Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2022

Top Drone: Mavericks
Raytheon Drone Project

Jaedyn Brown
Robert Carr
Javier Lopez
Ja'lun Morris
Pearl Iyayi

Revision History

Revision	Date	Author(s)	Description
0.1	11.13.2022	JB, RC, JM, JL, PI	document creation

Contents

List of Figures

List of Tables

1 INTRODUCTION

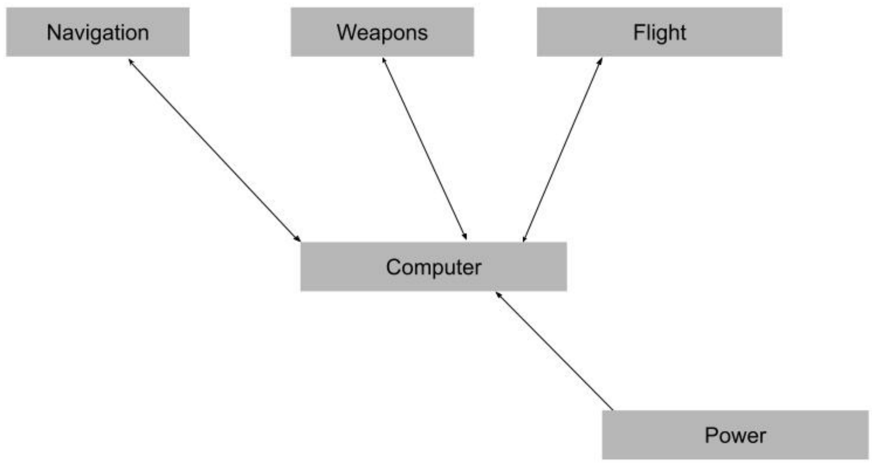
This document's purpose is to describe how the drone's system will be developed. The system is broken down into layers and smaller subsystems that interact with each other. This document will also demonstrate the flow of data shared between each layer and subsystem. The requirements in the statement of work provided by the sponsor and the System Requirement Specification document help define each layer. The main requirements of the drone are that it must fly autonomously and it must identify enemy and friendly ground vehicles. The drone will fly over an area where ground vehicles will be roaming. When an enemy vehicle is identified, the drone must fire its weapons at the target to disable it. If a friendly vehicle is identified, the drone must not take any action. The drone will determine the hostility of a vehicle using an ArUco Marker placed on each ground vehicle. It should be able to eliminate all enemy vehicles while leaving friendly vehicles unharmed. In the case of an emergency, the drone must have a "kill switch" that allows a pilot to manually override the drone.

Two crucial parts of the drone autonomy are its artificial intelligence and computer vision software. The software that will be used for autonomous flight will be on a computer that will function as the head of the system – meaning all layers will be connected to the computer layer. In this system, the computer is responsible for managing all data transferred between layers. From the computer layer stems out four other main components: navigation, weapons, flight, power (see diagram in section 2).

- The navigation layer will contain components such as GPS and RTK that will retrieve information for steering the drone.
- The weapons layer will contain the A.C.E. Combat System that is required by the statement of work. The weapons will be used to fire at hostile vehicles.
- The flight layer contains the flight controller and motors – it will allow the drone to generate lift and fly stably.
- The power layer will consist of the battery and any power modules required for components. It's job is simply to provide power to the rest of the system.

Each of these layers are essential and are needed for the drone to operate properly. They will work together to complete each objective given in the statement of work.

2 System Overview



2.1 Computer

The Computer Layer houses all features or functions that are processed on the computer. Whether the computer is located on board or connected wirelessly, the responsibility of the computer layer remains the same. The computer layer shares relationships with the rest of the layers as all data is communicated to and from the computer.

2.2 Navigation

The Navigation Layer houses all features or functions responsible for the drone's ability to perform paths around the workspace. Being that this layer is responsible for the drone's ability to perform it's autonomous movements, the relationship it has with the computer is bidirectional. The reason is information is relayed from the computer to the modules located within the Navigation layer, and the information that is produced in the Navigation layer is also relayed back to the computer.

2.3 Weapons

The Weapons Layer houses all features or functions responsible for the drone's ability to locate targets and fire upon them with the laser. This layer has a bidirectional relationship with the Computer Layer as the computer layer is running processes that help the weapons layer to identify targets, and the weapons layer is relaying data back to the Computer Layer.

