Semi-autonomous Drone System

-Drone delivery through QR scanning

submitted in fulfillment of the requirement

for the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY

PATIALA

October 2024

ABSTRACT

This project delves into the exciting realm of smart drones, with a focus on integrating QR code detection technology for enhanced precision during package identification. These autonomous drones navigate pre-programmed paths using onboard sensors and software, eliminating the need for human pilots. Cameras serve as the drone's eyes, not only capturing real-time video data for obstacle avoidance but also detecting QR codes on packages while hovering at a specific height. This ensures accurate package identification. QR code detection allows the drone to verify the package details mid-flight or just before landing, enhancing overall accuracy. Mission planning software plays a crucial role, enabling pre-planning of flight paths with waypoints, obstacle avoidance maneuvers, and real-time telemetry monitoring throughout the mission. However, safety remains paramount. Developing and adhering to protocols for takeoff, hovering, landing, and emergency situations is essential. Additionally, understanding and complying with drone regulations regarding payload weight, flight paths, and airspace restrictions is crucial for safe and legal operation. By integrating intelligent systems such as QR detection, smart drones offer the potential for precise operations, particularly in areas with challenging access, further pushing the boundaries of autonomous flight technology.

In conclusion, this innovative project showcases the potential of smart drones equipped with QR code detection technology for precise package identification and delivery. By combining autonomous navigation, real-time camera analysis, and intelligent mission planning, these drones can revolutionize various industries, from logistics and delivery to inspection and surveillance. As technology continues to advance, we can expect to see even more sophisticated drone applications that enhance efficiency, safety, and accessibility in diverse fields.

TABLES OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

BVLOS	Beyond Visual Line of Sight
EASA	European Union Aviation Safety
ESC	Electronic Speed Controller
FAA	Federal Aviation Administration
FSC6B	FlySky CT6B (A type of RC transmitter)
GES	Ground Equipment Station
GIT	Version Control System (often referred to as Git)
GPS	Global Positioning System
ISMS	Information Security Management System
kv	RPM per volt (used in motor specifications)
LIDAR	Light Detection and Ranging
Lipo	Lithium Polymer (battery type)
mah	Milliampere-Hour (battery capacity)
mhz	Megahertz (frequency unit)
NEO-7M	A GPS module (UBlox NEO-7M)
OpenSourcCV	Open Source Computer Vision Library
OS	Operating System

BVLOS	Beyond Visual Line of Sight
EASA	European Union Aviation Safety
PGM	Programmable (often used with electronic speed controllers)
PID	Proportional-Integral-Derivative (control loop mechanism)
QR Code	Quick Response Code
SAS	Sense and Avoid System
TIET	Thapar Institute of Engineering and Technology
VS Code	Visual Studio Code (code editor by Microsoft)

Chapter 1: INTRODUCTION

1.1- Project Overview

The **Semi-Autonomous Drone System** project is a simple college project aimed at developing a drone capable of detecting packages using QR codes while hovering above them. The system integrates essential hardware, including a sturdy frame, high-thrust motors, a Pixhawk flight controller, high-precision GPS, high-capacity batteries, and a camera for **QR code detection**.

On the software side, **Mission Planner** is used for mission planning and flight control, while **OpenCV-based QR code detection algorithms** ensure accurate package identification. The workflow involves navigating to the target location, hovering above the package, detecting it via the QR code, and then returning to base.

Key challenges include:

- Ensuring reliable **QR code detection** while the drone hovers
- Managing **battery life** to complete the mission
- **Drone weight management**, as the drone itself is heavy due to the battery and power bank used to power the Raspberry Pi and camera
- Securing communication and control systems

The testing phase focuses on validating QR code detection and stability in a controlled environment. Performance is optimized using flight logs and telemetry data.

This project aims to demonstrate the use of **QR code-based package detection** in a straightforward, semi-autonomous drone setup.

1.2- Motivation

The motivation behind this project stems from the growing demand for intelligent and precise autonomous systems, particularly in the context of QR code detection for accurate package identification while drones are hovering at a height. In today's fast-paced world, there is a need for drones that can efficiently carry out tasks such as package tracking and management, especially in industries requiring real-time identification of goods.

Traditional methods of package tracking often face limitations in accuracy and speed, especially when operating in complex or dynamic environments. As industries seek more innovative and reliable solutions, the integration of drones with advanced sensing capabilities—such as QR code detection—presents a promising opportunity. These autonomous drones can hover and scan packages from a specific height, eliminating human intervention and reducing errors associated with manual package handling.

One key advantage of smart drones is their ability to operate in environments where traditional systems may struggle, such as areas with limited access or in situations that demand rapid identification. By **incorporating QR code detection, drones can autonomously verify packages mid-flight, reducing delays and improving operational accuracy**. This feature is particularly valuable in settings where the wrong package identification could result in logistical challenges or costly delays.

Furthermore, the integration of smart technology in drones enables the automation of package management processes, reducing operational costs and increasing overall efficiency. This capability is critical as businesses aim to streamline operations while maintaining high standards of accuracy. Additionally, drones with QR detection can contribute to enhanced security by verifying packages before delivering or handing them over.

The motivation behind this project is to leverage drone technology with QR code detection to address the increasing need for precise, autonomous package identification, thereby improving efficiency, accuracy, and reliability across various sectors.

1.3- Assumptions and Constraints

1.3.1 Assumptions

ⅰ. Regulatory Compliance:

It is assumed that the drone system will adhere to relevant legal and regulatory requirements for semi-autonomous drone operations. This includes compliance with aviation authorities like the FAA (U.S.), EASA (Europe), or other regional bodies, especially for hover operations and QR code scanning over specified areas.

ⅱ. **Technological Feasibility:**

The project assumes that current technology, including high-resolution cameras, object detection algorithms, and real-time image processing, is capable of reliably detecting and scanning QR codes while the drone hovers at a specific height. This also assumes that the flight controllers and onboard processing units are sufficiently advanced to handle the computational load required for real-time QR detection.

ⅲ. **Hovering Stability:**

It is assumed that the drone is capable of maintaining a stable hover at a predefined altitude long enough to detect and process the QR codes. The drone's control systems, including altitude hold and position lock capabilities, are assumed to function effectively even in moderate wind or external disturbances.

ⅳ. **Battery Efficiency:**

The assumption is that the drone's battery will provide sufficient power for extended hovering durations needed for QR detection, including safe return to base or charging stations. Battery management systems are assumed to optimize power usage, especially during precision hovering.

ⅴ. **Environmental Conditions:**

The system assumes that the drone can effectively operate under standard weather conditions such as light wind or moderate rain. However, extreme weather conditions, including heavy rain, strong winds, or fog, may limit QR detection accuracy or drone stability and are considered outside the operational envelope.

ⅵ. **QR Code Size and Placement**:

It is assumed that the QR codes to be detected are of a size and quality that can be reliably scanned from the drone's operational hovering height. QR codes are assumed to be placed on packages or areas in a clear and visible manner to avoid any scanning obstructions

ⅶ. **Real-Time Processing**:

The project assumes that the onboard systems or ground control units have sufficient processing power to perform real-time QR detection and interpretation. It is also assumed that communication latency is minimal, ensuring that data from QR scanning can be transmitted without significant delay.

ⅷ. **Network Connectivity**:

Reliable and consistent network connectivity is assumed for real-time communication between the drone and ground control for telemetry monitoring, QR detection feedback, and any necessary remote overrides. In remote areas where network coverage may be limited, fallback systems like pre-programmed flight paths are assumed to function adequately.

