Table of Contents

1 ADS-B receiver connection

Download this subcategory as PDF file

See Disclaimer.

This VSM supports microADSB-USB receiver Please follow these steps to connect an ADS-B source to the UgCS:

1. To connect an ADS-B source to UgCS, a physical device and a driver for the operating system are required.
2. If using microADSB devices, highly advised to configure other VSM modules to exclude ports used by microADSB. See Excluded port name.
3. UgCS client will detect connected ADS-B sources. If at least one ADS-B source is connected, an indicator will light up in the top right corner of the Client.

4. Collision possibility calculation is based on three parameters: H - horizontal distance (meters) V ? vertical distance (meters) T ? warning time (seconds)
5. Warnings about possible collisions appear in the log window if vehicles, during the minimal convergence, would, in the future, violate both boundaries (H / V) of any other vehicle in a time less than T. A warning is not displayed if the minimal convergence occurred in the past and the vehicles fly apart from one another.

2 ADS-B safety parameters

Download this subcategory as PDF file

The parameters below specify a "safety cylinder" with radius H and height V. Each aircraft has its own cylinder, based on its type. A Kalman filter is used to detect potential convergence, i.e., the possibility that among each pair of aircraft at least one will enter another?s safety cylinder in T seconds. This is considered a pontential collision and a warning will be issued.

? The following values have been adapted for use with UgCS ADS-B collision warning service, among two classes of aircraft:

Values for vehicles controlled by UgCS (assumed to be drones):

Parameter	Parameter	Unit of measurement
H (horizontal distance)	20	Metres
V (vertical distance)	15	Metres
T (warning time)	60	Seconds

Values for vehicles observed by UgCS (assumed to be planes, values based on ICAO recommendations): Parameter Value Unit of measurement

Parameter	Value	Unit of measurement
H (horizontal distance)	9260 (5)	Metres (Nautical Miles)
V (vertical distance)	300	Metres
T (warning time)	60	Seconds

3 Common config 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.

3.1 UgCS server configuration

VSM can connect to UgCS in two different ways:
? Listen for connection form 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.

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`

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`

UgCS server address.
Optional.
? Name: ucs.address = [IP address]
? Description: UgCS server address to connect to.
? Example: `ucs.address = 1.2.3.4`

UgCS server port.
Optional.
? Name: ucs.port = [port number]
? Description: UgCS server port.
? Example: `ucs.port = 3335`

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

3.2 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`

3.3 Logging configuration

Level.
Optional.
? **Name:** log.level = [error|warning|info|debug]
? **Description:** Logging level.
? **Default:** info
? **Example:** `log.level = debug`

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`

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`

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`

4 Config file

Default configuration file of the ADS-B Receiver VSM suits most needs and it is generally not necessary to modify it.

Configuration file location:

? Microsoft Windows:
C:\Program Files (x86)\UgCS\bin\vsm-microadsb.conf

? GNU/Linux:
/etc/opt/ugcs/vsm-microadsb.conf

? Apple OS X:
/Users/[user name]/Library/Application Support/UGCS/configuration/vsm-microadsb.conf

4.1 Common parameters

All VSMs share a common set of configuration file parameters described in Common configuration file parameters. MicroADS-B Receiver VSM configuration file prefix is: `vsm-microadsb`

5 Connecting UgCS and microADS-B Receiver

Download Connecting UgCS and microADS-B Receiver as a PDF book

5.1 Connecting UgCS and microADS-B Receiver

6 Device communication

Download this subcategory as PDF file

VSM can communicate with Vehicle over different communication channels Currently supported channels are below:

6.1 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.

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`

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`

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`

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`

6.2 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.

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`

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`

6.3 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.

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`

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`

6.4 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.

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`

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`

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`

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`

6.5 Incoming UDP connection configuration

VSM can be configured to listen for UDP conncections from the vehicle. Vehicle must be actively sending heartbeat/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.

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`

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`

6.6 Incoming UDP connection configuration (any peer)

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

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`

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`

6.7 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`

6.8 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.

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`

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 congiguration on proxy side.
? **Example:** `connection.proxy.1.port = 5566`