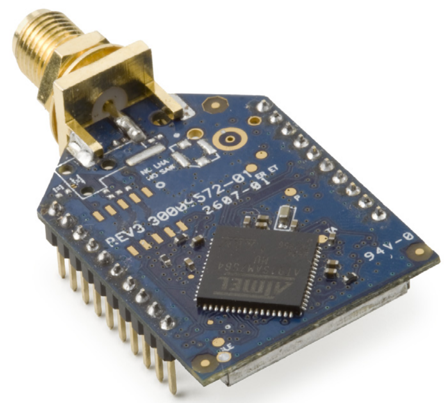
XBee-PRO® 868 OEM RF Modules



OEM RF Modules by Digi International
Firmware version:

102x XBee-PRO 868

Digi International Inc.
11001 Bren Road East
Minnetonka, MN 55343
877 912-3444 or 952 912-3444
http://www.digi.com

90001020_A
11/14/2008

Technical Support: Phone: (801) 765-9885

Live Chat: www.digi.com

E-support: http://www.digi.com/support/eservice/eservicelogin.jsp

Contents

1.XBee-PRO® 868 OEM RF Modules

The XBee-PRO® 868 OEM 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 ISM 868 MHz frequency band.

Key Features

High Performance, Low Cost

- Indoor/Urban: up to 1800 ft (550 m)
- Outdoor line-of-sight: up to 25 miles (40 km)
- Transmit Power Output: 1 mW to 315 mW (0dBm to +25dBm)
- Receiver Sensitivity: -112 dBm
- RF Data Rate: 24 kbps

Advanced Networking & Security

- Retries and Acknowledgements
- Point-to-point, point-to-multipoint and peer-to-peer topologies supported
- AES 128 bit encryption
- 16-bit Network ID

Low Power

XBee-PRO® 868

- TX Current: 85-500mA, depending on power level setting
- RX Current: 65mA (@3.3 V)

Easy-to-Use

- No configuration necessary for out-of-the box RF communications
- AT and API Command Modes for configuring module parameters
- Small, XBee-compatible form factor
- Extensive command set
- Free X-CTU Software (Testing and configuration software)

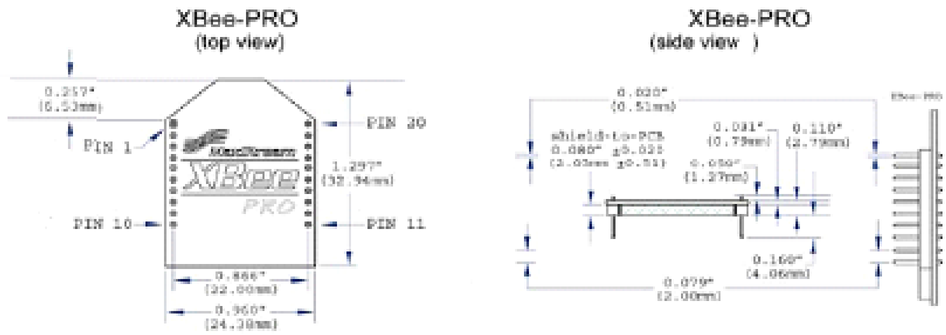
Specifications

Specifications of the XBee-PRO® 868 OEM RF Module

Specification	XBee-PRO® 868
Performance	
Indoor/Urban Range	up to 1800 ft (550 m)
Outdoor RF line-of-sight Range	up to 25 miles (40 km) w/ 2.0dBi dipole antenna up to 50 miles (80 km) w/ high gain antenna
Transmit Power Output	85-500mA, depending on power level setting
RF Data Rate	24 kbps
Throughput	2.4 kbps
Duty Cycle	10%
Receiver Sensitivity	-112dBm
Serial Interface	
UART	3.3V CMOS (5V 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	500mA Typical, (800 mA max)
Operating Current (Receive)	65 mA Typical
General	
Operating Frequency Band	SRD g3 Band (869.525 MHz)
Dimensions	0.962 in x 1.312 in (2.443 cm x 3.332 cm)
Operating Temperature	-40°C to 75°C @ 3.0 to 3.6V -40°C to 85°C @ 3.3 to 3.6V, 0 to 95% non-condensing
Connector Options	1/4 wave wire antenna, RPSMA RF connector, U.Fl RF connector
Networking & Security	
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 25mW Max (+14 dBm) Slovak Republic 10mW max (+10 dBm)

Mechanical Drawings

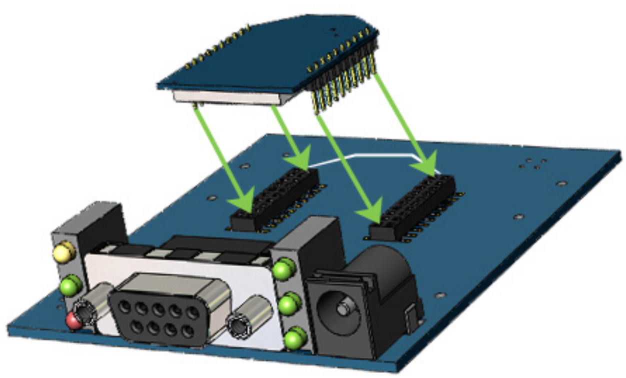
Mechanical drawings of the XBee-PRO® 868 OEM RF Modules (antenna options not shown)



Mounting Considerations

The XBee-PRO® 868 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.

XBee-PRO® 868 Module Mounting to an RS-232 Interface Board.

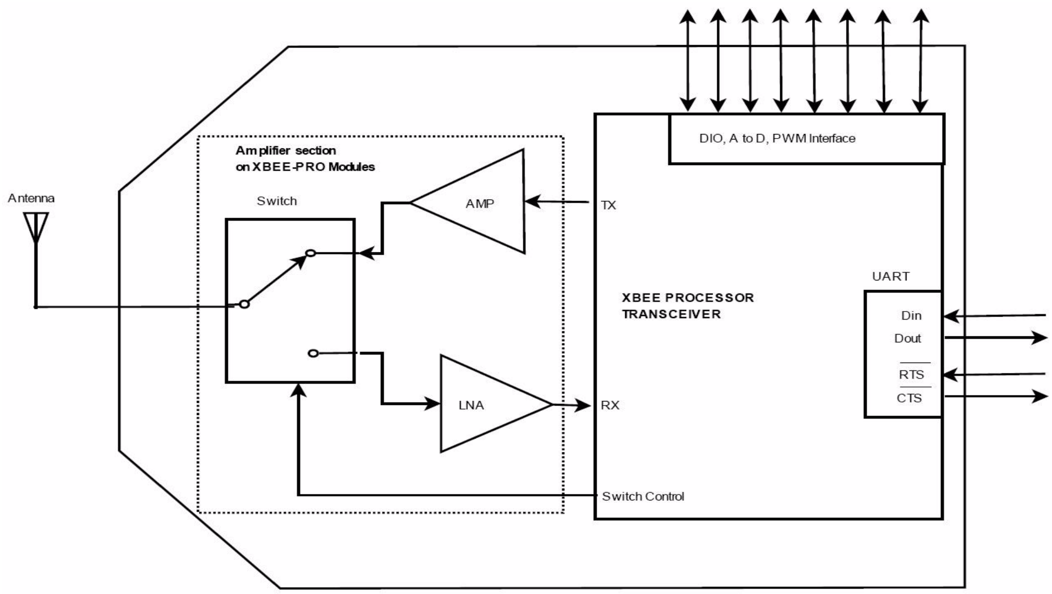


The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

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

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® 868 RF Module Pin Number

(top sides shown - shields on bottom)

