XBee® Wi-Fi RF Module

WiFi RF Modules by Digi International
Firmware version: 200x

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

90002180_A

© 2012 Digi International, Inc. All rights reserved
No part of the contents of this manual may be transmitted or reproduced in any form or by any means without the written permission of Digi International, Inc.
XBee® is a registered trademark of Digi International, Inc.

Technical Support Phone: (866) 765-9885 toll-free U.S.A. & Canada
(801) 765-9885 Worldwide
8:00 am - 5:00 pm [U.S. Mountain Time]
Online Support: http://www.digi.com/support/eservice/login.jsp
Email: rf-experts@digi.com

Contents

XBee® Wi-Fi RF Module 1
1.Overview 7
Specifications 7
General Specifications 7
RF Specifications 7
Electrical Specifications 12
Serial Communications Specifications 12
UART 12
SPI 13
GPIO Specifications 13
Agency Approvals 14
Pin Signals 14
Design Notes 15
Power Supply 15
Recommended Pin Connections 16
Board Layout 16
Mounting Considerations – Xbee Wi-Fi Through-hole 19
2. RF Module Operation 20
Serial Communications 20
UART Communications 20
SPI Communications 21
Serial Buffers 23
Serial Receive Buffer 23
Serial Transmit Buffer 23
UART Flow Control 24
Serial Interface Protocols 25
Transparent Operation 25
API Operation 25
A Comparison of Transparent and API Operation 26
Modes of Operation 27
Idle Mode 27
Transmit Mode 27
Receive Mode 27
Command Mode 27

Configuration Mode 29
Forcing Entry into Configuration Mode 29
Using X-CTU to Enter Configuration Mode 30
Sleep Mode 30
3. 802.11 bgn Networks 31
Infrastructure Networks 31
Ad Hoc Networks 31
Network Basics 32
XBee® Wi-Fi Standards 33
Encryption 33
Channels 33
4. XBee IP Services 35
XBee Application Service 35
Local Host 35
Network Client 36
Sending Configuration Commands 37
Sending Serial Data Command to XBee 38
Sending Over-the-Air Firmware Upgrades 39
Serial Communication Service 40
Transparent Mode 40
API Mode 40
5. Sleep 42
Using Sleep Mode: UART 42
Using Sleep Mode: SPI 42
Sleep Options 43
AP Associated Sleep 43
Deep Sleep (Non-Associated Sleep) 44
Sampling Data Using Sleep Modes 44
Sample Rate (ATIR) 45
Wake Host 45
6. Advanced Application Features 46
XBee Analog and Digital I/O Lines 46
I/O Sampling 47
Queried Sampling 48
Periodic I/O Sampling 48

I/O Examples 49
General Purpose Flash Memory 49
Accessing General Purpose Flash Memory 50
Working with Flash Memory 56
Over-the-Air Firmware Upgrades 56
Distributing the New Application 57
Verifying the New Application 58
Installing the Application 58
Things to Remember 58
7. API Operation 59
API Frame Specifications 59
API UART and SPI Exchanges 61
AT Commands 61
Transmitting and Receiving RF Data 62
Remote AT commands 62
Supporting the API 63
API Frames 64
TX (Transmit) Request: 64-Bit 64
AT Command 65
AT Command-Queue Parameter Value 66
Remote AT Command Request 67
Transmit (TX) Request: IPv4 68
Rx (Receive) Packet: 64-bit 69
AT Command Response 70
Modem Status 71
Transmission Status 72
IO Data Sample RX Indicator 73
Remote Command Response 75
RX (Receive) Packet: IPv4 76
8. XBee Command Reference Tables 77
Addressing 77
Networking Commands 78
Security Commands 78
RF Interfacing Commands 78
Serial Interfacing 79

I/O Settings 80
Diagnostics Interfacing 83
AT Command Options 84
Sleep Commands 84
Execution Commands 85
9. Module Support 86
X-CTU Configuration Tool 86
Serial Firmware Updates 86
Regulatory Compliance 86
10. Agency Certifications 87
United States FCC 87
Europe (ETSI) 94
OEM Labeling Requirements 94
Restrictions 95
Declarations of Conformity 95
Approved Antennas 96
Canada (IC) 97
Labeling Requirements 97
Transmitters with Detachable Antennas 97
Australia (C-Tick) 98
11. Warranty Information 99
1-Year Warranty 99
12.Glossary of Terms 100
Definitions 100

1.Overview

The XBee® Wi-Fi RF module provides wireless connectivity to end-point devices in 802.11 bgn networks. Using the 802.11 feature set, these modules are interoperable with other 802.11 bgn devices, including devices from other vendors. With XBee, users can have their 802.11 bgn network up-and running in a matter of minutes.

The XBee® Wi-Fi modules are compatible with other devices that use 802.11 bgn technology. These include Digi external 802.11x devices like the ConnectPort products and the Digi Connect Wi-SP, as well as embedded products like the ConnectCore series and Digi Connect series of products. More information on these Digi products can be found at:
http://www.digi.com/products/wireless/wifisolutions/

Specifications

General Specifications

Specification	XBee Wi-Fi Through-hole	XBee Wi-Fi Surface Mount
Dimensions	0.960 x 1.297 (2.438cm x 3.294cm)	0.866 x 1.330 in (2.200 x 3.378 cm)
Operating Temperature	-30 to 85° C	
Antenna Options	PCB Antenna, U.FL Connector, RPSMA Connector, or Integrated Wire	PCB Antenna, U.FL Connector, or RF Pad

RF Specifications

Specification	XBee Wi-Fi Through-hole		XBee Wi-Fi Surface Mount	
Frequency	ISM 2.4-2.5GHz			
Number of Channels	13			
Adjustable Power	Yes			
Wi-Fi Standards	802.11 b, g, and n			
Transmit Power Output (Average)	Up to +16 dBm (See table below)			
FCC/IC Test Transmit Power Range (Peak)	802.11b 802.11g 802.11n (800 ns GI) 802.11n (400 ns GI)	2.73 to 26.81 dBm 7.87 to 28.52 dBm 8.03 to 28.75 dBm 8.04 to 28.64 dBm	802.11b 802.11g 802.11n (800 ns GI) 802.11n (400 ns GI)	2.08 to 26.13 dBm 7.15 to 27.72 dBm 7.02 to 27.89 dBm 7.33 to 28.20 dBm
RF Data Rates	1 Mbps to 72.22 Mbps (See table below)			
Receiver Sensitivity (25 °C, <10% PER)	-93 to -71 dBm (See table below)			
Transmit Spectral Mask (25 °C, max output power)	Meets IEEE standards for 802.11 b, g, and n			

RF Data Rates

RF Data Rates			
Standard	**Data rates (Mbps)**		
802.11b	1, 2, 5.5, 11		
802.11g	6, 9, 12, 18, 24, 36, 48, 54		
Standard	**MCS index**	**Data rates (Mbps)**	
		800 ns guard interval	**400 ns guard interval**
802.11n	0	6.5	7.22
	1	13	14.44
	2	19.5	21.67
	3	26	28.89
	4	39	43.33
	5	52	57.78
	6	58.5	65
	7	65	72.22

Receiver Sensitivity

Receiver Sensitivity (25 °C, < 10% PER)		
Standard	**Data rate**	**Sensitivity (dBm)**
802.11b	1 Mbps	-93
	2 Mbps	-91
	5.5 Mbps	-90
	11 Mbps	-87
802.11g	6 Mbps	-91
	9 Mbps	-89
	12 Mbps	-88
	18 Mbps	-86
	24 Mbps	-83
	36 Mbps	-80
	48 Mbps	-76
	54 Mbps	-74
802.11n	MCS 0 6.5/7.22 Mbps	-91
	MCS 1 13/14.44 Mbps	-88
	MCS 2 19.5/21.67 Mbps	-85
	MCS 3 26/28.89 Mbps	-82
	MCS 4 39/43.33 Mbps	-78
	MCS 5 52/57.78 Mbps	-74
	MCS 6 58.5/65 Mbps	-73
	MCS 7 65/72.22 Mbps	-71

RF Transmit Power - Typical

RF Transmit Power (Average)			
Standard	**Data rate**	**Power (dBm)**	
		North America	**Europe**
802.11b	1 Mbps		
	2 Mbps		
	5.5 Mbps		
	11 Mbps	16	15
802.11g	6 Mbps		
	9 Mbps		
	12 Mbps		
	18 Mbps		
	24 Mbps		
	36 Mbps	16	15
	48 Mbps	14	14
	54 Mbps	14	14
802.11n	MCS 0 6.5/7.22 Mbps		
	MCS 1 13/14.44 Mbps		
	MCS 2 19.5/21.67 Mbps		
	MCS 3 26/28.89 Mbps		
	MCS 4 39/43.33 Mbps		
	MCS 5 52/57.78 Mbps	15	15
	MCS 6 58.5/65 Mbps	14	14
	MCS 7 65/72.22 Mbps	8.5	8.5

EVM – Typical, Maximum Output Power

EVM (25 °C, max output power)		
Standard	**Data rate**	**EVM (dB)**
802.11b	1 Mbps	-40
	2 Mbps	-40
	5.5 Mbps	-36
	11 Mbps	-38
802.11g	6 Mbps	-23
	9 Mbps	-23
	12 Mbps	-23
	18 Mbps	-23
	24 Mbps	-23
	36 Mbps	-23
	48 Mbps	-26
	54 Mbps	-26
802.11n	MCS 0 6.5/7.22 Mbps	-24
	MCS 1 13/14.44 Mbps	-24
	MCS 2 19.5/21.67 Mbps	-24
	MCS 3 26/28.89 Mbps	-25
	MCS 4 39/43.33 Mbps	-25
	MCS 5 52/57.78 Mbps	-25
	MCS 6 58.5/65 Mbps	-26
	MCS 7 65/72.22 Mbps	-28

Electrical Specifications

Specification	XBee Wi-Fi		
Supply Voltage	3.14 - 3.46 VDC		
Operating Current (transmit, max output power)	802.11b	1 Mbps	309 mA
		2 Mbps	
		5.5 Mbps	
		11 Mbps	
	802.11g	6 Mbps	275 mA
		9 Mbps	
		12 Mbps	
		18 Mbps	
		24 Mbps	
		36 Mbps	
		48 Mbps	231 mA
		54 Mbps	222 mA
	802.11n	MCS 0 6.5/7.22 Mbps	262 mA
		MCS 1 13/14.44 Mbps	
		MCS 2 19.5/21.67 Mbps	
		MCS 3 26/28.89 Mbps	
		MCS 4 39/43.33 Mbps	
		MCS 5 52/57.78 Mbps	234 mA
		MCS 6 58.5/65 Mbps	217 mA
		MCS 7 65/72.22 Mbps	184 mA
Operating Current (Receive)	100mA		
Deep Sleep Current	6 µA @25°C		
Associated Sleep current	2 mA asleep, 100 mA awake. (See AP Associated Sleep section for details.)		

Serial Communications Specifications

The XBee Wi-Fi RF modules support both UART (Universal Asynchronous Receiver/Transmitter) and SPI slave mode (Serial Peripheral Interface in slave mode only) serial connections.

UART

Specification	XBee Wi-Fi Through-hole	XBee Wi-Fi Surface Mount
UART Pins	Module Pin Number	Module Pin Number
DIO13/DOUT	2	3
DIO14/DIN	3	4
DIO7/nCTS	12	25
DIO6/nRTS	16	29

More information on UART operation is found in the UART section in chapter 2.

SPI

Specification	XBee Wi-Fi Through-hole	XBee Wi-Fi Surface Mount
SPI Pins	Module Pin Number	Module Pin Number
DIO2/SPI_SCLK	18	14
DIO3/SPI_nSSEL	17	15
DIO4/SPI_MOSI	11	16
DIO12/SPI_MISO	4	17
DIO1/SPI_nATTN	19	12

For more information on SPI operation see the SPI section in chapter 2.

GPIO Specifications

The XBee Wi-Fi modules have 14 GPIO (General Purpose Input Output) ports available. Those available will depend on the module configuration as some GPIO's are consumed by serial communication, etc.

See GPIO section for more information on configuring and using GPIO ports

Electrical Specification for GPIO pads

Parameter	Condition	Min	Max	Units
Input Low Voltage			0.3VDD	V
Input High Voltage		0.7VDD		V
Output high Voltage relative to VDD	Sourcing 2 mA, VDD=3.3 V	95		%
Output low voltage relative to VDD	Sinking 2 mA, VDD=3.3 V		5	%
Output fall time	2 mA drive strength and load capacitance CL=350-600pF.	20+0.1CL	250	ns
I/O pin hysteresis (VIOTHR+ - Viothr-)	VDD = 3.14 to 3.46 V	0.1VDD		V
Pulse width of pulses to be removed by the glitch suppression filter		10	50	ns

Agency Approvals

Specification	XBee Wi-Fi Through-hole	XBee Wi-Fi Surface Mount
United States (FCC Part 15.247)	FCC ID: MCQ-XBS6B	FCC ID: MCQ-S6BSM
Industry Canada (IC)	IC: 1846A-XBS6B	IC: 1846A-S6BSM
Europe (DC)	ETSI	ETSI
Australia	Pending	Pending
Brazil	Pending	Pending
Japan	Pending	Pending

FCC Approval (USA) Refer to Chapter 12 FCC Requirements. Systems that contain XBee Wi-Fi modules inherit Digi Certifications.

Pin Signals

Pin Assignment for the XBee Wi-Fi Through-hole module

(Low-asserted signals are distinguished with a lower case n before the signal name.)

Pin #	Name	Direction	Default State	Description
1	VCC	-	-	Power Supply
2	DIO13/DOUT	Both	Output	UART Data out
3	DIO14/Din/nConfig	Both	Input	UART Data In
4	DIO12/SPI_MISO	Both	Output	GPIO/ SPI slave out
5	nRESET	Input		Module Reset
6	DIO10/PWM0	Both		GPIO
7	DIO11/PWM1	Both		GPIO
8	reserved	-	Disabled	Do Not Connect
9	DIO8/nDTR/SLEEP_RQ	Both	Input	Pin Sleep Control line /GPIO
10	GND	-	-	Ground
11	DIO4/SPI_MOSI	Both		GPIO/SPI slave In
12	DIO7/nCTS	Both	Output	Clear-to-Send Flow Control/GPIO
13	DIO9/On_nSLEEP	Output	Output	Module Status Indicator/GPIO
14	VREF	Input	-	NC
15	DIO5/Associate	Both	Output	Associate Indicator/GPIO
16	DIO6/nRTS	Both	Input	Request-to-Send Flow Control/GPIO
17	DIO3/AD3 /SPI_nSSEL	Both		Analog Input/GPIO/SPI Slave Select
18	DIO2/AD2 /SPI_CLK	Both		Analog Input/GPIO/SPI Clock
19	DIO1/AD1 /SPI_nATTN	Both		Analog Input/GPIO/SPI Attention
20	DIO0/AD0	Both		Analog Input/GPIO

Pin Assignment for the XBee Wi-Fi Surface Mount module

(Low-asserted signals are distinguished with a lower case n before the signal name.)

Pin #	Name	Direction	Default State	Description
1	GND	-	-	Ground
2	VCC	-	-	Power Supply
3	DIO13/DOUT	Both	Output	UART Data Out
4	DIO14/DIN/Nconfig	Both	Input	UART Data In
5	DIO12	Both	-	GPIO
6	nRESET	Input	-	Module Reset
7	DIO10/PWM0	Both	-	GPIO
8	DIO11/PWM1	Both	-	GPIO
9	Reserved	-	-	Do Not Connect
10	DIO8/nDTR/SLEEP_RQ	Both	-	GPIO
11	GND	-	-	Ground
12	DIO19/SPI_nATTN	Both	-	GPIO/SPI Attention
13	GND	-	-	Ground
14	DIO18/SPI_CLK	Both	-	GPIO/SPI Clock
15	DIO17/SPI_nSSEL	Both	-	GPIO/SPI Slave Select
16	DIO16/SPI_SI	Both	-	GPIO/SPI Slave In
17	DIO15/SPI_SO	Both	-	GPIO/SPI Slave Out
18	Reserved	-	-	Do Not Connect
19	Reserved	-	-	Do Not Connect
20	Reserved	-	-	Do Not Connect
21	Reserved	-	-	Do Not Connect
22	GND	-	-	Ground
23	Reserved	-	-	Do Not Connect
24	DIO4	Both	-	GPIO
25	DIO7/nCTS	Both	Output	Clear-to-Send Flow Control/ GPIO
26	DIO9/On_nSLEEP	Both	-	Module Status Indicator/GPIO
27	VREF	-	-	NC
28	DIO5/ASSOC	Both	-	Associate Indicator/GPIO
29	DIO6/nRTS	Both	Input	Request-to-Send Flow Control/ GPIO
30	DIO3/AD3	Both	-	Analog Input/GPIO
31	DIO2/AD2	Both	-	Analog Input/GPIO
32	DIO1/AD1	Both	-	Analog Input/GPIO
33	DIO0/AD0	Both	-	Analog Input/GPIO
34	Reserved	-	-	Do Not Connect
35	GND	-	-	Ground
36	RF	Both	-	RF IO for RF Pad Variant
37	Reserved	-	-	Do Not Connect

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

Poor power supply can lead to poor radio performance, especially if the supply voltage is not kept within tolerance or is excessively noisy. To help reduce noise, a 1µF and 8.2pF capacitor are recommended to be placed 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 50mV peak to peak.

Recommended Pin Connections

The only required pin connections are VCC, GND, and either DOUT and DIN or SPI_CLK, SPI_nSSEL, SPI_MOSI, and SPI MISO. 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 30k internal pull-up resistors using the PR software command. No specific treatment is needed for unused outputs.

For applications that need to ensure the lowest sleep current, inputs should never be left floating. Use internal or external pull-up or pull-down resistors, or set the unused I/O lines to outputs.

Other pins may be connected to external circuitry for convenience of operation. For example, the Associate signal (pin 15) and the On_nSLEEP signal (pin 13) will change level or behavior based on the state of the module.

