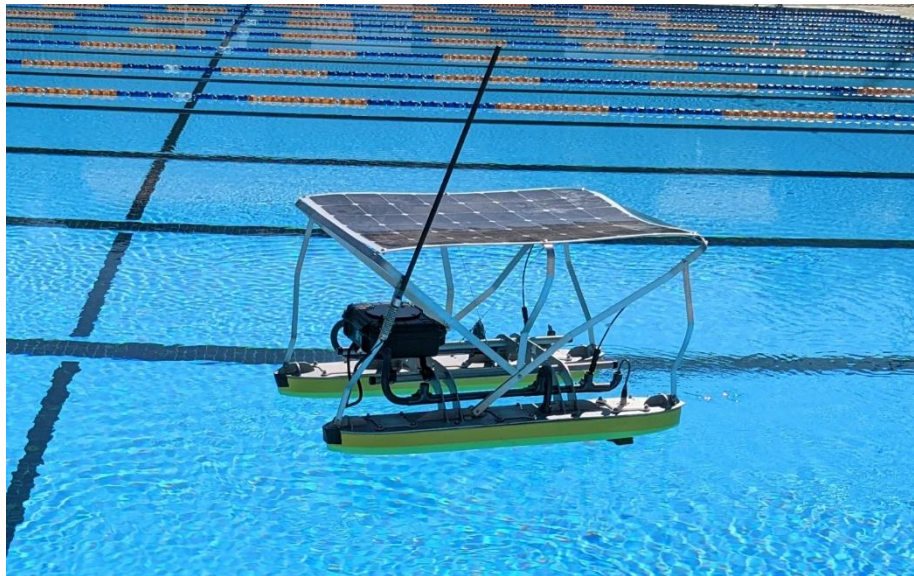


Solar Powered Autonomous Boat

User Manual



Author(s)	Lucie Cunningham, Tom Chan, Dylan Carpenter
Date Created	May 24, 2021
Version	1.0

CITS55512 Software Engineering Design Project
School of Computer Science and Software Engineering
The University of Western Australia

Contents

1. Project Overview.....	2
1.1. Background	2
1.2. Motivations	2
1.3. Vessel Hardware Summary	2
2. Installation and Setup	3
2.1. Requirements.....	3
2.2. Application Overview	3
2.3. SPAB-Data-Struct	3
2.4. SPAB-GUI.....	3
2.5. SPAB-Server.....	4
2.6. SPAB-Client.....	5
2.7. SPAB-Passthrough (Optional)	6
3. Controlling the Boat.....	7
3.1. Connecting to the Vessel	7
3.1.1 <i>Connecting via Radio</i>	7
3.1.2 <i>Connecting via the Internet (TCP)</i>	7
3.2. Manual Control	8
3.3. Automated Control	9
4. Using Website Interface.....	11

1. Project Overview

1.1. Background

The Solar-Powered Autonomous Boat is an ongoing project at UWA. The goal of the current iteration of the project is for the boat to perform a round trip of Rottnest Island while collecting sensor data.

1.2. Motivations

This project was inspired by other projects such as the SeaCharger project, which attempted to cross the Pacific Ocean with an autonomous vessel. A self-sustaining boat with the ability to autonomously navigate bodies of water is the next step in nautical exploration, and as such there are many international robotics competitions which strive to advance the field. UWA robotics is intending to enter an autonomous boat competition in 2022, building upon work on SPAB.

