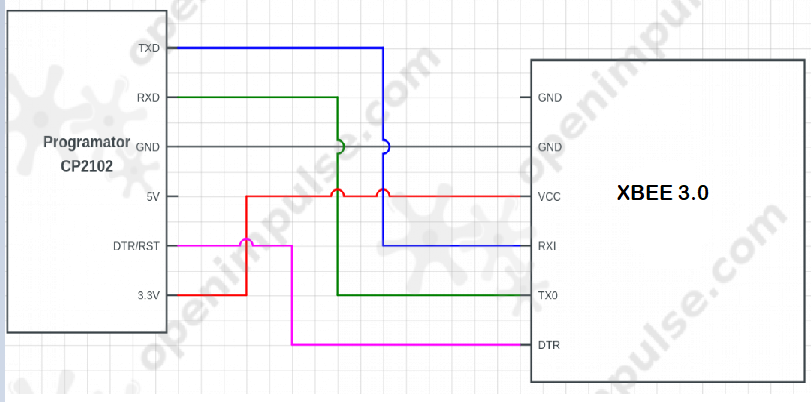
**HARDWARE**

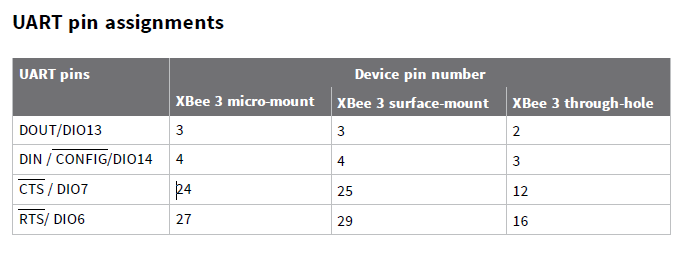
XBEE Uses UART SERIAL PROTOCOL for general purposes.

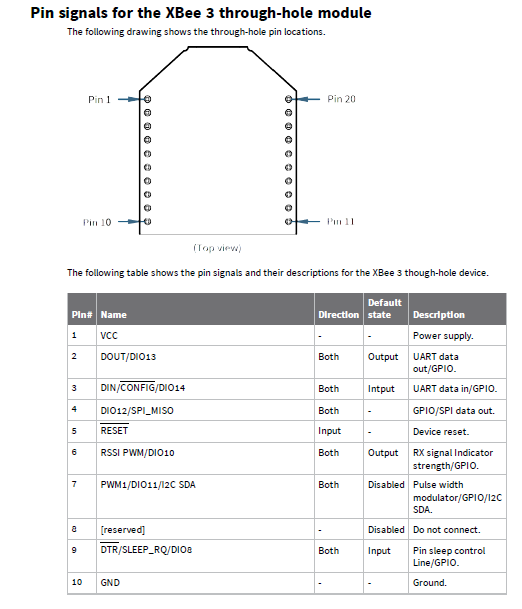
For setting up XBEE modules USB to UART connection is required

