Self-Stabilizing IoT Platform

Industry 4.0 Enabling Technologies

31 May 2025

Group Members

Lemuel

Ebrahim

Momen

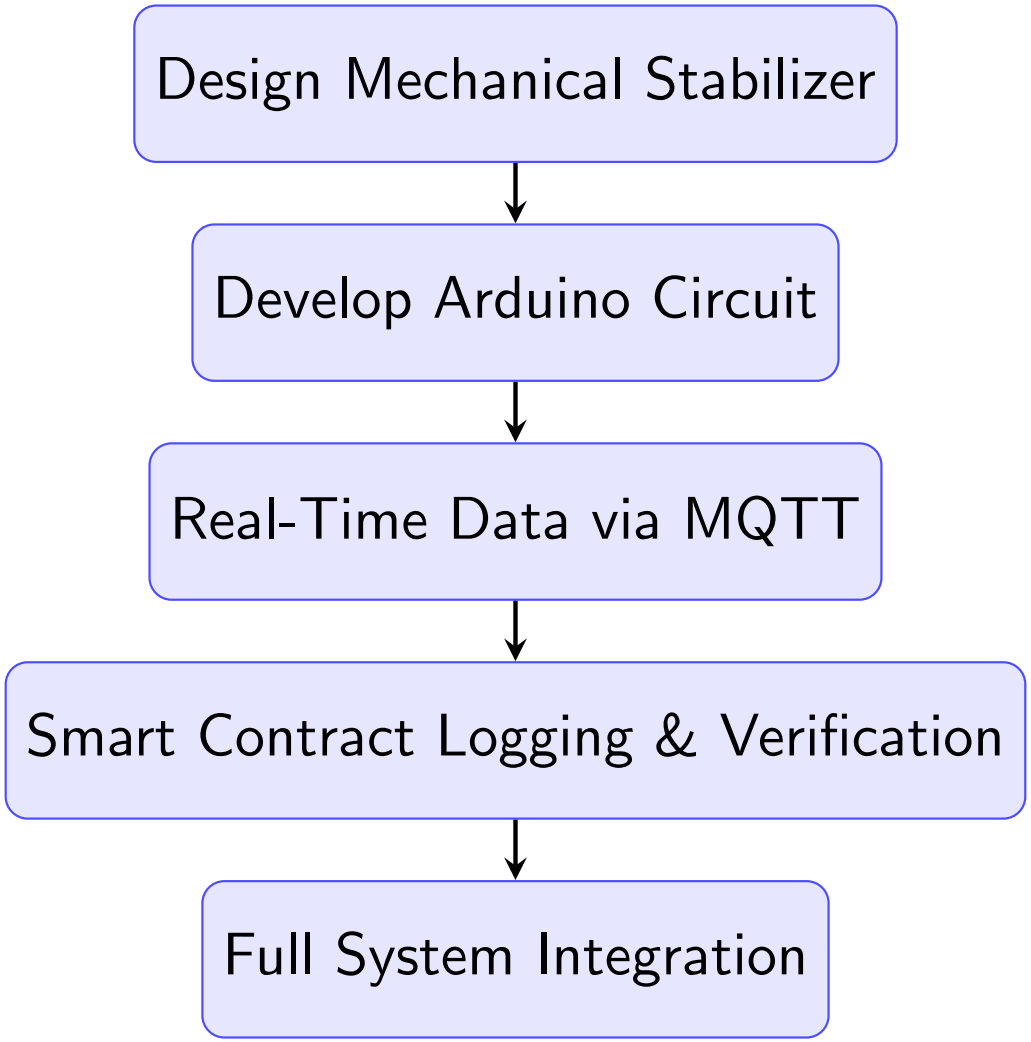
Problem Context

- Ships in **rough seas** experience constant **rolling and pitching** that can destabilize equipment and reduce operational safety.
- Traditional stabilization systems are often **bulky**, **expensive**, and **mechanically complex**.

Project Objective

- Develop a **lightweight**, **low-cost**, and **responsive stabilization platform**.
- Enable **real-time motion correction** using servo-driven mechanisms.
- Provide remote monitoring and alerting capabilities via IoT.
- Smart contract for mission control and logging