1.3. Vessel Hardware Summary

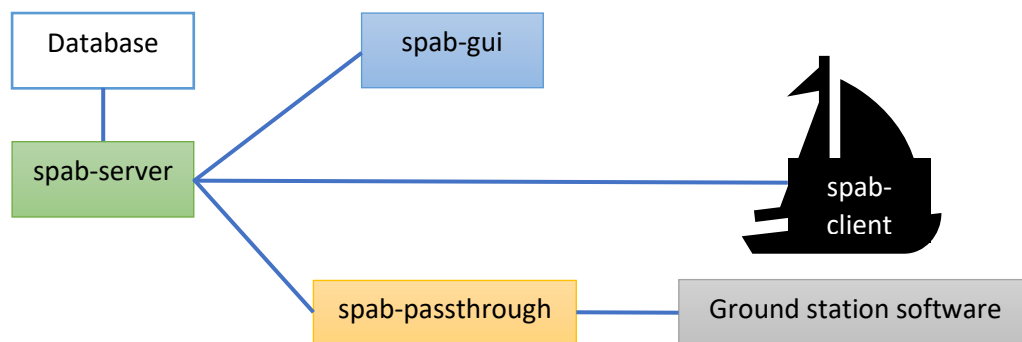
The hardware of the boat was built prior to the beginning of our team starting on the project. As seen on the cover page, the boat has dual fibreglass hulls with rear mounted propellers and a detachable solar canopy. It has electronic equipment such as a 3G modem, a raspberry pi, a flight controller, motor controllers and a short-range radio receiver.

2. Installation and Setup

2.1. Requirements

- Vessel
 - Onboard flight controller compatible with the MAVLink 2 communication protocol
 - Onboard computer/controller that supports Nodejs v14 and above
 - An internet connection via cellular signal or other means
 - (Optional) A long-range radio connection with the flight controller. If no radio connection is present, control can be established by using SPAB-Passthrough
- Server
 - Publicly addressable Linux server for running SPAB-Server
 - Nodejs v14 and above
 - PostgreSQL v9 and above

2.2. Application Overview



As shown above, the system contains four main components, which are:

- SPAB-Server: Server-side software
- SPAB-GUI: Web-based user interface
- SPAB-Client: Boat's software
- SPAB-Passthrough: Mavlink passthrough software for user's computer

Currently, the latest version of the server software is installed on the Bluehost server, which can be located at the path `/home4/therevpr/spab_2021s1`. On the other hand, the client software is installed on the desktop of boat's Raspberry Pi controller.

2.3. SPAB-Data-Struct

SPAB-Data-Struct contains the data structure used for data transmission in the system. It uses Google's Protocol Buffer format which is designed for reducing the size of structured data. SPAB-Data-Struct is a dependency for the following SPAB-Server, SPAB-GUI and SPAB-Client sub-projects. The repository comes with precompiled code. For more details, see the README.md file in spab-data-struct directory.

2.4. SPAB-GUI

SPAB-GUI contains the web-based user interface of the system. It is built with Angular framework and comes with precompiled code. For more details, see the README.md file in spab-gui directory.

2.5. SPAB-Server

SPAB-Server is the server-side software of the application. It requires Nodejs v14 and PostgreSQL v9 to operate correctly. To run the app, you need to install all necessary dependencies by running the following command line:

```
npm install
```

Next, you will need to maintain the directory structure as shown to ensure SPAB-GUI and SPAB-Data-Struct are accessible by SPAB-Server:

```
├── spab-server
│   ├── dist
│   │   └── ...
│   ├── static
│   │   └── ...
│   ├── private
│   │   └── ...
│   ├── config.js
│   └── ...
├── spab-gui
│   └── dist
│       └── ...
└── spab-data-struct
    ├── SpabDataStruct.d.ts
    ├── SpabDataStruct.js
    └── ...
```

After that, you might want to update the config.js file to configure the software, including database credentials and server domain name.

