# XBee/XBee-PRO S2C 802.15.4

## Radio Frequency (RF) Module

User Guide

# Revision history—90001500

| Revision | Date | Description |
|---|---|---|
| G | May 2018 | Added note on range estimation. Changed IC to ISED. |
| H | June 2018 | Changes to the Active Scan command. |
| J | May 2019 | Removed Brazilian certification information. |
| K | May 2019 | Added FCC publication 996369 related information. |
| L | July 2019 | Added the **NP** command. Added RS-485 support. |

## Trademarks and copyright

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

© 2018 Digi International Inc. All rights reserved.

## Disclaimers

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

## Warranty

To view product warranty information, go to the following website:

www.digi.com/howtobuy/terms

## Customer support

**Gather support information:** Before contacting Digi technical support for help, gather the following information:

- Product name and model
- Product serial number (s)
- Firmware version
- Operating system/browser (if applicable)
- Logs (from time of reported issue)
- Trace (if possible)
- Description of issue
- Steps to reproduce

**Contact Digi technical support**: Digi offers multiple technical support plans and service packages. Contact us at +1 952.912.3444 or visit us at www.digi.com/support.

## Feedback

To provide feedback on this document, email your comments to

techcomm@digi.com

Include the document title and part number (XBee/XBee-PRO S2C 802.15.4 RF Module User Guide, 90001500 E) in the subject line of your email.

# Contents


## XBee/XBee-PRO S2C 802.15.4 RF Module User Guide

Applicable firmware and hardware ..... 11

## Technical specifications

Performance specifications ..... 13
Power requirements ..... 13
General specifications ..... 14
Regulatory conformity summary ..... 14
Serial communication specifications ..... 15
- UART pin assignments ..... 15
- SPI pin assignments ..... 15

GPIO specifications ..... 15

## Hardware

Antenna options ..... 18
Mechanical drawings ..... 18
Mounting considerations ..... 19
Pin signals ..... 20
- Notes ..... 23

Design notes ..... 23
- Power supply design ..... 23
- Board layout ..... 23
- Antenna performance ..... 23
- Keepout area ..... 24
- RF pad version ..... 26

## Configure the XBee/XBee-PRO S2C 802.15.4 RF Module

Software libraries ..... 30
Configure the device using XCTU ..... 30
Over-the-air (OTA) firmware update ..... 30
XBee Network Assistant ..... 30
XBee Multi Programmer ..... 31

## Modes

Serial modes ..... 33

Transparent operating mode ..... 33
API operating mode ..... 33
Command mode ..... 34
Transceiver modes ..... 36
Idle mode ..... 36
Transmit mode ..... 36
Receive mode ..... 36

## Operation

Addressing ..... 38
Send packets to a specific device ..... 38
Addressing modes ..... 38
Encryption ..... 38
Maximum payload ..... 40
Maximum payload rules ..... 40
Maximum payload summary tables ..... 41
Work with Legacy devices ..... 41
Networking ..... 42
MAC Mode configuration ..... 42
XBee retries configuration ..... 43
Transmit status based on MAC mode and XBee retries configurations ..... 43
Peer-to-peer networks ..... 44
Master/slave networks ..... 44
Clear Channel Assessment (CCA) ..... 48
CCA operations ..... 48
Serial interface ..... 49
Select a serial port ..... 49
UART data flow ..... 50
Flow control ..... 51
SPI operation ..... 52
SPI signals ..... 52
SPI parameters ..... 53
SPI and API mode ..... 53
Full duplex operation ..... 53
Slave mode characteristics ..... 54
I/O support ..... 54
Digital I/O line support ..... 55
Analog input ..... 55
On demand I/O sampling ..... 55
Periodic I/O sampling ..... 57
Change Detect I/O sampling ..... 58
Wakeup I/O sampling ..... 59
Sample rate (interval) ..... 59
I/O line passing ..... 59
Output control ..... 60
Sleep support ..... 60
Sleep modes ..... 60
Sleep parameters ..... 61
Sleep current ..... 62
Sleep pins ..... 62
Direct and indirect transmission ..... 62
Node discovery ..... 63
Node discovery ..... 63
Node discovery in compatibility mode ..... 64

Directed node discovery ..... 64
Directed node discovery in compatibility mode ..... 64
Destination Node ..... 64
Remote configuration commands ..... 65
Send a remote command ..... 65
Apply changes on remote devices ..... 65
Remote command responses ..... 65

## AT commands

Special commands ..... 67
WR (Write) ..... 67
RE (Restore Defaults) ..... 67
FR (Software Reset) ..... 67
Networking and security commands ..... 67
C8 (802.15.4 Compatibility) ..... 67
CH (Operating Channel) ..... 69
ID (Network ID) ..... 69
DH (Destination Address High) ..... 69
DL (Destination Address Low) ..... 70
MY (Source Address) ..... 70
SH (Serial Number High) ..... 70
SL (Serial Number Low) ..... 70
MM (MAC Mode) ..... 71
RR (XBee Retries) ..... 71
RN (Random Delay Slots) ..... 72
ND (Network Discovery) ..... 72
NT (Node Discover Timeout) ..... 73
NO (Node Discovery Options) ..... 73
DN (Discover Node) ..... 74
CE (Coordinator Enable) ..... 74
SC (Scan Channels) ..... 75
SD (Scan Duration) ..... 75
A1 (End Device Association) ..... 76
A2 (Coordinator Association) ..... 77
AI (Association Indication) ..... 78
DA (Force Disassociation) ..... 78
FP (Force Poll) ..... 78
AS (Active Scan) ..... 79
ED (Energy Detect) ..... 80
EE (Encryption Enable) ..... 80
KY (AES Encryption Key) ..... 81
NI (Node Identifier) ..... 81
NP (Maximum Packet Payload Bytes) ..... 81
RF interfacing commands ..... 82
PL (TX Power Level) ..... 82
PM (Power Mode) ..... 83
CA (CCA Threshold) ..... 83
Sleep commands ..... 83
SM (Sleep Mode) ..... 84
ST (Time before Sleep) ..... 84
SP (Cyclic Sleep Period) ..... 84
DP (Disassociated Cyclic Sleep Period) ..... 85
SO (Sleep Options) ..... 85
Serial interfacing commands ..... 85

BD (Interface Data Rate) ..... 86
NB (Parity) ..... 87
RO (Packetization Timeout) ..... 87
D7 (DIO7/CTS) ..... 87
D6 (DIO6/RTS) ..... 88
AP (API Enable) ..... 88
I/O settings commands ..... 89
D0 (DIO0/AD0) ..... 89
D1 (DIO1/AD1) ..... 89
D2 (DIO2/AD2) ..... 90
D3 (DIO3/AD3) ..... 90
D4 (DIO4) ..... 91
D5 (DIO5/ASSOCIATED_INDICATOR) ..... 91
D8 (DIO8/SLEEP_REQUEST) ..... 92
P0 (RSSI/PWM0 Configuration) ..... 92
P1 (PWM1 Configuration) ..... 93
P2 (SPI_MISO) ..... 93
M0 (PWM0 Duty Cycle) ..... 94
M1 (PWM1 Duty Cycle) ..... 94
P5 (SPI_MISO) ..... 94
P6 (SPI_MOSI Configuration) ..... 95
P7 (SPI_SSEL ) ..... 95
P8 (SPI_SCLK) ..... 95
P9 (SPI_ATTN) ..... 96
PR (Pull-up/Down Resistor Enable) ..... 96
PD (Pull Up/Down Direction) ..... 97
IU (I/O Output Enable) ..... 97
IT (Samples before TX) ..... 97
IS (Force Sample) ..... 98
IO (Digital Output Level) ..... 98
IC (DIO Change Detect) ..... 98
IR (Sample Rate) ..... 99
RP (RSSI PWM Timer) ..... 99
I/O line passing commands ..... 99
IA (I/O Input Address) ..... 100
T0 (D0 Timeout) ..... 100
T1 (D1 Output Timeout) ..... 100
T2 (D2 Output Timeout) ..... 100
T3 (D3 Output Timeout) ..... 101
T4 (D4 Output Timeout) ..... 101
T5 (D5 Output Timeout) ..... 101
T6 (D6 Output Timeout) ..... 101
T7 (D7 Output Timeout) ..... 101
PT (PWM Output Timeout) ..... 102
Diagnostic commands ..... 102
VR (Firmware Version) ..... 102
VL (Version Long) ..... 102
HV (Hardware Version) ..... 102
DB (Last Packet RSSI) ..... 103
EC (CCA Failures) ..... 103
EA (ACK Failures) ..... 103
DD (Device Type Identifier) ..... 103
Command mode options ..... 104
CT (Command mode Timeout) ..... 104
CN (Exit Command mode) ..... 104

AC (Apply Changes) 104
GT (Guard Times) 105
CC (Command Character) 105

## Operate in API mode

API mode overview 107
API frame specifications 107
API operation (AP parameter = 1) 107
API operation-with escaped characters (AP parameter = 2) 107
Start delimiter 108
Length 108
Frame data 108
Checksum 109
Calculate and verify checksums 109
Escaped characters in API frames 110
Frame descriptions 111
TX Request: 64-bit address frame - 0x00 111
TX Request: 16-bit address - 0x01 112
AT Command frame - 0x08 113
AT Command - Queue Parameter Value frame - 0x09 115
Remote AT Command Request frame - 0x17 116
RX Packet: 64-bit Address frame - 0x80 117
Receive Packet: 16-bit address frame - 0x81 118
RX (Receive) Packet: 64-bit address IO frame- 0x82 119
RX Packet: 16-bit address I/O frame - 0x83 121
AT Command Response frame - 0x88 123
TX Status frame - 0x89 125
Modem Status frame - 0x8A 127
Remote Command Response frame - 0x97 128

## Regulatory information

United States (FCC) 130
OEM labeling requirements 130
FCC notices 130
FCC-approved antennas (2.4 GHz) 132
RF exposure 144
FCC publication 996369 related information 144
Europe (CE) 145
Maximum power and frequency specifications 145
OEM labeling requirements 145
Listen Before Talk requirement 146
Declarations of conformity 146
Antennas 146
ISED (Innovation, Science and Economic Development Canada) 147
Labeling requirements 147
For XBee S2C surface-mount 147
For XBee-PRO S2C surface-mount 147
For XBee S2C through-hole 147
For XBee-PRO S2C through-hole 147
Transmitters for detachable antennas 147
Detachable antenna 148
Australia (RCM) 148

South Korea ..... 148

## Load 802.15.4 firmware on ZB devices

Background ..... 154
Load 802.15.4 firmware ..... 154

## Migrate from XBee through-hole to surface-mount devices

Pin mapping ..... 157
Mount the devices ..... 158

## PCB design and manufacturing

Recommended solder reflow cycle ..... 161
Recommended footprint and keepout ..... 161
Flux and cleaning ..... 163
Rework ..... 163

# XBee/XBee-PRO S2C 802.15.4 RF Module User Guide

XBee/XBee-PRO S2C 802.15.4 RF Modules are embedded solutions providing wireless end-point connectivity to devices. These devices use the IEEE 802.15.4 networking protocol for fast point-to-multipoint or peer-to-peer networking. They are designed for high-throughput applications requiring low latency and predictable communication timing.

There are two footprints for the XBee/XBee-PRO S2C 802.15.4 RF Module hardware: through-hole (TH) and surface-mount (SMT). TH devices include a 20-pin header and require the placement of two 1x10 sockets on the carrier board for mounting the device. SMT devices include 37 pads. They are placed directly on the carrier board, which means they do not require holes or sockets for mounting.

The TH version may be useful for prototyping and production, but we recommend SMT for high-volume applications, as the component can be placed automatically by a pick-and-place machine and you save the cost of a socket on each board.

The XBee/XBee-PRO S2C 802.15.4 RF Module supports the needs of low-cost, low-power wireless sensor networks. The devices require minimal power and provide reliable delivery of data between devices. The devices operate within the ISM 2.4 GHz frequency band.

The XBee/XBee-PRO S2C 802.15.4 RF Module uses S2C hardware and the Silicon Labs EM357 chipset. As the name suggests, the 802.15.4 module is over-the-air compatible with our Legacy 802.15.4 module (S1 hardware), and the TH versions of the new product are also form factor compatible with designs that use the Legacy module.

**Note** OTA capability is only available when **MM** (Mac Mode) = 0 or 3

Applicable firmware and hardware ..... 11

# Applicable firmware and hardware

This manual supports the following firmware:

- 802.15.4 version 20xx

It supports the following hardware:

- XB24C TH
- XB24C SMT
- XBP24C TH
- XBP24C SMT

# Technical specifications

Performance specifications ..... 13
Power requirements ..... 13
General specifications ..... 14
Regulatory conformity summary ..... 14
Serial communication specifications ..... 15
GPIO specifications ..... 15

# Performance specifications

The following table describes the performance specifications for the devices.

**Note** Range figure estimates are based on free-air terrain with limited sources of interference. Actual range will vary based on transmitting power, orientation of transmitter and receiver, height of transmitting antenna, height of receiving antenna, weather conditions, interference sources in the area, and terrain between receiver and transmitter, including indoor and outdoor structures such as walls, trees, buildings, hills, and mountains.

| Specification | XBee value | XBee-PRO value |
|---|---|---|
| Indoor / urban range | Up to 200 ft (60 m) | Up to 300 ft. (90 m) |
| Outdoor RF line-of-sight range | Up to 4000 ft (1200 m) | Up to 2 miles (3200 m) |
| Transmit power output (software selectable) | 6.3 mW (8 dBm), Boost mode[1]<br>3.1 mW (5 dBm), Normal mode<br>Channel 26 max power is 0.3 mW (-5 dBm) | 63 mW (18 dBm)[2] |
| RF data rate | 250,000 b/s | 250,000 b/s |
| Maximum data throughput | Up to 96,000 b/s | Up to 96,000 b/s |
| UART interface data rate | 1200 b/s to 250,000 b/s | 1200 b/s to 250,000 b/s |
| SPI data rate | Up to 5 Mb/s (burst) | Up to 5 Mb/s (burst) |
| Receiver sensitivity | -102 dBm, Boost mode<br>-100 dBm, Normal mode | -101 dBm |

# Power requirements

The following table describes the power requirements for the XBee/XBee-PRO S2C 802.15.4 RF Module.

| Specification | XBee | XBee-PRO |
|---|---|---|
| Supply voltage | 2.1 - 3.6 V | 2.7 - 3.6 V |
| Transmit current (typical, VCC = 3.3 V) | 45 mA (8 dBm, Boost mode)<br>33 mA (5 dBm, Normal mode) | 120 mA (18 dBm) |
| Idle / receive current (typical, VCC = 3.3 V) | 31 mA (Boost mode)<br>28 mA (Normal mode) | 31 mA |
| Power-down current | <1 uA @ 25C | <1 uA @ 25C |

[1]Boost mode enabled by default; see PM (Power Mode).

[2]See Regulatory information for region-specific certification requirements.

# General specifications

The following table describes the general specifications for the devices.

| Specification | XBee | XBee-PRO |
|---|---|---|
| Operating frequency | ISM 2.4 GHz | |
| Supported channels | 11 - 26 | 12 - 23 |
| Form factor | TH: 2.438 x 2.761 cm (0.960 x 1.087 in)<br>SMT: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in) | TH: 2.438 x 3.294 cm (0.960 x 1.297 in)<br>SMT: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in) |
| Operating temperature | -40 to 85 ºC (industrial) | |
| Antenna options | TH: PCB antenna, U.FL connector, RPSMA connector, or integrated wire<br>SMT: RF pad, PCB antenna, or U.FL connector | |

# Regulatory conformity summary

This table describes the agency approvals for the devices.

| Country | XBee (surface-mount) | XBee-PRO (surface-mount) | XBee (through-hole) | XBee-PRO (through-hole) |
|---|---|---|---|---|
| United States (FCC Part 15.247) | FCC ID: MCQ-XBS2C | FCC ID: MCQ-PS2CSM | FCC ID: MCQ-S2CTH | FCC ID: MCQ-PS2CTH |
| Innovation, Science and Economic Development Canada (ISED) | IC: 1846A-XBS2C | IC: 1846A-PS2CSM | IC: 1846A-S2CTH | IC: 1846A-PS2CTH |
| FCC/IC test transmit power output range | -26 to +8 dBm | -0.7 to +19.4 dBm | -26 to +8 dBm | +1 to +19 dBm |
| Europe (CE) | Yes | - | Yes | - |
| Australia | RCM | RCM | RCM | RCM |
| Japan | R201WW10215369 | | R210- 105563 | |
| South Korea | MSIP-CRM-DIG-XBee-S2C | | MSIP-CRM-DIG-XBee-S2C-TH | |
| RoHS | Compliant | | | |

# Serial communication specifications

The XBee/XBee-PRO S2C 802.15.4 RF Module supports both Universal Asynchronous Receiver / Transmitter (UART) and Serial Peripheral Interface (SPI) serial connections.

## UART pin assignments

The SC1 (Serial Communication Port 1) of the Ember 357 is connected to the UART port. The following table provides the UART pin assignments.

| Specifications | Module pin number | |
|---|---|---|
| UART pins | XBee (surface-mount) | XBee (through-hole) |
| DOUT | 3 | 2 |
| DIN / $\overline{\text{CONFIG}}$ | 4 | 3 |
| $\overline{\text{CTS}}$ / DIO7 | 25 | 12 |
| $\overline{\text{RTS}}$ / DIO6 | 29 | 16 |

## SPI pin assignments

The SC2 (Serial Communication Port 2) of the Ember 357 is connected to the SPI port.

| Specifications | Module pin number | |
|---|---|---|
| SPI pins | XBee (surface-mount) | XBee (through-hole) |
| SPI_SCLK | 14 | 18 |
| SPI_$\overline{\text{SSEL}}$ | 15 | 17 |
| SPI_MOSI | 16 | 11 |
| SPI_MISO | 17 | 4 |
| SPI_$\overline{\text{ATTN}}$ | 12 | 19 |

# GPIO specifications

XBee/XBee-PRO S2C 802.15.4 RF Modules have 15 General Purpose Input / Output (GPIO) ports available. The exact list depends on the device configuration, as some GPIO pads are used for purposes such as serial communication.

| GPIO Electrical Specification | Value |
|---|---|
| Low Schmitt switching threshold | 0.42 - 0.5 x VCC |
| High Schmitt switching threshold | 0.62 - 0.8 x VCC |
| Input current for logic 0 | -0.5 µA |

| GPIO Electrical Specification | Value |
|---|---|
| Input current for logic 1 | 0.5 µA |
| Input pull-up resistor value | 29 kΩ |
| Input pull-down resistor value | 29 kΩ |
| Output voltage for logic 0 | 0.18 x VCC (maximum) |
| Output voltage for logic 1 | 0.82 x VCC (minimum) |
| Output source/sink current for pad numbers 3, 4, 5, 10, 12, 14, 15, 16, 17, 25, 26, 28, 29, 30, and 32 on the SMT modules | 4 mA |
| Output source/sink current for pin numbers 2, 3, 4, 9, 12, 13, 15, 16, 17, and 19 on the TH modules | 4 mA |
| Output source/sink current for pad numbers 7, 8, 24, 31, and 33 on the SMT modules | 8 mA |
| Output source/sink current for pin numbers 6, 7, 11, 18, and 20 on the TH modules | 8 mA |
| Total output current (for GPIO pads) | 40 mA |

# Hardware

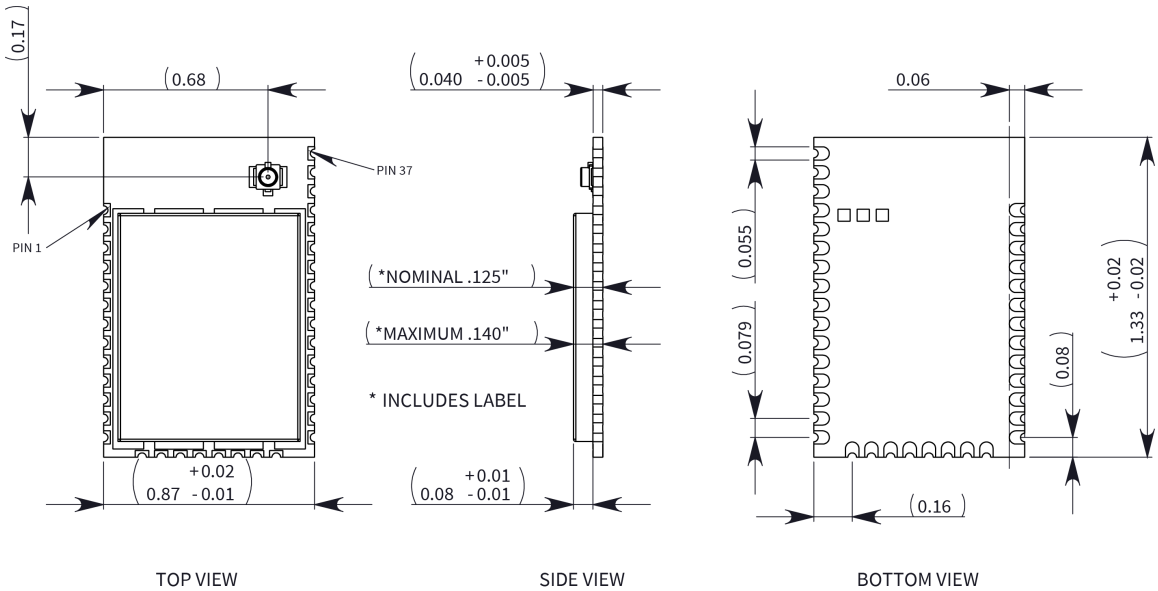
Antenna options ..... 18
Mechanical drawings ..... 18
Mounting considerations ..... 19
Pin signals ..... 20
Design notes ..... 23

## Antenna options

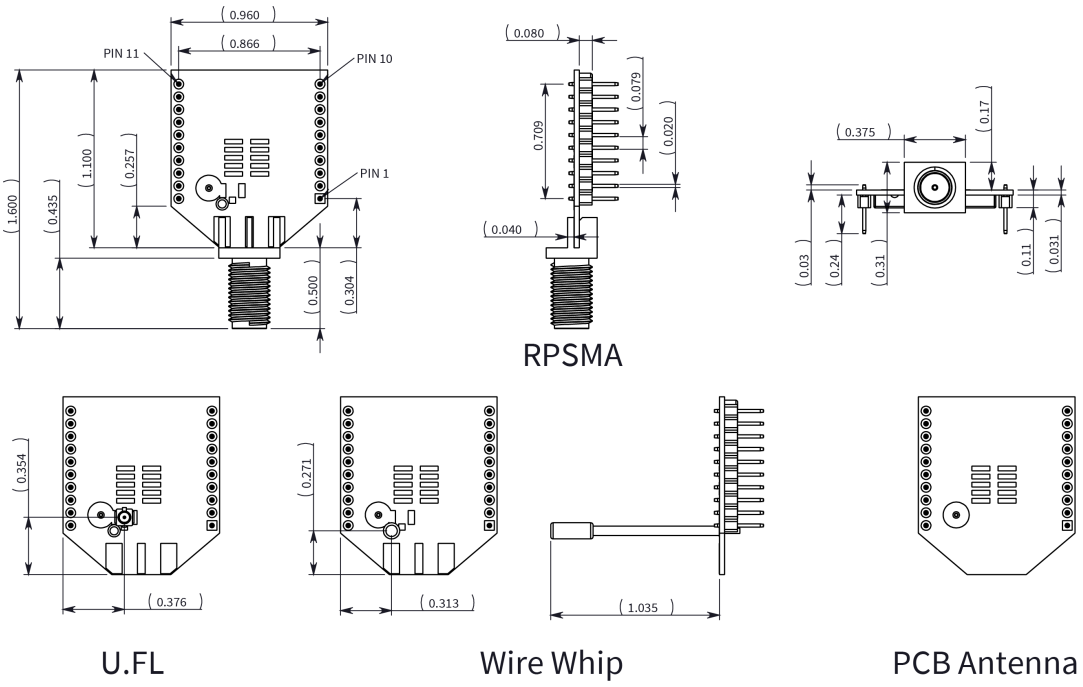
The ranges specified are typical for the integrated whip (1.5 dBi) and dipole (2.1 dBi) antennas. The printed circuit board (PCB) antenna option provides advantages in its form factor; however, it typically yields shorter range than the whip and dipole antenna options when transmitting outdoors. For more information, see XBee and XBee-PRO OEM RF Module Antenna Considerations Application Note.

## Mechanical drawings

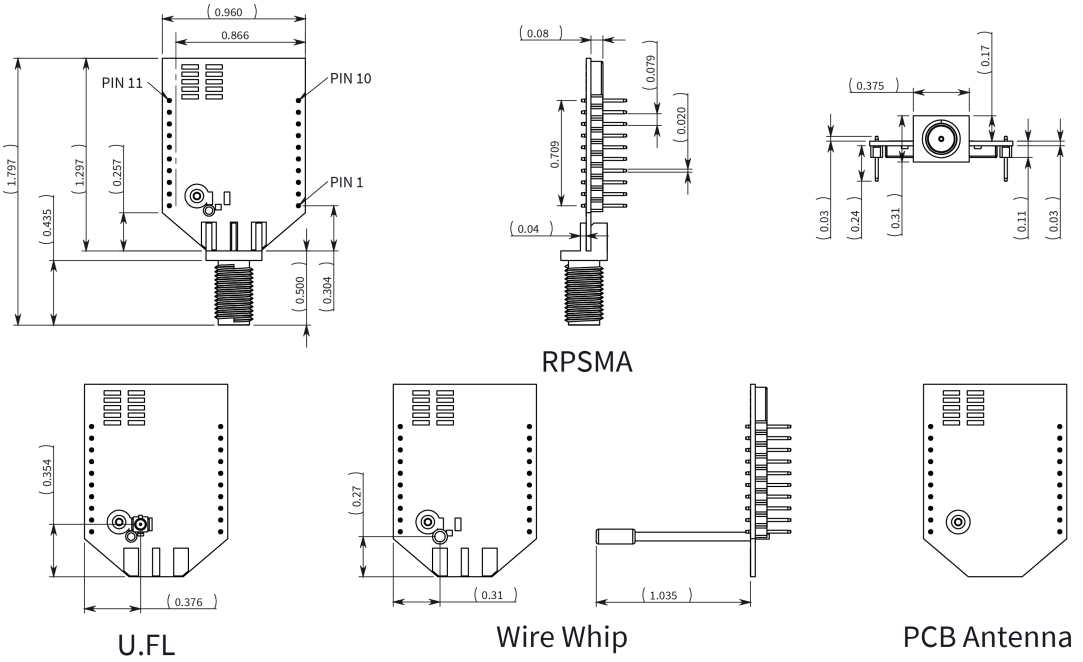
The following mechanical drawings of the XBee/XBee-PRO S2C 802.15.4 RF Module show all dimensions in inches. The first drawing shows the surface-mount device (antenna options not shown).




The following drawings show the standard (non-PRO) through-hole device.



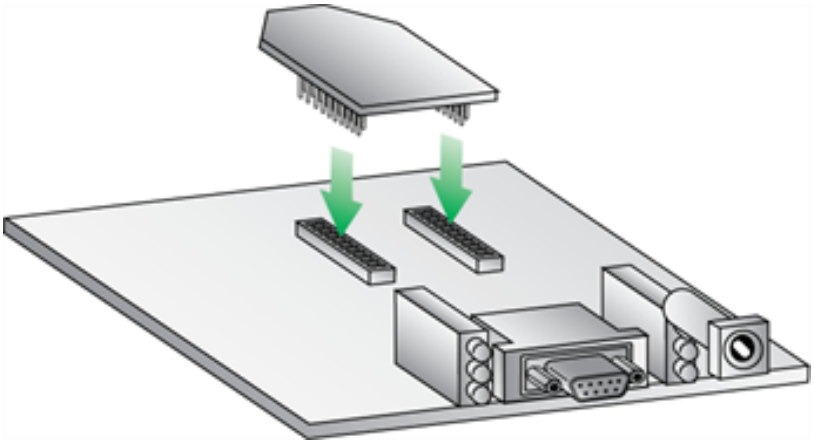

The following drawings show the XBee-PRO through-hole device.




# Mounting considerations

We design the through-hole module to mount into a receptacle so that you do not have to solder the module when you mount it to a board. The development kits may contain RS-232 and USB interface boards that use two 20-pin receptacles to receive modules.

The following illustration shows the module mounting into the receptacle on the RS-232 interface board.



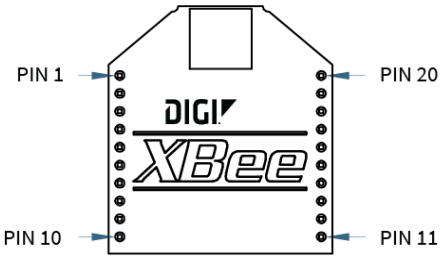
Century Interconnect and Samtec manufacture the 2 x 10 pin 2 mm spacing receptacles on Digi development boards. Several other manufacturers provide comparable mounting solutions; we currently use the following receptacles:

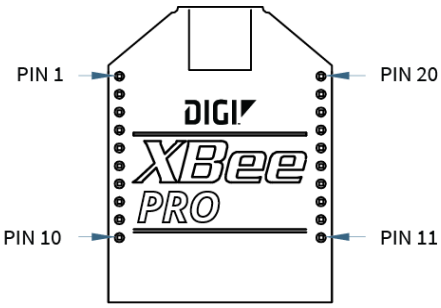
- Through-hole single-row receptacles: Samtec part number: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles: Century Interconnect part number: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles: Samtec part number: SMM-110-02-SM-S

**Note** We recommend that you print an outline of the module on the board to indicate the correct orientation for mounting the module.

# Pin signals

The following image shows the pin numbers; it shows the device's top sides, the shields are on the bottom.




The following table shows the pin assignments for the through-hole device. In the table, low-asserted signals have a horizontal line above signal name.

| Pin | Name | Direction | Description |
|---|---|---|---|
| 1 | VCC | - | Power supply |

| Pin | Name | Direction | Description |
|---|---|---|---|
| 2 | DOUT | Output | UART data out |
| 3 | DIN/$\overline{\text{CONFIG}}$ | Input | UART data In |
| 4 | SPI_MISO | Output | Serial Peripheral Interface (SPI) Data Out |
| 5 | $\overline{\text{RESET}}$ | Input | Module reset (reset pulse must be at least 200 ns). This must be driven as an open drain/collector. The device drives this line low when a reset occurs. Never drive this line high. |
| 6 | PWM0/RSSI PWM | Output | PWM output 0 / RX signal strength indicator |
| 7 | PWM1 | Output | PWM output 1 |
| 8 | [Reserved] | - | Do not connect |
| 9 | DI8/$\overline{\text{SLEEP_RQ}}$/$\overline{\text{DTR}}$ | Input | Pin sleep control line or digital input 8 |
| 10 | GND | - | Ground |
| 11 | DIO4/SPI_MOSI | Both | Digital I/O 4 / SPI Data In |
| 12 | DIO7/$\overline{\text{CTS}}$ | Both | Digital I/O 7 / Clear-to-send flow control |
| 13 | ON/$\overline{\text{SLEEP}}$ | Output | Device sleep status indicator |
| 14 | $V_{REF}$ | - | Feature not supported on this device. Used on other XBee devices for analog voltage reference. |
| 15 | DIO5/ASSOC | Both | Digital I/O 5 / Associated indicator |
| 16 | DIO6/$\overline{\text{RTS}}$ | Both | Digital I/O 6 / Request-to-send flow control |
| 17 | $\overline{\text{DIO3}}$/AD3/SPI_$\overline{\text{SSEL}}$ | Both | Digital I/O 3 / Analog input 3 / SPI select |
| 18 | DIO2/AD2/SPI_CLK | Both | Digital I/O 2 / Analog input 2 / SPI clock |
| 19 | $\overline{\text{DIO1}}$/AD1/SPI_$\overline{\text{ATTN}}$ | Both | Digital I/O 1 / Analog input 1 / SPI Attention |
| 20 | DIO0/AD0 | Both | Digital I/O 0 / Analog input 0 |

The following table shows the pin assignments for the surface-mount device.

| Pin | Name | Direction | Function |
|---|---|---|---|
| 1 | GND | - | Ground |
| 2 | VCC | - | Power supply |
| 3 | DOUT | Output | UART data out |

| Pin | Name | Direction | Function |
|---|---|---|---|
| 4 | DIN/$\overline{\text{CONFIG}}$ | Input | UART data in |
| 5 | [Reserved] | Output | Do not connect |
| 6 | $\overline{\text{RESET}}$ | Input | Module reset (reset pulse must be at least 200 ns). This must be driven as an open drain/collector. The device drives this line low when a reset occurs. Never drive this line high. |
| 7 | PWM0/RSSI PWM | Output | PWM output 0 / RX signal strength indicator |
| 8 | PWM1 | Output | PWM output 1 |
| 9 | [Reserved] | - | Do not connect |
| 10 | DI8/SLEEP_RQ/$\overline{\text{DTR}}$ | Input | Pin sleep control line or digital input 8 |
| 11 | GND | - | Ground |
| 12 | SPI_ $\overline{\text{ATTN}}$/$\overline{\text{BOOTMODE}}$ | Output | SPI Attention. Do not tie low on reset. |
| 13 | GND | - | Ground |
| 14 | SPI_CLK | Input | SPI clock |
| 15 | SPI_$\overline{\text{SSEL}}$ | Input | SPI select |
| 16 | SPI_MOSI | Input | SPI Data In |
| 17 | SPI_MISO | Output | SPI Data 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 | Digital I/O 4 |
| 25 | DIO7/$\overline{\text{CTS}}$ | Both | Digital I/O 7 / Clear-to-send flow control |
| 26 | ON/$\overline{\text{SLEEP}}$ | Output | Device sleep status indicator |
| 27 | $V_{REF}$ | - | Feature not supported on this device. Used on other XBee devices for analog voltage reference. |
| 28 | DIO5/ASSOC | Both | Digital I/O 5 / Associated indicator |
| 29 | DIO6/$\overline{\text{RTS}}$ | Both | Digital I/O 6 / Request-to-send flow control |

| Pin | Name | Direction | Function |
|---|---|---|---|
| 30 | DIO3/AD3 | Both | Digital I/O 3 / Analog input 3 |
| 31 | DIO2/AD2 | Both | Digital I/O 2 / Analog input 2 |
| 32 | DIO1/AD1 | Both | Digital I/O 1 / Analog input 1 |
| 33 | DIO0/AD0 | Both | Digital I/O 0 / Analog input 0 |
| 34 | [Reserved] | - | Do not connect |
| 35 | GND | - | Ground |
| 36 | RF | Both | RF connection |
| 37 | [Reserved] | - | Do not connect |

### Notes

Minimum connections: VCC, GND, DOUT and DIN.