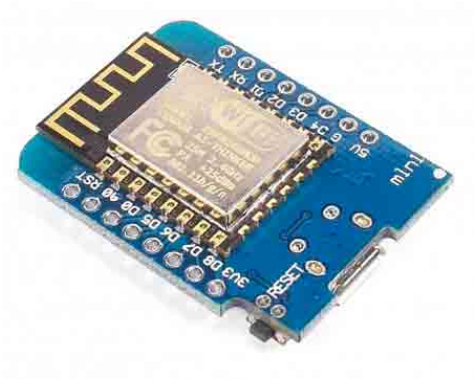
Project Workflow



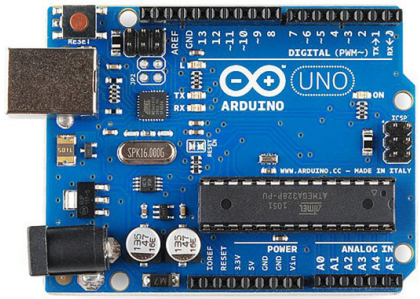
Tools and Components Used

- **CAD Software**: For 3D design of the stabilizer structure.
- **Fritzing**: For circuit layout and schematic design.
- **Arduino IDE**: For programming the microcontroller.
- **Arduino Uno** and **ESP8266 (Wemos D1 R1)**: Core IoT devices for connectivity.
- **GY-521 (MPU6050)**: Gyroscope/accelerometer module for motion detection.
- **Ultrasonic Sensor**: For measuring distance
- **MQTT Protocol**: Lightweight communication protocol for IoT data.
- **HiveMQ Broker**: Public MQTT broker (`broker.hivemq.com`) used for data exchange.
- **Remix IDE** and **Solidity**: For writing and deploying the smart contract on Ethereum.

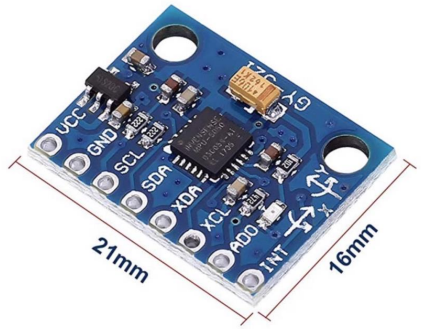
Tools and Components Used



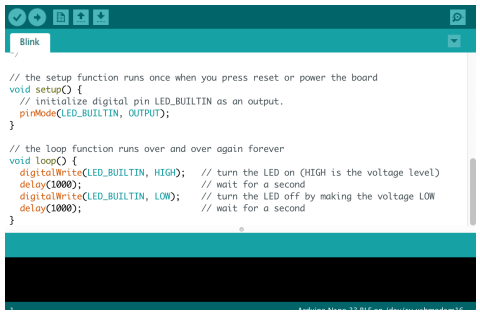
Arduino vemos(esp8266)