2.4 Flight

The Flight Layer houses all features or functions responsible for the drone's ability to fly. This layer focuses on the physical components that are responsible for the drone taking flight. The Flight Layer has a bidirectional relationship with the Computer Layer as it has internal sensors on the drone that sends data to the Computer Layer. Equally the Computer Layer sends data back to the Flight layer to communicate how the drone's parts should be operating to make specific paths.

2.5 Power

The Power Layer houses all features or functions responsible for the system's ability to draw power. This layer is solely responsible for the system's power source, and has a unidirectional relationship pointing at the Computer Layer as the system is powered, but there is no data flow from the computer back to the battery.

3 Subsystem Definitions & Data Flow

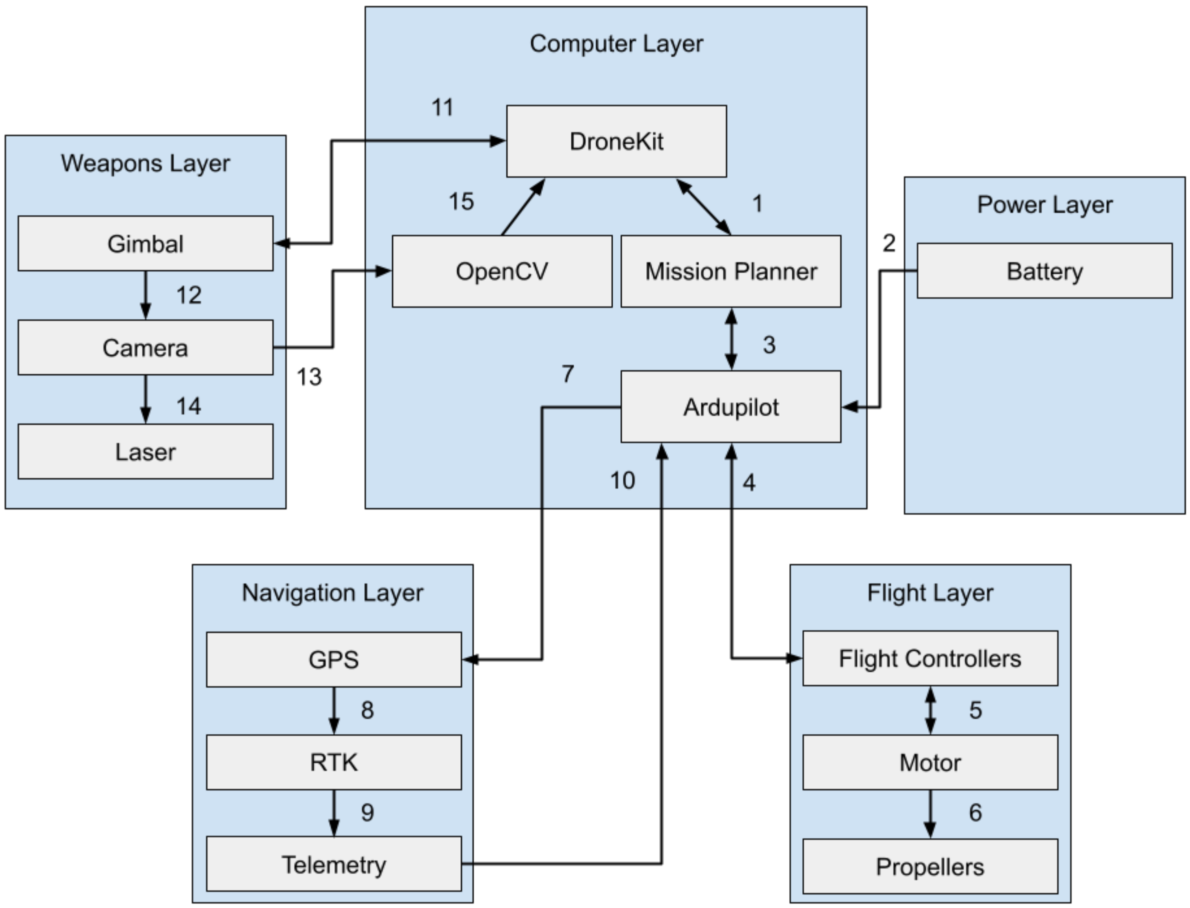


Figure 1: Subsystem Definitions & Data Flow diagram

4 Computer Layer Subsystems

4.1 DroneKit Subsystem

In addition to allowing developers to create applications that run on the in-flight companion computer, DroneKit (an open source software) will provide a low-latency connection between the flight controller and the onboard computer.