Minimum connections for updating firmware: VCC, GND, DIN, DOUT, RTS and DTR.

The table specifies signal direction with respect to the device.

The device includes a 50 kΩ pull-up resistor attached to RESET.

Use the **PR** (Pull-up/Down Resistor Enable) command to configure several of the input pull-ups.

You can connect other pins to external circuitry for convenience of operation including the Associate LED pin (pin 15). The Associate LED flashes differently depending on the state of the device.

Leave any unused pins disconnected.

## Design notes

The following guidelines help to ensure a robust design.

### Power supply design

A poor power supply can lead to poor device performance, especially if you do not keep the supply voltage within tolerance or if it is excessively noisy. To help reduce noise, place a 1.0 µF and 8.2 pF capacitor as near as possible to pin 1 on the PCB. If you are using a switching regulator for the power supply, switch the frequencies above 500 kHz. Limit the power supply ripple to a maximum 100 mV peak to peak.

### Board layout

We design XBee devices to be self sufficient and have minimal sensitivity to nearby processors, crystals or other printed circuit board (PCB) components. Keep power and ground traces thicker than signal traces and make sure that they are able to comfortably support the maximum current specifications. There are no other special PCB design considerations to integrate XBee devices, with the exception of antennas.

### Antenna performance

Antenna location is important for optimal performance. The following suggestions help you achieve optimal antenna performance. Point the antenna up vertically (upright). Antennas radiate and receive

the best signal perpendicular to the direction they point, so a vertical antenna's omnidirectional radiation pattern is strongest across the horizon.

Position the antennas away from metal objects whenever possible. Metal objects between the transmitter and receiver can block the radiation path or reduce the transmission distance. Objects that are often overlooked include:

- metal poles
- metal studs
- structure beams
- concrete, which is usually reinforced with metal rods

If you place the device inside a metal enclosure, use an external antenna. Common objects that have metal enclosures include:

- vehicles
- elevators
- ventilation ducts
- refrigerators
- microwave ovens
- batteries
- tall electrolytic capacitors

Do not place XBee devices with the chip or integrated PCB antenna inside a metal enclosure.

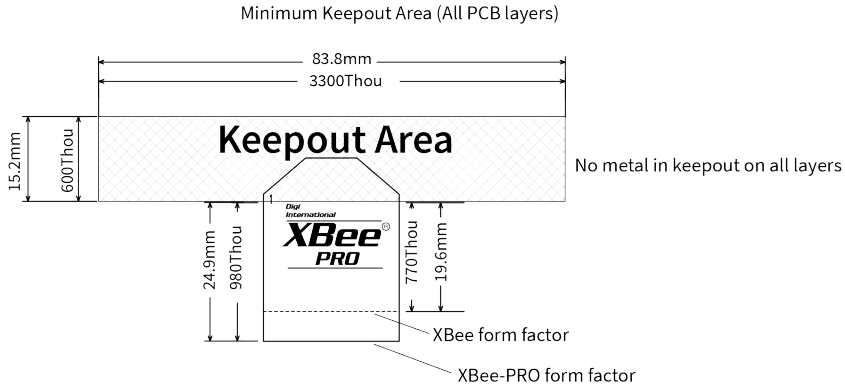
Do not place any ground planes or metal objects above or below the antenna.

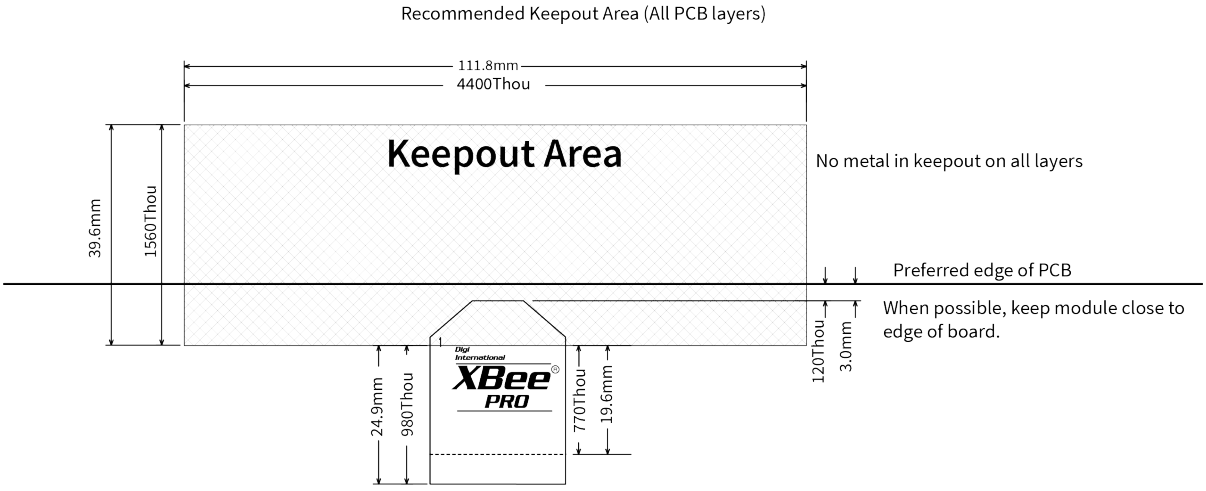
For the best results, mount the device at the edge of the host PCB. Ensure that the ground, power, and signal planes are vacant immediately below the antenna section.

## Keepout area

We recommend that you allow a “keepout” area, which the following drawings show.

### *Through-hole keepout*



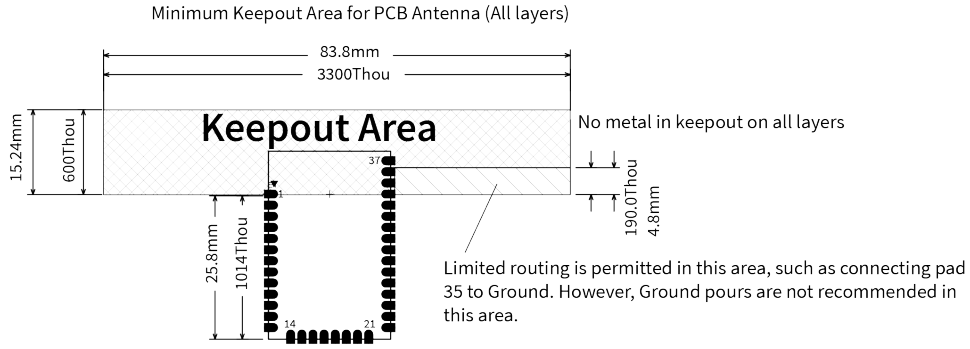



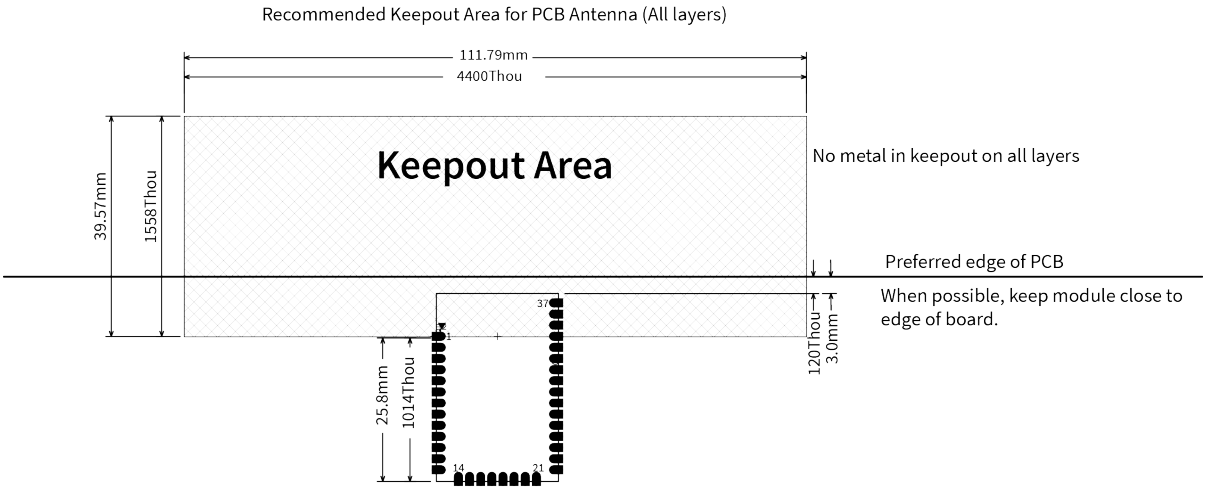

The antenna performance improves with a larger keepout area

**Notes**

1. We recommend non-metal enclosures. For metal enclosures, use an external antenna.
2. Keep metal chassis or mounting structures in the keepout area at least 2.54 cm (1 in) from the antenna.
3. Maximize the distance between the antenna and metal objects that might be mounted in the 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.

### *Surface-mount keepout*




The antenna performance improves with a larger keepout area

## RF pad version

The RF pad is a soldered antenna connection on the surface-mount device. The RF signal travels from pin 36 on the module to the antenna through a single ended RF transmission line on the PCB. This line should have a controlled impedance of 50 Ω.

For the transmission line, we recommend either a microstrip or coplanar waveguide trace on the PCB. We provide a microstrip example below, because it is simpler to design and generally requires less area on the host PCB than coplanar waveguide.

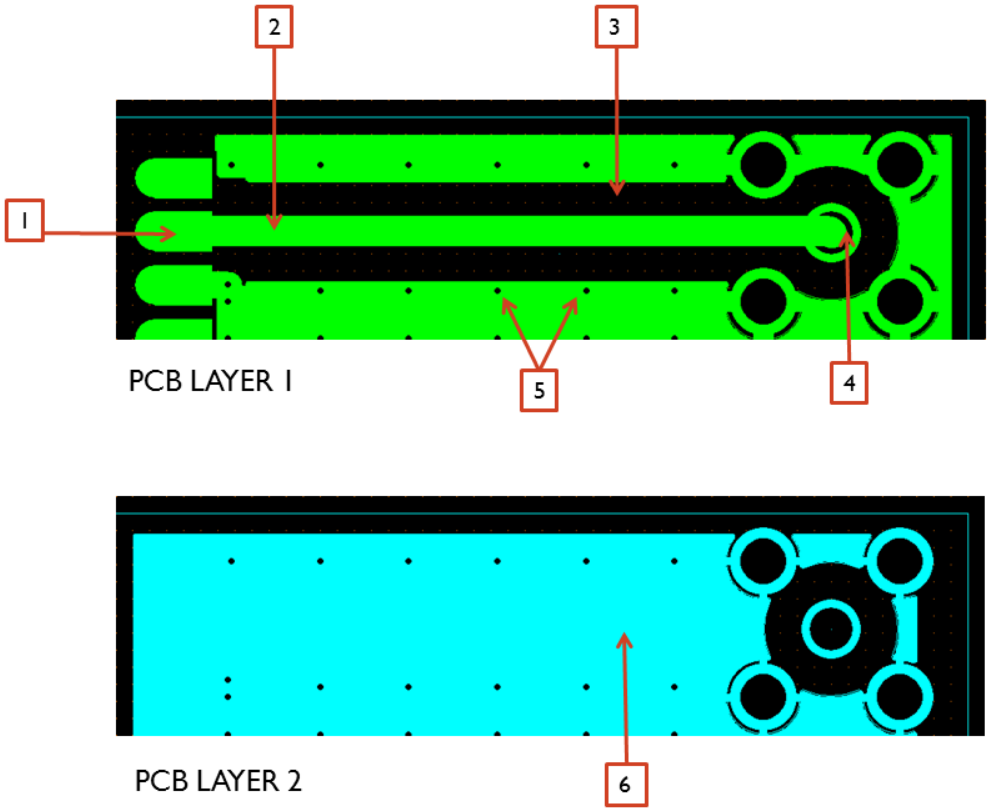
We do not recommend using a stripline RF trace because that requires routing the RF trace to an inner PCB layer, and via transitions can introduce matching and performance problems.

The following figure shows a layout example of a microstrip connecting an RF pad module to a through-hole RPSMA RF connector.

- The top two layers of the PCB have a controlled thickness dielectric material in between. The second layer has a ground plane which runs underneath the entire RF pad area. This ground plane is a distance *d*, the thickness of the dielectric, below the top layer.

- The top layer has an RF trace running from pin 36 of the device to the RF pin of the RPSMA connector. The RF trace's width determines the impedance of the transmission line with relation to the ground plane. Many online tools can estimate this value, although you should consult the PCB manufacturer for the exact width. Assuming *d* = 0.025 in, and that the dielectric has a relative permittivity of 4.4, the width in this example will be approximately 0.045 in for a 50 Ω trace. This trace width is a good fit with the module footprint's 0.060 in pad width.

We do not recommend using a trace wider than the pad width, and using a very narrow trace can cause unwanted RF loss. You can minimize the length of the trace by placing the RPSMA jack close to the module. All of the grounds on the jack and the module are connected to the ground planes directly or through closely placed vias. Space any ground fill on the top layer at least twice the distance *d* (in this case, at least 0.050 in) from the microstrip to minimize their interaction.




| Number | Description |
|---|---|
| 1 | XBee surface-mount pin 36 |
| 2 | 50 Ω microstrip trace |
| 3 | Back off ground fill at least twice the distance between layers 1 and 2 |

| Number | Description |
|---|---|
| 4 | RF connector |
| 5 | Stitch vias near the edges of the ground plane |
| 6 | Pour a solid ground plane under the RF trace on the reference layer |

Implementing these design suggestions helps ensure that the RF pad device performs to specifications.

# Configure the XBee/XBee-PRO S2C 802.15.4 RF Module

Software libraries ..... 30
Configure the device using XCTU ..... 30
Over-the-air (OTA) firmware update ..... 30
XBee Network Assistant ..... 30
XBee Multi Programmer ..... 31

## Software libraries

One way to communicate with the XBee/XBee-PRO S2C 802.15.4 RF Module is by using a software library. The libraries available for use with the XBee/XBee-PRO S2C 802.15.4 RF Module include:

- XBee Java library
- XBee Python library

The XBee Java Library is a Java API. The package includes the XBee library, its source code and a collection of samples that help you develop Java applications to communicate with your XBee devices.

The XBee Python Library is a Python API that dramatically reduces the time to market of XBee projects developed in Python and facilitates the development of these types of applications, making it an easy process.

## Configure the device using XCTU

XBee Configuration and Test Utility (XCTU) is a multi-platform program that enables users to interact with Digi radio frequency (RF) devices through a graphical interface. The application includes built-in tools that make it easy to set up, configure, and test Digi RF devices.

For instructions on downloading and using XCTU, see the XCTU User Guide.

Click **Discover devices** and follow the instructions. XCTU should discover the connected XBee/XBee-PRO S2C 802.15.4 RF Modules using the provided settings.

Click **Add selected devices**.The devices appear in the **Radio Modules** list. You can click a module to view and configure its individual settings. For more information on these items, see AT commands.

## Over-the-air (OTA) firmware update

The XBee/XBee-PRO S2C 802.15.4 RF Module supports OTA firmware updates using XCTU version 6.3.0 or higher. For instructions on performing an OTA firmware update with XCTU, see How to update the firmware of your modules in the XCTU User Guide.

OTA capability is only available when **MM** (Mac Mode) = 0 or 3

## XBee Network Assistant

The XBee Network Assistant is an application designed to inspect and manage RF networks created by Digi XBee devices. Features include:

- Join and inspect any nearby XBee network to get detailed information about all the nodes it contains.
- Update the configuration of all the nodes of the network, specific groups, or single devices based on configuration profiles.
- Geo-locate your network devices or place them in custom maps and get information about the connections between them.
- Export the network you are inspecting and import it later to continue working or work offline.
- Use automatic application updates to keep you up to date with the latest version of the tool.

See the *XBee Network Assistant User Guide* for more information.

To install the XBee Network Assistant:

1. Navigate to digi.com/xbeenetworkassistant.
2. Click **General Diagnostics, Utilities and MIBs**.
3. Click the **XBee Network Assistant - Windows x86** link.
4. When the file finishes downloading, run the executable file and follow the steps in the XBee Network Assistant Setup Wizard.

## XBee Multi Programmer

The XBee Multi Programmer is a combination of hardware and software that enables partners and distributors to program multiple Digi Radio frequency (RF) devices simultaneously. It provides a fast and easy way to prepare devices for distribution or large networks deployment.

The XBee Multi Programmer board is an enclosed hardware component that allows you to program up to six RF modules thanks to its six external XBee sockets. The XBee Multi Programmer application communicates with the boards and allows you to set up and execute programming sessions. Some of the features include:

- Each XBee Multi Programmer board allows you to program up to six devices simultaneously. Connect more boards to increase the programming concurrency.
- Different board variants cover all the XBee form factors to program almost any Digi RF device.

Download the XBee Multi Programmer application from: digi.com/support/productdetail?pid=5641

See the *XBee Multi Programmer User Guide* for more information.

# Modes

Serial modes ..... 33
Transceiver modes ..... 36

# Serial modes

The firmware operates in several different modes. Two top-level modes establish how the device communicates with other devices through its serial interface: Transparent operating mode and API operating mode. Use the **AP** command to choose Serial mode. XBee/XBee-PRO S2C 802.15.4 RF Modules use Transparent operation as the default serial mode.

The following modes describe how the serial port sends and receives data.

## Transparent operating mode

Devices operate in this mode by default. The device acts as a serial line replacement when it is in Transparent operating mode. The device queues all UART data it receives through the DIN pin for RF transmission. When a device receives RF data, it sends the data out through the DOUT pin. You can set the configuration parameters using Command mode.

**Note** Transparent operating mode is not available when using the SPI interface; see SPI operation.

The device buffers data in the serial receive buffer until one of the following causes the data to be packetized and transmitted:

- The device receives no serial characters for the amount of time determined by the **RO** (Packetization Timeout) parameter. If **RO** = 0, packetization begins when a character is received.
- The device receives the Command Mode Sequence (**GT** + **CC** + **GT**). Any character buffered in the serial receive buffer before the sequence is transmitted.
- The device receives the maximum number of characters that fits in an RF packet (100 bytes).

### *Serial-to-RF packetization*

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

1. No serial characters are received for the amount of time determined by the **RO** (Packetization Timeout) parameter. If **RO** = 0, packetization begins when a character is received.
2. The maximum number of characters that will fit in an RF packet (100) is received. The maximum payload depends on whether you use Compatibility mode or not. If you use it, the maximum payload is 100 characters, but if you do not use it, depending on encryption, the App header and addressing, the payload can be larger.
3. The Command Mode Sequence (**GT** + **CC** + **GT**) is received. Any character buffered in the DI buffer before the sequence is transmitted.

If the device cannot immediately transmit (for instance, if it is already receiving RF data), the serial data is stored in the DI Buffer. The data is packetized and sent at any **RO** timeout or when 100 bytes (maximum packet size without Compatibility mode) are received.

If the DI buffer becomes full, hardware flow control must be implemented in order to prevent overflow (loss of data between the host and device).

## API operating mode

Application programming interface (API) operating mode is an alternative to Transparent mode. It is helpful in managing larger networks and is more appropriate for performing tasks such as collecting data from multiple locations or controlling multiple devices remotely. API mode is a frame-based

protocol that allows you to direct data on a packet basis. It can be particularly useful in large networks where you need control over the operation of the radio network or when you need to know which node a data packet is from. The device communicates UART or SPI data in packets, also known as API frames. This mode allows for structured communications with serial devices.

For more information, see API mode overview.

## Command mode

Command mode is a state in which the firmware interprets incoming characters as commands. It allows you to modify the device's configuration using parameters you can set using AT commands. When you want to read or set any parameter of the XBee/XBee-PRO S2C 802.15.4 RF Module using this mode, you have to send an AT command. Every AT command starts with the letters **AT** followed by the two characters that identify the command and then by some optional configuration values.

The operating modes of the XBee/XBee-PRO S2C 802.15.4 RF Module are controlled by the AP (API Enable) setting, but Command mode is always available as a mode the device can enter while configured for any of the operating modes.

Command mode is available on the UART interface for all operating modes. You cannot use the SPI interface to enter Command mode.

### *Enter Command mode*

To get a device to switch into Command mode, you must issue the following sequence: **+++** within one second. There must be at least one second preceding and following the **+++** sequence. Both the command character (**CC**) and the silence before and after the sequence (**GT**) are configurable. When the entrance criteria are met the device responds with **OK\r** on UART signifying that it has entered Command mode successfully and is ready to start processing AT commands.

If configured to operate in Transparent operating mode, when entering Command mode the XBee/XBee-PRO S2C 802.15.4 RF Module knows to stop sending data and start accepting commands locally.

**Note** Do not press **Return** or **Enter** after typing **+++** because it interrupts the guard time silence and prevents you from entering Command mode.

When the device is in Command mode, it listens for user input and is able to receive AT commands on the UART. If **CT** time (default is 10 seconds) passes without any user input, the device drops out of Command mode and returns to the previous operating mode. You can force the device to leave Command mode by sending CN (Exit Command mode).

You can customize the command character, the guard times and the timeout in the device's configuration settings. For more information, see CC (Command Character), CT (Command mode Timeout) and GT (Guard Times).

### *Troubleshooting*

Failure to enter Command mode is often due to baud rate mismatch. Ensure that the baud rate of the connection matches the baud rate of the device. By default, BD (Interface Data Rate) = **3** (9600 b/s).

There are two alternative ways to enter Command mode:

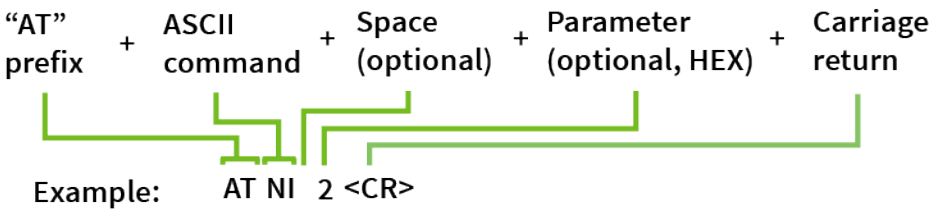
- A serial break for six seconds enters Command mode. You can issue the "break" command from a serial console, it is often a button or menu item.
- Asserting DIN (serial break) upon power up or reset enters Command mode. XCTU guides you through a reset and automatically issues the break when needed.

Both of these methods temporarily set the device's baud rate to 9600 and return an **OK** on the UART to indicate that Command mode is active. When Command mode exits, the device returns to normal operation at the baud rate that **BD** is set to.

### *Send AT commands*

Once the device enters Command mode, use the syntax in the following figure to send AT commands. Every AT command starts with the letters **AT**, which stands for "attention." The AT is followed by two characters that indicate which command is being issued, then by some optional configuration values.

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




The preceding example changes NI (Node Identifier) to **My XBee**.

#### Multiple AT commands

You can send multiple AT commands at a time when they are separated by a comma in Command mode; for example, **ATNIMy XBee,AC<cr>**.

The preceding example changes the **NI (Node Identifier)** to **My XBee** and makes the setting active through AC (Apply Changes).

#### Parameter format

Refer to the list of AT commands for the format of individual AT command parameters. Valid formats for hexidecimal values include with or without a leading **0x** for example **FFFF** or **0xFFFF**.

### *Response to AT commands*

When using AT commands to set parameters the XBee/XBee-PRO S2C 802.15.4 RF Module responds with **OK<cr>** if successful and **ERROR<cr>** if not.

For devices with a file system:

**ATAP1<cr>**

**OK<cr>**

When reading parameters, the device returns the current parameter value instead of an **OK** message.

**ATAP<cr>**

**1<cr>**

### *Apply command changes*

Any changes you make to the configuration command registers using AT commands do not take effect until you apply the changes. For example, if you send the **BD** command to change the baud rate, the actual baud rate does not change until you apply the changes. To apply changes:

1. Send AC (Apply Changes).
2. Send WR (Write).
   or:
3. Exit Command mode.

### *Make command changes permanent*

Send a WR (Write) command to save the changes. **WR** writes parameter values to non-volatile memory so that parameter modifications persist through subsequent resets.

Send as RE (Restore Defaults) to wipe settings saved using **WR** back to their factory defaults.

**Note** You still have to use **WR** to save the changes enacted with **RE**.

### *Exit Command mode*

1. Send CN (Exit Command mode) followed by a carriage return.
   or:
2. If the device does not receive any valid AT commands within the time specified by CT (Command mode Timeout), it returns to Transparent or API mode. The default Command mode timeout is 10 seconds.

For an example of programming the device using AT Commands and descriptions of each configurable parameter, see AT commands.

# Transceiver modes

The following modes describe how the transceiver sends and receives over-the-air (OTA) data.

## Idle mode

When not receiving or transmitting data, the device is in Idle mode. During Idle mode, the device listens for valid data on both the RF and serial ports.

## Transmit mode

Transmit mode is the mode in which the device is transmitting data. This typically happens when data is received from the serial port.

## Receive mode

This is the default mode for the XBee/XBee-PRO S2C 802.15.4 RF Module. The device is in Receive mode when it is not transmitting data. If a destination node receives a valid RF packet, the destination node transfers the data to its serial transmit buffer.

# Operation

Addressing ..... 38
Encryption ..... 38
Maximum payload ..... 40
Networking ..... 42
Clear Channel Assessment (CCA) ..... 48
Serial interface ..... 49
SPI operation ..... 52
I/O support ..... 54
Sleep support ..... 60
Node discovery ..... 63
Remote configuration commands ..... 65

# Addressing

Every RF data packet sent over-the-air contains a Source Address and Destination Address field in its header. The XBee/XBee-PRO S2C 802.15.4 RF Module conforms to the 802.15.4 specification and supports both short 16-bit addresses and long 64-bit addresses. A unique 64-bit IEEE source address is assigned at the factory and can be read with the **SL** (Serial Number Low) and **SH** (Serial Number High) commands. You must manually configure short addressing. A device uses its unique 64-bit address as its Source Address if its **MY** (16-bit Source Address) value is 0xFFFF or 0xFFFE.

## Send packets to a specific device

To send a packet to a specific device using 64-bit addressing:

- Set the Destination Address (**DL** + **DH**) of the sender to match the Source Address (**SL** + **SH**) of the intended destination device.

To send a packet to a specific device using 16-bit addressing:

1. Set the **DL** parameter to equal the **MY** parameter of the intended destination device.
2. Set the **DH** parameter to 0.

## Addressing modes

802.15.4 frames have a source address, a destination address, and a destination PAN ID in the over-the-air (OTA) frame. The source and destination addresses may be either long or short and the destination address may be either a unicast or a broadcast. The destination PAN ID is short and it may also be the broadcast PAN ID.

In Transparent mode, the destination address is set by the **DH** and **DL** parameters, but, in API mode, it is set by the TX Request: 64-bit address (0x00) or TX Request: 16-bit Address (0x01) frames. In either Transparent mode or API mode, the destination PAN ID is set with the **ID** parameter, and the source address is set with the **MY** parameter.

### *Broadcasts and unicasts*

Broadcasts are identified by the 16-bit short address of 0xFFFF. Any other destination address is considered a unicast and is a candidate for acknowledgments, if enabled.

### *Broadcast PAN ID*

The Broadcast PAN ID is also 0xFFFF. Its effect is to traverse all PANs in the vicinity. Typically, this only makes sense during association time when sending beacon requests to find PAN IDs.

### *Short and long addresses*

A short address is 16 bits and a long address is 64 bits. The short address is set with the **MY** parameter. If the short address is 0xFFFE or 0xFFFF, then the address of the device is long and it is the serial number of the device as read by the **SH** and **SL** parameters.

**Note** When an end device associates to a coordinator, it is assigned the short address of 0xFFFE.

# Encryption

The XBee/XBee-PRO S2C 802.15.4 RF Module supports AES 128-bit encryption. 128-bit encryption refers to the length of the encryption key entered with the **KY** command (128 bits = 16 bytes). The

802.15.4 protocol specifies eight security modes, enumerated as shown in the following table.

| Level | Name | Encrypted? | Length of message integrity check | Packet length overhead |
|---|---|---|---|---|
| 0 | N/A | No | 0 (no check) | 0 |
| 1 | MIC-32 | No | 4 | 9 |
| 2 | MIC-64 | No | 8 | 13 |
| 3 | MIC-128 | No | 16 | 21 |
| 4 | ENC | Yes | 0 (no check) | 5 |
| 5 | ENC-MIC-32 | Yes | 4 | 9 |
| 6 | ENC-MIC-64 | Yes | 8 | 13 |
| 7 | ENC-MIC-128 | Yes | 16 | 21 |

The XBee/XBee-PRO S2C 802.15.4 RF Module only supports security levels 0 and 4. It does not support message integrity checks. **EE** 0 selects security level 0 and **EE** 1 selects security level 4. When using encryption, all devices in the network must use the same 16-byte encryption key for valid data to get through. Mismatched keys will corrupt the data output on the receiving device. Mismatched **EE** parameters will prevent the receiving device from outputting received data.

Working from a maximum packet size of 116 bytes, encryption affects the maximum payload as shown in the following table.

| Factor | Effect on maximum payload | Comment |
|---|---|---|
| Compatibility mode | Force to 95 | If **C8** bit 0 is set, all packets are limited to 95 bytes, regardless of other factors listed below. This is how the Legacy 802.15.4 module (S1 hardware) functions. |
| Packet overhead | Reduce by 5 | This penalty for enabling encryption is unavoidable due to the 802.15.4 protocol. |
| Source address | Reduce by 6 | This penalty is unavoidable because the 802.15.4 requires encrypted packets to be sent with a long source address, even if a short address would otherwise be used. |
| Destination address | Reduce by 6 | This penalty only applies if sending to a long address rather than a short address. |
| App header | Reduce by 4 | The app header for encryption is 4 bytes long. This penalty only applies if **MM** = 0 or 3. |

Because of the two mandatory reductions when using encryption, no packet can exceed 116 - (5+6) =105 bytes. The other options may further reduce the maximum payload to 101 bytes, 99 bytes, or 95 bytes.

When operating in API mode and not using encryption, if the source address is long, the receiving device outputs an RX Indicator (0x80) frame for received data. But, if the source address is short, the

receiving device outputs a Receive Packet (0x81) frame for received data. These same rules apply for encryption if **MM** is 0 or 3. This is possible because the four-byte encryption App header includes the short address of the sender and the long received address is not used for API output. If encryption is enabled with **MM** of 1 or 2, then no App header exists, the source address is always long, and the receiving device in API mode always outputs a RX Packet: 64-bit Address frame - 0x80.

# Maximum payload

There is a maximum payload that you can send at one time, depending on the device's configuration.

