

User Guide

WiMA-Extension

Valentin Platzgummer

February 3, 2020

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
2	System Description	3
2.1	Key Features	3
2.2	Inventory	3
3	Menus and Instructions	3
3.1	Brief Summary	3
3.2	WiMA View	5
3.2.1	WiMA Tool Strip	5
3.2.2	WiMA Areas	5
3.2.3	Saving and Loading WiMA Missions	6
3.2.4	Clearing WiMA Missions	6
3.2.5	Uploading, Downloading and Clearing vehicle Missions	6
3.2.6	Inserting WiMA Areas	9
3.2.7	Generating a WiMA flight plan	9
3.2.8	Navigating the Map	9
3.2.9	How to display the generated flight plan	10
3.2.10	Circular Survey	10
3.2.11	How to synchronize the Data with Flight View	11
3.3	ArduPilot Simulator	12

1 Introduction

1.1 Purpose

The purpose of this document is to introduce the user to WiMA (**W**ireless **M**easurement **A**pplication) integrated in QGroundControl. QGroundControl is a ground control station software intended for the remote control of UAVs (Unmanned Aerial Vehicle). A Documentation of the QGroundControl program can be found under <https://docs.qgroundcontrol.com/en/>. The WiMA extends the functionality of QGroundControl to simplify mobile UAV based coverage measurements.

1.2 Scope

The current goal of the WiMA is to provide software able to generate flight plans form minimal user input. This guide contains information about how to use WiMA.

2 System Description

2.1 Key Features

The extension can generate flight plans based on minimal user input. This is done by defining different areas on a world map. Additionally WiMA offers to partition generated flight plans, which gives more flexibility. Partitioning makes it possibly to execute a subset of a flight plan, which makes it less tedious to cover large areas. As a security feature WiMA offers a smart return to launch routine. Most return to launch implementations choose the direct path for return, which can be blocked. WiMA can automatically calculate a save return path, based on user defined save areas. WiMA also offers fast adaption of flight altitude and speed.

2.2 Inventory

The source code can be cloned from the git repository <https://dummyurl.com>. The root directory of the project contains the folder "deploy" inside which a `.AppImage` of the program can be found. Under Linux QGroundControl can be launched from console or by double-clicking the `.AppImage`. It is also possible to build versions for Microsoft Windows, Android or Mac. For this purpose refer to the QGroundControl documentation under <https://docs.qgroundcontrol.com/en/>.

3 Menus and Instructions

3.1 Brief Summary

Figure 2 shows a detail view of the QGC main window. Relevant for this documentation are the flight view, the plan view and the WiMA view. Their function will now be briefly summarized.

The **plan view** is used to create flight plans and is part of standard QGroundControl. Flight plans can be stored as `.plan` files or directly be uploaded to a vehicle (either a real one or a simulated one). Flight plans consist of way-points. Each way-point stores, among others, a coordinate (latitude, longitude, altitude) and a command (take off, land, wait for x

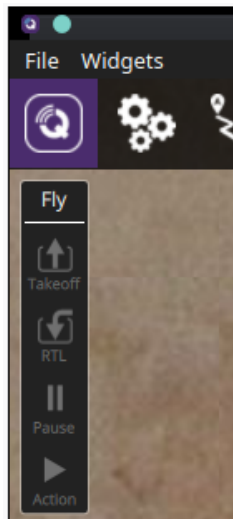


Figure 1: Flight view tool strip. Used to command the vehicle.

seconds, etc.). To define a sensible flight plan a sequence of way-points should begin with a take off command and end with a land command. Besides way-points more complex patterns can be defined, which is done by inserting a Survey, a Circular Survey, a Structure Scan or a Corridor Scan.

The **flight view** comes in necessary as soon as you want a vehicle to get in action. Once the vehicle established a connection (serial, TCP, UDP, etc.), QGroundControl starts to communicate automatically with it, in most cases. If not, check the settings menu of QGroundControl. A connected vehicle will appear as a red arrow on the map, showing the vehicles position and orientation. On the top indicator strip (fig. 3), telemetry data will be published. At the left edge a tool strip (fig. 1) will be activated, which can be used to command the vehicle.

The **WiMA view** is used to generate flight paths from minimal user input. The user can define a Measurement Area, a Service Area (for take off, land, supply tasks, etc.) and a Corridor, which connects the other two areas. Further the WiMA view and flight view will be described.

