

DJI VSM User Guide

UgCS 3.3.362



Copyright (c) 2018, Smart Projects Holdings Ltd

Contents

1	Connecting DJI autopilots to UgCS	1
1.1	Supported autopilots	1
1.2	First time vehicle connection	1
1.3	Known datalink connection issues	2
1.4	Mission execution specifics	2
1.5	Command execution specifics	4
1.5.1	Notes on Auto mode	4
1.5.2	Notes on Continue command	5
1.5.3	Notes on Return Home command	5
1.6	Joystick Mode	5
1.7	Command availability	5
1.8	Telemetry information specifics	5
1.9	Fail-safe actions	6
1.10	Waypoint turn types	6
1.11	General Purpose servo action setup	7
1.11.1	Configuring trigger_action parameters in vsm-dji.conf	7
1.12	GoPro video link	8
1.13	Configuration file	8
1.13.1	Common parameters	8
1.13.2	Serial port configuration	8
1.13.3	Waypoint turn type	8
1.13.4	NAZA/Wookong detection	9
1.13.5	Payload control	9
1.14	Common configuration file parameters	10
1.14.1	UgCS server configuration	10
1.14.2	Automatic service discovery	11
1.14.3	Logging configuration	11
1.14.4	Mission dump path	12
1.14.5	Command execution control	12
1.15	Communication with devices	13
1.15.1	Serial port configuration	14
1.15.2	Outgoing TCP connection configuration	15
1.15.3	Incoming TCP connection configuration	15
1.15.4	Outgoing UDP connection configuration	15
1.15.5	Incoming UDP connection configuration	16
1.15.6	Incoming UDP connection configuration (any peer)	17
1.15.7	Named pipes	17
1.15.8	Proxy configuration	17

2 Disclaimer

18

1 Connecting DJI autopilots to UgCS

See [Disclaimer](#).

1.1 Supported autopilots

This document relate to the following DJI autopilots and ready to fly vehicles:

- A2
- Wookong-M
- Naza-M V2
- Phantom 2

For Phantom 2 Vision+, Phantom 3 and Inspire1 please see separate documents on <http://ugcs.com> site.

1.2 First time vehicle connection

Please follow these steps to connect a DJI vehicle to the UgCS:

1. To connect DJI vehicle to UgCS you need the 2.4Hz datalink (<http://www.dji.com/2-4g-bluetooth-datalink>) or 900MHz datalink. Direct USB cable to DJI vehicle cannot be used to connect it to UgCS.
2. For Windows setup you also need to download from DJI site and install driver for 2.4GHz datalink (http://download.dji-innovations.com/downloads/driver/DJI_WIN_Driver_Installer.exe). For 900MHz datalink download and install this package (http://download.dji-innovations.com/downloads/driver/DJI_Datalink_Driver_Installer_1.0.zip). This step is not required if you are running UgCS on Linux or Mac.
3. Before connecting the vehicle to UgCS, please ensure all autopilot settings (fail-safe, control mode switch, compass calibration) are configured accordingly via DJI Assistant software. Please consult user manual of your autopilot for details.
4. Once the drone is connected it should appear in vehicles list. Both Uplink and Downlink connections should be available. 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:

