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

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2024

Verminator
Wasp Drone

Isaac Medrano
Luis Narez
Tyler Nguyen
Nishchal Shrestha
Rajesh Yaksho

Revision History

Revision	Date	Author(s)	Description
0.1	10.31.2024	TN	document creation
0.2	11.02.2024	TN, IM, RY, LN, NS	complete draft
1.0	11.04.2024	TN, IM, LN, RY, NS	official release
2.0	04.30.2025	TN, RY	Updated design with better reflection of final hardware.

CONTENTS

List of Figures

List of Tables

1 Introduction

The Wasp Drone is a remote-controlled device designed for safe and efficient wasp extermination, using a live-feed camera and an insecticide delivery system to neutralize wasp nests from a distance. The drone's spray system can be used for agricultural tasks like applying insecticides, herbicides, fertilizers, or water, making it a versatile tool for various applications. Key requirements include real-time video streaming, precise navigation, and customizable spray options, ensuring effective operation in diverse environments. The primary users include professional exterminators, property owners seeking an alternative to traditional pest control, farmers who can utilize its spraying capabilities, and hobbyists interested in modifying this open-source product for unique drone applications [1].

2 System Overview

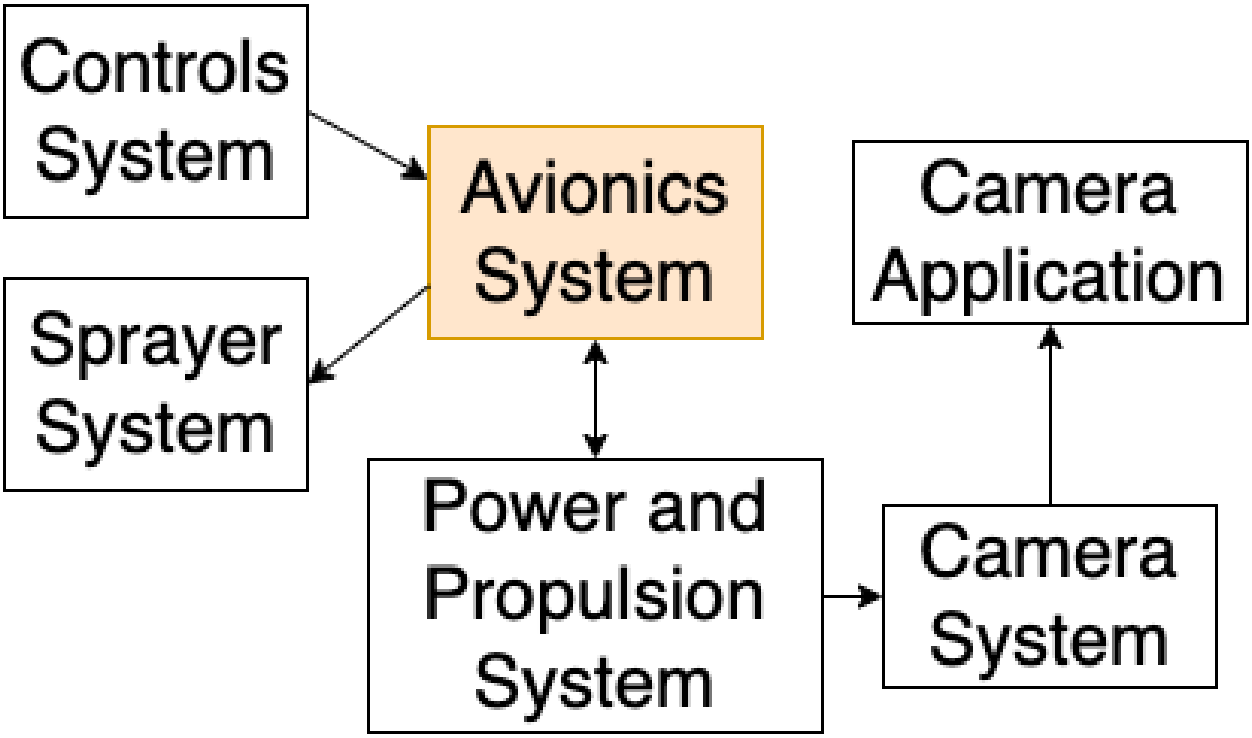

Figure 1: A simple architectural layer diagram

2.1 Avionics System Layer Description

The avionics system layer serves as the brain of the drone and contains the flight controller. User commands are fed into the flight controller through transmissions from the ground station or from a remote controller. Commands are passed to each individual component by the avionics system to perform an action. The avionics layer is composed of a flight controller (Cube Orange+ in this case), a telemetry subsystem (containing a telemetry radio, ground station, and related software for telemetry components), the GPS Module, and the Sprayer Firmware Module. The flight controller is used to control inherent drone functions and payload functions to carry out the drone's mission. The telemetry subsystem is needed to report flight data to the user and ground station. A GPS module is required for flight and is used to report location data to ground station.The sprayer firmware module controls and automates the operation of the drone's sprayer system.

