



# Digi XBee® Cellular

Embedded Modem

---

## User Guide

## Revision history—90001525

---

Revision	Date	Description
A	September 2016	Baseline release of the document.
B	December 2016	Added API functionality. Many changes to reflect the initial production release.

## 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.

© 2017 Digi International Inc. All rights reserved.

## Disclaimers

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

## Warranty

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

[www.digi.com/howtobuy/terms](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 +1 952.912.3444 or visit us at [www.digi.com/support](http://www.digi.com/support).

Support portal login: [www.digi.com/support/eservice](http://www.digi.com/support/eservice)

# Contents

---

## Digi XBee Cellular Embedded Modem User Guide

Applicable firmware and hardware .....	7
--	---

## Getting started with the XBee Cellular Modem Development Kit

Identify the kit contents .....	9
Cellular service .....	9
Connect the hardware .....	10
Configure the device using XCTU .....	11
Add a device .....	11
Update to the latest firmware .....	11
XBIB-U_DEV reference .....	12
XBee Cellular Modem examples .....	13
Check for cellular registration and connection .....	14
Send an SMS message to a phone .....	14
Connect to the ELIZA server .....	16
Connect to the echo server .....	17
Connect to the Daytime server .....	19
Connect to a TCP/IP address .....	21
Perform a (GET) HTTP request .....	22
Configure the XBee Cellular Modem using Device Cloud .....	23
Create a Device Cloud account .....	23
Get the XBee Cellular Modem IMEI number .....	23
Add a Digi XBee to Device Cloud .....	23
Update the firmware .....	24

## Technical specifications

Interface and hardware specifications .....	26
RF characteristics .....	26
Networking and carrier .....	26
Power requirements .....	26
Power consumption .....	27
Electrical specifications .....	27
Regulatory and carrier approvals .....	28

## Hardware

Mechanical drawings .....	30
Pin signals .....	30
Pin connection recommendations .....	31
Antenna connections .....	31

Antenna placement .....	32
SIM card .....	32
The Associate LED .....	32
Heat considerations .....	33
Heat sink considerations .....	33
Power supply considerations .....	34
Add a capacitor to the RESET line .....	34

## Cellular connection process

Connecting .....	37
Cellular network .....	37
Data network connection .....	37
Bypass mode note .....	37
Data communication with remote servers (TCP/UDP) .....	37
Disconnecting .....	38

## Modes

Select an operating mode .....	40
Transparent operating mode .....	41
API operating mode .....	41
Bypass operating mode .....	41
Enter Bypass operating mode .....	41
Leave Bypass operating mode .....	42
Boot the XBee Cellular Modem in Bypass operating mode .....	42
Restore cellular settings to default in Bypass operating mode .....	42
Command mode .....	42
Enter Command mode .....	42
Send AT commands .....	43
Apply command changes .....	43
Make command changes permanent .....	43
Exit Command mode .....	43

## AT commands

Special commands .....	46
AC (Apply Changes) .....	46
FR (Force Reset) .....	46
RE (Restore Defaults) .....	46
WR (Write) .....	46
Cellular commands .....	47
PH (Phone Number) .....	47
S# (ICCID) .....	47
IM (IMEI) .....	47
MN (Operator) .....	47
MV (Modem Firmware Version) .....	48
DB (Cellular Signal Strength) .....	48
Network commands .....	48
IP (IP Protocol) .....	48
TL (SSL/TLS Protocol Version) .....	48
TM (TCP Client Connection Timeout) .....	49
DO (Device Options) .....	49

Addressing commands .....	49
DL (Destination Address) .....	50
P# (Destination Phone Number) .....	50
DE (Destination Port) .....	50
TD (Text Delimiter) .....	50
MY (Module IP Address) .....	50
Serial interfacing commands .....	51
BD (Baud Rate) .....	51
NB (Parity) .....	51
SB (Stop Bits) .....	52
RO (Packetization Timeout) .....	52
FT (Flow Control Threshold) .....	52
AP (API Enable) .....	53
I/O sampling commands .....	53
TP (Temperature) .....	53
Command mode options .....	53
CC (Command Sequence Character) .....	53
CT (Command Mode Timeout) .....	54
GT (Guard Times) .....	54
Firmware version/information commands .....	54
VR (Firmware Version) .....	54
VL (Version Long) .....	55
HV (Hardware Version) .....	55
HS (Hardware Series) .....	55
CK (Configuration CRC) .....	55
Diagnostic interface commands .....	56
DI (Device Cloud Indicator) .....	56
CI (Protocol/Connection Indication) .....	56
Execution commands .....	58
AN (APN) .....	58
NR (Network Reset) .....	58
!R (Modem Reset) .....	58

## Operate in API mode

API frame specifications .....	61
API operation (AP parameter = 1) .....	61
API operation-with escaped characters (AP parameter = 2) .....	61
Start delimiter .....	62
Length .....	62
Frame data .....	62
Checksum .....	63
Calculate and verify checksums .....	63
API frames .....	63
AT Command - 0x08 .....	64
AT Command: Queue Parameter Value - 0x09 .....	65
Transmit (TX) SMS - 0x1F .....	66
Transmit (TX) Request: IPv4 - 0x20 .....	67
AT Command Response - 0x88 .....	68
Transmit (TX) Status - 0x89 .....	69
Modem Status - 0x8A .....	70
Receive (RX) Packet: SMS - 0x9F .....	71
Receive (RX) Packet: IPv4 - 0xB0 .....	72

## Socket behavior

Secure Sockets Layer (SSL) certificate checking .....	74
Socket timeouts .....	74

## Troubleshooting

Cannot find the serial port for the device .....	76
Correct a macOS Java error .....	77
Unresponsive cellular chipset in Bypass mode .....	78
Not on expected network after APN change .....	78

## Certifications

Modification statement .....	80
Interference statement .....	80
Wireless notice .....	80
FCC Class B digital device notice .....	81
Labelling requirements for the host device .....	81

# Digi XBee Cellular Embedded Modem User Guide

---

The XBee Cellular Modem is an embedded Long-Term Evolution (LTE) Category 1 cellular module that provides original equipment manufacturers (OEMs) with a simple way to integrate cellular connectivity into their devices.

The XBee Cellular Modem enables OEMs to quickly integrate cutting edge 4G cellular technology into their devices and applications without dealing with the painful, time-consuming, and expensive FCC and carrier end-device certifications.

With the full suite of standard XBee API frames and AT commands, existing XBee customers can seamlessly transition to this new device with only minor software adjustments. When OEMs add the XBee Cellular Modem to their product, they create a future-proof design with flexibility to switch between wireless protocols or frequencies as needed.

## Applicable firmware and hardware

This manual supports the following firmware:

- 1005

It supports the following hardware:

- XBC-V1-UT-xxx

# Getting started with the XBee Cellular Modem Development Kit

---

This section describes how to connect the hardware in the XBee Cellular Modem Development Kit, and provides some examples you can use to communicate with the device.

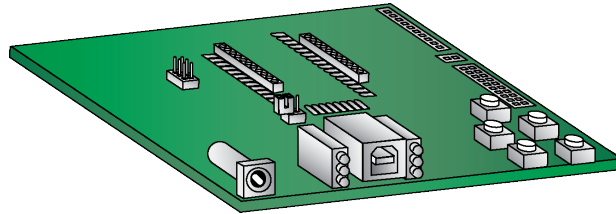
- Identify the kit contents ..... 9
- Cellular service ..... 9
- Connect the hardware .....10
- Configure the device using XCTU .....11
- XBIB-U\_DEV reference .....12
- XBee Cellular Modem examples .....13
- Configure the XBee Cellular Modem using Device Cloud .....23



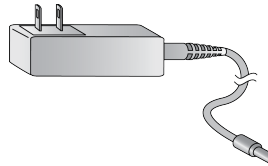
## Identify the kit contents

The Developer's kit includes the following:

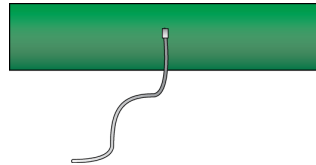
One XBIB-U-DEV board



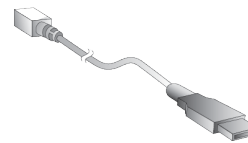
One 12 V power supply



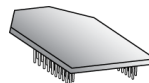
Two antennas with U.FL connectors



One USB cable



One XBee Cellular Modem

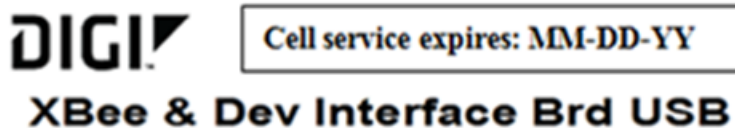


One SIM card



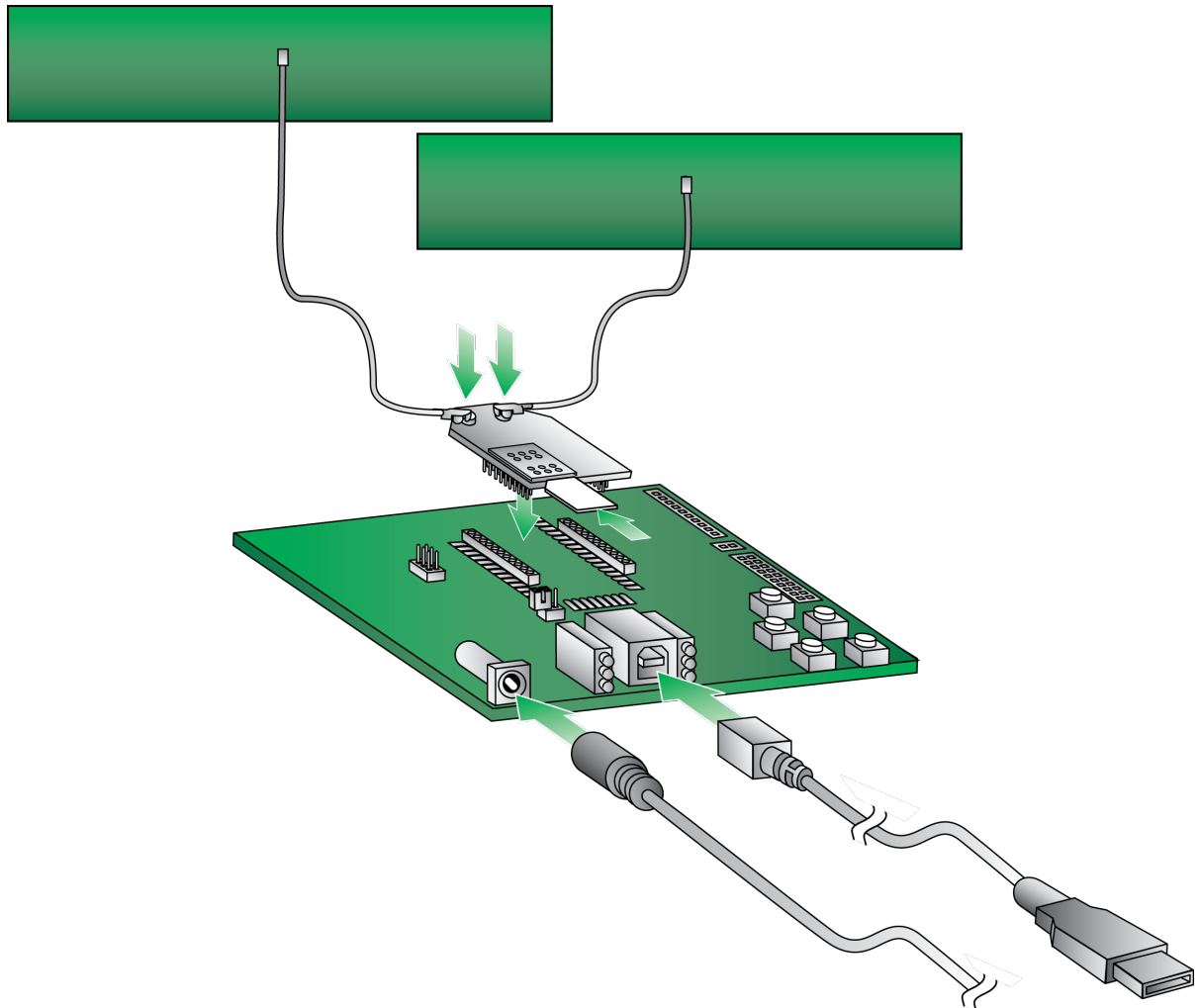
## Cellular service

The XBIB-U-DEV board comes in an anti-static bag. There is a label on the bag that provides the date that cellular service expires for the SIM card in the XBee Cellular Modem.



Note the expiration date and when cellular service expires, contact us at 1.877.890.4014 for further instructions on renewing cellular service for the device.

## Connect the hardware



1. Plug one XBee Cellular Modem into the XBIB-U-DEV board.
2. The SIM card should be inserted into the XBee Cellular Modem already. If not, install the SIM card into the XBee Cellular Modem.