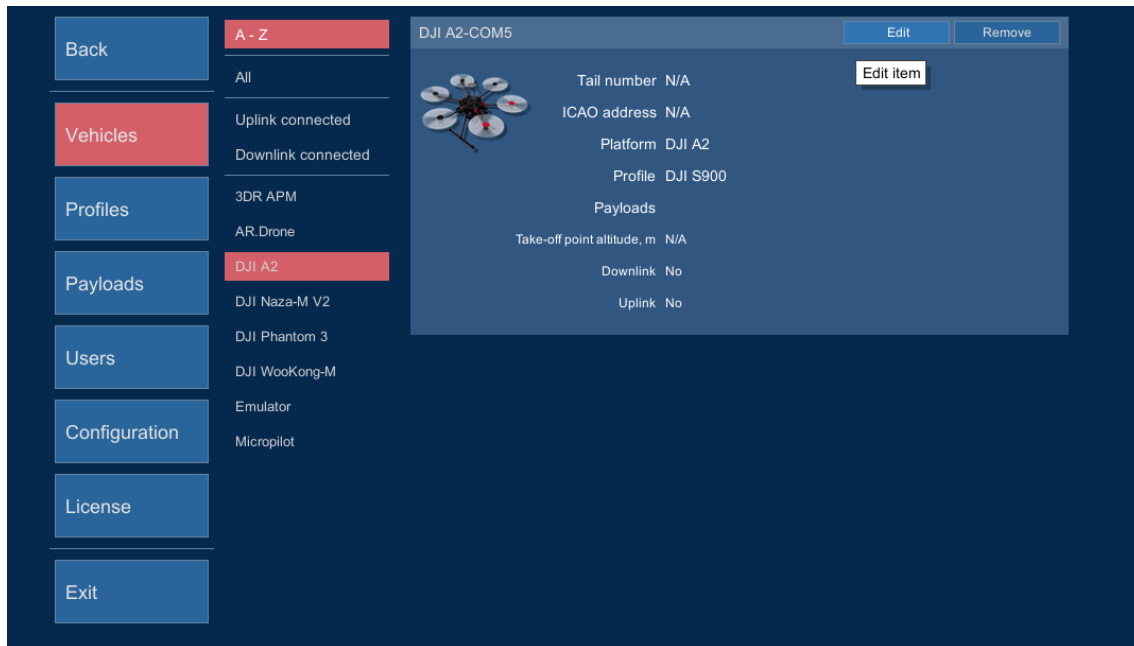


Figure 1: New DJI vehicle

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.

Note

For Wookong M pilots:

Sometimes UgCS detects WKM autopilot as NAZA-M V2. If that is the case, you can set the correct profile manually. After that it will tie the profile with the vehicle automatically.

1.3 Known datalink connection issues

- There is a known issue with DJI 2.4GHz datalink when radio signal is weak. It is possible that DJI Datalink device disappears from Windows device list. The issue manifests itself as disappearing telemetry from the vehicle. To restore connection to the vehicle the datalink must be disconnected from PC and plugged back after ~10-15 seconds. This is a bug in DJI Datalink USB driver which does not break the existing sessions when device is unplugged.
- 2.5 GHz datalink can be unreliable when used with Phantom2. We have experienced datalink loss even when vehicle flies more than 15 meters away from base station.

1.4 Mission execution specifics

- When vehicle acquires GPS lock it records the HOME location to current position.
- Mission upload sets the HOME location of the vehicle according to your mission configuration. (Home location in mission can be set explicitly or set automatically to 1st waypoint.)
- Fail-safe settings in mission properties are ignored.
- DJI has the following default fail-safe settings (these settings can be changed via DJI Assistant software):

Condition	Behavior	Notes
On GPS signal loss	Land	Happens when there are less than 6 satellites visible for more than 20 seconds
On RC signal loss	Return to HOME position	Default RTH altitude is 20m. See autopilot User Manual for more information.
On low battery	Land	See autopilot User Manual for more information.

- Please use DJI Assistant software to control the failsafe settings of autopilot.
- When using Bank or Adaptive bank turn types vehicle can experience random heading changes. It can be due to route segments being too small. Try increasing distance between WPTs.
- Mission waypoint actions supported by DJI:

Flight plan element / action	Support	Notes
Camera control	No	
Camera trigger	Yes	Only Single-shot camera mode is supported for A2 and Wookong autopilots when camera trigger is wired to General Purpose servo output see General Purpose servo action setup for details.
Wait	Yes	Only one wait action per waypoint is allowed.
Yaw	Partial	1) Only when hovering over the waypoint. Vehicle will always fly with nose pointing to next waypoint. 2) Only 1 Yaw action per Waypoint is supported. (In case of multiple yaw actions the last one will be used.) 3) For Yaw action to succeed it must be used together with "Wait" action.
Land	No	Vehicle will hover over the last waypoint until operator takes over the control.
Panorama	No	
Point Of Interest	No	
Camera by time	Yes	Supported for A2 and Wookong autopilots when camera trigger is wired to General Purpose servo output see General Purpose servo action setup for details.

