Kombucha Fermentation Control System

ECE 445: Senior Design Laboratory

Team #27:

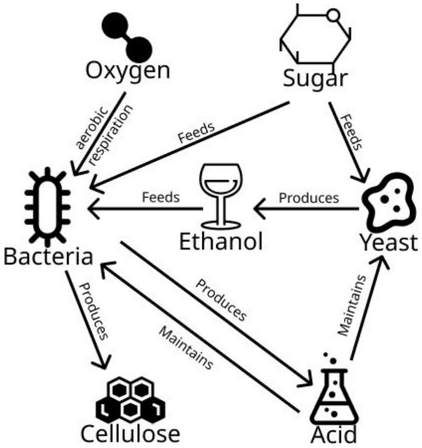
Rudy Beauchesne

John Puthiaparambil

Edwin Xiao

Introduction

- Kombucha is a fermented tea made by combining sweetened tea with a **SCOBY**
- Fermentation depends on maintaining a stable microbial environment over **several days**
- Many people brew kombucha at home, but most home setups rely on **manual observation**
- Our project develops a **low-cost automated fermentation platform** that monitors and helps regulate the brewing process.



Objective

Problem

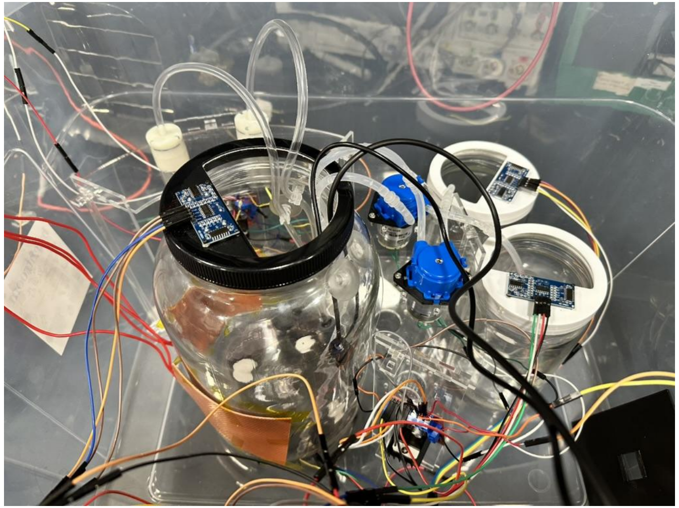
Home kombucha brewing often relies on minimal instrumentation, manual observation, and rough timing This leads to inconsistent fermentation, variable taste, excessive acidity, and uneven pellicle growth.

Requirements: Acidic, Warm, Clean, Dim, Oxygen, Nutrients

Solution

Design a **practical, low-cost closed-loop system** that:

- monitors key variables
- gives users real-time feedback
- improves repeatability over batches
- reduces the need for constant supervision



System Overview / Proposed Solution

The system continuously monitors fermentation using:

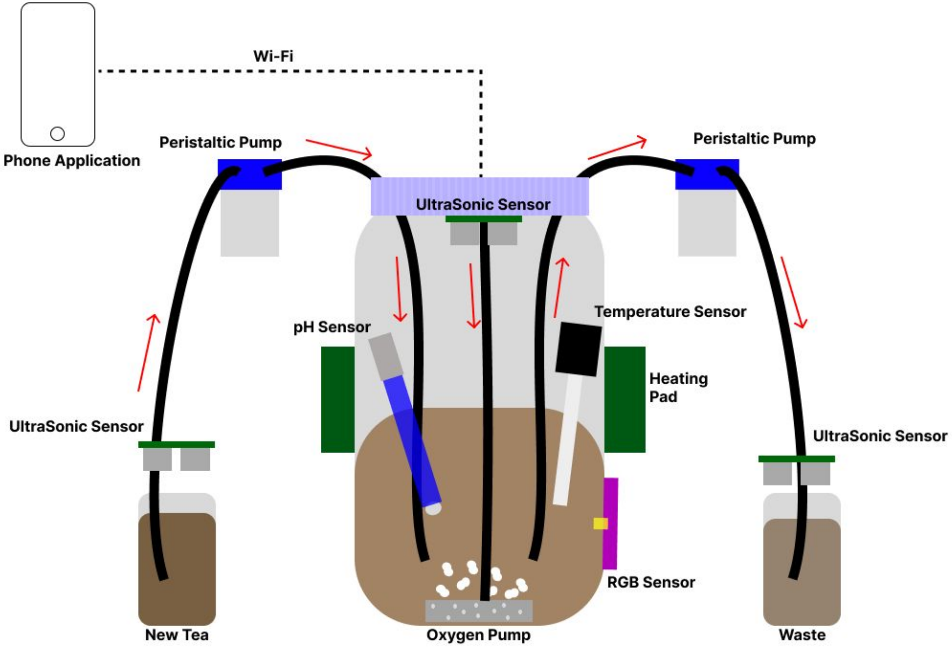
- temperature sensor
- pH sensor
- RGB color sensor
- ultrasonic level sensors