---

**WARNING!** Never insert or remove the SIM card while the device is powered!



- 
3. Connect the antennas to the XBee Cellular Modem by aligning the U.FL connectors carefully, then firmly pressing straight down to seat the connector. U.FL is fragile and is not designed for multiple insertions, so be careful when connecting or removing the antennas.
  4. Plug the 12 V power supply to the power jack on the development board.
  5. Connect the USB cable from a PC to the USB port on the development board. The computer searches for a driver, which can take a few minutes to install.

## Configure the device using XCTU

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

For instructions on downloading and using XCTU, see [the XCTU User Guide](#).



---

**Note** If you are on a macOS computer and encounter problems installing XCTU, see [Correct a macOS Java error](#).

---

### Add a device



These instructions show you how to add the XBee Cellular Modem to XCTU. If XCTU does not find your serial port, see [Cannot find the serial port for the device](#).

1. Launch XCTU .
2. Click the **Discover radio modules** button .
3. In the **Discover radio devices** dialog, select the serial ports where you want to look for XBee modules, and click **Next**.
4. In the **Set port parameters** window, maintain the default values and click **Finish**.
5. As XCTU locates radio modules, they appear in the **Discovering radio modules** dialog box.

If your module could not be found, XCTU displays the **Could not find any radio module** dialog providing possible reasons why the module could not be added.

### Update to the latest firmware

Firmware is the program code stored in the device's persistent memory that provides the control program for the device. Use XCTU to update the firmware.

1. Click the **Configuration working modes** button .
2. Select a local or remote XBee module from the **Radio Modules** list.
3. Click the **Update firmware** button .

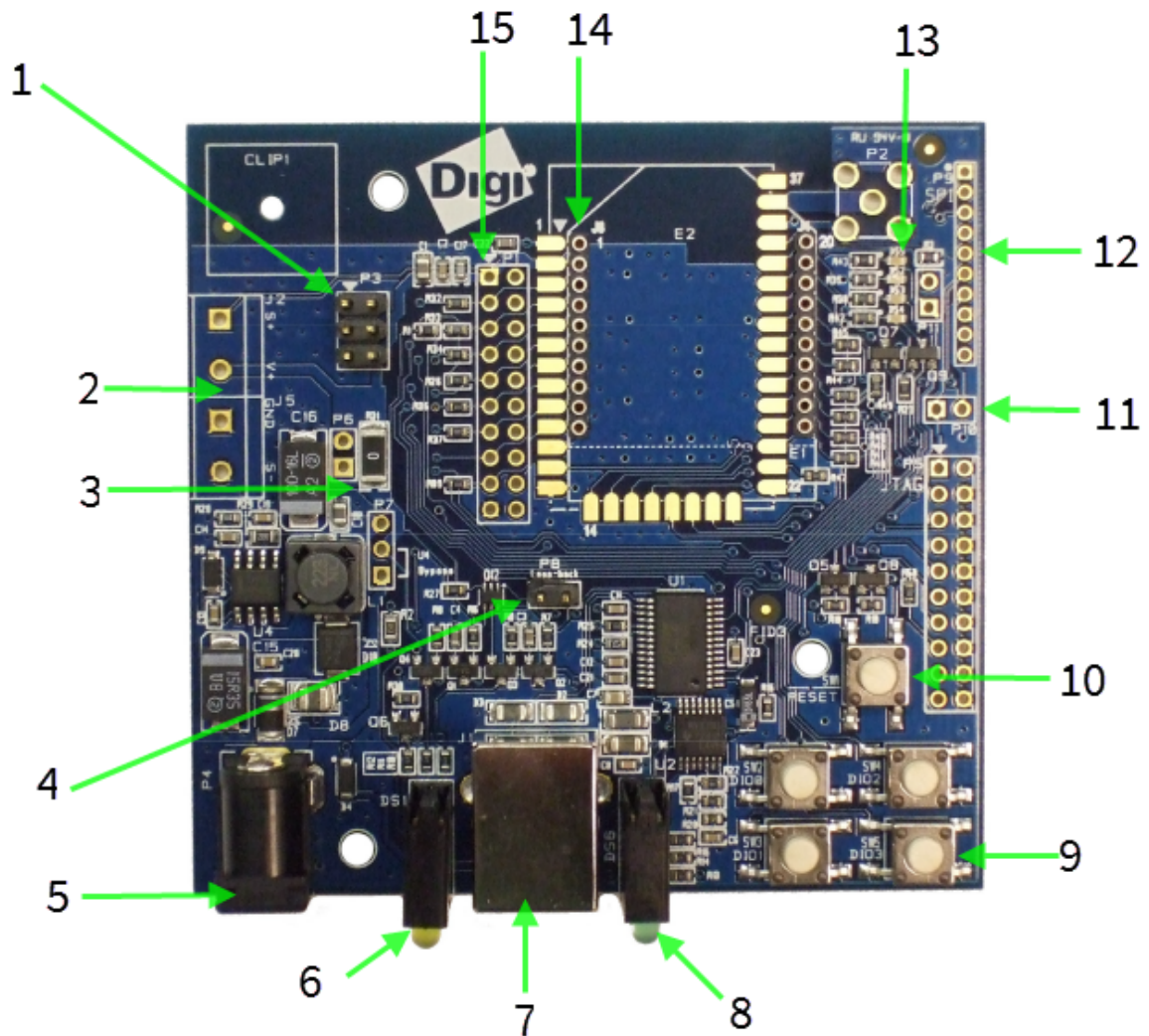
The **Update firmware** dialog displays the available and compatible firmware for the selected XBee module.

4. Select the product family of the XBee module, the function set, and the latest firmware version.
5. Click **Update**. A dialog displays update progress.


See [How to update the firmware of your modules](#) in the XCTU User Guide for more information.

## XBIB-U\_DEV reference

This picture shows the XBIB-U development board and the table that follows explains the callouts in the picture.



Number	Item	Description
1	Programming header	Header used to program XBee Programmable devices.

Number	Item	Description
2	Self power module	Advanced users only—voids the warranty. Depopulate R31 to power the device using V+ and GND from J2 and J5. You can connect sense lines to S+ and S- for sensing power supplies.   CAUTION: Voltage is not regulated. Applying the incorrect voltage can cause fire and serious injury. <sup>1</sup>
3	Current testing	Depopulating R31 allows a current probe to be inserted across P6 terminals. The current through P6/R31 powers the device only. Other supporting circuitry is powered by a different trace.
4	Loopback jumper	Populating P8 with a loopback jumper causes serial transmissions both from the device and from the USB to loopback.
5	DC barrel plug: 6-20 V	Greater than 500 mA loads require a DC supply for correct operation. Plug in the external power supply prior to the USB connector to ensure that proper USB communications are not interrupted.
6	LED indicator	Yellow: Modem sending serial/UART data to host. Green: Modem receiving serial/UART data from host. Red: Associate.
7	USB	
8	RSSI indicator	
9	User buttons	Connected to DIO lines for user implementation.
10	Reset button	
11	SPI power	Connect to the power board from 3.3 V.
12	SPI	Only used for surface-mount devices.
13	Indicator LEDs	DS5: ON/ <u>SLEEP</u> DS2: DIO12, the LED illuminates when driven low. DS3: DIO11, the LED illuminates when driven low. DS4: DIO4, the LED illuminates when driven low.
14	Through-hole XBee sockets	
15	20-pin header	Maps to standard through-hole XBee pins.

## XBee Cellular Modem examples

---


<sup>1</sup>Powering the board with J2 and J5 without R31 removed can cause shorts if the USB or barrel plug power are connected. Applying too high a voltage destroys electronic circuitry in the device and other board components and/or can cause injury.

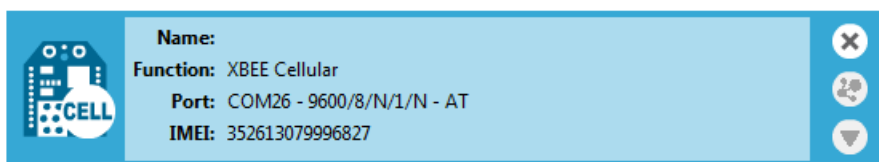
## Check for cellular registration and connection

In the following examples, proper cellular network registration and address assignment must occur successfully. The LED on the development board blinks when the XBee Cellular Modem registers to the cellular network; see [The Associate LED](#). If the LED remains solid, registration has not occurred properly. Registration can take several minutes.


**Note** Make sure you are in an area with adequate cellular network reception or the XBee Cellular Modem will not make the connection.

In addition to the LED confirmation, you can check the AT commands below in XCTU to check the registration and connection. To view these commands:

1. Open XCTU and click the **Configuration working mode**  button.
2. Select a device from the **Radio Modules** list. XCTU displays the current firmware settings for



that device.

3. On the Configuration toolbar, click the **Default** button  to load the default values established by the firmware, and click **Yes** to confirm.

The relevant commands are:

**Note** To search for an AT command in XCTU, use [the search box](#).

- **AI** (Association Indication) reads zero when the device successfully registers to the cellular network. If it reads 0x23 it is connecting to the Internet; 0x22 means it is registering to the cellular network.
- **MY** (Module IP Address) should display a valid IP address. If it reads **0.0.0.0**, it has not registered yet.
- **AN** (APN) should read **vzwinternet** when registration is successful. If **AN** is not on this network, see [Not on expected network after APN change](#).


## Send an SMS message to a phone

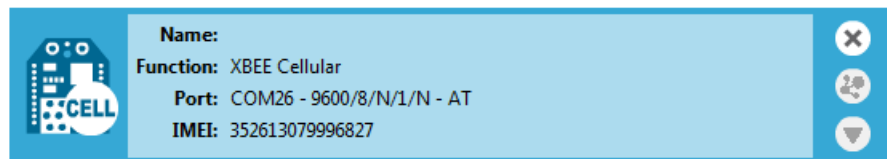
The XBee Cellular Modem can send and receive Short Message Service (SMS) transmissions (texts) while in Transparent mode. This allows you to send and receive text messages to and from an SMS capable device such as a mobile phone.

The following table explains the AT commands that you use in this example.




Command	Value	Description
<b>AP</b> (API Enable)	0	Set the device's API mode to Transparent mode.



Command	Value	Description
<b>IP</b> (IP Protocol)	2	Set the expected transmission mode to SMS communication.
<b>P#</b> (Destination Phone Number)	<Target phone number>	The target phone number that you send to, for example, your cellular phone. See <a href="#">P# (Destination Phone Number)</a> for instructions on using this command.
<b>TD</b> (Text Delimiter)	D (0x0D)	The text delimiter to be used for Transparent mode, as an ASCII hex code. No information is sent until this character is entered, unless the maximum number of characters has been reached. Set to zero to disable text delimiter checking. Set to D for a carriage return.
<b>PH</b> (Module's SIM phone number)	Read only	The value that represents your device's phone number as supplied by the SIM card. This is used to send text messages to the device from another cellular device.

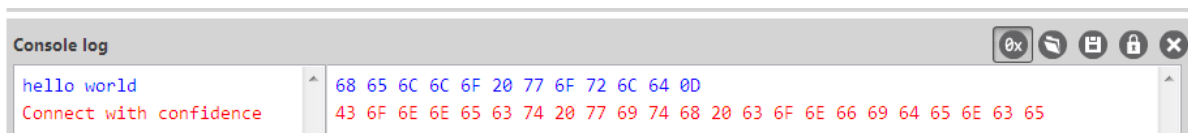
1. Ensure that the device is set up correctly with the SIM card installed and the antennas connected as described in [Connect the hardware](#).
2. Open XCTU and click the **Configuration working mode**  button.
3. Select a device from the **Radio Modules** list. XCTU displays the current firmware settings for



that device.

4. On the Configuration toolbar, click the **Default** button  to load the default values established by the firmware, and click **Yes** to confirm.
5. Factory settings are loaded but not written to the device. To write them, click the **Write** button  on the toolbar.
6. To switch to SMS communication, in the **IP** field, select 2 and click the **Write** button .
7. To enter your cell phone number, in the **P#** field, type the <**target phone number**> and click the **Write** button. Type the phone number using only numbers, with no dashes. You can use the + prefix if necessary. The target phone number is the phone number you wish to send a text to.
8. In the **TD** field, type **D** and click the **Write** button.
9. Note the number in the **PH** field; it is the XBee Cellular Modem phone number, which you see when it sends an SMS to your phone.

10. Click the **Consoles working mode** button  on the toolbar to open a serial console to the device. For instructions on using the Console, see [http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#reference/r\\_at\\_console.htm](http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#reference/r_at_console.htm)
11. Click the **Open** button  to open a serial connection to the device.
12. Click in the left pane of the **Console log**, type **hello world** and press **Enter**. The XBee Cellular Modem sends the message to the destination phone number set by the **P#** command.
13. When the phone receives the text, you can see that the sender's phone number matches the value reported by the XBee Cellular Modem with the **PH** command.
14. On the phone, reply with the text **connect with confidence** and the XBee Cellular Modem outputs this reply from the UART.