These maximums only apply in API mode. If you attempt to send an API packet with a larger payload than specified, the device responds with a Transmit Status frame (0x89) with the Status field set to 74 (Data payload too large).

In Transparent mode, the firmware splits the data as necessary to cope with maximum payloads.

## Maximum payload rules

**Note** Refer to NP (Maximum Packet Payload Bytes) which can provide maximum packet size, it always assumes a long destination address. This means that if you select a short destination address, you will be able to send up to **NP** + 6 bytes in a single packet.

1. If you enable transmit compatibility with the Legacy 802.15.4 module (S1 hardware):
    - There is a fixed maximum payload of 100 bytes if not using encryption
    - There is a fixed maximum payload of 95 bytes if using encryption
    - The rest of the rules do not apply. They apply only when you disable transmit compatibility with the Legacy 802.15.4 module.
2. The maximum achievable payload is 116 bytes. This is achieved when:
    - Not using encryption.
    - Not using the application header.
    - Using the short source address.
    - Using the short destination address.
3. If you are using the application header (MM (MAC Mode) set to **0** or **3**), the maximum achievable payload is reduced by:
    - 2 bytes if not using encryption.
    - 4 bytes if using encryption.
4. If you are using the long source address, the maximum achievable payload is reduced by 6 bytes (size of long address (8) - size of short address (2) = 6).
5. If you are using encryption, the short source addresses are promoted to long source addresses, so the maximum achievable payload is reduced by 6 bytes.
6. If you are using the long destination address, the maximum achievable payload is reduced by 6 bytes (the difference between the 8 bytes required for a long address and the 2 bytes required for a short address).
7. if you are using encryption, the maximum achievable payload is reduced by 5 bytes.

## Maximum payload summary tables

The following table indicates the maximum payload when using transmit compatibility with the Legacy 802.15.4 module (S1 hardware).

| Encryption | |
|---|---|
| **Enabled** | **Disabled** |
| 95 B | 100 B |

The following table indicates the maximum payload when using the application header and not using encryption. Increment the maximum payload in 2 bytes if you are not using the application header.

| | Destination Address | |
|---|---|---|
| Source Address | Short | Long |
| **Short** | 114 B | 108 B |
| **Long** | 108 B | 102 B |

The following table indicates the maximum payload when using the application header and using encryption. Increment the maximum payload in 4 bytes if you are not using the application header.

| | Destination Address | |
|---|---|---|
| Source Address | Short | Long |
| **Short** | 101 B | 95 B |
| **Long** | 101 B | 95 B |

## Work with Legacy devices

The Legacy 802.15.4 module (S1 hardware) transmits packets one by one. It does not transmit a packet until it receives all expected acknowledgments of the previous packet or the timeout expires.

The new XBee/XBee-PRO S2C 802.15.4 RF Modules enhance transmission by implementing a transmission queue that allows the device to transmit to several devices at the same time. Broadcast transmissions are performed in parallel with the unicast transmissions.

This enhancement in the XBee/XBee-PRO S2C 802.15.4 RF Module can produce problematic behavior under certain conditions if the receiver is a Legacy 802.15.4 module (S1 hardware).

The conditions are:

- The sender is a XBee/XBee-PRO S2C 802.15.4 RF Module, and the receiver is a Legacy 802.15.4 module.
- The sender has the Digi  Header enabled (**MM** = 0 or 3) and **RR** (XBee Retries) > 0.
- The sender sends broadcast and unicast messages at the same time to the Legacy 802.15.4 module without waiting for the transmission status of the previous packet.

The effect is:

- The receiver may display duplicate packets.

The solution is:

- Set bit 0 of the **C8** (802.15.4 Compatibility) parameter to 1 to enable TX compatibility mode in the XBee/XBee-PRO S2C 802.15.4 RF Module. This eliminates the transmission queue to avoid sending to multiple addresses simultaneously. It also limits the packet size to the levels of the Legacy 802.15.4 module.

# Networking

The following table describes some common terms we use when discussing networks.

| Term | Definition |
|---|---|
| Association | Establishing membership between end devices and a coordinator. |
| Coordinator | A full-function device (FFD) that provides network synchronization by polling nodes. |
| End device | When in the same network as a coordinator. Devices that rely on a coordinator for synchronization and can be put into states of sleep for low-power applications. |
| PAN | Personal Area Network. A data communication network that includes one or more end devices and optionally a coordinator. |

## MAC Mode configuration

Medium Access Control (MAC) Mode configures two functions:

1. Enables or disables the use of a Digi header in the 802.15.4 RF packet.
   When the Digi header is enabled (**MM** = 0 or 3), duplicate packet detection is enabled as well as certain AT commands.
   Modes 1 and 2 do not include a Digi header, which disables many features of the device. All data is strictly pass through.
2. Enables or disables MAC acknowledgment request for unicast packets.
   When MAC ACK is enabled (**MM** = 0 or 2), transmitting devices send packets with an ACK request so receiving devices send an ACK back (acknowledgment of RF packet reception) to the transmitter. If the transmitting device does not receive the ACK, it re-sends the packet up to three times or until the ACK is received.
   Modes 1 and 3 disable MAC acknowledgment. Transmitting devices send packets without an ACK request so receiving devices do not send an ACK back to the transmitter.
   Broadcast messages are always sent with the MAC ACK request disabled.

The following table summarizes the functionality.

| Mode | Digi Header | MAC ACK |
|---|---|---|
| 0 (default) | V | V |
| 1 | | |
| 2 | | V |
| 3 | V | |

The default value for the **MM** configuration parameter is 0 which enables both the Digi header and the MAC acknowledgment.

## XBee retries configuration

Configures the maximum number of retries the device executes in addition to the three retries provided by the 802.15.4 MAC. RR (XBee Retries) controls XBee retries configuration. It is also known as Application Retries.

For each XBee retry, the 802.15.4 MAC can execute up to three retries.

This only applies if MAC Mode configuration has Digi Header enabled.

## Transmit status based on MAC mode and XBee retries configurations

When working in API mode, a transmit request frame sent by the user is always answered with a transmit status frame sent by the device, if the frame ID is non-zero.

The following tables report the expected transmit status for unicast transmissions and the maximum number of MAC and application retries the device attempts.

The tables also report the transmit status reported when the device detects energy above the CCA threshold (when a CCA failure happens).

The following table applies in either of these cases:

- Digi Header is disabled.
- Digi Header is enabled and XBee Retries (**RR** parameter) is equal to 0 (default configuration).

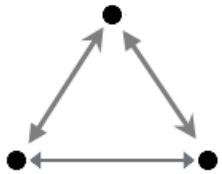
| Mac ACK Config | Destination reachable | | | Destination unreachable | | | CCA failure happened | | |
|---|---|---|---|---|---|---|---|---|---|
| | TX status | Retries | | TX status | Retries | | TX status | Retries | |
| | | MAC | App | | MAC | App | | MAC | App |
| Enabled | 00 (Success) | up to 3 | 0 | 01 (No acknowledgment received) | 3 | 0 | 02 (CCA failure) | 3 | 0 |
| Disabled | 00 (Success) | 0 | 0 | 00 (Success) | 0 | 0 | 02 (CCA failure) | 3 | 0 |

The following table applies when:

- Digi Header is enabled and XBee Retries (**RR** parameter) is bigger than 0.

| Mac ACK Config | Destination reachable | | | Destination unreachable | | | CCA failure happened | | |
|---|---|---|---|---|---|---|---|---|---|
| | TX status | Retries MAC | Retries App | TX status | Retries MAC | Retries App | TX status | Retries MAC | Retries App |
| Enabled | 00 (Success) | up to 3 per app retry | up to **RR** value | 21 (Network ACK Failure) | 3 | **RR** value | 02 (CCA failure) | 3 | **RR** value |
| Disabled | 00 (Success) | 0 | up to **RR** value | 21 (Network ACK Failure) | 0 | **RR** value | 02 (CCA failure) | 3 | **RR** value |

## Peer-to-peer networks



By default, XBee/XBee-PRO S2C 802.15.4 RF Module modules are configured to operate within a peer-to-peer network topology and therefore are not dependent upon master/slave relationships. This means that devices remain synchronized without the use of master/server configurations and each device in the network shares both roles of master and slave. Our peer-to-peer architecture features fast synchronization times and fast cold start times. This default configuration accommodates a wide range of RF data applications.

## Master/slave networks

In a Master Slave network, there is a coordinator and one or more end devices. When end devices associate to the coordinator, they become members of that Personal Area Network (PAN). As such, they share the same channel and PAN ID. PAN IDs must be unique to prevent miscommunication between PANs. Depending on the **A1** and **A2** parameters, association may assist in automatically assigning the PAN ID and the channel. These parameters are specified below based on the network role (end device or coordinator).

### *End device association*

End device association occurs if **CE** is **0** and **A1** has bit 2 set. See the following table.

| Bit | Hex value | Meaning |
|---|---|---|
| 0 | 0x01 | Allow PAN ID reassignment |
| 1 | 0x02 | Allow channel reassignment |
| 2 | 0x04 | Auto association |
| 3 | 0x08 | Poll coordinator on pin wake |

By default, **A1** is 0, which disables association and effectively causes an end device to operate in peer-to-peer mode. When bit 2 is set, the end device associates to a coordinator. This is done by sending out an active scan to detect beacons from nearby networks. The active scan selects one channel and transmits a Beacon Request command to the broadcast address and the broadcast PAN ID. It then listens on that channel for beacons from any coordinator operating on that channel. The listen time on each channel is determined by the **SD** parameter. Once that time expires, the active scan selects the next channel, repeating until all channels have been scanned.

If **A1** is 0x04 (bit 0 clear, bit 1 clear, and bit 2 set), then the active scan will reject all beacons that do not match both the configured PAN ID and the configured channel. This is the best way to join a particular coordinator.

If **A1** is 0x05 (bit 0 set, bit 1 clear, and bit 2 set), then the active scan will accept a beacon from any PAN ID, providing the channel matches. This is useful if the channel is known, but not the PAN ID.

If **A1** is 0x06 (bit 0 clear, bit 1 set, and bit 2 set), then the active scan will accept a beacon from any channel, providing the PAN ID matches. This is useful if the PAN ID is known, but not the channel.

If **A1** is 0x07 (bit 0 set, bit 1 set, and bit 2 set), then the active scan will accept a beacon from any PAN ID and from any channel. This is useful when the network does not matter, but the one with the best signal is desired.

Whenever multiple beacons are received that meet the criteria of the active scan, then the beacon with the best link quality is selected. This applies whether **A1** is 0x04, 0x05, 0x06, or 0x07.

Before the End Device joins a network, the Associate LED will be on solid. After it joins a network, the Associate LED will blink twice per second.

In the event that association parameters are changed after the end device is associated, the end device will leave the network and re-join in accordance with the new configuration parameters.

### *Coordinator association*

Coordinator association occurs if **CE** is 1 and **A2** has bit 2 set. See the following table.

| Bit | Hex value | Meaning |
|---|---|---|
| 0 | 0x01 | Allow PAN ID reassignment |
| 1 | 0x02 | Allow channel reassignment |
| 2 | 0x04 | Allow association |

By default, **A2** is 0, which prevents devices from associating to the coordinator. So, if **CE** is 1 and **A2** bit 2 is 0, the device still creates a network, but end devices are unable to associate to it.

If **A2** bit 2 is set, then joining is allowed after the coordinator forms a network.

If **A2** bit 0 is set, the coordinator issues an active scan. This means it will send out beacon requests to the broadcast address (0xFFFF) and the broadcast PAN ID (0xFFFF). Then, it will listen for beacons. The listen time is determined by the **SD** parameter. Then the same beacon request is sent out on the next channel and the device listens for beacon responses on that channel. This process repeats until each channel in the channel mask (**SC**) is scanned for **SD** time. If none of the beacons on any of the channels return a PAN ID equivalent with the **ID** parameter, then the coordinator will use the ID parameter for the PAN ID of the new network it forms. But, if a beacon response matches the PAN ID of the coordinator, the coordinator will form a PAN on a unique PAN ID.

If **A2** bit 0 is clear, then the coordinator will form a network on the PAN ID identified by the **ID** parameter, without regard to another network that might have the same PAN ID.

If **A2** bit 1 is set, the coordinator will issue an energy scan, similar to the active scan. It will listen on each channel specified in the **SC** parameter for the time indicated by the **SD** parameter. After the scan is complete, the channel with the least energy is selected to form the new network.

If **A2** bit 1 is clear, then no energy scan is performed and the **CH** parameter is used to select the channel of the new network.

If bits 0 and 1 of **A2** are both set, then an active scan is done followed by an energy scan. However, the channels on which the active scan finds a coordinator are eliminated as possible channels for the energy scan, unless such an action would eliminate all channels. If beacons are found on all channels in the channel mask, then then the energy scan behaves the same as it would if beacons are not found on any of those channels. Therefore, the active scan will be performed on all channels in the channel mask. Then, an energy scan will be performed on the channels in the channel mask that did not find a coordinator.

Depending on the result of the active scan, the set of channels for the energy scan varies. If a PAN ID is found on all the channels in the channel mask, then the energy scan operates on all the channels in the channel mask. If at least one of the channels in the channel mask did not find a PAN ID, then the channels with PAN IDs are eliminated from consideration for the energy scan. After the energy scan completes, the channel with the least energy is selected for forming the new network.

Whenever **CE**, **ID**, **A2**, or **MY** changes, the network will leave and association will recur with the new parameters. Any end devices associated to the coordinator prior to changing one of these parameters will lose association. For this reason, it is important not to change these parameters on a coordinator unless needed.

Before the Coordinator forms a network, the Associate LED will be on solid. After it forms a network, the Associate LED will blink once per second.

### *Indirect addressing*

The XBee/XBee-PRO S2C 802.15.4 RF Module may hold indirect messages until it runs out of buffers and the size of those messages does not matter. A brief summary follows:

A coordinator (**CE** = 1) must have **SP** set to a non-zero value to use indirect addressing. Otherwise, all frames will be sent directly, assuming the target device is awake. For best operation, **SP** and **ST** should be set to match the **SP** and **ST** values of the end nodes. The coordinator will hold onto an indirect message until it receives a poll from the device to which the message is addressed, or until a timeout, whichever occurs first. The timeout is 2.5 times the value of **SP**.

After an indirect message is sent, subsequent messages to the same address will be sent directly for a period of time equal to the **ST** parameter. The assumption is that the end device will remain awake for **ST** time after the last transmission or reception. Therefore, after a poll, every transmission and reception involving that address restart the **ST** timer. After the **ST** timer expires, the coordinator will again use indirect messaging (i.e it will hold messages awaiting a poll.)

End devices using cyclic sleep send a poll to the coordinator when they wake up unless **SO** bit 0 is set. End devices using pin sleep may be configured to send a poll on a pin wakeup by setting bit 3 of **A1**. Also the **FP** command can be used to send a poll upon exiting command mode. However, the poll is not sent while in command mode. Rather the poll is sent after changes are applied so that the module is ready to receive data after the poll is sent.

It is more difficult to use indirect addressing with pin sleep than with cyclic sleep because the end device must wake up periodically to poll for the data from the coordinator. Otherwise, the coordinator will discard the data. Therefore, cyclic sleep is the only recommended sleep mode for indirect addressing.

### *Association indicators*

There are two types of association indicators: Asynchronous device status messages, and on demand queries. Asynchronous device status messages occur whenever a change occurs and API mode is

enabled. On demand queries occur when the **AI** command is issued, which can occur in Command mode, in API mode, or as a remote command.

#### Device Status Messages

Not all device status messages are related with association, but for completeness all device status types reported by XBee/XBee-PRO S2C 802.15.4 RF Module are listed in the following table.

| Type | Meaning |
|---|---|
| 0x00 | Hardware reset. |
| 0x01 | Watchdog reset. |
| 0x02 | End device successfully associated with a coordinator. |
| 0x03 | End device disassociated from coordinator or coordinator failed to form a new network. |
| 0x06 | Coordinator formed a new network. |
| 0x0D | Input voltage on the XBee-PRO device is too high, which prevents transmissions. |

#### Association indicator status codes

The XBee/XBee-PRO S2C 802.15.4 RF Module code can potentially give any of the status codes in the following table.

| Code | Meaning |
|---|---|
| 0x00 | Coordinator successfully started, End device successfully associated, or operating in peer to peer mode where no association is needed. |
| 0x03 | Active Scan found a PAN coordinator, but it is not currently accepting associations. |
| 0x04 | Active Scan found a PAN coordinator in a beacon-enabled network, which is not a supported feature. |
| 0x05 | Active Scan found a PAN, but the PAN ID does not match the configured PAN ID on the requesting end device and bit 0 of **A1** is not set to allow reassignment of PAN ID. |
| 0x06 | Active Scan found a PAN on a channel does not match the configured channel on the requesting end device and bit 1 of **A1** is not set to allow reassignment of the channel. |
| 0x0C | Association request failed to get a response. |
| 0x13 | End device is disassociated or is in the process of disassociating. |
| 0xFF | Initialization time; no association status has been determined yet. |

### *Sleep*

Sleep is implemented to support installations where a mains power source is not available and a battery is required. In order to increase battery life, the device sleeps, which means it stops operating. It can be woken by a timer expiration or a pin.

For more information about sleep modes, see Sleep modes.

**Sleep conditions**

Since instructions stop executing while the device is sleeping, it is important to avoid sleeping when the device has work to do. For example, the device will not sleep if any of the following are true:

1. The device is operating in command mode, or in the process of getting into command mode with the +++ sequence.
2. The device is processing AT commands from API mode
3. The device is processing remote AT commands
4. Something is queued to the serial port and that data is not blocked by $\overline{\text{RTS}}$ flow control

If each of the above conditions are false, then sleep may still be blocked in these cases:

1. Enough time has not expired since the device has awakened.
    a. If the device is operating in pin sleep, the amount of time needed for one character to be received on the UART is enough time.
    b. If the device is operating in cyclic sleep, enough time is defined by a timer. The duration of that timer is:
        i. defined by **ST** if in **SM** 5 mode and it is awakened by a pin
        ii. 30 ms to allow enough time for a poll and a poll response
        iii. 750 ms to allow enough time for association, in case that needs to happen
    c. In addition, the wake time is extended by an additional **ST** time when new OTA data or serial data is received.
2. Sleep Request pin is not asserted when operating in pin sleep mode
3. Data is waiting to be sent OTA.

# Clear Channel Assessment (CCA)

Prior to transmitting a packet, the device performs a CCA (Clear Channel Assessment) on the channel to determine if the channel is available for transmission. The detected energy on the channel is compared with the **CA** (Clear Channel Assessment) parameter value. If the detected energy exceeds the **CA** parameter value, the device does not transmit the packet.

Also, the device inserts a delay before a transmission takes place. You can set this delay using the **RN** (Backoff Exponent) parameter. If you set **RN** to 0, then there is no delay before the first CCA is performed. The RN parameter value is the equivalent of the “minBE” parameter in the 802.15.4 specification. The transmit sequence follows the 802.15.4 specification.

By default, the **MM** (MAC Mode) parameter = 0. On a CCA failure, the device attempts to re-send the packet up to two additional times.

When in Unicast packets with **RR** (Retries) = 0, the device executes two CCA retries. Broadcast packets always get two CCA retries.

**Note** Customers in Europe who have the XBee 802.15.4 module must manage their CCA settings. See CA (CCA Threshold) for **CA** values.

## CCA operations

CCA is a method of collision avoidance that is implemented by detecting the energy level on the transmission channel before starting the transmission. The CCA threshold (defined by the **CA**

parameter) defines the energy level that it takes to block a transmission attempt. For example, if CCA is set to the default value of 0x2C (which is interpreted as -44 dBm) then energy detected above the -44 dBm level (for example -40 dBm) temporarily blocks a transmission attempt. But if the energy level is less than that (for example -50 dBm), the transmission is not blocked. The intent of this feature is to prevent simultaneous transmissions on the same channel.

CCA can be set down to 0x50 (or -80 dBm), which approaches the sensitivity level. Setting such a threshold may not work in a noisy environment.

In the event that the energy level exceeds the threshold, the transmission is blocked some random number of backoff periods. The number of backoff periods is defined by random(2^n - 1) where the initial value of n is defined by the **RN** parameter and it increments after each failure. When **RN** is set to its default value of 0, then 2^n -1 is 0, preventing any delay before the first energy detection on a new frame. However, n increments after each CCA failure, giving a greater range for the number of backoff periods between each energy detection cycle.

In the event that five energy detection cycles occur and each one detects too much energy, the application tries again 1 to 48 ms later. After the application retries are exhausted, then the transmission fails with a CCA error.

Whenever the MAC code reports a CCA failure, meaning that it performed five energy detection cycles with exponential random back-offs, and each one failed, the **EC** parameter is incremented. The EC parameter can be read at any time to find out how noisy the operating channel is. It continues to increment until it reaches its maximum value of 0xFFFF. To get new statistics, you can always set **EC** back to 0.

# Serial interface

The XBee/XBee-PRO S2C 802.15.4 RF Module interfaces to a host device through a serial port. The device can communicate through its serial port with:

- Through logic and voltage compatible universal asynchronous receiver/transmitter (UART).
- Through a level translator to any serial device, for example, through an RS-232 or USB interface board.
- Through a SPI, as described in SPI signals.

## Select a serial port

The device has two serial ports and only one is active at a time. To be active, a port must be enabled and in use.

The UART is always enabled. The SPI is enabled if it is configured. To be configured, SPI_MISO, SPI_MOSI, SPI_SSEL , and SPI_CLK must all be configured as peripherals. On the surface-mount device, these lines are configured as peripherals by setting P5, P6, P7, and P8 to 1. This is also the default configuration for surface-mount devices.

On the through-hole device, those pins are not available and SPI is disabled by default. Therefore, to configure the SPI pins on a through-hole device, hold DOUT low during a reset. If the UART is not hooked up, then DOUT can be treated as an input to force the device into SPI mode. It is best to follow this special operation by a **WR** operation so that the SPI port will still be enabled on future resets without forcing DOUT low.

Once the SPI port is enabled by either means, it is still not active until the external SPI master asserts SPI_SSEL low. After the SPI port is active, the device continues to use the SPI port until the next reset.

### *Serial receive buffer*

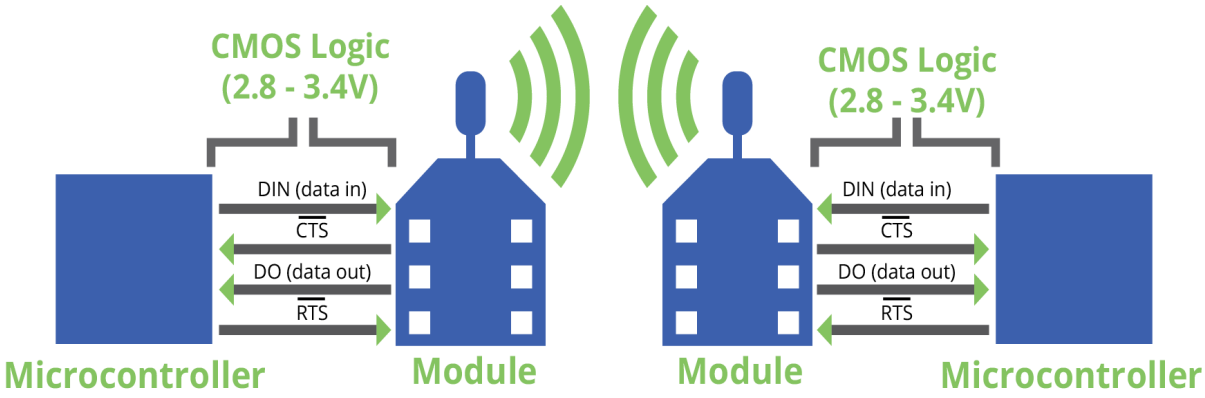
When serial data enters the device through the DIN pin (or the MOSI pin), it stores the data in the serial receive buffer until the device can process it. Under certain conditions, the device may not be able to process data in the serial receive buffer immediately. If large amounts of serial data are sent to the device such that the serial receive buffer would overflow, then it discards new data. If the UART is in use, you can avoid this by the host side honoring $\overline{\text{CTS}}$ flow control.

### *Serial transmit buffer*

When the device receives RF data, it moves the data into the serial transmit buffer and sends it out the UART or SPI port. If the serial transmit buffer becomes full and the system buffers are also full, then it drops the entire RF data packet. Whenever the device receives data faster than it can process and transmit the data out the serial port, there is a potential of dropping data.

## UART data flow

Devices that have a UART interface connect directly to the pins of the XBee/XBee-PRO S2C 802.15.4 RF Module as shown in the following figure. The figure shows system data flow in a UART-interfaced environment. Low-asserted signals have a horizontal line over the signal name.

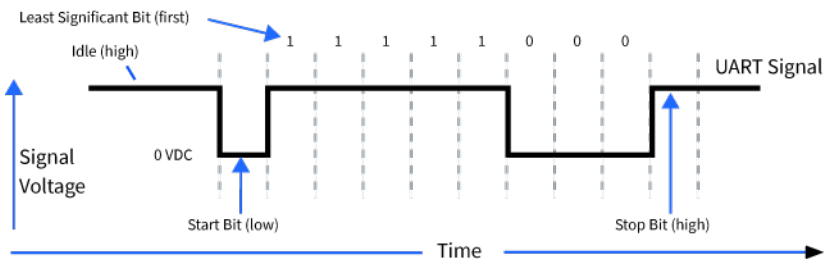



### *Serial data*

A device sends data to the XBee/XBee-PRO S2C 802.15.4 RF Module's UART through TH pin 3/SMT pin 4 DIN as an asynchronous serial signal. When the device is not transmitting data, the signals should idle high.

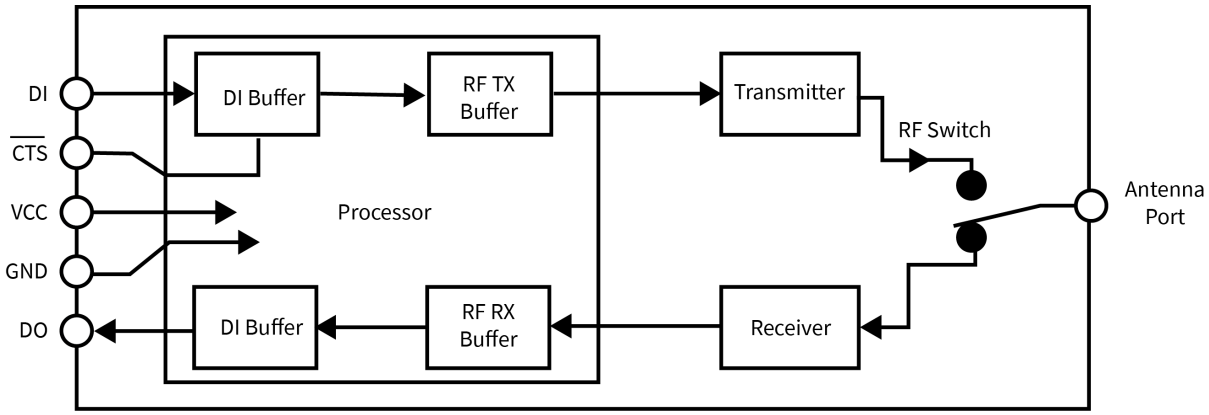
For serial communication to occur, you must configure the UART of both devices (the microcontroller and the XBee/XBee-PRO S2C 802.15.4 RF Module) with compatible settings for the baud rate, parity, start bits, stop bits, and data bits.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following diagram illustrates the serial bit pattern of data passing through the device. The diagram shows UART data packet 0x1F (decimal number 31) as transmitted through the device.




## Flow control

The XBee/XBee-PRO S2C 802.15.4 RF Module maintains buffers to collect serial and RF data that it receives. The serial receive buffer collects incoming serial characters and holds them until the device can process them. The serial transmit buffer collects the data it receives via the RF link until it transmits that data out the serial port. The following figure shows the process of device buffers collecting received serial data.




### *CTS flow control*

If you enable CTS flow control (by setting **D7** to 1), when the serial receive buffer is 17 bytes away from being full, the device de-asserts CTS (sets it high) to signal to the host device to stop sending serial data. The device reasserts CTS after the serial receive buffer has 34 bytes of space. The maximum space available for receiving serial data is 174 bytes, which is enough to hold 1.5 full packets of data.

#### Flow control threshold

Use the **FT** parameter to set the flow control threshold. Since the receive serial buffer is 174 bytes, you cannot set **FT** to more than 174-12 = 162 bytes. This allows up to 17 bytes of data to come in after CTS is de-asserted before data is dropped. The default value of **FT** is 109, leaving space for an external device that responds slowly to CTS being de-asserted. The minimum value of **FT** is 17, which is the minimal operational level.

### *RTS flow control*

If you send the **D6** command to enable RTS flow control, the device does not send data in the serial transmit buffer out the DOUT pin as long as RTS is de-asserted (set high). Do not de-assert RTS for long periods of time or the serial transmit buffer will fill. If the device receives an RF data packet and the serial transmit buffer does not have enough space for all of the data bytes, it discards the entire RF data packet.

If the device sends data out the UART when $\overline{RTS}$ is de-asserted (set high) the device could send up to five characters out the UART port after $\overline{RTS}$ is de-asserted.

Cases in which the DO buffer may become full, resulting in dropped RF packets:

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

### *Asynchronous Parameters*

Asynchronous communication over a UART is configured with a start bit, data bits, parity, stop bits, and baud rate. Out of these, only parity and baud rate are configurable on the device for 802.15.4. This means that the connecting micro-controller must match the the start bits (1), the data bits (8), and the stop bits (1) of the device for proper communication.

#### Parity

Use the **NB** command to configure parity; see NB (Parity).

# SPI operation

This section specifies how SPI is implemented on the device, what the SPI signals are, and how full duplex operations work.

## SPI signals

The XBee/XBee-PRO S2C 802.15.4 RF Module supports SPI communications in 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 device:

| Signal | SMT pin # | SMT applicable AT command | TH Pin # | TH applicable AT command |
|---|---|---|---|---|
| SPI_MOSI (Master out, Slave in) | 16 | **P6** | 11 | **D4** |
| SPI_MISO (Master in, Slave out) | 17 | **P5** | 4 | **P2** |
| SPI_SCLK (Serial clock) | 14 | **P8** | 18 | **D2** |
| SPI_$\overline{SSEL}$ (Slave select) | 15 | **P7** | 17 | **D3** |
| SPI_$\overline{ATTN}$ (Attention) | 12 | **P9** | 19 | **D1** |

By default, the inputs have pull-up resistors enabled. Use the **PR** command to disable the pull-up resistors. When the SPI pins are not connected but the pins are configured for SPI operation, then the device requires the pull-ups for proper UART operation.

### *Signal description*

**SPI_MISO**: When SPI_CLK is active, the device outputs the data on SPI_MISO at the SPI_CLK rate. If there are other SPI slave devices connected to the same SPI master, then the SPI_MISO output from

XBee device must be externally tri-stated when SPI_SSEL is de-asserted to prevent multiple devices from driving SPI_MISO.

**SPI_MOSI**: The SPI master outputs data on this line at the SPI_CLK rate after it selects the desired slave. When you configure the device for SPI operations, this pin is an input.

**SPI_SCLK**: The SPI master outputs a clock on this pin, and the rate must not exceed the maximum allowed, 5 Mb/s. This signal clocks data transfers on MOSI and MISO.

**SPI_$\overline{\text{SSEL}}$**: The SPI master outputs a low signal on this pin to select the device as an SPI slave. When you configure the device for SPI operations, this pin is an input. This signal enables serial communication with the slave.

**SPI_$\overline{\text{ATTN}}$**: The device asserts this pin low when it has data to send to the SPI master. When you configure this pin for SPI operations, it is an output (not tri-stated). This signal alerts the master that the slave has data queued to send. The device asserts this pin as soon as data is available to send to the SPI master and it remains asserted until the SPI master has clocked out all available data.

## SPI parameters

Most host processors with SPI hardware allow you to set the bit order, clock phase and polarity. For communication with all XBee/XBee-PRO S2C 802.15.4 RF Modules, the host processor must set these options as follows:

- Bit order: send MSB first
- Clock phase (CPHA): sample data on first (leading) edge
- Clock polarity (CPOL): first (leading) edge rises

All XBee/XBee-PRO S2C 802.15.4 RF Modules use SPI mode 0 and MSB first. Mode 0 means that data is sampled on the leading edge and that the leading edge rises. MSB first means that bit 7 is the first bit of a byte sent over the interface.

