



XBee[®] /XBee-PRO[®]

868 RF Module

User Guide

Revision history—90002010

Revision	Date	Description
A - E	N/A	Various editorial updates and technical content updates to keep current with product changes
F	May 2015	Updated the template and warranty information Various editorial updates Updated the capacitor requirements for the power supply in the Overview topic

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.

© 2015 Digi International. All rights reserved.

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.

Warranty

To view product warranties online, visit www.digi.com/howtobuy/terms.

Send comments

Documentation feedback: To provide feedback on this document, send your comments to techcomm@digi.com.

Customer support

Digi Technical Support: Digi offers multiple technical support plans and service packages to help our customers get the most out of their Digi product. For information on Technical Support plans and pricing, contact us at 877.912.3444 or visit www.digi.com/support.

Support portal login: www.digi.com/support.

Contents

XBee® 868 RF Modules

Key features	6
Regulatory compliance	7
Specifications	7
Mechanical drawings	8
Mounting considerations	9
Hardware diagram	10
Pin signals	10
Design notes	11
Power supply design	11
Recommended pin connections	11
Board layout	12
Antenna performance	12
Electrical characteristics	12

RF Module operation

Overview	14
Serial communications	14
UART data flow	14
Serial buffer	16
Serial flow control	16
Serial interface protocols	17
Idle mode	19
Transmit mode	19
Receive mode	19
Command mode	19
Sleep modes	21
Duty cycle	21

Advanced application features

Remote configuration commands	22
Sending a remote command	22
Applying changes on remote devices	22
Remote command responses	22
Network commissioning and diagnostics	22
Device configuration	23
Device placement	23
Node discovery	24
Commissioning pushbutton and associate LED	24
Commissioning pushbutton	24
I/O line monitoring	27
I/O samples	27
Queried sampling	27
Periodic I/O sampling	29
Digital I/O change detection	29

Sleep mode

Sleep modes	30
Normal mode (SM=0)	30
Asynchronous pin sleep mode (SM=1)	30
Asynchronous cyclic sleep mode (SM=4)	30
Asynchronous cyclic sleep with pin wake up mode (SM=5)	30
Asynchronous sleep operation	30
Wake timer	30
Indirect messaging and polling	31
Indirect messaging	31
Polling	31

Command reference tables

Special	32
Addressing	33
Serial Interfacing (I/O)	33
I/O commands	36
Diagnostics	41
AT command options	43
Node identification	44
Security	46
MAC Level	47

API operation

API frame specifications	48
API UART exchanges	50
AT commands	50
Transmitting and receiving RF data	50
Remote AT commands	51
Supporting the API	51
Frame data	52
AT command	52
AT command — queue parameter value	53
Transmit request	54
Explicit addressing command frame	56
Remote AT command request	58
AT command response	59
Modem status	60
Transmit status	61
Route information packet	62
Aggregate addressing update	64
Receive packet	65
Explicit Rx indicator	66
Data sample Rx indicator	68
Node identification indicator	70
Remote command response	72

Definitions

Definitions	74
-------------------	----

Agency certifications

CE labeling requirements	75
Declaration of Conformity	75
Important note	76
Restrictions	76

Antennas: 868 MHz

Antenna options: 868 MHz antenna listings	77
---	----

XBee[®] 868 RF Modules

The XBee and XBee-PRO 868 RF Modules were 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 SRD 868 MHz frequency band.

Key features

High performance, low cost	Low power
Indoor/urban: up to 1800 ft (550 m)	TX current: 85 - 500 mA, depending on power setting
Outdoor line-of-sight: up to 25 miles (40 km)	RX current: 65 mA (@3.3 V)
Transmit power output: 1 mW to 315 mW (0dBm to +25dBm)	
Receiver sensitivity: -112 dBm	
RF data rate: 24 kb/s	

Advanced networking and security	Easy-to-use
Retries and acknowledgments	No configuration necessary for out-of-box RF communications
Point-to-point, point-to-multi-point, and peer-to-peer topologies supported	AT and API command modes for configuring module parameters
AES 128-bit encryption	Small form factor
16-bit network ID	Extensive command set
	Free XCTU software (testing and configuration software)

Regulatory compliance

- SRD (short range device) **868 MHz frequency band**
- Manufactured under **ISO 9001:2000** registered standards. See [Agency certifications](#) for region-specific certification requirements.

Specifications

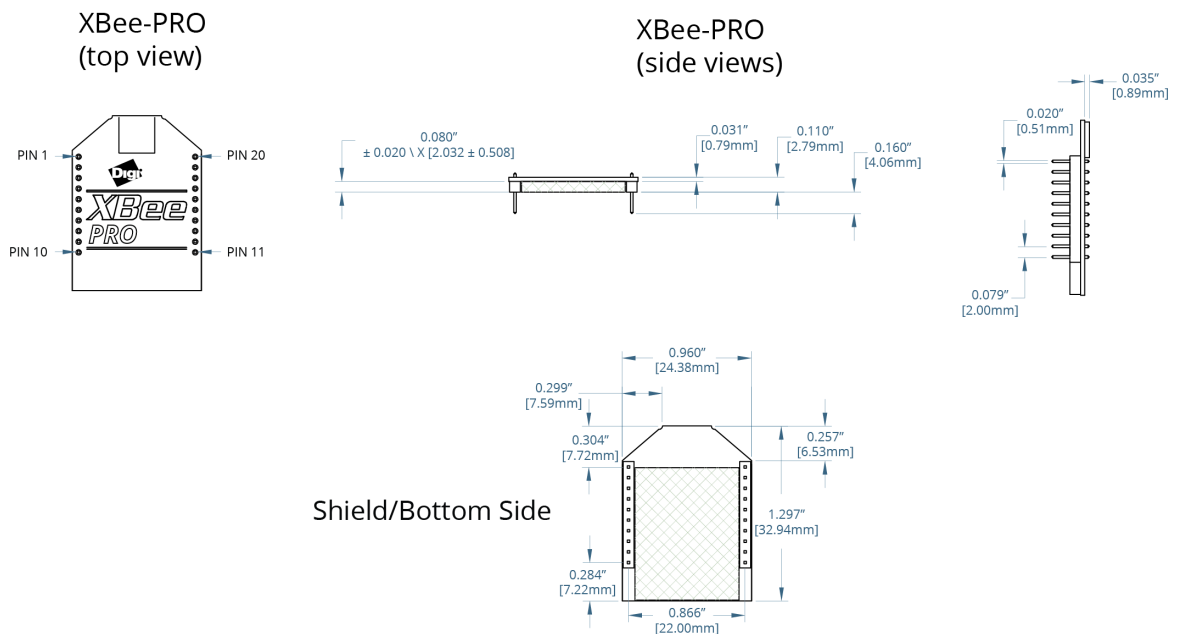
The following table contains specifications for the XBee-PRO 868 RF Module.