Board Layout

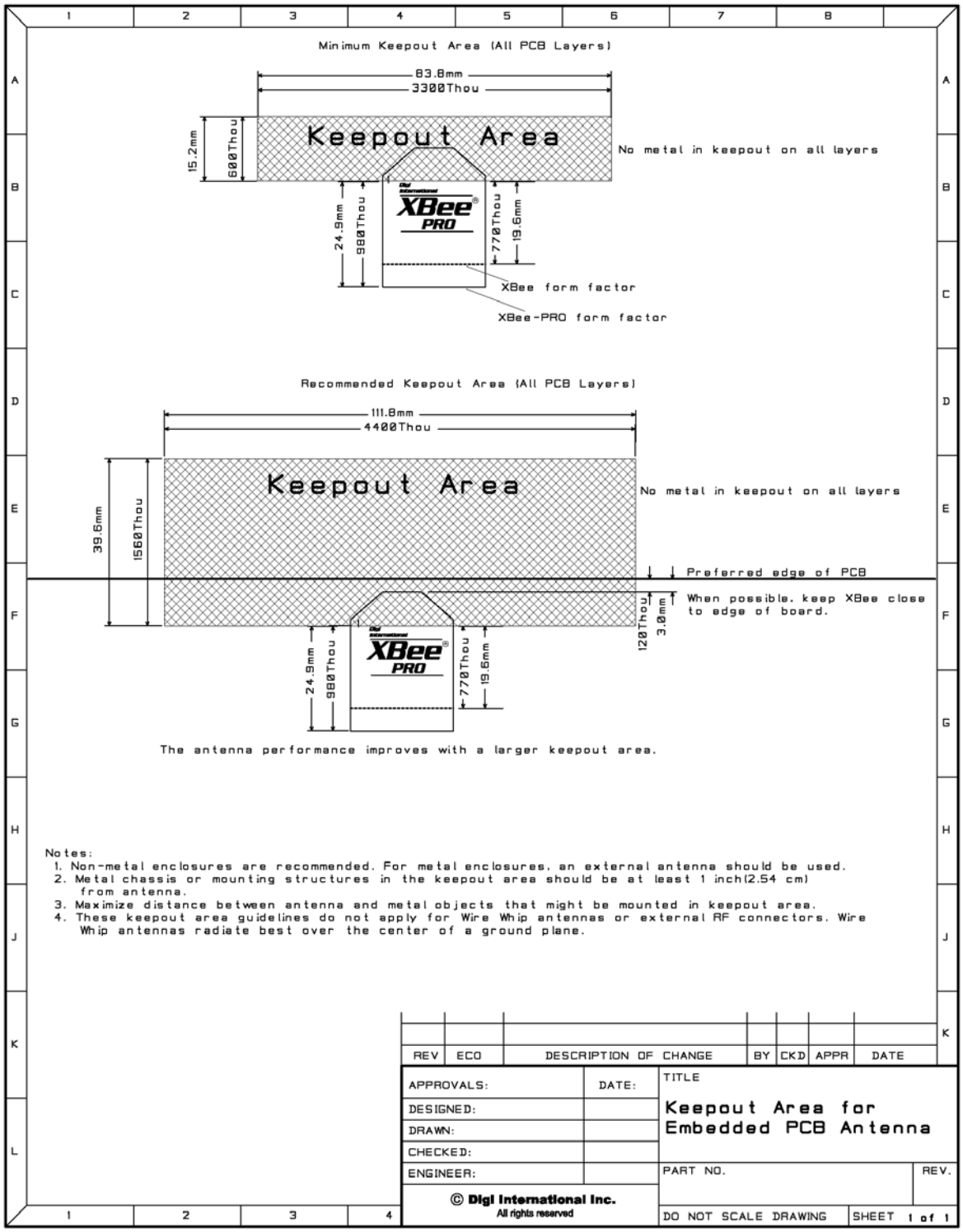
XBee modules do not have any specific sensitivity to nearby processors, crystals or other PCB components. Other than mechanical considerations, no special PCB placement is required for integrating XBee radios except for those with integral antennas. In general, Power and GND traces should be thicker than signal traces and be able to comfortably support the maximum currents.

The radios are also designed to be self sufficient and work with the integrated and external antennas without the need for additional ground planes on the host PCB. However, considerations should be taken on the choice of antenna and antenna location. Metal objects that are near an antenna cause reflections and may reduce the ability for an antenna to efficiently radiate. Using an integral antenna in an enclosed metal box will greatly reduce the range of a radio. For this type of application an external antenna would be a better choice.

External antennas should be positioned away from metal objects as much as possible. Metal objects next to the antenna or between transmitting and receiving antennas can often block or reduce the transmission distance. Some objects that are often overlooked are metal poles, metal studs or beams in structures, concrete (it is usually reinforced with metal rods), metal enclosures, vehicles, elevators, ventilation ducts, refrigerators and microwave ovens.

Antennas should reside above or away from any metal objects like batteries, tall electrolytic capacitors or metal enclosures. Antenna elements radiate perpendicular to the direction they point. Thus a vertical antenna emits across the horizon.

PCB Antennas should not have any ground planes or metal objects above or below the module at the antenna location. For best results, the module should be in a plastic enclosure instead of a metal one. It should be placed at the edge of the PCB to which it is mounted. The ground, power and signal planes should be vacant immediately below the antenna section. The following drawing shows the recommended PCB keepout area for XBee embedded antennas.


Minimum Keepout Area (All PCB Layers)
83.8mm
3300Thou
Keepout Area
No metal in keepout on all layers
15.2mm
600Thou
24.9mm
980Thou
770Thou
19.6mm
XBee PRO
XBee form factor
XBee-PRO form factor
Recommended Keepout Area (All PCB Layers)
111.8mm
4400Thou
Keepout Area
No metal in keepout on all layers
39.6mm
1560Thou
Preferred edge of PCB
When possible, keep XBee close to edge of board.
120Thou
3.0mm
24.9mm
980Thou
770Thou
19.6mm
XBee PRO
The antenna performance improves with a larger keepout area.
Notes:
1. Non-metal enclosures are recommended. For metal enclosures, an external antenna should be used.
2. Metal chassis or mounting structures in the keepout area should be at least 1 inch(2.54 cm) from antenna.
3. Maximize distance between antenna and metal objects that might be mounted in keepout area.
4. These keepout area guidelines do not apply for Wire Whip antennas or external RF connectors. Wire Whip antennas radiate best over the center of a ground plane.
REV
ECO
DESCRIPTION OF CHANGE
BY
CKD
APPR
DATE
APPROVALS:
DATE:
DESIGNED:
DRAWN:
CHECKED:
ENGINEER:
TITLE
Keepout Area for Embedded PCB Antenna
PART NO.
REV.
© Digi International Inc.
All rights reserved
DO NOT SCALE DRAWING
SHEET 1 of 1

Mounting Considerations – Xbee Wi-Fi Through-hole

XBee Through-hole modules were designed to mount into a receptacle (socket) and therefore do not require any soldering when mounting to a board. XBee interface boards provided in XBee Wi-Fi Development Kits have two ten pin receptacles for connecting the module.

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)
- Through-hole single-row receptacles - Mill-Max P/N: 831-43-0101-10-001000
- 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.

2. RF Module Operation

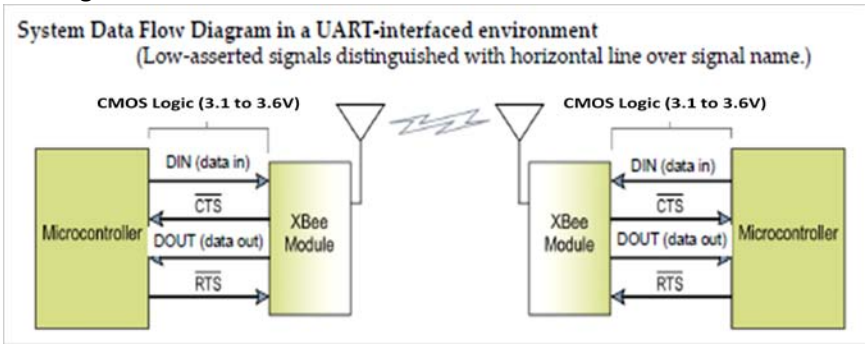
Serial Communications

The XBee RF Modules interface to a host device through a logic-level asynchronous serial port, or a Serial Peripheral Interface (SPI) port. Through its serial ports, the module can communicate with any logic and voltage compatible UART or SPI; or through a level translator to any serial device (for example: through a RS-232 or USB interface board).

UART Communications

UART Data Flow

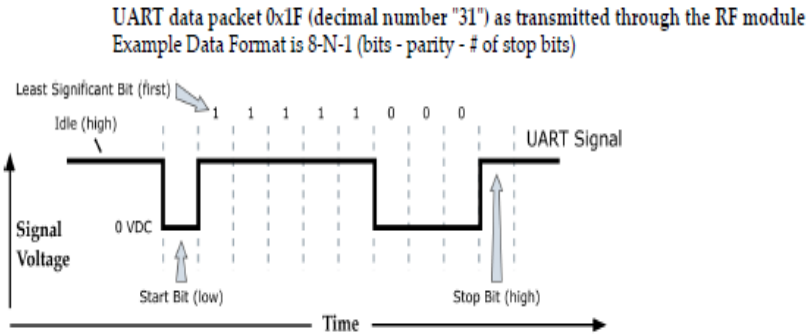
Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.



UART 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.



Serial communications depend on the two UARTs (the microcontroller's and the RF module's) to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

The UART baud rate, parity, and stop bits settings on the XBee module can be configured with the BD, NB, and SB commands respectively. See the command table in chapter 10 for details.

In the rare case that a radio has been configured with the UART disabled, the module may be recovered to the UART operation by holding DIN low at reset time. As always, DIN forces a default configuration on the UART at 9600 baud and it will bring up the module in command mode on the UART port. Appropriate commands can then be sent to the module to configure it for UART operation. If those parameters are written, then the module will come up with the UART enabled, as desired on the next reset.

SPI Communications

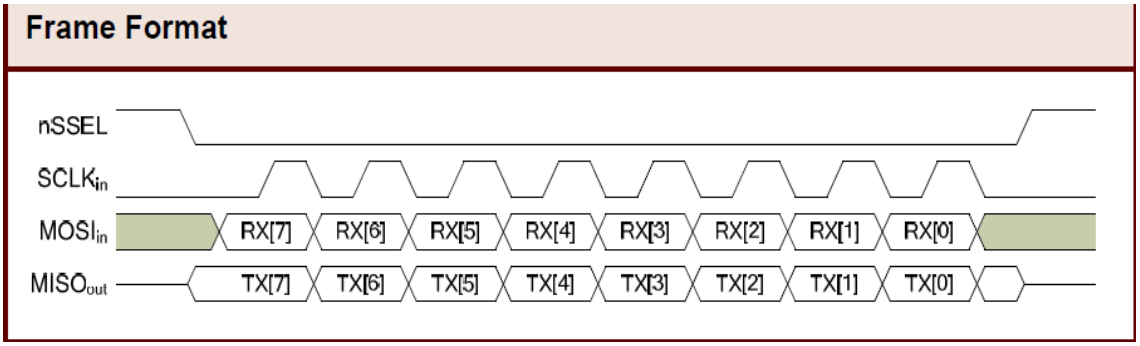
The XBee Wi-Fi module supports SPI communications in the slave mode. Slave mode receives the clock signal and data from the master and returns data to the master. The SPI port uses the following signals on the XBee:

- SPI_MOSI (Master Out, Slave In) – inputs serial data from the master
- SPI_MISO (Master In, Slave Out) – outputs serial data to the master
- SPI_SCLK (Serial Clock) – clocks data transfers on MOSI and MISO
- SPI_nSSEL (Slave Select) – enables serial communication with the slave
- SPI_nATTN(Attention) – alerts the master that slave has data queued to send. The XBee module will assert this pin as soon as data is available to send to the SPI master and it will remain asserted until the SPI master has clocked out all available data.

In this mode the following apply:

- SPI Clock rates up to 6 MHz are possible.
- Data is MSB first
- Frame Format mode 0 is used. This means CPOL=0 (idle clock is low) and CPHA=0 (data is sampled on the clock's leading edge). Mode 0 is diagramed below.
- SPI port is setup for API mode and is equivalent to AP=1.

Frame Format for SPI communications



SPI mode is chip to chip communication. Digi does not supply SPI communication option on the Device Development Evaluation Boards.

SPI mode can be forced by holding DIO13/DOUT (pin 2) low while resetting the module until SPI_nATTN asserts. By this means, the XBee Wi-Fi module will disable the UART and go straight into SPI communication mode. Once configuration is completed, a modem status frame is queued by the module to the SPI port which will cause the SPI_nATTN line to assert. The host can use this to determine that the SPI port has been configured properly. This method internally forces the configuration to provide full SPI support for the following parameters:

- D1 (note this parameter will only be changed if it is at a default of zero when method is invoked)
- D2
- D3
- D4
- P2

As long as a WR command is not issued, these configuration values will revert back to previous values after a power on reset. If a WR command is issued while in SPI mode, these same parameters will be written to flash. After a reset, parameters that were forced and then written to flash become the mode of operation. If the UART is disabled and the SPI is enabled in the written configuration, then the module will come up in SPI mode without forcing it by holding DOUT low. If both the UART and the SPI are enabled at the time of reset, then output will go to the UART until the host sends the first input. If that first input comes on the SPI port, then all subsequent output will go to the SPI port and the UART will be disabled. If the first input comes on the UART, then all subsequent output will go to the UART and the SPI will be disabled. Please note that once a serial port (UART or SPI) has been selected, all subsequent output will go to that port, even if a new configuration is applied. The only way to switch the selected serial port is to reset the module.

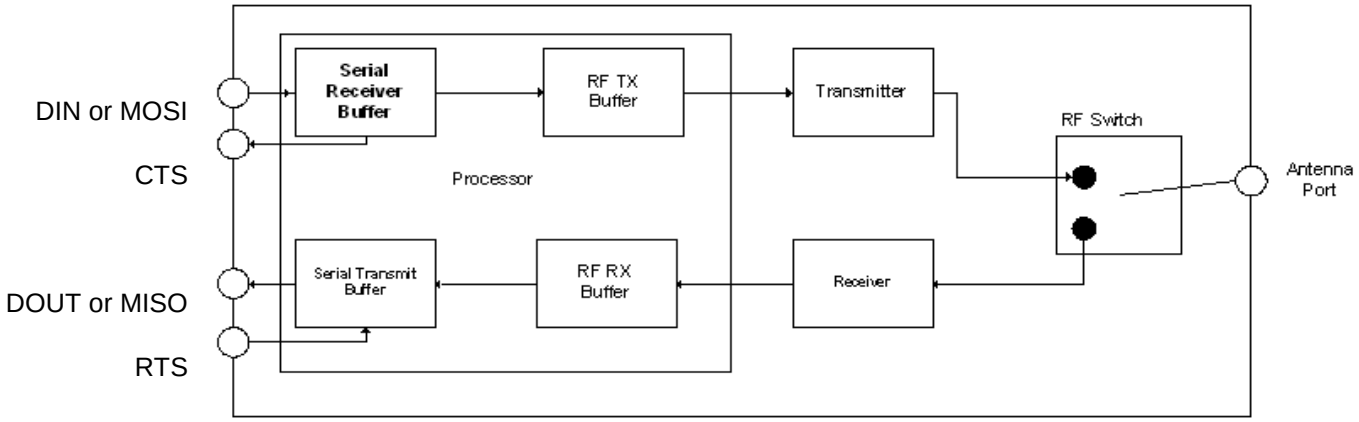
When the slave select (SPI_nSSEL) signal is asserted by the master, SPI transmit data is driven to the output pin SPI_MISO, and SPI data is received from the input pin SPI_MOSI. The SPI_nSSEL pin has to be asserted to enable the transmit serializer to drive data to the output signal SPI_MISO. A falling edge on SPI_nSSEL causes the SPI_MISO line to be tri-stated such that another slave device can drive it, if so desired.

If the output buffer is empty, the SPI serializer transmits the last valid bit repeatedly, which may be either high or low. Otherwise, the module formats all output in API mode 1 format, as described in chapter 7. The attached host is expected to ignore all data that is not part of a formatted API frame.

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 or SPI port.

Internal Data Flow Diagram



Serial Receive Buffer

When serial data enters the RF module through the DIN Pin (or the MOSI pin), 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 such that the serial receive buffer would overflow, then the new data will be discarded. If the UART is in use, this can be avoided by the host side honoring CTS flow control.

Serial Transmit Buffer

When RF data is received, the data is moved into the serial transmit buffer and sent out the UART or SPI port. If the serial transmit buffer becomes full and system buffers are also full, then the entire RF data packet is dropped. Whenever data is received faster than it can be processed and transmitted out the serial port, there is a potential of dropping data, even in TCP mode.

UART Flow Control

The nRTS and nCTS 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.

nCTS Flow Control

The FT command allows the user to specify how many bytes of data can be queued up in the serial transmit buffer before the module asserts CTS low. The serial receive buffer can hold up the 2100 bytes, but FT cannot be set any larger than 2083 bytes, leaving 17 bytes that can be sent by the host before the data is dropped.

By default, FT is 2035 (0x7F3), which allows the host to send 65 bytes to the module after the module asserts CTS before the data is dropped.

In either case, CTS will not be re-asserted until the serial receive buffer has FT-17 or less bytes in use.

nRTS 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 nRTS is de-asserted (set high). The host device should not de-assert nRTS 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.

Note: If RTS flow control is enabled and the XBee is sending data out the UART when nRTS is de-asserted (set high), the XBee could send up to 4 characters out the UART to clear its FIFO after nRTS is de-asserted. This implies that the user needs to de-assert nRTS by the time its receive capacity is within 4 bytes of full.

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 is queued up for RF transmission. When RF data is received, the data is sent out through the UART. The module configuration parameters are configured using the AT command mode interface. Please note that transparent operation is not an option when using SPI.

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 parameter. If RO is zero, data is packetized as soon as it is received, without delay. If RO is non-zero, the data is packetized after RO character times of no transitions on the DIN pin. However, if the time required for RO characters is less than 100 microseconds, then DIN must still be idle for at least 100 microseconds, which is the minimal idle time required for packetizing packets at any baud rate.
- The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the serial receive buffer before the sequence is packetized and transmitted before command mode is entered.
- 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 UART or SPI is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DIN pin (pin 3) or SPI_MOSI (pin 11)) include:

- RF Transmit Data Frame
- Local commands (equivalent to AT commands)
- Remote commands to be sent to another radio

Receive Data Frames (sent out the DOUT pin (pin 2) or SPI_MISO (pin 4)) include:

- RF-received data frames
- Local command responses
- Remote command responses
- I/O samples from a remote radio
- Event notifications such as transmission status, reset, associate, disassociate, etc.

The API provides an alternative means of configuring modules and of routing data at the local host application layer. A local 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 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, transmission status messages, and local radio status 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, (e.g. in a sensor node, or a simple application) then transparent operation might be suitable. It is acceptable to use a mixture of devices running API mode and transparent mode in a network.

Modes of Operation

Idle Mode

When not receiving or transmitting data, the RF module is in Idle Mode. 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)
- Sleep Mode
- Command Mode (Command Mode Sequence is issued)

Transmit Mode

When serial data is received and is ready to be packetized, 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 the API Operation chapter for an alternate means of configuring modules, which is the only method available for SPI mode. (Command mode is unavailable when using the SPI interface.)