Pin Assignments for the XBee-PRO® 868 Modules

(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description	Max Voltage
1	Vcc	-	Power supply	3.6V
2	DOUT	Output	UART Data Out	Vcc
3	DIN / $\overline{\text{CONFIG}}$	Input	UART Data In	5V
4	DIO12*	Either	Digital I/O 12	Vcc
5	$\overline{\text{RESET}}$	Input/Open drain output	Module Reset (reset pulse must be at least 100 us. This must be driven as a open drain/collector. The module will drive this line low when a reset occurs. This line should never be driven high.)	Vcc
6	PWM0 / RSSI / DIO10*	Either	PWM Output 0 / RX Signal Strength Indicator / Digital IO	Vcc
7	PWM / DIO11*	Either	Digital I/O 11	Vcc
8	[reserved]	-	Do not connect	n/a
9	$\overline{\text{DTR}}$ / SLEEP_RQ*/ DIO8*	Either	Pin Sleep Control Line or Digital IO 8	5V
10	GND	-	Ground	0V
11	DIO4*/AD4*	Either	Digital I/O 4	Vcc
12	$\overline{\text{CTS}}$ / DIO7*	Either	Clear-to-Send Flow Control or Digital I/O 7	Vcc
13	ON / $\overline{\text{SLEEP}}$	Output	Module Status Indicator or Digital I/O 9	Vcc
14	Vref	-	This line must be connected if AD sampling is desired. Must be between 2.6 V and Vcc.	Vcc
15	Associate / DIO5/AD5*	Either	Associated Indicator, Digital I/O 5	Vcc
16	$\overline{\text{RTS}}$ / DIO6*	Either	Request-to-Send Flow Control, Digital I/O	5V
17	AD3* / DIO3*	Either	Analog Input 3 or Digital I/O 3	Vcc
18	AD2* / DIO2*	Either	Analog Input 2 or Digital I/O 2	Vcc
19	AD1* / DIO1*	Either	Analog Input 1 or Digital I/O 1	Vcc
20	AD0* / DIO0*	Either	Analog Input 0, Digital I/O 0	Vcc

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections to support serial firmware upgrades: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module.
- Module includes an internal 5K Ohm resistor to $\overline{\text{RESET}}$.
- Several of the input pull-ups can be configured using the PR command. Disable internal pull up resistors to lines that will have 5 volts on the input. During reset and power up the internal pull-ups are enabled and can provide a higher current path for voltages above Vcc.
- Unused pins should be left disconnected.
- To ensure proper power up, Vcc SLOPE must be superior or equal to 6V/ms.

*Please note that I/O sampling and Sleep are not supported at this time.

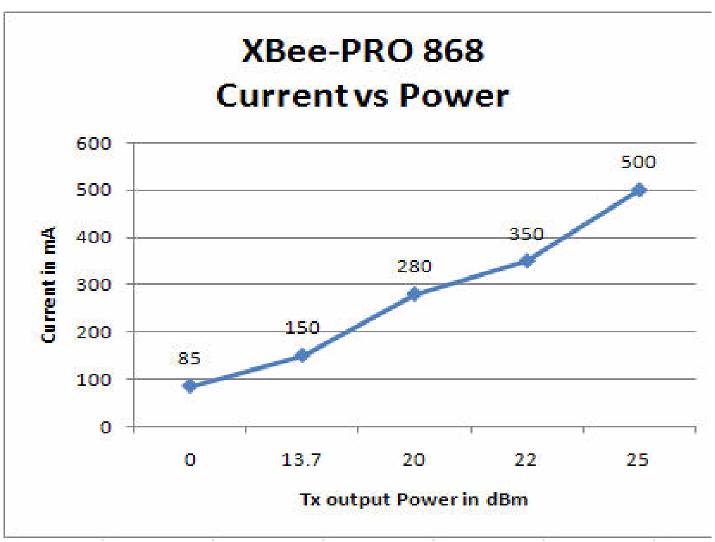
Electrical Characteristics

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 mA, VCC >= 3.0 V	-	-	0.18*VCC	V
V_{OH}	Output High Voltage	I_{OH} = 2 mA, VCC >= 3.0 V	0.82*VCC	-	-	V
II_{IN}	Input Leakage Current	V_{IN} = VCC or GND, all inputs, per pin	-	-	0.5uA	uA
Iout	Output Current	Dout, DIO(0,1,2,3,6,7,8), On/Sleep	-	-	8	mA
Iout	Output Current	DIO9, DIO10, DTR	-	-	16	mA
Iout	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 120mA.

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

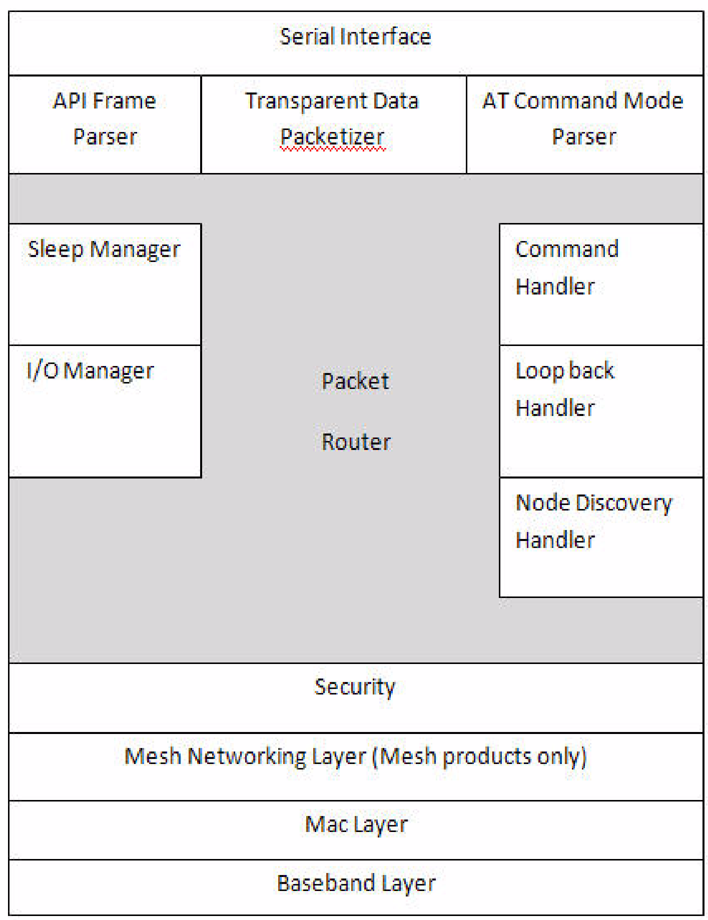
Note: These modules are designed to maximize the range of the radio. When testing modules on the same desk or in close proximity, use power level 0 and keep them at least 3 feet (1 meter) apart in order to avoid saturating the receiver. When using higher power levels keep the modules at least 24 feet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



