Smart Greenhouse Automation with Remote Management

EDIRISINGHE P.D.D.M.

BSc. Electrical &
Electronic Engineering

- Project Title (IoT Project) : Smart Greenhouse Automation with Remote Management
- Status : **Ongoing**

- Introduction :

This project aims to automate and monitor an agricultural plant using an ESP32 microcontroller, complemented by Node-RED, ESP Rainmaker and MQTT for enhanced control and communication. The system incorporates sensors for humidity and temperature monitoring, alongside three valves for precise water management. It includes a dashboard for remote control and utilizes WhatsApp notifications for real-time alerts.

The automation system optimizes water usage by activating valves based on environmental conditions and predefined schedules. Leveraging Wi-Fi connectivity, users can remotely access and manage the system through a Node-RED dashboard. MQTT protocol facilitates seamless communication between the ESP32 and Node-RED, ensuring reliable data exchange and command execution.

Node-RED's visual programming interface simplifies integration and automation tasks, empowering users to customize and extend functionalities without extensive programming knowledge. WhatsApp notifications provide timely updates on critical events, enabling proactive responses to plant conditions.

This project showcases the convergence of IoT technologies ESP32, Node-RED, MQTT, and WhatsApp integration to enhance agricultural efficiency, sustainability, and remote management capabilities. Finally, it is controlled using the ESP Rainmaker User interface.

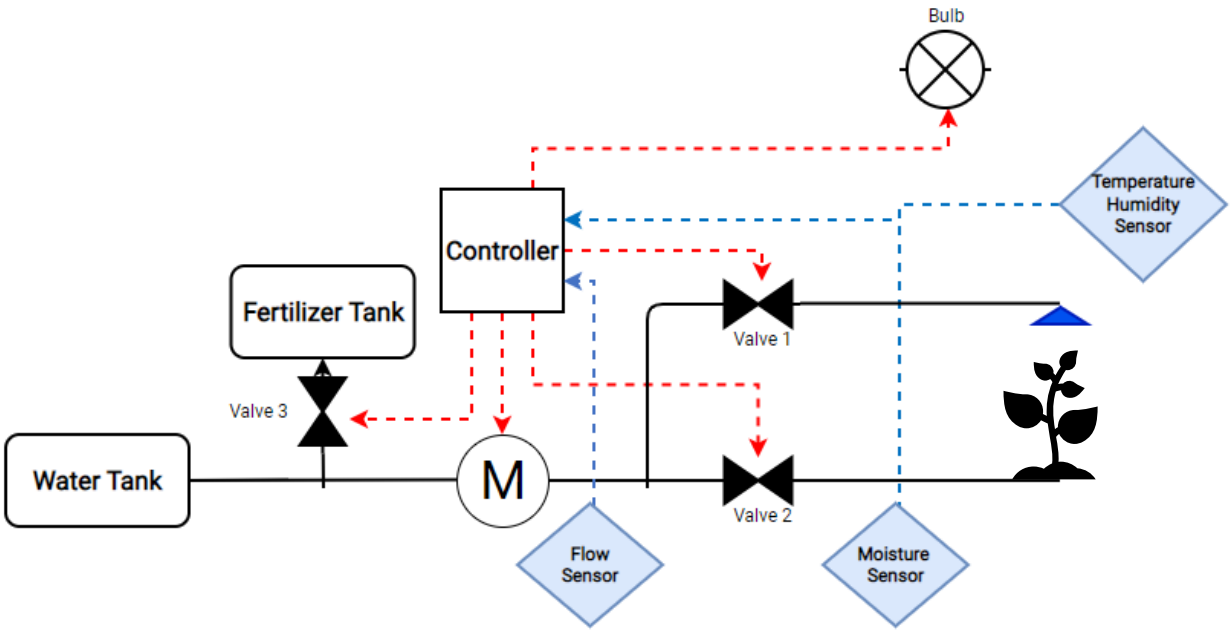

FIGURE 1: SENSOR AND ACTUATORS

Features

a. Wi-Fi offline and online dual mode operation
b. Time synchronized with the Sri Lanka time zone
c. Automated fertilized controller and water controller with time
d. Temperature and humidity are controlled by automatically

Components

a. ESP 32 microcontroller
b. LED dashboard
c. Four-channel relay module
d. Industrial DHT 11 temperature and humidity sensor
e. Batter charging and discharging module 5V
f. Moisture and flow sensor added later

Control Flow

The control logic for managing the irrigation system is implemented using an ESP32 microcontroller, which monitors temperature and humidity levels and operates three valves and a motor based on predefined conditions. The main components of the control logic are outlined below:

Valve 1 Control (Fogging)

- Valve 1 is responsible for irrigation based on temperature and humidity conditions.
- If the temperature exceeds 40°C or the humidity drops below 60%, Valve 1 is activated to start irrigation.
- Simultaneously, the motor is turned on to ensure water flow through Valve 1.
- Once the temperature falls below 40°C and humidity rises above 60%, Valve 1 is deactivated.

