



# XTend vB

Radio Frequency (RF) Module

---

User Guide

## Revision history—90001478

Revision	Date	Description
A	December, 2015	Baseline release of the document.
B	May, 2016	Added information on the Australian variant. Updated cyclic sleep numbers. Added the <b>HS</b> command.

## 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.

© 2016 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:

<http://www.digi.com/howtobuy/terms>

## Send comments

**Documentation feedback:** To provide feedback on this document, send your comments to [techcomm@digi.com](mailto:techcomm@digi.com).

## Customer support

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

Online: [www.digi.com/support/eservice](http://www.digi.com/support/eservice)

# Contents

---

Revision history—90001478 .....	2
---------------------------------	---

## XTend vB RF Module User Guide

Applicable firmware and hardware .....	9
XTend replacement numbers .....	9
Certification overview .....	9

## Technical specifications

General specifications .....	11
Performance specifications .....	11
Networking specifications .....	12
Power requirements .....	12
Cyclic sleep current (mA, average) .....	12
Regulatory approvals .....	13

## Hardware

Connect the hardware .....	15
Mechanical drawings .....	16
Pin signals .....	16

## Modes

Transparent and API operating modes .....	20
Transparent operating mode .....	20
API operating mode .....	20
Additional modes .....	20
Command mode .....	20
Binary Command mode .....	20
Idle mode .....	20
Receive mode .....	21
Sleep modes .....	21
Shutdown mode .....	21
Transmit mode .....	21
Enter Command mode .....	21
Send AT commands .....	22
Exit Command mode .....	22
Enter Binary Command mode .....	22
Exit Binary Command mode .....	23
Binary Command mode FAQs .....	23
Sleep modes .....	23

Pin Sleep (SM = 1) .....	24
Serial Port Sleep (SM = 2) .....	25
Cyclic Sleep Mode (SM = 4-8) .....	25

## Operation

Serial interface .....	28
UART data flow .....	28
Serial data .....	28
Flow control .....	29
Data In (DIN) buffer and flow control .....	29
Data Out (DO) buffer and flow control .....	30

## Configure the XTend vB RF Module

Configure the device using XCTU .....	32
---------------------------------------	----

## Program the XTend vB RF Module

Programming examples .....	33
Connect the device to a PC .....	33
Modify a device address .....	33
Restore device defaults .....	34
Send binary commands .....	34
Query binary commands .....	35

## XTend vB RF Module commands

Command mode options .....	38
AT (Guard Time After) .....	38
BT (Guard Time Before) .....	39
CC (Command Sequence Character) .....	39
CF (Number Base) .....	39
CN (Exit Command Mode) .....	40
CT (Command Mode Timeout) .....	40
E0 (Echo Off) .....	41
E1 (Echo On) .....	41
Diagnostic commands .....	41
%V (Board Voltage) .....	41
DB (Received Signal Strength) .....	42
GD (Receive Good Count) .....	42
HV (Hardware Version) .....	43
RC (Ambient Power - Single Channel) .....	43
RE (Restore Defaults) .....	43
RM (Ambient Power) .....	45
RP (RSSI PWM Timer) .....	45
SH (Serial Number High) .....	46
SL (Serial Number Low) .....	46
TP (Board Temperature) .....	47
TR (Transmit Error Count) .....	47
VL (Firmware Version - Verbose) .....	47
VR (Firmware Version - Short) .....	48
WA (Active Warning Numbers) .....	48

WN (Warning Data) .....	49
WS (Sticky Warning Numbers) .....	50
HS (Hardware Series) .....	51
MAC/PHY commands .....	51
AM (Auto-set MY) .....	51
DT (Destination Address) .....	51
HP (Preamble ID) .....	52
ID (Network ID) .....	52
MK (Address Mask) .....	53
MT (Multi-transmit) .....	53
MY (Source Address) .....	54
RN (Delay Slots) .....	54
RR (Retries) .....	55
TT (Streaming Limit) .....	55
RF interfacing commands .....	55
BR (RF Data Rate) .....	56
FS (Forced Synch Time) .....	56
MD (RF Mode) .....	57
PB (Polling Begin Address) .....	57
PD (Minimum Polling Delay) .....	57
PE (Polling End Address) .....	58
PK (Maximum RF Packet Size) .....	58
PL (TX Power Level) .....	58
TX (Transmit Only) .....	59
Security commands .....	60
KY (AES Encryption Key) .....	60
Serial interfacing commands .....	60
AP (API Enable) .....	60
BD (Interface Data Rate) .....	61
CD (GP02 Configuration) .....	62
CS (GP01 Configuration) .....	63
FL (Software Flow Control) .....	63
FT (Flow Control Threshold) .....	64
NB (Parity) .....	64
RB (Packetization Threshold) .....	64
RO (Packetization Timeout) .....	65
RT (GPI1 Configuration) .....	66
SB (Stop Bits) .....	66
Sleep commands .....	67
FH (Force Wakeup Initializer) .....	67
HT (Time before Wake-up Initializer) .....	67
LH (Wakeup Initializer Timer) .....	68
PW (Pin Wakeup) .....	68
SM (Sleep Mode) .....	69
ST (Time before Sleep) .....	69
Special commands .....	69
WR (Write) .....	69

## API operation

API mode overview .....	72
API frame specifications .....	72
Calculate and verify checksums .....	75
Escaped characters in API frames .....	75
XTend vB RF Module API frame overview .....	76

RF Module Status frame - 0x8A .....	77
Transmit Request: 16-bit address frame - 0x01 .....	78
Transmit Status frame - 0x89 .....	80
Receive Packet: 16-bit address frame - 0x81 .....	81

## Certifications

FCC (United States) .....	84
OEM labeling requirements .....	84
FCC notices .....	84
FCC antenna certifications .....	85
Antenna options .....	86
Industry Canada (IC) .....	91
Labeling requirements .....	91
Transmitters for detachable antennas .....	91
Detachable antennas .....	91
ACMA (Australia) .....	92
Power requirements .....	92

## Network configurations

Network topologies .....	94
Point-to-point networks .....	94
Point-to-multipoint networks .....	94
Peer to peer networks .....	95
Addressing .....	96
Address recognition .....	97
Basic communications .....	97
Streaming mode (default) .....	97
Multi-transmit mode .....	98
Repeater mode .....	99
Polling mode (basic) .....	103
Acknowledged communications: Acknowledged mode .....	104
Acknowledged mode connection sequence .....	104
Polling mode (acknowledged) .....	105

## Development Kit

Development Kit contents .....	108
Interface hardware .....	109
XTIB-R RS-232/485 Interface Board .....	109
Configuration switch .....	110
I/O and Power LEDs .....	110
Serial port .....	110
RSSI LEDs .....	110
Power connector .....	110
XTIB-R Dip switch .....	110
Adapters .....	112
NULL Modem Adapter (male-to-male) .....	113
NULL Modem Adapter (female-to-female) .....	113
Serial Loopback Adapter .....	113
Male DB-9 to RJ-45 Adapter .....	114
Female DB-9 to RJ-45 Adapter .....	114

Interface protocols .....	115
RS-232 operation .....	115
RS-485 (2-wire) operation .....	118
RS-485 (4-wire) and RS-422 operation .....	120

# XTend vB RF Module User Guide

---

The XTend vB RF Module was engineered to provide customers with an easy-to-use radio frequency (RF) solution that provides reliable delivery of critical data between remote devices. The module transfers a standard asynchronous serial data stream, operates within the ISM 900 MHz frequency band and offers two RF data rates of 10 kb/s and 125 kb/s for the United States and Canada variant. It offers two RF data rates of 10 kb/s and 105 kb/s for the Australia variant.

Applicable firmware and hardware .....	9
XTend replacement numbers .....	9
Certification overview .....	9

## Applicable firmware and hardware

As the name suggests, the XTend vB RF Module is form factor and over the air compatible with our XTend module.

This manual supports firmware versions 2xxx for XTend DigiMesh.

## XTend replacement numbers

The following table provides the part numbers you can use to replace XTend devices with the XTend vB RF Module.

Legacy part number	Replacement part number
XT09-MI	XTP9B-DPM-001
XT09-SI	XTP9B-DPS-001
XT09-MI-MESH	XTP9B-DMM-001
XT09-SI-MESH	XTP9B-DMS-001

## Certification overview

The XTend vB RF Module contains an FCC/IC approved RF module. A separate variant of the XTend vB RF Module contains an Australian approved RF module. For usage requirements, see [Certifications](#).

ISM (Industrial, Scientific and Medical) license-free 902-928 MHz frequency band.

Manufactured under ISO 9001:2000 registered standards.

# Technical specifications

---

The following tables provide the device's technical specifications.



**WARNING!** When operating at 1 W power output, observe a minimum separation distance of 6 ft (2 m) between devices. Transmitting in close proximity of other devices can damage the device's front end.

---

General specifications .....	11
Performance specifications .....	11
Networking specifications .....	12
Power requirements .....	12
Regulatory approvals .....	13

## General specifications

The following table provides the general specifications for the device.

Specification	Value
Dimensions (RF/pin connectors not included)	3.70 x 6.10 x 0.48 cm (1.457 x 2.402 x 0.190 in)
Weight	16 g
RoHS	Compliant
Manufacturing	ISO 9001:2000 registered standards
Connector	20 pin 2 mm pitch header
Antenna connector options	MMCX or RPSMA
Antenna impedance	50 ohms unbalanced
Operating temperature	-40°C to 85°C
Maximum input RF level at antenna port	6 dBm
Digital outputs	2 output lines

## Performance specifications

The following table provides the performance specifications for the device.

Specification	Value	
Frequency range	902 to 928 MHz US/Canada 915 to 928 MHz Australia	
RF data rate (software selectable)	10 kb/s to 125 kb/s US/Canada 10 kb/s to 105 kb/s Australia	
Transmit power (software selectable)	Up to 30 dBm (see <a href="#">Power requirements</a> )	
Channels	10 hopping sequences share 50 frequencies	
Outdoor line of sight	10 kb/s	Up to 40 miles <sup>1</sup>
	125 kb/s	Up to 7 miles
Indoor range line of sight	10 kb/s	Up to 1,000 feet (300 m)
	125 kb/s	Up to 500 feet (150 m)
Receiver sensitivity	10 kb/s	-110 dBm
	125 kb/s	-100 dBm
UART data rate	1200-230400 baud	

---

<sup>1</sup>Estimated based on a 9 mile range test with dipole antennas.

## Networking specifications

The following table provides the general specifications for the device.

Specification	Value
Modulation	Frequency Shift Keying
Spread Spectrum	Frequency Hopping Spread Spectrum (FHSS)
Supported Network Topologies (software selectable)	Peer-to-peer (master/slave relationship not required), point-to-point/point-to-multipoint, mesh
Encryption	256-bit or 128-bit AES CBC encryption depending on region. 256-bit is only available on the North America variant. 128-bit is only available on international variants.

## Power requirements

The following table provides the power requirements for the device. Specifications are given at 5 V, 25°C unless otherwise noted.

Requirement	Value
Supply voltage	2.8 to 5.5 VDC, 5 V typical
Receive current	@ 5 V 35 mA
Transmit current	See the following table
Shutdown mode current	1 $\mu$ A
Sleep current	< 147 $\mu$ A

### Cyclic sleep current (mA, average)

Sleep mode	Cycle time	RF data rate	Cyclic sleep current (mA, average)
<b>SM = 8</b>	16 seconds	<b>BR = 0</b>	0.65
		<b>BR = 1</b>	0.23
<b>SM = 7</b>	8 seconds	<b>BR = 0</b>	1.13
		<b>BR = 1</b>	0.31
<b>SM = 6</b>	4 seconds	<b>BR = 0</b>	2.06
		<b>BR = 1</b>	0.46
<b>SM = 5</b>	2 seconds	<b>BR = 0</b>	3.77
		<b>BR = 1</b>	0.77

Sleep mode	Cycle time	RF data rate	Cyclic sleep current (mA, average)
<b>SM = 4</b>	1 second	<b>BR = 0</b>	6.68
		<b>BR = 1</b>	1.36

Transmit power level	21.5 dBm	27 dBm	30 dBm
Supply voltage range	2.8 to 5.5 V	3.2 to 5.5 V	4.75 to 5.5 V
Transmit current (5 V, typical)	260 mA	470 mA	710 mA
Transmit current (3.3 V, typical)	340 mA	615 mA	N/A

## Regulatory approvals

The following table provides the regulatory approvals for the device.

Nation	Approval
United States	Contains FCC ID: MCQ-XBPSX
Canada	Contains IC: 1846A-XBPSX
Australia	RCM

## Hardware

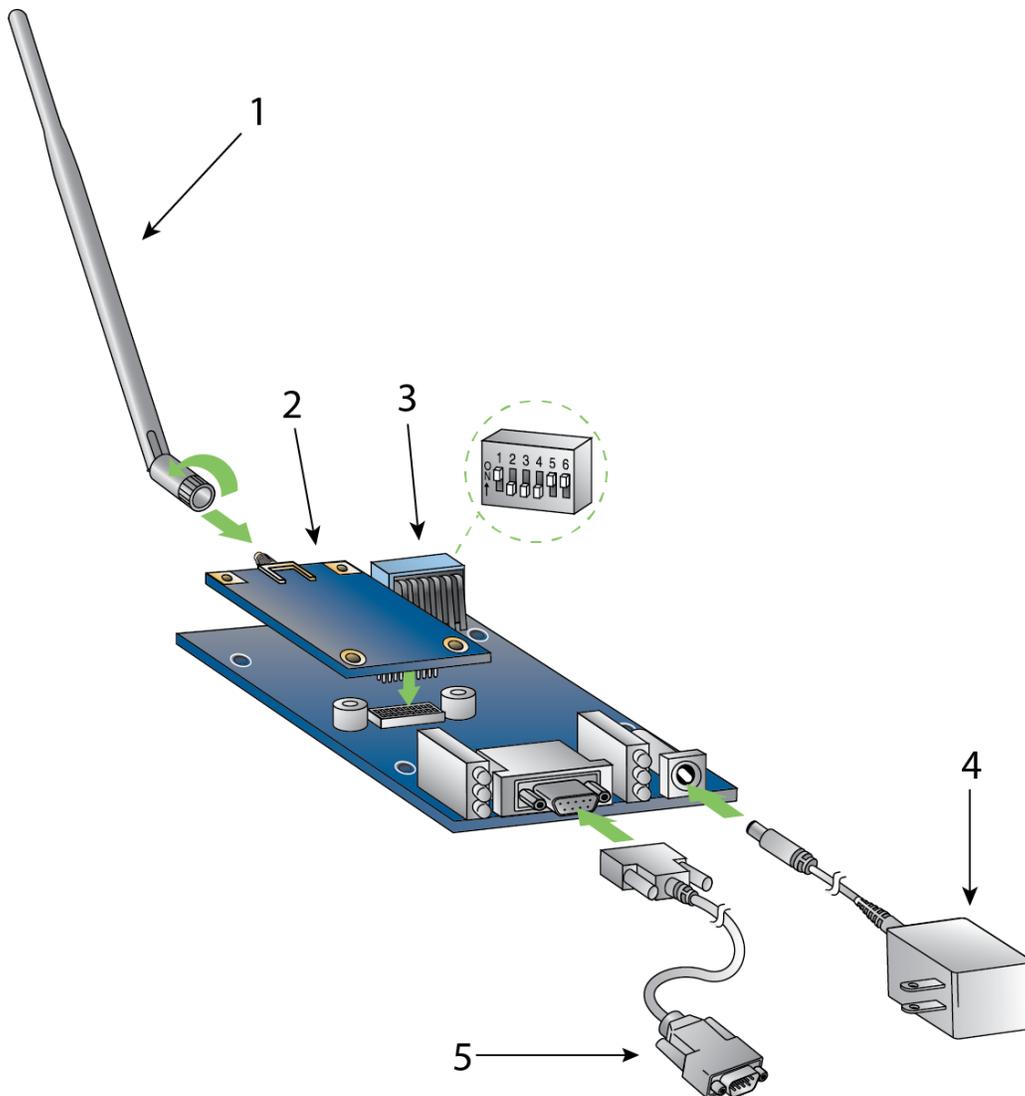
---

Connect the hardware .....	15
Mechanical drawings .....	16
Pin signals .....	16

## Connect the hardware

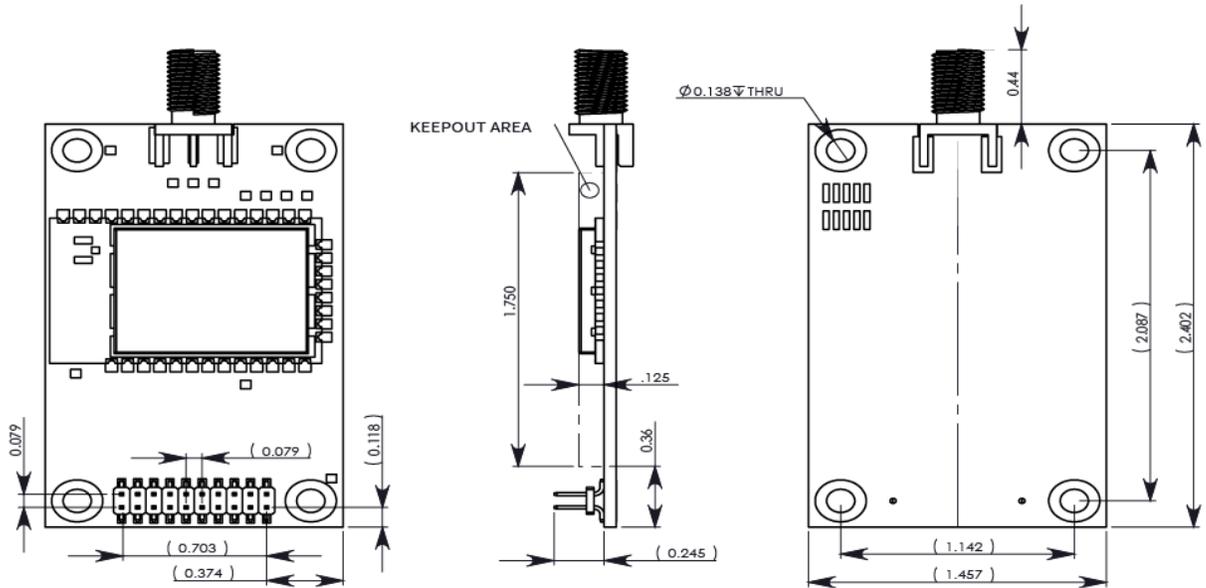
The following figure shows the XTend vB RF Module and accessories you need to get started and how to connect them. The accessories are in the XT09-DK development kit.

Item	Description
1	Antenna, RPSMA (female)
2	XTend vB module, RPSMA version shown
3	DIP switches
4	9 V power supply
5	DB9 serial cable



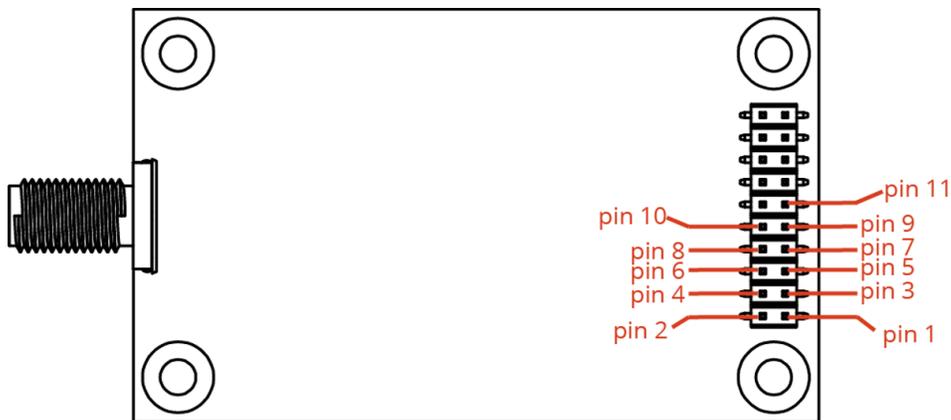
## Mechanical drawings

The following drawings show the dimensions of the device.



## Pin signals

The following drawing shows the location of the pins.



When integrating the module with a Host PC board, all lines not used should be left disconnected (floating).

Pin number	Name	I/O	High impedance during shutdown	Must connect	Function
1	GND	-	-	yes	Ground

Pin number	Name	I/O	High impedance during shutdown	Must connect	Function
2	VCC	I	-	yes	Power: 2.8 - 5.5 VDC
3	GPO2 / RX LED	O	-	yes	GPO2: General Purpose Output. Default ( <b>CD</b> = 2) drives this pin low. RX LED: Pin is driven high during RF data reception; otherwise, the pin is driven low. To enable this pin, see <a href="#">CD (GP02 Configuration)</a> .
4	$\overline{\text{TX\_PWR}}$	O	yes	-	Transmit_Power: Pin pulses low during RF transmission; otherwise, the pin is driven high to indicate power is on and the device is not in Sleep or Shutdown Mode.
5	DIN	I	yes	yes	Data In: Serial data entering the device (from the UART host). For more information, see .
6	DOUT	O	yes	-	Data Out: Serial data exiting the module (to the UART host). For more information, see .
7	$\overline{\text{SHDN}}$	I	no	yes	Shutdown: Drive this pin high to enable normal operation and low during Shutdown. Shutdown enables the lowest power mode available to the module.
8	SLEEP	I	yes	-	SLEEP: By default, SLEEP is not used. To configure this pin to enable Sleep modes, refer to <a href="#">Sleep modes</a> , <a href="#">SM (Sleep Mode)</a> and <a href="#">PW (Pin Wakeup)</a> .