2. RF Module Operation

Overview

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

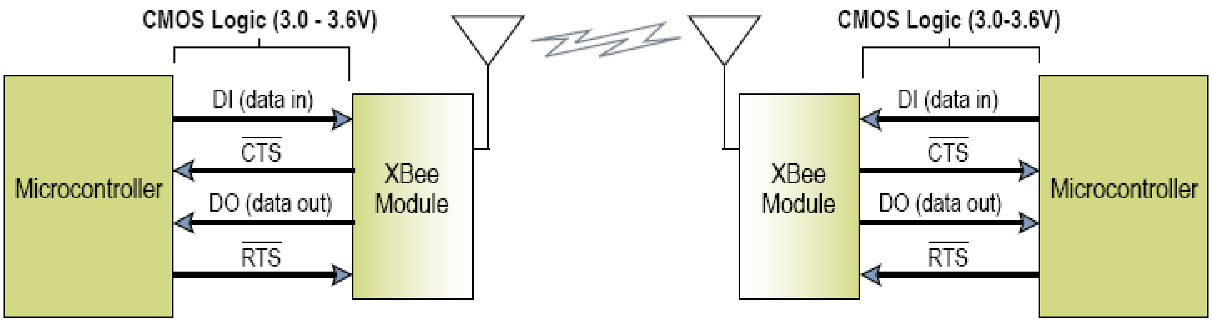


Serial Interface

The XBee module interfaces 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 (e.g. through a Digi RS-232 or USB interface board).

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

System Data Flow Diagram in a UART-interfaced environment
(Low-asserted signals distinguished with horizontal line over signal name.)

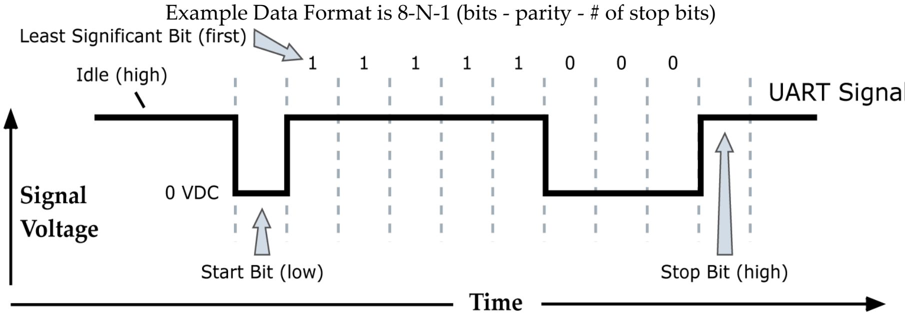


Serial Data

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

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

UART data packet 0x1F (decimal number "31") as transmitted through the RF module
Example Data Format is 8-N-1 (bits - parity - # of stop bits)

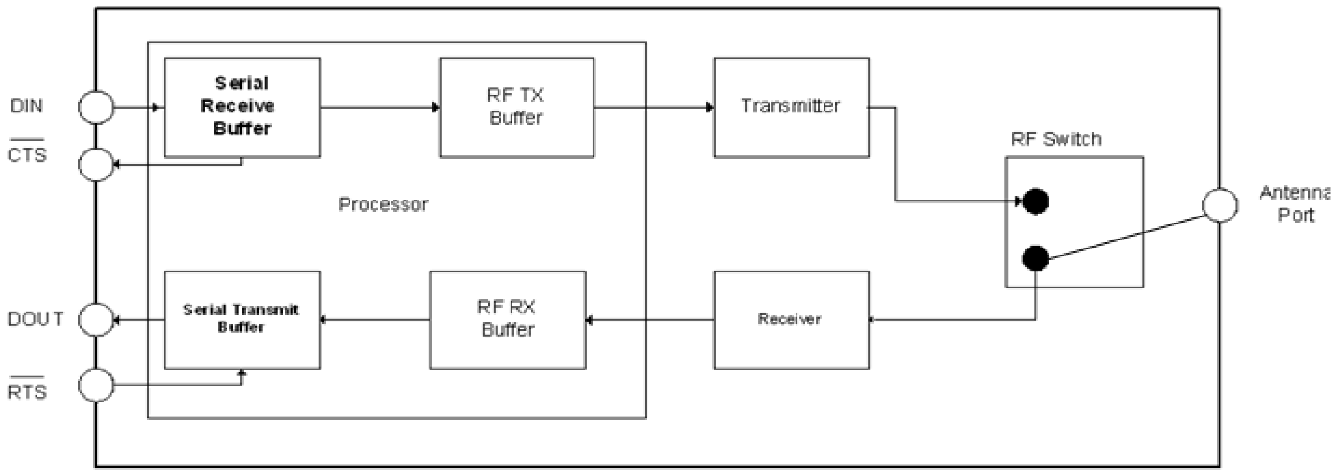


The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

Serial Buffers

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

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. The size of the serial receive buffer can be read with the FT command. This number may vary slightly with different firmware revisions.

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.
- After transmitting the data, the module may need to retransmit the data if an acknowledgment is not received, or if the transmission is a broadcast. These issues could delay the processing of data in the serial receive buffer.

Serial Transmit Buffer

When RF data is received, the data is moved into the serial transmit buffer and 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 interface data rate of the module is set lower than the RF data rate, the module could receive RF data faster than it can send data out the serial port.
- 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 (RTS).

Serial Flow Control

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

CTS Flow Control

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

Note: It is recommended to monitor the CTS line because of the duty cycle. If the radio cannot transmit because it will exceed the duty cycle, the UART buffer could be filled up with data and trigger the CTS line to de-assert.

RTS Flow Control

If RTS flow control is enabled (D6 command), data in the serial transmit buffer will not be sent out the DOUT pin as long as RTS is de-asserted (set high). The host device should not de-assert 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.

Transparent Operation

When operating in Transparent Operation, 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 the DOUT pin. The module configuration parameters are configured using the AT command mode interface. (See RF Module Operation --> Command Mode.)

Serial-to-RF Packetization

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 maximum number of characters that will fit (256) in an RF packet is received.

The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the serial receive buffer before the sequence is transmitted.

API Operation

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

Transmit Data Frames (received through the DIN pin (pin 3)) include:

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

Receive Data Frames (sent out the DOUT pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, etc.

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

To implement API operations, refer to the API Operation chapter 6.

Serial Configuration Options

The serial interface baud rate, parity, and flow control options can be configured using these AT commands: BD (baud rate), NB (parity), D6 ($\overline{RTS}$), and D7 ($\overline{CTS}$) .