ⅸ. **Obstacle Avoidance**:

It is assumed that the drone's obstacle detection systems will function effectively to avoid collisions during hovering and QR scanning. This includes sensors such as LIDAR, ultrasonic, or cameras to detect and avoid obstacles in close proximity to the drone.

ⅹ. **Safety Protocols**:

It is assumed that adequate safety protocols, such as emergency landing procedures and return-to-home functionalities, are in place to ensure safe operation during QR scanning, especially in case of unexpected failures or disruptions.

1.3.2 Constraints

ⅰ. **Regulatory Constraints:**

The drone must comply with aviation regulations, which may restrict flight altitude, hover time, and no-fly zones. Line-of-sight requirements could also limit the drone's operational range, particularly in urban areas or restricted airspace.

ⅱ. **Battery Limitations:**

The drone’s flight time and hovering duration are constrained by battery capacity, which directly impacts how long the drone can remain in the air for QR detection. Battery performance may also be affected by weather conditions and payload weight, limiting the system’s operational range.

ⅲ. **Hovering Stability:**

Maintaining a stable hover at a specific height for QR detection may be challenging in adverse weather conditions such as wind gusts or turbulence. This constraint can affect the accuracy of QR code scanning and may require additional stabilization measures or hardware upgrades.

ⅳ. **QR Detection Accuracy:**

The accuracy and efficiency of QR code detection are constrained by factors such as camera resolution, lighting conditions, and the distance between the drone and the QR code. Poor visibility or incorrect QR code placement can hinder successful scanning.

ⅴ. **Real-Time Processing Power:**

The onboard processing power available for real-time image recognition and QR code scanning may be constrained by hardware limitations. If the computational requirements for image processing are too high, it could delay or affect the accuracy of QR detection.

ⅵ. **Network Connectivity:**

Real-time communication between the drone and the control system depends on reliable network connectivity. In remote areas or regions with poor network infrastructure, delayed or lost signals may constrain the drone’s ability to transmit QR detection data or receive commands.

ⅶ. **Weather Conditions:**

The effectiveness of the drone’s operations, particularly for hovering and QR code detection, may be limited by adverse weather conditions such as rain, wind, or fog. These conditions can affect both the drone's stability and camera performance, limiting its ability to complete tasks.

ⅷ. **Security Concerns:**

Ensuring secure communication channels to prevent unauthorized access or interference is a critical constraint. Robust encryption and data protection protocols are required, adding complexity to the system and potentially impacting communication efficiency.

ⅸ.**Technical Integration:**

Seamless integration of multiple components such as sensors, cameras, flight controllers, and QR detection software poses a challenge. This constraint may extend development time and requires thorough testing to ensure reliable operation.

ⅹ.**Power Consumption:**

The energy requirements for running sensors, cameras, and processing systems alongside flight operations place a constraint on power usage, reducing overall flight time and limiting operational flexibility, especially for longer missions.

1.4- Novelty of Work

The Semi autonomous drone system project stands out due to several innovative aspects that push the boundaries of current drone technology and delivery solutions. It offers an autonomous end-to-end delivery system, eliminating the need for human intervention from package pickup to drop-off.

A typical drone system is designed to handle a significant payload, making it versatile for various applications, including delivering medical supplies, food, and e-commerce packages. It also boasts robust environmental adaptability, enabling reliable operation in adverse weather conditions such as light rain and moderate winds, thus extending its operational window beyond that of conventional drones. Seamless integration with Mission Planner provides advanced mission planning, real-time telemetry, and comprehensive flight control features, supporting complex waypoint navigation and automated takeoff and landing.

A key innovative feature of the system is its ability to precisely hover at a predetermined altitude to detect QR codes on packages. This precise hovering capability is crucial for accurate QR code scanning and package identification, ensuring efficient and reliable delivery. The drone's advanced flight control algorithms and sensors enable it to maintain a stable hover, even in challenging wind conditions, thereby enhancing the accuracy and reliability of the QR code detection process.

A smart battery management system enhances safety and efficiency by optimizing power consumption and providing timely alerts for low battery conditions, facilitating safe return-to-home maneuvers. A user-friendly interface allows customers to place orders, track deliveries in real-time, and receive notifications via web and mobile platforms, promoting wider adoption of the service.

Additionally, the modular design and software architecture allow for easy scalability and customization, making the system adaptable for various delivery scenarios and requirements, from urban settings to remote rural areas. These innovative features collectively make the Semi-autonomous Drone System a pioneering project, setting new standards for efficiency, reliability, and sustainability in drone-based delivery services.

Chapter 2: LITERATURE SURVEY

2.1-Literature Survey

Some references and research work that greatly influenced and helped us in building the initial mindset for drone technology and for further analysis also, its usage are mentioned below:

Moshref-Javadi and Winkenbach (2021) provide a structured framework for categorizing drone logistics, highlighting operational design and planning challenges relevant to semi-autonomous drones like those used in QR code detection. Their review reveals that, while multi-drone Pure-play Drone-based (PD) models are popular, there remains an underexplored area in Synchronized Multi-modal (SM) and Resupply Multi-modal (RM) models, particularly for real-world applications beyond e-commerce and emergency services. This insight is relevant for optimizing the design of semi-autonomous drones that can detect QR codes while hovering over packages, identifying specific operational models that enhance precision and functionality within constrained delivery scenarios.

Dolata and Schwabe (2023) emphasize the importance of public acceptance in drone applications beyond privacy and airspace concerns, focusing on perceived justice in operational use. For semi-autonomous drones tasked with package identification, such as QR code detection, incorporating public perceptions of fairness and transparency into the operational framework could enhance community support and acceptance. Leveraging organizational justice principles could help align the semi-autonomous drone's deployment strategy with public expectations, fostering legitimacy and addressing potential ethical considerations in routine, low-risk logistics scenarios.

Wu, Ye, and Du (2024) explore multi-objective reinforcement learning (MONRL) for autonomous drone navigation, relevant to drones that need to detect QR codes while hovering over packages. Their research addresses the challenge of navigating drones in complex urban environments with dynamic constraints, such as wind zones, which are critical for maintaining stability and control during flight. This aligns with the semi-autonomous drone's requirement to hover steadily for accurate QR code detection over packages, especially in environments where wind might disrupt precision. The MONRL approach, leveraging deep reinforcement learning and memory, enables drones to adapt navigation strategies using only camera-based data, eliminating the need for additional aerodynamic sensors. This technique could be adapted to improve the drone's hovering stability and accuracy in identifying package locations, even in the presence of varying environmental factors, contributing to the broader field of automated urban logistics and civil infrastructure tasks.

Yoo and Chankov (2018) discuss the potential of combining drone delivery with autonomous mobility to address last-mile delivery challenges, offering insights applicable to semi-autonomous drones aimed at QR code detection over packages. Their proposed Drone-delivery using Autonomous Mobility (DDAM) framework addresses high delivery demand, short lead times, and traffic congestion—factors also relevant to improving package detection and delivery accuracy in dense urban areas. This concept underscores the feasibility of autonomous drones for high-demand applications, suggesting that the combination of precise hovering for QR detection and autonomous adaptability to urban congestion could enhance the efficiency of package location and delivery in dynamic environments.

