMesh firmware variant. The following section applies only to the DigiMesh variant.

DigiMesh feature set

DigiMesh contains the following features

- **Self-healing**

  Any node may enter or leave the network at any time without causing the network as a whole to fail.
- **Peer-to-peer architecture**

  No hierarchy and no parent-child relationships are needed.
- **Quiet protocol**

  Routing overhead will be reduced by using a reactive protocol similar to AODV.
- **Route discovery**

  Rather than maintaining a network map, routes will be discovered and created only when needed.
- **Selective acknowledgments**

  Only the destination node will reply to route requests.
- **Reliable delivery**

  Reliable delivery of data is accomplished by means of acknowledgments.
- **Sleep modes**

  Low power sleep modes with synchronized wake are supported, with variable sleep and wake times.

Networking concepts

Device configuration

DigiMesh modules can be configured to act as routers or end devices with the CE command. By default modules are configured as routers and will actively relay network unicast and broadcast traffic as described below.

Network ID

DigiMesh networks are defined with a unique network identifier. This identifier is set with the ID command. For modules to communicate they must be configured with the same network identifier. The ID parameter allows multiple DigiMesh networks to co-exist on the same physical channel.

Operating channel

XBee-PRO modules utilize frequency hopping spread spectrum technology. There are 8 defined hopping patterns which operate on 12 physical channels. (The International variant has 4 hopping patterns on 5 physical channels). The hopping pattern is selected using the HP command.

For modules to communicate, the hop sequence (HP) and network identifier (ID) must be equal on all modules in the network.

Data transmission and routing

Unicast addressing

When transmitting while using Unicast communications, reliable delivery of data is accomplished using retries and acknowledgments. The number of retries is determined by the MR (Network Retries) parameter. RF data packets are sent up to MR + 1 times and ACKs (acknowledgments) are transmitted by the receiving node upon receipt. If a network ACK is not received within the time it would take for a packet to traverse the network twice, a retransmission occurs.

To send Unicast messages, set the DH and DL on the transmitting module to match the corresponding SH and SL parameter values on the receiving module.

Broadcast addressing

Broadcast transmissions will be received and repeated by all nodes in the network. Because ACKs are not used the originating node will send the broadcast multiple times as configured by the MT parameter. Essentially the extra transmissions become automatic retries without acknowledgments. A node which receives multiple copies of the same packet will discard the duplicates. This will result in all nodes repeating the transmission MT+1 times as well. In order to avoid RF packet collisions, a random delay is inserted before each node relays the broadcast message. (See the NN parameter for details on changing this random delay time.) Sending frequent broadcast transmissions can quickly reduce the available network bandwidth and as such should be used sparingly.

The broadcast address is a 64 bit address with the lowest 16 bits set to 1. The upper bits are set to

1.To send a broadcast transmission set DH to 0 and DL to 0xFFFF. In API mode the destination address would be set to 0x000000000000FFFF.

Routing

A module within a mesh network is able to determine reliable routes using a routing algorithm and table. The routing algorithm uses a reactive method derived from AODV (Ad-hoc On-demand Distance Vector). An associative routing table is used to map a destination node address with its next hop. By sending a message to the next hop address, either the message will reach its destination or be forwarded to an intermediate node which will route the message on to its destination. A message with a broadcast address will be received by all neighbors. All receiving neighbors will rebroadcast the message MT+1 times and eventually the message will reach all corners of the network. Packet tracking prevents a node from resending a broadcast message more than MT+1 times.

Route discovery

If the source node doesn't have a route to the requested destination, the packet is queued to await a route discovery (RD) process. This process is also used when a route fails. A route fails when the source node uses up its network retries without ever receiving an ACK. This results in the source node initiating RD.

RD begins by the source node broadcasting a route request (RREQ). Any node that receives the RREQ that is not the ultimate destination is called an intermediate node.

Intermediate nodes may either drop or forward a RREQ, depending on whether the new RREQ has a better route back to the source node. If so, information from the RREQ is saved and the RREQ is updated and broadcast. When the ultimate destination receives the RREQ, it unicasts a route reply (RREP) back to the source node along the path of the RREQ. This is done regardless of route quality and regardless of how many times an RREQ has been seen before.

This allows the source node to receive multiple route replies. The source node selects the route with the best round trip route quality, which it will use for the queued packet and for subsequent packets with the same destination address.

Throughput

Throughput in a DigiMesh network can vary by a number of variables, including: number of hops, encryption enabled/disabled, sleeping end devices, failures/route discoveries. Our empirical testing showed the following throughput performance in a robust operating environment (low interference).

Configuration	Data Throughput
1 hop, Encryption Disabled	87.1 kb/s
3 hop, Encryption Disabled	33.9 kb/s
6 hop, Encryption Disabled	17.0 kb/s
1 hop, Encryption Enabled	78.9 kb/s
3 hop, Encryption Enabled	32.8 kb/s
6 hop, Encryption Enabled	16.5 kb/s

Note Data throughput measurements were made setting the serial interface rate to 115200 b/s, and measuring the time to send 100,000 bytes from source to destination. During the test, no route discoveries or failures occurred.

Transmission timeouts

When a node receives an API TX Request (API configured modules) or an RO timeout occurs (modules configured for Transparent Mode) the time required to route the data to its destination depends on a number of configured parameters, whether the transmission is a unicast or a broadcast, and if the route to the destination address is known. Timeouts or timing information is provided for the following transmission types:

- Transmitting a broadcast
- Transmitting a unicast with a known route
- Transmitting a unicast with an unknown route
- Transmitting a unicast with a broken route.

Note The timeouts in this section are theoretical timeouts and not precisely accurate. The application should pad the calculated maximum timeouts by a few hundred milliseconds. When using API mode, Tx Status API packets should be the primary method of determining if a transmission has completed.

Unicast one hop time

A building block of many of the calculations presented below is the unicastOneHopTime. As its name indicates, it represents the amount of time it takes to send a unicast transmission between two adjacent nodes. It is largely dependent upon the mac retry setting (RR). DigiMesh networks assume that the average number of mac level retries across a multi-hop wireless link will be 3 or less. It is defined as follows:

RR (mac retries)	
0	unicastOneHopTime = 18 ms
1	unicastOneHopTime = 42 ms
2	unicastOneHopTime = 67 ms
3	unicastOneHopTime = 99 ms

Transmitting a broadcast

A broadcast transmission must be relayed by all routers in the network. The maximum delay would be when the sender and receiver are on the opposite ends of the network. The NH, NN, and MT parameters define the maximum broadcast delay as follows:

BroadcastTxTime=NN*NH*(MT+1)* 18mSec

Transmitting a unicast with a known route

When a route to a destination node is known the transmission time is largely a function of the number of hops and retries. The timeout associated with a unicast assumes the maximum number of hops is necessary (as specified by NH). The timeout can be estimated in the following manner:

knownRouteUnicast=2*NH*MR*unicastOneHopTime

Transmitting a unicast with an unknown route

If the route to the destination is not known the transmitting module will begin by sending a route discovery. If the route discovery is successful and a route is found then the data is transmitted. The timeout associated with the entire operation can be estimated as follows:

unknownRouteUnicast=BroadcastTxTime+NH*unicastOneHopTime+knownRouteUnicast

Transmitting a unicast with a broken route

If the route to a destination node has changed since the last time a route discovery was completed a node will begin by attempting to send the data along the previous route. After it fails a route discovery will be initiated and, upon completion of the route discovery, the data will be transmitted along the new route. The timeout associated with the entire operation can be estimated as follows:

brokenRouteUnicast=BroadcastTxTime+NH*unicastOneHopTime+2*knownRouteUnicast

Command Reference Tables

Special

Special commands

AT Command	Name and Description	Parameter Range	Default
AC	**Apply Changes**. Immediately applies new settings without exiting command mode.	-	-
FR	**Software Reset**. Reset module. Responds immediately with an “OK” then performs a reset 100 ms later.	-	-
RE	**Restore Defaults**. Restore module parameters to factory defaults.	-	-
WR	**Write**. Write parameter values to non-volatile memory so that parameter modifications persist through subsequent resets. **Note** Once WR is issued, no additional characters should be sent to the module until after the "OK\r" response is received.	-	-
R1	**Restore Compiled**. Restore module parameters to compiled defaults.	-	-

MAC/PHY level

MAC/PHY-level commands

AT Command	Name and Description	Parameter Range	Default
ID	Network ID. Set or read the user network identifier. Nodes must have the same network identifier to communicate. Changes to ID can be written to non-volatile memory using the WR command.	0x0000-0x7FFF	0x7FFF
MT	Broadcast Multi-Transmit. Set/Read the number of additional MAC-level broadcast transmissions. All broadcast packets are transmitted MT+1 times to ensure it is received.	0-0xF	3
RR	Unicast Mac Retries. Set/Read the maximum number of MAC level packet delivery attempts for unicasts. If RR is non-zero packets sent from the radio will request an acknowledgment, and can be resent up to RR times if no acknowledgments are received.	0-0xF	10
ED	Energy Detect. Start an Energy Detect Scan. This parameter is the time in milliseconds to scan all channels. The Module will loop through all the channels until the time elapses. The maximal energy on each channel is returned, and each value is followed by a comma with the list ending with a carriage return. The values returned reflect the detected energy level in units of -dBm.	0-0x0C	