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]

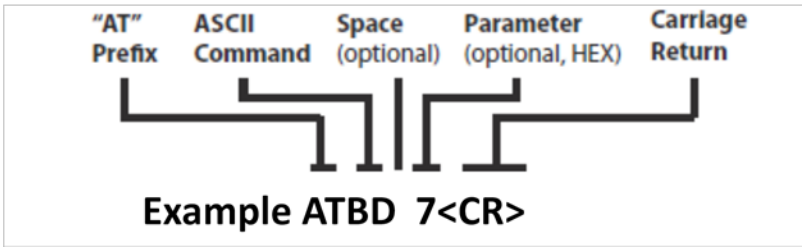
Once the AT command mode sequence has been issued, the module sends an "OK\r" out the UART. The "OK\r" characters can be delayed if the module has not finished transmitting received serial data.

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 UART.

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

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. By default, the BD (Baud Rate) parameter = 3 (9600 bps).

To Send AT Commands, send AT commands and parameters using the syntax shown below:



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

The preceding example would change the RF module baud rate to 7, which would allow operation at 115,200bps. To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

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

Command Response

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

Applying Command Changes

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

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

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, please see the Command Reference Table chapter.

Configuration Mode

The user may not always know the parameters with which the XBee module is configured. If those parameters affect the means by which command mode is entered (and the parameters were previously written to non-volatile memory), then command mode is not available to either read the parameters or to set them to known values. This makes configuration of the XBee difficult unless the user can successfully guess the configuration to allow entry into command mode. A common example of this problem is when the UART baud rate is unknown. In this case, the "+++" sequence to enter command mode would not be recognized due to a baud rate mismatch, preventing entry into command mode.

Forcing Entry into Configuration Mode

To overcome this issue, the XBee may be forced into command mode with a known configuration as follows: While holding DIN low (a.k.a. asserting the break key), reset the module. Rather than coming up in transparent mode, which is normal, it will come up in command mode and issue the OK prompt with the following default parameters applied for operation while in command mode:

- UART enabled (P3=1, P4=1)—only set for SPI-enabled modules.
- 9600 baud rate (BD=3)
- One stop bit (SB=0)
- No parity (NB=0)
- Three character times with no change on DIN before transmission (RO=3)
- No RTS flow control (D6=0)
- CTS flow control (D7=1)
- 65 characters left in transmission buffer before CTS is turned off (FT)
- '+' is used for command mode character (CC=0x2b)
- One second guard time (GT=0x3e8)
- Ten second command mode timeout (CT=0x64).

If configuration mode is left without setting any parameters (i.e. without changing parameter values), then all parameters will revert to their previous unknown state after exiting command mode. Also, any values queried will return the previously written settings rather than the temporarily applied default settings described above.

When the need arises to recover from an unknown configuration to a known configuration, the user should do the following:

1. Set up the interface to the XBee to match the default configuration as described above.
2. Press and hold DIN low while resetting the XBee module.
3. Release DIN (let it be pulled high) so that UART data may be received.
4. At the OK prompt, enter the desired configuration settings. (If desired, configuration settings which were unknown may be read before setting them in this state.)
5. Write the desired configuration to non-volatile memory using the WR command.
6. Set up the interface to the XBee to match the configuration just written to non-volatile memory.
7. Optionally, reset the module and then begin operation in the new mode.

Using X-CTU to Enter Configuration Mode

X-CTU is designed to support a forced configuration on a UART interface following the steps below. (Currently, X-CTU will not work over a SPI interface directly.)

1. Connect an asynchronous serial port of the PC (either RS-232 or USB) to the development board into which the XBee module is plugged.
2. Start X-CTU and go to the PC settings tab.
3. Set parameters as appropriate on the PC settings tab to match the default configuration previously described.
4. Go to the terminal tab and click on the break key. (This holds the DIN line low.)
5. Using the development board, press the reset button
6. Wait for the OK prompt to be displayed
7. Click to de-select the break key so that input can occur on DIN.
8. Within ten seconds of seeing the OK prompt, enter the desired configuration in AT command mode.
9. Enter the WR command to save the parameters to non-volatile memory.
10. Go back to the PC settings tab and set up the PC side of the interface as it was just configured on the XBee.
11. Optionally, reset the XBee module.
12. Go to the terminal tab and begin normal transparent operation.

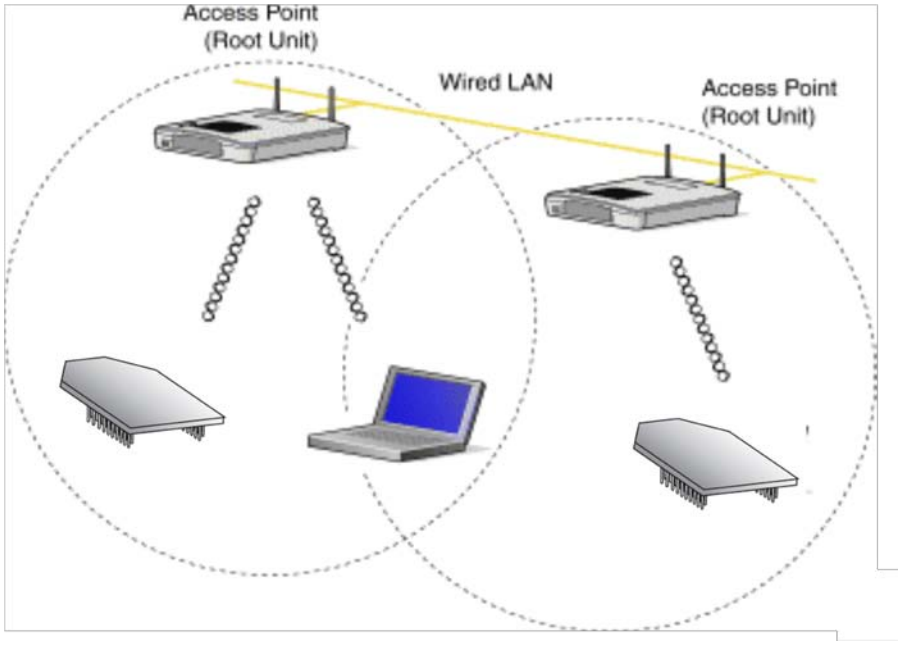
Sleep Mode

Sleep modes allow the RF module to enter states of low power consumption when not in use. The XBee Wi-Fi modules support both pin sleep (sleep mode entered on pin transition) and cyclic sleep (module sleeps for a fixed time). For both pin sleep and cyclic sleep the sleep level may be either deep sleep or associated sleep. XBee sleep modes are discussed in detail in Chapter 5.

3. 802.11 bgn Networks

Infrastructure Networks

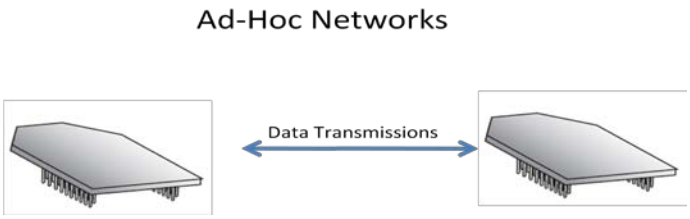
The main type of wireless network will involve a number of wireless devices (called stations) talking through a master wireless device known as an **Access Point** (AP for short). This type of setup is called an **Infrastructure or BSS** (Basic Service Set) network. Most wireless networks are of this type. An example of an infrastructure wireless network is shown below:



Infrastructure Wireless Network

Ad Hoc Networks

Wireless devices can get on a wireless network without an access point. This is called an **Ad Hoc or IBSS** (Independent Basic Service Set) network. An example of an ad hoc wireless network is shown below:



Note that ad hoc networks are point to point and that there can only be two nodes in the network, a creator and a joiner. Set up the creator first, and then the joiner.

Ad Hoc Creator

Set up the following parameters for the creator:

- AH1 designates the node as an Ad hoc creator.
- MA1 specifies static IP addresses. (No DHCP is supported in Ad Hoc mode.)
- EE0 specifies no security. (Security is not available in Ad Hoc mode.)
- CH may be any channel from 1 to 0xB.
- ID sets the SSID, which is any string of choice, as long as it isn't the same as another SSID in the vicinity.
- MY sets IP address of creator node.
- DL specifies IP address of joiner node.
- MK sets IP mask for both of the above addresses.

Ad Hoc Joiner

Set up the following parameters for the joiner:

- AH0 designates the node as an Ad hoc joiner.
- MA1 specifies static IP addresses. (No DHCP is supported in Ad Hoc mode.)
- EE0 specifies no security. (Security is not available in Ad Hoc mode.)
- ID sets the SSID, which must match the ID of the creator. Problems arise if it matches the SSID of an access point in the vicinity.
- MY sets IP address of joiner node.
- DL specifies IP address of creator node.
- MK sets IP mask for both of the above addresses.

Network Basics

Clients will need to join the wireless network before they can send data across it. This is called Association. In order for a device to associate it must know the following items about the desired wireless network:

- **SSID**: the name of the wireless network.
- **Encryption**: if and how the network encrypts or scrambles its data.
- **Authentication**: how and if the network requires its members to —prove their identity.
- **Channel**: what channel (frequency range) the wireless network uses.

Once a device is associated it can send and receive data from other associated devices on the same network. When the client is done or needs to leave, it then can Dis-associate and be removed from the wireless network.

XBee® Wi-Fi Standards

The XBee Wi-Fi module will operate in three of the available 802.11 standards.

802.11 b
The 802.11b standard was approved in July 1999 and can be considered the second generation. 802.11b operates in the 2.4 GHz frequency ISM band. The data rate is from 1 to 11 Mbps.

802.11 g
The 802.11g standard was approved in 2003. It provides a maximum data rate of 54 Mbps. In addition, the standard is also fully backwards-compatible with existing 802.11b wireless networks.

802.11 n
The 802.11n standard was approved in 2009. It provides for data rates up to 300Mbps. The XBee® Wi-Fi module uses the single stream n mode with 20MHz bandwidth and is capable of 65 Mbps over the air in n mode.

Encryption

Encryption is a method of scrambling a message that makes it unreadable to unwanted parties, adding a degree of secure communications. There are different protocols for providing encryption, and the XBee Wi-Fi module supports WPA, WPA2, and WEP.

Authentication
Authentication deals with proving the identity of the wireless device attempting to associate with the network. There are different methods of doing this. The XBee Wi-Fi module supports Open and Shared Key.

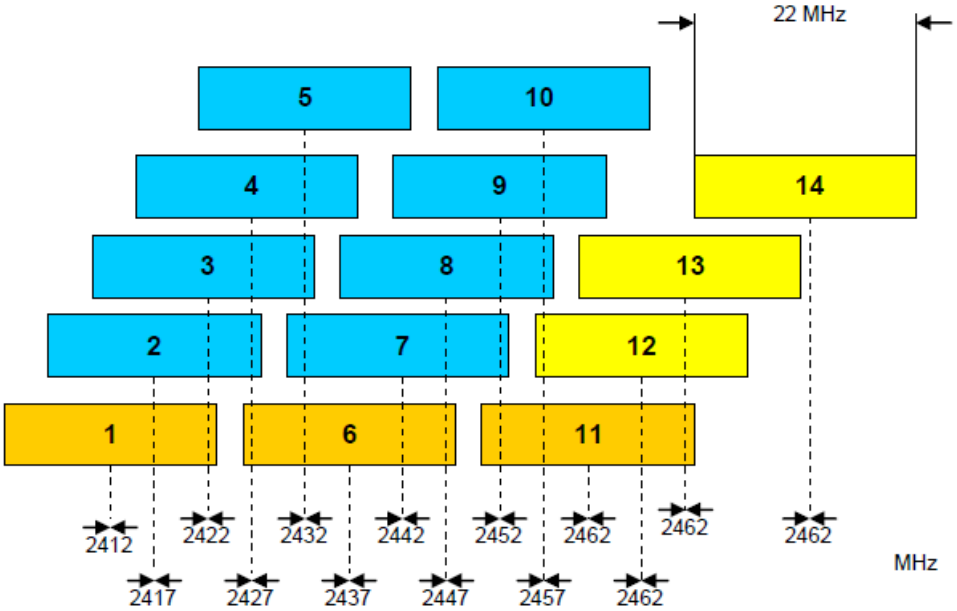
Open
Open Authentication is when the access point simply accepts the wireless devices identify without verifying or proving it. The benefits to this is simplicity and compatibility (all devices can do it).

Shared Key
Shared Key is when the wireless devices must present the proper key to get on the network. Although Shared Key has more security than Open Authentication it should not be considered secure. One of the benefits of Shared Key Authentication is simplicity.

Channels

The XBee® Wi-Fi modules operate in the 2.412-2.484 MHz range. The frequency range is broken down into 13 channels. Data is transmitted on a channel by radio frequencies over a certain frequency range. In order to avoid bad performance caused by the overlapping (“collision”) of channel frequencies in a wireless LAN environment, it is very important that the channels of neighboring access points are selected accordingly.

The center frequencies of the 14 possible channels range from 2.412 GHz to 2.484 GHz, with each channel being 22 MHz wide and centered in 5 MHz intervals. This means that only 3 channels (1, 6, and 11) in North America are not subject to overlapping.



4. XBee IP Services

The XBee provides services using IP (Internet Protocol) for XBee and other clients on the network. IP services provide functionality to allow XBee configuration and direct serial port access. There are two XBee services:

- XBee Application Service
- Serial Communication Service

XBee Application Service

This service primarily provides for XBee configuration. It also provides API compatibility for customers who have designed around other XBees. It uses UDP to transfer packets to and from port number 0xBEE. Packets are optionally acknowledged by the service but retries are not available. An extra header is added to the packet data to define commands for configuration and serial data transfer. The following sections describe how this service can be accessed from a local host or network client. C0 and DE are used to configure source and destination ports for the serial communication service. The XBee application service uses hard coded port 0xBEE for both source and destination and there is no option to configure another port.

Note: Do not configure C0 and/or DE to 0xBEE to use the XBee application service. Doing so will cause an error (AI=42), and the transceiver will neither send nor receive data.

Local Host

From a local host this functionality is accessed through XBee API frames. There are remote AT command frames as well as transmission frames. The API frames are listed as follows:

- TX request: 64-bit (TX64)
- RX indicator: 64-bit (RX64) (This frame is generated by the XBee module.)
- Remote AT command
- General Purpose Memory command

TX64 and RX64 API Frames

The intent of the XBee transmit and receive 64-bit API frames is to provide a standardized set of API frames to use for a point to multipoint network—a closed network of XBee Wi-Fi modules. The format of these frames has been standardized to work with other XBee products, such as the API frames of the 802.15.4 module. Please note that the XBee Wi-Fi module cannot communicate with an XBee 802.15.4 module.

Transmitting Data
The local host uses the TX64 frame to send data to another XBee using this service. When the frame is received through the serial port the XBee converts the contents of the frame to a serial data transfer command as defined by the XBee application service.

Receiving Data
A received Serial data transfer command will go to the serial port. The mode of the serial port will determine the format of the data. When in API mode the data will be sent to the host using the RX 64-bit frame.

Note: It is not recommended to use this service to send data to a network client. Use the serial communication service.

Remote AT Command Configuration

The Remote AT command frame is used to change configuration on a remote XBee. See Remote AT command frame in the API Operation chapter for more information.

Firmware Upgrades

Firmware upgrades from the local host can be done by sending ZigBee explicit API frames (type 0x11) to the IP address of the desired node with cluster ID 0x23. The format of the explicit frames is given in Chapter 7 and the sequence of operations to follow for firmware upgrades is given in Chapter 6.

Network Client

This port is accessed by sending a packet from the client using the UDP protocol on port 0xBEE. Data sent to this port must have an additional header preceding the data. The header description follows:

Field Name	Offset	Field Length	Description
Number1	0	2	Can be any random number
Number2	2	2	Number1 ^ 0x4242 (Exclusive OR of Number1 and constant 0x4242)
PacketID	4	1	Reserved for later use (0 for now)
EncPad	5	1	Reserved for later use (0 for now)
Command ID	6	1	0x00 = Data 0x02 = Remote Command 0x03 = General Purpose Memory Command 0x04 = I/O Sample 0x80 = Data Acknowledgement 0x82 = Response to remote command 0x83 = Response to General Purpose Memory Command
Command options	7	1	bit 0 – encrypted if set (Reserved for later use) bit 1 – set to request an ACK bits 2:7 - unused (Set to 0 for forward compatibility.)

All of the commands and command responses detailed below are preceded with the above application header.

Sending Configuration Commands

AT commands can be sent to the XBee Wi-Fi module from a network client. The following packet structure demonstrates how to query the SSID from a network client:

Packet Fields		Offset	Example	Description
Application Header	Number1	0	0x4242	
	Number2	2	0x0000	Number1 ^ Number2 = 0x4242
	Packet ID	4	0x00	Reserved for later use (0 for now)
	Encryption Pad	5	0x00	
	Command ID	6	0x02	Indicates Remote AT Command
	Command Options	7	0x00	Options are not available for this command
Command Specific Data	Frame ID	8	0x01	
	Configuration options	9	0x02	0 – Queue command parameter. Must send AC command or use apply changes option to apply changes. 2 – Apply changes to all changed commands
	AT Command	10	0x49 (I)	Command Name - Two ASCII characters that identify the AT command
		11	0x44(D)	
	Parameter Value	12		If present, indicates the requested parameter value to set the given command. If no characters present, command is queried.

The response will be sent back to the host with the following bytes.

Packet Fields		Offset	Example	Description
Application Header	Number1	0	0x4242	
	Number2	2	0x0000	Number1 ^ Number2 = 0x4242
	Packet ID	4	0x00	Reserved for later use (0 for now)
	Encryption Pad	5	0x00	
	Command ID	6	0x82	Indicates Remote AT Command Response
	Command Options	7	0x00	Options not available for this response
Command Specific Data	Frame ID	8	0x01	Copied from the command
	AT Command	9	0x49 (I)	Command Name - Two ASCII characters that identify the AT command
		10	0x44(D)	
	Status	11	0x00	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter
	Parameter Value	12	0x41 'A'	Data in binary or ASCII format, based on the command. For the ID command, the data is in ASCII format. If the command was set, then this field is not returned.
		13	0x63 'c'	
		14	0x63 'c'	
		15	0x65 'e'	
		16	0x73 's'	
		17	0x73 's'	
		18	0x50 'p'	
		19	0x6F 'o'	
		20	0x69 'i'	
		21	0x6E 'n'	
		22	0x74 't'	

Sending Serial Data Command to XBee

Using this service to send data out the serial port is not required. Most users will choose to use the Serial Communication Service (see below) for sending data from a network client. One reason to use the XBee Application Service to send the serial data command from a network client is to receive an acknowledgment when sending a UDP packet.