Arduino Uno

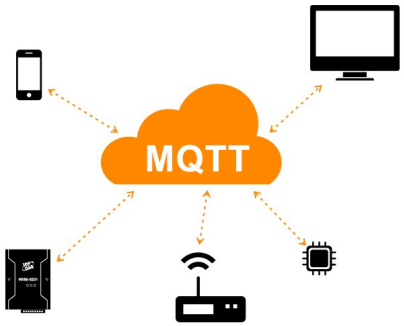


Gyroscope



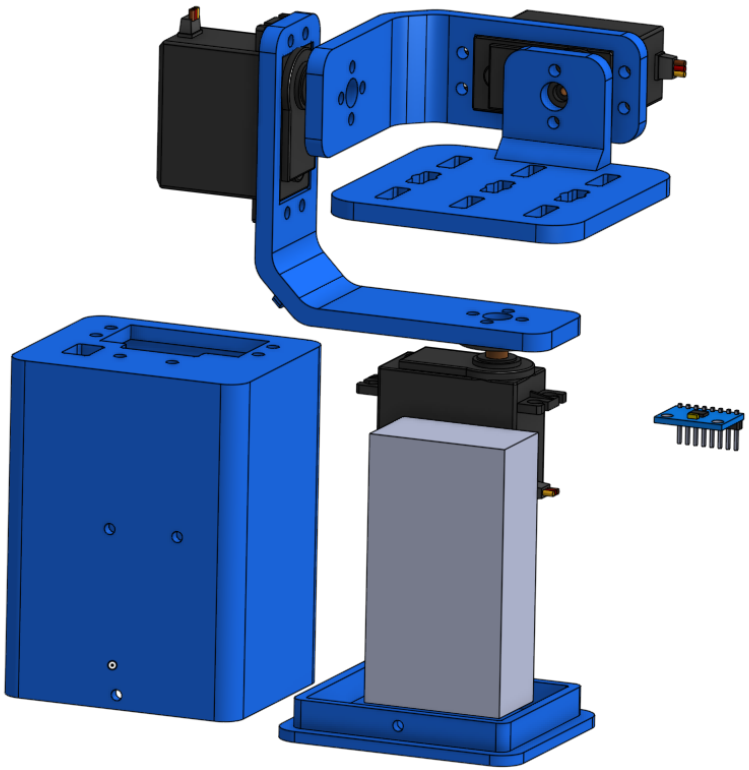
Arduino IDE environment

Ultrasonic sensor

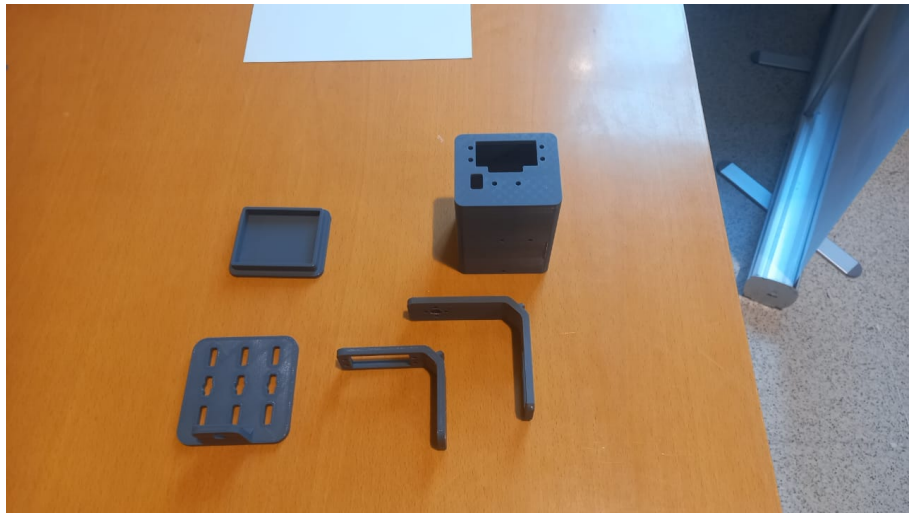


MQTT Protocol

Stabilizer design and structure



Platform in design step



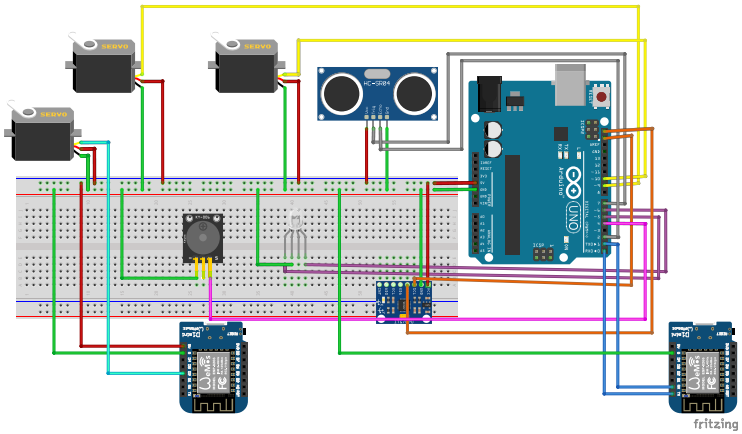
Real Platform after printing
Thanks to Prof. Sara

System Description

The platform consists of a mechanical base with **two servo motors** controlling