It responds using:

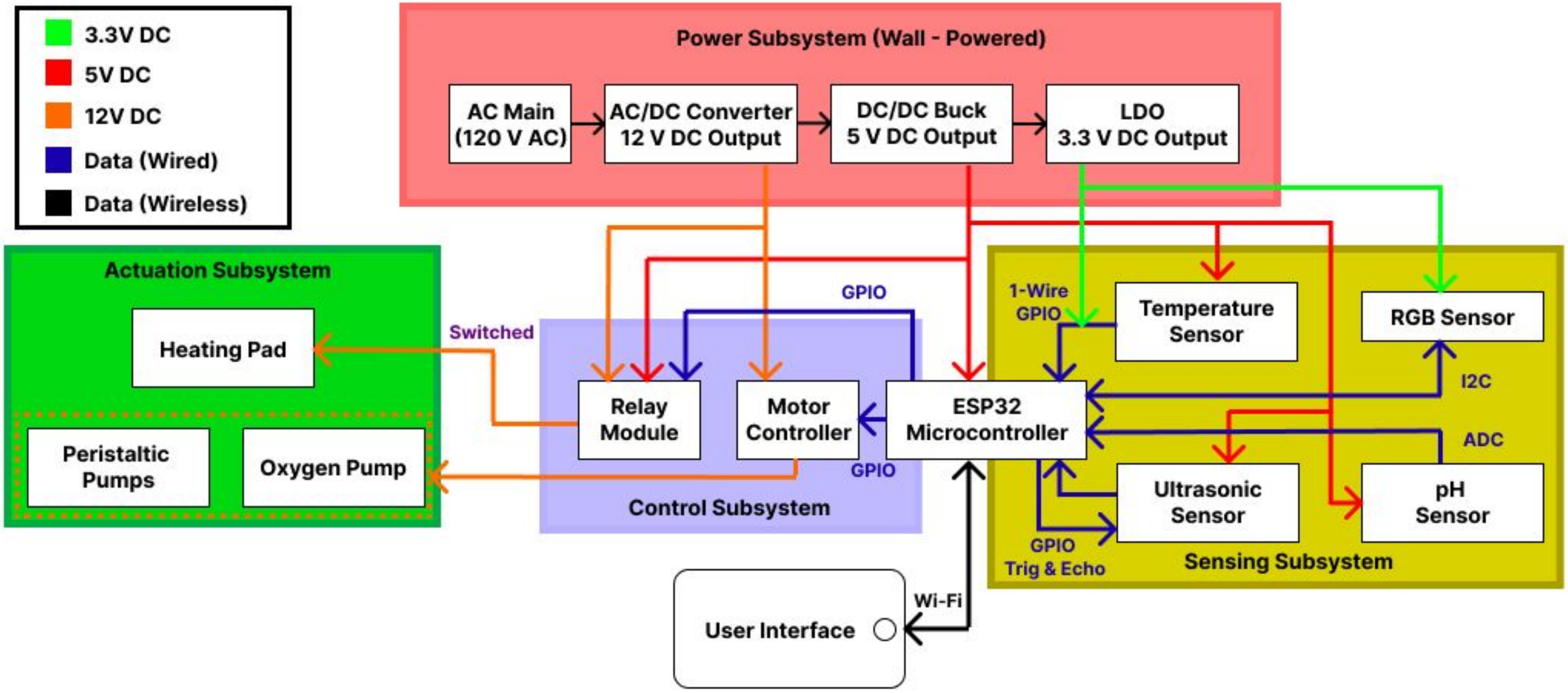
- heating pad
- peristaltic pumps
- aeration pump

And provides:

- Wi-Fi dashboard for real-time monitoring, logging, alerts



Block Diagram



Design Changes

12 V DC power architecture

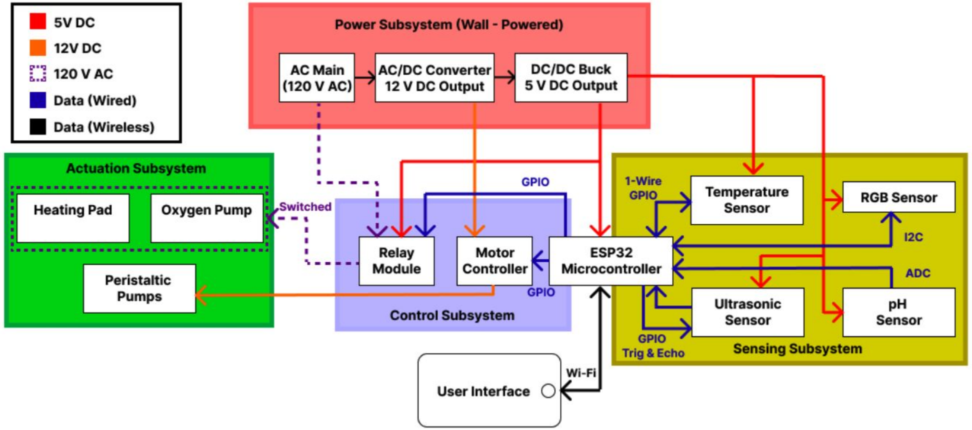
- Unified the system around a single 12 V rail with regulated 5 V and 3.3 V outputs
- **Why:** simpler power distribution and cleaner subsystem integration

All-DC actuation

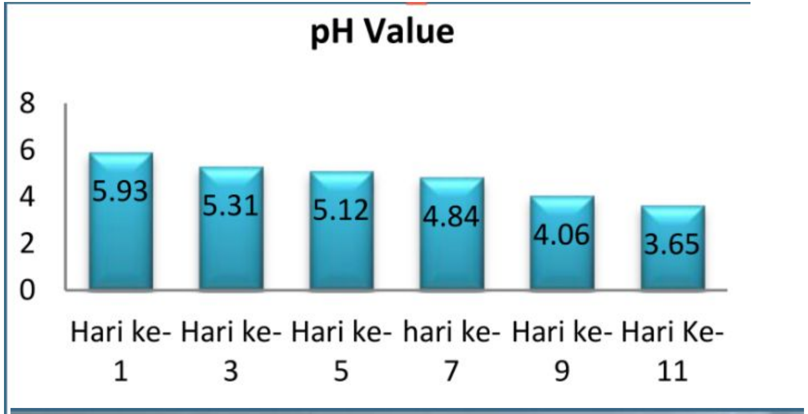
- Removed earlier AC-side assumptions for heating and pumping
- **Why:** reduced electrical complexity and improved implementation safety

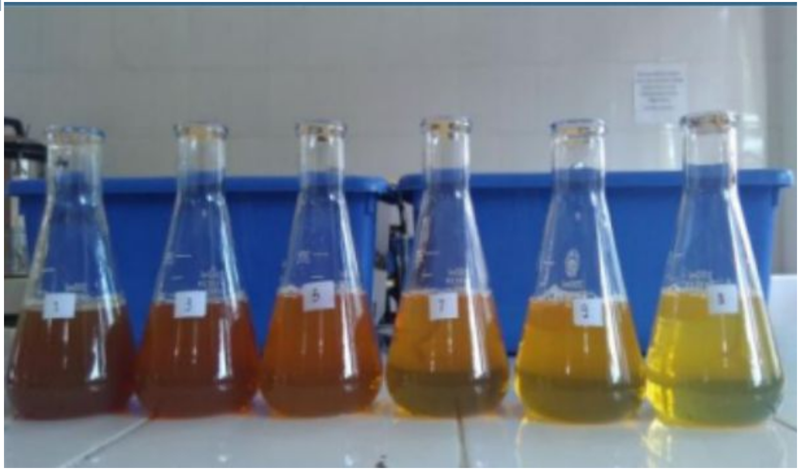
Cloud-based monitoring pipeline

- Replaced the local ESP32-hosted dashboard approach with **ESP32 → EMQX → ingest → Supabase → web dashboard**
- **Why:** enabled remote monitoring, historical data storage, and Discord alerts