See the specific commands for details.

Serial Data Processing

When serial data is received, the AT command mode parser evaluates the data to determine if the command mode sequence has been received. If the command mode sequence is received, the AT command mode parser starts the AT command mode timer and forwards all received serial data to the command mode handler. See the Command Mode section later in this chapter for details.

If the command mode sequence has not been received and command mode is not active, all received serial data is forwarded to either the API Frame Parser or the Transparent Packetizer, based on the AP command.

If AP is 0, the API frame parser is disabled, and all received serial characters are sent to the transparent packetizer. For all other AP values, serial data is forwarded to the API frame parser for processing. The following sections provide more detailed operation for these two entities.

API Frame Parser

If the AP command is set to 1 or 2, the API frame parser is used to send and receive serial data. When an API frame is received, the API frame parser determines the length of the API frame and continues processing bytes until the expected number of bytes have been received. After all bytes have been received, the packet checksum is computed and compared against the checksum at the end of the API frame. If the packet checksum matches the checksum byte read at the end of the API frame, the API frame is processed based on the API frame type (i.e. AT command, remote command, transmit data, etc). If the received API frame is not valid, it is discarded. If AP is 2 (escaping), the API frame parser will reset and begin processing a new API frame anytime a 0x7E character is received.

All received RF data is sent out the UART as API frames. When an RF data transmission is received, the API frame parser calculates a length, inserts the received data into the RF payload field, and calculates a checksum on the packet. If AP is 2, the API frame parser will also escape all bytes in the API frame that require escaping.

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

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

- RF Transmission frames (0x10, 0x11)
- AT command frames (0x08, 0x09, 0x17).Receive data frames (sent out the DOUT pin (pin 2)) include:
- RF-received data frames (0x90, 0x91)
- AT command responses (0x88, 0x97)
- Event notifications such as reset, associated, transmit status, etc (0x8A, 0x8B).

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

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

See the AP (serial API mode) and AO (API output) commands for details.

Transparent Packetizer

The XBee module operates in transparent mode by default (when AP is 0). In this mode, the radio behaves as a serial cable replacement. All received serial data is sent to the transparent packetizer where it is assembled into an RF packet for transmission. Received serial 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 is 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

When AP is 0 and the module is not in AT command mode, all received RF application data is forwarded to the transparent packetizer and sent out the UART. The transparent packetizer sends the RF payload of the received packet out the UART as it was received over the RF link.

See the AP, RO, FT, NP, DL, DH, CI, DE, and SE commands for details.

Comparing Transparent Operation and the API

The serial interface of the XBee module must be configured to use either transparent or API operation using the AP command. The following table compares the advantages of transparent and API operating modes.

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.
Explicit addressing support	API transmit and receive frames can expose explicit addressing fields including source and destination endpoints, cluster ID and profile ID.
Advanced networking diagnostics	API frames can provide indication of IO samples from remote devices, and node identification messages.
Remote Configuration	Set / read configuration commands can be sent to remote devices to configure them as needed using the API.

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

Packet Router

The Packet Router resides between the serial and RF interfaces and routes incoming serial and RF data packets as needed. For example, when the AT command mode parser receives an AT command, the packet router forwards the command to the Command Handler. When the Command Handler executes the AT command and generates a response, the Packet Router sends the response data to the serial interface layer for transmission out the UART.

Similarly, when RF packets are received, the Packet Router forwards the received RF data to the correct handler for processing. The Packet Router evaluates the destination endpoint, profile ID, and cluster ID fields in the data transmission to deliver the data to the correct handler. See the CI, DE, and SE commands and explicit API frame (0x11) for details.

The Packet Router has access to several handler entities shown in the previous figure. A description of each handler follows.

See the AP, RO, FT, NP, DL, DH, CI, DE, and SE commands for details.

Sleep Manager

The sleep manager determines when the XBee can enter a low power (sleep) state. See the SM, ST, SP, SN, and SO commands for details.

I/O Manager

The I/O manager monitors the configuration of the DIO and AIO lines on the module. It is also responsible for taking IO samples either at a periodic rate, or when explicitly queried. See the IS, IC, and IR commands for sampling options, and the D0 - D9, P0 - P2, M0, M1, PR, %V, IP, and R# commands for IO line configuration details.

Command Handler

The command handler receives AT commands and executes the commands. AT commands can be received over the RF link (remote API commands), or through the serial interface.

Loopback Handler

The loopback handler receives RF data on cluster ID 0x12 and sends a response transmission to the sender on cluster ID 0x11 to the source endpoint of the original transmission.

Node Discovery Handler

When a node discovery takes place (ATND or ATDN), the node discovery handler receives the RF request and generates a node discovery response transmission. See the ATND, ATDN, ATNI, and ATNT commands.

RF Transmission Layers

The RF transmission layers provide an interface between the Packet Manager and the RF port. It is divided into the following layers:

- Security
- Network Layer (ZigBee and DigiMesh products only)
- MAC Layer
- Baseband / PHY Layer.

Each of these is described below.

Security

The security layer encrypts outgoing data (for RF transmission) using AES-128 bit encryption. When RF data packets are received (and encrypted), the security layer is responsible for decrypting the packets before they reach the Packet Manager. See the EE, EO, and KY commands for details.

Network Layer

In ZigBee and DigiMesh products, the network layer populates the source and destination addresses for the transmission. If an RF transmission will traverse multiple hops, the network source and destination addresses are set to the addresses of the original source and the final destination.

When RF data is received, the network layer checks the network destination address against its own address. If the network destination address matches the address of the device, the packet is passed up to the Security layer. Otherwise, the device checks its routing table to transmit the packet to the next hop device.

Devices that implement the network layer (ZigBee and DigiMesh products) are able to establish multi-hop routes to route data from any source to any destination device in the network. The network layer performs route discovery and can support network retries to help ensure reliable end-to-end packet delivery. Note the network layer (and mesh routing) is not supported on all XBee platforms.

See the NH and MR commands in the command table.

Mac Layer

The Mac layer provides reliable packet delivery from one device to another. The Mac appends a CRC (cyclical redundancy check) to the end of outgoing RF packets to ensure packet integrity. In addition, it manages retries and acknowledgment processing to help improve reliable packet delivery. The Mac sends outgoing RF data packets to the Baseband layer for transmission over-the-air.

When an RF packet is received from the Baseband layer, the Mac layer verifies the destination address in the transmission matches the address of the device. If the addresses do not match, the packet is discarded. Otherwise, if the CRC is valid, the packet is sent to the Network Layer (for mesh platforms) or to the Security layer (for non-mesh platforms).

See the ID, HP, RR, MT, GD, ER, TR, TA, DB, SH, and SL commands for details.

Baseband

This layer is responsible for encoding and transmitting, or receiving and decoding RF data bits. It provides a direct interface between the Mac layer and the RF port.

See the CH (operating channel) and PL (power level) commands in the command table.