Camera by distance	Yes	Supported for A2 and Wookong autopilots when camera trigger is wired to General Purpose servo output see General Purpose servo action setup for details.
--------------------	-----	--

Warning

- Sometimes DJI autopilot will pause for 5-15 seconds on waypoint with turn type "Stop And Turn" before proceeding to next waypoint even when there is no "Wait" action specified. The issue has been observed with A2 and Wookong autopilots. The cause of said behaviour is unknown.
- Uploading mission in the air is possible but not recommended as it is possible to lose track of altitudes and "Return To Home" functionality can be affected.

1.5 Command execution specifics

Below is a list of commands supported by DJI autopilots.

Supported commands

Command	Support	Notes
ARM	No	
DISARM	No	
AUTOMODE	Yes	Take off and start the mission. See Notes on Auto mode below.
MANUALMODE	No	
CLICK & GO	Yes	
JOYSTICK	Yes	Sets <i>Joystick mode</i> . See also Joystick Mode below.
HOLD	Yes	Should be executed only during mission flight. Used to pause the mission
CONTINUE	Yes	Resume the mission. See Notes on Continue command below.
RETURN HOME	Yes	Vehicle will fly to preconfigured altitude (default is 20m) and return to HOME position and land. See Notes on Return Home command below.
TAKEOFF	No	
LAND	No	
EMERGENCYLAND	No	
CAMERA_TRIGGER	Yes	Trigger camera shutter. See General Purpose servo action setup for details.
DIRECT_PAYLOAD_CONTROL	Yes	Payload control via keyboard/joystick. See also Payload control below.

1.5.1 Notes on Auto mode

If vehicle does not move after successful "Auto mode" command please verify that you have uploaded the route to the vehicle. You can do that by checking log entries.

1.5.2 Notes on Continue command

- Continue command can perform erratically if turn type is *Bank* or *Adaptive bank*. That can happen if mission has small segments. In that case *Continue* command can direct vehicle to skip one or more WPTs.
- For WKM, NAZA and Phantom2 autopilots *Continue* command is ignored if vehicle is in Click&Go mode.
- A2 autopilot can continue mission in "Auto" mode from "Manual", "Click&Go", "Joystick" control modes.

1.5.3 Notes on Return Home command

WKM and NAZA autopilots sometimes ignore the "Return to home" command if mission is uploaded while the vehicle is armed (in the air).

1.6 Joystick Mode

DJI autopilots can be put into Joystick mode which allows user to manually control the vehicle in similar way to RC transmitter. Please note that Joystick mode is inherently more fragile than direct manual control via RC transmitter because it involves many different data links and components:

Joystick device → UgCS Client → UgCS Server → VSM → Ground Radio → Air Radio → Autopilot

If any of the above links fail, the joystick control is broken and vehicle will stay in Joystick mode until mode switch is flipped on the RC transmitter. This is why user should be very careful when working with Joystick control mode. It is recommended to have RC transmitter as backup controller at all times.

Note

There are differences in throttle axis behaviour while in Joystick Mode between autopilots:

- A2, Wookong and NAZA can be landed in Joystick Mode but it is not recommended because it is not possible to disarm the vehicle via joystick.
- NAZA-M and Phantom2 throttle control is somewhat different from RC transmitter: Throttle in the middle returns the vehicle to the altitude where joystick mode was enabled.
- A2 and Wookong throttle control behaves similarly to RC transmitter: Throttle in the middle holds current altitude.

1.7 Command availability

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.

Command availability:

State	Button highlighted	Button shaded
Armed	AUTOMODE, CAMERA_TRIGGER, RETURNHOME	
Disarmed	AUTOMODE, CAMERA_TRIGGER, RETURNHOME	

1.8 Telemetry information specifics

- RC link quality. When using DSM2 receivers sometimes Vehicle will report 100% RC link even when RC transmitter is turned off.
- Vehicle state (armed/disarmed) is controlled from RC transmitter. (Vehicle is armed automatically when "Auto Mode" command is issued)

- Flight mode meaning
 - Auto: Vehicle is executing mission or is returning to HOME position.
 - Manual: Vehicle is holding position.

Note

User can take over the control from any mode at any time by flipping the "Mode Switch" on RC transmitter from "GPS" to "ATTI" to "GPS"

Sometimes DJI autopilot can report mode as "Manual" but cannot be controlled via RC transmitter. To take over the control please flip the "Mode Switch" on RC transmitter to gain manual control.