Diagnostics

Diagnostics commands - MAC statistics and timeouts

AT Command	Name and Description	Parameter Range	Default
BC	**Bytes Transmitted**. The number of RF bytes transmitted. This count is incremented for every PHY-level byte transmitted. The purpose of this count is to estimate battery life by tracking time spent doing transmissions. This number rolls over to zero from 0xFFFF. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
DB	**Received Signal Strength**. This command reports the received signal strength of the last received RF data packet. The DB command only indicates the signal strength of the last hop. It does not provide an accurate quality measurement for a multihop link. The DB command value is measured in -dBm. For example if DB returns 0x60, then the RSSI of the last packet received was -96dBm.	n/a	n/a
GD	**Good packets**. Read the number of good frames with valid MAC headers that are received on the RF interface. When the value reaches 0xFFFF, it stays there.	n/a	n/a
EA	**MAC ACK Timeouts**. This count is incremented whenever a MAC ACK timeout occurs on a MAC-level unicast. Once the number reaches 0xFFFF, further events will not be counted. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
TR	**Transmission Errors**. Read the number of MAC frames that exhaust MAC retries without ever receiving a MA acknowledgment message from the adjacent node. When the value reaches 0xffff, it stays there.	n/a	n/a
UA	**MAC Unicast Transmission Count**. This count is incremented whenever a MAC unicast transmission occurs for which an ACK is requested. Once the number reaches 0xFFFF, further evens will not be counted. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
%H	**MAC Unicast One Hop Time**. The MAC unicast one hop timeout in milliseconds. Changing MAC parameters can change this value.	[read-only]	
%8	**MAC Broadcast One Hop Time**. The MAC broadcast one hop timeout in milliseconds. Changing MAC parameters can change this value.	[read-only]	
%V	**Supply Voltage**. Reads the voltage on the Vcc pin in mV. Read module voltage in millivolts.	0-0xF00	n/a

Network

Network commands: DigiMesh

AT Command	Name and Description	Parameter Range	Default
CE	**Node Type**. Set/read the node networking type. A module set as an end device will not propagate broadcasts and won't become and intermediate node on a route.	0 - Router 2 - End Device	0
BH	**Broadcast Radius**. Set/read the transmission radius for broadcast data transmissions. Set to 0 for maximum radius. If BH is set greater than NH then the value of NH is used.	0-0x20	0
NH	**Network Hops**. Set or read the maximum number of hops expected to be seen in a network route. This value doesn't limit the number of hops allowed, but it is used to calculate timeouts waiting for network acknowledgments.	1-0x20	7
NN	**Network Delay Slots**. Set or read the maximum random number of network delay slots before rebroadcasting a network packet. One network delay slot is approximately 13ms.	1-0x0A	3
MR	**Mesh Network Retries**. Set or read the maximum number of network packet delivery attempts. If MR is non-zero, packets sent will request a network acknowledgment, and can be resent up to MR+1 times if no acknowledgments are received.	0-7	1

Addressing

Addressing commands

AT Command	Name and Description	Parameter Range	Default
SH	**Serial Number High**. Read high 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled. This value is read-only and it never changes.	0-0xFFFFFFFF	Factory
SL	**Serial Number Low**. Read low 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled. This is read only and it is also the serial number of the node.	0-0xFFFFFFFF	Factory
DH	**Destination Address High**. Set/Get the upper 32 bits of the 64-bit destination address. When combined with DL, it defines the destination address used for transmission.	0-0xFFFFFFFF	0
DL	**Destination Address Low**. Set/Get the lower 32 bits of the 64-bit destination address. When combined with DH, DL defines the destination address used for transmission.	0-0xFFFFFFFF	0x0000FFFF

Addressing commands

AT Command	Name and Description	Parameter Range	Default
NI	**Node Identifier**. Stores a string identifier. The string accepts only printable ASCII data In AT Command Mode, the string can not start with a space. A carriage return or comma ends the command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ATND (Network Discover) command. This identifier is also used with the ATDN (Destination Node) command.	up to 20 byte ASCII string	a space character
NT	**Node Discover Timeout**. Set/Read the amount of time a node will spend discovering other nodes when ND or DN is issued.	0 - 0xFC [x 100 msec]	0x82 (130d)
NO	Network Discovery Options. Set/Read the options value for the network discovery command. The options bitfield value can change the behavior of the ND (network discovery) command and/or change what optional values are returned in any received ND responses or API node identification frames. Options include: 0x01 = Append DD value (to ND responses or API node identification frames) 0x02 = Local device sends ND response frame when ND is issued. 0x04 = Append RSSI (of the last hop for DigiMesh networks) to ND or FN responses or API node identification frames.	0-0x07 [bitfield]	0
CI	**Cluster Identifier**. Set/read application layer cluster ID value. This value will be used as the cluster ID for all data transmissions. The default value 0x11 (Transparent data cluster ID).	0-0xFFFF	0x11
DE	**Destination Endpoint**. Set/read application layer destination ID value. This value will be used as the destination endpoint for all data transmissions. The default value (0xE8) is the Digi data endpoint.	0-0xFF	0xE8
SE	**Source Endpoint**. Set/read the application layer source endpoint value. This value will be used as the source endpoint for all data transmissions. The default value 0xE8 (Data endpoint) is the Digi data endpoint.	0-0xFF	0xE8

Addressing discovery/configuration

Addressing discovery/configuration commands

AT Command	Name and Description	Parameter Range	Default
AG	**Aggregator Support**. The AG command sends a broadcast through the network that has the following effects on nodes which receive the broadcast: • The receiving node will establish a DigiMesh route back to the originating node, provided there is space in the routing table. • The DH and DL of the receiving node will be updated to the address of the originating node if the AG parameter matches the current DH/DL of the receiving node. • For API-enabled modules on which DH and DL are updated, an aggregate Addressing Update frame will be sent out the serial port. The AG command is only available on products that support DigiMesh.	Any 64-bit number	n/a
DN	**Discover Node - Destination Node**. Resolves an NI (Node Identifier) string to a physical address (case sensitive). The following events occur after the destination node is discovered: <AT Firmware> 1. DL and DH are set to the extended (64-bit) address of the module with the matching NI (Node Identifier) string. 2. OK (or ERROR)\r is returned. 3. Command Mode is exited to allow immediate communication <API Firmware> 0xFFFE and 64-bit extended addresses are returned in an API Command Response frame. If there is no response from a module within (NT * 100) milliseconds or a parameter is not specified (left blank), the command is terminated and an “ERROR” message is returned. In the case of an ERROR, Command Mode is not exited.	20 byte ASCII string	
ND	**Network Discover**. Discovers and reports all RF modules found. If the ND command is sent through a local API frame, each response is returned as a separate Local or Remote AT Command Response API packet, respectively.		

Addressing discovery/configuration commands

AT Command	Name and Description	Parameter Range	Default
FN	**Find Neighbors**. Discovers and reports all RF modules found within immediate RF range. The following information is reported for each module discovered. MY<CR> (always 0xFFFE) SH<CR> SL<CR> NI<CR> (Variable length) PARENT_NETWORK ADDRESS<CR> (2 Bytes) (always 0xFFFE) DEVICE_TYPE<CR> (1 Byte: 0=Coord, 1=Router, 2=End Device) STATUS<CR> (1 Byte: Reserved) PROFILE_ID<CR> (2 Bytes) MANUFACTURER_ID<CR> (2 Bytes) DIGI DEVICE TYPE<CR> (4 Bytes. Optionally included based on NO settings). RSSI OF LAST HOP<DR> (1 Byte. Optionally included based on NO settings.) <CR> If the FN command is issued in command mode, after (NT*100) ms + overhead time, the command ends by returning a <CR>. If the FN command is sent through a local API frame, each response is returned as a separate Local or Remote AT Command Response API packet, respectively. The data consists of the above listed bytes without the carriage return delimiters. The NI string will end in a “0x00” null character.		

