Hardware Project of ICT

ROCKET GUIDENCE SYSTEM

Submitted by:

1.SARA NASEER(24-CS-37)

2. EMAN ARIF(24-CS-105)

3.UMAR(24-CS-130)

Submitted to:

Dr. Muhammad Munwar Iqbal

Department of Computer Science

University of Engineering and Technology, Taxila

Fall 2023

ACKNOWLEDGEMENT

Firstly, I want to thank from the core of my heart to Almighty Allah who gave me the blessings in life and lifted me in need and give me firmness, zeal and competence for accomplishment and presentation of this research work.

I also want to put forward Durood on the Holy Prophet Hazrat Muhammad (Peace be Upon Him), who guided us to follow the correct path of Islam and instructed us the vital rules for living the balanced and comprehensive life.

I would like to thank my supervisor Assistant Professor. Dr. Muhammad Munwar Iqbal for providing the proper guidance right way to conduct my research work. He gave me the educational and amiable atmosphere to accomplish this research task in his supervision.

I want to present special thanks to my beloved parents. Who encouraged and helped me throughout the life and gave me quality education.

I would also like to say thanks to my dear sisters and brothers for their endless support and their affection. And I would also like to thank my colleagues and dear friends who backed and inspired me during my study period.

ABSTRACT

The main objective of this project is to design, develop, and implement an active fin stabilization system for rockets that ensures real-time course correction and enhanced flight stability. The system utilizes **gyro sensors** to detect angular deviations (pitch, yaw, and roll) and employs **servo-controlled fins** to counteract these deviations, maintaining the rocket's stability throughout its flight. The project aims to develop a sensor-driven stabilization mechanism that continuously monitors the rocket's flight attitude, implement an efficient control algorithm for real-time corrective actions, and design a cost-effective actuator system for precise adjustments. Additionally, the system will be robust under varying flight conditions and aims to achieve accurate flight paths, improving mission success and safety. Through the use of low-cost components, the system will offer a scalable and reliable solution for small- and medium-scale rockets, addressing the challenges of maintaining stability during flight.

TABLE OF CONTENTS

Introduction

The **Active Fin Stabilization System** (AFSS) is designed to tackle the complex challenges of maintaining stability during rocket flights, where even small deviations in orientation can significantly affect the trajectory and performance of the rocket. During flight, factors such as aerodynamic forces, wind gusts, and changes in the rocket's mass distribution can cause unwanted **pitch**, **yaw**, and **roll** deviations, potentially leading to mission failure. This system aims to address these issues by leveraging modern sensors and actuators that provide real-time adjustments to the rocket's control surfaces (fins). By continuously monitoring the rocket's orientation and making rapid, precise corrections, the AFSS ensures that the rocket maintains a stable and accurate flight path. This project explores a low-cost and efficient solution, making the technology accessible while still achieving the high level of control necessary for stable and successful rocket operations. Through this innovative approach, the AFSS minimizes the impact of destabilizing forces, enhancing both the performance and safety of the rocket throughout its flight.

Project Aim

To design and implement an active fin stabilization system for rockets to ensure real-time course correction and enhanced flight stability. The system utilizes gyro sensors to detect angular deviations and servo-controlled fins to counteract them.

Project Objective

The objective of this project is to design, develop, and implement an active fin stabilization system for rockets that ensures real-time course correction and enhanced flight stability. The system utilizes **gyro sensors** to detect angular deviations (pitch, yaw, and roll) and employs **servo-controlled fins** to counteract these deviations, maintaining the rocket's stability throughout its flight. The project aims to develop a sensor-driven stabilization mechanism that continuously monitors the rocket’s flight attitude, implement an efficient control algorithm for real-time corrective actions, and design a cost-effective actuator system for precise adjustments. Additionally, the system will be robust under varying flight conditions and aims to achieve accurate flight paths, improving mission success and safety. Through the use of low-cost components, the system will offer a scalable and reliable solution for small- and medium-scale rockets, addressing the challenges of maintaining stability during flight.

Project scope and limitation

Sensor Drift: Inaccuracies in the gyroscopic sensor readings can lead to deviation in the rocket's trajectory.

Battery Dependency: The system's performance is dependent on the power supply, which could deplete during long flights.