Specification	XBee-PRO
Performance	
Indoor/urban range	Up to 1800 ft (550 m)
Outdoor RF line-of-sight range	Up to 25 miles (40 km) with 2.0 dB dipole antenna Up to 50 miles (80 km) with high gain antenna
Transmit power output	85-500 mW, depending on power level setting
RF data rate	24 kb/s
Data throughput	2.4 kb/s
Duty cycle	10%
Receiver sensitivity	-112 dBm
Serial interface	
UART	3.3 V CMOS (5 V tolerant)
Data rate (software selectable)	1200 - 230400 bps (non-standard baud rates also supported)
Power requirements	
Supply voltage	3.0 to 3.6 VDC
Operating current	500 mA typical, (800 mA max)
Operating current (receive)	65 mA typical
Sleep current	55 µA typical @3.3 V
General	
Operating frequency band	SRD g3 band (869.525 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
Networking and security	

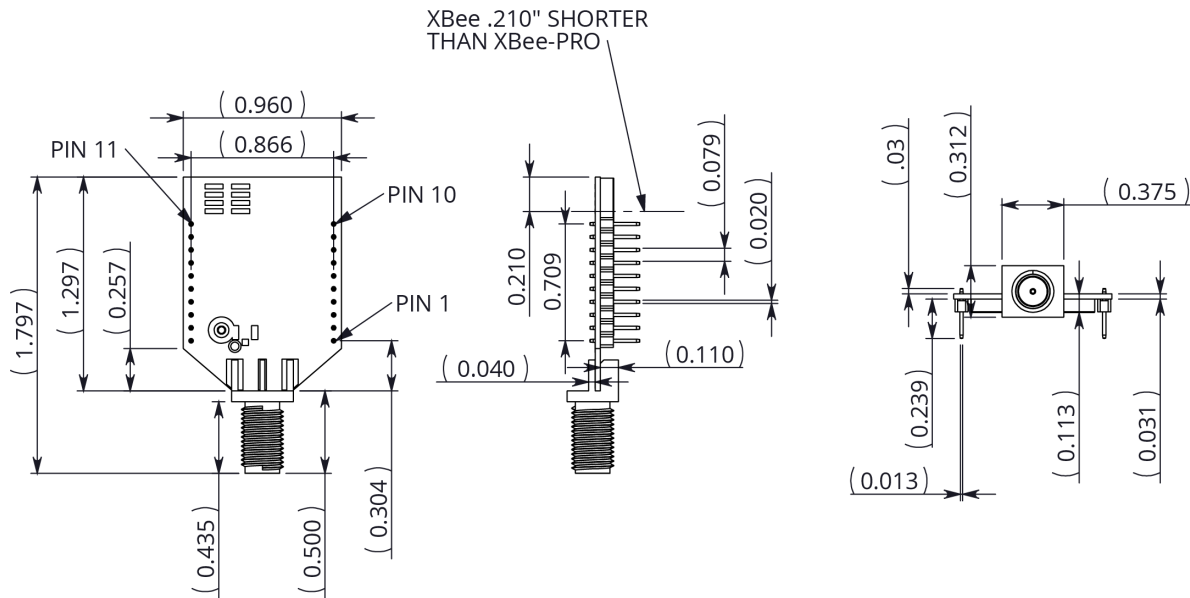
Specification	XBee-PRO
Supported network topologies	Point-to-point, point-to-multipoint, peer-to-peer
Number of channels	Single channel
Addressing options	Network ID, 64-bit addresses
Encryption	128 bit AES
Agency Approvals	
Europe (CE)	Yes Italy 25 mW Max (+14 dBm) Slovak Republic 10 mW max (+10 dBm)
RoHS	Lead-free and RoHS compliant

Mechanical drawings

The following figure shows mechanical drawings of the XBee-PRO RF Modules (antenna options are not shown).



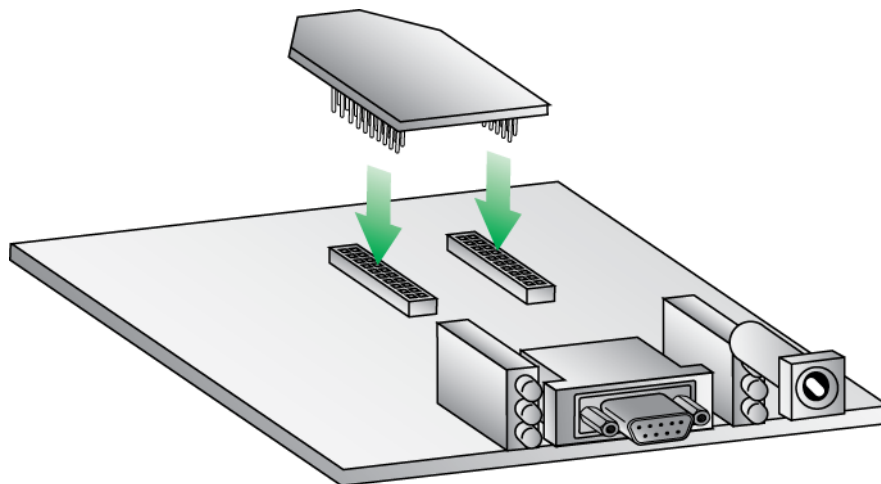
The following figure shows mechanical drawings for the RPSMA variant of the XBee-PRO RF Module.



Mounting considerations

The XBee/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.

The following figure shows how the XBee/XBee-PRO RF Module mounts to an RS-232 interface board.

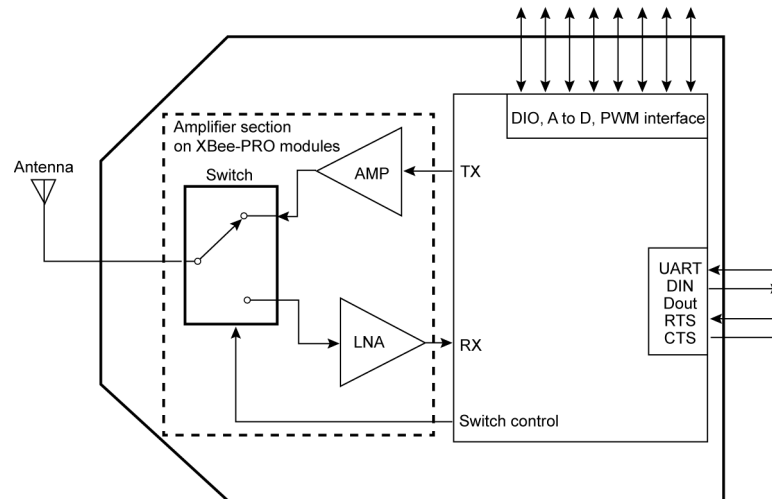


The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

- 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

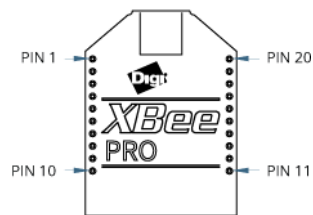
Note Digi also recommends printing an outline of the module on the board to indicate the orientation the module should be mounted.

Hardware diagram



Pin signals

XBee-PRO (top view)



The following table contains pin assignments for the XBee/XBee-PRO RF Modules. (Low-asserted signals have 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 (The 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

Pin #	Name	Direction	Description
7	PWM / DIO11	Either	PWM Output 1 / digital I/O 11
8	[reserved]	-	Do not connect
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 I/O 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, we recommend placing a 1.0 μF and 68 pF capacitor as near to pin 1 on the PCB as possible. If using a switching regulator for your power supply, switching frequencies above 500 kHz 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 2 mA; pins 9, 6, and 13 can source/sink a maximum of 16 mA; 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

The following table contains DC characteristics for the XBee-PRO (VCC = 3.0 - 3.6 VDC).