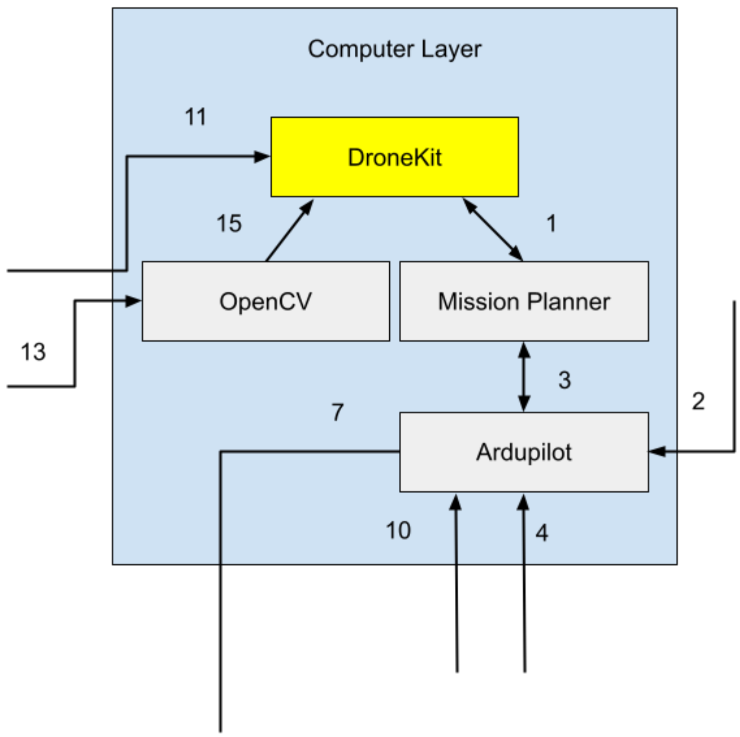


Figure 2: DroneKit Subsystem diagram

4.1.1 Assumptions

- DroneKit will run on our ground station.

4.1.2 Responsibilities

The drone will be able to communicate with DroneKit via MAVLink. In addition to giving the development team access to the drone's information (telemetry, parameter information), DroneKit will also allow them to control vehicle movements and operations directly.

4.1.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	Programmatic communication between DroneKit and Mission Planner	Mission Planner data	Mission Planner commands
#15	Receiving image recognition data from OpenCV	OpenCV data	Mission N/A

Table 2: DroneKit Subsystem interfaces

4.2 Mission Planner Subsystem

As the drone becomes autonomous, Mission Planner will play an increasingly significant role. Mission Planner will serve as a ground control station.

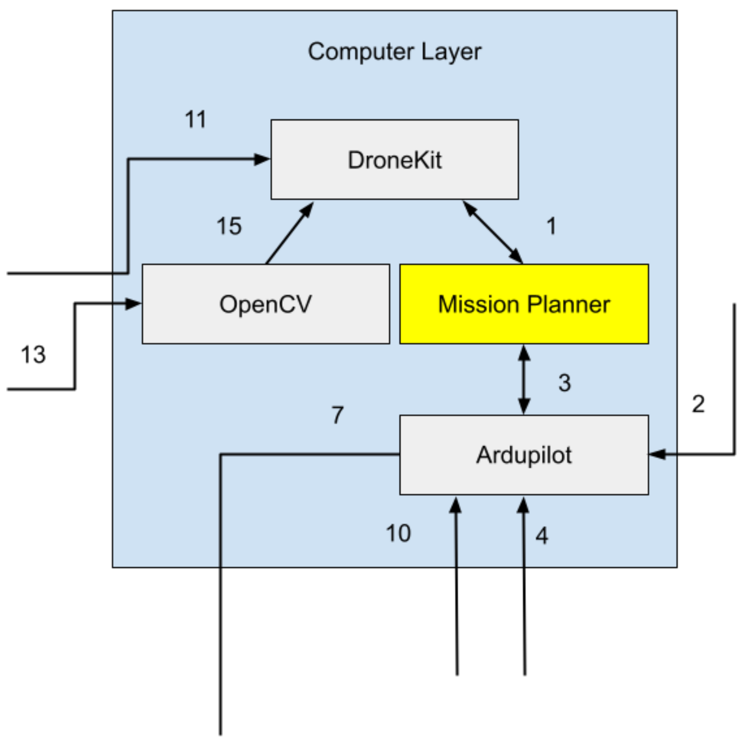


Figure 3: Mission Planner Subsystem diagram

4.2.1 Assumptions

- The connection to the drone will be stable enough to get DroneKit commands to the drone.

