



UGCS DDC

version 3.0

User Manual

Table of Contents

1.1.	Drawing and converting the dance paths	4
1.2.	Adapting dance paths for specific location and requirements	4
1.3.	Executing the dance and its specifics	4
1.4.	DANCE flight mode	5
3.1.	Configuring UCS Server	8
3.2.	Configuring Vehicle Specific Module (VSM)	9
3.2.1.	Connection related settings.....	9
3.2.2.	Regular GPS and RTK GPS related settings	9
3.2.3.	Settings related to light control using Flight controller	10
3.2.4.	Takeoff and moving to dance position time offset settings.....	10
3.2.5.	Other routing related settings.....	10
3.2.6.	Group Arducopter parameter change using vsm-ardupilot.conf file.....	11
3.3.	Configuring UgCS DDC	11
3.3.1.	Names of the vehicles used	11
3.3.2.	Vehicle group settings	11
4.1.	Geo-fence settings	13
4.1.1.	Geo-fence radius, altitude and polygon in DDCClient.exe.config file	13
4.1.2.	Geo-fence settings from vsm-ardupilot.conf file	14
5.1.	Actions and Options tab.....	16
5.1.1.	Dance menu (Commands to upload paths)	19
5.2.	Vehicle selection tab	19
5.3.	Filter by Operator or status tab.....	20
5.4.	Commands to start dance	20
5.5.	Commands to end dance	20
5.6.	Columns	20
5.6.1.	Last Cmd Column	21
5.6.2.	Start Time Column	21
5.6.3.	Onboard Time Column	21
5.6.4.	Path Column	21
5.6.5.	Mode Column	21
5.6.6.	Vehicle status column	21
5.6.7.	Altitude column.....	22
5.6.8.	Speed Columns	22
5.6.9.	Battery Column	22

5.6.10.	GPS Fix Column	22
5.6.11.	Vibrations Column	22
5.6.12.	Datalink column	22
5.6.13.	Path length column	22
5.6.14.	Offset To Start column	22
5.6.15.	Max Relative Alt Column	23
5.6.16.	Max Distance from Center Column.....	23
5.6.17.	Max Horizontal & Vertical Speed Columns.....	23
5.6.18.	FPS Column	23
5.6.19.	Comm Method Column	23
5.6.20.	Firmware Column	23
5.6.21.	Displays UAV firmware hash.....	23
6.1.	Before show preparations.....	24
6.2.	Connecting vehicles and selecting paths.....	24
6.3.	Setting the Dance Center	24
6.4.	Taking off and running the show	25
7.1.	Introduction.....	30
7.2.	Requirements.....	30
7.3.	Configuration.....	30
8.1.	Architecture.....	33
8.2.	Hardware	34
8.2.1.	Radios	34
8.2.2.	GPS.....	34
8.3.	Software	35
8.3.1.	Installing the Drivers.....	35
8.3.2.	RTK Client	35
8.3.3.	RedButton	37

1. Overview

DDC 3.0 group flights are based on animations, which are converted into special Path files. Flight process is time synchronized (GPS time) and runs autonomously. To provide dance functionality, the process utilizes special flight controller firmware, that features a custom Dance flight mode.

This section is meant to provide insight on how the Dance process works and what the requirements are.

1.1. Drawing and converting the dance paths

The way vehicles will fly and light up during the dance is first defined using animation software, for example Blender. Essentially each vehicle is represented by a dot in animation and the dots have 3-dimensional position that can be transferred to any geographic location. Vehicle light control can be defined in animation or controlled separately.

While drawing animation, it is obligatory to make sure that flight lines do not cross and that all vehicles have safe distance from each other at all times. After the animation is ready, it is can be proven for collisions using our script and then converted into special Path files which contain the actual flight trajectory and light settings.

1.2. Adapting dance paths for specific location and requirements

When a compatible animation is ready, it can be modified during conversion process to increase distance between vehicles. After the Path files are generated, operator can define the speed at which the animation will be executed – one can choose from 4 frames per second (typical) to 1 frame per second which will result in a slow and longer flight.

Before running the show, operator must define center coordinates, the angle at which the show will be executed relative to North and time to start the show after the vehicles will take off. This is done in **UgCS DDC**.

1.3. Executing the dance and its specifics

To run a **UgCS DDC 2.0** dance show, operator must follow these steps:

1. PLACEMENT

- Place vehicles in correct formation on ground and acquire stable position on them

2. UPLOAD PATH

- Upload paths to vehicles and check if all vehicles have rebooted

3. SET DANCE CENTER, HEADING AND FENCE

- Set correct dance formation center, heading and upload it, upload fence

4. CHECK OFFSET TO START
 - Check that horizontal and vertical offset to start is within safe limits for all vehicles
5. ARM VEHICLES
6. SET TIME & UPLOAD TIME
 - Set the required time to dance start and upload time
7. START DANCE
 - Until the set time reaches 20 seconds to start, click Dance and watch the show
8. LAND
 - When show is done, land the vehicles

1.4. DANCE flight mode

To perform time synchronized flights, SPH Engineering have developed custom Flight controller firmware, that includes special Dance flight mode.

In contrast to standard AUTO mode, in DANCE mode the vehicle flies along a given 3d-position function of time. The Dance trajectory may be described as a list of points with fixed time constant between each point. Each point has a local X-Y-Z coordinate and RGB color value. The trajectory length can be up to 1000 points. Common flight speed is 4 points per second, yielding maximum flight time about 4 minutes. It is possible to increase flight time by decreasing FPS (points per second).

Dance mode also allows to set the exact time of flight start. Therefore, operator is able to create consistent trajectories for any number of UAVs and run UAVs into an autonomous flight at the same time with help of GPS time. This also makes start of show precisely synchronous for all UAVs.

2. UgCS DDC installation

To successfully start using **UgCS Drone Dance Controller (DDC)**, it needs to be installed and configured accordingly.

There are no specific settings required to be changed during the installation. Although it is possible to install only those components which will be used on the specific computer (multi-node installation). It is recommended to install all components, since each of the components don't take up much disk space and can be used in emergency situations to replace any other PC.

If the user would like UCS Server (map and mission data storage, routing, mission calculation) to run automatically with computer start or configure advanced settings, one must select the following settings during **UgCS DDC** installation:

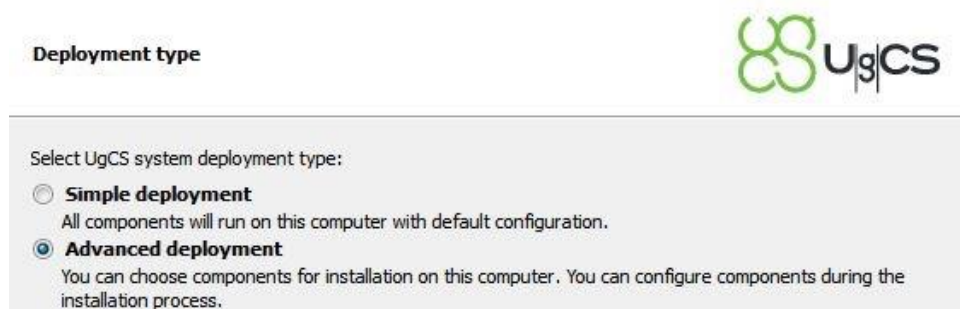


Image 1. Select “Advanced deployment” to automatically run UCS Server after Windows boot

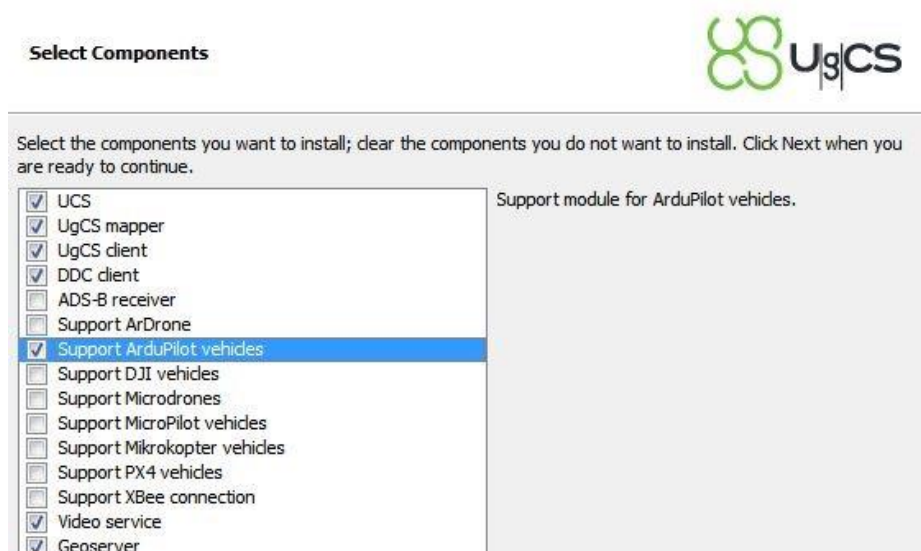


Image 2. Selecting essential components for installation

To make the components run during system startup, one must select to install **UgCS Server**, Pixhawk/APM Vehicle Specific Module (VSM) and Geoserver (used for custom map overlay and elevation data) as system service.