Modes of Operation

Idle Mode

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

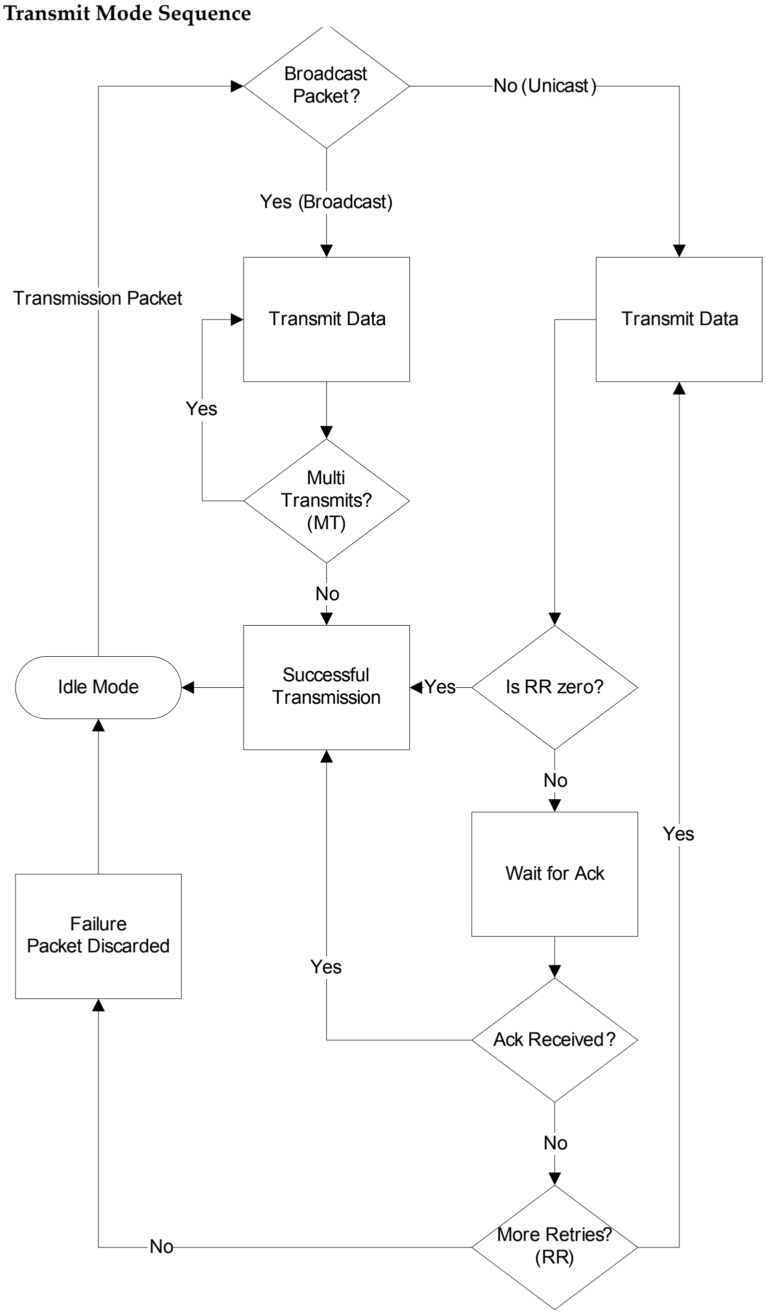
- Transmit Mode (Serial data in the serial receive buffer is ready to be packetized)
- Receive Mode (Valid RF data is received through the RF port)
- Command Mode (Command Mode Sequence is issued)

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.

If a route is not known, route discovery will take place for the purpose of establishing a route to the destination node. If a module with a matching network address is not discovered, the packet is discarded. The data will be transmitted once a route is established. If route discovery fails to establish a route, the packet will be discarded.

Transmit Mode Sequence



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

Duty Cycle

The duty cycle of this radio is 10% averaged over the period of 1 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 of 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°C.

Receive Mode

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

Command Mode

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

AT Command Mode

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

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

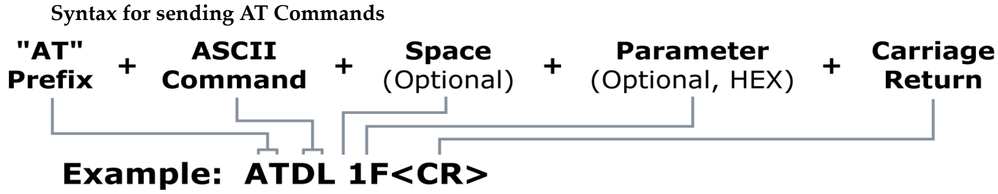
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters ("+++") within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

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. Ensure 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 below.

Syntax for sending AT Commands



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

The preceding example would change the RF module Destination Address (Low) to "0x1F". To store the new value to non-volatile 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.

System 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.

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

   [OR]

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

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

3. 868 Command Reference Tables

Special

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

Addressing

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 to 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. 0x0000 0000 0000 FFFF FFFF is the broadcast address.	0 to 0xFFFFFFFF	0x0000FFFF
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	0x-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 to 0xFFFFFFFF	Factory
SE	**Source Endpoint**. Set/read the application layer source endpoint value. If application layer addressing is enabled , this value will be used as the source endpoint for all data transmissions. SE is only supported in AT firmware.The default value 0xE8 (Data endpoint) is the Digi data endpoint	0 to 0xF0	0xE8
DE	**Destination Endpoint**. Set/read application layer destination ID value. If application layer addressing is enabled, this value will be used as the destination endpoint all data transmissions. DE is only supported in AT firmware. The default value (0xE8) is the Digi data endpoint.	0 to 0xF0	0xE8
CI	**Cluster Identifier**. Set/read application layer cluster ID value. If application layer addressing is enabled, this value will be used as the cluster ID for all data transmissions. CI is only supported in AT firmware.The default value0x11 (Transparent data cluster ID).	0 to 0xFF	0x11

Mac-level Commands

AT Command	Name and Description	Parameter Range	Default
MT	**Multiple Transmissions**. Specifies the number of additional broadcast retransmissions. All broadcast packets are transmitted MT+1 times to ensure the receiving radio(s) receive the packet.	0 to 0x0F	0x3
PL	**Power Level**. Set/view transmitter RF power. See the table under electrical characteristics in chapter 1.	0 to 4	4
RR	**Mac Retries**. Specifies the number of retries that can be sent for a given unicast RF packet. RR specifies the maximum number of retries, and a retry occurs whenever a unicast requesting an ACK is sent and a timeout occurs awaiting the ACK. If RR is 0, an ACK will not be requested nor expected, and no retries will occur.	0 to 0x0F	0xA

Serial Interfacing (I/O)

