UNIVERSITY OF LONDON

(CM3040)
Physical Computing and IoT Project
Final Report

Smart Hygiene Station

Ellen Faustine

220570273

Table of Content

Abstract

This project introduces an automated handwashing system consisting of a water dispenser, soap dispenser, and tissue dispenser, working sequentially to ensure a contactless hygiene process. Using ESP8266 microcontrollers, ultrasonic sensors, and relays, the system detects user presence and activates each unit accordingly. HTTP-based communication synchronizes the dispensers, while a web dashboard provides real-time monitoring. Designed for efficiency and accessibility, the system enhances hygiene by reducing direct contact, making it ideal for both public and private settings.

Introduction

Background

Maintaining hygiene is essential, but traditional hand washing setups can be challenging for individuals with disabilities due to physical barriers and the need for assistance. The Smart Hygiene Station leverages IoT technology to offer a fully automated, contactless solution, ensuring accessibility by sequentially dispensing water, soap, and tissue for a seamless user experience.

Aims & Objective

This project aims to develop a compact, portable IoT-enabled handwashing system that can be conveniently installed in homes or any location with a water source, helping to bridge the gap for individuals with disabilities to wash their hands independently. Designed to improve accessibility for individuals with disabilities, the system provides a fully automated, touch-free hand washing experience.

Hardware Design

Overview of Hardware Components

This project consists of three interconnected units: water dispenser, soap dispenser, and tissue dispenser, that operate sequentially to automate the handwashing process. The system is equipped with various sensors to detect user presence, indicate its operation, and activate the dispensers accordingly. The integration of these components enables a fully automated and accessible hand hygiene solution. All three units are housed on a cardboard platform to ensure system cohesiveness and portability.

Water Dispenser

The first unit of the station is the water dispenser, powered by a single ESP8266 microcontroller and equipped with two ultrasonic sensors for automated operation. One sensor detects hand presence, triggering the water pump for 5 seconds to dispense water. The second sensor monitors the water level in the container to ensure continuous functionality. To enhance accessibility, a blue LED and a buzzer activate during operation, providing both visual and auditory feedback for users with visual or hearing impairments. If the water level is low, a red LED lights up as a warning signal.

The water pump is housed inside a lightweight paper cup, chosen for portability, affordability, and ease of refilling. The cup's lid opening also conveniently supports the placement of the water tube. Since the ESP8266 cannot directly power the pump, an external battery pack is used, while a relay module regulates power flow. A 3.3V relay was selected to ensure compatibility with the ESP8266's voltage output, allowing automatic control of the pump. This unit ensures efficient water dispensing while maintaining a compact, portable, and accessible design.

Soap dispenser

The soap dispenser operates similarly to the water dispenser, with the primary difference being that it uses only one ultrasonic sensor instead of two. This single sensor is responsible for monitoring the soap level inside the container.

Positioned between the water and tissue dispensers, the soap dispenser ensures a logical and intuitive flow for the handwashing process. It utilizes an ESP8266 microcontroller to control its functions, and a water pump housed inside a paper cup for dispensing soap. The wide opening and strong suction allow for the use of low-viscosity liquid soaps, ensuring smooth operation.

Once activated, a yellow LED and a buzzer provide visual and auditory feedback, signaling that the dispenser is in use. This enhances accessibility for individuals with visual or hearing impairments, making the system more inclusive.

Tissue Dispenser

The tissue dispenser unit is powered by a single ESP8266 microcontroller and utilizes a servo motor as the tissue holder and dispenser mechanism. It also incorporates a roller ball limit switch to detect tissue availability.

The limit switch plays a crucial role in monitoring tissue levels. When the switch is pressed by the tissue roll, it confirms that tissue is available (Figure 1). If the switch is not pressed, it indicates that the tissue has run low, signaling the need for a refill.

A green LED lights up when tissue is present and the unit is in operation. The servo arm holds the tissue roll in place and rotates to dispense tissue when activated, ensuring a smooth and controlled dispensing process.

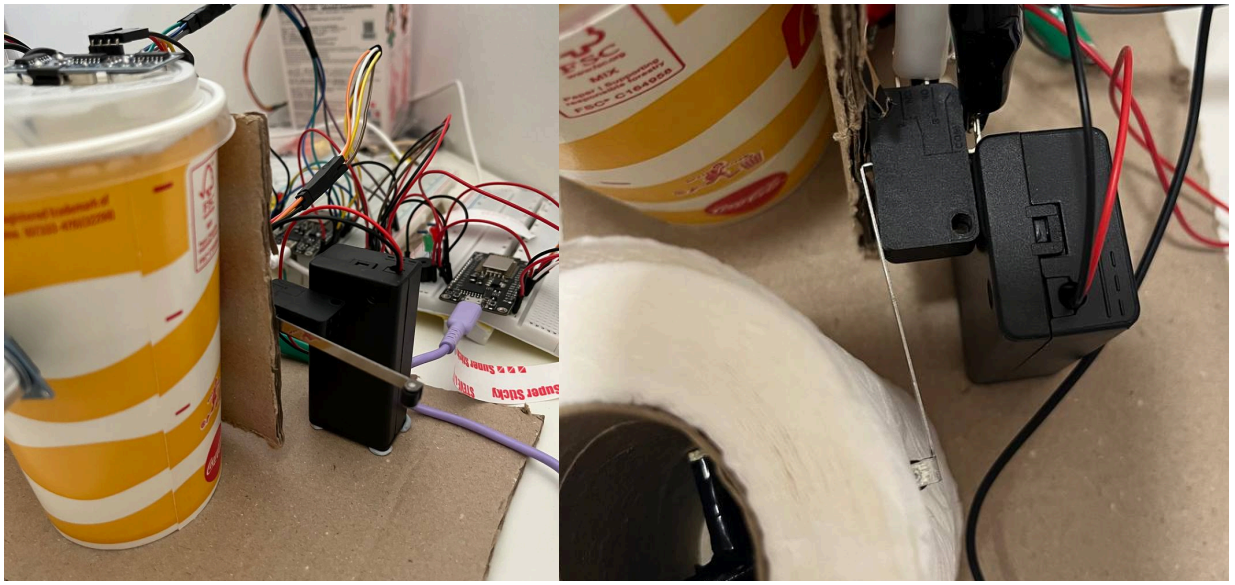


Figure 1. position of the limit switch

Development lifecycle

Planning & Requirement Analysis

The development of the automated hand washing system began with a comprehensive planning and requirement analysis phase to ensure seamless functionality and integration of all units. The primary goal was to design a system consisting of a water dispenser, soap dispenser, and tissue dispenser that would operate sequentially to automate the handwashing process. To achieve this, I identified key hardware components, including ESP8266 microcontrollers, ultrasonic sensors for user detection, relays for actuator control, and WiFi communication for inter-unit synchronization.

Research was conducted on HTTP-based communication to facilitate efficient signal exchange between the units, ensuring proper coordination in the dispensing process. Additionally, I explored various design options for each unit, focusing on sensor placement, response time, and power efficiency to enhance reliability and performance. Careful selection of components was essential to maintain accuracy in detection and dispenser activation. This planning phase established a clear foundation for implementation, ensuring all technical and functional requirements were well-defined before development began.