2.2 Power and Propulsion System Layer Description

The power and propulsion system layer ensures that the drone can safely and effectively operate its motors and provides power for other systems in the drone. This layer is mainly composed of motors, electronic speed controllers (ESCs), and a Lithium-ion polymer (LiPo) battery. The motors and ESCs in this layer are used to provide the thrust necessary to lift the drone and the mission payload.

2.3 Controls System Layer Description

The Controls System Layer serves as the primary interface for operator input, managing both the drone's flight and commands for attached payloads. It encompasses two distinct control methods: the Shell

Subsystem, which enables programmatic operation through a ground station command-line interface, and the RC Subsystem, facilitating direct, real-time manual control via a remote transmitter.

2.4 Camera Application Layer Description

The Camera Application layer provides visual feedback for the user. Through a desktop application, the user can view a live feed from the camera in order to help piloting outside line-of-sight.

2.5 Sprayer System Layer Description

The Sprayer Layer's operation centers on a liquid pump activated via a PWM-controlled switch. This mechanism connects to the flight controller's auxiliary outputs, highlighting the Verminator Wasp Drone's modular design for controlling various payload devices.

2.6 Camera System Layer Description

The Camera System Layer in the wasp-killing drone is designed to provide real-time visual monitoring and transmission capabilities, essential for target identification. This layer consists of three subsystems: the FPV Camera Subsystem, the Video Transmission Subsystem, and the Video Capture Subsystem. The FPV Camera Subsystem uses a very low-light capable camera to capture wide angle frames of the view from the drone. The Video Transmission Subsystem is used to send video frames from the camera to the host device using analog radio. The Video Capture Subsystem presents the analog video as a USB video device for the host to process.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

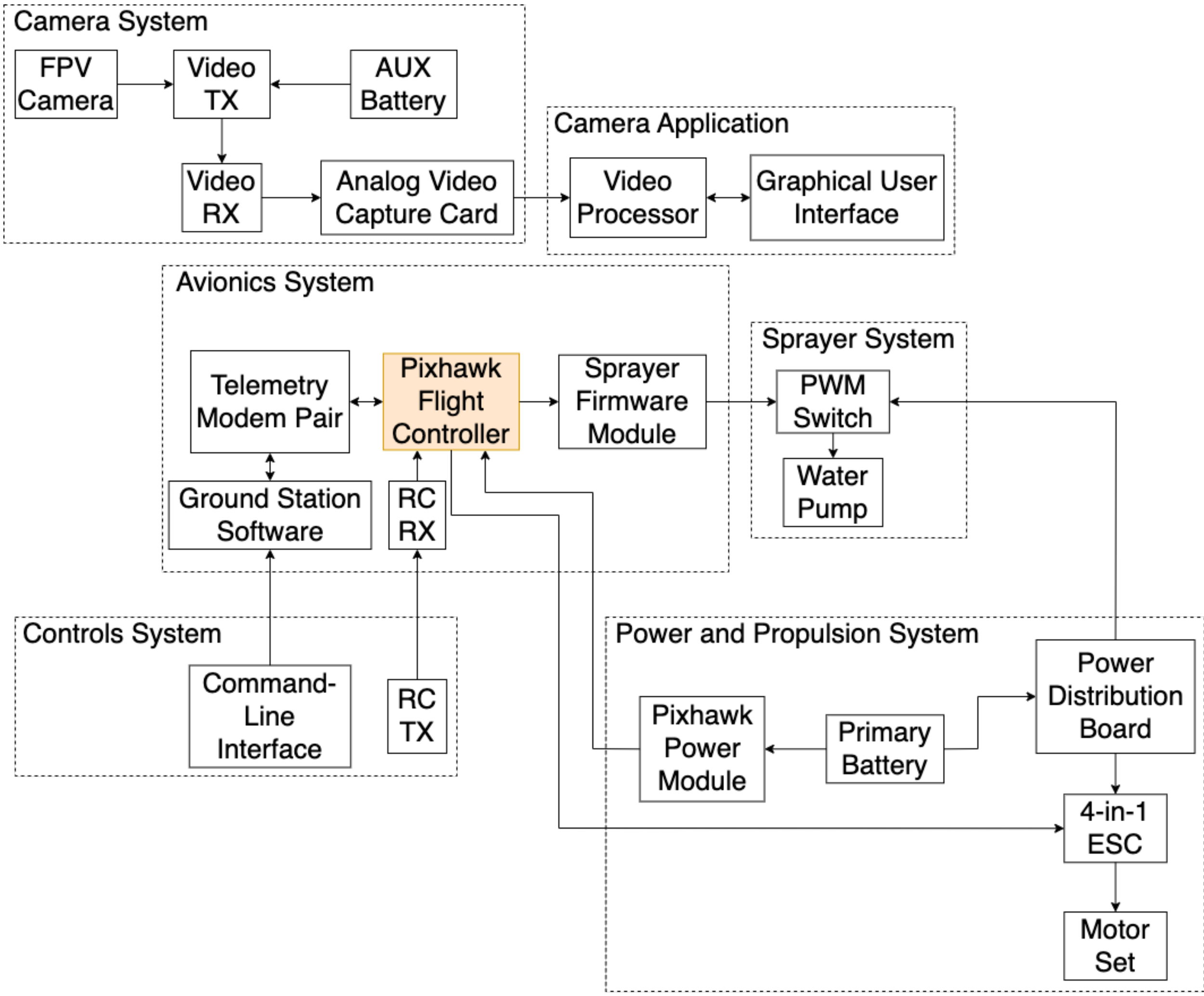

Figure 2: Full architectural design diagram showing data and power for the drone.