AT Command	Name and Description	Parameter Range	Default
AP	**API mode**. Set or read the API mode of the radio. The following settings are allowed: **0** API mode is off. All UART input and output is raw data and packets are delineated using the RO and RB parameters. **1** API mode is on. All UART input and output data is packetized in the API format, without escape sequences. **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.	0, 1, or 2	0
AO	API Output Format. Enables different API output frames. Options include: 0 Standard Data Frames (0x90 for RF rx) 1 Explicit Addressing Data Frames (0x91 for RF rx)	0, 1	0
BD	**Baud rate**. Set or read serial interface rate (speed for data transfer between radio modem and host). Values from 0-8 select preset standard rates. Values at 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: **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	0 to 8, and 0x39 to 0x1c9c38	0x03 (9600 bps)
RO	**Packetization Timeout**. Set/Read number of character times of inter-character silence required before packetization. Set (RO=0) to transmit characters as they arrive instead of buffering them into one RF packet.	0 to 0xFF [x character times]	3
FT	**Flow Control Threshhold**. Set or read flow control threshhold. 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.	17 to 319	0x013F = 319
NI	**Node Identifier.** Stores a string identifier. The register only accepts printable ASCII data. In AT Command Mode, a string can not start with a space. A carriage return ends the command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ND (Node Discover) command. This identifier is also used with the DN (Destination Node) command.	20-Byte printable ASCII string	Space Character
SB	**Stop Bits**. Set or read number of stop bits used for UART communications. The values from 0 to 4 are interpreted as follows: 0 =1 stop bit 1 = 2 stop bits	0 to 1	0 (1 stop bit)
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 to 1, 3 to 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 to1, 3 to 5	0

I/O Commands

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 to 5	1

AT Command	Name and Description	Parameter Range	Default
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 to 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 to 5	0
RP	**RSSI PWM Timer.** Time RSSI signal will be output after last transmission. When RP = 0xFF, output will always be on.	0 to 0xFF [x 100 ms]	2032 3.2 seconds)
D0	**AD0/DIO0 Configuration**. Configure options for the AD0/DIO0 line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored* 4 = Digital output low 5 = Digital output high	0, 3 to 5	0
D1	**AD1/DIO1 Configuration**. Configure options for the AD1/DIO1 line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored* 4 = Digital output low 5 = Digital output high	0, 3 to 5	0
D2	**AD2/DIO2 Configuration**. Configure options for the AD2/DIO2 line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored* 4 = Digital output low 5 = Digital output high	0, 3 to 5	0
D3	**AD3/DIO3 Configuration**. Configure options for the AD3/DIO3 line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored* 4 = Digital output low 5 = Digital output high	0, 3 to 5	0
D4	**AD4/DIO4 Configuration**. Configure options for the AD4/DIO4 line of the module. Options include: 0 = Input, unmonitored 3 = Digital input, monitored* 4 = Digital output low 5 = Digital output high	0, 3 to 5	0
D5	**AD5/DIO5 Configuration**. Configure options for the AD5/DIO5 line of the module. Options include: 0 = Input, unmonitored 1 = Power LED output 3 = Digital input, monitored* 4 = Digital output low 5 = Digital output high	0 to 1, 3 to 5	1
M0	**Set PWM0 Duty Cycle**. Be sure to set P0 to 2. The PWM period is 64usec. 0=0%, 0x1FF=50%, 0x3FF=100%.	0 to 0x3FF	0
M1	**Set PWM1 Duty Cycle** Be sure to set P1 to 2. The PWM period is 64usec. 0=0%, 0x1FF=50%, 0x3FF=100%.	0 to 0x3FF	0

AT Command	Name and Description	Parameter Range	Default
PR	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. Bits:" 0 - DIO4/AD4 (Pin 11) 1 - AD3 / DIO3 (Pin 17) 2 - AD2 / DIO2 (Pin 18) 3 - AD1 /DIO1 (Pin 19) 4 - AD0 / DIO0 (Pin 20) 5 - RTS / DIO6 (Pin 16) 6 - DTR / SLEEP_RQ/DI08 / DIO8 (Pin 9) 7 - DIN / Config (Pin 3) 8 - Associate / DIO5 (Pin 15) 9 - On/Sleep / DIO9 (Pin 13) 10 - DIO12 (Pin 4) 11 - PWM0 / RSSI / DIO10 (Pin 6) 12 - PWM1 / DIO11 (Pin 7) 13 - DIO7/CTS (Pin 12) 14 - DOUT (Pin 2) Several of the input pull-ups can be configured using the PR command. Disable internal pull up resistors to lines that will have 5 volts on the input. Keep in mind that during reset and power up the internal pull-ups are enabled and can provide a higher current path for voltages above Vcc.	0 to 7FFF	0x3F7F

*Please note that I/O sampling is not supported at this time.

Diagnostics

AT Command	Name and Description	Parameter Range	Default
VR	**Firmware Version**. Read firmware version of the module.	0 to 0xFFFF [read-only]	Firmware-set
HV	**Hardware Version**. Read hardware version of the module.	0 to 0xFFFF [read-only]	Factory-set
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%.	0 to 0x64	--
VL	**Version Long**. Shows detailed version information including application build date and time.	--	--
ER	**RF Errors** Read the number of times we found the end of a valid preamble with a valid hopping sequence and a valid packet type and some kind of error occurred in reading the rest of the frame. When the value reaches 0xffff, it stays there.	0 to 0xFFFF	0
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.	0 to 0xFFFF	0
RC	**RSSI for channel** Reads the DBM level of the designated channel.	0 to 11	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 to 0xFF	0x20 = 3.2 seconds
TR	**Transmission Errors** Read the number of MAC frames that exhaust MAC retries without ever receiving a MAC acknowledgement message from the adjacent node. When the value reaches 0xffff, it stays there.	0 to 0xFFFF	0
DB	**Received Signal Strength**. This command reports the received signal strength of the last received RF data packet. The value returned as though it is positive although the actual value is negative.	0 to 115	0x8000
NP	Maximum RF Payload Bytes. This value returns the maximum number of RF payload bytes that can be sent in a unicast transmission.	0 to 0x100	0x100
TA	**Transmit Acknowledgement Errors** Incremented once for each failed ack retry.	0 to 0xFFFF	0
TP	**Temperature** Read module temperature in Celsius. Negatives temperatures can be returned.	0xFF74 to 0x0258	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
%V	**Supply Voltage.** Reads the voltage on the Vcc pin in mV. Should be read module voltage in milivolts.	0 to 0xEA0	n/a

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.	0 to 0x28F	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 to 0xCE4	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 to 0xFF	0x2B

Node Identification

AT Command	Name and Description	Parameter Range	Default
ID	**Network ID**. Set or read the user network address. Nodes must have the same network address to communicate. Changes to ID should be written to non-volatile memory using the WR command.	0x0000 to 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.	0x20 to 0x2EE0 [x 100 msec]	0x82 (130d)
NO	**Network Discovery options.** Set/Read the options value for the network discovery command. The options bitfield value can change the behavior of the ND (network discovery) command and/or change what optional values are returned in any received ND responses or API node identification frames. Options include: 0x01 = Append DD value (to ND responses or API node identification frames) 0x02 = Local device sends ND response frame when ND is issued.	0 to 0x3 [bitfield]	0
DD	**Device Type Identifier.** Stores a device type value. This value can be used to differentiate multiple XBee-based products.	0 to 0xFFFFFFFF	Factory-set [read-only]
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 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	**Discover Node - Destination Node**. Resolves an NI (Node Identifier) string to a physical address (casesensitive). The following events occur after the destination node is discovered: <AT Mode> 1. DL & DH are set to the extended (64-bit) address of the module with the matching NI (Node Identifier) string. 2. OK (or ERROR)\r is returned. 3. Command Mode is exited to allow immediate communication <API Mode> The 16-bit network and 64-bit extended addresses are returned in an API Command Response frame. If there is no response from a module within (NT * 100) milliseconds or a parameter is not specified (left blank), the command is terminated and an “ERROR” message is returned. In the case of an ERROR, Command Mode is not exited.	20 byte ascii string	