Algorithm Challenges: Developing effective algorithms to adjust the rocket's fins accurately in real-time, especially during dynamic flight conditions.

·**Environmental Influence**: External factors like wind could affect the rocket's stability, requiring extensive testing under different conditions to ensure reliability.

Description of the Project

The **Rocket Guidance System: Active Fin Stabilization** project aims to enhance the stability and control of rocket flights by implementing an active fin stabilization system. This system utilizes an **MPU 6050 gyro sensor** to detect angular deviations in the rocket's orientation, such as **pitch**, **yaw**, and **roll**, and employs **servo-controlled fins** to correct these deviations in real time. The system is powered by an **Arduino Uno microcontroller**, which processes sensor data and sends corresponding PWM control signals to the servo motors attached to the fins. The project demonstrates how modern sensors, actuators, and a simple control algorithm can be integrated to improve rocket flight stability, using lightweight materials like PVC and plywood for the rocket's structure. With future enhancements, such as advanced guidance algorithms and full-scale flight testing, this project aims to provide a low-cost, efficient solution for ensuring stable rocket trajectories during flight.

Hardware Requirement

MPU 6050 Gyro Sensor:
Measures angular velocity and linear acceleration.
Provides real-time data for flight dynamics.
Servo Motors (4x):
Actuate the fins to correct angular deviations.
Controlled via PWM signals from a microcontroller.
Microcontroller (Arduino Uno):
Processes sensor data and sends control signals to the servos.
Power Supply:
USB From PC to power the servos and microcontroller.
Rocket Model:
A scaled-down prototype equipped with fins for testing.

Software Requirement

- The microcontroller is programmed in Arduino IDE.
- Key functionalities:
    - Read real-time data from the MPU 6050.
    - Calculate necessary corrections for pitch, yaw, and roll.
    - Send PWM signals to control the servo motors.

Description of Hardware Required

Arduino uno

The Arduino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to- serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to- serial converter.

Some Technical Specification of Arduino Uno are:

1.	Microcontroller	ATmega328P
2.	Operating Voltage	5V
3.	Input Voltage (recommended)	7-12V
4.	Input Voltage (limits)	6-20V
5.	Digital I/O Pins	14
6.	Analog Input Pins	6
7.	DC Current per I/O Pin	40 mA
8.	DC Current for 3.3V Pin	50 mA
9.	Flash Memory	32 KB of which 0.5 KB used by bootloader
10.	SRAM	2 KB
11.	EEPROM	1 KB
12.	Clock Speed	16 MHz