Pin number	Name	I/O	High impedance during shutdown	Must connect	Function
9	GPO1 / $\overline{\text{CTS}}$ / RS-485 TX_EN	O	yes	-	GPO1: General Purpose Output. Pin can be driven low or high. CTS (Clear-to-Send): CTS is enabled by default. When the pin is driven low, the UART host is permitted to send serial data to the device. For more information, see and <a href="#">CS (GP01 Configuration)</a> . RS-485 Transmit Enable: Enables RS-485 half and full-duplex communications. For more information, see and <a href="#">CS (GP01 Configuration)</a> .
10	$\overline{\text{RTS}}$ / CMD	I	yes	-	$\overline{\text{RTS}}$ (Request-to-Send): Not used by default. This pin can be configured to allow the UART host to regulate the flow of serial data exiting the module. For more information, see and <a href="#">RT (GPI1 Configuration)</a> .
11	$\overline{\text{CONFIG}}$ / RSSI	I <sup>1</sup>	no	-	Configuration: Pin can be used as a backup method for entering Command mode during power-up.
		O <sup>2</sup>	no	-	Receive Signal Strength Indicator: By default, pin is used as an RSSI PWM output after at the conclusion of the power-up sequence. The PWM output is 2.8 V-level. For more information, see <a href="#">RP (RSSI PWM Timer)</a> .
12-20	Reserved / do not connect				

---

<sup>1</sup>The RF module has a 10 k $\Omega$  internal pull-up resistor.

<sup>2</sup>The RF module has a 10 k $\Omega$  internal pull-up resistor.

## Modes

---

The RF device is in Receive Mode when it is not transmitting data. The device shifts into the other modes of operation under the following conditions:

- Transmit Mode (Serial data in the serial receive buffer is ready to be packetized)
- Sleep Mode
- Command Mode (Command Mode Sequence is issued, not available when using the SPI port)

Transparent and API operating modes .....	20
Additional modes .....	20
Sleep modes .....	23

## Transparent and API operating 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.

### 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 the AT Command interface.

### API operating mode

API operating mode is an alternative to Transparent mode. 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 originated from. The device communicates UART data in packets, also known as API frames. This mode allows for structured communications with serial devices. 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.

For more information, see [API frame specifications](#).

## Additional modes

In addition to the serial communication modes, several modes apply to how to configure devices and how devices communicate with each other.

### Command mode

Command mode is a state in which the firmware interprets incoming characters as commands. Command mode allows you to modify the device's firmware using parameters you can set using AT commands. When you want to read or set any setting of the device, you have to send it an AT command. Every AT command starts with the letters "AT" followed by the two characters that identify the command the device issues and then by some optional configuration values. For more details, see [Enter Command mode](#).

### Binary Command mode

Binary Command mode allows you to configure a device at a faster rate than AT commands will allow. Using binary commands to send and receive parameter values is the fastest way to change the operating parameters of the device. Use binary commands to:

- Sample signal strength and/or error counts;
- Change device addresses and channels for polling systems when a quick response is necessary.

For more details, see [Enter Binary Command mode](#) and [DB \(Received Signal Strength\)](#).

### Idle mode

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

## Receive mode

If a destination node receives a valid RF packet, the destination node transfers the data to its serial transmit buffer. For the serial interface to report receive data on the RF network, that data must meet the following criteria:

- ID match
- Channel match
- Address match

## Sleep modes

Sleep Modes enable the device to enter states of low-power consumption when not in use. The device support three software sleep modes:

- Pin Sleep: the host controls this
- Serial Port Sleep: wakes when it detects serial port activity
- Cyclic Sleep: wakes when it detects RF activity

For more information, see [Sleep modes](#).

## Shutdown mode

Shutdown mode offers the lowest power mode available to the device. This is helpful for applications that must keep power consumption to a minimum during idle periods.

When you drive the SHDN pin (pin 7) low, it forces the device into Shutdown mode. This halts any communication in progress (transmit or receive) and any buffered data is lost. For any other mode of operation, you must drive or pull SHDN high.

Immediately after the SHDN pin changes states from low to high, the device resets. After reset, the application must observe a delay time of <100 ms.

While SHDN is driven low, the device sets the following pins to high impedance: DCD, TX\_PWR, RX LED, DO and CTS. The SHDN line is driven low during shutdown.

The following input pins may continue to be driven by external circuitry when in shutdown mode: RTS, DI and SHDN.

Because the DO pin is set to high impedance during Shutdown, if the XTend vB RF Module is connected to a processor, the UART receive pin could be floating. Place a weak pull-up between the device and the microcontroller so that the application does not misinterpret noise as data.

## Transmit mode

When the device receives serial data and is ready to packetize it, the device exits Idle mode and attempts to transmit the serial data.

## Enter Command mode

There are two ways to enter Command mode:

1. To get a device to switch into this mode, you must issue a unique string of text in a special way: +++ (default). When the device sees a full second of silence in the data stream followed by the string +++ (without Enter or Return) and another full second of silence, it knows to stop sending data through and start accepting commands locally.  
Do not press Return or Enter after typing +++ because it will interrupt the guard time silence

and prevent you from entering Command mode.

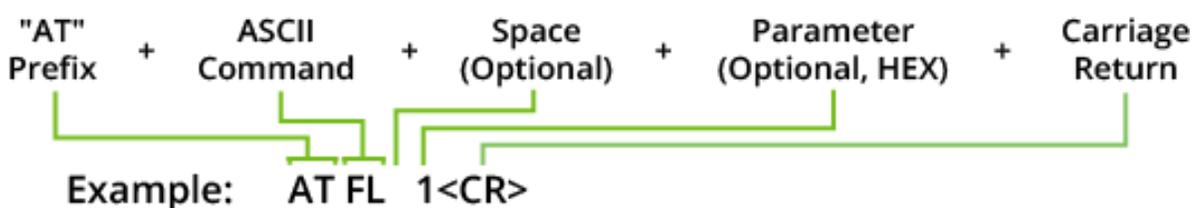
2. Assert (low) the CONFIG pin. Turn the power going to the device off and back on.

The device sends the letters **OK** followed by a carriage return out of the UART to indicate that it entered Command mode.

You can customize the guard times and timeout in the device's configuration settings. See [CC \(Command Sequence Character\)](#), [BT \(Guard Time Before\)](#) and [AT \(Guard Time After\)](#).

## 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 enables software flow control.

## Respond to AT commands

When you send a command to the device, the device parses and runs the command. If the command runs successfully, the device returns an **OK** message. If the command errors, the device returns an **ERROR** message.

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

## Exit Command mode

1. Send the **CN** (Exit Command Mode) command 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 20 seconds.

## Enter Binary Command mode

To enter Binary Command mode, you must first be in Command mode:

1. Set **RT** to 1; see [RT \(GPI1 Configuration\)](#).
2. Assert CMD by driving pin 10 high to enter Binary Command mode.
3. Disable hardware flow control.

$\overline{\text{CTS}}$  (pin ) is high when the firmware executes a command. That is why you must disable hardware flow control, because CTS holds off parameter bytes.

## Exit Binary Command mode

To exit Binary Command mode, de-assert CMD by driving pin 10 low.

## Binary Command mode FAQs

Since sending and receiving binary commands takes place through the same serial data path as live data, interference between the two types of data can be a concern. Some common questions about using binary commands are:

- What are the implications of asserting CMD while the device is sending or receiving live data?

You must assert the CMD pin (pin 10) in order to send binary commands to the device. You can assert the CMD pin to recognize binary commands anytime during the transmission or reception of data.

The device only checks the status of the CMD signal at the end of the stop bit as the byte shifts into the serial port.

The firmware does not allow control over when the device receives data, except by waiting for dead time between bursts of communication.

If the command is sent in the middle of a stream of payload data, the device executes the command in the order it is received. If the device is continuously receiving data, it waits for a break in the data it receives before executing the command.

- After sending serial data, is there a minimum time delay before you can assert CMD?
- Is a time delay required after CMD is de-asserted before payload data can be sent?

The host must observe a minimum time delay of 100  $\mu$ s after sending the stop bit of the command byte before the host de-asserts the CMD pin. The command executes after the host sends all of its associated parameters. If the device does not receive all of these parameters within 0.5 seconds, the device returns to Idle mode.

---

**Note** When a host sends parameters, they are two bytes long with the least significant byte sent first. Binary commands that return one parameter byte must be written with two parameter bytes. Example: to set **PL** to 3, send the following data: 0x3A 0x03 0x00 (Binary Command, LSB, MSB).

---

- How do I discern between live data and data received in response to a command?

To query command parameters using Binary Command mode, set the most significant bit of the binary command. This can be accomplished by logically ORing (bit-wise) the binary command with hexadecimal 0x80. The parameter bytes are returned in hexadecimal bytes with the least significant bit first (if multiple bytes are returned).

Example: to query **HP** in Binary Command mode, instead of setting it, send 0x11 (**HP** binary command) as 0x91 with no parameter bytes.

The device must be in Binary Command mode in order for the device to recognize a binary command; see [Enter Binary Command mode](#).

If the device is not in Binary Command mode (the **RT** parameter value is not 1), the device does not recognize that the CMD pin is asserted and therefore does not recognize the data as binary commands.

For an example of binary programming, see [Send binary commands](#).

## Sleep modes

For the device to enter one of the sleep modes, **SM** must have a non-zero parameter value, and it must meet one of the following conditions:

1. The device is idle (no data transmission or reception) for the amount of time defined by the **ST** parameter. **ST** is only active when **SM** = 4-5.
2. The host asserts SLEEP (pin 10). This only applies to the Pin Sleep option.

When in Sleep mode, the device does not transmit or receive data until it transitions to Idle mode.

Use the **SM** command to enable or disable all Sleep modes. The following table shows the transitions into and out of Sleep modes.

Sleep mode (setting)	Transition into Sleep mode	Transition out of Sleep mode (wake)	Related commands	Power consumption
Pin Sleep ( <b>SM</b> = 1)	Assert (high) SLEEP pin. A microcontroller can shut down and wake devices via the SLEEP pin. The device completes a transmission or reception before activating Pin Sleep.	De-assert (low) SLEEP pin	<b>SM</b>	< 147 $\mu$ A
Serial Port Sleep ( <b>SM</b> = 2)	Automatic transition to Sleep Mode occurs after a user-defined period of inactivity (no transmitting or receiving of data). Period of inactivity is defined by the <b>ST</b> command.	When a serial byte is received on the DI pin	( <b>SM</b> ), <b>ST</b>	7.3 mA
Cyclic Sleep ( <b>SM</b> = 4 - 8)	The device transitions in and out of Sleep Mode in cycles (you set the sleep interval of time using the <b>SM</b> command). The cyclic sleep interval of time must be shorter than the interval of time that is defined by the <b>LH</b> command. You can for the device into Idle Mode using the SLEEP pin if you issue the <b>PW</b> command.		( <b>SM</b> ), <b>ST</b> , <b>HT</b> , <b>LH</b> , <b>PW</b>	See <a href="#">Power requirements</a>

The **SM** (Sleep Mode) command is central to setting all Sleep Mode configurations. By default, Sleep Modes are disabled (**SM** = 0) and the device remains in Idle/Receive Mode. When in this state, the device remains constantly ready to respond to serial or RF activity.

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

After enabling Pin Sleep, the SLEEP pin controls whether the device is active or sleeping. When the host de-asserts SLEEP, the device is fully operational. When the host asserts SLEEP, the device transitions to Sleep mode and remains in its lowest power-consuming state until the host de-asserts the pin. This pin is only active if the device is setup to operate in this mode; otherwise the firmware ignores the pin.

Once in Pin Sleep, the device de-asserts (high)  $\overline{\text{CTS}}$  (pin 9), indicating that other devices should not send data to the device. The device also de-asserts (low) the TX\_PWR line (pin 4) when the device is in Pin Sleep mode.

You cannot assert the SLEEP (pin9) until the transmission of the second byte has started.

---

**Note** The device completes a transmission or reception before activating Pin Sleep.

---

## Serial Port Sleep (SM = 2)

- Wake on serial port activity

Serial Port Sleep is a Sleep mode in which the device runs in a low power state until it detects serial data on the DI pin.

The **ST** command determines the period of time that the device sleeps. Once it receives a character through the DI pin, the device returns to Idle mode and is fully operational.

## Cyclic Sleep Mode (SM = 4-8)

Cyclic Sleep modes allow device wakes according to the times designated by the cyclic sleep settings. If the device detects a wake-up initializer during the time it is awake, the device synchronizes with the transmitting device and receives data after the wake-up initializer runs its duration. Otherwise, the device returns to Sleep mode and continues to cycle in and out of activity until a wake-up initializer is detected.

While the device is in Cyclic Sleep mode, it de-asserts (high)  $\overline{\text{CTS}}$  (pin 9) to indicate not to send data to the device. When the device awakens to listen for data, it asserts CTS and transmits any data received on the DI pin. The device also de-asserts (low) the TX\_PWR (pin 4) when it is in Cyclic Sleep mode.

The device remains in Sleep mode for a user-defined period of time ranging from 1 second to 16 seconds (**SM** parameters 4 through 8). After this interval of time, the device returns to Idle mode and listens for a valid data packet. The listen time depends on the **BR** parameter setting. The default **BR** setting of 1 requires at least a 35 ms wake time, while the **BR** setting of 0 requires a wake time of up to 225 ms. If the device does not detect valid data on any frequency, it returns to Sleep mode. If it detects valid data, it transitions into Receive mode and receives the incoming RF packets. The device then returns to Sleep mode after a period of inactivity determined by the **ST** parameter.

You can also configure the device to wake from cyclic sleep when the SLEEP pin is de-asserted. To configure a device to operate in this manner, you must issue the **PW** (Pin Wake-up) command. When you de-assert the SLEEP pin, it forces the device into Idle mode and it can begin transmitting or receiving data. It remains active until it no longer detects data for the time that **ST** specifies, at which point it resumes its low-power cyclic state.

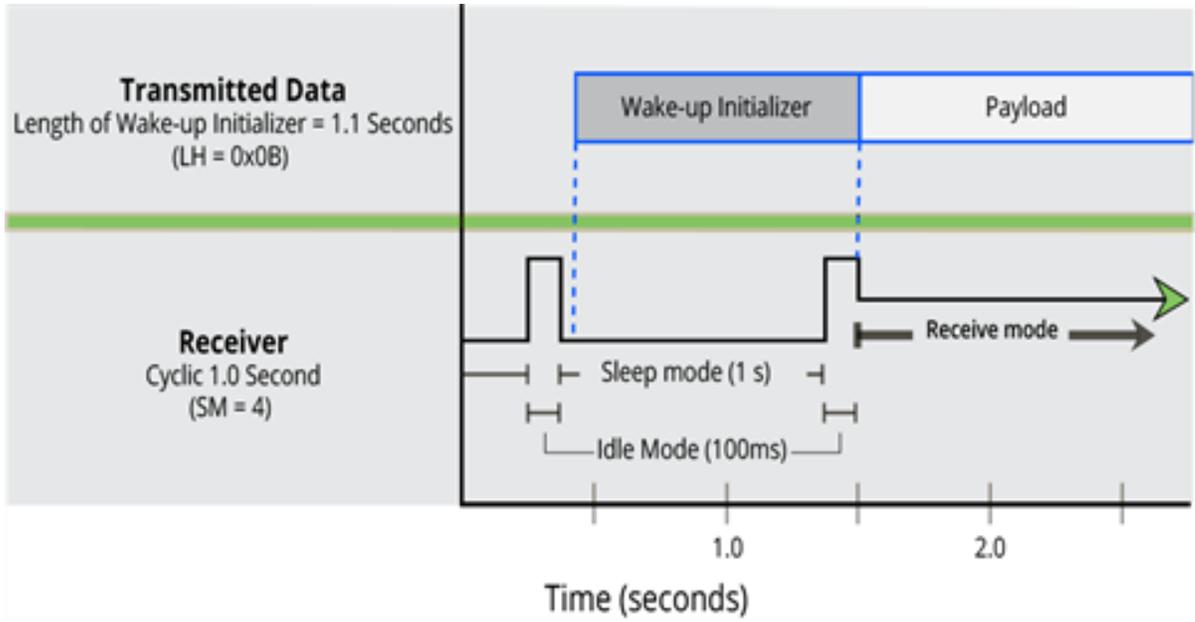
### Cyclic scanning

Each RF transmission consists of an RF initializer and payload. The RF initializer contains initialization information and all receiving devices must wake during the wake-up initializer portion of data transmission in order to synchronize with the transmitting device and receive the data.

The cyclic interval time defined by the **SM** (Sleep Mode) command must be shorter than the interval time defined by **LH** (Wake-up Initializer Timer) command.

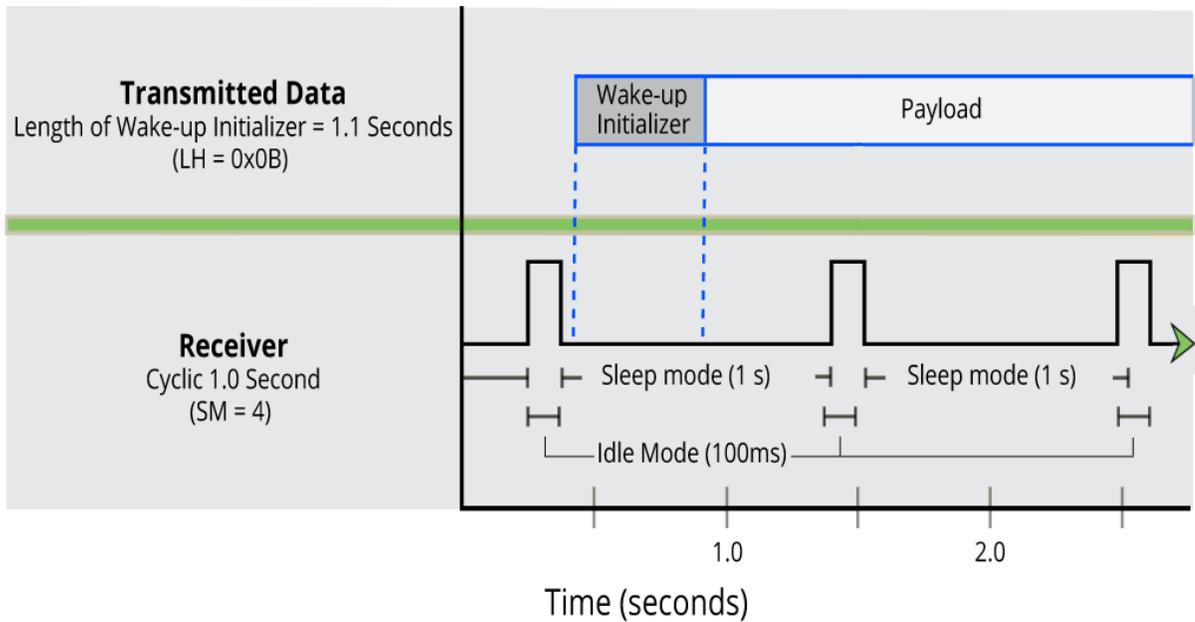
### Correct configuration (LH > SM)

In the following figure, the length of the wake-up initializer exceeds the time interval of Cyclic Sleep. The receiver is guaranteed to detect the wake-up initializer and receive the accompanying payload data.



**Incorrect configuration (LH < SM)**

Length of wake-up initializer is shorter than the time interval of Cyclic Sleep. This configuration is vulnerable to the receiver waking and missing the wake-up initializer (and therefore also the accompanying payload data).



## Operation

---



When operating at 1 W power output, observe a minimum separation distance of 6 ft (2 m) between devices. Transmitting in close proximity of other devices can damage the device's front end.

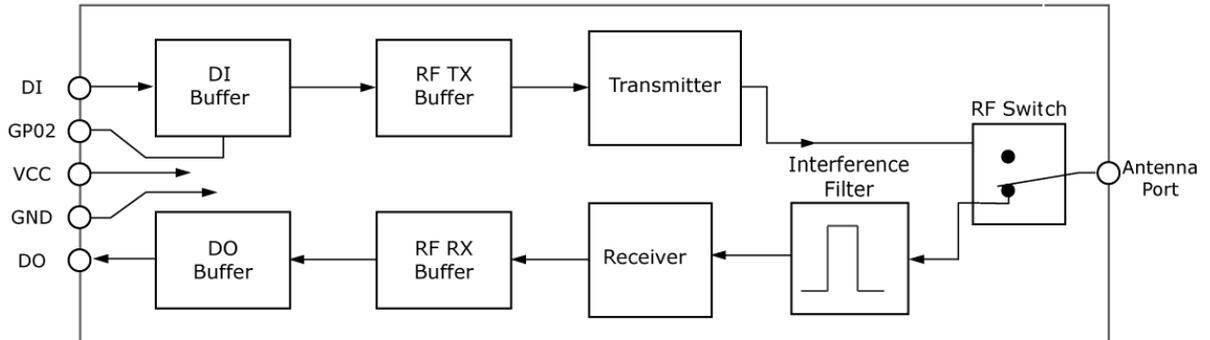
---

Serial interface .....	28
UART data flow .....	28
Serial data .....	28
Flow control .....	29



## Flow control

The  $\overline{\text{RTS}}$  and  $\overline{\text{CTS}}$  device pins provide  $\overline{\text{RTS}}$  and/or  $\overline{\text{CTS}}$  flow control.  $\overline{\text{CTS}}$  flow control signals the host to stop sending serial data to the device.  $\overline{\text{RTS}}$  flow control lets the host signal the device so it will not send the data in the serial transmit buffer out the UART. The following diagram shows the internal data flow, with the five most common pin signals.