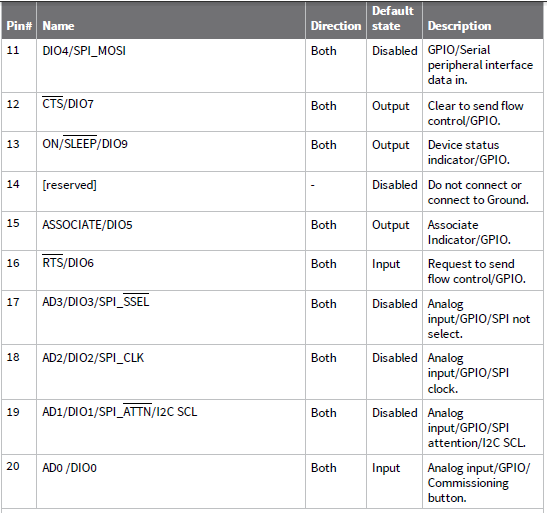
**USB to UART Wiring Diagram.**



**XBEE PIN OUT**

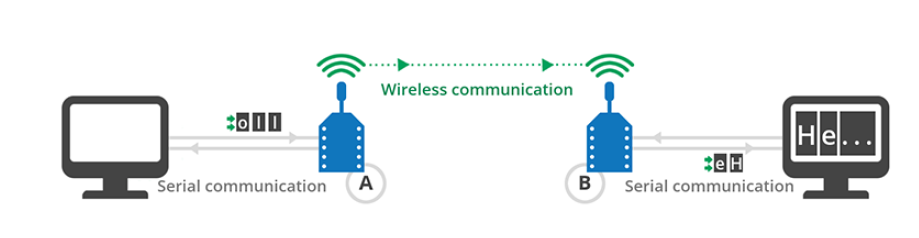




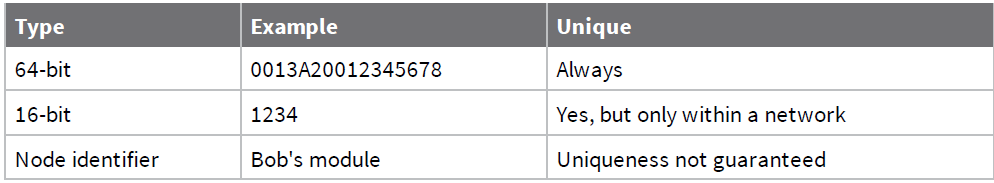


**How XBEE devices communicate**

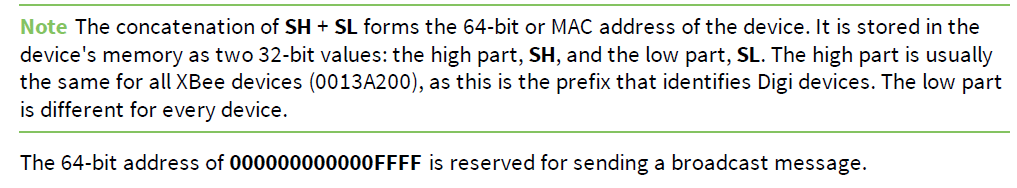
An XBEE communicates remotely with other XBEEs via wireless and locally with the intelligent device (microcontroller, computer) connected to it via the serial interface.



**ADDRESSING**



***64-bit address (MAC).***



***16-bit address (NETWORK ADDRESS)***

A device receives a random 16-bit address when it joins a ZigBee network, the value of the 16-bit address can be read through the 16-bit Network Address (**MY**) parameter.

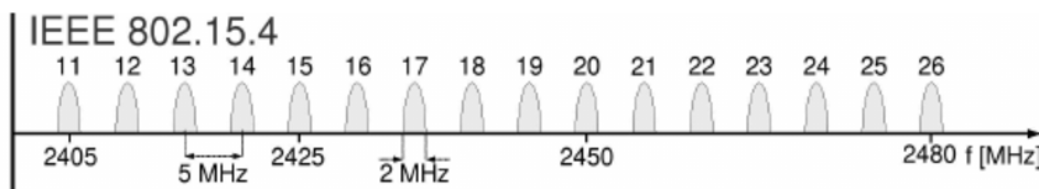
16-bit address of 0000 is reserved to the coordinator, while a value of FFFE means the device has not joined a PAN.

**PAN Addresses**

Zigbee networks are called personal area networks or PANs. A unique PAN identifier (PAN ID) defines each network and the identifier is common among all devices of the same network.

The value of the personal area network can be set through the PAN ID (**ID**) parameter. If this value is 0, the XBee automatically selects the PAN ID, so you can read it using the Operating PAN ID (**OP**) parameter.

**Channels**



To determine the specific channel where the device is operating, you must read the Operating Channel (**CH**) parameter.

you can select the operating channel by setting a single bit in the **SC** parameter.

That single bit forces a coordinator to operate on the channel specified by the single bit. It also prevents routers and end devices from joining a network on any channel but the one specified in **SC**.

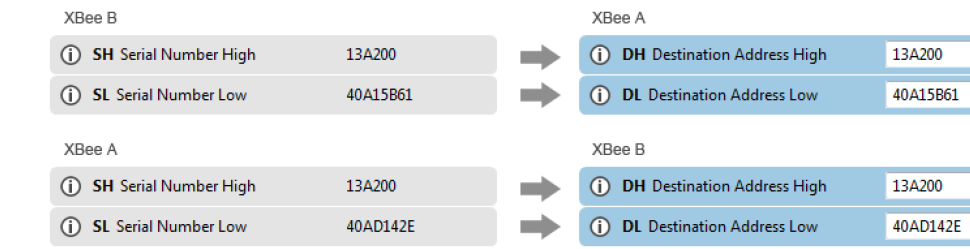
If the selected channel is not important, you can use the **SC** parameter to select multiple channels.