4.2.2 Responsibilities

It will be Mission Planner's responsibility to act as a ground station for the drone. Through Mission Planner, the development team can configure and calibrate the drone as well as set autonomous tasks and waypoints.

4.2.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	Programmatic communication between DroneKit and Mission Planner	Mission Planner data	Mission Planner commands
#15	Receiving image recognition data from OpenCV	OpenCV data	N/A

Table 3: Mission Planner Subsystem interfaces

4.3 Ardupilot Subsystem

Ardupilot is an open source software for controlling autonomous drones.

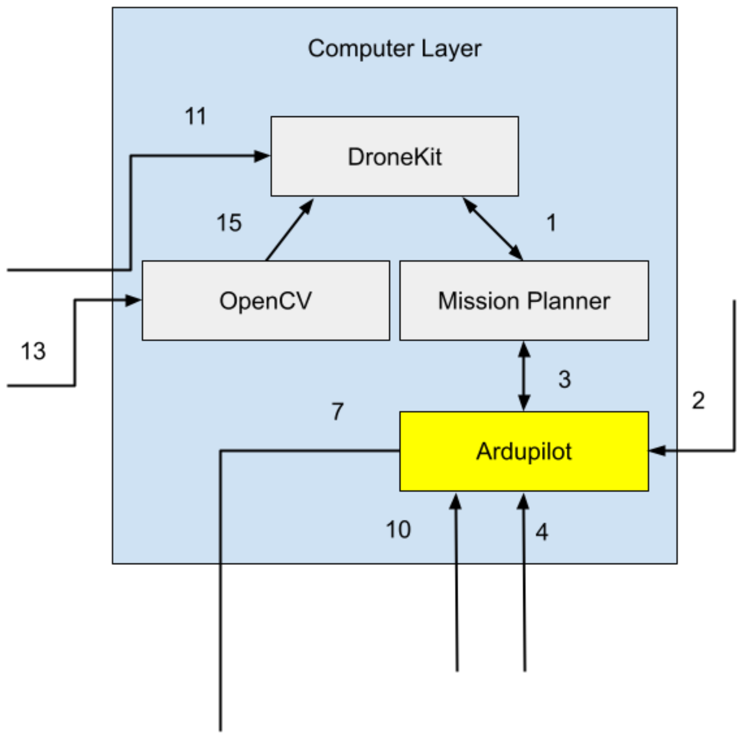


Figure 4: Ardupilot Subsystem diagram

4.3.1 Assumptions

- Ardupilot will have a stable enough signal to receive Mission Planner and DroneKit commands or information.

4.3.2 Responsibilities

In conjunction with ground station software such as Mission Planner, Ardupilot will be responsible for controlling the autonomous flight of the drone. A number of features are provided by Arduinopilot that the team will utilize when flying the drone, including fully autonomous, semi-autonomous, and manual flight modes as well as programmable missions with waypoints, geofencing, drone stabilization, flight simulation, navigation sensor support (including GPS and RTK), sensor communication, failsafes, brushless motor support, and video gimbal support.

4.3.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#02	Power from the battery to the drone	Power	N/A
#03	Programmatic communication between Ardupilot and Mission Planner	Mission Planner commands	Ardupilot data
#04	Programmatic communication between Ardupilot and the flight controllers	Flight controller data	Ardupilot command
#07	Send data to GPS via MAVLink	N/A	MAVLink data or commands
#10	Obtain telemetry data from Navigation subsystem	Telemetry data	N/A

Table 4: Ardupilot Subsystem interfaces

4.4 OpenCV Subsystem

An open-source library called OpenCV will be used to perform the image recognition part of the project. With this library, the drone can easily identify ArUco markers moving in conjunction with hostile ground vehicles. Data from the camera will be analyzed by OpenCV to provide results for image recognition.