## SPI and API mode

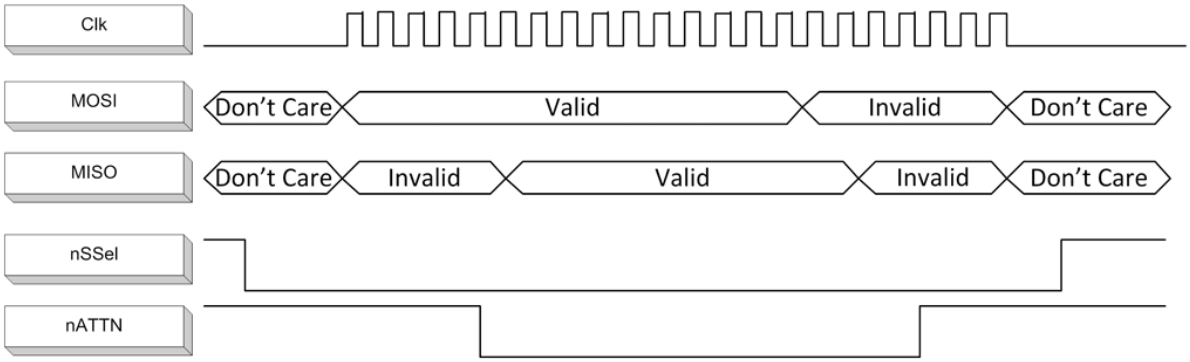
The SPI only operates in API mode 1. The SPI does not support Transparent mode or API mode 2 (with escaped characters). This means that the **AP** configuration only applies to the UART interface and is ignored while using the SPI.

## Full duplex operation

When using SPI on the XBee/XBee-PRO S2C 802.15.4 RF Module the device uses API operation without escaped characters to packetize data. The device ignores the configuration of **AP** because SPI does not operate in any other mode. SPI is a full duplex protocol, even when data is only available in one direction. This means that whenever a device receives data, it also transmits, and that data is normally invalid. Likewise, whenever a device transmits data, invalid data is probably received. To determine whether or not received data is invalid, the firmware places the data in API packets.

SPI allows for valid data from the slave to begin before, at the same time, or after valid data begins from the master. When the master sends data to the slave and the slave has valid data to send in the middle of receiving data from the master, a full duplex operation occurs, where data is valid in both directions for a period of time. Not only must the master and the slave both be able to keep up with the full duplex operation, but both sides must honor the protocol.

The following figure illustrates the SPI interface while valid data is being sent in both directions.

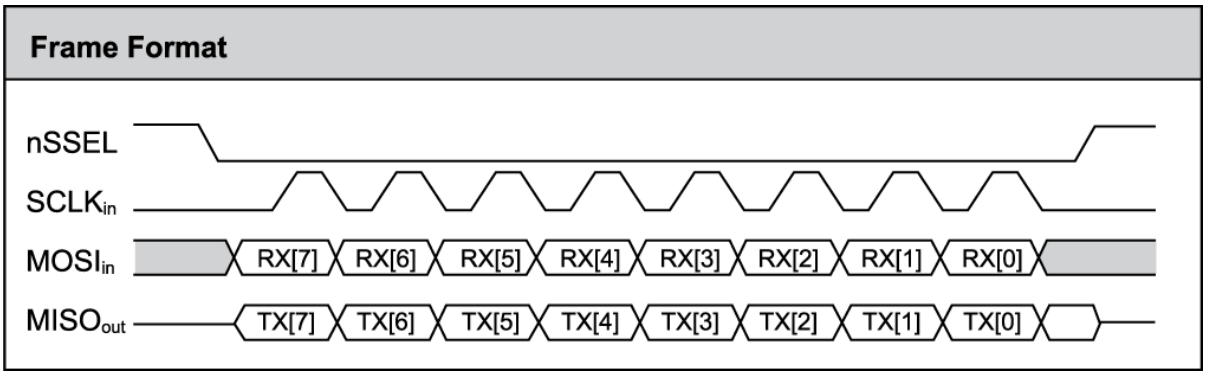



## Slave mode characteristics

In slave mode, the following apply:

- SPI Clock rates up to 5 MHz (5 Mb/s) are possible.
- Data is MSB first.
- It uses Frame Format Mode 0. This means CPOL= 0 (idle clock is low) and CPHA = 0 (data is sampled on the clock's leading edge). The picture below diagrams Mode 0.
- The SPI port is setup for API mode and is equivalent to **AP** = 1.

The following picture shows the frame format for SPI communications.




# I/O support

The following topics describe analog and digital I/O line support, line passing and output control.

## Digital I/O line support

Digital I/O is available on pins D0 through D8. Each of these pins may be configured as 3, 4, or 5 with the following meanings:

- 3 is digital input
- 4 is digital output low
- 5 is digital output high

Pins D0 through D8 are available for digital input, but only D0 through D7 are available for digital output.

| Function | SMT pin | TH pin | AT command |
|---|---|---|---|
| DIO0 | 33 | 20 | D0 (DIO0/AD0) |
| DIO1 | 32 | 19 | D1 (DIO1/AD1) |
| DIO2 | 31 | 18 | D2 (DIO2/AD2) |
| DIO3 | 30 | 17 | D3 (DIO3/AD3) |
| DIO4 | 24 | 11 | D4 (DIO4) |
| DIO5 | 28 | 15 | D5 (DIO5/ASSOCIATED_INDICATOR) |
| DIO6 | 29 | 16 | D6 (DIO6/RTS) |
| DIO7 | 25 | 12 | D7 (DIO7/CTS) |
| DIO8 | 10 | 9 | D8 (DIO8/SLEEP_REQUEST) |

## Analog input

Analog input is available on D0 through D3 by configuring these parameters to 2. Analog input is not available on D4 and D5.

## On demand I/O sampling

You can use the **IS** (Force Sample) command to sample pins configured as digital I/O and analog input. If no pins are configured in this manner (with the **DO** - **D8** commands set to 2, 3, 4, or 5), then the **IS** command returns an error.

In Command mode, the output is:

| Output | Description |
|---|---|
| 01 | Indicates one sample. That is the only possibility for Command mode. |
| 20E | Mask to indicate which lines are sampled (A0, D3, D2, and D1). |
| 00A | Digital sample indicates D3 high, D2 low, and D1 high. |
| 3FF | Analog sample for A0 indicates that A0 is reading maximum voltage of 1.2 V. |

In API mode, the output is:

7E 00 0C 83 00 00 00 00 01 03 3E 01 2A 02 10 FD

In this example, note the following:

83 indicates RX Packet: 16-bit Address I/O frame (0x83).

00 00 indicates 16-bit source address.

00 indicates RSSI (does not apply).

00 indicates options.

01 indicates the number of samples.

03 3E mask to indicate which lines are sampled (A0, D8, D5, D4, D3, D2, and D1).

01 2A digital sample that indicates that D8 is high, D5 is high, D4 is low, D3 is high, D2 is low, and D1 is high.

02 10 indicates that A0 has input voltage nearly half of capacity, where 03 FF would indicate the full voltage of 1.2 V = 1200 mV.

For a remote **IS** command sent to the device listed above with the same configuration, the output is:

7E 00 16 97 01 00 13 A2 00 40 E3 C0 15 00 00 49 53 00 01 03 3E 01 2A 02 10 9F

In this example, note the following:

97 indicates Remote AT Command Response frame (0x97).

01 is the frame ID.

00 13 A2 00 40 E3 C0 15 is the 64-bit source address.

00 00 indicates 16-bit source address.

49 53 (IS) indicates command response to the **IS** command.

00 indicates the status is OK.

01 indicates the number of samples.

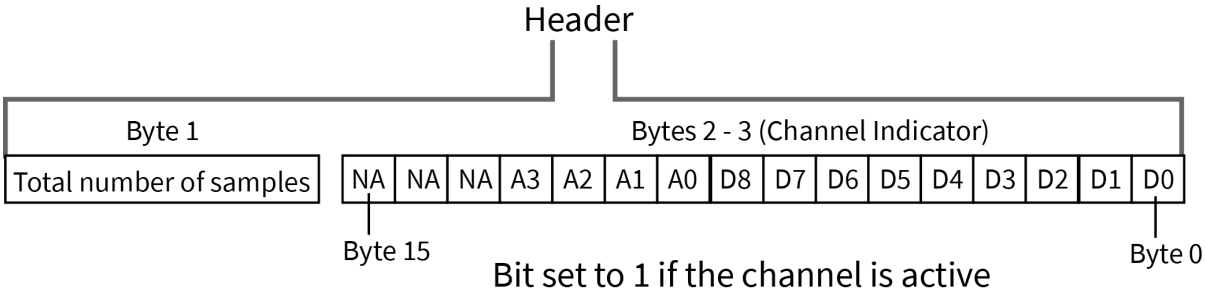
03 3E mask to indicate which lines are sampled (A0, D8, D5, D4, D3, D2, and D1).

01 2A digital sample that indicates that D8 is high, D5 is high, D4 is low, D3 is high, D2 is low, and D1 is high.

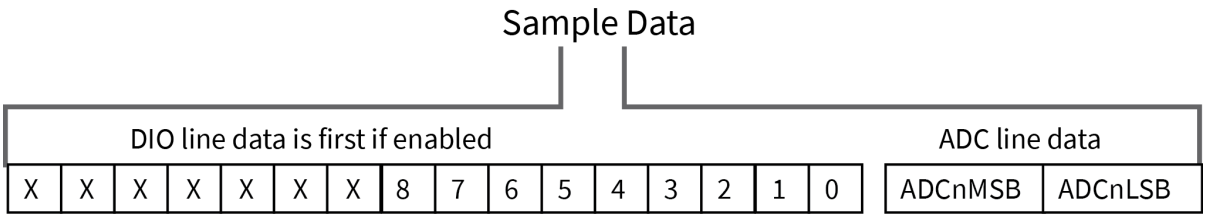
02 10 indicates that A0 has input voltage about half of capacity, where 03 FF would indicate full voltage of 1.2 V = 1200 mV.

### *I/O data format*

I/O data begins with a header. The first byte of the header defines the number of samples forthcoming. The last two bytes of the header (Channel Indicator) define which inputs are active. Each bit represents either a DIO line or ADC channel. The following figure illustrates the bits in the header.




Sample data follows the header and the channel indicator frame determines how to read the sample data. If any of the DIO lines are enabled, the first two bytes are the DIO sample. The ADC data follows. ADC channel data is represented as an unsigned 10-bit value right-justified on a 16- bit boundary. The following figure illustrates the sample data bits.




**Multiple samples per packet**

The **IT** parameter specifies how many I/O samples can be transmitted in a single OTA packet. Any single-byte value (0 - 0xFF) is accepted for input. However, the value is adjusted downward based on how many I/O samples can fit into a maximum size packet; see Maximum payload. A query of the **IT** parameter after changes are applied tells how many I/O samples actually occur.

Since **MM** must be 0 or 3 to send I/O samples, the maximum payload in the best of conditions (short source address, short destination address, and no encryption) is 114 bytes. Seven of those bytes are used by the command header and the I/O header, leaving 107 bytes for I/O samples. The minimum I/O sample is 2 bytes. Therefore the maximum possible usable value for **IT** is 53 (or 0x35).

### *API support*

I/O data is sent out the UART using an API frame. All other data can be sent and received using Transparent Operation or API frames if API mode is enabled (**AP** > 0).

API Operations support two RX (Receive) frame identifiers for I/O data (set 16-bit address to 0xFFFE and the device does 64-bit addressing):

- 0x82 for RX Packet: 64-bit Address I/O
- 0x83 for RX Packet: 16-bit Address I/O

The API command header is the same as shown in RX Packet: 64-bit Address frame - 0x80 and RX Packet: 16-bit address I/O frame - 0x83. RX data follows the format described in I/O data format.

## Periodic I/O sampling

Periodic I/O sampling is done by periodically gathering samples on the source node and sending the samples to the destination node to output on its serial port.

# Source node

On the source node, the **IR** parameter specifies how many milliseconds between samples. The **IT** parameter specifies how many samples will be sent OTA in a single packet, and the **DH**/**DL** parameters designate the node that will receive the samples. In addition **MM** must be 0 or 3 for I/O samples to be sent.

The maximum value of **IR** is 0xFFFF. If **IR** is 0, periodic I/O sampling is disabled. Otherwise, **IR** specifies the number of milliseconds between samples.

If **IT** is 0, than an I/O sample is sent every **IR** milliseconds. If **IR** is greater than 0, then an I/O sample is sent every **IR***IT milliseconds. Any single-byte value (0 - 0xFF) is accepted for input. However, the value is adjusted downward based on how many I/O samples can fit into a maximum size packet. A query of the **IT** parameter after changes are applied tells how many I/O samples will actually occur. Since **MM** must be 0 or 3 to send I/O samples, the maximum payload in the best of conditions (short source address, short destination address, and no encryption) is 114 bytes. Seven of those bytes are used by the command header and the I/O header, leaving 107 bytes for I/O samples. The minimum I/O sample is 2 bytes. Therefore the maximum possible usable value for **IT** is 53 (or 0x35).

Although samples may be taken every millisecond, **IR*IT** should be at least 20 milliseconds. This allows time for OTA transmission and output on the serial port of the receiving module.

# Destination node

On the destination node, the **IU** parameter enables the serial port to output I/O samples it receives. **IU** is set to 1 by default. If **IU** is set and the destination node is not in Command mode, it displays samples it receives on its serial port in API format. The **AP** parameter is ignored in this case.

# Data format

The I/O sample is formatted as follows:

| IT | Mask | Samples |
|---|---|---|
| 1 byte | 2 bytes | 2 * number of samples |

The **IT** field is described above.

The Mask is a bit field formatted as follows:

| Reserved | A3 - A0 | D8 - D0 |
|---|---|---|
| 3 bits | 4 analog bits | 9 digital bits |

If you set any of the digital bits, then a digital sample follows the Mask.

For every analog bit that is set, there is a 10-bit sample using two bytes where **A0** is included first and **A3** is included last.

The number of samples indicated by the mask is repeated **IT** times. The mask is not repeated.

The sending node sends I/O samples to the address specified by **DH**/**DL**. The receiving node displays the samples as described above, providing API mode is enabled, the node is not in Command mode, and the **IU** parameter is 1 (**IU** is 1 by default). If any of these are false, the I/O sample is discarded without being displayed.

## Change Detect I/O sampling

When you use the IC (DIO Change Detect) command to enable DIO Change Detect, DIO lines 0 - 7 are monitored. When a change is detected on a DIO line, the following occurs:

1. This packet does not contain DIO8 and does not contain any analog samples. However, it does contain a digital reading of the D0 - D7 pins configured for input or output, whether or not all those bits are set in **IC**.
2. Any queued samples are transmitted before the change detect data. This may result in receiving a packet with less than **IT** (Samples before TX) samples.

Change detect does not affect Pin Sleep wake-up. The D8 pin (DI8/SLEEP_RQ/$\overline{\text{DTR}}$ ) is the only line that wakes a device from Pin Sleep. If not all samples are collected, the device still enters Sleep mode after a change detect packet is sent.

Change detect is only supported when the **Dx** (DIOx Configuration) parameter equals 3, 4 or 5.

## Wakeup I/O sampling

Bit 2 of the **SO** parameter specifies whether or not the device automatically samples the ADC/DIO lines on wakeup. On the receiving side the I/O sample may go to the I/O pins, the serial port, both, or neither based on the settings of the **IA** (I/O Input Address) and **IU** (I/O Output Enable) parameters.

For more information, see SO (Sleep Options), IA (I/O Input Address), and IU (I/O Output Enable).

## Sample rate (interval)

The Sample Rate (Interval) feature allows enabled ADC and DIO pins to be read periodically on devices that are not configured to operate in Sleep Mode. When one of the Sleep Modes is enabled and the IR (Sample Rate) parameter is set, the device stays awake until IT (Samples before TX) samples have been collected.

Once a particular pin is enabled, the appropriate sample rate must be chosen. The maximum sample rate that can be achieved while using one A/D line is 1 sample/ms or 1 kHz. The device cannot keep up with transmission when **IR** and **IT** are equal to 1 and we do not recommend configuring the device to sample at rates greater than once every 20 ms.

## I/O line passing

You can configure XBee/XBee-PRO S2C 802.15.4 RF Modules to perform analog and digital line passing. When a device receives an RF I/O sample data packet, you can set up the receiving device to update any enabled outputs (PWM and DIO) based on the data it receives.

Digital I/O lines are mapped in pairs; pins configured as digital input on the transmitting device affect the corresponding digital output pin on the receiving device. For example: DI5 (pin 25) can only update DO5 (pin 25).

No I/O line passing occurs if the receiving device does not have the corresponding pins set for output.

For Analog Line Passing, the XBee/XBee-PRO S2C 802.15.4 RF Module has two PWM output pins that simulate the voltage measured by the ADC lines AD0 and AD1. For example, when configured as an ADC, AD0 (pin 33) updates PWM0 (pin 7); AD1 (pin 32) updates PWM1 (pin 8).

The default setup is for outputs to not be updated. Instead, a device sends I/O sample data out the serial interface in API mode, even if the destination node is not configured for API mode. You can use the **IU** command to disable sample data output.

To enable updating the outputs, set the **IA** (I/O Input Address) parameter with the address of the device that has the appropriate inputs enabled. This effectively binds the outputs to a particular device's input. This does not affect the ability of the device to receive I/O line data from other devices - only its ability to update enabled outputs. Set the **IA** parameter to 0xFFFF (broadcast address) to set up the device to accept I/O data for output changes from any device on the network.

For line passing to function, the device configured with inputs must generate sample data.

When outputs are changed from their configured state, the device can be setup to return the output level to its configured state after a timer expires. The timers are set using the **T*n*** (**D*n*** Output Timer) and **PT** (PWM Output Timeout) commands. The timers are reset every time the device receives a valid I/O sample packet with a matching **IA** address. You can adjust the **IT** (Samples before TX) and **IR** (Sample Rate) parameters on the transmitting device to keep the outputs set to their active output if the system needs more time than the timers can handle. Alternatively, the timers can be set to 0xFF, which prevents them from expiring (in other words, the outputs remain in the state specified by the input device indefinitely).

### *I/O line passing details*

The same message is received for both I/O sampling and for I/O line passing. But I/O line passing only occurs if **IA** matches the short or long address of the sending node or if **IA** is 0xFFFF to match a sample from any node. The default value of **IA** is 0xFFFFFFFF, which prevents I/O line passing from occurring on the node because no node has that address. Additionally, the receiving device must have a matching value for output. For example, if an ADC0 sample is received, then **P0** must be configured with 2 for PWM output. Otherwise, the analog signal will not be reflected with a matching PWM signal. Likewise, if the sample indicates that D2 is high, but **D2** is not set to 4 or 5 on the receiving device, then the D2 pin will not be affected by I/O line passing.

When a digital output pin is set to something different than its configured value, that pin may return to its configured value after the time specified for the corresponding timer. **T0** specifies how long D0 will hold its non-configured value and **T1** specifies how long D1 will hold its non-configured value. A value of 0xFF indicates that a pin holds the value of the input of the corresponding device indefinitely and a value less than 0xFF specifies how many tenth second units the pin holds the non-configured value.

For PWM outputs, PT timer applies to both PWM0 and PWM1. A value of 0xFF allows the PWM pin to output a duty cycle reflective of the analog input indefinitely and a smaller value indicates how many 10th second units before PWM output reverts to the duty cycle specified by **M0** or **M1**.

## Output control

The **IO** (Digital Output Level) command controls the output levels of D0 through D7 that are configured as output pins (either 4 or 5). These values override the configured output levels of the pins until they are changed again (the pins do not automatically revert to their configured values after a timeout.) You can use the **IO** command to trigger a sample on change detect.

# Sleep support

Set SO (Sleep Options) bit 1 to suppress automatic wake-up sampling.

When a device wakes, it always performs a sample based on any active ADC or DIO lines. This allows sampling based on the sleep cycle whether it be Cyclic Sleep (**SM** = 4 or 5) or Pin Sleep (**SM** = 1). Set the IR (Sample Rate) parameter to gather more samples when awake.

For Cyclic Sleep modes: If the **IR** parameter is set, the device stays awake until the IT (Samples before TX) parameter is met. The device stays awake for ST (Time before Sleep).

## Sleep modes

Sleep modes enable the device to enter states of low-power consumption when not in use. In order to enter Sleep mode, one of the following conditions must be met (in addition to the device having a non-zero **SM** parameter value):

- SLEEP_RQ/$\overline{\text{DTR}}$ (pin 9 on through-hole devices, pin 10 on surface-mount devices) is asserted and the device is in a pin sleep mode (**SM** = 1, or 5)
- The device is idle (no data transmission or reception) for the amount of time defined by the **ST** (Time before Sleep) parameter.

**Note** **ST** is only active when **SM** = 4 or 5.

The following table shows the sleep mode configurations.

| Sleep mode | Description |
|---|---|
| **SM** 0 | No sleep |
| **SM** 1 | Pin sleep |
| **SM** 4 | Cyclic sleep |
| **SM** 5 | Cyclic sleep with pin wake-up |

### *Pin Sleep mode (SM = 1)*

Pin Sleep mode minimizes quiescent power (power consumed when in a state of rest or inactivity). This mode is voltage level-activated; when Sleep_RQ (pin 9 for through-hole, pin 10 for surface-mount) is asserted, the device finishes any transmit, receive or association activities, enters Idle mode, and then enters a state of sleep. The device does not respond to either serial or RF activity while in pin sleep.

To wake a sleeping device operating in Pin Sleep mode, de-assert Sleep_RQ. The device wakes when Sleep_RQ is de-asserted and is ready to transmit or receive when the $\overline{\text{CTS}}$ line is low. When waking the device, the pin must be de-asserted at least two 'byte times' after $\overline{\text{CTS}}$ goes low. This assures that there is time for the data to enter the DI buffer.

### *Cyclic Sleep mode (SM = 4)*

The Cyclic Sleep modes allow devices to periodically check for RF data. When the **SM** parameter is set to 4, the XBee/XBee-PRO S2C 802.15.4 RF Module is configured to sleep, then wakes once per cycle to check for data from a coordinator. The Cyclic Sleep Remote sends a poll request to the coordinator at a specific interval set by the **SP** (Cyclic Sleep Period) parameter. The coordinator transmits any queued data addressed to that specific remote upon receiving the poll request.

ON_SLEEP goes high and $\overline{\text{CTS}}$ goes low each time the remote wakes, allowing for communication initiated by the remote host if desired.

### *Cyclic Sleep with Pin Wake-up mode (SM = 5)*

Use this mode to wake a sleeping remote device through either the RF interface or by de-asserting SLEEP_RQ for event-driven communications. The cyclic sleep mode works as described previously with the addition of a pin-controlled wake-up at the remote device. The SLEEP_RQ pin is level-triggered. The device wakes when a low is detected then set $\overline{\text{CTS}}$ low as soon as it is ready to transmit or receive.

Any activity resets the **ST** (Time before Sleep) timer, so the device goes back to sleep only after there is no activity for the duration of the timer. Once the device wakes (pin-controlled), it ignores further pin activity. The device transitions back into sleep according to the **ST** time regardless of the state of the pin.

## Sleep parameters

The following AT commands are associated with the sleep modes. See the linked commands for the parameter's description, range and default values.

- SM (Sleep Mode)
- SO (Sleep Options)
- A1 (End Device Association)
- ST (Time before Sleep)

- SP (Cyclic Sleep Period)
- DP (Disassociated Cyclic Sleep Period)

## Sleep current

The following table shows the sleep current during the XBee/XBee-PRO S2C 802.15.4 RF Module sleep modes.

| Sleep mode | SM command setting | Sleep current |
|---|---|---|
| Pin sleep | 1 | <1 µA @ 25°C |
| Cyclic sleep | 4 | <1 µA @ 25°C |
| Cyclic sleep with pin wake-up | 5 | <1 µA @ 25°C |

You can make devices use low sleep current by driving PWM outputs high during sleep and by using internal pull-ups/pull-downs on disabled/unused pins. The sleep pins are set up for sleeping as specified in Sleep pins. Additionally, pins that are outputs (other than PWM outputs) continue to output the same levels during sleep. Normally, this means that pins configured for output high or low will output high or low accordingly. However, if the output is overridden by I/O line passing, then the overridden output level is maintained during the sleep time.

## Sleep pins

The following table describes the three external device pins associated with sleep. For more details about the pins, see Pin signals.

| Pin name | Pin number | Description |
|---|---|---|
| SLEEP_RQ | TH pin 9/SMT pin 10 | For **SM** = 1, high puts the device to sleep and low wakes it up. For **SM** = 5, a high to low transition wakes the device up for **ST** time. The device ignores a low to high transition in **SM** = 5. |
| $\overline{\text{CTS}}$ | TH pin 12/SMT pin 25 | If **D7** = 1, high indicates that the device is asleep and low indicates that it is awake and ready to receive serial data. |
| $\overline{\text{ON_SLEEP}}$ | TH pin 13/SMT pin 26 | Low indicates that the device is asleep and high indicates that it is awake and ready to receive serial data. For the XBee/XBee-PRO S2C 802.15.4 RF Module, this pin cannot be configured to anything different. |

## Direct and indirect transmission

There are two methods to transmit data:

- Direct transmission: data is transmitted immediately to the Destination Address
- Indirect transmission: a packet is retained for a period of time and is only transmitted after the destination device (source address = destination address) requests the data.

Indirect transmissions can only occur on a Coordinator. Thus, if all nodes in a network are End Devices, only direct transmissions occurs. Indirect transmissions are useful to ensure packet delivery to a sleeping node. The Coordinator currently is able to retain up to five indirect messages.

### *Direct transmission*

A Coordinator can be configured to use only direct transmission by setting the **SP** (Cyclic Sleep Period) parameter to 0. Also, a Coordinator using indirect transmissions reverts to direct transmission if it knows the destination device is awake.

To enable this behavior, the **ST** (Time before Sleep) value of the Coordinator must be set to match the **ST** value of the End Device. Once the End Device either transmits data to the Coordinator or polls the Coordinator for data, the Coordinator uses direct transmission for all subsequent data transmissions to that device address until **ST** time occurs with no activity (at which point it reverts to using indirect transmissions for that device address). "No activity" means no transmission or reception of messages with a specific address. Broadcast messages do not reset the **ST** timer.

### *Indirect transmission*

To configure Indirect Transmissions in a Personal Area Network (PAN), the **SP** (Cyclic Sleep Period) parameter value on the Coordinator must be set to match the longest sleep value of any End Device. The sleep period value on the Coordinator determines how long (time or number of beacons) the Coordinator retains an indirect message before discarding it.

An End Device must poll the Coordinator once it wakes from Sleep to determine if the Coordinator has an indirect message for it. For Cyclic Sleep Modes, this is done automatically every time the device wakes (after **SP** time). For Pin Sleep Modes, the **A1** (End Device Association) parameter value must be set to enable Coordinator polling on pin wake-up (set in bit 3). Alternatively, an End Device can use the **FP** (Force Poll) command to poll the Coordinator as needed.

### *Acknowledgment*

If the transmission is not a broadcast message, the device expects to receive an acknowledgment from the destination node. If an acknowledgment is not received, the packet is resent up to three more times. If the acknowledgment is not received after all transmissions, an ACK failure is recorded.

# Node discovery

Node discovery has three variations as shown in the following table:

| Commands | Syntax | Description |
|---|---|---|
| Node Discovery | **ND** | Seeks to discover all nodes in the network (on the current PAN ID). |
| Directed Node Discovery | **ND** <NI String> | Seeks to discover if a particular node named <NI String> is found in the network. |
| Destination Node | **DN** <NI String> | Sets **DH**/**DL** to point to the MAC address of the node whose <NI String> matches. |

## Node discovery

The node discovery command (without an NI string designated) sends out a broadcast to every node in the PAN ID. Each node in the PAN sends a response back to the requesting node.

When the node discovery command is issued in AT command mode, all other AT commands are inhibited until the node discovery command times out, as specified by the **NT** parameter. After the timeout, an extra CRLF is output to the terminal window, indicating that new AT commands can be entered. This is the behavior whether or not there were any nodes that responded to the broadcast.

When the node discovery command is issued in API mode, the behavior is the same except that the response is output in API mode. If no nodes respond, there will be no responses at all to the node discover command. The requesting node is not able to process a new AT command until **NT** times out.

## Node discovery in compatibility mode

Node discovery (without an **NI** string parameter) in compatibility mode operates the same in compatibility mode as it does outside of compatibility mode with one minor exception:

If **C8** bit 1 is set and if requesting node is operating in API mode and if no responses are received by the time **NT** times out, then an API AT command response of OK (API frame type 0x88) is sent out the serial port rather than giving no response at all, which would happen if **C8** bit 1 is not set.

## Directed node discovery

The directed node discovery command (**ND** with an **NI** string parameter) sends out a broadcast to find a node in the network with a matching NI string. If such a node exists, it sends a response with its information back to the requesting node.

In Transparent mode, the requesting node will output an extra CRLF following the response from the designated node and the command will terminate, being ready to accept a new AT command. In the event that the requested node does not exist or is too slow to respond, the requesting node outputs an ERROR response after **NT** expires.

In API mode, the response from the requesting node will be output in API mode and the command will terminate immediately. If no response comes from the requested node, the requesting node outputs an error response in API mode after NT expires.

## Directed node discovery in compatibility mode

The behavior of the Legacy 802.15.4 module (S1 hardware) varies with the default behavior described above for the directed node discovery command. The Legacy module does not complete the command until **NT** expires, even if the requested node responds immediately. After **NT** expires, it gives a successful response, even if the requested node did not respond. To enable this behavior to be equivalent to the Legacy 802.15.4 module, set bit 1 of the **C8** parameter.

## Destination Node

The Destination Node command (**DN** with an **NI** string parameter) sends out a broadcast containing the **NI** string being requested. The responding node with a matching **NI** string sends its information back to the requesting node. The local node then sets **DH**/**DL** to match the address of the responding node. As soon as this response occurs, the command terminates successfully. If operating in AT command mode, an OK string is output and command mode exits. In API mode another AT command may be entered.

If an NI string parameter is not provided, the **DN** command terminates immediately with an error. If a node with the given NI string doesn't respond, the **DN** command terminates with an error after **NT** times out.

Unlike **ND** (with or without an **NI** string), **DN** does not cause the information from the responding node to be output; rather it simply sets **DH**/**DL** to the address of the responding node. If the responding node has a short address, then **DH**/**DL** is set to that short address (with **DH** at 0 and the ms 16 bits of

**DL** at 0). If the responding node has a long address (**MY** is FFFE), then **DH**/**DL** are set to the **SH**/**SL** of the responding node.

# Remote configuration commands

The API firmware has provisions to send configuration commands to remote devices using the Remote AT Command Request frame (0x17); see Remote AT Command Request frame - 0x17. You can use this frame to send commands to a remote device to read or set command parameters.

---

**CAUTION!** It is important to set the short address to 0xFFFE when sending to a long address. Any other value causes the long address to be ignored. This is particularly problematic in the case where nodes are set up with default addresses of 0 and the 16-bit address is erroneously left at 0. In that case, even with a correct long address the remote command goes out to all devices with the default short address of 0, potentially resulting in harmful consequences, depending on the command.

---

## Send a remote command

To send a remote command populate the Remote AT Command Request frame (0x17) with:

1. The 64-bit address and of the remote device.
2. The correct command options value.
3. The command and parameter data (optional). If and only if all nodes in the PAN have unique short addresses, then remote configuration commands can be sent to 16-bit short addresses by setting the short address in the API frame for Remote AT commands. In that case, the 64-bit address is unused and does not matter.

## Apply changes on remote devices

Any changes you make to the configuration command registers using AT commands do not take effect until you apply the changes. For example, if you send the **BD** command to change the baud rate, the actual baud rate does not change until you apply the changes. To apply changes:

1. Set the Apply Changes option bit in the Remote AT Command Request frame (0x17).
2. Issue an **AC** (Apply Changes) command to the remote device.
3. Issue a **WR** + **FR** command to the remote device to save changes and reset the device.

## Remote command responses

If the remote device receives a remote command request transmission, and the API frame ID is non-zero, the remote sends a remote command response transmission back to the device that sent the remote command. When a remote command response transmission is received, a device sends a remote command response API frame out its serial port. The remote command response indicates the status of the command (success, or reason for failure), and in the case of a command query, it includes the register value. The device that sends a remote command will not receive a remote command response frame if either of the following conditions exist:

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

# AT commands

Special commands ..... 67
Networking and security commands ..... 67
RF interfacing commands ..... 82
Sleep commands ..... 83
Serial interfacing commands ..... 85
I/O settings commands ..... 89
I/O line passing commands ..... 99
Diagnostic commands ..... 102
Command mode options ..... 104

# Special commands

The following commands are special commands.

## WR (Write)

Writes parameter values to non-volatile memory so that parameter modifications persist through subsequent resets.