4 Avionics System

The avionics system serves as the "brain" of the drone. Commands issued by the user to activate various equipment and control the drone must first pass through the avionics system. The avionics layer is composed of the flight controller, the telemetry subsystem, the GPS module, and the sprayer firmware module.

4.1 Flight Controller Subsystem

The flight controller functions as the central processing unit of the avionics system. It receives input from the user and various sensors, processes the data through its firmware, and then generates appropriate actions for the drone. As such, the flight controller and its firmware directly control all components of the drone.

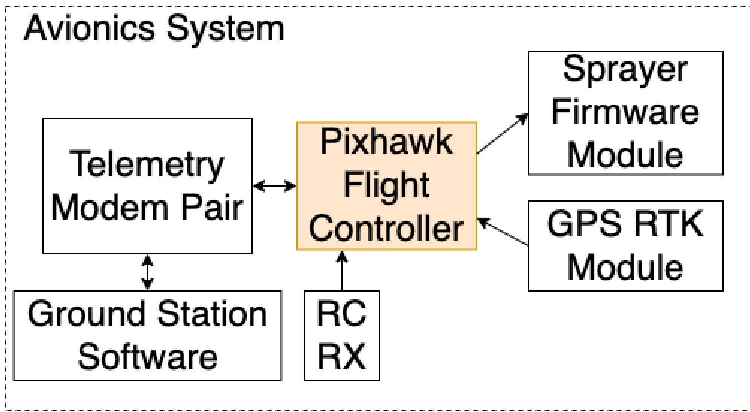

Figure 3: Avionics system highlighting flight controller

4.1.1 Assumptions

It is assumed that all sensors, peripherals, and connected subsystems (like the telemetry components, GPS, RC receiver, ESCs, power module, and sprayer module) are fully compatible with the Cube Orange+ flight controller and PX4 firmware.

4.1.2 Responsibilities

The flight controller subsystem is responsible for processing data from all connected sensors to determine drone's real-time state. Based on this sensor fusion and internal programming, it executes complex flight control algorithms necessary to stabilize the drone and enable it to follow desired flight paths.

It sends the appropriate control signals to the motor controllers and other actuators to achieve desired movements.

Furthermore, it serves as the primary interface for receiving and interpreting manual pilot commands from the RC receiver, exchanging telemetry data and accepting new commands from the ground station via the telemetry subsystem, and providing control interfaces for integrated payload systems, such as the sprayer firmware module.

4.1.3 Subsystem Interfaces

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	MAVLink protocol wireless interface to Telemetry Modem Pair	Data from Telemetry Modem Pair	Data to Telemetry Modem Pair
#02	Wireless RC Link (SBUS, PPM, CRSF) to RC receiver	RC Commands	None
#03	uORB topic publishing to sprayer firmware module	None	Actuator commands
#04	DroneCAN protocal wired interface to GPS RTK Module	Positional Data	None

4.2 Telemetry Subsystem

The telemetry subsystem enables the drone to communicate with the ground station. It works by receiving commands from the ground station and transmitting telemetry data back to it. Telemetry software is built into the flight controller, so only additional hardware is needed to facilitate a radio connection. In this case, antennas are connected to both the telemetry radio and the receiving ground station. Off-the-shelf software manages the implementation.

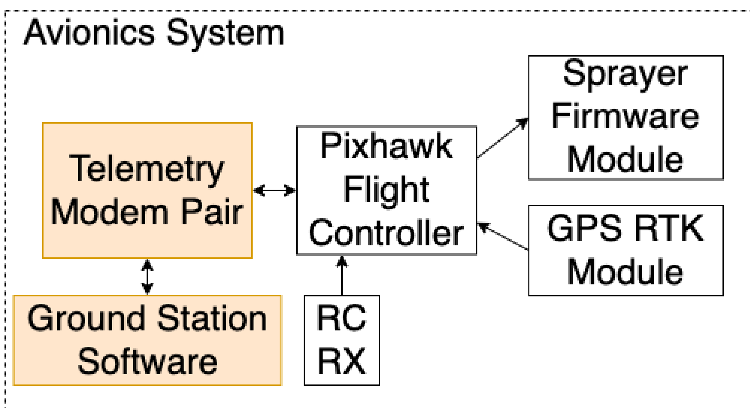

Figure 4: Avionics system highlighting telemetry subsystem

4.2.1 Assumptions

It is assumed that the telemetry components are fully compatible with the Cube Orange+ flight controller and operate using the MAVLink protocol.

4.2.2 Responsibilities

The telemetry subsystem is made up of a computer running ground station software, telemetry reporting firmware on the flight controller, and antennas for each component to transmit and receive data.