Sensing Subsystem



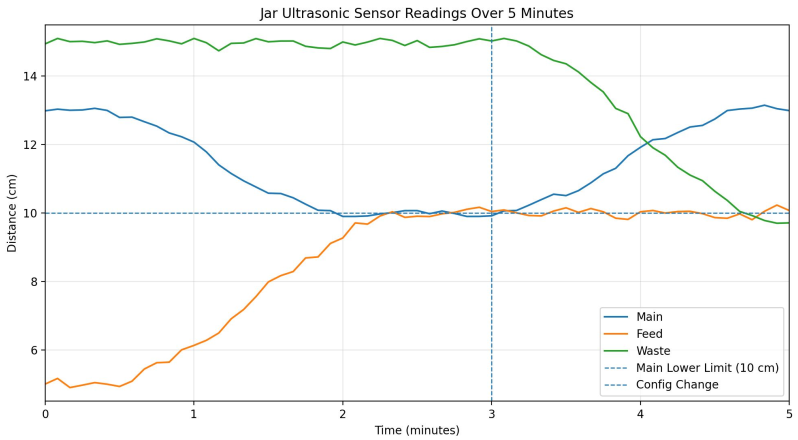


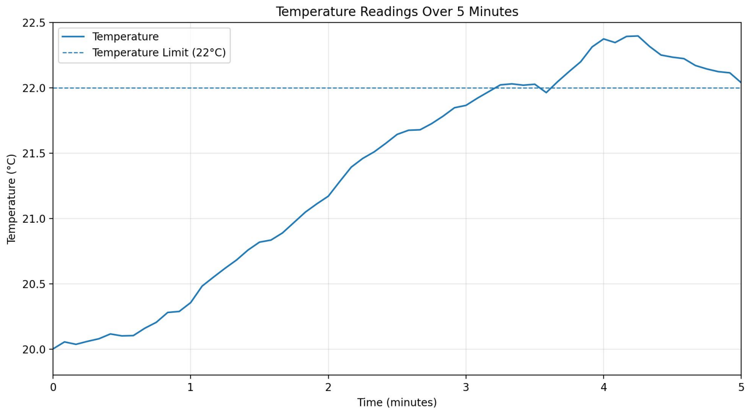
- pH: Increases in acidity overtime
- RGB: Loses saturation over time
- Temperature: Must stay consistent in 80-90℉
- Ultrasonic: Measure water level for main, waste, and feed jar

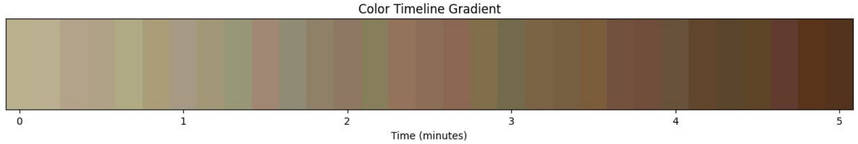
Temp: >= sample/s, stable to ±1°C	Serial log/website	Done
pH repeatability ± 0.3 over 5 min	Serial log/website	Done
Ultrasonic level ±1 cm repeatability	Fixed-level test	Done
RGB/HSL detects light-to-dark shift	Serial log/website	Done

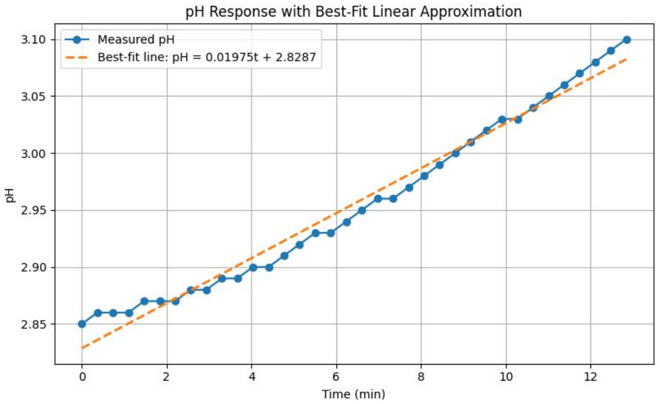
[Nurikasari et al. 2017]