The web server should also be configured correctly to redirect spab requests to the software. A sample configuration file for Apache webserver can be found at:

```
./server_config/.htaccess
```

Then, run this command to start the software:

```
node ./dist/index.js
```

Since it is impossible to setup a start-up task on the Bluehost server, a cron job is created in server's cPanel to start the software automatically when it is stopped. Please keep in mind that each cron job is running in an isolated environment so there is no way to stop a cron job through SSH and cPanel. However, you can use these four APIs to control the software:

List all running process on the server:

```
https://therevproject.com/spab_2021s1/php/spab_run.php?type=list
```

Kill all Nodejs processes:

```
https://therevproject.com/spab_2021s1/php/spab_run.php?type=killnode
```

Restart the server software:

```
https://therevproject.com/spab_2021s1/php/spab_run.php?type=restart
```

Stop the server software:

```
https://therevproject.com/spab_2021s1/php/spab_run.php?type=stop
```

For more details, see the README.md file in spab-server directory.

2.6. SPAB-Client

SPAB-Client is the software running on boat's Raspberry Pi controller for handling Mavlink connections, camera streams and sensor data. It requires Nodejs v14 to operate correctly. To run the software, you need to install all necessary dependencies by running the following command line:

```
npm install
```

Next, you will need to maintain the directory structure as shown to ensure SPAB-Data-Struct is accessible by SPAB-Client:

```
├─ spab-server
│   ├── dist
│   │   └─ ...
│   ├── config.js
│   └─ ...
└─ spab-data-struct
    ├── SpabDataStruct.d.ts
    ├── SpabDataStruct.js
    └─ ...
```

After that, you might want to update the config.js file to configure the software, including boat's credentials and server domain name.

Then, run this command to start the software:

```
node ./dist/index.js
```

If you want to setup a start-up script to start the software automatically when the boat is turned on, please follow these steps:

1. Run the command below to install PM2

```
npm install pm2 -g
```

2. Generate a start-up script

```
pm2 startup
```

3. Follow the prompt to install the start-up script

4. Set the working directory to SPAB-Client's location

```
cd {spab-client's path}
```

5. Start SPAB-Client with PM2

```
pm2 start ./dist/index.js
```

6. Check the status of SPAB-Client to verify if the software is running correctly

```
pm2 ls
```

7. Save the current running tasks to the start-up script

```
pm2 save
```

SPAB-Server has an administrator script for managing the users and boats. To use the admin script, run the following command:

```
node ./dist/Admin.js
```

The script supports these options:

```
3871: Rebuild database
1: List users
2: Create a new user
3: Remove a user
4: List clients
5: Create a new client
6: Remove a client
7: Remove client's logs
```

For more details, see the README.md file in spab-client directory.

2.7. SPAB-Passthrough (Optional)

SPAB-Passthrough is the software for handling Mavlink data passthrough on user's computer. It requires Nodejs v14 to operate correctly. To run the software, you need to install all necessary dependencies by running the following command line:

```
npm install
```

After that, you might want to update the config.js file to configure the software, including the local TCP port and server domain name.

Then, run this command to start the software:

```
node ./dist/index.js
```

For more details, see the README.md file in spab-passthrough directory.

3. Controlling the Boat

The boat can be controlled using the software QGroundControl which can be downloaded to function on a variety of systems here: <http://qgroundcontrol.com/downloads/>

This document shows the use of the most current Windows version of QGroundControl (v4.0.10) available at the time of writing. Other ground station software may also be capable of controlling the vessel.