Miranda et al. (2022) introduce an autonomous navigation system for delivery drones, addressing key considerations for drone-based delivery, including precise localization and smooth navigation. For a semi-autonomous drone focused on QR code detection while hovering over packages, their approach offers relevant insights, particularly in using GPS, IMU, and barometer for accurate positioning. The use of ArUco marker detection, combined with ultrawideband (UWB) devices and an extended Kalman filter for enhanced landing precision, aligns with the stability requirements necessary for accurate QR code scanning. Their vector field-based control method also minimizes vibrations, which is essential for maintaining focus on the target QR code, making this navigation system valuable for reliable and precise autonomous package delivery operations.

Xia, Guo, and Peng (2024) address the "last-hundred-feet" challenge in autonomous drone delivery, presenting Structural Semantic Segmentation (SSS) as a novel computer vision technique to enhance precise, house-aware navigation to designated drop-off points like doors or garages. This approach aligns with the goals of a semi-autonomous drone for QR code detection, as it demonstrates the feasibility of computer vision-based navigation for accurate positioning. The real-time semantic segmentation achieved by SSS could be adapted to improve the drone's capability to hover stably over packages for QR detection, enhancing delivery precision by identifying key locations without delays due to slow visual processing. Their implementation of SSS in an Android app validates its potential for streamlined and practical deployment in residential delivery scenarios.

Sachdeva et al. (2023) propose a decentralized, open-source framework for drone-based delivery systems, focusing on secure user verification and authentication to ensure reliable delivery directly to recipients. This approach is relevant for a semi-autonomous drone system designed to detect QR codes while hovering over packages, as it underscores the importance of secure delivery from sender to recipient. Their DIY autonomous quadcopter model emphasizes scalability and adaptability for urban environments, where logistical demands and security concerns are heightened. The framework's verification method could inspire further refinements in QR code-based package identification, ensuring secure and authenticated delivery within dense urban settings.

Yeh et al. (2023) develop a semi-autonomous UAV for food delivery, addressing the growing demand for contactless services heightened by the pandemic. Their project utilizes a GPS-based navigation system, enabling the drone to transport small food packages from a docking area to designated landing spots. Equipped with a carbon-fiber airframe, propellers, a flight controller, and electronic speed controllers, the UAV also integrates a camera with April Tags for precise landing. This approach highlights the potential for scalable and flexible drone systems that can be adapted to future automation needs in food delivery. The project exemplifies how technological advancements can streamline everyday services while ensuring safety and efficiency.

Shahzaad et al. (2023) present a context-aware framework to enhance drone delivery systems in smart cities, addressing the challenges of optimizing package delivery through aerial networks. By reformulating the delivery pathway problem into a drone service composition problem, the authors introduce a line-of-sight heuristic-based algorithm designed to select near-optimal delivery services. Extensive experiments using real datasets demonstrate the robustness of their approach, highlighting its potential to improve energy consumption and payload efficiency in urban logistics. This work contributes to the growing discourse on integrating drones into smart city infrastructures, emphasizing the need for context-awareness in delivery operations.

Mahmud Raivi et al. conduct a comprehensive survey on emerging drone routing algorithms for drone-based delivery systems, emphasizing critical design aspects such as trajectory planning, charging, and security. They address essential considerations for flexible and reliable parcel delivery, including route planning, payload weight, distance measurement, and customer location. The authors propose a novel taxonomy to categorize existing algorithms, reviewing each in terms of key features, operational characteristics, advantages, limitations, and performance. Their analysis highlights the challenges in designing effective routing solutions, paving the way for future research by identifying open research challenges in this rapidly evolving field. This work underscores the importance of practical design considerations to enhance the efficiency and reliability of drone delivery systems.

Gerrits and Schuur (2021) explore innovative concepts for last-mile delivery in smart cities, focusing on a synchronized delivery system involving trucks, drones, and street robots. They propose a model where trucks deploy a mixed fleet of drones and robots, highlighting the truck driver's active role in the process, which is often overlooked. The study aims to align the truck driver's service time with the delivery times of the drones and robots, utilizing a continuous approximation street delivery model. Their flexible and reusable simulation model allows logistics providers to make tactical decisions regarding the most suitable neighborhoods for deploying drones, robots, or their combinations, optimizing for makespan and energy consumption. This research contributes valuable insights into enhancing efficiency in urban logistics.

Marquez and Luna (2023) present a mathematical model aimed at analyzing the dynamic instabilities of an unmanned aerial vehicle (UAV) and its delivery system. By employing a Newton-Euler methodology, the authors establish the equations of motion for the UAV, which is modeled as a rigid body with six degrees of freedom—three for linear motion and three for rotational movement. This approach provides insights into the drone's maneuverability during operation. The delivery system, attached to the UAV, is also modeled to consider oscillations related to the platform's motion and the payload's impact. The study evaluates the UAV's response under various initial conditions, using power spectral density functions to clarify the dynamic characteristics of the drone, ultimately contributing to the optimization of drone design in delivery applications.

In this project, we have designed a Semi-autonomous Drone System for detecting packages using **QR codes** while the drone hovers above them. The drone does not involve obstacle avoidance or payload delivery but focuses solely on **QR code detection**. A camera is used to identify the QR code on the package, enabling precise detection. The drone's processing unit uses **real-time camera data** for accurate identification, ensuring reliable hovering while performing the detection task. Once the QR code is detected, the drone concludes its operation and returns to base.

Goal: Develop a precise, camera-guided system for **QR code-based package detection**.

The focus is on successful " hover and detect" operations for target applications. Scope:

● No additional payload or weight management aside from drone hardware

● Flight distance: Covers essential distance to hover over the target

● QR code detection as the main focus.

Need: Develop a precise, **QR code-based detection** system to demonstrate simple package identification through semi-autonomous drone operation.

We evaluate commercially available drone platforms and components for their suitability in providing QR code detection capabilities. This includes an analysis of hardware components such as Picamera v1.3, Raspberry Pi board model 4, motors, ESCs, GPS, and battery.

Additionally, we explore **open-source mission planning software** (e.g., Mission Planner, ArduPilot) for waypoint navigation and assess their ability to integrate QR code detection algorithms with semi-autonomous flight control. The research further involves reviewing existing projects that utilize QR code detection in autonomous drones to evaluate their success and relevance to our project.

2.2-Research Gap

- **Limited research on precise hovering for QR code scanning:** While drone hovering algorithms exist, there was a gap in research specifically addressing precise hovering at a predetermined altitude for accurate QR code scanning during package delivery. Our project focuses on developing robust algorithms for maintaining stable hovering while ensuring optimal QR code detection conditions.
- **Integration challenges between hardware, software, and QR code detection:** Integrating the camera, weight sensors, mission planning software, and QR code detection algorithms with the chosen drone platform might pose challenges. Research gaps could exist in areas like efficient data transmission between components, power management for extended flights, real-time processing limitations onboard the drone, and optimizing QR code detection algorithms for varying lighting conditions and package orientations.

2.3-Problem Definition and Scope

In today's world, there is a growing demand for fast, efficient, and convenient solutions in various sectors. Traditional delivery methods often face challenges such as traffic congestion, infrastructure limitations, and inefficiencies in remote locations. This project aims to address these challenges by developing a **Semi-autonomous Drone System** for detecting packages using **QR codes** while hovering above them.

This project focuses on creating a conceptual framework for a Semi-autonomous Drone System designed specifically for QR code-based package detection. The scope will encompass the following key areas:

- **Drone Technology**: Research and analyze different types of drones suitable for QR code detection, considering factors like hardware integration (camera, Raspberry Pi, power systems), flight stability, and control mechanisms.

