

PX4 VSM User Guide

UgCS 2.10.1123



Contents

1	Connecting PX4 autopilot to UgCS	1
1.1	First time vehicle connection	1
1.2	Mission execution specifics	2
1.3	Altitude	3
1.4	Home Location (HL) support.	3
1.4.1	Landing at Home Location	3
1.5	Command execution specifics	3
1.6	Command shading	4
1.7	Telemetry information specifics	4
1.7.1	Air speed	4
1.7.2	RC link quality	4
1.8	Fail-safe actions	4
1.9	Waypoint turn types	4
1.10	Connection using ZigBee interface	5
1.11	Configuration file	5
1.11.1	Common parameters	5
1.11.2	Communication channel configuration	5
1.11.3	Model name and serial number override	5
1.12	Common configuration file parameters	6
1.12.1	UgCS server configuration	6
1.12.2	Logging configuration	6
1.12.3	Mission dump path	7
1.12.4	Automatic service discovery	8
1.13	Communication with vehicle	8
1.13.1	Serial port configuration	8
1.13.2	TCP connection configuration	9
1.13.3	UDP connection configuration	9
1.13.4	Proxy configuration	10
2	Disclaimer	11

1 Connecting PX4 autopilot to UgCS



1.1 First time vehicle connection

See [Disclaimer](#).

Please follow these steps to connect an PX4 vehicle to the UgCS:

1. PX4 vehicle must be properly configured, calibrated and tested using tools and instruction from the official [PX4 web site](#) prior to using it with UgCS. UgCS does not support initial configuration, setup and calibration of PX4 driven vehicles.
2. If more than one PX4 vehicle is planned to be used with UgCS, it must be ensured that each vehicle has a unique system id as defined by the parameter `SYSID_THISMAV`, otherwise UgCS will not be able to distinguish between different vehicles and it will not be possible to operate vehicles normally. To change the parameter, please use the official PX4 configuration software like QGroundControl.
3. Turn on the vehicle and plug in the radio modem paired with the vehicle or direct USB cable from the PX4 board to the computer where VSM is running. UgCS uses serial ports for communication with PX4 vehicles. Standard communication devices like 3DR radio modems (and their analogs) and direct USB connections are supported, as long as OS driver for virtual serial port is installed and serial port is successfully created. Please refer to your communication equipment manufacturer documentation about driver installation instructions.
4. As soon as uplink and downlink connection is established, the vehicle should appear in the active vehicles list in main (map) view. Open *Vehicles* window from main menu and choose the corresponding vehicle for editing by clicking on the menu item and selecting *Edit* button. Now you can select the vehicle profile and change the default vehicle name to be convenient for you:



Figure 1: New PX4 vehicle

Vehicle profile needs to be assigned to allow mission planning with this vehicle. Vehicle avatar should be assigned in vehicle profile to properly see the vehicle location on map.

- Repeat steps above for each your PX4 vehicle.

Supported vehicle types:

- Copters

Supported PX4 firmware versions:

- 1.4.4

1.2 Mission execution specifics

Mission action support:

Flight plan element / action	Support	Notes
Change speed	Yes	
Wait	Yes	
Panorama	No	
Set camera mode	No	
Set camera by time	No	
Set camera by distance	No	

Set camera attitude	No	
Set POI	No	
Change yaw	No	

1.3 Altitude

PX4 reports two altitude types - AMSL (Above Mean Sea Level) and AHL (Above Home Location). Altitude AHL is displayed as Raw altitude in client.

Vehicle altitude AMSL is calculated from Raw altitude as:

$$\text{Vehicle altitude AMSL} = \text{Takeoff altitude} + \text{Reported altitude AHL}$$

If Takeoff altitude is not specified in UgCS then:

$$\text{Vehicle altitude AMSL} = \text{Reported altitude AMSL}$$

Current altitude AGL (Above Ground Level) is calculated as:

$$\text{Vehicle altitude AGL} = \text{Vehicle altitude AMSL} - \text{Terrain elevation AMSL at vehicle location}$$

1.4 Home Location (HL) support.

PX4 does not support setting HL from UgCS. HL is always automatically set to the current location when vehicle is armed.

UgCS is able to read the current HL from the vehicle and is display it on the map.

1.4.1 Landing at Home Location

Vehicle behavior (land or do not land) after returning at Home Location depends on on the autopilot configuration.

1.5 Command execution specifics

Command	Support	Notes
ARM	Yes	Arms vehicle.
DISARM	Yes	Disarms vehicle.
AUTOMODE	Yes	Start mission from first waypoint. Sets vehicle into <i>Mission</i> flight mode.
MANUALMODE	Yes	Sets <i>Manual</i> mode.
CLICK & GO	Yes	Sets <i>Click & Go</i> (single waypoint) mode.
JOYSTICK	No	
HOLD	Yes	Pause mission execution. The drone will loiter at its current position.
CONTINUE	Yes	Continue with mission execution from next waypoint. Works from <i>Manual</i> and <i>Click&Go</i> modes.
RETURN HOME	Yes	Vehicle will return to home location. See also Home Location (HL) support .