pitch and roll axes. An **MPU6050 sensor** is mounted on the moving platform to detect angular movement. An **ultrasonic sensor** is placed on the front edge to detect nearby obstacles.

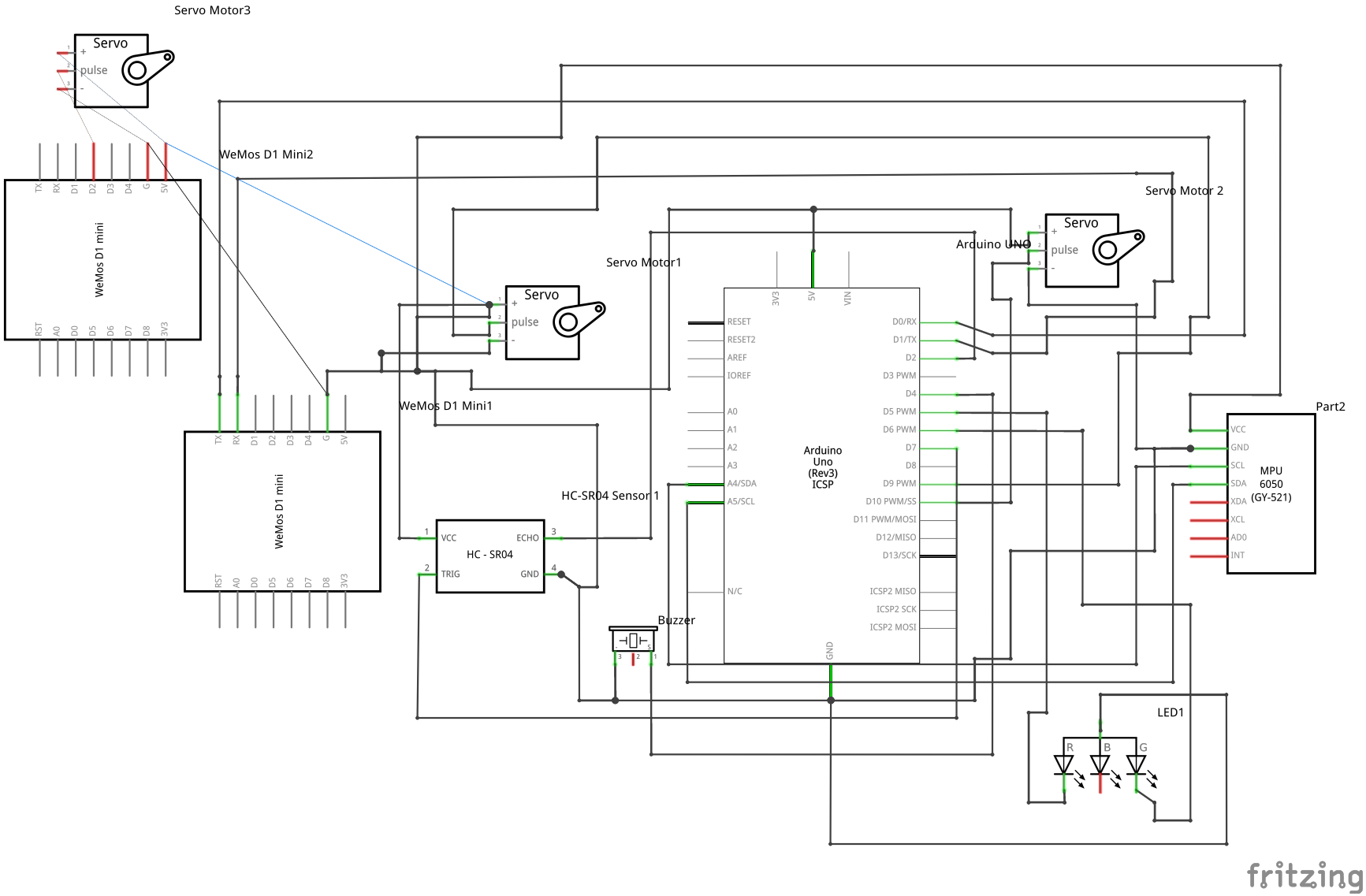
All sensors and actuators are connected to an **Arduino Uno**, which handles motion sensing and servo actuation. The **ESP8266** is wired via **serial UART** to the Arduino and is powered separately to manage Wi-Fi and MQTT communication.