The client can request an acknowledgement from the XBee but must wait to receive the acknowledgement before sending the next packet. The client is responsible for retransmissions due to missed acknowledgments. When resending packets, duplicates can be received at the destination due to a successful serial data command and a failed acknowledgment packet. The host in this case must be able to handle duplicate packets. The following packet structures are examples of sending data and receiving an acknowledgement using the XBee application service:

Serial Data Command:

Packet Fields		Offset	Example	Description
Application Header	Number1	0	0x4242	
	Number2	2	0x0000	Number1 ^ Number2 = 0x4242
	Packet ID	4	0x00	Reserved for later use (0 for now)
	Encryption Pad	5	0x00	
	Command ID	6	0x00	Indicates Transmission data
	Command Options	7	0x02	Request acknowledgment
Command Specific Data	Serial Data	8	0x48 'H'	Can be up to 1492 bytes. Data will be sent out the XBee's serial port.
		9	0x65 'e'	
		10	0x6C 'l'	
		11	0x6C 'l'	
		12	0x6F 'o'	

Serial Data command acknowledgment if requested:

Packet Fields		Offset	Example	Description
Application Header	Number1	0	0x4242	
	Number2	2	0x0000	Number1 ^ Number2 = 0x4242
	Packet ID	4	0x00	Reserved for later use (0 for now)
	Encryption Pad	5	0x00	
	Command ID	6	0x80	Indicates Transmission data
	Command Options	7	0x0	Options not available for this response
Command Specific Data	Serial Data	8		No command specific data

Receiving I/O sampled data

Sample data generated by the module will be sent to the address configured by the DL commands. This data can be sent to another XBee or to a network client. It will be sent using UDP from the 0xBEE port as with other XBee Application services. Sample data will be received by the client as follows:

Frame Fields		Offset	Example	Description
Application Header	Number1	0	0x4242	
	Number2	2	0x0000	Number1 ^ Number2 = 0x4242
	Packet ID	4	0x00	Reserved for later use (0 for now)
	Encryption Pad	5	0x00	
	Command ID	6	0x04	Indicates I/O Sample data
	Command Options	7	0x00	Options not available for this response
Command Specific Data	Number Samples	8	0x01	Indicates one sample set
	Digital Mask	MSB 9	0x01	Bit Mask. Each bit represents an enabled DIO line starting with DIO0 at bit 0.
		LSB 10	0x01	
	Analog Mask	11	0x02	Bit Mask. Each bit represents an enabled ADC starting with ADC0 at bit 0. This selects ADC1 for analog sampling.
	Digital Sample	MSB 12	0x00	This field is only present if at least one DIO line is enabled in the digital mask specified above. Each bit represents a DIO line. Start with bit 0 for DIO0.
		LSB 13	0x01	
	Analog Sample	MSB 14	0x02	0x200 indicates that reading is half of VREF. For a default VREF of 2.5V, 0x200 represents 1.25 volts on ADC1 in this example.
		LSB 15	0x00	

Sending Over-the-Air Firmware Upgrades

A network client can also use the XBee IP services to send a firmware upgrade to the module. This is done by sending a frame formatted with an application header, followed by a GPM header, following by GPM data. The format of the application header is given above. The format of the various GPM headers is given in chapter 6, but each of those GPM headers need to be preceded by an application header. The following frame shows an example of the final step of a firmware upgrade process:

Packet Fields		Offset	Example	Description
Application Header	Number1	0	0x4242	This is an easy number to create an accepted frame.
	Number2	2	0x0000	Number1 ^ Number2 = 0x4242 (This is an easy way to send a frame that software won't reject.)
	Packet ID	4	0x00	Reserved for later use (0 for now)
	EncPad	5	0x00	
	Command ID	6	0x00	Indicates Transmission data
	Command Options	7	0x02	Request acknowledgment
Command Specific Data	GPM_CMD_ID	8	0x06	Firmware verify and install command
	GPM_OPTIONS	9	0x00	Reserved for later use (0 for now)
	GPM_BLOCK_NUM	10	0x00	
	GPM_START_INDEX	12	0x00	
	GPM_NUM_BYTES	14	0x0000	
	GPM_DATA	16		This field is unused for this command

Serial Communication Service

The serial communication service connects an IP port to the serial peripheral (UART or SPI) of the XBee. No additional formatting or header is required and data will be transferred between the RF hardware and Serial Communication hardware as received. The IP ports are configured using the C0 and DE commands. Note that port 0xBEE is reserved for the XBee Application Service and should not be used for the Serial Communication Service. The behavior of this service varies based on the mode of the serial port and is discussed in the following sections.

Transparent Mode

Only one port is available and can be either UDP or TCP. It is configured through the IP command. Data received by the service is sent to the serial port without any additional processing.

UDP

When the IP command is configured for UDP, data received on the serial port will be packetized and sent to the IP address specified by the DL command and to the destination port specified by the DE command. The source port is defined by the C0 command.

TCP

TCP provides for a connection based protocol. When in transparent mode the module will only allow one connection at a time. A connection can be initiated by a local host or by a network client.

A local host initiates a connection by sending data to the serial port. A connection will be created based on the DL (IP address) and DE (destination port) commands. However, if DL is a broadcast address, then UDP will be used, ignoring the TCP configuration.

A network client establishing a TCP connection to the XBee will use the port defined by the C0 command. When established any data sent by the local host will not create a new connection based on DL and DE, but rather the existing connection will be utilized.

API Mode

In API mode, the module will listen on both the UDP and the TCP ports at the same time. The local host will utilize the IPv4 transmit frame to send data from the module and will receive data through the IPv4 received frame. These frames give greater IP control and visibility to the local host. See the API section for more information.

UDP packets are sent from the listening port on the sending module. If the listening port number doesn't match the source port of the sender, the packet is rejected and not sent. Also, if the specified IP address is a broadcast, it will be sent as a UDP packet, whether or not TCP is specified in the API frame.

TCP packets may be sent on an existing connection or on a new connection. In order to send data on an existing TCP connection, the destination IP address and port in the API packet must match the remote IP address and port in an existing socket. In a sample application, a packet may arrive that expects return data on the same socket. The API frame (Rx IPv4) will contain the remote IP address and port. While the remote IP address may be predicted, the remote IP address cannot. Therefore, the return data should be sent to the remote IP address and port by swapping the source and destination port numbers.

If the destination IP address and port don't match an existing connection, the frame will be discarded, unless the source port is 0. A source port of 0 allows the module to create a new TCP client socket if the requested socket connection doesn't already exist, and if a socket resource is available.

5. Sleep

The XBee Wi-Fi module supports two different sleep modes.

- Pin Sleep
- Cyclic Sleep

In addition the sleep mode current draw can be modified with the following sleep options.

- AP Associated Sleep
- Deep Sleep

Pin sleep allows an external microcontroller to determine when the XBee should sleep and when it should wake by using either the SleepRq pin (default) or the SPI_nSSEL pin. In contrast, cyclic sleep allows the sleep period and wake times to be configured through the use of AT commands. The module can stay associated to the access point or can enter a deeper sleep and associate to the access point for each sleep/wake occurrence. The sleep mode is configurable with the SM and SO commands.

Besides the four sleep modes mentioned above, each of them operate a little differently based on the serial interface (UART or SPI).

Using Sleep Mode: UART

When the serial interface is UART, the On/nSleep pin is used to indicate that the module is entering sleep mode, unless pin 13 is configured for a different usage. (See command reference table) If D9 is configured for On/nSleep, then it is driven low when asleep and high when awake, whether using pin sleep or cyclic sleep.

If CTS hardware flow control is enabled (D7 command), the CTS pin (pin 12) is de-asserted (high) when entering sleep to indicate that serial data should not be sent to the module. The module will not respond to serial or RF data when it is sleeping. Applications that utilize the UART are encouraged to observe CTS flow control in any of the sleep modes. When the XBee wakes from sleep with flow control enabled, the CTS pin is asserted (low).

If using pin sleep, D8 (mapped to XBee pin 9) must be configured for SleepRq (See command reference table) to put the module to sleep. Otherwise, there is no sleep at all, meaning the module will always stay awake in full power mode. When D8 is configured for SleepRq, the host should drive pin 9 high to put the module to sleep, and the host should drive pin 9 low to wake up the module.

Using Sleep Mode: SPI

When the serial interface is SPI, pin 19 is used as an attention indicator to tell the SPI master when it has data to send. Since SPI only operates in API mode, it will assert SPI/nATTN and send out a modem status indicator after initialization. The host can use this to know when the radio is ready to operate as a SPI slave. Since the function of pin

19 is to indicate when the XBee has data to send to the host, it may legitimately be driven high or low while the module is awake.

When using the SPI, either SleepRq or SPI_nSSEL may be used for pin sleep. If D8 is configured as a peripheral (1), then it will be used for pin sleep. If not, and SPI_nSSEL is configured as a peripheral (which it must be to enable SPI operation), then SPI_nSSEL is used for pin sleep.

Using SPI_nSSEL for pin sleep has the advantage of requiring one less physical pin connection to implement pin sleep on SPI. It has the disadvantage of putting the radio to sleep whenever the SPI master negates SPI_nSSEL, even if that wasn't the intent. Therefore, if the user can control SPI_nSSEL, whether or not data needs to be transmitted, then sharing the pin may be a good option. It makes the SleepRq pin available for another purpose, or it simply requires one less pin to the SPI interface.

Sleep Options

AP Associated Sleep

This option allows the module to sync up with beacons sent from the AP which contains the DTIM (Delivery Traffic Indication Message). The DTIM indicates when broadcast and multicast data will be sent on the network. This property is configured on the AP and is typically configured as the number of beacons between each beacon with DTIM. The current draw in associated sleep mode varies significantly. When the module is awake it draws approximately 100 mA. When it is asleep, it draws approximately 2 mA. Total current draw increases when the DTIM rate is higher and it decreases when the DTIM rate is lower on the access point.

The sleep modes are described as follows with this option enabled.

Pin Sleep Mode

The module remains associated to the AP and will wake based on the period of the DTIM. This wake period will not be seen by the local host unless data has been sent to the module. In this case the module will ‘wake’ by asserting the appropriate I/O lines. The local UART host is then required to de-assert the sleep pin (either SleepRq SPI_nSSEL) to awaken the module. Once the sleep pin is de-asserted the module will then send the data to the host.

SPI operation is similar except that the radio asserts nATTN when data becomes available and then the local host is expected to assert SPI_nSSEL and to provide a clock until the data available is sent out.

When the local UART host needs to send data it de-asserts SleepRq. Once the appropriate status I/O lines are asserted (CTS and/or On/nSleep) the module is ready to accept data. However data will be queued and not sent until the next DTIM.

When the local SPI host needs to send data it asserts SPI_nSSEL. If SPI_nSSEL is being used for pin sleep, asserting SPI_nSSEL is enough to awaken the module to receive the

incoming data. But, if SleepRq is being used to control sleep, then SPI_nSSEL must be asserted and SleepRq must be de-asserted to awaken the module to receive the data. This wakes up the module, which will then accept the incoming data. However data will be queued and not sent until the next DTIM.

Cyclic Sleep Mode

The module remains associated to the AP and will sleep based on the SP parameter. After SP expires, the module will awaken for 30 milliseconds to check for data from the AP and to allow the host to send data or commands. This time is factored in as part of the overall ST time. When data is received or sent within 30 ms, the module will remain awake for ST time and any further activity will not restart this time. When no data is received or sent within 30 ms, the module will resume sleeping immediately, without waiting for ST time-out.

Deep Sleep (Non-Associated Sleep)

This option allows the Wi-Fi circuitry to be powered down resulting in the lowest sleep current (about 6 µA) but at the expense of losing packets received during the time the module is asleep. This is because the access point will behave like the module is in full power mode while it is asleep and it will not hold back packets until the module wakes up.

Pin Sleep Mode

In this mode when SleepRq is asserted the module will power down the Wi-Fi circuitry. When SleepRq is de-asserted the Wi-Fi circuitry is powered up. This causes the module to associate to the access point for each wake event. If the module was associated when it went to sleep, it should be ready to transmit data as soon as the module indicates that it is awake, which occurs within one millisecond of the pin wake up. If not, a new association could take much longer to complete, especially if DHCP is used.

Cyclic Sleep Mode

In this mode the module will enter and exit sleep based on the SP, ST, and SA commands. SP specifies the sleep time and ST specifies the wake time of the module after it is associated. SA specifies the maximum time to wait for association before starting the ST timer. If SA expires before the association process completes, then the module will sleep anyway. When it awakens from this state, then it will start the SA timer again to seek to establish association.

Under normal conditions, SA is used for a time out for the first association following reset and ST is used for short wake cycles thereafter. To conserve battery power, SA should be long enough for association and ST should be as short as possible for the application.

Sampling Data Using Sleep Modes

Data can be sampled when waking from any sleep mode by enabling an ADC or digital input and setting IR appropriately with respect to ST to obtain the desired number of samples.

Sample Rate (ATIR)

If multiple samples are wanted during the wake period then IR can be used. This will provide ST/IR+1 samples. Each sample will be sent separately.

Wake Host

Wake host parameter (ATWH) delays UART and sample data from being initiated until the timer has expired. This allows the host to wake up before receiving data or a sensor to power up before an I/O sample is taken.

Digital outputs and special function outputs such as ON_SLEEP and CTS are not affected by WH. This is to allow these signals to be used to wake up devices.

Note that for deep sleep, both WH must be expired and the module must be associated before I/O samples are taken.

6. Advanced Application Features

XBee Analog and Digital I/O Lines

XBee Wi-Fi firmware supports a number of analog and digital I/O pins that are configured through software commands. Analog and digital I/O lines can be set or queried. The following table lists the configurable I/O pins and the corresponding configuration commands.

Pin name(s)	Module pin	AT cmd	Command Range
DIO12/SPI_MISO	4	P2	0,1,3-5
DIO10/PWM0	6	P0	0,2-5
DIO11/PWM1	7	P1	0,2-5
DIO8/nDTR/SLEEP_RQ	9	D8	0,1,3-5
DIO4/SPI_MOSI	11	D4	0-5
DIO7/nCTS	12	D7	0,1,3-7
DIO9/ON_nSLEEP	13	D9	0,1,3-5
DIO5/ASSOCIATE	15	D5	0,1,3-5
DIO6/nRTS	16	D6	0,1,3-5
DIO3/AD3/SPI_nSSEL	17	D3	0-5
DIO2/AD2/SPI_CLK	18	D2	0-5
DIO1/AD1/SPI_nATTN	19	D1	0-5
DIO0/AD0	20	D0	0,2-5

I/O Configuration

To enable an analog or digital I/O function on one or more XBee module pin(s), the appropriate configuration command must be issued with the correct parameter. After issuing the configuration command, changes must be applied on the module for the I/O settings to take effect. Pull-up/down resistors can be set for each digital input line using the PR command. The PR value updates the state of all pull-up/down resistors, and the PD command determines if a pull-up or pull-down is used. See Chapter 8 for information on these commands

Pin Command Parameter	Description
0	Disabled
1	Peripheral control
2	Analog input or PWM output
3	Data in monitored
4	Data out default low
5	Data out default High
6	RS485 enable low
7	RS485 enable high
>7	Unsupported

I/O Sampling

The XBee modules have the ability to monitor and sample the analog and digital I/O lines. I/O samples can be read locally or transmitted to a remote device to provide indication of the current I/O line states.

There are three ways to obtain I/O samples, either locally or remotely:

- Queried Sampling
- Periodic Sampling
- Change Detection Sampling.

IO sample data is formatted as shown in the table below

Bytes	Name	Description
1	Sample Sets	Number of sample sets in the packet. (Always set to 1.)
2	Digital Channel mask	Digital IO line on the module. • bit 0 = DIO0 • bit 1 = DIO1 • bit 2 = DIO2 • bit 3 = DIO3 • bit 4 = DIO4 • bit 5 = DIO5 • bit 6 = DIO6 • bit 7 = DIO7 • bit 8 = DIO8 • bit 9 = DIO9 • bit 10 = DIO10 • bit 11 = DIO11 • bit 12 = DIO12 For example, a digital channel mask of 0x002F means DIO0 1, 2, 3, and 5 are enabled as digital IO.
1	Analog Channel Mask	Indicates which lines have analog inputs enabled for sampling. Each bit in the analog channel mask corresponds to one analog input channel. • bit 0 = AD0 • bit 1 = AD1 • bit 2 = AD2 • bit 3 = AD3 • bit 4 = AD4
Variable	Sampled Data Set	If any digital IO lines are enabled, the first two bytes of the data set indicate the state of all enabled digital IO. Only digital channels that are enabled in the Digital Channel Mask bytes have any meaning in the sample set. If no digital IO is enabled on the device, these 2 bytes will be omitted. Following the digital IO data (if any), each enabled analog channel will return 2 bytes. The data starts with AD0 and continues sequentially for each enabled analog input channel up to AD4.

The sampled data set will include 2 bytes of digital I/O data only if one or more I/O lines on the device are configured as digital I/O. If no pins are configured as digital I/O, these 2 bytes will be omitted.

The digital I/O data is only relevant if the same bit is enabled in the digital I/O mask. Analog samples are 10 bit values and aligned on a 16 bit boundary. The analog reading is scaled such that 0x0000 represents 0V, and 0x3FF = VREF. VREF may be either 1.25 volts or 2.5 volts based on the setting of the AV command, where 2.5 volts is the

default. The analog inputs on the module are capped at 0x3FF. Analog samples are returned in order starting with AD0 and finishing with AD4. Only enabled analog input channels return data as shown in the example below.

To convert the A/D reading to mV, do the following:

AD (mV) = (A/D reading (converted to decimal) * VREF) / 1023 where VREF may be 1250 or 2500

Assuming that AV is set to the default value, the reading in the sample frame represents voltage inputs of 2385.14 mV (0x3D0) and 713.59 mV (0x124) for AD0 and AD1 respectively.

Queried Sampling

The IS command can be sent to a device locally, or to a remote device using the API remote command frame (see chapter 8 for details). When the IS command is sent and at least one I/O line is enabled as an input or an output, the receiving device samples all enabled digital I/O and analog input channels and returns an I/O sample. When no I/O line is enabled, IS will return nothing. If IS is sent locally, the I/O sample is sent out the UART or SPI port. If the IS command was received as a remote command, the I/O sample is sent over-the-air to the device that sent the IS command.

If the IS command is issued in command mode, the module returns a carriage return-delimited list containing the above-listed fields. If the IS command is issued in API mode, the module returns an API command response packet with the I/O data included in the command data portion of the response frame.

The following table shows an example of the fields in an IS response.

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

Periodic I/O Sampling

Periodic sampling allows the XBee module to take an I/O sample and transmit it to a remote device at a periodic rate. The periodic sample rate is set by the IR command. If IR is set to 0 or there are no active I/O lines, 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. However, the module cannot keep up with transmitting an I/O sample more often than every three milliseconds. Therefore, when IR is set to 1 or 2, many samples are lost. The DL command determines the destination address of the I/O samples. DL can be set to transmit to a network client or another XBee Wi-Fi module. Only modules with API mode enabled for the serial port can send I/O data samples out

their serial port. Network clients will receive the I/O data packet as described in the XBee IP Services chapter.

