People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Saad Dahlab University - Blida 1

Institute of Aeronautics and Space Studies

Aeronautical Construction Department

Manuscript

Towards the attainment of the Master's degree in

Aeronautics Engineering

Option: **Avionics**

Development Of Flight Controller For Drones By Using MCU

The startup project presented as part of the "1275" ministerial decree, provided by the Institute of Aeronautics and Space Studies.

Submitted by

- Nahr El Imene DORBANI
- Sara ZOUAOUI

Supervised by

- Senior Lect. Redha AMRI
- Pr. Mohaned LAGHA

Economic Partner:

- Mr. Ahmed KECHIDA

Jury Members:

- Mr. Mohamed KRIM — President
- Mr. Abdelhalime BENOUARED — Examiner
- Mr. Mohamed MEKARZIA — Examiner

2024/2025

Dedication

To my dearest parents, whose unwavering love, sacrifices, and constant encouragement have been my guiding light. Your faith in me, even in times of doubt, has been the foundation of my perseverance and resilience. I am forever indebted to you for your boundless support and belief in my dreams.

To my beloved sister and brother, Hadjer and Adem, who have not only been my companions but also my sources of strength and laughter. Your endless patience, understanding, and unshakeable belief in my abilities have lifted me in moments of discouragement. You have both inspired me to push forward . This thesis is as much yours as it is mine.

To my cherished grandparents, whose wisdom, kindness, and prayers have enveloped me with love and support from near and far. Your example of strength and grace has been a quiet force that has carried me through.

To my entire family, whose collective love and encouragement have formed the bedrock of my journey. Your pride in my accomplishments has given me the motivation to strive for excellence.

To my friends, whose companionship and shared moments of joy and struggle have made this journey not only bearable but memorable. Your support has meant the world to me.

To my professors and mentors, whose guidance, patience, and wisdom have inspired me and shaped my academic path.

With all my heart, I dedicate this thesis to each of you. This achievement is a reflection of the love, support, and inspiration you have all given me.

Nahr EL Imene

To my parents, an inexhaustible source of love, patience, and sacrifice, you are the foundation upon which I have built every step of this journey. Thank you for believing in me, even in my silences.

To my brothers, constant figures of strength and support, companions through laughter and quiet moments, your presence has been a quiet yet powerful encouragement. Thank you for your protection, your strength, and your unique way of lifting me up without needing words. You are an essential part of everything I've accomplished.

To my sisters, gentle and radiant souls, those whom life gifted me and who have become an inseparable part of who I am. We may not share the same blood, but we are bound by soul, by sincere love, shared tenderness, and a constant presence that carried me through my most fragile moments. Thank you for your affection, your strength, your laughter, and your silences. You are not just sisters at heart you are my family.

To my teachers, those generous souls who went beyond their roles to support me with rare kindness. To you who stood by me, more than mentors at times, more than family. I owe you far more than knowledge.

To my entire family, each of you, in your own way, has been a light, a breath, or a shoulder to lean on. To those who carried me when my strength was fading, to those who motivated me without even knowing sometimes with a simple word thank you, deeply.

To my friends, the true ones, the rare ones, the constants, thank you for reminding me I was never alone, even in the hardest times.

To all of you, this thesis is a little piece of you, a little piece of me, and a great deal of what we shared.Thank you from the bottom of my heart.

Sara

Acknowledgment

In the name of Allah, the Most Gracious, the Most Merciful

We begin by expressing our deepest gratitude to Allah Almighty, whose blessings, guidance, and strength have enabled us to carry out and complete this thesis.

We extend our sincere appreciation to our supervisors, **Mr. AMRI Redha** and **Mr. LAGHA Mohand**, for their exceptional guidance, invaluable advice, and continuous support throughout the course of this work. Their scientific expertise and constructive remarks have been essential to the quality and development of this research.

We also thank the members of the university and staff of the Institute Of Aeronautics and Space Studies for providing a stimulating academic environment and the resources necessary for the successful realization of this project.

We would like to express our sincere gratitude to **Mr. REZZOUK**, head of the Laboratoire des Sciences Aéronautiques, for his availability and the facilities he made accessible to us throughout our work.

We are especially grateful to our families for their unwavering encouragement, patience, and moral support during the entirety of our academic journey. Their presence has been a source of strength and motivation throughout.

Lastly, we wish to acknowledge the contributions of our fellow students, and friends who through their collaboration and insightful discussions, have positively influenced the outcome of this research.

To all those who supported us in one way or another, we extend our heartfelt thanks.

Abstract

This study focuses on the development of a flight controller for drones, utilizing both classic PIC microcontrollers and dsPICs to optimize performance. The main objective is to design and implement an embedded system capable of ensuring stable and precise control of a drone in flight.

The approach includes the integration of inertial sensors (accelerometers, gyroscopes) through the I^2C communication protocol, the implementation of PID control loops, and programming the microcontrollers using MikroC and MikroC Pro for dsPIC. Data acquisition and interaction with sensors were handled via I^2C and UART, ensuring accurate and synchronized measurements essential for flight stabilization. Tools such as SerialPlot, Docklight, and Advanced Serial Port Terminal were used for real-time data visualization and analysis during the testing phases.

The project combines modeling, simulation, and hardware/software implementation, with successive tests conducted to validate the controller's stability and responsiveness. The results demonstrate that the developed system effectively stabilizes the drone's main axes while addressing real-time processing constraints and precise sensor calibration.

This study highlights the complementarity between PIC and dsPIC microcontrollers in meeting the demands of complex embedded systems, providing a concrete contribution to the development of compact, reliable, and scalable flight controllers applicable in aeronautics and robotics.

Résumé

Cette étude présente le développement d'un contrôleur de vol pour drones, utilisant à la fois des microcontrôleurs PIC classiques et des dsPIC afin d'optimiser les performances. L'objectif principal est de concevoir et de mettre en œuvre un système embarqué capable d'assurer un contrôle stable et précis d'un drone en vol.

L'approche comprend l'intégration de capteurs inertiels (accéléromètres, gyroscopes) via le protocole de communication I^2C et UART, la mise en œuvre de boucles de régulation PID et la programmation des microcontrôleurs avec MikroC et MikroC Pro pour dsPIC. L'acquisition des données et l'interaction avec les capteurs ont été gérées via I^2C, garantissant des mesures précises et synchronisées, essentielles à la stabilisation du vol. Des outils tels que SerialPlot, Docklight et Advanced Serial Port Terminal ont été utilisés pour la visualisation et l'analyse des données en temps réel pendant les phases de test.

Le projet combine modélisation, simulation et implémentation matérielle/logicielle, avec des tests successifs pour valider la stabilité et la réactivité du contrôleur. Les résultats démontrent que le système développé stabilise efficacement les axes principaux du drone tout en répondant aux contraintes de traitement temps réel et de calibration précise des capteurs.

Cette étude met en évidence la complémentarité des microcontrôleurs PIC et dsPIC pour répondre aux exigences des systèmes embarqués complexes, apportant ainsi une contribution concrète au développement de contrôleurs de vol compacts, fiables et évolutifs, applicables en aéronautique et en robotique.

ملخص

تركز هذه الدراسة على تطوير وحدة تحكم طيران للطائرات المسيرة، باستخدام كل من متحكمات dsPIC الكلاسيكية ووحدات PIC لتحسين الأداء. الهدف الرئيسي هو تصميم وتنفيذ نظام مدمج قادر على ضمان تحكم مستقر ودقيق في الطائرة أثناء الطيران.

يشمل النهج دمج أجهزة استشعار بالقصور الذاتي (مقاييس التسارع، الجيروسكوبات) عبر بروتوكول اتصالات I²C و UART، وتنفيذ حلقات تحكم باستخدامMikroC Pro و MikroC وبرمجة وحدات التفاعل مع المستشعرات عبر I²C، مما يضمن قياسات دقيقة ومتزامنة ضرورية لاستقرار الطيران. استُخدمت أدوات مثل Advanced Serial Port Terminal و Docklight و SerialPlot لتصور البيانات وتحليلها في الوقت الفعلي خلال مراحل الاختبار.

يجمع المشروع بين النمذجة والمحاكاة وتطبيق الأجهزة/البرمجيات، مع إجراء اختبارات متتالية للتحقق من استقرار وحدة التحكم واستجابتها. تظهر النتائج أن النظام المطور نجح في تحقيق استقرار المحاور الرئيسية للطائرة بدون طيار بفعالية، مع معالجة قواعد المعالجة الآلية ومعايرة المستشعر بدقة.

تسلط هذه الدراسة الضوء على التكامل بين متحكمات dsPIC و PIC في تلبية متطلبات الأنظمة المدمجة المعقدة، مُقدمة مساهمة ملموسة في تطوير متحكمات طيران مدمجة وموثوقة وقابلة للتطوير، قابلة للتطبيق في مجال الطيران والروبوتات.

Contents

List of Acronyms

ADC	Analog-to-Digital Converter
AI	Artificial Intelligence
ALU	Arithmetic Logic Unit
ARM	Advanced RISC Machine
AVR	Advanced Virtual RISC
CAN	Controller Area Network
CCP	Capture/Compare/PWM
CU	Control Unit
DMP	Digital Motion Processor
DSP	Digital Signal Processing
EEPROM	Electrically Erasable Programmable Read-Only Memory
EXO	Exo Start Spherical
FC	Flight Controller
FPGA	Field-Programmable Gate Array
GPS	Global Positioning System
HAL	Hardware Abstraction Layer
HDL	Hardware Description Language
I/O	Input/Output
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit

LET	Lethal UAV
LR	Long Range
MALE	Medium Altitude Long Endurance
MCU	Microcontroller Unit
MPC	Model Predictive Control
MRAC	Model Reference Adaptive Control
MPLAB	Microchip Laboratory
MSP	Microcontroller Sensor Processor
P.A.E	Payload, Altitude, Endurance
PID	Proportional-Integral-Derivative
PIC	Peripheral Interface Controller
PIR	Passive Infrared
PPO	Proximal Policy Optimization
PWM	Pulse Width Modulation
RAM	Random Access Memory
RL	Reinforcement Learning
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
RTOS	Real-Time Operating System
SPI	Serial Peripheral Interface
SR	Short Range
STARTO	Stratospheric UAV
UART	Universal Asynchronous Receiver-Transmitter
UAV	Unmanned Aerial Vehicle
UAVs	Unmanned Aerial Vehicles
USB	Universal Serial Bus
VTOL	Vertical Takeoff and Landing

List of Figures

List of Tables

General Introduction

In recent years, the use of Unmanned Aerial Vehicles (UAVs), commonly known as drones, has expanded rapidly across various industries, including agriculture, logistics, surveillance, and entertainment. Among different types of UAVs, quadcopters have gained significant popularity due to their simplicity in design, vertical takeoff and landing capabilities, and high maneuverability. The core component that enables stable and autonomous flight in these vehicles is the flight controller, which serves as the system's brain by processing sensor data and controlling the propulsion system accordingly.[1] [2]

The flight controller relies on a microcontroller unit to execute control algorithms, process real-time sensor inputs, and manage communication protocols. With advancements in microcontroller technology, it is now feasible to develop efficient, low-cost, and high-performance flight controllers tailored to specific applications. This study focuses on the design and development of a flight controller using a PIC microcontroller, integrating essential sensors and implementing control strategies to ensure stability and precise maneuvering of a quadcopter.[3]

The structure of the present work is outlined as follows:

The first chapter provides a comprehensive overview of UAV systems, focusing on the classification of drones based on various criteria such as wing arrangement, engine type, and wings. Special attention is given to quadcopters, a subclass of multirotor UAVs characterized by their four-rotor configuration, which offers advantages in terms of mechanical simplicity and maneuverability. The chapter delves into the architecture of flight control systems, discussing the various types of flight controllers and their significance in ensuring stable and efficient flight. It also explores traditional and advanced stability methods, including classical control techniques, sensor fusion-based methods, and the incorporation of Artificial Intelligence (AI) and Machine Learning (ML) for enhanced flight stability.

The second chapter introduces the Peripheral Interface Controller (PIC) microcontrollers, highlighting their architecture, features, and suitability for embedded flight control applications. The discussion includes a comparison with other microcontroller families such as AVR, ARM, and FPGA, emphasizing the trade-offs in terms of processing power, energy efficiency, and development complexity.

The third chapter explores the development tools associated with PIC microcontrollers, including compilers like MPLAB and MikroC, which facilitate firmware development. It also covers the integration of sensors, focusing on the MPU-6050, a 6-axis MotionTracking device that combines a 3-axis gyroscope and a 3-axis accelerometer, essential for capturing the dynamic motion of UAVs.

At last, we offer a broad conclusion to wrap up our work.

Chapter 1

Drone Systems

1.1 Introduction

Drones are autonomous or remotely controlled multipurpose aerial vehicles driven by aerodynamic forces and capable of carrying a payload [4]. The word drone was initially used for the male bee, but was also introduced for aerial vehicles in the 1920s for the remote-controlled lightweight target aircraft Fairy Queen, which made a loud humming sound similar to that of a male bee [4][5].

Although the US Federal Aviation Administration adopted the term "unmanned aircraft vehicle/system" (UAV/UAS) in 2005 [6], the term "drone" is still preferred in peer-reviewed medical literature.

The performance of a drone is measured by its ability to react to various environmental factors, such as aerodynamic phenomena and disturbances (like wind). With advancements in technology in the field of electronics and embedded systems, the manufacturing of drones of different sizes and the simultaneous integration of multiple types of sensors have become possible. These sensors help minimize human intervention in complex and hazardous missions.

1.2 UAV Classification

1.2.1 Classification according to W.A.E

An interesting classification was proposed by the European Association of Unmanned Vehicle Systems (EUROUVS). This classification is based on the following parameters, referred to as W.A.E: maximum takeoff weight, maximum flight altitude, and endurance. [7]

Category	**Acronym/Type**	**Max Takeoff Weight (kg)**	**Max Flight Altitude (km)**	**Endurance (H)**	**Example**
Micro/ Mini Drones	Mav	0.10	0.25	1	MicroStar
	Mini	>30	0.15–0.3	<2	Mikado
Tactical Drones	Close Range (CR)	150	3	2–4	Phantom
	Short Range (SR)	200	3	3–6	Luna
	Medium Range (MR)	150–500	3–5	6–10	Hunter B
	Long Range (LR)	To be determined	5	6–13	Vigilante 502
	Endurance (EN)	500–1500	5–8	12–24	Aerosonde
	Medium Altitude Long Endurance (MALE)	1000–1500	5–8	24–48	Skyforce
Strategic Drones	High Altitude Long Endurance (HALE)	2500–125000	15–20	24–48	Condor
Specialized Drones	Lethal (LET)	250	3–4	3–4	Harpy
	Stratospheric (STARTO)	To be determined	20–30	> 48	Pegasus
	Exo Start Spherical (EXO)	To be determined	> 30	To be determined	Mac-1

Table 1.1: UAV Classification

1.2.2 Classification by Engine Type

UAVs utilize various engine types depending on their specific mission requirements. The most common engine types include turbofans, two-stroke, piston, rotary, turboprop, push-and-pull, electric, and propeller engines. Among these, electric and piston engines are the most frequently used in the UAVs examined in this project.

As in most aeronautical applications, an increase in aircraft weight necessitates a larger engine size, a trend that also applies to UAVs.

Smaller and lighter UAVs typically rely on electric motors, whereas heavier, combat-ready UAVs often use piston engines. The type of engine in a UAV significantly impacts its endurance and range. Selecting the appropriate engine enhances both performance factors, ensuring optimal efficiency. [8]

1.2.3 Classification by Wing Type

1.2.3.1. Flapping-Wing UAV

The control and piloting of this type of drone rely on flapping wings. Inspired by insects, these drones can perform low-speed hovering flights and operate in very confined spaces, making them suitable for specialized missions.

Figure 1.1: Flapping-wing drone

1.2.3.2. Fixed-Wing UAV

Fixed-wing drone features a rigid wing structure similar to conventional aircraft. It utilizes its extended wings to generate lift during flight, allowing for greater energy efficiency and the ability to cover vast distances. [9]

Figure 1.2: Fixed-Wing UAV

The Categories of Fixed-Wing UAVs are:

- **Large Fixed-Wing UAVs**: These drones typically offer high endurance, long-range capabilities, and a substantial payload capacity, making them suitable for demanding missions.

- **Medium Fixed-Wing UAVs**: While they maintain high endurance and long-range performance, they have a lower payload capacity compared to larger models. This category is among the most widely available fixed-wing UAV sizes on the market today.

- **Small Fixed-Wing UAVs**: Despite their ability to perform long-endurance and long-range flights, their limited wing area restricts their payload capacity, making them less suited for heavy-lift operations. [9]

1.2.3.3. Hybrid/VTOL UAV

Hybrid UAVs, also known as Vertical Takeoff and Landing (VTOL) UAVs, combine the advantages of both fixed-wing and rotary-wing aircraft. They can take off and land vertically like helicopters but transition to fixed-wing flight for greater efficiency in cruise mode. These UAVs are useful for missions requiring versatility in both hover and long-range flight.[10]

Figure 1.3: Hybrid/VTOL UAV

1.2.3.4. Rotary-Wing Drones

These drones can take off, fly, and land vertically. They rely on rotary wings or multiple rotors to hover in place, eliminating the need for a runway. This capability allows them to perform a wide range of missions that fixed-wing drones cannot accomplish. [11]

Classification of Rotary-Wing Drones

Rotary-wing drones can be categorized into four types:

- **Mono-Rotor Drones** are equipped with a main rotor that facilitates takeoff and landing. They are divided into three subcategories based on their configuration, which determines their movement along different axes (both translational and rotational) and prevents them from spinning uncontrollably due to the reaction torque of the main rotor.

Figure 1.4: Mono-Rotor Drone

- **Birotor Drones** are equipped with two rotors that spin in opposite directions at the same speed, canceling out torque effects. They are classified into three subcategories based on their configuration, which defines how they maneuver along different axes:

Figure 1.5: Birotor Drones

  - Cyclic and collective pitch birotors: These drones adjust their position by varying the cyclic and collective pitch of the blades.
  - Birotors with additional propellers to enhance stability and control.
  - Birotors with control surfaces, such as flaps, for improved maneuverability.

- **Tri-rotor drones** are designed with two front rotors rotating in opposite directions to control pitch, while a rear rotor manages roll. Although they operate similarly to quadrotors, their flight performance is generally less effective.

Figure 1.6: Tri-Rotor Drone

- **Multi-Rotor Drones** These aerial robots are typically equipped with four rotors, but variations exist with six or even eight rotors.[12]

Figure 1.7: Examples of multi-rotor drones

1.2.4 Quadcopter

A quadcopter is a type of rotary-wing aircraft that uses four rotors for lift. These rotors are usually positioned at the ends of a cross-shaped frame. To prevent unwanted rotation around the yaw axis, two propellers spin in one direction while the other two rotate in the opposite direction. For effective control, each pair of propellers spinning in the same direction must be placed at opposite ends of the cross arms. [13]

1.2.4.1. Quadcopter Configurations

There are many configurations. These are the most commonly used in industry and research, offering a good balance between stability, control dynamics, and practical application.[14]

- X-configuration: The most common setup, where the front and rear rotors form an "X" shape relative to the frame.
- Plus configuration: Similar to X, but the front rotor aligns with the forward direction.
- Coaxial quadcopters: Each arm has two stacked rotors spinning in opposite directions to increase lift.

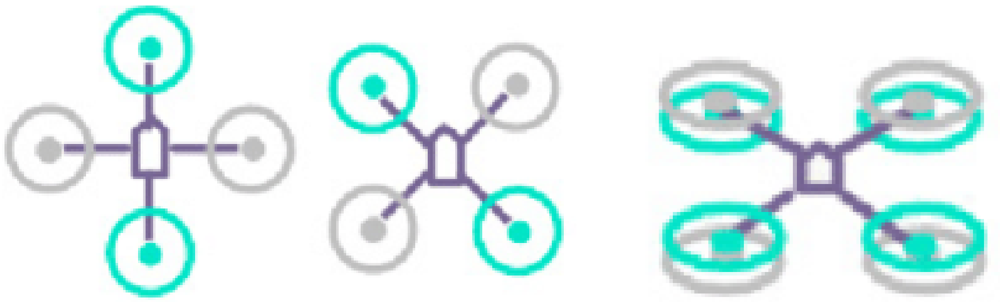

Figure 1.8: Examples of Quadcopter Configurations

1.2.4.2. Basic Principles of Quadcopter Flight

- Thrust and Lift The rotors generate lift by spinning at high speeds, creating downward airflow.
- Yaw Control By spinning two rotors clockwise and the other two counterclockwise, torque effects cancel out, allowing yaw control through differential rotor speeds.
- Pitch and Roll Tilting the quadcopter forward, backward, or sideways is achieved by increasing or decreasing the speed of specific rotors.[15]

1.2.4.3. Control System

A quadcopter's control system ensures stable flight by continuously adjusting rotor speeds based on sensor inputs.[16]

- Flight Controller The flight controller acts as the drone's brain, interpreting sensor data and making real-time corrections to maintain stability and control.
- Flight Controller The flight controller acts as the drone's brain, interpreting sensor data and making real-time corrections to maintain stability and control.

- Flight Controller The flight controller acts as the drone's brain, interpreting sensor data and making real-time corrections to maintain stability and control.
- Sensors: Various sensors provide essential data for navigation, altitude control, and orientation adjustments.
  - IMU (Inertial Measurement Unit): Measures acceleration and angular velocity.
  - GPS: Used for navigation and position holding.
  - Barometer: Measures altitude.
  - Magnetometer: Helps maintain heading.

1.2.4.4. Stability and Control Methods

Different control techniques are employed to enhance quadcopter stability and flight precision.[16]

- PID Control: A proportional-integral-derivative controller is commonly used to stabilize the quadcopter.
- Kalman Filter: A sensor fusion algorithm that combines data from different sensors for accurate positioning.
- Machine Learning-Based Control: Advanced systems use AI to enhance flight efficiency and adaptability.

1.2.4.5. Applications of Quadcopters

Quadcopters serve a wide range of applications across various industries.[17]

- Surveillance and Security: Used in military and law enforcement for reconnaissance.
- Agriculture: Crop monitoring, pesticide spraying.
- Aerial Photography and Filmmaking: Provides stable camera shots.
- Search and Rescue: Helps locate missing persons in difficult terrains.

1.3 Flight Controller system

A flight controller (FC) is the central processing unit of a drone, responsible for managing flight stability, navigation, and motor control. It integrates sensors, software, and communication modules to ensure smooth and responsive flight. Acting as the drone's "brain," the flight controller processes data and executes real-time adjustments. [18]

Core responsibilities of a flight controller include:

- Flight Stabilization: Uses gyroscopes and accelerometers to maintain balance and level flight.
- Command Processing: Interprets pilot inputs or autopilot instructions and adjusts motor speeds accordingly.
- Navigation Support: Some controllers incorporate GPS for autonomous flight, waypoints, and return-to-home features.
- Data Management: Communicates with external modules such as cameras, telemetry systems, and payloads.