The ground station is used to issue commands to the drone and receive telemetry data from the drone's flight controller.

Each antenna on the ground station and the telemetry radio is used to establish a connection between the telemetry system and the ground station.

4.2.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	MAVLink protocol 900 MHz wireless signal between telemetry radio and ground station	Commands	Telemetry Data

4.3 GPS Subsystem

The GPS subsystem determines the position of a drone and plays a critical role in navigation.

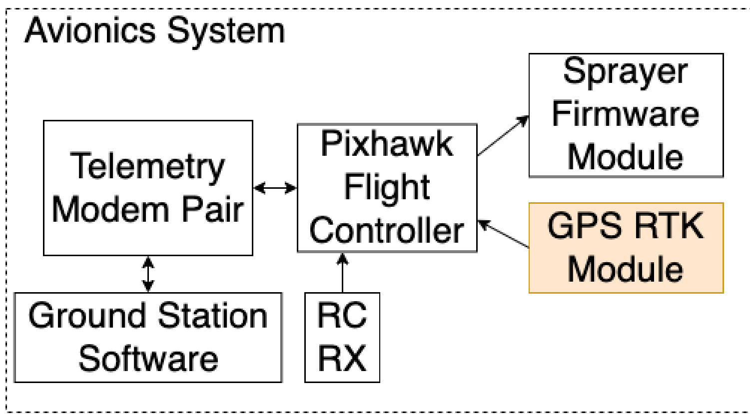

Figure 5: Avionics system highlighting GPS Module

4.3.1 Assumptions

The selected GPS module is assumed to be compatible with PX4 Autopilot systems and uses the DroneCAN protocol for communication.

4.3.2 Responsibilities

The GPS module is responsible for reporting data to the flight controller to aid in positioning and navigation. The chosen GPS module is a RTK DroneCAN GPS allowing for centimeter-level accuracy.

The GPS module is a plug-and-play subsystem that interacts with PX4 and Ardupilot systems out-of-the-box. It will only transmit navigational data to the flight controller.

4.3.3 Subsystem Interfaces

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	DroneCAN protocol over hardwire connection between GPS module and flight controller	None	Positional data

4.4 Sprayer Firmware Module Subsystem

The sprayer firmware module controls and automates the operation of the drone's sprayer system based on flight conditions or user commands.

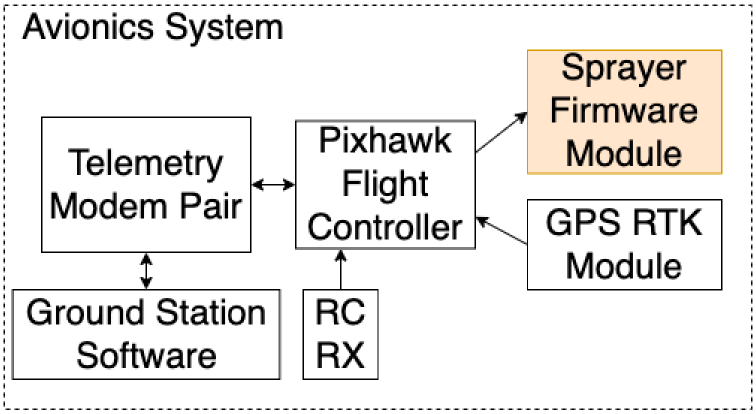

Figure 6: Avionics system highlighting sprayer modules.

4.4.1 Assumptions

It is assumed that the sprayer module is compatible with the Cube Orange+ flight controller and receives control commands through standard PX4/ArduPilot interfaces.

4.4.2 Responsibilities

The sprayer firmware module is responsible for managing the activation of the sprayer pump, typically based on GPS position, altitude, or preprogrammed mission triggers.

It interfaces with the flight controller to receive commands and controls the sprayer hardware accordingly. This subsystem is tightly integrated with the avionics layer for precise, autonomous spraying operations.

4.4.3 Subsystem Interfaces

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	uORB topic communication between flight controller and sprayer firmware module	Commands	Sprayer control

5 Power and Propulsion System

The power and propulsion system is a critical system for providing electrical energy to all components of the drone and generating the thrust for flight and maneuverability. This system is composed of the Propulsion Subsystem (motors and electronic speed controllers) and the Power Subsystem (lithium-polymer battery pack, power distribution board, and Pixhawk power module). The flight controller uses control firmware to command the Propulsion Subsystem, which then drives the motors to achieve desired flight characteristics.

5.1 Propulsion Subsystem

The Propulsion Subsystem is responsible for generating the thrust needed to lift and maneuver the drone. It consists of the brushless motors and the electronic speed controllers (ESCs) that control them. The arrangement of the four motors, typically in an alternating clockwise and anti-clockwise pattern, helps provide stability by cancelling out net torque.