The firmware has Hardware flow control ( $\overline{\text{CTS}}$ ) configured by default. You must configure  $\overline{\text{CTS}}$  flow control on the host side for it to work.

You must configure Software flow control (XON) on both the host and device side for it to work.

If you change the **CS** command from 0, then  $\overline{\text{CTS}}$  flow control will not work even if you have it configured on the host.

## Data In (DIN) buffer and flow control

When serial data enters the device through the DIN pin (pin 5), it stores the data in the DIN buffer until it can process the data.

When the firmware satisfies the **RB** and **RO** parameter thresholds, the device attempts to initialize an RF transmission. If the device is already receiving RF data, it stores the serial data in the device's DIN buffer.

The device creates and transmits data packets when it meets one of the following conditions:

1. The device does not receive any serial characters for the amount of time set with in the **RO** command; see [RO \(Packetization Timeout\)](#).
2. The device receives the maximum number of characters that fits in an RF packet.
3. The device receives the Command Mode sequence; see [Enter Command mode](#).

If the DIN buffer becomes full, you must implement hardware or software flow control in order to prevent overflow (loss of data between the host and the device). To eliminate the need for flow control:

1. Send messages that are smaller than the DIN buffer size. The size of the DIN buffer varies according to the packet size (**PK** parameter) and the parity setting (**NB** parameter) you use.
2. Interface at a lower baud rate (**BD** parameter) than the RF data rate of the firmware (**BR** parameter) of the firmware.

In the following situations, the DIN buffer may become full and overflow:

1. If you set the serial interface data rate higher than the RF data rate of the device, the device receives data from the host faster than it can transmit the data over-the-air.

2. If the device receives a continuous stream of RF data or if the device monitors data on a network, it places any serial data that arrives on the DIN pin (pin 5) in the DIN buffer. It transmits the data in the DIN buffer over-the-air when the device no longer detects RF data in the network.

### **Hardware flow control ( $\overline{\text{CTS}}$ )**

The firmware asserts  $\overline{\text{CTS}}$  before the DIN buffer is full so it has time to send the signal and the host has time to stop sending data.

When the DIN buffer is full, the firmware de-asserts  $\overline{\text{CTS}}$  (high) to signal the host to stop sending data; refer to [FT \(Flow Control Threshold\)](#) and [CS \(GP01 Configuration\)](#).

The firmware re-asserts  $\overline{\text{CTS}}$  after the DIN buffer has 34 bytes of memory available.

### **Hardware flow control ( $\overline{\text{RTS}}$ )**

If you enable  $\overline{\text{RTS}}$  for flow control (RT parameter = 2), the device will not send data out the DOUT buffer as long as the RTS pin (pin 10) is de-asserted.

### **Software flow control (XON/OFF)**

Use **FL** to enable XON/XOFF software flow control. This option only works with ASCII data.

## **Data Out (DO) buffer and flow control**

When a device receives RF data, the data enters the DOUT buffer and the device sends it out the serial port to a host device. Once the DOUT buffer reaches capacity, it loses any additional incoming RF data. The DOUT buffer stores at least 2.1 kB.

In the following situations, the DOUT buffer may become full and overflow:

1. If you set the interface data rate higher than the RF data rate of the device, the receiving device receives data from the transmitting device faster than it can send the data to the host.
2. If the host does not allow the device to transmit data out from the DOUT buffer because of being held off by hardware or software flow control.

### **Hardware flow control ( $\overline{\text{RTS}}$ )**

If you enable  $\overline{\text{RTS}}$  for flow control (RT Parameter = 2), data will not be sent out the DO Buffer as long as  $\overline{\text{RTS}}$  (pin 16) is de-asserted.

### **Software flow control (XOFF)**

You can enable XON/XOFF software flow control using the **FL** (Software Flow Control) Command. This option only works with ASCII data.

## **Configure the XTend vB RF Module**

---

Configure the device using XCTU .....32

## 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, go to:

<http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>

# Program the XTend vB RF Module

---

## Programming examples

For steps on sending AT commands to a device, refer to:

- [Send AT commands](#)
- [Exit Command mode](#)

For more information, refer to the XCTU online help at:

<https://docs.digi.com/display/XCTU/XCTU+Overview>

## Connect the device to a PC

The programming examples that follow require the installation of XCTU and a serial connection to a PC. Digi stocks connector boards to facilitate interfacing with a PC.

1. Download XCTU from the Digi website:  
<http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu#resources>
2. After the .exe file downloads to the PC, double-click the file to launch the XCTU Setup Wizard. Follow the steps in the wizard to completely install XCTU.
3. Mount the device to an interface board, then connect the assembly to a PC.
4. Launch XCTU and click the **Add devices** tab on the upper left corner of the screen.
5. Verify that the baud rate and parity settings of the Serial/USB port match those of the device.

---

**Note** Failure to enter Command mode is commonly due to baud rate mismatch. Ensure that the **Baud Rate:** setting on the Add radio device window matches the interface data rate of the device. By default, the BD parameter = 9600 b/s.

---

## Modify a device address

The following programming example shows you how to modify the device's destination address.

1. Once you add the device in XCTU, click on it in the **Radio Modules** pane to display the Configuration working mode. This mode shows most of the device's parameters that you can edit.
2. Scroll down in the Radio Configuration pane until you find the parameter you want to edit, in this case the DT (Destination Address) parameter, or use the search box and type DT. XCTU automatically scrolls to the selected parameter.
3. When you locate the parameter, change its value, for example to 1A0D. If you do not save the parameter, the color of the surrounding container is light green.

4. Click the **write** button to save the value to non-volatile memory; it is the pencil icon to the right of the parameter . If you change other parameters but have not saved them, you can use the **Write radio settings** button to save them. It is the white and blue pencil icon on the top of the configuration panel .

## Restore device defaults

The following programming example shows you how to restore a device's default parameters.

1. After establishing a connection between the device and a PC click the Configuration working mode tab of XCTU .
2. Click the **Load default firmware settings** button and agree to restore the default values. The button is the factory icon .
3. The restored parameters have a light green surrounding color, which means that they have been changed but not saved.
4. Click the **Write module settings** button  to save all of the parameters simultaneously.
5. All the parameters surrounding box must change to gray indicating that their values are now saved in the device's non-volatile memory.

## Send binary commands

### Example

Use XCTU's Serial Console tool to change the device's **DT** (Destination Address) parameter and save the new address to non-volatile memory.

This example requires XCTU and a serial connection to a PC.

To send binary commands:

1. Set the **RT** command to 1 to enable binary command programming; do this in Command mode or configure it through XCTU.
2. Drive pin 10 high to assert CMD by de-asserting the  $\overline{\text{RTS}}$  line in XCTU. The device enters Binary Command mode.
3. Send hexadecimal bytes (parameter bytes must be 2 bytes long). The next four lines are examples, not required values:  
 00 (Send binary command **DT**)  
 0D (Least significant byte of parameter bytes)  
 1A (Most significant byte of parameter bytes)  
 08 (Send binary command **WR**)
4. Drive pin 10 low to de-assert CMD. After you send the commands,  $\overline{\text{CTS}}$  (pin 9) de-asserts (driven low) temporarily. The device exits Binary Command mode.

The default flow control is **NONE**, so if you are using XCTU,  $\overline{\text{CTS}}$  is not an issue. However, you can still observe the behavior of the CTS line by monitoring the  $\overline{\text{CTS}}$  indicator in the terminal or console.

## Query binary commands

Example: use XCTU's Serial Console tool to query the device's **DT** (Destination Address) and **DB** (Received Signal strength) parameters. In order to query a parameter instead of setting it, you must logically OR the binary command byte with 0x80.

1. Set the **RT** command to 1 to enable binary command programming. To do this, you must either be in Command mode or use XCTU to configure the device.
2. Assert CMD by driving pin 29 high. To do this de-assert the  $\overline{\text{RTS}}$  line in XCTU.
3. Send hexadecimal bytes:  
80 (Binary command DT (0x00) ORed with 0x80)  
B6 (Binary command DB (0x36) ORed with 0x80)
4. Read the device's output for the parameter values of the two commands.
5. De-assert CMD by driving pin 29 low. The device exits Binary Command mode.

When querying commands in binary command mode, the output is the least significant byte followed by the most significant byte and is always represented in hexadecimal values.

## XTend vB RF Module commands

---

The following table lists the AT and binary commands in the XTend vB RF Module firmware and links to the description of the individual command.

By default, the device expects numerical values in hexadecimal since the default value of the CF (Number Base) Parameter is 1. Hexadecimal values are designated by the 0x prefix and decimal values by the d suffix.

AT command	Binary command
<a href="#">%V (Board Voltage)</a>	0x3B (59d)
<a href="#">AM (Auto-set MY)</a>	0x41 (65d)
<a href="#">AP (API Enable)</a>	--
<a href="#">AT (Guard Time After)</a>	0x05 (5d)
<a href="#">BD (Interface Data Rate)</a>	0x15 (21d)
<a href="#">BR (RF Data Rate)</a>	0x39 (57d)
<a href="#">BT (Guard Time Before)</a>	0x04 (4d)
<a href="#">CC (Command Sequence Character)</a>	0x13 (19d)
<a href="#">CD (GP02 Configuration)</a>	0x28 (40d)
<a href="#">CF (Number Base)</a>	--
<a href="#">CN (Exit Command Mode)</a>	0x09 (9d)
<a href="#">CS (GP01 Configuration)</a>	0x1F (31d)
<a href="#">CT (Command Mode Timeout)</a>	0x06 (6d)
<a href="#">DB (Received Signal Strength)</a>	0x36 (54d)
<a href="#">DT (Destination Address)</a>	0x00 (0d)
<a href="#">E0 (Echo Off)</a>	0x0A (10d)
<a href="#">E1 (Echo On)</a>	0x0B (11d)
<a href="#">ER (Receive Count Error)</a>	0x0F (15d)
<a href="#">FH (Force Wakeup Initializer)</a>	0x0D (13d)
<a href="#">FL (Software Flow Control)</a>	0x07 (7d)

AT command	Binary command
FS (Forced Synch Time)	0x3F (63d)
FT (Flow Control Threshold)	0x24 (36d)
GD (Receive Good Count)	0x10 (16d)
HP (Preamble ID)	0x11 (17d)
HS (Hardware Series)	--
HT (Time before Wake-up Initializer)	0x03 (3d)
HV (Hardware Version)	--
ID (Network ID)	0x27 (39d)
KY (AES Encryption Key)	0x43 (67d)
LH (Wakeup Initializer Timer)	0x0C (12d)
MD (RF Mode)	0x31 (49d)
MK (Address Mask)	0x12 (18d)
MT (Multi-transmit)	0x3E (62d)
MY (Source Address)	0x2A (42d)
NB (Parity)	0x23 (35d)
PB (Polling Begin Address)	0x45 (69d)
PD (Minimum Polling Delay)	0x47 (71d)
PE (Polling End Address)	0x46 (70d)
PK (Maximum RF Packet Size)	0x29 (41d)
PL (TX Power Level)	0x3A (58d)
PW (Pin Wakeup)	0x1D (29d)
RB (Packetization Threshold)	0x20 (32d)
RC (Ambient Power - Single Channel)	--
RE (Restore Defaults)	0x0E (14d)
RM (Ambient Power)	--
RN (Delay Slots)	0x19 (25d)
RO (Packetization Timeout)	0x21 (33d)
RP (RSSI PWM Timer)	0x22 (34d)
RR (Retries)	0x18 (24d)

AT command	Binary command
RT (GPI1 Configuration)	0x16 (22d)
SB (Stop Bits)	0x37 (55d)
SH (Serial Number High)	0x25 (37d)
SL (Serial Number Low)	0x26 (38d)
SM (Sleep Mode)	0x01 (1d)
ST (Time before Sleep)	0x02 (2d)
TP (Board Temperature)	0x38 (56d)
TR (Transmit Error Count)	0x1B (27d)
TT (Streaming Limit)	0x1A (26d)
TX (Transmit Only)	0x40 (64d)
VL (Firmware Version - Verbose)	--
VR (Firmware Version - Short)	0x14 (20d)
WA (Active Warning Numbers)	--
WN (Warning Data)	--
WR (Write)	0x08 (8d)
WS (Sticky Warning Numbers)	--

## Command mode options

The following commands are Command mode option commands.

### AT (Guard Time After)

Sets or reads the time-of-silence that follows the **CC** (Command Sequence Character) of the Command mode sequence (**BT + CC + AT**). By default, one second must elapse before and after the command sequence character.

The times-of-silence surrounding the Command Sequence Character prevent the device from inadvertently entering Command mode.

#### Binary command

0x05 (5 decimal)

#### Parameter range

#### Default

0xA (1 second)

#### Bytes returned

2

## BT (Guard Time Before)

Sets the DI pin silence time that must precede the Command Sequence Character (**CC** command) of the Command mode sequence. For more information about the Command mode sequence, see [Enter Command mode](#).

### Binary command

0x04 (4 decimal)

### Parameter range

### Default

0x0A (1 second)

### Bytes returned

2

## CC (Command Sequence Character)

Sets or reads the character the device uses between guard times of the AT Command mode sequence. The AT Command mode sequence causes the device to enter Command Mode (from Idle Mode).

### Binary command

0x13 (19 decimal)

### Parameter range

0x20 - 0x7F

### Default

0x2B (ASCII "+")

### Bytes returned

1

## CF (Number Base)

Sets or reads the command formatting setting.

The firmware always enters and reads the following commands in hex, no matter what the CF setting is:

**VR** (Firmware Version)

**HV** (Hardware Version)

**KY** (AES Encryption Key)

### Binary command

N/A

### Command type

Command mode options

**Parameter range**

0 - 2

Parameter	Configuration
0	Commands use the default number base; decimal commands may output units.
1	All commands are forced to unsigned, unit-less hex.
2	Commands use their default number base; no units are output.

**Default**

1

**Bytes returned**

1

**CN (Exit Command Mode)**

Makes the device exit Command mode.

**Binary command**

0x09 (9 decimal)

**Parameter range**

N/A

**Default**

N/A

**Bytes returned**

N/A

**CT (Command Mode Timeout)**

Set or read 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.

Use the **CN** (Exit Command mode) command to exit Command mode manually.

**Binary command**

0x06 (6 decimal)

**Parameter range**

**Default**

0xC8 (20 seconds)

**Bytes returned**

2

**E0 (Echo Off)**

Turns off the character echo in Command mode.

By default, echo is off.

**Binary command**

0x0A (10 decimal)

**Parameter range**

N/A

**Default**

N/A

**Bytes returned**

N/A

**E1 (Echo On)**

Enables character echo in Command mode. Each character that you type echoes back to the terminal when **E1** is active. **E0** (Echo Off) is the default.

**Binary command**

0x0B (11 decimal)

**Parameter range**

N/A

**Default**

N/A

**Bytes returned**

N/A

**Diagnostic commands**

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

**%V (Board Voltage)**

Reads the supply voltage to the module's VCC (pin 2).

The conversion of the hex value returned by **%V** to Volts is  $VAL/65536 = \text{Volts}$ .

Example:

2.8 VDC =  $2.8 * 65536 = 0x2CCCD$

3.3 VDC =  $3.3 * 65536 = 0x34CCD$

**Sample output**

3.27 V (when **CF** = 0)

345E3 (when **CF** = 1) <sup>1</sup>

3.27 (when **CF** = 2)

#### Binary command

0x3B (59 decimal)

#### Parameter range

[Read-only]:

0x2CCCA - 0x5BFFA (2.80 to 5.75 V)

#### Default

N/A

#### Bytes returned

4

## DB (Received Signal Strength)

This command reports the received signal strength of the last received RF data packet or APS acknowledgment. The **DB** command only indicates the signal strength of the last hop. It does not provide an accurate quality measurement for a multihop link. Set **DB** to 0 to clear it. The **DB** command value is measured in -dBm. For example, if **DB** returns 0x50, then the RSSI of the last packet received was -80 dBm.

#### Parameter range

0 - 0xFF

Observed ranges:

XBee-PRO - 0x1A - 0x58

XBee- 0x 1A - 0x5C

#### Default

N/A

## GD (Receive Good Count)

Sets or reads the number of RF packets with valid MAC headers that the device receives successfully on the RF interface. When the value reaches 0xFFFF, it stays there until you manually change the maximum count value or reset the device.

Its parameter value is reset to 0 after every device reset and is not non-volatile; the parameter value does not persist in the device's memory after a power-up sequence.

Pin, serial port or cyclic sleep modes do not reset the **GD** parameter.

#### Parameter range

0 - 0xFFFF

#### Default

0

---

<sup>1</sup>When **CF** = 1 (default), the firmware shows a hex integer that is equal to (voltage \* 65536d).

**Bytes returned**

2

**HV (Hardware Version)**

Reads the device's hardware version number.

**Binary command**

N/A

**Command type**

Diagnostics

**Parameter range**

[Read-only]: 0 - 0xFFFF

**Default**

N/A

**Bytes returned**

N/A

**RC (Ambient Power - Single Channel)**

Reads and reports the power level on a given channel.

**Sample output**-78 dBm (when **CF** = 0)4e (when **CF** = 1)-78 (when **CF** = 2)**Binary command**

N/A

**Parameter range**

[Read-only]: 0 - 0x31 [dBm]

**Default**

N/A

**Bytes returned**

1

**RE (Restore Defaults)**

Restore device parameters to factory defaults.

**RE** does not cause the device to store default values to non-volatile (persistent) memory. You must send the **WR** command prior to power-down or reset to save the default settings in the device's non-volatile memory.

**Binary command**

0x0E (14 decimal)

**Parameter range**

N/A

**Default**

N/A

**Bytes returned**

N/A

## RM (Ambient Power)

Reads and reports power levels on all channels. If you do not provide a parameter, the device scans the channels one time. If you do provide a parameter, the device scans the channels repeatedly for the number of seconds that the parameter calls for. The firmware reports the maximum power level seen for each channel (in other words, peak hold).

To implement a graphical spectrum analyzer, repeatedly send **RM** with no arguments and read the resulting 50 power levels. This is easiest to do when **CF** = 1 or 2.

Sample output when <b>CF</b> = 0:	Ch 0: -100 dBm
	Ch 1: -103 dBm
	...
	Ch 49: -99 dBm
Sample output when <b>CF</b> = 1: 64	64
	67
	...
	63
Sample output when <b>CF</b> = 2: 100	100
	-103
	...
	-99

### Binary command

N/A

### Command type

Diagnostics

### Parameter range

no parameter - 0x7D0

### Default

N/A

### Bytes returned

2

## RP (RSSI PWM Timer)

Enables a pulse-width modulated (PWM) output on the CONFIG /RSSI pin (pin 11 of the device). We calibrate the pin to show the difference between received signal strength and the sensitivity level of the device. PWM pulses vary from zero to 95 percent. Zero percent means the RF signal the device receives is at or below the device's sensitivity level.

The following table shows dB levels above sensitivity and PWM values. The total time period of the PWM output is 8.32 ms. PWM output consists of 40 steps, so the minimum step size is 0.208 ms.

dB above sensitivity	PWM percentage (high period / total period)
10	30%
20	45%
30	60%

A non-zero value defines the time that PWM output is active with the RSSI value of the last RF packet the device receives. After the set time when the device has not received RF packets, it sets the PWM output low (0 percent PWM) until the device receives another RF packet. It also sets PWM output low at power-up. A parameter value of 0xFF permanently enables PWM output and always reflects the value of the last received RF packet.

The PWM output and Config input share the CONFIG /RSSI pin. When the device is powered, the Config pin is an input. During the power-up sequence, if RP is a non-zero value, the firmware configures the Config pin as an output and sets it low until the device receives the first RF packet. With a non-zero RP parameter, the CONFIG pin is an input for RP ms after power up.

**Binary command**

0x22 (34 decimal)

**Parameter range**

0 - 0xFF [x 100 ms]

**Default**

0x20 (3.2 seconds)

**Bytes returned**

1

**SH (Serial Number High)**

**Binary command**

0x25 (37 decimal)

**Parameter range**

**Default**

**Bytes returned**

2

**SL (Serial Number Low)**

**Binary command**

0x26 (38 decimal)

**Parameter range**

0 - 0xFFFF [Read-only]

**Default****Bytes returned**

2

**TP (Board Temperature)**

The current module temperature in degrees Celsius in 8-bit two's complement format. For example 0x1A = 26°C, and 0xF6 = -10°C.