## Connect to the ELIZA server

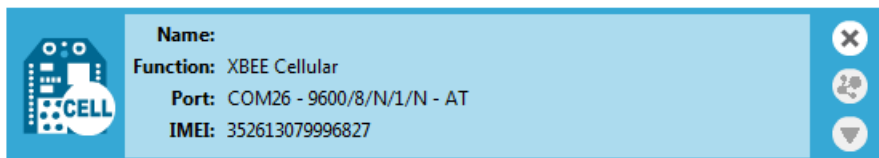
You can use the XBee Cellular Modem to chat with the ELIZA Therapist Bot. ELIZA is an artificial intelligence (AI) bot that emulates a therapist and can perform simple conversations.

The following table explains the AT commands that you use in this example.

At command	Value	Description
<b>IP</b> (IP Protocol)	1	Set the expected transmission mode to TCP communications
<b>DL</b> (Destination Address)	52.43.121.77	The target IP address of the Eliza server
<b>DE</b> (Destination Port)	0x2328	The target port number of the Eliza server






To communicate with the ELIZA Therapist Bot:

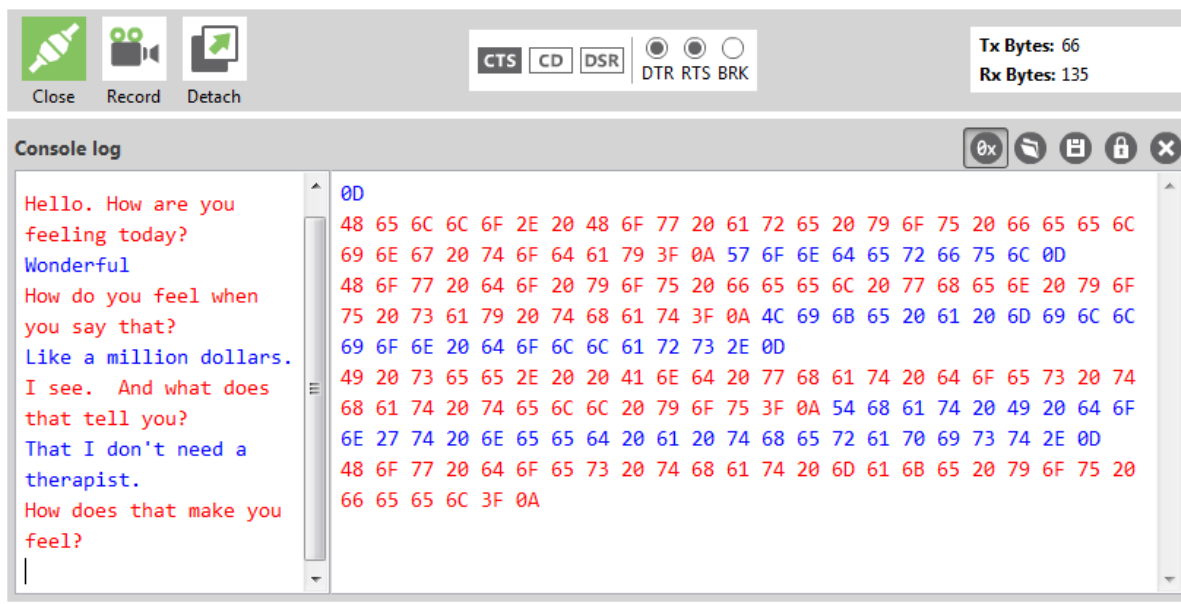
1. Ensure that the device is set up correctly with the SIM card installed and the antennas connected as described in [Connect the hardware](#).
2. Open XCTU and click the **Configuration working mode**  button.
3. Select a device from the **Radio Modules** list. XCTU displays the current firmware settings for



that device.



4. On the Configuration toolbar, click the **Default** button  to load the default values established by the firmware, and click **Yes** to confirm.
5. Factory settings are loaded but not written to the device. To write them, click the **Write** button  on the toolbar.
6. To switch to TCP communication, in the **IP** field, select 1 and click the **Write** button .
7. To enter the destination address of the ELIZA Therapist Bot, in the **DL** field, type **52.43.121.77** and click the **Write** button.
8. To enter the destination IP port number, in the **DE** field, type **2328** and click the **Write** button.
9. Click the **Consoles working mode** button  on the toolbar to open a serial console to the device. For instructions on using the Console, see [http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#reference/r\\_at\\_console.htm](http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#reference/r_at_console.htm)
10. Click the **Open** button  to open a serial connection to the device.
11. Click in the left pane of the **Console log**, then type in the Console to talk to the ELIZA Therapist Bot. The following screenshot provides an example of this chat.




## Connect to the echo server

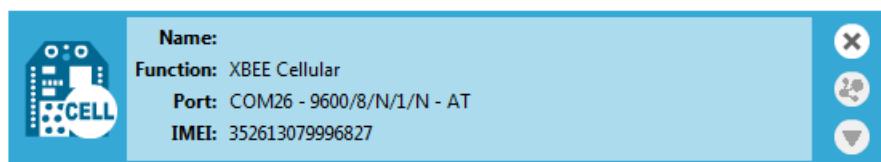
This server echoes back the messages you type.

The following table explains the AT commands that you use in this example.





At command	Value	Description
<b>IP</b> (IP Protocol)	1	Set the expected transmission mode to TCP communications
<b>TD</b> (Text Delimiter)	D (0x0D)	The text delimiter to be used for Transparent mode, as an ASCII hex code. No information is sent until this character is entered, unless the maximum number of characters has been reached. Set to zero to disable text delimiter checking. Set to D for a carriage return.
<b>DL</b> (Destination Address)	52.43.121.77	The target IP address of the echo server
<b>DE</b> (Destination Port)	0x2329	The target port number of the echo server


To communicate with the echo server:

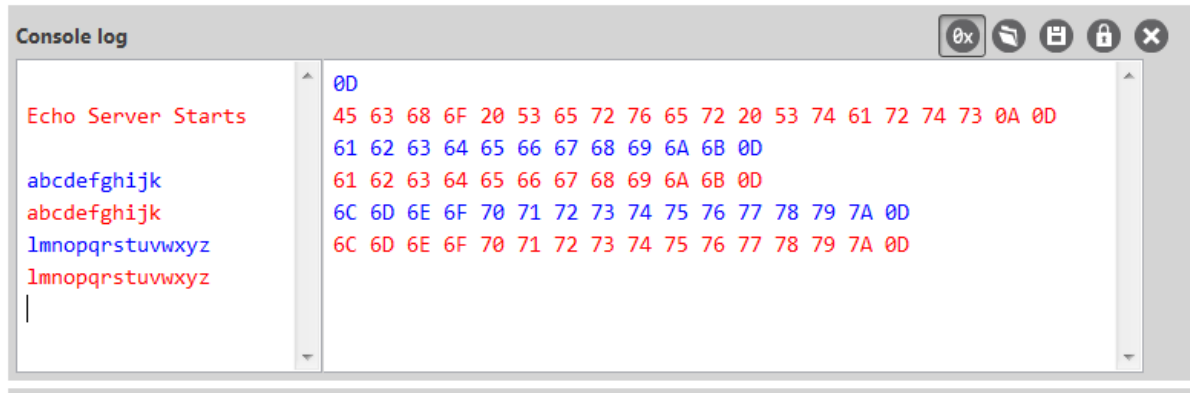
1. Ensure that the device is set up correctly with the SIM card installed and the antennas connected as described in [Connect the hardware](#).
2. Open XCTU and click the **Configuration working mode**  button.
3. Select a device from the **Radio Modules** list. XCTU displays the current firmware settings for



that device.

4. On the Configuration toolbar, click the **Default** button  to load the default values established by the firmware, and click **Yes** to confirm.
5. Factory settings are loaded but not written to the device. To write them, click the **Write** button  on the toolbar.
6. To switch to TCP communication, in the **IP** field, select 1 and click the **Write** button .
7. To enable the XBee Cellular Modem to recognize carriage return as a message delimiter, in the **TD** field, type **D** and click the **Write** button.
8. To enter the destination address of the echo server, in the **DL** field, type **52.43.121.77** and click the **Write** button.
9. To enter the destination IP port number, in the **DE** field, type **2329** and click the **Write** button.
10. Click the **Consoles working mode** button  on the toolbar to open a serial console to the device. For instructions on using the Console, see [http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#reference/r\\_at\\_console.htm](http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#reference/r_at_console.htm)

11. Click the **Open** button  to open a serial connection to the device.
12. Click in the left pane of the **Console log**, then type in the Console to talk to the echo server.  
The following screenshot provides an example of this chat.




## Connect to the Daytime server

The Daytime server reports the current Coordinated Universal Time (UTC) value responding to any user input.

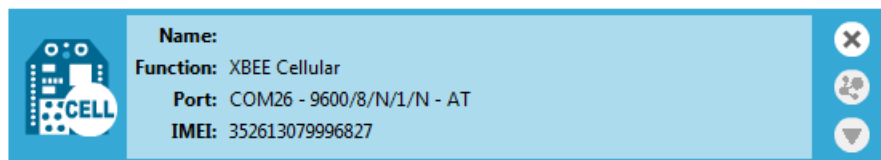
The following table explains the AT commands that you use in this example.

At command	Value	Description
<b>IP</b> (IP Protocol)	1	Set the expected transmission mode to TCP communications
<b>DL</b> (Destination Address)	52.43.121.77	The target IP of the Daytime server
<b>DE</b> (Destination Port)	0x232A	The target port number of the Daytime server
<b>TD</b> (Text Delimiter)	0	The text delimiter to be used for Transparent mode, as an ASCII hex code. No information is sent until this character is entered, unless the maximum number of characters has been reached. Set to zero to disable text delimiter checking.






To communicate with the Daytime server:

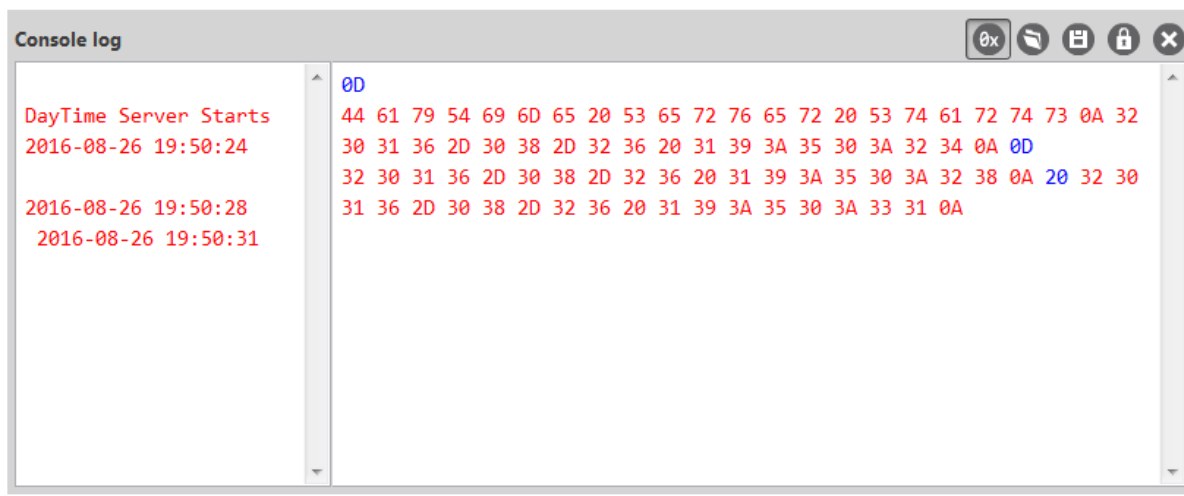
1. Ensure that the device is set up correctly with the SIM card installed and the antennas connected as described in [Connect the hardware](#).
2. Open XCTU and click the **Configuration working mode**  button.

3. Select a device from the **Radio Modules** list. XCTU displays the current firmware settings for



that device.

4. On the Configuration toolbar, click the **Default** button  to load the default values established by the firmware, and click **Yes** to confirm.
5. Factory settings are loaded but not written to the device. To write them, click the **Write** button  on the toolbar.
6. To switch to TCP communication, in the **IP** field, select 1 and click the **Write** button .
7. To enter the destination address of the daytime server, in the **DL** field, type **52.43.121.77** and click the **Write** button.
8. To enter the destination IP port number, in the **DE** field, type **232A** and click the **Write** button.
9. To disable text delimiter checking, in the **TD** field, type **0** and click the **Write** button.
10. Click the **Consoles working mode** button  on the toolbar to open a serial console to the device. For instructions on using the Console, see [http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#reference/r\\_at\\_console.htm](http://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#reference/r_at_console.htm)
11. Click the **Open** button  to open a serial connection to the device.
12. Click in the left pane of the **Console log**, then type in the Console to query the Daytime server. The following screenshot provides an example of this chat.