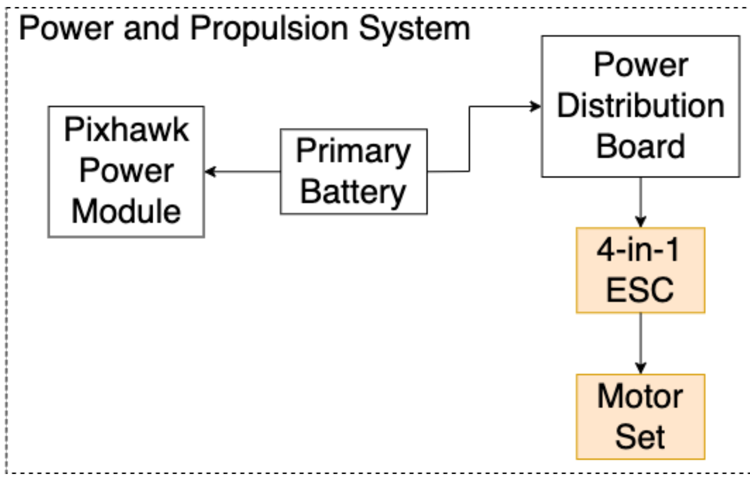

Figure 7: Propulsion system highlighting motors and ESC

5.1.1 Assumptions

It is assumed that the ESC firmware is compatible with PX4 Autopilot firmware using the DShot protocol or standard PWM, enabling precise and responsive motor control from the flight controller.

5.1.2 Responsibilities

It is responsible for providing the necessary thrust for the drone to carry out its mission. The motors need to provide enough thrust to lift the drone, battery, and payload while also being able to fly comfortably in a wide range of weather conditions. The ESCs are responsible for regulating the speed of each motor based on received commands and protecting the motors from overcurrent conditions by limiting power delivery.

5.1.3 Subsystem Interfaces

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Hardwire electrical connection between the motor and electronic speed controller.	Electrical power and control signals	Mechanical rotation
#02	Hardwire electrical connection between the motor and LiPo battery	None	None

5.2 Power Subsystem

The Power Subsystem is responsible for storing and distributing electrical energy to all components of the drone, including the Propulsion Subsystem, Flight Controller, and payload components.

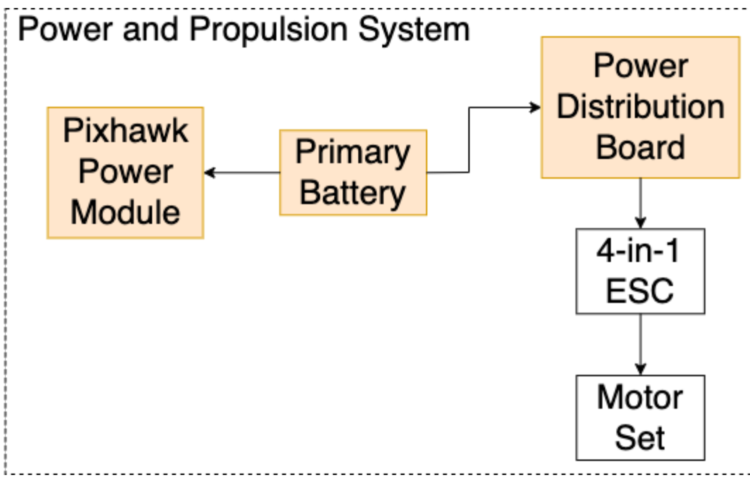

Figure 8: Propulsion system highlighting power subsystem

5.2.1 Assumptions

It is assumed that all power components, including the PDB, power module, and ESC, are compatible with PX4 and support XT60 and standard bullet connectors for power distribution.

5.2.2 Responsibilities

This subsystem is responsible for safely storing electrical energy, providing the necessary voltage and current to operate all drone components, distributing power efficiently through the PDB, and providing voltage and current telemetry data (via the Power Module) to the flight controller.

5.2.3 Subsystem Interfaces

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Battery to Pixhawk Power Module	Electrical Power from Battery	Electrical Power to Power Module
#02	Battery to PDB	Electrical Energy (Stored)	Electrical Power to PDB
#03	PDB/Power Module to Propulsion Subsystem Power	Electrical Power from Battery	Electrical Power for ESCs/Motors

6 Controls System

This section provides an overview of the control system, which includes multiple subsysterms for managing both manual and programmatic input. It includes the Radio Control (RC) and Shell Subsystem. The Flight Controls subsystem is responsible for stabilizing and maneuvering the drone, managing pitch, rotation, altitude and roll to ensure smooth flight. Both the RC Subsystem and the Shell Subsystem relay commands to the flight controller, enabling flexible operation modes either manual (via RC) or automated (via command-line interface).

6.1 Radio Control (RC) subsystem

The RC subsystem enables manual real-time control of the drone through a physical transmitter and onboard receiver.

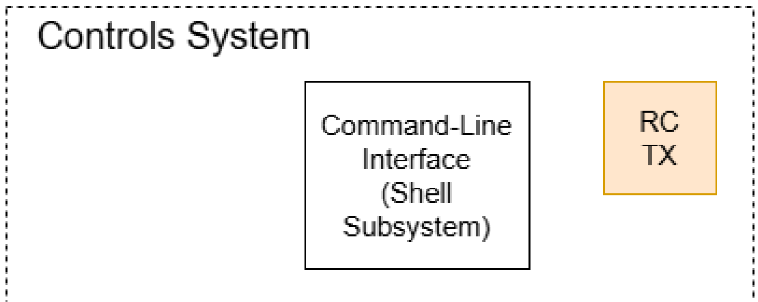

Figure 9: Controls system highlighting RC subsystem

6.1.1 Assumptions

The RC system is pre-configured and paired with the drone receiver. Standard 2.4 GHz frequency and channel settings are supported.