1.3.1 Types of Flight Control Systems

Different drones utilize various flight control systems depending on their intended use and user expertise [19]:

- Manual Systems: Fully controlled by the pilot, offering maximum maneuverability but requiring significant skill.
- Semi-Automatic Systems: Blend manual control with assistive features like altitude hold and GPS stabilization, making them ideal for intermediate users.
- Fully Autonomous Systems: Rely on advanced sensors and AI to execute automated flights, commonly used in professional and industrial applications.
- Hybrid Systems: Provide the flexibility to switch between manual and autonomous modes, allowing adaptation to different tasks.

1.3.2 The Importance of Flight Control Systems

Flight control systems are essential for drone functionality, ensuring stability, responsiveness, and adaptability. They continuously process real-time sensor data to counteract external disturbances such as wind or sudden altitude shifts. By providing precise control, these systems enable drones to perform tasks like aerial photography and industrial inspections with accuracy and efficiency.

The main advantages of flight control systems include:

- Enhanced Stability and Safety: Mitigates environmental factors for smooth and reliable operation.
- High-Precision Performance: Supports tasks requiring accuracy, such as filming and inspections.
- User-Friendly Operation: Features like auto-hover and return-to-home simplify drone handling.
- Advanced Autonomy: Integrates GPS, AI, and control algorithms for intelligent flight.
- Versatile Applications: Adapts to diverse environments and operational needs. [20]

1.4 Stability Methods

Drone stability is essential for safe and precise flight. To maintain balance and respond effectively to disturbances (like wind or sudden movements), flight controllers use various stability methods. These methods rely on mathematical models, sensor data, and real-time control adjustments.

Classical Control Methods

PID Control (Proportional-Integral-Derivative)

PID control remains the most widely used stability method due to its simplicity and effectiveness. It adjusts the motor outputs to counteract disturbances, keeping the drone level and responsive. [21]

$$u(t) = K_p e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt} \tag{1.1}$$

where:

- $u(t)$:control signal (output of the PID controller)
- K_p (Proportional gain): reduces instantaneous error.
- $e(t) = r(t) - y(t)$: error (difference between the reference input $r(t)$ and the system output $y(t)$)
- K_i (Integral gain): corrects long-term drift.
- K_d (Derivative gain): dampens rapid oscillations.

This method is simple, widely used, and effective for basic stabilization, offering fast response when properly tuned. However, it requires manual tuning for each drone and environment, struggles with sudden disturbances like wind gusts, and cannot dynamically adapt to changing flight conditions.

1.5 Sensor Fusion-Based Stability Methods

Kalman Filter (KF)

The Kalman Filter is a powerful sensor fusion algorithm that enhances flight stability by continuously estimating the drone's true position and orientation from noisy sensor data. [22]

The Kalman Filter operates in two main steps:

Prediction Step

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \tag{1.2}$$

Correction Step

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(y_k - H\hat{x}_{k|k-1}\right) \tag{1.3}$$

where:

- $\hat{x}$ is the estimated state (e.g., position, velocity).
- A is the state transition matrix.
- B is the control input matrix.
- K_k is the Kalman gain, adjusting how much trust is placed in new sensor data.
- y_k is the actual measurement at time k.
- H is the observation matrix.

This method significantly reduces noise from IMUs and GPS, resulting in smooth and accurate position estimation. However, it is computationally intensive, requiring real-time processing, and its performance declines in the presence of highly nonlinear dynamics.

1.6 Advanced Control Techniques

Model Predictive Control (MPC)

Model Predictive Control (MPC) predicts future states and selects control inputs that optimize stability while respecting system constraints (e.g., limited thrust, battery power).

This approach optimizes control actions by considering the future trajectory and effectively handles constraints, making it ideal for safe operations near obstacles. It requires high processing power, which limits its use in small drones, and its real-time implementation is more complex than methods such as PID or Kalman Filtering. [23]

Adaptive Control Systems

Adaptive controllers dynamically adjust control parameters in response to real-time environmental changes, such as payload variations or wind conditions.

Example Methods:

- Self-Tuning PID: Automatically adjusts K_p, K_i, K_d values.
- Model Reference Adaptive Control (MRAC): Adjusts flight dynamics to match a reference model.
- Neural Adaptive Control: Uses AI to learn optimal control strategies over time.

This method enhances stability in dynamic environments and adapts effectively to unknown system changes, such as payload shifts. It relies on real-time learning algorithms and is more complex to implement than standard control approaches.

1.7 AI & Machine Learning-Based Stability Methods

Reinforcement Learning (RL) for Drone Stability

Reinforcement Learning (RL) enables a drone to learn optimal flight control strategies by continuously improving based on trial and error.

Example Algorithms:

- Deep Q-Networks (DQN): Uses a neural network to map drone states to optimal control actions.
- Proximal Policy Optimization (PPO): A deep learning-based method for continuous control.

This approach enables learning of complex control behaviors without the need for explicit system modeling and allows adaptation to unexpected disturbances such as sudden wind gusts. However, it requires large datasets for effective training and incurs high computational costs, making real-time applications challenging.

1.8 Conclusion

This chapter provided an overview of UAV systems, their classification, flight control systems, and stability methods. We explored the differences between fixed-wing, rotary-wing, and hybrid drones, highlighting their unique advantages and applications.

The flight control system (FCS) was identified as the core of drone operation, responsible for stability, navigation, and execution of commands. Various control strategies, from manual to AI-driven autonomous systems, were discussed, emphasizing their role in ensuring precise and reliable flight.

To enhance stability, we examined PID control, model-based approaches, sensor fusion, and AI-driven techniques, each offering different levels of accuracy and adaptability.

This foundation sets the stage for the next chapter, which will focus on the design and implementation of a custom Flight Controller, covering both hardware and software aspects.

Chapter 2

PIC Microcontrollers and Development Tools

2.1 Introduction

A microcontroller (MCU) is a compact, cost-effective computer on a single chip, designed for controlling electronic devices. Unlike general-purpose processors, it includes built-in memory, I/O ports, and timers, making it ideal for dedicated control tasks in embedded systems. Widely used in consumer electronics and industrial applications, microcontrollers are the "brains" behind devices like microwaves, cameras, cars, and toys.

They dominate the chip market due to their affordability, ease of programming, and adaptability. As costs decrease and sizes shrink, MCUs are being integrated into even simpler devices, replacing traditional mechanical components. In embedded systems, understanding various types of microcontrollers such as PIC, AVR, ARM, and FPGAs is essential, as each offers different architectures and capabilities suited to specific applications.

2.2 Microcontrollers

2.2.1 FPGA

An FPGA (Field-Programmable Gate Array) is a programmable integrated circuit that allows users to create custom hardware logic for specific tasks. Unlike microcontrollers with fixed architectures, FPGAs offer flexibility through configurable logic blocks and interconnections, making them ideal for parallel processing, high-speed computing, and tailored hardware designs. Programming an FPGA involves defining hardware behavior using Hardware Description Languages like VHDL or Verilog, enabling parallel execution and superior performance.[24]

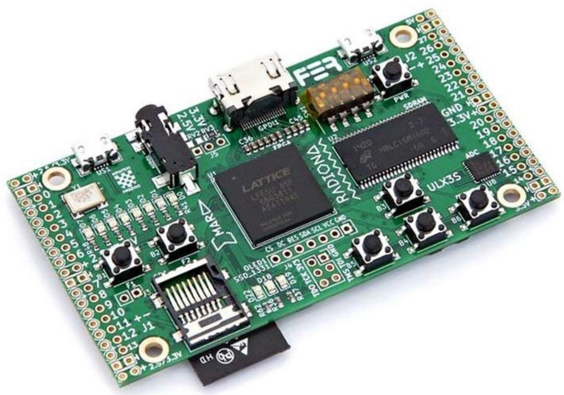

Figure 2.1: The Field Programmable Gate Array)

Programming FPGAs

Programming an FPGA involves using HDLs like VHDL or Verilog to describe hardware behavior, which is then compiled into a bitstream and loaded onto the FPGA. Once configured, it can execute multiple tasks in parallel, making it suitable for applications like DSP, machine learning, and image processing. Its reprogrammable nature allows updates even after deployment, offering great design flexibility.

2.2.2 AVR Microcontrollers

AVR microcontrollers, developed by Atmel (now Microchip Technology), are known for their simplicity and are widely used in embedded systems, especially by hobbyists and in education. The ATmega328P, used in Arduino boards, is a prominent example. Featuring a Harvard architecture with separate program and data memory, AVRs enable efficient processing and are easy to program and integrate into compact, low-power systems. [25]

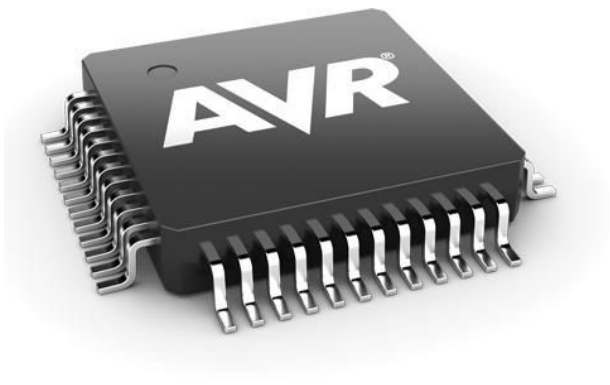

Figure 2.2: AVR Microcontroller

Programming AVR Microcontrollers

AVR microcontrollers are mainly programmed in C or assembly using IDEs like Atmel Studio or PlatformIO. Programming involves direct control of registers and peripherals, with many AVRs supporting bootloaders for easy USB or serial programming. The popularity of the AVR-GCC compiler and the simplicity of the architecture, along with open-source tools, make AVRs a favorite among hobbyists and DIY users.

2.2.3 PIC Microcontroller

The PIC (Peripheral Interface Controller) microcontroller family, developed by Microchip Technology, is another significant player in the embedded systems market. PIC microcontrollers are known for their versatility, broad availability, and low cost. They are available in a wide spectrum of models, ranging from low-power 8-bit devices to more powerful 32-bit variants.

A key characteristic of PIC microcontrollers is their flexibility, offering a range of integrated peripherals such as analog-to-digital converters (ADC), timers, PWM outputs, and communication interfaces (SPI, I^2C, UART). PICs are designed with a RISC (Reduced Instruction Set Computing) architecture, which simplifies their programming. [26]

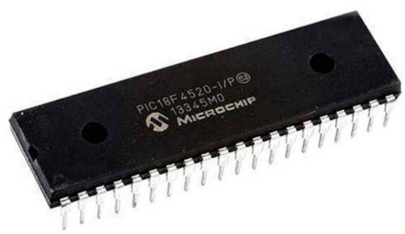

Figure 2.3: PIC Microcontroller

Programming PIC Microcontrollers

Programming a PIC microcontroller involves writing code in C or assembly language. Popular development environments include MPLAB X IDE, PICkit, and HI TECH C. Microchip provides extensive support through libraries and development tools, facilitating the initiation of PIC based projects for engineers.

A significant advantage of the PIC family is the extensive selection of available chips, from entry level, low cost models for basic tasks to more advanced models with increased memory and peripherals for complex undertakings.

2.2.4 ARM Microcontroller

The ARM microcontroller represents a class of microcontrollers based on the ARM architecture, developed by ARM Holdings (now part of Nvidia). ARM microcontrollers, particularly those based on the Cortex M series, have become the most favored choice for contemporary embedded systems. ARM microcontrollers offer a potent combination of performance, low power consumption, and extensive peripheral support.

The ARM architecture encompasses various cores, ranging from the low power Cortex M0 to the high performance Cortex M7 and Cortex M33. These cores are utilized by numerous manufacturers, including STMicroelectronics (STM32 series), NXP (LPC series), and Texas Instruments (Tiva C series). [27]

Figure 2.4: ARM Microcontroller

Programming ARM Microcontrollers

ARM microcontrollers are mainly programmed in C or C++ using IDEs like Keil MDK, IAR Embedded Workbench, or STM32CubeIDE, with assembly used for performance-critical tasks. They often run real-time operating systems like FreeRTOS for complex multitasking applications. Development is supported by libraries, HALs, middleware, and the CMSIS standard, which simplifies programming and hardware interaction.[28]

2.3 Microcontroller Types

The 8-bit microcontroller family is the most prevalent because its word size has proven suitable for the majority of tasks these devices are required to perform. A single byte is generally considered sufficient and offers the advantage of straightforward interfacing with the wide array of IC memories and logic circuits available. Additionally, serial ASCII data is also byte-sized, facilitating easy compatibility with microcontroller communication.

Given the vast diversity of potential microcontroller applications, most manufacturers offer a family of devices, each tailored to specific requirements. This approach avoids using a single, complex, and expensive device for all applications, where some of its features would remain unused. Microcontroller families typically share a common core instruction set, but individual members differ in their memory capacity and type, timer functionalities, and port options. This strategy results in cost-effective devices designed for particular manufacturing needs.

Memory expansion is possible using external RAM and/or ROM. Some family members may lack on-chip ROM, or their ROM can be either electrically programmable (EPROM) or electrically erasable (EEPROM), the latter often referred to as flash EEPROM, allowing for repeated erasure and rewriting of the program. Additional integrated features can include analog-to-digital converters (ADC), digital-to-analog converters (DAC), and analog comparators. Certain family members are also available with lower pin counts for simpler applications, minimizing costs [29].

2.3.1 PIC Microcontroller Architecture

The architecture of a PIC microcontroller encompasses a Central Processing Unit (CPU), Input/Output (I/O) ports, memory organization, an Analog-to-Digital (A/D) converter, timers/counters, interrupts, serial communication interfaces, an oscillator, and a Capture/Compare/PWM (CCP) module, all detailed below.

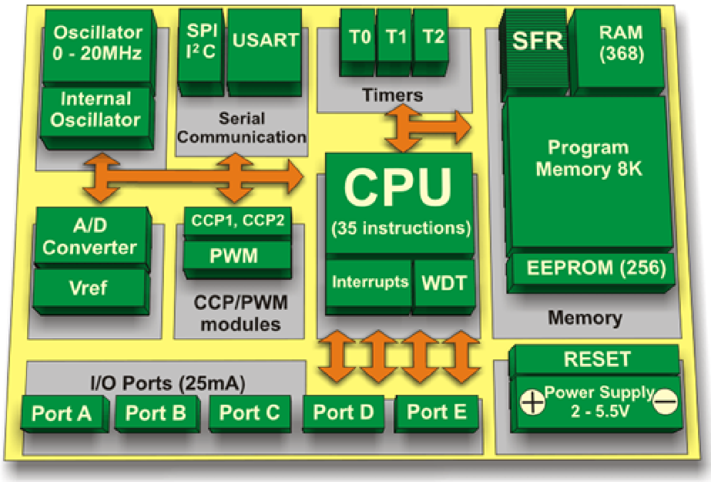

Figure 2.5: PIC Microcontroller Architecture

The PIC microcontroller architecture includes a CPU with an Arithmetic Logic Unit (ALU), Control Unit (CU), Memory Unit (MU), and an accumulator for processing tasks. It uses different memory types: RAM for temporary data storage, divided into General Purpose Registers (GPRs) for data manipulation and Special Function Registers (SFRs) for predefined control tasks; ROM/Flash memory for storing programs; and EEPROM for reprogrammable data storage.

The microcontroller also includes I/O ports (Ports A to E) for input/output operations, controlled by direction registers. Communication is managed via buses (data and address), and serial protocols like USART, SPI, and I2C for interfacing with external devices. Additional components include: A/D Converters: Convert analog signals to digital, with 5 or 8 channels depending on the chip. Timers/Counters: Support precise timing and counting operations in 8 or 16-bit modes. Interrupts: Handle 20 internal and 3 external events from peripherals. Oscillators: Provide clock signals using RC or crystal sources. CCP Module: Supports Capture, Compare, and PWM functions for signal timing and control. This architecture enables PIC microcontrollers to perform efficiently in embedded systems across various applications. [30].

2.4 PIC Microcontrollers

The bus width of a microcontroller, referring to the number of parallel lines for data transfer, directly impacts its precision and overall performance. Based on this bus width, microcontrollers are categorized into 8-bit, 16-bit, and 32-bit types.

2.4.1 8-bit Microcontrollers

These microcontrollers feature an 8-bit (1-byte) wide bus, enabling them to transfer and process 8 bits of data in a single cycle. A significant limitation lies in mathematical operations due to their 8-bit Arithmetic Logic Unit (ALU). Processing larger data, such as 16-bit values, requires multiple cycles for even simple mathematical functions, leading to lower overall performance.

Another key characteristic is their 8-bit timer, which has a maximum range from 0 (0x00) to 255 (0xFF). This limited range can introduce inaccuracies when generating time delay functions.

Common examples of 8-bit microcontroller families include 16F877, 16F877A, and 18F4520. [31]

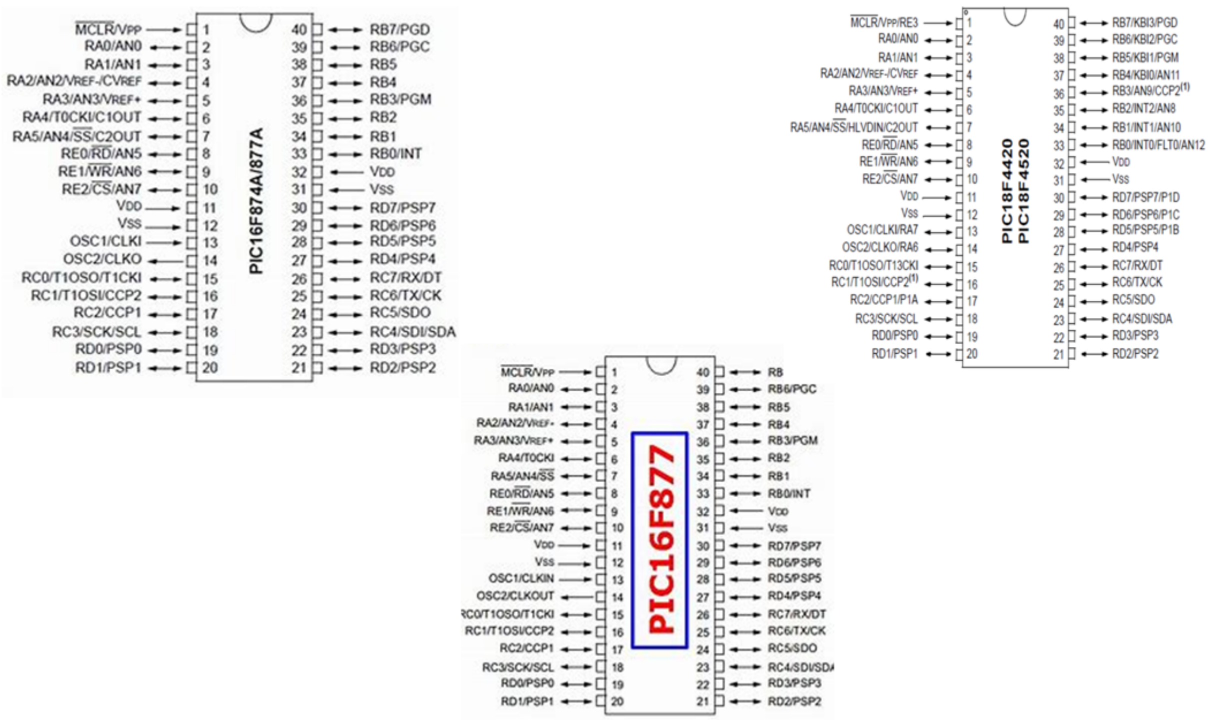

Figure 2.6: 16-bit Microcontroller

2.4.2 16-bit Microcontrollers

These microcontrollers have a 16-bit (2-byte) wide bus, allowing them to transfer and process 16 bits of data in a single cycle. Their 16-bit ALU offers more efficient performance compared to 8-bit counterparts. Additionally, their 16-bit timer provides a wider range from 0 (0x0000) to 65535 (0xFFFF), offering better accuracy per cost for applications requiring timer functionalities.

Common examples of 16-bit microcontroller families include 30F2010 (dsPIC30F Family) and 33FJ12 (dsPIC33F Family).

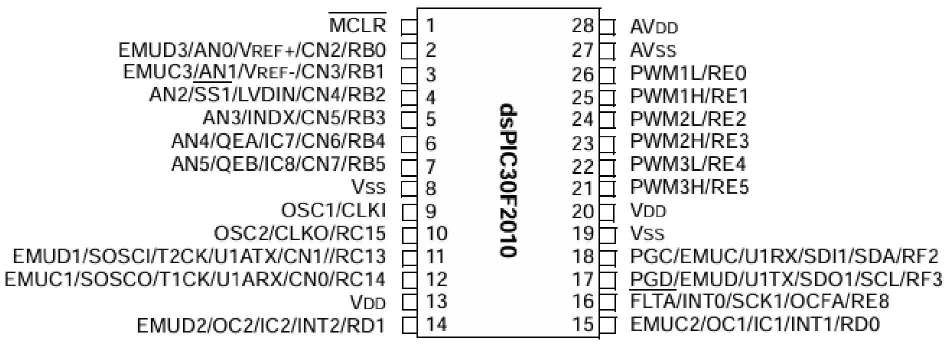

Figure 2.7: 8-bit Microcontrollers

2.4.3 32-bit Microcontrollers

A 32-bit microcontroller has a bus width of 32 bits (4 bytes). While offering higher performance and accuracy than 16-bit microcontrollers, they are also more expensive and consume more power.

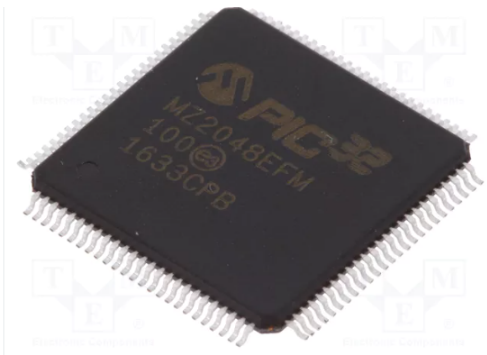

Figure 2.8: 32-bit Microcontrollers

Their higher processing speed makes them well-suited for complex tasks such as audio and video signal processing, and image processing. They can also support a larger number of peripherals commonly required in embedded systems, including Ethernet, Universal Serial Bus (USB), Universal Asynchronous Receiver-Transmitter (UART) devices, and a Controller Area Network (CAN) bus. Common examples of 32-bit microcontroller families include 32MZ2048EFM100, 32MX795F512L.[30]

2.5 Compilers

PIC compilers are essential tools that convert high-level programming languages into machine code for PIC microcontrollers, with two of the most popular being MPLAB XC and MikroC PRO for PIC. MPLAB XC, developed by Microchip, is highly trusted for its strong optimization capabilities, full device support, and direct integration with MPLAB X IDE, making it perfect for professional and industrial projects. MikroC PRO for PIC, developed by MikroElektronika, is designed specifically for standard PIC microcontrollers, offering a very easy-to-use environment with many built-in libraries that speed up development, especially for beginners and hobbyists. For more powerful devices like dsPIC and PIC24, MikroElektronika also offers MikroC PRO for dsPIC, which supports the advanced digital signal processing features and high-performance capabilities needed for more complex applications.