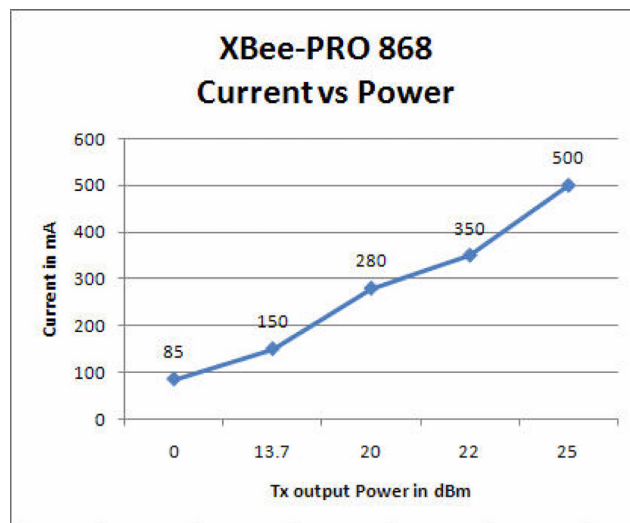
Symbol	Parameter	Condition	Min	Typical	Max	Units
V_{IL}	Input Low Voltage	All Digital Inputs	-	-	$0.2 * VCC$	V
V_{IH}	Input High Voltage	All Digital Inputs	$0.8 * VCC$	-	-	V
V_{OL}	Output Low Voltage	$I_{OL} = 2 \text{ mA}$, $VCC \geq 3.0 \text{ V}$	-	-	$0.18 * VCC$	V
V_{OH}	Output High Voltage	$I_{OH} = 2 \text{ mA}$, $VCC \geq 3.0 \text{ V}$	$0.82 * VCC$	-	-	V
I_{IN}	Input Leakage Current	$V_{IN} = VCC$ or GND, all inputs, per pin	-	-	0.5	μA
I_{out}	Output Current	D_{out} , DIO(0, 1, 2, 3, 6, 7, 8), On/Sleep	-	-	8	mA
I_{out}	Output Current	DIO9, DIO10, DTR	-	-	16	mA

Symbol	Parameter	Condition	Min	Typical	Max	Units
I_{out}	Output Current	DIO4, DIO5	-	-	2	mA
I-TX	Max TX current draw over Voltage and Temp	VCC=3.3V Power=+25dBm	-	500	800	mA

Note The sum of all the DIO current draw should not exceed 120 mA.

The following table contains PL commands for electrical characteristics.

ATPL	TX dBm	TX power (mW)	Typical Current (mA) @3.3V
0	0	1	85
1	13.7	23	150
2	20	100	280
3	22	158	350
4	25	316	500



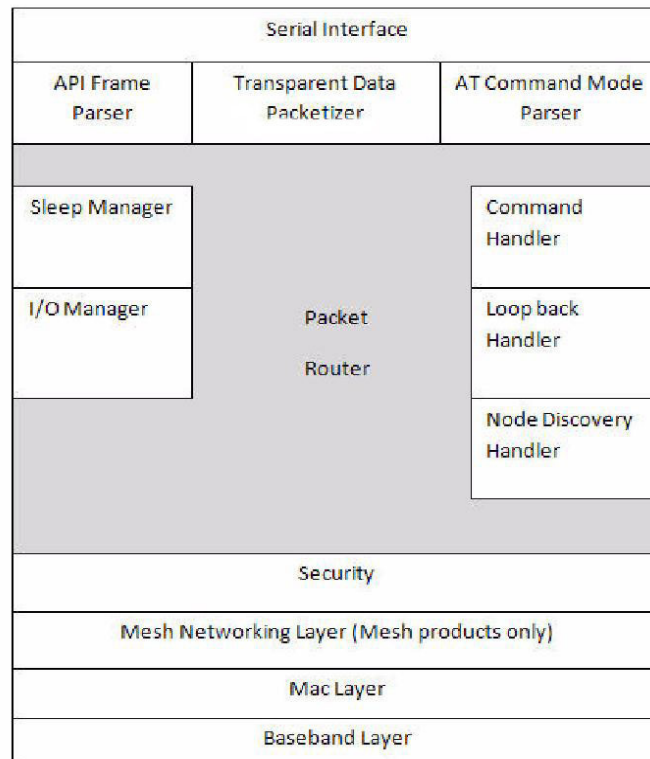
Note These modules are designed to maximize the range of the radio. When testing modules on the same desk or in close proximity, use power level 0 and keep them at least 3 feet (1 meter) apart in order to avoid saturating the receiver. When using higher power levels, keep the modules at least 24 feet (7 meters) apart.

RF Module operation

Overview

The XBee RF Module provides a serial interface to an RF link. The XBee RF 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 RF Module.



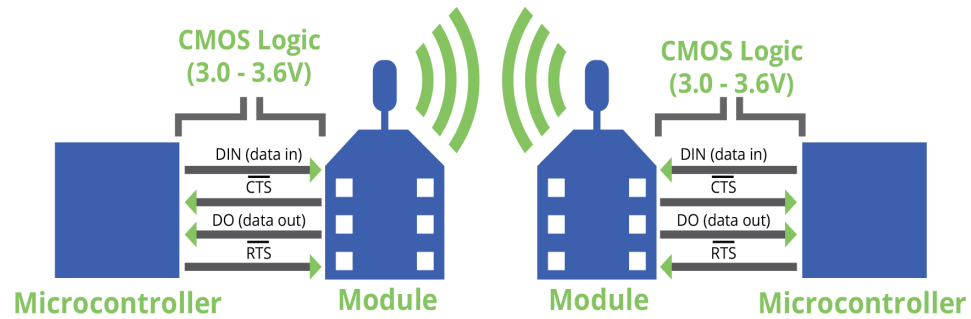
Serial communications

The XBee-PRO RF Module connects 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, such as 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.

The following figure shows a system data flow diagram in a UART-interfaced environment. (Low-asserted signals distinguished have a horizontal line over the signal name.)

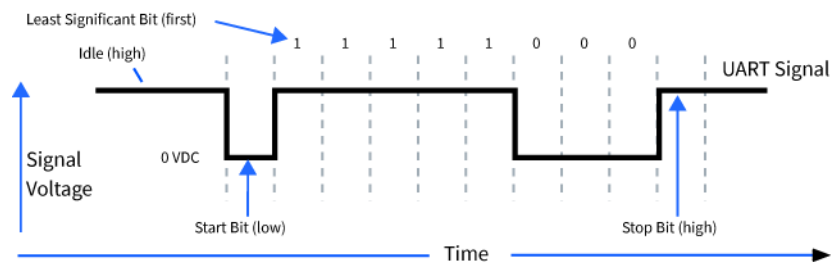


Serial data

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 following figure shows a UART data packet 0x1F (decimal number “31”) transmitting through the RF module. The example data format is 8-N-1 (bits - parity - # of stop bits).