## Connect to a TCP/IP address

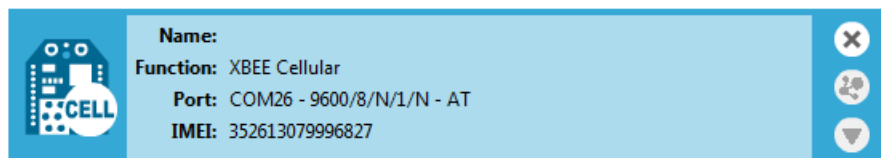
The XBee Cellular Modem can send and receive TCP messages while in Transparent mode; see [Transparent operating mode](#).

You can use this example as a template for sending and receiving data from a user. The following table explains the AT commands that you use in this example.




Command	Value	Description
<b>IP</b> (IP Protocol)	1	Set the expected transmission mode to TCP communication.
<b>DL</b> (Destination IP Address)	<Target IP address>	The target IP address that you send and receive from. For example, a data logging server's IP address that you want to send measurements to.
<b>DE</b> (Destination Port)	<Target port number>	The target port number that the device sends the transmission to. This is represented as a hexadecimal value.

To connect to a TCP/IP address:

1. Ensure that the device is set up correctly with the SIM card installed and the antennas connected as described in [Connect the hardware](#).
2. Open XCTU and click the **Configuration working mode**  button.
3. Select a device from the **Radio Modules** list. XCTU displays the current firmware settings for



that device.





4. On the Configuration toolbar, click the **Default** button  to load the default values established by the firmware, and click **Yes** to confirm.
5. Factory settings are loaded but not written to the device. To write them, click the **Write** button  on the toolbar.
6. In the **IP** field, select 1 and click the **Write** button .
7. In the **DL** field, type the <target IP address> and click the **Write** button. The target IP address is the IP address that you send and receive from.
8. In the **DE** field, type the <target port number>, converted to hexadecimal, and click the **Write** button.
9. Exit Command mode; see [Exit Command mode](#).

After exiting Command mode, any UART data sent to the device is sent to the destination IP address and port number after the **RO** (Packetization Timeout) occurs; see [RO \(Packetization Timeout\)](#).

## Perform a (GET) HTTP request

You can use the XBee Cellular Modem to perform a GET Hypertext Transfer Protocol (HTTP) request using XCTU. This example uses <http://httpbin.org/> (IP address: 54.175.219.8) as the target website that responds to the HTTP request.


To perform a GET request:

1. Ensure that the device is set up correctly with the SIM card installed and the antennas connected as described in [Connect the hardware](#).
2. Open XCTU and click the **Configuration working mode**  button.
3. Select a device from the **Radio Modules** list. XCTU displays the current firmware settings for that device.
4. To enter the destination address of the target website, in the **DL** field, type **54.175.219.8** and click the **Write** button .
5. To enter the HTTP request port number, in the **DE** field, type **50** and click the **Write** button. Hexadecimal **50** is 80 in decimal.
6. To switch to TCP communication, in the **IP** field, select **1** and click the **Write** button.
7. To move into Transparent mode, in the **AP** field, select **0** and click the **Write** button.
8. Wait for the **AI** (Association Indication) value to change to **0** (Connected to the Internet).
9. Click the **Consoles working mode** button  on the toolbar.
10. From the AT console, click the **Add new packet button**  in the Send packets dialog. The **Add new packet** dialog appears.
11. Enter the name of the data packet.
12. Type the following data in the **ASCII** input tab:  
GET /ip HTTP/1.1  
Host: httpbin.org
13. Click the **HEX** input tab and add **0A** (zero A) after each **0D** (zero D), and add an additional **0D 0A** at the end of the message body. For example, copy and past the following text into the **HEX** input tab:  
47 45 54 20 2F 69 70 20 48 54 54 50 2F 31 2E 31 0D 0A 48 6F 73 74 3A 20 68 74 74 70 62 69 6E  
2E 6F 72 67 0D 0A 0D 0A

---

**Note** The HTTP protocol requires an empty line (a line with nothing preceding the CRLF) to terminate the request.

---

14. Click **Add packet**.
15. Click the **Open** button .
16. Click **Send selected packet**.
17. A GET HTTP response from httpbin.org appears in the Console log.

## Configure the XBee Cellular Modem using Device Cloud

### Create a Device Cloud account

Digi Device Cloud is an on-demand service with no infrastructure requirements. Remote devices and enterprise business applications connect to Device Cloud through standards-based web services. This section describes how to configure and manage an XBee using Device Cloud. For detailed information on using Device Cloud, refer to the Device Cloud User Guide, available via the Documentation tab in Device Cloud.

Before you can manage an XBee with Device Cloud, you must create a Device Cloud account. To create a Device Cloud account:

1. Go to [www.digi.com/cloud/digi-device-cloud](http://www.digi.com/cloud/digi-device-cloud).
2. Click **Free Developer Account**.
3. Follow the online instructions to complete account registration. You can upgrade your Developer account to a paid account at any time.

When you are ready to deploy multiple XBee Cellular Modems in the field, upgrade your account to access additional Device Cloud features.

### Get the XBee Cellular Modem IMEI number

Before adding an XBee to your Device Cloud account inventory, you need to determine the International Mobile Equipment Identity (IMEI) number for the device. Use XCTU to view the IMEI number by querying the **IM** parameter.

### Add a Digi XBee to Device Cloud

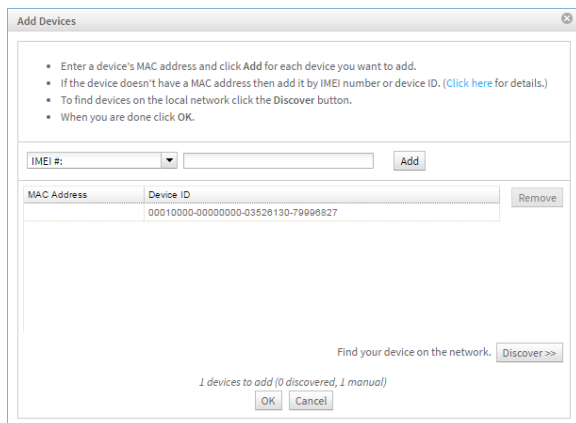
To add an XBee to your Device Cloud account inventory, follow these steps:

Go to [devicecloud.digi.com](http://devicecloud.digi.com).

1. Log in to your account
2. Click Device **Management > Devices**.
3. Click **Add Devices**. The Add Devices dialog appears.

You can use the **Discover** button to discover an XBee device if the XBee is connected to your local network.

4. Select **IMEI #**, and type or paste the IMEI number of the XBee you want to add. The **IM** command provides this number.



5. Click **Add** to add the device. The XBee is added to your inventory.
6. Click **OK** to close the Add Devices dialog and return to the Devices view.

## Update the firmware

XBee Cellular Modem supports Device Cloud firmware updates. To perform a firmware update, use the following steps.

1. Download the updated firmware file for your device from [Digi's support site](#). This is a zip file containing .ebin and .mxi files for import.
2. Unzip the file.
3. In your Device Cloud account, click **Device Management > Devices**.
4. Select the first device you want to update.
5. To select multiple devices (must be of the same type), press the Control key and select additional devices.
6. Click **More** in the Devices toolbar and select **Update Firmware** from the Update category of the More menu. The Update Firmware dialog appears.
7. Click **Browse** to select the .ebin file that you unzipped earlier.
8. Click **Update Firmware**. The updated devices automatically reboot when the updates are complete.



# Technical specifications

---

- Interface and hardware specifications .....26
- RF characteristics .....26
- Networking and carrier .....26
- Power requirements .....26
- Power consumption .....27
- Electrical specifications .....27
- Regulatory and carrier approvals .....28

## Interface and hardware specifications

The following table provides the interface and hardware specifications for the device.

Specification	Value
Dimensions	2.438 x 3.294 cm (0.960 x 1.297 in)
Weight	5 g (0.18 oz)
Operating temperature	-40 to +80 °C
Antenna connector	U.FL for primary and secondary antennas
Digital I/O	13 I/O lines
ADC	4 10-bit analog inputs

## RF characteristics

The following table provides the RF characteristics for the device.

Specification	Value
Modulation	LTE/4G – QPSK, 16 QAM
Transmit power	23 dBm
Receive sensitivity	-102 dBm
Over-the-air maximum data rate	10 Mb/s
Data throughput	TBD

## Networking and carrier

The following table provides the networking and carrier specifications for the device.

Specification	Value
Addressing options	TCP/IP and SMS
Carrier and technology	Verizon 4G LTE Cat 1
Supported bands	4 and 13
Security	SSL/TLS

## Power requirements

The following table provides the power requirements for the device.

Specification	Value
Supply voltage	3.0 to 5.5 V
Extended voltage range	2.7 to 5.5 VDC

## Power consumption

The peak current was measured from multiple tested units.

Specification	State	Average current	Measured peak current
Tx+RX current	Active transmit, 23 dBm @ 3.3 V	860 mA	1020 mA
Tx+RX current	Active transmit, 23 dBm @ 5.0 V	555 mA	630 mA
TX Only current	Active transmit, 23 dBm @ 3.3 V	680 mA	N/A
Rx + ACK current	Active receive @ 3.3 V	530 mA	N/A
Rx + ACK current	Active receive @ 5 V	360 mA	N/A
RX Only current	Active receive @ 3.3 V	300 mA	N/A
Idle current	Idle/connected, listening @ 3.3V	143 mA	N/A
Idle current	Idle/connected, listening @ 5 V	100 mA	N/A

## Electrical specifications

Symbol	Parameter	Condition	Min	Typical	Max	Units
VCCMAX	Maximum limits of VCC line		0		5.5	V
VDD_IO	Supply voltage for I/O	While in deep sleep and during initial power up	Min (VCC-0.3, 3.3)		3.3	V
VDD_IO	Supply voltage for I/O	In normal running mode		3.3 V		
VI	Voltage on any pin		-0.3		VDD_IO + 0.3	V
VIL	Input low voltage				0.3*VDD_IO	
VIH	Input high voltage		0.7*VDD_IO			
VOL	Voltage output low	Sinking 6 mA VDD_IO = 3.3 V			0.2*VDD_IO	

Symbol	Parameter	Condition	Min	Typical	Max	Units
VOH	Voltage output high	Sourcing 6 mA VDD_IO = 3.3 V	0.75*VDD_IO			
II_IN	Input leakage current	High Z state I/O connected to Ground or VDD_IO		0.1	100	nA
RPU	Internal pull-up resistor	Enabled		40		kΩ
RPD	Internal pull-down resistor	Enabled		40		kΩ

## Regulatory and carrier approvals

The following table provides the regulatory and carrier approvals for the device.

Specification	Value
United States	Contains FCC ID: RI7LE866SV1
Industry Canada	Contains IC: 5131A-LE866SV1
Europe (CE)	N/A
RoHS	Lead-free and RoHS compliant
Australia	N/A

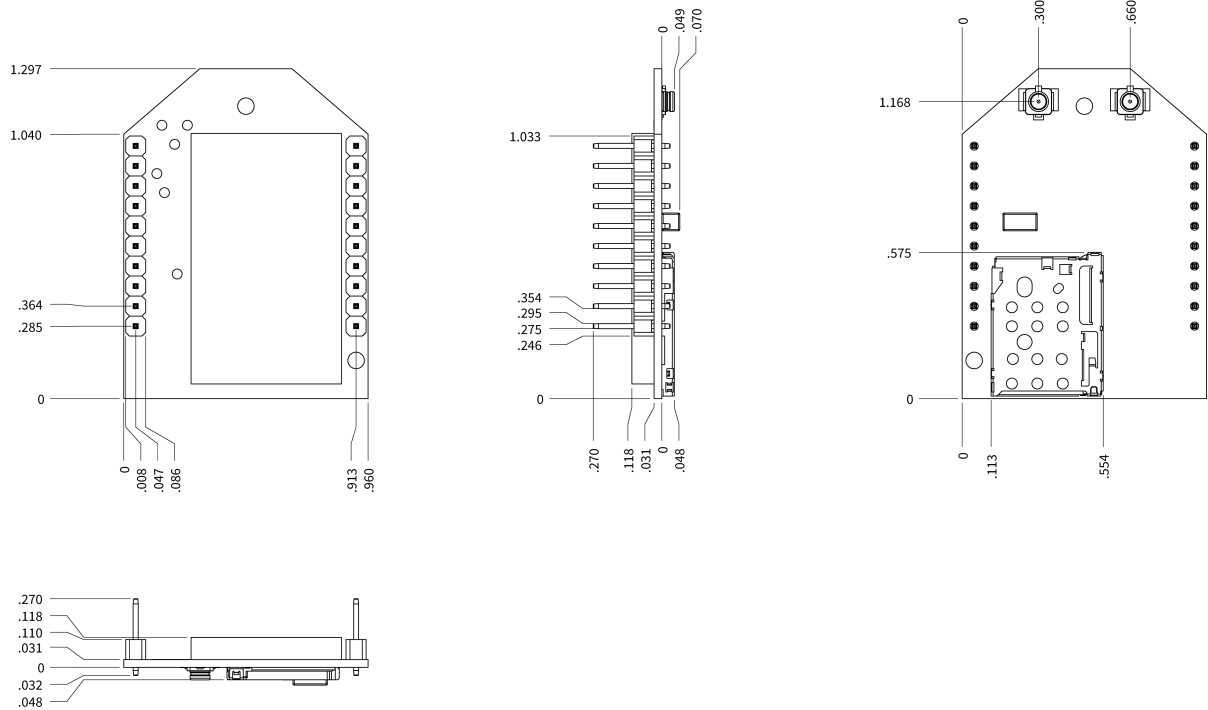
## Hardware

---

Mechanical drawings .....	30
Pin signals .....	30
Antenna connections .....	31
Antenna placement .....	32
SIM card .....	32
The Associate LED .....	32
Heat considerations .....	33
Heat sink considerations .....	33
Power supply considerations .....	34
Add a capacitor to the RESET line .....	34

## Mechanical drawings

The following figures show the mechanical drawings for the XBee Cellular Modem. The drawings do not show antenna options. All dimensions are in inches.