A **buzzer** is connected to the Arduino to trigger alerts when an object is detected within **100 cm**.

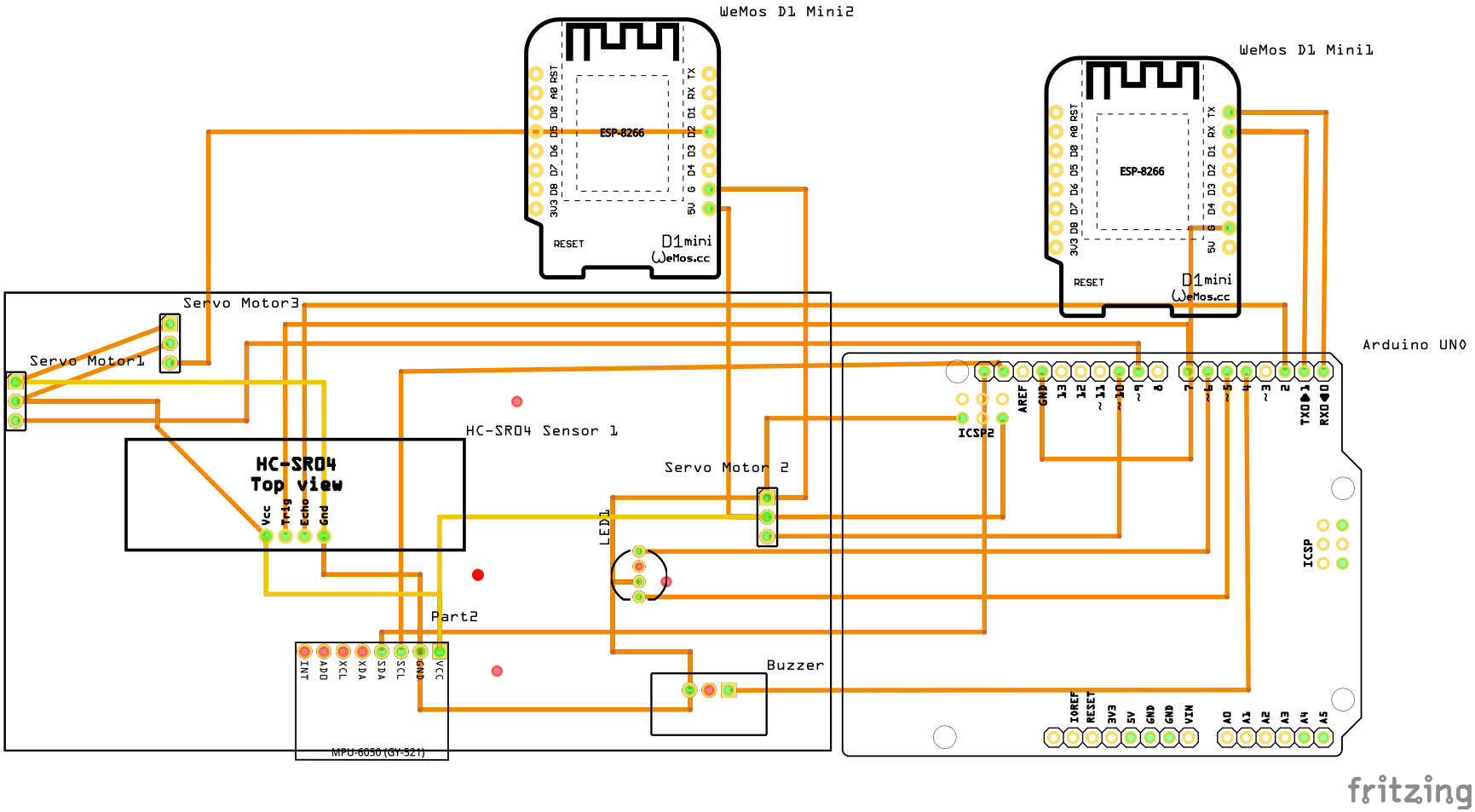


Hardware setup overview

Schematic layout



PCB layout


WeMos D1 Mini2
WeMos D1 Mini1
ESP-8266
RESET
D1mini
WeMos.cc
Servo Motor3
Servo Motor1
HC-SR04 Sensor 1
HC-SR04
Top view
Vcc
Trig
Echo
Gnd
Servo Motor 2
LED1
Part2
Buzzer
MPU-6050 (GY-521)
Arduino UNO
ICSP2
ICSP
fritzing

Working principle(1/2)

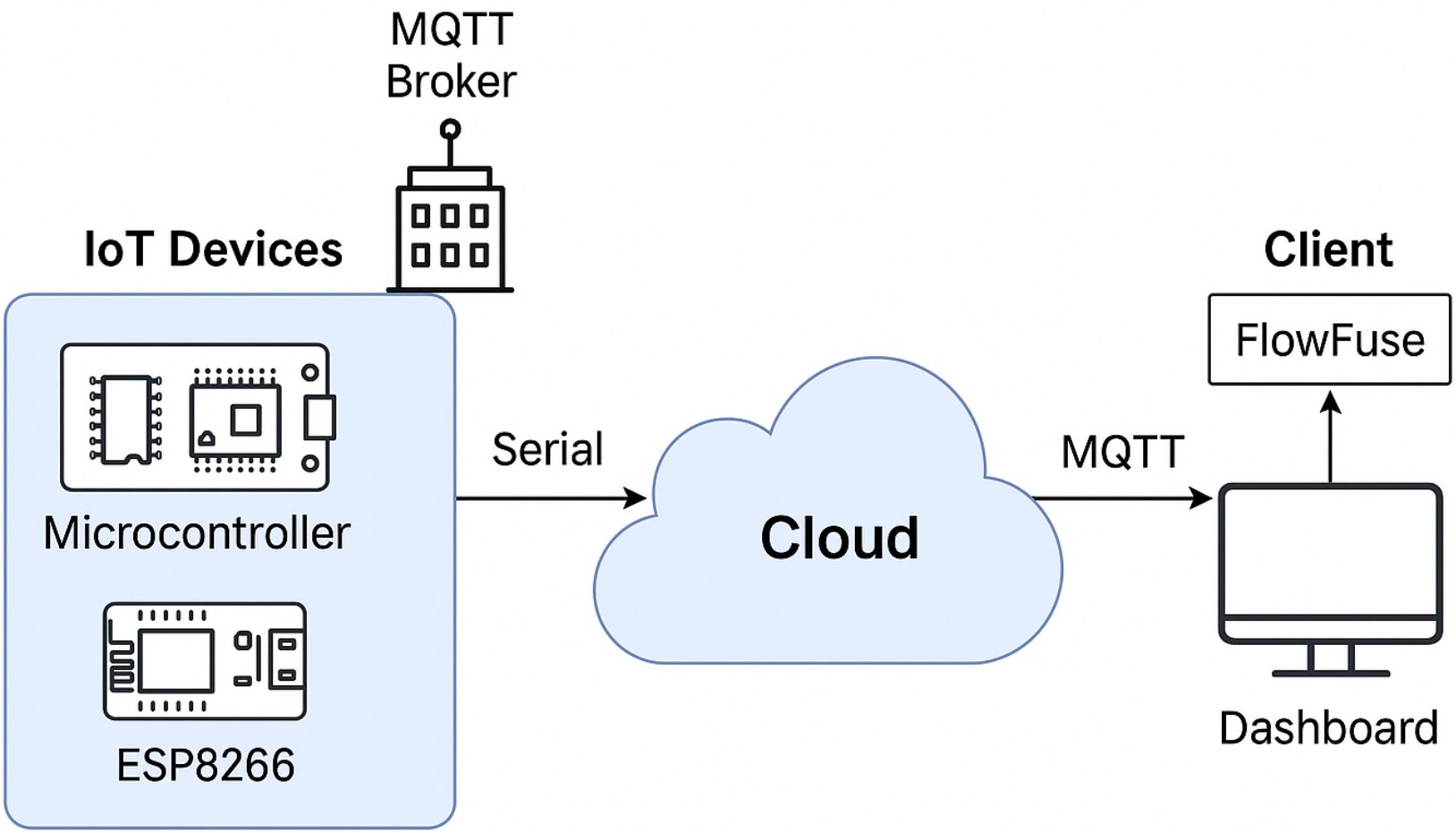
The system operates by combining real-time sensing, data processing, and cloud communication to maintain platform stability:

- **Motion Detection:**
  - **MPU6050** detects **pitch and roll** using gyroscope and accelerometer data.
  - Arduino Uno processes angle values and reads distance from the ultrasonic sensor.
- **Local Response:**
  - If an object is detected within **100 cm**, a **buzzer** is triggered for alert.
- **Data Transmission:**
  - Sensor values (pitch, roll, distance) are sent via **UART** to **ESP8266**.
  - ESP8266 formats data as a **comma-separated string** and publishes to MQTT topic via HiveMQ.