**SUMMARY NOTES:**

To communicate wirelessly, your modules must be part of the same network, so the **ID** and **CH** parameters must have identical values for all XBees in the network.

Every XBee module has a unique 64-bit address called MAC that distinguishes it from the rest of the devices. This address is formed by the concatenation of the parameters **SH** (Serial Number High) and **SL** (Serial Number Low).

**Transparent mode**



Communication in transparent mode yields the same result as if the two modules were connected by a wire, but wireless communication makes that physical wire unnecessary.

For two XBee modules to communicate, the sending module needs the address of the recipient.

Transparent mode has some limitations. For example, when you are working with several modules,

you must configure the destination before sending each message

You can set the device up to work in command mode in order to pass information to the local device for example if you want to modify its configuration.

**XBEE API mode Application Programming Interface**

API mode provides a structured interface where data is communicated through the serial interface in organized packets and in a determined order.

Since the data destination is included as part of the API frame structure, you can use API mode to transmit messages to multiple devices.

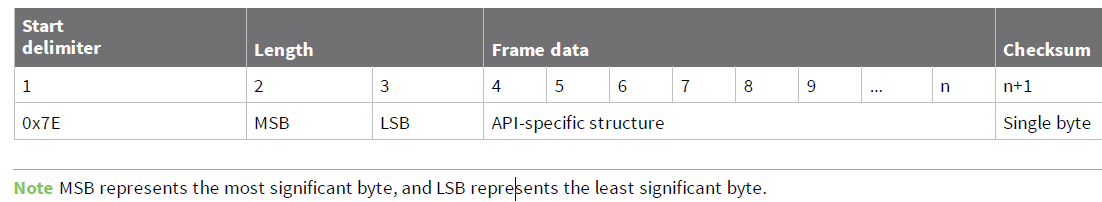
Since there are different frames for different purposes (such as configuration and

communication), you can configure a device without entering Command mode

The API frame includes the source of the message so it is easy to identify where data is coming from.

**API frame structure**

They are sent and received through the serial interface of the device and contain the wireless message itself as well as some extra information such as the destination/source of the data or the signal quality.

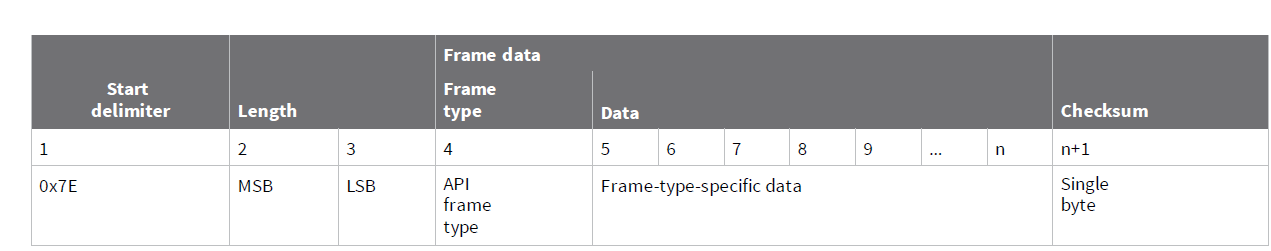


**Start delimiter** Its value is always 0x7E.This allows for easy detection of a new incoming frame.

**Length** The length field specifies the total number of bytes included in the frame data field. Its two-byte value.

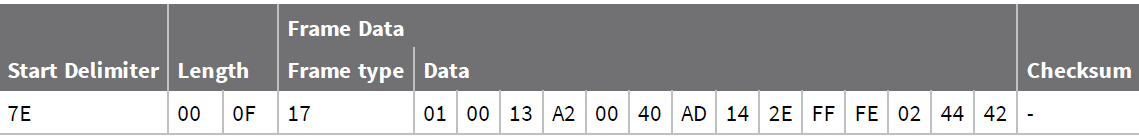
**Frame data** This field contains the information received or to be transmitted.