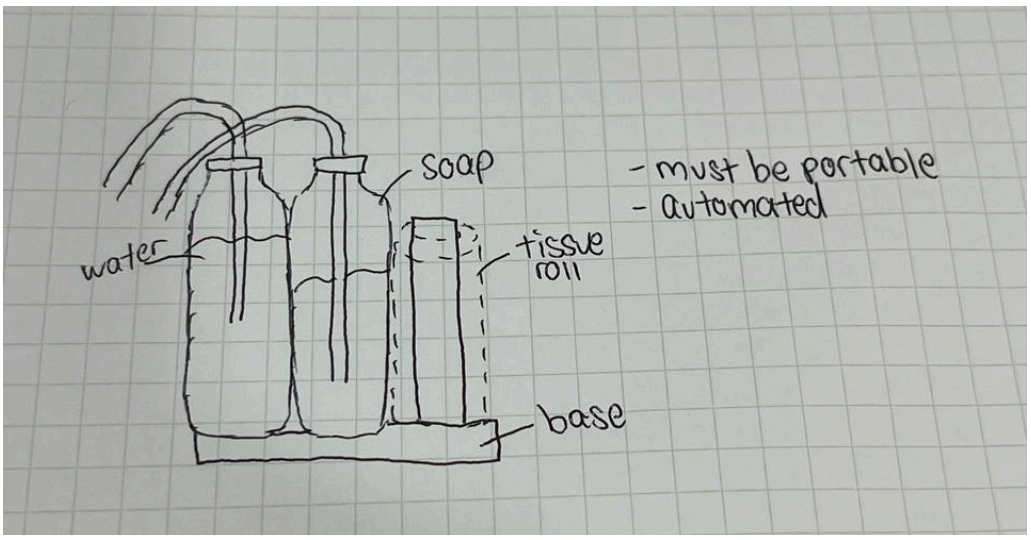


Figure 2. Design Plan

System Architecture

To ensure a seamless and synchronized workflow, the dispenser units communicate with each other after completing their respective processes. This is achieved through HTTP GET requests that allow each unit to notify the next one when it is time to activate.

Each ESP8266 microcontroller is programmed to send an HTTP request with specific parameters once its stage process is complete. The receiving unit extracts these parameters and updates its global variable, allowing it to transition to the next stage of the sequence. This method ensures that all dispensers operate in a coordinated manner, preventing unnecessary activations or delays.

For example, after the soap dispenser finishes dispensing, it sends a request to the water dispenser, signaling it to begin its process. The HTTP request might look like this:

http.begin(client, "http://<water_dispenser_IP>/update_stage?stage=2");

Upon receiving this request, the water dispenser extracts the "stage" parameter, updates its global variable, and proceeds with its operation. To enhance reliability, error handling mechanisms were implemented. If an HTTP request fails, the system attempts to resend it after a short delay, ensuring successful communication even in cases of temporary network instability.While HTTP-based communication effectively manages synchronization, future improvements could involve transitioning to MQTT messaging, which provides lower latency and better efficiency for IoT-based communication.

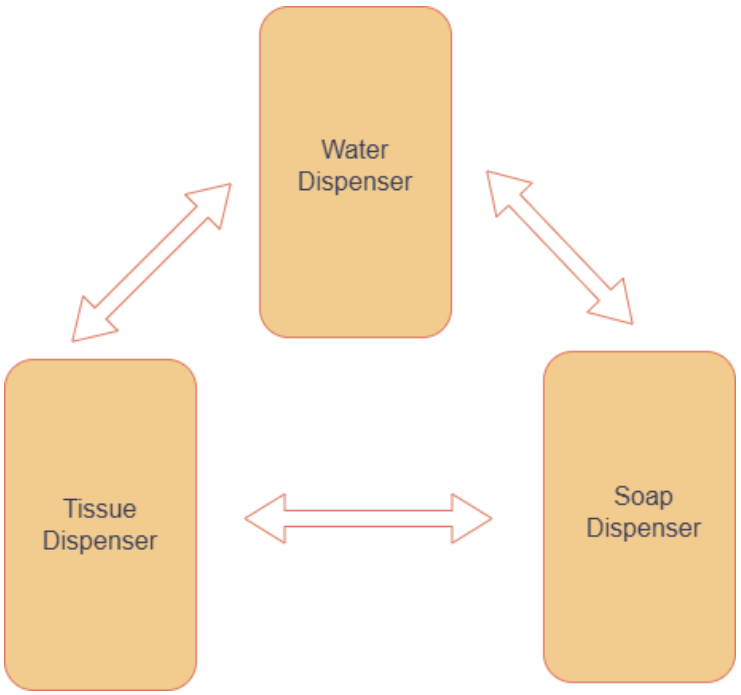


Figure 3. Smart Hygiene Station System Architecture

Implementation & Prototyping

For this project, I followed an iterative development approach, similar to the Agile methodology, where I conducted continuous testing and improvements throughout the development cycle. Since the system consists of multiple independent units, I developed and tested each unit separately before integrating them into the full system.

I began by developing the water dispenser, as it is a crucial part of the station. Initially, I created a simple circuit connection involving the pump, relay, and battery (Figure 4). Before adding complexity, I tested this basic setup to ensure that the pump operated correctly. Once the initial circuit was validated, I gradually introduced additional components, following a structured cycle of building, testing, and refining.

For example, after successfully testing the pump circuit, I added an ultrasonic sensor to detect hand presence and trigger dispensing (Figure 5). This incremental approach allowed me to identify and resolve issues early, ensuring that each component functioned correctly before moving forward. I repeated this cycle for each new sensor or feature, refining the system step by step until all dispensers were fully functional and synchronized.