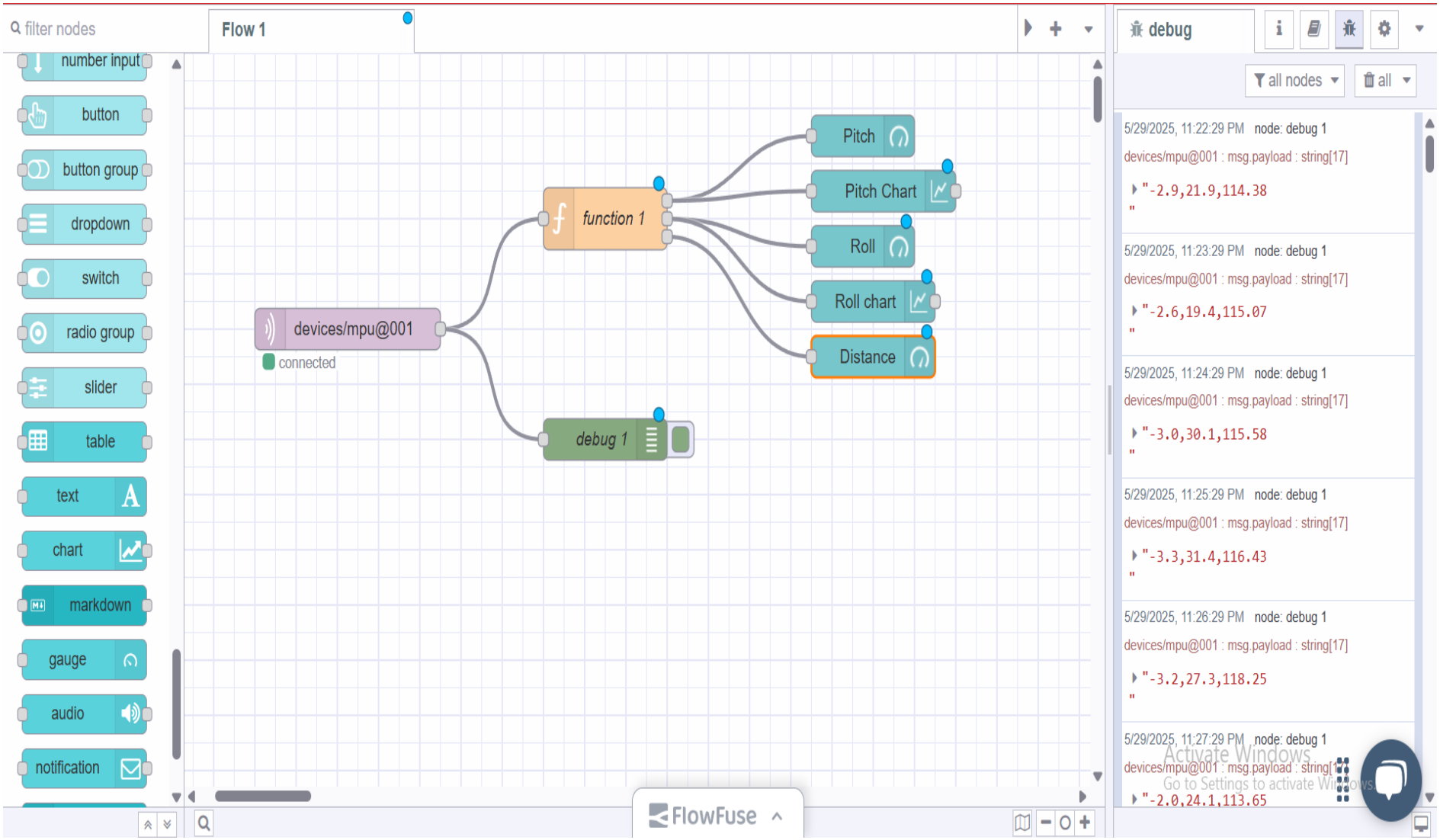
Working principle(2/2)

- **Cloud Processing:**
  - **FlowFuse** (Node-RED) subscribes to the MQTT topic.
  - A function node parses values and updates **dashboard** widgets (gauges, charts).
- **Platform Stabilization:**
  - Servo motors adjust platform orientation based on pitch and roll data.
- **Result:**
  - **Real-time monitoring**, automated response, and alerting—suitable for marine environments.