Frame data is structured based on the purpose of the API frame



**Checksum** Isthe last byte of the frame and helps test data integrity.

***Calculate the checksum of an API frame***



Add all bytes excluding the start delimiter and the length: 17 + 01 + 00 + 13 + A2 + 00 + 40 + AD + 14 + 2E + FF + FE+ 02 + 44 + 42 = 481

From the result, keep only the lowest 8 bits: 81.

Subtract that result from 0xFF: FF - 81 = 7E

2

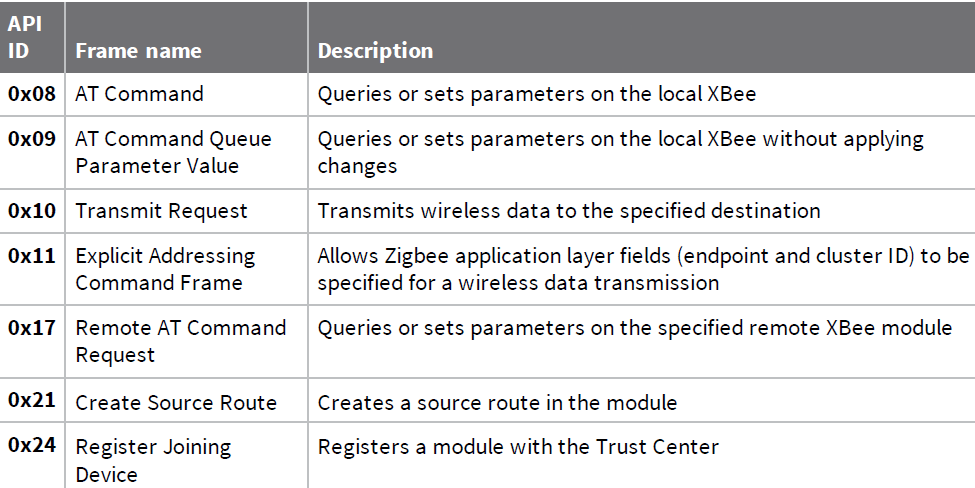
***Verify the checksum of a given API frame***

Add all bytes including the checksum (do not include the delimiter and length).

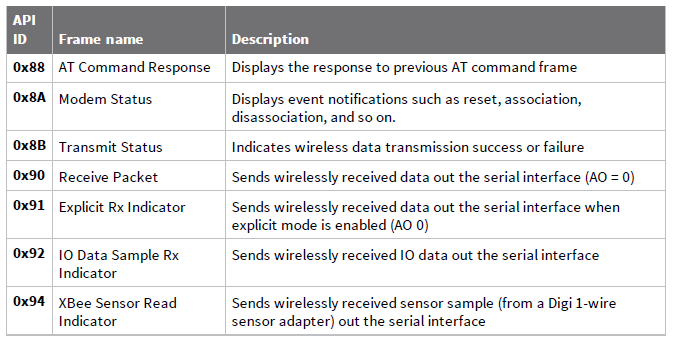
If the checksum is correct, the last two digits on the far right of the sum will equal FF.