3.1. Connecting to the Vessel

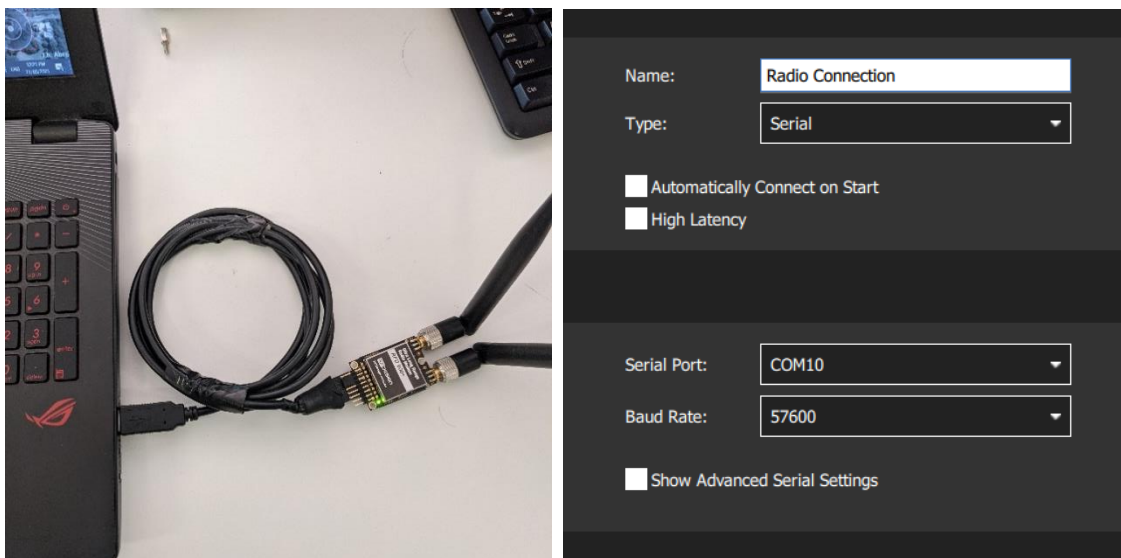
There are two different methods for connecting to SPAB for sending commands and receiving telemetry: over the internet with TCP or with a radio transceiver. Note that SPAB-Passthrough is required for connection over the internet and that not every ground control system (GCS) will support TCP connections.

3.1.1 Connecting via Radio

It is possible to connect directly to the SPAB flight controller over radio and bypass both SPAB-Client and SPAB-Server. This greatly decreases latency and packet-loss at the cost of restricted range and device compatibility. Steps for establishing a radio link may vary with the type of radio in use.

As an example, we will consider the *RFDDesign RFD900+*. To establish a link, one transceiver should be connected directly to the vessel's flight controller, and the other connected via USB to the GCS computer.

The communication link can then be setup in the GCS. The communication type should be "Serial" with the correct COM port and baud rate selected. For the *RFD900+* the baud rate is 57600.

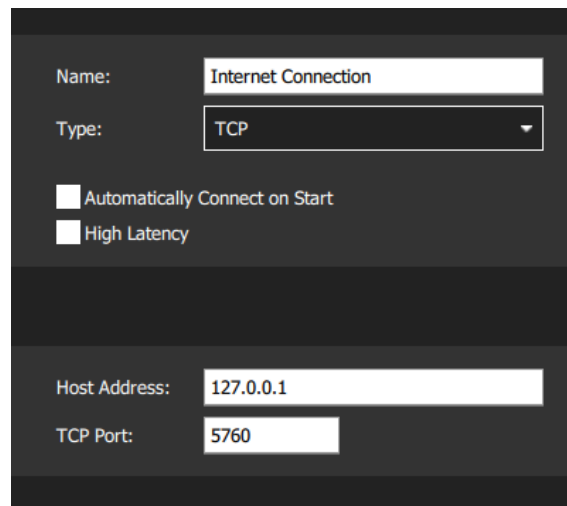


3.1.2 Connecting via the Internet (TCP)

To connect to the boat over the internet, MAVLink traffic must be routed through the SPAB-Server; an intermediary device to traverse any Network Address Translation (NAT) used in the boat's internet connection. SPAB-Passthrough must also be installed on the server device (see section 2.7).

To establish the connection between the GCS and the vessel, first turn the vessel on and wait for the internet connection to be established. This may take up to two minutes. If using the SPAB web interface, the vessel will show an "online" status once ready.

Then in the GCS, create a connection of type “TCP”, and enter the IP address and port number at which SPAB-Passthrough is running. A connection will then be established with the vessel. If during its journey the client (boat) loses connection with SPAB-Server, the GCS connection will be maintained until the client reconnects.