Security

Security commands

AT Command	Name and Description	Parameter Range	Default
EE	**Security Enable**. Enables or disables 128-bit AES encryption. This command parameter should be set the same on all devices.	0-1	0
KY	**Security Key**. Sets the 16 byte network security key value. This command is write-only. Attempts to read KY will return an OK status. This command parameter should be set the same on all devices.	128-bit value	n/a

Serial interfacing

Serial interfacing commands

AT Command	Name and Description	Parameter Range	Default
BD	**Baud rate**. Set or read serial interface rate (speed for data transfer between radio modem and host). Values from 0-8 select preset standard rates. Values at 0x7A and above select the actual baud rate, providing the host supports it. The values from 0 to 8 are interpreted as follows: 0 - 1,200 b/s 3 - 9,600 b/s 6 - 57,600 b/s 1 - 2,400 b/s 4 - 19,200 b/s 7 - 111,111 b/s 2 - 4,800 b/s 5 - 38,400 b/s	0-7, and 0x39-0xF4240	0x03 (9600 b/s)

Serial interfacing commands

AT Command	Name and Description	Parameter Range	Default
NB	**Parity**. Set or read parity settings for UART communications. The values from 0 to 4 are interpreted as follows: 0 No parity 3 Forced high parity 1 Even parity 4 Forced low parity 2 Odd parity	0-4	0 (No parity)
RO	**Packetization Timeout**. Set/Read number of character times of inter-character silence required before packetization. Set (RO=0) to transmit characters as they arrive instead of buffering them into one RF packet.	0-0xFF [x character times]	3
FT	**Flow Control Threshold**. Set or read flow control threshold. De-assert CTS and/or send XOFF when FT bytes are in the UART receive buffer. Re-assert CTS when less than FT - 16 bytes are in the UART receive buffer.	0x11-0xEE	0xBE=190d
AP	**API mode**. Set or read the API mode of the radio. The following settings are allowed: 0 API mode is off. All UART input and output is raw data and packets are delineated using the RO parameter. 1API mode is on. All UART input and output data is packetized in the API format, without escape sequences. 2API mode is on with escaped sequences inserted to allow for control characters (XON, XOFF, escape, and the 0x7e delimiter to be passed as data).	0, 1, or 2	0
AO	**API Output Format**. Enables different API output frames. Options include: 0 Standard Data Frames (0x90 for RF RX) 1 Explicit Addressing Data Frames (0x91 for RF RX)	0, 1	0

I/O settings

I/O Settings and Commands

AT Command	Name and Description	Parameter Range	Default
CB	**Commissioning Pushbutton**. This command can be used to simulate commissioning button presses in software. The parameter value should be set to the number of button presses to be simulated. For example, sending the ATCB1 command will execute the action associated with 1 commissioning button press.	0-4	n/a
D0	**AD0/DIO0 Configuration**. Configure options for the AD0/ DIO0 line of the module. Options include: 0 = Input, unmonitored 1 = Commissioning button enable 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0-5	1

I/O Settings and Commands

AT Command	Name and Description	Parameter Range	Default
D1	**AD1/DIO1 Configuration**. Configure options for the AD1/ DIO1 line of the module. Options include: 0 = Input, unmonitored 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
D2	**AD2/DIO2 Configuration**. Configure options for the AD2/ DIO2 line of the module. Options include: 0 = Input, unmonitored 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
D3	**AD3/DIO3 Configuration**. Configure options for the AD3/ DIO3 line of the module. Options include: 0 = Input, unmonitored 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
D4	**AD4/DIO4 Configuration**. Configure options for the AD4/ DIO4 line of the module. Options include: 0 = Input, unmonitored 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
D5	**AD5/DIO5 Configuration**. Configure options for the AD5/ DIO5 line of the module. Options include: 0 = Input, unmonitored 1 = Associate LED 2 = Analog Input 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0-5	1
D6	**DIO6 Configuration**. Configure options for the DIO6 line of the module. Options include: 0 = Input, unmonitored 1 = RTS flow control 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0-1, 3-5	0

I/O Settings and Commands

AT Command	Name and Description	Parameter Range	Default
D7	**DIO7 Configuration**. Configure options for the DIO7 line of the module. Options include: 0 = Input, unmonitored 1 = CTS flow control 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high 6 = RS-485 Tx enable, low TX (0V on transmit, high when idle) 7 = RS-485 Tx enable, high TX (high on transmit, 0V when idle)	0-1, 3-7	1
D8	**DIO8/SLEEP_RQ Configuration**. Configure options for the DIO8/SLEEP_RQ line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high When used as SLEEP_RQ, the D8 parameter should be configured in mode 0 or 3.	0,3-5	0
D9	**DIO9 / ON/$\overline{\text{SLEEP}}$ Configuration**. Configure options for the DIO9/ON/$\overline{\text{SLEEP}}$ line of the module. Options include: 0 = Input, unmonitored 1 = ON/$\overline{\text{SLEEP}}$ 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0,1,3-5	
P0	**DIO10/PWM0 Configuration**. Configure options for the DIO10/PWM0 line of the module. Options include: 0 = Input, unmonitored 1 = RSSI PWM 2 = PWM0 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0-5	1
P1	**DIO11/PWM1 Configuration**. Configure options for the DIO11/PWM1 line of the module. Options include: 0 = Input, unmonitored 2 = PWM1 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 2-5	0
P2	**DIO12 Configuration**. Configure options for the DIO12 line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored 4 = Digital output low 5 = Digital output high	0, 3-5	0

I/O Settings and Commands

<table>
<tr><th>AT Command</th><th>Name and Description</th><th>Parameter Range</th><th>Default</th></tr>
<tr><td>PR</td><td><b>Pull-up Resistor</b>. Set/read the bit field that configures the internal pull-up resistor status for the I/O lines. “1” specifies the pull-up resistor is enabled. “0” specifies no pullup.
<table>
<tr><th>Bit</th><th>I/O Line</th><th>Module pin</th><th>Range</th><th>Notes</th></tr>
<tr><td>0</td><td>DIO4/AD4</td><td>11</td><td>55k - 330k</td><td>1</td></tr>
<tr><td>1</td><td>DIO3/AD3</td><td>17</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>2</td><td>DIO2/AD2</td><td>18</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>3</td><td>DIO1/AD1</td><td>19</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>4</td><td>DIO0/AD0</td><td>20</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>5</td><td>DIO6/RTS</td><td>16</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>6</td><td>DIO8/SLEEP_RQ/DTR</td><td>9</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>7</td><td>DIN/Config</td><td>3</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>8</td><td>DIO5/Associate</td><td>15</td><td>55k - 330k</td><td>1</td></tr>
<tr><td>9</td><td>DIO9/On/Sleep</td><td>13</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>10</td><td>DIO12</td><td>4</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>11</td><td>DIO10/PWM0/RSSI</td><td>6</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>12</td><td>DIO11/PWM1</td><td>7</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>13</td><td>DIO7/CTS</td><td>12</td><td>5k - 23k</td><td>2</td></tr>
<tr><td>14</td><td>DOUT</td><td>2</td><td>5k - 23k</td><td>2</td></tr>
</table>
<b>Note</b> When set as a digital input with pull-up disabled, the leakage can be 9 µA in the worst case and 90 nA in typical case when the line is set externally at a low level.
<b>Note</b> When set as a digital input with pull-up enabled, the voltage of line will stabilize between Vcc-0.65 V and Vcc-0.45 V.</td><td>0 - 0x1FFF</td><td>0x1FFF</td></tr>
<tr><td>M0</td><td><b>PWM0 Output Level</b>. Set/read the output level of the PWM0 line. The line should be configured as a PWM output using the P0 command.</td><td>0-0x03FF</td><td>0</td></tr>
<tr><td>M1</td><td><b>PWM1 Output Level</b>. Set/read the output level of the PWM1 line. The line should be configured as a PWM output using the P1 command.</td><td>0-0x03FF</td><td>0</td></tr>
<tr><td>LT</td><td><b>Assoc LED Blink Time</b>. Set/Read the Associate LED blink time. If the Associate LED functionality is enabled (D5 command), this value determines the on and off blink times for the LED. If LT=0, the default blink rate will be used (500ms sleep coordinator, 250 ms otherwise). For all other LT values, LT is measured in 10 ms.</td><td>0x14-0xFF
(x10ms)</td><td>0</td></tr>
<tr><td>RP</td><td><b>RSSI PWM Timer</b>. Time RSSI signal will be output after last transmission. When RP = 0xFF, output will always be on.</td><td>0-0xFF [x 100 ms]</td><td>0x28
(4 seconds)</td></tr>
</table>

I/O Settings and Commands

AT Command	Name and Description	Parameter Range	Default
IC	**I/O Digital Change Detection**. Set/Read the digital I/O pins to monitor for changes in the I/O state. IC works with the individual pin configuration commands (D0-D9, P0-P2). If a pin is enabled as a digital input/output, the IC command can be used to force an immediate I/O sample transmission when the DIO state changes. IC is a bitmask that can be used to enable or disable edge detection on individual channels. Unused bits should be set to 0. Bit (I/O pin): 0 (DIO0) 1 (DIO1) 2 (DIO2) 3 (DIO3) 4 (DIO4) 5 (DIO5) 6 (DIO6) 7 (DIO7) 8 (DIO8) 9 (DIO9) 10 (DIO10) 11 (DIO11) 12 (DIO12)	0-0xFFFF	0
IF	**Sleep Sample Rate**. Set/read the number of sleep cycles that must elapse between periodic I/O samples. This allows I/O samples to be taken only during some wake cycles. During those cycles I/O samples are taken at the rate specified by IR.	1-0xFF	1
IR	**IO Sample Rate**. Set/Read the I/O sample rate to enable periodic sampling. For periodic sampling to be enabled, IR must be set to a non-zero value, and at least one module pin must have analog or digital I/O functionality enabled (see D0-D9, P0-P2 commands). The sample rate is measured in milliseconds.	0-0xFFFF (ms)	0
IS	**Force Sample**. Forces a read of all enabled digital and analog input lines.	n/a	n/a
1S	**XBee Sensor Sample**. Forces a sample to be taken on an XBee Sensor device. This command can only be issued to an XBee Sensor device using an API remote command.	-	-

