

XBee Connector User Guide

UgCS 2.9.1230



Contents

1	XBee Connector User Guide	1
1.1	Legal notice	1
1.2	Introduction	1
1.3	Interaction model	1
1.4	Prerequisites	2
1.5	XBee configuration	3
1.5.1	Ground station side	3
1.5.2	Vehicle side	4
1.6	Hardware connection	4
1.7	Configuration file	5
1.7.1	Common parameters	5
1.7.2	Serial port configuration	5
1.7.3	TCP port number for incoming connections from VSMS	5
1.8	Common configuration file parameters	6
1.8.1	Logging configuration	6
1.8.2	Serial port configuration	7
2	Disclaimer	8

1 XBee Connector User Guide



1.1 Legal notice

See [Disclaimer](#).

XBee® and **Digi®** are registered trademarks of **Digi International Inc.**

ZigBee® is a registered trademark of **ZigBee Alliance**.

1.2 Introduction

ZigBee is a wireless network standard targeted at wide development in wireless control and monitoring applications. It suits well for unmanned vehicles remote controlling.

There are several advantages of ZigBee remote control:

- It is possible to control a number of vehicles from one node.
- Multiple vehicles are able to form **wireless mesh network**.
- You can create encrypted radio channel between ground station and vehicle (ZigBee networks are secured by 128 bit symmetric encryption keys).

You can use all these advanced features in UgCS by virtue of XBee Connector. It allows to connect any supported vehicle to ground station via ZigBee protocol using Digi XBee modules.

Some of XBee module specifications:

- **RF interaction distance between two XBee:** up to 3200 meters in a line of sight
- **Data rate:** up to 250 Kbps
- **Radio frequency:** 2.4 GHz
- **Operating Temperature:** -40° C to +85° C

Warning

Note that only Pixhawk with ArduPilot firmware is supported at the moment.

1.3 Interaction model

All vehicle specific data exchange in UgCS is performed by VSMs. Every VSM are able to connect to XBee Connector by TCP. The only parameters to be configured in particular VSM are the host address and port number of XBee Connector running instance.

XBee Connector is responsible for detecting remote ZigBee nodes and routing data flow to correspondent VSM.