Additional connection parameters can be set during the installation, but those are meant for Virtual COM Port communication configuration and not TCP/UDP configuration.

3. UgCS DDC configuration and deployment

This section contains information on how to set up UgCS DDC for multi-node deployment.

For multi-node deployment the same UgCS DDC version should be installed on all PCs in question. It is recommended to install all UgCS DDC components on all PCs in question, but it is not obligatory to do so.

The recommended lay-out would be to have UCS Server and VSM running on one PC and UgCS DDC Client and UgCS Client running on the rest of available PCs.

NOTE: It is mandatory, that all PCs can access the PC running UCS Server and VSM trough network.

3.1. Configuring UCS Server

On the PC that will be running UCS Server, VSM and Geoserver processes, it is not necessary to run UgCS DDC Client and UgCS Client.

In order to set up the rest of the PCs running UgCS DDC Client and UgCS Client, it is necessary to know the IP address of the PC running UCS Server and VSM. Easiest way to find it out is to use “ipconfig” command from command prompt in Windows.

This IP address will be used on the PCs running UgCS DDC Client and UgCS Client to connect to.

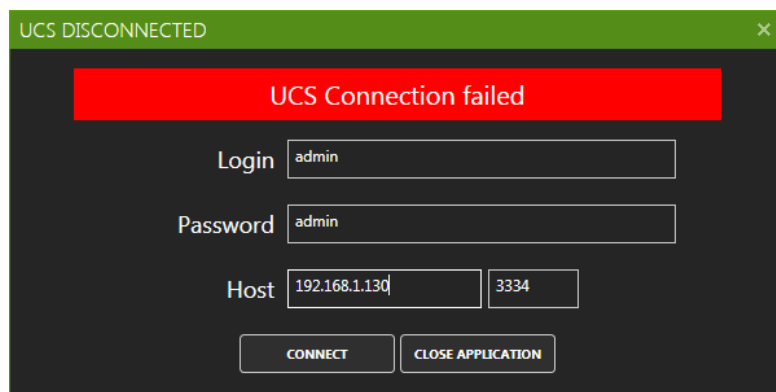


Image 3. Entering user credentials and IP address of the Server.

It is also recommended to change login and password and add new operator users to add another layer of security. To do so, one must open UgCS Client on the machine running UCS Server and go to “Users” tab in Main Menu. At first, there is one user – “admin” with password “admin”, change these parameters as preferred and create new users with “Operator” role for ease of use.

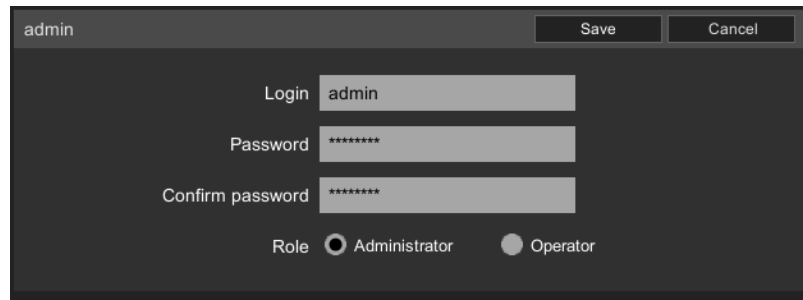


Image 4. User login and password settings in “Users” tab on Main Menu (UgCS Client).

3.2. Configuring Vehicle Specific Module (VSM)

Vehicle Specific Module is used to connect to the vehicle and governs advanced routing settings. To configure VSM it is necessary to edit the `vsm-ardupilot.conf` file which is typically located in

C:\Program Files (x86)\UgCS\bin or the specified installation directory.

3.2.1. Connection related settings

To start connecting the VSM to the available vehicles using UDP protocol, it is necessary to indicate which ports the VSM must listen for available devices (flight controllers).

The lines 36-38 of `vsm-ardupilot.conf` file contain the following text:

```
# Vehicle can be connected via UDP.
# UDP port which will listen for incoming mavlink messages.
connection.udp_in.1.local_port = 14550
```

To configure the connection to vehicles it is recommended to set the local port unique for each vehicle.

Example configuration for 3 vehicles:

```
connection.udp_in.1.local_port = 14561
connection.udp_in.2.local_port = 14562
connection.udp_in.3.local_port = 14563
```

Notice how both port and number of vehicle (1, 2, 3) changes. Make sure to restart VSM after making any changes to start using them.

3.2.2. Regular GPS and RTK GPS related settings

Within `vsm-ardupilot.conf` file there are two more very important settings which govern how **UgCS** calculates route altitude. By default these settings are configured for use with regular GPS modules.

NOTE: If the fleet of vehicles has RTK GPS setup and Base unit, these two lines within `vsm-ardupilot.conf` file must be uncommented (the “#” from beginning of the line must be removed) and should look like this:

```
vehicle.ardupilot.report_relative_altitude = no
```

```
vehicle.ardupilot.set_ground_alt_offset = no
```

- Report relative altitude – assumes that ALT value from flight controller is not absolute
- Set ground altitude offset – assumes that vehicle is on ground level and AGL altitude during route upload is 0

3.2.3. Settings related to light control using Flight controller

If one has chosen to follow [UgCS Dance Payload design](#) or has a different Light Payload that is controlled by pre-programmed servo actions from within flight controller, it is necessary to add the following lines to vsm-ardupilot.conf file.

```
# G
vehicle.ardupilot.parameter.SERVO9_FUNCTION = 82
# B
vehicle.ardupilot.parameter.SERVO10_FUNCTION = 83
# R
vehicle.ardupilot.parameter.SERVO11_FUNCTION = 81
```

Each line corresponds to a light color. The lights need to be mapped because the servos are controlled from within the flight controller and not from UgCS or DDC Client.

Note: fence will work both in Auto mode as well as all Manual modes. SET_SERVO will work while vehicles are on the ground. But when they are in DANCE mode, SET_SERVO will be overridden by DANCE RGB values. New function BLINK will work even when drones are dancing.

3.2.4. Takeoff and moving to dance position time offset settings

UAVs are intended to make autonomous takeoff and to fly to dance start position. To reduce potential chaos we introduce two parameters:

```
# Takeoff time offset
vehicle.ardupilot.parameter.DANCE_TK_OFF = 60
# Moving to position time offset
vehicle.ardupilot.parameter.DANCE_MV_POS = 45
```

Both of parameters means offset in seconds prior to Dance start. DANCE_TK_OFF sets time offset for taking off. So if DANCE_TK_OFF = 60, UAVs will take off a minute before dance start. DANCE_MV_POS sets time offset for drones to moving to dance starting position. So if DANCE_MV_POS = 45, UAVs will start to move to first path coordinate a 45 seconds before dance start.

In DDC client these values are shown in Start Time column.

3.2.5. Other routing related settings

Another important routing setting that can be configured within `vsm-ardupilot.conf` file is enabling and disabling of autoheading. If autoheading is enabled, the route will be created in a way that vehicle will always face the following waypoint. If autoheading is disabled, vehicle will not change its yaw angle unless changed using “Yaw” action for corresponding waypoint. The behavior is governed by changing the following line within `vsm-ardupilot.conf` file:

```
vehicle.ardupilot.autoheading = no
```

3.2.6. Group Arducopter parameter change using `vsm-ardupilot.conf` file

It is possible to change one or more Ardupilot configuration parameters by changing `vsm-ardupilot.conf` file. There is one sample line within `vsm-ardupilot.conf` file. When parameters are added to this file, they are changed within Ardupilot when VSM connects to vehicles. For example, if one would like to set action on loss of RC = N/A, to disable corresponding fail-safe, one could add the following line to `vsm-ardupilot.conf` file and restart the ArduPilot VSM:

```
vehicle.ardupilot.parameter.FS_THR_ENABLE=0
```

Once vehicles have been discovered by the VSM, the line can be removed from the config file.

3.3. Configuring UgCS DDC

To properly set up **UgCS DDC** one must edit the `DDClient.exe.config` file typically located in `C:\Program Files (x86)\UgCS\client-DDC`. The following settings should be changed in this configuration file:

- Names of the vehicles used
- Geo-fence radius, altitude and polygon
- Edit vehicle groups

```
<add internalid="1" name="APM-101" defaultroute="1" supervisor="User 1"
fenceradius="100" fencealtitude="120" polygonfence="false"/>
```

This is one line from `DDClient.exe.config` containing the vehicle settings for reference.