- **Operations**: Define the operational aspects of the system, including navigation to the designated location, hovering, QR code detection, and efficient return to base.
- **Regulatory Landscape**: Investigate current regulations governing drone use in your specific region and identify any potential challenges or limitations related to QR code detection and flight operations.
- **Technical Infrastructure**: Explore the technological infrastructure required to support the semi-autonomous system, including communication protocols, flight planning software, and data management systems.

Scope:

- **Developing a QR code detection system**: This project focuses on the conceptual design and feasibility analysis of the detection system, not on building a delivery system.
- **Detailed drone design**: The project will not cover the specific design and engineering details of building a physical prototype for a delivery system.
- **Large-scale implementation**: The project will not deal with large-scale deployment or commercialization aspects of a drone system for delivery.

Deliverables:

- A comprehensive report outlining the **problem definition, proposed solution framework**, and key considerations for QR code detection.
- A **feasibility analysis** of the proposed semi-autonomous system.
- Recommendations for **future research and development** related to drone-based QR code detection.

Some additional points to consider:

- Identifying **target use cases** for the system, such as academic projects, industrial automation, or small-scale logistics.
- Addressing any **regulatory concerns** regarding drone safety and compliance in the project.

- Considering the **technical limitations** of the drone, such as **battery life** and flight stability due to the hardware load.

By clearly defining the **problem and scope** of the project (hover and scan) the focus will remain on the most relevant aspects of **semi-autonomous drone systems** using QR code detection.

Chapter 3: FLOW CHART

3.1-System Architecture

The drone architecture flow chart illustrates the operational framework of a quadcopter equipped with QR scanning technology. It outlines the sequence from initial flight preparation and GPS-based navigation to real-time QR code detection for accurate package identification. The system enables the drone to autonomously adjust its path and perform precise landings by integrating sensor data and QR code information, enhancing accuracy and efficiency in delivery tasks.

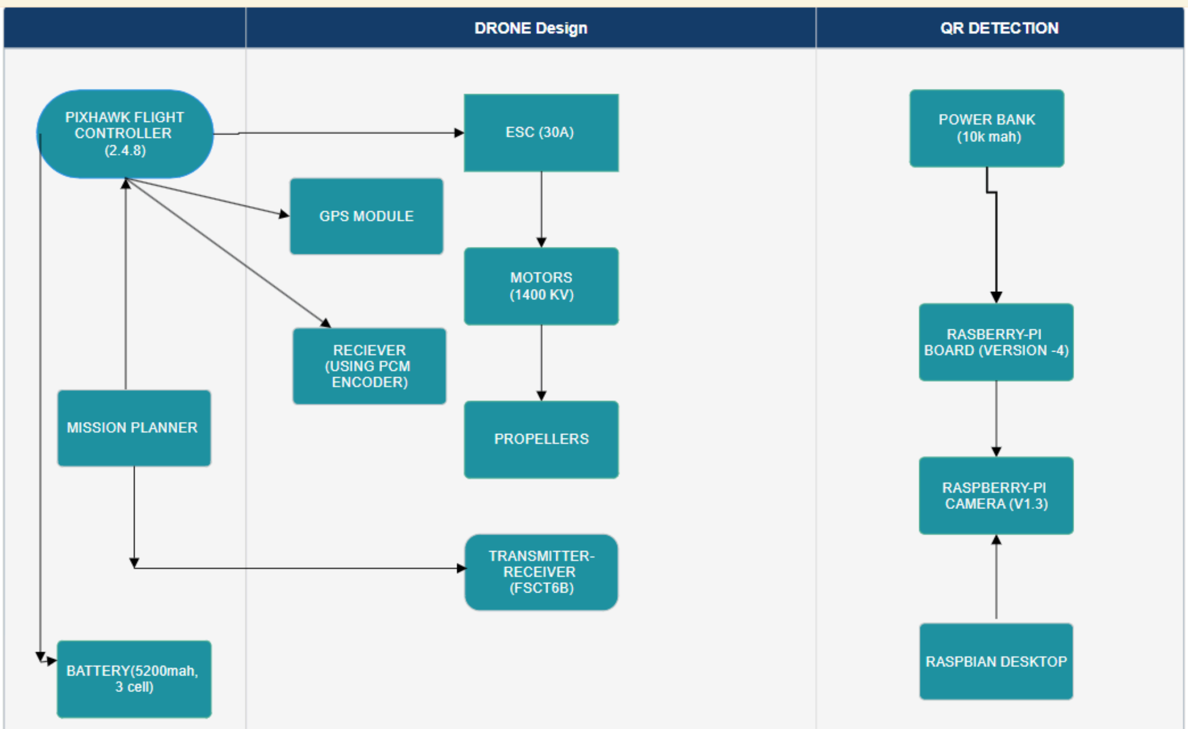

Fig 3.1 Drone System Architecture

The flow chart, fig 3.1, illustrates a drone design integrated with QR detection technology. It is divided into two main sections: **Drone Design** and **QR Detection**.

Components of the flow chart that mainly describes these two main sections are:

Drone Design	QR Detection

Pixhawk Flight Controller (2.4.8) GPS Module ESC (30A) Motors (1400 KV) Receiver (using PCM Encoder) Transmitter-Receiver (FSCT6B) Mission Planner Battery (5200mAh, 3 cell)	Power Bank (10,000 mAh) Raspberry Pi Board (Version 4) Raspberry Pi Camera (V1.3) Raspbian Desktop

Table 3.1 Main sections of the drone architecture flow chart

In operation, the drone's components work together for stable flight, while the Raspberry Pi system with a camera performs QR scanning for precise object identification and tracking during missions.

3.2-Analysis

The Semi-Autonomous Drone System aims to detect packages via QR codes while the drone hovers above the target. The flowchart highlights two major components of the project: **Drone Design** and **QR Detection**.

1. Drone Design

This segment is responsible for the flight operations and navigation of the drone. The system integrates essential hardware components to ensure reliable flight control and stable navigation.

- **Pixhawk Flight Controller (2.4.8)**: Acts as the central processing unit for the drone's flight dynamics and controls, interacting with sensors and other subsystems. It manages inputs from the GPS module, receiver, and mission planner.
- **GPS Module**: Provides precise positional data to ensure accurate navigation to the

designated area where the QR code will be detected.

- **Mission Planner**: A software tool that provides flight planning, configuration, and monitoring functions. It sends commands to the Pixhawk flight controller for autonomous navigation.
- **Battery (5200mAh, 3 cell)**: Supplies power to the drone's flight systems, ensuring the motors, flight controller, and other onboard electronics have sufficient energy for operation.
- **ESC (30A)**: The Electronic Speed Controller regulates the speed of the **Motors (1400 KV)** based on commands received from the Pixhawk. The motors provide thrust and lift for the drone to hover over the target location.
- **Propellers**: Paired with the motors, they generate the lift required for the drone's flight and stability.
- **Receiver (PGM Encoder)**: A critical component for communication, allowing the drone to receive commands and feedback. It connects to the transmitter-receiver (FSCT6B) for control functions, ensuring reliable communication during the mission.
- **Transmitter-Receiver (FSCT6B)**: This wireless communication system allows for manual intervention if necessary and provides telemetry data back to the base station.

QR Detection

This section focuses on detecting packages using QR codes. It involves the following components:

- **Raspberry Pi Board (Version 4)**: Handles the processing for the QR detection algorithms. It runs a Raspbian desktop environment where the QR code detection software operates.
- **Raspberry Pi Camera (v1.3)**: Captures real-time footage of the area beneath the drone. The camera is mounted on the drone and is essential for detecting the QR code on the package while the drone hovers.
- **Power Bank**: Provides dedicated power to the Raspberry Pi and camera, ensuring the QR detection system remains operational throughout the mission.
- **Raspbian Desktop**: The operating system on the Raspberry Pi where the QR detection

application, utilizing OpenCV, is run to identify the package.