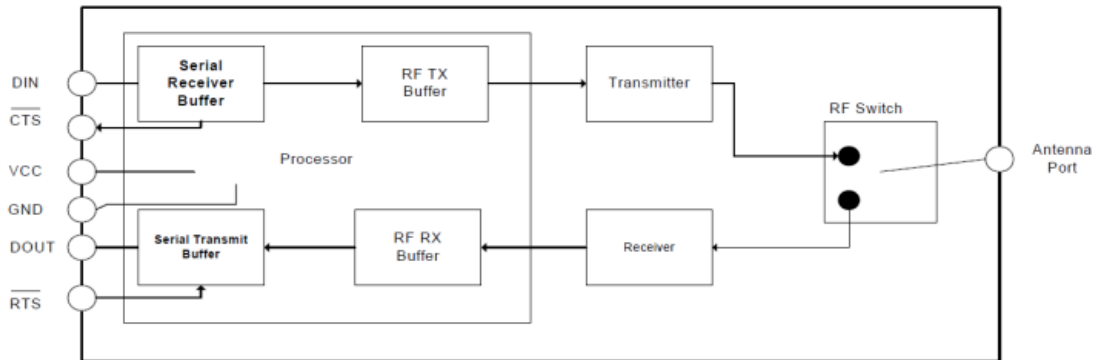


The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

Serial buffer

The XBee-PRO RF Module maintains buffers to collect received serial and RF data. 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.

The following figure shows the 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:

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.

Serial transmit buffer

When RF data is received, the data is moved into the serial transmit buffer and is sent out the serial port. 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:

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

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. (See command description for the FT command.)

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 API (application programming interface) 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's UART 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 on

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

A Comparison of 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 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.

As a general rule of thumb, API firmware is recommended 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.

Note To implement API operations, refer to [API operation](#).

Idle mode

When not receiving or transmitting data, the RF module is in idle mode. During idle mode, the RF module is 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.

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 [API operation](#) for an alternate means of configuring modules.

AT command mode

To enter AT command mode:

Send the three-character command sequence “+++” and observe guard times before and after the command characters. The following list provides additional information about AT command mode.

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.

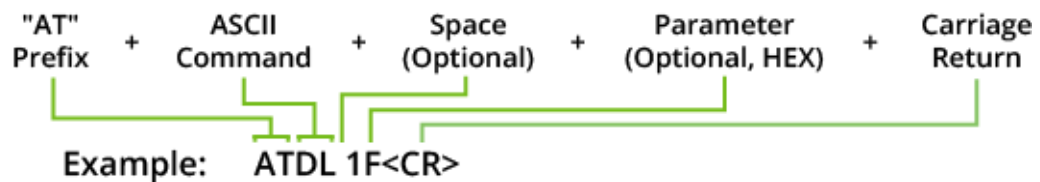
When command mode has been entered, the command mode timer is started (CT command), and the module is able to receive AT commands on the DIN pin.

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. When using XCTU, ensure that the ‘Baud’ setting on the “PC Settings” tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (9600 bps).

To send AT commands:

Send AT commands and parameters using the syntax shown in the following figure.



Note 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:

- AC (apply changes) command is issued
- AT command mode is exited

To exit AT command mode:

- Send the ATCN (exit command mode) command (followed by a carriage return).
[OR]
- 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.

Note For an example of programming the RF module using AT commands and descriptions of each configurable parameter, refer to [Command reference tables](#).

Sleep modes

Sleep modes allow the RF Module to enter states of low power consumption when not in use. The XBee RF Module supports both pin sleep (sleep mode entered on pin transition) and cyclic sleep (module sleeps for a fixed time). For more information about XBee sleep modes, see [Sleep mode](#).

Duty cycle

The duty cycle of this radio is 10% averaged over the period of one hour. Meaning, if the next transmission will push the running average duty cycle over the 10% limit, the module will not transmit until enough time has elapsed to stay under the duty cycle.

Because of heat restraints in the module, a 10% duty cycle over the period of 1 second will be enforced after the measured temperature of the module rises above 60 degrees C.

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 (for more information, see [API operation](#)). 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). The API field for a 16-bit address should be set to 0xFFFE. 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 devices

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 mesh 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.

Device placement

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.

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 loop-back 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 loop-back cluster ID depend on the following AP settings:

AT configuration (AP=0)

To send data to the loop-back 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 (0 for the coordinator, or 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 ZigBee Command 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.

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. However, this number can be misleading. 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. XBee-PRO 900 modules also have the limitation that only RSSI values near the sensitivity level of the radio are accurately reported. Signals received which are stronger than approximately -85 dBm will be reported as approximately -85 dBm.

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. XBee-PRO 900 modules also have the limitation that only RSSI values near the sensitivity level of the radio are accurately reported. Signals received which are stronger than approximately -85 dBm will be reported as approximately -85 dBm.

Node discovery

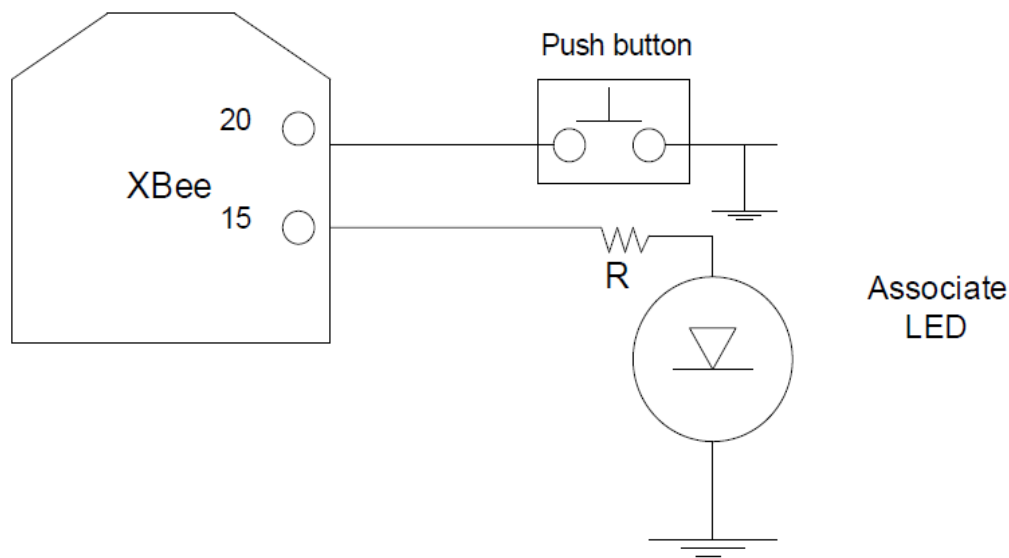
The node discover (ND) command can be used to discover all Digi modules that have joined a network. Issuing the ND command sends a broadcast node discovery command throughout the network. All devices that receive the command will send a response that includes the device's addressing information, node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

When a device receives the node 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).

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.

The following figure shows 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 associated LED functionalities.

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).

Button presses	Sleep configuration and sync status	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 sleep	No effect.
2	Configured for synchronous sleep	Causes a node which is configured with sleeping router nomination enabled (see description of the ATSO – sleep options command in the XBee module's Product Manual) to immediately nominate itself as the network sleep coordinator.
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. (For example, 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).

Having the commissioning button enabled during sleep will increase the sleeping current draw (especially in SM1 mode). It is necessary to hold down the commissioning button for up to 2 seconds to wake the module from sleep.

Associate LED

The Associate pin (pin 15) can provide indication of the device's sleep status and diagnostic 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 (500ms for sleep coordinator, 250ms otherwise).