TAKEOFF	Yes	
LAND	Yes	
EMERGENCYLAND	Yes	
CAMERA_TRIGGER	No	

1.6 Command shading

UGCS Client can show command buttons in different shades. You can always press all buttons disregarding of shade. Highlighted buttons suggest recommended commands, depending on vehicle current status.

1.7 Telemetry information specifics

1.7.1 Air speed

If there is no air speed sensor onboard, air speed will be shown as "Not available". If there is an air speed sensor onboard, the air speed value will be shown.

1.7.2 RC link quality

RC Link quality reporting is not supported.

1.8 Fail-safe actions

GPS Lost:

This parameter is not supported. Vehicle will behave as specified. By default it will wait for 30s. If GPS signal does not restore vehicle will land.

RC Lost:

Action	Result
Wait	Aircraft changes altitude to failsafe altitude and returns home
Land	Aircraft lands even if in loiter mode
Return Home	Aircraft changes altitude to failsafe altitude and returns home
Continue	Aircraft continues mission

Battery Low:

Action	Result
Land	Aircraft changes altitude to failsafe altitude and returns home
Return Home	Aircraft changes altitude to failsafe altitude and returns home
Continue	Aircraft continues mission

1.9 Waypoint turn types

Turn type	Support	Notes
Straight	Yes	The vehicle will fly a straight line to the location specified as a lat, lon and altitude.

Spline	No	
--------	----	--

1.10 Connection using ZigBee interface

There is a possibility to connect UgCS to PX4 vehicle using ZigBee interface. Connection is performed with two or more Digi XBee ZigBee modules (one on ground side, others on vehicles side) and dedicated UgCS software component called XBee Connector. Please refer to XBee Connector user guide for details.

In order to use such kind of connection you are to disable [Serial port configuration](#) and enable [Proxy configuration](#).

1.11 Configuration file

Default configuration file of the PX4 VSM 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-px4.conf
```

- **On GNU/Linux:**

```
/etc/opt/ugcs/vsm-px4.conf
```

- **On Apple OS X:**

```
/Users/[user name]/Library/Application Support/UGCS/configuration/vsm-px4.conf
```

1.11.1 Common parameters

All VSMs share a common set of configuration file parameters described in [Common configuration file parameters](#). PX4 VSM configuration file prefix is:

```
vehicle.px4
```

1.11.2 Communication channel configuration

There must be at least one communication channel defined, otherwise VSM will not try to connect to the vehicle. See [Communication with vehicle](#) for details

Default installation is configured to detect autopilot automatically on any available serial port at 57600 or 115Kbps.

1.11.3 Model name and serial number override

Optional.

- **Name:** vehicle.px4.custom.[name].system_id = [system id]
- **Name:** vehicle.px4.custom.[name].model_name = [model name]
- **Name:** vehicle.px4.custom.[name].serial_number = [serial number]
- **Description:** In UgCS each vehicle is identified by a unique combination of model name and serial number represented as text strings. By default, PX4 vehicles are identified with a model name *PX4* and serial number equal with the Mavlink system id read from the vehicle. It can be overridden by these parameters, where [name] is an arbitrary vehicle name, [system id] is the original Mavlink system id which should be overridden, [model name] is a new model name to be visible to the UgCS, [serial number] is a new serial number to be visible to the UgCS.

- **Example:**