3.3.1. Names of the vehicles used

It is necessary to specify the names of the vehicles that will be used, before any vehicles are displayed in **UgCS DDC**. By default the vehicle names in `DDClient.exe.config` file are set to 1, 2, 3.., etc. (`name="1"`, `name="2"`, `name="3"`,...). It is possible to either set the vehicle names for connected vehicles according to this within **UgCS** client or to change the `DDClient.exe.config` file to match your connected vehicles.

3.3.2. Vehicle group settings

Groups are useful to access quickly a certain subset of drones in DDC user interface. The grouping logic can be different and depend on the drone start positions, number of operators in control or formation peculiarities.

```

<groups>
  <add internalid="1" name="Group-1" vehicles="1,2,3,4,5,6,7"/>
  <add internalid="2" name="Group-2" vehicles="8,9,10,11,12,13, 14"/>
  <add internalid="3" name="Group-3" vehicles="15,16,17,18,19,20"/>
</groups>

```

This sample illustrates how 20 vehicles could be split up to be controlled by three operators. One vehicle can have infinite number of operators (two operators can control the same vehicle) and it is possible to add new groups just by creating a new line by the same pattern, for example:

```

<add internalid="4" name="Group-4" vehicles="7, 10, 14, 19"/>

```

Worth to mention is that if several DDC clients are being used, the configuration has to be performed for each node.

4. Safety settings and precautions

4.1. Geo-fence settings

Setting up geo fence helps prevent vehicles from flying out of defined flight space. It is possible to set Geo-fence using two ways – either define fence parameters for each vehicle in DDCClient.exe.config file or use vsm-ardupilot.config file to set fence. Setting fence from DDC Client allows more precise setting and specific settings for each drone.

Note: Geo-fence is calculated from the point where vehicles are started (Home Location). Operator must either set the fence in DDCClient.exe.config file or in vsm-ardupilot.config file. Never use both options at once.

4.1.1. Geo-fence radius, altitude and polygon in DDCClient.exe.config file

Geo-fence function consists of two parts:

- Cylindrical fence
- Polygonal fence

The **cylindrical radius fence** is the primary fence function. It can be used separately or together with polygon fence. By changing these two parameters – *fenceradius* and *fencealtitude* it is possible to define specific flight range limits for each vehicle. These parameters are defined in m (meters) from the starting point (Home location) – H as shown in Image 5. The action which vehicle will take after fence breach can be defined in Mission Planner.

Additionally to the cylindrical fence, it is possible to set **polygonal fence**. To enable polygon fence it is necessary to set parameter `polygonfence="true"`. When the polygonal fence is used together with cylindrical fence, fence breach is triggered as soon as vehicle breaches either cylindrical fence or polygonal fence border. In case of using both types of fence, the permitted flight area is where cylindrical radius fence and polygon fence overlap like shown in Image 5. In order to restrict the permitted flight area only by polygon fence, the radius fence should be larger than the intended polygon.

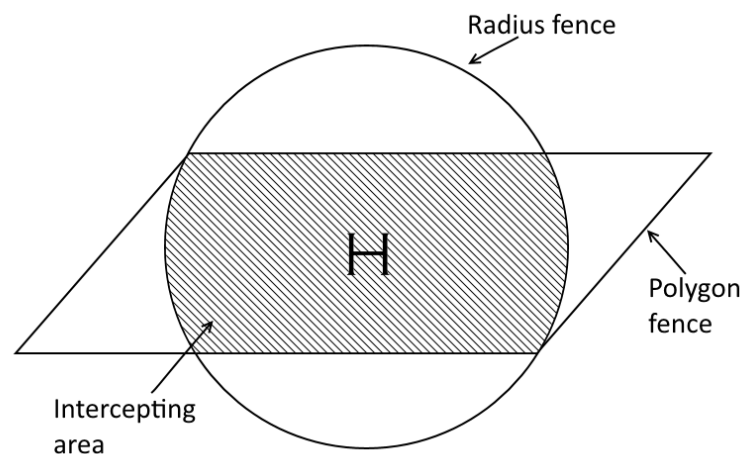


Image 5. Diagram displaying the permitted flight area within geo-fence

The setup of polygonal fence requires several steps. First, user needs to create a new route in **UgCS Client**



and mark the desired fence area using “Perimeter tool” in **UgCS**. When the area is marked, it is necessary to export the route as a *.XML file and open the file with a text editor. The necessary lines should look like this:

```
<ugcs-List name="points" type="FigurePoint" v0="id" v1="version" v2="order" v3="latitude"
v4="longitude" v5="wgs84Altitude" v6="aglAltitude" v7="altitudeType">
  <o v2="0" v3="0.5900877312384658" v4="-1.4685816967389724" v6="0.0" v7="AGL"/>
  <o v2="1" v3="0.5900910492826061" v4="-1.4685958825188987" v6="0.0" v7="AGL"/>
  <o v2="2" v3="0.5900957350709617" v4="-1.4685953519569892" v6="0.0" v7="AGL"/>
  <o v2="3" v3="0.5900981341526812" v4="-1.468586617752834" v6="0.0" v7="AGL"/>
</ugcs-List>
```

Next edit the *DDClient.exe.config* file and paste these lines within `<polyfence>` `</polyfence>` tags. Now save the changes and restart DDC client for the changes to take effect.

Note: fence will work both in DANCE mode as well as all Manual modes.

4.1.2. Geo-fence settings from *vsm-ardupilot.conf* file

If preferred by operator, it is possible to set Geo-fence using VSM parameters. The Geo-fence will be applied as soon as a vehicle is connected to VSM. Typical parameters include:

```
vehicle.ardupilot.parameter.RTL_ALT = 600
```

-Altitude (in centimeters) at which vehicle will return to Home Location

```
vehicle.ardupilot.parameter.RTL_ALT_FINAL = 600
```

- Altitude at which Return to Home operation will be finished (if set to 0, vehicle will land)

```
vehicle.ardupilot.parameter.FENCE_ACTION = 1
```

-If fence action is set to 1, vehicle will return to Home in case of fence breach

```
vehicle.ardupilot.parameter.FENCE_ALT_MAX = 15
```

-Maximum fence altitude in meters (from Home location)

```
vehicle.ardupilot.parameter.FENCE_RADIUS = 20
```

-Maximum fence radius in meters (from Home location)

```
vehicle.ardupilot.parameter.FENCE_ENABLE = 1
```

-If set to 1 fence is enabled

vehicle.ardupilot.parameter.FENCE_TYPE = 3

-Fence type 3 defines radius and altitude fence

5. UgCS Client and UgCS DDC Client show operation guide



Image 14. DDC Client main view with marked control areas

The UgCS DDC Client consists of 6 main control areas and a spreadsheet type list of vehicles. Each of the areas serve a different kind of purpose.

5.1. Actions and Options tab

Following commands are in “Actions” tab (Nr. 1 in the Image 14):

- **Reload Vehicle list** – will refresh the vehicle list and check for any changes
- **Load new dance** – opens up context menu to change to dance file folder
- **Set TO point altitude** – sets take-off altitude to align all vehicles evenly above ground (**entirely informative**)
- **Turn off the lights** – to power off lights on vehicle (if controlled by Flight Controller)
- **Blink UAVs** – lights the payload for two short periods (all colors)
- **Set servo** – use to set specific PWM on selected servo (can be used to change light or trigger fireworks)
- **Manual mode** – sets the selected vehicles to Manual mode
- **Adjust last point** – sets the last route point of selected vehicles to current vehicle location
- **Upload settings** – opens a dialog which allows operator to upload *.PARAM file with flight controller parameters

- **Reboot vehicles** – Restarts the flight controller to re-initialize all sensors
- **Exit** – Close this menu

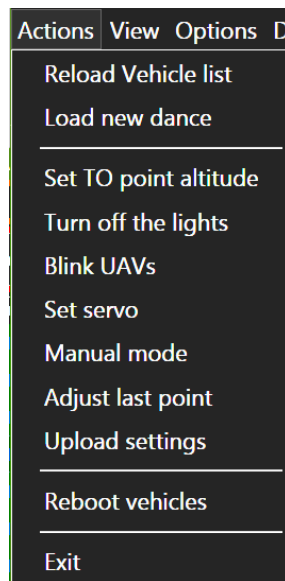


Image 15. “Actions” menu content

In “View” tab button “View server logs” displays the window with UCS Server log messages.