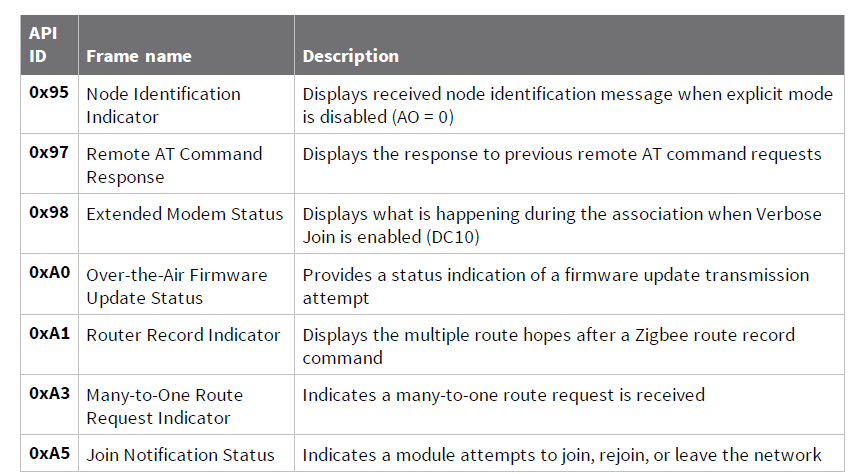
**Supported frames**

**Transmit data frames**



**Receive data frames**





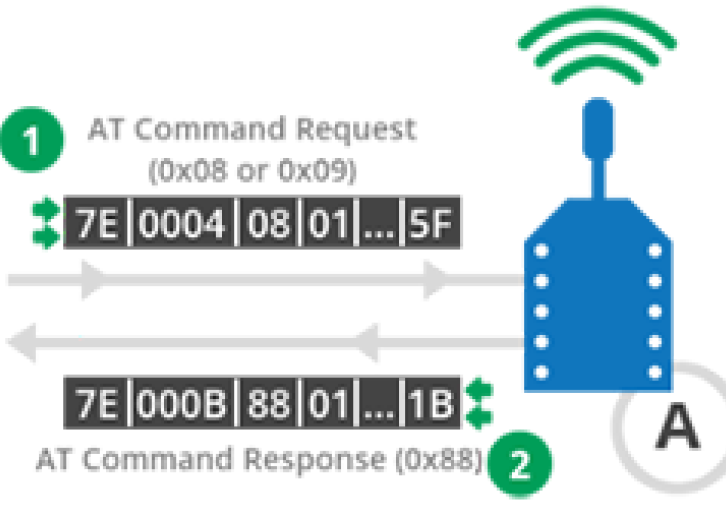
**XBEE frame exchange**

**AT Command: configure a local XBee device**

These are the same AT parameters and commands that are available in

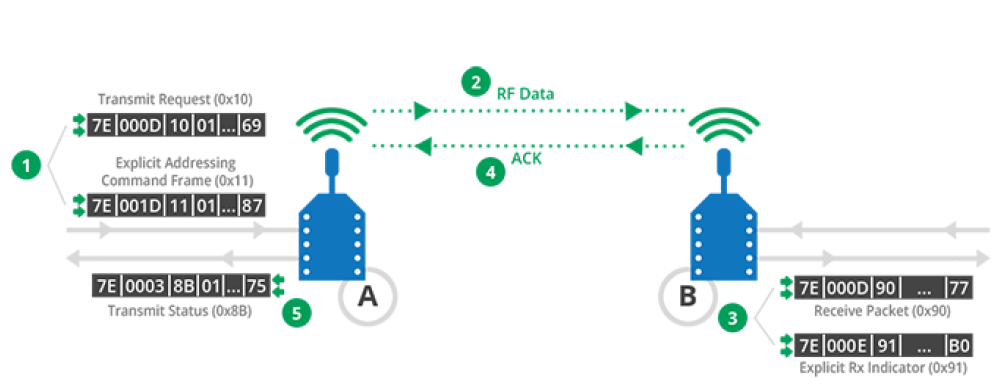
Transparent/Command mode, but included in an AT Command (0x08) frame.

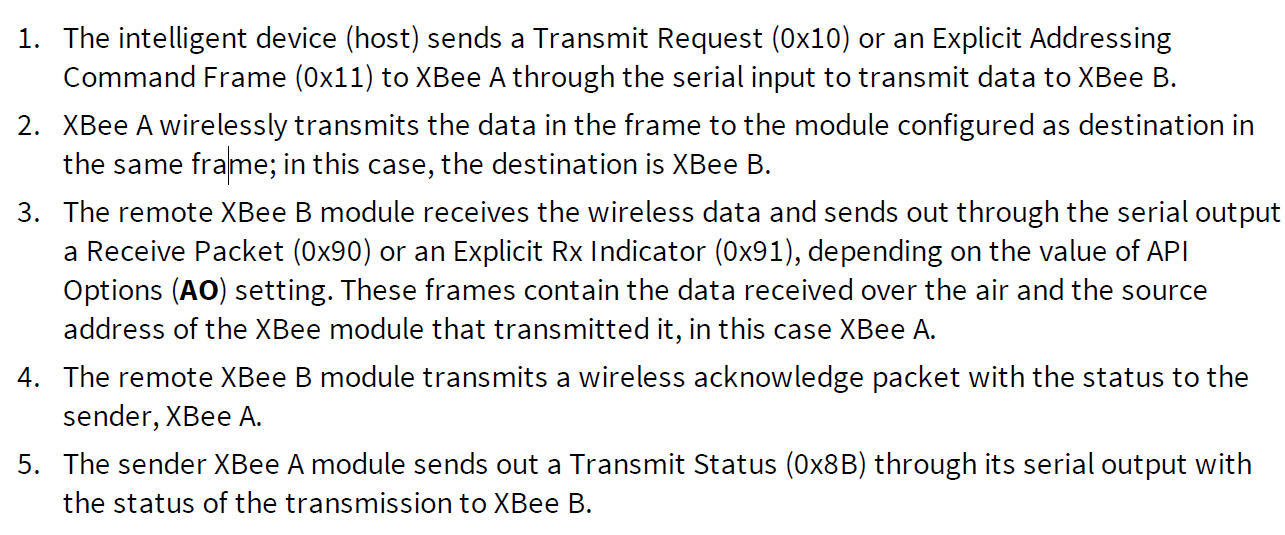
The response containing the result of the operation is sent back in an AT Command Response (0x88) frame.