**Sample output**26 C when **CF** = 01A when **CF** = 126 when **CF** = 2**Binary command**

0x38 (56 decimal)

**Parameter range**

0 - 0x7F [Read-only]

**Default****Bytes returned**

1

**TR (Transmit Error Count)**

Reads the number of RF packets where retries expire without receiving an ACK (when **RR** > 0).

**Binary command**

0x1B (27 decimal)

**Parameter range**

0 - 0xFFFF

**Default**

0

**Bytes returned**

2

**VL (Firmware Version - Verbose)**

Reads the verbose firmware version of the device.

**Binary command**

N/A

**Parameter range**

Returns a string

**Default**

0

**Bytes returned**

2

**VR (Firmware Version - Short)**

Reads the firmware version on a device.

Firmware versions contain four significant digits: A.B.C.D. If B = 2, the device is programmed for operation in Australia only.

**Binary command**

0x14 (20 decimal)

**Parameter range**

[Read-only]: 0 - 0xFFFF

**Default**

N/A

**Bytes returned**

2

**WA (Active Warning Numbers)**

Reports the warning numbers of all active warnings, one warning number per line. It does not show further information and does not reset warning counts. For information on what the warning numbers mean, see [WN \(Warning Data\)](#).

**Sample output (indicates warnings 1 and 3 are currently active)**

```
1
3
OK
```

**Binary command**

N/A

**Command type**

Diagnostics

**Parameter range**

Returns a string: one warning number per line.

**Default**

N/A

**Bytes returned**

N/A

**WN (Warning Data)**

Reports the following data for all active and sticky warnings:

- Warning number and description
- Number of occurrences since the last **WN** or **WS** command
- Whether the warning is currently active

**WN** does not display warnings that are not currently active and have not been active since the last issuance of the **WN** or **WS** commands. **WN** resets all non-zero warning counts except for warnings that are presently active, which are set to 1.

**Sample output**

Warning 4: Over-temperature  
5 occurrences; presently inactive.

Warning #	Description
1	Under-voltage. This is caused if the supply voltage falls below the minimum threshold for the lowest power level (2.8 V). If/when the voltage rises above the threshold, the warning is deactivated. The device does not transmit below this voltage threshold.
2	Deprecated.
3	Under-temperature. This is caused if the temperature sensed by the device is less than -40° C. The device does not artificially limit operation while this warning is active, but device functionality is not guaranteed.
4	Over-temperature. This is caused if the temperature sensed by the device is greater than 105° C. The device does not allow transmission nor reception while this warning is active. The warning is deactivated when the temperature falls below 100° C.
5	Power reduced. This is caused if the transmit power has to be reduced from the level programmed by PL due to insufficient supply voltage. PL4: 30 dBm (1 Watt) power level requires 4.75 V or higher. PL3: 27 dBm (500 mW) power level requires 3.2 V or higher. PL2 - PL0: 21.5 dBm (100 mW) power levels require 2.8 V or higher.
6	Deprecated.

Warning #	Description
7	Default configuration parameters in flash. This is caused if user-modifiable parameters (i.e. those stored by WR) in flash are all the compiled-in default values. This is caused if the user configuration is found to be not present or invalid at power-up and there is no custom configuration, or if no user-modifiable parameters have been modified from the compiled-in defaults. Modification of one or more parameters without the subsequent WR to commit the changes to flash will not deactivate this warning, since it reflects the status of the parameters in flash. This warning does not reflect usage of the custom configuration defaults, only usage of the compiled-in defaults.
8	Default factory configuration parameters in flash. This is caused if the factory parameters in flash are all the default values. This is caused if the factory configuration is found to be not present or invalid at power-up, or if no factory parameters have been modified.
9	Watchdog reset occurred.
10	<b>PK</b> was reduced by <b>BR</b> .
11	<b>RB</b> was reduced by <b>PK</b> .
12	One or more parameters overridden due to conflict.

**Binary command**

N/A

**Command type**

Diagnostics

**Parameter range**

Returns a string

**Default**

N/A

**Bytes returned**

N/A

**WS (Sticky Warning Numbers)**

Reports warning numbers of all warnings active since the last use of **WS** or **WN**, including any warnings that are currently active. **WS** also resets all non-zero warning counts, except for warnings that are presently active, which are set to 1.

**Binary command**

N/A

**Command type**

Diagnostics

**Parameter range**

[Read-only]: 1 - 8

**Default**

N/A

**Bytes returned**

1

The following AT commands are firmware commands.

**HS (Hardware Series)**

Read the device's hardware series number.

**Parameter range**

N/A

**Default**

0x2A00 - set in the factory

**MAC/PHY commands**

The following AT commands are MAC/PHY commands.

**AM (Auto-set MY)**

Sets the **MY** (Source Address) parameter from the factory-set serial number of the device. The address consists of bits 29, 28 and 13-0 of the serial number, in that order.

Sending **AM** displays the address.

**Binary command****Parameter range**

N/A

**Default**

N/A

**Bytes returned**

N/A

**DT (Destination Address)**

Sets or reads the networking address of a device. The devices use three filtration layers:

- Vendor ID Number (**ID**)
- Channel (**HP**)
- Destination Address (**DT**)

**Binary command**

0x00 (0 decimal)

**Parameter range**

0 - 0xFFFF

**Default**

0

**Bytes returned**

2

**HP (Preamble ID)**

Set or read the device's hopping channel number. A channel is one of three layers of filtration available to the device.

In order for devices to communicate with each other, the devices must have the same channel number since each channel uses a different hopping sequence. Devices can use different channels to prevent devices in one network from listening to transmissions of another.

When a device receives a packet it checks **HP** before the network ID, as it is encoded in the preamble and the network ID is encoded in the MAC header.

**Binary command**

0x11 (17 decimal)

**Parameter range**

0 - 9

**Default**

0

**Bytes returned**

1

**ID (Network ID)**

Sets or reads the Vendor Identification Number (VID) of the device. Devices must have matching VIDs in order to communicate. If the device uses OEM network IDs, 0xFFFF uses the factory value.

**Binary command**

0x27 (39 decimal)

**Parameter range**

0x11 - 0x7FFF (user-settable)

0 - 0x10 and 0x8000 - 0xFFFF (factory-set)

**Default**

0x3332

N/A

**Bytes returned**

2

**MK (Address Mask)**

Sets or read the device's Address Mask.

All RF data packets contain the Destination Address of the transmitting (TX) device. When a device receives a packet, the TX device's Destination Address is logically combined bitwise (in other words, joined with AND) with the Address Mask of the receiving (RX) device. The resulting value must match the Destination Address or Address Mask of the RX device for the packet to be received and sent out the RX device's DO (Data Out) pin. If the combined value does not match the Destination Address or Address Mask of the RX device, it discards the packet.

The firmware treats all 0 values as irrelevant and ignores them. For more information, see [Addressing](#).

**Binary command**

0x12 (18 decimal)

**Parameter range**

0 - 0xFFFF

**Default**

0xFFFF

**Bytes returned**

2

**MT (Multi-transmit)**

Enables multiple transmissions of RF data packets. When you enable Multi-transmit mode (**MT** > 0), packets do not request an ACK from the receiving devices. **MT** takes precedence over **RR**, so if both **MT** and **RR** are non-zero, then a device sends **MT**+1 packets with no ACK requests.

When a receiving device receives a packet with remaining forced retransmissions, it calculates the length of the packet and inhibits transmission for the amount of time required for all retransmissions. From that time on, the device inserts a random number of delay slots between 0 and **RN** before allowing transmission from the receiving devices. This prevents all listening devices from transmitting at once upon conclusion of a multiple transmission event (when **RN** > 0).

---

**Note** The actual number of forced transmissions is the parameter value plus one. For example, if **MT** = 1, a device sends two transmissions of each packet.

---

For more information, see [Multi-transmit mode](#).

**Binary command**

0x3E (62d)

**Command type**

MAC/PHY

**Parameter range**

0 - 0xFF

**Default**

0 (no forced retransmissions)

**Bytes returned**

1

**MY (Source Address)**

Sets or reads the Source Address of a device.

For more information, see [DT \(Destination Address\)](#) and [Addressing](#).

**Binary command**

0x2A (42 decimal)

**Parameter range**

0 - 0xFFFF

**Default**

0xFFFF (Disabled - **DT** (Destination Address) parameter serves as both source and destination address).

**Bytes returned**

2

---

**Note** This command is only supported on S3B modules.

---

**RN (Delay Slots)**

Sets or reads the time delay that the transmitting device inserts before attempting to resend a packet. If the transmitting device fails to receive an acknowledgment after sending a packet, it inserts a random number of delay slots (ranging from 0 to (**RN** minus 1)) before attempting to resend the packet. Each delay slot is 5 ms when **BR** = 1 and 54 ms when **BR** = 0.

If two devices attempt to transmit at the same time, the random time delay after packet failure only allows one device to transmit the packet successfully, while the other device waits until the channel is available for RF transmission.

**RN** is only applicable if:

- You enable retries using the **RR** command, or
- You insert forced delays into a transmission using the **TT** command

**Binary command**

0x19 (25 decimal)

**Parameter range**

0 - 0xFF [38 ms delay slots]

**Default**

0 (no delay slots inserted)

**Bytes returned**

1

**RR (Retries)**

Sets or reads the maximum number of retries sent for a given RF packet. When you enable RR (RR > 0), it enables RF packet retries and ACKs.

After transmitting a packet, the transmitting device waits to receive an ACK from a receiving device. If it does not receive the ACK in the time that **RN** specifies, it transmits the original packet again. The transmitting device transmits the RF packet repeatedly until it receives an ACK or until it sends the packet **RR** times.

---

**Note** You must have retries enabled for all modules in the network for retries to work.

---

**Binary command**

0x18 (24 decimal)

**Parameter range**

0 - 0xFF

**Default****Bytes returned**

1

**TT (Streaming Limit)**

Sets or reads the limit on the number of bytes that a device can send before issuing a random delay.

If a device is sending a continuous stream of RF data, it inserts a delay that stops its transmission and gives other devices time to transmit once it sends **TT** bytes of data. The random delay it inserts lasts between 1 and **RN** + 1 delay slots .

You can use **TT** to simulate full-duplex behavior.

**Binary command**

0x1A (26 decimal)

**Parameter range****Default****Bytes returned**

2

**RF interfacing commands**

The following AT commands are RF interfacing commands.

## BR (RF Data Rate)

Sets and reads the device's RF data rate (the rate at which the device transmits and receives RF data over-the-air).

### Binary command

0x39 (57 decimal)

### Parameter range

0 - 1

Parameter	RF data rate
0	10 kb/s
1	125 kb/s

### Default

1

### Bytes returned

1

## FS (Forced Synch Time)

The **FS** command only applies to streaming data. Normally, only the first packet of a continuous stream contains the full RF initializer. The RF devices then remain synchronized for subsequent packets of the stream.

You can use this parameter to periodically force an RF initializer during such streaming. Any break in UART character reception that is long enough to drain the DI buffer and cause a pause in RF data transmission also causes the firmware to insert an RF initializer on the next transmission.

### Binary command

0x3F (63 decimal)

### Command type

RF interfacing

### Parameter range

0 - 0xFFFF

[x 10 milliseconds]

### Default

0

### Bytes returned

2

## MD (RF Mode)

**Binary command****Parameter range****Default**

0

**Bytes returned**

1

## PB (Polling Begin Address)

Sets or reads the device's Polling Begin Address, which is the first address polled when you enable Polling mode.

**Binary command**

0x45 (69 decimal)

**Command type**

RF interface

**Parameter range**

0 - 0xFFFF

**Default**

0

**Bytes returned**

2

## PD (Minimum Polling Delay)

Sets or reads the Polling Delay (Base, **MD** = 3) or Polling Timeout (Remote, **MD** = 4).

Polling Delay (Base) is the time between polling cycles. The Polling Base starts the polling cycle after sending the first poll. After the polling cycle completes, the timer restarts.

Polling Timeout (Remote) is the amount of time the remote device holds data from the serial port before discarding it. The device transmits data entered within the **PD** time of the poll and does not discard it.

**Binary command**

0x47 (71 decimal)

**Command type**

RF interface

**Parameter range**

0 - 0xFFFF (Base: [x 1ms], Remote: [x 10ms])

**Default**

0x64

**Bytes returned**

2

**PE (Polling End Address)**

Sets or reads the device's Polling End Address; which is the last address polled when you enable Polling mode.

**Binary command**

0x46 (70 decimal)

**Command type**

RF interface

**Parameter range**

0 - 0xFFFF

**Default**

0

**Bytes returned**

2

**PK (Maximum RF Packet Size)**

Sets or reads the maximum size of RF packets that a device in Transparent operating mode (**AP** = 0) transmits. You can use the maximum packet size along with the **RB** and **RO** parameters to implicitly set the channel dwell time.

Changes to the **PK** parameter may have a secondary effect on the **RB** (Packetization Threshold) parameter. **RB** must always be less than or equal to **PK**. If you change **PK** to a value that is less than the current value of **RB**, the **RB** value lowers to be equal to PK.

**Binary command**

0x29 (41 decimal)

**Parameter range****Default****Bytes returned**

2

**PL (TX Power Level)**

Sets or reads the power level at which the device transmits conducted power.

**Binary command**

**Parameter range**

0 - 4

Parameter	Configuration
PL0	21.5 dBm
PL1	
PL2	
PL3	27 dBm
PL4	30 dBm (1 Watt)

**Default**

4

**Bytes returned**

1

**TX (Transmit Only)**

Sets or reads the transmit or receive behaviors of the device. Setting a device to TX-only (**TX** = 1) may reduce latency because the you can not limit the transmitting device to receiving data from other devices.

**Binary command**

0x40 (64d)

**Command type**

RF Interfacing

**Parameter range**

0 - 1

Parameter	Description
0	TX and RX
1	TX only

**Default**

0

**Bytes returned**

1

## Security commands

The following AT commands are security commands.

### KY (AES Encryption Key)

Sets the 256-bit Advanced Encryption Standard (AES) key for encrypting or decrypting data. Once set, you cannot read the key cannot out of the device by any means. The firmware encrypts the entire payload of the packet using the key and computes the CRC across the ciphertext. When you enable encryption, each packet carries an additional 16 bytes to convey the random cipher-block chaining (CBC) Initialization Vector (IV) to the receiver(s). Set 256-bit key (64 hex digits) on multiple devices for encrypted RF communication. Set to **0** to disable encryption. Reading the parameter returns a **0** (encryption disabled) or **1** (enabled). The key cannot be read for security reasons.

A device with the wrong key (or no key) receives encrypted data, but the data driven out the serial port is meaningless. Likewise, a device with a key receives unencrypted data sent from a device without a key, but the output is meaningless. Because it uses CBC mode, repetitive data appears differently in different transmissions due to the randomly-generated IV.

---

**Note** For international (non-U.S.) variants of XTC devices, the encryption key is 128-bit AES. The command operates the same except the key length is 16 bytes rather than 32 bytes. This pertains to part numbers ending with 128, no matter which firmware version is loaded. This also pertains to the Australia version of firmware 22xx.

---

#### Binary command

0x43 (67d)

#### Command type

Security

#### Parameter range

0 - (64 hex digits all set to 'F')

#### Default

0 (disabled)

#### Bytes returned

2

## Serial interfacing commands

The following AT commands are serial interfacing commands.

### AP (API Enable)

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.

Enable API Mode. The device ignores this command when using SPI. API mode 1 is always used.

**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 <b>RO</b> 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. The device passes XON (0x11), XOFF (0x13), Escape (0x7D), and start delimiter 0x7E as data.

Parameter	Description
0	API disabled (operate in Transparent mode)
1	API enabled
2	API enabled (with escaped control characters)

**Default**

0

**Bytes returned**

1

**BD (Interface Data Rate)**

Sets and reads the serial interface data rate (baud rate) between the device and the host. The baud rate is the rate that the host sends serial data to the device.

When you make an update to the interface data rate, the change does not take effect until the host issues the **CN** command and the device returns the **OK** response.

The **BD** parameter does not affect the RF data rate. If you set the interface data rate higher than the RF data rate, you may need to implement a flow control configuration.

**Non-standard interface data rates**

The firmware interprets any value within 0x4B0 - 0x2580 and 0x4B00 - 0x1C9468 as an actual baud rate. When the host sends a value above 0x4B0, the firmware stores the closest interface data rate represented by the number in the **BD** register. For example, to set a rate of 19200 b/s, send the following command line: **ATBD4B00**.

---

**Note** When using XCTU, you can only set and read non-standard interface data rates using the XCTU Serial Console tool. You cannot access non-standard rates through the configuration section of XCTU.

---

**Note** The device does not support nonstandard baud rates between 9601 and 19199 baud. If you attempt to set baud rates in this range, it will return an error.

---

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.

\* The S3 hardware variant is a legacy design that is obsolete. New and old designs should use the S3B hardware variant.

**Binary command**

0x15 (21 decimal)

**Parameter ranges**

Parameter	Configuration (b/s)
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600

**Default**

**Bytes returned**

4

**CD (GP02 Configuration)**

Selects or reads the behavior of the GPO2 line (pin 3).

**Binary command**

0x28 (40 decimal)

**Parameter range**

0 - 4

Parameter	Configuration
0	RX LED
1	Static high
2	Static low
3	Reserved
4	RX LED (valid address only)

**Default**

2

**Bytes returned**

1

**CS (GP01 Configuration)**

Sets or reads the behavior of the GP01 line (pin 9). This output can provide RS-232 flow control and controls the TX enable signal for RS-485 or RS-422 operations.

By default, GP01 provides RS-232 Clear-to-Send (CTS) flow control.

**Binary command**

0x1F (31 decimal)

**Parameter range**

0 - 4

Parameter	Configuration
0	RS-232 $\overline{\text{CTS}}$ flow control
1	RS-485 TX enable low
2	Static high
3	RS-485 TX enable high
4	Static low

**Default**

0

**Bytes returned**

1

**FL (Software Flow Control)**

The XON character used is 0x11 (17 decimal).

The XOFF character used is 0x13 (19 decimal).

**Binary command**

0x07 (7 decimal)

**Parameter range**

0 - 1

**Default**

0

**Bytes returned**

1

**FT (Flow Control Threshold)**

Sets or reads the flow control threshold.

De-assert  $\overline{\text{CTS}}$  when the number of bytes specified by the **FT** parameter are in the DIN buffer. Re-assert CTS when less than FT - 16 bytes are in the UART receive buffer.

**Binary command**

0x24 (36 decimal)

**Parameter range**

0x11 - 0xC00 [bytes]

**Default**

0xBBF (DI buffer size minus 0x11)

**Bytes returned**

2

**NB (Parity)**

Set or read the parity settings for UART communications.

**Parameter range**

Parameter	Configuration
0	8-bit (no parity )
1	8-bit even
2	8-bit odd
3	8-bit mark
4	8-bit space

**Default**

0

**Bytes returned**

1

**RB (Packetization Threshold)**

Sets or reads the character threshold value.

RF transmission begins after a device receives data in the DIN buffer and meets either of the following criteria:

- The UART receives **RB** characters
- The UART receive lines detect **RO** character times of silence after receiving at least 1 byte of data

If a device lowers **PK** below the value of **RB**, **RB** is automatically lowered to match the PK value.

If **RO** = 0, the device must receive **RB** bytes before beginning transmission.

**RB** and **RO** criteria only apply to the first packet of a multi-packet transmission. If data remains in the DIN buffer after the first packet, transmissions continue in a streaming manner until there is no data left in the DIN buffer.

#### Binary command

0x20 (32 decimal)

#### Parameter range

0 - **PK** parameter value  
(up to 0x800 bytes)

#### Default

0x800 (2048 bytes)

#### Bytes returned

2

## RO (Packetization Timeout)

Set or read the number of character times of inter-character silence required before transmission begins. For information on how **RO** works with the **RB** command, see [RB \(Packetization Threshold\)](#).

#### When RO is the transmission-beginning criteria:

The actual time between the reception of the last character from the UART and the beginning of RF transmission is at least 800  $\mu$ sec longer than the actual **RO** time to allow for transmission setup. It is also subject to 100-200  $\mu$ sec of additional uncertainty, which could be significant for small values of **RO** at high UART bit rates.

The firmware calculates the correct UART character time (10, 11, or 12 bits) based on the following criteria:

- 1 start bit
- 8 data bits
- 0 or 1 parity bit (as determined by the **NB** command)
- 1 or 2 stop bits (as determined by **SB** command)

#### Binary command

0x21 (33 decimal)

#### Parameter range

0 - 0x53E2 [x UART character times]

#### Default

3

**Bytes returned**

2

**RT (GPI1 Configuration)**

Sets or reads the behavior of the GPI1 pin (pin 10) of the device. You can configure the pin to enable Binary Command mode or RTS flow control.

**Binary command**

0x16 (22 decimal)

**Parameter range**

0 - 2

Parameter	Configuration
0	Disabled
1	Binary Command enable
2	RTS flow control enable

**Default**

0 (disabled)

**Bytes returned**

1

**SB (Stop Bits)**

Sets or reads the number of stop bits in the data packet.