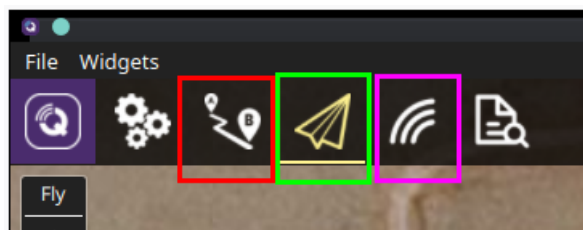


Figure 2: Detail view of the QGC window, which appears after start-up. Marked in red is the button for switching to the Plan View Window, green indicates the Flight View button (current window) and marked in magenta is the button for switching to the WiMA Main Window.

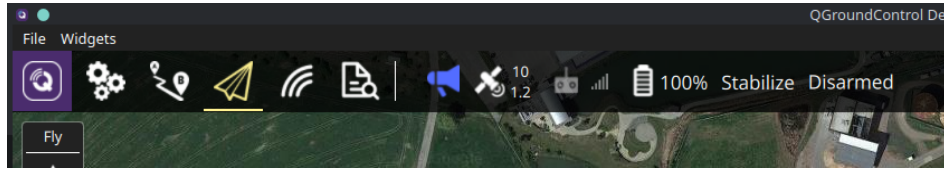


Figure 3: Flight view top indicator strip. Shows data about a connected vehicle. Among others satellite count, telemetry quality, battery level etc. are displayed.

3.2 WiMA View

By clicking the wave symbol (see fig. 2; magenta square) the WiMA view appears. After entering, at the left screen edge, the WiMA tool strip will appear (depicted in fig. 4).

3.2.1 WiMA Tool Strip

The WiMA tool strip is used for load/ save operations as well as editing tasks. The file button can be used to load, save and clear missions. Measurement and service area as well as corridor can be added by clicking the corresponding buttons. The calculate button can be used to regenerate the current mission. However with the current version regeneration is done automatically. The center and zoom buttons are used to navigate the map.

3.2.2 WiMA Areas

The three areas, namely the measurement area, the service area and the corridor, can be defined using the corresponding buttons (see 3.2.6). In the default mode the areas will be displayed as a half transparent polygon surrounded by a white lines. The polygon shape can be altered by using the drag and split handles shown in fig. 8, 9 and 10. All areas will be listed at the right side within the WiMA Item Editor (see fig. 6). The WiMA Item Editor can be used to modify area parameters. The Parameters **Altitude**, **Layers**, and **Layer Dist.** are reserved for future use and don't yet have any impact. The **Offset** parameter changes the distance between the area and its border polygon. The border polygon is a helper to provide sufficient clearance between the area and nearby obstacles, like trees or buildings. The border polygon can be deactivated by unchecking the **Border Polygon** check box. This can be use full for service area and corridor. Additionally the WiMA item editor shows some statistics about the polygon, like area and number of nodes.

The three areas have the following purpose. The WiMA measurement area defines, as the name indicates, the area of interest within which any measurements are performed. See 3.2.2 for

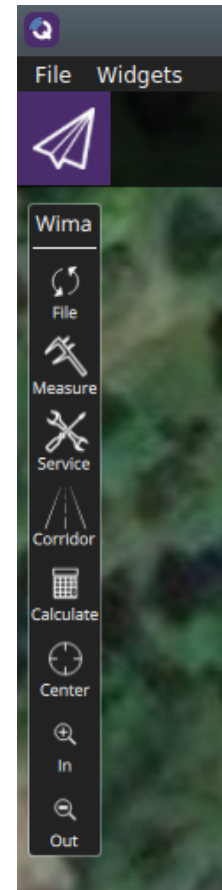


Figure 4: WiMA Toolstrip inside WiMA Main Window.

more information about WiMA areas. The WiMA service area is meant to be the place where takeoff and landing happens as well as battery exchange or repair work. See 3.2.2 for more information about WiMA areas. The WiMA corridor connects the measurement area and the service areas and defines a path which the vehicles use to travel between Service and Measurement Area.

A minimal configuration of areas consists of measurement and service area. In that case the two areas must be overlapping. The corridor can additionally be used to connect measurement and service area if overlapping of those two is not desired. A valid configuration including measurement, service area and corridor could look like in figure 11. The sum of all three areas is further referred to as the joined area.



Figure 5: WiMA area with border polygon. Shown is a measurement area (green).

3.2.3 Saving and Loading WiMA Missions

Pressing the **File** button (see fig. 12) of WiMA tool strip (see 3.2.1) opens a menu offering saving and loading operations. At one hand all WiMA areas and mission items can be saved using the `.wima` file extension, at the other hand the mission items only can be stored using the `.plan` file extension. The **Open** button can be used to load previously stored files.

3.2.4 Clearing WiMA Missions

Press the **File** button (see fig. 12) of WiMA tool strip (see 3.2.1) to open the dialog window. Pressing the **New** button deletes all contents within the WiMA Main Window.