Valve 2 and Valve 3 Control (Water and Fertilizer)

- Valve 2 is used for water supply and is scheduled to operate daily at 4:00 PM for five minutes.
- On specific days (Monday, Wednesday, and Friday), Valve 3 is also activated during this time to supply fertilizer alongside the water.
- The control logic checks the current day and time, and if it matches the conditions, it opens Valve 2. On the designated days, Valve 3 is simultaneously opened for fertilizer supply.

Motor Control

- The motor is controlled to ensure it operates only when water or fertilizer is being supplied.
- If any of the valves (Valve 1, Valve 2, or Valve 3) are open, the motor is turned on.
- If all valves are closed, the motor is turned off to conserve energy and prevent unnecessary operation.

Day Check for Valve 3

- A helper function (`isValve3Day`) checks whether the current day is Monday, Wednesday, or Friday.
- This function iterates through an array of designated days and returns true if the current day matches any of these days, enabling the logic to control Valve 3 accordingly.

By implementing this control logic, the system efficiently manages irrigation and fertilizer supply, ensuring optimal plant growth conditions while conserving resources. The automated operation based on environmental conditions and predefined schedules provides a robust solution for agricultural automation.

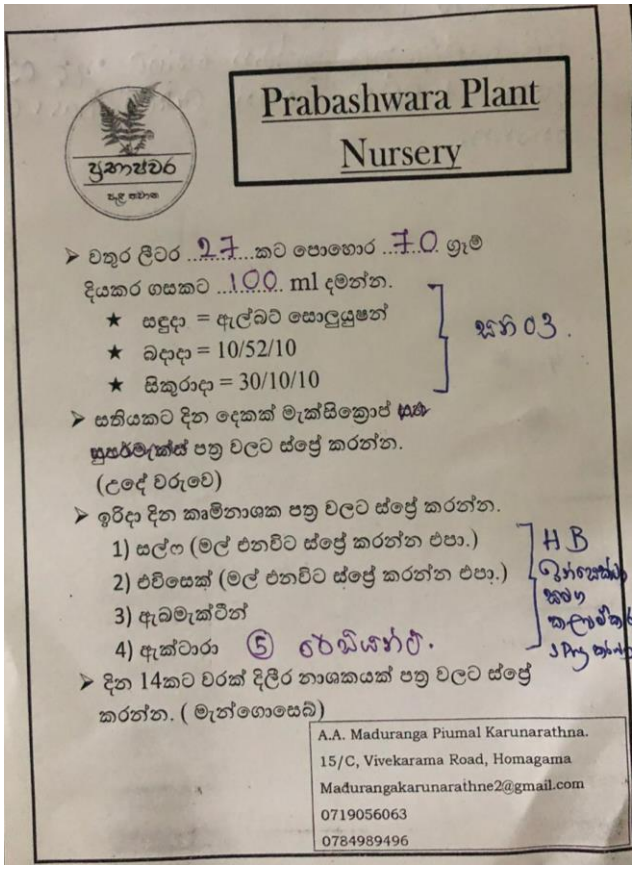

FIGURE 2: PLANT NURSERY

- LED Dash Board in controller

Red LED		Power ON
RGB LED	White	Valve 1 open (for smog)
	Blue	Valve 2 open (for water)
	Green	Valve 3 open (for fertilizer)
Orange LED		Motor ON