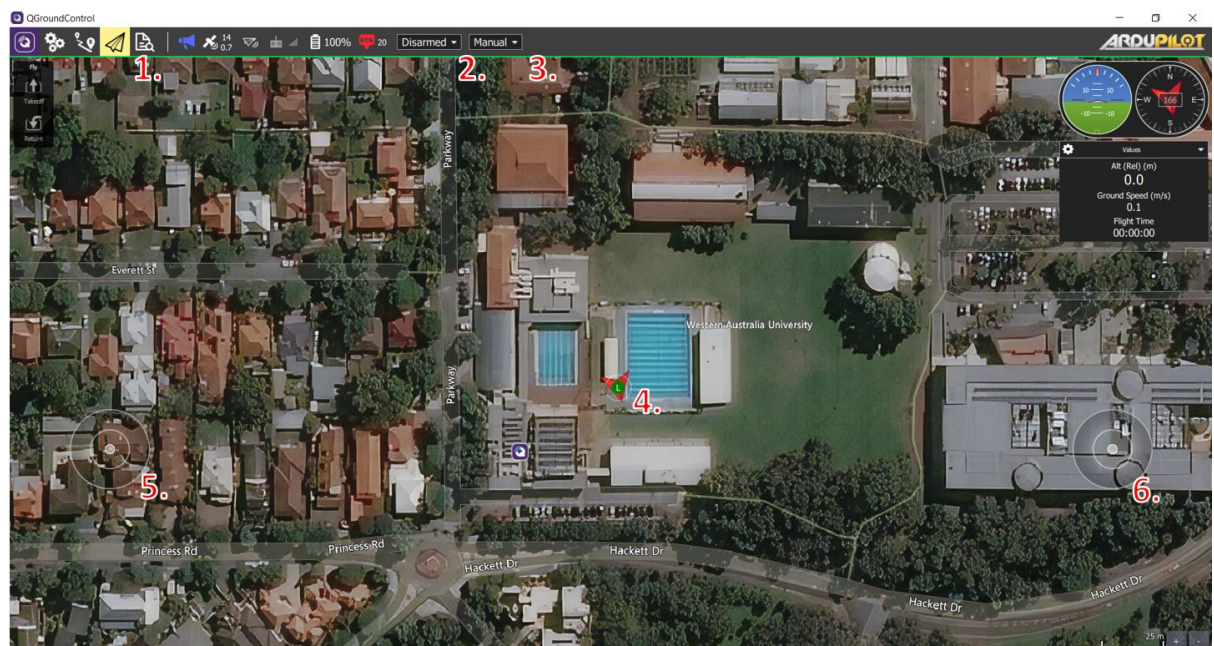


The screenshot shows the 'Internet Connection' configuration window in QGroundControl. The 'Name' field is set to 'Internet Connection'. The 'Type' is set to 'TCP'. There are two unchecked checkboxes: 'Automatically Connect on Start' and 'High Latency'. The 'Host Address' is set to '127.0.0.1' and the 'TCP Port' is set to '5760'.

3.2. Manual Control

Steps to control the boat manually:

- i. Ensure that QGroundControl is in “Flight Mode” and the boat is “Manual” mode.
- ii. Arm the vessel using the drop-down menu at the top of the interface.
- iii. Use the two onscreen joysticks to steer the vessel. The left joystick controls forwards and backwards motion using the stick’s Y-axis. The right joystick rotates the boat towards port and starboard using the stick X-axis.
- iv. Disarm the boat once driving is completed.