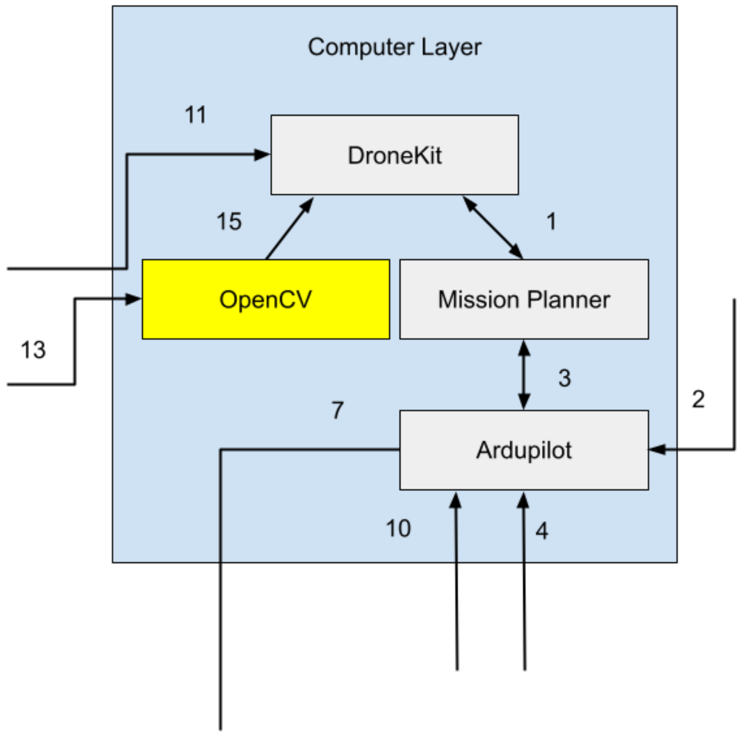


Figure 5: OpenCV Subsystem diagram

4.4.1 Assumptions

- OpenCV will be able to run continuously on the drone without any major issues.

4.4.2 Responsibilities

It will be OpenCV's responsibility to process the camera data and determine whether any ArUco markers can be seen. In the event that OpenCV does identify any ArUco markers, the ID of each marker must be obtained in order to establish whether the marker represents a hostile vehicle. For the drone to determine the next steps based on the data provided by OpenCV, the results will have to be communicated to the computer. Other software will be able to make decisions based on the data OpenCV generates.

4.4.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#13	Camera information is given to OpenCV	Camera data	N/A
#15	OpenCV image recognition data is given to DroneKit	N/A	OpenCV results

Table 5: OpenCV Subsystem interfaces

5 FLIGHT SUBSYSTEM

The flight aspect will be done using a mission planner to create paths for the drone to follow to complete certain tasks. While also, the flight system will be assisted with a cube orange that will provide flight data to the drone.

