**Hong Kong Institute of Vocational Education (LWL) Department of Information and Communications Technology**

**Final Year Project (ITP4913M)**

**Summary Report**

**(2019/2020)**

**Automated Data Collecting System for Environment Using UAVs and Smartphones**

|  |  |
| --- | --- |
| Supervisor: | Edward Ma |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | **Student No.** | **Contribution to the project (%)**  **(Total 100%)** | **Signature** |
| Wong Ming Yuen | 180275359 | 25% |  |
| Siu Chi Wang | 180042945 | 25% |  |
| Kwok Tsz Lung | 180047074 | 25% |  |
| Chau Yat Sum | 180478187 | 25% |  |

We declare that this is a group project and that no part of this submission has been copied from any other student’s work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

# Table of Contents

# Abstract

# Acknowledgement

# List of Tables and Figures

# Introduction

# The requirements

## Scope of the System

# UML Modeling

# Documentation for Detailed Design

# Implementation

# Test Plan and Test Result (Website)

# Critical Evaluation

# References

# Appendices

# Program Listing

Abstract

This project is to develop a Data Collection System. Our system will help the meteorologists to measure the weather. We analyzed their requirements to develop our system to provide easy deploy instruments and performing weather analysis.

This project is to develop a client-side system which consists of three parts including an unmanned aerial vehicle (UAV), a computer application and a mobile application.

As we all know, Smart devices are becoming much popular like Apple iPhone, iPad, Google Android devices...etc. So, we use a mobile device to replace the common UAV Flight Controller. The mobile application aims to easily control the UAV by directly connecting the UAV without another controller.

Regarding the computer application, we develop a system based on Window system. This system was designed to display the detail of the UAV and climatic data. A meteorological research team can use this system to monitor the status of the UAV and study the environment data instantly. Also, these collected data can be outputted by JSON file and the graphics. JSON is the best tool for the sharing data because the data is stored in the arrays. Therefore, the team can be handled the data easily for further study.

In the world, up to now, there is not any standard Personal Drone System for weather analyzing. The meteorological research team usually use simple environment data collected from the tools which are small and single or the environment data that providing from some large organization for study, but these ways is difficult to research a specific area for small organization. This system finally brings simplicity, ease of use to users in meteorological research field.

# Acknowledgement

We would like to thank Mr. Edward Ma who is our supervisor of this final year project. HIs guidance helped us make this project as a reality and become great success. With the help of his made valuable opinions, assistance and supports, we at last successfully made our UAV system.

We would also like to thank Ms. Ada Yuen who is our co-supervisor of this final year project. With her suggestions and opinions made, our system become more practical.

Finally, we would like to thank all the lecturers in this course. They have taught us these two years and making us more skillful and better when finishing this project.

# List of Tables and Figures

**Figure 1. reference from Dronecode**

**Figure 2. real-time view reference from DJI Go 4**

**Figure 3. reference from Dronecode**

**Figure 4. Mac OS App main interface**

**Figure 5. Flight instruments**

**Figure 6. Map view**

# Introduction of Project

Unmanned Aerial Vehicle (UAV) become more universal and mature in the modern society. The UAV is widely used for different fields including rescue operation, military affairs and physical distribution. For personal use, the UAV is mainly for high-altitude aerial photography. Some safety guideline and the legal is made for flying UAV by Hong Kong government. Because of the mature technology of UAV and social environment, we based on the UAV to develop an Automated Data Collecting System to user for environment data collection.

This project aims to build up a client-side application. The system build-up a communication between UAV and devices which include smartphone and computer. The UAV can transfer the real-time view and collected data to the devices. Also, the UAV can execute the instruction from the devices. Users can collect the climatic data easily from the high altitude through this system.

All collected data should be belonged to a mission which is users created before these data is collected. These data are not sent out from the device. The only aim of all collected data is for users' study. Also, these data are protected in users' device.

This system we expect that the system can suits professional environment analyst and normal user at the same time. we make the system become more user-friendly. Also, the system can meet the requirement of the environment data collection for professional users because the system can collect and display detailed environment data by specific UAV.

In this report, we will introduce the system design which describes how the UAV and devices communication to achieve live stream and data collection functions. Also, we consider the proposed solutions for hardware and software, the system's constraints and related regulation about UAV. In addition, we must make the project plan to arrange the time to make sure that the system can be completed on time

# Introduction

## The Description of Document Structure

This report has 8 major sections to help understanding process of the development of the new UAV System.

### Introduction

A general description of this document structure, the project and its background.

### The Problems

This part describes the problems we meet during the process.

### The Requirements

This part describes the scope of the system and the descriptions of the functions provided by the UAV.

### Documentation for Problems Analysis

This part is shown with the UML modeling to describe how the system works.

### Critical Evaluation

This part describes what potential difficulties we may face, or what problems

### Detailed Project Plan

This section will first show our project schedule and what deliverables will handout. Then, the resources we need for the project.

### References

There are some websites that we referred.

### Appendices

The original project plan with segmentation of activities and estimated times for those completion, which we submitted in our Initial Report, together with a revised plan, which records how you spent the time. Also, the copy of our project logbook.

# Description of the Project

Unmanned Aerial Vehicle (UAV) becomes more universal and mature in the modern society. It is widely used for different fields including rescue operation, military affairs and physical distribution. For personal use, the UAV is mainly used for high-altitude aerial photography. Some safety guidelines and the legal are made for flying UAV by the Hong Kong Government. Because of the mature technology of UAV and social environment, we based on the UAV to develop an Automated Data Collecting System to user for environment data collection.

This project aims to build up a client-side application. The system build-up a communication between UAV and devices which include smartphone and computer. The UAV can transfer the real-time view and collected data to the devices. Also, the UAV can execute the instruction from the devices. Users can collect the climatic data easily from the high altitude through this system.

All collected data should be belonged to a mission which is users created before these data is collected. These data are not sent out from the device. The only aim of all collected data is for users' study. Also, these data are protected in users' device.

We expect that this system can suit professional environment analyst and normal user at the same time. We make the system become more user-friendly. Also, the system can meet the requirement of the environment data collection for professional users because the system can collect and display detailed environment data by specific UAV.

In this report, we will introduce the system design which describes how the UAV and devices communication to achieve live stream and data collection functions. Also, we consider the proposed solutions for hardware and software, the system's constraints and related regulation about UAV. In addition, we must make the project plan to arrange the time to make sure that the system can be completed on time

In this report, we will present the design of the system and how it works. We will show you how the UAV and the devices communicate and how the data will be used. UML Diagrams will be shown to talk about the program design. Also, we have concerned about the limitation of developing the system based on reliability requirements, performance requirements, hardware, future extension and the implementation language. Problems that we may, or we have overcome and the proposed solutions in terms of software and hardware are included in this report.

# Background

As we know, nowadays, the environmental issue is increasingly severed. There is more and more exhaust emission because of industry high-speed development. This harm to personal health and the earth. So, people are highly worried about the environmental issue. A new environment data collecting system is expected. Our UAV could help people to collect climatic data accurately.

# Problem Description

Since the observatory cannot provide accurate data for every region, people can't get environmental data on a specific area.

First, the current climatic data collection tools normally work in the ground. The climatic analysis team can't collect climatic data from the sky. Many exhausts transmit to high altitude or exist in there. Therefore, the climatic analysis team can't get the most accurate data from the ground through the current data collection tool.

Second, someplace human can't arrive there easily such as disaster area and mountain. The climatic analysis team can't get the most accurate data from there. This will hinder some rescue operations or some research.

Third, the current climatic data tools may not include data recording function. The analysts must use another tool to record the climatic data manually. After collection, they also need to analyze these data record. This will increase the working process of climatic data analysis.

In this report, we

# Problem Environment

The data analysts are lack of electronic technology and information technology to create a new environment data collecting system. They can use the current climatic data tool only. Nowadays, the environmental issue is increasingly severed. A new environment data collecting system can help them to be more convenient and accurate to collect climatic data.

# The Requirements

## Scope of the system

This section is the discussion of system proposed to handle the problems encountered based on the design of the system, functionality and feasibility and the system requirements. Different aspects of problems have been concerned in terms of software and hardware layer. The discussed issues are as following:

* Mobile Application
  1. The way of UAV and smartphone to communication
  2. The structure of the mobile application
  3. The function of the mobile application
* MacOS Application
  1. The way of UAV and notebook computer to communication
  2. The structure of the MacOS application
  3. The function of the MacOS application
* UAV
  1. The inspection sensor of the UAV
* System Design
  1. How to make our application user-friendly
* Visualization for climatic data
* Description for the data outputting

## Mobile Application

### The way of UAV and smartphone to communication

Assume that the mobile application is installed to users' smartphone. The users will see the main page showing "no connection" when they open the mobile application. They should turn on the UAV and connect the smartphone to the Wi-Fi network of the UAVs. When the UAV and the smartphone are connected, the UAV will send the real-time view which is collected by the camera to the smartphone. The main page of the application will display the real-time view from the UAV when the users open the mobile application again. Also, the users can control the UAV by the smartphone including "rise", "decline", "go", "back", "turn" and "translation". These control request will send to the UAVs by the Wi-Fi network at real-time. The UAVs will response these control request.

### The structure of the mobile application

The mobile application is designed on one layer. The main view will display the real-time view from the UAV. On the right-top place, these are the attitude and the compass. Under the attitude and the compass, there is an instrument panel to show the UAV status including altitude, speed and flight time etc.  
On the main view, there will be two virtual joysticks for UAV control. The left joystick is for "rise", "decline" and "turn". The right joystick is for "go", "back" and "translation".

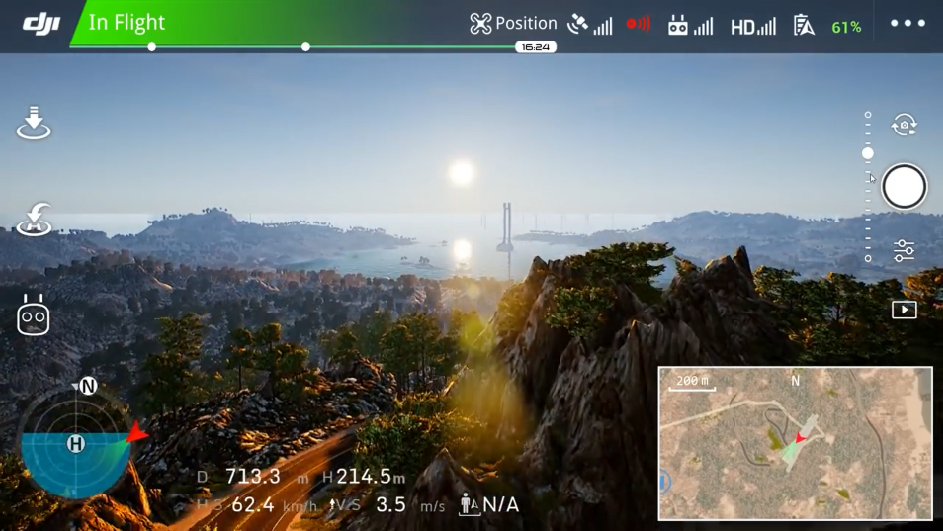


**Figure 1. reference from Dronecode**

## The function of the mobile application

### Display real-time view from UAV

Users can watch the real-time view of the UAV on the main view by the video streaming technology when the smartphone connects to the UAV. If the connection is failed, there are black view only.



**Figure 2. real-time view reference from DJI Go 4**

### Control UAV

Users could control the UAV by their mobile phones. The monitor will display the image captured by the camera on the UAV. Clients could control the UAV by virtual joysticks to move the UAV. The available actions include "rise", "decline", "go", "back", "turn" and "translation".

### Display details of UAV

Users can watch the UAV status information on the main view. This information includes attitude, direction, altitude, speed and flight time etc.



**Figure 3. reference from Dronecode**

### Take photo or video

Users can take photo during live stream. Also, users can take video that the UAV control period captured view is made a video.

### Connect UAV

The user should connect a UAV to the system. If the connection is invalid, the system cannot execute the main functions including streaming, displaying the details of the UAV, UAV controlling and taking photo or video.

### Disconnect UAV

When the UAV was connected to the system, the user can disconnect the UAV. After that, the main functions will be stopped.

## MacOS Application

### The way of UAV and notebook computer to communication

Assume that the Windows application is installed to users' notebook computer. The users will see the main page when they open the Windows application, but all view is black because of the invalid connection. They should turn on the UAV and connect the notebook computer to the Wi-Fi network of the UAV and press the "connect" button on the windows application to connect them. When the UAV and the notebook computer are connected, the UAV will send the real-time view which is collected by the camera to the notebook computer. The main page of the application will display the real-time view from the UAV, location on the map and the details of the UAV.

When the users want to collect the climatic data, they should press a "start" button to open all UAV's sensors. Then, the UAV will send the data to the notebook computer. And the notebook computer will show the real-time climatic data on the real-time view. There will be some charts to display the above data by the time period. At the same time, these data will be recorded.

### The structure of the MacOS application

The window application is designed on a layer. There is a menu bar on top. This will guide the user to execute different functions including "New Mission", "Output Result", "Quit", "Connect" and "Disconnect" functions. In addition, a "Start" button for starting data collection is placed a suitable position. These functions will be introduced after.

## The function of the MacOS application

### Build up new mission

Users can build up a new mission. The mission will keep all climatic data which the UAV will collect.

### Data collection

The real-time data will be collected as soon as the drone is connected to the application and the "Start" button is pressed. These collected real-time data will be shown by graph at a specific position where it is on the same page as real-time view.

### Save As

When the mission is done. The user can output all collected data with a JSON file and related graph to the directory which the user selects.

### Display real-time view from UAV

Users can watch the real-time view of the UAV on the view by the video streaming technology when the notebook computer connects to the UAV. If the connection is failed, there are black view only.

### Display location of UAV on map

Users can watch the location of the UAV on the map when the computer connects to the UAV. If the connection is failed, there are black view only.

### Display details of UAV

Users can watch the details of the UAV including airspeed, attitude, altitude, turn coordinator, heading and vertical speed on the view by when the computer connects to the UAV. If the connection is failed, there are black view only.

### Connect UAV

The user should connect a UAV to the system. If the connection is invalid, the system cannot execute the main functions including data collection, streaming, displaying location and details of the UAV.

### Disconnect UAV

If the user wants to stop using the UAV, the user can disconnect the UAV. After that, the main functions will be stopped, but all collected data will be kept.

## UAV

### The inspection sensor of the UAV

The system minimally requires a gyroscope, accelerometer, magnetometer (compass) and barometer. A GPS or other positioning system is needed to enable all automatic modes, and some assisted modes.

* MS5611 barometer
* U-blox M8N GLONASS/GPS/Beidou (antenna connector type MCX)

The system contains two 9DOF (degree of freedom) IMU - MPU9250 and LSM9DS1. Each of them combines a gyroscope, an accelerometer and a magnetometer in one device. IMU sensors are not only popular as a part of drone autopilot projects, but are also widely used in devices like cellphones, tablets, etc.

* MPU9250 9DOF IMU
* LSM9DS1 9DOF IMU

For Climatological data, we install below sensors:

* DHT22 Temperature-Humidity Sensor

+The DHT22 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air. It's fairly simple to use but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, sensor readings can be up to 2 seconds old.

* MQ-2 Semiconductor Sensor for Combustible Gas Sensor

Sensitive material of MQ-2 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exists, the sensor’s conductivity is higher along with the gas concentration rising. Using a simple electro circuit, convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensitivity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

* SDS011 Laser PM2.5 Sensor

The SDS011 using principle of laser scattering, can get the particle concentration between 0.3 to 10μm in the air. It with digital output and built-in fan is stable and reliable.

* UV Detection Sensor

Just a normal UV sensor.

## System Design

### How to make our application user-friendly

To increase the system acceptability, we will develop different prototype to test to make sure all users can use our system easily without professional knowledge. And the system will use simple English to reduce hinder when new user uses our system.

### Visualization for climatic data

The system will provide the visualization climatic data. The system will make some graphs using the UAV collected data to display the patterns or the trends of the environment changing to the user. These data include PM2.5, UV, temperature, humidity and gas etc. The system will generate the graphs instantly when the data is received from the UAV' sensors. The meteorological research team can research the environment by these graphs. Finally, these graphs can be outputted and keep for user to study further.

### Description for the data outputting

For the climatic data, it will be outputted by JSON format to a file. JSON is the best tool for the sharing data because the data is stored in the arrays. When the user wants to further handle these data, this format is suitable for different software. In the file, it will include all data during the whole running time, interval and unit for each sensor. The following is a sample of the JSON file.

{

" humidity":

{

"data" : [55, 50,55,57]

" unit" : "%RH"

" interval" : "5000"

}

"temperature":

{

" data" : [15.25,15.22,15.20,25.19]

"unit" : "C"

"interval" : "5000"

}

}

## Description of Hardware Requirements

### Client

any modern computer or mobile device. Performance will depend on the system environment, 3rd party applications, and available system resources. More capable hardware will provide a better experience. A computer with at least 8Gb RAM, an SSD, Nvidia or AMD graphics and an i5 or better CPU will be suitable for most applications.

Support Device:

Mobile with Google Android 5.1 or later.

Completer with MacOS 10.14

## Non-Functional Requirement

### User-Friendly GUI for the customer

We used the knowledge that we have learnt in the course called Human Computer Interaction and GUI Programming to make a user-friendly GUI for the users. As it is our first time doing this kind of projects, we have taken some references on some mobile applications that we will mark references on their design.

Constraints  
Wi-Fi stability problem

For our project, we use an AWUS036NHA Wi-Fi adapter to broadcast the video and control signal. This adapter will provide around 280mW output power. Ranges of several kilometers have been reported (with directional antennas though). But the big problem is the connection problems. Sometimes, the network connect was too slow, the mobile app needs a few times to update the information to keep the data consistency.