Physical Components:

These components provide the physical capabilities for the drone to fly:

- **Motors and Propellers:** The brushless DC motors convert electrical energy from the battery into power that spins the propellers, generating lift for the drone.
- **Frame:** This lightweight structure holds all the components together and provides a platform for attaching the motors, propellers, and other parts.
- **Battery:** This provides the electrical power for all the drone's systems, including the motors, sensors, and flight controller. The size and capacity of the battery will impact the drone's flight time and range.

Communication:

- **Transmitter-Receiver (FSCT6B):** This radio control system allows for two main functionalities:
- **Manual Control:** A human operator can take over control of the drone in case of emergencies or unexpected situations.
- **Backup Communication:** This system might serve as a secondary communication channel between the drone and a base station in case the primary data link fails.

Key Challenges Addressed in the Design

- **Power Management**: The weight of the drone, powered by a 5200mAh battery, is an important consideration. The power bank ensures that the Raspberry Pi and the camera can operate independently of the drone's main battery, thus avoiding excessive strain on the flight system.
- **QR Code Detection**: The accurate identification of the QR code is handled by the Raspberry Pi's onboard processing, using OpenCV. The challenge here is ensuring that the drone hovers stably above the target to allow the camera to clearly capture the QR code.

- **Flight Control**: The Pixhawk controller ensures that the drone can autonomously hover above the designated area. The integration of the GPS module provides accurate navigation, allowing the drone to reach the target location.
- **Drone weight management**: The drone itself is heavy due to the battery and power bank used to power the Raspberry Pi and camera

Conclusion

This analysis presents a streamlined approach to package identification using QR codes. The system's design efficiently divides responsibilities between the **drone's flight operations** and the **QR code detection system**. The modular approach makes it easier to optimize both aspects independently, ensuring reliable package detection and flight control in this simple, college-level project.

3.3-Tools and Technologies Used

The breakdown of the essential tools and technologies required for the smart drone delivery system depicted in the block diagram fig:3.1

1. Hardware Components

- **Pixhawk Flight Controller (2.4.8)**: The Pixhawk is an open-source flight control system used to manage the drone's flight dynamics and autonomy. It interacts with various sensors and subsystems to maintain stability, handle navigation, and execute flight missions.
- **GPS Module (NEO-7M)**: This module provides accurate positional data, allowing the drone to autonomously navigate to specific coordinates and hover precisely over the target package.
- **ESC (Electronic Speed Controller - 30A)**: The ESC regulates the speed of the brushless DC motors, ensuring smooth flight control based on the commands from the Pixhawk.
- **Brushless DC Motors (1400 KV)**: High-thrust motors that power the propellers, providing the necessary lift and stability for the drone to hover and navigate accurately.
- **Propellers (10x4.5R)**: These are attached to the motors to generate lift and control the drone's direction and stability.

- **Battery (5200mAh, 3-cell LiPo)**: This high-capacity battery provides power to the flight system, including the Pixhawk controller, GPS, and motors, ensuring the drone can operate for extended periods.
- **Raspberry Pi Board (Version 4)**: The Raspberry Pi acts as the onboard computer for QR code detection. It processes data from the camera and executes the QR code detection algorithm.
- **Raspberry Pi Camera (v1.3)**: A lightweight camera mounted on the drone, used to capture real-time video for detecting QR codes on packages.
- **Power Bank**: Dedicated to powering the Raspberry Pi and its camera, ensuring that the drone's primary battery is not drained by additional electronics, thus optimizing flight time.
- **Transmitter-Receiver (FSCT6B)**: Used for communication between the drone and the ground station, allowing for telemetry and remote control, if required.

2. Software Tools

- **Mission Planner**: An open-source ground control station (GCS) software that provides mission planning, configuration, and telemetry functions. It allows the operator to set waypoints, monitor real-time flight data, and control the drone autonomously.
- **OpenCV (Open Source Computer Vision Library)**: A key technology used for the QR code detection system. OpenCV provides image processing and computer vision capabilities that enable the Raspberry Pi to accurately detect and identify QR codes during flight.
- **Raspbian OS**: The official operating system for the Raspberry Pi. It provides the environment for running the QR detection software and interacting with the camera and other peripherals.
- **Python**: The programming language used for developing the QR code detection algorithm and controlling the Raspberry Pi. Python, along with OpenCV libraries, ensures smooth integration of image processing functions.

3. Communication and Telemetry

- **Telemetry Radio (433 MHz)**: Facilitates communication between the drone and the ground station for real-time monitoring, mission updates, and telemetry data transfer during flight.
- **PGM Encoder**: An interface used to decode the receiver signals and convert them into a format that can be processed by the flight controller, ensuring smooth remote control and communication.

4. Analysis and Debugging Tools

- **Flight Logs and Telemetry Data**: Collected via the Pixhawk and Mission Planner, flight logs and telemetry data provide detailed information about the drone's performance, including GPS data, battery usage, motor speeds, and flight path accuracy. These logs are crucial for analyzing the drone's performance and making necessary adjustments for system optimization.

5. Development Environment

- **Visual Studio Code (VS Code)**: A powerful code editor used for writing and debugging the QR detection code on the Raspberry Pi. It provides a flexible development environment with support for Python and other languages used in the project.
- **Git/GitHub**: Version control tools used for managing code changes and collaboration. GitHub repositories store the project code and allow for efficient tracking of updates during the development phase.

Chapter 4: PROJECT DESIGN AND DESCRIPTION

4.1- Description

This chapter provides a comprehensive overview of the design and description of our delivery drone project. The project encompasses both hardware and software components, integrated to achieve a functional and efficient delivery drone capable of autonomous navigation and manual control.

Hardware Design:

- **Frame:** The drone uses an F450 quadcopter frame, known for its robustness and suitability for medium payload capacities. The frame provides a stable platform for mounting all necessary components.
- **Motors:** Four A2212 10T 1400KV brushless motors are utilized. These motors offer a good balance between power and efficiency, essential for the drone's lift and maneuverability.
- **Electronic Speed Controllers (ESCs):** Each motor is controlled by a 30A ESC, which regulates the power supply from the battery to the motors, ensuring smooth and responsive control.
- **Propellers:** The propellers are chosen to match the motor specifications, providing optimal thrust and efficiency.
- **Battery:** A 5200 mAh LiPo battery powers the drone, offering a balance between flight time and weight.
- **Landing Gear:** The landing gear provides stability during takeoff and landing, protecting the drone's components.

Control System:

- **Flight Controller:** The flight controller manages the drone's stability and navigation. It receives input from the receiver and sensors, processing this data to control the motors.
- **Transmitter and Receiver:** The transmitter allows manual control of the drone, while the receiver on the drone interprets these signals and relays them to the flight controller.

Computing and Sensing:

- **Raspberry Pi 4:** Serving as the main computing unit, the Raspberry Pi 4 handles complex tasks such as image processing and decision-making algorithms.
- **Raspberry Pi Camera v1.3:** The camera is used for object detection, allowing the drone to recognize and navigate around obstacles.

Software Design:

- **Operating System:** The Raspberry Pi 4 runs on Raspbian, a Debian-based operating system optimized for the Raspberry Pi hardware.
- **Object Detection:** OpenCV is used for real-time image processing and object detection, enabling the drone to autonomously navigate.
- **Programming Languages:** Python is primarily used for software development due to its extensive libraries and ease of use.