Sensing Subsystem

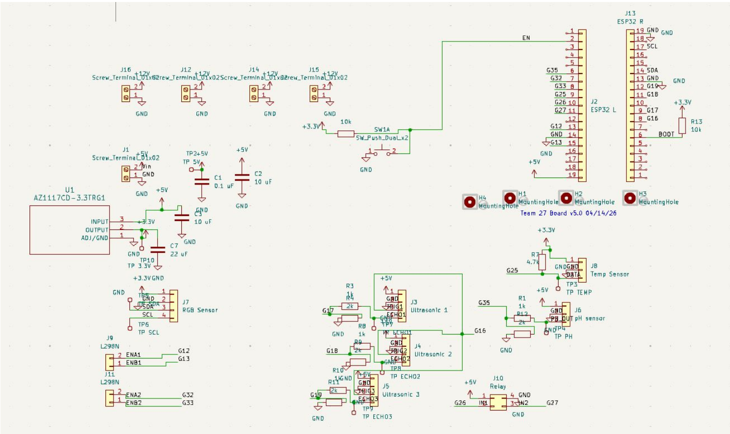








Sensing Subsystem

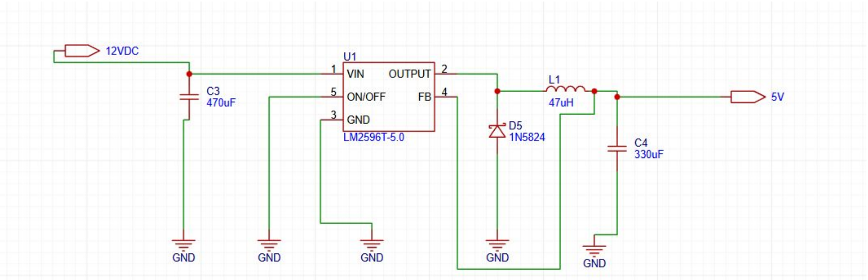


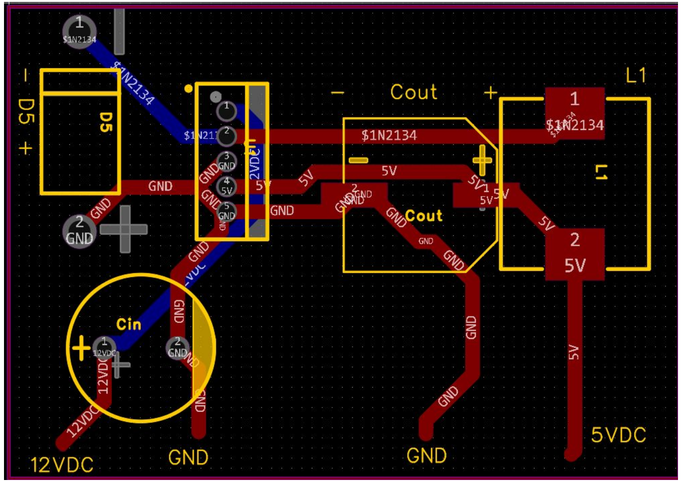
- pH: ADC (analog voltage)
- RGB: I2C
- Temperature: 1-Wire GPIO
- Ultrasonic: GPIO — shared TRIG pin, separate ECHO pins per sensor

Sensor	Interface	Voltage	Level-Shifting Needed
pH	ADC	5V output	Yes - resistor divider
RGB	I2C	3.3V	No
Temperature	1-Wire GPIO	3.3V	No
Ultrasonic	GPIO TRIG/ECHO	5V ECHO	Yes - resistor divider

Power Subsystem

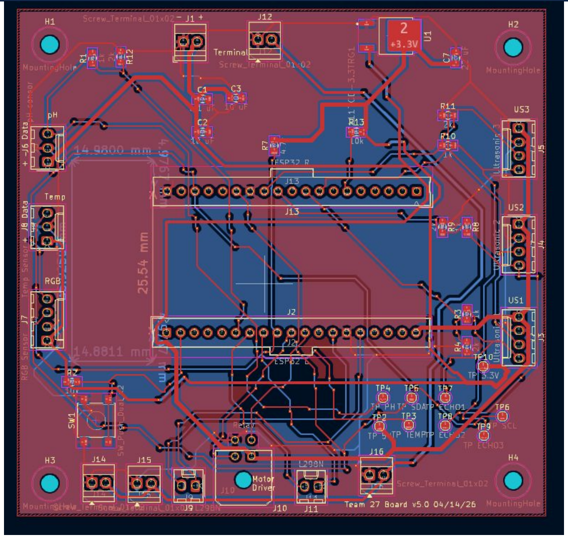
- Converts 120 VAC wall power to regulated 12 VDC using external AC/DC adapter
- Distributes 12 VDC rail to pumps, heating pads, and motor drivers
- Uses LM2596 buck converter to step 12 V → 5 V for control electronics
- Uses LDO to generate 3.3 V for ESP32 and low-voltage sensors(implemented on main PCB)
- Provides stable multi-rail power (12 V, 5 V, 3.3 V) for reliable sensing, control, and actuation