Also, we cannot confirm the range because we do not want to take the risk to lose our device.

## Battery capacity limitation

For our project, we consider using 4s3p Samsung 30Q LiPo battery pack. This Samsung 18650 battery will provide max 3000mAh of rated discharge capacity and max 20A discharge current.

For the flight time:

**Flight times = (Battery Capacity \* Battery Discharge /Average Amp Draw) \* 60 minutes**

Flight times=(9000mAh∗80%/20A)∗60 minutes=21.6 minutes

Under my calculation, this battery will provide 21.6 minutes of flight times.

## Fire risk with the LiPo battery

For our project, we use Lithium Polymer technology which allows considerable energy to be stored in a small package. But they are associated with fire risk and have been known to spontaneously catch fire while charging or if punctured. Most airlines restrict the numbers of batteries on board. Check with your airline for their policy on transporting batteries. It is recommended never to leave them unattended while charging and to transport and store them in a fireproof box.

## Weather problem

It is not recommended to fly drones in rain, snow, or even in the drizzle. Apart from any physical effects on the aircraft, there is a danger the electronics will be damaged, and communication between the controller and the drone can be affected when there is any kind of precipitation in the air.

## Hong Kong Drone Regulations

According to Hong Kong national aviation authority, the Hong Kong Civil Aviation Department (HKCAD), flying a drone is legal in Hong Kong, but they also set some rules:

**General Safety Guidelines for Flying UAS**

Any person who operates UAS shall observe the following general safety guidelines, unless otherwise permitted by CAD.

**1.Where you cannot fly**

(i)UAS shall not be flown in the vicinity of an airport and aircraft approach and take-off paths, which include:

-Areas within 5 km of Hong Kong International Airport;

-North Lantau coastal area;

-Coastal areas from Tai Lam Chung to Tsuen Wan and Tsing Yi Island;

-Victoria Harbour and its coastal areas; and

-Shek Kong area.

(ii)UAS shall not be flown over populated and congested areas.

(iii)UAS shall not be flown over, or close to, any object, installation or facility that would present a risk to safety in the event of damage due to any impact by the UAS, or in such a manner that the good order and discipline and control thereof may be or may likely be jeopardized.

3

**2.Good Choice of Flying Sites**

A good choice of flying site not only minimises the possibility of an accident but also enhances the pleasure of the flying experience.

Here are some tips to help you choose a good site. A good site should be:

(i)clear of persons, vessels, vehicles or structures;

(ii)away from helicopter landing pads;

(iii)clear of any power sources such as power lines, transformer stations, pylons, and transmitter towers etc., which might cause radio interference;

..3(iv)flat enough to enable safe take-off and landing; and

(v)free from visual obstruction, so that the operator can see the UAS in flight at all times.

**3.Altitude**

The altitude of operations shall not exceed 300 feet (~90 m) above ground level.

**4.Time of Operations**

The operations of UAS shall be conducted during daylight hours only.

**5.Weather Criteria**

(i)UAS shall only be operated under good visibility and weather conditions.

(ii)UAS shall not be operated when Rainstorm Warning, Tropical Cyclone Warning or Strong Monsoon Signal is in force.

**6.Heavy UAS**

Except with Civil Aviation Department's endorsement, heavy UAS weighing more than 7 kg (without its fuel) is not allowed to fly in Hong Kong.

**7.Control of UAS**

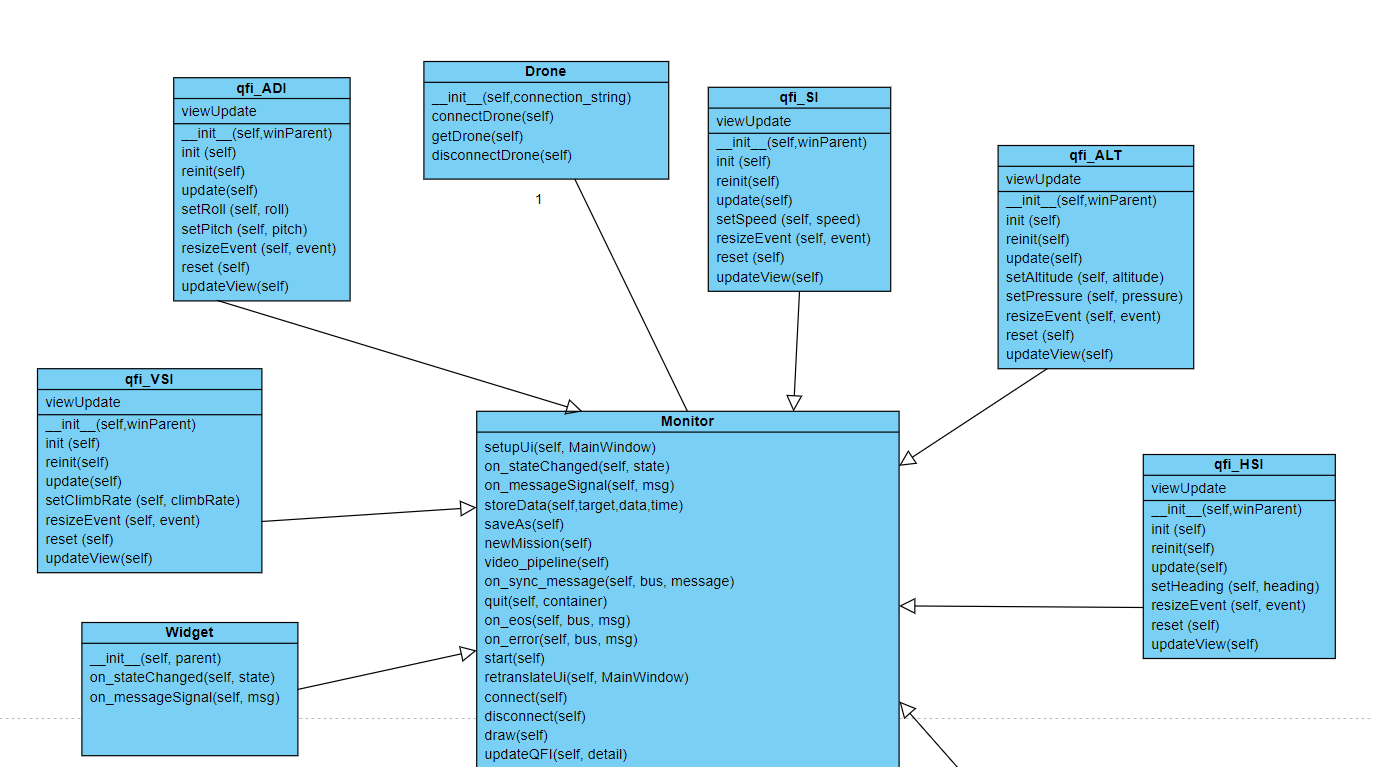
The UAS operator shall be on site and keep the UAS within Visual Line of Sight (VLOS) during the period of the flight. Operating within VLOS means that the UAS operator is able to maintain direct, unaided (other than corrective lenses) visual contact with the UAS, and is able to monitor the UAS flight path in relation to other aircraft, persons, vessels, vehicles and structures for the purpose of avoiding collisions.

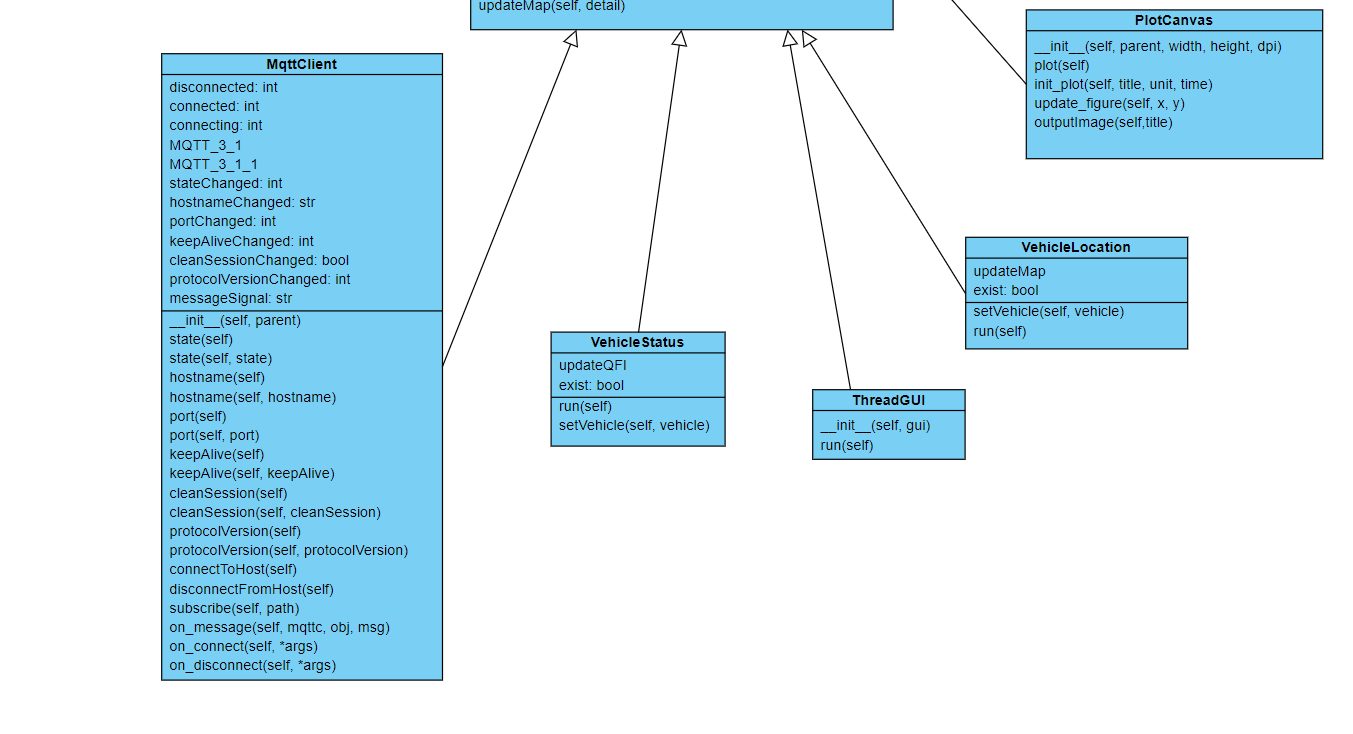
**8.Safety Check on UAS**

The UAS operator shall perform a safety check on UAS (including its fuel or lithium batteries) before operating UAS.

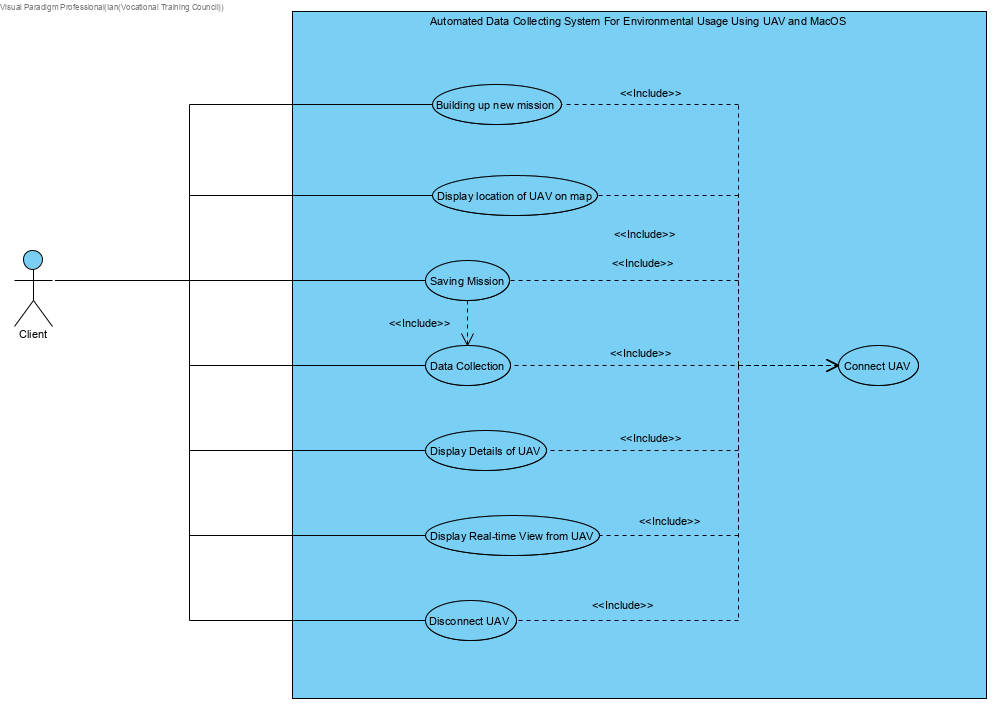
Documentation for problem analysis

## Class Diagram (Mac OS app)





## Use Case Diagram (Mac OS app)



Actor Description (MacOS App)

|  |  |
| --- | --- |
| Actor Name: | User |
| Description: | User is the end-user of the UAV. |

Initial Use Case Description (MacOS App)

|  |  |
| --- | --- |
| Use case name: | Connect UAV |
| Use case ID: | UC-0001 |
| Actor(s): | User |
| Brief description: | User connects UAV to the computer. |

|  |  |
| --- | --- |
| Use case name: | Display Real-time View from UAV |
| Use case ID: | UC-0002 |
| Actor(s): | User |
| Brief description: | User can see the live streaming from UAV. |

|  |  |
| --- | --- |
| Use case name: | Data Collection |
| Use case ID: | UC-0003 |
| Actor(s): | User |
| Brief description: | User can view the real-time climatic data on the graphs. |

|  |  |
| --- | --- |
| Use case name: | Building up new mission |
| Use case ID: | UC-0004 |
| Actor(s): | User |
| Brief description: | User can build up new mission to collect real-time climatic data. |

|  |  |
| --- | --- |
| Use case name: | Saving mission |
| Use case ID: | UC-0005 |
| Actor(s): | User |
| Brief description: | User can save the current mission by pressing “Save As” button. |

|  |  |
| --- | --- |
| Use case name: | Display location of UAV on map |
| Use case ID: | UC-0006 |
| Actor(s): | User |
| Brief description: | User can see the real-time location of the UAV on the map. |

|  |  |
| --- | --- |
| Use case name: | Display details of UAV |
| Use case ID: | UC-0007 |
| Actor(s): | User |
| Brief description: | User can see the status of UAV in the icons. |

|  |  |
| --- | --- |
| Use case name: | Disconnect UAV |
| Use case ID: | UC-0008 |
| Actor(s): | User |
| Brief description: | User can disconnect the UAV by pressing “Disconnect” button |

Refined Use Case Description (MacOS App)

|  |  |
| --- | --- |
| Use case name: | Connect UAV |
| Use case ID: | UC-0001 |
| Actor(s): | User |
| Brief description: | User connects UAV to the computer. |
| Preconditions | The computer has installed the app |
| Flow of events: | 1.open the application in the computer.  2. switch on the UAV  3. Connect both the UAV and notebook computer to the same Wi-Fi |
| Postconditions: | ／ |
| Alternative flows and exceptions: | User had to connect the correct Wi-Fi. Otherwise, it cannot connect the UAV. |
| Non-behavior requirements: | ／ |
| Assumptions: | The user uses Windows computer devices. |

|  |  |
| --- | --- |
| Use case name: | Display Real-time View from UAV |
| Use case ID: | UC-0002 |
| Actor(s): | User |
| Brief description: | User can see the live streaming from UAV. |
| Preconditions | The computer had connected to correct and available Wi-Fi |
| Flow of events: | 1.Include (Connect UAV)  2.switch on camera on UAV connected  3.Click view live button  4.The app displays the live image. |
| Postconditions: | ／ |
| Alternative flows and exceptions: | User had to connect the correct Wi-Fi. Otherwise, it cannot view live image the UAV. |
| Non-behavior requirements: | ／ |
| Assumptions: | The user uses MacOS computer devices. |

|  |  |
| --- | --- |
| Use case name: | Data Collection |
| Use case ID: | UC-0003 |
| Actor(s): | User |
| Brief description: | User can view the real-time climatic data on the graphs. |
| Preconditions | The computer had connected to correct and available Wi-Fi |
| Flow of events: | 1.Include (Connect UAV)  2.switch on the UAV and connects it to the computer  3.Real-time climatic data will be collected and display in the set position immediately after successfully connected both computer and UAV |
| Postconditions: | ／ |
| Alternative flows and exceptions: | User had to connect the correct Wi-Fi. Otherwise, it cannot view live image the UAV. |
| Non-behavior requirements: | ／ |
| Assumptions: | The user uses MacOS computer devices. |

|  |  |
| --- | --- |
| Use case name: | Build up new mission |
| Use case ID: | UC-0004 |
| Actor(s): | User |
| Brief description: | User can build up new mission to collect real-time climatic data. |
| Preconditions | The computer had connected to correct and available Wi-Fi |
| Flow of events: | 1.Include (Connect UAV)  2.switch on the UAV and connects it to the computer  3.Click “build up new mission” button  4.A new mission will be built. |
| Postconditions: | ／ |
| Alternative flows and exceptions: | User had to connect the correct Wi-Fi. Otherwise, it cannot view live image the UAV. |
| Non-behavior requirements: | ／ |
| Assumptions: | The user uses MacOS computer devices. |

|  |  |
| --- | --- |
| Use case name: | Saving mission |
| Use case ID: | UC-0005 |
| Actor(s): | User |
| Brief description: | User can save the current mission by pressing “Save As” button. |
| Preconditions | The computer had connected to correct and available Wi-Fi |
| Flow of events: | 1.Click “Save as” button  2.Collected data will be output as a json file and plot graph. |
| Postconditions: | ／ |
| Alternative flows and exceptions: | User had to connect the correct Wi-Fi. Otherwise, it cannot view live image the UAV. |
| Non-behavior requirements: | ／ |
| Assumptions: | The user uses MacOS computer devices. |