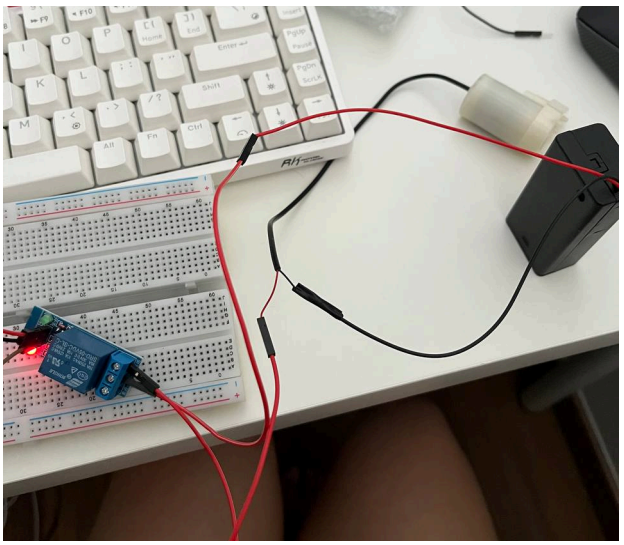


Figure 4. Initial circuit setup

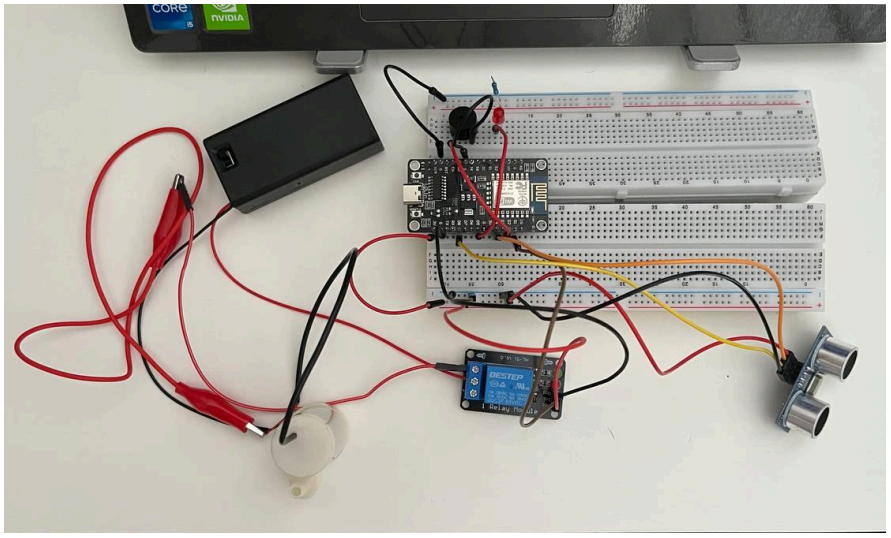


Figure 5. Adding ultrasonic sensor to the circuit

After successfully developing and testing the entire unit in its circuit form, I proceeded to integrate the physical structure (Figure 6), incorporating the paper cups as containers for the dispensers. This transition was essential to ensure that the system remained lightweight, portable, and easy to refill while maintaining functionality.

At this stage, I carefully positioned the water pump inside the cup, securing the tubing to align with the dispensing mechanism. I also ensured that the ultrasonic sensor was properly mounted to detect user presence without obstruction. This integration required adjustments to the sensor placement and wiring layout to maintain stability and avoid interference with the dispensing process.

Testing continued during this phase to confirm that the system performed consistently in its physical form, ensuring that the added structure did not affect sensor accuracy or pump efficiency.

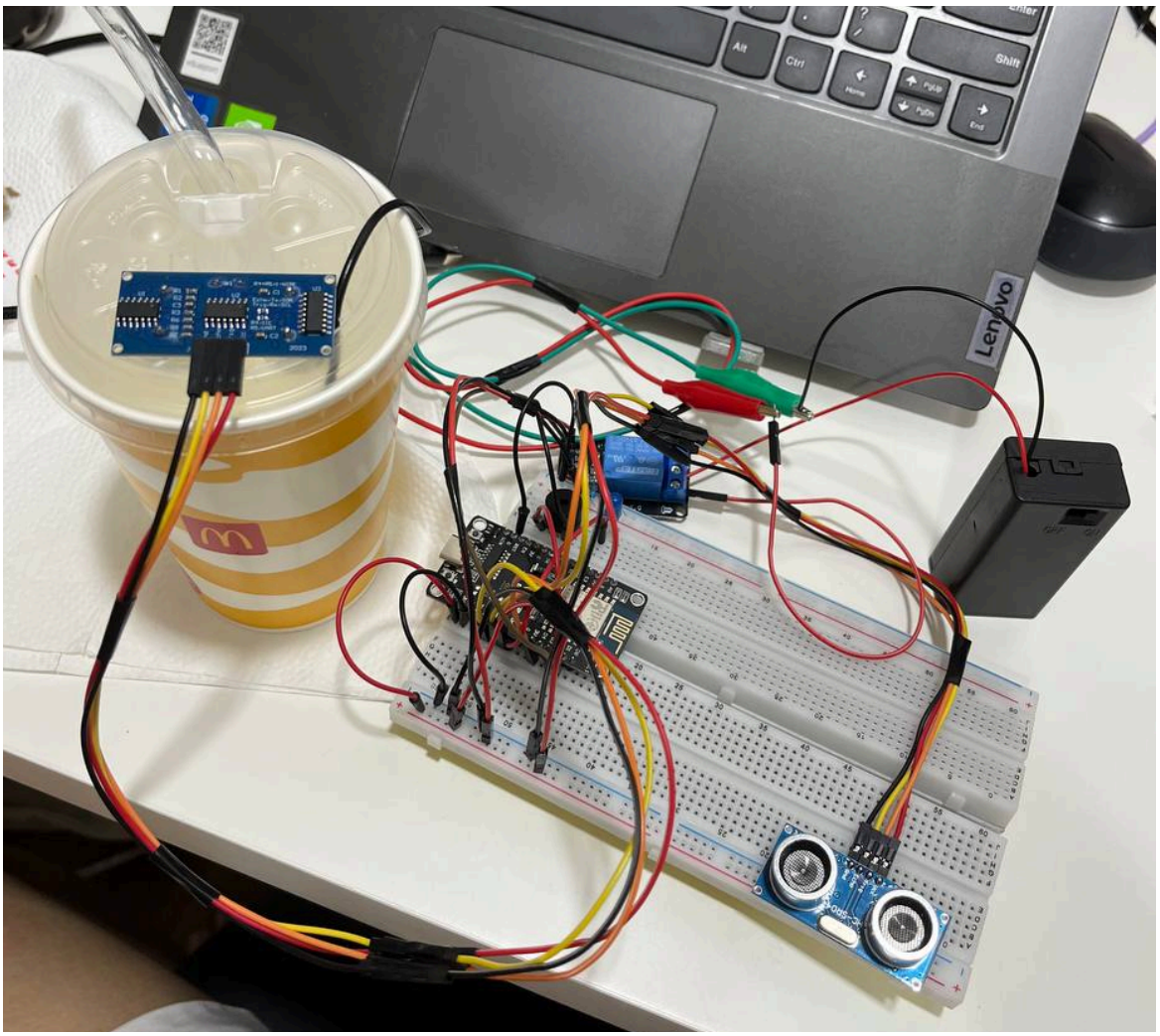


Figure 6. Integrating physical structure

Deployment and final integration

After successfully developing the water, soap, and tissue dispensers as independent units, the next step was to integrate them into a cohesive system. To ensure a systematic setup, I followed a step-by-step approach for assembling the units together.

First, I gathered all three dispensers and analyzed which components should be integrated first. Since the water and soap dispensers work in sequence, it was logical to set them up together initially. I mounted the water dispenser onto a cardboard platform and positioned the soap dispenser next to it, ensuring a stable and accessible layout while still leaving space for the tissue dispenser (Figure 7).

During this integration, I carefully arranged the wiring and sensor placement to prevent interference and ensure smooth operation. The goal was to maintain an intuitive flow for users while ensuring each unit could function without obstruction.

Figure 7. Integration

Once the water and soap dispensers were physically set up, the next step was to enable communication between the units. I implemented an HTTP-based signaling system, where each unit would send a signal upon completing its process, allowing the next dispenser to activate.