Power	5 V rail ± 5%	DMM	Done
Power	3.3 rail ± 5%	DMM	Done

Control + Actuation Subsystem

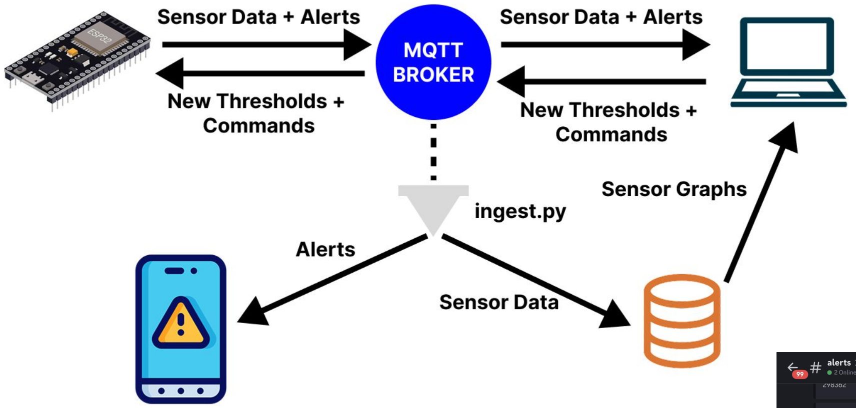


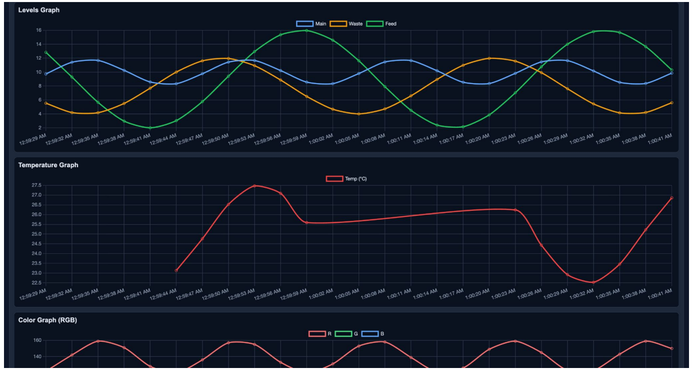
- The resistor divider for VADC = Vph *R2/(R1 + R2) with R1 = 1kΩ, R2 = 2kΩ giving 3.33 V max - safe for ESP32.
- Ultrasonic sensor -> peristaltic pump -> switches to waste at full capacity
- Oxygen pump on timer
- Temperature -> relay -> heater on/off
- The heating feasibility: $Q=mc\Delta T=(1.0)(4186)(3)=12558$ J, giving t=12558/50 s =251 s ≈ 4.2 min

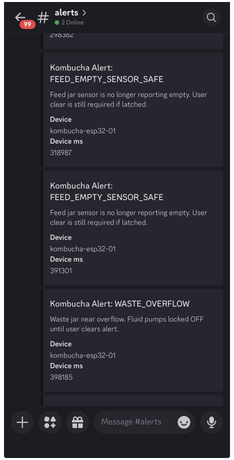
Actuation	Heating: >= 0.15°C/min for 10 min	Temperature log	Done
Control	Heater turns on below, off above	Observation	Done
Control	Oxygen pump activated on schedule	Observation	Done
Control	Liquid level maintained in the main jar via pump actuation	Observation	Done