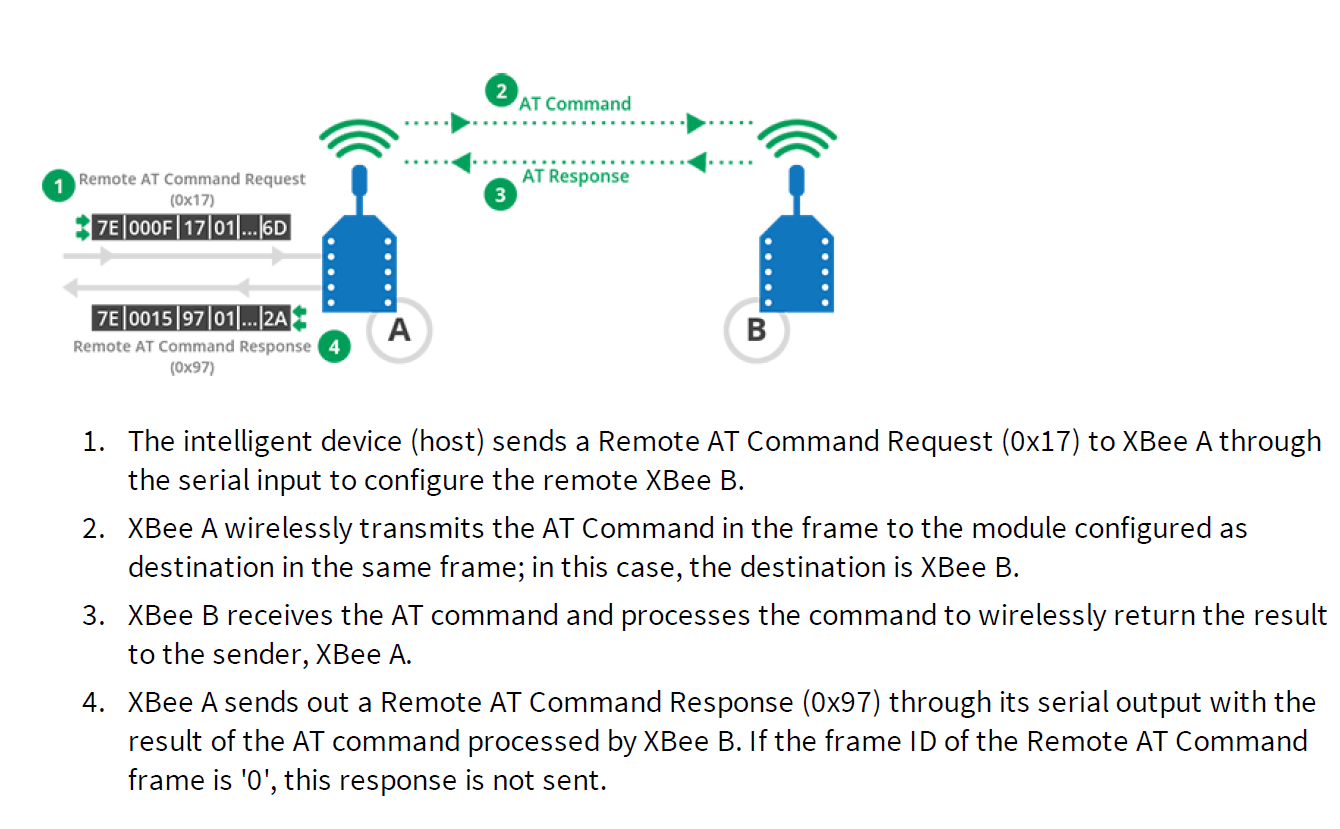
**Transmit Request/Receive Packet: Transmit and receive wireless**