**Binary command**

0x37 (55 decimal)

0x36 (54 decimal)

**Parameter range**

0 - 1

Parameter	Configuration
0	One stop bit
1	Two stop bits

**Default**

0

**Bytes returned**

1

## Sleep commands

### FH (Force Wakeup Initializer)

Forces the device to send a wake-up initializer on the next transmission.

Only use **FH** with cyclic sleep modes active on remote devices.

You do not need to issue the **WR** (Write) command with **FH**.

#### Binary command

0x0D (13 decimal)

#### Parameter range

N/A

#### Default

N/A

#### Bytes returned

N/A

### HT (Time before Wake-up Initializer)

Sets or reads the time of inactivity (no serial or RF data is sent or received) before a transmitting (TX) RF device sends a wake-up initializer. The main purpose of this command is to prevent devices from sending the Long Header with every data packet. For more information on long headers, see [LH \(Wakeup Initializer Timer\)](#).

For RX devices operating in Cyclic Sleep mode (**SM** = 4-8), set **HT** to be shorter than the **ST** command.

The TX device sends a wake-up initializer, which instructs all receiving (RX) devices to remain awake to receive RF data.

From the perspective of the RX device: after **HT** time elapses and the inactivity timeout (**ST** command) is met, the RX device goes into cyclic sleep. In cyclic sleep, the RX device wakes once per sleep interval (**SM** command) to check for a wake-up initializer. When it detects a wake-up initializer, the device stays awake to receive data. The wake-up initializer must be longer than the cyclic sleep interval to ensure that sleeping devices detect incoming data.

When **HT** time elapses, the TX device knows it needs to send a wake-up initializer for all RX devices to remain awake and receive the next transmission.

#### Binary command

0x03 (3 decimal)

#### Parameter range

#### Default

0xFFFF (wake-up initializer will not be sent)

#### Bytes returned

2

## LH (Wakeup Initializer Timer)

Sets or reads the duration of time during which the wake-up initializer is sent. When receiving devices are in Cyclic Sleep Mode, they power-down after a period of inactivity as specified by the **ST** parameter and will periodically wake and listen for data transmissions. In order for the receiving devices to remain awake, they must detect ~35 ms of the wake-up initializer.

You must use **LH** whenever a receiving device is operating in Cyclic Sleep mode. The wake-up initializer time must be longer than the cyclic sleep time, which is set by the **SM** (Sleep Mode) parameter. If the wake-up initializer time is less than the Cyclic Sleep interval, the connection is at risk of missing the wake-up initializer transmission.

### Binary command

0x0C (12 decimal)

### Parameter range

0 - 0xFF [x100 milliseconds]

### Default

1

### Bytes returned

1

## PW (Pin Wakeup)

Enables or disables the sleep pin.

Under normal operation, a device in Cyclic Sleep mode cycles from an active state to a low-power state at regular intervals until it is ready to receive data. If you set **PW** to 1, you can use the SLEEP pin (pin 26) to wake the device from Cyclic Sleep. When you de-assert (low) the SLEEP pin, the device is operational and will not go into Cyclic Sleep.

Once you assert the SLEEP pin, the device remains active for the period of time specified by the **ST** parameter and returns to Cyclic Sleep mode if no data is ready to transmit. **PW** is only valid if Cyclic Sleep is enabled.

### Binary command

0x1D (29 decimal)

### Parameter range

0 - 1

Parameter	Configuration
0	Disabled
1	Enabled

### Default

0

### Bytes returned

1

## SM (Sleep Mode)

### Binary command

0x01

### Parameter range

### Default

0

### Bytes returned

1

## ST (Time before Sleep)

You can only use this command if you use **SM** to select Cyclic Sleep or Serial Port Sleep mode settings; see [SM \(Sleep Mode\)](#).

### Binary command

0x02 (2 decimal)

### Parameter range

### Default

0x64 (10 seconds)

### Bytes returned

2

## 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.

If you make changes without writing them to non-volatile memory, the device reverts to previously saved parameters the next time it is powered on.

If the non-volatile user configuration is not correct, **WR** will re-attempt up to three times. If all three attempts fail, the command returns an ERROR alert.

### Binary command

0x08

### Command type

Special

### Parameter range

N/A

**Default**

N/A

**Bytes returned**

N/A

## API operation

---

API mode overview .....	72
XTend vB RF ModuleAPI frame overview .....	76

## API mode overview

By default, the XTend vB RF Module acts as a serial line replacement (Transparent operation), it queues all UART data that it receive through the DI pin for RF transmission. When the device receives an RF packet, it sends the data out the DO pin with no additional information.

The following behaviors are inherent to Transparent operation:

- If device parameter registers are to be set or queried, a special operation is required for transitioning the device into Command Mode; refer to [Enter Command mode](#).
- In point-to-multipoint systems, the host application must send eXTend vBa if the receiving device(s) need to distinguish between data coming from different remotes.

API operating mode is an alternative to transparent mode. 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 to control the destination of individual data packets or when you need to know which node a data packet was sent from. The device communicates UART data in packets, also known as API frames. This mode allows for structured communications with serial devices. 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 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:

1. **AP = 0:** Transparent operating mode, UART serial line replacement with API modes disabled. This is the default option.
2. **AP = 1:** API operation.
3. **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](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

#### 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 type	Frame data								Checksum
				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.

The XBee modules support the following API frames:

API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
ZigBee Transmit Request	0x10
Explicit Addressing ZigBee Command Frame	0x11
Remote Command Request	0x17
Create Source Route	0x21
AT Command Response	0x88
Modem Status	0x8A
ZigBee Transmit Status	0x8B
ZigBee Receive Packet (AO=0)	0x90
ZigBee Explicit Rx Indicator (AO=1)	0x91
ZigBee IO Data Sample Rx Indicator	0x92
XBee Sensor Read Indicator (AO=0)	0x94
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97
Extended Modem Status	0x98
Over-the-Air Firmware Update Status	0xA0
Route Record Indicator	0xA1
Many-to-One Route Request Indicator	0xA3

## Checksum

### 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 will equal 0xFF.

### Escaped characters in API frames

If operating in API mode with escaped characters (**AP** parameter = 2), when you send or receive an API frame, you must escape (flag) specific data values so they do not interfere with data frame sequencing. In API operating mode with escaped characters, you must escape the following data bytes:

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

API operating mode with escaped characters guarantees that all the 0x7E bytes a device receives are start delimiters: this character cannot be part of any of the other frame fields (length, data, or checksum) since you must escape it.

To escape a character:

1. Insert 0x7D, the escape character.
2. Append it with the byte you want to escape, XORed with 0x20.

In API operating 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.

#### Example: escape an API frame

To express the following API non-escaped frame in API operating mode with escaped characters:

Start delimiter	Length	Frame type	Frame Data	Checksum
			Data	
7E	00 0F	17	01 00 13 A2 00 40 AD 14 2E FF FE 02 4E 49 6D	

You must escape the 0x13 byte:

1. Insert a 0x7D.
2. XOR byte 0x13 with 0x20:  $13 \oplus 20 = 33$

The following figure shows the resulting frame. Note that the length and checksum are the same as the non-escaped frame.

Start delimiter	Length	Frame type	Frame Data	Checksum
			Data	
7E	00 0F 17	17	01 00 7D 33 A2 00 40 AD 14 2E FF FE 02 4E 49 6D	

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

## XTend vB RF Module API frame overview

The device sends multi-byte values in big-endian format. The devices support the following API frames:

API frame name	API ID
Transmit Request	0x01
RF Module Status	0x8A
Transmit Status	0x89
Receive Packet	0x81

## RF Module 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 0x01 = Watchdog timer reset

### Example

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

Frame fields	Offset	Example
Start Delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x02
Frame Type	3	0x8A
Status	4	0x00
Checksum	5	0x75

## Transmit Request: 16-bit address frame - 0x01

### Description

This frame causes the device to send data as an RF packet to a specific destination.

### 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. Setting Frame ID to 0 disables the response frame.
Destination address	5-6	MSB first, LSB last. Broadcast = 0xFFFF.
Options	7	0 = standard. 1 = disable ACK.
RF data	8-n	Up to 2048 bytes per packet. The payload size is limited by the PK command.

### Example

The following example shows how to send a transmission to a device with destination address 0x5642, and payload "TxData0A".

Frame fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x0D
Frame type	3	0x01
Frame ID	4	0x01
Destination address	MSB 5	0x56
	LSB 6	0x42
Options	7	0x00

Frame fields	Offset	Example
RF data	8	0x54
	9	0x78
	10	0x44
	11	0x61
	12	0x74
	13	0x61
	14	0x30
	15	0x41
Checksum	16	0xAE

## Transmit Status frame - 0x89

### Description

When a TX Request is completed, the device sends a TX Status message. This message will indicate if the packet was 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 data frame for the host to correlate with a subsequent ACK. Setting Frame ID to 0 disables the response frame.
Status	5	0 = success. 1 = all retries expired and no ACK received. 3 = a packet is purged due to a Polled Remote not receiving a poll.

### Example

In the following example, the destination device reports that a unicast data transmission was successful using a frame ID of 0x47.

Frame fields	Offset	Example
Start Delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x03
Frame type	3	0x89
Frame ID	4	0x47
Options	5	0x00
Checksum	6	0x2F

## 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 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 = ACK Bit 1 = indicate broadcast bits 2-7 = reserved
RF data	8-n	Up to 2048 bytes per packet.

### Example

In the following example, a device with a source address of 0xA35E sent a unicast data transmission to a remote device with a payload of "RxData". The receiving device would send the following frame out its UART:

Frame data fields	Offset	Example
Start Delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x0B
Frame type	3	0x81
Source address	MSB 4	0xA3
	LSB5	0x5E
RSSI	6	0x5D
Options	7	0x01

Frame data fields	Offset	Example
RF data	8	0x52
	9	0x78
	10	0x44
	11	0x61
	12	0x74
	13	0x61
Checksum	14	0xDB

## Certifications

---

FCC (United States) .....	84
Industry Canada (IC) .....	91
ACMA (Australia) .....	92

## FCC (United States)

These 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.

In order to operate under Digi's FCC Certification, integrators must comply with the following regulations:

1. The integrator must ensure that the text provided with this device (in the labeling requirements section that follows) is placed on the outside of the final product and within the final product operation manual.
2. The device may only be used with antennas that have been tested and approved for use with this device; refer to [FCC antenna certifications](#).

### OEM labeling requirements

The following text is the required FCC label for OEM products containing the XTend vB RF Module:

Contains FCC ID: MCQ-XBPSX

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

### FCC notices

**IMPORTANT:** These 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:** Integrators must test final product to comply with unintentional radiators (FCC sections 15.107 & 15.109) before declaring compliance of their final product to Part 15 of the FCC rules.

**IMPORTANT:** These RF modules have 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 antenna certifications

---



**WARNING!** This device has been tested with the antennas listed in the tables of this section. When integrated into products, fixed antennas require installation preventing end users from replacing them with non-approved antennas. Antennas not listed in the tables must be tested to comply with FCC Section 15.203 (unique antenna connectors) and Section 15.247 (emissions).

---

### ***Fixed base station and mobile applications***

Digi devices are pre-FCC approved for use in fixed base station and mobile applications. When the antenna is mounted at least 34 cm from nearby persons, the application is considered a mobile application.

### ***Portable applications and SAR testing***

When the antenna is mounted closer than 34 cm to nearby persons, then the application is considered "portable" and requires an additional test be performed on the final product. This test is called Specific Absorption Rate (SAR) testing and measures the emissions from the device and how they affect the person.

### ***RF exposure statement***

This statement must be included as a CAUTION statement in integrator product manuals.

---



**WARNING!** This equipment is approved only for mobile and base station transmitting devices. Antenna(s) used for this transmitter must be installed to provide a separation distance of at least 34 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.

---

## Antenna options

The following tables cover the antennas that are approved for use with the XTend vB RF Module. If applicable, the tables show the required cable loss between the device and the antenna.

Digi does not carry all of these antenna variants. Contact Digi Sales for available antennas.

### ***Dipole antennas***

All antenna part numbers followed by an asterisk (\*) are not available from Digi. Consult with an antenna manufacturer for an equivalent option.

Part Number	Type	Connector	Gain	Application
A09-HSM-7	Straight half-wave	RPSMA	2.1 dBi	Fixed / Mobile
A09-HASM-675	Articulated half-wave	RPSMA	2.1 dBi	Fixed / Mobile
A09-HABMM-P5I	Swivel half wave with 5" pigtail	MMCX	2.1 dBi	Fixed / Mobile
A09-HBMM-P5I	Straight half-wave with 6" pigtail	MMCX	2.1 dBi	Fixed / Mobile
A09-HASM-7	Articulated half-wave	RPSMA	2.1 dBi	Fixed
A09-HRSM*	Right angle half-wave	RPSMA	2.1 dBi	Fixed
A09-HG*	Glass mounted half-wave	RPSMA	2.1 dBi	Fixed
A09-HATM*	Articulated half-wave	RPTNC	2.1 dBi	Fixed
A09-H*	Half-wave dipole	RPSMA	2.1 dBi	Fixed

### ***Yagi antennas***

All antenna part numbers followed by an asterisk (\*) are not available from Digi. Consult with an antenna manufacturer for an equivalent option.

Part Number	Type	Gain	Connector	Required Antenna Cable Loss	Application
A09-Y6NF*	2 element Yagi	6.1 dBi	N	2.0 dB	Fixed/Mobile
A09-Y7NF*	3 element Yagi	7.1 dBi	N	3.0 dB	Fixed/Mobile
A09-Y8NF	4 element Yagi	8.1 dBi	N	4.0 dB	Fixed/Mobile
A09-Y9NF*	4 element Yagi	9.1 dBi	N	5.0 dB	Fixed/Mobile
A09-Y10NF*	5 element Yagi	10.1 dBi	N	6.0 dB	Fixed/Mobile
A09-Y11NF	6 element Yagi	11.1 dBi	N	7.0 dB	Fixed/Mobile
A09-Y12NF*	7 element Yagi	12.1 dBi	N	8.0 dB	Fixed/Mobile
A09-Y13NF*	9 element Yagi	13.1 dBi	N	9.0 dB	Fixed/Mobile
A09-Y14NF*	14 element Yagi	14.0 dBi	N	9.9 dB	Fixed/Mobile
A09-Y6TM*	2 element Yagi	6.1 dBi	RPTNC	2.0 dB	Fixed/Mobile
A09-Y7TM*	3 element Yagi	7.1 dBi	RPTNC	3.0 dB	Fixed/Mobile
A09-Y8TM*	4 element Yagi	8.1 dBi	RPTNC	4.0 dB	Fixed/Mobile
A09-Y9TM*	4 element Yagi	9.1 dBi	RPTNC	5.0 dB	Fixed/Mobile
A09-Y10TM-P10I	5 element Yagi	10.1 dBi	RPTNC	6.0 dB	Fixed/Mobile
A09-Y11TM*	6 element Yagi	11.1 dBi	RPTNC	7.0 dB	Fixed/Mobile
A09-Y12TM*	7 element Yagi	12.1 dBi	RPTNC	8.0 dB	Fixed/Mobile
A09-Y13TM*	9 element Yagi	13.1 dBi	RPTNC	9.0 dB	Fixed/Mobile
A09-Y14TM*	14 element Yagi	14.0 dBi	RPTNC	9.9 dB	Fixed/Mobile

### **Omni-directional base station antennas**

All antenna part numbers followed by an asterisk (\*) are not available from Digi. Consult with an antenna manufacturer for an equivalent option.

Part Number	Type	Gain	Connector	Required Antenna Cable Loss	Application
A09-F0NF*	Fiberglass Base Station	0 dBi	N	-	Fixed
A09-F1NF*	Fiberglass Base Station	1.0 dBi	N	-	Fixed
A09-F2NF-M	Fiberglass Base Station	2.1 dBi	N	-	Fixed
A09-F3NF*	Fiberglass Base Station	3.1 dBi	N	-	Fixed
A09-F4NF*	Fiberglass Base Station	4.1 dBi	N	-	Fixed
A09-F5NF-M	Fiberglass Base Station	5.1 dBi	N	-	Fixed
A09-F6NF*	Fiberglass Base Station	6.1 dBi	N	0.9 dB	Fixed
A09-F7NF*	Fiberglass Base Station	7.1 dBi	N	1.9 dB	Fixed
A09-F8NF-M	Fiberglass Base Station	8.1 dBi	N	2.9 dB	Fixed
A09-F0SM*	Fiberglass Base Station	0 dBi	RPSMA	-	Fixed
A09-F1SM*	Fiberglass Base Station	1.0 dBi	RPSMA	-	Fixed
A09-F2SM*	Fiberglass Base Station	2.1 dBi	RPSMA	-	Fixed
A09-F3SM*	Fiberglass Base Station	3.1 dBi	RPSMA	-	Fixed
A09-F4SM*	Fiberglass Base Station	4.1 dBi	RPSMA	-	Fixed
A09-F5SM*	Fiberglass Base Station	5.1 dBi	RPSMA	-	Fixed
A09-F6SM*	Fiberglass Base Station	6.1 dBi	RPSMA	0.9 dB	Fixed
A09-F7SM*	Fiberglass Base Station	7.1 dBi	RPSMA	1.9 dB	Fixed
A09-F8SM*	Fiberglass Base Station	8.1 dBi	RPSMA	2.9 dB	Fixed
A09-F0TM*	Fiberglass Base Station	0 dBi	RPTNC	-	Fixed
A09-F1TM*	Fiberglass Base Station	1.0 dBi	RPTNC	-	Fixed
A09-F2TM*	Fiberglass Base Station	2.1 dBi	RPTNC	-	Fixed

Part Number	Type	Gain	Connector	Required Antenna Cable Loss	Application
A09-F3TM*	Fiberglass Base Station	3.1 dBi	RPTNC	-	Fixed
A09-F4TM*	Fiberglass Base Station	4.1 dBi	RPTNC	-	Fixed
A09-F5TM*	Fiberglass Base Station	5.1 dBi	RPTNC	-	Fixed
A09-F6TM*	Fiberglass Base Station	6.1 dBi	RPTNC	0.9 dB	Fixed
A09-F7TM*	Fiberglass Base Station	7.1 dBi	RPTNC	1.9 dB	Fixed
A09-F8TM*	Fiberglass Base Station	8.1 dBi	RPTNC	2.9 dB	Fixed
A09-W7*	Wire Base Station	7.1 dBi	RPN	1.9 dB	Fixed
A09-W7SM*	Wire Base Station	7.1 dBi	RPSMA	1.9 dB	Fixed
A09-W7TM*	Wire Base Station	7.1 dBi	RPTNC	1.9 dB	Fixed

### ***Dome antennas***

All antenna part numbers followed by an asterisk (\*) are not available from Digi. Consult with an antenna manufacturer for an equivalent option.

Part Number	Type	Gain	Connector	Required Antenna Cable Loss	Application
A09-D3PNF*	Omnidirectional permanent mount	3.0 dBi	N	0.4 dB	Fixed/Mobile
A09-D3NF*	Omnidirectional magnetic mount	3.0 dBi	N	0.4 dB	Fixed/Mobile
A09-D3PTM*	Omnidirectional permanent mount	3.0 dBi	RPTNC	0.4 dB	Fixed/Mobile
A09-D3PSM*	Omnidirectional permanent mount	3.0 dBi	RPSMA	0.4 dB	Fixed/Mobile

### ***Monopole antennas***

All antenna part numbers followed by an asterisk (\*) are not available from Digi. Consult with an antenna manufacturer for an equivalent option.

Part Number	Type	Gain	Connector	Required Antenna Cable Loss	Application
A09-QRAMM	3" Quarter wave wire	2.1 dBi	MMCX	-	Fixed/Mobile
A09-QRSM-2.1*	Quarter wave 2.1" right angle	3.3 dBi	RPSMA	0.4 dB	Fixed/Mobile
A09-QW*	Quarter wave wire	1.9 dBi	Permanent	-	Fixed/Mobile
A09-QSM-3*	Quarter wave straight	1.9 dBi	RPSMA	-	Fixed/Mobile
A09-QSM-3H*	Heavy duty quarter wave straight	1.9 dBi	RPSMA	-	Fixed/Mobile
A09-QBMM-P6I*	Quarter wave w/ 6" pigtail	1.9 dBi	MMCX	-	Fixed/Mobile
A09-QHSM-2*	2" straight	1.9 dBi	RPSMA	-	Fixed/Mobile
A09-QHRSM-2*	2" right angle	1.9 dBi	RPSMA	-	Fixed/Mobile
A09-QHRSM-170*	1.7" right angle	1.9 dBi	RPSMA	-	Fixed/Mobile
A09-QRSM-380*	3.8" right angle	1.9 dBi	RPSMA	-	Fixed/Mobile
A09-QAPM-520*	5.2" articulated screw mount	1.9 dBi	Permanent	-	Fixed/Mobile
A09-QSPM-3*	3" straight screw mount	1.9 dBi	Permanent	-	Fixed/Mobile
A09-QAPM-3*	3" articulated screw mount	1.9 dBi	Permanent	-	Fixed/Mobile
A09-QAPM-3H*	3" articulated screw mount	1.9 dBi	Permanent	-	Fixed/Mobile

## Industry Canada (IC)

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

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

### Labeling requirements

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

Contains Model XBPSX Radio, IC: 1846A-XBPSX

The integrator is responsible for its product to comply with IC ICES-003 and 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 antenna certifications](#) 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 ci-dessous et ayant un gain admissible maximal et l'impédance requise pour chaque type d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.*

### Detachable antennas

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

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

## ACMA (Australia)

### Power requirements

Regulations in Australia stipulate a maximum of 30 dBm EIRP (Effective Isotropic Radiated Power). The EIRP equals the sum (in dBm) of power output, antenna gain and cable loss and cannot not exceed 30 dBm.