1.9 Fail-safe actions

Fail-safe actions can be set only in DJI Assistant software.

1.10 Waypoint turn types

There are 3 different routing planning modes for DJI autopilots: fixed-point turn mode (Stop and Turn), coordinated turn mode (Bank Turn) and adaptive coordinated turn mode (Adaptive Bank Turn). You can choose turn type for each Waypoint, Circle, Perimeter. The default turn mode in the system is Stop and turn.

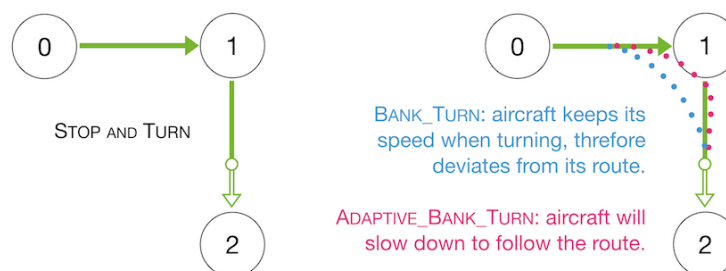


Figure 2: Turn type

Turn type	Support	Notes
Stop and Turn	Yes	Aircraft flies to the first fixed point accurately, stays at the fixed point and then flies to the next fixed point. See note below.
Bank Turn	Yes	The route of aircraft is calculated with turning speed and turning angle. The aircraft would fly from one point to another point without stopping.
Adaptive Bank Turn	Yes	It is almost the same performance with Bank Turn mode. But the flight routine will be more accurately detailed with a planned flight routine in this mode.

Note

When using "Stop and Turn" Vehicle sometimes will wait on a waypoint for several seconds even when there is no wait action specified. Can happen in windy weather.

You can find more information about turning modes and supporting autopilots on the site <http://wiki.dji.-com/>.

1.11 General Purpose servo action setup

DJI A2 and Wookong autopilots support general purpose servo action. It can be used to trigger camera.

You will need a device which can trigger camera remotely. Possible solutions:

- Gimbal with camera trigger connector built in which supports PWM signal. (For example: Zenmuse Z15)
- Stand alone device to trigger camera via PWM signal (For example: http://copter.ardupilot.-com/wiki/common-pixhawk-auto-camera-trigger-without-chdk/#22_IR_trigger-_device)

Steps to configure camera trigger for use with UgCS mission flight:

- Connect gimbal or triggering device to general purpose servo output on autopilot:
 - A2: output "F2"
 - Wookong: output "F1"
 (Consult DJI autopilot manual for more details on how to use GP servo outputs.)
- Configure the servo action in configuration file. By default the "vehicle.dji.trigger_action" parameter in vsm-dji.conf file is set to -1000, 1000, for additional configuration see [Configuring trigger_action parameters in vsm-dji.conf](#) below.
- Create a mission with waypoint actions Camera-mode Single-shot.

1.11.1 Configuring trigger_action parameters in vsm-dji.conf

Parameter trigger_action accepts seven comma separated integers.

Parameter number	Range	Description
1	-1000..1000	Default servo position
2	-1000..1000	1st servo position on action
3	0..9	seconds to hold 1st servo position
4	-1000..1000	2nd servo position on action
5	0..9	seconds to hold 2nd servo position
6	-1000..1000	3rd servo position on action
7	0..9	seconds to hold 3rd servo position

If your trigger needs only 1 servo position change then last 4 parameters can be omitted.

Example:

The following line in vsm-dji.conf:

```
vehicle.dji.trigger_action = 100, -200, 1
```

Produces the behavior:

- Autopilot will keep the GP servo at position 100 during the whole mission.
- On each waypoint which has the "Camera-mode: Single-shot" action the connected servo will:
 - move to position -200,
 - wait 1 second,
 - move back to position 100.

1.12 GoPro video link

Vehicle can be configured to carry GoPro camera. In that case live video stream can be obtained via UgCS video streamer component which must be running on the host which is connected to GoPro WiFi access point.