Fig 4.1 Real image of the Project Model (The Drone)

Fig 4.1 is a quadcopter drone, used in our project, with four propellers, GPS module, brushless motors, ESCs, and central battery for balance along with powerbank for the raspberry pi board, designed for outdoor delivery.

4.2- U.G. Subjects

The delivery drone project integrates knowledge and skills from various undergraduate subjects, providing a multidisciplinary approach to solving complex engineering problems. Key subjects include:

- **Electronics and Circuit Design:** This involves understanding electronic components, circuit design, and power management, along with practical skills in assembling and testing electronic circuits.
- **Mechanical Engineering:** Principles of aerodynamics, material science, and structural design are applied to ensure the stability and integrity of the drone's frame.
- **Computer Science and Programming:** Proficiency in programming languages, particularly Python, is essential for developing software algorithms and integrating various hardware components.
- **Control Systems:** Knowledge of control theory and its application in flight control systems, including the implementation of PID controllers for stable flight dynamics, is critical.
- **Wireless Communication:** Understanding wireless communication principles is necessary for the effective use of the transmitter and receiver, ensuring reliable control of the drone.
- **Image Processing and Computer Vision:** Techniques in image processing and computer vision, especially using OpenCV, are implemented for real-time object detection and navigation.

4.3- Standards used

To ensure the quality, safety, and interoperability of our delivery drone, we adhere to several industry standards and best practices:

IEEE Standards:

- **IEEE 1930.1-2022** - Recommended Practice for Software Engineering - Software Architecture - Part 1: Recommended Practices for Architectural Description of Software-Intensive Systems

  This standard provides guidelines for creating architectural descriptions of software-intensive systems. It covers various aspects of architectural modeling, including views, perspectives, and artifacts. By following this standard, organizations can improve the communication, understanding, and analysis of their software architectures.

- **IEEE 1936.1-2021** - Guide for Developing Software Measurement Plans

  This standard offers a comprehensive guide for developing effective software measurement plans. It covers key steps such as defining measurement objectives, selecting appropriate metrics, collecting and analyzing data, and reporting results.

- **IEEE 27001-2017** – Information Technology - Security techniques - Information security management systems – Requirements

  This international standard establishes a framework for implementing an information security management system (ISMS). It outlines a set of requirements for organizations to protect their sensitive information from various threats.

- **IEEE 1451** – A Suite of Smart Transducer Interface Standards

  This suite of standards defines a common interface for smart transducers, enabling seamless communication and interoperability between different devices. It covers various aspects of transducer interfaces, including physical, electrical, and communication characteristics. By adopting these standards, organizations can promote the development of innovative and compatible smart transducer technologies.

Chapter 5: IMPLEMENTATION AND EXPERIMENTAL RESULTS

The implementation phase of the smart autonomous drone system for QR detection focused on integrating hardware components, configuring the necessary software, and conducting tests to ensure the system could reliably detect and scan QR codes while hovering at a specified height. This section details the steps taken during the implementation and presents the results from the initial tests.

Hardware Integration:

1. **Frame Assembly:**
   - The F450 quadcopter frame was assembled, ensuring all components were securely mounted and balanced. The frame provided a stable platform to mount motors, electronic speed controllers (ESCs), the flight controller, and sensors.
   - The lightweight yet durable frame ensured the drone maintained structural integrity while carrying additional components such as the camera and QR detection sensors.
2. **Motor and ESC Installation:**
   - Four A2212 1400KV brushless motors were mounted on the quadcopter frame. The motors were connected to 30A ESCs, and proper calibration was conducted to ensure synchronized motor operation and smooth flight control.
   - The ESCs were linked to the flight controller to provide real-time power modulation based on flight conditions, ensuring stable hover during QR detection.
3. **Propeller Attachment:**
   - Propellers were selected based on the motor specifications, ensuring optimal thrust and efficiency. Proper alignment was confirmed to maintain balance and stability during flight and hovering. Each propeller was securely attached, ensuring that vibrations were minimized to avoid interference with the QR detection process.
4. **Power System Setup:**
   - A 5200mAh LiPo battery was integrated into the power distribution board,

supplying power to all drone systems, including the motors, flight controller, and camera.
    - A battery monitoring system was configured to track voltage and current consumption, ensuring sufficient power for both QR detection and safe return flights.
5. **Landing Gear Installation:**
    - Lightweight landing gear was attached to the frame, providing protection during takeoff and landing while keeping the drone stable during QR detection operations.

Control System Configuration:

1. **Transmitter and Receiver Binding:**
    - A FS-CT6B transmitter and receiver system was used for manual control of the drone. The transmitter was bound to the receiver, and channels were configured to control the drone's throttle, yaw, pitch, and roll.
    - Additional channels were assigned for camera control and flight mode switching (e.g., from manual to autonomous flight modes).
2. **Pixhawk Flight Controller Configuration:**
    - The Pixhawk flight controller was programmed with parameters specific to the drone's design, including motor calibration, ESC synchronization, and altitude control.
    - Altitude hold mode was configured to allow the drone to hover at a specified height during QR detection.

Software Development:

1. **Operating System Configuration:**
    - Raspbian OS was installed on the Raspberry Pi 4, which was mounted on the drone to process real-time video data. Python and OpenCV libraries were installed to handle image processing and QR detection tasks.
    - Other necessary packages and dependencies for interfacing with sensors and the flight controller were also configured.

2. **QR Detection Algorithm Development:**
    - Using the OpenCV library, a Python-based QR detection algorithm was developed. The algorithm processed the video feed from the Raspberry Pi Camera v1.3 in real-time, identifying and decoding QR codes on the packages.
    - The algorithm was optimized to ensure accurate QR detection even when the drone was hovering at varying heights or slight angles.
3. **Height Stabilization and Hovering Control:**
    - The altitude hold functionality was integrated with the QR detection system to ensure the drone maintained a constant height above the target package for optimal QR scanning.
    - The flight controller received altitude data from the ultrasonic sensor, allowing for minor adjustments to stabilize the drone during QR detection.

Testing and Experimental Results:

1. **Preliminary Flight Tests:**
    - Initial flight tests were conducted to verify the drone's ability to hover at a consistent height and to ensure stable flight during manual and autonomous operations. The altitude hold mode performed well, maintaining a steady height for extended periods.
    - Manual override functionality was tested to ensure smooth transitions between autonomous and manual control when necessary.
2. **QR Detection Testing:**
    - During preliminary tests, the drone successfully detected and decoded QR codes on static packages. The camera captured real-time video, and the QR detection algorithm processed the video feed to identify QR patterns at varying distances and angles.
    - Adjustments to the algorithm were made to improve detection accuracy under different lighting conditions and heights.
3. **Battery Performance and Power Management:**
    - Power consumption tests were conducted to ensure the battery provided sufficient flight time for completing QR scanning. The system operated efficiently, allowing

for extended hover times while preserving enough power for a safe return.

5.1-Simulation Results

Simulation was not extensively conducted for this phase of the project; however, basic functional tests and preliminary checks were performed to ensure system coherence and readiness for hardware implementation.

Preliminary Functional Tests:

- **Basic Stability Check:** Initial checks were conducted to ensure the drone's components, such as the motors, responded correctly to input commands in a simulated environment using basic control software.
- **Communication Verification:** Simulations verified the communication between the Raspberry Pi 4, ESCs, and ground control software, ensuring data was transmitted and received correctly.