Writing parameters to non-volatile memory does not apply the changes immediately. However, since the device uses non-volatile memory to determine initial configuration following reset, the written parameters are applied following a reset.

**Note** Once you issue a **WR** command, do not send any additional characters to the device until after you receive the **OK** response.

**Parameter range**

N/A

**Default**

N/A

## RE (Restore Defaults)

Restore device parameters to factory defaults. Does not exit out of Command mode.

**Parameter range**

N/A

**Default**

N/A

## FR (Software Reset)

Resets the device. The device responds immediately with an **OK** and performs a reset 100 ms later.

If you issue **FR** while the device is in Command Mode, the reset effectively exits Command mode.

**Parameter range**

N/A

**Default**

N/A

# Networking and security commands

The following AT commands are networking and security commands.

## C8 (802.15.4 Compatibility)

Sets or displays the operational compatibility with the Legacy 802.15.4 module (S1 hardware).

**Parameter range**

0 - 3

**Bit field**:

| Bit | Meaning | Setting | Description |
|---|---|---|---|
| 0[1] | TX compatibility | 0 | Transmissions are optimized as follows:<br>1. Maximum transmission size is affected by multiple factors (**MM**, **MY**, **DH**, **DL**, and **EE**). See Maximum payload rules. In the best case, with no app header, short source and destination addresses, and no encryption, the maximum transmission size is 116 bytes.<br>2. Multiple messages can be present simultaneously on the active queue, providing they are all destined for different addresses. This improves performance. |
| | | 1 | Transmissions operate like the Legacy 802.15.4 module, which means the following:<br>1. Maximum transmission size is 95 bytes for encrypted packets and 100 bytes for un-encrypted packets. These maximum transmission sizes are not adjusted upward for short addresses or for lack of an APP header.<br>2. Only one transmission message can be active at a time, even if other messages in the queue would go to a different destination address. |
| 1 | Node Discovery compatibility | 0 | Node discovery operates like other XBee devices and not like the Legacy 802.15.4 module. This means the following:<br>1. A directed **ND** request terminates after the single response arrives. This allows the device to process other commands without waiting for the **NT** to time out.<br>2. The device outputs an error response to the directed **ND** request if no response occurs within the time out. |
| | | 1 | The module operates like the Legacy 802.15.4 module, which has the following effect:<br>1. When the expected response arrives, the command remains active until **NT** times out. (**NT** defaults to 2.5 seconds.) This prevents the device from processing any other AT command, even if the desired response occurs immediately.<br>2. When the timeout occurs, the command silently terminates and indicates success, whether or not a response occurred within the **NT** timeout. |

[1]This bit does not normally need to be set. However, when the XBee/XBee-PRO S2C 802.15.4 RF Module is streaming broadcasts in transparent mode to a Legacy 802.15.4 module (S1 hardware), and **RR** > 0, then this bit does need to be set to avoid a watchdog reset on the Legacy 802.15.4 module.

**Default**

0

## CH (Operating Channel)

Set or read the operating channel devices used to transmit and receive data. The channel is one of three addressing configurations available to the device. The other configurations are the PAN ID (**ID** command) and destination addresses (**DL** and **DH** commands).

In order for devices to communicate with each other, they must share the same channel number. A network can use different channels to prevent devices in one network from listening to the transmissions of another. Adjacent channel rejection is 23 dB.

The command uses 802.15.4 channel numbers. Center frequency = 2405 MHz + (**CH** - 11 decimal) * 5 MHz.

**Parameter range**

0xB - 0x1A (XBee)

0x0C - 0x17 (XBee-PRO)

**Default**

0xC (12 decimal)

## ID (Network ID)

Set or read the user network identifier.

Devices must have the same network identifier to communicate with each other.

Devices can only communicate with other devices that have the same network identifier and channel configured.

Setting the **ID** parameter to **0xFFFF** indicates a global transmission for all PANs. It does not indicate a global receive.

**Parameter range**

0 - 0xFFFF

**Default**

0x3332 (13106 decimal)

## DH (Destination Address High)

Set or read the upper 32 bits of the 64-bit destination address. When you combine **DH** with **DL**, it defines the destination address that the device uses for transmissions in Transparent mode.

The destination address is also used for I/O sampling in both Transparent and API modes.

To transmit using a 16-bit address, set **DH** to 0 and **DL** less than 0xFFFF.

0x000000000000FFFF is the broadcast address. It is also used as the polling address when the device functions as end device.

**Parameter range**

0 - 0xFFFFFFFF

**Default**

0

## DL (Destination Address Low)

Set or display the lower 32 bits of the 64-bit destination address. When you combine **DH** with **DL**, it defines the destination address that the device uses for transmissions in Transparent mode.

The destination address is also used for I/O sampling in both Transparent and API modes.

**0x000000000000FFFF** is the broadcast address. It is also used as the polling address when the device functions as end device.

**Parameter range**

0 - 0xFFFFFFFF

**Default**

0

## MY (Source Address)

Sets or displays the device's 16-bit source address. Set **MY** = 0xFFFF to disable reception of packets with 16-bit addresses. Regardless of **MY**, messages addressed to the 64-bit long address of the device are always delivered.

**Parameter range**

0 - 0xFFFF

**Default**

0

## SH (Serial Number High)

Displays the upper 32 bits of the unique IEEE 64-bit extended address assigned to the product family in the factory.

The 64-bit source address is always enabled. This value is read-only and it never changes.

**Parameter range**

0 - 0xFFFFFFFF [read-only]

**Default**

Set in the factory

## SL (Serial Number Low)

Displays the lower 32 bits of the unique IEEE 64-bit RF extended address assigned to the product family in the factory.

The device's serial number is set at the factory and is read-only.

**Parameter range**

0 - 0xFFFFFFFF [read-only]

**Default**

Set in the factory

## MM (MAC Mode)

The **MM** command is used to set and read the MAC Mode value. The **MM** command disables/enables the use of a Digi header contained in the 802.15.4 RF packet. By default (**MM** = 0), Digi Mode is enabled and the module adds an extra header to the data portion of the 802.15.4 packet. This enables the following features:

- **ND** and **DN** command support
- Duplicate packet detection when using ACKs
- Remote command support
- **RR** command
- DIO/AIO sampling support
- OTA firmware updates

The **MM** command allows users to turn off the use of the extra header. Modes 1 and 2 are strict 802.15.4 modes. If the Digi header is disabled, the features above are also disabled.

When **MM** = **1** or **3**, MAC retries are not supported.

When the Digi header is disabled, encrypted data that is not valid will be sent out of the UART and not filtered out.

**Parameter range**

0 - 3

| Parameter | Configuration |
|---|---|
| 0 | Digi Mode (802.15.4 + Digi header) |
| 1 | 802.15.4 (no ACKs) |
| 2 | 802.15.4 (with ACKs) |
| 3 | Digi Mode (no ACKs) |

**Default**

0

## RR (XBee Retries)

Set or reads the maximum number of retries the device executes in addition to the three retries provided by the 802.15.4 MAC. For each device retry, the 802.15.4 MAC can execute up to three retries.

The following applies for broadcast messages: If **RR** = **0**, only one packet is broadcast. If **RR** is > **0**, **RR** + 2 packets are sent on each broadcast. No acknowledgments are returned on a broadcast.

This value does not need to be set on all devices for retries to work. If retries are enabled, the transmitting device sets a bit in the Digi RF Packet header that requests the receiving device to send an ACK. If the transmitting device does not receive an ACK within 200 ms, it re-sends the packet within a random period up to 48 ms. Each device retry can potentially result in the MAC sending the packet four times (one try plus three retries). Retries are not attempted for indirect messages that are purged.

**Parameter range**

0 - 6

**Default**

0

## RN (Random Delay Slots)

Sets or displays the minimum value of the back-off exponent in the CSMA-CA algorithm. The Carrier Sense Multiple Access - Collision Avoidance (CSMA-CA) algorithm was engineered for collision avoidance(random delays are inserted to prevent data loss caused by data collisions.

If **RN** = 0, there is no delay between a request to transmit and the first iteration of CSMA-CA.

Unlike CSMA-CD, which reacts to network transmissions after collisions have been detected, CSMA-CA acts to prevent data collisions before they occur. As soon as a device receives a packet that is to be transmitted, it checks if the channel is clear (no other device is transmitting). If the channel is clear, the packet is sent over-the-air. If the channel is not clear, the device waits for a randomly selected period of time, then checks again to see if the channel is clear. After a time, the process ends and the data is lost.

**Parameter range**

0 - 3 (exponent)

**Default**

0

## ND (Network Discovery)

Discovers and reports all devices found in the network. For each discovered device, the following information is returned after a jittered time delay:

Node discover response when issued in Command mode:

MY<CR> (2 bytes) (always 0xFFFE)

SH<CR> (4 bytes)

SL<CR> (4 bytes)

DB<CR> (Contains the detected signal strength of the response in negative dBm units)

NI <CR> (variable, 0-20 bytes plus 0x00 character)

PARENT_NETWORK ADDRESS<CR> (2 bytes)

DEVICE_TYPE<CR> (1 byte: **0** = Coordinator, **1** = Router, **2** = End Device)

STATUS<CR> (1 byte: reserved)

PROFILE_ID<CR> (2 bytes)

MANUFACTURER_ID<CR> (2 bytes)

DIGI DEVICE TYPE<CR> (4 bytes. Optionally included based on **NO** settings.)

RSSI OF LAST HOP<CR> (1 byte. Optionally included based on **NO** settings.)

A second carriage return will indicate the network discovery timeout (**NT**) has expired.

When operating in API mode and a Network Discovery is issued as a 0x08 or 0x09 frame, the response will contain binary data except for the **NI** string in the following format:

2 bytes for Short Source Address

4 bytes for Upper Long Address

4 bytes for Lower Long Address

1 byte for the signal strength in -dBm (two's compliment representation)

NULL-terminated string for **NI (Node Identifier)** value (maximum 20 bytes without NULL terminator)

Each device that responds to the request will generate a separate AT Command Response frame - 0x88.

Broadcast an **ND** command to the network. If the command includes an optional node identifier string parameter, only those devices with a matching **NI** string respond without a random offset delay. If the command does not include a node identifier string parameter, all devices respond with a random offset delay.

The **NT** setting determines the maximum timeout (13 seconds by default), this value is sent along with the discovery broadcast and determines the random delay the remote nodes use to prevent the responses from colliding.

For more information about options that affect the behavior of the **ND** command Refer to the description of the **NO** command for options which affect the behavior of the **ND** command.

---

**WARNING!** If the **NT** setting is small relative to the number of devices on the network, responses may be lost due to channel congestion. Regardless of the **NT** setting, because the random offset only mitigates transmission collisions, getting responses from all devices in the network is not guaranteed.

---

**Parameter range**

20-byte printable ASCII string

**Default**

N/A

## NT (Node Discover Timeout)

Sets the amount of time a base node waits for responses from other nodes when using the **ND** (Node Discover) command. The **NT** value is transmitted with the **ND** command.

Remote nodes set up a random hold-off time based on this time. Once the **ND** command has ended, the base discards any response it receives.

**Parameter range**

0x1 - 0xFC (x 100 ms)

**Default**

0x19 (2.5 decimal seconds)

## NO (Node Discovery Options)

Use **NO** to suppress or include a self-response to **ND** (Node Discover) commands. When **NO** bit 1 = 1, a device performing a Node Discover includes a response entry for itself.

**Parameter range**

0 - 1

**Default**

0x0

## DN (Discover Node)

Resolves an **NI** (Node identifier) string to a physical address (case sensitive).

The following events occur after **DN** discovers the destination node:

When **DN** is sent in Command mode :

1. The device sets **DL** and **DH** to the address of the device with the matching **NI** string.
   The address selected (either 16-bit short address or 64-bit extended address) is chosen based on the destination device's **MY** command configuration.
2. The receiving device returns **OK** (or **ERROR**).
3. The device exits Command mode to allow for immediate communication. If an ERROR is received, then Command mode does not exit.

When **DN** is sent as a local AT Command API frame:

1. The receiving device returns the 16-bit network and 64-bit extended addresses in an API Command Response frame.
2. If there is no response from a module within (**NT*** 100) milliseconds or you do not specify a parameter (by leaving it blank), the receiving device returns an ERROR message. In the case of an ERROR, the device does not exit command mode. Set the radius of the **DN** command using the **BH** command.

When **DN** is sent as a local AT Command frame - 0x08:

1. The receiving device returns a success response in a AT Command Response frame - 0x88.
2. If there is no response from a module within (**NT** * 100) milliseconds or you do not specify a parameter (by leaving it blank), the receiving device returns an ERROR message.

**Parameter range**

20-byte ASCII string

**Default**

N/A

## CE (Coordinator Enable)

The routing and messaging mode of the device.

**Parameter range**

0 - 1

| Parameter | Description |
|---|---|
| 0 | End Device |
| 1 | Coordinator |

**Default**

0

## SC (Scan Channels)

Sets or displays the list of channels to scan for all Active and Energy Scans as a bit field. This affects scans initiated in the **AS** (Active Scan) and **ED** (Energy Scan) commands in Command mode and during End Device Association and Coordinator startup.

**Parameter range**

0 - 0xFFFF (bit field)

**Bit field mask:**

| Bit | Parameter |
|---|---|
| 0 | 0x0B (not available on XBee-PRO) |
| 1 | 0x0C |
| 2 | 0x0D |
| 3 | 0x0E |
| 4 | 0x0F |
| 5 | 0x10 |
| 6 | 0x11 |
| 7 | 0x12 |
| 8 | 0x13 |
| 9 | 0x14 |
| 10 | 0x15 |
| 11 | 0x16 |
| 12 | 0x17 |
| 13 | 0x18 (not available on XBee-PRO) |
| 14 | 0x19 (not available on XBee-PRO) |
| 15 | 0x1A (not available on XBee-PRO) |

**Default**

0x1FFE

## SD (Scan Duration)

Sets or displays the scan duration exponent.

**Coordinator:** If you set the **ReassignPANID** option on the coordinator (refer to A2 (Coordinator Association)), **SD** determines the length of time the coordinator scans channels to locate existing

PANs. If you set the **ReassignChannel** option, **SD** determines how long the coordinator performs an Energy Scan to determine which channel it will operate on.

**End Device:** Duration of Active Scan during Association. In a Beacon system, set **SD**=**BE** of the coordinator. **SD** must be set at least to the highest **BE** parameter of any Beaconing Coordinator with which an end device or coordinator wants to discover.

Scan Time is measured as:

([# of channels to scan] * (2 ^**SD**) * 15.36 ms) + (38 ms * [# of channels to scan]) + 20 ms

Use the **SC** (Scan Channels) command to set the number of channels to scan. The XBee can scan up to 16 channels (**SC** = 0xFFFF). The XBee-PRO can scan up to 13 channels (**SC**= 0x1FFE).

**SD** influences the time the MAC listens for beacons or runs an energy scan on a given channel.

**Example**

The following table shows the results for a thirteen channel scan.

| SD setting | Time |
|---|---|
| 0 | 0.18 s |
| 2 | 0.74 s |
| 4 | 2.95 s |
| 6 | 11.80 s |
| 8 | 47.19 s |
| 10 | 3.15 min |
| 12 | 12.58 min |
| 14 | 50.33 min |

**Parameter range**

0 - 0x0F (exponent)

**Default**

4

## A1 (End Device Association)

Sets or displays the End Device association options.

**Parameter range**

0 - 0x0F (bit field)

**Bit field:**

| Bit | Meaning | Setting | Description |
|---|---|---|---|
| 0 | Allow PanId reassignment | 0 | Only associates with Coordinator operating on PAN ID that matches device ID. |
| | | 1 | May associate with Coordinator operating on any PAN ID. |
| 1 | Allow Channel reassignment | 0 | Only associates with Coordinator operating on matching **CH** channel setting. |
| | | 1 | May associate with Coordinator operating on any channel. |
| 2 | Auto Associate | 0 | Device will not attempt association. |
| | | 1 | Device attempts association until success. |
| 3 | Poll coordinator on pin wake | 0 | Pin Wake does not poll the Coordinator for indirect (pending) data. |
| | | 1 | Pin Wake sends Poll Request to Coordinator to extract any pending data. |
| 4 - 7 | Reserved | | |

**Default**

0

## A2 (Coordinator Association)

Sets or displays the Coordinator association options.

**Parameter range**

0 - 7 (bit field)

**Bit field:**

| Bit | Meaning | Setting | Description |
|---|---|---|---|
| 0 | Allow Pan ID reassignment | 0 | Coordinator will not perform Active Scan to locate available PAN ID. It operates on ID (PAN ID). |
| | | 1 | Coordinator performs an Active Scan to determine an available **ID** (PAN ID). If a PAN ID conflict is found, the **ID** parameter will change. |
| 1 | Allow Channel reassignment | 0 | Coordinator will not perform Energy Scan to determine free channel. It operates on the channel determined by the **CH** parameter. |
| | | 1 | Coordinator performs an Energy Scan to find the quietest channel, then operates on that channel. |
| 2 | Allow Association | 0 | Coordinator will not allow any devices to associate to it. |
| | | 1 | Coordinator allows devices to associate to it. |
| 3 - 7 | Reserved | | |

The binary equivalent of the default value (0x06) is 00000110. ‘Bit 0’ is the last digit of the sequence.

**Default**

0

## AI (Association Indication)

Reads the Association status code to monitor association progress. The following table provides the status codes and their meanings.

| Status code | Meaning |
|---|---|
| 0x00 | Coordinator successfully started, End device successfully associated, or operating in peer to peer mode where no association is needed. |
| 0x03 | Active Scan found a PAN coordinator, but it isn't currently accepting associations. |
| 0x04 | Active Scan found a PAN coordinator in a beacon-enabled network, which is not a supported feature. |
| 0x05 | Active Scan found a PAN, but the PAN ID doesn't match the configured PAN ID on the requesting end device and bit 0 of **A1** is not set to allow reassignment of PAN ID. |
| 0x06 | Active Scan found a PAN on a channel that does not match the configured channel on the requesting end device and bit 1 of **A1** is not set to allow reassignment of the channel. |
| 0x0C | Association request failed to get a response. |
| 0x13 | End device is disassociated or is in the process of disassociating. |
| 0xFF | Initialization time; no association status has been determined yet. |

**Parameter range**

0 - 0x13 [read-only]

**Default**

N/A

## DA (Force Disassociation)

Causes the End Device to immediately disassociate from a Coordinator (if associated) and re-attempt to associate.

**Parameter range**

-

**Default**

-

## FP (Force Poll)

The **FP** command is deferred until changes are applied. This prevents indirect messages from arriving at the end device while it is operating in Command mode.

**Parameter range**

N/A

**Default**

N/A

## AS (Active Scan)

Sends a Beacon Request to a Broadcast address (**0xFFFF**) and Broadcast PAN (**0xFFFF**) on every channel in **SC**. **SD** determines the amount of time the device listens for Beacons on each channel. A PanDescriptor is created and returned for every Beacon received from the scan. Each PanDescriptor contains the following information:

CoordAddress (**SH** + **SL** parameters)<CR>

**Note** If **MY** on the coordinator is set less than 0xFFFF, the **MY** value is displayed.

CoordPanID (**ID** parameter)<CR>

CoordAddrMode <CR>

0x02 = 16-bit Short Address

0x03 = 64-bit Long Address

Channel (**CH** parameter) <CR>

SecurityUse<CR> - will always report 0x00

ACLEntry<CR> - will always report 0x00

SecurityFailure<CR> - will always report 0x00

SuperFrameSpec<CR> (2 bytes):

bit 15 - Association Permitted (MSB) - depending on bit 3 of A2 (Coordinator Association)

bit 14 - PAN Coordinator

bit 13 - Reserved

bit 12 - Battery Life Extension

bits 8-11 - Final CAP Slot

bits 4-7 - Superframe Order

bits 0-3 - Beacon Order

GtsPermit<CR>

RSSI<CR> (- RSSI is returned as -dBm)

TimeStamp<CR> (3 bytes)

<CR> (A carriage return <CR> is sent at the end of the **AS** command)

The Active Scan is capable of returning up to five PanDescriptors in a scan. The actual scan time on each channel is measured as:

Time = [(2 ^ (**SD** Parameter)) * 15.36] ms.

Total scan time is this time multiplied by the number of channels to be scanned (as determined by the **SC** parameter).

Refer to the scan table in SD (Scan Duration) to determine scan times. If using API Mode, no <CR>'s are returned in the response. For more information, see Operate in API mode. If no PANs are discovered during the scan, only one carriage return is printed.

**Parameter range**

0 - 6

**Default**

N/A

## ED (Energy Detect)

Starts an energy detect scan. This command accepts an argument to specify the time in milliseconds to scan all channels. The device loops through all the available channels until the time elapses. It returns the maximal energy on each channel, a comma follows each value, and the list ends with a carriage return. The values returned reflect the energy level that **ED** detects in -dBm units.

**Parameter range**

0 - 6

**Default**

N/A

## EE (Encryption Enable)

Enables or disables 128-bit Advanced Encryption Standard (AES) encryption.

Set this command parameter the same on all devices in a network.

The firmware uses the 802.15.4 Default Security protocol and uses AES encryption with a 128-bit key. AES encryption dictates that all devices in the network use the same key, and that the maximum RF packet size is 95 bytes if Tx compatibility is enabled (you set bit 0 of **C8**). If **C8**, bit 0 is not set, see Maximum payload.

When encryption is enabled, the device always uses its 64-bit long address as the source address for RF packets. This does not affect how the **MY** (Source Address), **DH** (Destination Address High) and **DL** (Destination Address Low) parameters work.

If **MM** (MAC Mode) is set to 1 or 2 and **AP** (API Enable) parameter > 0:

> With encryption enabled and a 16-bit short address set, receiving devices can only issue RX (Receive) 64-bit indicators. This is not an issue when **MM** = 0 or 3.

If a device with a non-matching key detects RF data, but has an incorrect key:

When encryption is enabled, non-encrypted RF packets received are rejected and are not sent out the UART.

**Parameter range**

0 - 1

| Parameter | Description |
|---|---|
| 0 | Encryption Disabled |
| 1 | Encryption Enabled |

**Default**

0

## KY (AES Encryption Key)

Sets the 128-bit network security key value that the device uses for encryption and decryption.

This command is write-only. If you attempt to read **KY**, the device returns an **OK** status.

Set this command parameter the same on all devices in a network.

The entire payload of the packet is encrypted using the key and the CRC is computed across the ciphertext.

**Parameter range**

128-bit value (up to 16 bytes)

**Default**

0

## NI (Node Identifier)

Stores the node identifier string for a device, which is a user-defined name or description of the device. This can be up to 20 ASCII characters.

- XCTU prevents you from exceeding the string limit of 20 characters for this command. If you are using another software application to send the string, you can enter longer strings, but the software on the device returns an error.

Use the **ND** (Network Discovery) command with this string as an argument to easily identify devices on the network.

The **DN** command also uses this identifier.

**Parameter range**

A string of case-sensitive ASCII printable characters from 1 to 20 bytes in length. The string cannot start with the space character. A carriage return or a comma automatically ends the command.

**Default**

0x20 (an ASCII space character)

## NP (Maximum Packet Payload Bytes)

Reads the maximum number of RF payload bytes that you can send in a transmission.

**NP** is based on multiple factors including the length of the source address, the length of the destination address, the length of the APP header, and whether or not encryption is enabled.

For the purposes of this command, it always assumes a long destination address. This means that if you select a short destination address, you will be able to send up to **NP** + 6 bytes in a single packet.

**Note** **NP** returns a hexadecimal value. For example, if **NP** returns 0x66, this is equivalent to 102 bytes.

**Parameter range**

0 - 0xFFFF (bytes) [read-only]

**Default**

N/A

# RF interfacing commands

The following AT commands are RF interfacing commands.

## PL (TX Power Level)

Sets or displays the power level at which the device transmits conducted power. Power levels are approximate.

For XBee-PRO, **PL**= 4 is calibrated and the remaining power levels are approximate. The device recalibrates its power setting every 15 seconds based on factory calibration settings and the current temperature.

For XBee, **PL** = 4, **PM** = 1 is tested at the time of manufacturing. Other power levels are approximate. On channel 26, transmitter power will not exceed -5 dBm.

**Parameter range**

0 - 4

The following table shows the TX power versus the **PL** setting.

**XBee modules**

| PL setting | PM setting | Channel(s) | TX power* (dBm) |
|---|---|---|---|
| 4 | 1 | 11 to 25 | 8 |
| 4 | 0 | 11 to 25 | 5 |
| 3 | 1 | 11 to 25 | 6 |
| 3 | 0 | 11 to 25 | 3 |
| 2 | 1 | 11 to 25 | 4 |
| 2 | 0 | 11 to 25 | 1 |
| 1 | 1 | 11 to 25 | 2 |
| 1 | 0 | 11 to 25 | -1 |
| 0 | 1 | 11 to 25 | -2 |
| 0 | 0 | 11 to 25 | -5 |
| X | 1 | 26 | -5 |
| X | 0 | 26 | -8 |
| * Highest power level is tested during manufacturing. Other power levels are approximate. | | | |

**XBee-PRO modules**

| PL setting | Channel(s) | TX power* (dBm) |
|---|---|---|
| 0 | 12 to 23 | 0 |
| 1 | 12 to 23 | 12 |

| PL setting | Channel(s) | TX power* (dBm) |
|---|---|---|
| 2 | 12 to 23 | 15 |
| 3 | 12 to 23 | 16 |
| 4 | 12 to 23 | 18 |
| * Highest power level is tested during manufacturing. Other power levels are approximate. | | |

**Default**

4

## PM (Power Mode)

Set or read the power mode of the device. Enabling boost mode improves the receive sensitivity by 2dB and increase the transmit power by 3dB.

**Parameter range**

0 - 1

| Setting | Meaning |
|---|---|
| 0 | Boost mode disabled |
| 1 | Boost mode enabled |

**Default**

1

## CA (CCA Threshold)

Set or read the Clear Channel Assessment (CCA) threshold. Prior to transmitting a packet, the device performs a CCA to detect energy on the channel. If the device detects energy above the CCA threshold, it will not transmit the packet.

The **CA** parameter is measured in units of -dBm.

**Note** If device is operating in Europe, this value must be set to 0x34 to comply with EN 300 328 Listen Before Talk requirements. Alternatively the device can be set to **PL3** as explained in Europe (CE).

**Parameter range**

0x28 - 0x50

**Default**

0x2C (-44 decimal dBm)

# Sleep commands

The following AT commands are sleep commands.

## SM (Sleep Mode)

Sets or displays the sleep mode of the device.

By default, Sleep Modes are disabled (**SM** = **0**) and the device remains in Idle/Receive mode. When in this state, the device is constantly ready to respond to either serial or RF activity.

**Parameter range**

0, 1, 4, 5

| Parameter | Description |
|---|---|
| 0 | No sleep (disabled) |
| 1 | Pin sleep |
| 2 | Reserved |
| 3 | Reserved |
| 4 | Cyclic Sleep Remote |
| 5 | Cyclic Sleep Remote with pin wakeup |

**Default**

0

## ST (Time before Sleep)

Sets or displays the wake time of the device.

The **ST** parameter is only valid for end devices configured with Cyclic Sleep settings (**SM** = 4 - 5) and for coordinators.

Coordinator and End Device **ST** values must be equal.

**Parameter range**

1 - 0xFFFF (x 1 ms)

**Default**

0x1388 (5 seconds)

## SP (Cyclic Sleep Period)

Sets and reads the duration of time that a remote device sleeps. After the cyclic sleep period is over, the device wakes and checks for data. If data is not present, the device goes back to sleep.

The **SP** parameter is only valid if you configure the end device to operate in Cyclic Sleep (**SM** = 4-5). Coordinator and End Device **SP** values should always be equal.

To send direct messages on a coordinator, set **SP** = **0**.

**End Device**: **SP** determines the sleep period for cyclic sleeping remotes.

**Coordinator**: If non-zero, **SP** determines the time to hold an indirect message before discarding it. A Coordinator discards indirect messages after a period of (2.5 * **SP**).

**Parameter range**

0 - 0x15F900

**Default**

0

## DP (Disassociated Cyclic Sleep Period)

Sets or displays the sleep period for cyclic sleeping remotes that are configured for Association but that are not associated to a Coordinator. For example, if a device is configured to associate and is configured as a Cyclic Sleep remote, but does not find a Coordinator, it sleeps for **DP** time before reattempting association.

**Parameter range**

1 - 0xFFFF

**Default**

0x3E8 (10 seconds)

## SO (Sleep Options)

Set or read the sleep options bit field of a device. This command is a bitmask.

You can set or clear any of the available sleep option bits.

**Parameter range**

0 - 0x3

**Bit field:**

| Bit | Setting | Meaning | Description |
|---|---|---|---|
| 0x01 | 0 | Normal operations | A device configured for cyclic sleep will poll for data on waking |
| | 1 | Disable wakeup poll | A device configured for cyclic sleep will not poll for data on waking |
| 0x02 | 0 | Normal operations | A device configured in a sleep mode with ADC/DIO sampling enabled will automatically perform a sampling on wakeup |
| | 1 | Suppress sample on wakeup | A device configured in a sleep mode with ADC/DIO sampling enabled will not automatically sample on wakeup |
| Set all other option bits to 0. | | | |

**Default**

0

# Serial interfacing commands

The following AT commands are serial interfacing commands.

## BD (Interface Data Rate)

To request non-standard baud rates with values between 1200 b/s and 250,000 b/s (0x4B0 and 0x3D090), you can use the Serial Console toolbar in XCTU to configure the serial connection (if the console is connected), or click the **Connect** button (if the console is not yet connected).

When you send non-standard baud rates to a device, it stores the closest interface data rate represented by the number in the **BD** register. Read the **BD** command by sending **ATBD** without a parameter value, and the device returns the value stored in the **BD** register.

The RF data rate is not affected by the **BD** parameter.

### *Non-standard interface data rates*

The firmware interprets any value from 0x4B0 through 0x3D090 as an actual baud rate. When the firmware cannot configure the exact rate specified, it configures the closest approximation to that rate. For example, to set a rate of 57600 b/s send the following command line: **ATBDE100**. Then, to find out the closest approximation, send **ATBD** to the console window. It sends back a response of 0xE0D1, which is the closest approximation to 57600 b/s attainable by the hardware.

**Note** When using XCTU, you can only set and read non-standard interface data rates using the XCTU **Terminal** tab. You cannot access non-standard rates through the **Modem Configuration** tab.

When you send the **BD** command with a non-standard interface data rate, the UART adjusts to accommodate the interface rate you request. In most cases, the clock resolution causes the stored **BD** parameter to vary from the sent parameter. Sending **ATBD** without an associated parameter value returns the value actually stored in the device's **BD** register.

The following table provides the parameters sent versus the parameters stored.

| BD parameter sent (HEX) | Interface data rate (b/s) | BD parameter stored (HEX) |
|---|---|---|
| 0 | 1200 | 0 |
| 4 | 19,200 | 4 |
| 7 | 115,200 | 7 |
| E100 | 57,600 | E0D1 |
| 1C200 | 115,200 | 1C2B8 |

**Parameter range**

Standard baud rates: 0x0 - 0x8

| Parameter | Description |
|---|---|
| 0x0 | 1200 b/s |
| 0x1 | 2400 b/s |
| 0x2 | 4800 b/s |
| 0x3 | 9600 b/s |
| 0x4 | 19200 b/s |

| Parameter | Description |
|---|---|
| 0x5 | 38400 b/s |
| 0x6 | 57600 |
| 0x7 | 115200 b/s |
| 0x8 | 230400 b/s |

**Default**

0x03 (9600 b/s)

## NB (Parity)

Set or read the serial parity settings for UART communications.

The device does not actually calculate and check the parity. It only interfaces with devices at the configured parity and stop bit settings.

**Parameter range**

0x00 - 0x04

| Parameter | Description |
|---|---|
| 0x00 | No parity |
| 0x01 | Even parity |
| 0x02 | Odd parity |
| 0x03 | Mark parity (forced high) |

**Default**

0x00

## RO (Packetization Timeout)

Set or read the number of character times of inter-character silence required before transmission begins when operating in Transparent mode. RF transmission will also commence when the maximum payload (see Maximum payload) is received in the DI buffer.

Set **RO** to **0** to transmit characters as they arrive instead of buffering them into one RF packet.

**Parameter range**

0 - 0xFF (x character times)

**Default**

3

## D7 (DIO7/CTS)

Sets or displays the DIO7/$\overline{\text{CTS}}$ configuration (TH pin 12/SMT pin 25).