## Pin signals

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

Pin	Name	Direction	Default	Description
Pin	Name	Direction	Default	Description
1	V <sub>CC</sub>			Power supply
2	DOUT	Output	Output	UART Data Out
3	DIN / <u>CONFIG</u>	Input	Input	UART Data In
4	DIO12	Either		Digital I/O 12
5	<u>RESET</u>	Input		
6	PWM0 / RSSI / DIO10	Either	Output	PWM Output 0 / RX Signal Strength Indicator / Digital I/O10

Pin	Name	Direction	Default	Description
7	PWM1 / DIO11	Either		PWM Output 1 / Digital I/O 11
8	[reserved]			Do not connect
9	$\overline{\text{DTR}}$ / SLEEP_RQ/ DIO8			Pin Sleep Control Line or Digital I/O 8
10	GND			Ground
11	DIO4	Either		Digital I/O 4
12	$\overline{\text{CTS}}$ / DIO7	Either	Output	Output Clear-to-Send Flow Control or Digital I/O 7
13	ON / $\overline{\text{SLEEP}}$	Output	Output	Module Status Indicator or Digital I/O 9
14	VREF	Unused		This line is not connected. It is used on other XBee products.
15	Associate / DIO5	Either	Output	Associated Indicator, Digital I/O 5
16	$\overline{\text{RTS}}$ / DIO6	Either	Disabled	Input Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Disabled	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Disabled	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Disabled	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Input	Analog Input 0, Digital I/O 0

## Pin connection recommendations

The only required pin connections are VCC, GND, DIN, DOUT,  $\overline{\text{RTS}}$ ,  $\overline{\text{DTR}}$  and  $\overline{\text{RESET}}$ . Firmware updates require access to these pins.

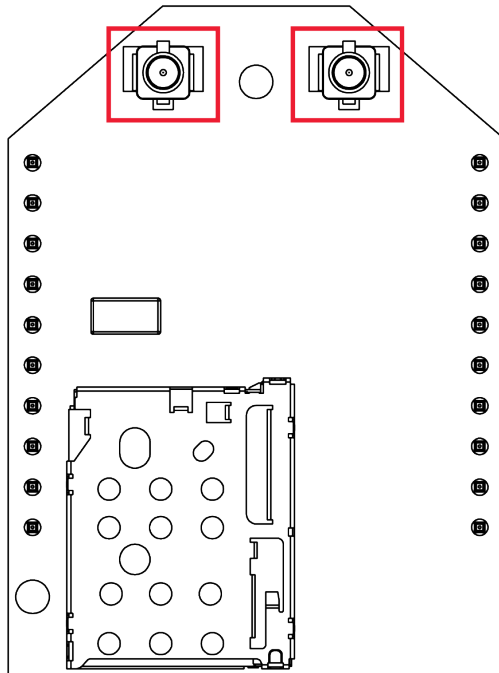
## Antenna connections



**CAUTION!** The XBee Cellular Modem will not function properly with only the secondary antenna port connected!

The XBee Cellular Modem has two U.FL antenna ports; a primary on the upper left of the board and a secondary port on the upper right, see the drawing below. You must connect the primary port and the

secondary port is optional. The secondary antenna improves receive performance in certain situations, so we recommend it for best results.



## Antenna placement

It is important to keep the antenna as far away from the XBee Cellular Modem and other metal objects as possible. Often, small antennas are desirable, but at the cost of increasing size of dead zones because of reduced range and efficiency.

We recommend that antennas do not touch each other, but the XBee Cellular Modem works if they do. To optimize receive performance, orient the two antennas at right angles to each other.

## SIM card

The XBee Cellular Modem uses the 4FF/nano SIM card size.

---

**CAUTION!** Never insert or remove SIM card while the power is on!

---



## The Associate LED

The following table describes the Associate LED functionality. For the location of the Associate LED on the XBIB-U development board, see number 6 on the [XBIB-U\\_DEV reference](#).



LED status	Blink timing	Meaning
On, solid		Not joined to a mobile network
Double blink	½ second	The last TCP/UDP/SMS attempt failed
Standard single blink	1 second	Normal operation

Error conditions cause the LED to blink at ½ second. If it is blinking at ½ second, you may need to check [DI \(Device Cloud Indicator\)](#) or [CI \(Protocol/Connection Indication\)](#) for the cause of the error.

## Heat considerations

Depending on the use case, your application may require a heat sink. Use a non-conductive thermal gasket to make contact to the heat sink. The gasket should be thick enough to ensure that contact with the tallest component does not cause damage when pressure is applied to a secure heat sink.

We recommend connecting the heat sink to the top side of the XBee Cellular Modem without covering the U.FL connectors.

We do not recommend operating the unit above 80 °C.

The operation temperature of the unit can be approximated for different current draws and extreme cases by the equation below. For best results, attach a temperature probe just above the center of the XBee Cellular Modem. Alternatively use the AT command **TP** to query the current temperature of the device's processor.

$$MaxOperatingAmbientTemp^{\circ}C = 80^{\circ}C - (XBeeBoardTemp - RoomTemp) * ScenarioMaxCurrent / AverageCurrentDuringTest$$

Where

1. *XBeeBoardTemp* is the temperature of the XBee Cellular Modem at steady state.
  - a. Use the **TP** command to help estimate the temperature when attaching a temperature probe is not practical, but for reliable results, you must use a temperature probe.
2. *RoomTemp* is the temperature of the ambient air.
3. *AverageCurrentDuringTest* is the average current measured during test.
4. *ScenarioMaxCurrent* is the maximum current expected for the device.

## Heat sink considerations

The following table provides a set of typical heat sink examples for scenarios that customers may encounter.

The results may vary by implementation and scenario. You should always perform sufficient testing to ensure that the XBee Cellular Modem does not exceed temperature specifications.

Scenario	3.3 V average current	Scenario example	Peak MB consumed per hour	Maximum ambient temperature		
				No heat sink	Heat sink	Heat sink and fan
Maximum current	950 mA	Running video camera	200 to 2000	NR <sup>1</sup>	31 °C	70 °C
50% current	475 mA	Running low-resolution camera	80 to 500	42 °C	55 °C	75 °C
20% current	200 mA	Sending 2 MB picture of traffic every minute	10 to 100	64 °C	70 °C	78 °C
Device available	170 mA	Updating traffic sign	1 to 10	66 °C	71 °C	78 °C
Sleep current scenario—high latency	20 mA	Wakes at most once per hour to send and/or receive data	Less than 0.1	78 °C	78 °C	78 °C

## Power supply considerations

When considering a power supply, use the following design practices.

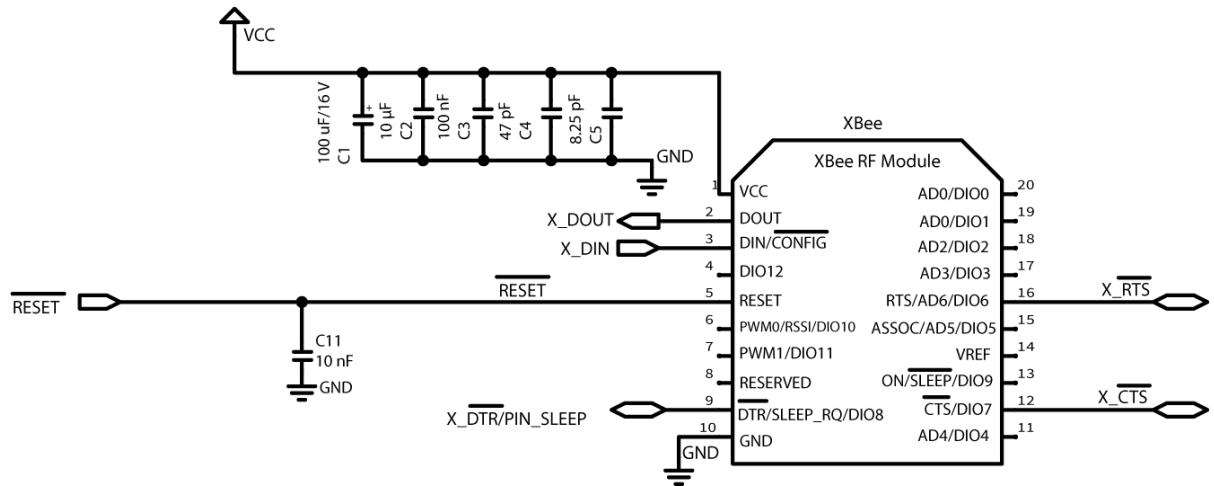
1. Keep power supply ripple below 75 mV.
2. The supply should handle 1.5 A.
3. Place sufficient bulk capacitance on the VCC to maintain voltage above the minimum specification during inrush current. Inrush current is about 2 A during initial power up of cellular communications and during cellular restart conditions.
4. Place smaller high frequency ceramic capacitors very close to the XBee Cellular Modem VCC pin to decrease high frequency noise.
5. Power supplies traces must be thick enough to handle the current requirements with minimal voltage drop when peak currents are reached. Lower voltages require more current to achieve the same power output. Supply voltage steady state at VCC pin 1 should change by less than 0.1 V if drawing 0.5 W compared to drawing 2.8 W.

## Add a capacitor to the RESET line

In high EMI noise environments, we recommend adding a 10 nF ceramic capacitor very close to pin 5.

---

<sup>1</sup>NR = not recommended.



# Cellular connection process

---

- Connecting .....37
- Cellular network .....37
- Data network connection .....37
- Bypass mode note .....37
- Data communication with remote servers (TCP/UDP) .....37
- Disconnecting .....38

## Connecting

In normal operations, the XBee Cellular Modem automatically attempts both a cellular network connection and a data network connection on power-up. The sequence of these connections is as follows:

### Cellular network

1. The device powers on.
2. It looks for cellular towers.
3. It chooses a candidate tower with the strongest signal.
4. It negotiates a connection.
5. It completes cellular registration; the phone number and SMS are available.

### Data network connection

1. The network enables the evolved packet system (EPS) bearer with an access point name (APN).
2. The device negotiates a data connection with the access point.
3. The device receives its IP configuration and address.
4. The [AI \(Association Indication\)](#) command now returns a **0** and the sockets become available.

### Bypass mode note

When the XBee Cellular Modem starts up in Bypass mode, a cellular network connection is attempted by default, but you must request all other operations manually. Bypass mode is intended for advanced users with special needs only.

### Data communication with remote servers (TCP/UDP)

Once sockets become available, communication with remote servers can be initiated in several ways:

- Transparent mode data received over a serial connection (see [TD \(Text Delimiter\)](#) and [RO \(Packetization Timeout\)](#) for timing).
- API mode: [Transmit \(TX\) Request: IPv4 - 0x20](#) received over the serial connection.
- Digi Remote Manager connectivity begins.

Data communication begins when:

1. A socket opens to the remote server.
2. Data is sent.
3. The XBee Cellular Modem listens for incoming data.

Data connectivity ends when:

1. The server closes the connection.
2. The **TM** timeout expires (see [TM \(TCP Client Connection Timeout\)](#)).

3. The cellular network may also close the connection after a timeout set by the network operator.

## **Disconnecting**

When the XBee Cellular Modem is put into Airplane mode or deep sleep is requested:

1. Sockets are closed, cleanly if possible.
2. The EPS bearer data connection is shut down.
3. The cellular connection is shut down.
4. The cellular component is powered off.