The EIRP formula for Australia is:

$$\text{power output} + \text{antenna gain} - \text{cable loss} \leq 30 \text{ dBm}$$

---

**Note** The maximum EIRP for the FCC (United States) and IC (Canada) is 36 dBm.

---

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

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

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

## Network configurations

---

Network topologies .....	94
Addressing .....	96
Basic communications .....	97
Acknowledged communications: Acknowledged mode .....	104

## Network topologies

The device supports three different network topologies:

- Point-to-point
- Point-to-multipoint
- Peer-to-peer

### Point-to-point networks

This following section provides the RF communication type and RF mode for point-to-point networks.

#### Definition

Point-to-point means an RF data link between two devices.



#### Sample network profile (Broadcast communications)

Use the default values for all devices.

#### Sample network profile (Acknowledged communications)

---

**Note** Assume the default value for all parameters that are not in this list. These profiles do not reflect addressing implementations.

---

1. Use XCTU or another terminal program to send the AM command. See [AM \(Auto-set MY\)](#) for details.
2. Set the destination address to 0xFFFF, send: **DT FFFF**

#### Basic RF modes

Streaming, Multi-Transmit, Repeater.

#### Acknowledged RF mode

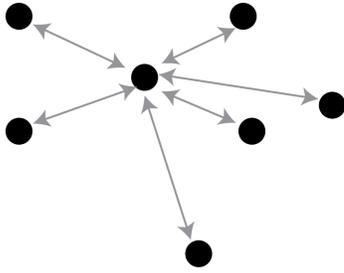
Acknowledged mode.

### Point-to-multipoint networks

This following section provides the RF communication type and RF mode for point-to-multipoint networks.

#### Definition

Point-to-multipoint means a network with RF data links between one base and multiple remotes.



### Sample network profile (Broadcast communications)

**Note** Assume the default value for all parameters that are not in this list. These profiles do not reflect addressing implementations.

Base:

1. Send **MY 0** to set the source address to 0x00.
2. Send **DT FFFF** to set the destination address to 0xFFFF.

Remotes:

1. Use XCTU or another terminal program to send the **AM** command. See [AM \(Auto-set MY\)](#) for details.
2. Send **DT 0** to set the destination address to 0x00.

### Sample network profile (Acknowledged communications)

**Note** Assume the default value for all parameters that are not in this list. These profiles do not reflect addressing implementations.

Base:

1. Send **MY 0** to set the source address to 0x00.
2. Send **DT FFFF** to set the destination address to 0xFFFF.
3. Send **RR 3** to set the number of retries to 3.

Remotes:

1. Use XCTU or another terminal program to send the **AM** command.
2. Send **DT FFFF** to set the destination address to 0xFFFF.
3. Send **RR 3** to set the number of retries to 3.

### Basic RF modes

Streaming, Multi-Transmit, Repeater, and Polling.

### Acknowledged RF mode

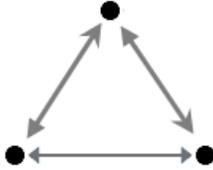
Acknowledged and Polling.

## Peer to peer networks

This following section provides the RF communication type and RF mode for peer-to-peer networks.

**Definition**

In Peer-to-peer networks, RF devices remain synchronized without the use of master/slave dependencies. Each device shares the roles of master and slave. Digi's peer-to-peer architecture features fast synch times (35 ms to synchronize devices) and fast cold start times (50 ms before transmission).

**Sample network profile (Broadcast communications)**


---

**Note** Assume the default value for all parameters that are not in this list. These profiles do not reflect addressing implementations.

---

Use the default values for all devices.

**Sample network profile (Acknowledged communications)**


---

**Note** Assume the default value for all parameters that are not in this list. These profiles do not reflect addressing implementations.

---

All devices:

1. Send **MY 0** to set the source address to 0x00.
2. Send **DT FFFF** to set the destination address to 0xFFFF.
3. Send **RR 3** to set the number of retries to 3.

**Basic RF modes**

Streaming.

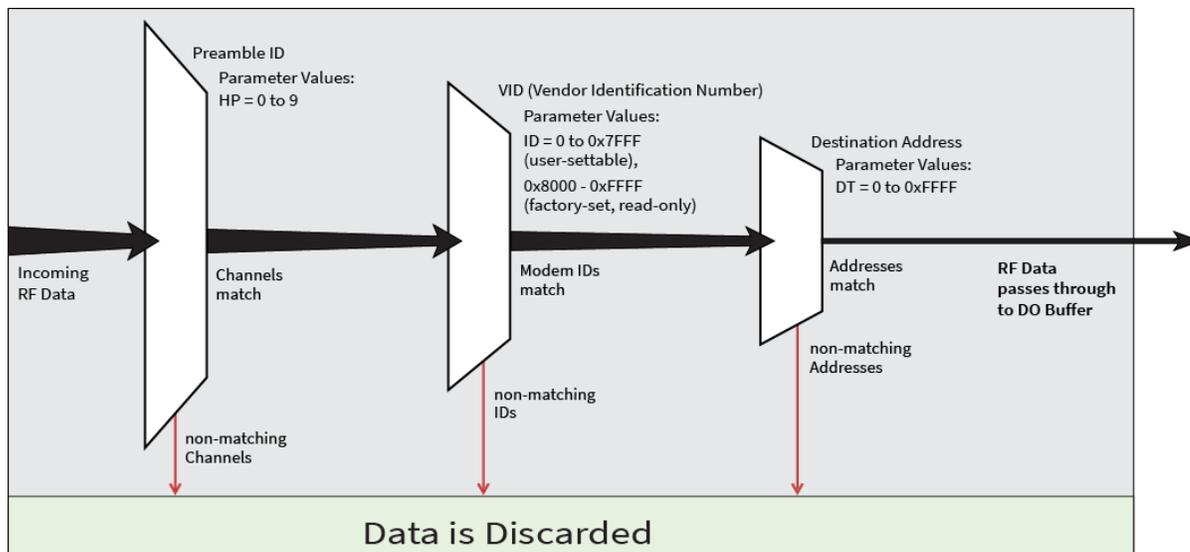
**Acknowledged RF mode**

Acknowledged.

## Addressing

Each RF packet contains addressing information that the receiving devices use to filter incoming RF data. Receiving devices inspect the Preamble ID (**HP** parameter), Vendor Identification Number (**ID** parameter) and Destination Address (**DT** parameter) in each RF packet. A receiving device discards all data that does not pass through all three network security layers.

The following image illustrates the addressing layers in the RF packet header.



### Address recognition

The transmitting device can address transmissions to a specific device or group of devices using the **DT** and **MK** commands. A receiving device only accepts a packet if it determines that the packet is addressed to it, either as a global or local packet. The receiving device makes this determination by inspecting the destination address of the packet and comparing it to its own address and address mask.

The transmitting device determines whether the packet is for a specific node (local address) or multiple nodes (global address) by comparing the packet's destination address (**DT**) and its own address mask (**MK**). This assumes you program the address masks on the transmitting device and receiving device to the same value for proper operation in each RF communication mode.

For more information, see [DT \(Destination Address\)](#) and [MK \(Address Mask\)](#).

## Basic communications

Basic communications includes two sub-types:

- Broadcast. By default, the XTend vB RF Modules communicate through Broadcast communications and within a peer-to-peer network topology. When any device transmits, all other devices within range receive the data and pass it directly to their host device.
- Addressed. If addressing parameters match, the device forwards the RF data it receives to the DOUT buffer; otherwise, it discards the RF data.

When using Basic communications, the integrator handles any functions, such as acknowledgments, at the application layer. The Broadcast modes provide transparent communications, meaning that the RF link replaces a wired link.

### Streaming mode (default)

Streaming mode is most appropriate for data systems that are more sensitive to latency and/or jitter than to occasional packet loss; for example: streaming audio or video.

Characteristics	Highest data throughput Lowest latency and jitter Reduced immunity to interference Transmissions never acknowledged (ACK) by receiving device (s)
Required parameter values (TX device)	RR = 0
Related commands	Networking (DT, MK, MY), Serial interfacing (PK, RB, RO, TT)

### **Streaming mode connection sequence**

Events and processes in this mode are common to all of the other RF modes.

When streaming data, the firmware only observes the RB and RO parameters on the first packet.

After transmission begins, the transmission event continues without interruption until the DIN buffer is empty or the device reaches the streaming limit (TT parameter). As with the first packet, the payload of each subsequent packet includes up to the maximum packet size (PK parameter).

The TX (transmitting) device specifies the TT parameter as the maximum number of bytes the TX device can send in one transmission event. After the device reaches the TT parameter threshold, the TX device forces a random delay of 1 to RN delay slots; exactly 1 delay slot if RN = 0.

The TX device sends subsequent packets without an RF initializer since RX (receiving) devices remain synchronized with the TX device for the duration of the transmission (from preceding packet information). However, due to interference, some RX devices may lose data (and synchronization to the TX device), particularly during long transmission events.

Once the TX device has sent all pending data or has reached the TT limit, the transmission event ends. The TX device does not transmit again for exactly RN delay slots if the local (for example the TX device's) RN parameter is set to a nonzero value. The RX device(s) do not transmit for a random number of delay slots between 0 and (RN-1) if the local (for example the RX device's) RN parameter is set to a non-zero value. These delays lessen congestion following long bursts of packets from a single TX device, during which several RX devices may have become ready to transmit.

### **Multi-transmit mode**

Use Multi-transmit mode for applications that require reliable delivery without using retries and acknowledgments.

Characteristics	Reliable delivery through forcing the transmission of every RF packet. Every RF packet is sent exactly MT + 1 times, with no delays between packets. Diminished throughput and increased latency.
Required parameter values (TX device)	MT ≥ 1.
Related commands	Networking (DT, MK, MY, RN, TT), Serial interfacing (BR, PK, RB, RO), RF interfacing (FS).

### Multi-transmit mode connection sequence

In Multi-transmit mode, the device re-transmits each packet MT times, for a total of (MT+1) transmissions. There is no delay between retransmissions, and the TX (transmitting) device never receives RF data between retransmissions. Each retransmission includes an RF initializer. A transmission event may include follow-on packets, each of which retransmit MT times. Devices ignore the Forced Sync (FS) parameter in Multi-Transmit Mode.

The firmware does not apply the RB and RO parameters to follow-on packets, meaning that once transmission has begun, it continues without interruption until the DIN buffer is empty or the device reaches the streaming limit (TT parameter). As with the first packet, the payload of each follow-on packet includes up to the maximum packet size (PK parameter) bytes, and the TX device checks for more pending data near the end of each packet. The device does not send follow-on packets until it finishes all retransmissions of the previous packet.

The TX device specifies the streaming limit (TT) as the maximum number of bytes that the TX device can send in one transmission event, which may consist of many packets. If the device reaches the TT parameter limit, the TX device forces a random delay of 1 to RN delay slots (exactly 1 delay slot if RN is zero). In Multi-transmit mode, the firmware counts each packet only once when tracking the streaming limit (TT), no matter how many times it is retransmitted.

When an RX (receiving) device receives a Multi-transmit packet, it calculates the amount of time remaining in the Multi-transmit event, and inhibits its own transmissions for the duration of the Multi-transmit event, plus a random number of delay slots between 0 and (RN-1). If the local RN parameter is zero, the delay is only for the calculated duration of the event. An RX device only needs to receive one of the transmissions, and it keep the channel off until the TX device is complete. If follow-on packets are coming, the RX devices move to the new frequency and listen for the follow-on packet for a specific period of time.

### Repeater mode

Use Repeater mode in networks where you need intermediary devices to relay data to devices beyond the transmission range of the base device.

<p>Characteristics</p>	<p>Low power consumption.                  Minimizes interference                  Tags each RF packet with a unique Packet ID (PID).                  Each repeater only repeats a packet once (the PID tracks the packet).                  Increases latency and decreases throughput. The number of hops determine latency and throughput, not the number of repeaters. Multiple repeaters within range of a source node count as one hop.                  All RF packets propagate to every device in the network (filtering rules apply).                  Packet destination addresses (DT) determine which packets the device sends out the serial port and/or retransmits.                  Broadcast communications: each packet comes out every node exactly once.                  Addressed communications: all devices see every packet. Only the devices with a matching address forward it to the DOUT buffer.</p>
<p>Constraints</p>	<p>Requires that each device have a unique MY parameter.                  System must introduce only one packet at a time to the network for transmission.                  The PK parameter determines the maximum number of bytes.                  Each hop (H) decreases network throughput by a factor of 1/(H+1). Additional repeaters add network redundancy without decreasing throughput.</p>

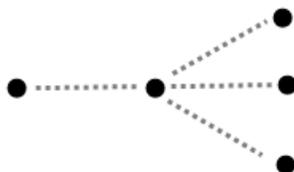
Suggestions	Insert a variable delay before repeating packets to avoid collisions (based on RSSI). Buffer any incoming serial data and delay response packet transmissions until the previous packet clears out of the network. For best results, use the RO and RB commands to ensure that the RF packets align with the underlying protocol packets as the network can only accept one RF packet at a time.
Required parameter values (TX device)	MD = 5 or 6. MY = unique value. You can accomplish this by issuing the AM and WR commands to all devices in the network.
Related commands	Networking (MD, DT, MY, AM), Serial interfacing (RN, PK, RO, RB)

### Repeater mode theory of operation

You can extend the effective range and reliability of your data system by forwarding traffic through one or more repeaters. Instead of using routing tables and path discovery to establish dynamic paths through a network, the repeater system uses a sophisticated algorithm to propagate each RF packet through the entire network.

The network supports RF packets up to 2048 bytes when the RF data rate is set at 9600 bps (BR = 0). The repeater network can operate using broadcast or addressed communications for multi-drop networks, and it works well in many systems with no special configuration.

When in Repeater mode, the network repeats each message among all available devices exactly one time. This mechanism eliminates the need for configuring specific routes. The following figure illustrates the Repeater network topology.



### Configure a repeater network

If an RF link is weak, a device is out-of-range or a difficult RF environment is present; you can use repeaters to extend the effective range and reliability of the network.

A network may consist of End Nodes (EN), End/Repeater Nodes (ERN) and a Base Node (BN). The base node initiates all communications. Both Repeater Nodes and End Nodes can source data, allowing connection to host devices. Repeater Nodes however, are able to repeat information in a simple store and forward fashion. As an example, one End Node (which can be a base or remote) must send a message to another End Node. Because the End Node is out of range of the base device, you can use a repeater to forward information from the Base to the End Node.

You can configure a repeater network to operate using Basic Broadcast or Basic Addressed communications. The addressing capabilities of the device allow integrators to send a packet as a global packet (**DT** = 0xFFFF) and shift out of every device in the network (Basic Broadcast). Alternately, you can send the packet with a specific **DT** parameter so that only a specific remote node accepts it (Basic Addressed).

**Repeater network: configure communications**

To configure a Repeater network for Basic broadcast communications:

1. Assign each device a unique **MY** (source) address. Use the **AM** command to configure a unique source address based on the device serial number. This is essential because a unique packet ID on each RF packet is based on the originator’s **MY** value.
2. Set **DT** = 0xFFFF to enable Basic Broadcast communications OR Basic Addressed communications (**DT** specifies a specific destination).
3. Configure **PK**, **RO** and **RB** to ensure that the RF packet aligns with the protocol packet. For example:  
**PK** = 0x100  
**RB** = 0x100  
**RO** depends on the baud rate
4. Set **MD** = 5 to configure one or more devices that you do not intend to be repeaters as repeater End Nodes in the system.
5. Set **MD** = 6 to configure remote nodes as destinations. This ensures that the remote node waits for the repeater traffic to subside before it transmits a response.

To configure a Repeater network for Basic addressed communications, use **DT** to assign unique addresses to each device in the network.

**AT commands to configure Repeater network functions**

The following table lists the AT commands you use to configure repeater functions.

AT Command	Binary Command	AT Command Name	Range	# Bytes Returned	Factory Default
<b>AM</b>	0x3A (58d)	Auto-set MY	-	-	-
<b>DT</b>	0x00 (0d)	Destination Address	0 - 0xFFFF	2	0
<b>MD</b>	0x3C (60d)	RF Mode	0 - 6	1	0
<b>MY</b>	0x2A (42d)	Source Address	0 - 0xFFFF	2	0xFFFF
<b>RN</b>	0x19 (25d)	Delay Slots	0 - 0xFF [slots]	1	0
<b>WR</b>	0x08 (8d)	Write	-	-	-

**Repeater network algorithm details**

The firmware uses an algorithm to propagate each RF packet through the entire repeater network. Within a repeater network, the firmware only defines Repeater Nodes and repeater End Nodes. Repeater Nodes forward messages on to other devices within range; End Nodes do not.

The algorithm maintains a list of messages previously received in a buffer. The firmware discards messages already in the buffer. This eliminates End Nodes receiving multiple copies of a packet from more than one source, and also eliminates multiple repeaters within range of each other from continually passing messages in an infinite loop.

Packet ID (PID) is composed of the TX (transmitting) device MY address and the packet sequence number.

The firmware ignores incoming packets with a PID already in the buffer.

Each device maintains a PID buffer 4-deep of previously received packets (managed as FIFO).

The firmware may shift packets out the serial port and/or repeat them depending on the DT parameter in the RF packet. The following table shows the basis for these decisions.

Address Match	Send out serial port?	Repeat?
Global	Yes	Yes
Local	Yes	No
None	No	Yes

### **Repeat delay based on RSSI**

A transmitted packet may be received by more than one repeater at the same time. In order to reduce the probability that the repeaters will transmit at the same instant, resulting in a collision and possible data loss; the firmware uses an algorithm that allows a variable back-off prior to a repeater retransmitting the packet. The algorithm allows devices that receive the packet with a stronger RF signal (RSSI) to have the first opportunity to retransmit the packet.

Use the RN (Delay Slots) parameter to configure this delay. Set RN = 0 (no delays) for small networks with few repeaters or repeaters that are not within range of each other. Set RN = 1 for systems with two to five repeaters that may be within range of each other.

The actual length of the delay is computed by the formula:

$$\text{Delay (ms)} = L * DS$$

$$DS = (-41 - \text{RSSI}) / 10 * RN + \text{RandomInt}(0, RN)$$

Where L is the length of the transmitted packet in milliseconds, DS is the number of delay slots to wait, RSSI is the received signal strength in dBm, RN is the value of the RN register and RandomInt(A,B) is a function that returns a random integer from A to B-0

### **Response packet delay**

As a packet propagates through the repeater network, if any node receives the data and generates a quick response, the network needs to delay the response so as not to collide with subsequent retransmissions of the original packet. To reduce collisions, both repeater and end node devices in a repeater network delay transmission of data shifted in the serial port to allow any repeaters within range to complete their retransmissions.

The time for this delay is computed by the formula:

$$\text{Maximum Delay (ms)} = L * DS$$

$$DS = ((-41 - (-100)) / 10) * RN + RN + 1$$

Where L is the length of the transmitted packet in milliseconds, DS is the number of delay slots to wait, RSSI is the received signal strength in dBm, and RN is the value of the RN register.

### **Bandwidth considerations**

Using broadcast repeaters in a network reduces the overall network data throughput as each repeater must buffer an entire packet before retransmitting it. For example: if the destination is within range of the transmitter and the packet is 32-bytes long, the transmission takes 12 ms on a device operating at 115,200 baud. If the same packet must propagate through two repeaters, it takes 12 ms to arrive at the first repeater, 12 ms to get to the second and a final 12 ms to reach the destination for a total of 36 ms. Accounting for UART transfer times (~1ms/byte at 9600 baud), the time for a server to send a 32-byte query and receive a 32-byte response is about 200 ms, allowing for

5 polls per second. With the two repeaters in the path, the same query/response sequence would take about 500 ms for two polls per second.

Generally, network throughput decreases by a factor of  $1/(R+1)$ , with R representing the number of repeaters between the source and destination.

### Polling mode (basic)

Polling mode (basic) and Polling mode (acknowledged) operate in the same way. The only difference between the two modes is in their means of achieving reliable delivery of data. Polling mode (basic) uses multiple transmissions to achieve reliable delivery.