IR can be used with sleep. A module will transmit periodic I/O samples at the IR rate until the ST timer expires, the SleepRq line is asserted, and the device can resume sleeping. Even if the IR rate is set longer than the ST defined wake time, at least one I/O sample will still be sent before the module returns to sleep because it sends one immediately upon wake up. If it is not desired that a sample is sent every wake cycle, the IF command can be used to configure how many wake cycles should elapse before sending I/O samples at the IR rate.

Change Detection Sampling

Modules can be configured to transmit a data sample immediately whenever a monitored digital I/O pin changes state. The IC command is a bitmask that can be used to set which digital I/O lines should be monitored for a state change. If one or more bits in IC is set, an I/O sample will be transmitted as soon as a state change is observed in one of the monitored digital IO lines. Change detection samples are transmitted to the IPv4 address specified by DL.

I/O Examples

Example 1: Configure the following I/O settings on the XBee
Configure DIO1/AD1 as a digital input with pull-up resistor enabled

Configure DIO2/AD2 as an analog input

Configure DIO4 as a digital output, driving high.

To configure DIO1/AD1 as an input, issue the ATD1 command with a parameter of 3 ("ATD13"). To enable pull-up resistors on the same pin, the PR command should be issued with bit 3 set (e.g. ATPR8, ATPR1FFF, etc.). The ATD2 command should be issued with a parameter of 2 to enable the analog input ("ATD22"). Finally, DIO4 can be set as an output, driving high by issuing the ATD4 command with a parameter value of 5 ("ATD45").
After issuing these commands, changes must be applied before the module I/O pins will be updated to the new states. The AC or CN commands can be issued to apply changes (e.g. ATAC).

General Purpose Flash Memory

The XBee Wi-Fi RF modules provide 160 4096-byte blocks of flash memory which can be read and written by the user application. This memory provides a non-volatile data storage area which can be used for a multitude of purposes. Some common uses of this data storage include: storing logged sensor data, buffering firmware upgrade data for a host microcontroller, or storing and retrieving data tables needed for calculations performed by a host microcontroller. The General Purpose Memory (GPM) is also used to store a firmware upgrade file for over-the-air firmware upgrades of the XBee module itself.

Accessing General Purpose Flash Memory

The GPM of a target node can be accessed from the XBee serial port or from a non-XBee network client.

Serial port access is done by sending explicit API frames to the MEMORY_ACCESS cluster ID (0x23) on the DIGI_DEVICE endpoint (0xE6) of the target node. (Explicit API frames have frame identifier 0x11 and are described in the API Operation section.)

Access from a non-XBee network client is done by sending UDP frames to the target node on port 0x0BEE. The payload begins with an application header followed by the GPM header described below. (Refer to the Network Client section of the XBee Application Service section to learn how to format the application header.)

The following header is used to generate a GPM command. It should be used whether using serial port access or network client access. For network client access, an application header needs to precede the GPM header. To keep things simple, this section is written from the perspective of serial port access, without the application header. Do not forget to precede each frame with an application header if using a network client for GPM access.

Byte Offset in Payload	Number of Bytes	Field Name	General Field Description
0	1	GPM_CMD_ID	Specific GPM commands are described below
1	1	GPM_OPTIONS	Command-specific option
2	2*	GPM_BLOCK_NUM	The block number addressed in the GPM
4	2*	GPM_START_INDEX	The byte index within the addressed GPM block
6	2*	GPM_NUM_BYTES	Then number of bytes in the GPM_DATA field, or in the case of a READ, the number of bytes requested
8	Varies	GPM_DATA	

*Multi-byte parameters should be specified with big-endian byte ordering.

When a GPM command is sent to a radio via a unicast the receiving radio will unicast a response back to the requesting radio's source endpoint specified in the request packet. No response is sent for broadcast requests. If the source endpoint is set to the DIGI_DEVICE endpoint (0xE6) or explicit API mode is enabled on the requesting radio then a GPM response will be output as an explicit API RX indicator frame on the requesting node (assuming API mode is enabled.) The format of the response is very similar to the request packet:

Byte Offset in Payload	Number of Bytes	Field Name	General Field Description
0	1	GPM_CMD_ID	This field will be the same as the request field
1	1	GPM_STATUS	Status indicating whether the command was successful
2	2*	GPM_BLOCK_NUM	The block number addressed in the GPM
4	2*	GPM_START_INDEX	The byte index within the addressed GPM block
6	2*	GPM_NUM_BYTES	Then number of bytes in the GPM_DATA field
8	Varies	GPM_DATA	

*Multi-byte parameters should be specified with big-endian byte ordering.

The following commands exist for interacting with GPM:

PLATFORM_INFO_REQUEST (0x00):
A PLATFORM_INFO_REQUEST frame can be sent to query details of the GPM structure.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to PLATFORM_INFO_REQUEST (0x00).
GPM_OPTIONS	This field is unused for this command. Set to 0.
GPM_BLOCK_NUM	
GPM_START_INDEX	
GPM_NUM_BYTES	
GPM_DATA	No data bytes should be specified for this command.

PLATFORM_INFO (0x80):
When a PLATFORM_INFO_REQUEST command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to PLATFORM_INFO (0x80).
GPM_OPTIONS	A 1 in the least significant bit indicates an error occurred. All other bits are reserved at this time.
GPM_BLOCK_NUM	Indicates the number of GPM blocks available.
GPM_START_INDEX	Indicates the size of a GPM block in bytes.
GPM_NUM_BYTES	The number of bytes in the GPM_DATA field. For this command, this field will be set to 0.
GPM_DATA	No data bytes should be specified for this command.

Example:

A **PLATFORM_INFO_REQUEST** sent to a radio with a serial number of 0x0013a200407402AC should be formatted as follows (spaces added to delineate fields):

7E 001C 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 00 00 00 0000 0000 0000 24

Assuming all transmissions were successful, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76
7E 001A 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 80 00 0077 0200 0000 EB

ERASE (0x01):
The ERASE command erases (writes all bits to binary 1) one or all of the GPM flash blocks. The ERASE command can also be used to erase all blocks of the GPM by setting the GPM_NUM_BYTES field to 0.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to ERASE (0x01).
GPM_OPTIONS	There are currently no options defined for the ERASE command. Set this field to 0.
GPM_BLOCK_NUM	Set to the index of the GPM block that should be erased. When erasing all GPM blocks, this field is ignored (set to 0).
GPM_START_INDEX	The ERASE command only works on complete GPM blocks. The command cannot be used to erase part of a GPM block. For this reason, GPM_START_INDEX is unused (set to 0).
GPM_NUM_BYTES	Setting GPM_NUM_BYTES to 0 has a special meaning. It indicates that every flash block in the GPM should be erased (not just the one specified with GPM_BLOCK_NUM). In all other cases, the GPM_NUM_BYTES field should be set to the GPM flash block size.
GPM_DATA	No data bytes should be specified for this command.

ERASE_RESPONSE (0x81):
When an ERASE command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to ERASE_RESPONSE (0x81).
GPM_STATUS	A 1 in the least significant bit indicates an error occurred. All other bits are reserved at this time.
GPM_BLOCK_NUM	Matches the parameter passed in the request frame.
GPM_START_INDEX	
GPM_NUM_BYTES	The number of bytes in the GPM_DATA field. For this command, this field will be set to 0.
GPM_DATA	No data bytes should be specified for this command.

Example:

To erase flash block 42 of a target radio with serial number of 0x0013a200407402ac, an ERASE packet should be formatted as follows (spaces added to delineate fields):

7E 001C 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 C0 01 00 002A 0000 0200 37

Assuming all transmissions were successful, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76

7E 001A 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 81 00 002A 0000 0000 39

WRITE (0x02) and **ERASE_THEN_WRITE (0x03)**:
The WRITE command writes the specified bytes to the GPM location specified. Before writing bytes to a GPM block it is important that the bytes have been erased previously. The ERASE_THEN_WRITE command performs an ERASE of the entire GPM block specified with the GPM_BLOCK_NUM field prior to doing a WRITE.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to WRITE (0x02) or ERASE_THEN_WRITE (0x03).
GPM_OPTIONS	There are currently no options defined for the ERASE command. Set this field to 0.
GPM_BLOCK_NUM	Set to the index of the GPM block that should be written.
GPM_START_INDEX	Set to the byte index within the GPM block where the given data should be written.
GPM_NUM_BYTES	Set to the number of bytes specified in the GPM_DATA field. Only one GPM block can be operated on per command. For this reason, GPM_START_INDEX + GPM_NUM_BYTES cannot be greater than the GPM block size. It is also important to remember that the number of bytes sent in an explicit API frame (including the GPM command fields) cannot exceed the maximum payload size of the radio. The maximum payload size can be queried with the NP AT command.
GPM_DATA	The data to be written.

WRITE _RESPONSE (0x82) and ERASE**_THEN_WRITE_RESPONSE(0x83)**:
When a WRITE or ERASE_THEN_WRITE command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to WRITE_RESPONSE (0x82) or ERASE_THEN_WRITE_RESPONSE (0x83).
GPM_STATUS	A 1 in the least significant bit indicates an error occurred. All other bits are reserved at this time.
GPM_BLOCK_NUM	Matches the parameter passed in the request frame.
GPM_START_INDEX	
GPM_NUM_BYTES	The number of bytes in the GPM_DATA field. For this command, this field will be set to 0.
GPM_DATA	No data bytes should be specified for this command.

Example:
To write 15 bytes of incrementing data to flash block 22 of a target radio with serial number of 0x0013a200407402ac, a WRITE packet should be formatted as follows (spaces added to delineate fields):

7E 002B 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 C0 02 00 0016 0000 000F
0102030405060708090A0B0C0D0E0F C5

Assuming all transmissions were successful and that flash block 22 was previously erased, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00
76 7E 001A 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 82 00 0016 0000 0000 4C

READ (0x04):
The READ command can be used to read the specified number of bytes from the GPM location specified. Data can be queried from only one GPM block per command.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to READ (0x04).
GPM_OPTIONS	There are currently no options defined for the ERASE command. Set this field to 0.
GPM_BLOCK_NUM	Set to the index of the GPM block that should be read.
GPM_START_INDEX	Set to the byte index within the GPM block where the given data should be read.
GPM_NUM_BYTES	Set to the number of data bytes to be read. Only one GPM block can be operated on per command. For this reason, GPM_START_INDEX + GPM_NUM_BYTES cannot be greater than the GPM block size. It is also important to remember that the number of bytes sent in an explicit API frame (including the GPM command fields) cannot exceed the maximum payload size of the radio. The maximum payload size can be queried with the NP AT command.
GPM_DATA	No data bytes should be specified for this command.

READ _RESPONSE (0x84):
When a READ command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to READ_RESPONSE (0x84).
GPM_STATUS	A 1 in the least significant bit indicates an error occurred. All other bits are reserved at this time.
GPM_BLOCK_NUM	Matches the parameter passed in the request frame.
GPM_START_INDEX	
GPM_NUM_BYTES	The number of bytes in the GPM_DATA field.
GPM_DATA	The bytes read from the GPM block specified.

Example:
To read 15 bytes of previously written data from flash block 22 of a target radio with serial number of 0x0013a200407402ac, a READ packet should be formatted as follows (spaces added to delineate fields):

7E 001C 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 C0 04 00 0016 0000 000F 3B

Assuming all transmissions were successful and that flash block 22 was previously written with incrementing data, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76
7E 0029 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 84 00 0016 0000 000F
0102030405060708090A0B0C0D0E0F C3

FIRMWARE_VERIFY (0x05) and **FIRMWARE_VERIFY_AND_INSTALL(0x06):**
The FIRMWARE_VERIFY and FIRMWARE_VERIFY_AND_INSTALL commands are used when remotely updating firmware on a module. Remote firmware upgrades are covered in detail in the next section. These commands check if the General Purpose Memory contains a valid over-the-air update file. For the FIRMWARE_VERIFY_AND_INSTALL command, if the GPM contains a valid firmware image then the module will reset and begin using the new firmware.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to FIRMWARE_VERIFY (0x05) or FIRMWARE_VERIFY_AND_INSTALL (0x06).
GPM_OPTIONS	There are currently no options defined for the ERASE command. Set this field to 0.
GPM_BLOCK_NUM	These fields are unused for this command. Set to 0.
GPM_START_INDEX	
GPM_NUM_BYTES	
GPM_DATA	This field is unused for this command.

FIRMWARE_VERIFY _RESPONSE (0x85):
When a FIRMWARE_VERIFY command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to FIRMWARE_VERIFY_RESPONSE (0x85).
GPM_OPTIONS	A 1 in the least significant bit indicates the GPM does not contain a valid firmware image. A 0 in the least significant bit indicates the GPM does contain a valid firmware image. All other bits are reserved at this time.
GPM_BLOCK_NUM	These fields are unused for this command. Set to 0.
GPM_START_INDEX	
GPM_NUM_BYTES	
GPM_DATA	This field is unused for this command.

FIRMWARE_VERIFY _AND_INSTALL_RESPONSE (0x86):
When a FIRMWARE_VERIFY_AND_INSTALL command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame only if the GPM memory does not contain a valid image. If the image is valid, the module will reset and begin using the new firmware.

Field Name	Command –Specific Description
GPM_CMD_ID	Should be set to FIRMWARE_VERIFY_AND_INSTALL_RESPONSE (0x86).
GPM_OPTIONS	A 1 in the least significant bit indicates the GPM does not contain a valid firmware image. All other bits are reserved at this time.
GPM_BLOCK_NUM	These fields are unused for this command. Set to 0.
GPM_START_INDEX	
GPM_NUM_BYTES	
GPM_DATA	This field is unused for this command.

Example:
To verify a firmware image previously loaded into the GPM on a target radio with serial number of 0x0013a200407402ac, a FIRMWARE_VERIFY packet should be formatted as follows (spaces added to delineate fields):

7E 001C 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 00 05 00 0000 0000 0000 1F

Assuming all transmissions were successful and that the firmware image previously loaded into the GPM is valid, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76
7E 001A 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 85 00 0000 0000 0000 5F

Working with Flash Memory

When working with the General Purpose Memory, the user should be aware of a number of limitations associated with working with flash memory:

- Flash memory write operations are only capable of changing binary 1's to binary 0's. Only the erase operation can change binary 0's to binary 1's. For this reason it is usually necessary to erase a flash block before performing a write operation.
- A flash memory block must be erased in its entirety when performing an erase operation. A block cannot be partially erased.
- Flash memory has a limited lifetime. The flash memory on which the GPM is based is rated at 20,000 erase cycles before failure. Care must be taken to ensure that the frequency of erase/ write operations allows for the desired product lifetime. Digi's warranty will not cover products whose number of erase cycles has been exceeded.
- Over-the-Air firmware upgrades (described in the next section) require the entire GPM be erased. Any user data stored in the GPM will be lost during an over-the-air upgrade.

Over-the-Air Firmware Upgrades

The XBee Wi-Fi RF modules provide two methods of updating the firmware on the module. Firmware can be updated locally via X-CTU (a free testing and configuration

utility provided by Digi) using the radio's serial port interface. Firmware can also be updated using the radios' RF interface (Over-the-Air Updating.)

The over-the-air firmware upgrading method provided is a robust and versatile technique which can be tailored to many different networks and applications. It has been engineered to be reliable and minimize disruption of normal network operations.

There are three phases of the over-the-air upgrade process: distributing the new application, verifying the new application, and installing the new application. In the following section the node which will be upgraded will be referred to as the target node. The node providing the update information will be referred to as the source node. In most applications the source node will be locally attached to a PC running update software.

Distributing the New Application

The first phase of performing an over-the-air upgrade on a module is transferring the new firmware file to the target node. The new firmware image should be loaded in the target node's GPM prior to installation XBee Wi-Fi RF modules use an encrypted binary (.ebin) file for both serial and over-the-air firmware upgrades. These firmware files are available on the Digi Support website.

The contents of the .ebin file should be sent to the target radio using general purpose memory WRITE commands. The entire GPM should be erased prior to beginning an upload of an .ebin file. The contents of the .ebin file should be stored in order in the appropriate GPM memory blocks. The number of bytes that are sent in an individual GPM WRITE frame is flexible and can be catered to the user application.

Example:

If the size of the .ebin file is 217,088 bytes, then it could be sent to the module in 1024 byte blocks as follows:

CPT_BLOCK_NUM	GPM_START_INDEX	GPM_NUM_BYTES	.ebin bytes
0	0	1024	0 to 1023
0	1024	1024	1024 to 2047
0	2048	1024	2048 to 3071
0	3072	1024	3071 to 4095
1	0	1024	4096 to 5119
1	1024	1024	5120 to 6143
-	-	-	-
-	-	-	-
-	-	-	-
52	1024		214,016 to 215,039
52	2048		215,040 to 216,063
52	3072		216,064 to 217,087

Verifying the New Application

For an uploaded application to function correctly every single byte from the .ebin file must be properly transferred to the GPM. To guarantee that this is the case GPM VERIFY functions exist to ensure that all bytes are properly in place. The FIRMWARE_VERIFY function reports whether or not the uploaded data is valid. The FIRMWARE_VERIFY_AND_INSTALL command will report if the uploaded data is invalid. If the data is valid it will begin installing the application. No installation will take place on invalid data.

Installing the Application

When the entire .ebin file has been uploaded to the GPM of the target node a FIRMWARE_VERIFY_AND_INSTALL command can be issued. Once the target receives the command it will verify the .ebin file loaded in the GPM. If it is found to be valid then the module will install the new firmware. This installation process can take up to 8 seconds. During the installation the module will be unresponsive to both serial and RF communication. To complete the installation the target module will reset. AT parameter settings which have not been written to flash (using the WR command) will be lost.

Things to Remember

- The firmware upgrade process requires that the module resets itself. Because of this reset parameters which have not been written to flash will be lost after the reset. To avoid this, write all parameters with the WR command before doing a firmware upgrade.
- Because explicit API Tx frames can be addressed to a local node (accessible via the SPI or UART) or a remote node (accessible over the RF port) the same process can be used to update firmware on a module in either case.

7. 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 or SPI Data Frame.

Please note that Digi may add new API frames to future versions of firmware, so please 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 or SPI data frame structure is defined as follows:

UART or SPI Data Frame Structure:

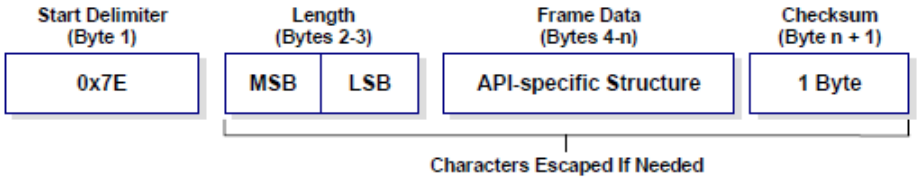
MSB = Most Significant Byte, LSB = Least Significant Byte

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), SPI mode is not supported and the UART 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 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.