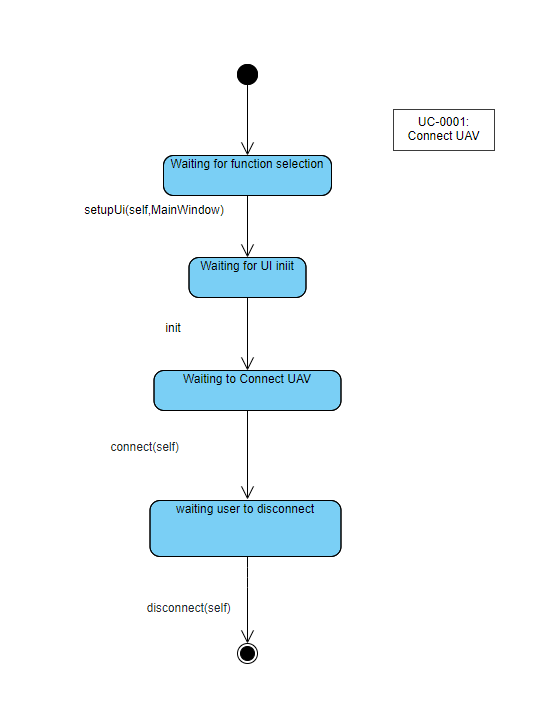
|  |  |
| --- | --- |
| Use case name: | Display location of UAV on map |
| Use case ID: | UC-0006 |
| Actor(s): | User |
| Brief description: | User can see the real-time location of the UAV on the map. |
| Preconditions | The computer had connected to correct and available Wi-Fi |
| Flow of events: | 1.Include (Connect UAV)  2.switch on the UAV and connects it to the computer  3. Real-time location of UAV shown on the map |
| Postconditions: | ／ |
| Alternative flows and exceptions: | User had to connect the correct Wi-Fi. Otherwise, it cannot view live image the UAV. |
| Non-behavior requirements: | ／ |
| Assumptions: | The user uses MacOS computer devices. |

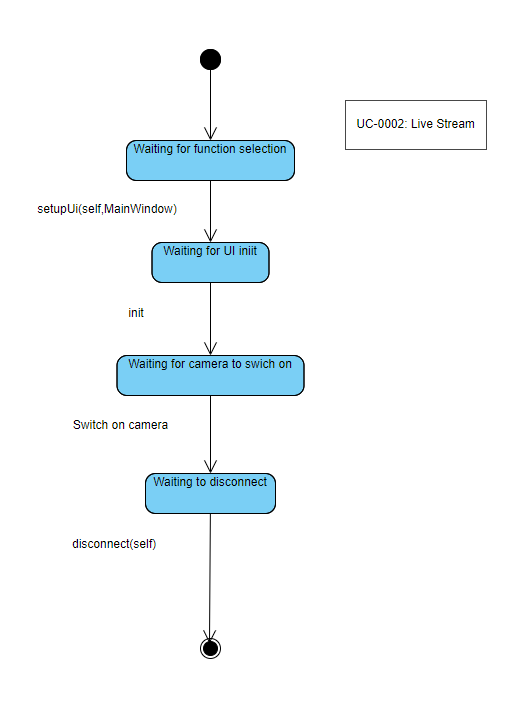
|  |  |
| --- | --- |
| Use case name: | Display details of UAV |
| Use case ID: | UC-0007 |
| Actor(s): | User |
| Brief description: | User can see the status of UAV in the icons. |
| Preconditions | The computer had connected to correct and available Wi-Fi |
| Flow of events: | 1.Include (Connect UAV)  2.switch on the UAV and connects it to the computer  3.Details of the UAV will be updated in the icons |
| Postconditions: | ／ |
| Alternative flows and exceptions: | User had to connect the correct Wi-Fi. Otherwise, it cannot view live image the UAV. |
| Non-behavior requirements: | ／ |
| Assumptions: | The user uses MacOS computer devices. |

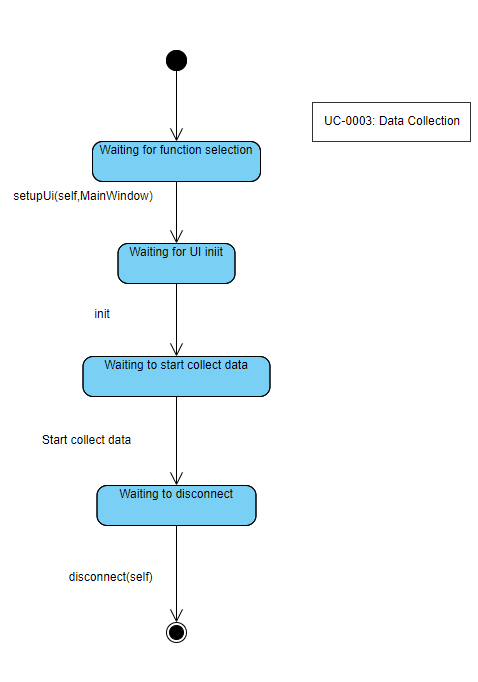
|  |  |
| --- | --- |
| Use case name: | Disconnect UAV |
| Use case ID: | UC-0008 |
| Actor(s): | User |
| Brief description: | Disconnect UAV and the devices |
| Preconditions | User can disconnect the UAV by pressing “Disconnect” button |
| Flow of events: | 1.Include (Connect UAV)  2.switch on the UAV and connects it to the computer  3.Click “Disconnect” button  4. UAV will be disconnected from the devices |
| Postconditions: | ／ |
| Alternative flows and exceptions: | User had to connect the correct Wi-Fi. Otherwise, it cannot view live image the UAV. |
| Non-behavior requirements: | ／ |
| Assumptions: | The user uses MacOS computer devices. |

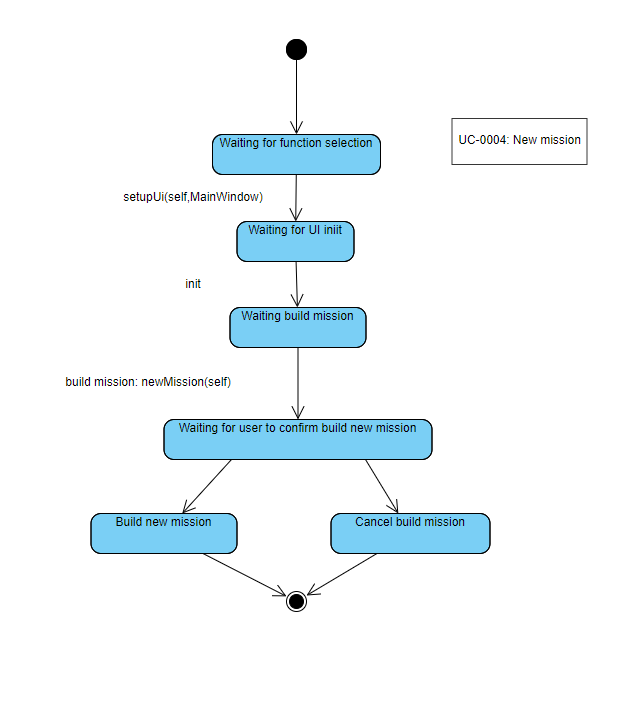
## State Transition Diagram

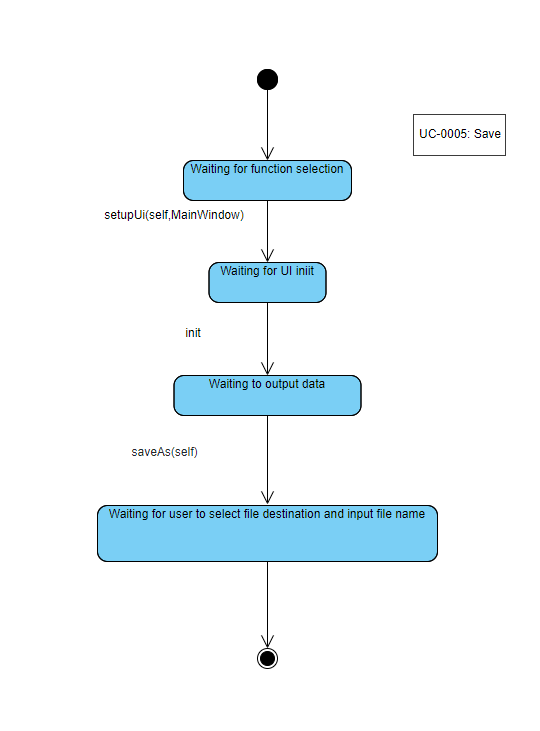
MacOS Application

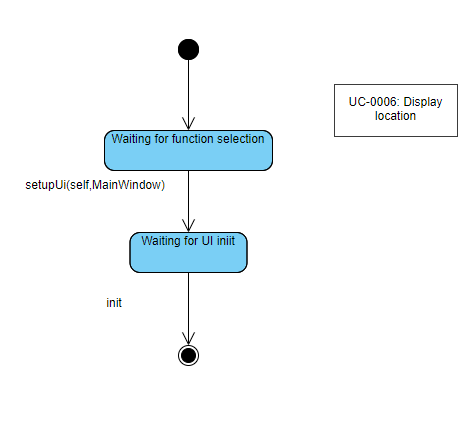


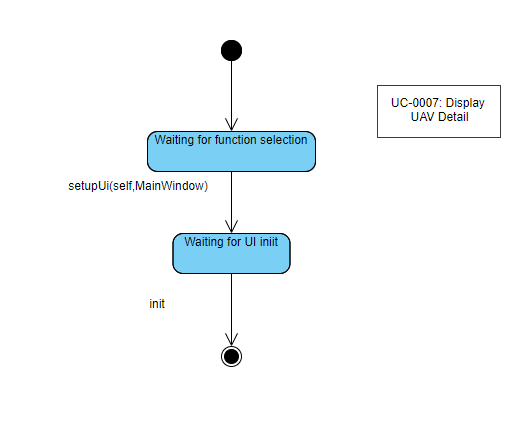


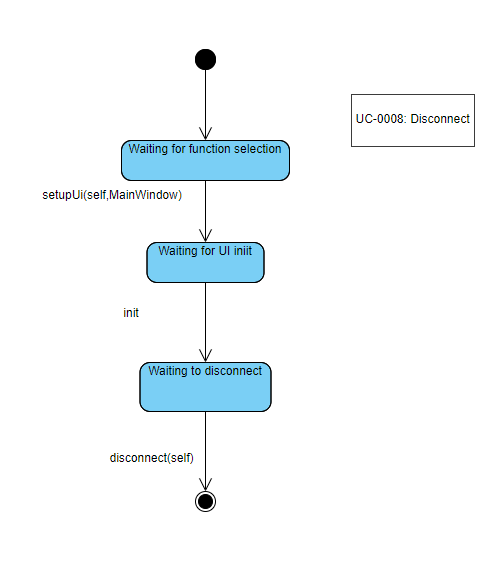




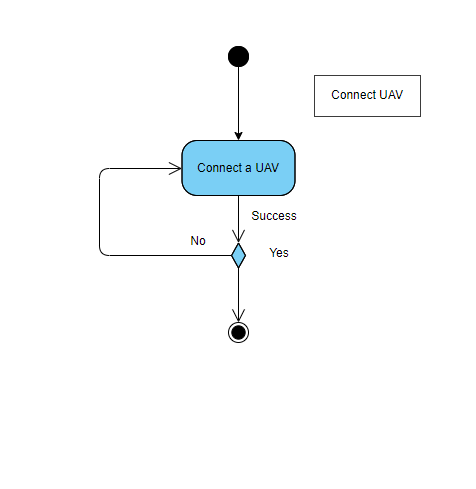


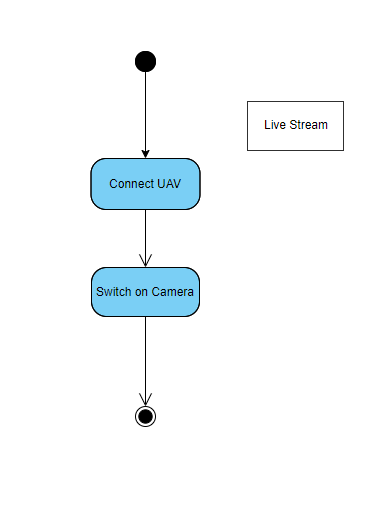


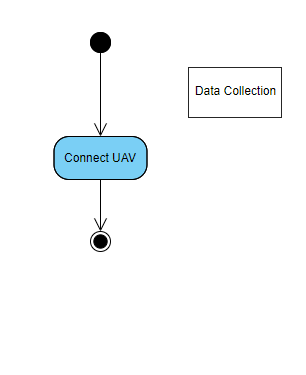


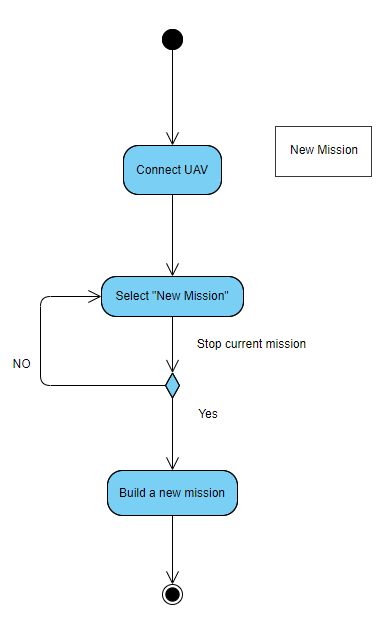


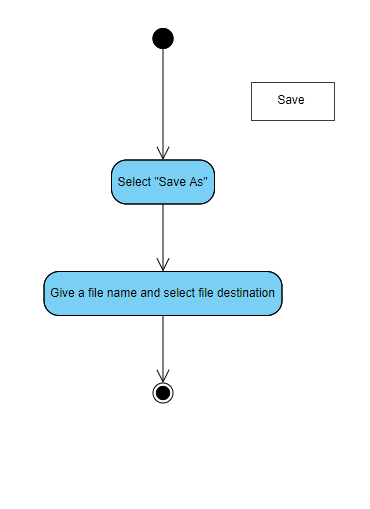
Activity Diagram (Mac OS App)