In “Options” tab the “General options” button opens up a window with following controls:

- **Arm confirm** – enables or disables confirmation message before executing Arm command, same for Disarm, Hold, Land, Return to home, Reboot and Set servo commands
- **Arm interval** – adds time in s (seconds) after which the Arm command will actually be sent to the vehicles, same for Disarm, Hold, Land and Return to home commands
- **Blink color Value [0-255]** – sets the color intensity of each color (Blue, Green, Red) during “Blink” action
- **Blink rate [Hz]** – sets the frequency of “Blink” action
- **Blink time (millisecond)** – sets the duration of “Blink” action in milliseconds
- **Dance Center AMSL** – defines the altitude at which dance will be executed (meters, AMSL)
- **Dance center Heading** – defines the angle to North at which the dance will be executed (degrees)
- **Dance center Latitude/Longitude** – defines Latitude and Longitude of the Dance center
- **Dance Max Altitude to First Point (meters)** – sets the maximum safe altitude from take-off to first point (informative)
- **Dance Max Distance to First Point (meters)** – sets the maximum safe distance from take-off to first point (informative)

- **Dance Min Altitude to First Point (meters)** – sets the minimum safe altitude from take-off to first point (informative)
- **Dance Start Interval (in seconds)** – sets the time offset (will be added to current GPS time) in seconds after which dance will be executed
- **Default timeout for command** – timeout after which command will be regarded as not executed or failed
- **Execute hold before RTL** – enables or disables UgCS DDC executing “hold” command before executing Return to Home command
- **Host** – sets the host IP which to connect to (UCS Server)
- **Login** – sets the login username
- **Measurement** - Sets between Imperial or Metric measurement units
- **Password** – allows to change Users password
- **Port** - allows to change connection port
- **Vibration threshold XY, Z** – sets the amount of maximum vibrations on specified axis (informative)

In “Help” tab buttons “Support” and “Contacts” are located, that lead to according pages on UgCS website.

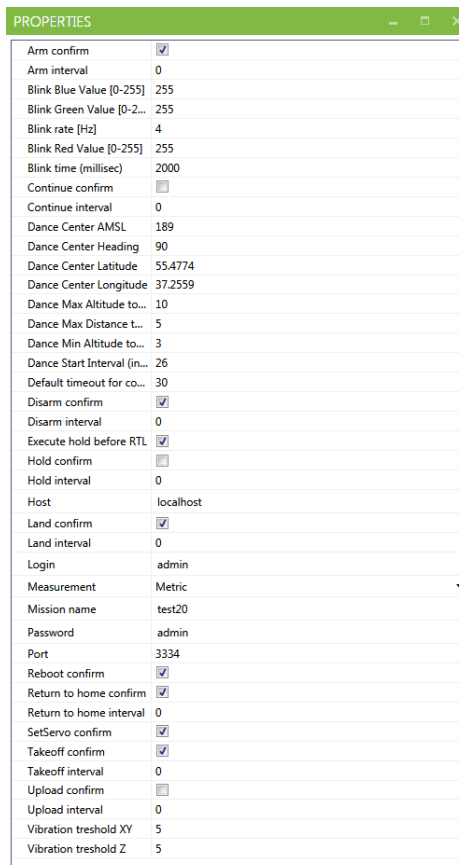


Image 16. General options window showing the available controls

5.1.1. Dance menu (Commands to upload paths)

Dance menu contains the buttons associated with Path and Fence upload. It consists of the following commands:

- **Upload Path** – uploads the selected Path to vehicle
- **Upload Location** – uploads the location (Dance Center) set in Options menu to vehicle
- **Upload Fence** – uploads the fence settings set in DDCClient.exe.config file

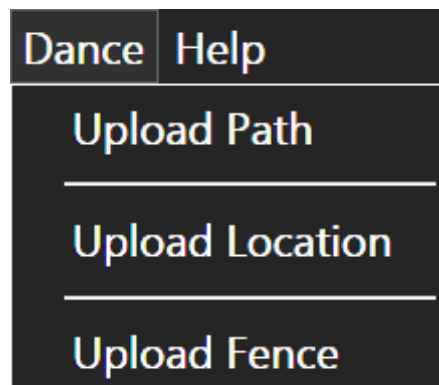


Image 17. Dance menu showing the available controls

5.2. Vehicle selection tab

The vehicle selection tab (Nr. 2 in the Image 14) consists of the following commands:

- **Select All** - selects all vehicles
- **Clear All** – clears the selection of all selected vehicles
- **Select Failed CMD** – selects only the vehicles which last command had failed
- **Select by Mode** – selects vehicles which state conform to the state set. There are three states available to select:
 - Connected
 - Disconnected
 - Disarmed
 - Armed

5.3. Filter by Operator or status tab

The filter by Operator or status tab (Nr. 3 in the Image 14) allows operator to instantly show only the vehicles which conform to the status chosen in this tab. The following options are available:

- **All** – displays all vehicles
- **Armed** – displays only the vehicles which are armed
- **Disarmed** – displays only the disarmed vehicles

Group – By default there are three Groups available which different operators may use for displaying only the vehicles needed. It is also possible to add new groups.

5.4. Commands to start dance

The following commands are available in Route starting tab (Nr. 4 in the Image 14):

- **SET TIME** – Adds the defined start time from “General Options menu” to current vehicle time (sets the beginning of show)
- **UPLOAD TIME** – Uploads the defined start time to vehicles
- **DANCE** – Sets the vehicles to Dance mode (starts the flight)
- **ARM** – arms the motors (should be executed before Set Time, Upload Time and Dance commands)

5.5. Commands to end dance

The commands in this tab (Nr. 5 in the Image 14) controls the actions needed to end route execution.

- **Return to Home** – Makes the vehicle reach pre-set return to home altitude and return to home location
- **Hold** - Pauses mission execution and makes vehicle hover in current position
- **Land** - Executes landing algorithm – vehicle will land in current position
- **Disarm** – Disarms the motors

5.6. Columns

5.6.1. Last Cmd Column

Displays last command name that was send to UAV. It is green if command was successful and red if it was not.

5.6.2. Start Time Column

Displays absolute time (UTC) of dance starting and time offsets. Start time column has the following format: HH:MM:SS (X/Y/Z) where

HH:MM:SS - absolute UTC dance starting time

X - time left to takeoff for corresponding UAV (in seconds)

Y - time left to moving to start position for corresponding UAV (in seconds)

Z - time left to start the dance for corresponding UAV (in seconds)

5.6.3. Onboard Time Column

Displays current onboard time (UTC) of UAV. This time is taken from GPS sensor.

5.6.4. Path Column

Allow to manual select path-file for UAV. Can be filled automatically with Assign Path command in Dance menu.

5.6.5. Mode Column

- **Disarmed, Manual**– Vehicle is ready for ARM and DANCE
- **Armed, Manual** – Vehicle is ready for DANCE command and take-off
- **Armed, Auto, Flying to point** – Vehicle is executing DANCE
- **Armed, Auto, Landing** – Vehicle is Landing
- **Disarmed, Auto, Landing** – After flight vehicle must be set to Manual mode (Actions/Manual mode) to execute DANCE again

5.6.6. Vehicle status column

For essential information about vehicle status in **UgCS DDC** a column displays current vehicle state. There are four different statuses:

- **Standby** – Vehicle is ready for take-off
- **Airborne** – Vehicle is currently airborne
- **Not ready** – Vehicle is not yet ready for take-off
- **Critical** – Vehicle battery is low or flight controller reports sensor problems
-

5.6.7. Altitude column

Displays GPS altitude and altitude above ground level. Altitude column has the following format: X/Y (Z) where

X – raw GPS altitude, directly from sensor

Y – filtered GPS altitude

Z - Above Ground Level altitude

5.6.8. Speed Columns

Vertical and Ground Speed columns display current speed of vehicle.

5.6.9. Battery Column

Displays Battery status in percent and Volts. Can be specified in **UgCS** client according to battery used.

5.6.10. GPS Fix Column

The following GPS Fix Statuses are possible:

- **No Fix** – GPS position has not been fixed, not ready for flight
- **2D Fix** – GPS Position fixed in two dimensions, not ready for flight
- **3D Fix** – GPS Position fixed in three dimensions, if not using RTK GPS receiver, ready for flight
- **DGPS Fix** - GPS Position fixed in three dimensions with improved accuracy, if not using RTK GPS receiver, ready for flight
- **RTK Float** – RTK GPS Position fixed with acceptable accuracy, ready for flight
- **RTK Fix** – RTK GPS Position fixed with perfect accuracy, ready for flight

5.6.11. Vibrations Column

Displays current vibrations of autopilot. If vibrations are larger than threshold text is colored red.