6.1.2 Responsibilities

This subsystem is responsible for transmitting user input to the flight controller to control drone orientation and throttle.It should provide immediate manual override in case of automated mission failure or deviation.

6.1.3 Subsystem Interfaces

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
#04	RC Tranmission	Manual Joystick Control	Radio signals to receiver
#05	Receiver to Flight Controller	Electrical connection	PWM/SBUS signals

6.2 Shell subsystem

The Shell subsystem provides a programmatic interface for mission automation and direct command-line interaction with the flight controller.

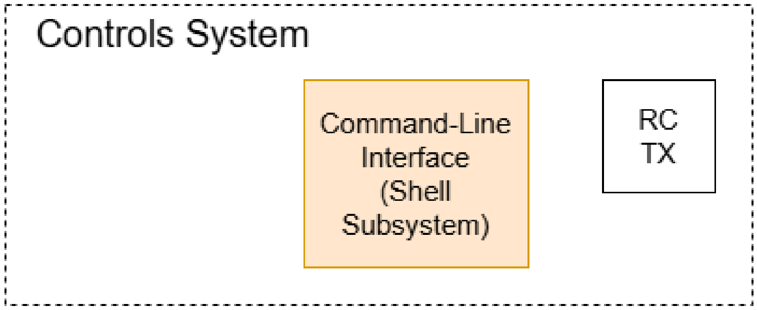

Figure 10: Controls system highlighting Shell subsystem

6.2.1 Assumptions

It is assumed that the PX4 firmware with shell access is installed on the flight controller. QGroundControl is used as the primary GCS for interfacing with the Shell subsystem.

6.2.2 Responsibilities

This subsystem is responsible for executing mission-level scripts and direct commands. It should provide debugging and direct configuration interface for developers or advanced users.

6.2.3 Subsystem Interfaces

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#06	Shell Command Input	CLI commands via GCS	Autopilot system instructions
#07	Shell Telemetry Feedback	Flight controller status	Output to GCS terminal

7 Camera Application System

This section describes the User Interface layer in detail, focusing on its subsystems and their interactions. The layer supports user interaction with the drone system by presenting essential telemetry, navigation, and stability data alongside the real-time camera feed. Each subsystem's description includes assumptions, responsibilities, and interface details.

7.1 Video Processor Subsystem

The Video Processor Subsystem is a software component that is designed to control a USB video device, perform image processing, and pass the final video frames to the Graphical User Interface Subsystem. While the core video streaming functionality is handled by the app's software, additional hardware such as a high-resolution USB camera and a compatible display interface are required to ensure optimal performance. Currently, the Video Processor performs blending algorithms and image format conversion as part of its processing pipeline. Future additions can be made to perform other image processing algorithms due to Video Processor's modular design.

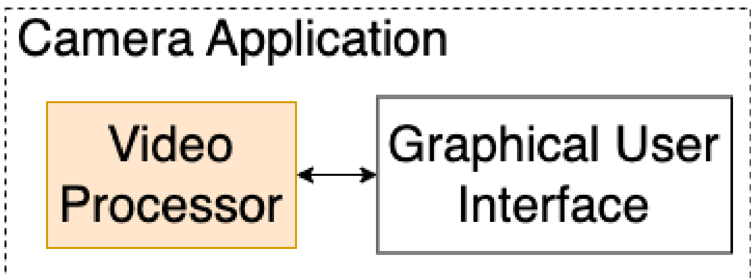

Figure 11: Camera application system highlighting the video processor subsystem.

7.1.1 Assumptions

It is assumed that the Video Processor Subsystem will receive a continuous stream of frames from any USB video device. The analog camera used in this project is connected to an analog video capture card which will present the video stream as a digital USB device for the Video Processor Subsystem to process.

7.1.2 Responsibilities

The Camera Application should capture and stream live video from the drone's camera to the information overlay for the user. It must minimize latency in the video feed to ensure real-time monitoring capabilities. Additionally, it should allow the selection of various USB video and camera devices and facilitate the superimposition of the crosshair for enhanced targeting accuracy.

7.1.3 Subsystem Interfaces

Table 10: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Video Stream Interface	Camera Transmission	Live video feed to GUI
#02	Information overlay interface	Video feed Overlay images	Overlay information and crosshair onto video

7.2 Graphical User Interface Subsystem

The GUI Subsystem overlays critical information on the live video feed provided by the Video Processor. Integrates telemetry data (speed, altitude, battery), GPS data, and stability information, visually presenting them to assist navigation. The subsystem includes a built-in crosshair to assist in tracking and alignment of objects during flight operations.