Sleep

Sleep commands

AT Command	Name and Description	Parameter Range	Default
SM	**Sleep Mode**. Set/read the sleep mode of the module. 0 - No sleep mode enabled 1 - Pin sleep. In this mode, the sleep/wake state of the module is controlled by the SLEEP_RQ line. 4- Asynchronous cyclic sleep. In this mode, the module periodically sleeps and wakes based on the SP and ST commands. 5- Asynchronous cyclic sleep with pin wake-up. When the SLEEP_REQUEST pin is asserted, the module will enter a cyclic sleep mode similar to (SM=4). When the SLEEP_REQUEST pin is de-asserted, the module will immediately wake up. The module will not sleep when the SLEEP_REQUEST pin is de-asserted. 7- Sleep support mode. 8- Synchronous cyclic sleep mode.	0, 1, 4, 5, 7, 8	0
SO	**Sleep Options**. Set/read the sleep options of the module. This command is a bitmask. For synchronous sleep modules, the following sleep options are defined: bit 0 = Preferred sleep coordinator bit 1 = Non-sleep coordinator bit 2 = Enable API sleep status messages bit bit 3 = Disable early wake-up bit 4 = Enable node type equality bit 5 = Disable lone coordinator sync repeat For asynchronous sleep modules, the following sleep options are defined: bit 8 = Always wake for ST time	Any of the available sleep option bits can be set or cleared. Bit 0 and bit 1 cannot be set at the same time.	0x02
SN	**Number of Sleep Periods**. Set/read the number of sleep periods value. This command controls the number of sleep periods that must elapse between assertions of the ON_SLEEP line during the wake time of asynchronous cyclic sleep. During cycles when the ON_SLEEP line is not asserted, the module will wake up and check for any serial or RF data. If any such data is received, then the ON_SLEEP line will be asserted and the module will fully wake up. Otherwise, the module will return to sleep after checking. This command does not work with synchronous sleep modules.	1-0xFFFF	1
SP	**Sleep Period**. Set/read the sleep period of the module. This command defines the amount of time the module will sleep per cycle.	1-1440000 (x 10 ms)	2 seconds

Sleep commands

AT Command	Name and Description	Parameter Range	Default
ST	**Wake Time**. Set/read the wake period of the module. For asynchronous sleep modules, this command defines the amount of time that the module will stay awake after receiving RF or serial data. For synchronous sleep modules, this command defines the amount of time that the module will stay awake when operating in cyclic sleep mode. This value will be adjusted upwards automatically if it is too small to function properly based on other settings.	0x45-0x36EE80	0x7D0 (2 seconds)
WH	**Wake Host**. Set/Read the wake host timer value. If the wake host timer is set to a non-zero value, this timer specifies a time (in millisecond units) that the device should allow after waking from sleep before sending data out the UART or transmitting an I/O sample. If serial characters are received, the WH timer is stopped immediately. When in synchronous sleep, the device will shorten its sleep period by the value specified by the WH command to ensure that it is prepared to communicate when the network wakes up. When in this sleep mode, the device will always stay awake for the WH time plus the amount of time it takes to transmit a one-hop unicast to another node.	0-0xFFFF (x 1ms)	0

Sleep diagnostics

Diagnostics - sleep status timing

AT Command	Name and Description	Parameter Range	Default
SS	**Sleep Status**. The SS command can be used to query a number of Boolean values describing the status of the module. Bit 0: This bit will be true when the network is in its wake state. Bit 1: This bit will be true if the node is currently acting as a network sleep coordinator. Bit 2: This bit will be true if the node has ever received a valid sync message since the time it was powered on. Bit 3: This bit will be true if the node has received a sync message in the current wake cycle. Bit 4: This bit will be true if the user has altered the sleep settings on the module so that the node will nominate itself and send a sync message with the new settings at the beginning of the next wake cycle. Bit 5: This bit will be true if the user has requested that the node nominate itself as the sleep coordinator (using the commissioning button or the CB2 command). bit 6 = This bit will be true if the node is currently in deployment mode. All other bits: Reserved - All non-documented bits can be any value and should be ignored.	n/a	n/a

Diagnostics - sleep status timing

AT Command	Name and Description	Parameter Range	Default
OS	**Operational Sleep Period**. Read the sleep period that the node is currently using. This number will oftentimes be different from the SP parameter if the node has synchronized with a sleeping router network. Units of 10 msec	n/a	n/a
OW	**Operational Wake Period**. Read the wake time that the node is currently using. This number will oftentimes be different from the ST parameter if the node has synchronized with a sleeping router network. Units of 1 ms	n/a	n/a
MS	**Number of Missed Syncs**. Read the number of wake cycles that have elapsed since the last sync message was received.		
SQ	**Missed Sync Count**. Count of the number of syncs that have been missed. This value can be reset by setting ATSQ to 0. When the value reaches 0xFFFF it will not be incremented anymore.	n/a	n/a

AT command options

AT command options commands

AT Command	Name and Description	Parameter Range	Default
CC	**Command Character**. Set or read the character to be used between guard times of the AT Command Mode Sequence. The AT Command Mode Sequence causes the radio modem to enter Command Mode (from Idle Mode).	0-0xFF	0x2B
CT	**Command Mode Timeout**. Set/Read the period of inactivity (no valid commands received) after which the RF module automatically exits AT Command Mode and returns to Idle Mode.	2-0x1770	0x64 (100d)
CN	**Exit Command Mode**. Explicitly exit the module from AT Command Mode.	-	-
GT	**Guard Times**. Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT + CC + GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	0-0xFFFF	0x3E8 (1000d)

Firmware commands

Firmware version/information

AT Command	Name and Description	Parameter Range	Default
VL	**Version Long**. Shows detailed version information including application build date and time.	-	-
VR	**Firmware Version**. Read firmware version of the module.	0-0xFFFFFFFF [read-only]	Firmware-set
HV	**Hardware Version**. Read hardware version of the module.	0-0xFFFF [read-only]	Factory-set
DD	**Device Type Identifier**. Stores a device type value. This value can be used to differentiate multiple XBee-based products.	0-0xFFFFFFFF [read only]	0x40000
HP	**Hopping Channel**. Set/read the spread spectrum channel on which the module communicates. Separate channels minimize interference between multiple networks operating in the same vicinity.	0-0x7 (standard variant) 0-0x3 (international variant)	0
NP	**Maximum RF Payload Bytes**. This value returns the maximum number of RF payload bytes that can be sent in a unicast transmission based on the current configurations.	0-0xFFFF	n/a
CK	**Configuration Code**. Read the configuration code associated with the current AT command configuration.The code returned can be used as a quick check to determine if a node has been configured as desired.	0-0xFFFFFFFF	n/a

API operation

As an alternative to Transparent Operation, Application Programming Interface (API) Operations are available. API operation requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and module status messages are sent and received from the module using a UART data frame.

Digi may add new API frames to future versions of firmware, so build the ability to filter out additional API frames with unknown API identifiers into your software interface.

API frame specifications

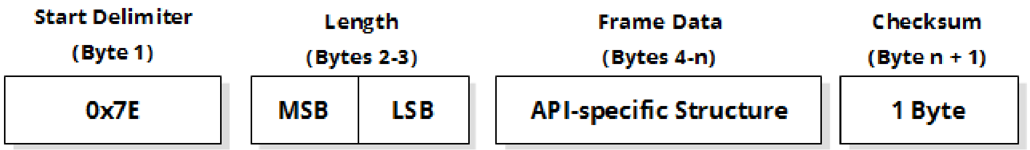
Two API modes are supported and both can be enabled using the AP (API Enable) command. Use the following AP parameter values to configure the module to operate in a particular mode:

- AP = 1: API Operation
- AP = 2: API Operation (with escaped characters)

API operation (AP parameter = 1)

When this API mode is enabled (AP = 1), the UART data frame structure is defined as follows:

Figure 9: UART data frame structure.



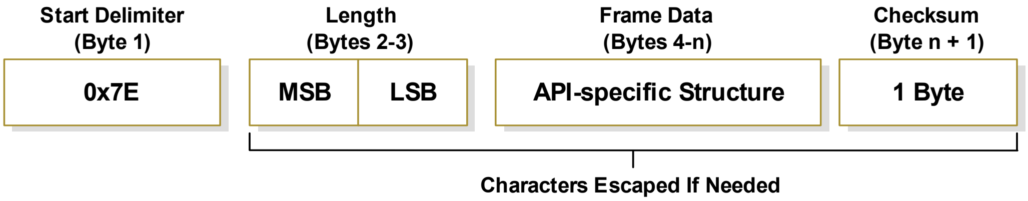
MSB = Most Significant Byte, LSB = Least Significant Byte

Any data received prior to the start delimiter is silently discarded. If the frame is not received correctly or if the checksum fails, the module will discard the packet.

API operation - with escape characters (AP parameter = 2)

When this API mode is enabled (AP = 2), the UART data frame structure is defined as follows:

Figure 10: UART data frame structure with escape control characters.



MSB = Most Significant Byte, LSB = Least Significant Byte

Escape characters

When sending or receiving a UART data frame, specific data values must be escaped (flagged) so they do not interfere with the data frame sequencing. To escape an interfering data byte, insert 0x7D and follow it with the byte to be escaped XOR'd with 0x20.

Data bytes that need to be escaped:

- 0x7E – Frame Delimiter
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

Example - Raw UART Data Frame (before escaping interfering bytes):

0x7E 0x00 0x02 0x23 0x11 0xCB

0x11 needs to be escaped which results in the following frame:

0x7E 0x00 0x02 0x23 0x7D 0x31 0xCB

Note In the above example, the length of the raw data (excluding the checksum) is 0x0002 and the checksum of the non-escaped data (excluding frame delimiter and length) is calculated as: 0xFF - (0x23 + 0x11) = (0xFF - 0x34) = 0xCB.