Circuit Diagram

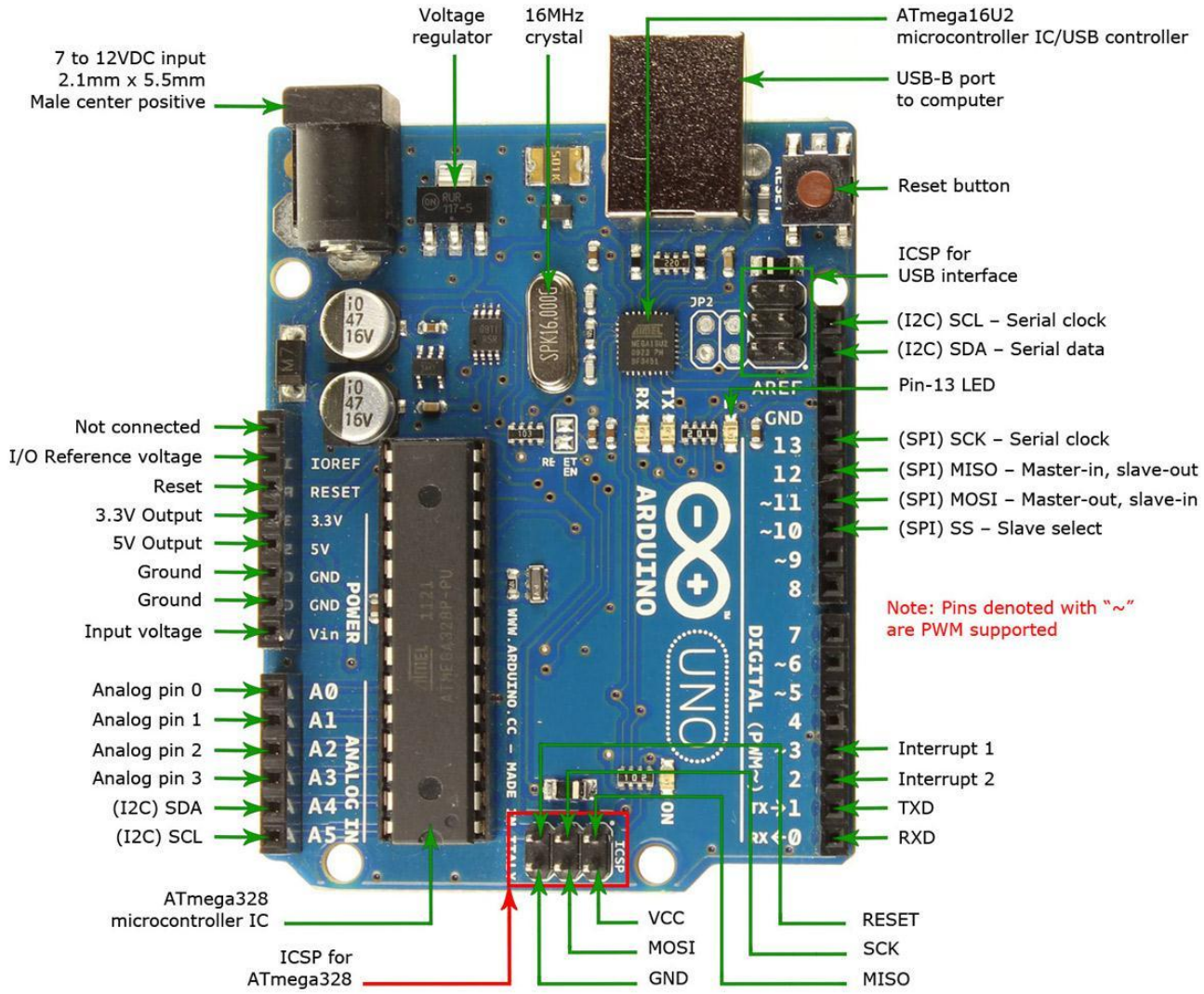


Figure 01. Circuit diagram for component of Arduino UNO

MPU 6050 GYRO SENSOR:

The **MPU 6050** is a compact and versatile sensor that combines both an **accelerometer** and a **gyroscope** into a single module, making it ideal for motion tracking and stabilization applications. It measures linear acceleration along three axes (X, Y, Z) through its accelerometer and detects rotational movement or angular velocity along the same axes using its gyroscope. The sensor communicates with microcontrollers via the **I2C interface**, allowing for easy integration into various systems. Additionally, the **onboard Digital Motion Processor (DMP)** processes motion data internally, offloading computational tasks from the microcontroller for improved efficiency. In the context of the **Active Fin Stabilization System**, the MPU 6050 provides real-time data on the rocket's orientation and movement, enabling the system to detect pitch, yaw, and roll deviations and make the necessary adjustments to stabilize the flight path.