2.5.1 MPLAB

MPLAB is the official development ecosystem created by Microchip Technology for programming and debugging their line of PIC microcontrollers, dsPIC Digital Signal Controllers, and SAM ARM microcontrollers.

The term ”MPLAB” usually refers to MPLAB X IDE (Integrated Development Environment), but historically it has included multiple tools (IDE, compilers, debuggers, and code generators) designed to help engineers write, build, test, and program embedded applications. Whether you’re building a basic 8-bit application or a complex 32-bit embedded system, there’s an XC compiler optimized for your target device.

Figure 2.9: The MPLAB

2.5.2 MikroC

The MikroC compiler, developed by MikroElektronika, is a specialized C programming tool for PIC microcontrollers. It is a next-generation compiler known for its user-friendly interface and advanced features. MikroC includes several integrated tools that enhance the development process, such as a simulator for testing code without physical hardware, a communication terminal for data exchange with connected devices, and a 7-segment display manager for controlling numeric displays.

Additionally, it offers a statistical analyzer to assess code performance, an error corrector to identify and fix syntax errors, and a code explorer for efficient navigation through source files. MikroC also supports a wide range of industry-standard peripherals, including the I^{2}C bus, 1-Wire protocol, SPI bus, RS485, CAN bus, CompactFlash cards, PWM signals, LCDs, and 7-segment displays.

With these capabilities, MikroC provides developers with a complete solution for creating applications on PIC microcontrollers. Designed to simplify system development, it includes an extensive library of hardware components and thorough documentation, making it a popular choice for both professionals and hobbyists seeking to maximize the potential of PIC microcontrollers. [31].

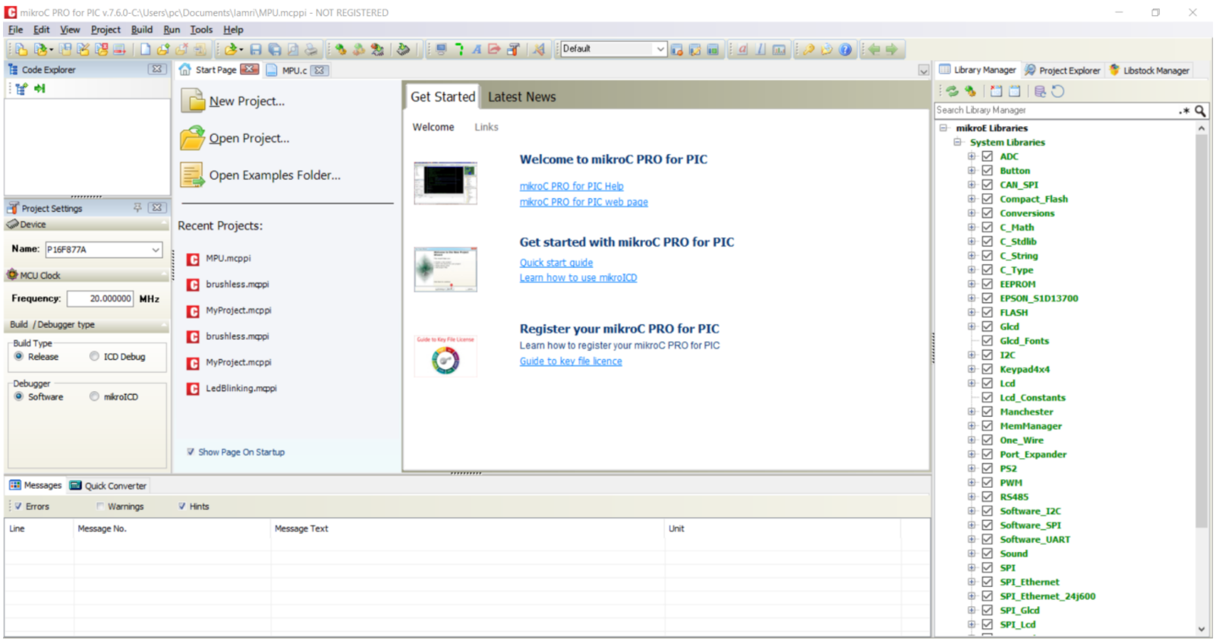

Figure 2.10: MikroC PRO Software Interface for PIC

2.6 Sensors

Sensors play a critical role in embedded systems by enabling microcontrollers to perceive and respond to changes in their environment. They serve as the primary means for acquiring real world data, which is essential for applications such as automation, robotics, environmental monitoring, and healthcare systems.

PIC microcontrollers, developed by Microchip Technology, are widely recognized for their robustness, low power consumption, and versatile features. They are equipped with modules like Analog to Digital Converters (ADC), digital input/output interfaces, I^2C, SPI, and UART communication ports, making them highly suitable for integrating a wide range of sensors.

Sensors used with PIC microcontrollers can generally be classified into two categories: analog and digital. Analog sensors, such as the LM35 temperature sensor and MQ gas sensors, provide a continuous voltage output that is interpreted through the PIC's ADC module. Digital sensors, including PIR motion detectors and ultrasonic distance sensors like the HC-SR04, either output simple HIGH/LOW signals or communicate using serial protocols like I^2C or SPI.

Through the integration of various sensors, PIC based systems are capable of monitoring environmental parameters, detecting physical movements, and making decisions based on sensor inputs. This ability is fundamental to the development of smart systems across a wide range of industries.

2.6.1 PIC Microcontroller Sensor Interface Guide

PIC microcontrollers are equipped with a variety of modules that allow them to communicate efficiently with different types of sensors. Depending on the sensor's nature analog or digital, and its communication protocol, different interfacing methods are used.

2.6.1.1. Analog Sensors

Analog sensors output a continuous voltage that represents the measured parameter. PIC microcontrollers with an internal ADC (Analog to Digital Converter) can read this voltage and convert it into a digital value for processing.

Examples: LM35 (temperature sensor), LDR (light sensor), MQ2 (gas sensor), differential pressure sensors (for Pitot tubes).

Sonde de Vitesse (Differential Pitot Tube):

This is a durable and eco-friendly sensor designed for accurate airspeed measurement. It plays a crucial role in aviation by detecting the difference between static and dynamic air pressure, allowing pilots to determine the aircraft's speed relative to the surrounding air.

Figure 2.11: Differential Pitot Tube

Differential Pressure Sensor (MS4525/MS56 Series):

Pitot tubes measure airspeed by comparing dynamic and static air pressures. Since the Pitot tube itself has no electronics, it must be connected to a differential pressure sensor like:

- **MS4525DO:** digital output (I^2C)
- **MPX5010DP:** analog output
- **MS5607 / MS5611:** digital output (I^2C or SPI, high precision barometric sensors)

The MS56 series (such as the MS5607 and MS5611) are highly accurate barometric pressure sensors widely used for altitude measurement, weather monitoring, and drone stabilization.

These sensors provide precise pressure and temperature readings through an I^2C or SPI digital interface, making them compatible with microcontrollers like the PIC series. To interface the MS56 sensor with a PIC microcontroller, you first configure the PIC's I^2C module to communicate with the sensor using its unique device address. After initializing the communication, the PIC sends commands to the MS56 to trigger pressure and temperature measurements. The sensor then responds with raw data, which the PIC processes using the provided calibration coefficients stored in the sensor's memory. These coefficients allow the PIC to convert the raw data into human-readable values for pressure and temperature. By implementing this setup, applications such as altitude tracking, environmental monitoring, or navigation systems can be effectively developed, leveraging the high resolution and reliability of the MS56 sensor in combination with the processing capabilities of the PIC microcontroller. [32]

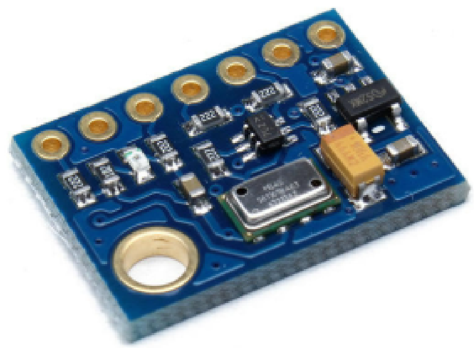

Figure 2.12: MS5611 Sensor

Figure 2.13: MS5607 Sensor

2.6.1.2. Digital Sensors (Simple High/Low Output)

Some sensors have a digital output that simply toggles between HIGH and LOW based on a threshold.

Examples: PIR motion sensor (HC-SR501), TTP223 touch sensor.

Sensors Using Serial Communication

More complex sensors send structured data over serial communication protocols like I^2C, SPI, or UART.

- **I²C Sensors:**for example MPU6050 (accelerometer + gyroscope), BMP280 (pressure sensor)
- **SPI Sensors:**such as BMP280 (alternative SPI mode), some high-speed ADCs
- **UART Sensors:**like GPS modules, serial output sensors

2.6.2 MPU-6050

The MPU-6050 is a popular 6-axis motion tracking sensor that combines a 3-axis gyroscope and a 3-axis accelerometer into one integrated circuit. It also includes an onboard Digital Motion Processor (DMP) for advanced tasks such as orientation detection and sensor fusion. The sensor communicates via the I²C interface, making it compatible with PIC microcontrollers[33].

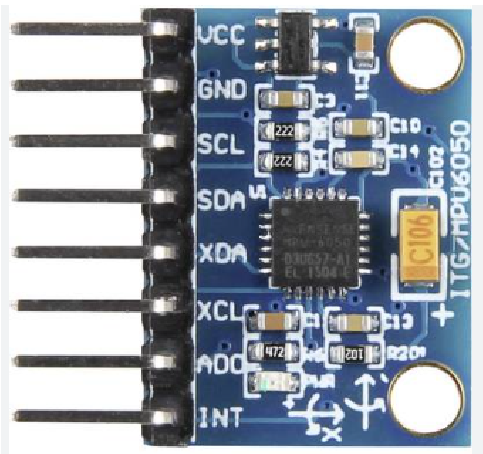

Figure 2.14: MPU-6050 Sensor

The MPU-6050 can be interfaced with a PIC microcontroller to enable applications involving motion detection, orientation tracking, and stabilization.

2.7 Conclusion

This chapter offered a detailed examination of various microcontroller technologies and their development environments, with a particular focus on PIC microcontrollers. It began by comparing major microcontroller families FPGA, AVR, PIC, and ARM emphasizing their architectures, programming approaches, and typical applications.

The discussion then shifted to PIC microcontrollers, detailing their architecture and categorizing them into 8-bit, 16-bit, and 32-bit versions. Development tools like MPLAB and MikroC were also introduced, showing how these environments support coding and debugging.

Additionally, sensor integration with PIC microcontrollers was covered extensively, focusing on analog, digital, and serial-based sensors such as the MPU-6050. This foundational knowledge equips the reader with the skills to design embedded systems using PIC microcontrollers.

Chapter 3

Quadcopter Flight Controller

3.1 Introduction

To ensure stable and responsive flight, a quadcopter requires a flight controller that continuously processes sensor data and adjusts motor speeds in real time. Without this system, maintaining balance, orientation, and executing precise maneuvers would be nearly impossible, especially under varying conditions. In this part, we will develop a flight controller capable of reading data from sensors like gyroscopes and accelerometers, calculating orientation, and applying control algorithms such as PID to stabilize the quadcopter and respond to user commands or autonomous instructions.

3.2 Quadcopter Flight Controller

3.2.1 Quadcopter components

The selection of components for building our quadcopter drone was guided by practical constraints such as weight, cost, performance, and compatibility. The drone includes the essential hardware for stable flight and control, using widely available, cost-effective, and proven technologies.

Frame: The DJI F450 plastic frame was selected for its robustness, modularity, and compatibility with standard drone hardware. It features lightweight yet durable arms and a central hub that securely houses the electronics. This frame supports easy assembly and future upgrades.

Figure 3.1: The Frame

Motors: We select for our work the 1000 KV brushless DC motors, which offer an efficient balance between speed and torque. These motors are ideal for multirotor drones due to their smooth operation, reliability, and suitability for PID-based control systems.

Figure 3.2: The Motors

Electronic Speed Controllers (ESCs): ESCs rated at 30A convert flight controller signals into precise motor speeds. These ESCs are matched to the selected motors and battery, enabling responsive throttle control required for flight stabilization.

Figure 3.3: The Electronic Speed Controllers

Battery: A 3s Cells LiPo battery powers the system, providing high current output and energy density. Its lightweight design allows for extended flight time while supporting the power demands of both motors and control electronics.

Figure 3.4: Lipo Battery

Propellers: The propellers 10 x 4.5 inches, are used to ensure compatibility with the motors. Their size and pitch are optimized for efficient thrust generation, stable hovering, and precise maneuvering.

Figure 3.5: Propellers 10x4.5 inches

FlySky FS-i6X Transmitter: This 2.4 GHz digital RC transmitter provides manual control through multiple channels. It sends pilot inputs like throttle, pitch, roll, and yaw to the drone's receiver, enabling real-time flight adjustments.

Figure 3.6: FlySky FS-i6X Transmitter

FlySky iA6B Receiver: The iA6B receiver receives signals from the FS-i6X transmitter and transmits them to the flight controller via i-BUS or PPM. With dual antennas and failsafe functionality, it ensures reliable, low-latency communication.

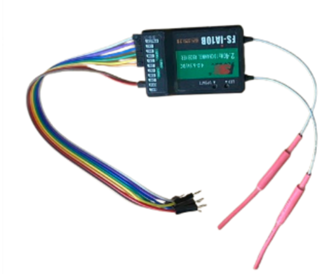

Figure 3.7: FlySky iA6B Receiver

Additional Components:

Other essential materials include wiring, connectors, motor mounts, and screws. These were selected for electrical reliability and mechanical robustness. The integration was done with attention to weight distribution and modularity, ensuring the drone's performance and ease of maintenance.

3.2.2 Flight controller

3.2.2.1. PIC Microcontroller type dsPIC30F4013

Our electronic circuit is based on a high-performance 16-bit PIC microcontroller from Microchip. Among the various available options, we chose the dsPIC30F4013 because it is well-suited to the requirements of our work.

This microcontroller, housed in a 40-pin package, operates at a clock speed of up to 120 MHz and offers complete control in a single chip. It features special function registers, power-on reset, interrupt management, user RAM for program data storage, EPROM program memory, 32-bit timers, a rich instruction set, low power consumption, and integrated analog-to-digital converters (ADCs).

In addition to these capabilities, the dsPIC30F4013 includes architectural enhancements that make it an ideal choice for many high-performance and power-sensitive applications.

In parallel, we also used another microcontroller with a 28-pin package that shares similar characteristics with the dsPIC30F4013, allowing us to maintain functionality while optimizing for size in specific parts of the design. Figure 8 shows the architecture of the microcontroller. Below are the main characteristics of the dsPIC30F4013 microcontroller. [34]

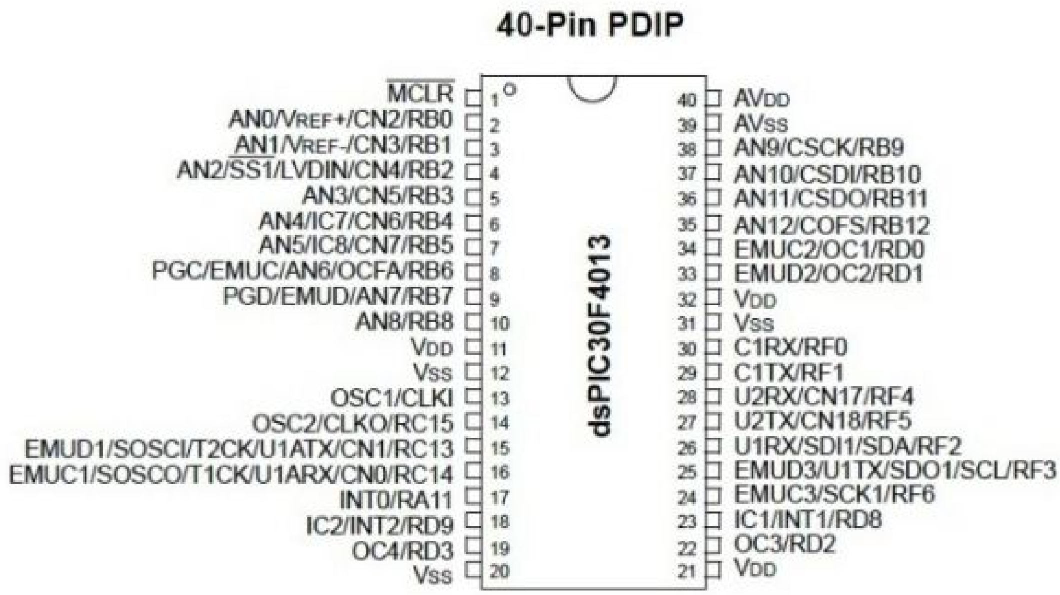

Figure 3.8: dsPIC30F4013 Microcontroller

Feature	**Specification**
Operating Voltage (VDD)	2.5V to 5.5V
Maximum Clock Frequency	120 MHz (up to 30 MIPS with internal DSP)
Pin Count	40 or 44 pins (depending on package type)
ADC (Analog-to-Digital Converter)	13 input channels, 12-bit resolution
Timers	3 Timers (Timer1, Timer2, Timer3)
PWM Modules	2 Enhanced Capture/Compare/PWM (ECCP) modules
External Interrupts	3 programmable INTx lines
Communication Interfaces	2 UARTs, 1 SPI, 1 I^2C
DSP Engine	Yes – 17-bit by 17-bit single-cycle multiplier
Signal Handling	Capable of analog and digital signal processing
Power Supply Options	USB (via external regulator) or external DC
Typical Operating Current	∼25 mA (varies with peripheral usage)
Flash Program Memory	48 KWords (96 KBytes)
SRAM (Data Memory)	2 KBytes
EEPROM	1 KByte

Table 3.1: Microcontroller Feature and Specification Summary

3.2.2.2. Description of the Flight Controller Circuit

The electronic circuit of our flight controller is centered on the dsPIC30F4013 16-bit microcontroller, running at 120 MHz, offering the processing power needed for real-time drone stabilization. An MPU6050 inertial sensor, connected via I2C at a frequency of 1MHz, provides 6-axis motion data (accelerometer + gyroscope) for attitude estimation.

The microcontroller processes this data and generates PWM signals to control four 30A ESCs, which in turn drive 1000 KV brushless DC motors. A stable crystal oscillator ensures precise timing, while the design also includes a regulated power supply and essential protections.

The full system is simulated in Proteus, illustrating all key connections for validation and testing.

The MPU6050 sensor was connected to the dspic30F4013 microcontroller via the I^2C protocol at 1MHz frequency. The sensor's VCC pin was connected to a stable 5V power supply, and the GND pin was connected to the microcontroller's ground. The SCL (clock) line was connected to pin 25 (SCL1) of the dsPIC30F4013, and the SDA (data) line to pin 26 (SDA1). Pull-up resistors of 4.7were added on both lines to ensure reliable I^2C communication.

For data monitoring on the PC, a USB-to-serial FTDI module (FT232RL) was used in receiver-only mode. Only the TX pin of the dsPIC (pin 15, U1TX) was connected to the RXD pin of the FTDI module to transmit data from the microcontroller to the PC. The RX pin of the dsPIC was left unconnected, as no data was received from the computer. The GND of the FTDI was tied to the dsPIC's ground, and the module was powered via USB. Data sent by the microcontroller was received and visualized on the computer using Docklight software, configured at a baud rate of 115200 bps.

Two brushless motors were used for testing, each driven by an ESC (Electronic Speed Controller). The control signals were output as PWM from the dsPIC: motor 1 was connected to pin 19 (OC1) and motor 2 to pin 22 (OC2). The ESCs received their power from an external battery source, and their GNDs were connected to the microcontroller's ground to ensure proper signal reference. We used the very high-precision 32-bit timers of the dsPIC microcontroller to generate accurate PWM signals for driving the brushless motors.

Thanks to this precision, we were also able to implement the main flight control loop time with a fixed period of 4 milliseconds (loop frequency of 250 Hz). This control loop estimates the orientation angles of the system in real time, using data from the gyroscope (angular velocities and linear accelerations).

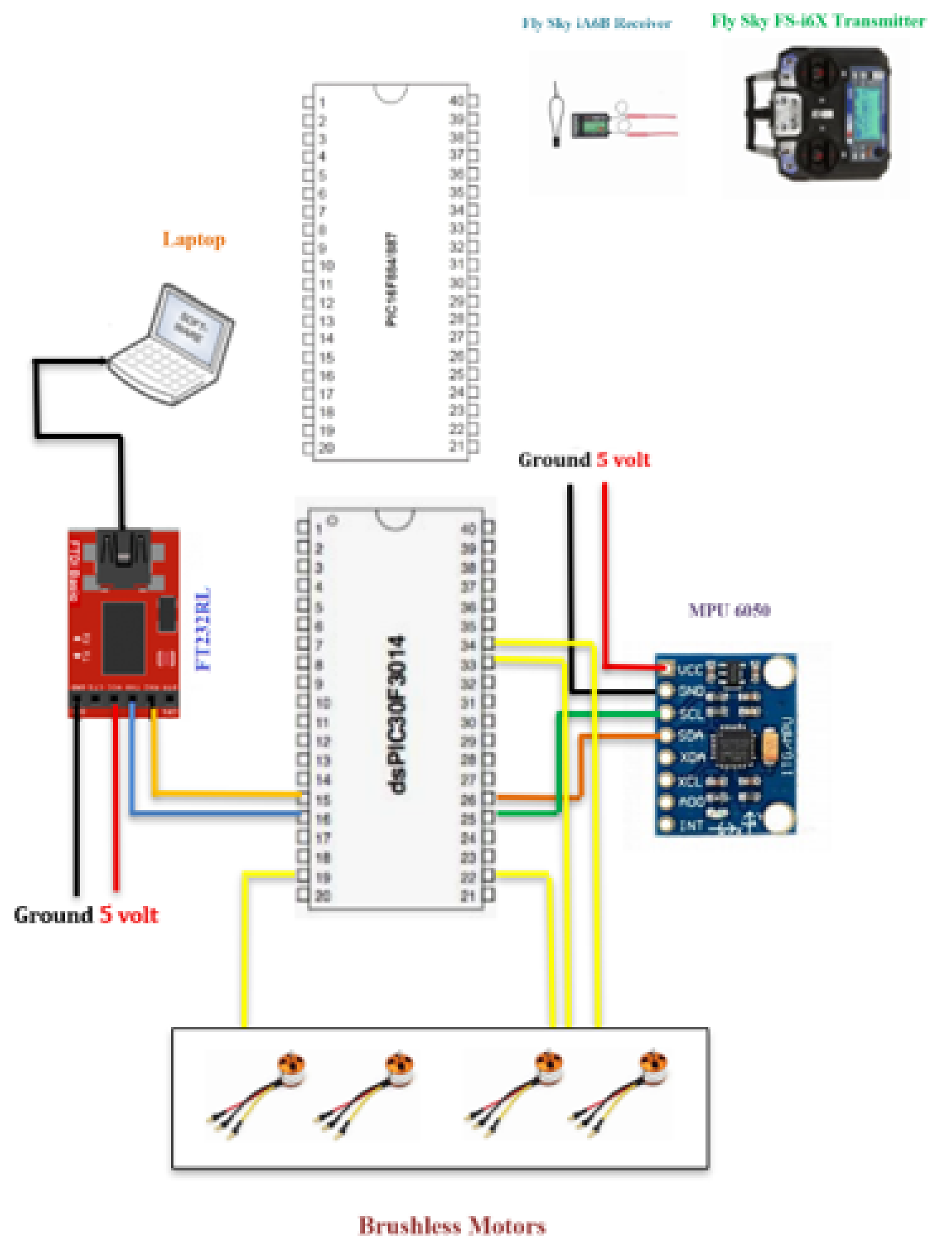

Figure 3.9: Circuit Schematic

3.2.2.3. Flight controller development

a. Flight controller development methodology:

To realize the flight controller, we first designed the schematic and tested individual parts of the overall circuit separately through virtual simulation in Proteus.