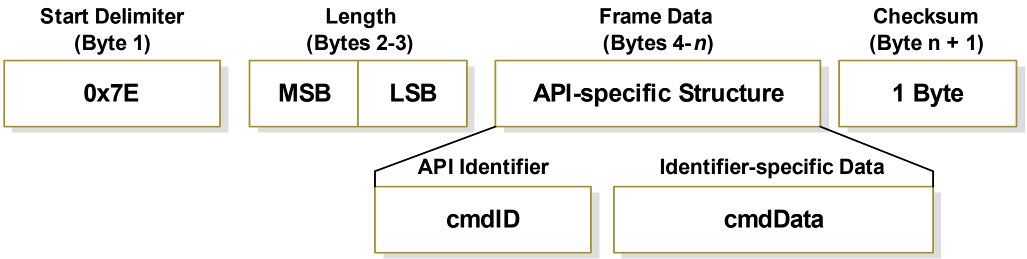
Length

The length field has two-byte value that specifies the number of bytes that will be contained in the frame data field. It does not include the checksum field.

Frame data

Frame data of the UART data frame forms an API-specific structure as follows:

Figure 11: UART data frame and API-specific structure.



The cmdID frame (API-identifier) indicates which API messages will be contained in the cmdData frame (Identifier-specific data). Note that multi-byte values are sent big endian.The XBee modules support the following API frames:

Table 5: API frame names and values

API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
Transmit Request	0x10
Explicit Addressing Command Frame	0x11
Remote Command Request	0x17
AT Command Response	0x88
Modem Status	0x8A
Transmit Status	0x8B
Receive Packet (AO=0)	0x90
Explicit Rx Indicator (AO=1)	0x91
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97

Checksum

To test data integrity, a checksum is calculated and verified on non-escaped data.

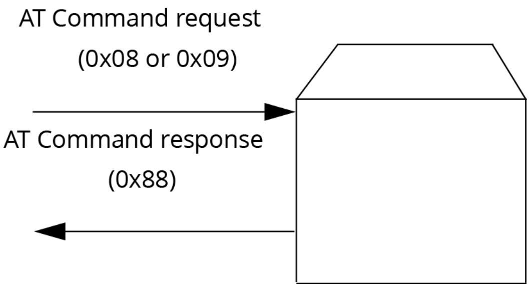
To calculate: Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract the result from 0xFF.

To verify: Add all bytes (include checksum, but not the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

API UART exchanges

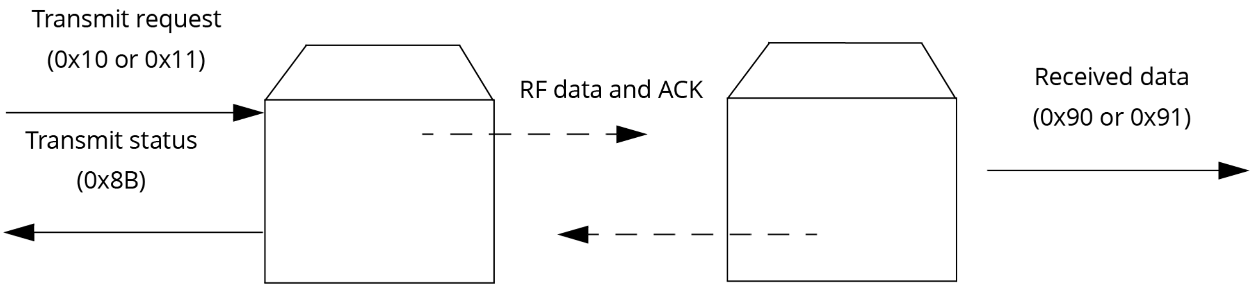
AT commands

The following image shows the API frame exchange that takes place at the serial interface when sending an AT command request to read or set a module parameter. The response can be disabled by setting the frame ID to 0 in the request.



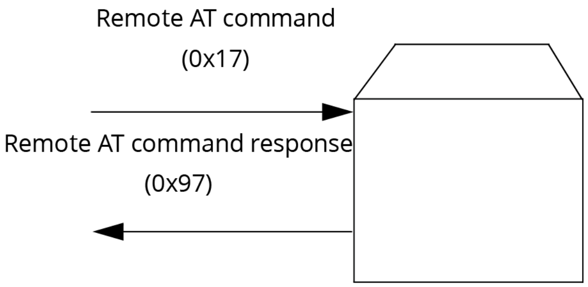
Transmitting and receiving RF data

The following image shows the API exchanges that take place at the serial interface when sending RF data to another device. The transmit status frame is always sent at the end of a data transmission unless the frame ID is set to 0 in the TX request. If the packet cannot be delivered to the destination, the transmit status frame will indicate the cause of failure. The received data frame (0x90 or 0x91) is set by the AP command.



Remote AT commands

The following image shows the API frame exchanges that take place at the serial interface when sending a remote AT command. A remote command response frame is not sent out the serial interface if the remote device does not receive the remote command.



Supporting the API

Applications that support the API should make provisions to deal with new API frames that may be introduced in future releases. For example, a section of code on a host microprocessor that handles received serial API frames (sent out the module's DOUT pin) might look like this:

```
void XBee_HandleRxAPIFrame(_apiFrameUnion *papiFrame){
   switch(papiFrame->api_id){
       case RX_RF_DATA_FRAME:
          //process received RF data frame
          break;


       case RX_IO_SAMPLE_FRAME:
          //process IO sample frame
          break;


       case NODE_IDENTIFICATION_FRAME:
          //process node identification frame
          break;


       default:
          //Discard any other API frame types that are not being used
          break;
   }
}
```

Frame data

The following sections illustrate the types of frames encountered while using the API.

AT command

Frame type: 0x08

Used to query or set module parameters on the local device. This API command applies changes after executing the command. (Changes made to module parameters take effect once changes are applied.) The API example below illustrates an API frame when modifying the NH parameter value of the module.

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x04	
Frame-specific Data	Frame Type	3	0x08	
	Frame ID	4	0x52 (R)	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgment). If set to 0, no response is sent.
	AT Command	5	0x4E (N)	Command Name - Two ASCII characters that identify the AT Command.
		6	0x48 (H)	
	Parameter Value (optional)			If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
Checksum		8	0x0F	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

The example above illustrates an AT command when querying an NH value.

AT command - Queue Parameter Value

Frame type: 0x09

This API type allows module parameters to be queried or set. In contrast to the “AT Command” API type, new parameter values are queued and not applied until either the “AT Command” (0x08) API type or the Apply Changes (AC) command is issued. Register queries (reading parameter values) are returned immediately.

Example: Send a command to change the baud rate (BD) to 115200 baud, but don't apply changes yet. (Module will continue to operate at the previous baud rate until changes are applied).

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x05	

Frame Fields		Offset	Example	Description
Frame-specific Data	Frame Type	3	0x09	
	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgment). If set to 0, no response is sent.
	AT Command	5	0x42 (B)	Command Name - two ASCII characters that identify the AT Command.
		6	0x44 (D)	
	Parameter Value (ATBD7 = 115200 baud)		0x07	If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
Checksum		8	0x68	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Note In this example, the parameter could have been sent as a zero-padded 2-byte or 4-byte value.

Transmit request

Frame type: 0x10

A Transmit Request API frame causes the module to send data as an RF packet to the specified destination.

The 64-bit destination address should be set to 0x000000000000FFFF for a broadcast transmission (to all devices). For unicast transmissions the 64 bit address field should be set to the address of the desired destination node. The reserved field should be set to 0xFFFE.

This example shows if escaping is disabled (AP=1).

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x16	
Frame-specific Data	Frame Type	3	0x10	
	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgment). If set to 0, no response is sent.
	64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF - Broadcast address
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x0A	
		11	0x01	
		LSB 12	0x27	
	Reserved	13	0xFF	Set to 0xFFFE.
		14	0xFE	
	Broadcast Radius	15	0x00	Sets maximum number of hops a broadcast transmission can occur. If set to 0, the broadcast radius will be set to the maximum hops value.
	Transmit Options	16	0x00	Bitfield: bit 0: Disable ACK bit 1: Don't attempt route Discovery. All other bits must be set to 0.
	RF Data	17	0x54	Data that is sent to the destination device.
		18	0x78	
		19	0x44	
		20	0x61	
		21	0x74	
		22	0x61	
		23	0x30	
		24	0x41	
Checksum		25	0x13	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: The example above shows how to send a transmission to a module where escaping is disabled (AP=1) with destination address 0x0013A200 400A0127, payload “TxData0A”. If escaping is enabled (AP=2), the frame should look like:

0x7E 0x00 0x16 0x10 0x01 0x00 0x7D 0x33 0xA2 0x00 0x40 0x0A 0x01 0x27

0xFF 0xFE 0x00 0x00 0x54 0x78 0x44 0x61 0x74 0x61 0x30 0x41 0x7D 0x33

The checksum is calculated (on all non-escaped bytes) as [0xFF - (sum of all bytes from API frame type through data payload)].