User Subsystem

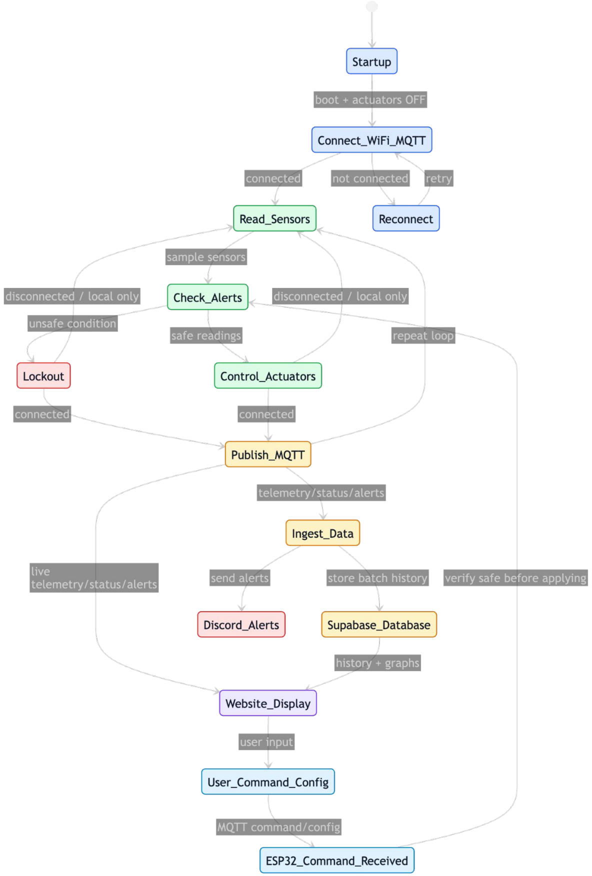
- Provides remote monitoring, visualization, and user interaction through a web dashboard and phone alerts.
- The ESP32 publishes sensor data and alert conditions to an MQTT broker
- The dashboard displays live readings and historical sensor graphs from stored data.
- Users can send threshold updates and commands back to the controller.
- An ingestion script stores telemetry in the database and forwards alerts to the user.







Software Architecture



- The system starts by turning all actuators off, then connects to WiFi/MQTT.
- The ESP32 continuously reads sensors and checks for unsafe conditions.
- If readings are safe, it controls the pumps, heater, and oxygen system.
- If an alert occurs, the system enters a lockout state to protect the hardware and fermentation setup.
- When connected, telemetry and alerts are published to the website and sent through `ingest.py`.
- `ingest.py` stores batch history in Supabase and can send Discord alerts.
- The website displays live data, history graphs, and allows the user to send commands/configuration back to the ESP32.
- If WiFi/MQTT disconnects, the ESP32 keeps running local control and resumes publishing once reconnected.