Sleep mode	Sleep status	LED status	Meaning
0		On, blinking green	The device is powered and operating properly.
1, 4, 5		Off	The device is in a low power mode.
1, 4, 5		On, blinking green	The device is powered, awake and is operating properly.
7		On, solid green	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 green (500 ms blink time)	The device is acting as the network sleep coordinator and is operating properly.
7, 8		On, fast blinking green (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 green	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 1 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

The following table outlines typical parameters for the pin configuration commands.

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

The following table shows the commands you use when configuring a 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

For more information, see [Command reference tables](#). Pull-up 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	<p>Indicates which digital IO lines have sampling enabled. Each bit corresponds to one digital IO line on the module.</p> <ul style="list-style-type: none"> • 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 <p>For example, a digital channel mask of 0x002F means DIO 0, 1, 2, 3, and 5 are enabled as digital IO.</p>
1	Analog channel mask	<p>Indicates which lines have analog inputs enabled for sampling. Each bit in the analog channel mask corresponds to one analog input channel.</p> <ul style="list-style-type: none"> • 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	<p>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.</p> <p>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.</p>

If the IS command is issued from AT command mode, then a carriage return delimited list will be returned containing the fields listed in the previous table. 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]

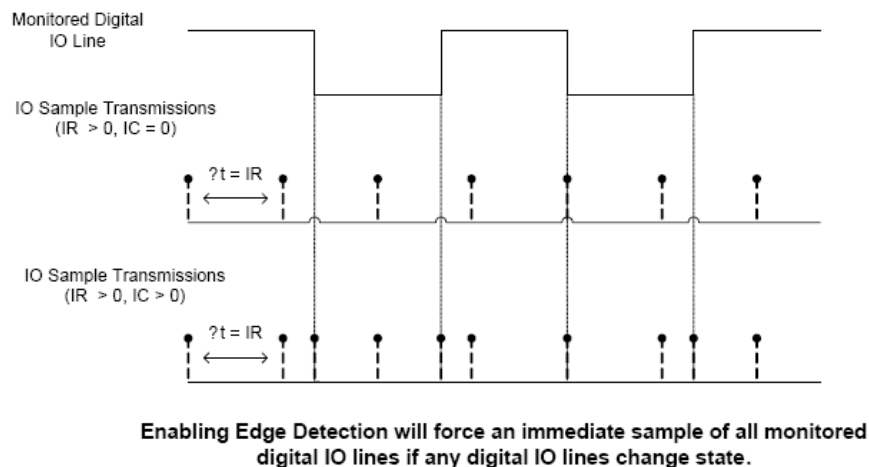
Periodic I/O sampling

Periodic sampling allows an XBee-PRO 900 module to take an IO 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 IO samples at the IR rate until the ST time expires and the device can resume sleeping. For more information, see [Sleep mode](#).

Digital I/O change detection

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



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.

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.

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). Upon waking up, a pin sleep module will begin polling its parent as described in the [Indirect messaging and polling](#).

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 its parent for any buffered data messages before returning to sleep again. Cyclic sleep mode is enabled by setting the SM command to 4. In cyclic sleep, the module sleeps for a specified time, and then wakes and sends a poll request to its parent to discover if the parent has any pending data for the end device.

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 a slight variation on (SM=4) that allows the module to be woken prematurely by asserting the **Sleep_RQ** pin (pin 9). In (SM=5), the XBee can wake after the sleep period expires, or if a high-to-low transition occurs on the **Sleep_RQ** pin.

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. While the module is awake, it will send poll request transmissions every 100 ms to check its parent for buffered data. The module returns to sleep when the sleep timer expires.

Indirect messaging and polling

To enable reliable communication with sleeping devices, the messaging mode command (CE) can be used to enable indirect messaging and polling.

Indirect messaging

Indirect messaging is a communication mode designed for communicating with asynchronous sleeping devices. A module can enable indirect messaging by making itself an indirect messaging coordinator with the CE command. An indirect messaging coordinator does not immediately transmit a unicast when it is received over the serial port. Instead, the module holds on to the data until it is requested via a poll. On receiving a poll, the indirect messaging coordinator will send out any data packets it may have queued up.

Because it is possible for a polling device to be eliminated, a mechanism is in place to purge unrequested data packets. If the coordinator holds an indirect data packet for an end device longer than 2.5 times its SP value, the packet is then purged. Users are encouraged to set the SP of the coordinator to the same value as the highest SP time that exists among the end devices in the network. If the coordinator is in API mode, a TxStatus message is generated for a purged data packet with a status of 0x75 (INDIRECT_MESSAGE_UNREQUESTED).

An indirect messaging coordinator will queue up as many data packets as it has buffers available. After the coordinator has used up all of its available buffers, it will hold all transmission requests unprocessed on the serial input queue. After the serial input queue is full, CTS will be de-asserted (if hardware flow control is enabled). After receiving a poll or purging data from the indirect messaging queue, the buffers become available again.

Indirect messaging has no effect on broadcasts. Broadcasts are sent immediately when received over the serial port and are not put on the indirect messaging queue.

Polling

Polling is the automatic process by which a node can request data from an indirect messaging coordinator. Polling can be enabled on a device by configuring it as an end device with the CE command. When polling is enabled, the module will send a poll request at least once every 100 ms. When normal data is sent to the destination specified by the DH/DL of an end device module, that data will also function as a poll.

When a polling device is also an asynchronous sleeping device, it will send a poll shortly after waking from sleep. After that first poll is sent, the module will send polls in the normal manner described above until it has returned to sleep.

Command reference tables

Special

The following table contains special AT commands.

AT command	Name and description	Parameter range	Default
WR	Write. Writes 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.	--	--
RE	Restore defaults. Restores module parameters to factory defaults.	--	--
FR	Software reset. Resets module. Responds immediately with an “OK” then performs a reset 100 ms later.	--	--
AC	Apply changes. Immediately applies new settings without exiting command mode.	--	--
R1	Restore compiled. Restores module parameters to compiled defaults.	--	--
VL	Version long. Shows detailed version information including application build date and time.	--	--

Addressing

The following table contains addressing commands.

AT command	Name and description	Parameter range	Default
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
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
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
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
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
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
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
CE	Coordinator/End Device. Set/read the messaging mode of the module as follows: 0 — Normal mode 1 — Indirect messaging coordinator 2 — Polling device	0 - 2	0

Serial Interfacing (I/O)

The following table contains serial interfacing commands.

AT command	Name and description	Parameter range	Default
AP	<p>API mode. Set or read the API mode of the radio. The following settings are allowed:</p> <p>0 — API mode is off. All UART input and output is raw data and packets are delineated using the RO and RB parameters.</p> <p>1 — API mode is on. All UART input and output data is packetized in the API format, without escape sequences.</p> <p>2 — API mode is on with escaped sequences inserted to allow for control characters (XON, XOFF, escape, and the 0x7e delimiter to be passed as data).</p>	0, 1, or 2	0
AO	<p>API output format. Enables different API output frames. Options include:</p> <p>0 — Standard Data Frames (0x90 for RF RX)</p> <p>1 — Explicit Addressing Data Frames (0x91 for RF RX)</p>	0, 1	0
BD	<p>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 0x39 and above select the actual baud rate. Providing the host supports it. Baud rates can go as high as 1.875Mbps. The values from 0 to 8 are interpreted as follows:</p> <p>0 — 1,200bps 3 — 9,600bps 6 — 57,600bps 1 — 2,400bps 4 — 19,200bps 7 — 115,200bps 2 — 4,800bps 5 — 38,400bps 8 — 230,400bps</p>	0 - 8 0x39 - 0x1c9c38	0x03 (9600 bps)
RO	<p>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.</p>	0 - 0xFF [x character times]	3
FT	<p>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.</p>	17 - 319	0x13F =319
NB	<p>Parity. Set or read parity settings for UART communications. The values from 0 to 4 are interpreted as follows:</p> <p>0 — No parity 3 — Forced high parity 1 — Even parity 4 — Forced low parity 2 — Odd parity</p>	0 - 4	0 (No parity)
SB	<p>Stop bits. Set or read number of stop bits used for UART communications.</p> <p>0 — 1 stop bit 1 — 2 stop bits</p>	0 - 1	0 (1 stop bit)

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	0
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 commands

The following table contains input and output (I/O) commands. (Low-asserted signals have a horizontal line above signal name.)