3.2.5 Uploading, Downloading and Clearing vehicle Missions

Press **File** button (see fig. 12) of WiMA tool strip (see 3.2.1) to open the dialog window. With the **Upload**, **Download** and **Clear Vehicle Mission** button, the mission items (if present)

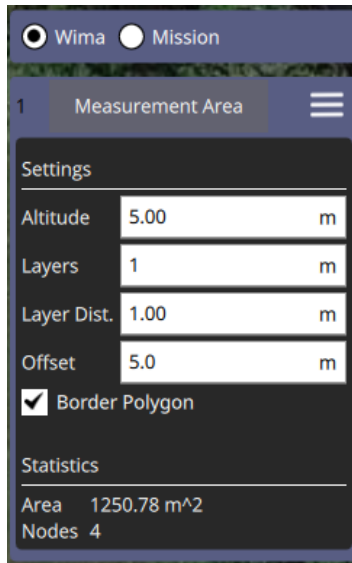


Figure 6: WiMA Item Editor at the right edge of the WiMA Main Window.

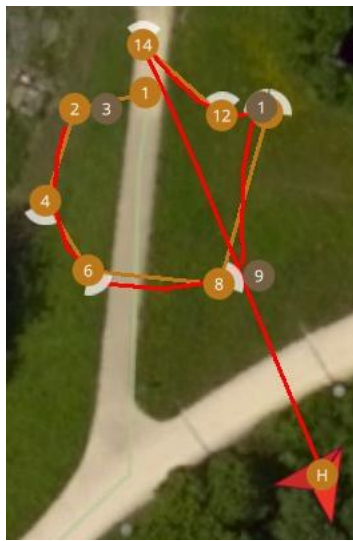


Figure 7: Flight path of a real vehicle (red line).

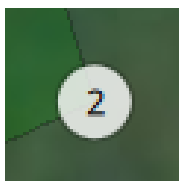


Figure 8: Adjust the vertex by dragging it's handle.



Figure 9: Add vertices by hitting the plus sign.

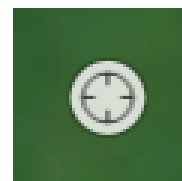


Figure 10: Move any area by dragging its drag handle.



Figure 11: A valid configuration of Measurement, Service Area and Corridor.

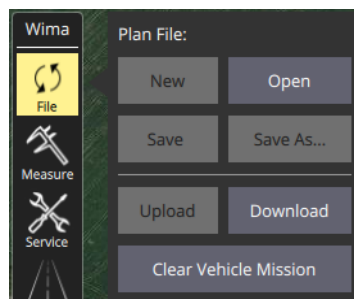


Figure 12: WiMA tool strip save/load dialog.

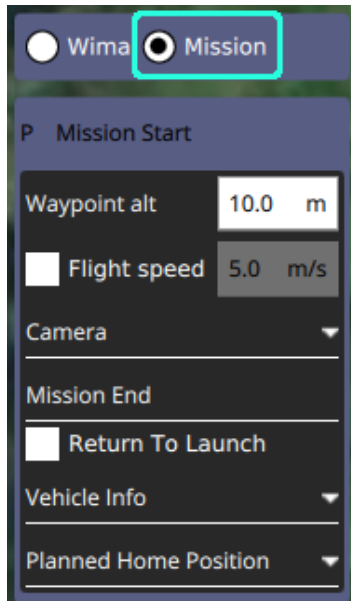


Figure 13: Checking the Mission radio button, on the right hand side, displays the mission items.

can be uploaded, downloaded or deleted from the vehicle.

3.2.6 Inserting WiMA Areas

The **Measure**, **Service** and **Corridor** buttons of the WiMA tool strip (see 3.2.1) are used to insert a measurement, service and corridor area, respectively. Only one of each can be inserted. To generate a flight plan, at least a measurement and a service area must be defined. In that case both must be overlapping.

3.2.7 Generating a WiMA flight plan

The current version of WiMA automatically updates the flight plan, after terminated user interaction. However using the **Calculate** button (see fig. 12) of the WiMA tool strip (see 3.2.1), manual regeneration can be triggered. The flight plan will be generated such that all way-points are within or at least at the edge of the joined¹ area. However it should be taken in account, that the vehicle could still leave the save area, even under fully functional operation. Depending on the software implementation of the vehicles flight stack (firmware), the flight controller could perform a path change. Additionally drifts caused by wind gusts could occur. An example can be seen in fig. 7. The vehicles flight controller has no information about any WiMA Areas, they exist exclusively within the QGroundControl application.