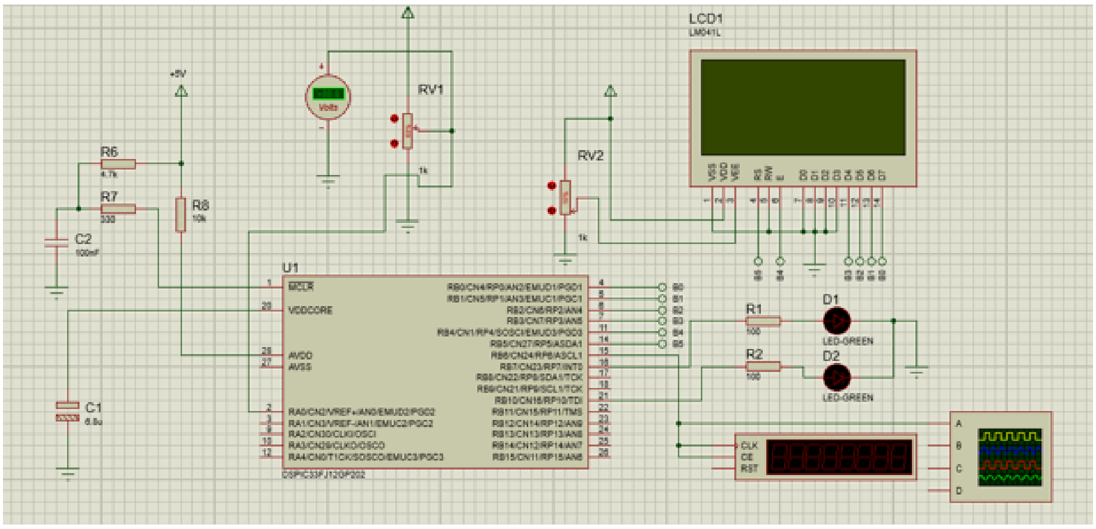

Figure 3.10: Flight controller development methodology

After completing the simulation, the circuit was built and tested on a breadboard to verify its practical functionality and ensure that all components operated as expected in a real environment.

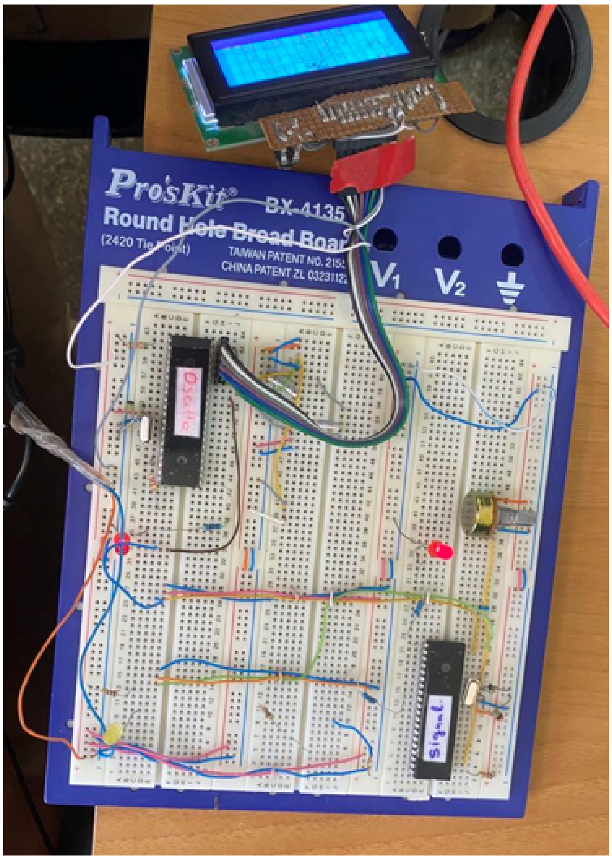

Figure 3.11: Practical Implementation on Breadboard

Following the successful verification on the breadboard, the circuit was finalized by soldering the components onto a perforated board, providing a stable and durable setup for further testing and integration.

After confirming the connections, we proceeded to assemble the hardware. We soldered the dsPIC30F4013 microcontroller, crystal oscillator, voltage regulator, and header pins for the MPU6050 and ESC connections on a custom PCB.

Each connection was carefully checked using a digital multimeter to ensure there were no shorts or open circuits. Once the hardware was complete, we uploaded the firmware and verified I^2C communication with the MPU6050. PWM outputs were tested using an oscilloscope to confirm correct signal generation. Final integration included connecting the ESCs and motors, followed by bench testing for stability and control response.

b. Soldering, integration and verification tests:

Before any soldering took place, we created a precise layout plan to organize all components efficiently on the perforated. This diagram helped us anticipate wiring paths, component placement, and space optimization. To facilitate the integration process and reduce soldering errors, we used black female pin headers (or IC sockets) to temporarily position the components. This approach allowed us to verify alignment and fit before committing to permanent solder joints.

Once the layout was validated, we proceeded with the systematic construction of our flight controller. The following steps summarize the full assembly process.

1. Material Placement Planning: A detailed hand drawn layout was created to organize component placement and ensure logical routing between modules.

2. Placing and Soldering the PIC Microcontroller: The PIC microcontroller, central to our homemade flight controller was carefully positioned and soldered alongside basic passive components such as resistors. Pin headers were used to support modular connections.

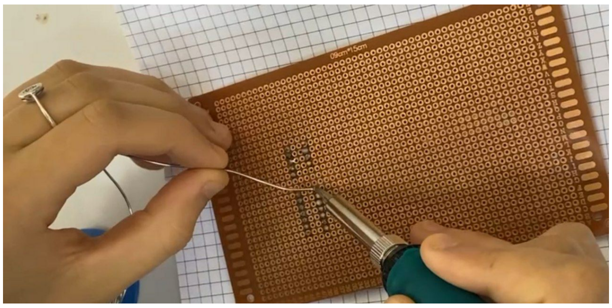

Figure 3.12: Soldering the PIC Microcontroller

3. Soldering the Gyroscope Module: We installed the gyroscope module with precise pin matching, since it plays a key role in detecting drone movement and orientation.

4. Connecting the FTDI Module: The FTDI interface was integrated to allow USB-to-serial communication with a PC for programming and debugging purposes.

5. Connecting the PICKIT Programmer: The PICKIT programmer was connected to flash our custom firmware into the PIC and perform real-time code verification.

6. Soldering Signal Pins: Signal interface pins were added to connect sensors, actuators, and external modules like ESCs or RC receivers

7. Placing Jumper Wires : Jumper wires were soldered to interconnect all board sections, maintaining clear routing and avoiding wire tangling.

8. Adding Indicator LED LEDs were added to indicate power, status, or errors during testing, helping us monitor the controller's behavior in real time.

9. Soldering the External Oscillator : An external oscillator was mounted to provide a stable clock source for accurate microcontroller operation.

10. Performing the Continuity Test : Before powering the system, we conducted a full continuity check to ensure all connections were correctly made and that no shorts were present.

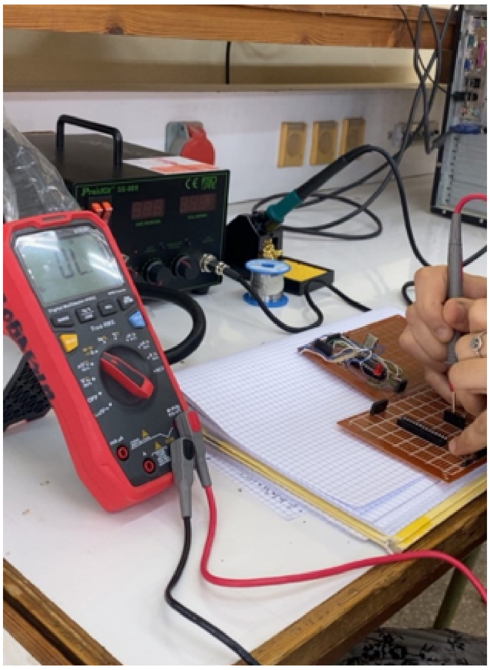

Figure 3.13: Performing the Continuity Test

Two circuits were realized: the first one (Figure3.14) aimed to validate our design and ensure possible stability on a single axis (in this design the dspic30F2010 microcontroller was used), while the second one was developed with the capability for flight control on three axes (Figure3.15).

This last includes two dsPIC microcontrollers one dedicated to detecting the signal from the remote controller, and the other responsible for maintaining stability.

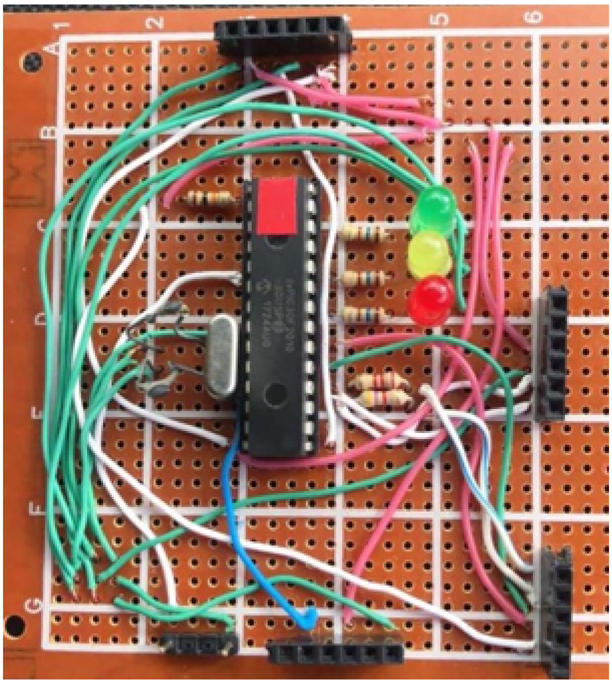

Figure 3.14: One axis stability Circuit using dspic30F2010 microcontroller

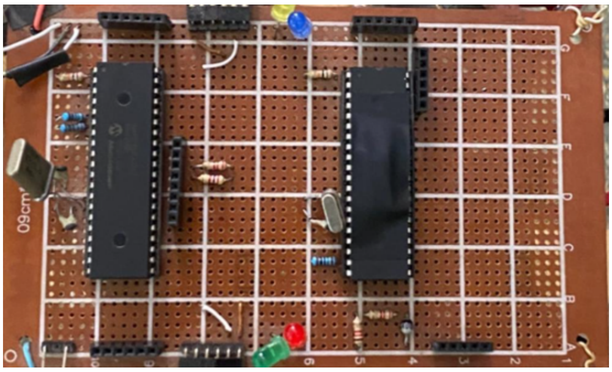

Figure 3.15: Flight control circuit using dspic30F4013 microcontroller

3.2.2.4. Stability Code

The stability of our quadcopter is achieved using a PID controller, implemented in C language using MikroC Pro for dsPIC, specifically for the 16 bit dsPIC30F4013 microcontroller running at 118 MHz..The code reads orientation data (roll, pitch, yaw) from the MPU6050 gyroscope/accelerometer over I^2C at 115200 Hz, then applies PID corrections every 4 milliseconds using Timer1 (32 bit) configured for high-precision interrupts. This ensures consistent motor speed updates for stable flight. The PID algorithm calculates the error between the desired and actual angles and adjusts motor PWM outputs accordingly (via ESCs). To enhance safety, the system includes a 10 seconds calibration delay at startup to ensure ESCs initialize properly, and a manual potentiometer-based arming system is used to prevent accidental motor activation. Additionally, the code features a failsafe routine that shuts down the motors if the receiver signal is lost or sensor data becomes invalid. A simplified PID flowchart is shown below.

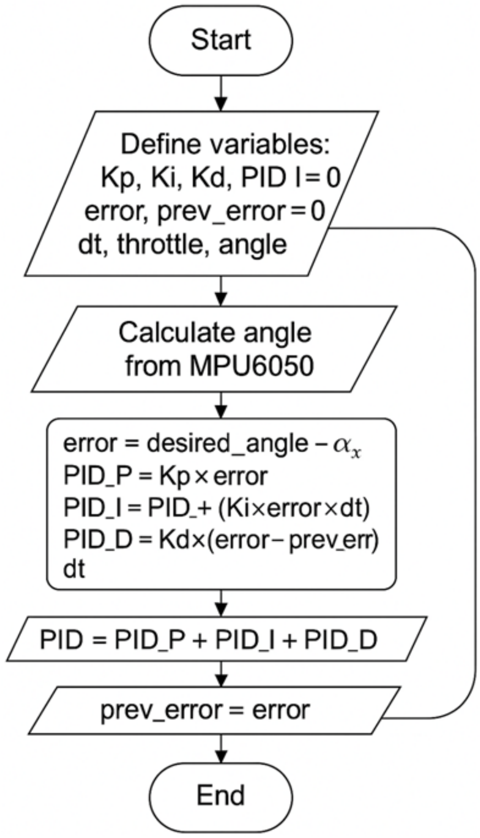

Figure 3.16: A simplified PID flowchart

3.3 Experimental Testing

The hexacopter assembly was configured with two motors per axis, with the current setup focusing on controlling only a single axis. This simplified configuration enabled preliminary validation of the control logic and integration of the MPU6050 sensor prior to scaling to a full quadcopter system. The dual-motor arrangement per axis enhances system stability during operation. The flight controller is based on the dsPIC30F2010 microcontroller, and communication with the PC is established through a UART serial link for data monitoring and debugging.

Figure 3.17: One Axis stability using Two brushless motors

3.3.1 PID Tuning and Testing Overview

To achieve optimal flight stability for the quadcopter, several PID gain configurations were tested. The aim was to find the best balance between responsiveness and stability. Through iterative tuning, different combinations of proportional (K_p), integral (K_i), and derivative (K_d) gains were applied, and the system's behavior was carefully observed. The table below summarizes all test cases, their outcomes, and remarks.

Test	K_p	K_i	K_d	**Observed Response**	**Comment**
01	00.500	00.000	00.000	Very slow, no correction	System too passive
02	01.000	00.000	00.000	Still unstable, slight drift	Low proportional gain
03	01.200	00.000	00.000	Minor improvement	Still underdamped
04	01.500	00.000	00.000	Reactive but unstable	No damping effect
05	01.700	00.000	00.000	Slightly better	Still overshoots
06	01.800	00.000	00.000	Fast response, but oscillatory	Missing derivative control
07	02.000	00.000	00.000	Too aggressive, overshooting	K_p too high
08	03.000	00.000	00.000	Very unstable	Uncontrolled response
09	01.500	00.000	01.000	Slight damping	Small improvement
10	01.500	00.000	05.000	Less oscillation	Smoother, still not optimal
11	01.500	00.000	10.000	Reactive, more stable	Near optimal K_d
12	01.500	00.000	13.000	Responsive and well-damped	Good balance
13	01.500	00.000	16.000	Fast correction, minimal overshoot	Very close to best
14	01.500	00.000	18.000	Smooth but slightly undercorrected	Good result
15	01.500	00.000	20.000	Slower response, too much damping	K_d too high
16	01.800	00.000	18.000	Stable, fast return, slight drift	Almost ideal
17	01.800	00.001	18.000	Very stable, smooth, no overshoot	Best result – retained

Table 3.2: PID Parameter Testing and Observed Responses

Test 01

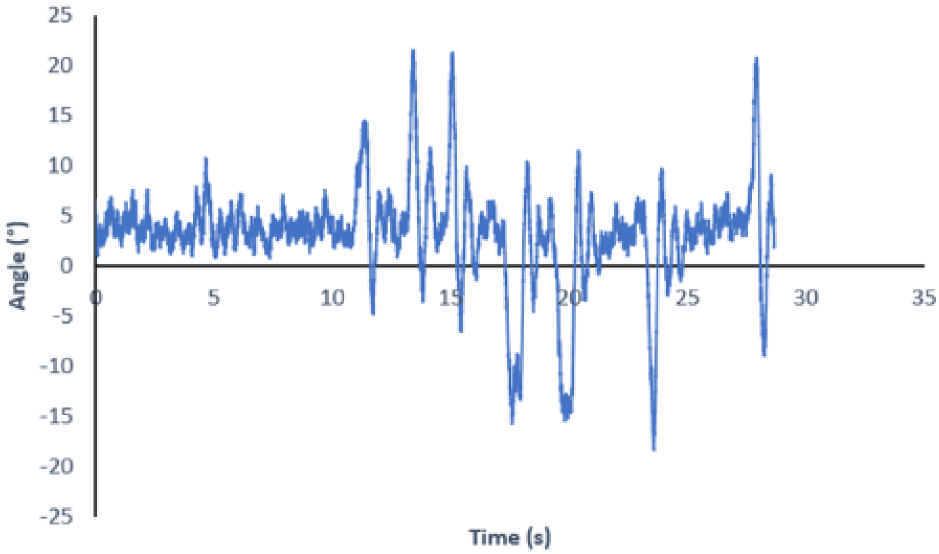

Figure 3.18: With $K_p = 0.500$, $K_i = 0.000$ and $K_d = 0.000$

Note: The system is too passive and applies no correction.

Test 02

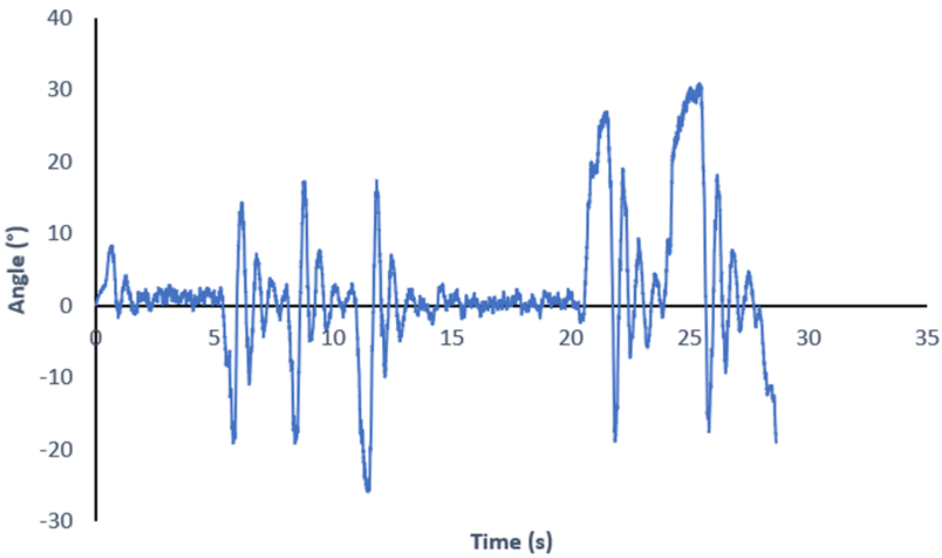

Figure 3.19: With $K_p = 1.000$, $K_i = 0.000$ and $K_d = 0.000$

Note: Due to the low proportional gain, the system exhibits a slight drift.

Test 03

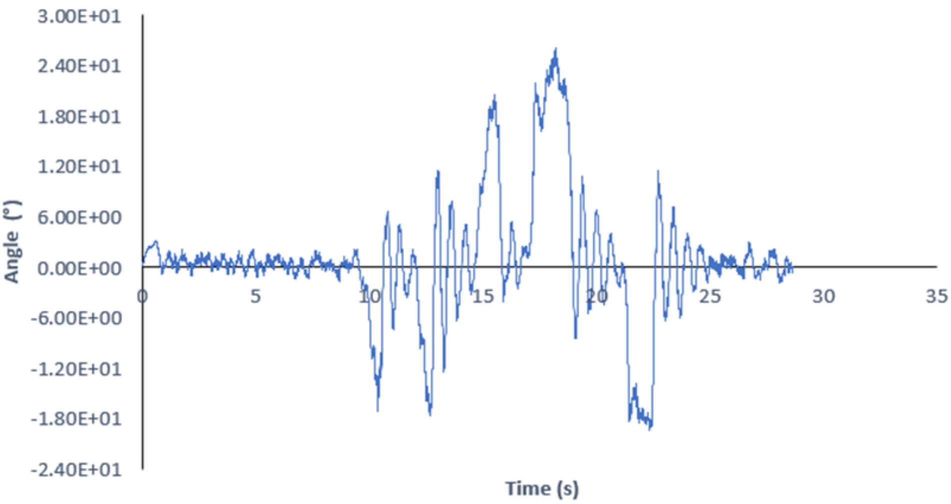

Figure 3.20: With $K_p = 1.200$, $K_i = 0.000$ and $K_d = 0.000$

Note: There is a minor improvement, but the system is still underdamped.

Test 04

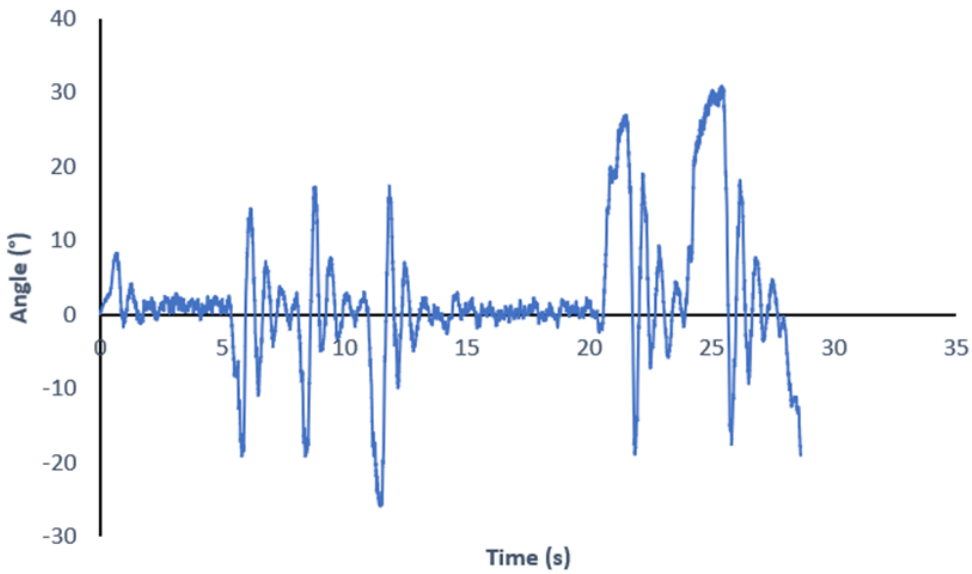

Figure 3.21: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 0.000$

Note: The system is reactive but unstable, with no damping.