**Data**





**Remote AT Command: Remotely configure an XBee module**



**Source routing: Create and obtain the route of a packet and other examples pages 60-69 (Xbee Zigbee Mesh Manual)**

**Startup operations**

When you power on an XBee module, it performs several operations depending on the role assigned.

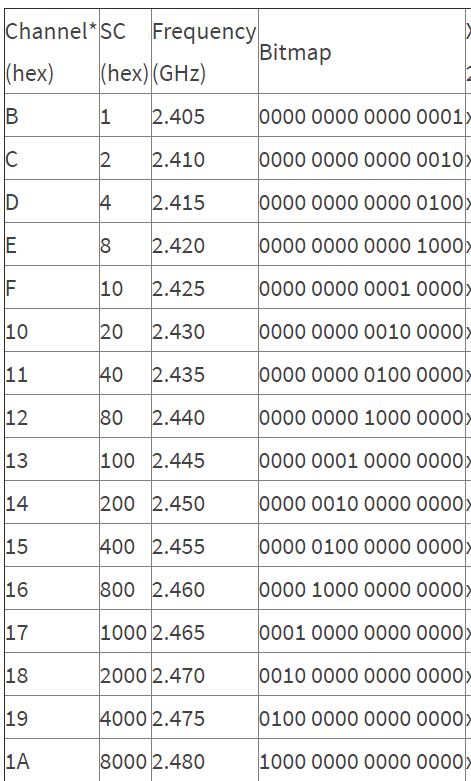
**Coordinator**

To ensure the coordinator starts on a good channel and unused PAN ID, it performs a series of scans to discover any RF activity on different channels (energy scan) and to discover any nearby operating PANs (active scan).

PAN ID (**ID**). Determines the PAN ID. If set to 0 (default), the device selects a random PAN ID.

Scan Channels (**SC**). Determines the scan channels bitmask the coordinator uses to form a

network. The coordinator performs an energy scan on all enabled SC channels.



You can also specify multiple channels for the module to choose from. For example, if you wanted  
to use only channels 12, 13, and 14, you would add all the corresponding bitmaps together.  
  
0000 0000 1000 0000 + 0000 0001 0000 0000 + 0000 0010 0000 0000 = 0000 0011 1000 0000 =  
0x0380

Scan Duration (**SD**). Sets the scan duration, which determines how long the coordinator

performs an energy or active scan on a given channel.

The permit joining attribute is configurable with the Node Join Time (**NJ**) command.

**Router**

To discover nearby networks, the router performs an active scan, just like the coordinator does when it starts the network. When a router joins a network, it receives a randomly selected 16-bit address from the device that allowed the join.

Join Verification (**JV**). If enabled, the XBee attempts to discover the address of the coordinator

when it first joins a network.

Network Watchdog Timeout (**NW**). Used for a powered router to periodically check for the

presence of a coordinator to verify network connectivity.

**End device**

Similar to routers, end devices must also discover and join a valid Zigbee network before they can

participate in it. End devices also discover networks by issuing an active scan, and when they join a

network they receive a randomly selected 16-bit address from the device that allowed the join.

Coordinators and routers maintain a table of all child devices that have joined. This table has a finite

size and determines how many end devices can join. You can use the Number of Remaining Children (**NC**) setting to determine how many additional end devices can join a coordinator or router.

**Transmission methods**

**Broadcast transmission** Broadcastmeans to transmit the same data to all nodes on a network.

You can address broadcast transmissions using either the 64-bit broadcast address or the 16-bit

broadcast address:

If the **64-bit broadcast address** (000000000000FFFF) is used, set the 16-bit address to

unknown address (FFFE).