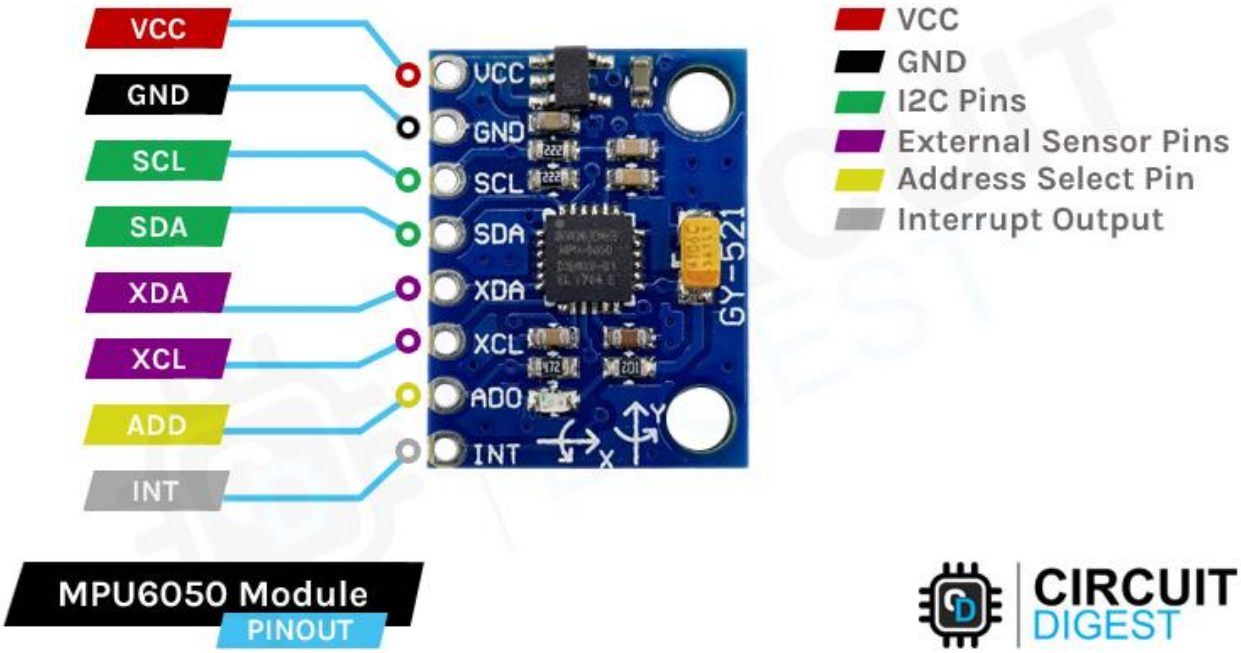


Figure 02. MPU 6050 Gyro Sensor

-

MPU 6050 Gyro SensorInterfacing with Arduino UNO

Interfacing the **MPU 6050** gyroscope and accelerometer sensor with an **Arduino** involves connecting the sensor to the Arduino using the **I2C** communication protocol. The connections include VCC to 5V, GND to ground, SCL to the Arduino's A5 pin (clock), and SDA to A4 pin (data). After wiring, the **Wire** library for I2C and the **MPU6050** library are used in the Arduino IDE to simplify communication with the sensor. The sensor's data can be read using the mpu.getMotion6() function, which retrieves both accelerometer (X, Y, Z axes) and gyroscope (X, Y, Z axes) values. These values are then sent to the **Serial Monitor** for real-time debugging. The MPU 6050 provides important motion data, which can be used in applications like real-time flight stabilization, motion tracking, or robotics.