**Parameter range**

0, 1, 3 - 7

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | $\overline{\text{CTS}}$ flow control |
| 2 | N/A |
| 3 | Digital input |
| 4 | Digital output, low |
| 5 | Digital output, high |
| 6 | RS-485 Tx enable, low Tx (0 V on transmit, high when idle) |
| 7 | RS-485 Tx enable high, high Tx (high on transmit, 0 V when idle) |

**Default**

0x1

## D6 (DIO6/RTS)

Sets or displays the DIO6/$\overline{\text{RTS}}$ configuration (TH pin 16/SMT pin 29).

**Parameter range**

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | $\overline{\text{RTS}}$ flow control |
| 2 | N/A |
| 3 | Digital input |
| 4 | Digital output, low |
| 5 | Digital output, high |

**Default**

0

## AP (API Enable)

Set or read the API mode setting. The device can format the RF packets it receives into API frames and send them out the serial port.

When you enable API, you must format the serial data as API frames because Transparent operating mode is disabled.

Enables API Mode.

**Parameter range**

0 - 2

| Parameter | Description |
|---|---|
| 0 | Transparent mode, API mode is off. All UART input and output is raw data and the device uses the **RO** parameter to delineate packets. |
| 1 | API Mode Without Escapes. The device packetizes all UART input and output data in API format, without escape sequences. |
| 2 | API Mode With Escapes. The device is in API mode and inserts escaped sequences to allow for control characters. When XON (0x11), XOFF (0x13), Escape (0x7D), and start delimiter 0x7E are data, these bytes are escaped so that they never appear in the data and so that they can always be interpreted with their special meanings. |

**Default**

0

# I/O settings commands

The following AT commands are I/O settings commands.

## D0 (DIO0/AD0)

Sets or displays the DIO0/AD0 configuration (TH pin 20/SMT pin 33).

**Parameter range**

0, 2 - 5

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 0 | Unmonitored digital input |
| 1 | N/A |
| 2 | ADC |
| 3 | Digital input |
| 4 | Digital output, low |
| 5 | Digital output, high |

**Default**

0

## D1 (DIO1/AD1)

Sets or displays the DIO1/AD1 configuration (TH pin 19/SMT pin 32).

**Parameter range**

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | Commissioning button |
| 1 | SPI_$\overline{\text{ATTN}}$ for the through-hole device |
| 2 | ADC |
| 3 | Digital input |
| 4 | Digital output, low |
| 5 | Digital output, high |
| 6 | PTI_EN |

**Default**

0

## D2 (DIO2/AD2)

Sets or displays the DIO2/AD2 configuration (TH pin 18/SMT pin 31).

**Parameter range**

0 - 5

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | SPI_CLK for through-hole devices |
| 2 | ADC |
| 3 | Digital input |
| 4 | Digital output, low |
| 5 | Digital output, high |

**Default**

0

## D3 (DIO3/AD3)

Sets or displays the DIO3/AD3 configuration (TH pin 17/SMT pin 30).

**Parameter range**

0 - 5

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 0 | Unmonitored digital input |
| 1 | SPI_$\overline{\text{SSEL}}$ for the through-hole device |
| 1 | SPI slave select |
| 2 | ADC |
| 3 | Digital input |
| 4 | Digital output, low |
| 5 | Digital output, high |

**Default**

0

## D4 (DIO4)

Sets or displays the DIO4 configuration (TH pin 11/SMT pin 24).

**Parameter range**

0, 1, 3 - 5

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 0 | Unmonitored digital input |
| 1 | SPI_MOSI for the through-hole device |
| 2 | N/A |
| 3 | Digital input |
| 4 | Digital output, low |
| 5 | Digital output, high |

**Default**

0

## D5 (DIO5/ASSOCIATED_INDICATOR)

Sets or displays the DIO5/ASSOCIATED_INDICATOR configuration (TH pin 15/SMT pin 28).

**Parameter range**

0, 1, 3 - 5

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | Associate LED indicator - blinks when associated |
| 2 | N/A |
| 3 | Digital input |
| 4 | Digital output, default low |
| 5 | Digital output, default high |

**Default**

1

## D8 (DIO8/SLEEP_REQUEST)

Sets or displays the DI8/DTR/SLP_RQ configuration (TH pin 9/SMT pin 10).

The XBee/XBee-PRO S2C 802.15.4 RF Module does not support sleep. The SLEEP_REQUEST option is provided for compatibility purposes and does not affect the device.

This line is also used with Pin Sleep, but pin sleep ignores the **D8** configuration. It is always used to control pin sleep, regardless of configuration of **D8**.

**Note** When using Pin sleep, the internal pull up/down resistor is not used

**Parameter range**

0, 3

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | N/A |
| 2 | N/A |
| 3 | Digital input |
| 4 | N/A |
| 5 | N/A |

**Default**

0

## P0 (RSSI/PWM0 Configuration)

Sets or displays the RSSI/PWM0 configuration ().

This command enables the option of translating incoming data to a PWM so that the output can be translated back into analog form.

If the **IA** (I/O Input Address) parameter is correctly set and **P0** is configured as PWM0 output, incoming AD0 samples automatically modify the PWM0 value.

**Parameter range**

0 - 2

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | RSSI PWM0 output |
| 2 | PWM0 output |

**Default**

1

## P1 (PWM1 Configuration)

Sets or displays the PWM1 configuration (TH pin 7/SMT pin 8).

If **IA** (I/O Input Address) is correctly set and **P1** is configured as PWM1 output, incoming AD0 samples automatically modify the PWM1 value.

**Parameter range**

0, 2

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 2 | PWM1 output |

**Default**

0

## P2 (SPI_MISO)

Sets or displays the SPI_MISO configuration (TH pin 4/SMT pin 17). This only applies to through-hole devices.

**Parameter range**

0 - 1

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | SPI_MISO |

**Default**

0

## M0 (PWM0 Duty Cycle)

The duty cycle of the PWM0 line (TH pin 6/SMT pin 7).

Use the **P0** command to configure the line as a PWM output.

If the **IA** (I/O Input Address) parameter is correctly set and **P0** is configured as PWM0 output, incoming AD0 samples automatically modify the PWM0 value.

Before setting the line as an output:

1. Enable PWM0 output (**P0** = **2**).
2. Apply settings (use **CN** or **AC**).

To configure the duty cycle of PWM0:

The PWM period is 64 µs and there are 0x03FF (1023 decimal) steps within this period. When **M0** = **0** (0% PWM), 0x01FF (50% PWM), 0x03FF (100% PWM), and so forth.

**Parameter range**

0 - 0x3FF

**Default**

0

## M1 (PWM1 Duty Cycle)

The duty cycle of the PWM1 line (TH pin 7/SMT pin 8).

Use the **P1** command to configure the line as a PWM output.

Before setting the line as an output:

1. Enable PWM1 output (**P1** = **2**).
2. Apply settings (use **CN** or **AC**)

**Parameter range**

0 - 0x3FF

**Default**

0

## P5 (SPI_MISO)

Sets or displays the SPI_MISO configuration.

This only applies to surface-mount devices.

**Parameter range**

0, 1

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | SPI_MISO |

**Default**

1

## P6 (SPI_MOSI Configuration)

Sets or displays the SPI_MOSI configuration.
This only applies to surface-mount devices.

**Parameter range**

0, 1

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | SPI_MOSI |

**Default**

1

## P7 (SPI_SSEL )

Sets or displays the SPI_$\overline{\text{SSEL}}$ configuration.
This only applies to surface-mount devices.

**Parameter range**

1, 2

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | SPI_$\overline{\text{SSEL}}$ |

**Default**

1

## P8 (SPI_SCLK)

Sets or displays the DIO18/SPI_CLK configuration.
This only applies to surface-mount devices.

**Parameter range**

1, 2

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | SPI_SCLK |

**Default**

1

## P9 (SPI_ATTN)

Sets or displays the SPI_$\overline{\text{ATTN}}$ configuration (pin 12).
This only applies to surface-mount devices.

**Parameter range**

1, 2

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | SPI_$\overline{\text{ATTN}}$ |

**Default**

1

## PR (Pull-up/Down Resistor Enable)

The bit field that configures the internal pull-up/down resistor status for the I/O lines.

- If you set a **PR** bit to 1, it enables the pull-up/down resistor
- If you set a **PR** bit to 0, it specifies no internal pull-up/down resistor.

The **PD** (Pull Direction) parameter determines the direction of the internal pull-up/down resistor.

**Note** When using Pin sleep, the internal pull up/down resistor is not used

**PR** and **PD** only affect lines that are configured as digital inputs or disabled.
The following table defines the bit-field map for **PR** and **PD** commands.

| Bit | I/O line |
|---|---|
| 0 | DIO4 (pin 11) |
| 1 | AD3/DIO3 (pin 17) |
| 2 | AD2/DIO2 (pin 18) |
| 3 | AD1/DIO1 (pin 19) |
| 4 | AD0/DIO0 (pin 20) |
| 5 | $\overline{\text{RTS}}$/DIO6 (pin 16) |
| 6 | DI8/SLEEP_RQ (pin 9) |
| 7 | DIN/CONFIG (pin 3) |

**Parameter range**

0 - 0xFF (bit field)

**Default**

0xFF

**Example**

Sending the command **ATPR 6F** turn bits 0, 1, 2, 3, 5 and 6 ON, and bits 4 and 7 OFF. The binary equivalent of **0x6F** is **01101111**. Bit 0 is the last digit in the bit field.

## PD (Pull Up/Down Direction)

The resistor pull direction bit field (1 = pull-up, 0 = pull-down) for corresponding I/O lines that are set by the **PR** command.

See PR (Pull-up/Down Resistor Enable) for the bit mappings.

**Parameter range**

0x0 - 0xFF

**Default**

0xFF

## IU (I/O Output Enable)

The **IU** command disables or enables I/O UART output. When enabled (**IU** = 1), received I/O line data packets are sent out the UART. The data is sent using an API frame regardless of the current **AP** parameter value.

Enable or disable the serial output of received I/O sample data when I/O line passing is enabled. **IU** only affects the device's behavior when **IA** is set to a non-default value.

When **IU** is enabled, any received I/O sample data is sent out the UART/SPI interface using an API frame. Sample data is only generated if the local device is operating in API mode (**AP** = 1 or 2).

**Parameter range**

0 - 1

| Parameter | Description |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

**Default**

1

## IT (Samples before TX)

Sets or displays the number of samples to collect before transmitting data. The maximum number of samples is dependent on the number of enabled I/O lines and the maximum payload available.

If **IT** is set to a number too big to fit in the maximum payload, it is reduced such that it will fit. A query of **IT** after setting it reports the actual number of samples in a packet.

**Parameter range**

0x1 - 0xFF

**Default**

1

## IS (Force Sample)

Forces a read of all enabled digital and analog input lines. The data is returned through the UART or SPI.

When operating in Transparent mode (**AP** = 0), the data is returned in the following format:

All bytes are converted to ASCII:

number of samples<CR>

channel mask<CR>

DIO data<CR> (If DIO lines are enabled)

ADC channel Data<CR> (This will repeat for every enabled ADC channel)

<CR> (end of data noted by extra <CR>)

When operating in API mode (**AP** = 1), the command immediately returns an **OK** response. The data follows in the normal API format for DIO data.

**Parameter range**

N/A

**Default**

N/A

## IO (Digital Output Level)

Sets digital output levels. This allows DIO lines setup as outputs to be changed through Command mode.

**Parameter range**

8-bit bit map; each bit represents the level of an I/O line set up as an output

**Default**

N/A

## IC (DIO Change Detect)

Set or read the digital I/O pins to monitor for changes in the I/O state.

Each bit enables monitoring of DIO0 - DIO7 for changes. If detected, data is transmitted with DIO data only. Any samples queued waiting for transmission is sent first.

Set unused bits to 0.

**Parameter range**

0 - 0xFFFF (bit field)

**Default**

0

## IR (Sample Rate)

Set or read the I/O sample rate to enable periodic sampling. When set, this parameter causes the device to sample all enabled DIO and ADC at a specified interval.

To enable periodic sampling, set **IR** to a non-zero value, and enable the analog or digital I/O functionality of at least one device pin (see D0 (DIO0/AD0) -D8 (DIO8/SLEEP_REQUEST), P0 (RSSI/PWM0 Configuration) -P2 (SPI_MISO).

**WARNING!** If you set **IR** to 1 or 2, the device will not keep up and many samples will be lost.

Set or read the I/O sample rate to enable periodic sampling.

When set, this parameter causes the device to sample all enabled digital I/O and analog inputs at a specified interval. Samples will be sent to the address specified by the **DH** and **DL** commands. The target device must be operating in API mode in order to output the received sample data.

To enable periodic sampling, set **IR** to a non-zero value, and enable the analog or digital I/O functionality of at least one device pin (**D0** – **D9**, **P0** – **P9**).

**Parameter range**

0 - 0xFFFF (x 1 ms)

**Default**

0

## RP (RSSI PWM Timer)

The PWM timer expiration in 0.1 seconds. **RP** sets the duration of pulse width modulation (PWM) signal output on the RSSI pin. The signal duty cycle updates with each received packet and shuts off when the timer expires.

When **RP** = **0xFF**, the output is always on.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0x28 (four seconds)

# I/O line passing commands

The following AT commands are I/O line passing commands.

I/O Line Passing allows the digital and analog inputs of a remote device to affect the corresponding outputs of the local device.

You can perform Digital Line Passing on any of the Digital I/O lines. Digital Inputs directly map to Digital Outputs of each digital pin.

Analog Line Passing can be performed only on the first two ADC lines:

- ADC0 corresponds with PWM0
- ADC1 corresponds with PWM1

## IA (I/O Input Address)

The source address of the device to which outputs are bound. Setting all bytes to 0xFF disables I/O line passing. Setting **IA** to 0xFFFF allows any I/O packet addressed to this device (including broadcasts) to change the outputs.

The source address of the device to which outputs are bound. If an I/O sample is received from the address specified, any pin that is configured as a digital output or PWM changes its state to match that of the I/O sample.

Set **IA** to 0xFFFFFFFFFFFFFFFF to disable I/O line passing.

Set **IA** to 0xFFFF to allow any I/O packet addressed to this device (including broadcasts) to change the outputs.

**Parameter range**

0 - 0xFFFF FFFF FFFF FFFF

**Default**

0xFFFFFFFFFFFFFFFF (I/O line passing disabled)

## T0 (D0 Timeout)

Specifies how long pin D0 holds a given value before it reverts to configured value. If set to 0, there is no timeout.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

## T1 (D1 Output Timeout)

Specifies how long pin D1 holds a given value before it reverts to configured value. If set to 0, there is no timeout.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

## T2 (D2 Output Timeout)

Specifies how long pin D2 holds a given value before it reverts to configured value. If set to 0, there is no timeout.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

## T3 (D3 Output Timeout)

Specifies how long pin D3 holds a given value before it reverts to configured value. If set to 0, there is no timeout.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

## T4 (D4 Output Timeout)

Specifies how long pin D4 holds a given value before it reverts to configured value. If set to 0, there is no timeout.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

## T5 (D5 Output Timeout)

Specifies how long pin D5 holds a given value before it reverts to configured value. If set to 0, there is no timeout.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

## T6 (D6 Output Timeout)

Specifies how long pin D6 holds a given value before it reverts to configured value. If set to 0, there is no timeout.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

## T7 (D7 Output Timeout)

Specifies how long pin D7 holds a given value before it reverts to configured value. If set to 0, there is no timeout.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

### PT (PWM Output Timeout)

Specifies how long both PWM outputs (**P0**, **P1**) output a given PWM signal before it reverts to the configured value (**M0**/**M1**). If set to 0, there is no timeout. This timeout only affects these pins when they are configured as PWM output.

**Parameter range**

0 - 0xFF (x 100 ms)

**Default**

0xFF

## Diagnostic commands

The following AT commands are diagnostic commands. Diagnostic commands are typically volatile and will not persist across a power cycle.

### VR (Firmware Version)

Reads the firmware version on a device.

Firmware version numbers have four significant digits. The reported number shows three or four numbers in hexadecimal notation. A version is reported as **ABCD**. Digits ABC are the main release number and D is the revision number from the main release. **D** is not required and if it is not present, a zero is assumed for **D**. **B** is a variant designator.

**Parameter range**

0x2000 - 0x20FF

**Default**

Set in the factory

### VL (Version Long)

Shows detailed version information including the application build date and time.

**Parameter range**

N/A

**Default**

N/A

### HV (Hardware Version)

Display the hardware version number of the device.

**Parameter range**

0 - 0xFFFF [read-only]

**Default**

Set in firmware

## DB (Last Packet RSSI)

Reports the RSSI in -dBm of the last received RF data packet. **DB** returns a hexadecimal value for the -dBm measurement.

For example, if **DB** returns 0x60, then the RSSI of the last packet received was -96 dBm.

If the XBee/XBee-PRO S2C 802.15.4 RF Module has been reset and has not yet received a packet, **DB** reports **0**.

This value is volatile (the value does not persist in the device's memory after a power-up sequence).

**Parameter range**

0 - 0xFF [read-only]

**Default**

N/A

## EC (CCA Failures)

ReSets or displays the count of Clear Channel Assessment (CCA) failures. This register increments when the device does not transmit a packet because it detected energy above the CCA threshold level set with **CA** command. This count saturates at its maximum value. Set the count to zero to reset the count.

**Parameter range**

0 - 0xFFFF

**Default**

N/A

## EA (ACK Failures)

Resets or displays the count of acknowledgment failures. This register increments when the device expires the retries without receiving an ACK on a packet transmission. This count saturates at its maximum value. Set the count to zero to reset the count.

**Parameter range**

0 - 0xFFFF

**Default**

N/A

## DD (Device Type Identifier)

Stores the Digi device type identifier value. Use this value to differentiate between multiple XBee devices.

If you change **DD**, RE (Restore Defaults) will not restore defaults. The only way to get **DD** back to default values is to explicitly set it to defaults.

**Parameter range**

0 - 0xFFFFFFFF

**Default**

0x10000

# Command mode options

The following commands are Command mode option commands.

## CT (Command mode Timeout)

Sets or displays the Command mode timeout parameter. If a device does not receive any valid commands within this time period, it returns to Idle mode from Command mode.

**Parameter range**

2 - 0x1770 (x 100 ms)

**Default**

0x64 (10 seconds)

## CN (Exit Command mode)

Exits Command mode and returns the product family to Idle mode.

**Parameter range**

N/A

**Default**

N/A

## AC (Apply Changes)

Immediately applies new settings without exiting Command mode.

Applying changes means that the device is re-initialized based on changes made to its parameter values. Once changes are applied, the device immediately operates according to the new parameter values.

This behavior is in contrast to issuing the **WR** (Write) command. The **WR** command saves parameter values to non-volatile memory, but the device still operates according to previously saved values until the device is rebooted or the **CN** (Exit AT Command Mode) or **AC** commands are issued. For more information, see AT Command - Queue Parameter Value frame - 0x09.

**Parameter range**

N/A

**Default**

N/A

## GT (Guard Times)

Set the required period of silence before and after the command sequence characters of the Command mode sequence (**GT** + **CC** + **GT**). The period of silence prevents inadvertently entering Command mode.

**Parameter range**

0x2 - 0xCE4 (x 1 ms)

**Default**

0x3E8 (one second)

## CC (Command Character)

The character value the device uses to enter Command mode.

The default value (**0x2B**) is the ASCII code for the plus (**+**) character. You must enter it three times within the guard time to enter Command mode. To enter Command mode, there is also a required period of silence before and after the command sequence characters of the Command mode sequence (**GT** + **CC** + **GT**). The period of silence prevents inadvertently entering Command mode.

**Parameter range**

0 - 0xFF

**Default**

0x2B (the ASCII plus character: **+**)

# Operate in API mode

API mode overview ..... 107
API frame specifications ..... 107
Escaped characters in API frames ..... 110
Frame descriptions ..... 111

# API mode overview

As an alternative to Transparent operating mode, you can use API operating mode. API mode provides a structured interface where data is communicated through the serial interface in organized packets and in a determined order. This enables you to establish complex communication between devices without having to define your own protocol. The API specifies how commands, command responses and device status messages are sent and received from the device using the serial interface or the SPI interface.

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

# API frame specifications

The firmware supports two API operating modes: without escaped characters and with escaped characters. Use the **AP** command to enable either mode. To configure a device to one of these modes, set the following **AP** parameter values:

| AP command setting | Description |
|---|---|
| **AP** = 0 | Transparent operating mode, UART serial line replacement with API modes disabled. This is the default option. |
| **AP** = 1 | API operation. |
| **AP** = 2 | API operation with escaped characters (only possible on UART). |

The API data frame structure differs depending on what mode you choose.

The firmware silently discards any data it receives prior to the start delimiter. If the device does not receive the frame correctly or if the checksum fails, the device discards the frame.

## API operation (AP parameter = 1)

We recommend this API mode for most applications. The following table shows the data frame structure when you enable this mode:

| Frame fields | Byte | Description |
|---|---|---|
| Start delimiter | 1 | 0x7E |
| Length | 2 - 3 | Most Significant Byte, Least Significant Byte |
| Frame data | 4 - n | API-specific structure |
| Checksum | n + 1 | 1 byte |

## API operation-with escaped characters (AP parameter = 2)

Set API to 2 to allow escaped control characters in the API frame. Due to its increased complexity, we only recommend this API mode in specific circumstances. API 2 may help improve reliability if the serial interface to the device is unstable or malformed frames are frequently being generated.

When operating in API 2, if an unescaped 0x7E byte is observed, it is treated as the start of a new API frame and all data received prior to this delimiter is silently discarded. For more information on using this API mode, refer to the following knowledge base article:

http://knowledge.digi.com/articles/Knowledge_Base_Article/Escaped-Characters-and-API-Mode-2

The following table shows the structure of an API frame with escaped characters:

| Frame fields | Byte | Description | |
|---|---|---|---|
| Start delimiter | 1 | 0x7E | |
| Length | 2 - 3 | Most Significant Byte, Least Significant Byte | Characters escaped if needed |
| Frame data | 4 - n | API-specific structure | |
| Checksum | n + 1 | 1 byte | |

### *Escape characters*

When sending or receiving a UART data frame, you must escape (flag) specific data values 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. If not escaped, 0x11 and 0x13 are sent as is.

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 previous 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.

## Start delimiter

This field indicates the beginning of a frame. It is always 0x7E. This allows the device to easily detect a new incoming frame.

## Length

The length field specifies the total number of bytes included in the frame's data field. Its two-byte value excludes the start delimiter, the length, and the checksum.

## Frame data

This field contains the information that a device receives or will transmit. The structure of frame data depends on the purpose of the API frame:

| Start delimiter | Length | | Frame data | | | | | | | | Checksum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Frame type | Data | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | n | n+1 |
| 0x7E | MSB | LSB | API frame type | Data | | | | | | | Single byte |

- **Frame type** is the API frame type identifier. It determines the type of API frame and indicates how the Data field organizes the information.
- **Data** contains the data itself. This information and its order depend on the what type of frame that the Frame type field defines.

## Checksum

Checksum is the last byte of the frame and helps test data integrity. It is calculated by taking the hash sum of all the API frame bytes that came before it, except the first three bytes (start delimiter and length).

The device does not process frames sent through the serial interface with incorrect checksums, and ignores their data.

## Calculate and verify checksums

To calculate the checksum of an API frame:

1. Add all bytes of the packet, except the start delimiter 0x7E and the length (the second and third bytes).
2. Keep only the lowest 8 bits from the result.
3. Subtract this quantity from 0xFF.

To verify the checksum of an API frame:

1. Add all bytes including the checksum; do not include the delimiter and length.
2. If the checksum is correct, the last two digits on the far right of the sum equal 0xFF.

### *Example*

Consider the following sample data packet: **7E 00 0A 01 01 50 01 00 48 65 6C 6C 6F B8**

| Byte(s) | Description |
|---|---|
| 7E | Start delimiter |
| 00 0A | Length bytes |
| 01 | API identifier |
| 01 | API frame ID |
| 50 01 | Destination address low |
| 00 | Option byte |

| Byte(s) | Description |
|---|---|
| 48 65 6C 6C 6F | Data packet |
| B8 | Checksum |

To calculate the check sum you add all bytes of the packet, excluding the frame delimiter **7E** and the length (the second and third bytes):

**7E 00 0A 01 01 50 01 00 48 65 6C 6C 6F B8**

Add these hex bytes:

01 + 01 + 50 + 01 + 00 + 48 + 65 + 6C + 6C + 6F = 247

Now take the result of 0x247 and keep only the lowest 8 bits which, in this example, is 0x47 (the two far right digits). Subtract 0x47 from 0xFF and you get 0xB8 (0xFF - 0x47 = 0xB8). 0xB8 is the checksum for this data packet.

If an API data packet is composed with an incorrect checksum, the XBee/XBee-PRO S2C 802.15.4 RF Module will consider the packet invalid and will ignore the data.

To verify the check sum of an API packet add all bytes including the checksum (do not include the delimiter and length) and if correct, the last two far right digits of the sum will equal FF.

01 + 01 + 50 + 01 + 00 + 48 + 65 + 6C + 6C + 6F + B8 = 2FF

# Escaped characters in API frames

If operating in API mode with escaped characters (**AP** parameter = 2), when sending or receiving a serial 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'ed with 0x20).

The following data bytes need to be escaped:

- 0x7E: start delimiter
- 0x7D: escape character
- 0x11: XON
- 0x13: XOFF

To escape a character:

1. Insert 0x7D (escape character).
2. Append it with the byte you want to escape, XOR'ed with 0x20.

In API mode with escaped characters, the length field does not include any escape characters in the frame and the firmware calculates the checksum with non-escaped data.

# Frame descriptions

The following sections describe the API frames.

## TX Request: 64-bit address frame - 0x00

# Description

This frame causes the device to send payload data as an RF packet.

# Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x00 |
| Frame ID | 4 | Identifies the data frame for the host to correlate with a subsequent ACK, which is a 0x89 (Tx status) frame that indicates the packet was transmitted successfully. If set to **0**, the device does not send a response. |
| 64-bit destination address | 5-12 | MSB first, LSB last. Set to the 64-bit address of the destination device.<br>If set to 0x000000000000FFFF, the broadcast address is used. |
| Options | 13 | 0x01 = Disable ACK<br>0x04 = Send packet with Broadcast PAN ID.<br>Set all other bits to 0. |
| RF data | 14-n | The RF data length can be up to 110 bytes, but may be less depending on other factors discussed in Maximum payload. |

### TX Request: 16-bit address - 0x01

## Description

A TX Request message causes the device to transmit data as an RF Packet.

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x01 |
| Frame ID | 4 | Identifies the data frame for the host to correlate with a subsequent ACK (0x89). If set to **0**, the device does not send a response. |
| 16-bit destination address | 5-6 | MSB first, LSB last. Set to the 16-bit address of the destination device. Broadcast = 0xFFFF. |
| Options | 7 | 0x01 = Disable ACK.<br>0x04 = Send packet with Broadcast PAN ID.<br>Set all other bits to 0. |
| RF data | 8-n | The RF data length can be up to 116 bytes, but may be less depending on other factors discussed in Maximum payload. |

### AT Command frame - 0x08

## Description

Use this frame to query or set device parameters on the local device. This API command applies changes after running the command. You can query parameter values by sending the 0x08 AT Command frame with no parameter value field (the two-byte AT command is immediately followed by the frame checksum).

A 0x88 response frame is populated with the parameter value that is currently set on the device. The Frame ID of the 0x88 response is the same one set by the command in the 0x08 frame.

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x08 |
| Frame ID | 4 | Identifies the data frame for the host to correlate with a subsequent ACK (0x88). If set to **0**, the device does not send a response. |
| AT command | 5-6 | Command name: two ASCII characters that identify the AT command. |
| Parameter value | 7-n | If present, indicates the requested parameter value to set the given register. If no characters are present, it queries the register. |

## Example

The following example illustrates an AT Command frame where the baud rate (**BD**) for the device is set to 1200.

| Frame data fields | Offset | Example |
|---|---|---|
| Start delimiter | 0 | 0x7E |
| Length | MSB 1 | 0x00 |
| | LSB 2 | 0x04 |
| Frame type | 3 | 0x08 |
| Frame ID | 4 | 0x52 (R) |

| Frame data fields | Offset | Example |
|---|---|---|
| AT command | 5 | 0x4E (N) |
| | 6 | 0x48 (H) |
| Parameter value (optional) | | |
| Checksum | 7 | 0x0F |

The following example illustrates an AT Command frame when you modify the device's **DL** parameter value.

| Frame data fields | Offset | Example |
|---|---|---|
| Start delimiter | 0 | 0x7E |
| Length | MSB 1 | 0x00 |
| | LSB 2 | 0x08 |
| Frame type | 3 | 0x08 |
| Frame ID | 4 | 0x4D (M)[1] |
| AT command | 5 | 0x44 (D) |
| | 6 | 0x4C (L) |
| Parameter value | 7-10 | 0x00000FFF |
| Checksum | 11 | 0x0C |

[1]M was arbitrarily selected.

### AT Command - Queue Parameter Value frame - 0x09

## Description

This frame allows you to query or set device parameters. In contrast to the AT Command (0x08) frame, this frame queues new parameter values and does not apply them until you issue either:

- The **AT** Command (0x08) frame (for API type)
- The **AC** command

When querying parameter values, the 0x09 frame behaves identically to the 0x08 frame. The device returns register queries immediately and not does not queue them. The response for this command is also an **AT** Command Response frame (0x88).

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x09 |
| Frame ID | 4 | Identifies the data frame for the host to correlate with a subsequent ACK. If set to **0**, the device does not send a response. |
| AT command | 5-6 | Command name: two ASCII characters that identify the AT command. |
| Parameter value (optional) | 7-n | If present, indicates the requested parameter HEX (non-ASCII) value to set the given register. If no characters are present, queries the register. |

### Remote AT Command Request frame - 0x17

## Description

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

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x17 |
| Frame ID | 4 | Identifies the data frame for the host to correlate with a subsequent ACK (0x97). If set to **0**, the device does not send a response. |
| 64-bit destination address | 5-12 | MSB first, LSB last. Set to the 64-bit address of the destination device.<br>If the 16-bit parameter has a different value from 0xFFFE the device uses the 64-bit address is used. Otherwise, it uses the 16-bit address. |
| 16-bit destination address | 13-14 | MSB first, LSB last. Set to match the 16-bit network address of the destination device, if known.<br>Set to 0xFFFE if the address is unknown, or if sending a broadcast. |
| AT command | 16-17 | Command name: two ASCII characters that identify the command. |
| Command parameter | 18-n | If present, indicates the parameter value you request for a given register.<br>If no characters are present, it queries the register. |

### RX Packet: 64-bit Address frame - 0x80

## Description

When a device receives an RF data packet from a device configured to use 64-bit addressing (**MY** = FFFF or FFFE), it sends this frame out the serial interface.

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x80 |
| 64-bit source address | 4-11 | The sender's 64-bit address. Set to 0xFFFFFFFFFFFFFFFF (unknown 64-bit address) if the sender's 64-bit address is unknown. MSB first, LSB last. |
| RSSI | 12 | Received Signal Strength Indicator. The Hexadecimal equivalent of (-dBm) value. For example if RX signal strength is -40 dBm, then 0x28 (40 decimal) is returned. |
| Options | 13 | Bit field:<br>0 = [reserved]<br>1 = Address broadcast<br>2 = PAN broadcast<br>3-7 = [reserved] |
| Received data | 14-n | The RF data length can be up to 110 bytes, but may be less depending on other factors discussed in Maximum payload. |

### Receive Packet: 16-bit address frame - 0x81

## Description

When the device receives an RF packet from a device configured to use 16 bit addressing (**MY** < FFFE), it sends this frame out the serial interface.