# Modes

---

- Select an operating mode ..... 40
- Transparent operating mode ..... 41
- API operating mode ..... 41
- Bypass operating mode ..... 41
- Command mode ..... 42

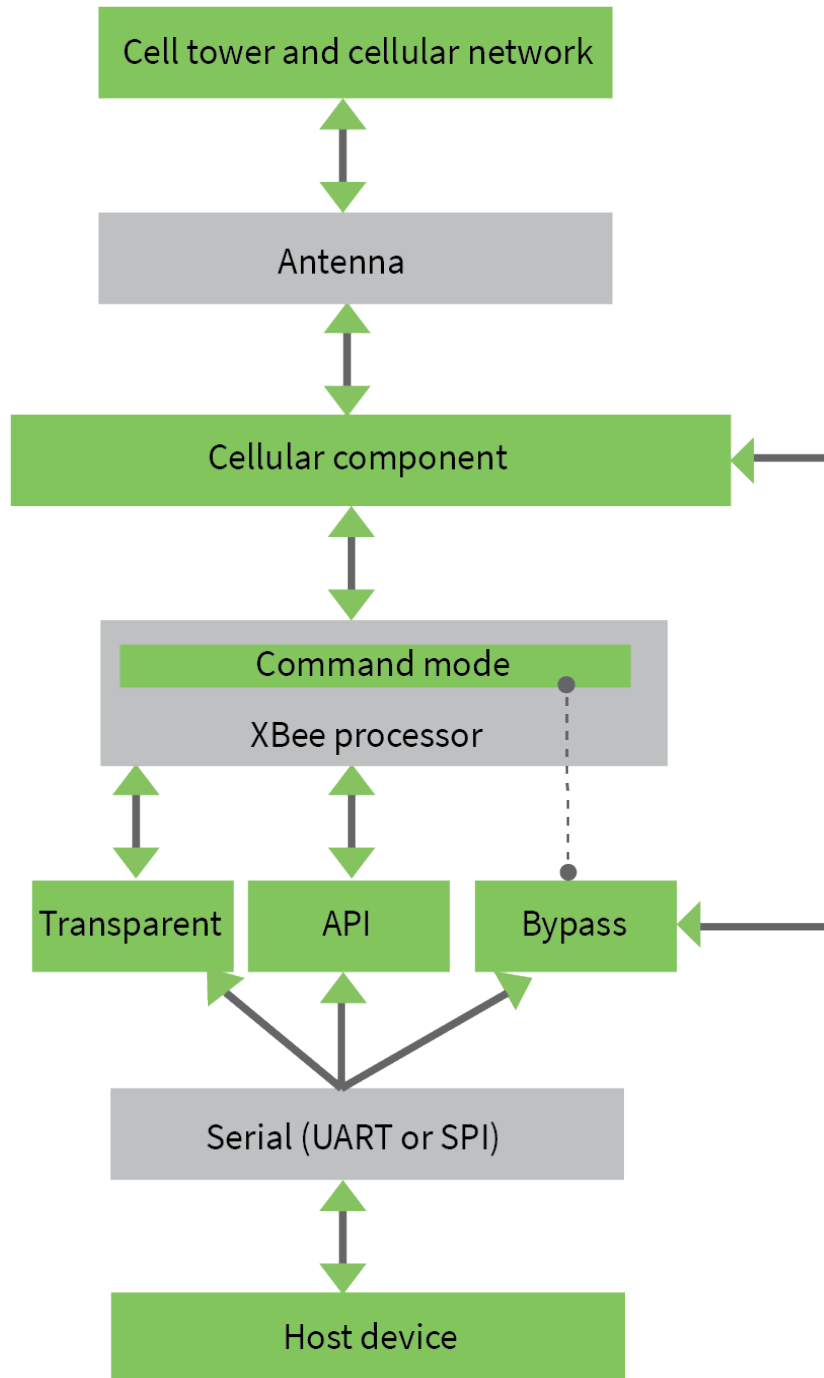
## Select an operating mode

The XBee Cellular Modem interfaces to a host device such as a microcontroller or computer through a logic-level asynchronous serial port. It uses a [UART](#) for serial communication with those devices.

The XBee Cellular Modem supports three operating modes: Transparent operating mode, API operating mode, and Bypass operating mode. The default mode is Transparent operating mode.

Use the [AP \(API Enable\)](#) command to select a different operating mode.

The following flowchart illustrates how the modes relate to each other.





## 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 serial data it receives through the DIN pin for RF transmission. When a device receives RF data, it sends the data out through the DOUT pin. You can set the configuration parameters using Command mode.

The [IP \(IP Protocol\)](#) command setting controls how Transparent operating mode works for the XBee Cellular Modem.

## API operating mode

API operating mode is an alternative to Transparent operating mode. API mode is a frame-based protocol that allows you to direct data on a packet basis. The device communicates UART data in packets, also known as API frames. This mode allows for structured communications with computers and microcontrollers.

The advantages of API operating mode include:

- It is easier to send information to multiple destinations
- The host receives the source address for each received data frame
- You can change parameters without entering Command mode

## Bypass operating mode



**CAUTION!** Bypass operating mode is an alternative to Transparent and API modes for advanced users with special configuration needs. Changes made in this mode might change or disable the device and we do not recommend it for most users.

In Bypass mode, the device acts as a serial line replacement to the cellular chipset. In this mode, the XBee Cellular Modem exposes all control of the cellular chipset's AT port through the UART.

**Note** The cellular chipset can become unresponsive in Bypass mode. See [Unresponsive cellular chipset in Bypass mode](#) for help in this situation.

When Bypass mode is active, most of the XBee Cellular Modem's AT commands do not work. For example, **IM** (IMEI) may never return a value, and **DB** does not update.

When entering Bypass mode, the device terminates all existing socket connections. This includes the Device Cloud connection if it is currently enabled. When leaving Bypass mode, the device re-establishes a Device Cloud connection if the Device Cloud connection setting is currently enabled.

Command mode is available while in Bypass mode; see [Enter Command mode](#) for instructions.

## Enter Bypass operating mode

To configure a device for Bypass operating mode:

1. Set the **AP** (API Mode) parameter value to 5.
2. Use **AC** (Apply Changes) or **CN** (Exit Command Mode) to apply the changes.
3. To verify that Bypass operating mode is active, query **AI** (Association Indication) and confirm that it returns a value of 0x2F.

It may take a moment for Bypass operating mode to become active.

## Leave Bypass operating mode

To configure a device to leave Bypass operating mode:

1. Set the **AP** (API Mode) parameter value to something other than 5.
2. Use **AC** (Apply Changes) or **CN** (Exit Command Mode) to apply the changes.
3. To verify that Bypass operating mode is not active, query **AI** (Association Indication) and confirm that it returns something other than 0x2F.

## Boot the XBee Cellular Modem in Bypass operating mode

If the host sends **WR** (Write) when **AP** (API Mode) is set to 5, it enables Bypass operating mode when the XBee Cellular Modem turns on. In this configuration, the firmware does not test communication with the cellular chipset (which it does by sending AT commands). This is useful in case you have reconfigured the cellular chipset in a way that makes it incompatible with the firmware.

The option to boot with Bypass operating mode enabled exists for users who wish to communicate directly with the cellular chipset settings and do not intend to use XBee Cellular Modem software features such as API mode.

---

**Note** When the XBee Cellular Modem has been booted with Bypass operating mode enabled, disabling Bypass operating mode causes the XBee Cellular Modem firmware to reboot the cellular modem.

---

## Restore cellular settings to default in Bypass operating mode

Send **AT&F1** to reset the cellular chipset to its factory profile.

## Command mode

The three operating modes are controlled by the [AP \(API Enable\)](#) setting, but Command mode is always available as a mode the device can enter while configured for any of the operating modes.

### Enter Command mode

To get a device to switch into this mode, you must issue the following sequence: +++ within one second. There must be at least one second preceding and following the +++ sequence. Both the command character (**CC**) and the silence before and after the sequence (**GT**) are configurable. When the device sees a full second of silence in the data stream (the guard time, **GT**) followed by the string +++ (without Enter or Return) and another full second of silence, it knows to stop sending data and start accepting commands locally.

---

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

---

Once the device is in Command mode, it listens for user input and is able to receive AT commands on the UART. If **CT** time (default is 10 seconds) passes without any user input, the device drops out of Command mode and returns to the previous operating mode (Transparent, Bypass, API, Python, and so forth).

You can customize the command character, the guard times and the timeout in the device's configuration settings. For information on how to do this, see [CC \(Command Sequence Character\)](#), [CT \(Command Mode Timeout\)](#) and [GT \(Guard Times\)](#).

## Troubleshooting

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

There are two alternative ways to enter Command mode:

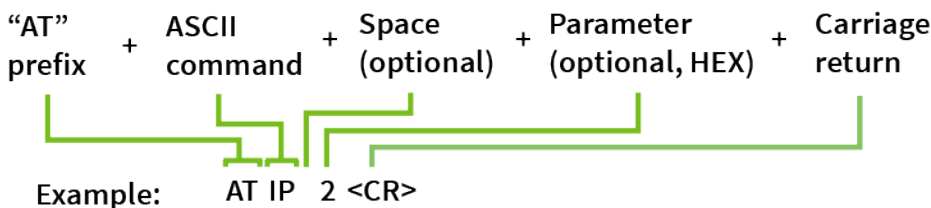
- A serial break for six seconds enters Command mode. The "break" command can be issued from a serial console, and is often a button or menu item.
- Asserting DIN (serial break) upon power up or reset enters Command mode. XCTU guides you through a reset and automatically issues the break when needed.

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

## Send AT commands

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

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



The preceding example changes the IP protocol to SMS.

## Respond to AT commands

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

## Apply command changes

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

1. Send the **AC** (Apply Changes) command.
- or:
2. Exit Command mode.

## Make command changes permanent

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

## 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 10 seconds.

For an example of programming the device using AT Commands and descriptions of each configurable parameter, see [AT commands](#).

## AT commands

---

Special commands .....	46
Cellular commands .....	47
Network commands .....	48
Addressing commands .....	49
Serial interfacing commands .....	51
I/O sampling commands .....	53
Command mode options .....	53
Firmware version/information commands .....	54
Diagnostic interface commands .....	56
Execution commands .....	58

## Special commands

The following commands are special commands.

### AC (Apply Changes)

Immediately applies new settings without exiting Command mode.

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

This behavior is in contrast to issuing the **WR** (Write) command. The **WR** command saves parameter values to non-volatile memory, but the device still operates according to previously saved values until the device is rebooted or you issue the **CN** (Exit AT Command Mode) or **AC** commands.

#### Parameter range

N/A

#### Default

N/A

### FR (Force Reset)

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

#### Parameter range

N/A

#### Default

N/A

### RE (Restore Defaults)

Restore device parameters to factory defaults.

The **RE** command does not write restored values to non-volatile (persistent) memory. Issue the **WR** (Write) command after issuing the **RE** command to save restored parameter values to non-volatile memory.

#### Parameter range

N/A

#### Default

N/A

### WR (Write)

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

---

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

---

**Parameter range**

N/A

**Default**

N/A

## Cellular commands

The following AT commands are cellular configuration and data commands.

### PH (Phone Number)

Reads the SIM card phone number. If **PH** is blank, the XBee Cellular Modem is not registered to the network.

**Parameter range**

N/A

**Default**

Set by the cellular carrier via the SIM card

### S# (ICCID)

Reads the Integrated Circuit Card Identifier (ICCID) of the inserted SIM.

**Parameter range**

N/A

**Default**

Set by the SIM card

### IM (IMEI)

Reads the device's International Mobile Equipment Identity (IMEI).

**Parameter range**

N/A

**Default**

Set in the factory

### MN (Operator)

Reads the network operator on which the device is registered.

**Parameter range**

N/A

**Default**

Verizon

## MV (Modem Firmware Version)

Read the firmware version string for cellular chipset communications. See the related [VR \(Firmware Version\)](#) command.

### Parameter range

N/A

### Default

Set in the currently loaded firmware

## DB (Cellular Signal Strength)

Reads the absolute value of the current signal strength to the cell tower in dB. If DB is blank, the XBee Cellular Modem has not received a signal strength from the cellular chipset.

### Parameter range

0x71 - 0x33 (-113 dBm to -51 dBm) [Read-only]

### Default

N/A

## Network commands

The following commands are network commands.

## IP (IP Protocol)

Sets or displays the IP protocol used for client and server socket connections in IP socket mode.

### Parameter range

Value	Description
0x00	UDP
0x01	TCP
0x02	SMS
0x03	Reserved
0x04	SSL over TCP communication

### Default

0x01

## TL (SSL/TLS Protocol Version)

Sets the SSL/TLS protocol version used for the SSL socket. If you change the **TL** value, it does not affect any currently open sockets. The value only applies to subsequently opened sockets.



**Note** Due to known vulnerabilities in prior protocol versions, we strongly recommend that you use the latest TLS version whenever possible.