```
vehicle.px4.custom.my_drone.system_id = 2
vehicle.px4.custom.my_drone.model_name = PX4Quad
vehicle.px4.custom.my_drone.serial_number = 123456
```

1.12 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-ardupilot.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.

1.12.1 UgCS server configuration

1.12.1.1 Listening address

Mandatory.

- **Name:** ucs.local_listening_address = [IP address]
- **Description:** Local TCP address to listen for incoming connections from UgCS server. Specify *0.0.0.0* to listen from all local addresses.
- **Example:** ucs.local_listening_address = 0.0.0.0

1.12.1.2 Listening port

Mandatory.

- **Name:** ucs.local_listening_port = [port number]
- **Description:** Local TCP port to listen for incoming connections from UgCS server. Default is 5556.
- **Example:** ucs.local_listening_port = 5556

1.12.2 Logging configuration

1.12.2.1 Level

Optional.

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

1.12.2.2 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-ardupilot/vsm-ardupilot.log`
- **Example:** `log.file` = `C:\\Users\\John\\AppData\\Local\\UGCS\\logs\\vsm-ardupilot\\vsm-ardupilot.log`

1.12.2.3 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

1.12.2.4 Maximum number of old log files

Optional.

- **Name:** `log.max_file_count` = [number]
- **Description:** Log rotation feature. Maximum number of old log files to keep. After reaching `single_max_size` of current log file VSM will rename it with current time in extension and start new one. VSM will delete older logs so the number of old logs does not exceed the `max_file_count`.
- **Default:** 1
- **Example:** `log.max_file_count` = 5

1.12.3 Mission dump path

Optional.

- **Name:** `[prefix].mission_dump_path` = [path to a file]
- **Description:** File to dump all generated missions to. Timestamp is appended to the name. Delete the entry to disable mission dumping. All directories in the path to a file should be already created.
- **Example:** `vehicle.ardupilot.mission_dump_path` = `C:\\tmp\\ardupilot_dump`

1.12.4 Automatic service discovery

VSM can respond to automatic service discovery requests from UgCS server.

When this parameter is not configured, service discovery is disabled.

Optional.

- **Name:** `service_discovery.vsm_name = [Service name]`
- **Description:** Human readable service name.
- **Example:** `service_discovery.vsm_name = Ardupilot VSM`

1.13 Communication with vehicle

VSM can communicate with Vehicle over different communication channels

Currently supported channels:

- Serial port, see [Serial port configuration](#) for details.
- TCP link, see [TCP connection configuration](#) for details.
- UDP link, see [UDP connection configuration](#) for details.
- vsm-proxy (XBee), see [Proxy configuration](#) for details.

1.13.1 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.

1.13.1.1 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`

1.13.1.2 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:** `vehicle.ardupilot.serial_port.1.baud.1 = 9600`
- **Example:** `vehicle.ardupilot.serial_port.1.baud.2 = 57600`
- **Example:** `vehicle.ardupilot.serial_port.2.baud = 38400`

1.13.1.3 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 an arbitrary indexing name used when more than one exclude names are defined.
- **Example:** vehicle.ardupilot.serial_port.exclude.1 = /dev/ttyS.*
- **Example:** vehicle.ardupilot.serial_port.exclude = com1

1.13.1.4 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:** vehicle.ardupilot.serial_port.use_serial_arbiter = no

1.13.2 TCP connection configuration

Optional. VSM which communicates with vehicles over TCP should define at least one network connection, otherwise VSM will not try to connect to vehicles. [prefix] is unique for each VSM.

1.13.2.1 IP-address for outgoing TCP connection

Optional.

- **Name:** [prefix].detector.[con index].address = [IP-address]
- **Description:** IP-address of vehicle to connect to. Typically used for vehicle simulators.
- **Example:** vehicle.ardupilot.detector.1.address = 10.0.0.111

1.13.2.2 remote TCP port

Optional.

- **Name:** [prefix].detector.[con index].tcp_port = [port number]
- **Description:** Remote port to connect to.
- **Example:** vehicle.ardupilot.detector.1.tcp_port = 5762

1.13.3 UDP connection configuration

Optional. VSM which communicates with vehicles via network should define at least one network connection, otherwise VSM will not try to connect to vehicles. [prefix] is unique for each VSM.

1.13.3.1 Local IP-address for UDP

Optional.

- **Name:** [prefix].detector.[con index].udp_local_address = [IP-address]
- **Description:** Local IP-address to listen for incoming UDP packets on. Specify 0.0.0.0 if you want to listen on all local addresses.
- **Example:** vehicle.ardrone.detector.1.udp_local_address = 0.0.0.0

1.13.3.2 Local UDP port

Optional.

- **Name:** [prefix].detector.[con index].udp_local_port = [port number]
- **Description:** Local UDP port to listen for incoming packets on.
- **Example:** vehicle.ardrone.detector.1.udp_local_port = 14550

1.13.3.3 Remote IP-address for UDP

Optional.

- **Name:** [prefix].detector.[con index].udp_address = [IP-address]
- **Description:** Remote IP-address to send outgoing UDP packets to.
- **Example:** vehicle.ardrone.detector.1.udp_address = 192.168.1.1

1.13.3.4 Remote UDP port

Optional.

- **Name:** [prefix].detector.[con index].udp_port = [port number]
- **Description:** Remote UDP port to send outgoing packets to.
- **Example:** vehicle.ardrone.detector.1.udp_port = 14551

1.13.4 Proxy configuration

Optional. VSM is able to communicate with vehicle via proxy service which redirects dataflow received from vehicle through TCP connection to VSM and vice versa using specific protocol. In other words proxy service appears as a router between vehicle and VSM. At the moment there is one implementation of proxy in UgCS called XBee Connector which retranslates data from ZigBee network to respective VSM.

1.13.4.1 IP-address for proxy

Optional.

- **Name:** [prefix].tcp.[con index].proxy = [IP-address]
- **Description:** IP-address to connect proxy to. Specify local or remote address.
- **Example:** vehicle.ardupilot.tcp.1.proxy = 127.0.0.1

1.13.4.2 TCP port for proxy

Optional.

- **Name:** [prefix].tcp.[con index].port = [port number]
- **Description:** TCP port to be connected with proxy through. Should be the same as in configuration on proxy side.
- **Example:** vehicle.ardupilot.tcp.1.port = 5566

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.