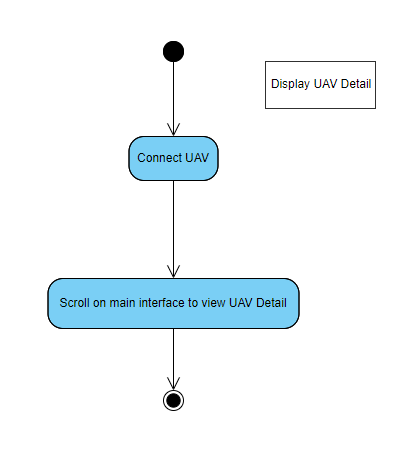


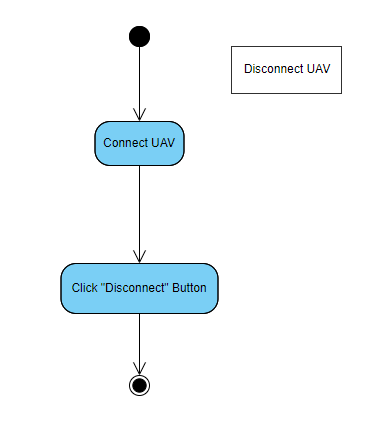






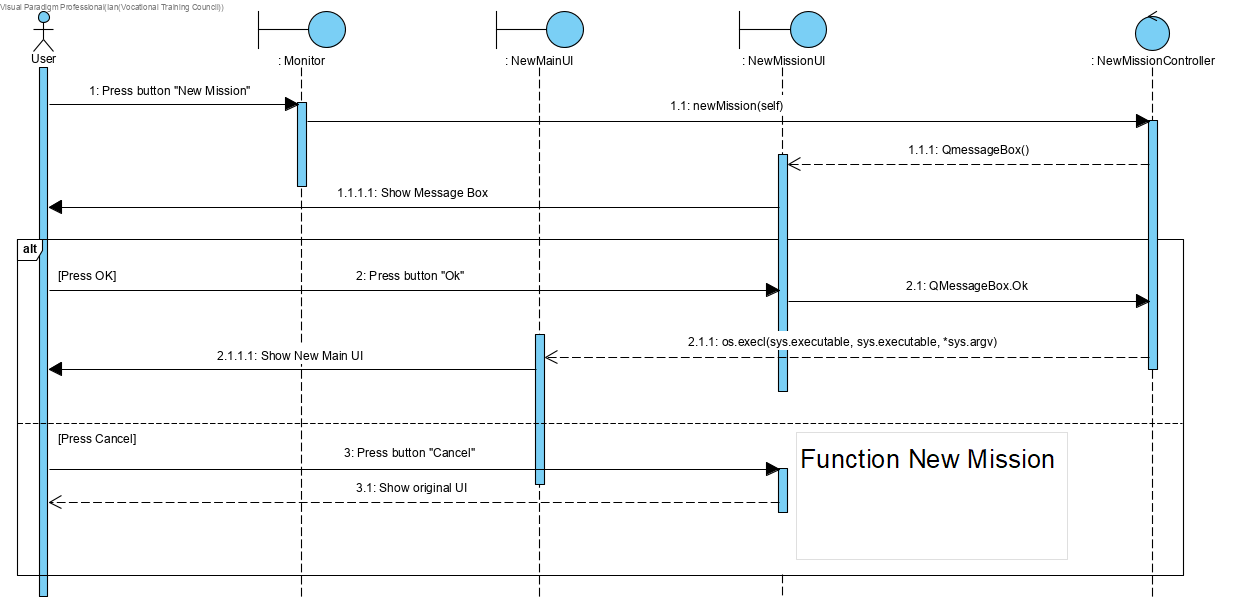




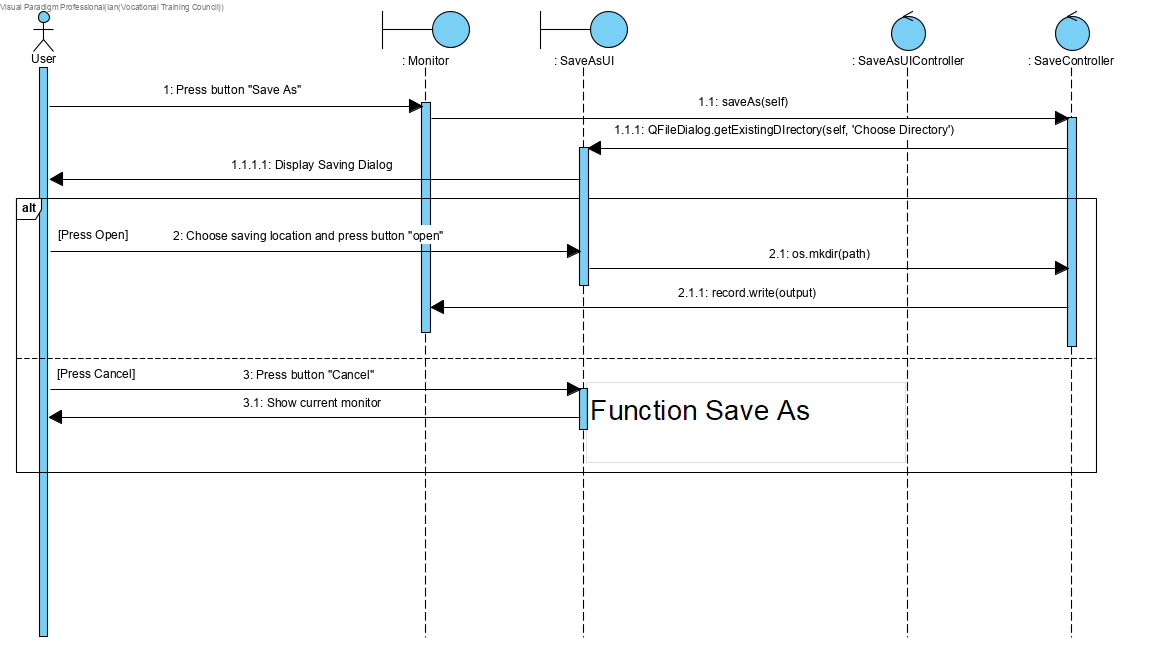


## Sequence Diagram (Window App)

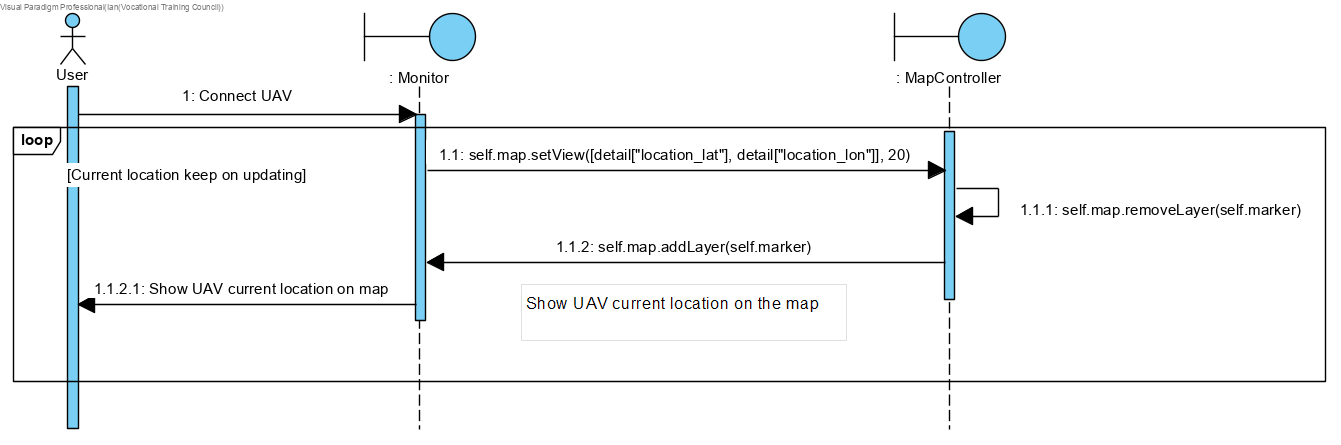
### Function “New Mission”



### Function “Save As”

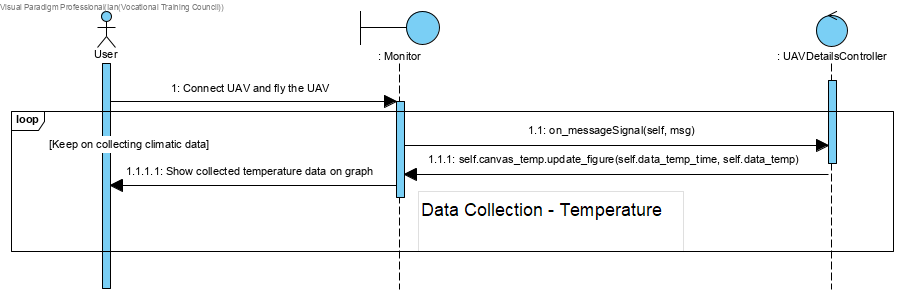


### Function “Display location of UAV on map”

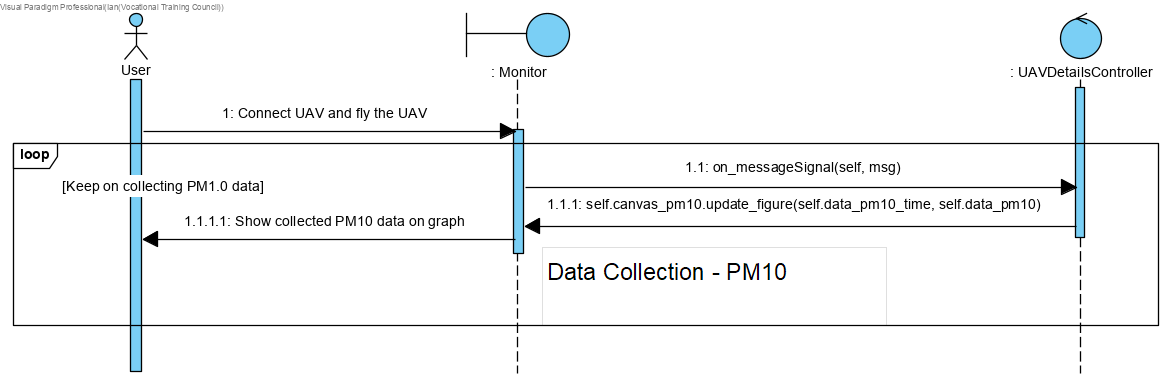


### Function “Data Collection”

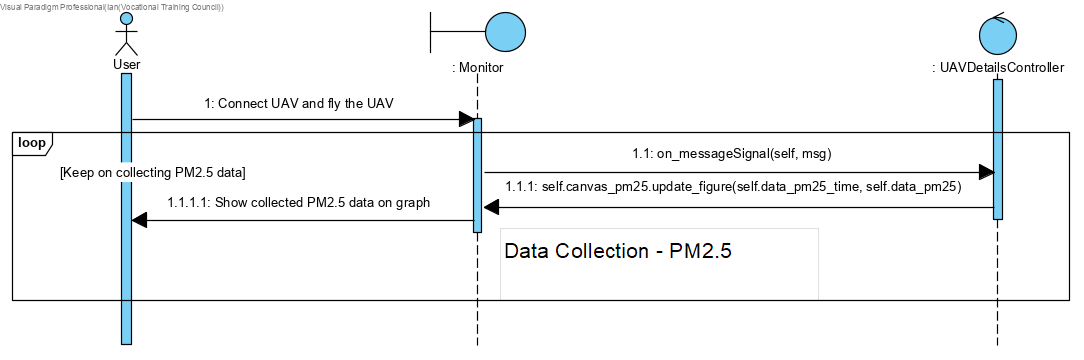
#### Temperature



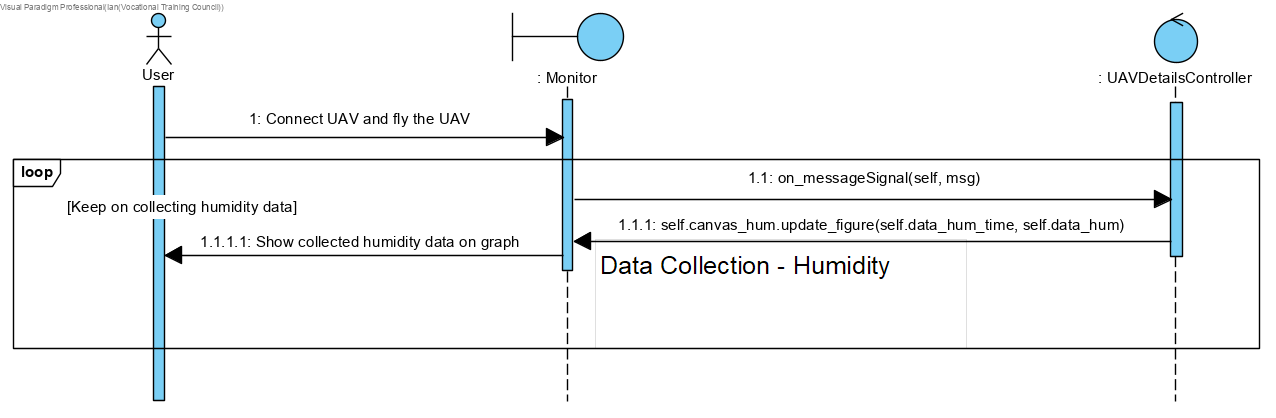
#### PM10



#### PM2.5

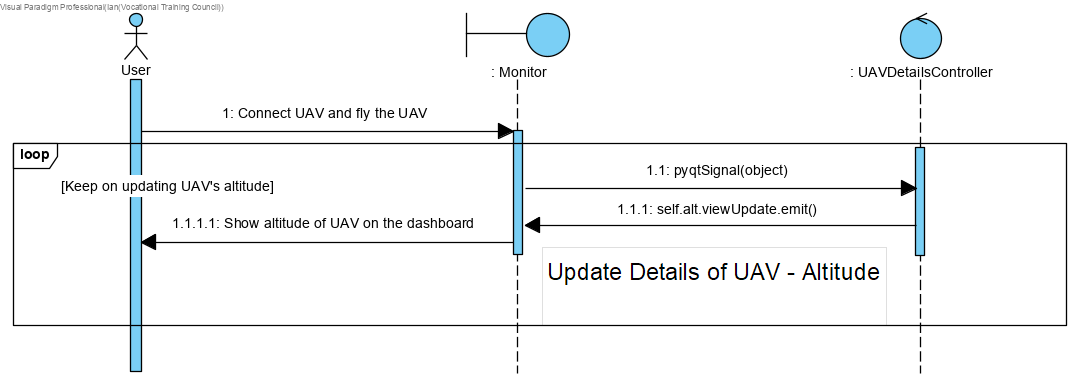


#### Humidity

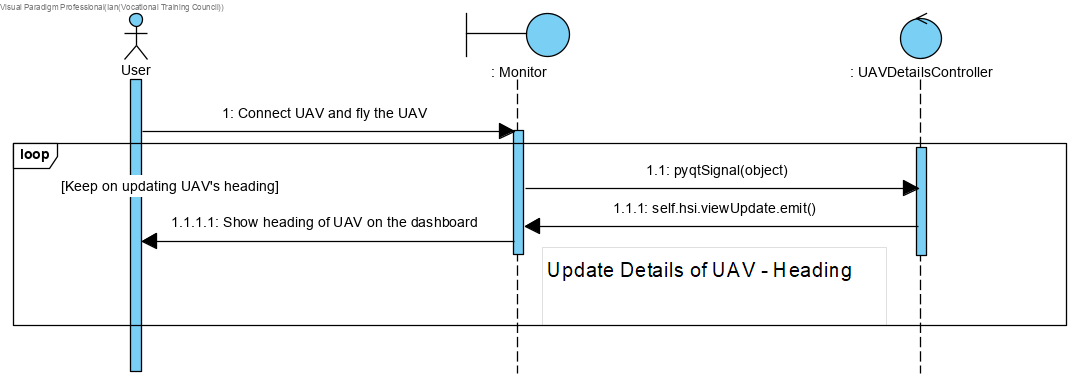


### Function “Display Details of UAV”

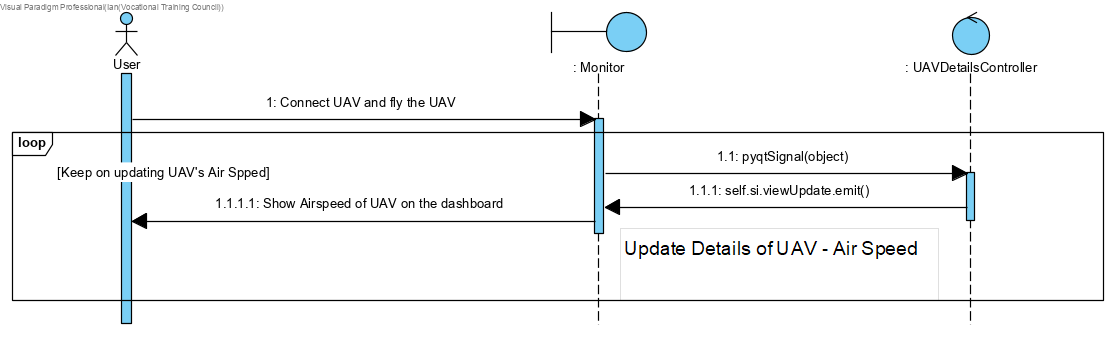
#### Altitude



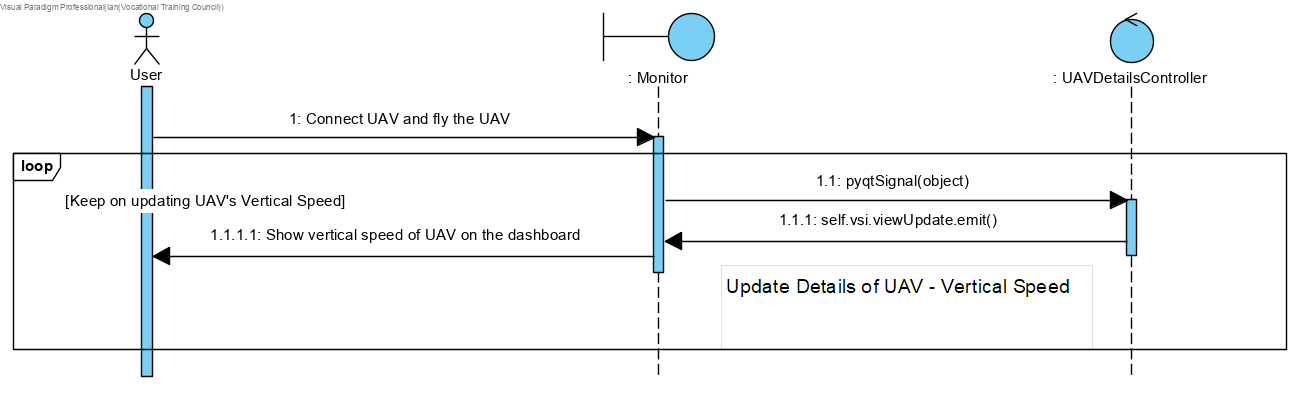
#### Heading



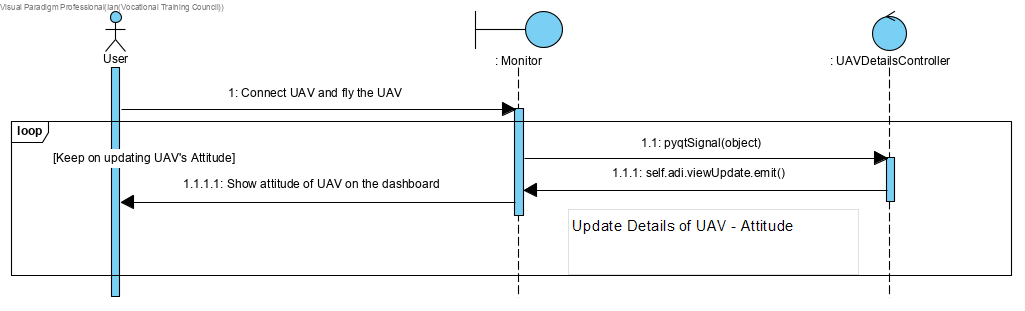
#### Airspeed



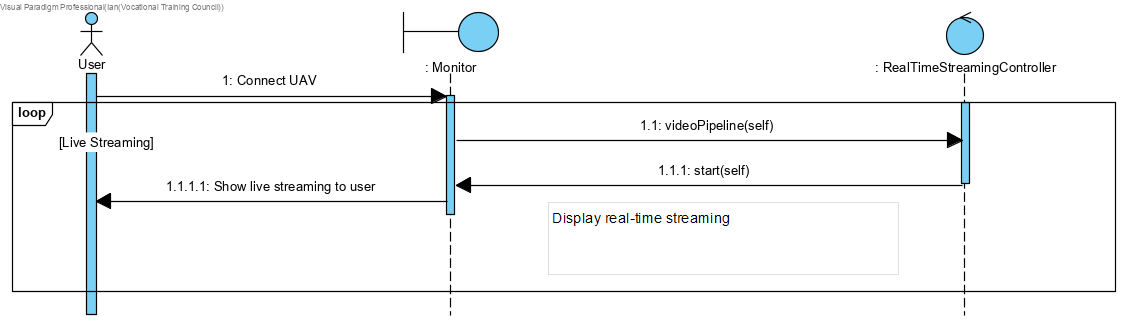
#### Vertical Speed



#### Attitude



## Function “Display real-time view from UAV”



# Data Dictionary

{

"Temperature":{

"Data":[],

"Unit":C,

"CollectedTime":[]