1.13 Configuration file

Default configuration file of the DJI 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-dji.conf
```

- **On GNU/Linux:**

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

- **On Apple OS X:**

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

1.13.1 Common parameters

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

```
vehicle.dji
```

1.13.2 Serial port configuration

Typically DJI autopilot is connected to UgCS via 2.4GHz or 900MHz datalink. It appears as serial port when USB cable is plugged in. See [Serial port configuration](#) for details.

- **Example:**

```
connection.serial.1.name = COM21  
connection.serial.1.baud = 115200
```

1.13.3 Waypoint turn type

Optional. Used to override the turn type for all waypoints in uploaded mission. Please see DJI GS manual for detailed explanation of turn types.

- **Name:** vehicle.dji.turn_type_override
- **Possible values:** stop, bank, adaptive_bank
- **Default:** stop
- **Example:**

```
vehicle.dji.turn_type_override = adaptive_bank
```

1.13.4 NAZA/Wookong detection

Optional.

There is no deterministic way to differentiate between NAZA and Wookong autopilots. VSM tries its best to autodetect the autopilot correctly but sometimes it can wrongly detect Wookong as NAZA. This setting can be used to workaround the issue. Set this setting to "no" if you are using only NAZA based vehicles.

- **Name:** vehicle.dji.report_naza_as_wookong
- **Possible values:** yes, no
- **Default:** no
- **Example:**

```
vehicle.dji.report_naza_as_wookong = yes
```

1.13.5 Payload control

UgCS supports direct payload control from GCS via joystick or keyboard on four different axes: pitch, roll, yaw, zoom. The following parameters must be set for joystick control to work.

- **Name:** vehicle.dji.payload.data_rate
- **Possible values:** positive integer
- **Default:** 5
- **Description:** Max number of payload control messages per second to send to the vehicle. The bigger the value the smoother the servo movement.
- **Example:**

```
vehicle.dji.payload.data_rate = 8
```

- **Name:** vehicle.dji.payload.[pitch|roll|yaw|zoom].fchannel
- **Possible values:** 1,2,3,4 for A2; 1,2 for Wookong.
- **Description:** F-Channel number on autopilot this axis is connected to. This number should not conflict with [General Purpose servo action setup](#). See above.
- **Example:**

```
vehicle.dji.payload.pitch.fchannel = 1
```

- **Name:** vehicle.dji.payload.[pitch|roll|yaw|zoom].mode
- **Possible values:** position, speed
- **Default:** position
- **Description:** **position** mode maps the payload control input directly to F-Channel servo output. **speed** mode translates control input to rate of change of F-Channel.
- **Example:**

```
vehicle.dji.payload.pitch.mode = speed
```

- **Name:** vehicle.dji.payload.[pitch|roll|yaw|zoom].gain
- **Possible values:** positive real number.
- **Default:** 1.0

- **Description:** Controls the rate of change applied to FChannel.
- **Example:**