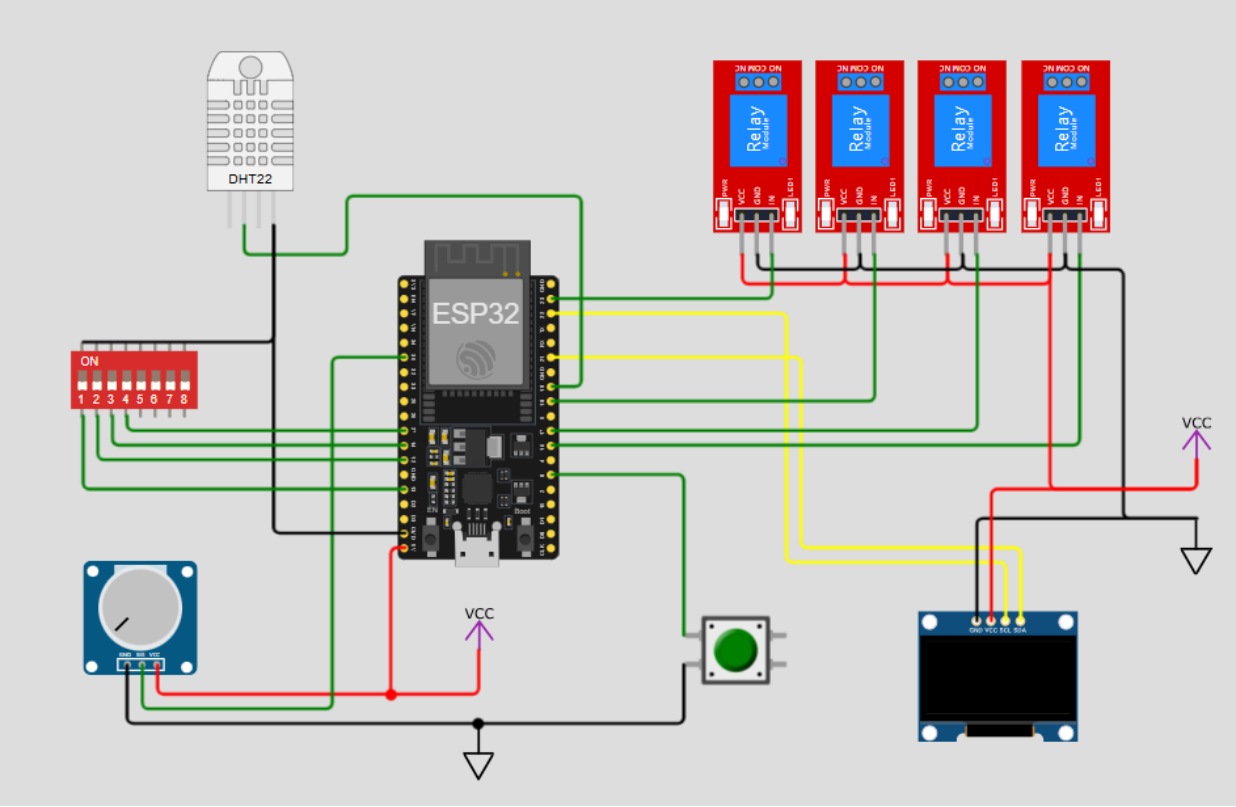

Figure 4: Controller Electrical Connection

Developing stages

1st Version

Write the control flow (**Automated Program**) only in the ESP 32 device.

2nd Version

Gave the **notification for WhatsApp**

Use the Twilio *WhatsApp API* and Thing ESP *operator*

3rd Version

Local Hosted Dashboard development using the Node-Red GUI and MQQT server

Esp device and we want another local host computer or Raspberry Pi board

4th Version

Rainmaker App (User interface can automation and scheduling)

ESP RainMaker is a platform from Espressif Systems that streamlines the process of building and managing Internet of Things (IoT) devices, particularly those using Espressif's ESP32-S2 SoC. It achieves this by providing a comprehensive suite of tools that handle various aspects of IoT development

- Device Programming:

  ESP RainMaker offers a Device SDK for your ESP32 device. This SDK simplifies communication with the RainMaker cloud and allows you to define how your device interacts with sensors, actuators, and other peripherals.

- Mobile App Configuration (Self-Adapting)
- Transparent Cloud Middleware
- Easy Device Provisioning