To verify communication, I tested whether each dispenser could send and receive HTTP requests successfully. I monitored the global variable updates to ensure that when a unit finished its stage, the next unit would be triggered accordingly. After confirming that the signal was received correctly and the variables were updated, I proceeded to integrate the tissue dispenser into the system.

With all three dispensers now set up and synchronized (Figure 8), I conducted final testing to ensure smooth operation. This involved verifying that the sequence was correctly followed, that no unit activated prematurely, and that all feedback mechanisms (LEDs, buzzers) functioned as intended.

Figure 8. Final deployment

Software & Firmware Development

For this project, I initially developed separate codes for each unit in Arduino IDE to maintain a modular structure, allowing each dispenser to manage its own specific actions independently. I started by writing the software for a basic circuit setup, ensuring each unit could function on its own before integrating communication between them.

Once the individual units were operational, I implemented HTTP communication between the nodes. To ensure reliability, I first established successful communication between two nodes before integrating the third. This step-by-step approach allowed me to debug and optimize the data exchange process, ensuring seamless synchronization.

Each unit features its own individual dashboard, displaying relevant information such as water, soap, and tissue levels, pump status, and system progress. The dashboards were developed using a combination of HTTP and JSON, utilizing the ESP8266WiFi library and HTTPClient to send and receive data between units. These dashboards provide real-time

sensor readings and track the station’s progress, ensuring efficient monitoring and control of the handwashing system.