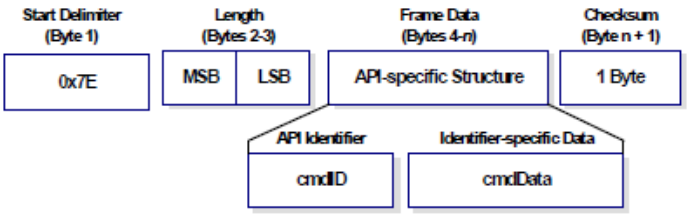
Length

The length field has a 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.

Framed Data

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

UART or SPI Data Frame & 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 and Values

API Frame Names	API ID
Tx64 Request	0x00
AT Command	0x08
AT Command - Queue Parameter Value	0x09
Remote Command Request	0x07
TX IPv4	0x20
Rx64 Indicator	0x80
AT Command Response	0x88
TX Status	0x89
Modem Status	0x8A
IO Data Sample Rx Indicator	0x8F
Remote Command Response	0x87
RX IPv4	0xB0

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 Examples

Example: Create an API AT command frame to configure an XBee baud rate to 230,400 (set BD to 0x08). The frame should look like (in hex):

7E 00 05 08 01 42 44 08 68

Where:

0x0005 = length excluding checksum
0x08 = AT Command API frame type
0x01 = Frame ID (set to non-zero value for transmit status)
0x4244 = AT Command ('BD')
0x08 = value to set command to
0x68 = Checksum

The checksum is calculated as [0xFF - (0x08 + 0x01 + 0x42 + 0x44 + 0x08)]

Example: Send a remote command to a module who's IP address is 192.168.0.103 (C0 A8 00 67) to set DIO1/AD1 as a digital input (D1=3) and apply changes to force the IO update. The API remote command frame should look like (in hex):

7E 00 0E 07 01 00 00 00 00 C0 A8 01 64 02 44 31 03 B0

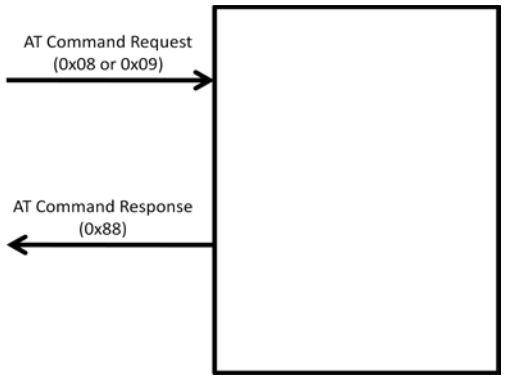
Where:

0x000E = length (14 bytes excluding checksum)
0x07 = Remote Command API frame type
0x01 = Frame ID
0x00000000 C0A80067 = Remote address (Pad first 4 bytes with 00)
0x02 = Apply Changes (Remote Command Options)
0x4431 = AT command ('D1')
0xB0 = Checksum

API UART and SPI Exchanges

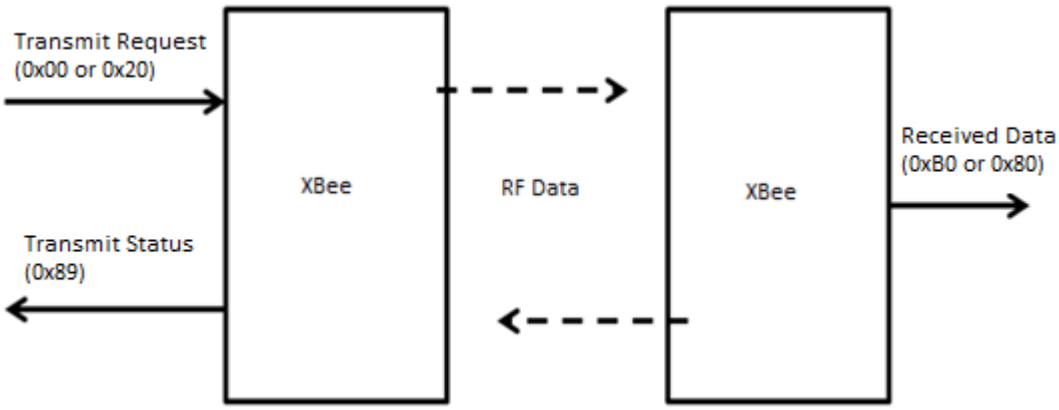
AT Commands

The following image shows the API frame exchange that takes place at the UART or SPI 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.



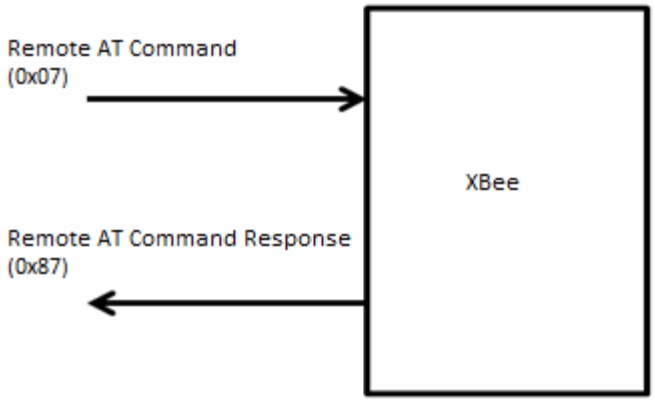
Transmitting and Receiving RF Data

The following image shows the API exchanges that take place at the UART or SPI 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 (0x80 or 0xB0) is set by the AP command.



Remote AT commands

The following image shows the API frame exchanges that take place at the UART or SPI when sending a remote AT command. A remote command response frame is not sent out the UART or SPI 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;