Cloud architecture

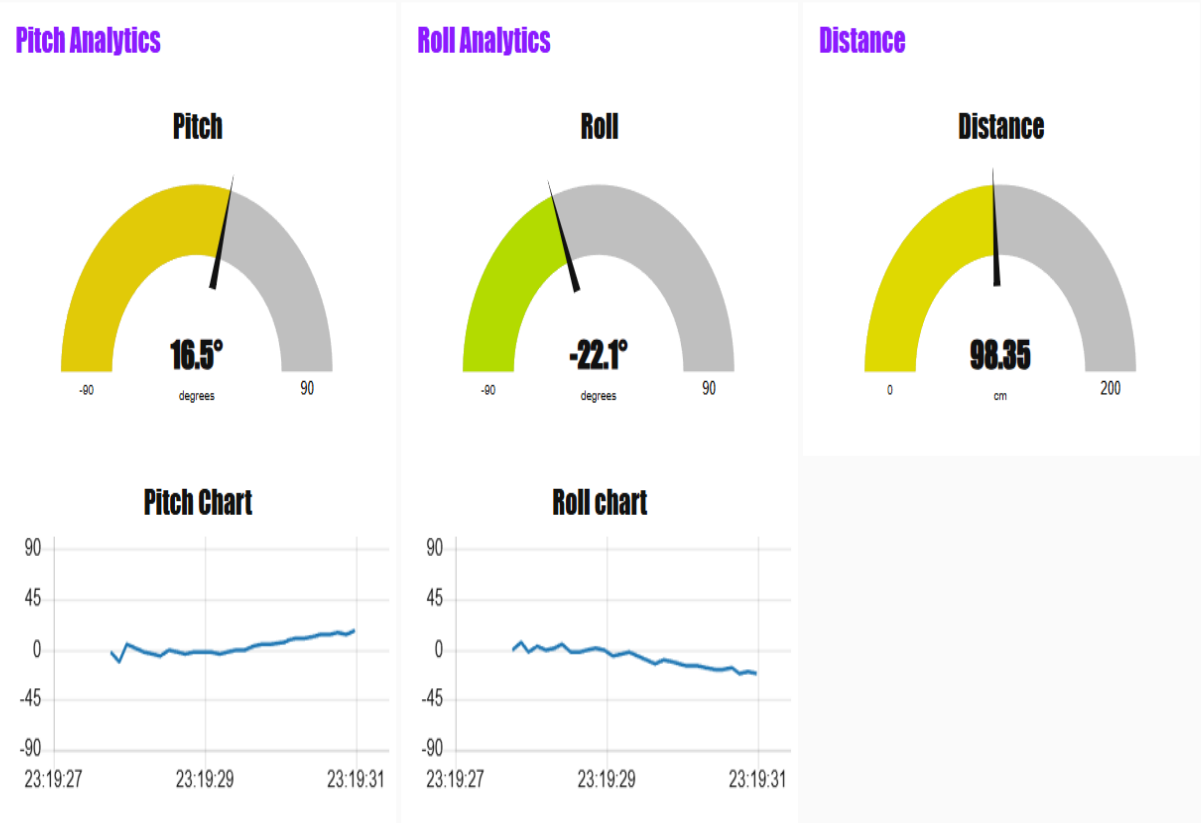


Device-to-Cloud Communication Flow



Dashboard

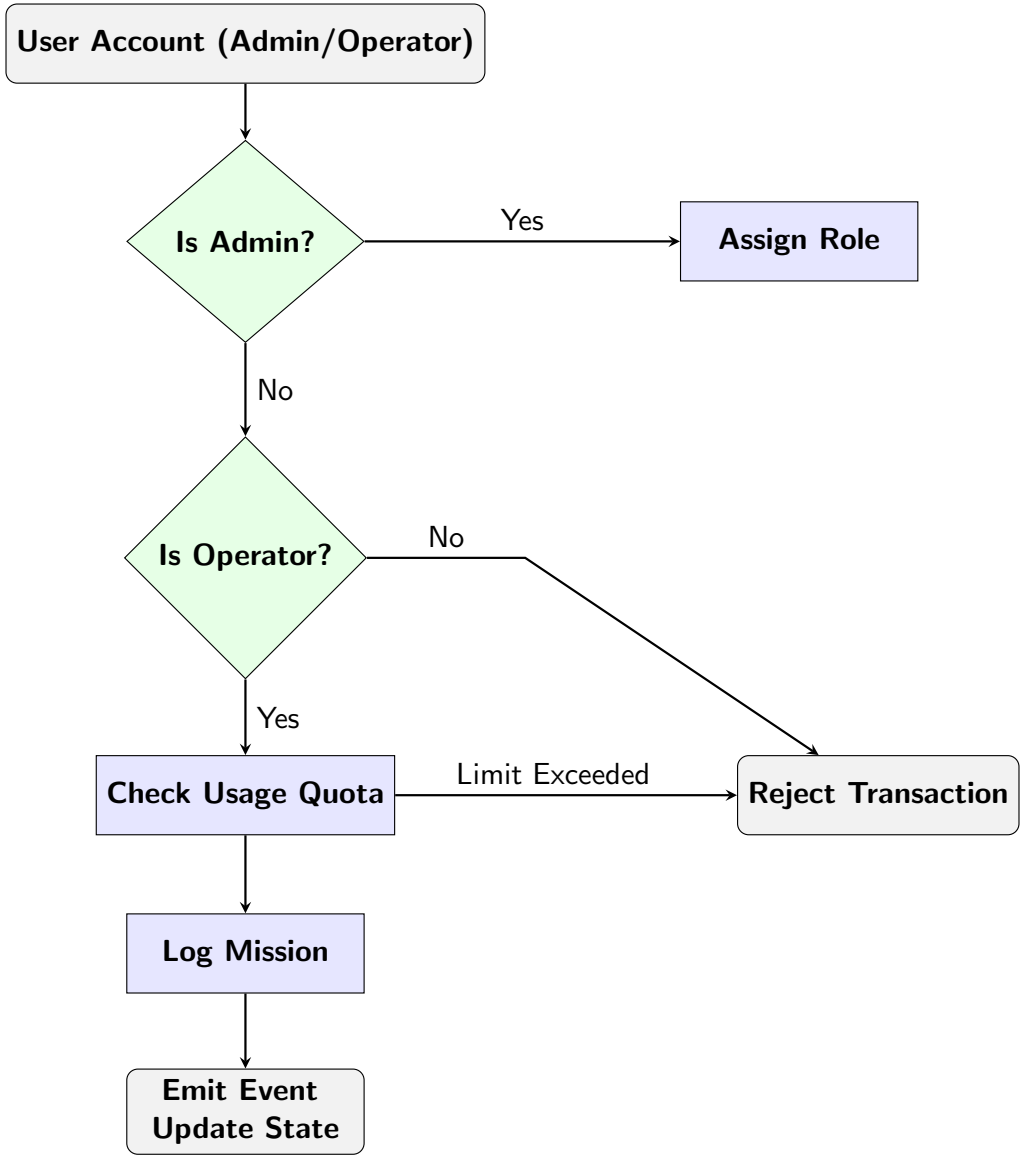
Tab 2



Smart contract

The project incorporates a Solidity-based smart contract to securely log and manage platform usage. Deployed via Remix IDE, the contract defines two roles: Admin and Operator. The Admin can assign operators and set daily usage limits, while the Operator logs missions with metadata such as timestamp, duration, and purpose.
Each mission entry is recorded immutably on the blockchain and emits an event for transparency. The contract enforces daily usage restrictions to prevent overuse or unauthorized access, making it suitable for regulated environments such as maritime or industrial monitoring systems. This ensures that the stabilization platform's deployment and usage are both secure and auditable.

Smart contract



Conclusion

This project successfully demonstrates the development of a self-stabilizing IoT platform capable of maintaining balance in dynamic environments, such as on-board ships. By integrating motion and distance sensing, servo-based actuation, and real-time cloud communication via MQTT, the system achieves both autonomous stabilization and remote monitoring through a FlowFuse (Node-RED) dashboard.

The addition of a smart contract further enhances the solution by introducing secure, transparent mission logging and operational control. The overall design is modular, low-cost, and scalable, making it suitable for a wide range of marine and industrial applications.

Applications

- **Precision Instrument Support**: Ensures stable operation of sensitive equipment like gyroscopes, cameras, radar, or satellite dishes.