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x81 |
| Source address | 4-5 | MSB first<br>LSB last |
| RSSI | 6 | RSSI = hexadecimal equivalent of -dBm value. For example, if RX signal strength = -40 dBm, it returns 0x28 (40 decimal). |
| Options | 7 | Bit 0 = [reserved]<br>Bit 1 = Address broadcast<br>Bit 2 = PAN broadcast<br>Bits 3 - 7 = [reserved] |
| RF data | 8-n | The RF data length can be up to 116 bytes, but may be less depending on other factors discussed in Maximum payload. |

### RX (Receive) Packet: 64-bit address IO frame- 0x82

## Description

When the device receives an I/O sample from a remote device configured to use 64-bit addressing, the I/O data is sent out the UART using this frame type

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Total number of samples | Description |
|---|---|---|---|
| Frame type | 3 | N/A | 0x82 |
| 64-bit source address | 4-11 | N/A | MSB first, LSB last. |
| RSSI | 12 | N/A | RSSI: Hexadecimal equivalent of (-dBm) value. For example, if RX signal strength = -40 dBm, the device returns 0x28 (40 decimal). |
| Status | 13 | N/A | bit 0 = reserved<br>bit 1 = Address broadcast<br>bit 2 = PAN broadcast<br>bits 3-7 = [reserved] |
| Number of samples | 14 | N/A | Total number of samples. |
| Channel Indicator (see bit field table below) | 15 | MSB | Indicates which inputs have sampling enabled (if any).<br>Each bit represents either a DIO line or ADC channel. Bit set to 1 if channel is active |
| | 16 | LSB | |
| Digital samples (if enabled) (see bit field table below) | 17 | MSB | If any of the DIO lines are enabled in the Channel indicator, these two bytes contain samples for all enabled DIO lines. DIO lines that do not have sampling enabled return 0. If no DIO line is enabled, no bytes are included in the frame. |
| | 18 | LSB | |

| Frame data fields | Offset | Total number of samples | Description |
|---|---|---|---|
| | 19 | ADC0 MSB | If the sample set includes any ADC data, each enabled analog input returns a two-byte value indicating the A/D measurement of that input.<br>ADC channel data is represented as an unsigned 10-bit value right-justified on a 16-bit boundary.<br>Analog samples are ordered sequentially from AD0 to AD5. |
| | 20 | ADC0 LSB | |
| | ... | N/A | |
| | n -1 | ADCn MSB | |
| | n | ADCn LSB | |

| Bit field | Description |
|---|---|
| Reserved | 3 bits |
| A3 - A0 | 4 analog bits |
| D8 - D0 | 9 digital bits |

### RX Packet: 16-bit address I/O frame - 0x83

## Description

When the device receives an I/O sample from a remote device configured to use 16-bit addressing, the I/O data is sent out the UART using this frame type.

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Total number of samples | Description |
|---|---|---|---|
| Frame type | 3 | N/A | 0x83 |
| Source Address | 4-5 | N/A | MSB first, LSB last. |
| RSSI | 6 | N/A | RSSI: Hexadecimal equivalent of (-dBm) value. For example, if RX signal strength = -40 dBm, the device returns 0x28 (40 decimal). |
| Options | 7 | N/A | bit 0 = reserved<br>bit 1 = Address broadcast<br>bit 2 = PAN broadcast<br>bits 3-7 = [reserved] |
| Number of samples | 8 | N/A | Total number of samples. |
| Channel Indicator (see bit field table below) | 9 | MSB | Indicates which inputs have sampling enabled (if any). Each bit represents either a DIO line or ADC channel.<br>Bit set to 1 if channel is active. |
| | 10 | LSB | |
| Digital Samples (if enabled) (see bit field table below) | 11 | MSB | If any of the DIO lines are enabled in the Channel indicator, these two bytes contain samples for all enabled DIO lines. DIO lines that do not have sampling enabled return 0. If no DIO line is enabled, no bytes are included in the frame. |
| | 12 | LSB | |

| Frame data fields | Offset | Total number of samples | Description |
|---|---|---|---|
| Analog samples | 13 | ADC0 MSB | If the sample set includes any ADC data, each enabled analog input returns a two-byte value indicating the A/D measurement of that input. ADC channel data is represented as an unsigned 10-bit value right-justified on a 16-bit boundary. Analog samples are ordered sequentially from AD0 to AD5. |
| | 14 | ADC0 LSB | |
| | ... | | |
| | n - 1 | ADCn MSB | |
| | n | ADCn LSB | |

The following table shows the Channel Indicator bit field.

| Bit field | Description |
|---|---|
| Reserved | 3 bits |
| A3 - A0 | 4 analog bits |
| D8 - D0 | 9 digital bits |

### AT Command Response frame - 0x88

## Description

A device sends this frame in response to an AT Command (0x08) frame and a queued AT command 0x09. Some commands send back multiple frames; for example, the **ND** command. This command ends by sending a frame with a status of **0** (OK) and no value. In the particular case of **ND**, a frame is received via a remote node in the network and when the process is finished, the AT command response is received. For details on the behavior of **ND**, see ND (Network Discovery).

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x88 |
| Frame ID | 4 | Identifies the data frame for the host to correlate with a subsequent ACK (0x08 or 0x09). If set to **0**, the device does not send a response. |
| AT command | 5-6 | Command name: two ASCII characters that identify the command. |
| Command status | 7 | 0 = OK<br>1 = ERROR<br>2 = Invalid command<br>3 = Invalid parameter |
| Command data | 8-n | The register data in hexadecimal format. If the host sets the register, the device does not return this field. |

## Example

If you change the **BD** parameter on a local device with a frame ID of 0x01, and the parameter is valid, the user receives the following response.

| Frame data fields | Offset | Example |
|---|---|---|
| Start delimiter | 0 | 0x7E |
| Length | MSB 1 | 0x00 |
| | LSB 2 | 0x05 |
| Frame type | 3 | 0x88 |

| Frame data fields | Offset | Example |
|---|---|---|
| Frame ID | 4 | 0x01 |
| AT command | 5 | 0x42 (B) |
| | 6 | 0x44 (D) |
| Command status | 7 | 0x00 |
| Command data | | (No command data implies the parameter was set rather than queried) |
| Checksum | 8 | 0xF0 |

### TX Status frame - 0x89

## Description

When a TX request: 64-bit address (0x00) or 16-bit address (0x01) is complete, the device sends a TX Status frame. This message indicates if the packet transmitted successfully or if there was a failure.

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x89 |
| Frame ID | 4 | Identifies the TX Request frame being reported.<br>If the Frame ID = 0 in the TX Request, no TX Status frame is given. |
| Status | 5 | 0x00 = standard<br>0x01 = no ACK received<br>0x02 = CCA failure<br>0x03 = Transmission was purged because a coordinator tried to send to an end device, but it timed out waiting for a poll from the end device that never occurred<br>0x21 = Network ACK failure<br>0x22 = Transmission failed because an end device was not joined to the network<br>0x31 = Internal error<br>0x32 = Transmission failed due to resource depletion (for example, out of buffers, especially for indirect messages from coordinator)<br>0x74 = The payload in the frame was larger than allowed |

**Notes:**

- STATUS = 1 occurs when all retries are expired and no ACK is received.
- If transmitter broadcasts (destination address = 0x000000000000FFFF), status 0x01 and 0x21 will never be returned.
- STATUS = 3 occurs when Coordinator times out of an indirect transmission. Timeout is defined as (2.5 x **SP** (Cyclic Sleep Period) parameter value).

## Example

The following example shows a successful status received.

| Frame data fields | Offset | Example |
|---|---|---|
| Start delimiter | 0 | 0x7E |
| Length | MSB 1 | 0x00 |
| | LSB 2 | 0x03 |
| Frame type | 3 | 0x89 |
| Frame ID | 4 | 0x01 |
| Status | 5 | 0x00 |
| Checksum | 6 | 0x75 |

### Modem Status frame - 0x8A

## Description

Devices send the status messages in this frame in response to specific conditions.

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x8A |
| Status | 4 | 0x00 Hardware reset<br>0x01 Watchdog timer reset<br>0x02 End device successfully associated with a coordinator<br>0x03 End device disassociated from coordinator or coordinator failed to form a new network<br>0x06 Coordinator formed a new network<br>0x0D Input voltage on the XBee-PRO device is too high, which prevents transmissions |

## Example

When a device powers up, it returns the following API frame.

| Frame data fields | Offset | Example |
|---|---|---|
| Start delimiter | 0 | 0x7E |
| Length | MSB 1 | 0x00 |
| LSB 2 | LSB 2 | 0x02 |
| Frame type | 3 | 0x8A |
| Status | 4 | 0x00 |
| Checksum | 5 | 0x75 |

### Remote Command Response frame - 0x97

## Description

If a device receives this frame in response to a Remote Command Request (0x17) frame, the device sends an AT Command Response (0x97) frame out the serial interface.

Some commands, such as the **ND** command, may send back multiple frames. For details on the behavior of **ND**, see ND (Network Discovery).

## Format

The following table provides the contents of the frame. For details on frame structure, see API frame specifications.

| Frame data fields | Offset | Description |
|---|---|---|
| Frame type | 3 | 0x97 |
| Frame ID | 4 | This is the same value that is passed in to the request.<br>The request is a 0x17 frame. |
| 64-bit source (remote) address | 5-12 | The address of the remote device returning this response. |
| 16-bit source (remote) address | 13-14 | Set to the 16-bit network address of the remote device returning this response. |
| AT commands | 15-16 | The name of the command. |
| Command status | 17 | 0 = OK<br>1 = ERROR<br>2 = Invalid Command<br>3 = Invalid Parameter<br>4 = No response |
| Command data | 18-n | The value of the requested register in hexadecimal notation (non-ASCII). |

# Regulatory information

United States (FCC) ..... 130
Europe (CE) ..... 145
ISED (Innovation, Science and Economic Development Canada) ..... 147
Australia (RCM) ..... 148
South Korea ..... 148

# United States (FCC)

XBee/XBee-PRO S2C 802.15.4 RF Modules comply with Part 15 of the FCC rules and regulations. Compliance with the labeling requirements, FCC notices and antenna usage guidelines is required.

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

1. The system integrator must ensure that the text on the external label provided with this device is placed on the outside of the final product.
2. RF Modules may only be used with antennas that have been tested and approved for use with the modules.

## OEM labeling requirements

**WARNING!** As an Original Equipment Manufacturer (OEM) you must ensure that FCC labeling requirements are met. You must include a clearly visible label on the outside of the final product enclosure that displays the following content:

### *Required FCC Label for OEM products containing the XBee-PRO S2C SMT RF Module*

Contains FCC ID: MCQ-PS2CSM

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.

### *Required FCC Label for OEM products containing the XBee S2C TH RF Module*

Contains FCC ID: MCQ-S2CTH

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.

### *Required FCC Label for OEM products containing the XBee-PRO S2C TH RF Module*

Contains FCC ID: MCQ-PS2CTH

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.

## FCC notices

**IMPORTANT**: XBee/XBee-PRO S2C 802.15.4 RF Modules have 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 and XBee-PRO RF Modules can be installed using 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 for the channels indicated in the tables below. If the antenna is mounted at least 25 cm (10 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).

The antennas in the tables below have been approved for use with this module. Cable loss is required when using gain antennas as shown in the tables. Digi does not carry all of these antenna variants. Contact Digi Sales for available antennas.

All antenna part numbers followed by an asterisk (*) are not available from Digi. Consult with an antenna manufacturer for an equivalent option.

### *XBee S2C SMT RF module*

The following table shows the antennas approved for use with the XBee S2C SMT RF module.

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | | |
|---|---|---|---|---|---|---|---|
| | | | | | Channels 11-24 | Channel 25 | Channel 26 |
| **Integral antennas** | | | | | | | |
| 29000313 | Integral PCB antenna | 0.0 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-QI | Monopole (Integrated whip) | 1.5 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| **Dipole antennas** | | | | | | | |
| A24-HASM-450 | Dipole (Half-wave articulated RPSMA - 4.5”) | 2.1 | Fixed | 25 cm | N/A | N/A | N/A |
| A24-HABSM* | Dipole (Articulated RPSMA) | 2.1 | Fixed | 25 cm | N/A | N/A | N/A |
| 29000095 | Dipole (Half-wave articulated RPSMA - 4.5”) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-HABUF-P5I | Dipole (Half-wave articulated bulkhead mount U.FL. w/ 5” pigtail) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | | |
|---|---|---|---|---|---|---|---|
| | | | | | Channels 11-24 | Channel 25 | Channel 26 |
| A24-HASM-525 | Dipole (Half-wave articulated RPSMA - 5.25") | 2.1 | Fixed | 25 cm | N/A | N/A | N/A |
| **Omni-directional antennas** | | | | | | | |
| A24-F2NF | Omni-directional (Fiberglass base station) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-F3NF | Omni-directional (Fiberglass base station) | 3.0 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-F5NF | Omni-directional (Fiberglass base station) | 5.0 | Fixed | 25 cm | N/A | N/A | N/A |
| A24-F8NF | Omni-directional (Fiberglass base station) | 8.0 | Fixed | 2 m | N/A | N/A | 0.1 |
| A24-F9NF | Omni-directional (Fiberglass base station) | 9.5 | Fixed | 2 m | N/A | N/A | 1.6 |
| A24-F10NF | Omni-directional (Fiberglass base station) | 10.0 | Fixed | 2 m | N/A | N/A | 2.1 |
| A24-F12NF | Omni-directional (Fiberglass base station) | 12.0 | Fixed | 2 m | N/A | N/A | 4.1 |
| A24-W7NF | Omni-directional (Fiberglass base station) | 7.2 | Fixed | 2 m | N/A | N/A | N/A |
| A24-M7NF | Omni-directional (Mag-mount base station) | 7.2 | Fixed | 2 m | N/A | N/A | N/A |
| A24-F15NF | Omni-directional (Fiberglass base station) | 15.0 | Fixed | 2 m | 1.1 | 1.1 | 7.1 |
| **Panel antennas** | | | | | | | |
| A24-P8SF | Flat Panel | 8.5 | Fixed | 2 m | N/A | N/A | 6.1 |
| A24-P8NF | Flat Panel | 8.5 | Fixed | 2 m | N/A | N/A | 6.1 |
| A24-P13NF | Flat Panel | 13.0 | Fixed | 2 m | N/A | 3.1 | 10.6 |
| A24-P14NF | Flat Panel | 14.0 | Fixed | 2 m | N/A | 4.1 | 11.6 |
| A24-P15NF | Flat Panel | 15.0 | Fixed | 2 m | N/A | 5.1 | 12.6 |
| A24-P16NF | Flat Panel | 16.0 | Fixed | 2 m | N/A | 6.1 | 13.6 |

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | | |
|---|---|---|---|---|---|---|---|
| | | | | | Channels 11-24 | Channel 25 | Channel 26 |
| A24-P19NF | Flat Panel | 19.0 | Fixed | 2 m | 1.1 | 9.1 | 16.6 |
| **Yagi antennas** | | | | | | | |
| A24-Y6NF | Yagi (6-element) | 8.8 | Fixed | 2 m | N/A | N/A | 3.9 |
| A24-Y7NF | Yagi (7-element) | 9.0 | Fixed | 2 m | N/A | N/A | 4.1 |
| A24-Y9NF | Yagi (9-element) | 10.0 | Fixed | 2 m | N/A | N/A | 5.1 |
| A24-Y10NF | Yagi (10-element) | 11.0 | Fixed | 2 m | N/A | 0.6 | 6.1 |
| A24-Y12NF | Yagi (12-element) | 12.0 | Fixed | 2 m | N/A | 1.6 | 7.1 |
| A24-Y13NF | Yagi (13-element) | 12.0 | Fixed | 2 m | N/A | 1.6 | 7.1 |
| A24-Y15NF | Yagi (15-element) | 12.5 | Fixed | 2 m | N/A | 2.1 | 7.6 |
| A24-Y16NF | Yagi (16-element) | 13.5 | Fixed | 2 m | N/A | 3.1 | 8.6 |
| A24-Y16RM | Yagi (16-element, RPSMA connector) | 13.5 dBi | Fixed | 2 m | N/A | 3.1 | 8.6 |
| A24-Y18NF | Yagi (18-element) | 15.0 | Fixed | 2 m | 1.1 | 4.6 | 10.1 |

### *XBee S2C TH RF Module*

The following table shows the antennas approved for use with the XBee S2C TH RF Module.

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | | |
|---|---|---|---|---|---|---|---|
| | | | | | Channels 11-24 | Channel 25 | Channel 26 |
| **Integral antennas** | | | | | | | |
| 29000294 | Integral PCB antenna | -0.5 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-QI | Monopole (Integrated whip) | 1.5 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| **Dipole antennas** | | | | | | | |
| A24-HASM-450 | Dipole (Half-wave articulated RPSMA - 4.5”) | 2.1 | Fixed | 25 cm | N/A | N/A | N/A |
| A24-HABSM | Dipole (Articulated RPSMA) | 2.1 | Fixed | 25 cm | N/A | N/A | N/A |
| 29000095 | Dipole (Half-wave articulated RPSMA - 4.5”) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-HABUF-P5I | Dipole (Half-wave articulated bulkhead mount U.FL. w/ 5” pigtail) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-HASM-525 | Dipole (Half-wave articulated RPSMA - 5.25") | 2.1 | Fixed | 25 cm | N/A | N/A | N/A |
| **Omni-directional antennas** | | | | | | | |
| A24-F2NF | Omni-directional (Fiberglass base station) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-F3NF | Omni-directional (Fiberglass base station) | 3.0 | Fixed/Mobile | 25 cm | N/A | N/A | N/A |
| A24-F5NF | Omni-directional (Fiberglass base station) | 5.0 | Fixed | 25 cm | N/A | N/A | N/A |
| A24-F8NF | Omni-directional (Fiberglass base station) | 8.0 | Fixed | 2 m | N/A | N/A | N/A |

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | | |
|---|---|---|---|---|---|---|---|
| | | | | | Channels 11-24 | Channel 25 | Channel 26 |
| A24-F9NF | Omni-directional (Fiberglass base station) | 9.5 | Fixed | 2 m | N/A | N/A | 0.9 |
| A24-F10NF | Omni-directional (Fiberglass base station) | 10.0 | Fixed | 2 m | N/A | N/A | 1.4 |
| A24-F12NF | Omni-directional (Fiberglass base station) | 12.0 | Fixed | 2 m | N/A | N/A | 3.4 |
| A24-W7NF | Omni-directional (Fiberglass base station) | 7.2 | Fixed | 2 m | N/A | N/A | N/A |
| A24-M7NF | Omni-directional (Mag-mount base station) | 7.2 | Fixed | 2 m | N/A | N/A | N/A |
| A24-F15NF | Omni-directional (Fiberglass base station) | 15.0 | Fixed | 2 m | 0.4 | 0.4 | 6.4 |
| **Panel antennas** | | | | | | | |
| A24-P8SF | Flat Panel | 8.5 | Fixed | 2 m | N/A | N/A | 4.9 |
| A24-P8NF | Flat Panel | 8.5 | Fixed | 2 m | N/A | N/A | 4.9 |
| A24-P13NF | Flat Panel | 13.0 | Fixed | 2 m | N/A | 3.4 | 9.4 |
| A24-P14NF | Flat Panel | 14.0 | Fixed | 2 m | N/A | 4.4 | 10.4 |
| A24-P15NF | Flat Panel | 15.0 | Fixed | 2 m | N/A | 5.4 | 11.4 |
| A24-P16NF | Flat Panel | 16.0 | Fixed | 2 m | N/A | 6.4 | 12.4 |
| A24-P19NF | Flat Panel | 19.0 | Fixed | 2 m | 0.4 | 9.4 | 15.4 |
| **Yagi antennas** | | | | | | | |
| A24-Y6NF | Yagi (6-element) | 8.8 | Fixed | 2 m | N/A | N/A | 4.7 |
| A24-Y7NF | Yagi (7-element) | 9.0 | Fixed | 2 m | N/A | N/A | 4.9 |
| A24-Y9NF | Yagi (9-element) | 10.0 | Fixed | 2 m | N/A | 0.4 | 5.9 |
| A24-Y10NF | Yagi (10-element) | 11.0 | Fixed | 2 m | N/A | 1.4 | 6.9 |

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | | |
|---|---|---|---|---|---|---|---|
| | | | | | Channels 11-24 | Channel 25 | Channel 26 |
| A24-Y12NF | Yagi (12-element) | 12.0 | Fixed | 2 m | N/A | 2.4 | 7.9 |
| A24-Y13NF | Yagi (13-element) | 12.0 | Fixed | 2 m | N/A | 2.4 | 7.9 |
| A24-Y15NF | Yagi (15-element) | 12.5 | Fixed | 2 m | N/A | 2.9 | 8.4 |
| A24-Y16NF | Yagi (16-element) | 13.5 | Fixed | 2 m | N/A | 3.9 | 9.4 |
| A24-Y16RM | Yagi (16-element, RPSMA connector) | 13.5 | Fixed | 2 m | N/A | 3.9 | 9.4 |
| A24-Y18NF | Yagi (18-element) | 15.0 | Fixed | 2 m | 0.4 | 5.4 | 10.9 |

### *XBee-PRO S2C SMT RF Module*

The following table shows the antennas approved for use with the XBee-PRO S2C SMT RF Module.

| Part Number | Type (Description) | Gain (dBi) | Application* | Min Separation | Required antenna cable loss (dB) | |
|---|---|---|---|---|---|---|
| | | | | | Channels 11-23† | Channel 24† |
| **Internal antennas** | | | | | | |
| 29000313 | Integral PCB antenna | 0.0 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-QI | Monopole (Integrated whip) | 1.5 | Fixed/Mobile | 25 cm | N/A | N/A |
| **Dipole antennas** | | | | | | |
| A24-HASM-450 | Dipole (Half-wave articulated RPSMA - 4.5”) | 2.1 | Fixed | 25 cm | N/A | N/A |
| A24-HABSM | Dipole (Articulated RPSMA) | 2.1 | Fixed | 25 cm | N/A | N/A |
| 29000095 | Dipole (Half-wave articulated RPSMA - 4.5”) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-HABUF-P5I | Dipole (Half-wave articulated bulkhead mount U.FL. w/ 5” pigtail) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-HASM-525 | Dipole (Half-wave articulated RPSMA - 5.25") | 2.1 | Fixed | 25 cm | N/A | N/A |
| **Omni-directional antennas** | | | | | | |
| A24-F2NF | Omni-directional (Fiberglass base station) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-F3NF | Omni-directional (Fiberglass base station) | 3.0 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-F5NF | Omni-directional (Fiberglass base station) | 5.0 | Fixed | 25 cm | N/A | N/A |
| A24-F8NF | Omni-directional (Fiberglass base station) | 8.0 | Fixed | 2 m | N/A | N/A |

| Part Number | Type (Description) | Gain (dBi) | Application* | Min Separation | Required antenna cable loss (dB) | |
|---|---|---|---|---|---|---|
| | | | | | Channels 11-23† | Channel 24† |
| A24-F9NF | Omni-directional (Fiberglass base station) | 9.5 | Fixed | 2 m | N/A | N/A |
| A24-F10NF | Omni-directional (Fiberglass base station) | 10 | Fixed | 2 m | N/A | N/A |
| A24-F12NF | Omni-directional (Fiberglass base station) | 12 | Fixed | 2 m | N/A | 1.6 |
| A24-W7NF | Omni-directional (Fiberglass base station) | 7.2 | Fixed | 2 m | N/A | N/A |
| A24-M7NF | Omni-directional (Mag-mount base station) | 7.2 | Fixed | 2 m | N/A | N/A |
| A24-F15NF | Omni-directional (Fiberglass base station) | 15.0 | Fixed | 2 m | 1.1 | 4.6 |
| **Panel antennas** | | | | | | |
| A24-P8SF | Flat Panel | 8.5 | Fixed | 2 m | N/A | 2.1 |
| A24-P8NF | Flat Panel | 8.5 | Fixed | 2 m | N/A | 2.1 |
| A24-P13NF | Flat Panel | 13.0 | Fixed | 2 m | 2.7 | 6.6 |
| A24-P14NF | Flat Panel | 14.0 | Fixed | 2 m | 3.7 | 7.6 |
| A24-P15NF | Flat Panel | 15.0 | Fixed | 2 m | 4.7 | 8.6 |
| A24-P16NF | Flat Panel | 16.0 | Fixed | 2 m | 5.7 | 9.6 |
| A24-P19NF | Flat Panel | 19.0 | Fixed | 2 m | 8.7 | 12.6 |
| **Yagi antennas** | | | | | | |
| A24-Y6NF | Yagi (6-element) | 8.8 | Fixed | 2 m | N/A | 1.9 |
| A24-Y7NF | Yagi (7-element) | 9.0 | Fixed | 2 m | N/A | 2.1 |
| A24-Y9NF | Yagi (9-element) | 10.0 | Fixed | 2 m | N/A | 3.1 |
| A24-Y10NF | Yagi (10-element) | 11.0 | Fixed | 2 m | 0.6 | 4.1 |

| Part Number | Type (Description) | Gain (dBi) | Application* | Min Separation | Required antenna cable loss (dB) | |
|---|---|---|---|---|---|---|
| | | | | | Channels 11-23† | Channel 24† |
| A24-Y12NF | Yagi (12-element) | 12.0 | Fixed | 2 m | 1.6 | 5.1 |
| A24-Y13NF | Yagi (13-element) | 12.0 | Fixed | 2 m | 1.6 | 5.1 |
| A24-Y15NF | Yagi (15-element) | 12.5 | Fixed | 2 m | 2.1 | 5.6 |
| A24-Y16NF | Yagi (16-element) | 13.5 | Fixed | 2 m | 3.1 | 6.6 |
| A24-Y16RM | Yagi (16-element, RPSMA connector) | 13.5 | Fixed | 2 m | 3.1 | 6.6 |
| A24-Y18NF | Yagi (18-element) | 15.0 | Fixed | 2 m | 4.6 | 8.1 |

### *XBee-PRO S2C TH RF Module*

The following table shows the antennas approved for use with the XBee-PRO S2C TH RF Module.

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | |
|---|---|---|---|---|---|---|
| | | | | | Channels 11-23† | Channel 24† |
| **Integral antennas** | | | | | | |
| 29000294 | Integral PCB antenna | -0.5 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-QI | Monopole (Integrated whip) | 1.5 | Fixed/Mobile | 25 cm | N/A | N/A |
| **Dipole antennas** | | | | | | |
| A24-HASM-450 | Dipole (Half-wave articulated RPSMA - 4.5”) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-HABSM | Dipole (Articulated RPSMA) | 2.1 | Fixed | 25 cm | N/A | N/A |
| 29000095 | Dipole (Half-wave articulated RPSMA - 4.5”) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-HABUF-P5I | Dipole (Half-wave articulated bulkhead mount U.FL. w/ 5” pigtail) | 2.1 | Fixed | 25 cm | N/A | N/A |
| A24-HASM-525 | Dipole (Half-wave articulated RPSMA - 5.25") | 2.1 | Fixed/ Mobile | 25 cm | N/A | N/A |
| **Omni-directional antennas** | | | | | | |
| A24-F2NF | Omni-directional (Fiberglass base station) | 2.1 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-F3NF | Omni-directional (Fiberglass base station) | 3.0 | Fixed/Mobile | 25 cm | N/A | N/A |
| A24-F5NF | Omni-directional (Fiberglass base station) | 5.0 | Fixed | 25 cm | N/A | N/A |
| A24-F8NF | Omni-directional (Fiberglass base station) | 8.0 | Fixed | 2 m | N/A | N/A |

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | |
|---|---|---|---|---|---|---|
| | | | | | Channels 11-23† | Channel 24† |
| A24-F9NF | Omni-directional (Fiberglass base station) | 9.5 | Fixed | 2 m | N/A | N/A |
| A24-F10NF | Omni-directional (Fiberglass base station) | 10.0 | Fixed | 2 m | N/A | N/A |
| A24-F12NF | Omni-directional (Fiberglass base station) | 12.0 | Fixed | 2 m | N/A | 1.4 |
| A24-W7NF | Omni-directional (base station) | 7.2 | Fixed | 2 m | N/A | N/A |
| A24-M7NF | Omni-directional (Mag-mount base station) | 7.2 | Fixed | 2 m | N/A | N/A |
| A24-F15NF | Omni-directional (Fiberglass base station) | 15.0 | Fixed | 2 m | 0.4 | 4.4 |
| **Panel antennas** | | | | | | |
| A24-P8SF | Flat Panel | 8.5 | Fixed | 2 m | N/A | 0.4 |
| A24-P8NF | Flat Panel | 8.5 | Fixed | 2 m | N/A | 0.4 |
| A24-P13NF | Flat Panel | 13 | Fixed | 2 m | 2.4 | 4.9 |
| A24-P14NF | Flat Panel | 14 | Fixed | 2 m | 3.4 | 5.9 |
| A24-P15NF | Flat Panel | 15.0 | Fixed | 2 m | 4.4 | 6.9 |
| A24-P16NF | Flat Panel | 16.0 | Fixed | 2 m | 5.4 | 7.9 |
| A24-19NF | Flat Panel | 19.0 | Fixed | 2 m | 8.4 | 10.9 |
| **Yagi antennas** | | | | | | |
| A24-Y6NF | Yagi (6-element) | 8.8 | Fixed | 2 m | N/A | 1.2 |
| A24-Y7NF | Yagi (7-element) | 9.0 | Fixed | 2 m | N/A | 1.4 |
| A24-Y9NF | Yagi (9-element) | 10.0 | Fixed | 2 m | N/A | 2.4 |
| A24-Y10NF | Yagi (10-element) | 11.0 dBi | Fixed | 2 m | 0.4 | 3.4 |

| Part number | Type (description) | Gain (dBi) | Application* | Min. separation | Required antenna cable loss (dB) | |
|---|---|---|---|---|---|---|
| | | | | | Channels 11-23† | Channel 24† |
| A24-Y12NF | Yagi (12-element) | 12.0 | Fixed | 2 m | 1.4 | 4.4 |
| A24-Y13NF | Yagi (13-element) | 12.0 | Fixed | 2 m | 1.4 | 4.4 |
| A24-Y15NF | Yagi (15-element) | 12.5 | Fixed | 2 m | 1.9 | 4.9 |
| A24-Y16NF | Yagi (16-element) | 13.5 | Fixed | 2 m | 2.9 | 5.9 |
| A24-Y16RM | Yagi (16-element, RPSMA connector) | 13.5 | Fixed | 2 m | 2.9 | 5.9 |
| A24-Y18NF | Yagi (18-element) | 15.0 | Fixed | 2 m | 4.4 | 7.4 |

* 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 25 cm 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.

† Although certified to operate on channels 11-24, currently this product only supports channels 12-23.

## RF exposure

If you are an integrating the XBee into another product, you must include the following Caution statement in OEM product manuals to alert users of FCC RF exposure compliance:

---

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

---

## FCC publication 996369 related information

In publication 996369 section D03, the FCC requires information concerning a module to be presented by OEM manufacturers. This section assists in answering or fulfilling these requirements.

### *2.1 General*

No requirements are associated with this section.

### *2.2 List of applicable FCC rules*

This module conforms to FCC Part 15.247.

### *2.3 Summarize the specific operational use conditions*

Certain approved antennas require attenuation for operation. For the XBee/XBee-PRO S2C 802.15.4 RF Module, see FCC-approved antennas (2.4 GHz).

Host product user guides should include the antenna table if end customers are permitted to select antennas.

### *2.4 Limited module procedures*

Not applicable.

### *2.5 Trace antenna designs*

While it is possible to build a trace antenna into the host PCB, this requires at least a Class II permissive change to the FCC grant which includes significant extra testing and cost. If an embedded trace antenna is desired, select the XBee module variant with the preferred antenna.

### *2.6 RF exposure considerations*

For RF exposure considerations see RF exposure and FCC-approved antennas (2.4 GHz).

Host product manufacturers need to provide end-users a copy of the “RF Exposure” section of the manual: RF exposure.

### *2.7 Antennas*

A list of approved antennas is provided for the XBee/XBee-PRO S2C 802.15.4 RF Modules. See FCC-approved antennas (2.4 GHz).

### 2.8 Label and compliance information

Host product manufacturers need to follow the sticker guidelines outlined in OEM labeling requirements.

### 2.9 Information on test modes and additional testing requirements

Contact a Digi sales representative for information on how to configure test modes for the XBee/XBee-PRO S2C 802.15.4 RF Module.

### 2.10 Additional testing, Part 15 Subpart B disclaimer

All final host products must be tested to be compliant to FCC Part 15 Subpart B standards. While the XBee/XBee-PRO S2C 802.15.4 module was tested to be complaint to FCC unintentional radiator standards, FCC Part 15 Subpart B compliance testing is still required for the final host product. This testing is required for all end products, and XBee/XBee-PRO S2C 802.15.4 module Part 15 Subpart B compliance does not affirm the end product's compliance.

See FCC notices for more details.

# Europe (CE)

The XBee/XBee-PRO S2C 802.15.4 RF Modules (non-PRO variants) have been tested for use in several European countries. For a complete list, refer to www.digi.com/resources/certifications.

If XBee/XBee-PRO S2C 802.15.4 RF Modules are incorporated into a product, the manufacturer must ensure compliance of the final product with articles 3.1a and 3.1b of the Radio Equipment Directive. A Declaration of Conformity must be issued for each of these standards and kept on file as described in the Radio Equipment Directive.

Furthermore, the manufacturer must maintain a copy of the XBee/XBee-PRO S2C 802.15.4 RF Module user guide documentation and ensure the final product does not exceed the specified power ratings, antenna specifications, and/or installation requirements as specified in the user guide.