### Range

Value	Description
0x00	SSL v3
0x01	TLS v1.0
0x02	TLS v1.1
0x03	TLS v1.2

### Default

0x03

## TM (TCP Client Connection Timeout)

Sets or reads the TCP client connection timeout. If there is no activity for this timeout then the connection is closed. If you set **TM** to 0, the connection is closed immediately after the device sends data.

If you change the **TM** value, it does not affect any currently open sockets. The value only applies to subsequently opened sockets.

### Parameter range

0 - 0xFFFF [x 100 ms]

### Default

0xBB8 (5 minutes)

## DO (Device Options)

Sets or displays whether Device Cloud support is enabled.

### Range

Value	Description
0x00	Device Cloud support is disabled.
0x01	Device Cloud support is enabled.

### Default

0x01

## Addressing commands

The following AT commands are addressing commands.

## DL (Destination Address)

The destination IPv4 address. The IP address must be a dotted quad (for example XXX.XXX.XXX.XXX).

### Parameter range

0- 15 IPv4 characters

### Default

0.0.0.0

## P# (Destination Phone Number)

Sets or displays the destination phone number used for SMS when **IP** (IP Protocol) = 2. Phone numbers must be fully numeric, 7 to 20 ASCII digits, for example: 8889991234.

**P#** allows international numbers with or without the + prefix. If you omit + and are dialing internationally, you need to include the proper International Dialing Prefix for your calling region, for example, 011 for the United States.

### Range

7 - 20 ASCII digits including an optional + prefix

### Default

N/A

## DE (Destination Port)

Sets or reads the destination IP port number.

### Parameter range

0x0 - 0xFFFF

### Default

0x2616

## TD (Text Delimiter)

The ASCII character used as a text delimiter for Transparent mode. When you select a character, information received over the serial port in Transparent mode is not transmitted until that character is received. To use a carriage return, set to 0xD. Set to zero to disable text delimiter checking.

### Parameter range

0 - 0xFF

### Default

0x0

## MY (Module IP Address)

Reads the device's IP address. This command is read-only because the IP address is assigned by the mobile network.

In API mode, the address is represented as the binary four byte big-endian numeric value representing the IPv4 address.

In Transparent or Command mode, the address is represented as a dotted-quad string notation.

**Parameter range**

0- 15 IPv4 characters

**Default**

0.0.0.0

## Serial interfacing commands

The following AT commands are serial interfacing commands.

### BD (Baud Rate)

Sets or reads the serial interface baud rate for communication between the device's serial port and the host.

**Parameter range**

0x0 - 0xA (standard rates) 0x5B9 to 0x5B8D80 (non-standard rates up to 6 Mb/s)

Parameter	Description
0x0	1200 b/s
0x1	2400 b/s
0x2	4800 b/s
0x3	9600 b/s
0x4	19200 b/s
0x5	38400 b/s
0x6	57600 b/s
0x7	115200 b/s
0x8	230400 b/s
0x9	460800 b/s
0xA	921600 b/s

**Default**

3 (9600 b/s)

### NB (Parity)

Set or display the parity settings for UART communications.

**Parameter range**

0x00 - 0x02

Parameter	Description
0x00	No parity
0x01	Even parity
0x02	Odd parity

**Default**

0x00

**SB (Stop Bits)**

Sets or reads the number of stop bits for UART communications.

**Parameter range**

0x00 - 0x01

Parameter	Configuration
0x00	One stop bit
0x01	Two stop bits

**Default**

0x00

**RO (Packetization Timeout)**

Set or read the number of character times of inter-character silence required before transmission begins when operating in Transparent mode.

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

Set to FF for realtime typing by humans. Also, see [TD \(Text Delimiter\)](#).

**Parameter range**

0 - 0xFF (x character times)

**Default**

3

**FT (Flow Control Threshold)**

Set or display the flow control threshold.

De-assert CTS when **FT** bytes are in the UART receive buffer.

**Parameter range**

0x9D - 0x82D

**Default**

0x681

**AP (API Enable)**

The API mode setting. The device can format the RF packets it receives into API frames and send them out the UART. When API is enabled the UART data must be formatted as API frames because Transparent mode is disabled. See [Modes](#) for more information.

**Parameter range**

0x00 - 0x05

Parameter	Description
0x00	API disabled (operate in Transparent mode)
0x01	API enabled
0x02	API enabled (with escaped control characters)
0x03	N/A
0x04	N/A
0x05	Bypass mode

**Default**

0

**I/O sampling commands**

The following AT command configures I/O sampling parameters.

**TP (Temperature)**

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

**Parameter range**

This is a read-only parameter

**Default**

N/A

**Command mode options**

The following commands are Command mode option commands.

**CC (Command Sequence Character)**

The character value the device uses to enter Command mode.

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

**Parameter range**

0 - 0xFF

Recommended: 0x20 - 0x7F (ASCII)

**Default**

0x2B (+ ASCII)

**CT (Command Mode Timeout)**

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

**Parameter range**

2 - 0x1770 (x 100 ms)

**Default**

0x64 (10 seconds)

**GT (Guard Times)**

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

**Parameter range**

0x2 - 0x576 (x 1 ms)

**Default**

0x3E8 (one second)

**Firmware version/information commands**

The following AT commands are firmware version/information commands.

**VR (Firmware Version)**

Reads the firmware version on a device.

The firmware version returns four hexadecimal values (2 bytes) **ABCD**. Digits **ABC** are the main release number and **D** is the revision number from the main release. **B** is a variant designator where **0** means standard release.

**Parameter range**

0 - 0xFFFF [Read-only]

**Default**

Set in firmware

**VL (Version Long)**

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

**Parameter range**

N/A

**Default**

Set in firmware

**HV (Hardware Version)**

Read the device's hardware version. Use this command to distinguish between different hardware platforms. The upper byte returns a value that is unique to each device type. The lower byte indicates the hardware revision.

**Parameter range**

0 - 0xFFFF [Read-only]

**Default**

Set in the factory

**HS (Hardware Series)**

Read the device's hardware series number.

**Parameter range**

N/A

**Default**

Set in the factory

**CK (Configuration CRC)**

Displays the cyclic redundancy check (CRC) of the current AT command configuration settings.

This command allows you to detect an unexpected configuration change on a device. Use the code that the device returns to determine if a node has the configuration you want.

After a firmware update this command may return a different value.

**Parameter range**

0 - 0xFFFFFFFF

**Default**

N/A

## Diagnostic interface commands

The following AT commands are diagnostic interface commands.

### DI (Device Cloud Indicator)

Displays the current Device Cloud status for the XBee.

#### Range

Value	Description
0x00	Connected
0x01	Before connection to the Internet
0x02	Device Cloud connection in progress
0x03	Disconnecting from Device Cloud
0x04	Not configured for Device Cloud

#### Default

N/A

### CI (Protocol/Connection Indication)

Displays information regarding the last IP connection (when the **IP** command = 0, 1 or 4) or SMS transmission (when **IP** = 2).

The following table provides the parameter's meaning when **IP** = 0 for UDP connections.

Parameter	Description
0x00	The socket is open
0x01	Tried to send but could not
0x02	Invalid parameters (bad IP/host)
0x03	TCP not supported on this cellular chipset
0x10	Not registered to the cell network
0x11	Cellular chipset not identified yet
0x20	Bad handle
0x21	User closed
0x22	Unknown server - DNS lookup failed
0x23	Connection lost
0x24	Unknown
0xFF	No known status



The following table provides the parameter's meaning when **IP** = 1 or 4 for TCP connections.

Parameter	Description
0x00	The socket is open
0x01	Tried to send but could not
0x02	Invalid parameters (bad IP/host)
0x03	TCP not supported on this cellular chipset
0x10	Not registered to the cell network
0x11	Cellular chipset not identified yet
0x20	Bad handle
0x21	User closed
0x22	No network registration
0x23	No internet connection
0x24	No server - timed out on connection
0x25	Unknown server - DNS lookup failed
0x26	Connection refused
0x27	Connection lost
0x28	Unknown
0xFF	No known status

The following table provides the parameter's meaning when **IP** = 2.

Parameter	Description
0x00	SMS successfully sent
0x01	SMS failed to send
0x02	Invalid SMS parameters - check <b>P#</b> (Destination Phone Number)
0x03	SMS not supported
0x10	No network registration
0x11	Cellular chipset stack error
0xFF	No SMS state to report (no SMS messages have been sent)

The value of this parameter resets to 0xFF when the device switches to SMS mode (**IP** = 2).

#### Parameter range

0 - 0xFF (read-only)

**Default**

-

## Execution commands

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

### AN (APN)

Sets or reads the Access Point Name (APN) the device uses to connect. The APN must match a valid value for the carrier.




---

**CAUTION!** Send **ATNR0** to reset Internet connectivity any time that you change **AN**. Failure to do so can cause usability problems with the XBee Cellular Modem.

---

If **AN** runs with no value, the command responds with a cached copy of the active APN value. When the XBee Cellular Modem is turned on, this value is empty and is queried from the cellular chipset settings and is cached once per minute.

If **AN** runs with a value, the XBee Cellular Modem holds that value in memory until the next time Internet connectivity is established.

**Parameter range**

0- 99 ASCII characters

**Default**

vzwinternet

### NR (Network Reset)

**NR** resets the network layer parameters.

The XBee Cellular Modem responds immediately with an **OK** on the UART and then causes a network restart.

If **NR** = 0, the XBee Cellular Modem tears down any TCP/UDP sockets and resets Internet connectivity. You can also send **NR**, which acts like **NR** = 0.

**Parameter range**

0

**Default**

N/A

### !R (Modem Reset)

Forces the cellular chipset to reboot.



**CAUTION!** This command is for advanced users, and you should only use it if the cellular chipset becomes completely stuck while in Bypass mode. Normal users should never need to run this command. See the [FR \(Force Reset\)](#) command instead.

---

**Range**

N/A

**Default**

N/A

# Operate in API mode

---

API frame specifications .....	61
Calculate and verify checksums .....	63
API frames .....	63

## API frame specifications

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

AP command setting	Description
<b>AP = 0</b>	Transparent operating mode, UART serial line replacement with API modes disabled. This is the default option.
<b>AP = 1</b>	API operation.
<b>AP = 2</b>	API operation with escaped characters (only possible on UART).
<b>AP = 3</b>	Reserved, do not use.
<b>AP = 4</b>	Reserved, do not use.
<b>AP = 5</b>	Bypass mode. This mode is for direct communication with the underlying chip and is only for advanced users.

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

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

### Frame data

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

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

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

## Checksum

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

## Calculate and verify checksums

To calculate the checksum of an API frame:

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

To verify the checksum of an API frame:

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

## API frames

The XBee Cellular Modem supports the following API frames:

- [AT Command - 0x08](#)
- [AT Command: Queue Parameter Value - 0x09](#)
- [Transmit \(TX\) SMS - 0x1F](#)
- [Transmit \(TX\) Request: IPv4 - 0x20](#)
- [AT Command Response - 0x88](#)
- [Transmit \(TX\) Status - 0x89](#)
- [Modem Status - 0x8A](#)
- [Receive \(RX\) Packet: SMS - 0x9F](#)
- [Receive \(RX\) Packet: IPv4 - 0xB0](#)

## AT Command - 0x08

### Description

Use this frame to query or set parameters on the local device. Changes this frame makes to device parameters take effect after executing the AT command.

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame type	0x08	Byte	
Frame ID		Byte	Identifies the data frame for the host to correlate with a subsequent ACK. If set to <b>0</b> , the device does not send a response.
AT command		Byte	Command name: two ASCII characters that identify the AT command.
Parameter value		Byte	If present, indicates the requested parameter value to set the given register. If no characters are present, it queries the register.



## AT Command: Queue Parameter Value - 0x09

### Description

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

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

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

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame type	0x09	Byte	
Frame ID		Byte	Identifies the data frame for the host to correlate with a subsequent ACK. If set to <b>0</b> , the device does not send a response.
AT command		Byte	Command name: two ASCII characters that identify the AT command.
Parameter value		Byte	If present, indicates the requested parameter value to set the given register. If no characters are present, it queries the register.

## Transmit (TX) SMS - 0x1F

### Description

Transmit an SMS message. The frame allows international numbers with or without the + prefix. If you omit + and are dialing internationally, you need to include the proper International Dialing Prefix for your calling region, for example, 011 for the United States.

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame type	0x1F	Byte	
Frame ID		Byte	Reference identifier used to match status responses. Zero disables the TX Status frame.
Options		Byte	Reserved for future use.
Phone number		20 byte string	String representation of phone number terminated with a null (0x0) byte. Use numbers and the + symbol only, no other symbols or letters.
Payload		Variable (160 characters maximum)	Data to send as the body of the SMS message.

## Transmit (TX) Request: IPv4 - 0x20

### Description

A TX Request message causes the device to transmit data in IPv4 format.