Test 05

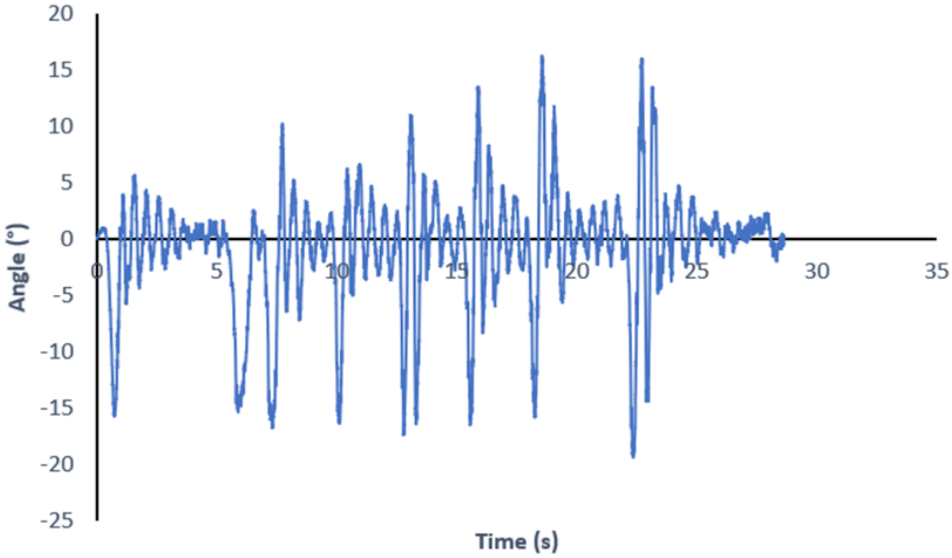

Figure 3.22: With $K_p = 1.700$, $K_i = 0.000$ and $K_d = 0.000$

Note: Response is slightly improved, though overshooting remains.

Test 06

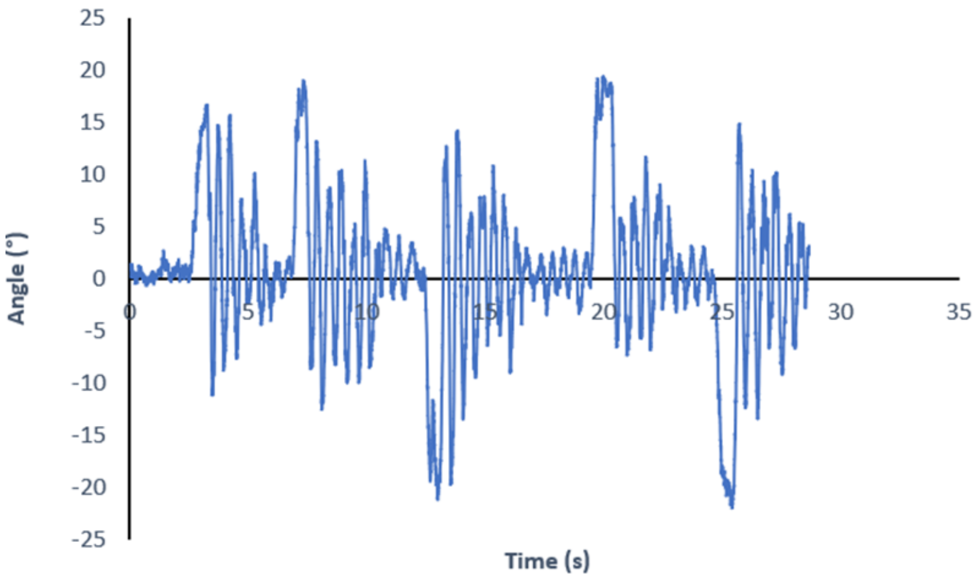

Figure 3.23: With $K_p = 1.800$, $K_i = 0.000$ and $K_d = 0.000$

Note: While the response is fast, it is oscillatory.

Test 07

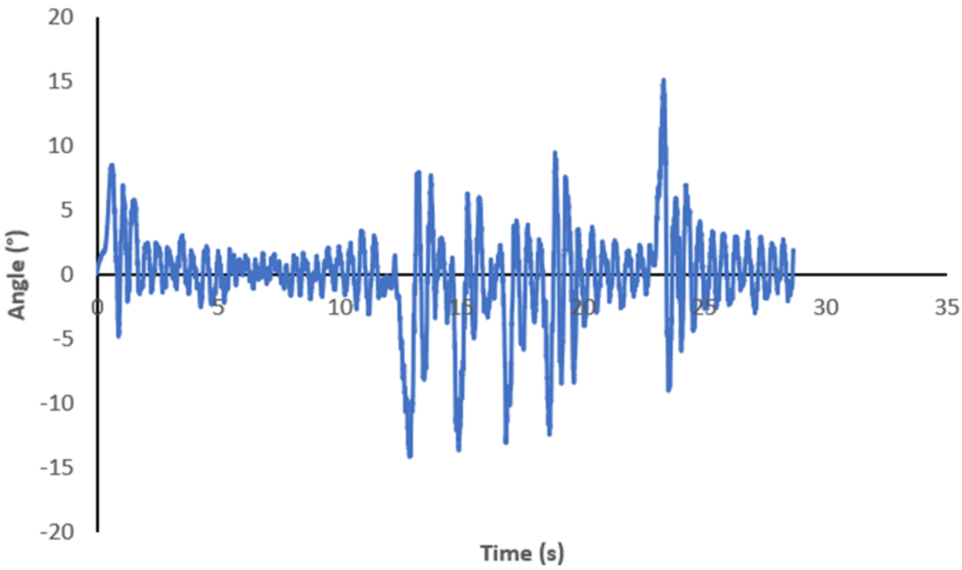

Figure 3.24: With $K_p = 2.000$, $K_i = 0.000$ and $K_d = 0.000$

Note: The system is too aggressive and overshoots.

Test 08

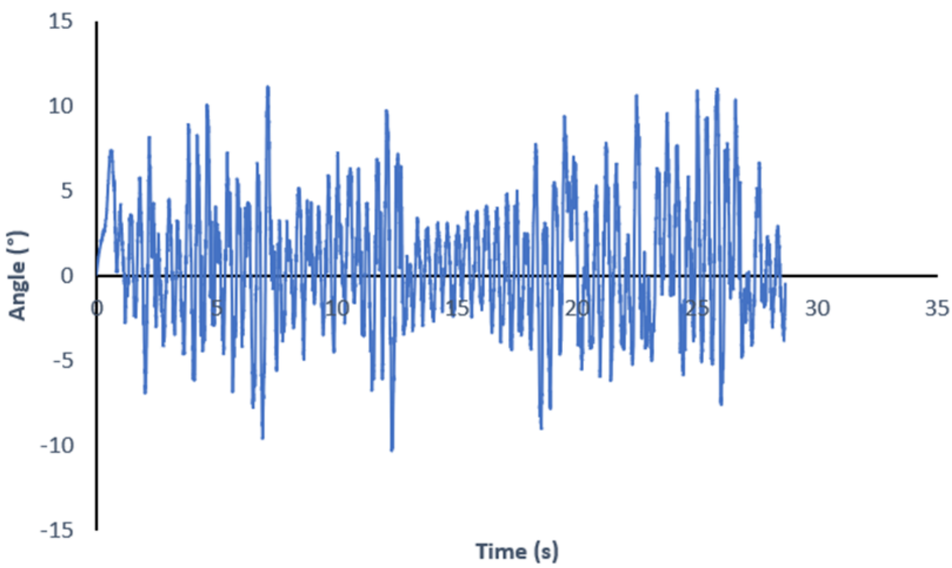

Figure 3.25: With $K_p = 3.000$, $K_i = 0.000$ and $K_d = 0.000$

Note: The system is highly unstable and lacks control.

Test 09

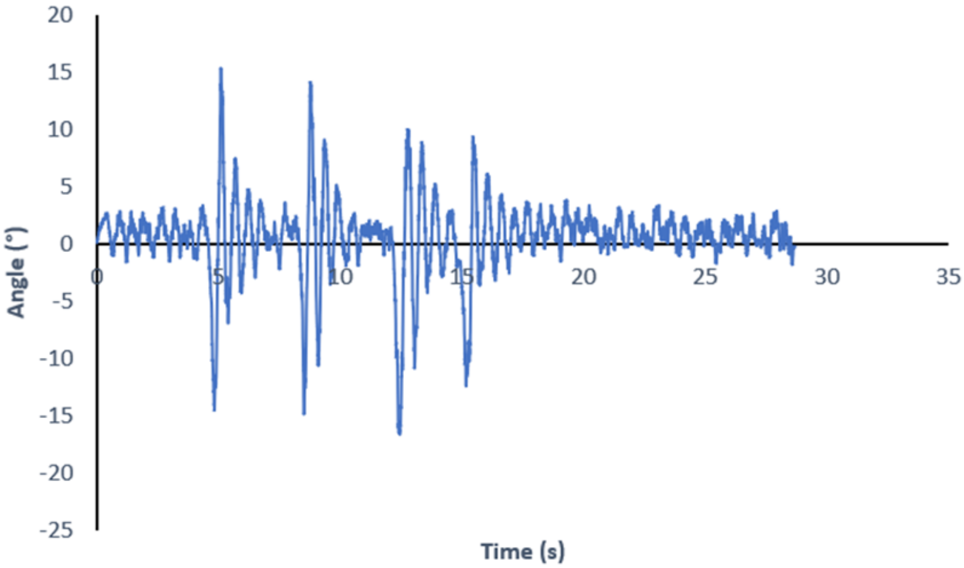

Figure 3.26: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 1.000$

Note: Slight damping leads to a small improvement.

Test 10

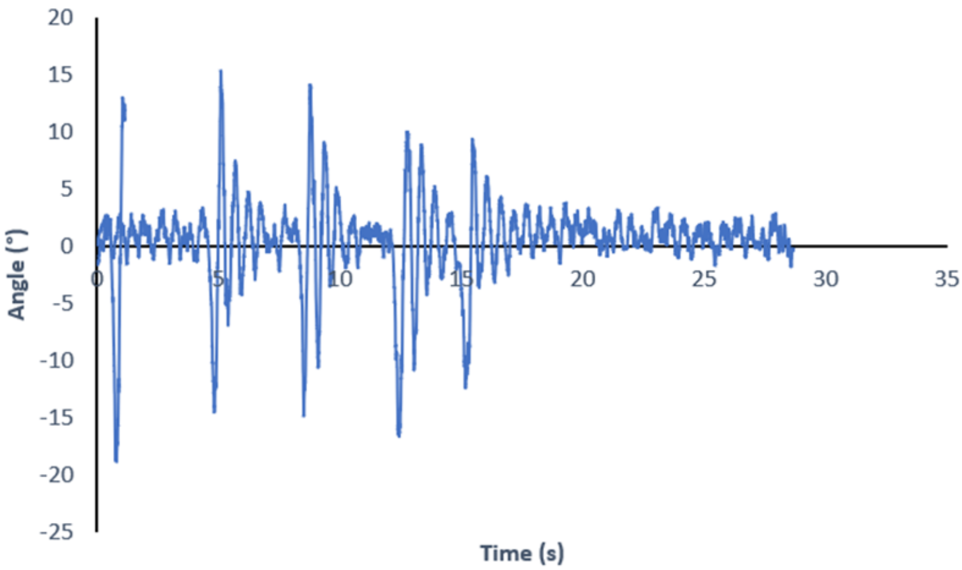

Figure 3.27: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 5.000$

Note: The system shows a smoother output and reduced oscillation.

Test 11

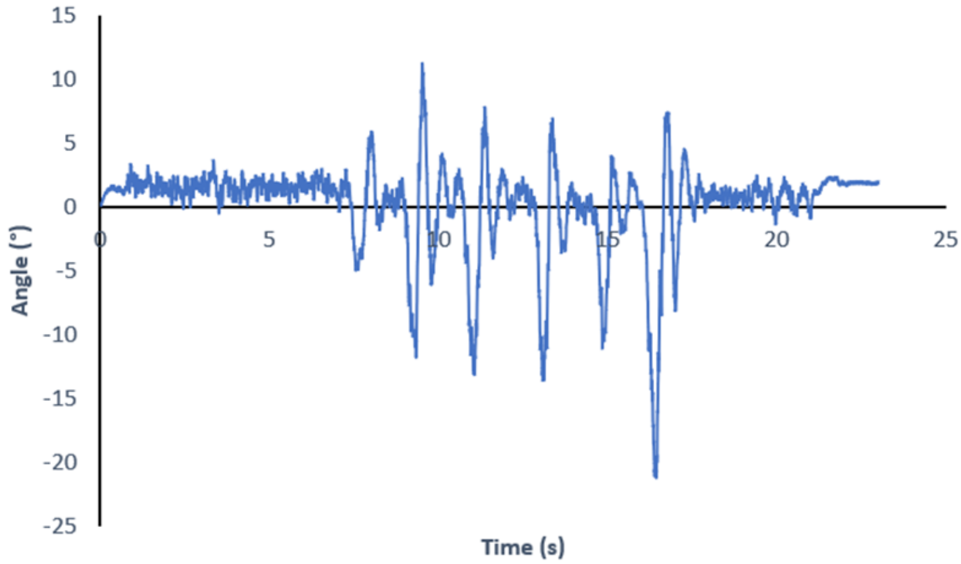

Figure 3.28: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 10.000$

Note: System shows increased stability at near optimal K_d.

Test 12

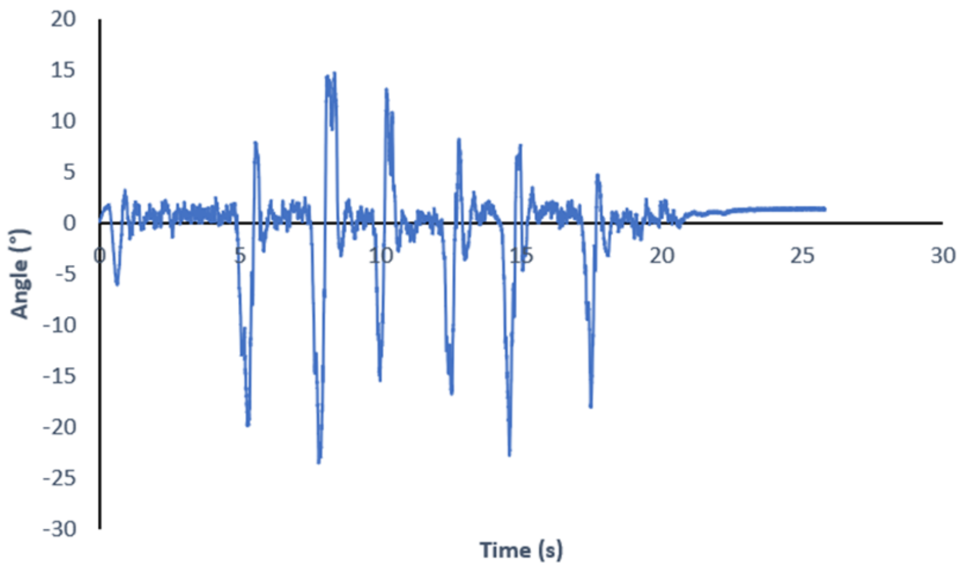

Figure 3.29: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 13.000$

Note: The system is well damped and exhibits good balance.

Test 13

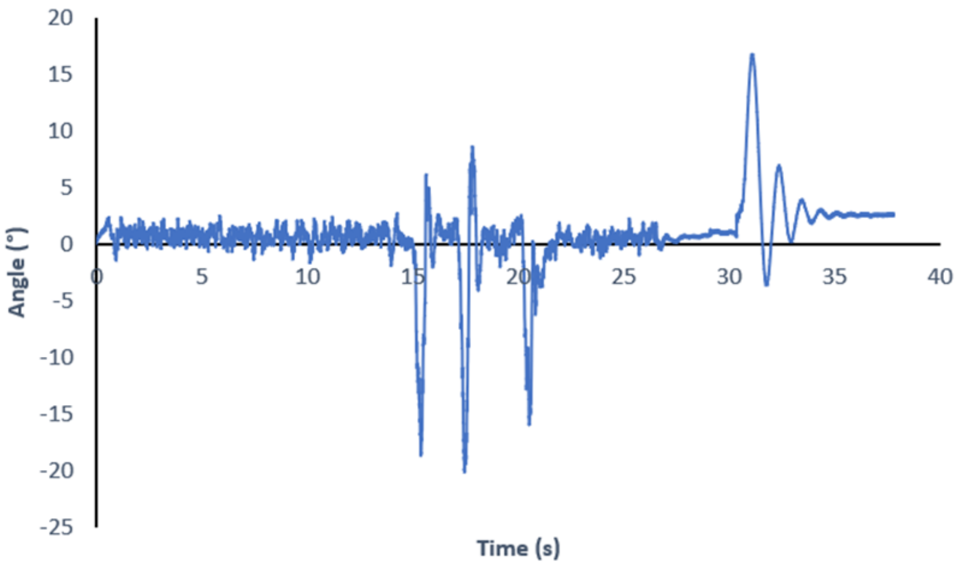

Figure 3.30: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 16.000$

Note: Fast correction, minimal overshoot.

Test 14

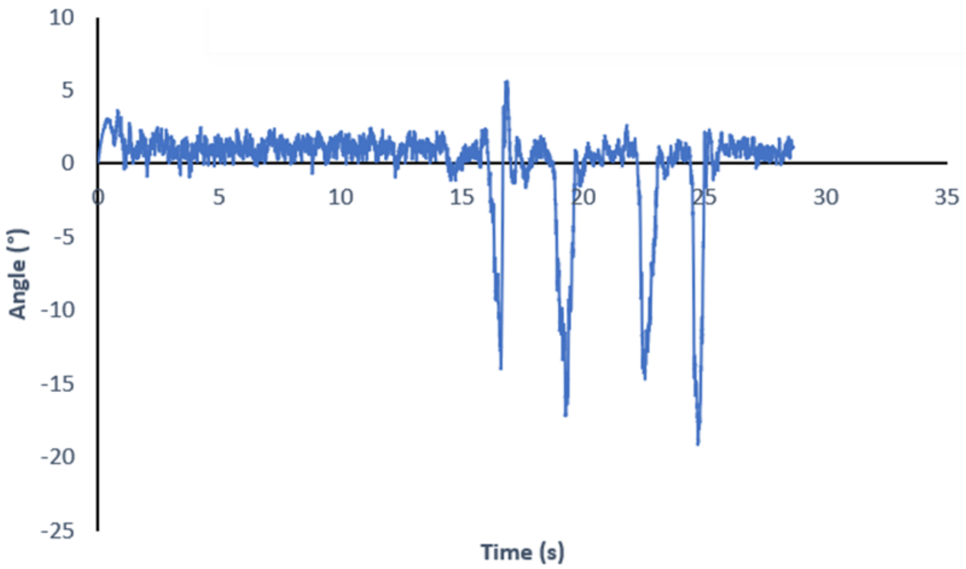

Figure 3.31: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 18.000$

Note: Smooth response, good result.

Test 15

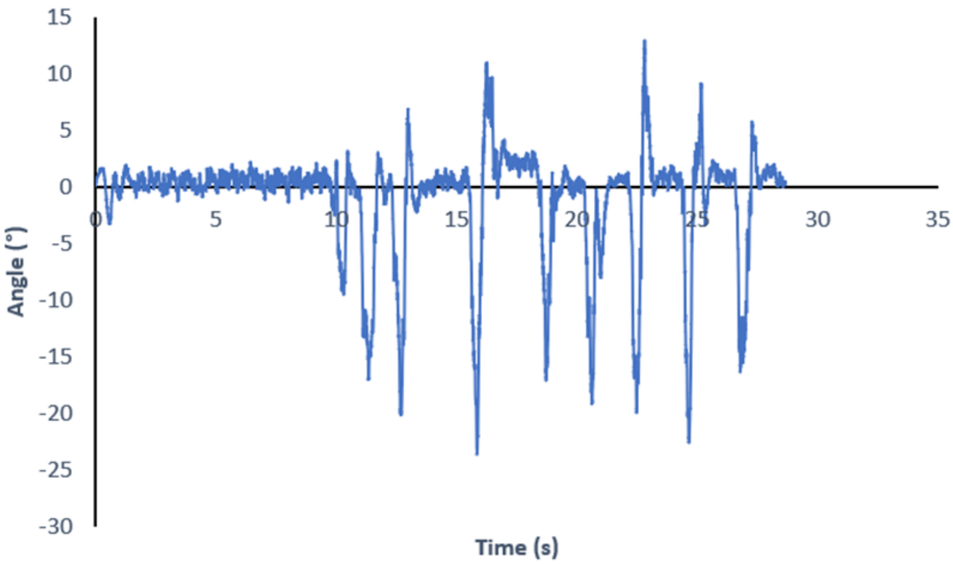

Figure 3.32: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 20.000$

Note: Too much damping, slower response.

Test 16

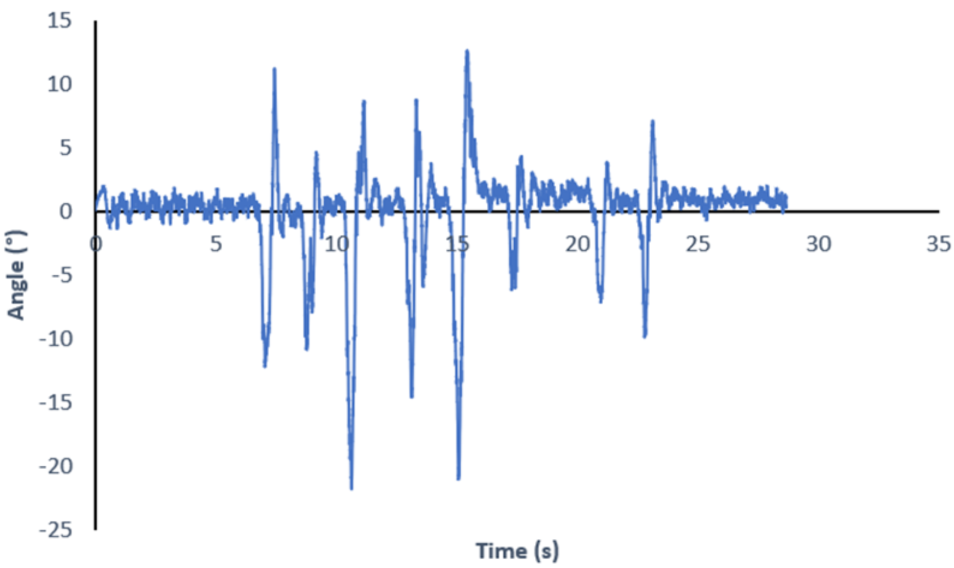

Figure 3.33: With $K_p = 1.800$, $K_i = 0.000$ and $K_d = 18.000$

Note: Stable and fast, almost ideal.