Explicit Addressing Command frame

Frame type: 0x11

Allows application layer fields (endpoint and cluster ID) to be specified for a data transmission. Similar to the Transmit Request, but also requires application layer addressing fields to be specified (endpoints, cluster ID, profile ID). An Explicit Addressing Request API frame causes the module to send data as an RF packet to the specified destination, using the specified source and destination endpoints, cluster ID, and profile ID.

The 64-bit destination address should be set to 0x000000000000FFFF for a broadcast transmission (to all devices). For unicast transmissions the 64 bit address field should be set to the address of the desired destination node. The reserved field should be set to 0xFFFE.

The broadcast radius can be set from 0 up to NH to 0xFF. If the broadcast radius exceeds the value of NH then the value of NH will be used as the radius. This parameter is only used for broadcast transmissions.

The maximum number of payload bytes can be read with the NP command.

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x1A	
Frame-specific data	Frame Type	3	0x11	
	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgment). If set to 0, no response is sent.
	64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF - Broadcast address
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x01	
		10	0x23	
		11	0x84	
		LSB12	0x00	
	Reserved	13	0xFF	Set to 0xFFFE.
		14	0xFE	
	Source Endpoint	15	0xA0	Source endpoint for the transmission.
	Destination Endpoint	16	0xA1	Destination endpoint for the transmission.
	Cluster ID	17	0x15	Cluster ID used in the transmission.
		18	0x54	
	Profile ID	19	0xC1	Profile ID used in the transmission.
		20	0x05	
	Broadcast Radius	21	0x00	Sets the maximum number of hops a broadcast transmission can traverse. If set to 0, the transmission radius will be set to the network maximum hops value.
	Transmit Options	22	0x00	Bitfield: bit 0: Disable ACK bit 1: Don't attempt route Discovery. All other bits must be set to 0.

Frame Fields		Offset	Example	Description
Frame-specific data	Data Payload	23	0x54	
		24	0x78	
		25	0x44	
		26	0x61	
		27	0x74	
		28	0x61	
Checksum		29	0xDD	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: The above example sends a data transmission to a radio with a 64 bit address of 0x0013A20001238400 using a source endpoint of 0xA0, destination endpoint 0xA1, cluster ID =0x1554, and profile ID 0xC105. Payload will be “TxData”.

Remote AT Command request

Frame type: 0x17

Used to query or set module parameters on a remote device. For parameter changes on the remote device to take effect, changes must be applied, either by setting the apply changes options bit, or by sending an AC command to the remote.

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x10	

Frame Fields		Offset	Example	Description
Frame-specific Data	Frame Type	3	0x17	
	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgment). If set to 0, no response is sent.
	64-bit Destination Address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF - Broadcast address.
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x40	
		11	0x11	
		LSB 12	0x22	
	Reserved	13	0xFF	Set to 0xFFFE.
		14	0xFE	
	Remote Command Options	15	0x02 (apply changes)	0x02 - Apply changes on remote. If not set, AC command must be sent before changes will take effect. All other bits must be set to 0.
	AT Command	16	0x42 (B)	Name of the command.
		17	0x48 (H)	
Frame-specific data	Command Parameter	18	0x01	If present, indicates the requested parameter value to set the given register. If no characters present, the register is queried.
Checksum		18	0xF5	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: The above example sends a remote command to change the broadcast hops register on a remote device to 1 (broadcasts go to 1-hop neighbors only), and apply changes so the new configuration value immediately takes effect. In this example, the 64-bit address of the remote is 0x0013A200 40401122.

AT Command Response

Frame type: 0x88

In response to an AT Command message, the module will send an AT Command Response message. Some commands will send back multiple frames (for example, the ND (Node Discover) command).

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x05	

Frame Fields		Offset	Example	Description
Frame-specific Data	Frame Type	3	0x88	
	Frame ID	4	0x01	Identifies the UART data frame being reported. **Note** If Frame ID = 0 in AT Command Mode, no AT Command Response will be given.
	AT Command	5	‘B’ = 0x42	Command Name - Two ASCII characters that identify the AT Command.
		6	‘D’ = 0x44	
	Command Status	7	0x00	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter
	Command Data			Register data in binary format. If the register was set, then this field is not returned, as in this example.
Checksum		8	0xF0	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: Suppose the BD parameter is changed on the local device with a frame ID of 0x01. If successful (parameter was valid), the above response would be received.

Modem Status

Frame type: (0x8A)

RF module status messages are sent from the module in response to specific conditions.

Example: The following API frame is returned when an API device powers up.

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x02	
Frame-specific Data	Frame Type	3	0x8A	
	Status	4	0x00	0x00 = Hardware reset 0x01= Watchdog timer reset 0x0B = Network Woke Up 0x0C = Network Went To Sleep
Checksum		5	0x75	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Transmit Status

Frame type: 0x8B

When a TX Request is completed, the module sends a TX Status message. This message will indicate if the packet was transmitted successfully or if there was a failure.

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x07	
Frame-specific Data	Frame Type	3	0x8B	
	Reserved	4	0x47	Reserved.
		5	0xFF	
		6	0xFE	
	Transmit Retry Count	7	0x00	The number of application transmission retries that took place.
	Delivery Status	8	0x00	0x00 = Success 0x01 = MAC ACK Failure 0x15 = Invalid destination endpoint 0x21 = Network ACK Failure 0x25 = Route Not Found 0x74 = Data Payload Too Large
	Discovery Status	9	0x02	0x00 = No Discovery Overhead 0x02 = Route Discovery
Checksum		10	0x2E	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: In the above example, a unicast data transmission was sent successfully to a destination device.

Receive Packet

Frame type: (0x90)

When the module receives an RF packet, it is sent out the UART using this message type.

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x12	
Frame-specific Data	Frame Type	3	0x90	
	64-bit Source Address	MSB 4	0x00	64-bit address of sender.
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		LSB 11	0xAA	
	Reserved	12	0xFF	Reserved.
		13	0xFE	
	Receive Options	14	0x01	0x01 - Packet acknowledged. 0x02 - Packet was a broadcast packet.
	Received Data	15	0x52	Received RF data.
		16	0x78	
		17	0x44	
		18	0x61	
		19	0x74	
		20	0x61	
Checksum		21	0x11	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: In the above example, a device with a 64-bit address of 0x0013A200 40522BAA sends a unicast data transmission to a remote device with payload “RxData”. If AO=0 on the receiving device, it would send the above frame out its UART.

Explicit Rx Indicator

Frame type:0x91

When the modem receives an RF packet it is sent out the UART using this message type (when AO=1).

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x18	
Frame-specific Data	Frame Type	3	0x91	
	64-bit Source Address	MSB 4	0x00	64-bit address of sender.
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		LSB 11	0xAA	
	Reserved	12	0xFF	Reserved.
		13	0xFE	
	Source Endpoint	14	0xE0	Endpoint of the source that initiated the transmission.
	Destination Endpoint	15	0xE0	Endpoint of the destination the message is addressed to.
	Cluster ID	16	0x22	Cluster ID the packet was addressed to.
		17	0x11	
	Profile ID	18	0xC1	Profile ID the packet was addressed to.
		19	0x05	
	Receive Options	20	0x02	0x01 – Packet acknowledged 0x02 – Packet was a broadcast packet
	Received Data	21	0x52	Received RF data.
		22	0x78	
		23	0x44	
		24	0x61	
		25	0x74	
		26	0x61	
Checksum		27	0x56	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: In the above example, a device with a 64-bit address of 0x0013A200 40522BAA sends a broadcast data transmission to a remote device with payload “RxData”. Suppose the transmission was sent with source and destination endpoints of 0xE0, cluster ID=0x2211, and profile ID=0xC105. If AO=1 on the receiving device, it would send the above frame out its UART.

Data Sample Rx Indicator

Frame type:0x92

When the modem receives an RF packet it is sent out the UART using this message type (when AO=1).

Frame Fields		Offset	Example	Description
Start		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x14	
Frame-specific Data	Frame Type	3	0x92	
	64-bit Source Address	MSB 4	0x00	64-bit address of sender.
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		LSB 11	0xAA	
	16-bit Source Network Address	MSB 12	0x7D	16-bit address of sender.
		LSB 13	0x84	
	Receive Options	14	0x01	0x01 - Packet acknowledged 0x02 - Packet was a broadcast packet
	Number of samples	15	0x01	Number of sample sets included in the payload. Always set to 1.
	Digital Channel Mask*	16	0x00	Bitmask field that indicates which digital IO lines on the remote have sampling enabled (if any).
		17	0x1C	
	Analog Channel Mask**	18	0x02	Bitmask field that indicates which analog IO lines on the remote have sampling enabled (if any).
	Digital Samples (if included)	19	0x00	If the sample set includes any digital IO lines (Digital Channel Mask > 0), these two bytes contain samples for all enabled digital IO lines. DIO lines that do not have sampling enabled return 0. Bits in these 2 bytes map the same as they do in the Digital Channels Mask field.
		20	0x14	

Frame Fields		Offset	Example	Description
Frame-specific data	Analog Sample	21	0x02	If the sample set includes any analog input lines (Analog Channel Mask > 0), each enabled analog input returns a 2-byte value indicating the A/D measurement of that input. Analog samples are ordered sequentially from AD0/DIO0 to AD3/DIO3, to the supply voltage.
		22	0x25	
Checksum		23	0xF5	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Node Identification Indicator

Frame type:0x95

This frame is received when a module transmits a node identification message to identify itself (when AO=0). The data portion of this frame is similar to a network discovery response frame (see ND command).