Characteristics	<p>Uses a high percentage of available network bandwidth.</p> <p>Eliminates collisions.</p> <p>Works with reliable delivery (RR or MT parameters).</p> <p>Supports binary data transfers.</p> <p>Base device requests packets from remote device by polling a sequential range of addresses.</p> <p>Base device is configured to specify the range of addresses being polled.</p> <p>Uses inter-character delay to create RF packet lengths aligned with protocol packet lengths up to 2048 bytes long.</p>
Constraints	<p>The minimum time interval between polling cycles is configurable. However, if the remote devices cannot all be processed within that time interval, the polling cycle is ineffective (i.e. it will impose no additional delay). In order to ensure a pause between polling cycles, you must set PD to a value that is large enough to accommodate the pause.</p>
Recommended use	<p>Use for point-to-multipoint applications that require Reliable Delivery of data. Use this mode when it is critical that a base device be able to discern data coming from multiple devices.</p>
Required parameter values (Base)	<p>MD (RF Mode) = 3</p> <p>PB (Polling Begin Address)</p> <p>PE (Polling End Address)</p>
Required parameter values (Remote)	<p>MD (RF Mode) = 4</p>
Related commands	<p>Networking: MT, PD, DT, MY, and AM</p>

### Polling mode theory of operation

A Polling Base device cycles through a sequential range of addresses. The Polling Base polls each Polling Remote device, waits for a response, then poll the next remote address in the sequence. Each Polling Remote responds by sending the data from its DIN buffer following the RB and RO parameters. When there is no eligible data to send, the Polling Remote does not respond. The Polling Base polls the next address in the polling sequence after a short delay.

### Configure a Polling Base

To configure a device as a Polling Base:

1. Set **MD** = 3.
2. Set **MY** = 0.
3. Set the sequential range of polling addresses using **PB** and **PE**.
4. (Optional) Enable Basic Reliable Delivery (**MT**  $\geq$  0). The firmware also supports Acknowledged Reliable Delivery. For more information, see [Polling mode \(acknowledged\)](#).
5. (Optional) Use **PD** to configure a delay between polls to slow down the system, if needed.
6. (Optional) Enable API Mode to address remote devices within polling range on a packet-by-packet basis.

### Configure a Polling Remote

To configure a device as a Polling Remote:

1. Set **MD** = 4.
2. Configure sequential source addresses for all remote devices using **MY**.
3. Set **DT** to point to the Polling Base (**DT** = 0x0000).
4. (Optional) Enable Basic Reliable Delivery (**MT**  $\geq$  0). The firmware also supports Acknowledged Reliable Delivery. For more information, see [Polling mode \(acknowledged\)](#).

## Acknowledged communications: Acknowledged mode

Use Acknowledged mode for applications that need reliable delivery. If messages are smaller than 256 bytes, use the RB and RO commands to align RF packets to application packets.

Characteristics	Reliable delivery through positive acknowledgments for each packet. Throughput, latency and jitter vary depending on the quality of the channel and the strength of the signal.
Required parameter values (TX device)	RR (Retries) $\geq$ 1
Related commands	Networking (DT, MK, RR), Serial Interfacing (PK, RN, RO, RB, TT)

### Acknowledged mode connection sequence

After sending a packet while in Acknowledged mode, the TX (transmitting) device listens for an acknowledgment (ACK). If it receives the ACK, it either moves on to sending a subsequent packet if more transmit data is pending or waits for exactly RN random delay slots before allowing another transmission if no more data is pending transmit.

If the TX device does not receive the ACK within the allotted time, it retransmits the packet with a new RF initializer following the ACK slot. There is no delay between the first ACK slot and the first retransmission. Subsequent retransmissions incur a delay of a random number of delay slots, between 0 and RN. If RN is set to 0 on the TX device, there are never any back-off delays between retransmissions. During back-off delays, the TX device goes into Idle Mode and may receive RF data. This can have the effect of increasing the back-off delay, as the device cannot return to Transmit (or retransmit) mode as long as it is receiving RF data.

After receiving and acknowledging a packet, the RX (receiving) device moves to the next frequency and listens for either a retransmission or new data for a specific period of time. Even if the TX device

indicates that it has no more pending transmit data, it may not have received the previous ACK, and so may retransmit the packet, possibly with no delay after the ACK slot. In this case, the RX device always detects the immediate retransmission, which holds off the communications channel and reduces collisions. RX devices acknowledge each retransmission they receive, but they only pass the first copy of a packet they receive out the UART.

The device does not apply the RB and RO parameters to subsequent packets, meaning that once transmission begins, it continues uninterrupted until the DIN buffer is empty or it reaches the streaming limit (TT parameter). As with the first packet, the payload of each subsequent packet includes up to the maximum packet size (PK parameter), and the TX device checks for more pending data near the end of each packet.

The TT parameter specifies the maximum number of bytes that the TX device sends in one transmission event, which may consist of many packets and retries. If a device reaches the TT parameter limit, the TX device forces a random delay of 1 to RN delay slots (exactly 1 delay slot if RN is zero). Each packet counts only once toward TT, no matter how many times the packet is retransmitted.

Subsequent packets in Acknowledged mode are similar to those in Streaming mode, with the addition of an ACK between each packet, and the possibility of retransmissions. The device sends subsequent packets without an RF initializer, as the RX devices are already synchronized to the TX device from the preceding packet(s) and they remain synchronized for the duration of the transmission event. Each packet retransmission includes an RF initializer.

Once the TX device sends all pending data or reaches the TT limit, the acknowledged transmission event is complete. The TX device does not transmit again for exactly RN delay slots, if the local RN parameter is set to a non-zero value. The RX device does not transmit for a random number of delay slots between 0 and (RN-1), if the local RN parameter is set to a non-zero value. The intent of these delays is to lessen congestion following long bursts of packets from a single TX device, during which several RX devices may have themselves become ready to transmit.

### Polling mode (acknowledged)

Polling mode (acknowledged) and Polling mode (basic) operate in the same way. The difference between the two modes is in their means of achieving the reliable delivery of data. In Polling mode (acknowledged), the firmware achieves reliable delivery using retries and acknowledgments.

<p>Characteristics</p>	<p>Uses a high percentage of available network bandwidth.                  Eliminates collisions.                  Works with reliable delivery (RR or MT parameters).                  Supports binary data transfers.                  Base device requests packets from remote device by polling a sequential range of addresses.                  Base device is configured to specify the range of addresses being polled.                  Uses inter-character delay to create RF packet lengths aligned with protocol packet lengths up to 2048 bytes long.</p>
<p>Constraints</p>	<p>The minimum time interval between polling cycles is configurable. However, if the remote devices cannot all be processed within that time interval, the polling cycle is ineffective (i.e. it will impose no additional delay). In order to ensure a pause between polling cycles, PD must be set to a value which is large enough to accommodate the pause.</p>

Recommended use	Use for point-to-multipoint applications that require Reliable Delivery of data. Use this mode when it is critical that a base device be able to discern data coming from multiple devices.
Required parameter values (Base)	MD (RF Mode) = 3, PB (Polling Begin Address) PE (Polling End Address)
Required parameter values (Remote)	MD (RF Mode) = 4
Related commands	Networking (RR, PD, DT, MY, AM)

For configuration and theory of operation information, see [Polling mode theory of operation](#), [Configure a Polling Base](#) and [Configure a Polling Remote](#).

## Development Kit

---

Development Kit contents .....	108
Interface hardware .....	109
XTIB-R RS-232/485 Interface Board .....	109
Adapters .....	112
Interface protocols .....	115

## Development Kit contents

The XTend vB RF Module Development Kit includes the hardware and software you need to rapidly create long-range wireless links between devices. The following table shows the contents of the kit.

Item	Qty	Description	Part number
XTend vB RF Module	1	Long range 900 MHz RF Module(with RPSMA connector)	XT09-SI
XTend vB RF Module	1	Long range 900 MHz RF Module (with MMCX antenna)	XT09-MI
Antenna	1	900 MHz RPSMA, 6" half-wave, dipole, articulating, RPSMA	A09-HASM-675
Antenna	1	900 MHz RPSMA, 7" half-wave, dipole, articulating, w/ pigtail, MMCX	A09-HABMM-P5I
RS-232 Interface Board	2	Enables communication to RS-232 devices	XTIB-R
RS-232 Cable (6')	2	Connects interface board to devices having an RS-232 serial port	JD2D3-CDS-6F
Serial loopback adapter	1	Connects to the female RS-232 (DB-9) serial connector of the Digi Interface Board and can be used to configure the device to function as a repeater (for range testing)	JD2D3-CDL-A
NULL modem adapter (male-to-male)	1	Connects to the female RS-232 (DB-9) serial connector of the Digi Interface Board and can be used to connect the device to another DCE (female DB9) device	JD2D2-CDN-A
NULL modem adapter (female-to-female)	1	Used to bypass radios to verify serial cabling is functioning properly	JD3D3-CDN-A
Male DB-9 to RJ-45 adapter	1	Facilitates adapting the DB-9 connector of the Digi Interface Board to a CAT5 cable (male DB9 to female RJ45)	JE1D2-CDA-A
Female DB-9 to RJ-45 adapter	1	Facilitates adapting the DB-9 connector of the Digi Interface Board to a CAT5 cable (female DB9 to female RJ45)	JE1D3-CDA-A
Power adapter	2	Allows you to power the Interface Board with a 110 V AC power supply (not included with international (-INT) development kits)	JP4P2-9V10-6F

## Interface hardware

The XTend vB RF Module Development Kit includes a pair of RS-232 interface boards that support the RS-232, RS-485 and RS-422 protocols. When you mount the devices to the interface boards, the boards provide the following development tools:

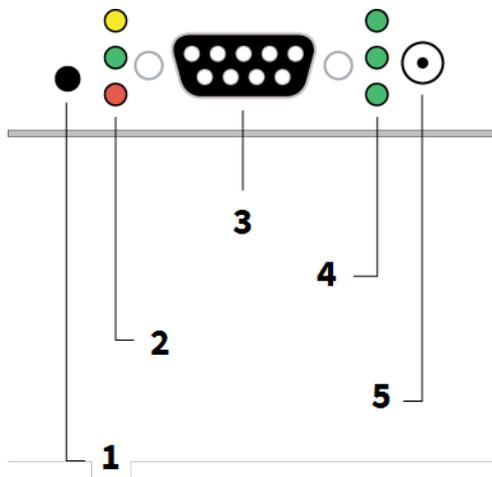
- Fast and direct connection to serial devices (such as PCs) and easy access to the device registries. The parameters stored in the registry allow you to customize the devices to suit the specific needs of your data systems.
- External DIP switch to automatically configure common device profiles.
- Signal conversion between TTL levels and RS-232 levels.

The Digi Interface board can connect the device to any device that has an available RS-232, RS-485 or RS-422 connection.

This documentation refers to a XTend vB RF Module mounted to an interface board as a "Module Assembly."

### XTIB-R RS-232/485 Interface Board

The following figure shows a front view of the board. The table explains the numbered callouts in the figure.



Number	Description
1	Configuration switch
2	I/O and Power LEDs
3	DB-9 Serial port
4	RSSI LEDs
5	Power connector

## Configuration switch

The Configuration switch provides an alternate method for entering Command mode. To enter Command mode at the device's default RF data rate, hold the Configuration switch down for two seconds.

## I/O and Power LEDs

The LEDs visualize status information and indicate device activity as follows:

LED color	LED location	Indication
Yellow	Top	Serial data out (to the host)
Green	Middle	Serial data in (from the host)
Red	Bottom	Power/TX indicator (the red light is on when powered, it pulses on and off briefly during RF transmission)

## Serial port

The serial port is a standard female DB-9 (RS-232) connector. You can also use this connector for RS-485 and RS-422 connections.

## RSSI LEDs

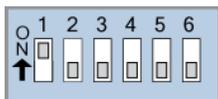
The RSSI LEDs indicate the amount of fade margin present in an active wireless link. Fade margin is the difference between the incoming signal strength and the device's receiver sensitivity. The LED indications are as follows:

Number of LEDs on	Indicates
3	Very strong signal (> 30 dB fade margin)
2	Strong signal (> 20 dB fade margin)
1	Moderate signal (> 10 dB fade margin)
0	Weak signal (< 10 dB fade margin)

## Power connector

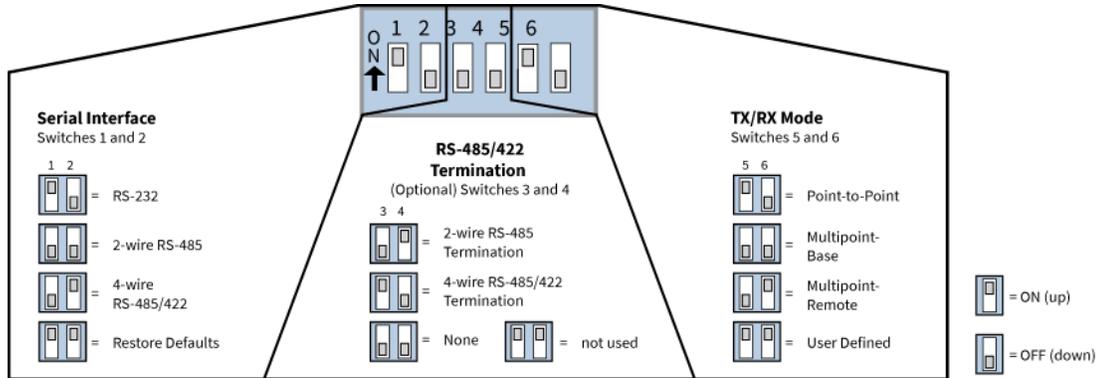
7-28 VDC power connector (center positive, 5.5/2.1 mm). The XTIB-R interface board can accept voltages as low as 5 V. Contact Digi Technical Support to enable this option.

## XTIB-R Dip switch



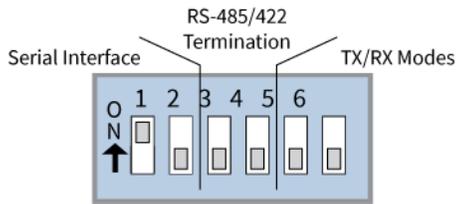
The DIP switch automatically configures the device to operate in different modes during the power-on sequence. Each time the Module Assembly (interface board with a device) powers-on, intelligence on the board programs the attached device according to the positions of the DIP Switch.

The following figure illustrates the DIP Switch settings.



### Automatic DIP switch configurations

Each time you power on the Module Assembly, the firmware sends AT commands to the on-board RF module as dictated by the positions of the DIP switches. The following figure shows the DIP switch with the various configurations.



The following table shows the commands that the firmware sends to the module as a result of DIP switch settings. DIP switches 3 and 4 are only used for RS-485/422 termination and do not affect the configuration of the device. In the tables, SW means switch.

Switch condition	Behavior	Commands sent during power-up
<b>Switches 1 and 2: Restore defaults / serial interfacing</b>		
SW1: ON (up) SW2: ON (up)	Restore defaults	RE (restore defaults) WR (write defaults to non-volatile memory)
SW1: ON (up) SW2: OFF (down)	RS-232 operation	CS 0 (RS-232, $\overline{\text{CTS}}$ flow control)
SW1: OFF (down) SW2: OFF (down)	RS-485/422 operation	CS 3 (RS-485 or RS-422 operation)

Switch condition	Behavior	Commands sent during power-up
<b>Switches 5 and 6: TX/TX modes</b>		
SW5: OFF (down) SW6: OFF (down)	Multipoint base	MY 0 (Source address = 0) DT FFFF (Destination address = broadcast address) MT 3 (Enable 3 multi-transmit retries)
SW5: OFF (down) SW6: ON (up)	Multipoint remote	AM (Generate unique source address) DT 0 (Destination address = 0) MT 0 (Disable multi-transmit) RR A (Enable 10 unicast retries)
SW5: ON (up) SW6: OFF (down)	Point-to-point	AM (Generate unique source address) DT FFFF (Set destination address to broadcast) MT 3 (Enable 3 multi-transmit retries)
SW5: ON (up) SW6: ON (up)	User defined	No addressing commands are sent to the device, refer to the following table for details.

The following table shows the user-defined mode, when switches 5 and 6 are ON (up). The behavior of pin 9 (GPO1) varies depending on the state of the DIP switches and the CS command parameter upon power-up.

Switch condition	CS Condition	Command sent during power-up
SW1: ON (up) SW2: OFF (down) SW5: ON (up) SW6: ON (up)	If CS = 0, 1, 2 or 4	CS parameter remains the same
	If CS = 3	CS 0 (RS-232 operation with $\overline{\text{CTS}}$ flow control)
SW1: OFF (down) SW2: ON (up) SW5: ON (up) SW6: ON (up)	If CS = 2	CS parameter remains the same
	If CS = 0, 1, 3 or 4	CS 3 (RS-485 or RS-422 operation)
SW1: OFF (down) SW2: OFF (down) SW5: ON (up) SW6: ON (up)	If CS = 2	CS parameter remains the same
	If CS = 0, 1, 3 or 4	CS 3 (RS-485 or RS-422 operation)

## Adapters

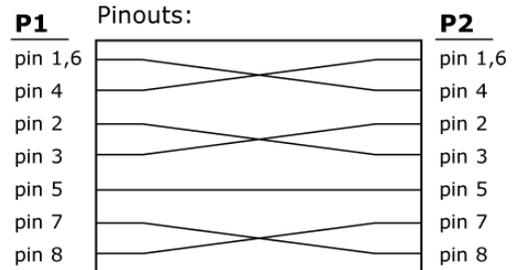
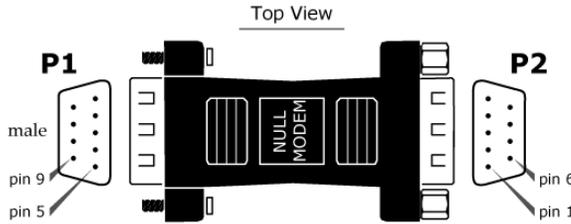
The development kit includes several adapters that facilitate the following functions:

- Performing range tests
- Testing cables
- Connecting to other RS-232 DCE and DTE devices
- Connecting to terminal blocks or RJ-45 (for RS-485/422 devices)

### NULL Modem Adapter (male-to-male)

Part Number: **JD2D2-CDN-A** (Black, DB-9 M-M)

The male-to-male NULL modem adapter is used to connect two DCE devices. A DCE device connects with a straight-through cable to the male serial port of a computer (DTE). The following image shows the Male NULL modem adapter and its pinouts.

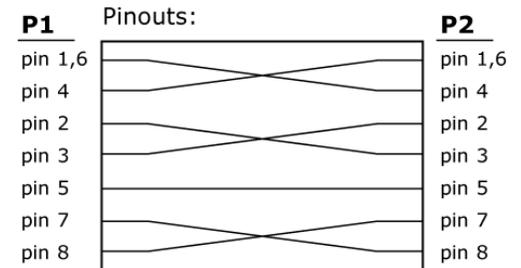
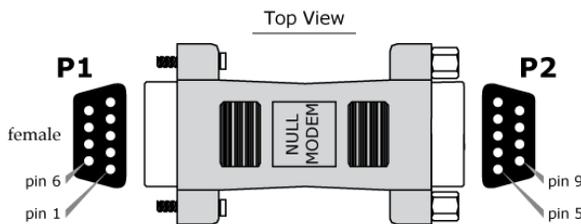


### NULL Modem Adapter (female-to-female)

Part Number: **JD3D3-CDN-A** (Gray, DB-9 F-F)

Use the female-to-female NULL modem adapter to verify that serial cabling is functioning properly. To test cables, insert the female-to-female NULL modem adapter in place of a pair of device assemblies (RS-232 interface board and XTend vB RF Module) and test the connection without devices in the connection.

The following figure shows the adapter and its pinouts.

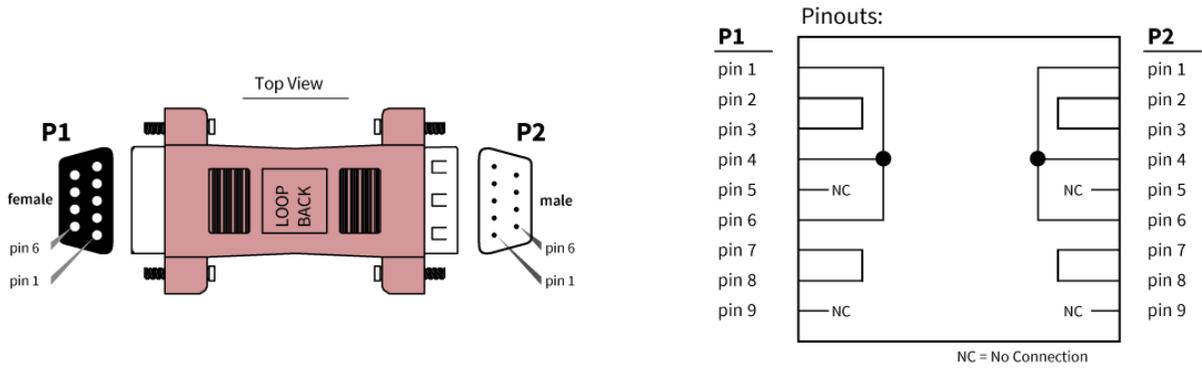


### Serial Loopback Adapter

Part Number: **JD2D3-CDL-A** (Red, DB-9 M-F)

Use the serial loopback adapter for range testing. During a range test, the serial loopback adapter causes the device to function as a repeater by looping serial data back into the device for retransmission.

The following image shows the adapter and its pinouts.