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

        }
}
```

API Frames

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

TX (Transmit) Request: 64-Bit

Frame Type: 0x0

This frame type uses the XBee Application Service. This command allows for software compatibility with other XBee module such as the 802.15.4 module.

	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	0x0D	
	API Frame Specific Data	API Frame Identifier	3	0x00	
		Frame ID	4	0x01	
		64-Bit Destination Address	5	0x00	Align IP address to low 32-bits of the field. The other bytes set to 0. IP address is in hex. The address in this example is 192.168.0.100
				0x00	
				0x00	
				0x00	
				0xC0	
				0xA8	
				0x00	
				0x64	
		TX Options	13	0x00	0x01 – Disable ACK All other bits must be set to 0.
		Data	14	0x1516	Max is 1392 bytes. Data will be sent to the XBee application service port.
	Checksum			0xD9	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

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 NI parameter value of the module.

	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	
	API Frame Specific Data	API Frame Identifier	3	0x08	
		Frame ID	4	0x01	
		AT Command	MSB 5	0x4E(N)	Command Name - Two ASCII characters that identify the AT command
			LSB 6	0x49(I)	
		Parameter Value	-	-	If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
	Checksum		7	0x5E	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

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
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x05	
	API Frame Specific Data	API Frame Identifier	3	0x09	
		Frame ID	4	0x01	
		AT Command	MSB 5	0x42 (B)	Command Name - Two ASCII characters that identify the AT command
			LSB 6	0x44 (D)	
		Parameter Value	7	0x07	If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
	Checksum		8	0x68	0xFF minus 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.

Remote AT Command Request

Frame Type: 0x07

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.

Example: Send a remote command to query the DL register on a remote device. In this example, the IP address of the remote is 192.168.0.100.

	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	0x0D	
	API Frame Specific Data	API Frame Identifier	3	0x07	
		Frame ID	4	0x01	
		64-Bit Destination Address	5	0x00	Align IP address to low 32-bits of the field. The other bytes set to 0. IP address is in hex. The address in this example is 192.168.0.100
			6	0x00	
			7	0x00	
			8	0x00	
			9	0xC0	
			10	0xA8	
			11	0x00	
			12	0x64	
		Command Options	13	0x02	0x02 – Apply changes on the remote. If not set then the AC command must be sent or the last remote command sent must set this option.
		AT Command	MSB 14	0x44(D)	Command Name - Two ASCII characters that identify the AT command
			LSB 15	0x4C(L)	
		Parameter Value	-		If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
	Checksum		16	0x99	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

Transmit (TX) Request: IPv4

Frame Type: 0x20

This frame type utilizes the serial data service. The frame gives greater control to the application over the IP setting for the data.

	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	0x11	
	API Frame Specific Data	API Frame Identifier	3	0x20	
		Frame ID	4	0x01	Set to a value that will be passed back in the Tx Status frame. 0 disables the Tx Status frame.
		IPv4 32 bit Destination Address	MSB 5	0xC0	Use 0xFFFFFFFF for broadcast when protocol is UDP. The address in the example is for a destination of 192.168.0.100
			6	0xA8	
			7	0x00	
			8	0x64	
		16 Bit Destination Port	MSB 9	0x26	UDP or TCP port number
			LSB 10	0x16	
		16 bit Source Port	MSB 11	0x26	UDP or TCP port number To send a UDP packet, this must match the port number of the listening port as specified by C0. To send a TCP packet on a new connection, this must be 0.
			LSB 12	0x16	
		Protocol	13	0x00	0 = UDP, 1= TCP - Protocol use for the transmitted data
		Transmit Options Bitfield	14	0x00	Bit field: BIT 1 = 1 - Terminate socket after tx complete 0 - Leave socket open (use TCP timeout). Ignore bit for UDP packets. All other bits are reserved and should be 0.
		RF Data	15	0x48('H')	Up to 1400 bytes of data
			16	0x65('e')	
			17	0x6C('l')	
			18	0x6C('l')	
			19	0x6F('o')	
	Checksum		20	0xA6	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

Rx (Receive) Packet: 64-bit

Frame Type: 0x80

This frame type is used by XBee when RF data is received using the XBee application service. It allows for software compatibility with other XBee modules such as 802.15.4. An example of this frame type is given below:

	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	
	API Frame Specific Data	API Frame Identifier	3	0x80	
		64-Bit Source Address	4	0x00	Align IP address to low 32-bits of the field. The other bytes set to 0. IP address is in hex. The example uses address 192.168.0.103
			5	0x00	
			6	0x00	
			7	0x00	
			8	0xC0	
			9	0xA8	
			10	0x00	
			11	0x67	
		RSSI	12	0x2E	RSSI in terms of dBm above sensitivity (link margin)
		Options	13	0x00	None currently defined
		RF Data	14	0x48 'H'	Up to 1392 bytes of data
			15	0x65 'e'	
			16	0x6C 'l'	
			17	0x6C 'l'	
			18	0x6F 'o'	
	Checksum		19	0x8E	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

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 AS (Active Scan) command).

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

	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	
	API Frame Specific Data	API Frame Identifier	3	0x88	
		Frame ID	4	0x01	
		AT Command	MSB 5	0x42 (B)	Command Name - Two ASCII characters that identify the AT command
			LSB 6	0x44 (D)	
		Status		0x00	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter
		Parameter Value	7		Register data in binary format. If the register was set, then this field is not returned, as in this example.
	Checksum		8	0xF0	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

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 a module is powered on in API mode.

	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	
	API Frame Specific Data	API Frame Identifier	3	0x8A	
		Status	4	0x00	0 = Hardware reset or power up 1 = Watchdog timer reset 2 = Joined 3 = No longer joined to access point 4 = IP configuration error Whenever the most significant bit of the Status byte is set, the WiFi transceiver is reporting a problem. These are the most likely modem status codes from the WiFi transceiver: 0x82 = Send or join command issued without first connecting from access point 0x83 = Access point not found 0x84 = PSK not configured 0x87 = SSID not found 0x88 = Failed to join with security enabled 0x8A = Invalid channel 0x8E = Failed to join access point
	Checksum		5	0x75	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

Note: New modem status codes may be added in future firmware releases.

Transmission Status

Frame Type: (0x89)

RF transmission status messages are sent from the module in response to transmission attempts.

Example: The following API frame is returned when a successful transmission occurs on an API transmission using frame ID 01.

	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	0x03	
	API Frame Specific Data	API Frame Identifier	3	0x89	
		Frame ID	4	0x01	Identifies the frame for which status is being reported. This number corresponds with the Frame ID provided in the transmission. If that frame ID was 0, then this frame will not be generated.
		Status	5	0x00	0x00 = Success 0x03 = Transmission was purged because it was attempted before stack was completely up. 0x04 = Physical error occurred on the interface with the WiFi transceiver. 0x21 = TX64 transmission timed out awaiting an acknowledgement from the remote device. 0x32 = Resource Error. Either buffers or sockets were depleted, preventing a transmission from occurring. 0x74 = Message not sent because it was too long 0x76 = Attempt to create a client socket failed 0x77 = TCP connection to given IP address and port doesn't exist. Source port is non-zero so that a new connection is not attempted. 0x78 = Source port on a UDP transmission doesn't match a listening port on the transmitting module.
	Checksum		6	0x75	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

Note: New transmission status codes may be added in future firmware releases.

IO Data Sample RX Indicator

Frame Type: 0x8F

When the module receives an IO sample frame from a remote device, it sends the sample out the UART or SPI using this frame type. Only modules running API mode will be able to receive IO samples.

Example: The following is the IO sample response from a radio at IP address 192.168.0.103 reporting one active DIO (DIO8) and one active analog input (AN1).

	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	0x13	
	API Frame Specific Data	API Frame Identifier	3	0x8F	
		64-Bit Source Address	4	0x00	Align IP address to low 32-bits of the field. The other bytes set to 0. IP address is in hex. The example uses address 192.168.0.103
			5	0x00	
			6	0x00	
			7	0x00	
			8	0xC0	
			9	0xA8	
			10	0x00	
			11	0x67	
		RSSI in terms of link margin	12	0x2E	
		Receive Options	13	0x00	None currently defined
		Number of samples	14	0x01	Number of sample sets included in the payload. (Always set to 1)
		Digital Channel Mask*	MSB 15	0x01	Bitmask field that indicates which digital IO lines on the remote have sampling enabled (if any). In this example DIO8 is active.
			LSB 16	0x00	
		Analog Channel Mask**	17	0x81	Bitmask field that indicates which analog IO lines on the remote have sampling enabled (if any). The most significant bit signals that the Vcc value is included in the frame. In this example Analog input 1 and Vcc are active.
		Digital Samples (if included)	MSB 18	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. The bits in these 2 bytes map the same as they do in the Digital Channels Mask field. In this example, DIO8 has value 0.
			LSB 19	0x00	
		Analog Sample	MSB 20	0x03	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 DIO0/AD0 to DIO3/AD3, to the supply voltage.
			LSB 21	0xB5	
	Checksum		22	0x38	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

*	N/A	N/A	N/A	CD/DIO 12	PWM/DIO11	RSSI/DIO10	N/A	N/A
	CTS/DIO7	RTS/DIO6	ASSOC DIO5	DIO4	AD3/DIO3	AD2/DIO2	AD1/DIO1	AD0/DIO0
**	Supply Voltage	N/A	N/A	N/A	AD3	AD2	AD1	AD0

Remote Command Response

Frame Type: 0x87

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 or SPI.

Example: If a remote command is sent to a remote device with an IP address of 192.168.0.103 to set the D1 parameter to 3 (digital input), the response is shown in the example API frame in the table below.

	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	0x0D	
	API Frame Specific Data	API Frame Identifier	3	0x87	
		Frame ID	4	0x01	
		64-Bit Responder Address	5	0x00	Align IP address to low 32-bits of the field. The other bytes set to 0. Value is in hex. In this example the IP address is 192.168.0.103
			6	0x00	
			7	0x00	
			8	0x00	
			9	0xC0	
			10	0xA8	
			11	0x00	
			12	0x67	
		AT Command	MSB 13	0x44 (D)	Command Name - Two ASCII characters that identify the AT command
			LSB 14	0x31 (1)	
		Status	15	0x00	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter 4 = Tx Failure
		Parameter Value		-	If present, indicates value of the requested parameter. If not present, this is not a response to a query command.
	Checksum		16	0x33	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

RX (Receive) Packet: IPv4

Frame Type: 0xB0

This frame is used by XBee when RF data is received using the Serial Data service on the port defined by the C0 command.

Example: When a module in API mode receives an IPv4 transmission, it will produce an RX notification (0xB0) and send it out the UART or SPI. This example is the response to a UDP transmission to IP address 192.168.0.103 with data 'Hello' from the source address 192.168.0.104.

	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	
	API Frame Specific Data	API Frame Identifier	3	0xB0	
		IPv4 32 bit Source Address	MSB 4	0xC0	The address in the example is for a source address of 192.168.0.104
			5	0xA8	
			6	0x00	
			7	0x68	
		16 Bit Destination Port	MSB 8	0x26	Same value as the C0 command.
			LSB 9	0x16	
		16 bit Source Port	MSB 10	0x26	
			LSB 11	0x16	
		Protocol	MSB 12	0x00	0 = UDP, 1= TCP - Protocol use for the transmitted data
		Status	13	0x00	Reserved
		RF Data	14	0x48 'H'	Up to 1400 bytes of data
			15	0x65 'e'	
			16	0x6C 'l'	
			17	0x6C 'l'	
			18	0x6F 'o'	
	Checksum		19	0x13	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

8. XBee Command Reference Tables

Addressing

AT Command	Name and Description	Parameter Range	Default
DL	**Destination Address Low**. Set/Get the 32 bits of the IPv4 destination address. Using AT command mode this value is entered using dotted notation (example 192.168.0.100).	0.0.0.0 – 255.255.255.255	255.255.255.255
MY	**IP Network Address**. Read the 32-bit network address of the module when using DHCP. Set/Read values when using static IP address.	0.0.0.0 – 255.255.255.255	0.0.0.0
MK	**IP Address Mask.** This command is read only when DHCP is enabled.	0.0.0.0 – 255.255.255.255	0.0.0.0
GW	**Gateway IP address.** This command is read only when DHCP is enabled.	0.0.0.0 – 255.255.255.255	0.0.0.0
SH	**Serial Number High**. Read the high 16 bits of the module's unique 48-bit address.	0 - 0xFFFFFFFF [read-only]	factory-set
SL	**Serial Number Low**. Read the low 32 bits of the module's unique 48-bit address.	0 - 0xFFFFFFFF [read-only]	factory-set
NI	**Node Identifier.** Stores a string identifier. The register only accepts printable ASCII data. In AT Command Mode, a string cannot start with a space. A carriage return ends the command. Command will automatically end when maximum bytes for the string have been entered.	20-Byte printable ASCII string	ASCII space character (0x20)
DE	**Destination Port**. Set/Get destination UDP/TCP port value.	0 - 0xFFFF	0x2616
C0	**Serial Communication Service Port.** Set/Get port number used to provide the serial communication service. Data sent to this port will come out of the serial port of the module. The protocol used is set by the IP command when UART is in transparent mode.	0 – 0xFFFF	0x2616
DD	**Device Type Identifier**. Stores a device type value. This value can be used to differentiate different XBee-based devices. Digi reserves the range 0 - 0xFFFFFF.	0-0xFFFFFFFF	0x90000
NP	**Maximum RF Payload Bytes**. This value returns the maximum number of RF payload bytes that can be sent in a transmission **Note**: NP returns a hexadecimal value. (e.g. if NP returns 0x54, this is equivalent to 84 bytes)	0 - 0xFFFF	[read-only]

Networking Commands

AT Command	Name and Description	Parameter Range	Default
ID	**SSID**. Set/read the SSID of the access point, which may be up to 31 ASCII characters	Up to 31 bytes of printable ASCII	NULL
AH	**Network Type.** Set/read network type. Network types supported are Infrastructure (using an access point) and Adhoc (IBSS).	0-IBSS Joiner 1-IBSS Creator 2 - Infrastructure	2
IP	**IP Protocol**. Set/Read the protocol used for the serial communication service. This is the port used by the C0 command.	0 – UDP 1 - TCP	0
MA	**IP Addressing Mode**. Set / read the IP addressing mode.	0 – DHCP 1 – Static	0
TM	**TCP timeout.** Set/Read the timeout for connection on TCP client sockets. If 0, socket closes immediately after data sent.	0-0xFFFF [x 100 msec]	0x64
TS	**TCP Server Socket Timeout.** Set/Read the timeout for connection on a TCP server socket. This is a socket whose connection was initiated at the other end.	0 x000A– 0xFFFF * 100 ms.	0x0258 (1 minute)

Security Commands

AT Command	Name and Description	Parameter Range	Default
EE	**Encryption Enable**. Set/Read the encryption enable setting.	0 – No security 1 – WPA 2 – WPA2 3 - WEP	0
PK	**Security Key**. Set the security key used for WEP, WPA, and WPA2 security. This command is write only; PK cannot be read.	0 -31-ASCII characters for WPA and WPA2, Either 5 or 13 ASCII characters should be used for the WEP password, based on the access point key length (64 or 128 bits respectively).	

RF Interfacing Commands

AT Command	Name and Description	Parameter Range	Default
PL	**Power Level**. Select/Read the power level at which the RF module transmits conducted power.	0 – 0 dBm 1 – 5 dBm 2 – 10 dBm 3 – 15 dBm 4 – Max power	4
CH	**Channel**. Read the channel number of the access point or 0xFF if not associated. Channel can be set when AH is configured for Adhoc creator mode. Note when using Adhoc mode, not all channels are available in all countries. It is the responsibility of the installer to use the appropriate channels.	1-0xB	[read only]

Serial Interfacing

AT Command	Name and Description	Parameter Range	Default
AP	**API Enable**. Enable API Mode.	0 = Transparent mode 1 = API-enabled 2 = API-enabled (w/escaped control characters)	1
AO	**API Output Options.** Indicates the type of frame to output when data is received on the IP services port	0=ZigBee Rx 1=Explicit Zigbee Rx 2=RX64	2 (RX64)
BD	**Interface Data Rate**. Set/Read the serial interface data rate for communication between the module serial port and host. Any value above 0x0A will be interpreted as an actual baud rate. When a value above 0x0A is sent, the closest interface data rate represented by the number is stored in the BD register.	1 - 7 (standard baud rates) 1 = 2400 bps 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200 8 = 230400 9 = 460,800 0xA = 921,600 0X5B9 - 0X5B8D80 (non-standard rates up to 6mps)	3
NB	**Serial Parity**. Set/Read the serial parity setting on the module.	0 = No parity 1 = Even parity 2 = Odd parity	0
SB	**Stop Bits**. Set/read the number of stop bits for the UART. (Two stop bits are not supported if mark parity is enabled.)	0 = 1 stop bit 1 = 2 stop bits	0
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 . Regardless of how small RO is, the inter-character silence required to trigger a transmission of the data is 100 usec.	0 - 0xFF [x character times]	3
FT	**Flow Control Threshold.** De-assert CTS when FT bytes are in the UART receive buffer	0x11 – 0x823	0x7F3
D7	**DIO7 Configuration**. Select/Read options for the DIO7 line of the RF module.	0 = Disabled 1 = CTS Flow Control 3 = Digital input 4 = Digital output, low 5 = Digital output, high 6 = RS-485 transmit enable (low enable) 7 = RS-485 transmit enable (high enable)	1
D6	**DIO6 Configuration.** Configure options for the DIO6 line of the RF module.	0 = Disabled 1 = RTS flow control 3 = Digital input 4 = Digital output, low 5 = Digital output, high	0

I/O Settings

AT Command	Name and Description	Parameter Range	Default
IS	**Force Sample** Forces a read of all enabled digital and analog input lines.	-	-
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-D8, P0-P2 commands). The sample rate is measured in milliseconds. WARNING: If IR is set to 1 or 2, the module will not keep up and many samples will be lost.	0-0xFFFF (x 1 ms)	0 – no sampling
IC	I**O 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.	0 - 0xFFFF	0
IF	**Sample from Sleep Rate.** The number of sleep cycles that must elapse between periodic I/O samples. This allows I/O samples to be taken only during some wake cycles. During those cycles I/O samples are taken at the rate specified by IR. IR can be 0 which will cause only one sample to be taken.	1-0xFF (1 gives you a sample every sleep cycle)	1
P0	**DIO10 Configuration**. Select/Read function for the DIO10 line of the RF module.	0 = Disabled, 2 = PWM0 Output 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	0
P1	**DIO11 Configuration**. Select/Read function for the DIO11 line of the RF module.	0 = Disabled 2 = PWM1 Output 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	0
P2	**DIO12 Configuration**. Select/Read function for the DIO12 line of the RF module.	0 = Disabled 1 = SPI_MISO 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	0
P3	**DOUT.** Enables or disables output on UART port	0 = Disabled 1 = Enabled	1
P4	**DIN.** Enables or disables input on UART port	0 = Disabled 1 = Enabled	1

AT Command	Name and Description	Parameter Range	Default
D0	**DIO0/AD0 Configuration**. Select/Read function for DIO0/AD0.	0 = Disabled 2 = Analog input 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	0
D1	**DIO1/AD1 Configuration**. Select/Read function for DIO1/AD1	0 - Disabled 1 = SPI Attention 2 = Analog input 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	0
D2	**DIO2/AD2 Configuration**. Select/Read function for DIO2/AD2	0 = Disabled 1 = SPI Clock 2 = Analog input 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	0
D3	**DIO3/AD3 Configuration**. Select/Read function for DIO3/AD3	0 = Disabled 1 = SPI Slave Select 2 = Analog input 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	0
D4	**DIO4 Configuration**. Select/Read function for DIO4	0 = Disabled 1 = SPI_MOSI 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	0
D5	**DIO5 Configuration**. Select/Read function for DIO5	0 = Disabled 1 = Associated LED 3 = Digital input 4 = Digital output, default low 5 = Digital output, default high	1

AT Command	Name and Description	Parameter Range	Default
D8	**DIO8 Configuration**. Select/Read function for DIO8	0 = Disabled 1 = SleepRq 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	1
D9	**DIO9 Configuration.** Select/Read function for DIO9	0 = Disabled 1 = On/Sleep indicator 3 = Digital input, monitored 4 = Digital output, default low 5 = Digital output, default high	1
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 when the module has joined a network. If LT=0, the default blink rate of 250ms will be used. For all other LT values, LT is measured in 10ms.	0, 0x14 - 0xFF (200 - 2550 ms)	0
PR	**Pull-up Resistor**. Set/read the bit field that configures the internal resistor status for the I/O lines. "1" specifies the resistor is enabled. "0" specifies no resistor. The PD command specifies whether the resistor is pull-up or pull-down. Bits:* 0 – DIO4 (Pin 11) 1 – DIO3 / AD3 (Pin 17) 2 – DIO2 /AD2 (Pin 18) 3 – DIO1/AD1 (Pin 19) 4 – DIO0 / AD0 (Pin 20) 5 – DIO6 / RTS (Pin 16) 6 – DIO8 / DTR / Sleep Request (Pin 9) 7 – DIN / Config (Pin 3) 8 – DIO5 / Associate (Pin 15) 9 – DIO9 / On/Sleep (Pin 13) 10 – DIO12 (Pin 4) 11 – DIO10 / PWM0 (Pin 6) 12 – DIO11/PWM1 (Pin 7) 13 – DIO7 / CTS (Pin 12)	0 - 0x7FFF	0x7FFF
PD	**Pull Direction**. Set/Read resistor direction for the corresponding bits set in PR. If the bit is not set in PR, then PD is unused.	0 – 0x7FFF	0x7FFF
DS	**Drive Strength** Set/Read the output drive strength (output amperes) for DIO lines. Bits are mapped the same as the PR and PD commands. If the bit is set, the drive strength is 6mA . Otherwise, it is 2mA.	0 – 0x7FFF	0
AV	**Analog Voltage Reference**. Set/Read the analog voltage reference. This specifies the volts for an analog reading of 0x03ff, where a reading of 0x200 indicates a voltage input that is half of VREF. VREF may be one of these two values: 0 – 1.25Volts 1 – 2.5 Volts	0-1	1
M0	**PWM0 Duty cycle**. Sets the duty cycle of PWM0 for P0=2, where a value of 0x200 is a 50% duty cycle.	0 – 0x03FF	0
M1	**PWM1 Duty cycle**. Sets the duty cycle of PWM1 for P1=2, where a value of 0x200 is a 50% duty cycle.	0 – 0x03FF	0

Diagnostics Interfacing

AT Command	Name and Description	Parameter Range	Default
VR	**Firmware Version**. Read firmware version of the module. The firmware version returns 4 hexadecimal values (2 bytes) "ABCD". Digits ABC are the main release number and D is the revision number from the main release. "B" is a variant designator where 0 means standard release.	0 - 0xFFFF [read-only]	Factory-set
HV	**Hardware Version**. Read the hardware version of the module. This command can be used to distinguish among different hardware platforms. The upper byte returns a value that is unique to each module type. The lower byte indicates the hardware revision. XBee WiFi modules return 0x1Fxx for the HV command.	0 - 0xFFFF [read-only]	Factory-set
HS	**Hardware Series.** Indicates the hardware series number of the module. This module should indicate 0x601 for S6B.		
AI	**Association Indication**. Read information regarding last node join request: 0x00 - Successfully joined an access point, established IP addresses and IP listening sockets. 0x01 - WiFi transceiver initialization in progress. 0x02 - WiFi transceiver initialized, but not yet scanning for access point. 0x13 - Disconnecting from access point. 0x23 – SSID not configured. 0x24 - Encryption key invalid (either NULL or invalid length for WEP) 0x27 – SSID was found, but join failed. 0x41 – Module joined a network and is waiting for IP configuration to complete, which usually means it is waiting for a DHCP provided address. 0x42 – Module is joined, IP is configured, and listening sockets are being set up. 0xFF– Module is currently scanning for the configured SSID. **Note**: New non-zero AI values may be added in later firmware versions. Applications should read AI until it returns 0x00, indicating a successful startup.	0 - 0xFF [read-only]	-
AS	**Active Scan.** Scan for access points in the vicinity. This command can only be issued when SSID is NULL, which can be forced by first issuing the NR command. If SSID is not NULL, then the active scan command returns an error. This command may be issued in command mode or in API mode. In either case, the following information is returned for each access point found: 02 – Indicates scan type of 802.11 in this format unique to S6B. CH - Channel number in use by access point ST – Security type where: 00=open, 01=WPA, 02=WPA2, and 03=WEP LM - Link Margin (Signal strength in dBm above sensitivity) ID = SSID of access point found. When this command is issued in command mode, the above record is displayed, one per line for each access point found. Readable ASCII characters are outputs with a carriage return and each field on a new line. When it is issued in API mode, each record (i.e. each access point) outputs a separate AT command response of type 0x88 with the above fields in binary format. Note that this command is not available as a remote command.	-	-