}

"Humidity":{

"Data":[],

"Unit":%,

"CollectedTime":[]

}

"PM2.5":{

"Data":[],

"Unit":µg/m³,

"CollectedTime":[]

}

"PM10":{

"Data":[],

"Unit":µg/m³,

"CollectedTime":[]

}

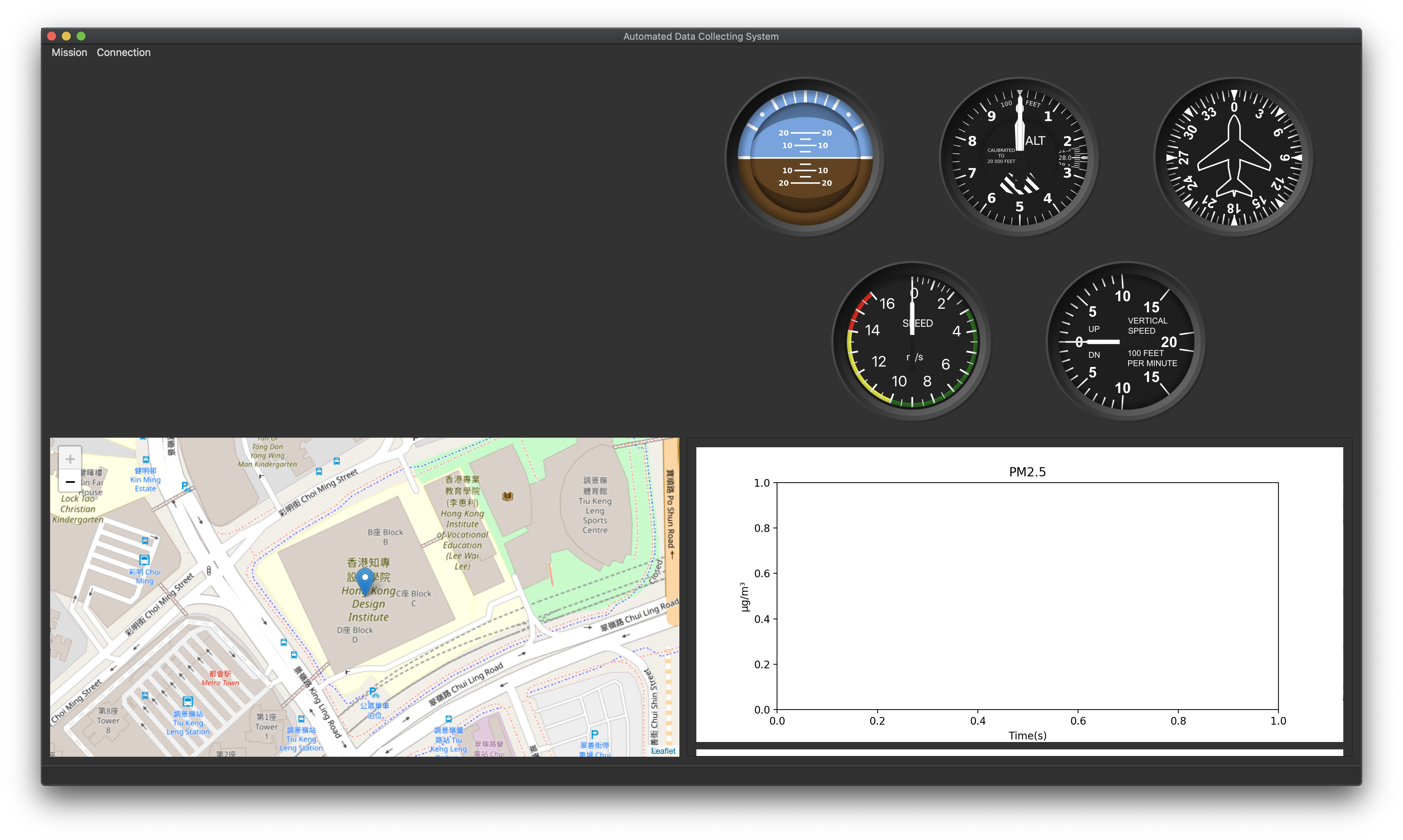
}

# Documentation for Detailed Design

## UI Design

### MacOS Application User Interface Design

The main view consists of four-part including the live view, details of the UAV, the map and the graphs. The live view displays the real-time view from the UAV by live-streaming technology.



**Figure 4. Mac OS App main interface**

### Flight Instruments

The detail of the UAV displays the UAV's information including airspeed, attitude, altitude, heading and vertical speed by the flight instruments.



**Figure 5. Flight instruments**

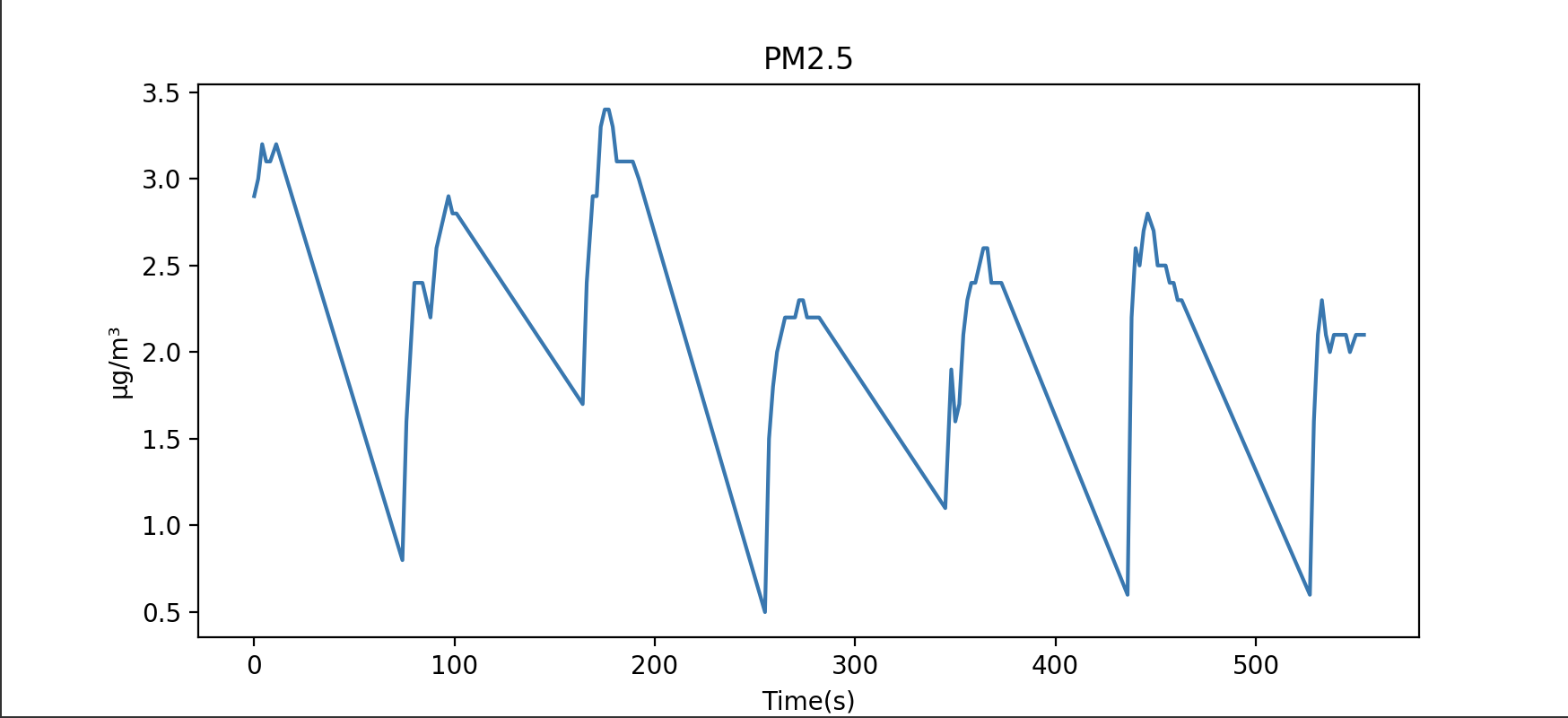
### Map

The map view displays the location of the UAV.

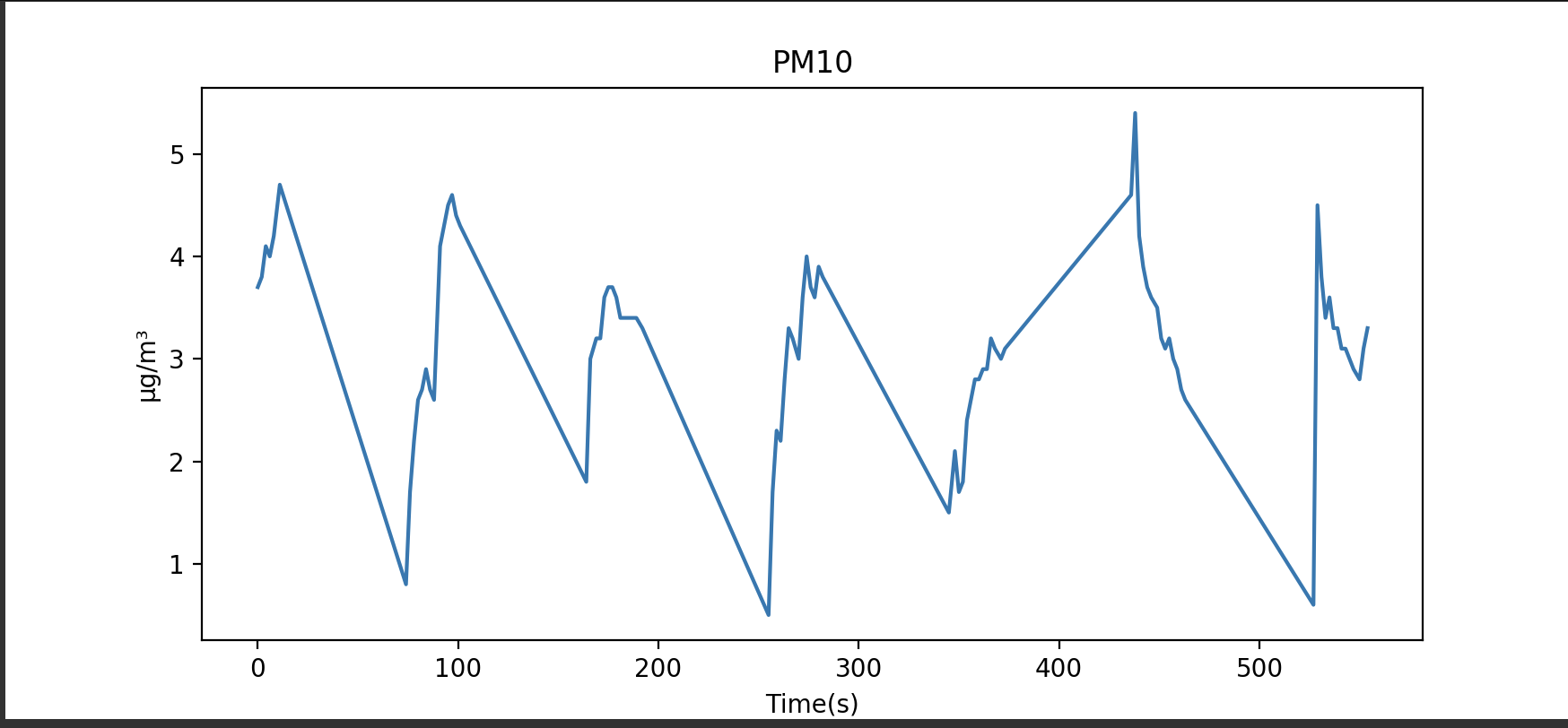
**Figure 6. Map view**

### Graph

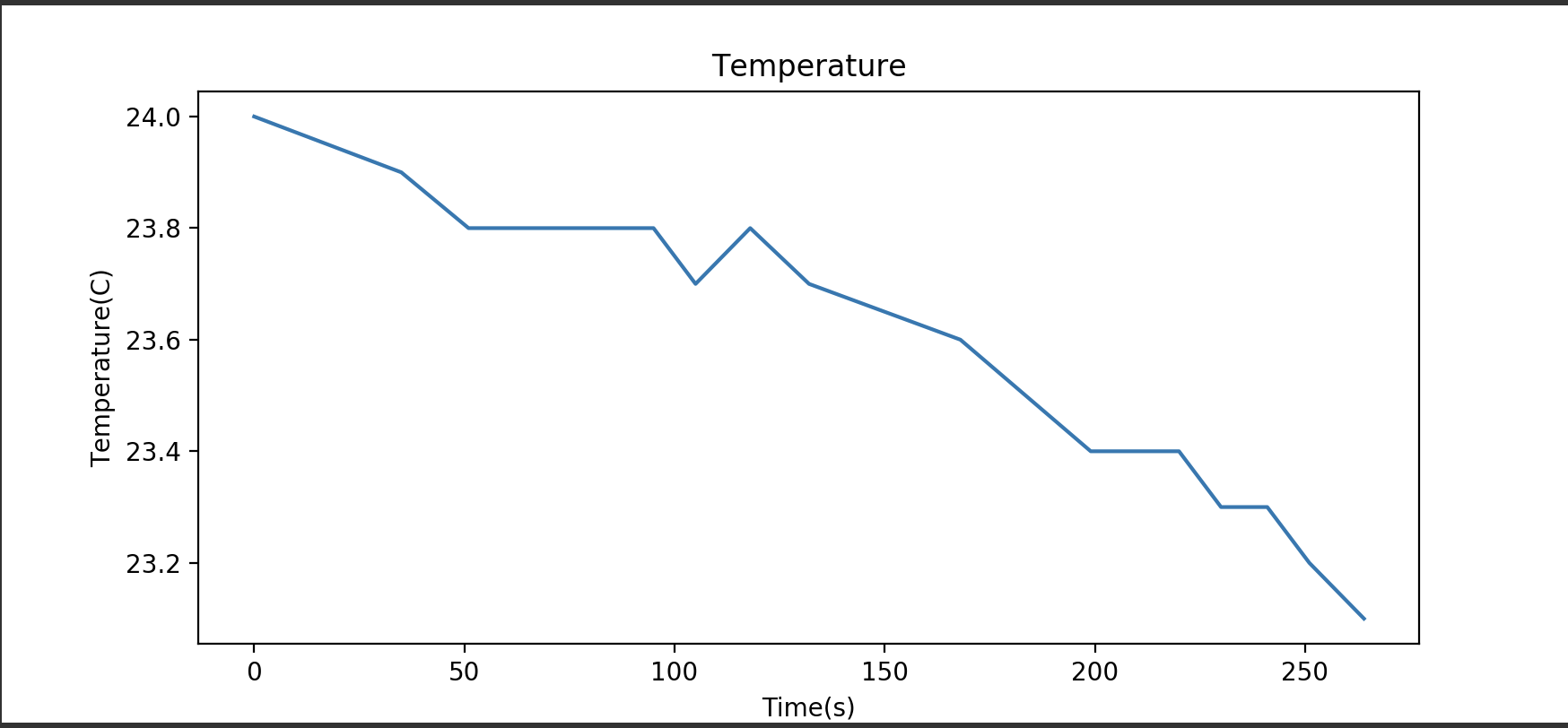
For the data collection, the system will draw the graphs by the collected data. There are four graphs. The user can use the scroll bar to watch another graph.