- **Cargo Stability Monitoring**: Helps detect and correct shifting of cargo or containers to prevent imbalance and damage.
- **Landing Platforms for Drones/Helicopters**: Provides a level surface during rough sea conditions for safe UAV or crew transfer operations.
- **Weapon and Surveillance Systems**: Maintains accurate targeting and surveillance by compensating for ship motion.
- **Medical and Laboratory Stations**: Stabilizes medical equipment and experimental setups for reliable operation in marine research vessels.
- **Crew Comfort and Navigation Consoles**: Stabilizes seating, desks, or control panels to reduce fatigue and improve crew performance.
- **Communications Equipment Alignment**: Keeps antennas and communication systems aligned with satellites or shore stations.

Lessens learned

- Sensor Integration and Calibration
- Data Filtering is Essential
- PWM and Pin Limitations
- MQTT Communication
- Smart Contract

Lemuel

Ebrahim

Momen

Future Work and Enhancements

- **Add Yaw Control** – Expand stabilization to all 3 axes (pitch, roll, yaw).
- **Implement Feedback Loop** – Use PID or Kalman filters for auto-correction.
- **Optimize Power** – Add battery/sleep modes and solar charging for autonomy.
- **Develop Mobile App** – Enable remote access, configuration, and alerts.
- **Cloud Data Storage** – Log sensor data for analysis and maintenance.
- **Predictive AI Models** – Use machine learning to anticipate motion.
- **Modular Hardware Design** – Create plug-and-play sensor and actuator modules.
- **Enhanced Smart Contracts** – Add roles, payment access, and advanced logging.

Thank You!

Questions are welcome.

For more information or source code:

`Ask group leader!!!`