AT Command	Name and Description	Parameter Range	Default
ND	**Node Discover** Discovers and reports all RF modules found. The following information is reported for each module discovered. MY<CR> SH<CR> SL<CR> NI<CR> (Variable length) PARENT_NETWORK ADDRESS (2 Bytes)<CR> DEVICE_TYPE<CR> (1 Byte: 0=Coord) STATUS<CR> (1 Byte: Reserved) PROFILE_ID<CR> (2 Bytes) MANUFACTURER_ID<CR> (2 Bytes) <CR> 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.		

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 to 1	0
KY	**Security Key** Sets the 16 byte - 128-bit 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.	0 to (32 hex digits all set to 'F')	n/a

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.

API Frame Specifications

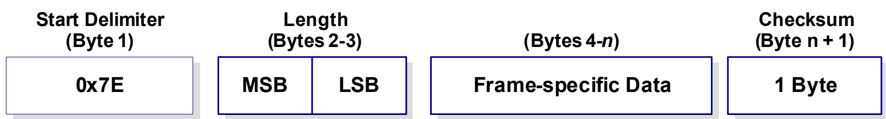
Two API modes are supported and both can be enabled using the AP (API Enable) command. Use the following AP parameter values to configure the module to operate in a particular mode:

- 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:



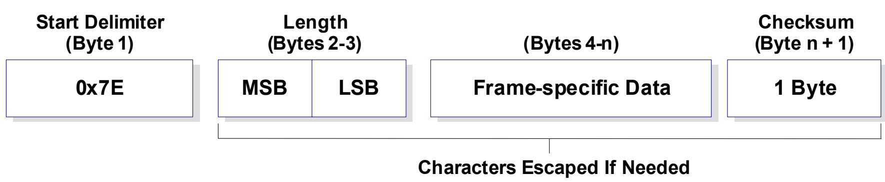
MSB = Most Significant Byte, LSB = Least Significant Byte

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 - with Escape Characters (AP parameter = 2)

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

UART Data Frame Structure - with escape control characters:



MSB = Most Significant Byte, LSB = Least Significant Byte

Escape characters. When sending or receiving a UART data frame, specific data values must be escaped (flagged) so they do not interfere with the data frame sequencing. To escape aninterfering 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.

Checksum

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

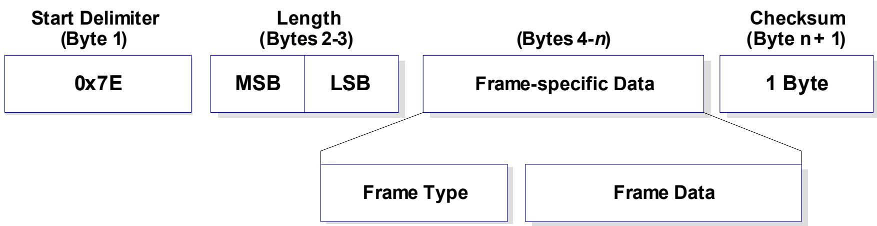
To calculate: Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract the result from 0xFF.

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

API Frames

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

UART Data Frame & API-specific Structure:



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

API Frame Names and Values

API Frame Names	Frame Type
Modem Status Frame	0x8A
AT Command Frame	0x08
AT Command - Queue Parameter Value Frame	0x09
AT Command Response Frame	0x88
Remote Command Request Frame	0x17
Remote Command Response Frame	0x97
Transmit Frame	0x10
Explicit Addressing Transmit Frame	0x11
Transmit StatusFrame	0x8B
Receive Frame (AO=0)	0x90
Explicit Addressing Receive Frame	0x91

Modem Status Frame

Frame Type: (0x8A)

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

	Frame Fields		Offset	Example	Description
API Packet	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	0 = Hardware reset 1 = Watchdog timer reset
	Checksum		5	0x75	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: The modem sends this API frame with the modem is reset or powered on.

AT Command Frame

Frame Type: (0x08)
Allows for module parameter registers to be queried or set.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x06	
	Frame-specific Data	Frame Type	3	0x08	
		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.
		AT Command	5	'I' = 0x49	Command Name - Two ASCII characters that identify the AT Command.
			6	'D' = 0x44	
		AT Command Data	7	0x01	Register data is sent as binary values. Note there are certain commands which take a null terminated array of bytes.
			8	0x23	
	Checksum		9	0x45	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: Send a command to set the network ID parameter to 123.

A string parameter used with the NI (Node Identifier), ND (Node Discover) and DH (Destination Address High) command is terminated with a 0x00 character.

AT Command - Queue Parameter Value Frame

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.

	Frame Fields		Offset	Example	Description
API Packet	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 (acknowledgement). If set to 0, no response is sent.
		AT Command	5	'B' = 0x42	Command Name - Two ASCII characters that identify the AT Command.
			6	'D' = 0x44	
		AT Command Data	7	0x07	Register data is sent as binary values. Note there are certain commands which take a null terminated array of bytes.
	Checksum		8	0x68	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

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.)

AT Command Response Frame

Frame Type: (0x88)
Response to previous command.

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

	Frame Fields		Offset	Example	Description
API Packet	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 following response would be received.

Remote AT Command Request Frame

Frame Type: (0x17)

Allows for module parameter registers on a remote device to be queried or set

	Frame Fields		Offset	Example	Description
API Packet	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 being reported. Note: If Frame ID = 0 in AT Command Mode, no AT Command Response will be given. .
		64-bit Destination Address	MSB 5	0x00	
			6	0x13	Indicates the 64-bit address of the remote module that is responding to the Remote AT Command request
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x40	
			11	0x11	
			LSB 12	0x22	
		16- Destination Network Address	MSB 13	0xFF	Set to the 16-bit network address of the remote. Set to 0xFFFE if unknown.
			LSB 14	0xFE	
		Command Options	15	0x02	0x00 = Queue Parameter Value 0x02 = Apply Value Change
		Command Name	16	'B' = 0x42	Name of the command. Two ASCII characters that identify the AT command
			17	'H' = 0x48	
		Command Data	18	0x01	The register data is sent as binary values
	Checksum		19	0xF5	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Send 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 0x0013A20040401122, and the destination 16-bit address is unknown.