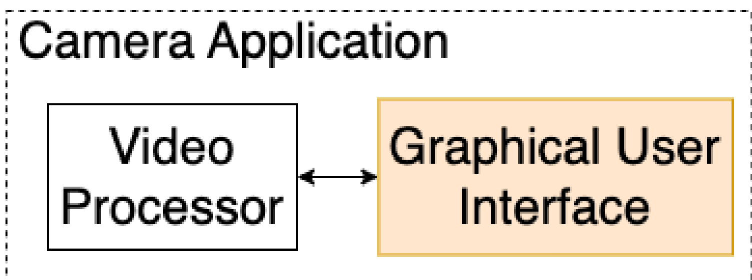

Figure 12: User interface system highlighting information overlay

7.2.1 Assumptions

The system is assumed to receive data from the Telemetry Data Source (if the drone is equipped with a telemetry OSD module). If no hardware module exists, then the camera should be able to continue displaying frames with simple overlaid images. The layout can overlay telemetry and other data on top of the video feed without obstructing essential information. Moreover, it is expected that there will be stable integration with the Camera app to display live video alongside other info overlay elements. It can also be assumed that the Video Processor Subsystem will pass frames in order and with low-latency while receiving input from the Camera System on a perfect medium. The design of the Camera Application will always display a frame, so internal error handling will allow the Video Processor to recover and continue to pass frames to the GUI.

7.2.2 Responsibilities

It should display the real-time video feed from the Camera app. The system should overlay telemetry data, including speed, altitude, and battery level, on the video feed for user visibility. It should also show GPS and stabilization information as part of the info overlay to assist with navigation and balance monitoring. Crosshair functionality must be integrated to allow users to track targets and enhance their navigation experience.

7.2.3 Subsystem Interfaces

Table 11: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Video Feed Interface	Processed Video Frames	Display Frame on Video Label

8 Sprayer System

The Sprayer System enables the controlled delivery of insecticide during drone flight. It is made up of only electrical hardware components that work together to perform precision spraying. This section outlines the assumptions, responsibilities, and interfaces associated with the subsystem.

8.1 Control Logic Via Flight Controller

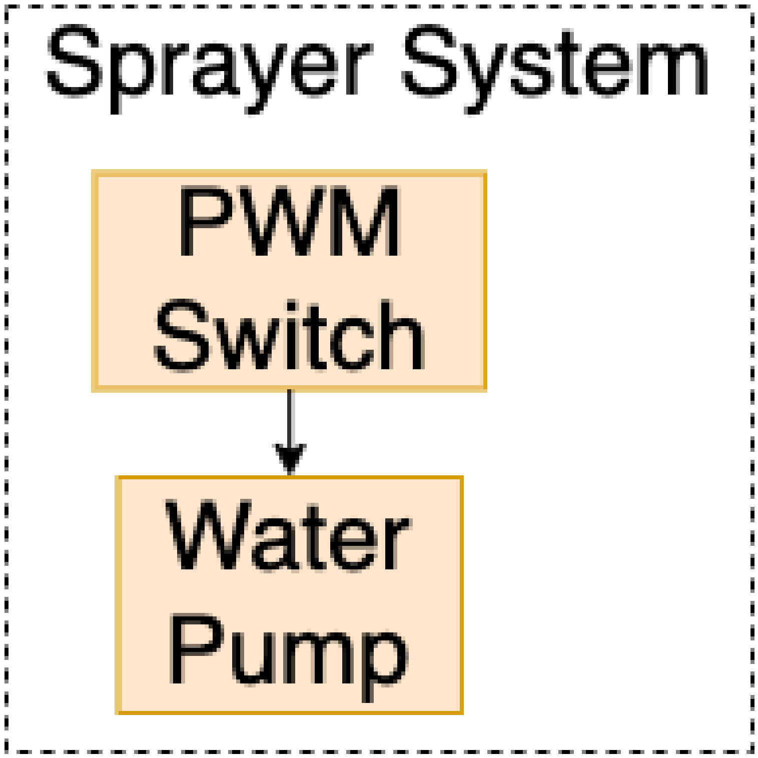

Figure 13: Sprayer system highlighting PWM-controlled switch and pump

8.1.1 Assumptions

It is assumed that the flight controller sends a valid PWM signal to the PWM-controlled switch to toggle the electric pump. It is also assumed that the power and ground connections to the pump are stable and correctly configured.

8.1.2 Responsibilities

The flight controller, via custom firmware, generates high or low PWM signals to toggle the state of the electric pump. When activated, the pump draws insecticide from the reservoir and expels it through the spray nozzle.

8.1.3 Subsystem Interfaces

Table 12: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Pump Activation	PWM signal from flight controller	Pump on/off toggle signal

9 Camera System

The Camera System enables real-time video monitoring and navigation of the drone by capturing, transmitting, and digitizing video. It is composed of three subsystems: the FPV Camera, the Video Transmitter, and the Video Capture Subsystem. Together, these components provide operators with a continuous video feed during operation.