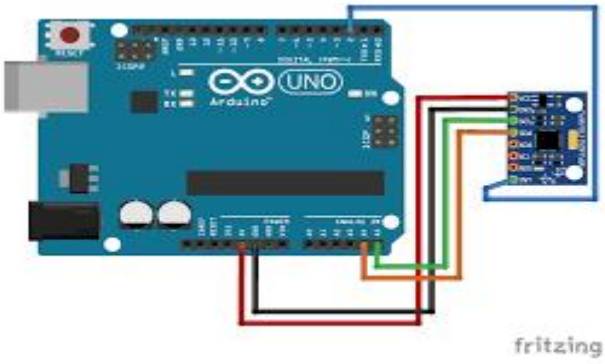
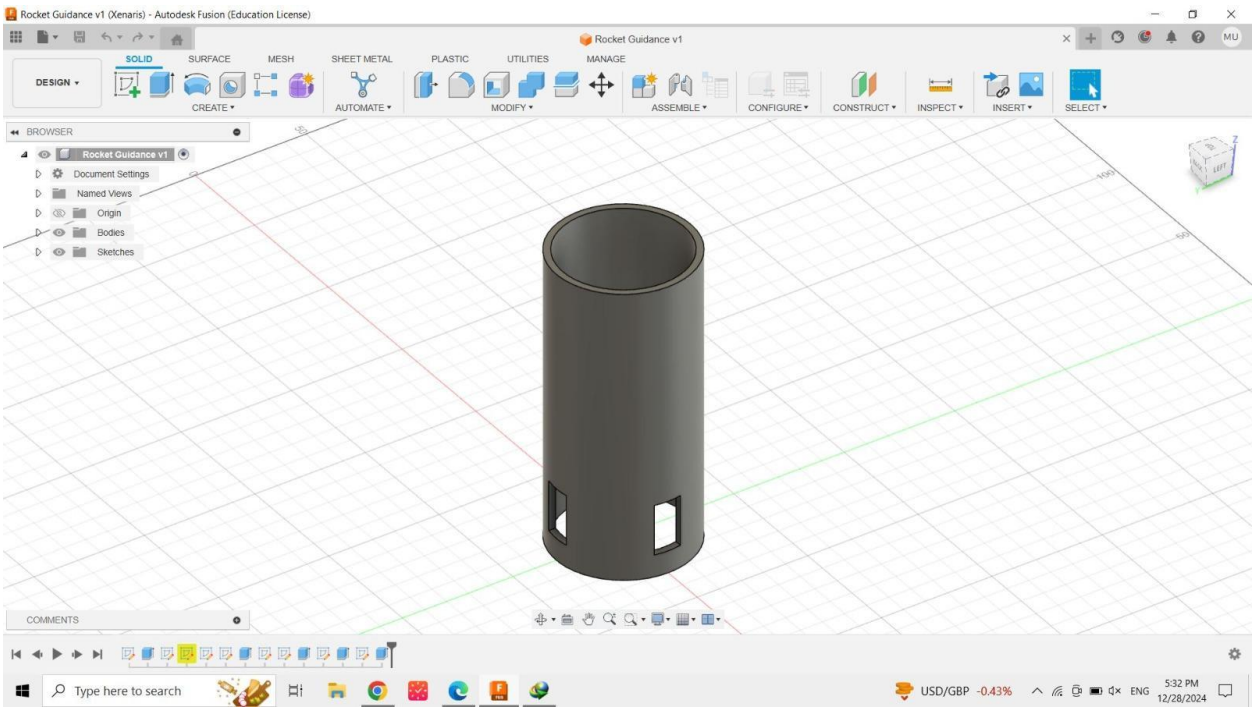


Figure 03. MPU 6050 Gyro Sensor Interfacing with Arduino UNO

CAD Design:

A detailed CAD model of the Guidance Unit was created using Fusion 360.



Design and Implementation

The **Active Fin Stabilization System** for rockets uses an **MPU 6050 gyro sensor** to detect angular deviations in flight, which are corrected in real-time by **servo-controlled fins**. The system is powered by an **Arduino Uno** microcontroller, which processes the sensor data and adjusts the servo positions via PWM signals. The rocket prototype is made with lightweight **PVC pipes** and **plywood fins** to minimize weight and ensure structural integrity. The system provides **real-time stabilization** during flight, correcting pitch, yaw, and roll deviations. Future improvements include scaling for larger rockets and optimizing the system for higher performance.