Test 17 (Final PID Performance)

To illustrate the effectiveness of the selected PID gains, the system response from test 17 was plotted on excel . This configuration, with $K_p = 1.8$, $K_i = 0.001$, and $K_d = 18$, demonstrated stable and responsive behavior. The plot shows how the drone reacts to disturbances and returns to equilibrium, highlighting the improved control performance.

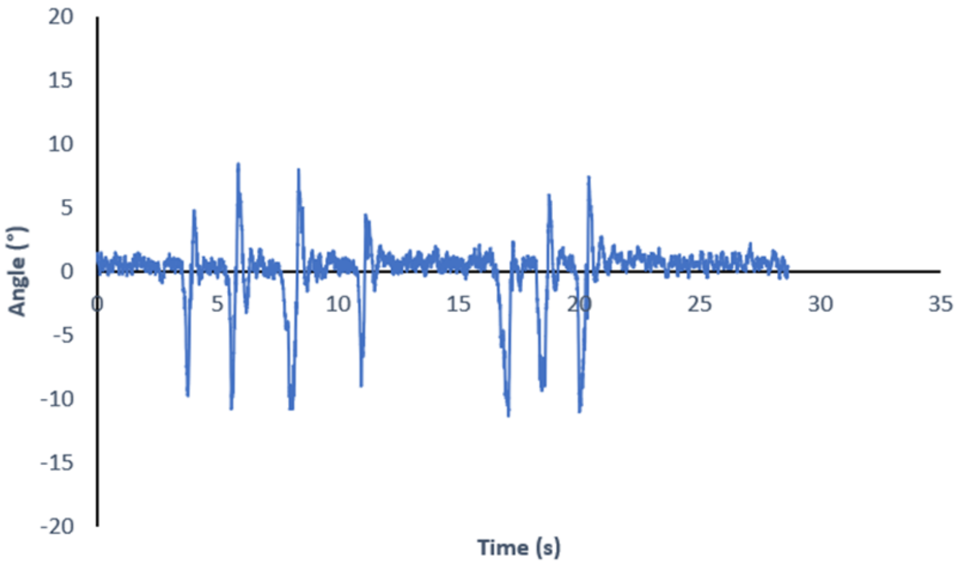

Figure 3.34: With $K_p = 1.800$, $K_i = 0.000$ and $K_d = 18.000$

3.3.1.1. PID Tuning Results Summary

During the PID tuning phase, various gain values were tested to optimize the drone's stabilization. After multiple iterations, test 17 was identified as the most effective configuration, with $K_p = 1.8$, $K_i = 0.001$, and $K_d = 18$. These parameters offered a well-balanced trade-off between responsiveness and stability. The system responded quickly to disturbances, with minimal overshoot and oscillation, and maintained steady control during hover and small maneuvers. The low integral gain helped reduce steady-state error without introducing instability, while the relatively high derivative gain efficiently damped high-frequency noise and sudden input changes. These experimental results confirmed that both our algorithm and our circuit design were correct and effective for achieving stable and responsive flight control. This success enabled us to move forward and implement full 3D stabilization, expanding control over all three axes of motion.

3.3.1.2. Use of the Remote Controller

Instead of using a traditional potentiometer, a custom-built remote controller was employed to command the drone. Before conducting stabilization tests, all buttons and the transmitter signal were verified to ensure proper communication. The joystick was mapped to control angular movement, allowing manual input in the range of approximately –10° to +10°. This setup provided a more realistic way to simulate user commands and observe the drone's response under real control conditions.

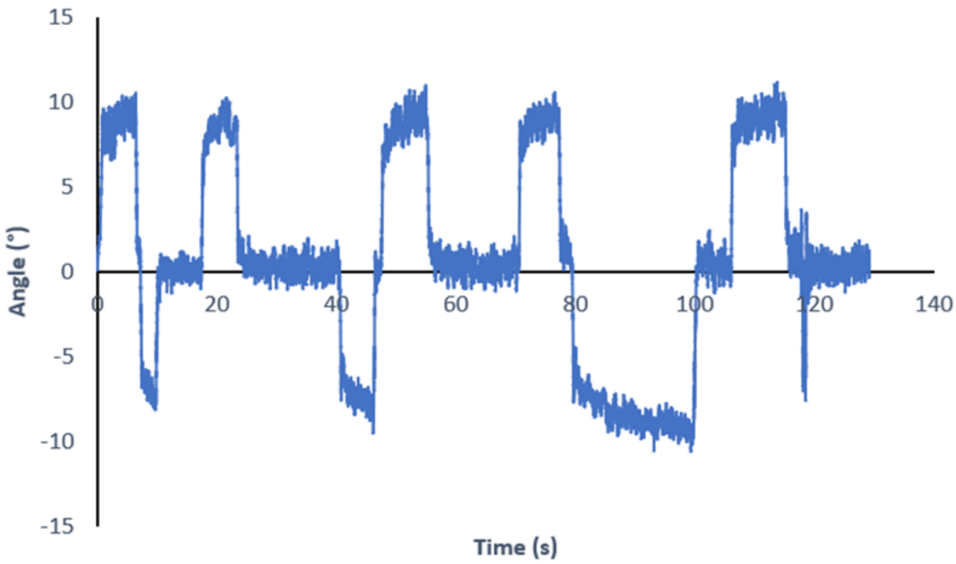

Figure 3.35: Angular Response of the Drone to Manual Joystick Inputs Over Time

3.3.1.3. Stability using FlySky FS-i6X Transmitter

A dedicated interface circuit was designed to control motors speed and to connect the FlySky receiver to the flight stability system. We implemented a circuit on a breadboard, incorporating two main functionalities:

1.Manual Control of Brushless Motors:

A PIC16F877 microcontroller, referred to as "Signal", was used to generate the PWM signal required to drive two brushless motors.The motor speed was adjusted using a manual potentiometer.The characteristics of the PWM signal pulse width, period, duty cycle, and frequency were measured and displayed on an LCD screen by a second microcontroller, designated as "Oscillo".

2.Control via FlySky Receiver:

A FlySky receiver was integrated into the breadboard circuit. The "Oscillo" microcontroller was subsequently connected to the flight controller.Once powered on, the transmitter enabled control of the hexacopter's roll axis by transmitting three predefined angle values: –10°, 0°, and 10°. The corresponding responses were monitored and displayed in real time on the LCD screen.

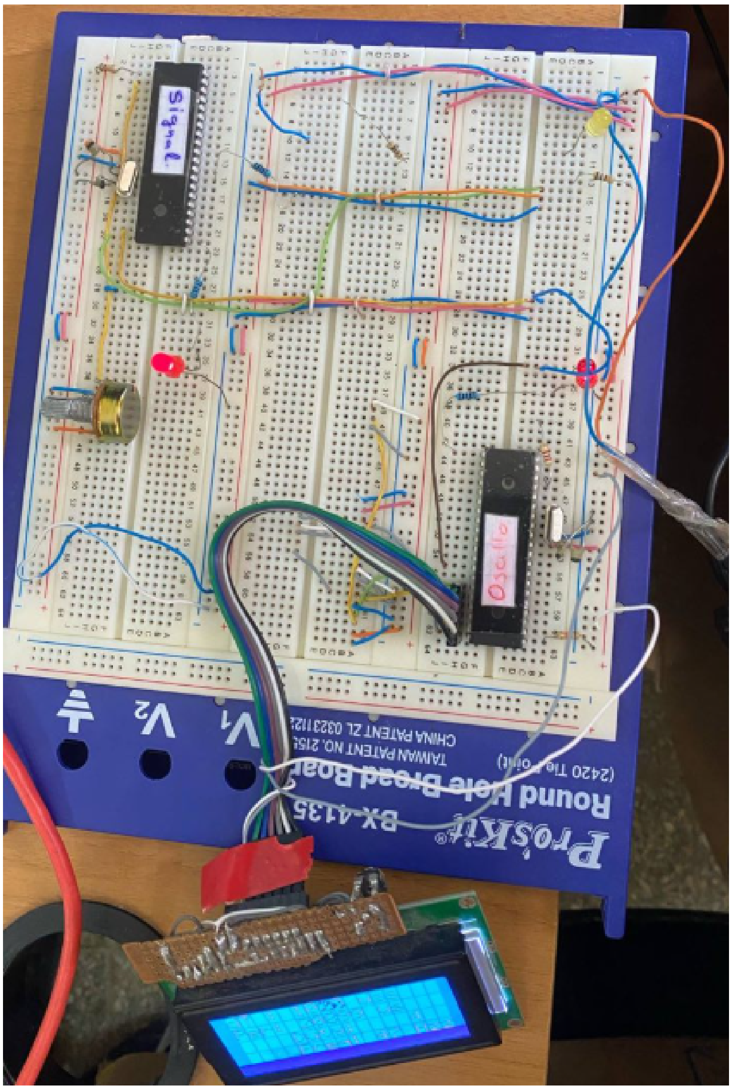

Figure 3.36: Electronic Interface Circuit for Stability Testing Using the FlySky FS-i6X Transmitter

3.3.1.4. Stability input via FlySky iA6B Receiver

This setup allowed us to vary the desired angle using RCFlysky within a limited range of - 10 °, 0 °, and +10 °, providing a simple and intuitive way to test the stability control loop.

Later, we transitioned to a more realistic input method by integrating the FlySky RC receiver. Our goal was to replace the manual potentiometer with a joystick channel from the RC transmitter, allowing us to test the system under conditions closer to actual flight scenarios.

After examining the output signals from the receiver, we identified the correct joystick channel by monitoring the PWM signal changes while interacting with the transmitter. We focused on identifying the channel that produced three distinct signal states corresponding to our desired test range (e.g., left, center, right).

The correct signal was found on the yellow wire, which corresponds to the right button/joystick on the FlySky transmitter. This channel was then mapped as the roll input in our control algorithm, replacing the potentiometer and allowing the user to command the hexacopter more naturally via the transmitter.

Figure 3.37: Stability Input via FlySky iA6B Receiver

3.3.2 One Axis stability using Four brushless motors

The assembly of the quadcopter circuit involved carefully mounting the four motors aligned along a single axis to ensure balanced thrust and stability. All electronic components, including the two dsPIC microcontrollers, were securely mounted on the frame and connected according to the schematic. Special attention was paid to minimize vibrations and ensure a symmetric weight distribution to maintain consistent behavior during the testing.

For the test setup, the quadcopter was placed on a stable platform that allowed free rotation around the tested axis, usually the roll axis. This configuration enabled us to isolate and evaluate the controller's performance under semi-constrained conditions. We monitored the system reaction to the transmitter control input, measured by the MPU6050 sensor, and verified that the motor responses were consistent with the expected output of the PID controller.

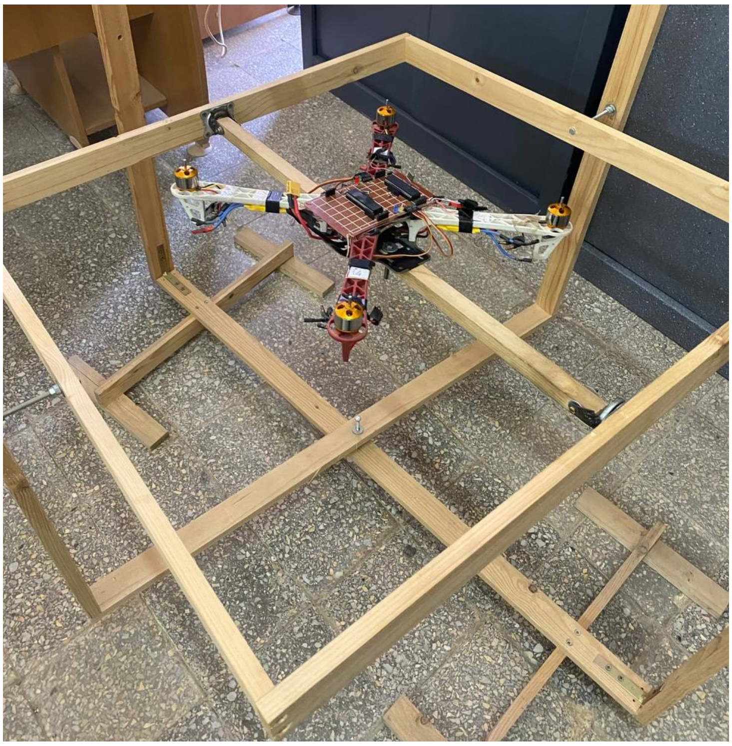

Figure 3.38: One Axis stability using Four brushless motors

3.3.3 PID Tuning and Testing Overview, Roll Axis (Four Motors)

To ensure precise control and flight stability of the quadcopter, the roll axis was selected for initial PID tuning. This axis is directly influenced by the speed difference between pairs of opposing motors. Although all four motors are active, the roll movement is primarily adjusted by increasing or decreasing thrust on motors M1 and M2 versus M3 and M4.

The PID controller was iteratively tuned by adjusting the Proportional (K_p), Integral (K_i), and Derivative (K_d) gains. The objective was to achieve:

- Minimal overshoot.
- Fast settling time.
- Stable oscillation-free response.
- Good disturbance rejection.

Each configuration was tested under the same conditions to ensure consistency. The behavior of the quadcopter's roll axis was observed through logs, real-time angle plots, and visual flight testing.

The table below summarizes all test cases, their outcomes, and remarks.

Test	K_p	K_i	K_d	**Observed Response**	**Comment**
01	0.500	0.000	0.000	Very slow, no correction	System too passive
02	1.000	0.000	0.000	Still unstable, slight drift	Low proportional gain
03	1.500	0.000	0.000	Minor improvement	Still underdamped
04	1.300	0.000	0.000	Reactive but unstable	No damping effect
05	1.300	0.000	1.000	Slightly better	Still overshoots
06	1.300	0.000	5.000	Fast response, but oscillatory	Missing derivative control
07	1.300	0.000	10.000	Too aggressive, overshooting	K_d too high
08	1.300	0.000	15.000	Very unstable	Uncontrolled response
09	1.300	0.001	15.000	Slight damping	Introduction of K_i begins to correct steady-state errors.
10	1.300	0.003	15.000	Smoother response	More integral action improves stability, reduces oscillations.
11	1.300	0.005	15.000	Better balance	K_i now significantly contributes to stabilization and precision.
12	1.300	0.000	18.000	Slight damping	Small improvement
13	1.300	0.005	18.000	Less oscillation, smoother	Best result – balanced performance

Table 3.3: PID Parameter Testing and Observed Responses

Test 01

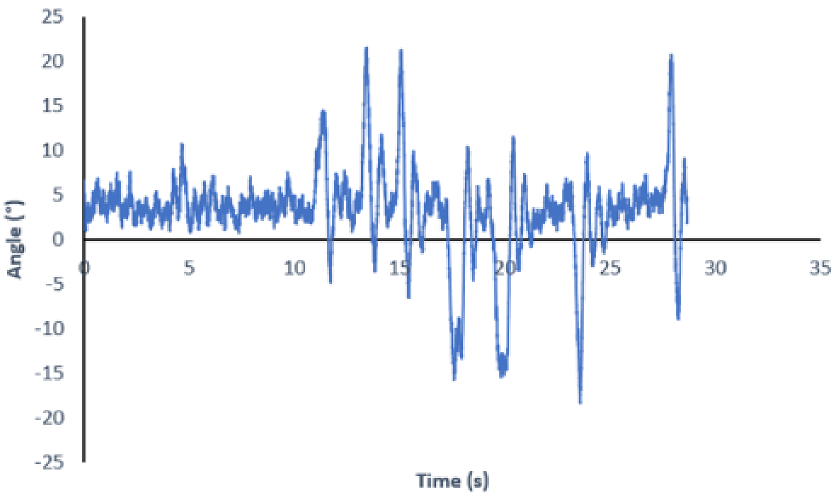

Figure 3.39: With $K_p = 0.500$, $K_i = 0.000$ and $K_d = 0.000$

Note: The system is too passive, with a very slow response.

Test 02

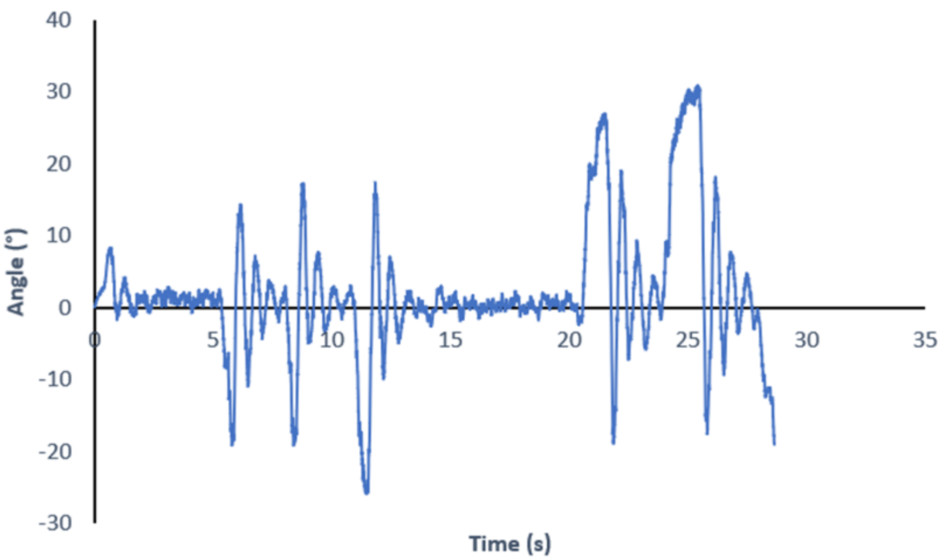

Figure 3.40: With $K_p = 1.000$, $K_i = 0.000$ and $K_d = 0.000$

Note: The system remains unstable and exhibits slight drift.

Test 03

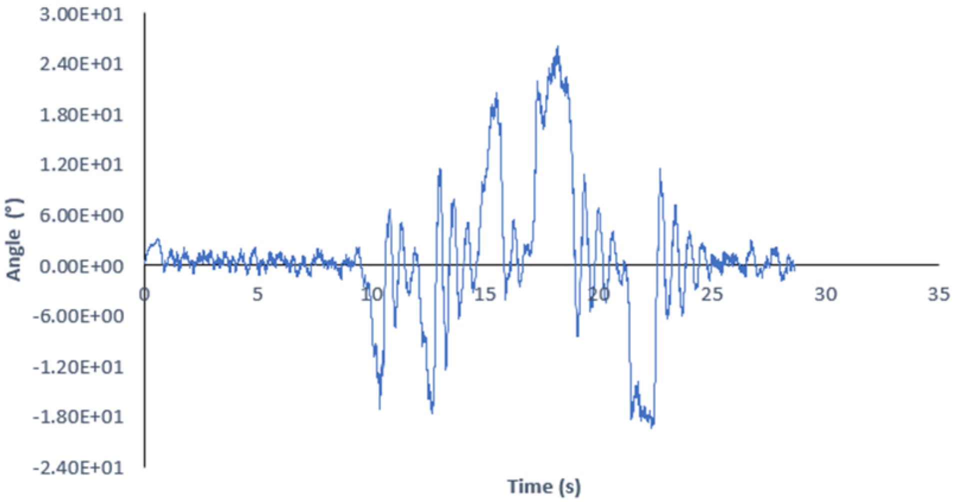

Figure 3.41: With $K_p = 1.500$, $K_i = 0.000$ and $K_d = 0.000$

Note:Although improved slightly, the system is still underdamped.

Test 04

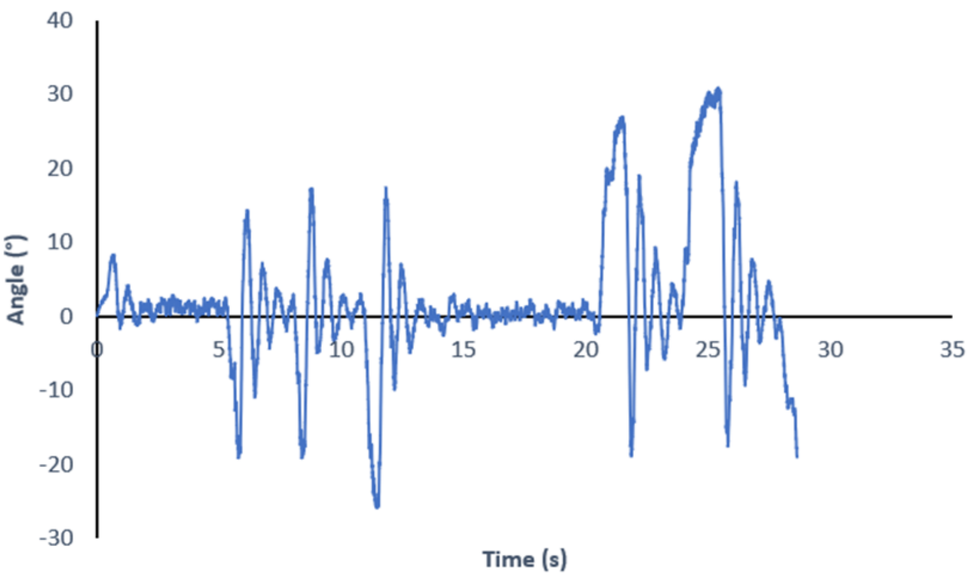

Figure 3.42: With $K_p = 1.300$, $K_i = 0.000$ and $K_d = 0.000$

Note: The system reacts quickly but lacks stability and damping.