5.6.12. Datalink column

Displays current datalink level. 100% means best level and 0% means no datalink.

5.6.13. Path length column

Displays length of path currently loaded in autopilot memory. Measured in number of trajectory points. So path with length of 160 points will be flown in 40 seconds with FPS = 4 (160 / 4 = 40).

5.6.14. Offset To Start column

Displays horizontal and vertical distances to first path point from current UAV location in meters. Useful for detection problems of mission location. When values are larger than a threshold column is painted with red.

5.6.15. Max Relative Alt Column

Displays maximum relative altitude UAV will climb during current loaded path.

5.6.16. Max Distance from Center Column

Displays maximum distance from dance center (not from current UAV position!) UAV will flow away during current loaded path.

5.6.17. Max Horizontal & Vertical Speed Columns

Display maximum horizontal and vertical speed UAV will reach during current loaded path.

5.6.18. FPS Column

Displays current FPS settings UAV will flow loaded path with. For example path with length of 160 points with FPS = 4 will be flown in 40 seconds ($160 / 4 = 40$). Lower FPS to increase the duration of mission and to decrease the speed of UAV and vice versa.

5.6.19. Comm Method Column

Displays current connection type and address of corresponded UAV.

5.6.20. Firmware Column

5.6.21. Displays UAV firmware hash.

6. Typical workflow

This section contains information on how a typical drone dance event is planned and executed. It is meant to introduce users to main functions and steps required to run a drone show with UgCS DDC.

6.1. Before show preparations

When planning a show, it is mandatory to have an appropriate set of *.PATH files ready and tested. Show operator must know the formation of vehicles from which the dance starts and best to plan ahead the exact position each vehicle will take off from. All vehicles should be tested in flight individually prior to doing a group (dance) flight.

6.2. Connecting vehicles and selecting paths

First step to start off is to create a new Vehicle Group in DDCClient.exe.config which will contain only the vehicles planned for the show. Next when first time opening UgCS DDC it will ask to point to the folder which contains the .PATH files for the show.

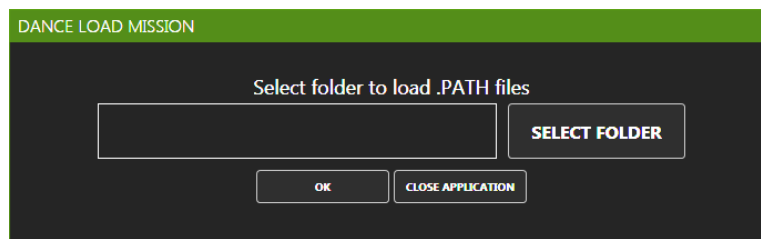


Image 16. Displaying the .PATH file location selection

Next it is necessary to make sure that all the vehicles can be connected and are displayed both in UgCS client and UgCS DDC Client (Correct group must be selected). Read mode about connection and Groups in sections Configuring Vehicle Specific Module (VSM) and Configuring UgCS DDC.



Image 17. Displaying three properly connected vehicles in UgCS with good telemetry status of all sensors

If some vehicles do not appear on the list in UgCS client, make sure to check if they can be seen in ‘Uplink connected’ list under *Menu/Vehicles/Uplink* connected or look for the vehicles by clicking on “+” button next to connected vehicles in Image 16.

After all the involved vehicles have been confirmed to connect and display correct status both in UgCS and UgCS DDC, operator may select the appropriate PATH file for each vehicle.

6.3. Setting the Dance Center

As the .PATH files only contain relative coordinates from a position defined in animation, it is necessary to define a Dance Center using Latitude, Longitude and Altitude parameters.

To assess Dance Center the vehicles should be placed in their positions and have acquired stable position (GPS and ALT). Now in **UgCS** client one can see all the vehicles on a 3D Globe and by placing a single waypoint in the center of the formation (or different position depending on animation) and noting its coordinates and altitude a dance center can be set.

To make sure that the center and fence has been set correctly, follow these steps:

1. Select correct PATH file for all vehicles
2. Select all vehicles and click Dance/Upload Path
3. Wait for the vehicles to initialize and Click Dance/Upload location
4. Click Dance/Upload Fence

Now the flight line from current vehicle location to the first point of dance will be displayed in **UgCS** client with a blue line as well as the Fence is uploaded to the vehicles. The offsets can also be viewed in **UgCS DDC** in the vehicles list under “OFFSET TO START”. Using these methods, operator should make sure that all vehicles are in correct positions and Dance Center is where it is intended. If some of the vehicles in **UgCS** client are displayed too low or too high over terrain, operator may use the “Set TO point altitude” command under “Actions” menu to correct this.

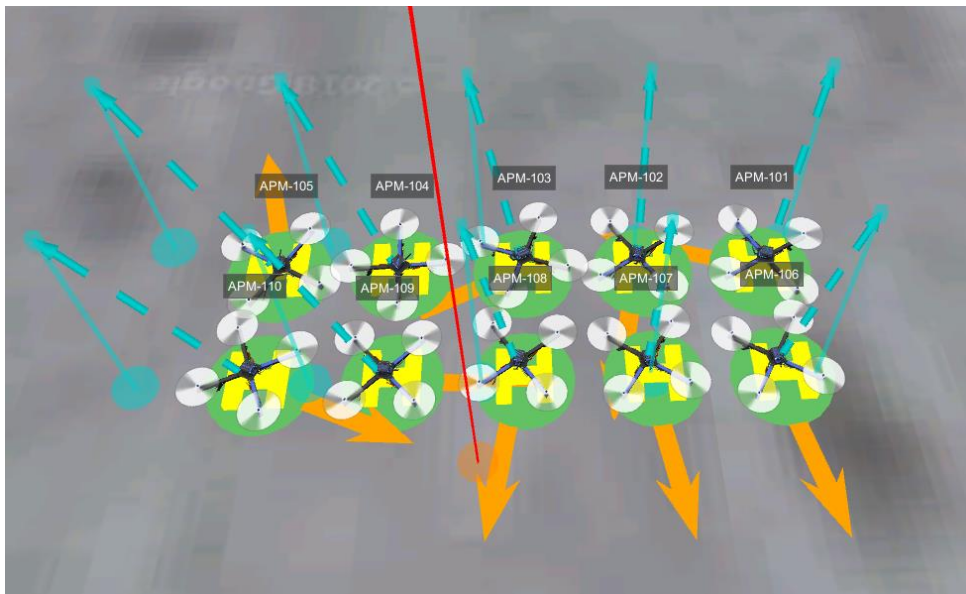


Image 18. Displaying connected vehicles in UgCS with flight lines to first point of the dance

6.4. Taking off and running the show

To easily assess current readiness for take-off, operator can pay attention to the “Dance Ready” status column. It displays actions that are still due if the column is red and will be ready only when all steps of dance process are completed.

Note: Operator should always keep in mind the steps taken and never rely solely on “Dance Ready” status. If in doubt, it is better to repeat all steps.

When the Paths and fence have been uploaded and Dance center has been set correctly, operator may continue to final dance step –uploading time and switching to dance mode.

ACTIVE	NAME	LAST CMD	START TIME	ONBOARD T	DANCE READY	PATH	MODE	STATUS	ALT GPS(AGL)	VERT SPD	GRND SPD	BATTERY
<input type="checkbox"/>	APM-101	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-101	Disarmed, Manual	Standby	1.0/1.0 (-2.6)	0 m/s	0.1 m/s	99% (12.59v)
<input type="checkbox"/>	APM-102	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-102	Disarmed, Manual	Standby	1.0/1.0 (-2.7)	0 m/s	0.1 m/s	99% (12.59v)
<input type="checkbox"/>	APM-103	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-103	Disarmed, Manual	Standby	1.0/1.0 (-2.9)	0 m/s	0.1 m/s	99% (12.59v)
<input type="checkbox"/>	APM-104	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-104	Disarmed, Manual	Standby	1.0/1.0 (-3.0)	0 m/s	0 m/s	99% (12.59v)
<input type="checkbox"/>	APM-105	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-105	Disarmed, Manual	Standby	1.0/1.0 (-3.1)	0 m/s	0.1 m/s	99% (12.59v)
<input type="checkbox"/>	APM-106	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-106	Disarmed, Manual	Standby	1.0/1.0 (-2.7)	0 m/s	0.1 m/s	99% (12.59v)
<input type="checkbox"/>	APM-107	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-107	Disarmed, Manual	Standby	1.0/1.0 (-2.8)	0 m/s	0.1 m/s	99% (12.59v)
<input type="checkbox"/>	APM-108	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-108	Disarmed, Manual	Standby	1.0/1.0 (-2.9)	0 m/s	0.1 m/s	99% (12.59v)
<input type="checkbox"/>	APM-109	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-109	Disarmed, Manual	Standby	1.0/1.0 (-3.0)	0 m/s	0.1 m/s	99% (12.59v)
<input type="checkbox"/>	APM-110	SetDanceLocat	00:00:00	12:02:47	NO (Size Center Time FPAlt Fence)	APM-110	Disarmed, Manual	Standby	1.0/1.0 (-3.2)	0 m/s	0.1 m/s	99% (12.59v)