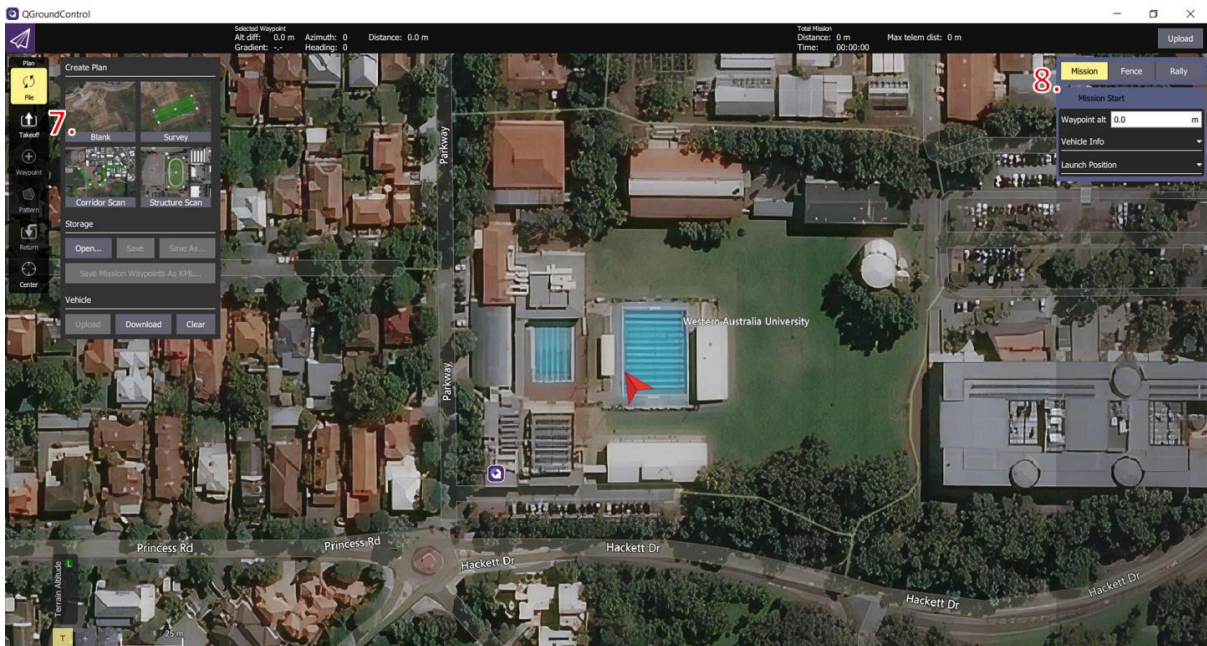
Legend:

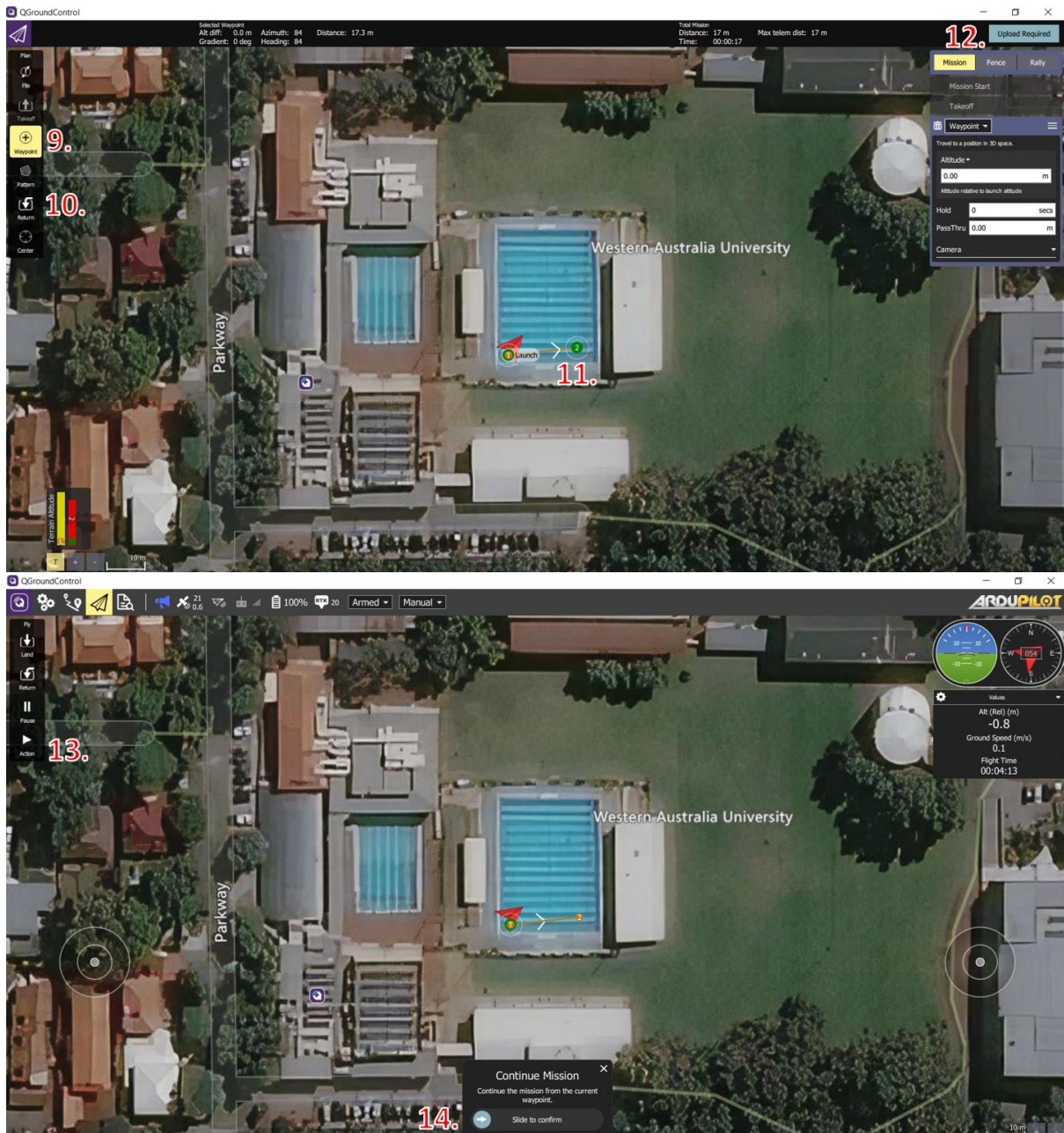
1. Current mode highlighted in yellow (currently in "Flight Mode")
2. Dropdown which dictates whether boat is "Armed" or "Disarmed"
3. Dropdown which dictates whether boat is in "Manual" or "Auto"
4. Shows where and in what orientation the boat is
5. Virtual Joystick 1, controls forwards/backwards movement (joystick y-axis)
6. Virtual Joystick 2, turns the vessel (joystick x-axis)

3.3. Automated Control

Steps to plan boat missions with waypoints in QGroundControl:

- i. Switch QGroundControl into the "Planning Mode" by pressing the button next to the "Flight Mode" button.
- ii. Select the "take-off" and move the circle with a T on it to the desired take-off point.
- iii. Add waypoints to the mission by selecting the waypoint button and clicking on the desired location. Waypoints can be dragged around the map to change their location.
- iv. Press the return button if you wish the boat to return to the take-off point once all waypoints have been visited.
- v. Press the "Upload" button to send the waypoint information to the vessel.
- vi. Return QGroundControl to "Flight Mode" and arm the vehicle.
- vii. Change the vessel into "Auto" mode and press the "Action" button
- viii. Disarm the boat when finished or if the mission needs to be aborted.