3.2.8 Navigating the Map

The three buttons **Center**, **In** and **Out** (see fig. 12) of the WiMA tool strip (see 3.2.1) can be used to center the view to a object of interest, and for zooming. The Map can be navigated by dragging the mouse pointer or touch screen.

¹The joined area is the sum of measurement, service area and corridor (see 3.2.2).



Figure 14: The flight path generated from the areas depicted in fig. 11.

3.2.9 How to display the generated flight plan

The flight plan can be displayed by checking the **Mission** radio button above the WiMA item editor (see. fig. 13), as depicted in fig 13. An example of a generated flight path can be seen in figure 14. Within the green measurement area the program has generated a circular survey (see 3.2.10). The survey has an entry and an exit point. The program has routed paths from the take off point to the surveys entry point and from the surveys exit point to the landing point. The paths have been chosen to be within the joined area and to be as short as possible.

3.2.10 Circular Survey

The Circular Survey (green area in fig. 14, containing circular tracks) is a flight pattern provided by WiMA. It is similar to the Survey pattern of the standard QGroundControl application. The Circular Survey was adapted for mobile coverage measurements. It consists of circle segments referred to as transects. Ideally, the transects are connected with each other such that the path between entry and exit point is as short as possible. Currently this goal is approximated by a heuristics. The solution might not always be the global optimum. The **Reference** point (Circle with R symbol and Reference tag, see fig. 14) defines the center of the circle segments. It is supposed to be placed on the base stations position.

Circular survey parameters can be edited with the circular survey editor (see 15). The **Altitude** input box adjusts the altitude of the survey's way-points. The distance between adjacent circle segments can be modified by changing the **Delta R** value. The circle segments are approximated by polygonal structures, consisting of way-points. Two arbitrary adjacent way-points of a circle segment have a maximum angle between them. This angle can be ad-

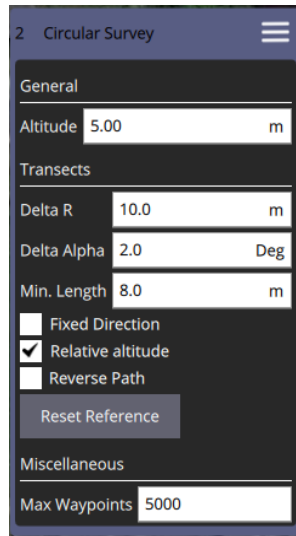


Figure 15: The circular survey item editor. Can be found at the right edge of the screen.

justed by editing the **Delta Alpha** input box. The **Delta Alpha** value can assume everything between 0.3° and 90° . The higher the value the rougher the circle appears. The survey will contain the less way-points the higher the **Delta R** and the **Delta Alpha** values are. The number of way-points impacts the recreation and upload time. In general the following holds. The less way-points the better. Use only as many way-point as you really need.

Short transects are often undesired, to remove them, the minimal transect length can be specified by the **Min. Length** value.

If it is desired that the UAV flies with the same direction for all transects the **Fixed Direction** check box can be activated. The **Relative Altitude** check box enables relative height. With the **Reverse Path** check box path reversal can be obtained. The **Reset Reference** button the reference handle can be set to the survey area center. This can be helpful if the reference handle is hidden.

To reduce calculation effort, the maximum number of way-points can be set using the **Max Waypoints** input box. Transects are hidden and calculations aborted, if the number of way-points exceeds **Max Waypoints**.

3.2.11 How to synchronize the Data with Flight View

3.3 ArduPilot Simulator

For tasks like debugging, program verification or flight plan testing a simulated vehicle can be very useful. It can save time, money and prevent you from excessive sunburns, if you forgot that you are actually outside, starring on your screen, exposed to the hot summer sun.

For this task the ArduPilot simulator can be used. It simulates a vehicle running the ArduPilot flight stack (firmware) on your local machine. Data is being published by the simulator via UDP and should ideally connect to QGroundControl without any further tweaks.

The simulator is part of the ArduPilot project, which can be downloaded from Github: <https://github.com/ArduPilot/ardupilot>. It is recommended to fork the repository. After cloning the repository, the submodules must be initialized and updated. Execute the following code to do this.

```
git clone https://github.com/ArduPilot/ardupilot
cd ardupilot
git submodule init
git submodule update
```

The simulator is launched by a Python script, thus Python must be installed on your machine. The simulator can be launched from your ArduPilot root directory with the following command.

```
./Tools/autotest/sim_vehicle.py -v ArduCopter
```

Thereby the `-v` option specifies the vehicle type. The `-l` option can be used to define a custom start location. See the `sim_vehicle.py --help` option for further information. After launching, the vehicle should appear inside QGC.