TP	**Temperature.** Read temperature of module in degrees Celsius.	-40 to 85C	-
CK	**Configuration Code.** Read the configuration code associated with the current AT command configuration	2 bytes	-
%V	**Supply Voltage.** Read supply voltage in millivolt units.	3.1 to 3.6V	-
LM	**Link Margin.** Reads the received signal strength (RSSI) in terms of dBm units above sensitivity. It will report 0xff until the first reception after connection to access point.	0 – 0xFF	

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. This time can be up to ten minutes.	2 - 0x1770 [x 100 ms]	0x64 (100d)
CN	**Exit Command Mode.** Explicitly exit the module from AT Command Mode. (Whether command mode is left by the CN command or by CT timing out, changes will be applied upon exit.	-	-
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.	2 - 0x0CE4 [x 1 ms] (max of 3.3 decimal sec)	0x3E8 (1000d)
CC	**Command Mode Character** Set/read the command mode character used between guard times of the AT Command Mode Sequence (GT + CC + CC + CC + GT). This sequence allows the module to enter into AT Command Mode.	0 - 0xFF	0x2B ('+' ASCII)

Sleep Commands

AT Command	Name and Description	Parameter Range	Default
SM	**Sleep Mode** Sets the sleep mode on the RF module. Sleep mode is also affected by the SO command, option bit 6. See the "Sleep" chapter for a full explanation of the various sleep modes.	0 = No sleep 1 = Pin sleep 4 = Cyclic sleep 5 = Cyclic sleep, pin wake	0
SP	**Sleep Period.** This value determines how long the device will sleep at a time, up to 24 hours or 86,400 seconds. This corresponds to 0x83d600 in 10ms units.	1 - 0x83D600 x 10ms	0xC8 (2 seconds)
SO Command	**Sleep Options**. Configure options for sleep. Unused option bits should be set to 0. Sleep options include: 0x40 – Stay associated with AP during sleep. Draw more current during sleep with this option enabled, but also awake from sleep more rapidly. 0x100 – For cyclic sleep, ST specifies the time before returning to sleep. With this bit set, new receptions from either the serial or the RF port will NOT restart the ST timer. Current implementation does not support this bit being turned off.	0 - 0x01FF	0x100
WH	**Wake Host**. Set/Read the wake host timer value. If the wake host timer is set to a non-zero value, this timer specifies a time (in millisecond units) that the device should allow after waking from sleep before sending data out the UART or transmitting an IO sample. If serial characters are received, the WH timer is stopped immediately.	0 - 0xFFFF (x 1ms)	0
ST	**Wake Time.** Wake time for cyclic modes. New data will not refresh the timer. However, if there is data to transmit or receive after ST expires, those actions will occur before the module goes to sleep. Max wake time is 3600 seconds.	0x1 – 0x36EE80 (x 1 ms)	0x7D0
SA	**Association Timeout.** Time to wait for association before entering deep sleep. (Wakeup from deep sleep is much faster if association occurs before going to sleep.)	0x1 – 0x36EE80 (x1 ms)	0x2710 (10 seconds)

Execution Commands

Where most AT commands set or query register values, execution commands cause an action to be executed on the module. Execution commands are executed immediately and do not require changes to be applied.

AT Command	Name and Description	Parameter Range	Default
AC	**Apply Changes**. Applies changes to all command registers causing queued command register values to be applied. For example, changing the serial interface rate with the BD command will not change the UART interface rate until changes are applied with the AC command. The CN command and 0x08 API command frame also apply changes.	-	-
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. The WR command should be used sparingly to preserve flash.	-	-
RE	**Restore Defaults**. Restore module parameters to factory defaults.	-	-
FR	**Software Reset.** Reset module. Responds immediately with an OK status, and then performs a software reset about 2 seconds later.	-	-
NR	**Network Reset**. Reset network layer. For WiFi, this means to disassociate from the access point and set SSID to NULL, thereby preventing the node from immediately establishing the same connection with the same access point. This also allows the active scan command to be executed so that access point candidates can be selected from the resultant list. Note that NR and NR0 both do the same thing and may be used interchangeably.	0	-

9. Module Support

This chapter provides customization information for the XBee Wi-Fi module. In addition to providing an extremely flexible and powerful API, the XBee module is a robust development platform that has passed FCC and ETSI testing.

X-CTU Configuration Tool

Digi provides a Windows X-CTU configuration tool for configuring module parameters and updating firmware. The XCTU has the capability to do the following:

- Update firmware on a local module (requires USB or serial connection)
- Read or write module configuration parameters on a local
- Save and load configuration profiles containing customized settings.

Contact Digi support for more information about the X-CTU.

Serial Firmware Updates

Serial firmware updates make use of the XBee bootloader which ships in all modules. This bootloader allows firmware to be updated. Normally, the running application can be told to invoke the bootloader through a command from X-CTU. If that command is not available in the currently loaded firmware, the bootloader includes a modified entry mechanism using pins 3, 9, and 16 (DIN, DTR, and RTS, respectively). By driving pin 3 low, pin 9 low, and pin 16 high at the time the module is reset, the XBee bootloader is forced to run, allowing a new version of firmware to load. This method works even when the current firmware version does not support the firmware upgrade feature.

The X-CTU program can update firmware on the XBee module over the UART port, but not currently over the SPI port. Contact Digi support for details.

Regulatory Compliance

XBee modules are certified for FCC and IC operation on all 11 channels (1-11) allowable, and ETSI certified for all 13 channels (1-13) allowable.

10. Agency Certifications

United States FCC

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (*1*) this device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operation.

The XBee Wi-Fi Module complies with Part 15 of the FCC rules and regulations. Compliance with the labeling requirements, FCC notices and antenna usage guidelines is required.
To fulfill FCC Certification, the OEM must comply with the following regulations:

1. The system integrator must ensure that the text on the module label is placed on the outside of the final product.
2. XBee Wi-Fi Module may only be used with antennas that have been tested and approved for use with this module [refer to the antenna tables in this section].

OEM Labeling Requirements

WARNING: The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the final product enclosure.

Required FCC Label for OEM products containing the XBee Wi-Fi S6B Through-hole Module

Contains FCC ID: MCQ-XBS6B
The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (*i*.) this device may not cause harmful interference and (*ii*.) this device must accept any interference received, including interference that may cause undesired operation.

Required FCC Label for OEM products containing the XBee Wi-Fi S6B Surface Mount Module

Contains FCC ID: MCQ-S6BSM
The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (*i*.) this device may not cause harmful interference and (*ii*.) this device must accept any interference received, including interference that may cause undesired operation.

The integrator is responsible for its product to comply with FCC Part 15, Sub. B - Unintentional Radiators.

FCC Notices

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

IMPORTANT: OEMs must test final product to comply with unintentional radiators (FCC section 15.107 & 15.109) before declaring compliance of their final product to Part 15 of the FCC Rules.
IMPORTANT: The RF module has been certified for remote and base radio applications. If the module will be used for portable applications, the device must undergo SAR testing.
This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

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

FCC-Approved Antennas (2.4 GHz)

The XBee Wi-Fi Module can be installed utilizing antennas and cables constructed with non-standard connectors (RPSMA, RPTNC, etc.). An adapter cable may be necessary to attach the XBee connector to the antenna connector.

The modules are FCC approved for fixed base station and mobile applications. If the antenna is mounted at least 20cm (8 in.) from nearby persons, the application is considered a mobile application. Antennas not listed in the table must be tested to comply with FCC Section 15.203 (Unique Antenna Connectors) and Section 15.247 (Emissions). XBee Wi-Fi RF Modules have been approved for use with all the antennas listed in the tables below. (Cable-loss is required when using gain antennas as shown below.) Digi does not carry all of these antenna variants. Contact Digi Sales for available antennas.

Antennas approved for use with the XBee Wi-Fi Through-hole Module

Integrated Antennas							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
29000294	Integral PCB antenna	-0.5 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A
A24-QI	Monopole (Integrated Whip)	1.5 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A

Dipole Antennas					
					Minimum Cable Loss/Power Reduction/Attenuation Required

Part Number	Type (Description)	Gain	Application	Min Separation	b mode	g mode	n mode
A24-HASM-450	Dipole (Half-wave articulated RPSMA-4.5")	2.1 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A
A24-HABSM	Dipole (Articulated RPSMA)	2.1 dBi	Fixed	20 cm	N/A	N/A	N/A
A24-HABUF-P5I	Dipole (Half-wave bulkhead mount U.FL s/ 5" pigtail)	2.1 dBi	Fixed	20 cm	N/A	N/A	N/A
A24-HASM-525	Dipole (Half-wave articulated RPSMA-5.25")	2.1 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A

Omni-Directional Antennas							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
A24-F2NF	Omni-Directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A
A24-F3NF	Omni-Directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A
A24-F5NF	Omni-Directional (Fiberglass base station)	5.0 dBi	Fixed	20 cm	N/A	N/A	N/A
A24-F8NF	Omni-Directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	N/A	0.4 dB	0.4 dB
A24-F9NF	Omni-Directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	0.4 dB	2.4 dB	2.4 dB
A24-F10NF	Omni-Directional (Fiberglass base station)	10 dBi	Fixed	2 m	0.9 dB	2.9 dB	2.9 dB
A24-F12NF	Omni-Directional (Fiberglass base station)	12 dBi	Fixed	2 m	2.9 dB	4.9 dB	4.9 dB
A24-F15NF	Omni-Directional (Fiberglass base station)	15 dBi	Fixed	2 m	5.9 dB	7.9 dB	7.9 dB
A24-W7NF	Omni-Directional (base station)	7.2 dBi	Fixed	2 m	N/A	0.1 dB	0.1 dB
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	N/A	0.1 dB	0.1 dB

PANEL CLASS ANTENNAS							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	N/A	1.4 dB	1.4 dB
A24-P8NF	Flat Panel	8.5 dBi	Fixed	3 m	N/A	1.4 dB	1.4 dB
A24-P13NF	Flat Panel	13 dBi	Fixed	4 m	3.9 dB	5.9 dB	5.9 dB
A24-P14NF	Flat Panel	14 dBi	Fixed	5 m	4.9 dB	6.9 dB	6.9 dB
A24-P15NF	Flat Panel	15 dBi	Fixed	2 m	5.9 dB	7.9 dB	7.9 dB
A24-P16NF	Flat Panel	16 dBi	Fixed	2 m	6.9 dB	8.9 dB	8.9 dB
A24-19NF	Flat Panel	19 dBi	Fixed	2 m	9.9 dB	11.9 dB	11.9 dB

YAGI CLASS ANTENNAS							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
A24-Y6NF	Yagi (6 element)	8.8 dBi	Fixed	2 m	N/A	1.7 dB	1.7 dB
A24-Y7NF	Yagi (7 element)	9.0 dBi	Fixed	2 m	N/A	1.9 dB	1.9 dB
A24-Y9NF	Yagi (9 element)	10.0 dBi	Fixed	2 m	0.9 dB	2.9 dB	2.9 dB
A24-Y10NF	Yagi (10 element)	11.0 dBi	Fixed	2 m	1.9 dB	3.9 dB	3.9 dB
A24-Y12NF	Yagi (12 element)	12.0 dBi	Fixed	2 m	2.9 dB	4.9 dB	4.9 dB
A24-Y13NF	Yagi (13 element)	12.0 dBi	Fixed	2 m	2.9 dB	4.9 dB	4.9 dB
A24-Y15NF	Yagi (15 element)	12.5 dBi	Fixed	2 m	3.4 dB	5.4 dB	5.4 dB
A24-Y16NF	Yagi (16 element)	13.5 dBi	Fixed	2 m	4.4 dB	6.4 dB	6.4 dB
A24-Y16RM	Yagi (16 element, RPSMA connector)	13.5 dBi	Fixed	2 m	4.4 dB	6.4 dB	6.4 dB
A24-Y18NF	Yagi (18 element)	15.0 dBi	Fixed	2 m	5.9 dB	7.9 dB	7.9 dB

Antennas approved for use with the XBee Wi-Fi Surface Mount Module

Integrated Antennas							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
31000005-01	Integral PCB antenna	0 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A
A24-QI	Monopole (Integrated Whip)	1.5 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A

Dipole Antennas							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
A24-HASM-450	Dipole (Half-wave articulated RPSMA-4.5")	2.1 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A
A24-HABSM	Dipole (Articulated RPSMA)	2.1 dBi	Fixed	20 cm	N/A	N/A	N/A
A24-HABUF-P5I	Dipole (Half-wave bulkhead mount U.FL s/ 5" pigtail)	2.1 dBi	Fixed	20 cm	N/A	N/A	N/A
A24-HASM-525	Dipole (Half-wave articulated RPSMA-5.25")	2.1 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A

Omni-Directional Antennas							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
A24-F2NF	Omni-Directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A
A24-F3NF	Omni-Directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	N/A	N/A	N/A
A24-F5NF	Omni-Directional (Fiberglass base station)	5.0 dBi	Fixed	20 cm	N/A	N/A	N/A
A24-F8NF	Omni-Directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	N/A	N/A	1.5 dB
A24-F9NF	Omni-Directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	N/A	1.5 dB	1.0 dB
A24-F10NF	Omni-Directional (Fiberglass base station)	10 dBi	Fixed	2 m	0.5 dB	2.0 dB	2.5 dB
A24-F12NF	Omni-Directional (Fiberglass base station)	12 dBi	Fixed	2 m	2.5 dB	4.0 dB	4.5 dB
A24-F15NF	Omni-Directional (Fiberglass base station)	15 dBi	Fixed	2 m	5.5 dB	7.0 dB	7.5 dB
A24-W7NF	Omni-Directional (base station)	7.2 dBi	Fixed	2 m	N/A	N/A	N/A
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	N/A	N/A	N/A

PANEL CLASS ANTENNAS							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	N/A	0.5 dB	0.9 dB
A24-P8NF	Flat Panel	8.5 dBi	Fixed	3 m	N/A	0.5 dB	0.9 dB
A24-P13NF	Flat Panel	13 dBi	Fixed	4 m	3.5dB	5.0 dB	5.5 dB
A24-P14NF	Flat Panel	14 dBi	Fixed	5 m	4.5dB	6.0 dB	6.5 dB
A24-P15NF	Flat Panel	15 dBi	Fixed	2 m	5.5dB	7.0 dB	7.5 dB
A24-P16NF	Flat Panel	16 dBi	Fixed	2 m	6.5dB	8.0 dB	8.5 dB
A24-19NF	Flat Panel	19 dBi	Fixed	2 m	9.5dB	11.0 dB	11.5 dB

YAGI CLASS ANTENNAS							
					Minimum Cable Loss/Power Reduction/Attenuation Required		
Part Number	**Type (Description)**	**Gain**	**Application**	**Min Separation**	**b mode**	**g mode**	**n mode**
A24-Y6NF	Yagi (6 element)	8.8 dBi	Fixed	2 m	N/A	0.8 dB	1.2 dB
A24-Y7NF	Yagi (7 element)	9.0 dBi	Fixed	2 m	N/A	1.0 dB	1.5 dB
A24-Y9NF	Yagi (9 element)	10.0 dBi	Fixed	2 m	0.5 dB	2.0 dB	2.5 dB
A24-Y10NF	Yagi (10 element)	11.0 dBi	Fixed	2 m	1.5 dB	3.0 dB	3.5 dB
A24-Y12NF	Yagi (12 element)	12.0 dBi	Fixed	2 m	2.5 dB	4.0 dB	4.5 dB
A24-Y13NF	Yagi (13 element)	12.0 dBi	Fixed	2 m	2.5 dB	4.0 dB	4.5 dB
A24-Y15NF	Yagi (15 element)	12.5 dBi	Fixed	2 m	3.0 dB	4.5 dB	5.0 dB
A24-Y16NF	Yagi (16 element)	13.5 dBi	Fixed	2 m	4.0 dB	5.5 dB	6.0 dB
A24-Y16RM	Yagi (16 element, RPSMA connector)	13.5 dBi	Fixed	2 m	4.0 dB	5.5 dB	6.0 dB
A24-Y18NF	Yagi (18 element)	15.0 dBi	Fixed	2 m	5.5 dB	7.0 dB	7.5 dB

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

RF Exposure

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

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

Europe (ETSI)

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

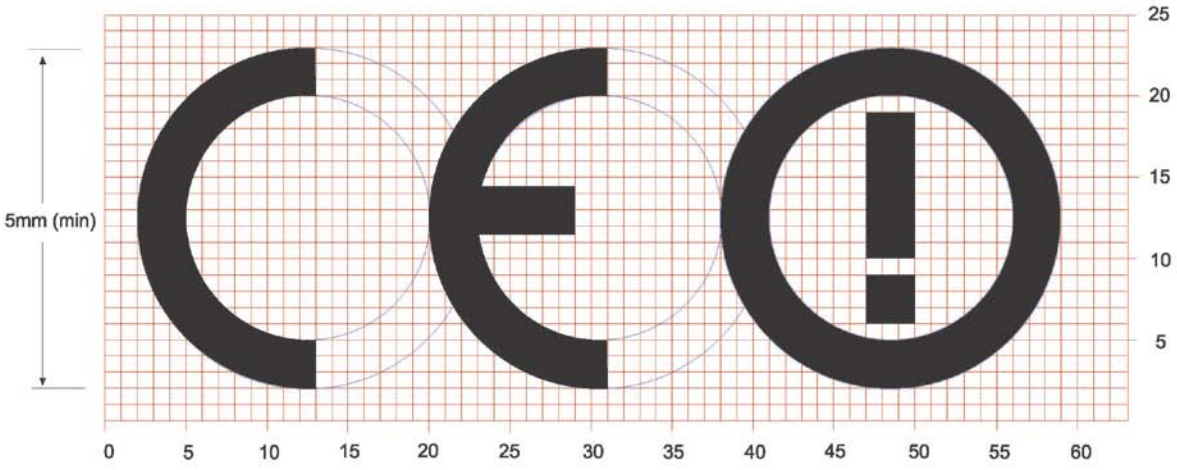
If the module is incorporated into a product, the manufacturer must ensure compliance of the final product to the European harmonized EMC and low-voltage/safety standards. A Declaration of Conformity must be issued for each of these standards and kept on file as described in Annex II of the R&TTE Directive.

Furthermore, the manufacturer must maintain a copy of the XBee 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.

OEM Labeling Requirements

The 'CE' marking must be affixed to a visible location on the OEM product.

CE Labeling Requirements



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

- If the CE! alert marking is reduced or enlarged, the proportions given in the above graduated drawing must be respected.
- The CE! alert 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! alert marking must be affixed visibly, legibly, and indelibly.

Restrictions

Declarations of Conformity

Digi has issued Declarations of Conformity for the XBee RF Modules concerning emissions, EMC and safety. Files can be obtained by contacting Digi Support.

Annex	Country	Restriction	Reason/remark
***Annex 3 Band A* Wideband Data Transmission systems 2400.0-2483.5 MHz**	France	Limited implementation	Outdoor use limited to 10 mW e.i.r.p. within the band 2454-2483.5 MHz. Military Radiolocation use. Refarming of the 2.4 GHz has been ongoing in recent years to allow current relaxed regulation. Full implementation planned 2012
	Italy		For private use, a general authorisation is required if WAS/RLAN's are used outside own premises. For public use, a general authorisation is required
	Norway	Implemented	This subsection does not apply for the geographical area within a radius of 20 km from the centre of Ny-Ålesund
	Russian Federation	Limited implementation	**1. SRD with FHSS modulation** 1.1. Maximum 2.5 mW e.i.r.p. 1.2. Maximum 100 mW e.i.r.p. Permitted for use SRD for outdoor applications without restriction on installation height only for purposes of gathering telemetry information for automated monitoring and resources accounting systems. Permitted to use SRD for other purposes for outdoor applications only when the installation height is not exceeding 10 m above the ground surface. 1.3.Maximum 100 mW e.i.r.p. Indoor applications **2. SRD with DSSS and other than FHSS wideband modulation** 2.1. Maximum mean e.i.r.p. density is 2 mW/MHz. Maximum 100 mW e.i.r.p. 2.2. Maximum mean e.i.r.p. density is 20 mW/MHz. Maximum 100 mW e.i.r.p. Permitted to use SRD for outdoor applications only for purposes of gathering telemetry information for automated monitoring and resources accounting systems or security systems. 2.3. Maximum mean e.i.r.p. density is 10 mW/MHz. Maximum 100 mW e.i.r.p. Indoor applications
	Ukraine	Limited implementation	e.i.r.p. ≤100 mW with built-in antenna with amplification factor up to 6 dBi

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 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/.

Approved Antennas

When integrating high-gain antennas, European regulations stipulate EIRP power maximums. The following antennas have been tested and approved for use with the embedded XBee RF Module:

- Dipole (2.1 dBi, Omni-directional, Articulated RPSMA, Digi part number A24-HABSM)
- PCB Antenna (-0.5 dBi)
- Wire Whip Antenna (1.5 dBi)

Canada (IC)

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes : (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement

Labeling Requirements

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

XBee Wi-Fi Through-hole:

Contains Model XBEES6B Radio, IC: 1846A-XBS6B

XBee Wi-Fi Surface Mount:

Contains Model S6BSM Radio, IC: 1846A-S6BSM

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

Transmitters with Detachable Antennas

This radio transmitter (IC: 1846A-XBS6B and IC: 1846A-S6BSM) has been approved by Industry Canada to operate with the antenna types listed in the tables above with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Le présent émetteur radio (IC: 1846A-XBS6B et IC: 1846A-S6BSM) a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés ci-dessous et ayant un gain admissible maximal et l'impédance requise pour chaque type d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

Detachable Antenna

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that necessary for successful communication.

Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire àl'établissement d'une communication satisfaisante.

Australia (C-Tick)

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

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

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

11. Warranty Information

1-Year Warranty

XBee RF Modules from Digi International, Inc. (the "Product") are warranted against defects in materials and workmanship under normal use, for a period of 1-year from the date of purchase. In the event of a product failure due to materials or workmanship, Digi will repair or replace the defective product. For warranty service, return the defective product to Digi International, shipping prepaid, for prompt repair or replacement.

The foregoing sets forth the full extent of Digi International's warranties regarding the Product. Repair or replacement at Digi International's option is the exclusive remedy. THIS WARRANTY IS GIVEN IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, AND DIGI SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL DIGI, ITS SUPPLIERS OR LICENSORS BE LIABLE FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, FOR ANY LOSS OF USE, LOSS OF TIME, INCONVENIENCE, COMMERCIAL LOSS, LOST PROFITS OR SAVINGS, OR OTHER INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT, TO THE FULL EXTENT SUCH MAY BE DISCLAIMED BY LAW. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES. THEREFORE, THE FOREGOING EXCLUSIONS MAY NOT APPLY IN ALL CASES. This warranty provides specific legal rights. Other rights which vary from state to state may also apply.

12.Glossary of Terms

Definitions

Local Host
A device which is electrically connected to an XBee. Typically this is a microcontroller connected to the serial pins of the module.

MAC address
A unique network identifier. All network devices are required to have their own unique MAC address. The MAC address is on a sticker on your Digi device server. The number is displayed as 12 hexadecimal digits, usually starting with 00:40:9D.

Network Client
A device which communicates with an XBee through the 802.11 network.

Static IP address assignment
The process of assigning a specific IP address to a device. Contrast with assigning a device through Dynamic Host Configuration Protocol (DHCP), or Automatic Private IP Addressing (APIPA or Auto-IP).

TCP
See Transmission Control Protocol.

Temporal Key Integrity Protocol (TKIP)
Part of the IEEE 802.11i encryption standard for wireless LANs. TKIP is the next generation of the Wired Equivalent Privacy (WEP), which is used to secure 802.11 wireless LANs. TKIP provides per-packet key mixing, a message integrity check and a re-keying mechanism, and addresses several design shortcomings of the original WEP.

Transmission Control Protocol (TCP)
A set of rules (protocol) used along with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. While IP handles the actual delivery of the data, TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.
For example, when an HTML file is sent to you from a Web server, the Transmission Control Protocol (TCP) program layer in that server divides the file into one or more packets, numbers the packets, and then forwards them individually to the IP program layer. Although each packet has the same destination IP address, it may get routed differently through the network. At the other end (the client program in your computer), TCP reassembles the individual packets and waits until they have arrived to forward them to you as a single file.
TCP is known as a connection-oriented protocol, which means that a connection is established and maintained until such time as the message or messages to be exchanged by the application programs at each end have been exchanged. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. In the Open

Systems Interconnection (OSI) communication model, TCP is in layer 4, the Transport Layer.

UDP
See User Datagram Protocol.

User Datagram Protocol (UDP)
A communications protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP). UDP is an alternative to the Transmission Control Protocol (TCP) and, together with IP, is sometimes referred to as UDP/IP. Like the Transmission Control Protocol, UDP uses the Internet Protocol to actually get a data unit (called a datagram) from one computer to another. Unlike TCP, however, UDP does not provide the service of dividing a message into packets (datagrams) and reassembling it at the other end. Specifically, UDP does not provide sequencing of the packets in which the data arrives, nor does it guarantee delivery of data. This means that the application program that uses UDP must be able to make sure that the entire message has arrived and is in the right order. Network applications that want to save processing time because they have very small data units to exchange (and therefore very little message reassembling to do) may prefer UDP to TCP. The Trivial File Transfer Protocol (TFTP) uses UDP instead of TCP.
UDP provides two services not provided by the IP layer. It provides port numbers to help distinguish different user requests and, optionally, a checksum capability to verify that the data arrived intact.

Wi-Fi Protected Access (WPA)
A data encryption/ user authentication method for 802.11 wireless LANs. WPA uses the Temporal Key Integrity Protocol (TKIP).

Wired Equivalency Protocol (WEP)
A security algorithm that uses an RC4 stream cipher, but which has multiple known flaws.

WPA
See Wi-Fi Protected Access.

WPA2/802.11i
WPA with AES-based encryption (CCMP)