MicroPilot VSM User Guide

UgCS 2.11.227

Contents

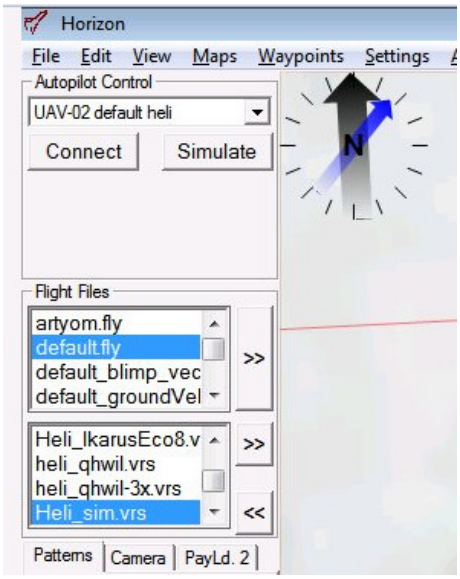
1 MicroPilot VSM User Guide

1.1 First time vehicle connection

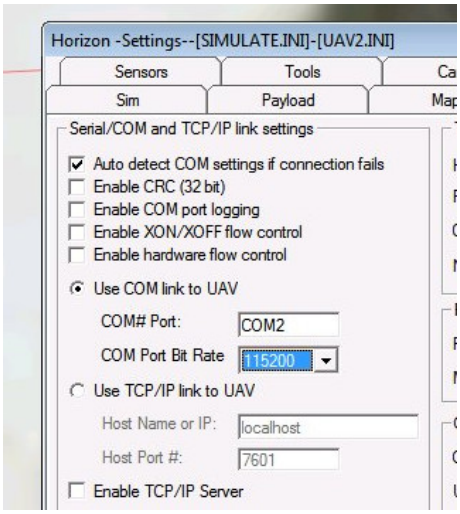
See Disclaimer.

The following instructions are applicable for Horizon simulator only. For working with real Micropilot hardware they may significantly differ due to high flexibility of the autopilot. Please contact UgCS support for assistance.

Select these built-in profiles in Horizon:



Set up serial port for simulation in Horizon settings "Comms" tab:



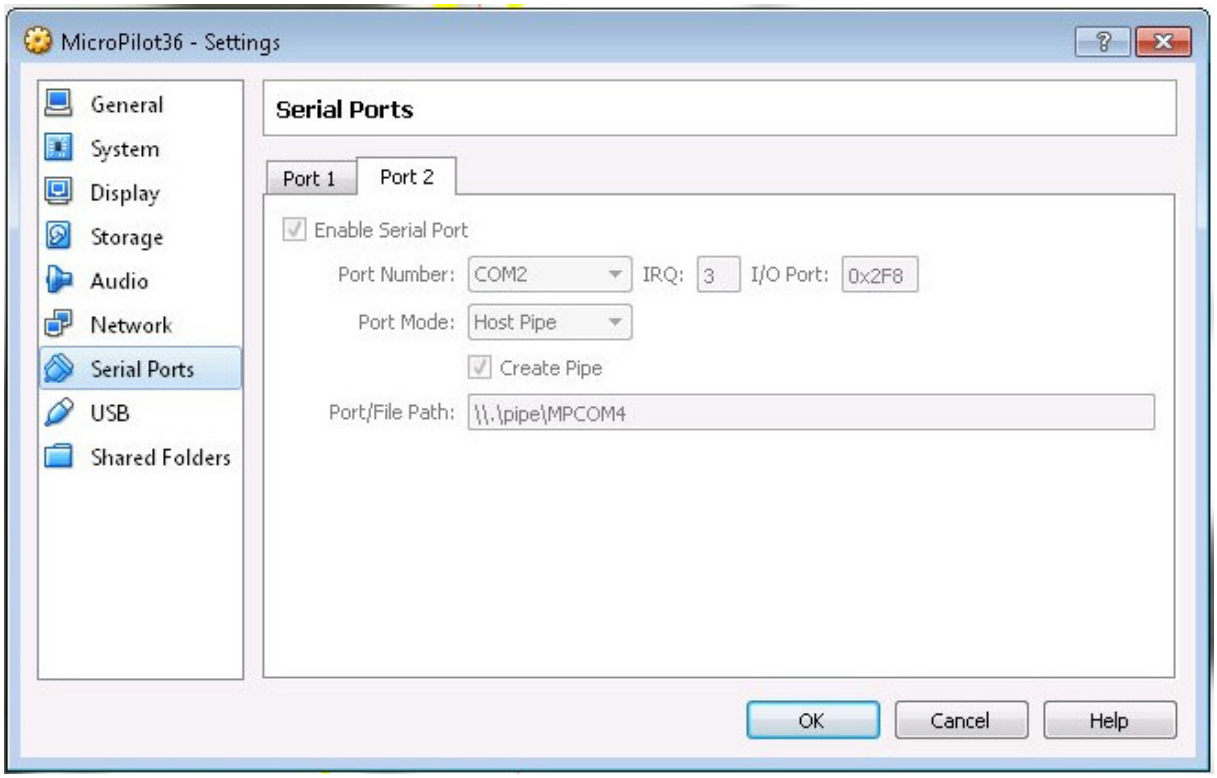
After that you can start simulation by pressing "Simulate" button. The simulated autopilot will be available on the specified serial port. You can connect to it by using some other serial port hardware connected to computer which runs Micropilot VSM. The VSM should have proper port settings which correspond to settings entered in Horizon.