```
vehicle.dji.payload.pitch.gain = 1.8
```

Warning

F1 channel on Wookong can behave unexpectedly when used for payload control. Sometimes it jumps to maximum value for short moment. Please test the configured servo outputs before flight.

1.14 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.

1.14.1 UgCS server configuration

VSM can connect to UgCS in two different ways:

- Listen for connection from the UgCS server. See [Listening address](#) and [Listening port](#).
When VSM is configured in listening mode automatic VSM discovery can be set up, too. See [Automatic service discovery](#)
- Initiate connection to UgCS server. See [UgCS server address](#) and [UgCS server port](#).

At least one of the above must be configured for VSM to work.

1.14.1.1 Listening address

Optional.

- **Name:** ucs.local_listening_address = [IP address]
- **Description:** Local address to listen for incoming connections from UgCS server.
- **Default:** 0.0.0.0 (listen on all local addresses)
- **Example:** ucs.local_listening_address = 10.0.0.2

1.14.1.2 Listening port

Optional.

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

1.14.1.3 UgCS server address

Optional.

- **Name:** ucs.address = [IP address]
- **Description:** UgCS server address to connect to.
- **Example:** ucs.address = 1.2.3.4

1.14.1.4 UgCS server port

Optional.

- **Name:** ucs.port = [port number]
- **Description:** UgCS server port.
- **Example:** ucs.port = 3335

1.14.1.5 Retry timeout

Optional.

- **Name:** ucs.retry_timeout = [seconds]
- **Description:** Retry timeout for outgoing server connections in seconds.
- **Default:** 10
- **Example:** retry_timeout = 11

1.14.2 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

1.14.3 Logging configuration

1.14.3.1 Level

Optional.

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

1.14.3.2 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`

1.14.3.3 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

1.14.3.4 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

1.14.4 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`

1.14.5 Command execution control

When vehicle is connected via unreliable link VSM will retry each command several times before failing. This section describes the parameters which control the command execution.

1.14.5.1 Command try count

- **Name:** `vehicle.command_try_count = <number of="" times>="">`
- **Description:** Number of times the command will be issued before declaring it as failed. Must be greater than zero.
- **Default:** 3
- **Example:** `vehicle.command_try_count = 5`

1.14.5.2 Command timeout

- **Name:** `vehicle.command_timeout = <timeout in="" seconds>="">`
- **Description:** Time to wait for response on issued command before retrying.
- **Unit:** second
- **Default:** 1

1.14.5.3 Detection timeout

- **Name:** `vehicle.detection_timeout = <timeout in="" seconds>="">`
- **Description:** Time to wait for vehicle to respond after connection is established.
- **Unit:** second
- **Default:** 6

1.14.5.4 Vehicle serial prefix

- **Name:** `vehicle.serial_prefix = string`
- **Description:** String value used as prefix for all vehicle serial numbers connected to this VSM. Can be used to connect vehicles with equal serial numbers to the same server via different VSMs
- **Default:** not defined
- **Example:** `vehicle.serial_prefix = group1:`

1.15 Communication with devices

VSM can communicate with Vehicle over different communication channels

Currently supported channels:

- Serial port, see [Serial port configuration](#) for details.
- Outgoing TCP, see [Outgoing TCP connection configuration](#) for details.
- Incoming TCP, see [Incoming TCP connection configuration](#) for details.
- Outgoing UDP, see [Outgoing UDP connection configuration](#) for details.
- Incoming UDP, see [Incoming UDP connection configuration](#) for details.
- Incoming UDP (any peer), see [Incoming UDP connection configuration \(any peer\)](#) for details.
- Named pipe , see [Named pipes](#) for details.
- vsm-proxy (XBee), see [Proxy configuration](#) for details.

1.15.1 Serial port configuration

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.

1.15.1.1 Port name

Required.

- **Name:** `connection.serial.[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. [index] is a arbitrary port indexing name.
- **Example:** `connection.serial.1.name = /dev/ttyUSB[0-9]+|com[0-9]+`
- **Example:** `connection.serial.2.name = com42`

1.15.1.2 Port baud rate

Required.

- **Name:** `connection.serial.[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. [index] is an arbitrary port indexing name.
- **Example:** `connection.serial.1.baud.1 = 9600`
- **Example:** `connection.serial.1.baud.2 = 57600`
- **Example:** `connection.serial.2.baud = 38400`

1.15.1.3 Excluded port name

Optional.

- **Name:** `connection.serial.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:** `connection.serial.exclude.1 = /dev/ttyS.*`
- **Example:** `connection.serial.exclude = com1`

1.15.1.4 Serial port arbiter

Optional.

- **Name:** `connection.serial.use_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:** `connection.serial.use_arbiter = no`