Test 05

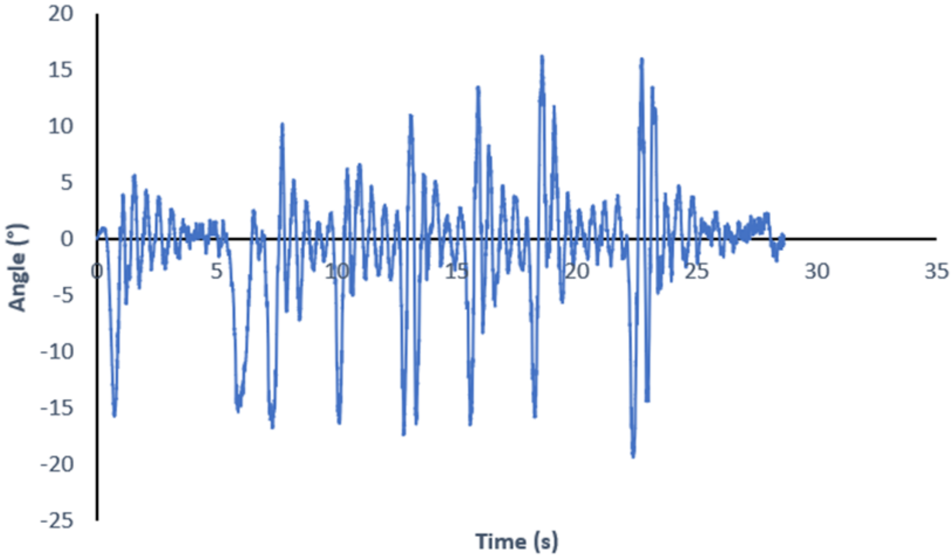

Figure 3.43: With $K_p = 1.300$, $K_i = 0.000$ and $K_d = 1.000$

Note: The response is slightly improved, yet it still overshoots.

Test 06

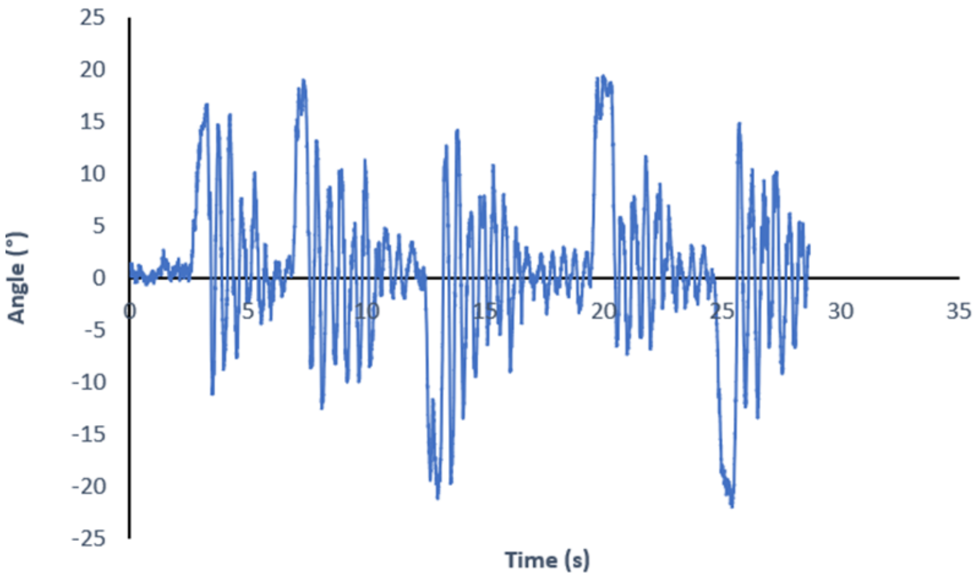

Figure 3.44: With $K_p = 1.300$, $K_i = 0.000$ and $K_d = 5.000$

Note: Faster response observed, oscillatory behavior persists.

Test 07

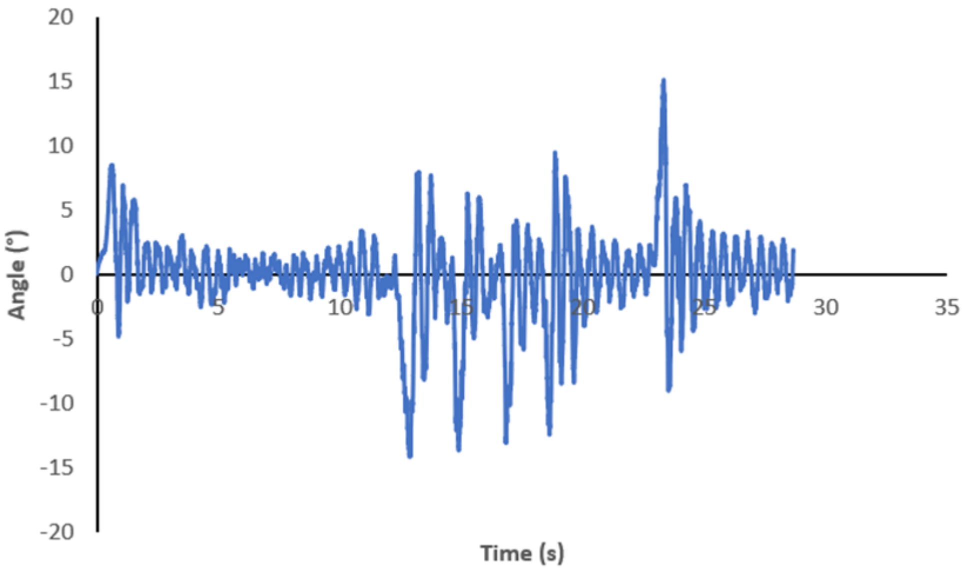

Figure 3.45: With $K_p = 1.300$, $K_i = 0.000$ and $K_d = 10.000$

Note: The system is too aggressive and produces a large overshoot.

Test 08

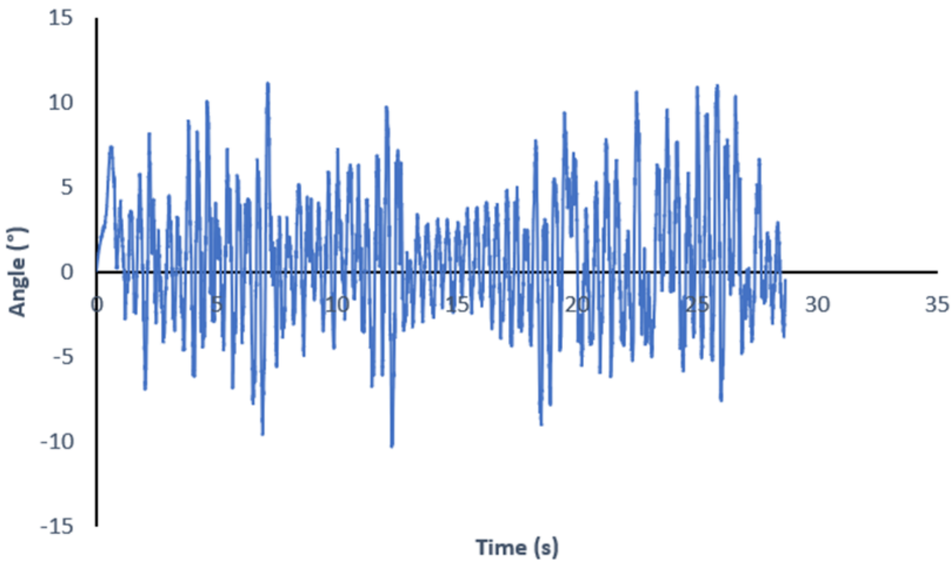

Figure 3.46: With $K_p = 1.300$, $K_i = 0.000$ and $K_d = 15.000$

Note: The system is highly unstable and lacks control.

Test 09

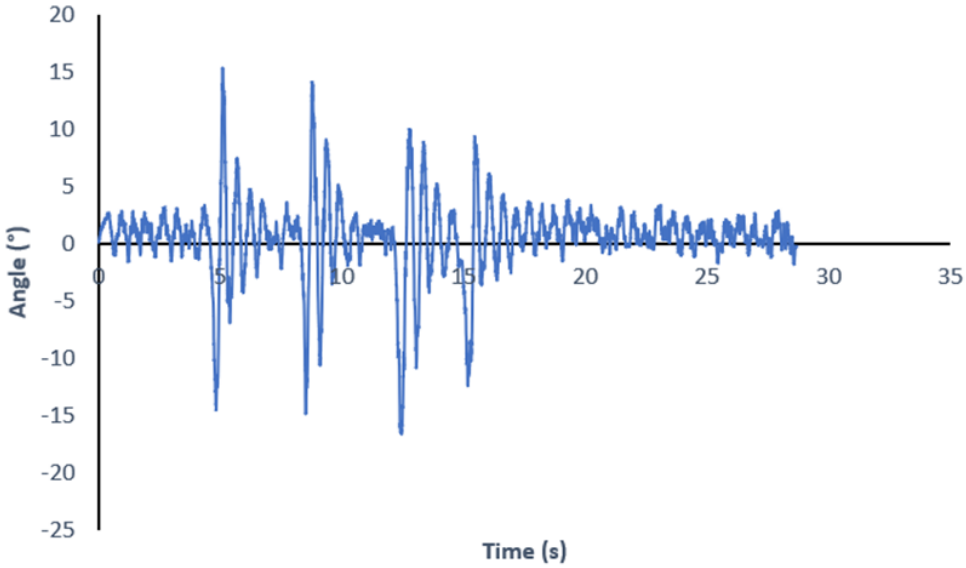

Figure 3.47: With $K_p = 1.300$, $K_i = 0.001$ and $K_d = 15.000$

Note: Light damping begins after introducing a small K_i.

Test 10

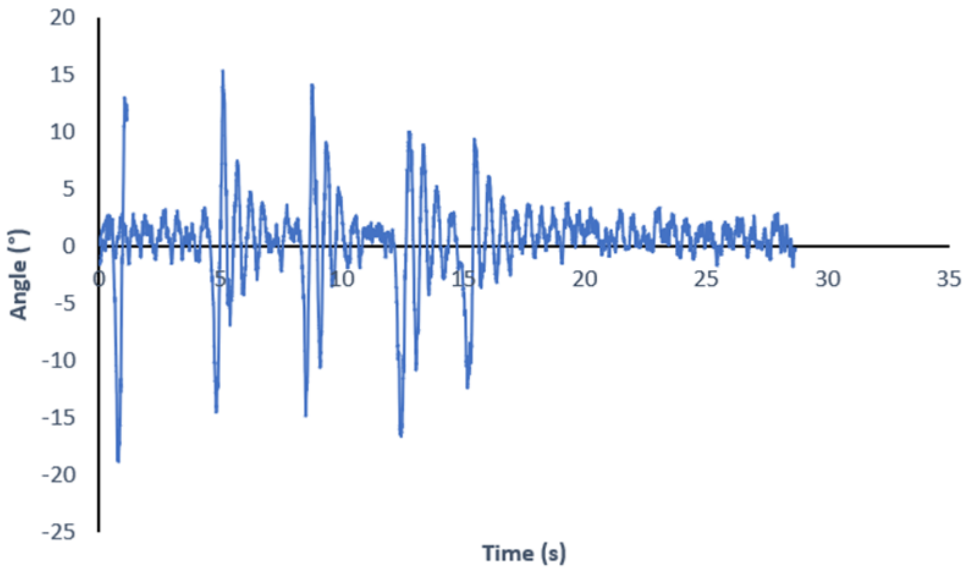

Figure 3.48: With $K_p = 1.300$, $K_i = 0.003$ and $K_d = 15.000$

Note: The system exhibits smoother behavior due to increased K_i .

Test 11

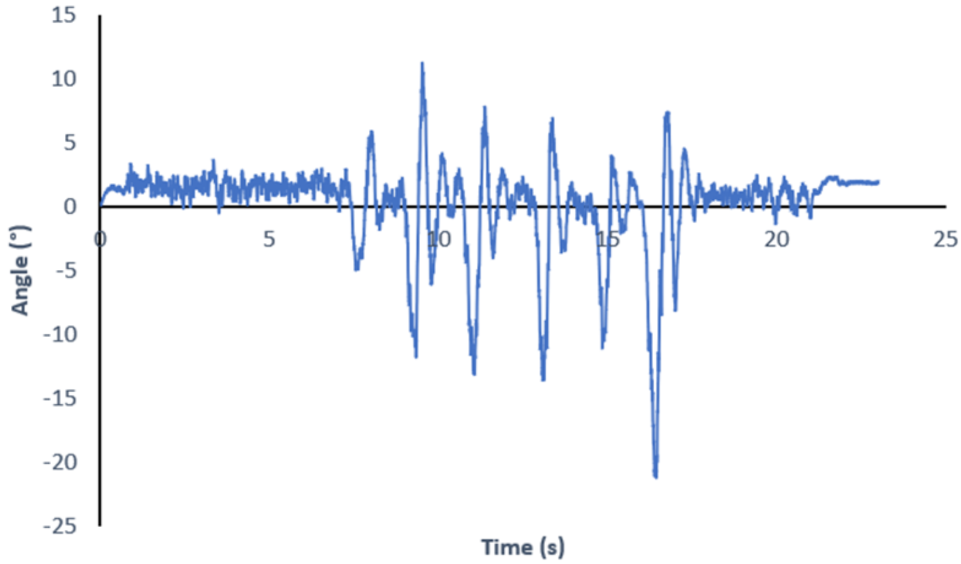

Figure 3.49: With $K_p = 1.300$, $K_i = 0.005$ and $K_d = 15\ .000$

Note: System balance improves with K_i contributing to control and accuracy.

Test 12

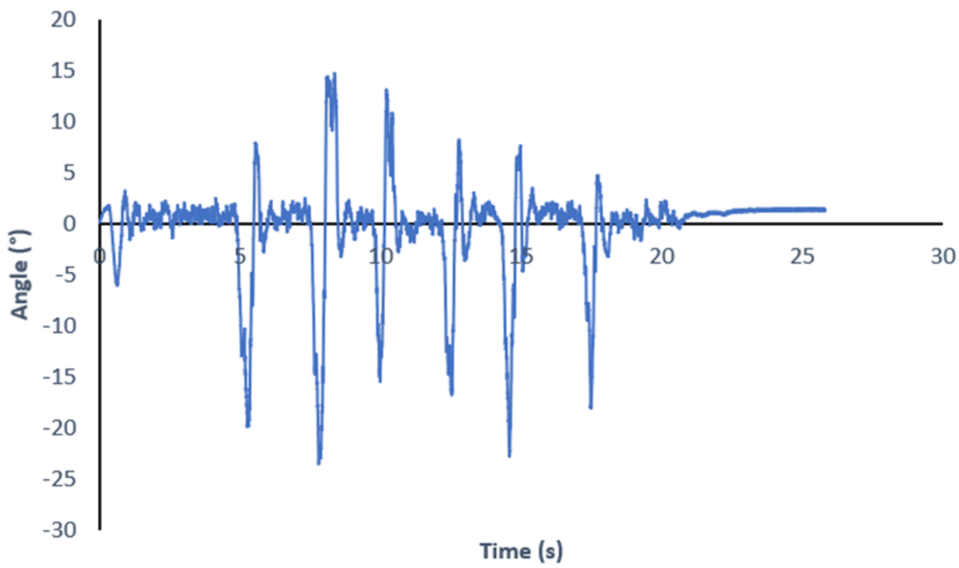

Figure 3.50: With $K_p = 1.300$, $K_i = 0.000$ and $K_d = 18.000$

Note: The system shows slight damping and minor improvement.

Test 13 (Final PID Performance)

To illustrate the effectiveness of the selected PID gains, the system response from Test 13 was plotted in Excel.

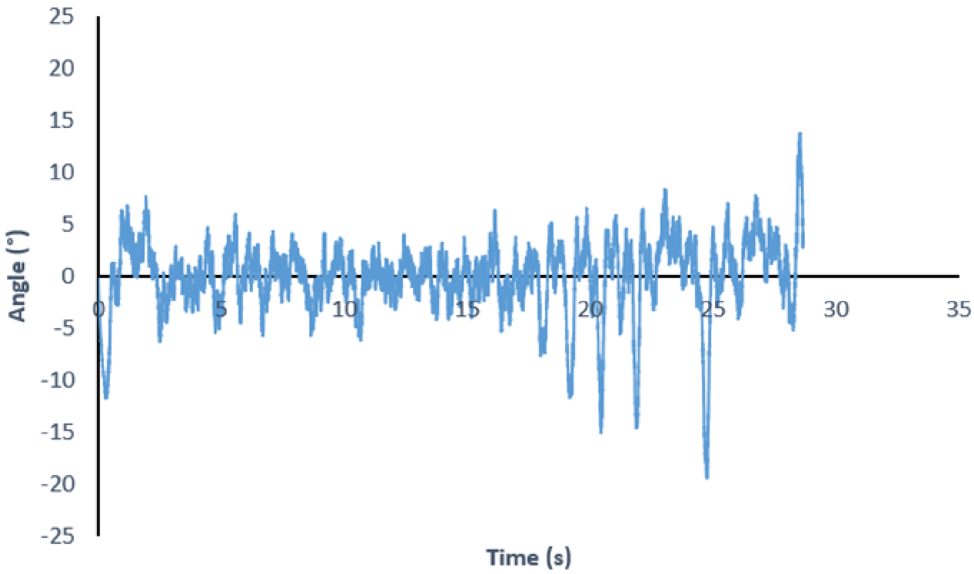

Figure 3.51: With $K_p = 1.300$, $K_i = 0.005$ and $K_d = 18.000$

This configuration, with $K_p = 1.300$, $K_i = 0.005$, and $K_d = 18.000$, demonstrated a stable and responsive behavior.The plot clearly shows how the quadcopter reacts to external disturbances and quickly returns to equilibrium, highlighting the improved control performance and smoother system dynamics.

3.3.3.1. PID Tuning Results Summary

During the PID tuning process, several combinations of gain values were tested to achieve optimal stabilization of the drone.Through successive trials and careful observation of the system's behavior, Test 13 emerged as the most effective configuration.

With $K_p = 1.300$, $K_i = 0.005$, and $K_d = 18.000$, the controller provided a balanced trade-off between speed and stability.The proportional gain ensured sufficient responsiveness, while the low integral gain minimized steady-state error without introducing instability. The derivative gain was critical in damping oscillations and handling abrupt changes in motion or disturbances.This tuning resulted in minimal overshoot, reduced oscillation, and stable hover performance, making Test 13 the best compromise between accuracy, responsiveness, and robustness.It allowed the drone to maintain smooth control during both steady flight and corrective maneuvers.

3.4 Conclusion

This chapter presented a progressive and experimental approach to achieving the stabilization of a multicopter, starting with simplified configurations and gradually increasing in complexity.

We began by analyzing the stability of a single axis of a hexacopter using two motors, which allowed us to validate the basic principles of attitude control. A potentiometer was then integrated to simulate a manual angle command, enabling controlled variations and observation of the system's response.

In the next phase, we developed an electronic setup using two PIC microcontrollers: one was responsible for generating the PWM signal simulating motor control, while the other, paired with an LCD display, acted as a visual analyzer. This "mini oscilloscope" displayed in real time key parameters such as period, duty cycle, and motor speed, providing valuable insight into the behavior of the control signal.

The following step involved the integration of a FlySky transmitter and its receiver module, allowing remote control of the system. The receiver circuit was capable of reading and decoding the PPM signal and then issuing angle commands to the hexacopter within a range of +10° to -10°, simulating a realistic manual input scenario.

Finally, we transitioned to a quadcopter with single-axis control using four motors. This stage required precise PID gain tuning to ensure the system's dynamic stability. Multiple configurations were tested to achieve an optimized response, balancing reaction speed, overshoot, oscillations, and steady state stability.

Overall, this practical work successfully validated the entire stabilization loop of a drone, from control signal generation to effective stabilization, including signal acquisition, processing, and dynamic response.

General Conclusion and Perspectives

General Conclusion

The study has explored the complete design, construction, and validation of a custom flight controller tailored for quadcopter type Unmanned Aerial Vehicles (UAVs), using PIC microcontrollers as the central processing elements. The study integrated theoretical concepts in aeronautical control, embedded systems design, sensor interfacing, and real time testing to address the critical challenges associated with drone stabilization and control.

The theoretical framework established a detailed understanding of UAV systems, including their aerodynamic classifications and the structural differences between fixed wing, rotary wing, and hybrid configurations. A particular focus was placed on flight control systems (FCS), highlighting their indispensable role in maintaining attitude, responding to user inputs, and ensuring autonomous behavior. Stability enhancing strategies such as PID control, model based approaches, sensor fusion, and intelligent control methods were reviewed to evaluate their impact on performance, robustness, and adaptability.

In parallel, the investigation into microcontroller based embedded systems emphasized the selection of a cost-effective and reliable hardware platform. PIC microcontrollers were chosen for their maturity, flexibility, and developer support. Development environments such as MPLAB and MikroC were assessed for their ability to facilitate firmware design and debugging. Emphasis was also placed on sensor integration specifically the MPU 6050 inertial measurement unit as a critical element for acquiring real time data to feed the control algorithm.

A significant contribution of this work lies in the practical realization of a fully functional flight control prototype. A modular architecture was constructed, incorporating dual PIC microcontrollers one responsible for PID control and signal generation, and the other configured as a real time analyzer with LCD visualization. A progressive experimental methodology was adopted, initially validating control principles using a simplified one axis setup, then gradually transitioning to a complete quadcopter configuration with four brushless motors. The integration of a FlySky radio transmitter and receiver module enabled manual control of angular position, simulating realistic operating conditions. The final prototype demonstrated effective stabilization through iterative PID tuning and real time evaluation of performance parameters such as duty cycle, overshoot, and system responsiveness.

The results obtained confirm the feasibility of developing a low cost, modular, and reliable drone flight control platform using general purpose microcontrollers. The system exhibited favorable behavior in terms of control accuracy and dynamic stability, thereby validating the design choices and implementation strategy adopted throughout the project.

Perspectives

While the goals of this thesis were met, several promising directions for future improvement and development can be identified: First, expanding the current implementation to support full three-axis control (roll, pitch, and yaw) would enable complete spatial stabilization, thereby achieving full flight autonomy and greater maneuverability.

Then, the integration of advanced sensor fusion algorithms, such as Kalman filtering or complementary filters, could significantly enhance the accuracy and robustness of orientation estimation, particularly in the presence of sensor noise or external disturbances.

Next, more sophisticated control methodologies such as Sliding Mode Control (SMC), Linear Quadratic Regulation (LQR), or Model Predictive Control (MPC) should be explored to improve the system's performance in rapidly changing or uncertain flight conditions.

In addition, incorporating machine learning techniques may allow for adaptive gain tuning, predictive modeling of disturbances, or optimization of the control response based on real time data, paving the way for intelligent flight controllers.

Furthermore, developing wireless telemetry interfaces and a ground control station would provide operators with the ability to monitor system variables in real time, remotely adjust parameters, and log flight data for post mission analysis.

Subsequently, the design and fabrication of a custom printed circuit board (PCB) that integrates all control, sensing, and power management components would lead to a more compact, lightweight, and reliable flight controller suitable for deployment in real world UAV applications.

Finally, the integration of GPS based navigation systems and the implementation of autonomous mission planning algorithms would allow the platform to support complex tasks such as waypoint tracking, surveying, mapping, and automated delivery.

This work contributes a robust foundation for the development of embedded flight control systems and opens the door to further innovation in the field of autonomous and intelligent unmanned aerial platforms.

Bibliography

[1] Lav Gupta, Raj Jain, and Gabor Vaszkun. Survey of important issues in uav communication networks. *IEEE communications surveys & tutorials*, 18(2):1123–1152, 2015.