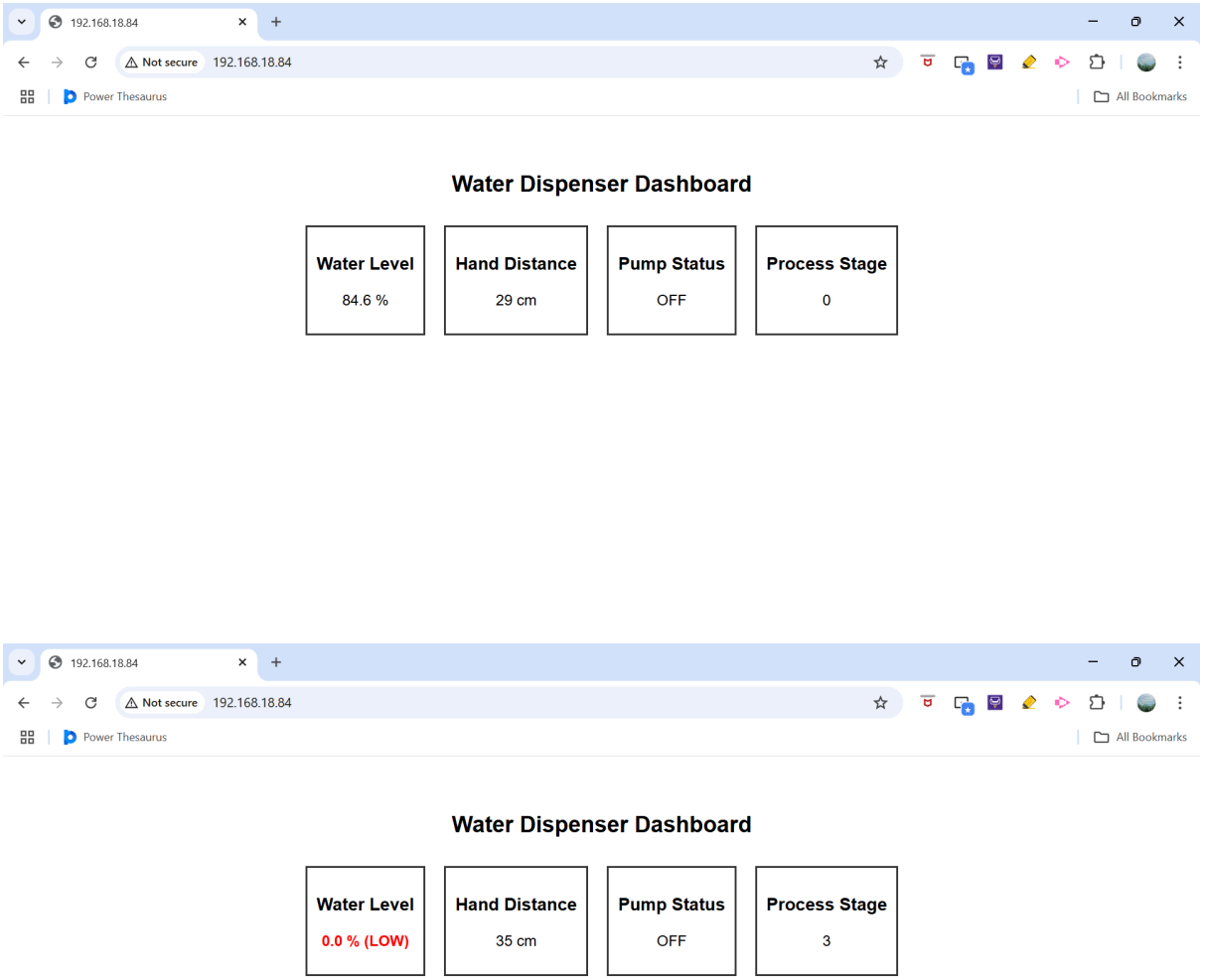


Figure 9. Water Dispenser Dashboard

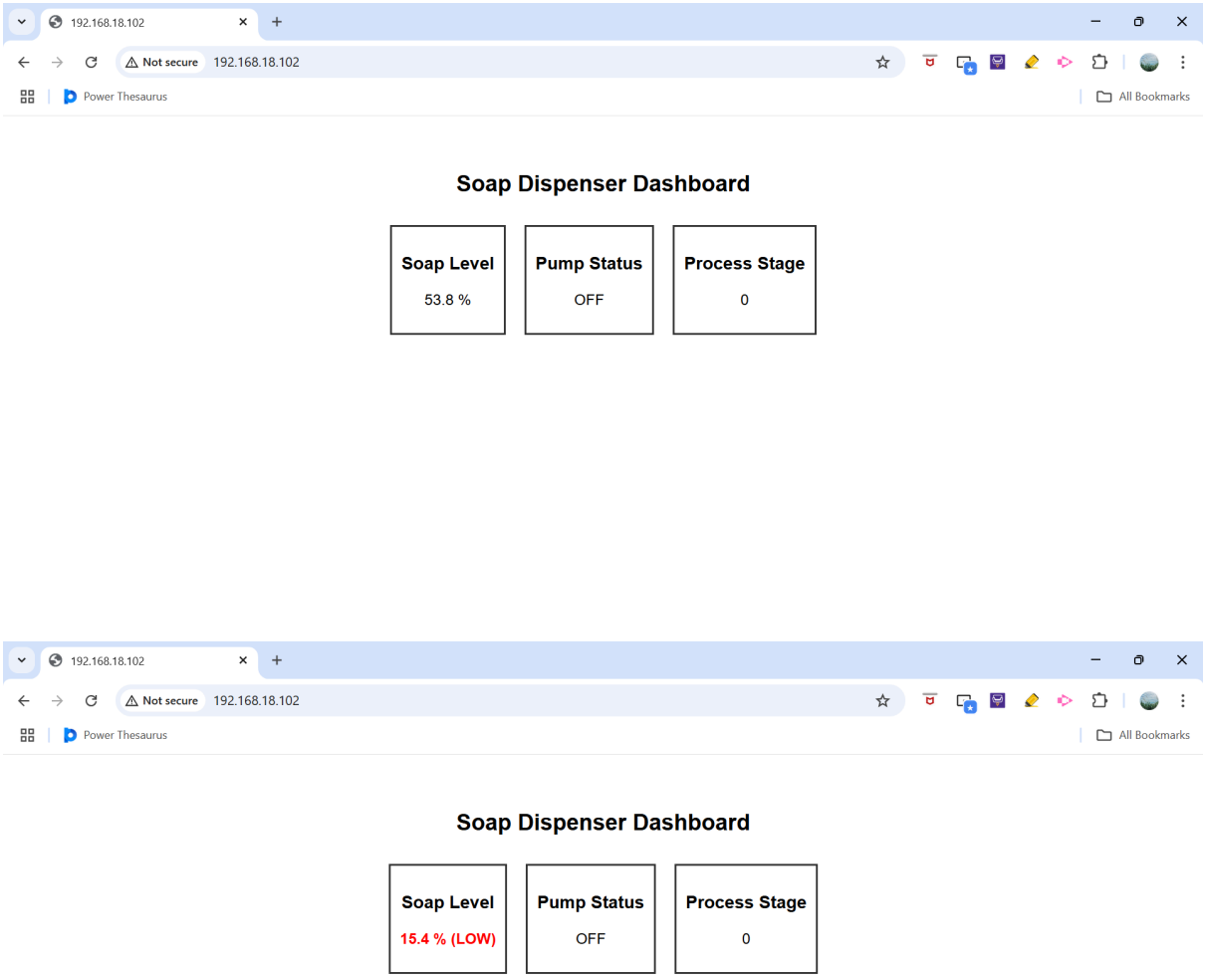


Figure 10. Soap Dispenser Dashboard

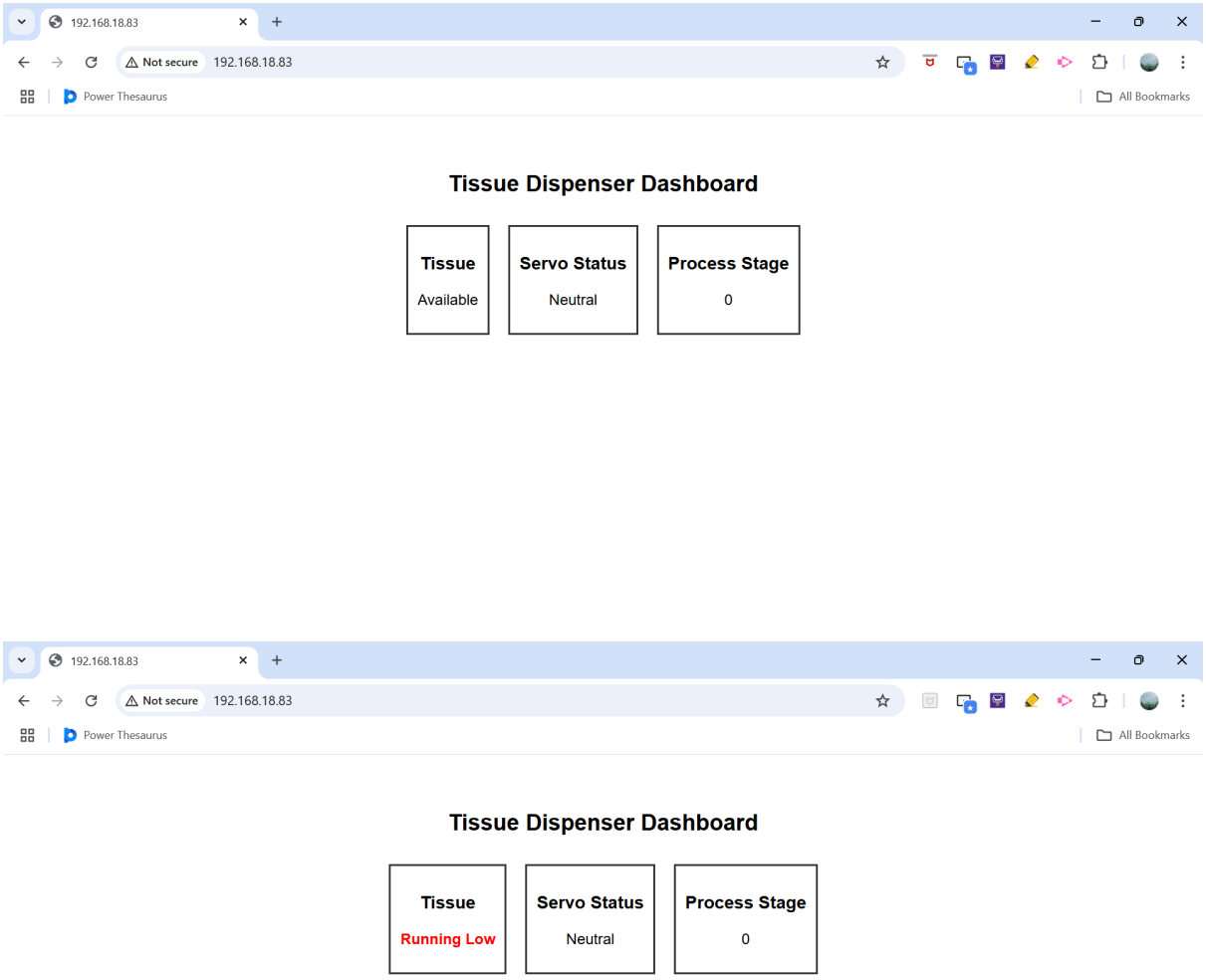


Figure 11. Tissue Dispenser Dashboard

A crucial part of the automation cycle is the sendCompletionSignal() and updateStage() functions. Without these two functions, the seamless coordination between dispenser units would not be possible.

The sendCompletionSignal() function (Figure 12) is responsible for notifying the other nodes when a unit has completed its process. It does this by sending an HTTP GET request to the next unit, signaling it to update to the next process stage. This ensures that each dispenser activates in the correct sequence without manual intervention.

The updateStage() function (Figure 13) processes incoming HTTP requests by checking for the "stage" argument in the request parameters. It then converts the received string value into an integer and stores it in the global variable processStage. This allows each unit to recognize its current stage in the sequence and take the appropriate action.