AT command	Name and description	Parameter range	Default
P0	DIO10/PWM0 configuration. Configure options for the DIO10/PWM0 line of the module. Options include: 0 — Input, unmonitored 1 — RSSI 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
RP	RSSI PWM timer. Time RSSI signal will be output after last transmission. When RP = 0xFF, output will always be on.	0 - 0xFF [x 100 ms]	2032 3.2 seconds
1S	XBee sensor sample. Forces a sample to be taken on an XBee Sensor device. This command can only be issues to an XBee Sensor device using an API remote command.		
D0	AD0/DIO0 configuration. Configure options for the AD0/DIO0 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

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

AT command	Name and description	Parameter range	Default
D8	DIO8 configuration. Configure options for the DIO8 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
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	

AT command	Name and description					Parameter range	Default
PR	Pull-up resistor. 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.					0 - 7FFF	0x3F7F
	0	DIO4/AD4	11	55K - 330K	1		
	1	AD3 / DIO3	17	5K - 23K	2		
	2	AD2 / DIO2	18	5K - 23K	2		
	3	AD1 / DIO1	19	5K - 23K	2		
	4	AD0 / DIO0	20	5K - 23K	2		
	5	RTS / DIO6	16	5K - 23K	2		
	6	DTR / SLEEP_RQ/DIO8	9	5K - 23K	2		
	7	DIN / Config	3	5K - 23K	2		
	8	Associate / DIO5	15	55K - 330K	1		
	9	On/Sleep / DIO9	13	5K - 23K	2		
	10	DIO12	4	5K - 23K	2		
	11	PWM0 / RSSI / DIO10	6	5K - 23K	2		
	12	PWM1 / DIO11	7	5K - 23K	2		
	13	DIO7/CTS	12	5K - 23K	2		
	14	DOUT	2	5K - 23K	2		
	Note 1: When set as a digital input with pull-up disabled, the leakage can be 9 uA in the worst case and 90nA in typical case when the line is set externally at a low level. Note 2: When set as a digital input with pull-up enabled, the voltage of line will stabilize between $V_{CC}-0.65V$ and $V_{CC}-0.45V$.						
M0	PWM0 output level. Set/read the output level of the PWM0 line. The line should be configured as a PWM output using the P0 command.					0 - 0x03FF	0
M1	PWM1 output level. Set/read the output level of the PWM1 line. The line should be configured as a PWM output using the P1 command.					0 - 0x03FF	0

AT command	Name and description	Parameter range	Default
LT	Assoc LED blink time. 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, 250ms otherwise). For all other LT values, LT is measured in 10 ms.	0x14 - 0xFF (x10ms)	0
IS	Force sample. Forces a read of all enabled digital and analog input lines.	n/a	n/a
IC	<p>IO digital change detection. Set/Read the digital IO pins to monitor for changes in the IO 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 IO 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.</p> <p>Bit (IO pin):</p> <ul style="list-style-type: none"> • 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
IR	IO sample rate. Set/Read the IO 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 IO functionality enabled (see D0-D9, P0-P2 commands). The sample rate is measured in milliseconds.	0 - 0xFFFF (ms)	0
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

Diagnostics

The following table contains diagnostics commands.

AT command	Name and description	Parameter range	Default
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
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
ER	RF errors. Read the number of times a packet was received which contained integrity errors of some sort. When the value reaches 0xFFFF, it stays there.	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
RP	RSSI PWM timer. Set or read the time that the RSSI output (indicating signal strength) will remain active after the last reception. Time units are measured in tenths of seconds.	1 - 0xff	0x20 = 3.2 seconds
TR	Transmission errors. Read the number of MAC frames that exhaust MAC retries without ever receiving a MAC acknowledgment message from the adjacent node. When the value reaches 0xffff, it stays there.	n/a	n/a
TP	Temperature. Read module temperature in Celsius. Negatives temperatures can be returned.	0xff74 - 0x0258	n/a
DB	<p>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 -96 dBm.</p> <p>An XBee-PRO900 module will only report RSSI values within approximately 15 dBm of the sensitivity level of the module. Signals which exceed approximately -85 dBm will be reported as approximately -85 dBm.</p>	n/a	n/a

AT command	Name and description	Parameter range	Default
DC	Duty cycle. Returns a current usage percentage of the 10% duty cycle. This is measured over the period of 1 hour. For example, if the radio had averaged 2% duty cycle, then this would return 20%.		
RC	RSSI for channel. Reads the dBm level of the designated channel.	0	n/a
R#	Reset number. Tells the reason for the last module reset. 0 = Power up reset 2 = Watchdog reset 3 = Software reset 4 = Reset line reset 5 = Brownout reset	n/a	0
TA	Transmit acknowledgment errors. Incremented once for each failed ack retry.	0 - 0xFFFF	0
%V	Supply voltage. Reads the voltage on the V _{CC} pin in mV. Read module voltage in millivolts.	0 - 0x/F00	n/a

AT command options

The following table contains AT command options.

AT command	Name and description	Parameter range	Default
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)
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

Node identification

The following table contains node identification 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
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)
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
DN	<p>Discover node — destination node. Resolves an NI (Node Identifier) string to a physical address (case sensitive).</p> <p>The following events occur after the destination node is discovered:</p> <p><AT Firmware></p> <ol style="list-style-type: none"> 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. <p><API Firmware></p> <p>0xFFFE and 64-bit extended addresses are returned in an API command Response frame.</p> <p>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.</p>	20 byte ASCII string	

AT command	Name and description	Parameter range	Default
ND	<p>Node discover — node discover. Discovers and reports all RF modules found. The following information is reported for each module discovered.</p> <p>MY<CR> SH<CR> SL<CR> NI<CR> (Variable length) PARENT_NETWORK ADDRESS (2 Bytes)<CR> 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) <CR></p> <p>After (NT * 100) milliseconds, the command ends by returning a <CR>. ND also accepts a node identifier (NI) as a parameter (optional). In this case, only a module that matches the supplied identifier will respond. If ND is sent through the API, each response is returned as a separate AT_CMD_Response packet. The data consists of the above listed bytes without the carriage return delimiters. The NI string will end in a “0x00” null character.</p>		
NO	<p>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.</p> <p>Options include:</p> <p>0x01 = Append DD value (to ND responses or API node identification frames)</p> <p>0x02 = Local device sends ND response frame when ND is issued.</p>	0 - 0x03 [bitfield]	0

Security

The following table contains 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

MAC Level

The following table contains MAC-level commands.