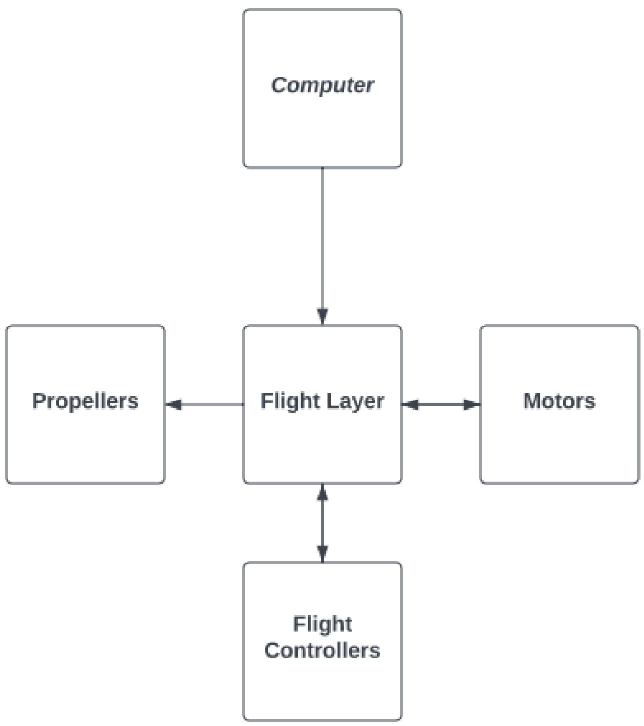


Figure 6: A simple architectural layer diagram

5.1 Motors

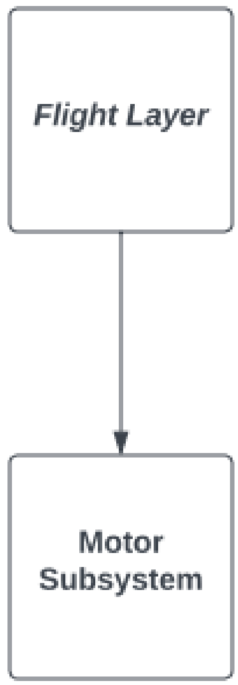


Figure 7: Motor subsystem diagram

There will be four motors that will be mounted onto each arm of the drone.

5.1.1 Assumptions

The motors will be light and powerful to lift and fly the drone. The motors will be connected to electronic speed controllers, which will in turn be connected to the dedicated flight controller.

5.1.2 Responsibilities

The motors will have the responsibilities of being able to generate enough lift for the drone to be able to fly.

5.1.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	Electronic Speed controller	Motor Control signals	Flight lift

Table 6: Motor Subsystem interfaces

5.2 PROPELLER

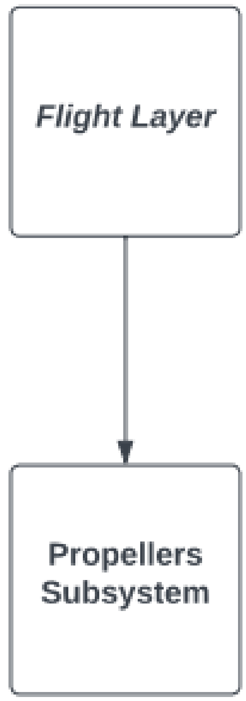


Figure 8: Propeller subsystem diagram

There will be four propellers. There will be two clockwise turn propellers, in addition to two counter clockwise turn propellers. These propellers must be opposite of each other.

5.2.1 Assumptions

The propellers must be strong enough to withstand strong wind. When flying each propeller must be attached at all times with a propeller guard.

5.2.2 Responsibilities

It is the propeller's responsibility to provide lift, by spinning and creating airflow. The rotation also keeps the drone stable and propels it to move forwards.

5.2.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	Propeller Guard	N/A	N/A
#02	3D printer	Object file	3D object

Table 7: Propeller Subsystem interfaces

5.3 Flight Controllers

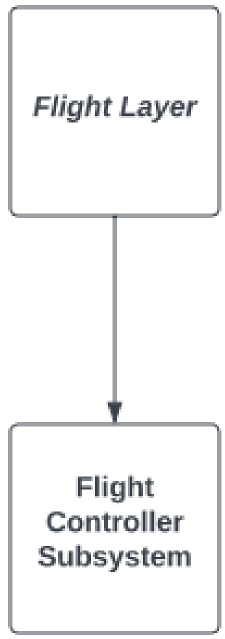


Figure 9: Flight Controller subsystem diagram

The flight controller would be connected to several components such as the GPS and the RTK.

5.3.1 Assumptions

The flight controller would be mounted to the drone to control its motor speeds.

5.3.2 Responsibilities

It is the flight controllers responsibility to provide flight data to the drone, to help assist the drone with autonomous flight.

5.3.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	GPS/RTK	Data from satellites	Position of the Drone
#02	SBC	Data from laptop	Computation to drone