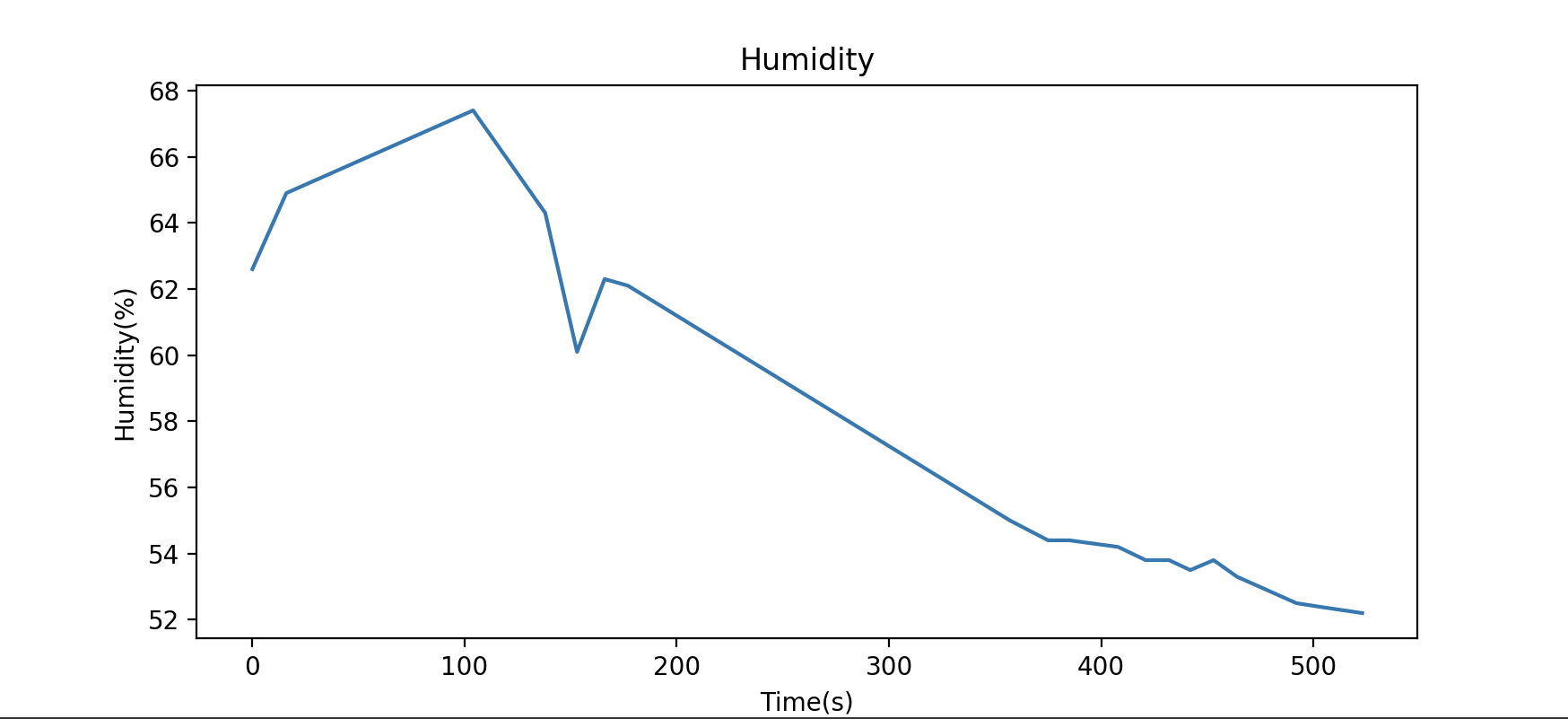
**Figure 7. PM2.5 graph**



**Figure 8. PM10 graph**



**Figure 9. Temperature graph**



**Figure 10. Humidity**

# Critical Evaluation

## Sensor side

The sensor is a module who purpose is to detect changes in its environment and send the information to computer. In our case, we use MQ-2 gas sensor, UV sensor, DHT22 temperature-humidity sensor, SDS011 PM sensor. We use USB interface to connect SDS011 to raspberry pi (rpi). DHT22 connect to the rpi’s gpio. MQ-2 and UV sensor need an analog-to-digital converter (adc) to convert an analog signal into a digital signal that computer can understand. There is a problem that the adc use i2c bus to communication, but Navio2 (Flight Controller) uses all i2c bus, so I try to use a i2c to usb adapter to make the sensors working.

## Laptop side

The laptop side just like the first officer in an aircraft. The laptop provides map, Live streaming from the drone, flight instruments and the data from the sensors.

# Implementation

## MacOS Application Test Plan and Test Result

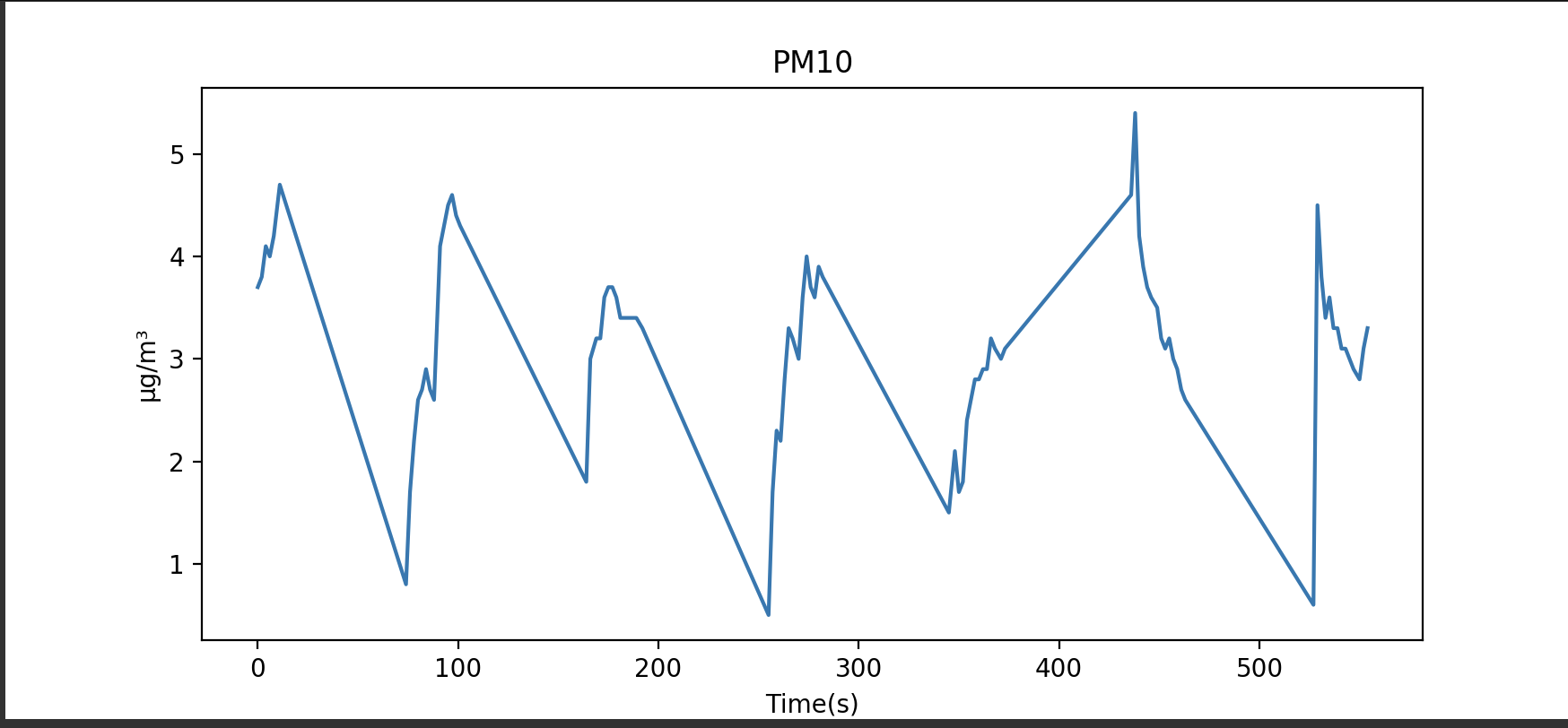
### View collected data of PM2.5

|  |  |  |
| --- | --- | --- |
| Test Case ID | Test Case Description | Expected Result |
| 1 | View PM2.5 data | Successful |

### 

### View collected data of PM10

|  |  |  |
| --- | --- | --- |
| Test Case ID | Test Case Description | Expected Result |
| 2 | View PM10 data | Successful |



### View collected data of Temperature

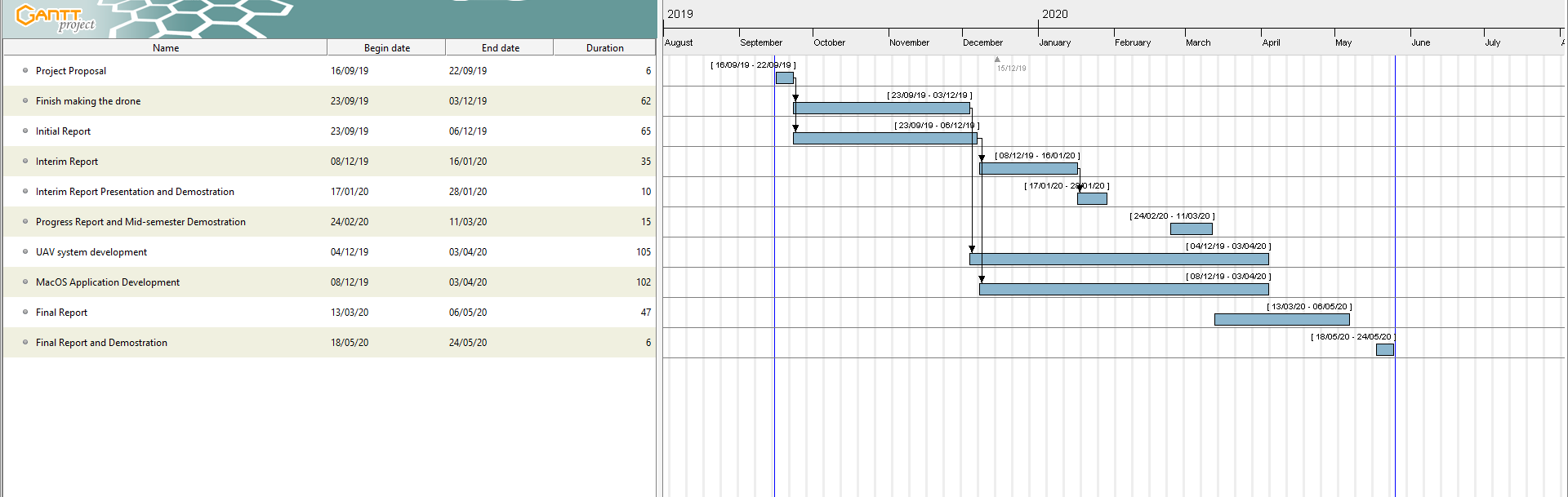
|  |  |  |
| --- | --- | --- |
| Test Case ID | Test Case Description | Expected Result |
| 3 | View temperature data | Successful |

Reference

* [DroneKit Python](https://github.com/dronekit/dronekit-python) - The drone API used
* [QGroundControl](http://qgroundcontrol.com/) - The ground control station used on smartphone
* [create\_ap](https://github.com/oblique/create_ap) - The Access Point API used
* [GStreamer](https://gstreamer.freedesktop.org/) - The media-handling component used
* [PyQt](https://riverbankcomputing.com/software/pyqt/intro) - The GUI framework used
* [GTK](https://www.gtk.org/) - The GUI framework used
* [Eclipse Paho](https://www.eclipse.org/paho/) - The MQTT API used
* [PyGObject](https://pygobject.readthedocs.io/en/latest/index.html) - It provides bindings for GObject based libraries such as GTK, GStreamer.
* [pyqtlet](https://github.com/skylarkdrones/pyqtlet) - The map wrapper used
* [Matplotlib](https://matplotlib.org/) - The library for creating graphics and charts used
* [QFlightInstruments](https://github.com/JdeRobot/ThirdParty/tree/master/qflightinstruments) - The flight instrument used

# Appendices

## A1. Project Plan



## A2. Log Sheet

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 01 )**

|  |  |
| --- | --- |
| Course/Class: ITP114105 2A | Project ID: 9 |
| Supervisor: Edward Ma | Date: 17/9/2019 |
| Project Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s):  Wong Ming Yuen, Chau Yat Sum, Kwok Tsz Lung, Siu Chi Wang\_ | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Discuss the project title |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| Confirm the project title |  |
| Plan of work before the next recording | Logged by |
| Confirm the project title, purchasing list, allocation of working of the project proposal |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 02 )**

|  |  |
| --- | --- |
| Course/Class: ITP114105 2A | Project ID: 9 |
| Supervisor: Edward Ma | Date: 20/9/2019 |
| Project Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s):  Wong Ming Yuen, Chau Yat Sum, Kwok Tsz Lung, Siu Chi Wang\_ | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Discuss the project title |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| Confirm the project title |  |
| Plan of work before the next recording | Logged by |
| Confirm the project title, purchasing list, allocation of working of the project proposal |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 03 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 24/9/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Confirmed the project title, finished project proposal, allocation of working of each teammate |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| Purchasing list |  |
| Plan of work before the next recording | Logged by |
| Confirm the purchase list, work on the Initial Report, work on the UAV system we are making |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 04 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 30/9/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Confirmed the project title, finished project proposal, allocation of working of each teammate |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| Purchasing list |  |
| Plan of work before the next recording | Logged by |
| Confirm the purchase list, work on the Initial Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 05 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 8/10/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| Confirmation of purchase list, working on Initial Report |  |
| Plan of work before the next recording | Logged by |
| Confirm the purchase list, work on the Initial Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 06 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 22/10/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Confirmation of the purchase list |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| Wait for the purchased items for the project, working on Initial Report |  |
| Plan of work before the next recording | Logged by |
| Wait for the purchased items for the project, work on the Initial Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 07 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 29/10/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| Wait for the purchased items for the project, working on Initial Report |  |
| Plan of work before the next recording | Logged by |
| Wait for the purchased items for the project, work on the Initial Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 08 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 12/11/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| UAV parts arrived |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on Initial Report |  |
| Plan of work before the next recording | Logged by |
| work on the Initial Report, working on the UAV system we are making |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 09 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 26/11/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on Initial Report, working on the UAV system we are making |  |
| Plan of work before the next recording | Logged by |
| work on the Initial Report, working on the UAV system we are making |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 10 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 9/12/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on Initial Report, working on the UAV system we are making |  |
| Plan of work before the next recording | Logged by |
| work on the Initial Report, working on the UAV system we are making |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 11 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 16/12/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on Initial Report, working on the UAV system we are making |  |
| Plan of work before the next recording | Logged by |
| work on the Initial Report, working on the UAV system we are making |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 12 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 19/12/2019 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on Initial Report, working on the UAV system we are making |  |
| Plan of work before the next recording | Logged by |
| work on the Initial Report, working on the UAV system we are making |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 13 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 10/01/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Finished Initial Report |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, work on Interim Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 14 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 15/01/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, work on Interim Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, work on Interim Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 15 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 24/01/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, work on Interim Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, work on Interim Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 16 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 31/01/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, work on Interim Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, work on Interim Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 17 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 6/2/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Finished Interim Report |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, work on Interim Report Presentation and Demonstration |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 18 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 13/2/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on Interim Report Presentation and Demonstration |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, work on Interim Report Presentation and Demonstration |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 19 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 22/2/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Finished working on Interim Report Presentation and Demonstration |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on Interim Report Presentation and Demonstration |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, start working on progress report and Mid-semester Demostration |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 20 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 27/2/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Finished working on Interim Report Presentation and Demonstration |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, start working on progress report and Mid-semester Demonstration |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 21 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 4/3/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on progress report and Mid-semester Demonstration |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, start working on progress report and Mid-semester Demonstration |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 22 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 10/3/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| Finished progress report and Mid-semester Demonstration |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, start working on Final Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 23 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 18/3/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on Final Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, working on Final Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 24 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 31/3/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on Final Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, working on Final Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 25 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 16/3/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on Final Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, working on Final Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 26 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 30/3/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on Final Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, working on Final Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 27 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 8/5/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on Final Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, working on Final Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

**Hong Kong Institute of Vocational Education**

**Discipline of Information Technology**

**Final Year Project - Log Sheet (No. 28 )**

|  |  |
| --- | --- |
| Course/Class: IT114105-2A | Project ID: 9 |
| Supervisor: Mr Edward Ma\_\_ | Date: 14/5/2020 |
| Project Title: Automated Data Collecting System for Environment Using UAVs and Smartphones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Student Name(s): Wong Ming Yuen, Siu Chi Wang, Kwok Tsz Lung, Chau Yat Sum | |

|  |  |
| --- | --- |
| Work done & findings since last recording | Logged by |
| -Nil- |  |
| State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan | Logged by |
| working on the UAV system we are making, working on Final Report |  |
| Plan of work before the next recording | Logged by |
| working on the UAV system we are making, working on Final Report |  |

|  |
| --- |
| Supervisor’s comment on student’s work |
|  |

|  |  |  |
| --- | --- | --- |
| Student’s Signature: |  | Supervisor Signature |

A3. Job Division

|  |  |
| --- | --- |
| Members | Job in this group project |
| Siu Chi Wang | * Build UAV * Raspberry Pi Configuration * Python Application development (Monitor):   Video Streaming,  IoT,  Flight Instruments,  README   * MQTT Server on UAV * MQTT Service for publish data to Monitor * Co-operation with other members’ job |
| Wong Ming Yuen | * Python Application development (Monitor):   GUI,  Charts & Graphs,  Map,   * MQTT Service for subscribe data form UAV * Co-operation with other members’ job |
| Kwok Tsz Lung | Use Case Diagram,  Sequence Diagram   * Co-operation with other members’ job |
| Chau Yat Sum | Class Diagram,  State Machine Diagram,   * Co-operation with other members’ job |

Program Listing

monitor.py

*# -\*- coding: utf-8 -\*-*  
*import* math  
import os  
import time  
from datetime **import** datetime  
import sys  
from PyQt5 **import** QtCore, QtWidgets  
from PyQt5.QtCore **import** QThread  
from PyQt5.QtWidgets **import** QFileDialog, QMessageBox, QApplication, QMainWindow, QWidget  
  
from drone **import** Drone  
from vehicleStatus **import** VehicleStatus  
from vehicleLocation **import** VehicleLocation  
  
from pyqtlet **import** L, MapWidget  
  
import gi  
from gi.overrides **import** Gtk  
  
gi.require\_version(**'Gst'**, **'1.0'**)  
gi.require\_version(**'GstVideo'**, **'1.0'**)  
from gi.repository **import** Gst, GObject  
  
GObject.threads\_init()  
Gst.init(**None**)  
  
from mqttClient **import** MqttClient  
from plotCanvas **import** PlotCanvas  
  
from threadGUI **import** ThreadGUI  
from qfi **import** qfi\_ADI, qfi\_ALT, qfi\_SI, qfi\_HSI, qfi\_VSI, qfi\_TC  
  
  
class Monitor(QMainWindow):  
 **def** setupUi(self, MainWindow):  
 self.drone = **None**  
MainWindow.setObjectName(**"MainWindow"**)  
 MainWindow.resize(1792, 1008)  
 sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed, QtWidgets.QSizePolicy.Fixed)  
 sizePolicy.setHorizontalStretch(0)  
 sizePolicy.setVerticalStretch(0)  
 sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightForWidth())  
 MainWindow.setSizePolicy(sizePolicy)  
 self.centralwidget = QtWidgets.QWidget(MainWindow)  
 self.centralwidget.setObjectName(**"centralwidget"**)  
 self.gridLayout\_1 = QtWidgets.QGridLayout(self.centralwidget)  
 self.gridLayout\_1.setObjectName(**"gridLayout\_1"**)  
  