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame type	0x20	Byte	
Frame ID		Byte	Reference identifier used to match status responses. Zero disables the TX Status frame.
Destination address		32-bit big endian	
Destination port		16-bit big endian	
Source port		16-bit big endian	UDP or TCP port number. To send a UDP packet, this must match the port number of the listening port. To send a TCP packet on a new connection, this must be <b>0</b> .
Protocol		Byte	0 = UDP 1 = TCP 4 = SSL over TCP
Transmit options		Byte bitfield	Bit 1: When set, close TCP socket immediately after transmit success. All other bits are reserved and should be set to zero.
Payload		Variable	Data to be transferred to the destination, may be up to 1500 bytes.

## AT Command Response - 0x88

### Description

A device sends this frame in response to an AT Command (0x08) frame. Some commands send back multiple frames.

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame type	0x88	Byte	
Frame ID		Byte	Identifies the data frame for the host to correlate with a subsequent ACK. If set to <b>0</b> , the device does not send a response.
AT command		Byte	Command name: two ASCII characters that identify the AT command.
Status	##	Byte	0 = OK 1 = ERROR 2 = Invalid command 3 = Invalid parameter
Parameter value		Byte	Register data in binary format. If the register was set, then this field is not returned.

## Transmit (TX) Status - 0x89

### Description

Indicates the success or failure of a transmit operation.

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame type	0x89	Byte	
Frame ID		Byte	Refers to the frame ID specified in a previous transmit frame
Status		Byte	Status code (see the table below)

The following table shows the SMS status codes.

Code	Description
0x0	Successful transmit
0x21	Failure to transmit to cell network
0x22	Not registered to cell network
0x2c	Invalid frame values (check the phone number)
0x31	Internal error
0x32	Resource error (retry operation later)
0x74	Message too long
0x78	Invalid UDP port
0x79	Invalid TCP port
0x7A	Invalid host address
0x7B	Invalid data mode
0x80	Connection refused
0x81	Socket connection lost
0x82	No server
0x83	Socket closed
0x84	Unknown server
0x85	Unknown error

## Modem Status - 0x8A

### Description

Cellular chipset status messages are sent from the device in response to specific conditions.

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame type	0x8A	Byte	
Status	##	Byte	0 = Hardware reset or power up 1 = Watchdog timer reset 2 = Registered with cellular network 3 = Unregistered with cellular network 0x0E = Device Cloud connected 0x0F = Device Cloud disconnected

## Receive (RX) Packet: SMS - 0x9F

### Description

This XBee Cellular Modem uses this frame when it receives an SMS message.

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame Type	0x9F	Byte	
Phone number		20 byte string	String representation of the phone number, padded out with null bytes (0x0).
Payload		Variable	Body of the received SMS message.

## Receive (RX) Packet: IPv4 - 0xB0

### Description

Needed

### Format

The following table provides the contents of the frame. For details on frame structure, see [API frame specifications](#).

Field name	Field value	Data type	Description
Frame type	0xB0	Byte	
Source address		32-bit big endian	
Destination port		16-bit big endian	
Source port		16-bit big endian	
Protocol		Byte	0 = UDP 1 = TCP 4 = SSL over TCP
Status		Byte	Reserved
Payload		Variable	Data received from the source. The maximum size is 1500 bytes.



## Socket behavior

---

Secure Sockets Layer (SSL) certificate checking .....	74
Socket timeouts .....	74

## Secure Sockets Layer (SSL) certificate checking

Currently, the XBee Cellular Modem is not capable of checking the certificate of the remote end of the socket connection. We will add this support in the future.

This means that the device cannot authenticate the remote end of the connection. Although the connection is encrypted, there is no way to verify that the remote server is the expected server.

## Socket timeouts

The XBee Cellular Modem implicitly opens the socket any time there is data to be sent, and closes it according to the timeout settings. The [TM \(TCP Client Connection Timeout\)](#) command controls the timeout settings.

# Troubleshooting

---

This section contains troubleshooting steps for the XBee Cellular Modem.


- Cannot find the serial port for the device .....76
- Correct a macOS Java error .....77
- Unresponsive cellular chipset in Bypass mode .....78
- Not on expected network after APN change .....78

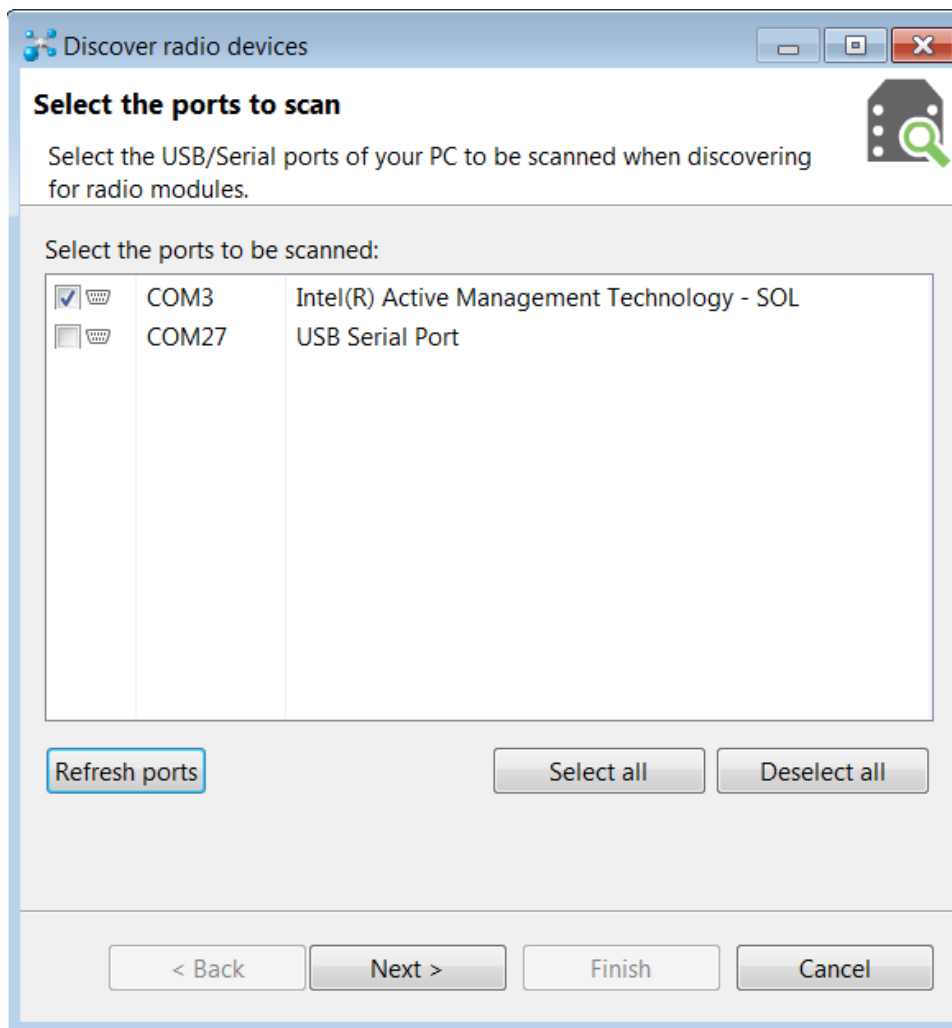
## Cannot find the serial port for the device

### Condition

In XCTU, the serial port that your device is connected to does not appear.

### Solution

1. Click the **Discover radio modules** button .
2. Select all of the ports to be scanned.
3. Click **Next** and then **Finish**. A dialog notifies you of the devices discovered and their details.



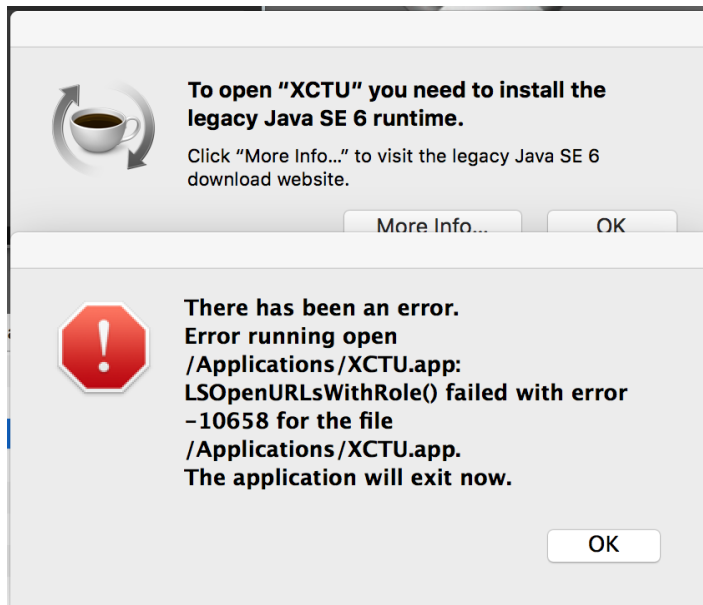
4. Remove the development board from the USB port and view which port name no longer appears in the **Discover radio devices** list of ports. The port name that no longer appears is the correct port for the development board.

## Correct a macOS Java error

When you use XCTU on macOS computer, you may encounter a Java error.

### Condition

When opening XCTU for the first time on a macOS computer, you may see the following error:



### Solution

1. Click **More info** to open a browser window.
2. Click **Download** to get the file javaforosx.dmg.
3. Double-click on the downloaded javaforosx.dmg.
4. In the dialog, double-click the JavaForOSX.pkg and follow the instructions to install Java.

## Unresponsive cellular chipset in Bypass mode

When in Bypass mode, the XBee Cellular Modem does not reset or reboot the cellular chipset if the cellular chipset becomes unresponsive.

### Condition

The cellular chipset is unresponsive.

### Solution

1. Query the [AI \(Association Indication\)](#) parameter to determine whether the cellular chipset is connected to the XBee Cellular Modem software:

Status code	Meaning
0x2F	The cellular chipset and XBee Cellular Modem are connected.
0xC2	The cellular chipset is not working.

2. You can send the [!R \(Modem Reset\)](#) command to reset only the cellular chipset.

## Not on expected network after APN change

### Condition

The XBee Cellular Modem is not on the expected network after a change to the [AN \(APN\)](#) command.

### Solution

Send **ATNR0** to reset Internet connectivity. See [NR \(Network Reset\)](#) for more information.

# Certifications

---

Modification statement .....	80
Interference statement .....	80
Wireless notice .....	80
FCC Class B digital device notice .....	81
Labelling requirements for the host device .....	81

## Modification statement

Digi International has not approved any changes or modifications to this device by the user. Any changes or modifications could void the user's authority to operate the equipment.

*Digi International n'approuve aucune modification apportée à l'appareil par l'utilisateur, quelle qu'en soit la nature. Tout changement ou modification peuvent annuler le droit d'utilisation de l'appareil par l'utilisateur.*

## Interference statement

This device complies with Part 15 of the FCC Rules and 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.*

## Wireless notice

This equipment complies with FCC and IC radiation exposure limits set forth for an uncontrolled environment. The antenna should be installed and operated with minimum distance of 20 cm between the radiator and your body. Antenna gain must be below:

Frequency band	Gain
Band 4 (1700 MHz)	12.9 dBi
Band 13 (700 MHz)	6.0 dBi

This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

*Cet appareil est conforme aux limites d'exposition aux rayonnements de la IC pour un environnement non contrôlé. L'antenne doit être installée de façon à garder une distance minimale de 20 centimètres entre la source de rayonnements et votre corps. Gain de l'antenne doit être ci-dessous:*

Bande de fréquence	Gain
Band 4 (1700 MHz)	12.9 dBi
Band 13 (700 MHz)	6.0 dBi

Band 4 (1700 MHz) 12.9 dBi

Band 13 (700 MHz) 6.0 dBi

*L'émetteur ne doit pas être colocalisé ni fonctionner conjointement avec à autre antenne ou autre émetteur.*



## FCC Class B digital device notice

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:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

## Labelling requirements for the host device

The device shall be properly labelled to identify the product within the host device. The certification label of the module shall be clearly visible at all times when installed in the host device, otherwise the host device must be labelled to display the FCC ID and IC of the module, preceded by the words "Contains transmitter module", or the word "Contains", or similar wording expressing the same meaning, as follows:

Contains FCC ID: RI7LE866SV1

Contains IC: 5131A-LE866SV1

*L'appareil hôte doit être étiqueté comme il faut pour permettre l'identification des modules qui s'y trouvent. L'étiquette de certification du module donné doit être posée sur l'appareil hôte à un endroit bien en vue en tout temps. En l'absence d'étiquette, l'appareil hôte doit porter une étiquette donnant le FCC ID et le IC du module, précédé des mots « Contient un module d'émission », du mot « Contient » ou d'une formulation similaire exprimant le même sens, comme suit :*

Contains FCC ID: RI7LE866SV1

Contains IC: 5131A-LE866SV1

CAN ICES-3 (B) / NMB-3 (B)

This Class B digital apparatus complies with Canadian ICES-003.

*Cet appareil numérique de classe B est conforme à la norme canadienne ICES-003.*