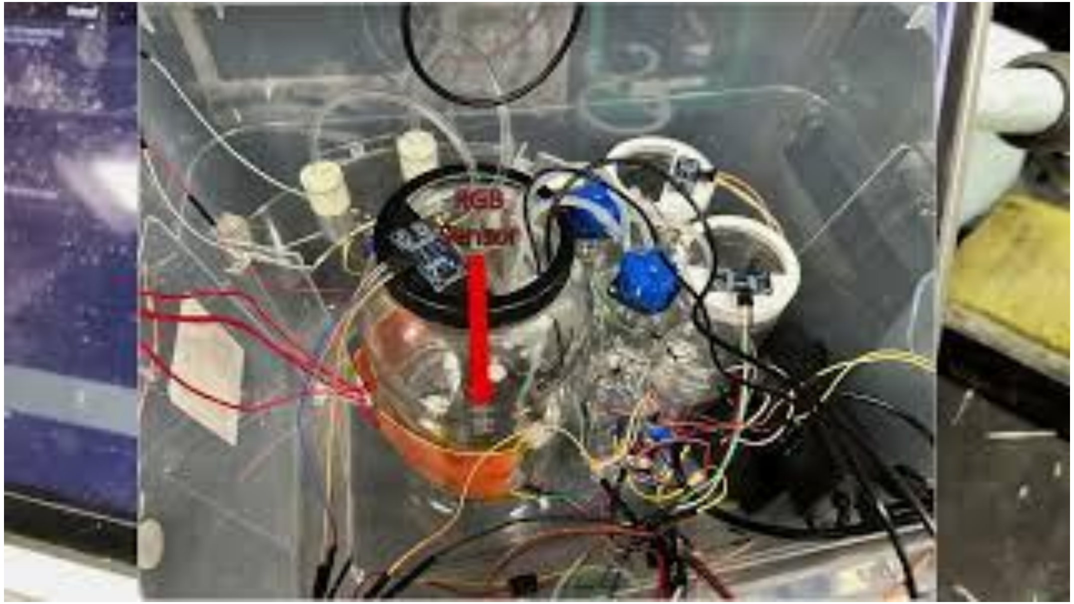
Requirements and Verification

Subsystem	Requirement	Verification	Status
Power	5 V rail ± 5%	DMM measurement	Done
Power	3.3 rail ± 5%	DMM measurement	Done
Sensing	Temp: >= sample/s, stable to ±1°C	Serial log/website	Done
Sensing	pH repeatability ± 0.3 over 5 min	Serial log/website	Done
Sensing	Ultrasonic level ±1 cm repeatability	Fixed-level test	Done
Sensing	RGB/HSL detects light-to-dark shift	Serial log/website	Done
Actuation	Heating: >= 0.15°C/min for 10 min	Temperature log	Done
Control	Heater turns on below, off above	Observation	Done
Control	Oxygen pump activated on schedule	Observation	Done
Control	Liquid level maintained via pumps	Observation	Done

Functional Test Results

What's working

- Power rails verified at 5V, 12V, and 3.3V.
- Ultrasonic/volume feeding loop functional.
- Oxygen pumps physically creating bubbles in the liquid.
- RGB sensor detecting color change
- pH sensor calibrated and tested
- Temperature sensor reading liquid temperature accurately
    a. Used probes to double check measurements (pH & temperature)
- Dashboard showing all sensor readings with timestamped logs.



Ethics/Safety

- **Sensor fault detection:** Invalid temperature, pH, color, or level readings are flagged instead of being trusted blindly.
- **Alert system:** The dashboard clearly shows unsafe conditions such as feed empty, waste overflow, main jar too full, or sensor faults.
- **Fluid lockouts:** Pumps are blocked when liquid levels indicate overflow, empty reservoirs, or unsafe jar conditions.
- **Heater lockout:** If the temperature sensor fails, the heater is disabled to prevent overheating.
- **User acknowledgement:** Some alerts stay latched until the user physically fixes the issue and clears the warning.
- **Food-safe design:** Materials in contact with kombucha should be cleanable and safe for acidic liquids.
- **Electrical safety:** The design separates low-voltage sensor/control electronics from higher-power actuator hardware.

Learned/Needed Improvements

Needed Improvements

RGB sensing: Translucent broth produced responsive but less reliable readings, and calibration was difficult.

pH sensing: The pH sensor was hard to keep calibrated and broke after being placed too deep in the broth.

Batch tracking: Hard to retrieve correct batch data from Database

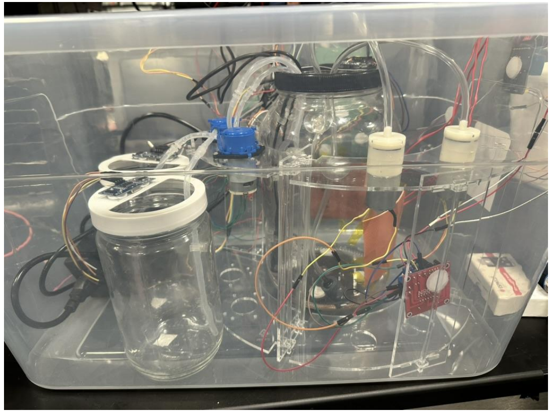
Heating system: The heating pads did not stick securely to the jars.

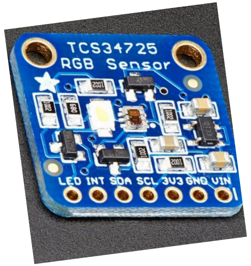
Mechanical design: The jars and sensor setup were messy/ difficult to clean and maintain.

Onsite Controls: If the wifi was disconnected there is no way to know the problems with system

What we Learned

- **PCB development**
- **IoT pipeline**
- **Real-time embedded processing**
- **Proper testing mindset**





Conclusion

Design Summary

Built a closed-loop kombucha fermentation system with an ESP32-based custom PCB, integrating temperature, pH, RGB, and ultrasonic sensors with a heating pad, two peristaltic pumps, and two oxygen pumps, all monitored via a Wi-Fi dashboard.

Future work

- Enclosure/physical cleanup
- Full multi-day fermentation run (longer than 20 minutes)
- Long-term logging (one week)
- Polish of website
- Mold/contamination detection alerts.
- Indicators of operation on system itself
- Addition of pressure controlled environment for second stage
    a. Requires sealed, rigid container and pressure sensors

Thank You

Q & A