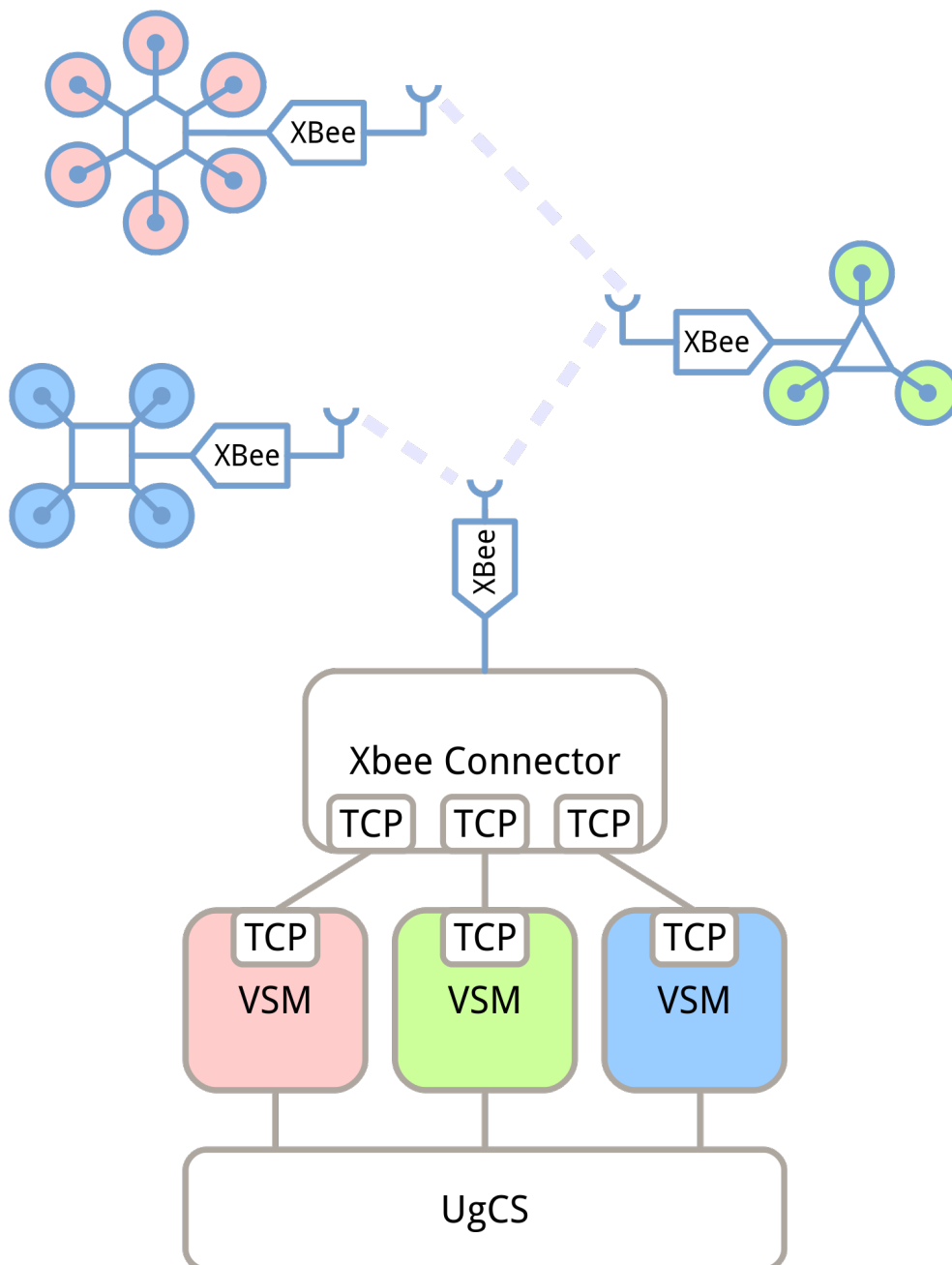


Figure 1: XBee interaction model

If one of the vehicles is not in a line of sight in terms of ZigBee link (e.g. due to being too far or behind an obstacle) then this vehicle can interact with ground station through other vehicle which is located in between.

1.4 Prerequisites

You need at least two Digi XBee ZigBee modules in order to connect your vehicle via ZigBee interface. You can choose options that would best fit your needs on manufacturer's website:

<http://www.digi.com/products/xbee-rf-solutions/modules/xbee-zigbee>

We recommend through-hole modules instead of SMT due to more simple connecting procedures.

Also you will need an appropriate USB-UART adapter for connecting XBee module to PC and autopilot. For example you can use DFRobot's one:

http://www.dfrobot.com/index.php?route=product/product&product_id=588

1.5 XBee configuration

There is a dedicated software for XBee modules configuration from Digi called **XCTU**. Please refer to Digi website in order to download it:

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

After installing software you are to configure XBee modules.

1.5.1 Ground station side

First module should be appointed as **ZigBee Coordinator API**. Such kind of appointment is performed by firmware update:

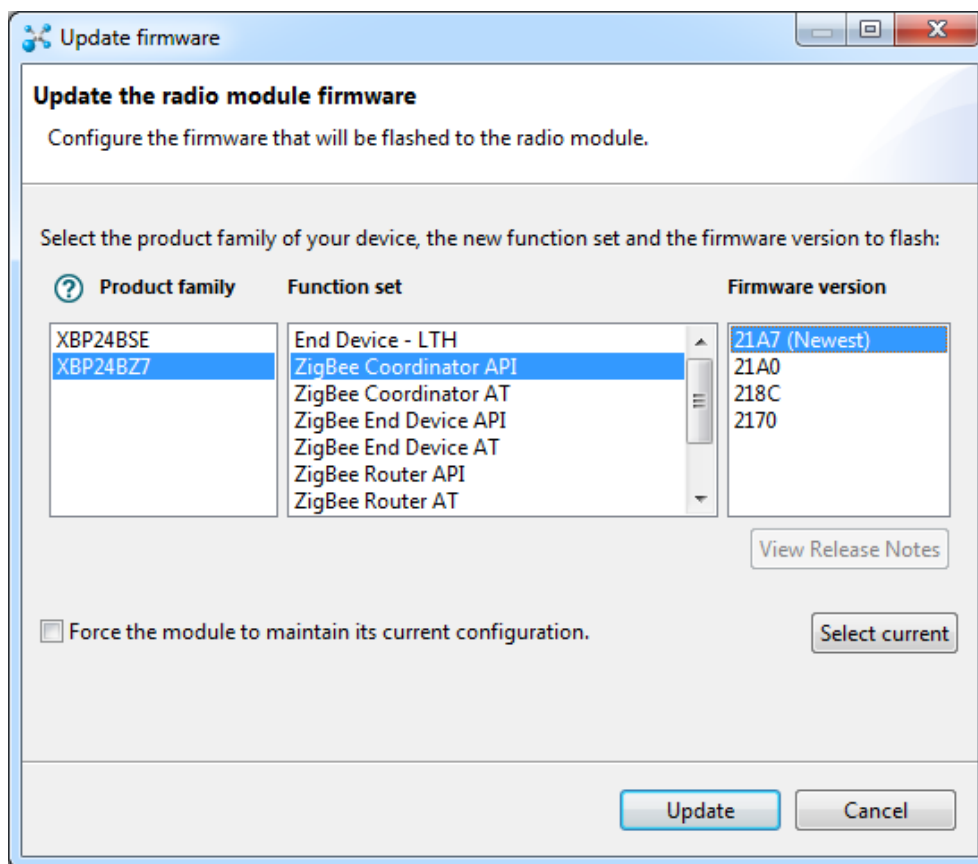


Figure 2: ZigBee Coordinator API firmware

This module is to be connected to PC via mentioned USB-UART adapter.

The next step is to configure Coordinator's parameters. Most important are:

- **ID** (PAN ID): **105**
- **NI** (Node Identifier): **ONGROUND**

Other parameters may be set to default values. Just in case check some of them:

- **PL** (Power Level): **Highest** (4)

- **BD** (Baud Rate): **115200** (7)

1.5.2 Vehicle side

A role of the second module (and subsequent if any) may be **ZigBee Router AT** as well as **ZigBee End Device AT**. ZigBee Router option is more preferable because it brings more networking capabilities.

Choose correspondent firmware and update the module:

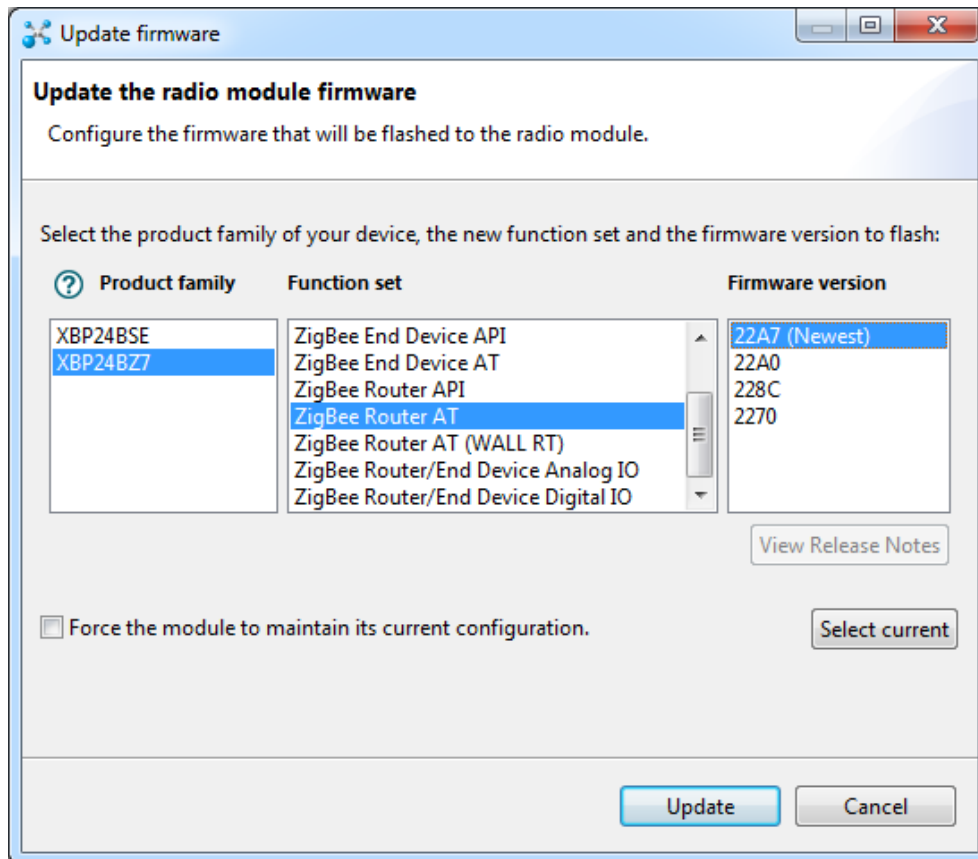


Figure 3: ZigBee Router AT firmware

Router's parameters should be as following:

- **ID** (PAN ID): **105** (*same value as on Coordinator*)
- **NI** (Node Identifier): **ONBOARD1** (*ONBOARD2, ONBOARD3 etc. for subsequent XBee modules*)
- **PL** (Power Level): **Highest** (4)
- **BD** (Baud Rate): **57600** (6)

1.6 Hardware connection

Connect XBee Coordinator to USB using USB-UART adapter mentioned above. Check out whether drivers are installed correctly and new serial port became available.

Connect XBee Router to Pixhawk TELEM1 connector. Connection diagram (in case of using DFRobot XBee USB Adapter V2):

Pixhawk TELEM1 (DF13-6)	Adapter UART (PLS-5)
1 (+5V)	1 (+5V)
2 (TX, out)	4 (RX, in)
3 (RX, in)	3 (TX, out)
6 (GND)	2 (GND)

After powering autopilot up you can use XCTU software for performing remote nodes discovery in order to make sure that wiring and settings are correct.

1.7 Configuration file

Default configuration file of the XBee Connector suits most needs and it is generally not necessary to modify it.

Configuration file location:

- **On Microsoft Windows:**

```
C:\Program Files (x86)\UgCS\bin\vsm-xbee.conf
```

- **On GNU/Linux:**

```
/etc/opt/ugcs/vsm-xbee.conf
```

- **On Apple OS X:**

```
/Users/[user name]/Library/Application Support/UGCS/configuration/vsm-xbee.conf
```

1.7.1 Common parameters

XBee Connector shares a common set of configuration file parameters with VSMS which are described in [Common configuration file parameters](#). XBee Connector configuration file prefix is:

```
xbee
```

1.7.2 Serial port configuration

Mandatory. At least one serial port should be configured, otherwise XBee Connector will not try to connect to the Coordinator.

- **Name:** xbee.serial_port
- **Description:** Serial port configuration, for more details see [Serial port configuration](#). Default XBee Coordinator serial port communication speed is 115200 bps.
- **Example:**

```
xbee.serial_port.1.name = com1
xbee.serial_port.1.baud = 115200
```

1.7.3 TCP port number for incoming connections from VSMS

Mandatory.

- **Name:** xbee.tcp.[number].port = [port number]
- **Description:** This port number is used to establish TCP connections between XBee Connector and VSMS. It should have the same value as `vehicle.[VSM].tcp.[number].port` parameter of particular VSM configuration.
- **Example:**

```
xbee.tcp.1.port = 5566
```

1.8 Common configuration file parameters

VSM configuration file is a text file specified via command line argument `--config` of the VSM application. Example:

```
--config /etc/opt/ugcs/vsm-xbee.conf
```

Each configuration parameter is defined as a line in the configuration file with the following structure:

```
name1.name2...nameX = value
```

where name1, name2 ... nameX are arbitrary names separated by dots to divide a variable into logical blocks and a value which can be a number value or a text string depending on the context. See below the description about common VSM configuration parameters.

1.8.1 Logging configuration

1.8.1.1 Level

Optional.

- **Name:** log.level = [error|warning|info|debug]
- **Description:** Logging level.
- **Default:** info
- **Example:** log.level = debug

1.8.1.2 File path

Optional.