See Serial port configuration section for serial port configuration details in the VSM.

1.1.1 Setup example with virtual machine

Since Horizon simulator provides so ugly interface for accessing simulated vehicle (via hardware serial port), some technique can be used to make it more convenient. It might be using some software which creates virtual serial port (or virtually connected ports pair). Another way is to use virtual machine and its emulated serial port. VirtualBox is used in this example.

Configure emulated serial port. Map it to pipe.



VSM does not support working with pipes so some utility software should be used for proxying data between a pipe and TCP connection. "piper" utility was created for this task (based on VSM SDK) and is available for download on GitHub: https://github.com/UgCS/piper/releases. (Here is other alternative: http://www.hw-group.com/products/hw_vsp/index_en.html#download). Example of configuration ("piper.conf" file which is located aside with piper executable):

```
pipe.1.first.type = pipe
pipe.1.first.name = \\\\.\\pipe\\MPPORT4
pipe.1.second.type = tcp_in
pipe.1.second.local_port = 60004
```

This configuration instructs the utility to open pipe "\\\\.\\pipe\\MPPORT4" and proxy it to TCP port 60004 (on all interfaces). Then you can connect VSM to that port by placing the following lines in VSM configruation:

```
vehicle.micropilot.connection.1.address = 10.0.0.1
vehicle.micropilot.connection.1.tcp_port = 60004
```

IP-address should correspond to machine where the "piper" is running.

1.2 Mission execution specifics

Command	Support	Notes
ARM	Yes	After armed the vehicle starts counting down. If take-off has not been issued after 30 seconds the vehicle is disarmed automatically.
DISARM	No	
AUTOMODE	Partial	Can be used once right after arming to take-off and execute the mission. There is some time after the take-off when the vehicle does not accept any command.
MANUALMODE	No	
CLICK-TO-GO	Partial	Works but has some specifics. Can be issued during mission execution. Once issued, the vehicle should reach the specified point before the mission can be continued by CONTINUE command. Next "Click to go" can be issued only after the mission is continued. Heading specifying is not supported.
JOYSTICK	No	
HOLD	Yes	
CONTINUE	Yes	
RETURNHOME	Yes	
TAKEOFF	No	Send AUTOMODE to take-off and start the mission.
LAND	Yes	
EMERGENCYLAND	No	
CAMERA_TRIGGER	No	

Mission can be executed only once after powering on. When the mission is completed the vehicle lands, and a power cycle and a new mission uploading should be done to launch the vehicle again with a mission.

Flight plan element / action	Support	Notes
Camera control	Yes	
Camera trigger	No	
Panorama	No	
Take-off	Yes	Should be the first mission item if used.
Landing	Yes	Should be the last mission item if used.
Set camera by time	No	
Set camera by distance	No	
POI	Yes	
Change yaw	No	
Wait	Yes	

1.3 Fail-safe actions

GPS Lost:

Action	Result
Wait	Aircraft tries to maintain altitude
Land	Aircraft slowly decends

RC Lost:

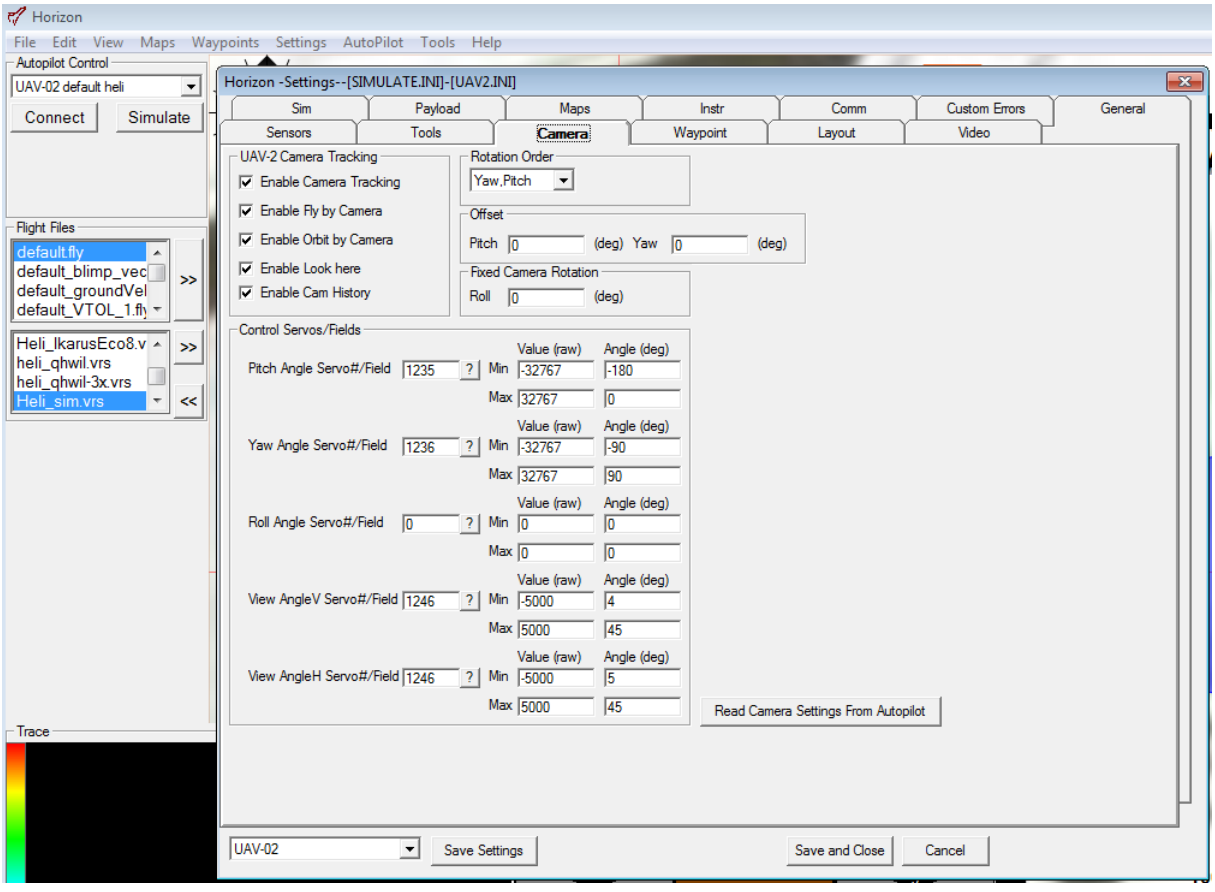
Action	Result
Wait	Aircraft returns home and lands
Land	Aircraft returns home and lands
Return Home	Aircraft changes altitude to failsafe value and returns home
Continue	Aircaft continues mission

Battery Low:

Action	Result
Wait	Stop and wait for user interaction.
Land	Land at current position.
Return Home	If possible aircraft returns home and lands, if not possible slowly decends
Continue	Aircaft continues mission

1.4 Camera setup in Horizon when using simulator

In order to properly see camera footprint in Horizon simulation the camera servo should be configured as on the picture bellow:



Tested with Horizon 3.7.436.0.

1.5 Configuration file

Configuration file location:

- **On Microsoft Windows:**

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

- **On GNU/Linux:**

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

- **On Apple OS X:**

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

1.5.1 Common parameters

All VSMs share a common set of configuration file parameters described in Common configuration file parameters. MicroPilot VSM configuration file prefix is:

```
vehicle.micropilot
```

1.5.2 Mission upload retry count

During mission upload commands can get lost. Retry each mission item this number of times before failing the mission upload. See also Mission upload command timeout. Optional.

- **Name:** vehicle.micropilot.1.command_retry_count
- **Default:** 5
- **Example:**

```
vehicle.micropilot.1.command_retry_count = 3
```

1.5.3 Mission upload command timeout

Milliseconds to wait for accept from vehicle before resending the mission item. See also Mission upload retry count. Optional.

- **Name:** vehicle.micropilot.1.command_timeout
- **Default:** 500
- **Example:**

```
vehicle.micropilot.1.command_timeout = 500
```

1.5.4 Protocol parameters

- **Name:** micropilot.use_crc
- **Description:** Use CRC-32 in autopilot messages when set to non-zero. Otherwise single byte checksum is used.
- **Example:**

```
micropilot.use_crc = 1
```

1.6 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.

1.6.1 UgCS server configuration

1.6.1.1 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

1.6.1.2 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

1.6.2 Logging configuration

1.6.2.1 Level

Optional.

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

1.6.2.2 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

1.6.2.3 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

1.6.2.4 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

1.6.3 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

1.6.4 Automatic service discovery

VSM can respond to automatic service discovery requests form 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

1.7 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.

1.7.1 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.

1.7.1.1 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

1.7.1.2 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

1.7.1.3 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 a 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

1.7.1.4 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

1.7.2 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.

1.7.2.1 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

1.7.2.2 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

1.7.3 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.

1.7.3.1 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

1.7.3.2 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

1.7.3.3 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

1.7.3.4 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

1.7.4 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.

1.7.4.1 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

1.7.4.2 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 congiguration on proxy side.
- **Example:** vehicle.ardupilot.tcp.1.port = 5566

2 Disclaimer

DISCLAIMER OF WARRANTIES AND LIMITATIONS ON LIABILITY.

(a) SPH ENGINEERING MAKE NO REPRESENTATIONS OR WARRANTIES REGARDING THE ACCURACY OR COMPLETENESS OF ANY CONTENT OR FUNCTIONALITY OF THE PRODUCT AND ITS DOCUMENTATION.

(b) SPH ENGINEERING 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.