9.1 FPV Camera Subsystem

The FPV Camera provides real-time video capture from the drone's perspective. Its primary function is to deliver a continuous video stream that enables operators to visually monitor the drone's surroundings and identify potential wasp targets. The FPV Camera directly interfaces with the Video Transmission Subsystem, which transmits this live feed to an external receiver, allowing for remote observation and control adjustments based on the visual data.

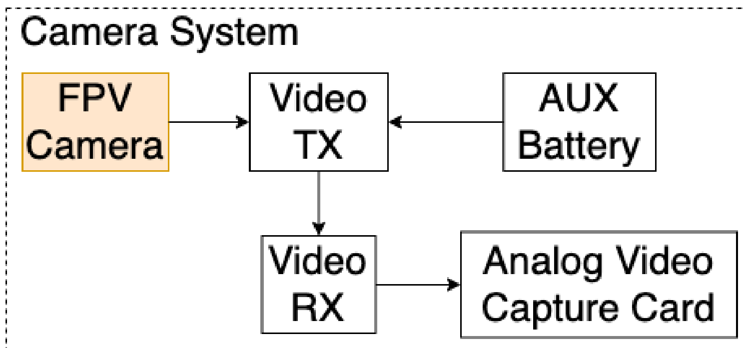

Figure 14: Camera system highlighting FPV camera

9.1.1 Assumptions

The FPV Camera provides continuous video streaming and does not perform any image processing.

9.1.2 Responsibilities

The Video Transmission Subsystem takes the video feed generated by the FPV Camera and transmits it to an external receiver, such as a control station on the ground. This allows operators to monitor the live video stream and make decisions based on the visual data.

9.1.3 Subsystem Interfaces

Table 13: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	FPV Camera Output Interface	Power input	Analog video signal

9.2 Video Transmission Subsystem

The Video Transmission Subsystem sends the video feed to an external receiver. This allows operators to view live footage remotely, enabling real-time response and monitoring during the drone's operation.

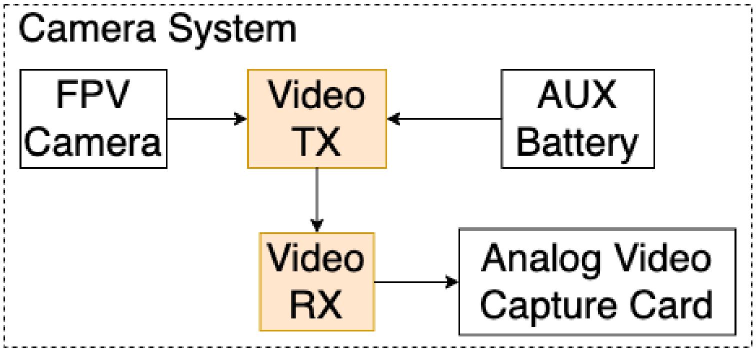

Figure 15: Camera system highlighting video transmitter

9.2.1 Assumptions

It is assumed to have a reliable signal range sufficient to transmit live video back to a ground station or monitoring device.

9.2.2 Responsibilities

The Video Transmitter should take the video feed generated by the FPV Camera and transmit it to an external receiver, such as a control station on the ground. This allows operators to monitor the live video stream and make decisions based on the visual data.

9.2.3 Subsystem Interfaces

Table 14: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Video Transmitter Interface	Video frame NTSC format	Video frame NTSC format (5.8 GHz radio)

9.3 Video Capture Subsystem

The Video Capture Subsystem receives the transmitted analog video via a ground station receiver, digitizes it, and provides it as a USB video input to a Linux device or similar host.

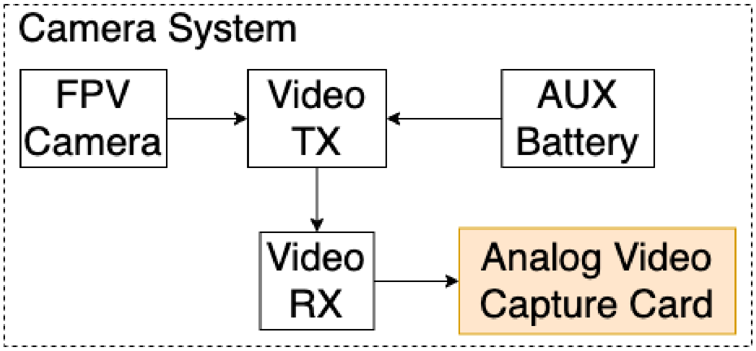

Figure 16: Camera system highlighting video transmitter

9.3.1 Assumptions

It is assumed that the host system supports USB Video Class (UVC) devices for video capture functionality. On desktop, the capture card will appear as a USB video device with all of the correct USB headers for that type of device.

9.3.2 Responsibilities

It is responsible for converting the analog video to digital format using a capture card. It should provide a real-time digital video stream to a user application. Analog to digital conversion is critical as the Camera Application can only interface with USB video devices.

9.3.3 Subsystem Interfaces

Table 15: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Video Digitization Interface	Analog Video (RCA)	Digital Video via USB

References

[1] Luis Narez Rajesh Yaksho Nishchal Shrestha Tyler Nguyen, Isaac Medrano. System requirements specifications, 2024.