Remote Command Response Frame

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
API Packet	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	0x97	
		Frame ID	4	0x55	This is the same value passed in to the request. .
		64-bit Destination Address	MSB 5	0x00	The address of the remote radio returning this response.
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x40	
			11	0x11	
			LSB 12	0x22	
		16-bit Destination Network Address	MSB 13	0xFF	Set to match the 16-bit network address of the destination, MSB first, LSB last. Set to 0xFFFE for broadcast TX, or if the network address is unknown.
			LSB 14	0xFE	
		Command Name	15	'B' = 0x42	Name of the command
			16	'H' = 0x48	
		Command Status	17	0x00	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter 4 = Tx Failure
		Command Data	18	0x01	Register data in binary format. If the register was set, then this field is not returned.
	Checksum		19	0x23	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: This example shows a response to the command to query the broadcast hops (BH) register, which was returned as 1. The orginal remote command had a Frame ID of 0x55 and was sent to the node with an address of 0x0013A20040401122.

Transmit Frame

Frame Type: (0x10) A TX Request message will cause the module to send RF Data as an RF Packet.

	Frame Fields		Offset	Example	Description
API Packet	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 this frame, so when you receive a Transmit Status frame, you can tell that it matches this request. Setting Frame ID to '0' will disable the Transmit Status frame.
		64-bit Destination Address	MSB 5	0x00	Broadcast = 0x000000000000 FFFF
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x0A	
			11	0x01	
			LSB 12	0x27	
		16-bit Destination Network Address	MSB 13	0xFF	Set to 0xFFFE for this product.
			LSB 14	0xFE	
		Broadcast Radius	15	0x00	Set to 0
		Transmit Options Bitfield	16	0x00	0x01-Disable ACK 0x10-Duty purge packet. (If a packet would be delayed because of duty cycle, then purge the packet.)
		RF Data	17	0x54	Up to 256 Bytes per packet
			18	0x78	
			19	0x44	
			20	0x61	
			21	0x74	
			22	0x61	
			23	0x30	
			24	0x41	
	Checksum		25	0x64	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: Send a transmission to a module with destination address 0x0013A200 40014011, payload "TxData1B".

Explicit Addressing Command Frame

Frame Type: (0x11)

Allows application addressing layer fields (endpoint and cluster ID) to be specified for a data transmission.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x19	
	Frame-specific Data	Frame Type	3	0x11	
		Frame ID	4	0x01	Set to a value that will be passed back in the Tx Status frame. 0 disables the Tx Status frame.
		64-bit Destination Address	MSB 5	0x00	Broadcast = 0x000000000000 FFFF
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x01	
			10	0x02	
			11	0x03	
			LSB 12	0x04	
		16-bit Destination Network Address	MSB 13	0xFF	Set to 0xFFFE for this product.
			LSB 14	0xFE	
		Source Endpoint	15	0xE8	
		Destination Endpoint	16	0xE8	
		ClusterID	MSB 17	0x00	
			LSB 18	0x11	
		Profile ID	MSB 19	0xC1	
			LSB 20	0x05	
		Broadcast Radius	21	0x00	0x00 for this product.
		Transmit Options Bitfield	22	0x00	0x01 - Disable ACK 0x10 - If the packet would be delayed due to duty cycle then purge it."
		RF Data	23	'H' (0x48)	Up to 72 Bytes per packet
			24	'e' (0x65)	
			25	'l' (0x6c)	
			26	'l' (0x6c)	
			27	'o' (0x6f)	
	Checksum		28	0x96	0xFF - the 8 bit sum of bytes from offset 3 to this byte

Example: The following API frame sends the string "Hello" to a destination with 64-bit address: 0x0013A200 01020304. It is sent to cluster ID of 0x11, profile ID of 0xC105, and endpoints of 0xE8. The application frame ID is set to 0x01.

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
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x07	
	Frame-specific Data	Frame Type	3	0x8B	
		Frame ID	4	0x01	This value was used on the Transmit or Explicit transmit frame so that the status can be correlated.
		16-bit Destination Network Address	MSB 5	0xFF	Set to 0xFFFE for this product.
			LSB 6	0xFE	
		Retries	7	0x00	The number of retries it took to send the packet.
		Delivery Status	8	0x00	0x00 = Success 0x01 = Mac Ack Failure 0x03 = Purged
		Discovery Status	9	0x00	0x00 = No Discovery Overhead
	Checksum		10	0x76	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Example: This example represents a status from a previously sent transmit frame with a Frame ID of 0x01.

Receive Packet Frame

Frame Type: (0x90)

When the module receives an RF packet, it is sent out the UART using this message type if AO is set to 0.

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x12	
	Frame-specific Data	Frame Type	3	0x90	
		64-bit Source address	MSB 4	0x00	Source address of the packet
			5	0x13	
			6	0xA2	
			7	0x00	
			8	0x40	
			9	0x52	
			10	0x2B	
			LSB 11	0xAA	
		16-bit Source network address	MSB 12	0xFF	Always 0xFFFE for this product.
			LSB 13	0xFE	
		Receive Options Bitfield	14	0x01	0x01 - Packet Acknowledged 0x02 - Packet was a broadcast packet
		RF Data	15	0x52	Up to 256 bytes.
			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: Suppose a device with a 64-bit address of 0x0013A200 sends a unicast data transmission to a remote device with payload "RxData".

Explicit Addressing Receive Frame

Frame Type:(0x91)

When the modem receives a RF packet it is sent out the UART using this message type if AO is set to 1.

	Frame Fields		Offset	Example	Description
API Packet	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	4	0x00	Destination 64-bit (extended) address. Set to 0xFFFF for broadcast.
			5	0x13	
			6	0xA2	
			7	0x00	
			8	0x40	
			9	0x52	
			10	0x2B	
			11	0xAA	
		16-bit Source network address	12	0xFF	Always 0xFFFE for this product.t
			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 Bitfield	20	0x02	0x01 - Packet Acknowledged 0x02 - Packet was a broadcast packet
		RF Data	21	0x52	Up to 256 bytes.
			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: Suppose 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 of 0x2211, and profile ID of 0xC105. If AO is1 on the receiving device, it would send the following frame out its UART.

Appendix A: Definitions

Definitions

Terms and Definitions

Network ID	Radios with different network IDs will not receive packets from each other and can be used to create separate unique RF networks.
Network Address	The 16-bit address is used for compatibility and has an address of 0xFFFE

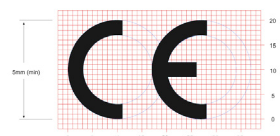
Appendix B: Agency Certifications

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

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 5mm 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® 868 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® 868 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: Most European countries require that the power output of the XBee-PRO 868 RF modules must not exceed 27 dBm. The power level is set using the PL command. Italy and Slovak Republic have unique restrictions.

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

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

Appendix C: 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:

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

Where:

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

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 10dBm 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.

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
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-HAUM-560	Dipole	U.FL Female	2	0	Fixed/Moble	4	25	0
A08-HASM-560	Dipole	RPSMA-F	2	0	Fixed/Moble	4	25	0
Integrated	Mono-pole	Integrated	1.7	-0.3	Fixed/Moble	4	25	0