AT command	Name and description	Parameter range	Default
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
PL	Power level. Set/view transmitter RF power. For more information, see Electrical characteristics .	0 - 4	4

API operation

As an alternative to transparent operation, API (application programming interface) 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.

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

API frame specifications

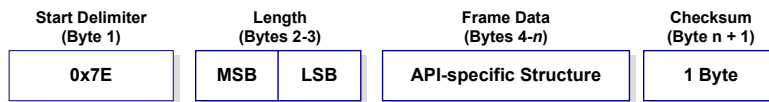
Two API modes are supported and both can be enabled using the AP (API enable) command. Use the following AP parameter values to configure the module to operate in a particular mode:

- 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:

UART data frame structure:



Note 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 reply with a module status frame indicating the nature of the failure.

API operation — escape characters (AP parameter = 2)

When this API mode is enabled (AP = 2), the UART data frame structure is defined (see the following figure).

UART data frame structure — with escape control characters:



Note 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:
 $0xFF - (0x23 + 0x11) = (0xFF - 0x34) = 0xCB$.

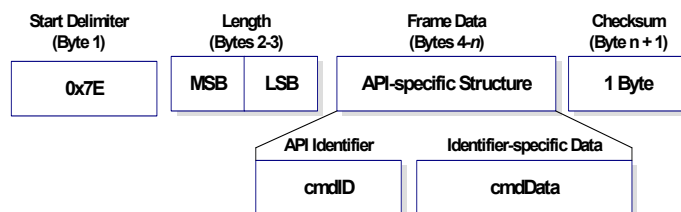
Length

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, shown in the following figure.

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:

API frame names	API ID
AT command	0x08
AT command — queue parameter value	0x09
Transmit request	0x10
Explicit addressing command frame	0x11

API frame names	API ID
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.

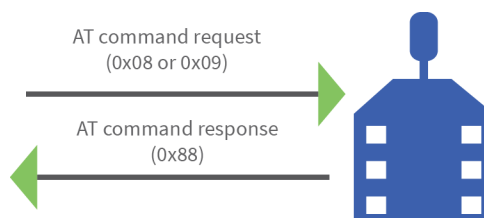
To calculate: Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract the result from 0xFF.

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 UART 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.



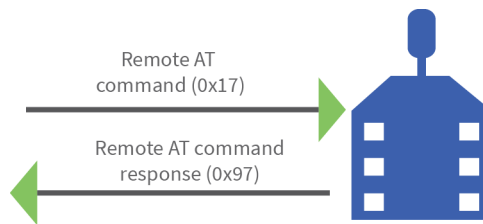
Transmitting and receiving RF data

The following image shows the API exchanges that take place at the UART 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 transmit 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 UART when sending a remote AT command. A remote command response frame is not sent out the UART 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	Number of bytes between the length and the checksum.
Length		MSB 1	0x00	
		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 above example 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 AC (apply changes) 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-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).

The antennas in the tables below have been approved for use with this module. Digi does not carry all of these antenna variants. Contact Digi Sales for available antennas.

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

Frame fields		Offset	Example	Description
	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 40014011, 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 (acknowledgement). 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	

Frame fields		Offset	Example	Description
	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: <ul style="list-style-type: none"> • bit 0: Disable ACK • bit 1: Don't attempt route Discovery • All other bits must be set to 0
	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-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 this is not set, then the 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)	
	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 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-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	Number of bytes between the length and the checksum.
Length		MSB 1	0x00	
		LSB 2	0x07	
Frame-specific Data	Frame type	3	0x8B	
	Frame ID	4	0x47	Identifies the UART data frame being reported. Note: If Frame ID = 0 in AT command mode, no AT command response will be given.
	Reserved	5	0xFF	Reserved
		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
	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 using a frame ID of 0x47.

Route information packet

Frame type: 0x8D

A route information packet can be output for unicast transmissions on which the NACK enable or the trace route enable TX option is enabled.

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	Number of bytes between the length and the checksum
Length		MSB 1	0x00	
		LSB 2	0x2A	
Frame-specific data	Frame type	3	0x8D	
	Source event	4	0x12	0x11 = NACK, 0x12 = trace route
	Length	5	0x2B	Number of bytes that follow (excluding checksum). If length increases, the new items have been added to the end of the list (for future revisions).
	Timestamp	MSB 6	0x9C	System timer value on the node generating the route information packet.
		7	0x93	
		8	0x81	
		LSB 9	0x7F	
	ACK timeout count	10	0x00	The number of MAC ACK timeouts that occurred.
	Reserved	11	0x00	Reserved
	Reserved	12	0x00	Reserved
	Destination address	MSB 13	0x00	Address of the final destination node of this network level transmission.
		14	0x13	
		15	0xA2	
		16	0x00	
		17	0x40	
		18	0x52	
		19	0xAA	
		LSB 20	0xAA	

Frame fields		Offset	Example	Description
	Source address	MSB 21	0x00	Address of the source node of this network level transmission.
		22	0x13	
		23	0xA2	
		24	0x00	
		25	0x40	
		26	0x52	
		27	0xDD	
		LSB 28	0xDD	
	Responder address	MSB 29	0x00	Address of the node that generated this route information packet after sending (or attempting to send) the packet to the next hop (the receiver node).
		30	0x13	
		31	0xA2	
		32	0x00	
		33	0x40	
		34	0x52	
		35	0xBB	
		LSB 36	0xBB	
	Receiver address	MSB37	0x00	Address of the node to which the data packet was just sent (or attempted to be sent to).
		38	0x13	
		39	0xA2	
		40	0x00	
		41	0x40	
		42	0x52	
		43	0xCC	
		LSB 44	0xCC	
Checksum		45	0xCE	0xFF — The 8 bit sum of bytes from offset 3 to this byte.

Example: The above example represents a possible route information frame that could be received when doing a trace route on a transmission from a radio with serial number 0x0013a2004052AAAA to a radio with serial number 0x0013a2004052DDDD. This particular frame indicates that the transmission was successfully forwarded from the radio with serial number 0x0013a2004052BBBB to the radio with serial number 0x0013a2004052CCCC.

Aggregate addressing update

Frame type: 0x8E

An Aggregate Addressing Update frame is output on an API-enabled node when an address update frame (generated by the AG command being issued on a node in the network) causes the node to update its DH and DL registers.

Frame Fields		Offset	Example	Description
Start delimiter		0	0x7E	Number of bytes between the length and the checksum.
Length		MSB 1	0x00	
		LSB 2	0x12	
Frame-specific data	Frame type	3	0x8E	
	Format ID	4	0x00	Byte reserved to indicate format of additional packet information which may be added in future firmware revisions. In the current firmware revision, 0x00 is returned in this field.
	New address	MSB 5	0x00	Address to which DH and DL are being set.
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x52	
		11	0xBB	
		LSB 12	0xBB	
	Old address	13	0x00	Address to which DH and DL were previously set.
		14	0x13	
		15	0xA2	
		16	0x00	
		17	0x40	
		18	0x52	
		19	0xAA	
		20	0xAA	
Checksum		21	0x2E	0xFF — The 8 bit sum of bytes from offset 3 to this byte.

Example: In the above example a radio which had a destination address (DH/DL) of 0x0013A2004052AAAA updated its destination address to 0x0013A2004052BBBB.