Together, these two functions enable the synchronized operation of the water, soap, and tissue dispensers, ensuring that the handwashing cycle progresses smoothly from one stage to the next.

```
// Function to Notify Nodes
void sendCompletionSignal() {
  WiFiClient client;
  HTTPClient http;

  // Notify Water Dispenser to update to stage 2
  String serverPath1 = "http://" + String(node1_IP) + "/update_stage?stage=2";
  Serial.println("Sending signal to Water Dispenser (Node 1)...");
  http.begin(client, serverPath1);
  int httpResponseCode1 = http.GET();
  if (httpResponseCode1 > 0) {
    Serial.println("Node 1 Response: " + String(httpResponseCode1));
  } else {
    Serial.println("Error sending request to Node 1.");
  }
  http.end();

  // Notify Tissue Dispenser to update to stage 2
  String serverPath2 = "http://" + String(node3_IP) + "/update_stage?stage=2";
  Serial.println("Sending signal to Tissue Dispenser (Node 3)...");
  http.begin(client, serverPath2);
  int httpResponseCode2 = http.GET();
  if (httpResponseCode2 > 0) {
    Serial.println("Node 3 Response: " + String(httpResponseCode2));
```

Figure 12. sendCompleteSignal code snippet

```
// Handle Process Stage Updates
void updateStage() {
  if (server.hasArg("stage")) {
    processStage = server.arg("stage").toInt();
    Serial.print("Received processStage: ");
    Serial.println(processStage);
    server.send(200, "text/plain", "Stage Updated");
  } else {
    server.send(400, "text/plain", "Missing stage data");
  }
}
```

Figure 13. updateStage code snippet

Potential Improvements and Reflections

One major challenge was water level measurement accuracy, as the ultrasonic sensor occasionally produced fluctuating readings due to water surface disturbances and sensor positioning. This inconsistency affected system performance and could have been improved by using a float sensor or capacitive sensor, which offer more stable and precise measurements. Additionally, selecting the right container was difficult, as it needed to be tall enough for sufficient water capacity, lightweight for portability, and compact enough to fit within the system's design constraints. A taller, lightweight alternative could have improved efficiency while maintaining ease of use.

The tissue dispenser mechanism also required enhancement. A stronger motor with a gear or roller system could have ensured smoother dispensing, preventing issues like tearing or jamming. Exploring alternative dispensing mechanisms, such as vacuum suction or a friction-based roller, might have provided more reliable operation and prevented multiple tissues from being dispensed at once.

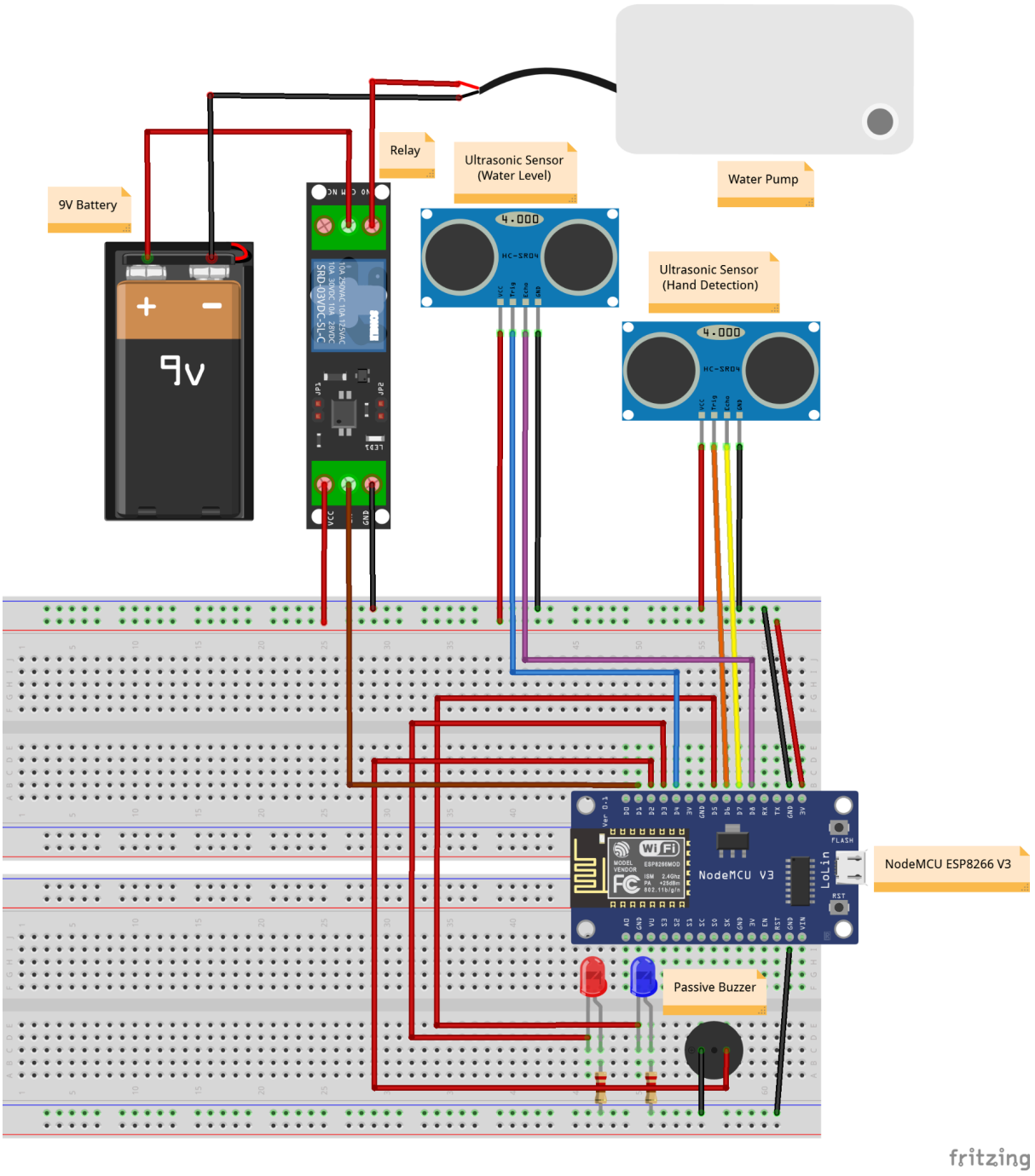
Additionally, using a Painless Mesh network for communication could improve device synchronization and eliminate dependency on a central access point, enhancing the system's overall efficiency. Despite these challenges, the system successfully automated the handwashing process, and future iterations could refine these aspects for greater accuracy, efficiency, and reliability.