Image 19. Displaying connected vehicles in UgCS with flight lines to first point of the dance

First, operator should check status columns in UgCS DDC. The following columns are important before setting time:

- **ONBOARD TIME** – displays the time in flight controller, this time should be equal on all vehicles up to two seconds error
- **DANCE READY** – before setting time, the status should read *NO (Time)* and nothing else
- **ALT GPS(AGL)** – should be the same for all vehicles up to 1m precision
- **PATH LENGTH** – depending on dance path plan, should be equal or similar for all vehicles
- **MAX RELATIVE ALT** – displays the maximum altitude from take-off position during dance
- **MAX DIST FROM CENTER** – maximum distance the vehicle will have from Dance center during dance
- **MAX HOR/VER SPEED** – maximum speed the vehicle should reach during dance
- **FPS** – must be equal for all vehicles

GPS FIX	VIBRATIONS	DATALINK	PATH LENGTH	OFFSET TO START	MAX RELATIVE ALT	MAX DIST FROM CENTER	MAX HOR SPEED	MAX VER SPEED	FPS
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:1.2 m, V:9.0 m	52.9 m	16.9 m	2.1 m/s	2.1 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:0.8 m, V:9.0 m	52.1 m	16.9 m	2 m/s	2 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:1.3 m, V:9.0 m	49.7 m	16.9 m	2 m/s	1.7 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:2.0 m, V:9.0 m	44.8 m	16.9 m	2.8 m/s	1.5 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:2.9 m, V:9.0 m	48.6 m	18.4 m	3.4 m/s	1.7 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:0.8 m, V:9.0 m	42.6 m	18.4 m	3.4 m/s	1 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:0.2 m, V:9.0 m	42.8 m	16.9 m	2.8 m/s	0.8 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:1.0 m, V:9.0 m	46 m	16.9 m	2.1 m/s	1.2 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:1.9 m, V:9.0 m	55.7 m	16.9 m	2.1 m/s	2.1 m/s	4
RTK fix (10)	0.01,0.01,0.01	99 %	867	H:2.8 m, V:9.0 m	52.1 m	16.9 m	2 m/s	2 m/s	4

If all status parameters are safe, and all vehicles have status: Dance ready (time) – meaning that everything is ready for dance and operator must set the time at which dance will start. The offset to start parameter is defined under Options/General options/Dance start interval



Now the commands must be executed in the following order:

1. ARM
2. SET TIME
3. UPLOAD TIME
4. DANCE

After the dance operator should execute the LAND command.

Using SITL simulator In order to run dancing swarm, the following requirements must be met:

Operation System: Ubuntu 16.04 or later

- **Preparing environment:**

If the operation system is 64 bit, use sitl64.zip

1. Unpack provided archive into the working directory:
 - a. **unzip sitl(64).zip -d SITL**
2. SITL – is the directory name in which SITL files will be located.
3. Enter directory and add execution permissions:
 - a. **cd SITL**
 - b. **chmod +x swarm.bash Tools/autotest/run_in_terminal_window.sh build/sitl/bin/arducopter**
4. Install required packages or make sure they are already installed:
 - a. **sudo apt-get install python-pip**
 - b. **sudo pip install future pymavlink MAVProxy**
 - c. **sudo apt-get install gnome-terminal**

Configuring swarm size:

Open **swarm.bash** with any text editor. Find the following lines :

```
# Range of system ids to use.
SYSID_FIRST=1
SYSID_LAST=20
```

These two lines define the range of drone IDs and the number of simulators generated. Here 20 drone instances will be initialized in a dance swarm.

- **Configuring drone positions:**

Open **locations.py** with any text editor, and find the following part:

```

Y_count = 7
X_count = 5
Distance_between_drones = 3 # meters

CNT_LAT = 56.973501 # Latitude of Drone formation center
CNT_LON = 24.071459 # Longitude of Drone formation center
Heading = 0
Altitude = 1

```

This file generates initial location of each drone in swarm, to avoid all vehicles to be placed in a single point. After running the **python location.py** script, the following output will be displayed:

```

S1=56.9734471011,24.0713106617,1,202
S2=56.9734471011,24.0713601078,1,78
S3=56.9734471011,24.0714095539,1,0
...

```

It is necessary to insert this output into **Tools/autotest/locations.txt**

Now the swarm can be started:

```

Execute
./swarm.bash

```

Wait until all instances got initialized and “Press any key...” in “swarm.bash” terminal to terminate all processes. First run is required for setting up sitl environments for each drone and is not valid for swarm operations yet.

As a result new directory will be created **ddc-cache**. It contains subdirs for each drone identity. (1, 2, 3, 4...) Each drone keeps its logs, parameters and mission paths in its corresponding folder.

Run swarm again

Script will start instances. Each instance is available on a tcp port following the formula:

port= 5760 + I*10 + 2

For example:

```

5772 for drone with id= 1
5782 for drone with id= 2
5792 for drone with id= 3
5802 for drone with id= 4

```

In order to connect DDC-Client to instances, VSM should be configures respectively:

You can use **vsm.py** script to optimize configuration generation. If SITL instances are running on a host with IP **192.168.1.39** and their number is 5, execute the following line:

```

python vsm.py --host 192.168.1.39 --num 5

```

Copy an output and place it into your **vsm-arducopter.config** file.

N 1

connection.tcp_out.1.port = 5772

connection.tcp_out.1.address = 192.168.1.39

N 2

connection.tcp_out.2.port = 5782

connection.tcp_out.2.address = 192.168.1.39

N 3

connection.tcp_out.3.port = 5792

connection.tcp_out.3.address = 192.168.1.39

N 4

connection.tcp_out.4.port = 5802

connection.tcp_out.4.address = 192.168.1.39

7. Configuring RTK corrections broadcast

7.1. Introduction

If the vehicles are equipped with RTK GPS modules, the same telemetry link that is used to send and receive information to vehicle might be used to send RTK corrections to vehicles. This kind of set-up does not require additional data link for the RTK GPS only and therefore can simplify the whole fleet. For the configuration described in this document to work it is required to use WiFi modules – B&B Airborne, ESP8266 based, XBee WiFi.

7.2. Requirements

A specific RTK Base unit or usually the same Rover module as on the vehicle can be used for this configuration. The unit must send the corrections in RTCM 3.0 protocol format.

Example base configurations:

- North SmarTK base unit
- North RTKite unit configured as base
- Ublox Neo-M8P configured as base
- Here+ RTK GPS ground unit

To broadcast the corrections to several vehicles at once, a custom installation of Mission Planner is required. The installation can be downloaded from here: <https://github.com/ugcs/MissionPlanner/releases>

To receive the RTK corrections on the vehicle, ESP-07 modules (or similar) based on ESP8266 chip using custom firmware available from here: <https://github.com/ugcs/mavesp8266/releases> are required. Please refer to document “[ESP WiFi Module configuration and connection for UDP Protocol](#)” for detailed information. To prevent interference with other 2.4 GHz devices, Airborne devices should be used: read more: https://github.com/ugcs/ddc/tree/master/WiFi/Airborne_5ghz. XBee WiFi modules can be used as well.

7.3. Configuration

To start using the custom Mission Planner for broadcast, first connect the base unit to PC and only then open up Mission Planner. Then go to section “INITIAL SETUP” on upper menu and chose “Optional hardware” from left menu then click on “RTK/GPS Inject”.

See 0.0 where the important settings are highlighted. First it is necessary to select correct COM port, if unsure which port is assigned to the Base unit, it can be viewed in Windows DeviceManager. Depending on the base unit, it needs to be configured to transmit the RTK corrections to the PC. For Ublox units, Mission Planner offers automated configuration, see “M8P autoconfig” checkbox on 0.

As for network settings, the PC running Mission Planner must be in the same network as all the vehicles. It is necessary to define the correct IP address mask and UDP port. This depends on the network

settings configured on the router and WiFi modules. The IP address (or broadcast address) should be structured depending on subnet. Typically, it should be 192.168.0.255 or 192.168.1.255 or other. The same applies to port, but the incoming port for the WiFi modules should be the same for all modules.