Designing the Circuit

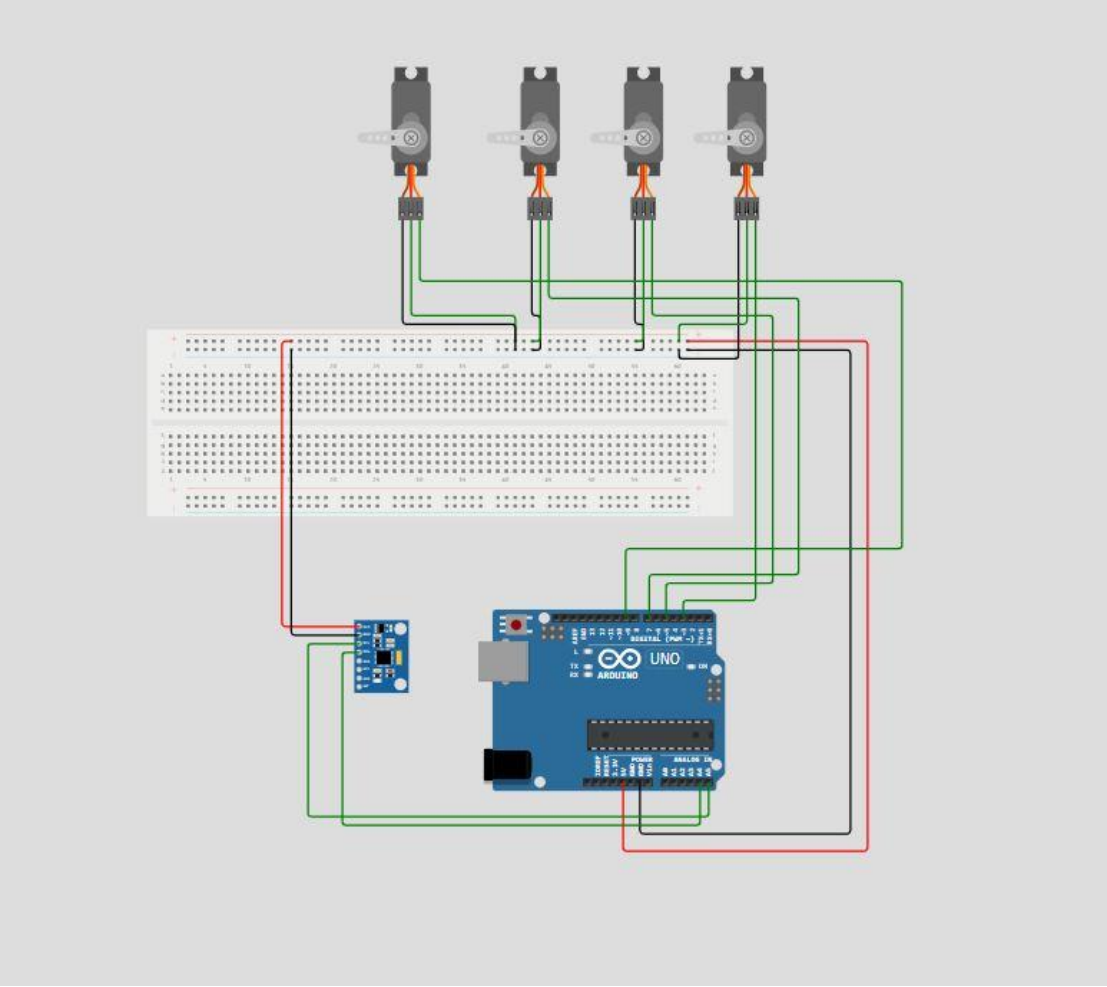


Figure 04. Designing the Circuit

Technical Specification for this project

· **Sensor: MPU 6050**

- **Type**: 3-axis accelerometer, 3-axis gyroscope
- **Communication**: I2C interface
- **Range**: ±2g, ±4g, ±8g, ±16g for accelerometer; ±250, ±500, ±1000, ±2000 degrees/second for gyroscope
- **Power Supply**: 3.3V to 5V

· **Microcontroller: Arduino Uno**

- **Processor**: ATmega328P microcontroller
- **Operating Voltage**: 5V
- **Clock Speed**: 16 MHz
- **I/O Pins**: 14 digital pins, 6 analog pins
- **Communication**: I2C (for MPU 6050), PWM (for servo motors)

· **Servo Motors (4x)**

- **Type**: Standard Servo Motors
- **Torque**: 9-12 kg/cm (depends on the servo used)
- **Control Method**: PWM signals from the Arduino
- **Range**: 0 to 180 degrees

· **Power Supply**

- **Source**: USB from PC or external battery (5V for Arduino and servos)
- **Current Rating**: Adequate to power the servos and Arduino simultaneously

· **Rocket Model**

- **Material**: PVC pipes for the body (lightweight, durable), plywood for the fins
- **Size**: Scaled-down prototype suitable for testing the stabilization system

· **Software**

- **Development Environment**: Arduino IDE
- **Libraries Used**:Wire.h for I2C communicationMPU6050.h for interacting with the MPU 6050 sensorServo.h for controlling the servo motors
- **Control Algorithm**: Real-time calculation of pitch and roll based on sensor data, mapped to servo positions.

· **Stabilization System**

- **Real-time Feedback**: Constant data acquisition from the MPU 6050 sensor to detect angular deviations.
- **Course Correction**: Adjustments made via PWM signals to the servo motors to reposition the fins and correct any flight instability (pitch, yaw, or roll).