Appendices

Instructions

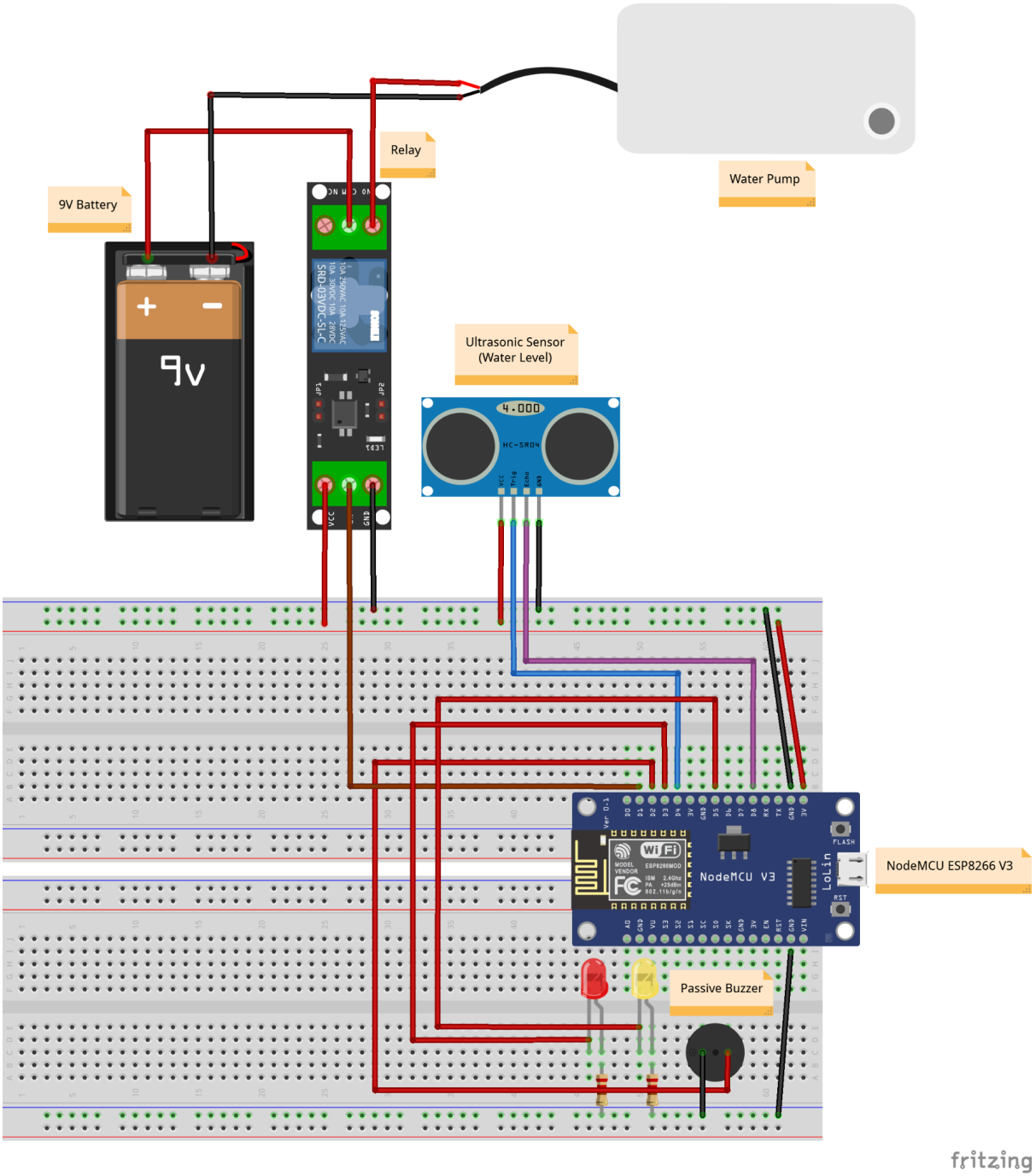
Assembling the Water Dispenser

1. Mount the water pump into the container.
2. Connect the tube to the water pump
3. Connect the pump to a relay module, which will be controlled by the ESP8266.
4. Attach an ultrasonic sensor to detect user presence.
5. Attach another ultrasonic sensor face down on top of the container
6. Wire the components to the ESP8266



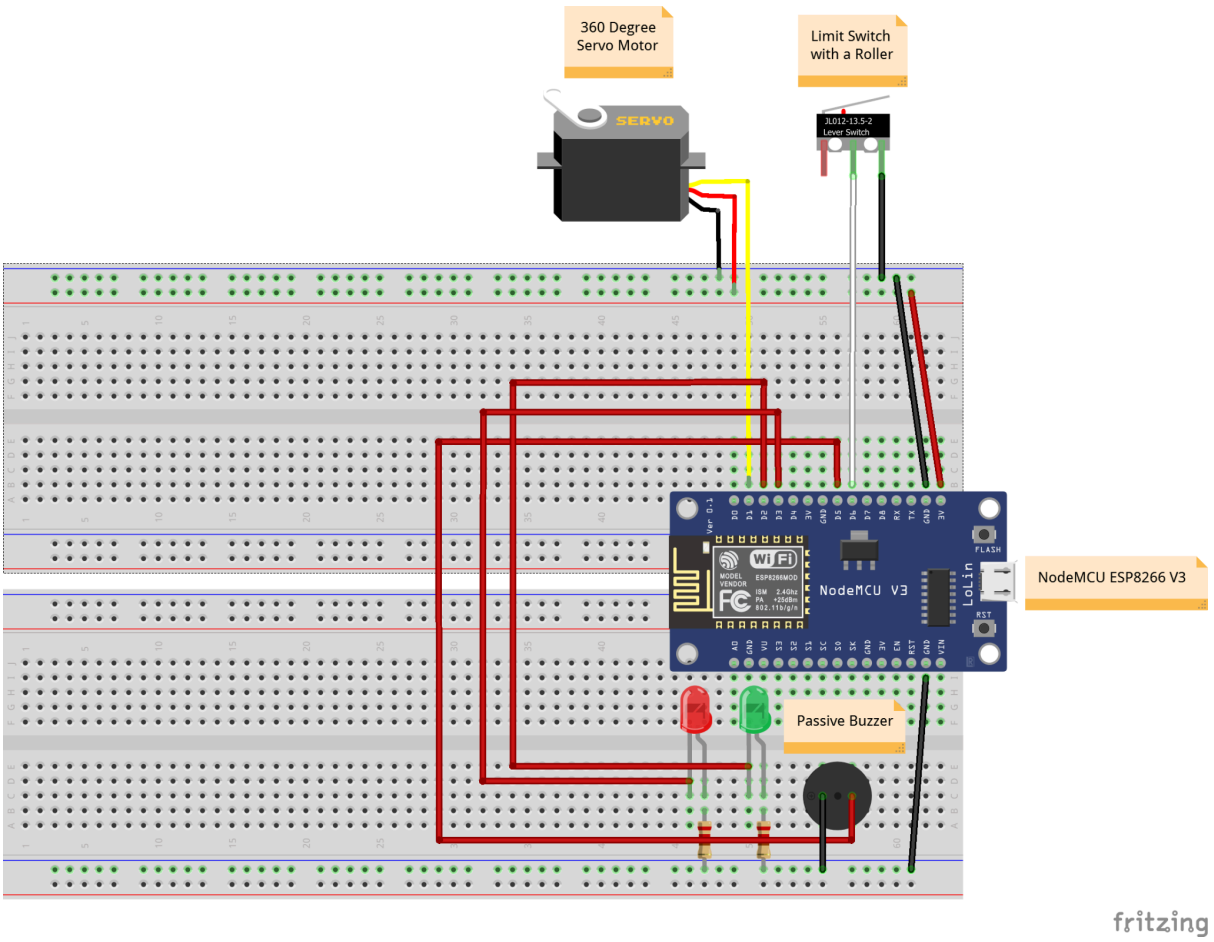
Assembling the Soap Dispenser

1. Mount the water pump into the container.
2. Connect the tube to the water pump
3. Connect the pump to a relay module, which will be controlled by the ESP8266.
4. Attach an ultrasonic sensor to detect user presence.
5. Wire the components to the ESP8266