## Maximum power and frequency specifications

For the through-hole device:

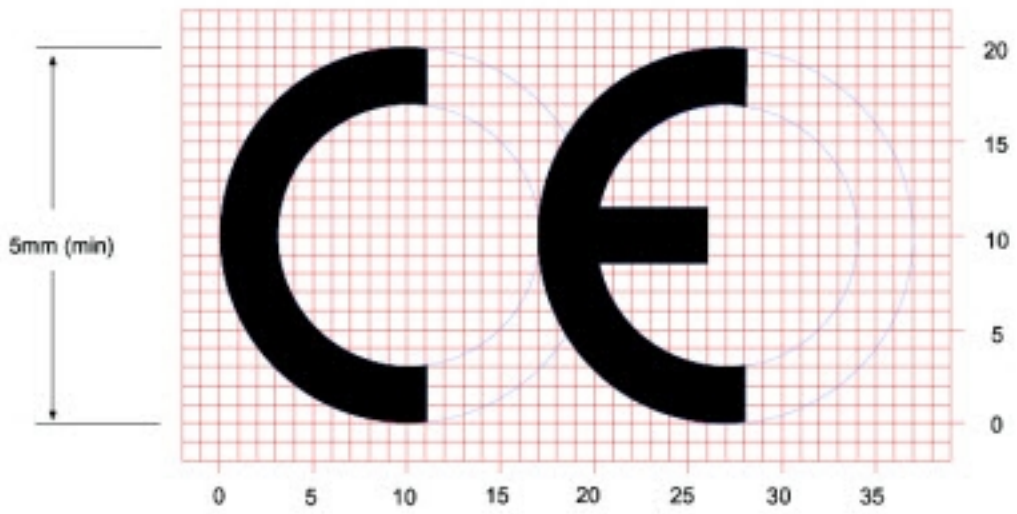
- Maximum power: 9.82 mW (9.92 dBm) Equivalent Isotropically Radiated Power (EIRP) at normal condition.
- Frequencies: 5 MHz channel spacing, beginning at 2405 MHz and ending at 2480 MHz.

For the surface-mount device:

- Maximum power: 12.65 mW (11.02 dBm) EIRP.
- Frequencies: 5 MHz channel spacing, beginning at 2405 MHz and ending at 2480 MHz.

## OEM labeling requirements

The "CE" marking must be affixed to a visible location on the OEM product. The following figure shows CE labeling requirements.




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

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

Digi customers assume full responsibility for learning and meeting the required guidelines for each country in their distribution market. Refer to the radio regulatory agency in the desired countries of operation for more information.

## Listen Before Talk requirement

The XBee/XBee-PRO S2C 802.15.4 RF Module must be configured to comply with the Listen Before Talk (LBT) requirements in the EN 300 328 standard. This can be accomplished by one of the following options:

1. Set the **PL** command to 3 (6 dBm) or lower, which ensures that the maximum transmitter power is under the limit at which LBT is required.
   or
2. Set the **CA** command as described in CA (CCA Threshold) to enable LBT at the required noise threshold level.

## Declarations of conformity

Digi has issued Declarations of Conformity for the XBee RF Modules concerning emissions, EMC, and safety. For more information, see www.digi.com/resources/certifications.

## Antennas

The following antennas have been tested and approved for use with the XBee/XBee-PRO S2C 802.15.4 RF Module:

All antenna part numbers followed by an asterisk (*) are not available from Digi. Consult with an antenna manufacturer for an equivalent option.

- Dipole (2.1 dBi, Omni-directional, Articulated RPSMA, Digi part number A24-HABSM)
- PCB Antenna (0.0 dBi)
- Monopole Whip (1.5 dBi)

# ISED (Innovation, Science and Economic Development Canada)

## 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:

## For XBee S2C surface-mount

Contains Model XBee S2C Radio, IC: 1846A-XBS2C

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.

## For XBee-PRO S2C surface-mount

Contains Model PS2CSM Radio, IC: 1846A-PS2CSM

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.

## For XBee S2C through-hole

Contains Model S2CTH Radio, IC: 1846A-S2CTH

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.

## For XBee-PRO S2C through-hole

Contains Model PS2CTH Radio, IC: 1846A-PS2CTH

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 for detachable antennas

This radio transmitter has been approved by Industry Canada to operate with the antenna types listed in the tables in FCC-approved antennas (2.4 GHz) 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. The required antenna impedance is 50 ohms.

*Le présent émetteur radio a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés 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 (EIRP) 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 (RCM)

XBee S2C 802.15.4 and XBee-PRO S2C 802.15.4 modules comply with requirements to be used in end products in Australia and New Zealand. All products with EMC and radio communications must have registered RCM and R-NZ marks. Registration to use the compliance mark will only be accepted from Australia or New Zealand manufacturers or importers, or their agents.

In order to have a RCM or R-NZ mark on an end product, a company must comply with **a** or **b** below.

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

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

## South Korea

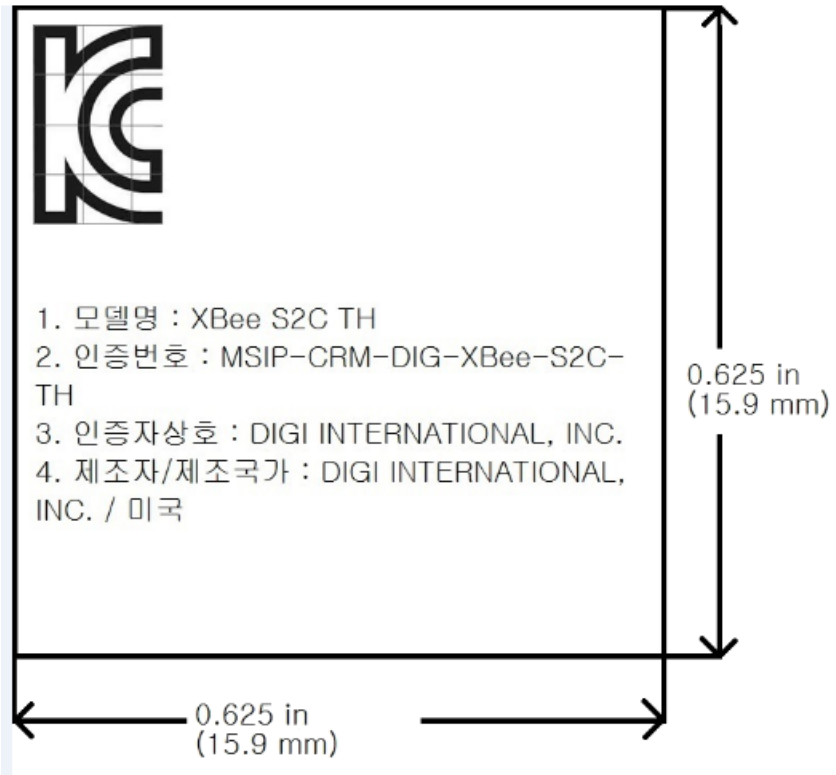
The low-power XBee S2C TH and XBee S2C devices have received South Korean approvals. To show conformity to the certificate, you must add a label with the South Korean product information to the XBee S2C 802.15.4 RF Module.

For the through-hole device, you can place the label on the reverse side.

Recommended label material: Abraham Technical (700342) MFG P/N TAAE-014250.

The label size is: 15.9 mm x 15.9 mm (0.625 in x 0.625 in)

The complete label information is as follows:

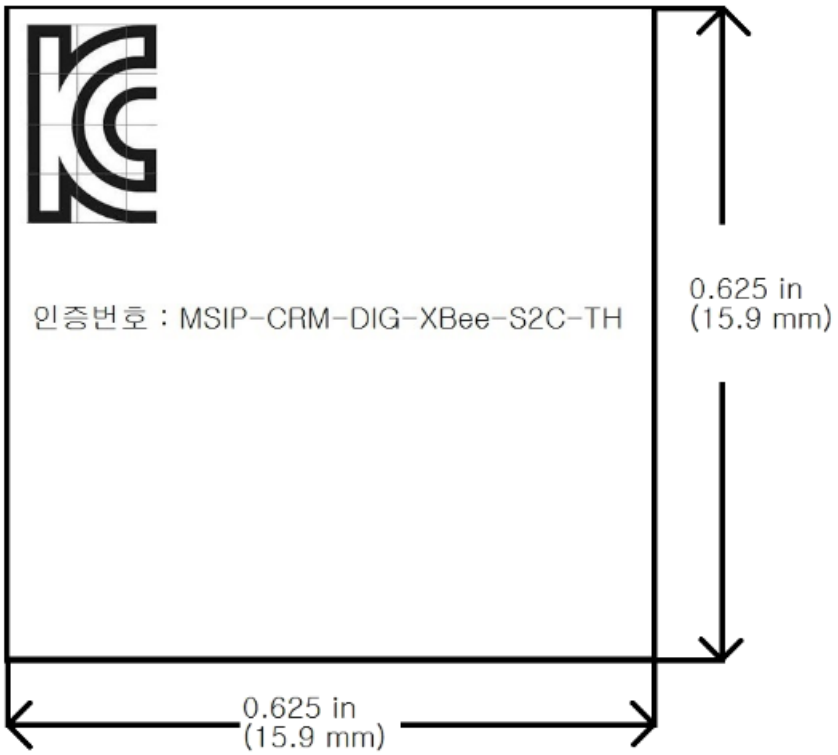



The KCC logo must be at least 5 mm tall.

The text shown in the label is:

1. 모델명 : XBee S2C TH
2. 인증번호 : MSIP-CRM-DIG-XBee-S2C-TH
3. 인증자상호 : DIGI INTERNATIONAL, INC.
4. 제조자/제조국가 : DIGI INTERNATIONAL, INC. / 미국

If the label size does not accommodate the required content, you can use abbreviated information, as follows:




The KCC logo must be at least 5 mm tall.

The text shown on the label is:

인증번호 : MSIP-CRM-DIG-XBee-S2C-TH

For the surface-mount version, the label will overlay the existing product label.

---

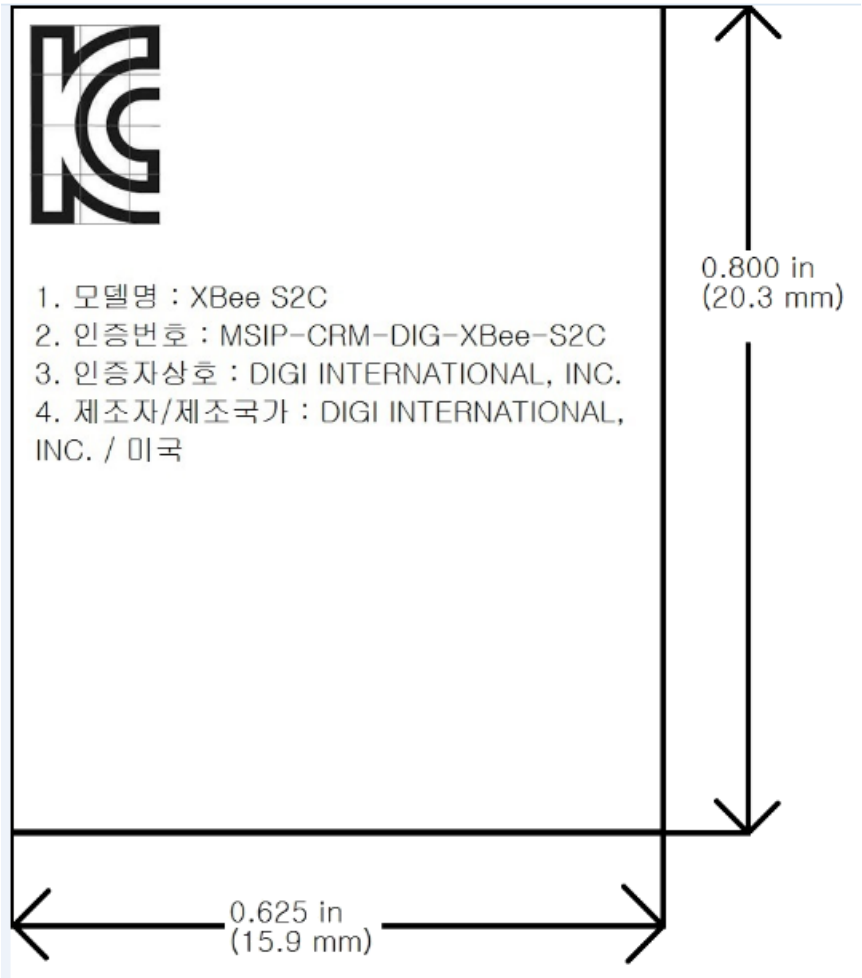
**CAUTION!** By placing a label over the existing label, the certifications for Europe (CE), Australia, New Zealand (RCM), and Japan will no longer apply.

---

Recommended label material: Abraham Technical TELT-000465.

The label size is: 15.9 mm x 20.3 mm (0.625 in x 0.8 in)

The complete label information is as follows:

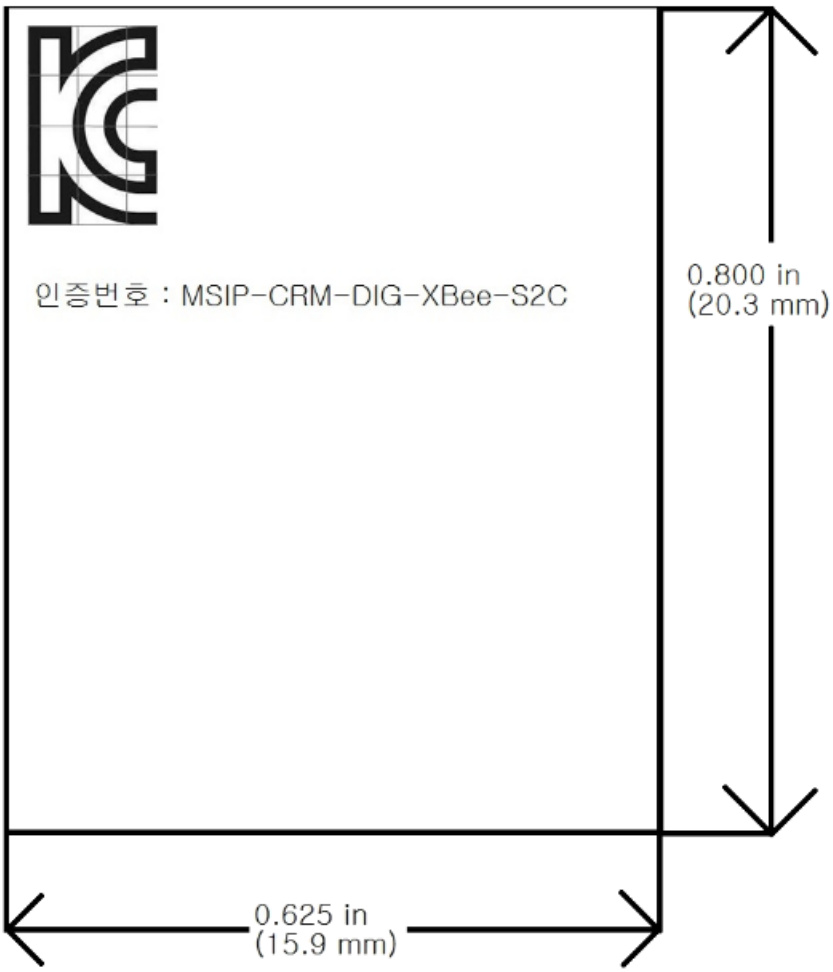



The KCC logo must be at least 5 mm tall.

The text shown in the label is:

1. 모델명 : XBee S2C
2. 인증번호 : MSIP-CRM-DIG-XBee-S2C
3. 인증자상호 : DIGI INTERNATIONAL, INC.
4. 제조자/제조국가 : DIGI INTERNATIONAL, INC. / 미국

If the label size does not accommodate the required content, you can use the abbreviated information, as follows:




The KCC logo must be at least 5 mm tall.

The text shown in the label is:

인증번호 : MSIP-CRM-DIG-XBee-S2C

# Load 802.15.4 firmware on ZB devices

Background ..... 154
Load 802.15.4 firmware ..... 154

# Background

Our XBee/XBee-PRO ZB RF modules are built on the same hardware as the XBee/XBee-PRO S2C 802.15.4 RF Module. It is possible to load 802.15.4 firmware on existing ZB modules. The table below shows which part numbers are compatible with 802.15.4 firmware.

**Note** Currently the 802.15.4 firmware is approved for use only in the United States, Canada, Europe, Australia and Japan. You can find region-specific regulatory information for the firmware in Regulatory information.

**CAUTION!** The antenna cable loss requirements for the 802.15.4 firmware are different than the ZB firmware for gain antennas exceeding 2.1 dBi. If you migrate a ZB device to 802.15.4 firmware, and are using gain antennas, you must adhere to the cable loss requirements found in Regulatory information.

| XBee/XBee-PRO ZB S2C part numbers | Revision | Form factor | Hardware version (HV) |
|---|---|---|---|
| XB24CZ7PIS-004<br>XB24CZ7RIS-004<br>XB24CZ7UIS-004 | All | XBee SMT | 0x22 |
| XB24CZ7PIT-004<br>XB24CZ7SIT-004<br>XB24CZ7UIT-004<br>XB24CZ7WIT-004 | All | XBee TH | 0x2E |
| XBP24CZ7PIS-004<br>XBP24CZ7RIS-004<br>XBP24CZ7UIS-004 | Rev L<br>(and later) | XBee SMT | 0x30 |
| XBP24CZ7PIT-004<br>XBP24CZ7SIT-004<br>XBP24CZ7UIT-004<br>XBP24CZ7WIT-004 | All | XBee TH | 0x2D |

In addition to the differences between the 802.15.4 and Zigbee protocols, some of the operational features are different between the two firmware versions. For example, the XBee-PRO 802.15.4 supports fewer channels than the Zigbee firmware. It is important that you read and understand this user guide before developing with the 802.15.4 firmware.

# Load 802.15.4 firmware

To load 802.15.4 firmware on an existing ZB module, use the following instructions. You must use the serial interface to perform this update. The device does not support OTA updates for changing 802.15.4 to ZB firmware or vice versa.

1. Verify that your device's part number (listed on the label) is included in the list shown in Background.
2. Install the device in a Digi development board and connect it to your PC.
3. The next steps involve loading firmware using XCTU. To download XCTU and read detailed instructions about it, go to:
   https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu

4. When you get to the **Update firmware** dialog box, in the **Function set** area, click the **802.15.4** option, and the newest firmware version.
5. Click **Update** and follow the instructions.
6. When the updating process successfully completes, your device runs 802.15.4 firmware. You can change back to Zigbee firmware at any time by following the same process and selecting the Zigbee firmware option instead.

# Migrate from XBee through-hole to surface-mount devices

We design the XBee surface-mount and through-hole devices to be compatible with each other and offer the same basic feature set. The surface-mount form factor has more I/O pins. Because the XBee device was originally offered in only the through-hole form factor, we offer this section to help you migrate from the through-hole to the surface-mount form factor.

Pin mapping ..... 157
Mount the devices ..... 158

# Pin mapping

The following table shows the pin mapping for the surface-mount (SMT) pins to the through-hole (TH) pins. The pin names are from the XBee S2C SMT device.

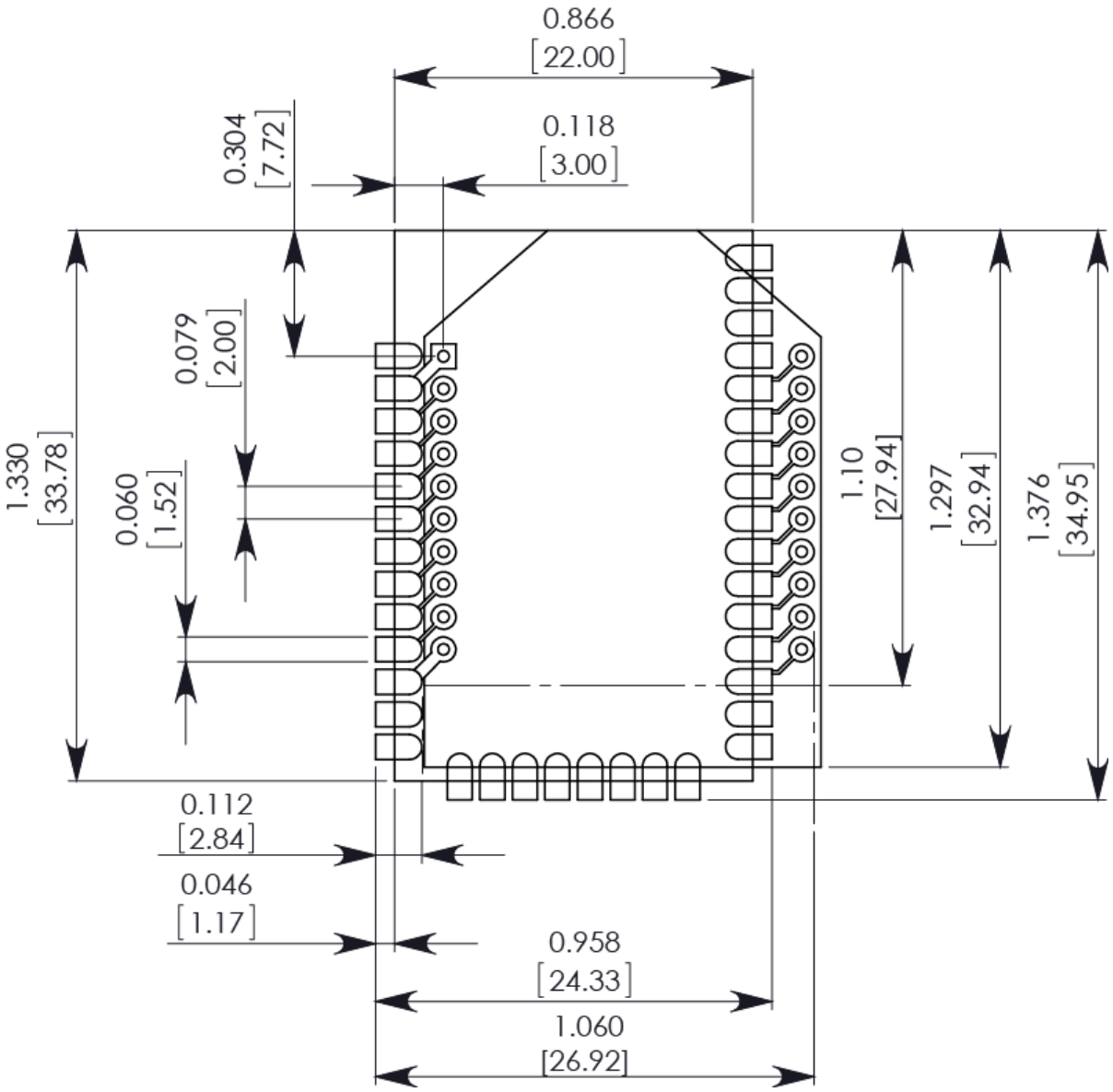
| SMT Pin # | Name | TH Pin # |
|---|---|---|
| 1 | GND | |
| 2 | VCC | 1 |
| 3 | DOUT | 2 |
| 4 | DIN/$\overline{\text{CONFIG}}$ | 3 |
| 5 | [Reserved] | 4 |
| 6 | $\overline{\text{RESET}}$ | 5 |
| 7 | PWM0/RSSI PWM | 6 |
| 8 | PWM1 | 7 |
| 9 | [Reserved] | 8 |
| 10 | DI8/SLEEP_RQ/$\overline{\text{DTR}}$ | 9 |
| 11 | GND | 10 |
| 12 | SPI_$\overline{\text{ATTN}}$ /$\overline{\text{BOOTMODE}}$ | |
| 13 | GND | |
| 14 | SPI_CLK | |
| 15 | SPI_$\overline{\text{SSEL}}$ | |
| 16 | SPI_MOSI | |
| 17 | SPI_MISO | |
| 18 | [Reserved] | |
| 19 | [Reserved] | |
| 20 | [Reserved] | |
| 21 | [Reserved] | |
| 22 | GND | |
| 23 | [Reserved] | |
| 24 | DIO4 | 11 |
| 25 | DIO7/$\overline{\text{CTS}}$ | 12 |
| 26 | On/$\overline{\text{SLEEP}}$ | 13 |
| 27 | $V_{REF}$ | 14 |

| SMT Pin # | Name | TH Pin # |
|---|---|---|
| 28 | DIO5/ASSOC | 15 |
| 29 | DIO6/$\overline{\text{RTS}}$ | 16 |
| 30 | DIO3/AD3 | 17 |
| 31 | DIO2/AD2 | 18 |
| 32 | DIO1/AD1 | 19 |
| 33 | DIO0/AD0 | 20 |
| 34 | [Reserved] | |
| 35 | GND | |
| 36 | RF | |
| 37 | [Reserved] | |

## Mount the devices

One important difference between the SMT and TH devices is the way they mount to a printed circuit board (PCB). Each footprint requires different mounting techniques.

We designed a footprint that allows you to attach either device to a PCB. The following drawing shows the layout.




The round holes in the diagram are for the TH design, and the semi-oval pads are for the SMT design. Pin 1 of the TH design is lined up with pad 1 of the SMT design, but the pins are actually offset by one pad; see Pin mapping. By using diagonal traces to connect the appropriate pins, the layout will work for both devices.

PCB design and manufacturing contains information on attaching the SMT device.

# PCB design and manufacturing

The XBee/XBee-PRO S2C 802.15.4 RF Module is designed for surface-mount on the OEM PCB. It has castellated pads to allow for easy solder attach inspection. The pads are all located on the edge of the module, so there are no hidden solder joints on these modules.

Recommended solder reflow cycle ..... 161
Recommended footprint and keepout ..... 161
Flux and cleaning ..... 163
Rework ..... 163

# Recommended solder reflow cycle

The following table provides the recommended solder reflow cycle. The table shows the temperature setting and the time to reach the temperature; it does not show the cooling cycle.

| Time (seconds) | Temperature (degrees C) |
|---|---|
| 30 | 65 |
| 60 | 100 |
| 90 | 135 |
| 120 | 160 |
| 150 | 195 |
| 180 | 240 |
| 210 | 260 |

The maximum temperature should not exceed 260 °C.

The device will reflow during this cycle, and therefore must not be reflowed upside down. Take care not to jar the device while the solder is molten, as this can remove components under the shield from their required locations.

Hand soldering is possible and should be performed in accordance with approved standards.

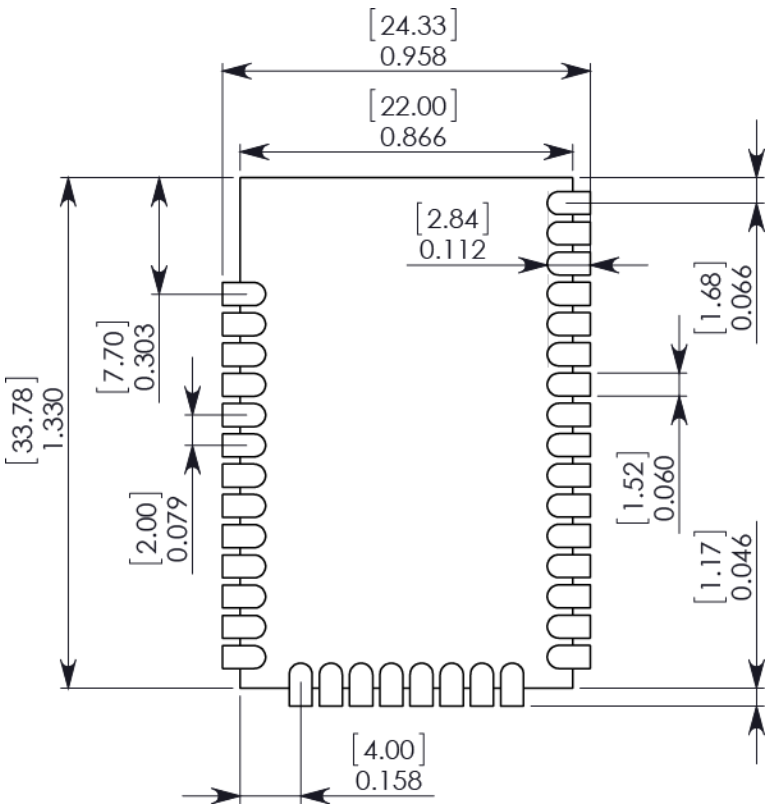
The device has a Moisture Sensitivity Level (MSL) of 3. When using this product, consider the relative requirements in accordance with standard IPC/JEDEC J-STD-020.

In addition, note the following conditions:

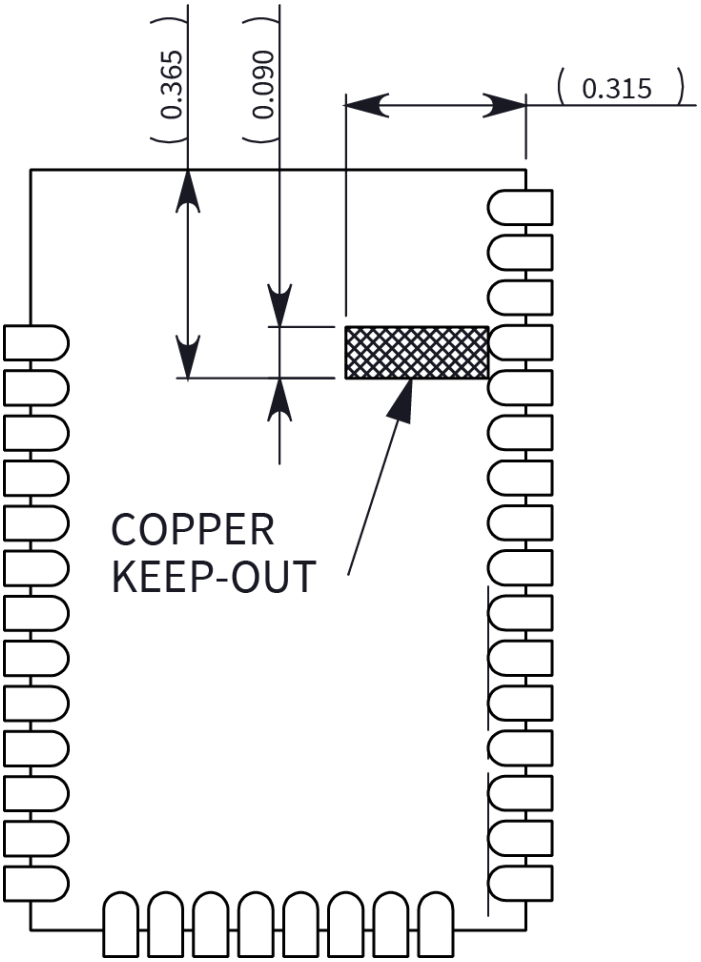
a. Calculated shelf life in sealed bag: 12 months at < 40 °C and < 90% relative humidity (RH).
b. Environmental condition during the production: 30 °C /60% RH according to IPC/JEDEC J-STD-033C, paragraphs 5 through 7.
c. The time between the opening of the sealed bag and the start of the reflow process cannot exceed 168 hours if condition b) is met.
d. Baking is required if conditions b) or c) are not met.
e. Baking is required if the humidity indicator inside the bag indicates a RH of 10% more.
f. If baking is required, bake modules in trays stacked no more than 10 high for 4-6 hours at 125 °C.

# Recommended footprint and keepout

We recommend that you use the following PCB footprints for surface-mounting. The dimensions without brackets are in inches, and those in brackets are in millimeters.




While the underside of the module is mostly coated with solder resist, we recommend that the copper layer directly below the module be left open to avoid unintended contacts. Copper or vias must not interfere with the three exposed RF test points on the bottom of the module (see below). Furthermore, these modules have a ground plane in the middle on the back side for shielding purposes, which can be affected by copper traces directly below the module.




## Flux and cleaning

We recommend that you use a “no clean” solder paste in assembling these devices. This eliminates the clean step and ensures that you do not leave unwanted residual flux under the device where it is difficult to remove. In addition:

- Cleaning with liquids can result in liquid remaining under the device or in the gap between the device and the host PCB. This can lead to unintended connections between pads.
- The residual moisture and flux residue under the device are not easily seen during an inspection process.

## Rework

**CAUTION!** Any modification to the device voids the warranty coverage and certifications.

Rework should never be performed on the module itself. The module has been optimized to give the best possible performance, and reworking the module itself will void warranty coverage and certifications. We recognize that some customers will choose to rework and void the warranty; the

following information is given as a guideline in such cases to increase the chances of success during rework, though the warranty is still voided.

The module may be removed from the OEM PCB by the use of a hot air rework station, or hot plate. Care should be taken not to overheat the module. During rework, the module temperature may rise above its internal solder melting point and care should be taken not to dislodge internal components from their intended positions.