- **Name:** log.file_path = [path to a file]
- **Description:** Absolute or relative (to the current directory) path to a logging file. Logging is disabled if logging file is not defined. File should be writable. Backslash should be escaped with a backslash.
- **Example:** log.file = /var/opt/ugcs/log/vsm-xbee/vsm-xbee.log
- **Example:** log.file = C:\\Users\\John\\AppData\\Local\\UGCS\\logs\\vsm-xbee\\vsm-xbee.log

1.8.1.3 Maximum single file size

Optional.

- **Name:** log.single_max_size = [size]
- **Description:** Maximum size of a single log file. When maximum size is exceeded, existing file is renamed by adding a time stamp and logging is continued into the empty file. [size] should be defined as a number postfixed by a case insensitive multiplier:
 - Gb, G, Gbyte, Gbytes: for Giga-bytes
 - Mb, M, Mbyte, Mbytes: for Mega-bytes
 - Kb, K, Kbyte, Kbytes: for Kilo-bytes
 - no postfix: for bytes
- **Default:** 100 Mb
- **Example:** log.single_max_size = 500 Mb

1.8.2 Serial port configuration

Optional. VSM which communicates with vehicles via serial ports should define at least one serial port, otherwise VSM will not try to connect to the vehicles. Port name and baud rate should be both defined. [prefix] is unique for each VSM.

1.8.2.1 Port name

Optional.

- **Name:** [prefix].[port index].name = [regular expression]
- **Description:** Ports which should be used to connect to the vehicles by given VSM. Port names are defined by a [regular expression] which can be used to define just a single port or create a port filtering regular expression. Expression is case insensitive on Windows. [port index] is a arbitrary port indexing name.
- **Example:** vehicle.ardupilot.serial_port.1.name = /dev/ttyUSB[0-9]+|com[0-9]+
- **Example:** vehicle.ardupilot.serial_port.2.name = com42

1.8.2.2 Port baud rate

Optional.

- **Name:** [prefix].[port index].baud.[baud index] = [baud]
- **Description:** Baud rate for port opening. [baud index] is an optional arbitrary name used when it is necessary to open the same serial port using multiple baud rates. [port index] is an arbitrary port indexing name.
- **Example:** xbee.serial_port.1.baud.1 = 115200
- **Example:** xbee.serial_port.1.baud.2 = 57600
- **Example:** xbee.serial_port.2.baud = 38400

1.8.2.3 Excluded port name

Optional.

- **Name:** [prefix].exclude.[exclude index] = [regular expression]
- **Description:** Ports which should not be used for vehicle access by this VSM. Port names are defined by a [regular expression] which can be used to define just a single port or create a port filtering regular expression. Filter is case insensitive on Windows. [exclude index] is a arbitrary indexing name used when more than one exclude names are defined.
- **Example:** xbee.serial_port.exclude.1 = /dev/ttyS.*
- **Example:** xbee.serial_port.exclude = com1

1.8.2.4 Serial port arbiter

Optional.

- **Name:** [prefix].use_serial_arbiter = [yes|no]
- **Description:** Enable (yes) or disable (no) serial port access arbitration between VSMs running on the same machine. It is recommended to have it enabled to avoid situation when multiple VSMs try to open the same port simultaneously.
- **Default:** yes
- **Example:** xbee.serial_port.use_serial_arbiter = no

2 Disclaimer

DISCLAIMER OF WARRANTIES AND LIMITATIONS ON LIABILITY.

(a) SMART PROJECTS HOLDINGS LTD MAKE NO REPRESENTATIONS OR WARRANTIES REGARDING THE ACCURACY OR COMPLETENESS OF ANY CONTENT OR FUNCTIONALITY OF THE PRODUCT AND ITS DOCUMENTATION.

(b) SMART PROJECTS HOLDINGS LTD DISCLAIM ALL WARRANTIES IN CONNECTION WITH THE PRODUCT, AND WILL NOT BE LIABLE FOR ANY DAMAGE OR LOSS RESULTING FROM YOUR USE OF THE PRODUCT, INCLUDING BUT NOT LIMITED TO INJURY OR DEATH OF USER OR ANY THIRD PERSONS OR DAMAGE TO PROPERTY.

(c) THE SOFTWARE IS SUPPLIED AS IS WITH NO WARRANTIES AND CAN BE USED ONLY AT USERS OWN RISK.