.Assembling the Tissue Dispenser

1. Mount a servo motor to control on a flat surface
2. Attach a tissue to the servo motor
3. Mount the limit switch on a wall so that the switch can be easily clicked but the it stays put
4. Adjust so that the tissue roll presses on the limit switch until you hear a click and that it stays that way



Component List

Water Dispenser

Components	Quantity
ESP8266 WiFi Module	1
Ultrasonic Sensor	2
Power Adapter/ 9V battery	1
Water Container (Cup)	1
Tube	1
Vertical Water Pump	1
3.3V Relay	1

Blue LED	1
Red LED	1
Buzzer	1
220Ω resistor	2

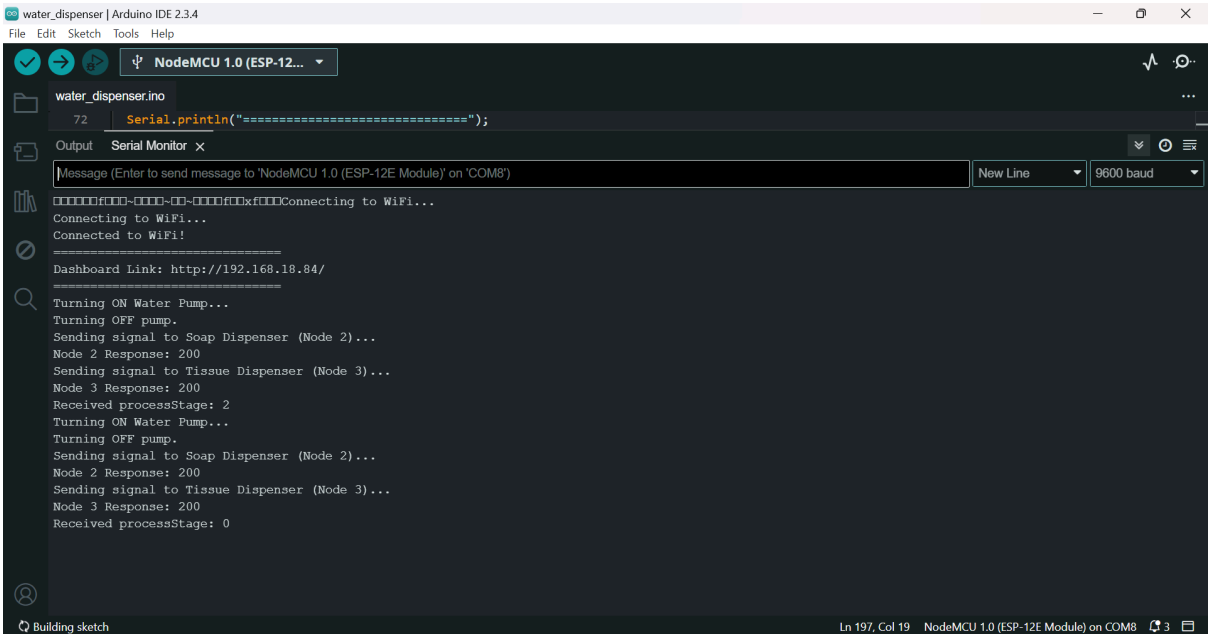
Soap Dispenser

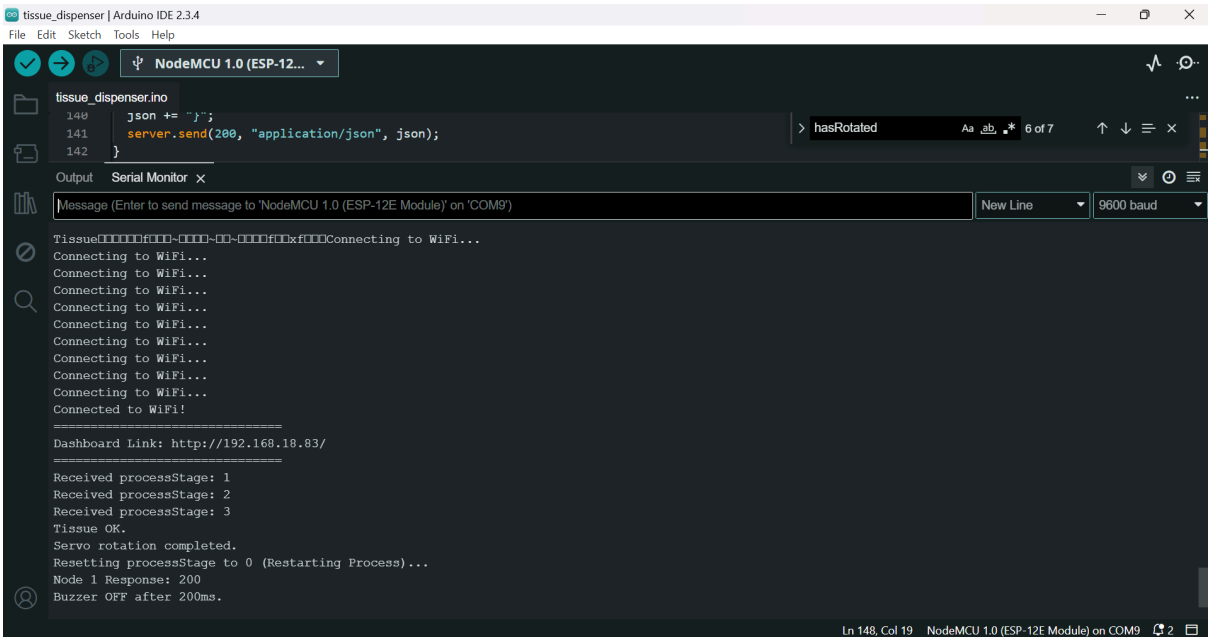
Components	Quantity
ESP8266 WiFi Module	1
Ultrasonic Sensor	1
Power Adapter/ 9V battery	1
Water Container (Cup)	1
Tube	1
Vertical Water Pump	1
3.3V Relay	1
Yellow LED	1
Red LED	1
220Ω resistor	2
Buzzer	1

Tissue Dispenser

Components	Quantity
ESP8266 WiFi Module	1
360° Servo Motor	1
Roller Ball Limit Switch	1
Green LED	1
Red LED	1
Buzzer	1
220Ω resistor	2

Screenshots





Pretty-print ☑


```
{
  "water_level": 6,
  "hand_distance": 72,
  "pump_running": false,
  "process_stage": 0
}
```


Pretty-print ☑


```
{
  "soap_level": 8,
  "pump_running": false,
  "process_stage": 0
}
```


Pretty-print ☑


```
{"process_stage":0,"microswitch":false,"servo_rotated":false,}
```