Table 8: Propeller Subsystem interfaces

6 WEAPON SYSTEM

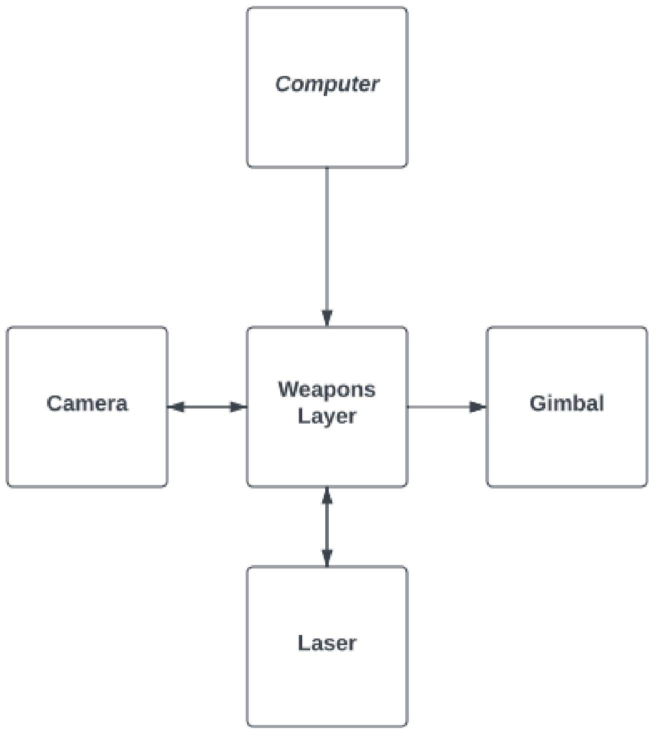


Figure 10: Weapons Subsystem diagram

The weapons subsystem consists of a camera, laser, and gimbal. These components must work together to search for targets and disable the target.

6.1 CAMERA

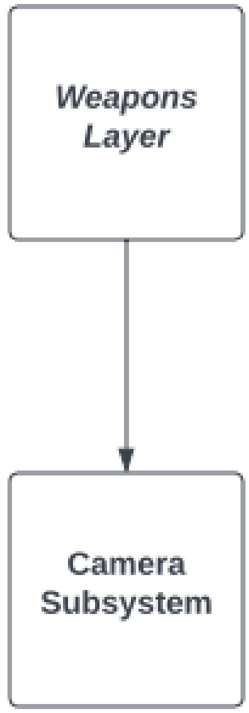


Figure 11: Camera Subsystem diagram

There will be a camera mounted onto the drone, that will be connected to a receiver that will be collecting data from the camera.

6.1.1 Assumptions

The camera would be able to detect ArUco markers. The camera would be able to turn and locate aim at enemy vehicles.

6.1.2 Responsibilities

The responsibilities for the camera would be to search for an QR code on top of a ground vehicle, while using an OpenVC algorithm

6.1.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	OpenCV algorithm	Python library from Python IDE	Image Processing

6.2 LASER

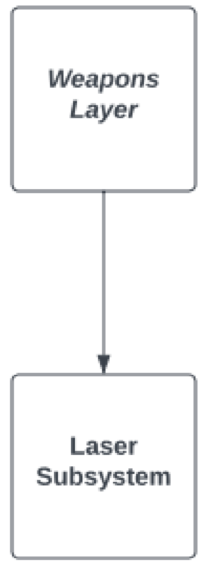


Figure 12: Laser Subsystem diagram

The laser would be attached to the camera, the closer the laser is to the camera the more accurate the laser would be.

6.2.1 Assumptions

When the drone shoots the laser an LED light will show, and also the drone will play a buzzer sound to indicate the laser has been fired.

6.2.2 Responsibilities

The responsibilities of the laser would be to shoot a beam at a ground vehicle once located. Using a software, it would output each time the drone was fired tracking its timestamps.

6.2.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	Timestamps	Python code from Python IDE	A timestamp would print out, every time laser shot.
#01	Sound	Every time laser is triggered	Buzzer sound

6.3 Gimbal

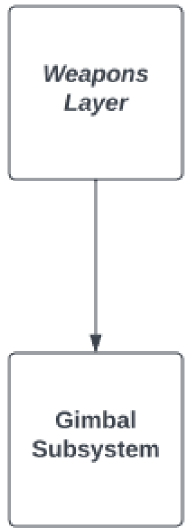


Figure 13: Gimbal Subsystem diagram

The Gimbal component helps stabilize and support the camera when mounted onto the drone.

6.3.1 Assumptions

When in the air the gimbal would rotate the camera to help locate each ArUco marker.

6.3.2 Responsibilities

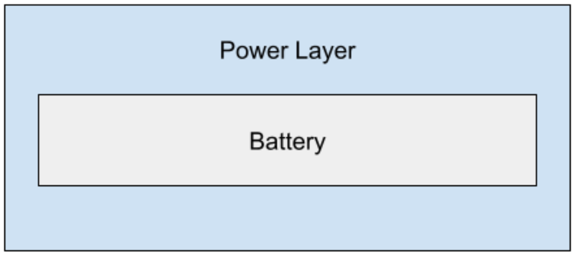
The Gimbal responsibilities would be to rotate the camera across each axis. With the gimbal it is possible that obtaining image recognition would be much easier.