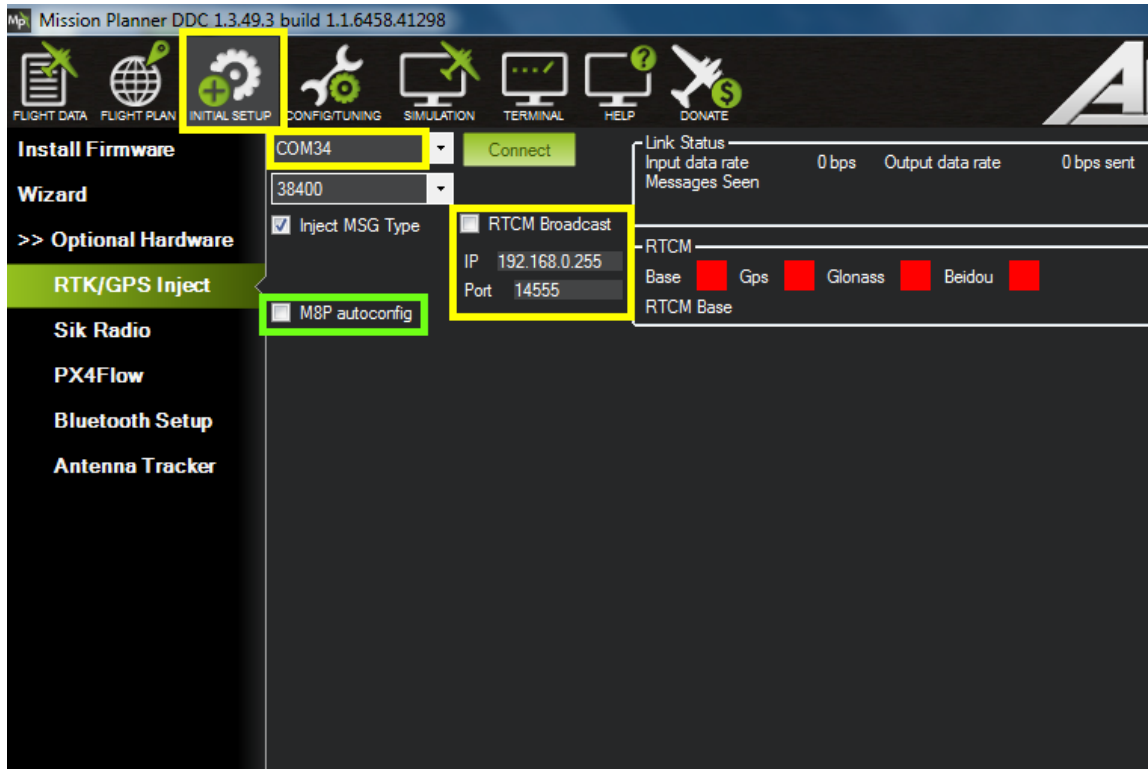


Image 20. The RTK/GPS Inject section prior to connecting a base unit

After successful configuration and connection to the base (click “Connect” next to COM port setting) the Mission Planner should start displaying which satellites are available to the base and display them like shown on Image 21. To start UDP broadcast enable RTCM Broadcast checkbox.



Image 21. Displaying available satellite status for the base.

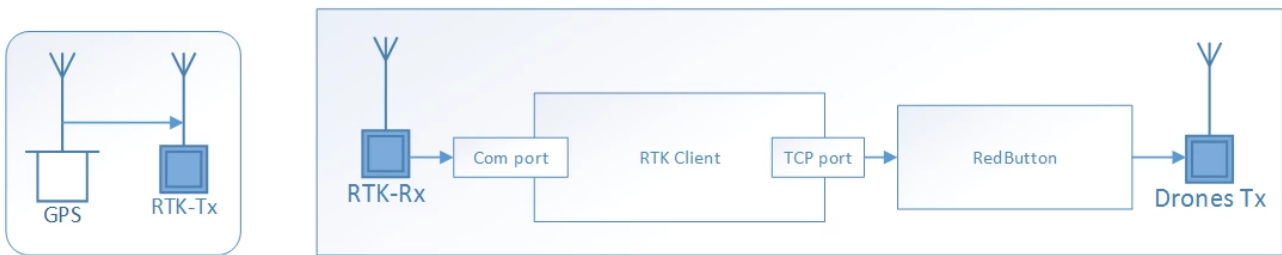
Now check if the vehicles change status to RTK Float or RTK Fix.

8. Configuring 2nd datalink

During the drone show we can't guarantee strong Wi-Fi connection with all the drones. Even during autonomous Dance, they still require RTK-GPS corrections and must respond to emergency commands (like hold the mission execution, do land, return home or even disarm). Secondary channel was designed to reduce risks of leaving fleet uncontrollable

8.1. Architecture

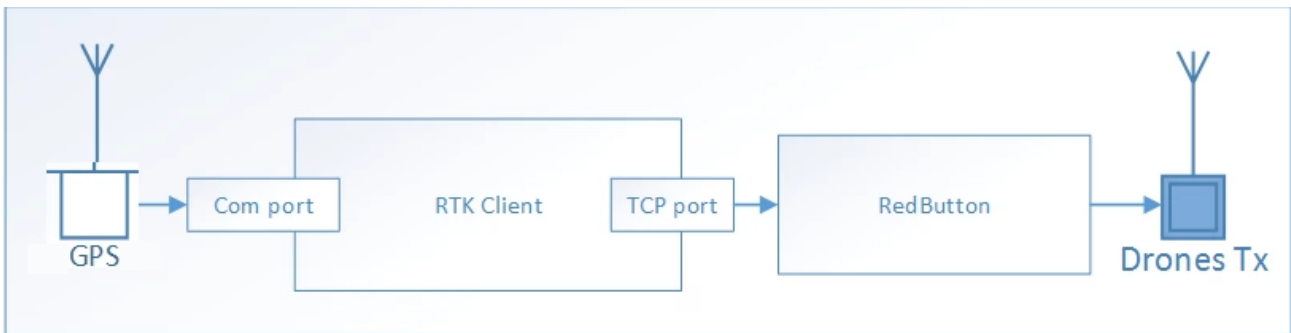
Reading RTCM corrections through radio link



This case nicely fits when RTK Station should be located away from operators.

Direct

connection:



Direct connection is good when high interference is expected.

Base GPS-RTK Station is running on its stationary location and sends its RTCM data to **RTK Client** application.

RTK Client parses incoming packages, wraps them into Mavlink and retransmits them to consuming sinks. **Red Button** is responsible for sending commands to the swarm. Reads data from one of **RTK Client's** sinks and broadcasts packages through to the fleet.

8.2. Hardware

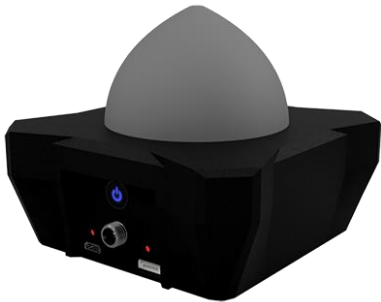
8.2.1. Radios

3DR Radio modules were chosen as a secondary channel link between drones and ground station. Depending on region, different band versions could be used to satisfy local requirements. (433, 868 or 915 MHz versions)



8.2.2. GPS

U-blox NEO-M8P-2 chip based module is used as a base station. [here](#)



Base station can be connected and configured through USB locally or using radio modems remotely. RTKClient application suites perfect for such purpose.

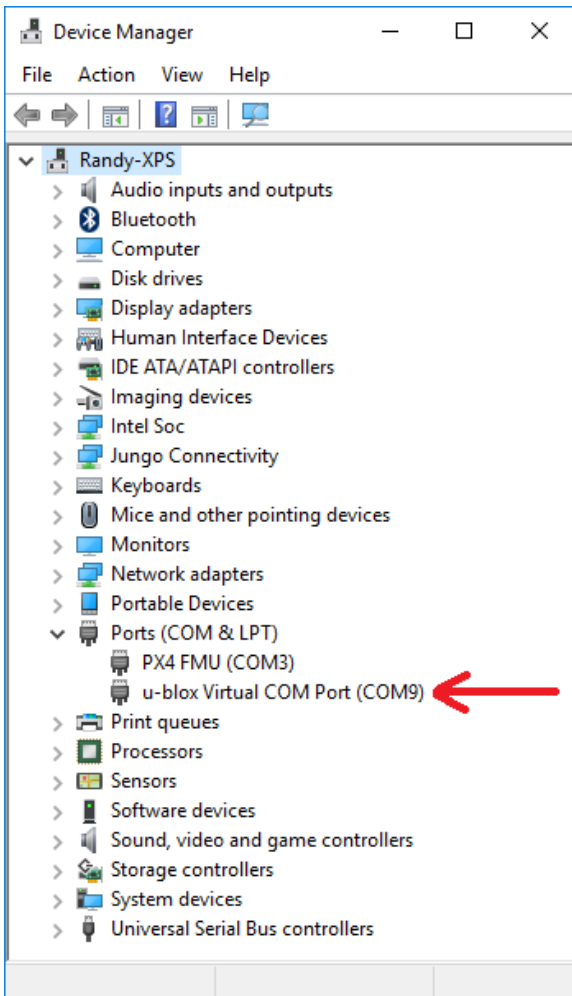
8.3. Software

8.3.1. Installing the Drivers

Windows 10 users (and perhaps others) will need to download and install the [u-blox GNSS Standard Driver for Windows, v1.2.0.8](#).