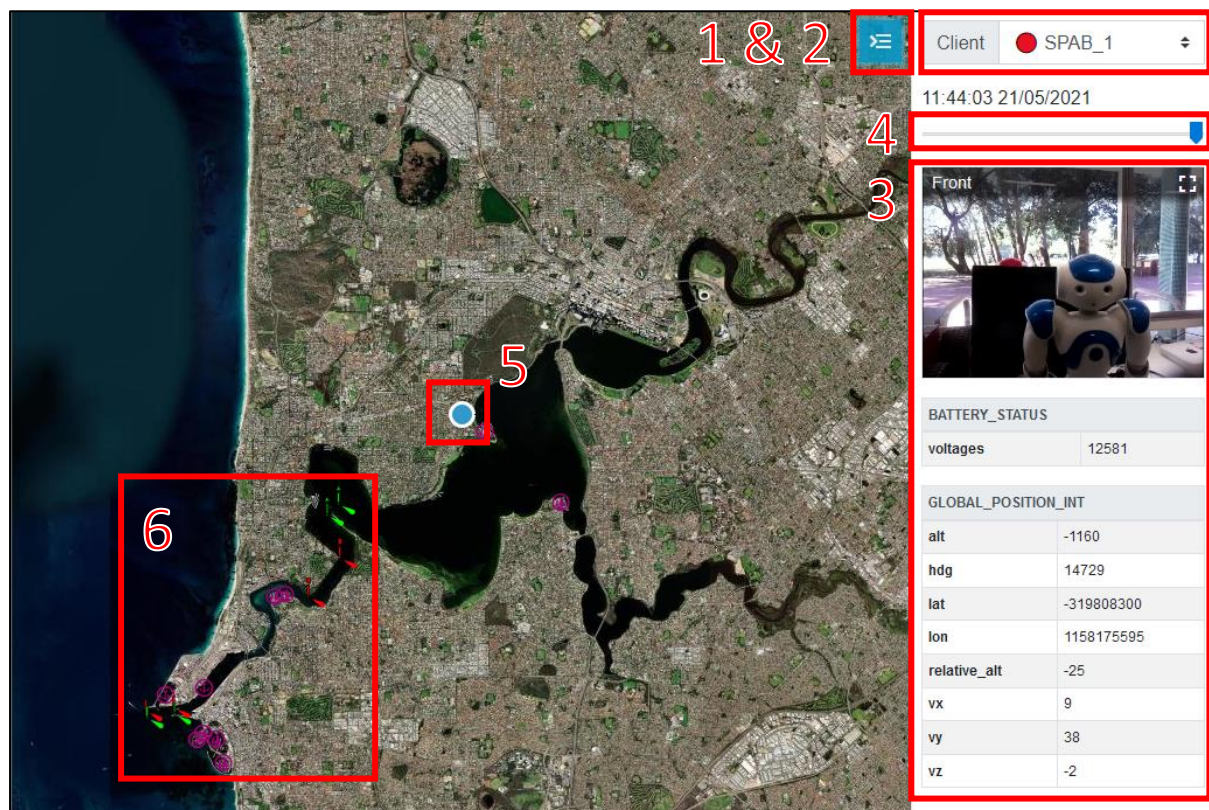
Legend:

- | | |
|---------------------------------|--|
| 7. The take-off button | 8. The list showing the planned mission |
| 9. The waypoint button | 10. The return to starting position button |
| 11. The visualised mission plan | 12. The upload button |
| 13. The action button | 14. The confirmation pop-up |

4. Using Website Interface

Steps to use the web interface:

- i. Go to the web address: https://therevproject.com/spab_2021s1/ and enter the username and password at the login screen. Section 2.6 details how to create a new user login.
- ii. Click on the menu icon and select "SPAB_1" as the client from the dropdown menu
- iii. If the boat is currently turned on, real-time camera images, battery status and sensor data can be seen in the side panel.
- iv. Moving the slider above the camera image from side to side will scroll through previously collected sensor data and camera images.



Legend:

1. The menu icon
2. Client drop-down with SPAB_1 selected
3. Camera images and sensor data
4. Slider to pan through historical data
5. Current location of SPAB
6. Nautical map data from *OpenSeaMaps*