 *# creat Simple Window*  
self.container = QWidget(self)  
 self.container.setFixedSize(854,480)  
 self.container.setWindowTitle(**'Test1'**)  
  
 *# container.connect('destroy', self.quit)*  
self.setCentralWidget(self.container)  
 self.winId = self.container.winId()  
 *#self.resize(480, 320)*  
  
 *# Create GStreamer pipeline*  
self.videoPipeline()  
  
 *# Create bus to get events from GStreamer pipeline*  
self.bus = self.pipeline.get\_bus()  
 self.bus.add\_signal\_watch()  
 self.bus.enable\_sync\_message\_emission()  
 self.bus.connect(**'message::error'**, self.on\_error)  
 self.bus.connect(**'message::eos'**, self.on\_eos)  
 self.bus.connect(**'sync-message::element'**, self.on\_sync\_message)  
  
 *# uav detail ui*  
self.gridLayout = QtWidgets.QGridLayout()  
  
 self.adi = qfi\_ADI.qfi\_ADI(self)  
 self.adi.resize(240, 240)  
 self.adi.reinit()  
 self.gridLayout.addWidget(self.adi, 0, 0)  
  
 self.alt = qfi\_ALT.qfi\_ALT(self)  
 self.alt.resize(240, 240)  
  
 self.alt.reinit()  
 self.gridLayout.addWidget(self.alt, 0, 1)  
  
 self.hsi = qfi\_HSI.qfi\_HSI(self)  
 self.hsi.resize(240, 240)  
 self.hsi.reinit()  
 self.gridLayout.addWidget(self.hsi, 0, 2)  
  
 self.si = qfi\_SI.qfi\_SI(self)  
 self.si.resize(240, 240)  
 self.si.reinit()  
 self.gridLayout.addWidget(self.si, 1, 0,1,2,alignment=QtCore.Qt.AlignCenter)  
  
 self.vsi = qfi\_VSI.qfi\_VSI(self)  
 self.vsi.resize(240, 240)  
 self.vsi.reinit()  
 self.gridLayout.addWidget(self.vsi, 1, 1,1,2,alignment=QtCore.Qt.AlignCenter)  
  
 self.tc = qfi\_TC.qfi\_TC(self)  
 self.tc.resize(240, 240)  
 self.tc.reinit()  
  
 self.setLayout(self.gridLayout)  
  
 *# end uav detail ui*  
  
  
self.gridLayout\_1.addWidget(self.container,0,0)  
 self.gridLayout\_1.addLayout(self.gridLayout,0,1)  
  
 *# Map*  
self.mapWidget = MapWidget()  
 sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred, QtWidgets.QSizePolicy.Preferred)  
 sizePolicy.setHorizontalStretch(1)  
 sizePolicy.setVerticalStretch(1)  
 sizePolicy.setHeightForWidth(self.mapWidget.sizePolicy().hasHeightForWidth())  
 self.mapWidget.setSizePolicy(sizePolicy)  
 self.gridLayout\_1.addWidget(self.mapWidget,1,0)  
 self.map = L.map(self.mapWidget)  
  
 self.map.setView([22.305711, 114.253426], 20)  
 L.tileLayer(**'http://{s}.tile.osm.org/{z}/{x}/{y}.png'**).addTo(self.map)  
  
 self.marker = L.marker([22.305711, 114.253426])  
 self.marker.bindPopup(**'No connection'**)  
 self.map.addLayer(self.marker)  
 self.count = 0  
  
 *#build scrollArea which is placed the graphs*  
self.scrollArea = QtWidgets.QScrollArea(self.centralwidget)  
 self.scrollArea.setVerticalScrollBarPolicy(QtCore.Qt.ScrollBarAlwaysOn)  
 self.scrollArea.setHorizontalScrollBarPolicy(QtCore.Qt.ScrollBarAlwaysOff)  
 self.scrollArea.setWidgetResizable(**True**)  
 self.scrollArea.setObjectName(**"scrollArea"**)  
 self.scrollAreaWidgetContents = QtWidgets.QWidget()  
 self.scrollAreaWidgetContents.setGeometry(QtCore.QRect(0, 0, 849, 1024))  
 self.scrollAreaWidgetContents.setObjectName(**"scrollAreaWidgetContents"**)  
 self.verticalLayout\_graphs = QtWidgets.QVBoxLayout(self.scrollAreaWidgetContents)  
 self.verticalLayout\_graphs.setObjectName(**"verticalLayout\_graphs"**)  
  
 *# draw graphs*  
 *# 1 PM2.5*  
self.data\_pm25 = []  
 self.data\_pm25\_time = []  
 self.data\_pm25\_collectTime = []  
 self.canvas\_pm25 = PlotCanvas(self, width=1, height=4)  
 self.canvas\_pm25.init\_plot(**"PM2.5"**, **"µg/m³"**, **"Time(s)"**)  
 self.canvas\_pm25.setMinimumSize(self.canvas\_pm25.size())  
 self.verticalLayout\_graphs.addWidget(self.canvas\_pm25)  
 *# 2 PM10*  
self.data\_pm10 = []  
 self.data\_pm10\_time = []  
 self.data\_pm10\_collectTime = []  
 self.canvas\_pm10 = PlotCanvas(self, width=1, height=4)  
 self.canvas\_pm10.init\_plot(**"PM10"**, **"µg/m³"**, **"Time(s)"**)  
 self.canvas\_pm10.setMinimumSize(self.canvas\_pm10.size())  
 self.verticalLayout\_graphs.addWidget(self.canvas\_pm10)  
 *# 3 temperature*  
self.data\_temp = []  
 self.data\_temp\_time = []  
 self.data\_temp\_collectTime = []  
 self.canvas\_temp = PlotCanvas(self, width=1, height=4)  
 self.canvas\_temp.init\_plot(**"Temperature"**, **"Temperature(C)"**, **"Time(s)"**)  
 self.canvas\_temp.setMinimumSize(self.canvas\_temp.size())  
 self.verticalLayout\_graphs.addWidget((self.canvas\_temp))  
 *# 4 humidity*  
self.data\_hum = []  
 self.data\_hum\_time = []  
 self.data\_hum\_collectTime = []  
 self.canvas\_hum = PlotCanvas(self, width=1, height=4)  
 self.canvas\_hum.init\_plot(**"Humidity"**, **"Humidity(%)"**, **"Time(s)"**)  
 self.canvas\_hum.setMinimumSize(self.canvas\_hum.size())  
 self.verticalLayout\_graphs.addWidget((self.canvas\_hum))  
  
 self.scrollArea.setWidget(self.scrollAreaWidgetContents)  
 self.gridLayout\_1.addWidget(self.scrollArea,1,1)  
  
 *# Menu*  
self.menubar = QtWidgets.QMenuBar(MainWindow)  
 self.menubar.setGeometry(QtCore.QRect(0, 0, 1792, 22))  
 self.menubar.setObjectName(**"menubar"**)  
 self.menuMission = QtWidgets.QMenu(self.menubar)  
 self.menuMission.setObjectName(**"menuFile"**)  
 self.menuConnection = QtWidgets.QMenu(self.menubar)  
 self.menuConnection.setObjectName(**"menuConnection"**)  
 MainWindow.setMenuBar(self.menubar)  
 self.statusbar = QtWidgets.QStatusBar(MainWindow)  
 self.statusbar.setObjectName(**"statusbar"**)  
 MainWindow.setStatusBar(self.statusbar)  
  
 *# new mission*  
self.actionNew\_Mission = QtWidgets.QAction(MainWindow)  
 self.actionNew\_Mission.setObjectName(**"actionNew\_Mission"**)  
 self.actionNew\_Mission.triggered.connect(self.newMission)  
 self.actionView\_Mission = QtWidgets.QAction(MainWindow)  
 self.actionView\_Mission.setObjectName(**"actionView\_Mission"**)  
  
 *# save as*  
self.actionSave = QtWidgets.QAction(MainWindow)  
 self.actionSave.setObjectName(**"actionSave"**)  
 self.actionSave.triggered.connect(self.saveAs)  
  
 *# quit*  
self.actionClose = QtWidgets.QAction(MainWindow)  
 self.actionClose.setObjectName(**"actionClose"**)  
 self.actionClose.triggered.connect(QApplication.quit)  
  
 *# connect*  
self.actionConnect = QtWidgets.QAction(MainWindow)  
 self.actionConnect.setObjectName(**"actionConnect"**)  
 self.actionConnect.triggered.connect(self.connect)  
  
 *#disconnect*  
self.actionDisconnect = QtWidgets.QAction(MainWindow)  
 self.actionDisconnect.setObjectName(**"actionDisconnect"**)  
 self.actionDisconnect.triggered.connect(self.disconnect)  
  
 self.menuMission.addAction(self.actionNew\_Mission)  
 self.menuMission.addAction(self.actionSave)  
 self.menuMission.addAction(self.actionClose)  
 self.menuConnection.addAction(self.actionConnect)  
 self.menuConnection.addAction(self.actionDisconnect)  
 self.menubar.addAction(self.menuMission.menuAction())  
 self.menubar.addAction(self.menuConnection.menuAction())  
  
 *# menu button setting*  
self.actionConnect.setDisabled(**False**)  
 self.actionDisconnect.setDisabled(**True**)  
  
 self.retranslateUi(MainWindow)  
 QtCore.QMetaObject.connectSlotsByName(MainWindow)  
 self.menubar.setNativeMenuBar(**False**) *# False for current window, True for parent window*  
  
 *# update detail*  
self.updateQFI\_thread = VehicleStatus()  
 self.updateQFI\_thread.updateQFI.connect(self.updateQFI)  
 self.updateMap\_thread = VehicleLocation()  
 self.updateMap\_thread.updateMap.connect(self.updateMap)  
  
 *#build thread*  
self.thread1 = QThread()  
 self.thread2 = QThread()  
 self.updateQFI\_thread.moveToThread(self.thread1)  
 self.updateMap\_thread.moveToThread(self.thread2)  
  
 *# start thread*  
self.thread1.started.connect(self.updateQFI\_thread.run)  
 self.thread1.start()  
 self.thread2.started.connect(self.updateMap\_thread.run)  
 self.thread2.start()  
  
 MainWindow.setCentralWidget(self.centralwidget)  
  
  
 @QtCore.pyqtSlot(int)  
 **def** on\_stateChanged(self, state):  
 **if** state == MqttClient.Connected:  
 print(state)  
 self.client.subscribe([(**"/IoTSensor/DHT22"**,0),(**"/IoTSensor/SDS011"**,1)])  
  
 @QtCore.pyqtSlot(str)  
 **def** on\_messageSignal(self, msg):  
 val = msg  
 print(val)  
 type = val.split(**" "**)[1].split(**"="**)[0]  
 **if** type == **"Temperature"**:  
 val = val.replace(**"Time="**, **""**)  
 val = val.replace(**"Temperature="**, **""**)  
 val = val.replace(**"Humidity="**, **""**)  
 val = val.split(**" "**)  
 sTime = val[0] *#String format*  
dTime = datetime.strptime(sTime, **'%H:%M:%S'**) *#Time format*  
temp = val[1].replace(**"C"**, **""**)  
 hum = val[2].replace(**"%"**, **""**)  
 self.storeData(self.data\_temp,temp,self.data\_temp\_time,self.data\_temp\_collectTime,sTime,dTime)  
 self.storeData(self.data\_hum,hum, self.data\_hum\_time, self.data\_hum\_collectTime, sTime, dTime)  
 **elif** type == **"PM25"**:  
 val = val.replace(**"Time="**, **""**)  
 val = val.replace(**"PM25="**, **""**)  
 val = val.replace(**"PM10="**, **""**)  
 val = val.split(**" "**)  
 sTime = val[0] *#String format*  
dTime = datetime.strptime(sTime, **'%H:%M:%S'**) *#Time format*  
pm25 = val[1]  
 pm10 = val[2]  
 self.storeData(self.data\_pm25,pm25,self.data\_pm25\_time,self.data\_pm25\_collectTime,sTime,dTime)  
 self.storeData(self.data\_pm10,pm10,self.data\_pm10\_time,self.data\_pm10\_collectTime,sTime,dTime)  
 self.draw()  
  
  
 **def** storeData(self, target, data,target\_time,target\_collectTime, sTime,dTime):  
 print(data)  
 target.append(float(data))  
  
 **if** (len(target\_time) != 0):  
 lTime = datetime.strptime(target\_collectTime[-1], **'%H:%M:%S'**) *#lastest collect time*  
timeDiff = dTime-lTime  
 target\_time.append(target\_time[-1] + int(str(timeDiff.seconds)))  
 target\_collectTime.append(sTime)  
 **else**:  
 print(**'first'**)  
 target\_time.append(0)  
 target\_collectTime.append(sTime)  
  
 **def** draw(self):  
 *# print("draw")*  
self.canvas\_pm25.update\_figure(self.data\_pm25\_time,self.data\_pm25)  
 self.canvas\_pm10.update\_figure(self.data\_pm10\_time, self.data\_pm10)  
 self.canvas\_temp.update\_figure(self.data\_temp\_time, self.data\_temp)  
 self.canvas\_hum.update\_figure(self.data\_hum\_time, self.data\_hum)  
  
 **def** saveAs(self):  
 path = QFileDialog.getExistingDirectory(self, **'Choose Directory'**)  
 directory = time.strftime(**'%d-%m-%Y'**) + **' '** + time.strftime(**'%H-%M-%S'**)  
  
 **if** path != **""**:  
 path = path + **"/"** + directory  
 os.mkdir(path)  
 record = open(path + **"/"** + **'raw\_data.txt'**, **'a+'**)  
 output\_temp = **""**  
output\_hum = **""**  
output\_pm25 = **""**  
output\_pm10 = **""**  
output\_temp\_collectTime = **""**  
output\_hum\_collectTime = **""**  
output\_pm25\_collectTime = **""**  
output\_pm10\_collectTime = **""**  
  
 **if** len(self.data\_temp) != 0:  
 output\_temp = [**"%.1f"** % number **for** number **in** self.data\_temp]  
 output\_temp = **','**.join(output\_temp)  
 output\_temp\_collectTime = **','**.join(self.data\_temp\_collectTime)  
 **if** len(self.data\_hum) != 0:  
 output\_hum = [**"%.1f"** % number **for** number **in** self.data\_hum]  
 output\_hum = **','**.join(output\_hum)  
 output\_hum\_collectTime = **','**.join(self.data\_hum\_collectTime)  
 **if** len(self.data\_pm25) != 0:  
 output\_pm25 = [**"%.1f"** % number **for** number **in** self.data\_pm25]  
 output\_pm25 = **','**.join(output\_pm25)  
 output\_pm25\_collectTime = **','**.join(self.data\_pm25\_collectTime)  
 **if** len(self.data\_pm10) != 0:  
 output\_pm10 = [**"%.1f"** % number **for** number **in** self.data\_pm10]  
 output\_pm10 = **','**.join(output\_pm10)  
 output\_pm10\_collectTime = **','**.join(self.data\_pm10\_collectTime)  
  
  
 output = **"{\n"** \  
 **"\"Temperature\":{"** \  
 **"\n\t\"Data\":["** + output\_temp + **"],\n\t\"Unit\":C,"** \  
 **"\n\t\"CollectedTime\":["**+output\_temp\_collectTime+**"]\n\t}"**\  
 **"\n\"Humidity\":{"**\  
 **"\n\t\"Data\":["** + output\_hum + **"],\n\t\"Unit\":%,"**\  
 **"\n\t\"CollectedTime\":["**+output\_hum\_collectTime+**"]\n\t}"** \  
 **"\n\"PM2.5\":{"** \  
 **"\n\t\"Data\":["** + output\_pm25 + **"],\n\t\"Unit\":µg/m³,"** \  
 **"\n\t\"CollectedTime\":["** + output\_pm25\_collectTime + **"]\n\t}"** \  
 **"\n\"PM10\":{"** \  
 **"\n\t\"Data\":["** + output\_pm10 + **"],\n\t\"Unit\":µg/m³,"** \  
 **"\n\t\"CollectedTime\":["** + output\_pm10\_collectTime + **"]\n\t}"** \  
 **"\n}"**  
record.write(output)  
 self.canvas\_temp.outputImage(path + **"/"** + **"Temperature"**)  
 self.canvas\_hum.outputImage(path + **"/"** + **"Humidity"**)  
 self.canvas\_pm25.outputImage(path + **"/"** + **"PM2.5"**)  
 self.canvas\_pm10.outputImage(path + **"/"** + **"PM10"**)  
  
 **def** newMission(self):  
 msgBox = QMessageBox()  
 msgBox.setText(**"Do you want to create a new mission now? If yes, then the current mission will be closed."**)  
 msgBox.setStandardButtons(QMessageBox.Cancel | QMessageBox.Ok)  
 option = msgBox.exec\_()  
 **if** option == QMessageBox.Ok:  
 *# os.execl(sys.executable, 'python', \_\_file\_\_, \*sys.argv[1:])*  
os.execl(sys.executable, sys.executable, \*sys.argv)  
 *# os.execl(sys.executable, '"{}"'.format(sys.executable), \*sys.argv)*  
  