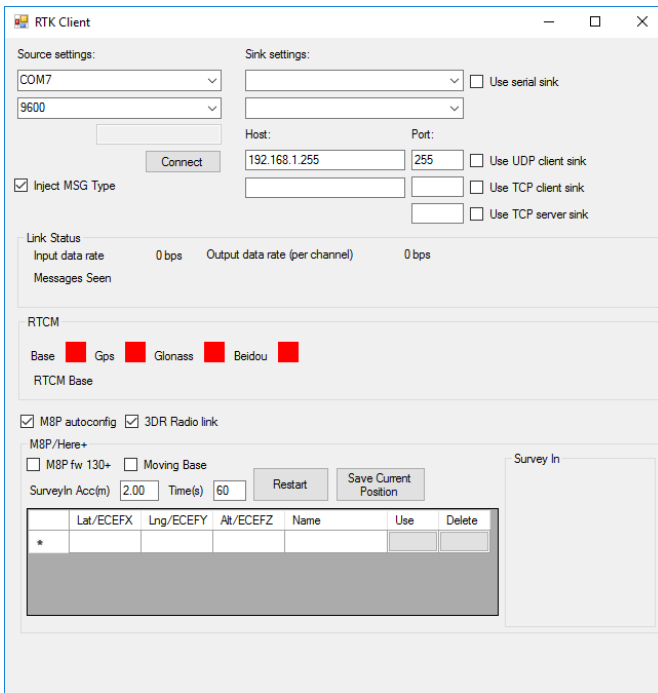
If this link is broken you may find a newer version is available under the “Driver” section of this [ublox page](#). When first installed, the device may show up in the “sensors” category in device manager. If this happens, please right click on it and “update driver” select “Browse my computer for driver software” then select “Let me pick from a list of devices” the next screen will show a list of possible drivers. Select. “USB Serial Device” Select next and finish.

If installed correctly, when plugged in, the GPS should appear in the Device Manager as “USB Serial Device”.



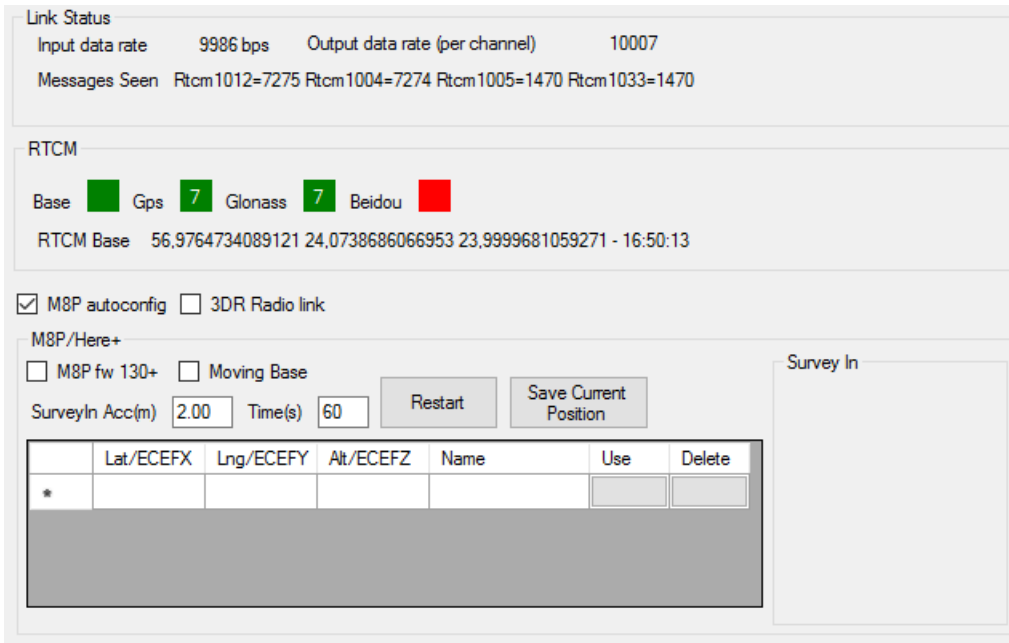
8.3.2. RTK Client

This tool was designed for configuration, to read corrections from RTK base station and retransmit data as a Mavlink messages to the multiple consumers.



Typical use case:

- Connect RTK Base station to your workstation.
- Identify COM port and baud rate used by GPS base. (9600 is default baud rate after gps restart) Refer to VCP Device driver manual (<https://www.u-blox.com/en/product/u-center-windows>)
- Choose port and speed in Source settings
- Choose consumers sinks. Fill text fields and click check box:
 - UDP client (Broadcast UDP corrections to drones Wifi network.)
 - TC client (Direct TCP connection to remote host)
 - TCP server port (Opens a server port and supplies RTCP corrections to external clients)
- Click “Connect”



If RTK base is connected to the workstation using USB cable, then the GPS module will be automatically configured when **M8P autoconfig** check box is selected)

If RTK base is connected with radio modems like 3DR Radios, then select 3DR Radio link option checked, so it will not try to reconfigure RTK-base to use higher speed. Only required messages will be passed, so bandwidth and baud rate may remain relatively low.

In the SurveyIn Acc section, enter the absolute geographic accuracy that you expect your RTK base station to achieve. In the Time column, enter the minimum survey time you expect. Click on Restart, the ground station will transfer the data you have entered to the RTK base module, the base module will start a new round of surveying.

Survey In section will display : InProgress shows whether the survey is still ongoing. A completed survey of the current base position is indicated by the status messages SurveyIn Valid: True, and InProgress: false.

Duration: the duration of the survey in seconds

Obs: number of observations used for the survey

Acc: current accuracy in meters

- When the survey is completed as indicated by the SurveyIn: True message, you may click the ‘Save current pos’ button and enter a name for the saved position.
- The RTK messages from base station are sent to vehicle by Mavlink message through consumers sinks.

8.3.3. RedButton

This app is a supporting tool for the main DDC Client. Its main responsibility is to take control of individual drone or the whole fleet. It also sends RTCM corrections through extra radio channel, not WiFi only.

Main screen shows all drones in a fleet with buttons against each one.

9. Troubleshooting

This section consists of various non-standard situations which might occur during mission execution.

The vehicle cannot be armed

- Make sure, that communications link is good, check if GPS/Compass and other sensor status is sufficient. If that does not help, try power cycling the vehicle. If everything fails, contact support@ugcs.com

The vehicle cannot be armed and in UgCS DDC error “The path cannot be found. Please try to change the location of the basic point(s) or values or parameters.” Is shown.

- Make sure that the current location of vehicle is within Geo Fence radius. The radius is calculated from Home Location.

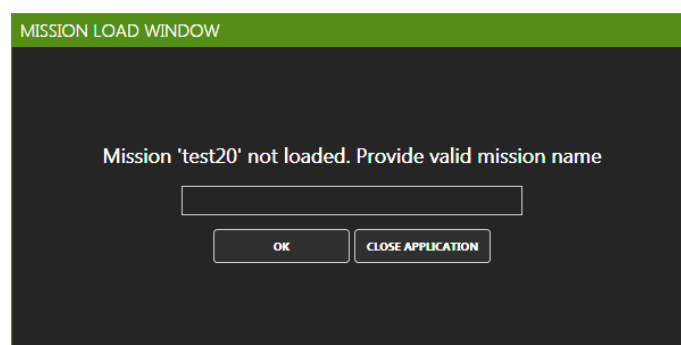
Route cannot be uploaded to a vehicle

- There is a high possibility that the communications link is weak and therefore **UgCS** is having trouble sending route data to the vehicle. Try placing the vehicle closer to ground end of communications module.

During flight vehicle suddenly returns home

- Check Geo-fence radius and altitude, there is a high possibility that vehicle has reached fence limits

When starting UgCS DDC Client the following message appears:



- This message is caused because the mission that is configured to work with **UgCS DDC** is either removed or renamed. Either enter the new mission name in the input field or change the mission name to previous one in **UgCS Client**. Note: **UgCS DDC** needs to be restarted if the name of mission was changed.