[2] Azade Fotouhi, Haoran Qiang, Ming Ding, Mahbub Hassan, Lorenzo Galati Giordano, Adrian Garcia-Rodriguez, and Jinhong Yuan. Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security challenges. *IEEE Communications surveys & tutorials*, 21(4):3417–3442, 2019.

[3] Georgy Skorobogatov, Cristina Barrado, and Esther Salamí. Multiple uav systems: A survey. *Unmanned Systems*, 8(02):149–169, 2020.

[4] J.C. Rosser Jr., V. Vignesh, B.A. Terwilliger, and B.C. Parker. Surgical and medical applications of drones: a comprehensive review. *JSLS*, 22, 2018.

[5] G. Dekoulis. *Introductory Chapter: Drones, Drones – Applications.* IntechOpen, 2018.

[6] US Department of Defense. Unmanned aerial vehicles roadmap, 2005–2030. Technical report, Department of Defense, Office of the Secretary of Defense, Washington, DC, 2005.

[7] Maria de Fatima Bento. Unmanned aerial vehicles: An overview, 2010. Technical report.

[8] Maziar Arjomandi. Classification of unmanned aerial vehicles, 2007.

[9] Aile fixe - uav model. Accessed: 2025-04-18.

[10] Syed Agha Hassnain Mohsan, Muhammad Asghar Khan, Fazal Noor, Insaf Ullah, and Mohammed H Alsharif. Towards the unmanned aerial vehicles (uavs): A comprehensive review. *Drones*, 6(6):147, 2022.

[11] Med Rida BENAISSA. Étude & implémentation d'une loi de commande pour le pilotage d'un drone quadrirotor, 2017.

[12] MH Sabour, P Jafary, and S Nematiyan. Applications and classifications of unmanned aerial vehicles: A literature review with focus on multi-rotors. *The Aeronautical Journal*, 127(1309):466–490, 2023.

[13] Nassim SACI. Simulation d'un drone sous matlab : cas d'étude quad-copter, 2016.

[14] Omar I Dallal Bashi, WZ Hasan, Norhafiz Azis, Suhaidi Shafie, and Hiroaki Wagatsuma. Unmanned aerial vehicle quadcopter: A review. *Journal of Computational and Theoretical Nanoscience*, 14(12):5663–5675, 2017.

[15] Deepak BBVL and Pritpal Singh. A survey on design and development of an unmanned aerial vehicle (quadcopter). *International Journal of Intelligent Unmanned Systems*, 4(2):70–106, 2016.

[16] Muhammad Maaruf, Magdi Sadek Mahmoud, and Alfian Ma'arif. A survey of control methods for quadrotor uav. *International Journal of Robotics and Control Systems*, 2(4):652–665, 2022.

[17] Omar I Dallal Bashi, WZ Hasan, Norhafiz Azis, Suhaidi Shafie, and Hiroaki Wagatsuma. Unmanned aerial vehicle quadcopter: A review. *Journal of Computational and Theoretical Nanoscience*, 14(12):5663–5675, 2017.

[18] What is a drone flight controller? Accessed: 2025-04-21.

[19] Weibin Gu, Kimon P Valavanis, Matthew J Rutherford, and Alessandro Rizzo. Uav model-based flight control with artificial neural networks: A survey. *Journal of Intelligent & Robotic Systems*, 100(3):1469–1491, 2020.

[20] Drone technology: Flight control systems. Accessed: 2025-04-21.

[21] Ivan Lopez-Sanchez and Javier Moreno-Valenzuela. Pid control of quadrotor uavs: A survey. *Annual Reviews in Control*, 56:100900, 2023.

[22] Chingiz Hajiyev, Halil Ersin Soken, and Sıtkı Yenal Vural. *State Estimation and Control for Low-cost Unmanned Aerial Vehicles.* Springer, 2015.

[23] Nadir Abbas, Zeshan Abbas, Xiaodong Liu, Saad Saleem Khan, Eric Deale Foster, and Stephen Larkin. A survey: Future smart cities based on advance control of unmanned aerial vehicles (uavs). *Applied Sciences*, 13(17):9881, 2023.

[24] Shubham Gandhare and B Karthikeyan. Survey on fpga architecture and recent applications. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, pages 1–4. IEEE, 2019.

[25] Si Jung Kim, Yunhwan Jeong, Sujin Park, Kihyun Ryu, and Gyuhwan Oh. A survey of drone use for entertainment and avr (augmented and virtual reality). *Augmented reality and virtual reality: empowering human, place and business*, pages 339–352, 2018.

[26] Kenichi Nishikawa, Ioana Duţan, Christoph Köhn, and Yosuke Mizuno. Pic methods in astrophysics: simulations of relativistic jets and kinetic physics in astrophysical systems. *Living Reviews in Computational Astrophysics*, 7(1):1, 2021.

[27] Daniel Yokoyama, Bruno Schulze, Fábio Borges, and Giacomo Mc Evoy. The survey on arm processors for hpc. *The Journal of Supercomputing*, 75:7003–7036, 2019.

[28] *Exploring C for Microcontrollers: A Hands-on Approach.* Jivan S. Parab, Vinod G. Shelake, Rajanish K. Kamat, Gourish M. Naik, 2007.

[29] David Calcutt, Frederick Cowan, and Hassan Parchizadeh. *8051 Microcontroller: An Applications Based Introduction.* Newnes, 2003.

[30] Martin Bates. *PIC Microcontrollers: An Introduction to Microelectronics.* unknown, 2011.

[31] D. Menesplier. Microcontrôleurs pic 16f876 et 16f877. documentation interne ele (enac). Technical report, 2001.

[32] *Make: Sensors: A Hands-On Primer for Monitoring the Real World with Arduino and Raspberry Pi.* Make Community, 2014. Paperback – July 1, 2014.

[33] Thermopedia - capteurs. Accessed: 2025-06-01.

[34] Dspace. (2024). site officiel de dspace. Accessed: 2025-06-01.

Appendix

Development Of Flight Controller For Drones By Using MCU

Startup Project under Ministerial Decree N° 1275

République Algérienne Démocratique Et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche scientifique

Université Saad Dahlab - Blida 1

Institut d'études aéronautiques et spatiales IAES

Mémoire de fin d'études

En vue de l'obtention d'Attestation de projet d'entreprise économique dans le cadre de l'arrêté ministériel N° 1275-008

Domaine : Sciences et Technologies

Département : Construction Aéronautique

Spécialité :Avionique

Intitulé du Projet :

"Development of flight controller for drones by MCU "

Code de projet : IAES/0007/2025

Projet **Startup** présenté dans le cadre de l'arrête ministériel **"1275", assurée par Institut d'Aéronautique et des Etudes Spatiales IAES**

Présenté par :	**Encadré par :**	**Formateurs :**
1- DORBANI Nahr El Imene	**Encadreur :** MCA. AMRI Redha	MCB. KRIM Mohamed
2- ZOUAOUI Sara	**Co-encadreur :** PR. LAGHA Mohand	MCB. LEBSIR Abdelkadir

Année académique : 2024/2025

Annexe

Projet startup 1275 - 008

République Algérienne Démocratique Et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche scientifique

Université Saad Dahlab - Blida 1

Institut d'études aéronautiques et spatiales IAES

Mémoire de fin d'études

En vue de l'obtention d'Attestation de projet d'entreprise économique dans le cadre de l'arrêté ministériel N° 1275-008

Domaine : Sciences et Technologies
Département :Construction Aéronautique
Spécialité : Avionique

Intitulé du Projet :

" Development of Flight Controller for Drones by using MCU "

Projet de **Startup** présenté dans le cadre de l'arrête ministériel **"1275-008 ",** assurée par Institut d'Aéronautique et des Etudes Spatiales. IAES

Devant le jury composé de :

Membres des jurys	Nom et prénom	Grade	Etablissement	Signature
Président	KRIM Mohamed	MCB	Univ. Blida 1	
Encadreur	AMRI Redha	MCA	Univ. Blida 1	
Co-encadreur	LAGHA Mohand	PROFESSEUR	Univ. Blida 1	
Examinateur	BENOUARED Abdelhalime	MAA		
Représentant incubateur	TIDJANI Naoual	MCA	Univ. Blida 1	
Représentant CDE	/	/	/	/
Représentant du partenaire économique	KECHIDA Ahmed	DIRECTEUR	CRTI	
Responsable du Centre d'Appui à la Technologie et à l'Innovation	/	/	/	/

Année académique : 2024/2025

Carte d'information

A propose de l'équipe d'encadrement du groupe de travail

Equipe de l'Encadrement du Groupe principale	SPECIALITE	FACULTE	ETABLISMENT
Encadrant 01 : MCA. AMRI Redha	Propulsion	IAES	Univ. Blida 1
Encadrant 02 : Pr. LAGHA Mohand	Avionique	IAES	Univ. Blida 1

A propose de l'équipe du projet du groupe de travail

Equipe du projet du Groupe principale (à titre indicatif)	SPECIALITE	FACULTE	ETABLISMENT
Etudiante 01 : DORBANI Nahr El Imene	Avionique	IAES	Univ. Blida 1
Etudiante02 : ZOUAOUI Sara	Avionique	IAES	Univ. Blida 1

Introduction :

L'importance de toute étude scientifique est évidente dans l'étendue de sa contribution à mettre en évidence le problème à résoudre après avoir défini ses variables avec précision. Cela permet de déterminer les dimensions de l'étude et d'évaluer sa valeur théorique et pratique.

Contrairement aux solutions commerciales comme le Pixhawk, qui fonctionnent comme des boîtes noires et rendent le système dépendant de technologies étrangères, mon projet vise le développement d’un contrôleur de vol (Flight Controller) 100% personnalisé à partir d’un microcontrôleur (MCU). L’objectif est de concevoir un système totalement transparent, dans lequel les signaux ne sont ni piratables ni cryptés à l'insu de l’utilisateur, garantissant ainsi la souveraineté numérique et la sécurité des données de vol. Ce système est pensé pour s'adapter à des drones polyvalents pouvant opérer dans plusieurs domaines d’application (surveillance, agriculture, recherche, transport, etc.). L’accent est mis sur la stabilité dynamique, avec l’implémentation d’algorithmes avancés de contrôle (comme le PID, LQR, etc.) permettant des performances optimales, même dans des environnements instables ou perturbés. Ce projet représente une contribution concrète à la démocratisation des systèmes de pilotage de drones, avec une orientation vers l’autonomie technologique, la sécurité des communications, et la flexibilité d’adaptation à différents usages, notamment dans les secteurs sensibles où les solutions propriétaires sont inacceptables.

L’objectif d’étude :

L’objectif principal de ce projet est de concevoir et développer un contrôleur de vol (Flight Controller) 100 % personnalisé, basé sur un microcontrôleur (MCU), destiné à équiper des drones à voilure fixe et multirotors, capables de s’adapter à plusieurs domaines d'application.

Ce contrôleur vise à offrir une solution souveraine, ouverte et totalement transparente, en opposition aux systèmes existants de type « boîte noire » comme le Pixhawk. Il intègre des algorithmes de contrôle avancé assurant une stabilité dynamique optimale, même dans des conditions complexes.

En plus de la maîtrise complète du traitement du signal et des données embarquées, le système garantit la confidentialité et la sécurité des communications, en éliminant les risques de piratage ou de verrouillage propriétaire.

Ce projet a pour ambition de fournir une plateforme technologique évolutive, adaptable aux besoins de la recherche scientifique, la sécurité civile, l’agriculture de précision, la cartographie, et d’autres secteurs stratégiques, tout en assurant un haut niveau de performance, de fiabilité et d’indépendance technologique.

Obstacles à l'étude :

Lors de la préparation de notre mémoire de projet startup, plusieurs obstacles ont été rencontrés. L'un des plus importants a été la complexité liée à la programmation, un aspect essentiel de la mise en œuvre technique du projet.

Cette difficulté a initialement freiné notre progression et limité le temps consacréà la recherche, à la rédaction et à la réalisation concrète du travail. Heureusement, ce problème a pu être surmonté grâce à l'encadrement et au soutien technique des enseignants, qui ont su nous orienter et nous aider à débloquer les points critiques du développement.

Approche de l'étude :

De nombreux chercheurs considèrent que la démarche scientifique est essentielle pour atteindre des résultats objectifs et révéler des vérités dans divers domaines technologiques. Elle repose sur un ensemble de règles méthodologiques rigoureuses qui guident le travail du chercheur, de la formulation du problème jusqu'à l'obtention de résultats concrets.

Dans le cadre de cette étude, orientée vers la conception et le développement d'un contrôleur de vol (flight controller) autonome, sécurisé et totalement conçu en interne, une démarche scientifique structurée a été adoptée. L'objectif étant de proposer une alternative fiable, transparente et souveraine aux solutions existantes du marché, telles que le Pixhawk, souvent considérées comme des boîtes noires difficiles à personnaliser ou à auditer.

L'étude s'inscrit dans une recherche à la fois descriptive et analytique, particulièrement adaptée à une startup technologique émergente développant une solution innovante dans le domaine des drones. L'approche descriptive permet de présenter les caractéristiques, les fonctionnalités attendues et les contraintes techniques du système embarqué à concevoir.

L'approche analytique, quant à elle, permet d'examiner de manière approfondie les données issues des tests de vol, de la stabilité dynamique, de la réponse du contrôleur, ainsi que de la sécurité des signaux et des algorithmes embarqués.

La combinaison de ces deux approches favorise une compréhension complète du problème, tout en apportant des solutions concrètes et des recommandations techniques pertinentes pour le développement d'une plateforme de vol indépendante, optimisée, et ouverte à l'innovation.

Technique utilisée et outils de collecte de données :

Dans cette recherche, les outils requis pour le développement algorithmique d'un contrôleur de vol autonome et sécurisé, rivalisant avec les solutions existantes comme le Pixhawk, ont été utilisés. L'approche adoptée s'inscrit dans le cadre d'un plan de développement technologique orienté startup, qui a constitué un pilier essentiel de l'étude. Ce plan a permis d'optimiser la collecte des données système (capteurs inertiels, signaux de commande, réponses dynamiques), en s'appuyant sur une méthode structurée incluant un ensemble de paramètres et de critères définis par le chercheur. Ces éléments ont servi à analyser, concevoir et affiner l'architecture matérielle et logicielle du contrôleur, avec pour objectif d'assurer une stabilité dynamique élevée, une intégrité des signaux, et une cybersécurité embarquée. Les résultats obtenus ont permis d'apporter des réponses précises et des solutions concrètes aux défis techniques identifiés, tout en clarifiant les aspects critiques liés à la conception d'un système de vol intelligent, personnalisable et souverain.L'observation indispensable était invoquée dans toute étude, quelle que soit la recherche scientifique.

Définition de projet startup :

Une startup est une jeune entreprise innovante en phase de démarrage qui se caractérise par son fort potentiel de croissance et son ambition de perturber ou de révolutionner un marché existant. Elle se distingue par son caractère innovant, son agilité et sa recherche constante de solutions novatrices pour répondre aux besoins et aux demandes changeants du marché. Les startups sont souvent fondées par des entrepreneurs passionnés qui cherchent à résoudre des problèmes spécifiques en proposant des produits, des services ou des technologies disruptifs. Elles sont généralement axées sur la croissance rapide et l'expansion internationale, et sont souvent soutenues par des investisseurs et des incubateurs qui fournissent des ressources financières, des conseils stratégiques et un réseau d'experts pour favoriser leur développement. Les startups peuvent opérer dans divers secteurs, tels que la technologie, la santé, les services financiers, l'énergie, etc., et ont le potentiel de devenir des acteurs majeurs et influents de l'économie.

Business Model Canvas

Business Model Canvas - *BMC*

Porteurs de projet:	*Promoteurs :*	*Code de projet:*
1- DORBANI Nahr El Imene 2- ZOUAOUI Sara	P. AMRI Rehda CO-P. LAGHA MOHAND	IAES/0007/2025

Projet Startup : Development of flight controller for drones by MCU

<table>
<tr><th>Partenaires clés :</th><th>Activités Clés :</th><th>Propositions de valeur :</th><th>Relation avec les clients :</th><th>Segments de Clientèle :</th></tr>
<tr><td rowspan="3">-Universités et laboratoires en électronique et IA
-Fournisseurs de composants(capteurs, caméras, moteurs, batteries, etc.)
-Ministères (Environnement,Agriculture,Intérieur)
- ONG environnementales et institutions de protection
-Distributeurs / intégrateurs tech</td><td>--Conceptionmatérielle (dsPIC,IA)
-Développement logiciel embarqué
-Tests terrain multi-secteurs-Personnalisation par cas d’usage
-Formation et support client</td><td rowspan="3">- Drones modulaires, intelligents et solaires
- Autonomie et IA embarquée
-Polyvalence sectorielle
- Fabrication locale, économique
-Écologiques et faciles à maintenir</td><td>-Nous croyons que la construction des relations solides avec nos clients est la clé de notre succès.
-Fournir un support client de qualité pour répondre aux besoins de nos clients.
-Fournir des formations sur l’utilisation de notre solution.
-Offrir un soutien continu pour assurer que nos clients sont en mesure de tirer le meilleur parti de notre technologie.</td><td rowspan="3">-Exploitants agricoles
- Photographe pro / vidéo
- Institutions publiques (pompiers, environnement)
- SONATRACH, NAFTAL, SONELGAZ
- ONG de surveillance forestière</td></tr>
<tr><th>Ressources clés :</th><th>Canaux de distribution :</th></tr>
<tr><td>- Équipe R&D IA / électronique
- Atelier d’assemblage
- Plateformes IA & bases de données
-Infrastructure logistique
- Réseau de distribution</td><td>-Plateformes B2B.
-Des publicités.
-Réseaux sociaux.
-Site Web.
-Participation aux foires technologiques et aux évènements autour de l’entreprenariat.
-livraison à domicile.</td></tr>
</table>

Structure des coûts :	Sources et revenus :
-Les coûts de conception et de fabrication des drones. -Les coûts de maintenance et de réparation. -Les coûts de développement de logiciels d’IA. -Commercialisation de notre solution.	-Vente directe de notre solution. -La formation du personnel et le soutien continu. -les publicités.

القيمة المقترحة أو العرض المقدّم

تحديد المشكل الذي يواجهه الزبون **Déterminer le problème rencontré par le client**	
Je souhaite résoudre le problème de la dépendance aux contrôleurs de vol fermés comme le Pixhawk, qui fonctionnent comme des "boîtes noires", sans contrôle sur la gestion interne des données ni possibilité de personnalisation avancée, ce qui expose à des risques de piratage ou d'utilisation non sécurisée des signaux.	**ما هي المشكلة التي تريد حلها؟**
D'après mon étude des systèmes comme le Pixhawk, l'accès complet au code n'est pas garanti, et les protocoles de sécurité sont parfois opaques. De plus, des rapports dans le domaine de l'aéronautique et des systèmes embarqués signalent des incidents causés par des vulnérabilités logicielles ou des interférences radio non sécurisées.	**ما هي البيانات المتوفرة لديك التي تدل على وجود المشكلة المحددة؟**
Il n y'a pas d'autresétudes existantes.	**ما هي المشاريع الأخرى التي استهدفت نفس المشكلة والتي جرى تنفيذها؟**
Objectifs : • Développer un contrôleur de vol 100% personnel et transparent. • Garantir la sécurité des signaux et données par des protocoles maison. • Assurer la compatibilité multi-domaine (drones agricoles, militaires, civils...). • Réduire la dépendance aux solutions étrangères. Résultatsattendus : • Première version fonctionnelle du FC personnalisé. • Validation par test en vol. • Base solide pour un développement industriel ou scientifique.	**ماهي أهداف مشروعك و/أو نتائجه المتوقعة؟**

La fiche technique de projet startup 1275

البطاقة التقنية للمشروع

La fiche technique de projet

Carte d'information	
DORBANI NAHR EL IMENE ZOUAOUI SARA	**الاسم و اللقب** **Your first and last Name** **Votre prénom et nom**
Development of flight controller for drones by MCU	**الاسم التجاري للمشروع** **Intitulé de votre projet** **Title of your Project**
Légal en cas d'autorisation	**الصفة القانونية للمشروع** **Votre statut juridique** **Yourlegalstatus**
0699551387 0672952923	**رقم الهاتف** **Votre numéro de téléphone** **Your phone number**
dorbaninahr2002@gmail.com **Saarah76@hotmail.com**	**البريد الالكتروني** **Votre adresse e-mail** **Your email address**
Blida Oulad-Yaich	**مقر مزاولة النشاط (الولاية- البلدية)** **Votre ville ou commune d'activité** **Your city or municipality of activity**

طبيعة المشروع (طبيعة الابتكار)

La nature de projet

Vente de services.	**المنتوج ذو طابع إنتاجي أو خدماتي** **Vente de marchandises ou de services** **Sale of goods or services**

Proposition de valeur ou l'offre faite

القيمة المقترحة وفق المعايير التالية

La valeur proposée selon les critères suivants

القيمة المقترحة وفق المعايير التالية	
القيمة بالتخصيص	/
القيمة بالسعر	/
القيمة بالتصميم	/
القيمة بالأداء العالي	/
القيمة بالخدمة الشاملة	/
قيم أخرى	/

محتوى مخطط العملي للمشروع

Business Model Canvas – *BMC*

1. الشركاء الأساسيون Key Partners

الشركاء	معلومات حول الشركاء	طبيعة الشراكة
الشريك الأول	/	-
الشريك الثاني	/	-
الشريك الثالث	/	-
الشريك الرابع	/	-
الشريك الخامس	/	-

2. قنوات التوزيع Channels

قنوات التوزيع Channels	
المبيعات المباشرة	/
تجار الجملة	/
الموزعون	/
توزيع التجزئة	/

3. العلاقة مع العملاء Customer Relationship

كيف تدير علاقاتك مع العملاء؟	-Etablissez des canaux de communication solides avec les clients, tels que les publicités, le chat en direct, et nous serons disponibles pour répondre aux demandes des clients et fournir le support nécessaire. -Formation et sensibilisation : Nous proposerons des sessions de formation et de sensibilisation aux clients sur l'utilisation de notre drone, afin de nous assurer que nos clients comprennent comment utiliser notre service. -En fournissant un contenu précieux : en fournissant un contenu précieux aux clients, tels que des articles, des guides et des conseils utiles liés à l'utilisation de notre solution, cela renforcera la confiance et montrera aux clients que nous sommes des experts dans le domaine. -Réponse rapide : en assurant une réponse rapide à toute demande ou plainte que nous recevons, la réponse immédiate contribue à établir une relation solide et améliore la satisfaction de la clientèle. -Les techniques d'analyse graphique peuvent également être utilisées pour comprendre les besoins des clients et anticiper leurs tendances et leurs demandes.
ماهية أهم البرامج التي ستعتمد عليها في ادارة العلاقة مع الزبون	-Les réseaux sociaux. -des conventions. -site Web.

تطور حجم الأعمال في السنوات الثلاث