Program Code

```
#include <Wire.h>
#include <Servo.h>
#include <MPU6050.h>

// Create MPU6050 object
MPU6050 mpu;

// Create servo objects
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;

// Pin definitions
const int servo1Pin = 3;
const int servo2Pin = 5;
const int servo3Pin = 6;
const int servo4Pin = 9;

void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Initialize MPU6050
  Wire.begin();
  mpu.initialize();

  if (!mpu.testConnection()) {
    Serial.println("MPU6050 connection failed!");
    while (1);
  }

  Serial.println("MPU6050 connected successfully.");

  // Attach servos to their pins
  servo1.attach(servo1Pin);
  servo2.attach(servo2Pin);
  servo3.attach(servo3Pin);
  servo4.attach(servo4Pin);

  // Center servos initially
  servo1.write(90);
  servo2.write(90);
  servo3.write(90);
  servo4.write(90);
}
```

```
void loop() {
  // Read raw accelerometer and gyroscope data
  int16_t ax, ay, az, gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  // Calculate pitch and roll from accelerometer data
  float pitch = atan2(ax, sqrt(ay * ay + az * az)) * 180 / PI;
  float roll = atan2(ay, sqrt(ax * ax + az * az)) * 180 / PI;

  // Print pitch and roll for debugging
  Serial.print("Pitch: ");
  Serial.print(pitch);
  Serial.print("\tRoll: ");
  Serial.println(roll);

  // Map pitch and roll to servo positions (0 to 180 degrees)
  int servo1Pos = constrain(map(pitch, -45, 45, 45, 135), 0, 180);
  int servo2Pos = constrain(map(roll, -45, 45, 45, 135), 0, 180);
  int servo3Pos = constrain(map(pitch, -45, 45, 135, 45), 0, 180);
  int servo4Pos = constrain(map(roll, -45, 45, 135, 45), 0, 180);

  // Write positions to servos
  servo1.write(servo1Pos);
  servo2.write(servo2Pos);
  servo3.write(servo3Pos);
  servo4.write(servo4Pos);

  // Delay for stability
  delay(50);
}
```

Pros of Rocket Guidence System:

Improved Flight Stability:The system provides real-time course correction, which helps in maintaining stability during rocket flight, preventing deviations in pitch, yaw, and roll.

Cost-Effective:By using low-cost components like the **MPU 6050** gyro sensor and **Arduino Uno**, this system offers a practical solution for both educational and small-scale applications without requiring expensive hardware.

Real-Time Feedback and Control:The active fin stabilization system can instantly react to changes in orientation and make necessary adjustments, which ensures that the rocket maintains a steady flight path throughout its ascent.

Scalability:The system is versatile and can be scaled for different types of rockets, from small prototype models to medium-sized experimental rockets, making it adaptable for various applications.

Cons of Rocket guidence system:

Limited to Small-Scale Rockets:Due to the lightweight materials used for construction and the relatively small scale of the components, the system might not be suitable for larger, high-speed rockets without significant modifications.

Servo Motor Limitations: The performance of servo motors, such as speed and precision, may not be sufficient for higher-speed rockets. High-performance servos may be required for better control at higher speeds or more demanding conditions.

Applications

- Using this project, we can turn on or off appliances remotely i.e. using a phone or tablet.
- The project can be further expanded to a smart home automation system by including some sensors like light sensors, temperature sensors, safety sensors etc. and automatically adjust different parameters like room lighting, air conditioning (room temperature), door locks etc. and transmit the information to our phone.
- Additionally, we can connect to internet and control the home from remote location over internet and also monitor the safety.

Future Development of the project

- Integrating advanced trajectory control algorithms for precise path optimization.
- Conducting flight tests using a full-scale model for real-world validation.
- Optimizing system components for weight and power efficiency to increase performance and reduce energy consumption.

Conclusion

This project demonstrates the feasibility of an active fin stabilization system for rockets. By integrating hardware and software components, significant progress has been made toward achieving reliable in-flight stability.

Reference

The web sites that provide the informations:

[1]. https://wokwi.com/projects/418714140982163457

[2]. https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/model-rockets/

[3]. https://www.researchgate.net/publication/220049724_Model_Rocket_Workshop_A_Project-Based_Learning_Experience_for_Engineering_Students

[4]. https://www.researchgate.net/publication/366278128_Arduino_Rocket_Flight_Computer

[5]. https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/bgmr.html