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x20	
Frame-specific Data	Frame Type	3	0x95	
	64-bit Source Address	MSB 4	0x00	64-bit address of sender.
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		LSB 11	0xAA	
	16-bit Source Network Address	MSB 12	0xFF	Set to 0xFFFE.
		LSB 13	0xFE	
	Receive Options	14	0x02	0x01 - Packet acknowledged 0x02 - Packet was a broadcast packet
	Source 16-bit address	15	0xFF	Set to 0xFFFE.
		16	0xFE	

Frame Fields		Offset	Example	Description
Frame-specific data	64-bitNetwork address	17	0x00	Indicates the 64-bit address of the remote module that transmitted the node identification frame.
		18	0x13	
		19	0xA2	
		20	0x00	
		21	0x40	
		22	0x52	
		23	0x2B	
		24	0xAA	
	NI String	25	0x20	Node identifier string on the remote device. The NI- String is terminated with a NULL byte (0x00).
		26	0x00	
	Parent 16-bit address	27	0xFF	Set to 0xFFFE.
		28	0xFE	
Checksum		29	0xF4	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: If the commissioning push button is pressed on a remote router device with 64-bit address 0x0013A200 40522BAA and default NI string, the following node identification indicator would be received.

Remote Command Response

Frame type: 0x97

If a module receives a remote command response RF data frame in response to a Remote AT Command Request, the module will send a Remote AT Command Response message out the UART. Some commands may send back multiple frames--for example, Node Discover (ND) command.

Frame Fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x13	
Frame-specific Data	Frame Type	3	0x97	
	Frame ID	4	0x55	This is the same value passed in to the request.
	64-bit Source (remote) Address	MSB 5	0x00	The address of the remote radio returning this response.
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x52	
		11	0x2B	
		LSB 12	0xAA	
	Reserved	13	0xFF	Reserved
		14	0xFE	
	AT Commands	15	0x53	Name of the command.
		16	0x4C	
	Command Status	17	0x00	0 = OK 1 = ERROR 2 = Invalid command 3 = Invalid parameter 4 = Transmit failure
	Command Data	18	0x40	The value of the required register.
		19	0x52	
		20	0x2B	
		21	0xAA	
Checksum		22	0xF4	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: If a remote command is sent to a remote device with 64-bit address 0x0013A200 40522BAA to query the SL command, and if the frame ID=0x55, the response would look like the above example.

Definitions

Terms and definitions

Term	Definition
PAN	Personal Area Network. A data communication network that includes a coordinator and one or more routers/end devices.
Network address	The 16-bit address assigned to a node after it has joined to another node. The coordinator always has a network address of 0.
Route request	Broadcast transmission sent by a coordinator or router throughout the network in attempt to establish a route to a destination node.
Route reply	Unicast transmission sent back to the originator of the route request. It is initiated by a node when it receives a route request packet and its address matches the Destination Address in the route request packet.
Route discovery	The process of establishing a route to a destination node when one does not exist in the Routing Table. It is based on the Ad-hoc On-demand Distance Vector routing (AODV) protocol.
DigiMesh Protocol	
Election	An election takes place to resolve which node will function as the network's sleep coordinator if multiple nodes nominate themselves at the same time.
Hopping	One direct host-to-host connection forming part of the route between hosts.
Network identifier	A user configurable string used to identify a node apart from its address.
Network address	The 64-bit address assigned to a node after it has joined to another node.
Nomination	Nomination is the process where a node becomes a sleep coordinator.
Route request	Broadcast transmission sent by a coordinator or router throughout the network in attempt to establish a route to a destination node.
Route reply	Unicast transmission sent back to the originator of the route request. It is initiated by a node when it receives a route request packet and its address matches the Destination Address in the route request packet.
Route discovery	The process of establishing a route to a destination node when one does not exist in the Routing Table. It is based on the AODV protocol.
Sleep coordinator	Node used to send sync messages in a cyclic sleeping network.
Sync message	A transmission used in a cyclic sleeping network to maintain synchronization.

Agency certifications

United States FCC

The XBee-PRO® 900 RF Module complies with Part 15 of the FCC rules and regulations. Compliance with the labeling requirements, FCC notices and antenna usage guidelines is required.

To fulfill FCC Certification, the OEM must comply with the following regulations:

1. The system integrator must ensure that the text on the external label provided with this device is placed on the outside of the final product.
2. The XBee-PRO 900 RF Module may only be used with antennas that have been tested and approved for use with this module; refer to the antenna tables in this section.

OEM labeling requirements

The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the final product enclosure that displays the contents shown in the figure below.

Required FCC Label for OEM products containing the XBee-PRO 900 RF Module:

> Contains FCC ID:MCQ-XBEE09P
>
> The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (i.) this device may not cause harmful interference and (ii.) this device must accept any interference received, including interference that may cause undesired operation.

FCC notices

IMPORTANT: The XBee-PRO 900 RF Module has been certified by the FCC for use with other products without any further certification (as per FCC section 2.1091). Modifications not expressly approved by Digi could void the user's authority to operate the equipment.

IMPORTANT: OEMs must test final product to comply with unintentional radiators (FCC section 15.107 & 15.109) before declaring compliance of their final product to Part 15 of the FCC Rules.

IMPORTANT: The RF module has been certified for remote and base radio applications. If the module will be used for portable applications, the device must undergo SAR testing.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: Re-orient or relocate the receiving antenna, Increase the separation between the equipment and receiver, Connect equipment and receiver to outlets on different circuits, or consult the dealer or an experienced radio/TV technician for help.

FCC-approved antennas (900 MHz)

The XBee-PRO 900 RF Module can be installed using antennas and cables constructed with standard connectors (Type-N, SMA, TNC, etc.) if the installation is performed professionally and according to FCC guidelines. For installations not performed by a professional, non-standard connectors (RPSMA, RPTNC, etc.) must be used.

The modules are FCC approved for fixed base station and mobile applications. If the antenna is mounted at least 20cm (8 in.) from nearby persons, the application is considered a mobile application. Antennas not listed in the table must be tested to comply with FCC Section 15.203 (Unique Antenna Connectors) and Section 15.247 (Emissions). XBee-PRO 900 have been tested and approved for use with all the antennas listed in Antennas: 900 MHz on page 91.

*** If using the RF module in a portable application** (For example - If the module is used in a handheld device and the antenna is less than 20cm from the human body when the device is in operation): The integrator is responsible for passing additional SAR (Specific Absorption Rate) testing based on FCC rules 2.1091 and FCC Guidelines for Human Exposure to Radio Frequency Electromagnetic Fields, OET Bulletin and Supplement C. The testing results will be submitted to the FCC for approval prior to selling the integrated unit. The required SAR testing measures emissions from the module and how they affect the person.

RF exposure

CAUTION! To satisfy FCC RF exposure requirements for mobile transmitting devices, a separation distance of 20 cm or more should be maintained between the antenna of this device and persons during device operation. To ensure compliance, operations at closer than this distance are not recommended. The antenna used for this transmitter must not be co-located in conjunction with any other antenna or transmitter.

The preceding statement must be included as a CAUTION statement in OEM product manuals in order to alert users of FCC RF Exposure compliance.

Canada (IC)

Labeling requirements for Industry Canada are similar to those of the FCC. A clearly visible label on the outside of the final product enclosure must display the following text:

Contains Model: XBEE09P, IC: 1846A-XBEE09P

Integrator is responsible for its product to comply with IC ICES-003 & FCC Part 15, Sub. B - Unintentional Radiators. ICES-003 is the same as FCC Part 15 Sub. B and Industry Canada accepts FCC test report or CISPR 22 test report for compliance with ICES-003.

Transmitter antennas

This device has been designed to operate with the antennas listed in Antennas: 900 MHz on page 91, and having a maximum gain of 15.1 dB. Antennas not included in this list or having a gain greater than 15.1 dB are strictly prohibited for use with this device. The required antenna impedance is 50 ohms.

Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that permitted for successful communication.

Australia (C-Tick)

These products comply with requirements to be used in end products in Australia. All products with EMC and radio communications must have a registered C-Tick mark. Registration to use the compliance mark will only be accepted from Australian manufacturers or importers, or their agent, in Australia.

Labeling requirements

In order to have a C-Tick mark on an end product, a company must comply with (a) or (b) below:

a. Have a company presence in Australia

b. Have a company/distributor/agent in Australia that will sponsor the importing of the end product

Contact Digi for questions related to locating a contact in Australia.

Antennas: 900 MHz

The following antennas have been approved for use with the XBee-PRO 900 RF Module. Digi does not carry all of these antenna variants. Contact Digi Sales for available antennas.