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	Number of bytes between the length and the checksum.
Length		MSB 1	0x00	
		LSB 2	0x12	
Frame-specific data	Frame type	3	0x90	64-bit address of sender
	64-bit Source address	MSB 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

Frame fields		Offset	Example	Description
	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 delimiter		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	

Frame fields		Offset	Example	Description
	Receive options	14	0x01	0x01 — Packet acknowledged 0x02 — Packet was a broadcast packet All other bits are reserved and should be ignored.
	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 17 0x1C 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).
		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.
	Digital samples (if included)	20	0x14	
	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.

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.

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 the ND command in [Node identification](#)).

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
Frame-specific data		LSB 2	0x25	
	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	0x74	
		10	0x02	
		LSB 11	0xAC	
	16-bit source network address	MSB 12	0x7D	16-bit address of sender
		LSB 13	0x84	
	Receive options	14	0x2	0x01 — Packet acknowledged 0x02 — Packet was a broadcast packet
	Source 16-bit address	15	0x7D	Set to the 16-bit network address of the remote. Set to 0xFFFE if unknown.
		16	0x84	
	64-bit 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	

Frame fields		Offset	Example	Description
	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	Indicates the 16-bit address of the remote's parent or 0xFFFE if the remote has no parent.

Example: If the commissioning push button is pressed on a remote router device with 64-bit address 0x0013a200407402ac and default NI string, the following node identification indicator would be received: 0x7e 0025 9500 13a2 0040 7402 acff fec2 fffe 0013 a200 4074 02ac 2000 fffe 0101 c105 101e 000c 0000 2e33.

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
	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 0x0013A20040522BAA to query the SL command, and if the frame ID=0x55, the response would look like the above example.

Definitions

Definitions

The following table contains terms and definitions for this guide.

Term	Definition
PAN	Personal area network — A data communication network that includes a coordinator and one or more routers/end devices. Synonymous with “network ID”
Network address	16-bit address used for compatibility and has an address of 0xFFFE

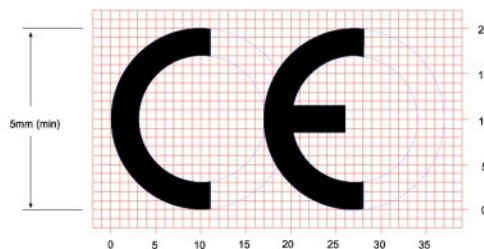
Agency certifications

The XBee-PRO RF Module has been certified for use in several European countries. For a complete list, refer to www.digi.com.

If the XBee-PRO RF Modules are incorporated into a product, the manufacturer must ensure compliance of the final product to the European harmonized EMC and low-voltage/safety standards. A Declaration of Conformity must be issued for each of these standards and kept on file as described in Annex II of the R&TTE Directive.

Furthermore, the manufacturer must maintain a copy of the XBee-PRO user manual documentation and ensure the final product does not exceed the specified power ratings, antenna specifications, and/or installation requirements as specified in the user manual. If any of these specifications are exceeded in the final product, a submission must be made to a notified body for compliance testing to all required standards.

CE labeling requirements



The CE mark shall consist of the initials "CE" taking the following form:

- If the CE marking is reduced or enlarged, the proportions given in the above graduated drawing must be respected.
- The CE marking must have a height of at least 5 mm except where this is not possible on account of the nature of the apparatus.
- The CE marking must be affixed visibly, legibly, and indelibly.

Declaration of Conformity

Digi has issued Declarations of Conformity for the XBee-PRO RF Modules concerning emissions, EMC and safety. Files are located in the 'documentation' folder of the Digi CD.

Important note

Digi does not list the entire set of standards that must be met for each country. Digi customers assume full responsibility for learning and meeting the required guidelines for each country in their distribution market. For more information relating to European compliance of an OEM product incorporating the XBee-PRO RF Module, contact Digi, or refer to the following web sites:

- **CEPT ERC 70-03E** — Technical Requirements, European restrictions and general requirements: Available at www.ero.dk.
- **R&TTE Directive** — Equipment requirements, placement on market; available at www.ero.dk.

Restrictions

Power Output: The power output of the XBee-PRO 868 RF modules must not exceed 27 dBm E.R.P. (conducted RF power). The power level is set using the PL command.

Italy: Italy imposes a 25 mW (14 dBm) maximum limit. (Use PL=1)

Slovak Republic: Slovak Republic imposes a 10 mW (10 dBm) maximum limit (Use PL=0)

Antennas: 868 MHz

Most European countries require that the transmit output power, antenna and cable included, remain less than or equal to 27 dBm. You can calculate the allowable antenna gain for any PL setting using the following equations:

- $\text{dBi} = \text{dBd} + 2$ (This helps you find the dBi gain of the antenna, since some are listed in dBd.)
- $P + G - L - 1.14 \leq 27 \text{ dBm}$ (This equation has 1 dB of headroom for variations that occur).

Where:

P = The transmit power level of the module. (0 dBm, +13.7 dBm, +20 dBm, +22 dBm, +25 dBm).

G = The gain of the antenna in dBi.

L = The cable loss between the module and the antenna.

Italy allows 14dBm radio output power plus a 2 dBi gain antenna. Slovak Republic allows 10 dBm radio output power plus a 2 dBi gain antenna.

For optimal link budget and better performance, the power level (P) should be decreased to keep under the legal limit, rather than increase the loss (L) in the cable.

Antenna options: 868 MHz antenna listings

The antennas in the tables below have been approved for use with this module. Digi does not carry all of these antenna variants. Contact Digi Sales for available antennas.

Digi Model	Type	Connector	Gain (dBi)	Gain (dBd)	Application	Max Power ATPL	XBEE-868 Max Power Level w/ no attenuation or cable loss	Attenuation or cable loss needed for +25dBm output
A08-Y6NF	Yagi	N-Female	6	4	Fixed	3	22	2.86
A08-Y9NF	Yagi	N-Female	9	7	Fixed	1	13.7	5.86
A08-Y14NF	Yagi	N-Female	14	12	Fixed	1	13.7	10.86
A08-Y11NF	Yagi	N-Female	11	9	Fixed	1	13.7	7.86

Digi Model	Type	Connector	Gain (dBi)	Gain (dBd)	Application	Max Power ATPL	XBEE-868 Max Power Level w/ no attenuation or cable loss	Attenuation or cable loss needed for +25dBm output
A08-Y13NF	Yagi	N-Female	13	11	Fixed	1	13.7	9.86
A08-Y14NF-1	Yagi	N-Female	14	12	Fixed	1	13.7	10.86
A08-F2NF	Omni	N-Female	2	0	Fixed	4	25	0
A08-F5NF	Omni	N-Female	5	3	Fixed	3	22	1.86
A08-F8NF	Omni	N-Female	8	6	Fixed	2	20	4.86
A08-P8NF	Patch	N-Female	8	6	Fixed	2	20	4.86
A08-P7NF	Patch	N-Female	7	5	Fixed	2	20	3.86
A08-P5NF	Patch	N-Female	5	3	Fixed	3	22	1.86
A08-HABUF-P51	Dipole	U.FL Female	2	0	Fixed/Mobile	4	25	0
A08-HASM-560	Dipole	RPSMA-F	2	0	Fixed/Mobile	4	25	0
Integrated	Mono-pole	Integrated	1.7	-0.3	Fixed/Mobile	4	25	0

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

DIGI:

[A08-HASM-560](#)