Software Algorithm Verification:

- **Object Detection:** We tested the object detection algorithm. To verify its ability to identify and track objects before deploying it on hardware, we tested it using video feeds in a simulated environment on a Raspberry Pi 4.

5.2-Hardware Results

Following the preliminary simulation checks, hardware testing was conducted to validate the real-world performance of the drone's components and software integration. The following results were obtained from the hardware testing phase:

Motor and ESC Test:

- **Motor Operation:** Initial testing confirmed that each motor and ESC combination was functioning correctly. Motors responded to direct input commands from the ESCs, but full synchronization with the transmitter and receiver has yet to be tested.

Raspberry Pi and Camera Test:

- **System Boot:** The Raspberry Pi 4 booted correctly, and the camera provided a clear video feed. The setup was stable, and no connectivity issues were observed.
- **Object Detection:** Initial object detection tests were performed using the Raspberry Pi Camera v1.3 and OpenCV/Tensorflow. The system successfully identified and tracked objects in a controlled environment, demonstrating basic functionality of the detection algorithm.

Control System Test:

- **Transmitter and Receiver Binding:** The transmitter and receiver were successfully bound, and initial configuration was performed. However, comprehensive testing of motor control through the transmitter has not yet been completed.

QR Detection:

- **Basic Familiarization:** Initial exploration of the Mission Planner software was conducted to understand its features and capabilities related to **QR code detection**. This involved identifying the relevant modules, parameters, and configuration options for integrating QR code detection functionality into the drone's flight missions.

Chapter 6: OUTCOMES AND PROSPECTIVE LEARNING

6.1- Scope and Outcomes

Flight Controller and Navigation:

- **Scope:** Implement and configure the Pixhawk flight controller to enable stable autonomous flight with precise altitude control for QR code detection while hovering.
- **Outcome:** The Pixhawk flight controller was successfully configured, providing reliable altitude hold and smooth flight control. This allowed the drone to maintain a stable hover, which is critical for accurate QR code scanning. The successful integration ensures that the drone can autonomously navigate to a designated location and hover at a set height for detection tasks.

Motor & Communication System:

- **Scope:** Evaluate and integrate the motor system and communication system (FS-CT6B) for reliable flight control and manual overrides.
- **Outcome:** The motor system (A2212 1400KV brushless motors) and ESCs were successfully connected, calibrated, and tested, providing efficient thrust for stable flight. The transmitter-receiver system was bound and tested, allowing for seamless communication and manual control. This ensured that the drone could be manually overridden during critical moments while also functioning autonomously.

QR Code Detection Algorithm:

- **Scope:** Develop and test the QR code detection algorithm for accurate scanning of packages while the drone hovers at a specific height.
- **Outcome:** The QR detection algorithm was developed using OpenCV and Python and tested successfully. The system accurately detected and decoded QR codes in real-time, even while the drone was hovering at varying heights. This outcome demonstrated the reliability of the drone’s ability to scan packages autonomously, a crucial function of the project.

Raspberry Pi Camera:

- **Scope:** Integrate and validate the Raspberry Pi Camera v1.3 for real-time image capture and QR code scanning during flight.
- **Outcome:** The Raspberry Pi Camera v1.3 was successfully integrated and tested, capturing clear images and video in real-time. The camera's video feed provided the necessary visual input for the QR detection algorithm. This allows for precise QR code identification, ensuring that the drone accurately identifies packages from the air.

Raspberry Pi Upgrade:

- **Scope:** Upgrade to Raspberry Pi 4 for improved processing power to handle real-time QR detection and control tasks.
- **Outcome:** The Raspberry Pi 4 was integrated, significantly boosting the computational capacity required for processing real-time video feeds and running the QR detection algorithm. The enhanced processing power improved system performance, making the drone more efficient in executing autonomous tasks such as image processing and sensor data analysis.

Power and Battery Management:

- **Scope:**
  Evaluate battery life and optimize power consumption to ensure sufficient flight time for autonomous QR detection missions. Additionally, ensure that the Raspberry Pi, responsible for processing real-time video and running the QR detection algorithm, has a reliable power source independent of the main flight battery.
- **Outcome:**
  The 5200mAh LiPo battery was tested, powering the drone's flight system, motors, and sensors. Power management systems were implemented to optimize energy consumption, ensuring that the drone could carry out both QR detection and return trips with sufficient battery life. Additionally, a portable power bank was integrated to power the Raspberry Pi 4 independently, ensuring uninterrupted operation of the onboard computer for

real-time video processing and QR code detection. This dual power setup allowed the drone to maintain stable operation without overloading the main flight battery, contributing to extended flight times and overall system reliability.

6.2- Prospective Learning

i. Enhanced Flight Control: Achieving stable QR code detection from varying altitudes will require enhanced flight control through sensor fusion, incorporating data from cameras, sensors and mission planner. This integrated approach will provide superior flight control and stability, particularly under adverse conditions such as high wind or rain. Advanced algorithms implemented by mission planner and complementary filters will merge sensor data, giving precise estimates of the drone's position and orientation while it hovers or adjusts altitude. Additionally, vision-based landing techniques will be employed for accurate payload delivery by detecting and locking onto landing zones, adjusting for uneven terrain or moving targets as needed.

ii. Regulatory Navigation: This project prioritizes compliance with regulations for safe and legal flight operations. Adhering to airspace restrictions and payload limits, the project also explores future opportunities for Beyond Visual Line of Sight (BVLOS) operations. Developing Sense & Avoid Systems (SAS) will ensure the drone remains within regulatory standards for autonomous flight. By understanding and meeting current regulatory requirements, the project lays the groundwork for more advanced QR code detection capabilities and autonomous flight operations in the future.

iii. Integration and Knowledge Acquisition: By addressing these challenges, the project will gain valuable experience in integrating various hardware and software components for effective QR code detection by a drone from a height. This includes ensuring real-time QR code scanning accuracy while the drone hovers or moves at different altitudes, along with maintaining superior flight control and balancing the drone's range, stability, and safety during flight. The integration of technologies such as machine learning for image recognition, sensor fusion for precise positioning, and advanced navigation systems will provide practical insights into optimizing system reliability and performance at height. This experience will not only enhance the current project but also prepare the team for future ventures in autonomous aerial systems and the rapidly advancing field of drone technology.

iv. QR Code Detection: QR code detection from a height poses unique challenges, as it requires the drone to maintain precision and stability while scanning packages from above. The project integrates high-resolution cameras and real-time image processing algorithms to ensure accurate QR code detection even at varying altitudes. The system uses advanced vision techniques, including object tracking and image stabilization, to focus on the QR code during flight, compensating for any drone movement or environmental factors. Machine learning models will enhance detection accuracy by improving the system's ability to recognize QR codes under different lighting conditions, angles, and distances. This ensures that package identification is reliable, whether the drone is hovering or approaching for delivery.

6.3- Conclusion

We explored the outcomes and prospective learnings derived from our QR code detection by drone project. We achieved several key milestones, including the theoretical analysis of Mission Planner software, the successful evaluation of the motor and communication system, the integration of the Raspberry Pi Camera version 1.3, and the upgrade to a Raspberry Pi model 4. These advancements laid a solid foundation for the project's technical capabilities and operational potential, particularly for accurate QR code scanning from specific heights during drone flight.