1.15.2 Outgoing TCP connection configuration

VSM can be configured to connect to the vehicle via TCP. VSM will try to establish connection to the specified address:port.

Used to connect to vehicle simulator or when vehicle is equipped with WiFi adapter.

1.15.2.1 Remote TCP port

Required.

- **Name:** connection.tcp_out.[index].port = [port number]
- **Description:** Remote port to connect to.
- **Example:** connection.tcp_out.1.port = 5762

1.15.2.2 IP-address for outgoing TCP connection

Required.

- **Name:** connection.tcp_out.[index].address = [IP-address]
- **Description:** IP-address of vehicle to connect to.
- **Example:** connection.tcp_out.1.address = 10.0.0.111

1.15.3 Incoming TCP connection configuration

VSM can be configured to listen for incoming TCP connections from the vehicle. Multiple vehicles are supported on the same port.

Used to connect to vehicle equipped with WiFi adapter.

1.15.3.1 Local listening TCP port

Required.

- **Name:** connection.tcp_in.[index].local_port = [port number]
- **Description:** Remote port to connect to.
- **Example:** connection.tcp_in.1.local_port = 5762

1.15.3.2 Local IP address

Optional.

- **Name:** connection.tcp_in.[index].local_address = [IP-address]
- **Description:** Local ip address to bind to.
- **Default:** 0.0.0.0 (all interfaces)
- **Example:** connection.tcp_in.1.local_address = 127.0.0.1

1.15.4 Outgoing UDP connection configuration

VSM can be configured to connect to the vehicle via UDP. VSM will try to establish UDP connection to the specified address:port.

1.15.4.1 Remote IP-address for UDP

Required.

- **Name:** connection.udp_out.[index].address = [IP-address]
- **Description:** Remote IP-address to send outgoing UDP packets to.
- **Example:** connection.udp_out.1.address = 192.168.1.1

1.15.4.2 Remote UDP port

Required.

- **Name:** connection.udp_out.[index].port = [port number]
- **Description:** Remote UDP port to send outgoing packets to.
- **Example:** connection.udp_out.1.port = 14551

1.15.4.3 Local IP-address for UDP

Optional.

- **Name:** connection.udp_out.[index].local_address = [IP-address]
- **Description:** Local ip address to bind to.
- **Default:** 0.0.0.0 (bind to all interfaces)
- **Example:** connection.udp_out.1.local_address = 0.0.0.0

1.15.4.4 Local UDP port

Optional.

- **Name:** connection.udp_out.[index].local_port = [port number]
- **Description:** Local UDP port to listen for incoming packets on.
- **Default:** 0 (bind to random port)
- **Example:** connection.udp_out.1.local_port = 14550

1.15.5 Incoming UDP connection configuration

VSM can be configured to listen for UDP connections from the vehicle. Vehicle must be actively sending heart-beat/telemetry on specified UDP port before it can be detected by VSM. VSM will automatically detect multiple vehicles on the same port. This is very useful for "drone swarm" setups as there is no need to specify connector for each vehicle and no need to know the IP address of each vehicle in advance.

1.15.5.1 Local UDP port

Required.

- **Name:** connection.udp_in.[index].local_port = [port number]
- **Description:** Local UDP port to listen for incoming packets on.
- **Example:** connection.udp_in.1.local_port = 14550

1.15.5.2 Local IP-address for UDP

Optional.

- **Name:** connection.udp_in.[index].local_address = [IP-address]
- **Description:** Local ip address to bind to.
- **Default:** 0.0.0.0 (bind to all interfaces)
- **Example:** connection.udp_in.1.local_address = 0.0.0.0

1.15.6 Incoming UDP connection configuration (any peer)

This connection type is similar to "udp_in" with the exception that all incoming traffic will be received as one stream. It is used for special purpose connections and cannot be used to connect vehicles.

1.15.6.1 Local UDP port

Required.

- **Name:** connection.udp_any.[index].local_port = [port number]
- **Description:** Local UDP port to listen for incoming packets on.
- **Example:** connection.udp_any.1.local_port = 14550

1.15.6.2 Local IP-address for UDP

Optional.

- **Name:** connection.udp_any.[index].local_address = [IP-address]
- **Description:** Local ip address to bind to.
- **Default:** 0.0.0.0 (bind to all interfaces)
- **Example:** connection.udp_any.1.local_address = 0.0.0.0

1.15.7 Named pipes

VSM is able to communicate with vehicle over named pipe. The pipe must already exist.

- **Name:** connection.pipe.[index].port = pipe_name
- **Description:** Pipe name
- **Example:** connection.pipe.1.name = \\pipe\my_named_pipe

1.15.8 Proxy configuration

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.

1.15.8.1 IP-address for proxy

Required.

- **Name:** connection.proxy.[index].address = [IP-address]
- **Description:** IP-address to connect proxy to. Specify local or remote address.
- **Example:** connection.proxy.1.address = 127.0.0.1

1.15.8.2 TCP port for proxy

Required.

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

2 Disclaimer

DISCLAIMER OF WARRANTIES AND LIMITATIONS ON LIABILITY.

(a) SMART PROJECTS HOLDINGS 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 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.