6.3.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	Movement	Imagine Recognition Algorithms	Rotation from the gimbal.

7 Power System

7.1 Battery



7.1.1 Assumptions

Battery levels are not read by the computer.

7.1.2 Responsibilities

The battery is attached to the drone and is responsible for providing power to all parts attached to the drone's frame.

7.1.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#01	Power	N/A	Power to the system

8 Navigation Subsystems

8.1 GPS

The GPS will be used to identify the location of the drone. It receives signals from Ardupilot and sends signals to the RTK unit. this communication will assist with stabilizing the coordinates and flying in autopilot mode.

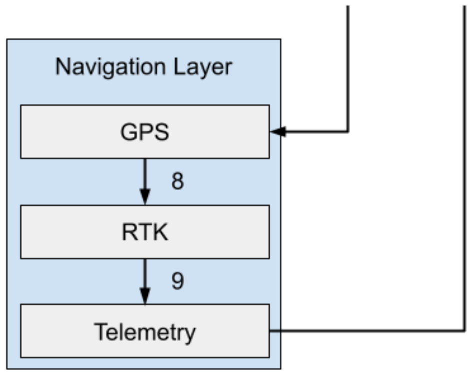


Figure 14: GPS Subsystem diagram

8.1.1 Assumptions

Assuming that the showcase will take place in an outdoor setting.

8.1.2 Responsibility

Responsible for mapping the directions to the desired location and also to tell where the drone is at a given point.

8.1.3 subsystem Interfaces

ID	Description	Inputs	Outputs
#07	Ardupilot	Desired location	N/A
#08	RTK	N/A	Current Location

8.2 RTK

The RTK or Real-Time Kinematic is a corrective surveying technique used by many Global Navigation Satellite Systems (GNSS)/Global Positioning Systems (GPS) to improve significantly the accuracy of their receivers. Compared to conventional satellite navigation systems (which have a positional accuracy of about two to four meters). The addition of RTK to the system increases the accuracy by a hundred times.

8.2.1 Assumptions

Assuming that the showcase will take place in an outdoor setting.

8.2.2 Responsibility

Responsible for sending signals to satellites to get precise coordinates. Also to is there has been a drift due too strong wind.

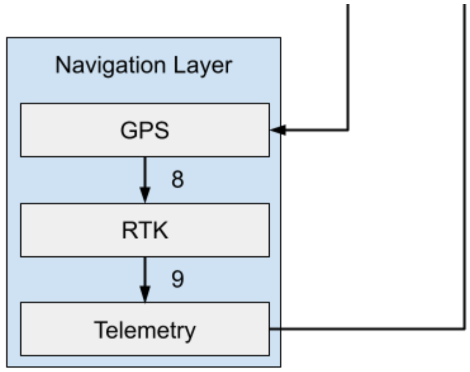


Figure 15: RTK Subsystem diagram

8.2.3 Subsystem Interfaces

ID	Description	Inputs	Outputs
#08	GPS	Estimated location	N/A
#09	Telemetry	N/A	precise Location

8.3 Telemetry

Telemetry data provides the ability to track UAV status in real-time, allowing pilots to monitor position, and altitude to ensure smooth and efficient flight.

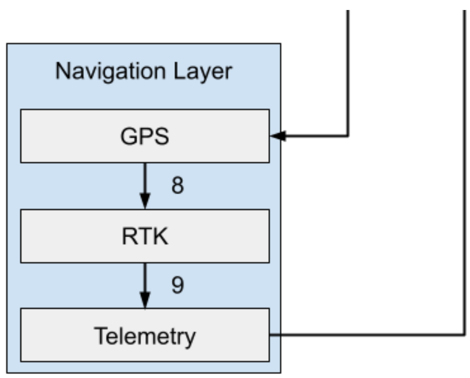


Figure 16: Telemetry Subsystem diagram

8.3.1 Assumptions

Assuming that the showcase will take place in an outdoor setting.

8.3.2 Responsibility

Keeping track of data in relation to the drone.

8.3.3 SUBSYSTEM INTERFACES

ID	Description	Inputs	Outputs
#09	RTK	Precise Location	N/A
#10	Ardupilot	N/A	GPS data