The scope of our project extended beyond basic implementations, delving into advanced QR code detection at varying altitudes, enhanced flight control, and regulatory navigation. We emphasized the importance of differentiating between moving obstacles using various techniques, combined with machine learning for real-time obstacle recognition. By integrating sensor data through advanced algorithms, we aimed to achieve superior flight control and stability, ensuring that the drone can accurately detect QR codes from above, even under challenging environmental conditions. Our focus on regulatory navigation ensured that safety and compliance remained at the forefront, preparing the project for future autonomous flight operations.

Through these efforts, we gained invaluable experience in integrating diverse hardware and software components, honing our skills in real-time QR code detection, flight control optimization, and regulatory adherence. This knowledge equips us not only for the successful completion of the current project but also for future endeavors in the rapidly evolving field of autonomous systems and drone technology.

Our project represents a significant step forward in the development of sophisticated drone-based QR code detection systems. The learnings and insights gathered will contribute to the broader field of robotics and autonomous vehicles, enhancing the capabilities and safety of these systems in various applications. By continuously pushing the boundaries of what is possible, we aim to lead the way in the next generation of drone technology, ensuring that it meets the highest standards of performance, safety, and reliability.

Chapter 7: PROJECT TIMELINE

7.1- Work breakdown and Gantt Chart

Our work primarily focuses on the research component at the beginning, alongside different sources, articles, and research papers during . Later, we dedicate our time to Mission Planner, exploring and learning about its various features. We then assemble the drone and test the proper functioning of the motors and ESC (Electronic Speed Controller).

Sr. No.	Task	Jan	Feb	March	April	May	June	July	Aug	Sep	Oct	Nov
1	**Analysis/Research**											
2	**OS Integration**											
3	**Object Detection**											
4	**Manual Controller**											
5	**Drone Assembly**											
6	**Pixhawk**											
7	**Algorithm Analysis**											
8	**Mission Planner Integration**											
9	**QR Scan**											
10	**Stabilization and testing**											

	indicates the further analysis continued		indicates the initial or the start

Table 7.1 Project breakdown and Gantt Chart

7.2- Project Timeline

DURATION	TASKS	DESCRIPTION
JAN-FEB 2024	Analysis/Research	Research-based on the project
MARCH-APRIL, 2024	OS Integration	Updating and installing the required software
	Object Detection	Integrating sensors and algorithms
MAY, 2024	Manual Controller	Calibration and connection
	Drone Assembly	Integrating components.
AUGUST, 2024	Pixhawk	Final assembly of the drone
SEPTEMBER, 2024	Algorithm Analysis	QR Detection Algorithms
	Mission Planner Integration	Configuration of hardware and software components
OCTOBER, 2024	QR Scan	Hovering and scanning qr on packages
	Stabilization	Drone stabilization with payload
	Testing	Final testing

Table 7.2 Project timeline

REFERENCES

[1] ***M. Moshref-Javadi and M. Winkenbach,*** *"Applications and Research Avenues for Drone-Based Models in Logistics: A Classification and Review," Expert Systems with Applications, vol. 177, pp. 114131, Apr. 2021. https://www.sciencedirect.com/science/article/abs/pii/S0957417421002955*

[2] ***M. Dolata and G. Schwabe,*** *"Moving Beyond Privacy and Airspace Safety: Guidelines for Just Drones in Policing," Expert Systems with Applications, vol. 211, pp. 118542, Nov. 2023.https://www.sciencedirect.com/science/article/pii/S0740624X23000746*

[3] ***J. Wu, Y. Ye, and J. Du,*** *"Multi-Objective Reinforcement Learning for Autonomous Drone Navigation in Urban Areas with Wind Zones," Expert Systems with Applications, vol. 215, pp. 119017, Dec. 2023. https://www.sciencedirect.com/science/article/abs/pii/S0926580523005137*

[4] ***HH. D. Yoo and S. M. Chankov,*** *"Drone-Delivery Using Autonomous Mobility: An Innovative Approach to Future Last-Mile Delivery Problems," in Proc. IEEE, Bremen, Germany: IEEE, pp. 1–10. https://ieeexplore.ieee.org/abstract/document/8607829?casa_token=IFKtJcHmMBgAAAAA:5G9lFxyUG7Kk4NJzFJDwF6qwuKU0ejTXCk6nhydy92xeRMC4rEl0g3tIGTLgfjiI9CtxHkaOMoaC*

[5] ***V. R. F. Miranda, A. M. C. Rezende, T. L. Rocha, H. Azpúrua, L. C. A. Pimenta, and G. M. Freitas,*** *"Autonomous Navigation System for a Delivery Drone," Journal of Control, Automation and Electrical Systems, vol. 33, pp. 141–155, Oct. 2021. https://link.springer.com/article/10.1007/s40313-021-00828-4*

[6] ***S. Xia, J. Guo, and C. Peng,*** *"SSS: Towards Autonomous Drone Delivery to Your Door Over House-Aware Semantics," in Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications (HOTMOBILE '24), pp. 33–39, Feb. 2024. https://dl.acm.org/doi/abs/10.1145/3638550.3641129*

[7] ***P. Sachdeva, J. Kaur, A. Huhn, and T. Schwotzer,*** *"Open-Source, Decentralized Autonomous Drone-Based Delivery System," in Proceedings of the International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2022), pp. 523–528, Nov. 2022. https://link.springer.com/chapter/10.1007/978-3-031-21333-5_52*

[8] ***E. Yeh, J. Mohanroy, C. Ciobanu, and C. Scrosati,*** *"Autonomous Food Delivery Drone," SOECS Senior Project Demonstration, University of the Pacific, Stockton, CA, USA, Jun. 2023. https://scholarlycommons.pacific.edu/purcc/2023/events/81/*

[9] ***B. Shahzaad, B. Alkouz, J. Janszen, and A. Bouguettaya,*** *"Optimizing Drone Delivery in Smart Cities," IEEE Internet Computing, vol. 27, no. 4, pp. [page numbers], Jul.–Aug. 2023. https://ieeexplore.ieee.org/abstract/document/10105302?casa_token=qz70TA4sGHAAAAAA:PqbJ8CaXnRKqPJayPBlK-eZeqhp_5EzbcLeFss089Jw3xZTEj_sWCfeivyQDcg9M1vwKluSfZjLQ*

[10] ***A. M. Raivi, S. M. A. Huda, M. M. Alam, and S. Moh,*** *"Drone Routing for Drone-Based Delivery Systems: A Review of Trajectory Planning, Charging, and Security," Sensors, vol. 23, no. 3, p. 1463, Jan. 2023. https://www.mdpi.com/1424-8220/23/3/1463*

[11] ***B. Gerrits and P. Schuur,*** *"Parcel Delivery For Smart Cities: A Synchronization Approach For Combined Truck-Drone-Street Robot Deliveries," in Proc. 2021 Winter Simulation Conf. (WSC), Dec. 2021. https://ieeexplore.ieee.org/abstract/document/9715506?casa_token=obUSSIlDHH8AAAAA:nz0I7O1eqPXm2dPxXHd5QD1JY2YuZYQ1u6BXhngzFFDs8zuiGnyI01P4OiCBX1L5twbkfsEIybvT*

[12] ***E. Marquez and I. Luna****, "Modelling of a Drone to Analyze Dynamic Instabilities With its Delivery System," in Proc. ASME 2023 Int. Mechanical Engineering Congress and Exposition (IMECE), New Orleans, LA, USA, Oct. 29–Nov. 2, 2023, paper no. IMECE2023-116608, V004T05A001, pp. 1–6. https://asmedigitalcollection.asme.org/IMECE/proceedings-abstract/IMECE2023/87615/1195727*