**def** videoPipeline(self):  
 self.pipeline = Gst.Pipeline()  
 self.tcpsrc = Gst.ElementFactory.make(**'tcpclientsrc'**, **'tcpsrc'**)  
 self.tcpsrc.set\_property(**"host"**, **'192.168.12.1'**)  
 self.tcpsrc.set\_property(**"port"**, 5000)  
  
 self.gdepay = Gst.ElementFactory.make(**'gdpdepay'**, **'gdepay'**)  
 self.rdepay = Gst.ElementFactory.make(**'rtph264depay'**, **'rdepay'**)  
 self.avdec = Gst.ElementFactory.make(**'avdec\_h264'**, **'avdec'**)  
 self.vidconvert = Gst.ElementFactory.make(**'videoconvert'**, **'vidconvert'**)  
 self.asink = Gst.ElementFactory.make(**'autovideosink'**, **'asink'**)  
 self.asink.set\_property(**'sync'**, **False**)  
  
 self.pipeline.add(self.tcpsrc)  
 self.pipeline.add(self.gdepay)  
 self.pipeline.add(self.avdec)  
 self.pipeline.add(self.rdepay)  
 self.pipeline.add(self.vidconvert)  
 self.pipeline.add(self.asink)  
  
 self.tcpsrc.link(self.gdepay)  
 self.gdepay.link(self.rdepay)  
 self.rdepay.link(self.avdec)  
 self.avdec.link(self.vidconvert)  
 self.vidconvert.link(self.asink)  
  
 **def** on\_sync\_message(self, bus, message):  
 **if** message.get\_structure().get\_name() == **'prepare-window-handle'**:  
 message.src.set\_property(**'force-aspect-ratio'**, **True**)  
 message.src.set\_window\_handle(self.winId)  
  
 **def** quit(self, container):  
 self.pipeline.set\_state(Gst.State.NULL)  
 Gtk.main\_quit()  
  
 **def** on\_eos(self, bus, msg):  
 print(**'on\_eos(): seeking to start of video'**)  
 self.pipeline.seek\_simple(  
 Gst.Format.TIME,  
 Gst.SeekFlags.FLUSH | Gst.SeekFlags.KEY\_UNIT,  
 0  
 )  
  
 **def** on\_error(self, bus, msg):  
 print(**'on\_error():'**, msg.parse\_error())  
  
 **def** start(self):  
 self.pipeline.set\_state(Gst.State.PLAYING)  
  
 **def** updateQFI(self, detail):  
 **if**(detail[**"airspeed"**] != **""**):  
 self.adi.setRoll(math.degrees(detail[**"attitude\_roll"**]))  
 self.adi.setPitch(math.degrees(detail[**"attitude\_pitch"**]))  
 self.alt.setAltitude(detail[**"altitude"**])  
 self.si.setSpeed(detail[**"airspeed"**])  
 self.hsi.setHeading(detail[**"heading"**])  
 self.vsi.setClimbRate(detail[**"verticalSpeed"**])  
  
 self.adi.viewUpdate.emit()  
 self.alt.viewUpdate.emit()  
 self.si.viewUpdate.emit()  
 self.hsi.viewUpdate.emit()  
 self.vsi.viewUpdate.emit()  
  
 **def** updateMap(self, detail):  
  
 *# Working with the maps with pyqtlet*  
**if** (detail[**"location\_lat"**] != **""**):  
 self.map.removeLayer(self.marker)  
 **if** self.count == 0:  
 self.map.setView([detail[**"location\_lat"**], detail[**"location\_lon"**]], 20)  
 self.count += 1  
 **if** self.count % 10 == 0:  
 self.map.setView([detail[**"location\_lat"**], detail[**"location\_lon"**]], 20)  
 self.marker = L.marker([detail[**"location\_lat"**], detail[**"location\_lon"**]])  
 self.marker.bindPopup(**'UAV Here'**)  
 self.map.addLayer(self.marker)  
  
  
  
 **def** retranslateUi(self, MainWindow):  
 \_translate = QtCore.QCoreApplication.translate  
 MainWindow.setWindowTitle(\_translate(**"MainWindow"**, **"Automated Data Collecting System"**))  
 self.menuMission.setTitle(\_translate(**"MainWindow"**, **"Mission"**))  
 self.menuConnection.setTitle(\_translate(**"MainWindow"**, **"Connection"**))  
 self.actionNew\_Mission.setText(\_translate(**"MainWindow"**, **"New Mission"**))  
 self.actionView\_Mission.setText(\_translate(**"MainWindow"**, **"View Mission"**))  
 self.actionSave.setText(\_translate(**"MainWindow"**, **"Save As"**))  
 self.actionClose.setText(\_translate(**"MainWindow"**, **"Quit"**))  
 self.actionConnect.setText(\_translate(**"MainWindow"**, **"Connect"**))  
 self.actionDisconnect.setText(\_translate(**"MainWindow"**, **"Disconnect"**))  
  
 **def** connect(self):  
 self.drone = Drone(**'udp:0.0.0.0:14550'**)  
 **if** self.drone.isconnect == **True**:  
 self.vehicle = self.drone.getDrone()  
 self.actionConnect.setDisabled(**True**)  
 self.actionDisconnect.setDisabled(**False**)  
  
 *#MQtt start*  
self.client = MqttClient(self)  
 self.client.stateChanged.connect(self.on\_stateChanged)  
 self.client.messageSignal.connect(self.on\_messageSignal)  
 self.client.hostname = **"192.168.12.1"**  
self.client.connectToHost()  
  
 *#start Map and QFI*  
self.updateMap\_thread.setVehicle(self.vehicle)  
 self.updateQFI\_thread.setVehicle(self.vehicle)  
 self.start()  
  
 *# Animation thread*  
t2 = ThreadGUI(self.gridLayout)  
 t2.daemon = **True**  
t2.start()  
 **else**:  
 self.disconnect()  
  
  
 **def** disconnect(self):  
 self.drone.disconnectDrone()  
 self.actionConnect.setDisabled(**False**)  
 self.actionDisconnect.setDisabled(**True**)  
 self.updateMap\_thread.exist = **False**  
self.updateQFI\_thread.exist = **False**  
self.quit(self.container)  
  
 self.adi.setRoll(0)  
 self.adi.setPitch(0)  
 self.alt.setAltitude(0)  
 self.si.setSpeed(0)  
 self.hsi.setHeading(0)  
 self.vsi.setClimbRate(0)  
  
 self.adi.viewUpdate.emit()  
 self.alt.viewUpdate.emit()  
 self.si.viewUpdate.emit()  
 self.hsi.viewUpdate.emit()  
 self.vsi.viewUpdate.emit()  
  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 app = QApplication(sys.argv)  
 MainWindow = QMainWindow()  
 ui = Monitor()  
 ui.setupUi(MainWindow)  
 MainWindow.show()  
 sys.exit(app.exec\_())

Drone.py

**from** PyQt5.QtWidgets **import** QMessageBox  
from dronekit **import** connect, VehicleMode  
class Drone:  
 isconnect = **False**  
 **def** \_\_init\_\_(self,connection\_string):  
 self.connection\_string=connection\_string  
 **try**:  
 print(**"connect"**)  
 self.drone = connect(self.connection\_string, wait\_ready=**True**)  
 print(**"connected"**)  
 self.isconnect = **True**  
 **except**:  
 msgBox = QMessageBox()  
 msgBox.setText(**"The connection to the drone timed out."**)  
 msgBox.exec\_()  
 *#self.sitl = dronekit\_sitl.start\_default()*  
 *#self.drone = connect('tcp:127.0.0.1:5760', wait\_ready=True)*  
  
**def** connectDrone(self):  
 self.drone = connect(self.connection\_string, wait\_ready=**True**)  
  
 **def** getDrone(self):  
 **return** self.drone  
 **def** disconnectDrone(self):  
 **if** self.isconnect == **True**:  
 self.drone.close()  
 *#self.sitl.stop()*

*MqttClient.py*

**from** PyQt5 **import** QtCore, QtWidgets  
import paho.mqtt.client **as** mqtt  
  
  
class MqttClient(QtCore.QObject):  
 Disconnected = 0  
 Connecting = 1  
 Connected = 2  
  
 MQTT\_3\_1 = mqtt.MQTTv31  
 MQTT\_3\_1\_1 = mqtt.MQTTv311  
  
 connected = QtCore.pyqtSignal()  
 disconnected = QtCore.pyqtSignal()  
  
 stateChanged = QtCore.pyqtSignal(int)  
 hostnameChanged = QtCore.pyqtSignal(str)  
 portChanged = QtCore.pyqtSignal(int)  
 keepAliveChanged = QtCore.pyqtSignal(int)  
 cleanSessionChanged = QtCore.pyqtSignal(bool)  
 protocolVersionChanged = QtCore.pyqtSignal(int)  
  
 messageSignal = QtCore.pyqtSignal(str)  
  
 **def** \_\_init\_\_(self, parent=**None**):  
 super(MqttClient, self).\_\_init\_\_(parent)  
  
 self.m\_hostname = **""**  
self.m\_port = 1883  
 self.m\_keepAlive = 60  
 self.m\_cleanSession = **True**  
self.m\_protocolVersion = MqttClient.MQTT\_3\_1  
  
 self.m\_state = MqttClient.Disconnected  
  
 self.m\_client = mqtt.Client(clean\_session=self.m\_cleanSession,  
 protocol=self.protocolVersion)  
  
 self.m\_client.on\_connect = self.on\_connect  
 self.m\_client.on\_message = self.on\_message  
 self.m\_client.on\_disconnect = self.on\_disconnect  
  
  
 @QtCore.pyqtProperty(int, notify=stateChanged)  
 **def** state(self):  
 **return** self.m\_state  
  
 @state.setter  
 **def** state(self, state):  
 **if** self.m\_state == state: **return**  
self.m\_state = state  
 self.stateChanged.emit(state)  
  
 @QtCore.pyqtProperty(str, notify=hostnameChanged)  
 **def** hostname(self):  
 **return** self.m\_hostname  
  
 @hostname.setter  
 **def** hostname(self, hostname):  
 **if** self.m\_hostname == hostname: **return**  
self.m\_hostname = hostname  
 self.hostnameChanged.emit(hostname)  
  
 @QtCore.pyqtProperty(int, notify=portChanged)  
 **def** port(self):  
 **return** self.m\_port  
  
 @port.setter  
 **def** port(self, port):  
 **if** self.m\_port == port: **return**  
self.m\_port = port  
 self.portChanged.emit(port)  
  
 @QtCore.pyqtProperty(int, notify=keepAliveChanged)  
 **def** keepAlive(self):  
 **return** self.m\_keepAlive  
  
 @keepAlive.setter  
 **def** keepAlive(self, keepAlive):  
 **if** self.m\_keepAlive == keepAlive: **return**  
self.m\_keepAlive = keepAlive  
 self.keepAliveChanged.emit(keepAlive)  
  
 @QtCore.pyqtProperty(bool, notify=cleanSessionChanged)  
 **def** cleanSession(self):  
 **return** self.m\_cleanSession  
  
 @cleanSession.setter  
 **def** cleanSession(self, cleanSession):  
 **if** self.m\_cleanSession == cleanSession: **return**  
self.m\_cleanSession = cleanSession  
 self.cleanSessionChanged.emit(cleanSession)  
  
 @QtCore.pyqtProperty(int, notify=protocolVersionChanged)  
 **def** protocolVersion(self):  
 **return** self.m\_protocolVersion  
  
 @protocolVersion.setter  
 **def** protocolVersion(self, protocolVersion):  
 **if** self.m\_protocolVersion == protocolVersion: **return**  
 **if** protocolVersion **in** (MqttClient.MQTT\_3\_1, MQTT\_3\_1\_1):  
 self.m\_protocolVersion = protocolVersion  
 self.protocolVersionChanged.emit(protocolVersion)  
  
 *#################################################################*  
@QtCore.pyqtSlot()  
 **def** connectToHost(self):  
 **if** self.m\_hostname:  
 self.m\_client.connect(self.m\_hostname,  
 port=self.port,  
 keepalive=self.keepAlive)  
  
 self.state = MqttClient.Connecting  
 self.m\_client.loop\_start()  
  
 @QtCore.pyqtSlot()  
 **def** disconnectFromHost(self):  
 self.m\_client.disconnect()  
  
 **def** subscribe(self, path):  
 **if** self.state == MqttClient.Connected:  
 self.m\_client.subscribe(path)  
  
 *#################################################################*  
 *# callbacks*  
**def** on\_message(self, mqttc, obj, msg):  
 mstr = msg.payload.decode(**"ascii"**)  
 *# print("on\_message", mstr, obj, mqttc)*  
self.messageSignal.emit(mstr)  
  
 **def** on\_connect(self, \*args):  
 *# print("on\_connect", args)*  
self.state = MqttClient.Connected  
 self.connected.emit()  
  
 **def** on\_disconnect(self, \*args):  
 *# print("on\_disconnect", args)*  
self.state = MqttClient.Disconnected  
 self.disconnected.emit()  
  
  
class Widget(QtWidgets.QWidget):  
 **def** \_\_init\_\_(self, parent=**None**):  
 super(Widget, self).\_\_init\_\_(parent)  
  
 lay = QtWidgets.QVBoxLayout(self)  
 self.lcd\_number = QtWidgets.QLabel()  
 lay.addWidget(self.lcd\_number)  
  
 self.client = MqttClient(self)  
 self.client.stateChanged.connect(self.on\_stateChanged)  
 self.client.messageSignal.connect(self.on\_messageSignal)  
  
 self.client.hostname = **"192.168.12.1"**  
self.client.connectToHost()  
  
 @QtCore.pyqtSlot(int)  
 **def** on\_stateChanged(self, state):  
 **if** state == MqttClient.Connected:  
 print(state)  
 self.client.subscribe([(**"/IoTSensor/DHT22"**,0),(**"/IoTSensor/SDS011"**,1)])  
  
 @QtCore.pyqtSlot(str)  
 **def** on\_messageSignal(self, msg):  
 val = msg  
 print(val)  
 self.lcd\_number.setText(val)  
  
plotCanvas.py

**from** PyQt5.QtWidgets **import** QSizePolicy  
  
from matplotlib.backends.backend\_qt5agg **import** FigureCanvasQTAgg **as** FigureCanvas  
from matplotlib.figure **import** Figure  
from PyQt5.QtCore **import** QTimer  
  
  
class PlotCanvas(FigureCanvas):  
  
 **def** \_\_init\_\_(self, parent=**None**, width=1, height=1, dpi=100):  
 self.fig = Figure(figsize=(width, height), dpi=dpi)  
 self.axes = self.fig.add\_subplot(111)  
  
 FigureCanvas.\_\_init\_\_(self, self.fig)  
 self.setParent(parent)  
  
 FigureCanvas.setSizePolicy(self,  
 QSizePolicy.Expanding,  
 QSizePolicy.Expanding)  
 FigureCanvas.updateGeometry(self)  
  
 **def** plot(self):  
 timer = QTimer(self)  
 timer.timeout.connect(self.update\_figure)  
 timer.start(1000)  
  
 **def** init\_plot(self, title, unit, time):  
 self.title = title  
 self.unit = unit  
 self.time = time  
 self.axes.set\_title(self.title)  
 self.axes.set\_ylabel(self.unit)  
 self.axes.set\_xlabel(self.time)  
  
 **def** update\_figure(self, x, y):  
 self.axes.cla()  
 self.axes.set\_title(self.title)  
 self.axes.set\_ylabel(self.unit)  
 self.axes.set\_xlabel(self.time)  
 self.axes.plot(x, y)  
 self.draw()  
  
 **def** outputImage(self, title):  
 self.fig.savefig(title + **'.png'**)

vehicleLocal.py

**from** PyQt5.QtCore **import** pyqtSignal, QObject  
import time  
  
  
class VehicleLocation(QObject):  
 updateMap = pyqtSignal(object)  
 exist = **False**  
  
 **def** run(self):  
 **while True**:  
 **if** (self.exist):  
 detail = {**"location\_lat"**: self.vehicle.location.global\_frame.lat,  
 **"location\_lon"**: self.vehicle.location.global\_frame.lon}  
 **else**:  
 detail = {**"location\_lat"**: **""**, **"location\_lon"**: **""**}  
 self.updateMap.emit(detail)  
 time.sleep(5)  
  
 **def** setVehicle(self, vehicle):  
 self.vehicle = vehicle  
 self.exist = **True**

VehicleStatus.py

**from** PyQt5.QtCore **import** pyqtSignal, QObject  
import time  
  
  
class VehicleStatus(QObject):  
 updateQFI = pyqtSignal(object)  
 exist = **False**  
  
 **def** run(self):  
 **while True**:  
 **if** (self.exist):  
 *# split attitude*  
attitude = str(self.vehicle.attitude).split(**':'**)  
 attitude = attitude[1].split(**'='**)  
 attitude\_pitch = float(attitude[1].split(**','**)[0])  
 attitude\_yaw = float(attitude[2].split(**','**)[0])  
 attitude\_roll = float(attitude[3].split(**','**)[0])  
 detail = {**"airspeed"**: self.vehicle.airspeed,  
 **"attitude\_pitch"**: attitude\_pitch,  
 **"attitude\_yaw"**: attitude\_yaw,  
 **"attitude\_roll"**: attitude\_roll,  
 **"altitude"**: self.vehicle.location.global\_relative\_frame.alt,  
 **"groundspeed"**: format(float(self.vehicle.groundspeed), **'0.3f'**),  
 **"heading"**: self.vehicle.heading,  
 **"verticalSpeed"**: self.vehicle.velocity[2]}  
 **else**:  
 detail = {**"airspeed"**: **""**, **"attitude\_pitch"**: **""**, **"attitude\_yaw"**: **""**, **"attitude\_roll"**: **""**, **"altitude"**: **""**,  
 **"groundspeed"**: **""**, **"heading"**: **""**, **"verticalSpeed"**: **""**}  
  
 self.updateQFI.emit(detail)  
  
 time.sleep(0.2)  
  
 **def** setVehicle(self, vehicle):  
 self.vehicle = vehicle  
 self.exist = **True**