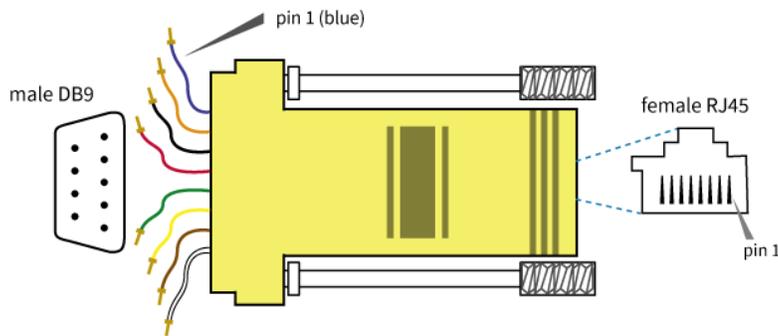
### Male DB-9 to RJ-45 Adapter

Part Number: **JD2D2-CDN-A** (Yellow)

This adapter facilitates adapting the DB-9 connector of the Interface Board to a CAT5 cable (male DB9 to female RJ45).

For connection guidelines, see [RS-485 \(4-wire\)](#) and [RS-422 operation](#).

The following image shows the adapter and its pinouts.



RJ-45 Pin Specifications

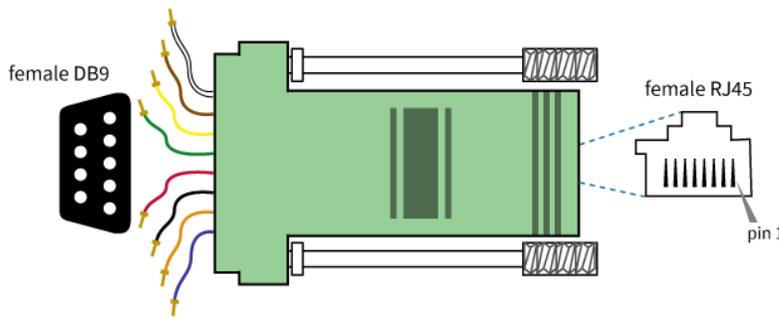
RJ45 Pin	Wire Color
1	blue
2	orange
3	black
4	red
5	green
6	yellow
7	brown
8	white

### Female DB-9 to RJ-45 Adapter

Part Number: **JD3D3-CDN-A** (Green) This adapter facilitates adapting the DB-9 Connector of the Interface Board to a CAT5 cable (female DB9 to female RJ45).

For connection guidelines, see ‘RS-485 (4-wire) & RS-422 Operation’ sections.

The following image shows the adapter and its pinouts.



RJ-45 Pin Specifications

RJ45 Pin	Wire Color
1	blue
2	orange
3	black
4	red
5	green
6	yellow
7	brown
8	white

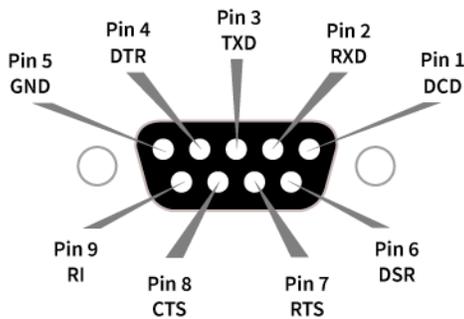
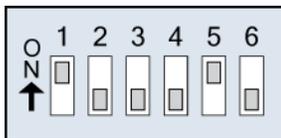
## Interface protocols

The XTend vB RF Module Assembly supports the following interfacing protocols:

- RS-232
- RS-485 (2-wire) half-duplex
- RS-485 (4-wire) and RS-422

### RS-232 operation

The following figures show the RS-232 DIP switch settings and the pins used on the female RS-232 (DB-9) serial connector. The Module Assembly reads and applies the DIP switch settings only during power-on.

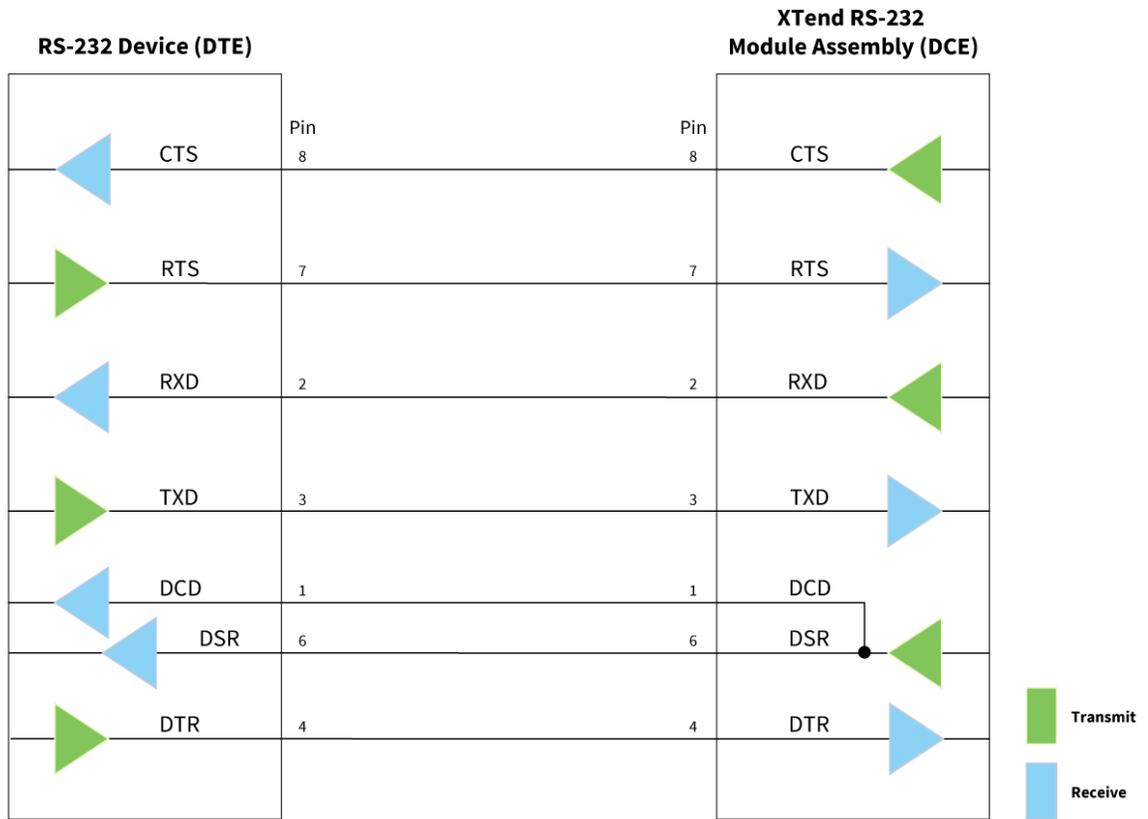


The following table provides the RS-232 signals and their implementations on the Module Assembly. Low-asserted signals have a horizontal line over the pin name.

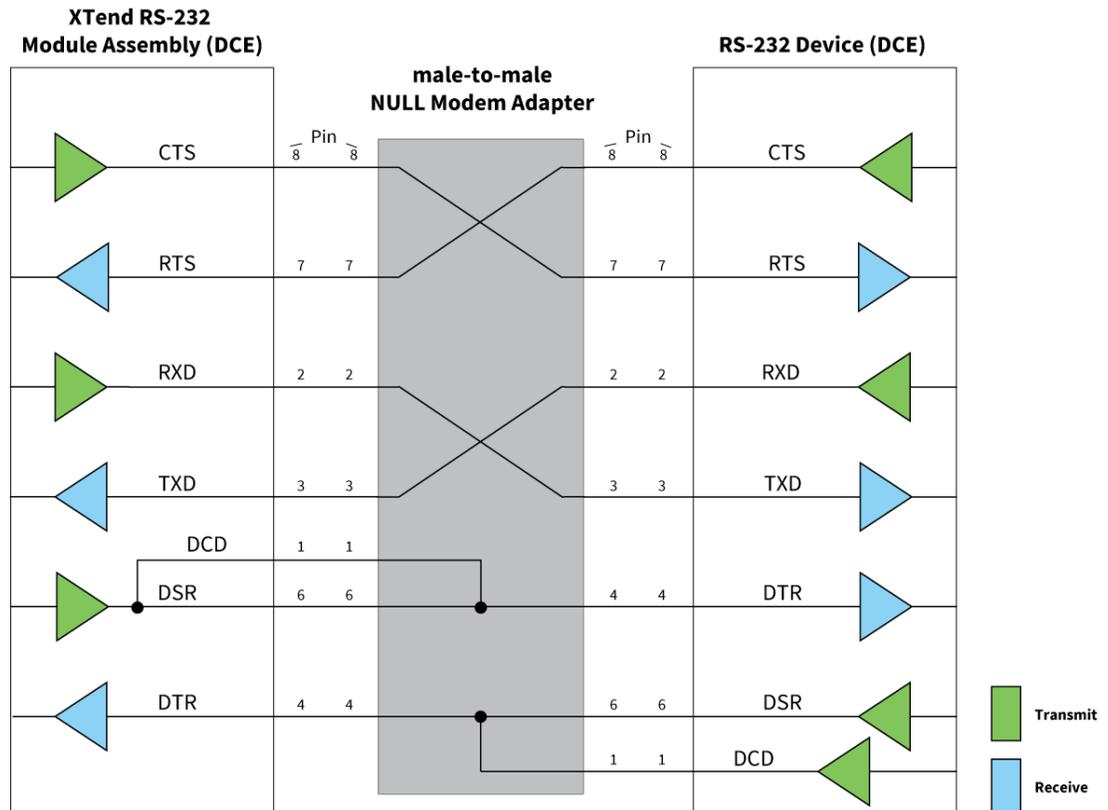
DB-9 pin	RS-232 name	XCTU name	Description	Implementation
1	DCD	GPO2	Data-Carrier-Detect	Connected to DSR (pin6 of DB-9)
2	RXD	DOUT	Received Data	Serial data exiting the Module Assembly (to host)
3	TXD	DIN	Transmitted Data	Serial data entering into the Module Assembly (from host)
4	DTR	GPI2	Data-Terminal-Ready	Can enable power-down on the Module Assembly
5	GND	-	Ground Signal	Ground
6	DSR	GPO2	Data-Set-Ready	Connected to DCD (pin1 of DB-9)
7	$\overline{\text{RTS}} / \text{CMD}$	GPI1	Request-to-Send / Command Mode	Provides $\overline{\text{RTS}}$ flow control or enables Command mode
8	$\overline{\text{CTS}}$	GPO1	Clear-to-Send	Provides $\overline{\text{CTS}}$ flow control
9	RI	-	Ring Indicator	Optional power input that is connected internally to the positive lead of the front power connector

### ***RS-232 wiring diagrams***

The following diagram shows the DTE device (RS-232, male DB-9 connector) wired to a DCE Module Assembly (female DB-9).



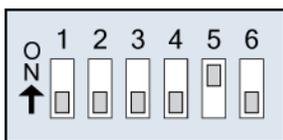
The following diagram shows the DCE Module Assembly (female DB-9 connector) wired to a DCE device (RS-232, male DB-9).



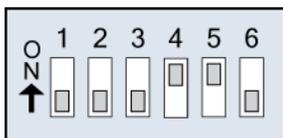
### RS-485 (2-wire) operation

When operating within the RS-485 protocols, all communications are half-duplex. The Module Assembly reads and applies the DIP switch settings only during power-on.

The following figure shows the RS-485 (2-wire) half-duplex DIP switch settings.

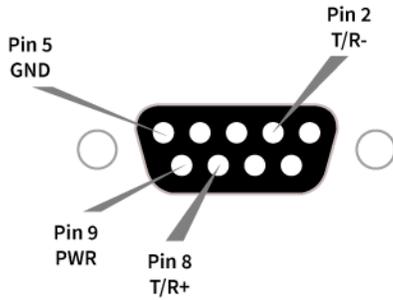


The following figure shows the RS-485 (2-wire) with termination (optional) DIP switch settings.



Enabling termination activates a 120 Ω resistor between T+ and T-.

The following figure shows the pins that the female RS-232 (DB-9) serial connector uses.



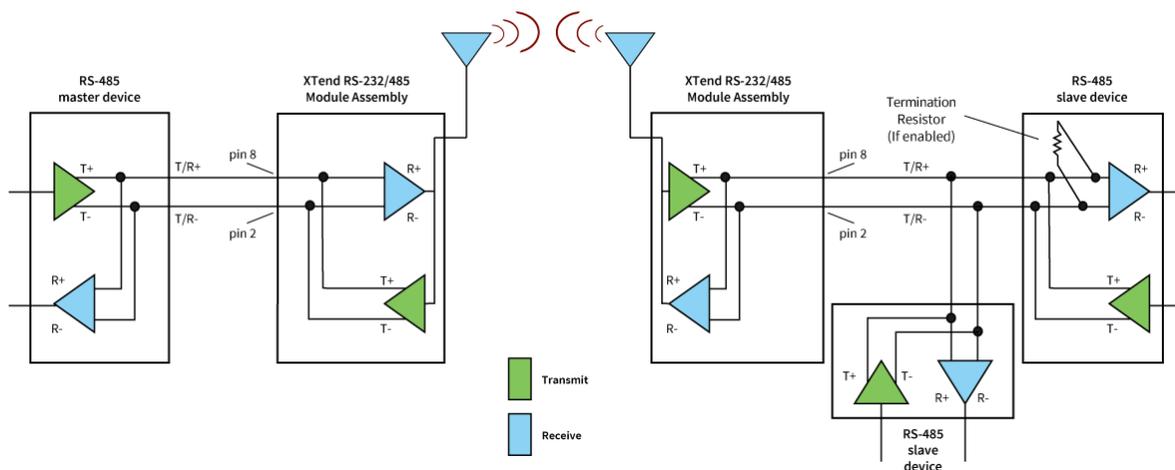
For the RJ-45 connector pin designations to use in RS-485/422 environments see [Male DB-9 to RJ-45 Adapter](#) and [Female DB-9 to RJ-45 Adapter](#).

The following table provides the RS-485 (2-wire half-duplex) signals and their implementations on the Module Assembly.

DB-9 Pin	RS-485 name	Description	Implementation
2	T/R- (TRA)	Negative data line	Transmit serial data to and from the Module Assembly
5	GND	Ground signal	Ground
8	T/R+ (TRB)	Positive data line	Transmit serial data to and from the Module Assembly
9	PWR	Power	Optional power input that is connected internally to the front power connector
1, 3, 4, 6, 7	not used		

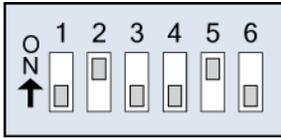
### RS-485 wiring diagrams

The following diagram shows the Module Assembly in an RS-485 (2-wire) half-duplex environment.

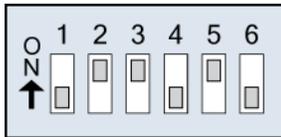


### RS-485 (4-wire) and RS-422 operation

The Module Assembly reads and applies the DIP switch settings only during power-on. The following figure shows the RS-485 (4-wire) and RS-422 DIP switch settings.

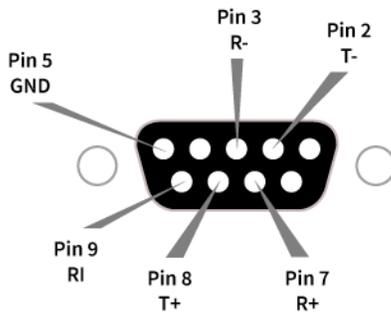


The following figure shows the RS-485 (4-wire) and RS-422 DIP switch settings with termination (optional).



Enabling termination activates a 120 Ω resistor between T+ and T-.

The following figure shows the pins that the female RS-232 (DB-9) serial connector uses.



For the RJ-45 connector pin designations to use in RS-485/422 environments see [Male DB-9 to RJ-45 Adapter](#) and [Female DB-9 to RJ-45 Adapter](#).

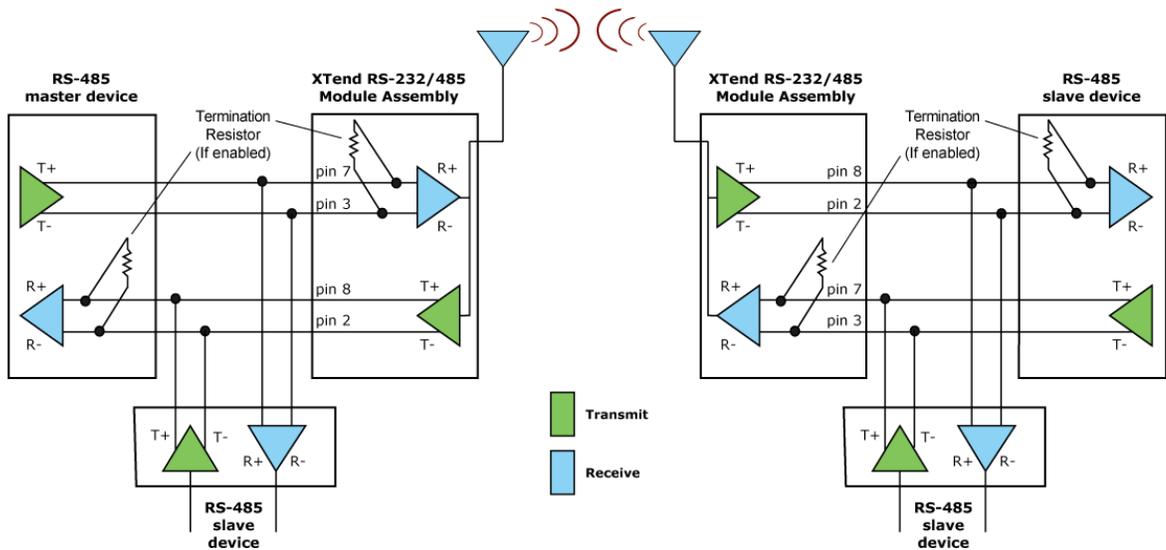
The following table provides the RS-485/422 (4-wire) signals and their implementations on the Module Assembly.

DB-9 pin	RS-485/422 name	Description	Implementation
2	T- (TA)	Transmit negative data line	Serial data sent from the Module Assembly
3	R- (RA)	Receive negative data line	Serial data received by the Module Assembly
5	GND	Signal ground	Ground
7	R+ (RB)	Receive positive data line	Serial data received by the Module Assembly

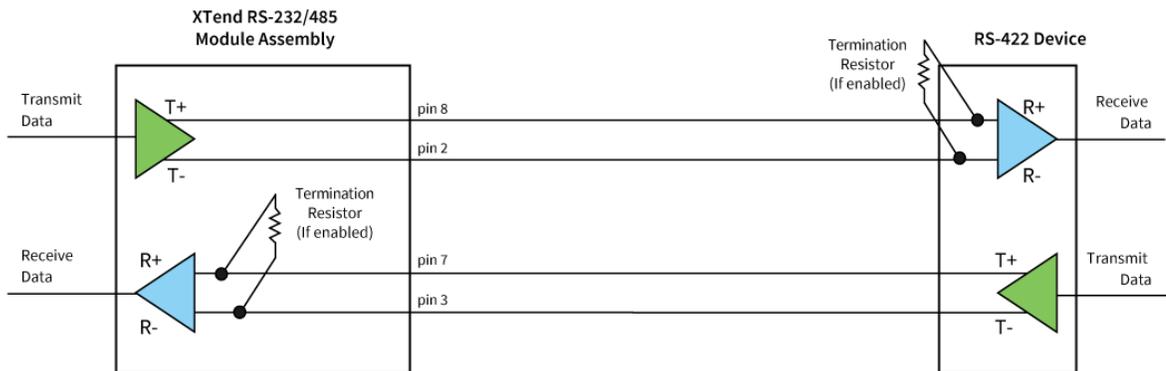
DB-9 pin	RS-485/422 name	Description	Implementation
8	T+ (TB)	Transmit positive data line	Serial data sent from the Module Assembly
9	PWR	Power	Optional power input that is connected internally to the front power connector
1, 4, 6	not used		

**RS-422 wiring diagrams**

The following figure shows the Module Assembly in an RS-485 (4-wire) environment.



The following figure shows the Module Assembly in an RS-422 environment.



**XTend vB RF Module RS-485/422 connection guidelines**

The RS-485/422 protocol provides a solution for wired communications that can tolerate high noise and push signals over long cable lengths. RS-485/422 signals can communicate as far as 4000 feet

(1200 m). RS-232 signals are suitable for cable distances up to 100 feet (30.5 m).

RS-485 offers multi-drop capability in which you can connect up to 32 nodes. Use the RS-422 protocol for point-to-point communications.

To integrate the XTend vB RF Module with the RS-485/422 protocol, we suggest the following:

1. When using Ethernet twisted pair cabling: connect T+ and T- to each wire in a twisted pair. Likewise, connect R+ and R- to a twisted pair. For example, tie the green and white/green wires to T+ and T-.
2. For straight-through Ethernet cable (not cross-over cable), the following wiring pattern works well: Pin 3 to T+, Pin 4 to R+, Pin 5 to R-, Pin 6 to T-.
3. The connecting cable only requires 4 wires, even though there are 8 wires.
4. When using phone cabling (RJ-11), Pin 2 in the cable maps to Pin 3 on the opposite end of cable and Pin 1 maps to Pin 4 respectively.