Part Number	Type (Description)	Connector	Gain	Application
900 MHz Antenna Listings				
A09-F0	Fiberglass Base	RPN	0 dBi	Fixed
A09-F1	Fiberglass Base	RPN	1.0 dBi	Fixed
A09-F2	Fiberglass Base	RPN	2.1 dBi	Fixed
A09-F3	Fiberglass Base	RPN	3.1 dBi	Fixed
A09-F4	Fiberglass Base	RPN	4.1 dBi	Fixed
A09-F5	Fiberglass Base	RPN	5.1 dBi	Fixed
A09-F6	Fiberglass Base	RPN	6.1 dBi	Fixed
A09-F7	Fiberglass Base	RPN	7.1 dBi	Fixed
A09-F8	Fiberglass Base	RPN	8.1 dBi	Fixed
A09-F9	Base Station	RPSMAF	9.2 dBi	Fixed
A09-W7	Wire Base Station	RPN	7.1 dBi	Fixed
A09-F0	Fiberglass Base	RPSMA	0 dBi	Fixed

Part Number	Type (Description)	Connector	Gain	Application
A09-F1	Fiberglass Base	RPSMA	1.0 dBi	Fixed
A09-F2	Fiberglass Base	RPSMA	2.1 dBi	Fixed
A09-F3	Fiberglass Base	RPSMA	3.1 dBi	Fixed
A09-F4	Fiberglass Base	RPSMA	4.1 dBi	Fixed
A09-F5	Fiberglass Base	RPSMA	5.1 dBi	Fixed
A09-F6	Fiberglass Base	RPSMA	6.1 dBi	Fixed
A09-F7	Fiberglass Base	RPSMA	7.1 dBi	Fixed
A09-F8	Fiberglass Base	RPSMA	8.1 dBi	Fixed
A09-M7	Base Station	RPSMAF	7.2 dBi	Fixed
A09-W7SM	Wire Base Station	RPSMA	7.1 dBi	Fixed
A09-F0TM	Fiberglass Base	RPTNC	0 dBi	Fixed
A09-F1TM	Fiberglass Base	RPTNC	1.0 dBi	Fixed
A09-F2TM	Fiberglass Base	RPTNC	2.1 dBi	Fixed
A09-F3TM	Fiberglass Base	RPTNC	3.1 dBi	Fixed
A09-F4TM	Fiberglass Base	RPTNC	4.1 dBi	Fixed
A09-F5TM	Fiberglass Base	RPTNC	5.1 dBi	Fixed
A09-F6TM	Fiberglass Base	RPTNC	6.1 dBi	Fixed
A09-F7TM	Fiberglass Base	RPTNC	7.1 dBi	Fixed
A09-F8TM	Fiberglass Base	RPTNC	8.1 dBi	Fixed
A09-W7TM	Wire Base Station	RPTNC	7.1 dBi	Fixed
A09-HSM-7	Straight half-wave	RPSMA	3.0 dBi	Fixed / Mobile
A09-HASM-675	Articulated half-wave	RPSMA	2.1 dBi	Fixed / Mobile
A09-HABMM-P6I	Articulated half-wave	MMCX	2.1 dBi	Fixed / Mobile
A09-HABMM-6-P6I	Articulated half-wave	MMCX	2.1 dBi	Fixed / Mobile
A09-HBMM-P6I	Straight half-wave w/	MMCX	2.1 dBi	Fixed / Mobile
A09-HRSM	Right angle half-wave	RPSMA	2.1 dBi	Fixed
A09-HASM-7	Articulated half-wave	RPSMA	2.1 dBi	Fixed
A09-HG	Glass mounted half-	RPSMA	2.1 dBi	Fixed
A09-HATM	Articulated half-wave	RPTNC	2.1 dBi	Fixed
A09-H	Half-wave dipole	RPSMA	2.1 dBi	Fixed

Part Number	Type (Description)	Connector	Gain	Application
A09-HBMMP6I	1/2 wave antenna	MMCX	2.1dBi	Mobile
A09-QBMMP6I	1/4 wave antenna	MMCX	1.9 dBi	Mobile
A09-QI	1/4 wave integrated wire antenna	Integrated	1.9 dBi	Mobile
29000187	Helical	Integrated	-2.0 dBi	Fixed/Mobile
A09-QW	Quarter-wave wire	Permanent	1.9 dBi	Fixed / Mobile
A09-QRAMM	3" Quarter-wave wire	MMCX	2.1 dBi	Fixed / Mobile
A09-QSM-3	Quarter-wave straight	RPSMA	1.9 dBi	Fixed / Mobile
A09-QSM-3H	Heavy duty quarter-	RPSMA	1.9 dBi	Fixed / Mobile
A09-QBMM-P6I	Quarter-wave w/ 6"	MMCX	1.9 dBi	Fixed / Mobile
A09-QHRN	Miniature Helical Right	Permanent	-1 dBi	Fixed / Mobile
A09-QHSN	Miniature Helical Right	Permanent	-1 dBi	Fixed / Mobile
A09-QHSM-2	2" Straight	RPSMA	1.9 dBi	Fixed / Mobile
A09-QHRSM-2	2" Right angle	RPSMA	1.9 dBi	Fixed / Mobile
A09-QHRSM-170	1.7" Right angle	RPSMA	1.9 dBi	Fixed / Mobile
A09-QRSM-380	3.8" Right angle	RPSMA	1.9 dBi	Fixed / Mobile
A09-QAPM-520	5.2" Articulated Screw	Permanent	1.9 dBi	Fixed / Mobile
A09-QSPM-3	3" Straight screw	Permanent	1.9 dBi	Fixed / Mobile
A09-QAPM-3	3" Articulated screw	Permanent	1.9 dBi	Fixed / Mobile
A09-QAPM-3H	3" Articulated screw	Permanent	1.9 dBi	Fixed / Mobile
A09-DPSM-P12F	omni directional	RPSMA	3.0 dBi	Fixed
A09-D3NF-P12F	omni directional	RPN	3.0 dBi	Fixed
A09-D3SM-P12F	omni directional w/ 12ft	RPSMA	3.0 dBi	Fixed
A09-D3PNF	omni directional	RPN	3.0 dBi	Fixed
A09-D3TM-P12F	omni directional w/ 12ft	RPTNC	3.0 dBi	Fixed
A09-D3PTM	omni directional	RPTNC	3.0 dBi	Fixed
A09-M0SM	Mag Mount	RPSMA	0 dBi	Fixed
A09-M2SM	Mag Mount	RPSMA	2.1 dBi	Fixed
A09-M3SM	Mag Mount	RPSMA	3.1 dBi	Fixed
A09-M5SM	Mag Mount	RPSMA	5.1 dBi	Fixed
A09-M7SM	Mag Mount	RPSMA	7.1 dBi	Fixed

Part Number	Type (Description)	Connector	Gain	Application
A09-M8SM	Mag Mount	RPSMA	8.1 dBi	Fixed
A09-M0TM	Mag Mount	RPTNC	0 dBi	Fixed
A09-M2TM	Mag Mount	RPTNC	2.1 dBi	Fixed
A09-M3TM	Mag Mount	RPTNC	3.1 dBi	Fixed
A09-M5TM	Mag Mount	RPTNC	5.1 dBi	Fixed
A09-M7TM	Mag Mount	RPTNC	7.1 dBi	Fixed
A09-M8TM	Mag Mount	RPTNC	8.1 dBi	Fixed
A09-Y6	2 Element Yagi	RPN	6.1 dBi	Fixed / Mobile
A09-Y7	3 Element Yagi	RPN	7.1 dBi	Fixed / Mobile
A09-Y8	4 Element Yagi	RPN	8.1 dBi	Fixed / Mobile
A09-Y9	4 Element Yagi	RPN	9.1 dBi	Fixed / Mobile
A09-Y10	5 Element Yagi	RPN	10.1 dBi	Fixed / Mobile
A09-Y11	6 Element Yagi	RPN	11.1 dBi	Fixed / Mobile
A09-Y12	7 Element Yagi	RPN	12.1 dBi	Fixed / Mobile
A09-Y13	9 Element Yagi	RPN	13.1 dBi	Fixed / Mobile
A09-Y14	10 Element Yagi	RPN	14.1 dBi	Fixed / Mobile
A09-Y14	12 Element Yagi	RPN	14.1 dBi	Fixed / Mobile
A09-Y15	13 Element Yagi	RPN	15.1 dBi	Fixed / Mobile
A09-Y15	15 Element Yagi	RPN	15.1 dBi	Fixed / Mobile
A09-Y6TM	2 Element Yagi	RPTNC	6.1 dBi	Fixed / Mobile
A09-Y7TM	3 Element Yagi	RPTNC	7.1 dBi	Fixed / Mobile
A09-Y8TM	4 Element Yagi	RPTNC	8.1 dBi	Fixed / Mobile
A09-Y9TM	4 Element Yagi	RPTNC	9.1 dBi	Fixed / Mobile
A09-Y10TM	5 Element Yagi	RPTNC	10.1 dBi	Fixed / Mobile
A09-Y11TM	6 Element Yagi	RPTNC	11.1 dBi	Fixed / Mobile
A09-Y12TM	7 Element Yagi	RPTNC	12.1 dBi	Fixed / Mobile
A09-Y13TM	9 Element Yagi	RPTNC	13.1 dBi	Fixed / Mobile
A09-Y14TM	10 Element Yagi	RPTNC	14.1 dBi	Fixed / Mobile
A09-Y14TM	12 Element Yagi	RPTNC	14.1 dBi	Fixed / Mobile
A09-Y15TM	13 Element Yagi	RPTNC	15.1 dBi	Fixed / Mobile
A09-Y15TM	15 Element Yagi	RPTNC	15.1 dBi	Fixed / Mobile