If the **16-bit broadcast address** (FFFF) form is used, set the 64-bit address to unknown

address (FFFFFFFFFFFFFFFF).

**Unicast transmission**

A **unicast** transmission consists of sending messages to a single node on the network identified by a unique address.

If you use the **64-bit address**, set the network address to unknown address (FFFE).

If you use the **16-bit address**, set the 64-bit address must be set to unknown address

(FFFFFFFFFFFFFFFF).

The 16-bit address 0000 and the 64-bit address 0000000000000000 are reserved for the

coordinator.

**Signal strength and the RSSI pin**

The Received Signal Strength Indicator (RSSI) measures the amount of power present in a radio

signal. It is an approximate value for signal strength received on an antenna.

If a distant transmitter is moved closer to a receiver, the strength of the transmitted signal at the receiving antenna increases. Likewise, if a transmitter is moved farther away, signal strength at the receiving antenna decreases.

The RSSI is measured in dBm. A greater negative value (in dBm) indicates a weaker signal. Therefore, -50 dBm is better than -60 dBm.

XBee module's pin 6 can be configured as an RSSI pin that outputs a PWM (pulse-width modulation) signal representing this value. To do so, configure **P0** as RSSI [1]

Configure the amount of time the RSSI pin is active, by modifying the RSSI PWM Timer (RP) setting.

**RP** value is expressed in hexadecimal notation. For example, a configured value of 0x1E is equivalent to 30 in decimal and means that the pin will be active for three seconds (30\*100=3000ms).

After the **RP** time has elapsed and no data has been received, the pin will be set to low.

A value of 0xFF permanently enables the pin; when configured in this way, it will always reflect the RSSI value of the last-received data packet.

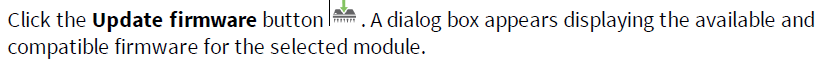
The RSSI value can also be obtained by reading the XBee **DB** parameter value. It represents the RSSI absolute value of the last received data packet expressed in hexadecimal notation.

**SOFTWARE**

1. Add a radio module manually / Discover local radio modules
2. Read radio module configuration

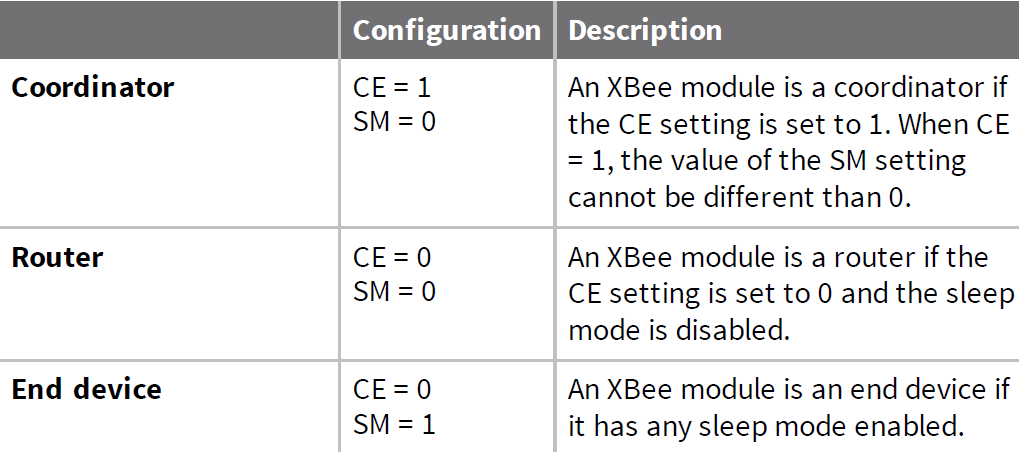
Select a radio module from the device list. XCTU displays the current firmware settings for that.

1. Update firmware



1. Configure the device type of an XBEE module

The device type of an XBee is determined by the value of two parameters: Coordinator Enable (**CE**) and Sleep Mode (**SM**). The first setting determines if an XBee module is coordinator or not, and the second one determines if the module is router or end device. Coordinators and routers cannot sleep.



1. Configure the ZigBee mesh Network.