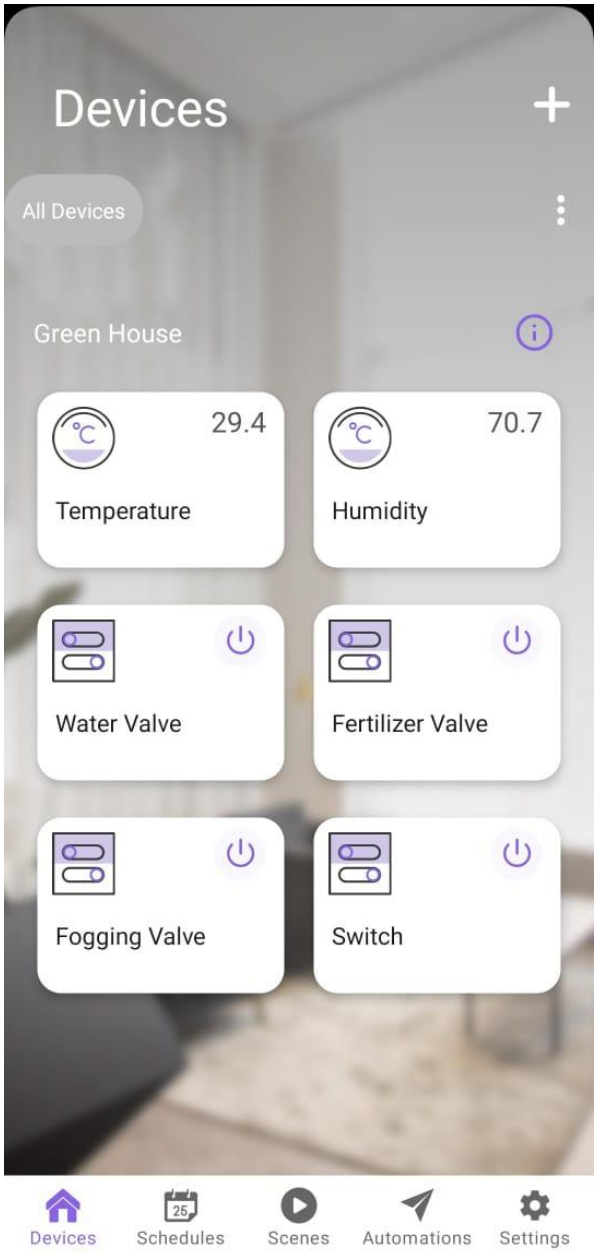

Figure 5: Mobile APP Interface

Here's a breakdown of the typical workflow

1. Device Development

   Program your ESP32 device using the RainMaker SDK, defining functionalities and sensor data to be exposed.

2. Flashing the Firmware

   The compiled code is uploaded (flashed) onto your ESP32 device.

3. Mobile App Setup

   The user downloads the ESP RainMaker app and creates an account.

4. Device Pairing

   The user opens the app and initiates the pairing process by scanning a QR code displayed on the device's serial monitor.

5. Wi-Fi Provisioning

   The app uses BLE to transfer Wi-Fi credentials to the device securely.

6. Cloud Connection

   The device connects to the RainMaker cloud using the provided credentials.

7. Remote Control and Monitoring

The user can now interact with your device's features and view sensor data through the intuitive interface in the RainMaker app.

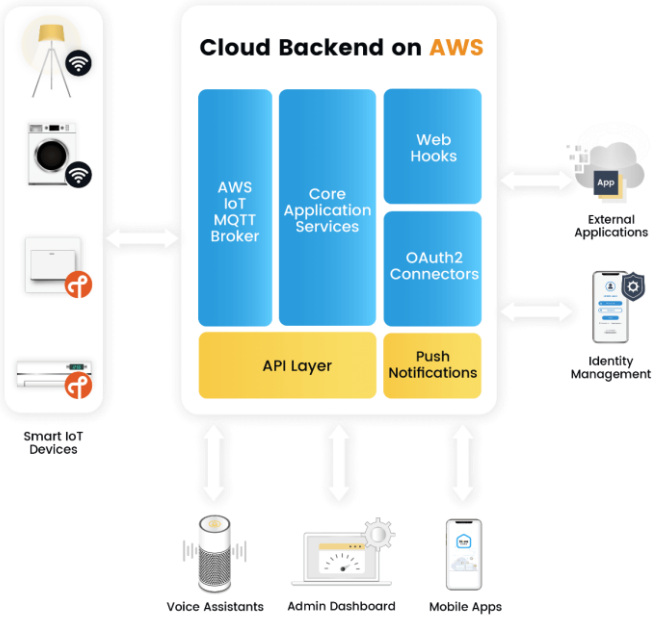

Figure 6: Architecture of the RainMaker

BOM

	Quantity	*Cost (Rs)*
Esp32 Microcontroller	*1*	*1790*
Four channel relays	*1*	*890*
Relay module	*1*	*220*
On off switch	*5*	*60 × 5 = 300*
Jumper wire set	*1*	*160*
230V to 5V converter	*1*	*400*
Box	*1*	*200*
Plug wire	*1*	*200*
Connector Bar	*1*	*220*
Total controller only		***4380***
Solenoid Valve	*3*	*1290 × 3 = 3870*
Total		***8250***

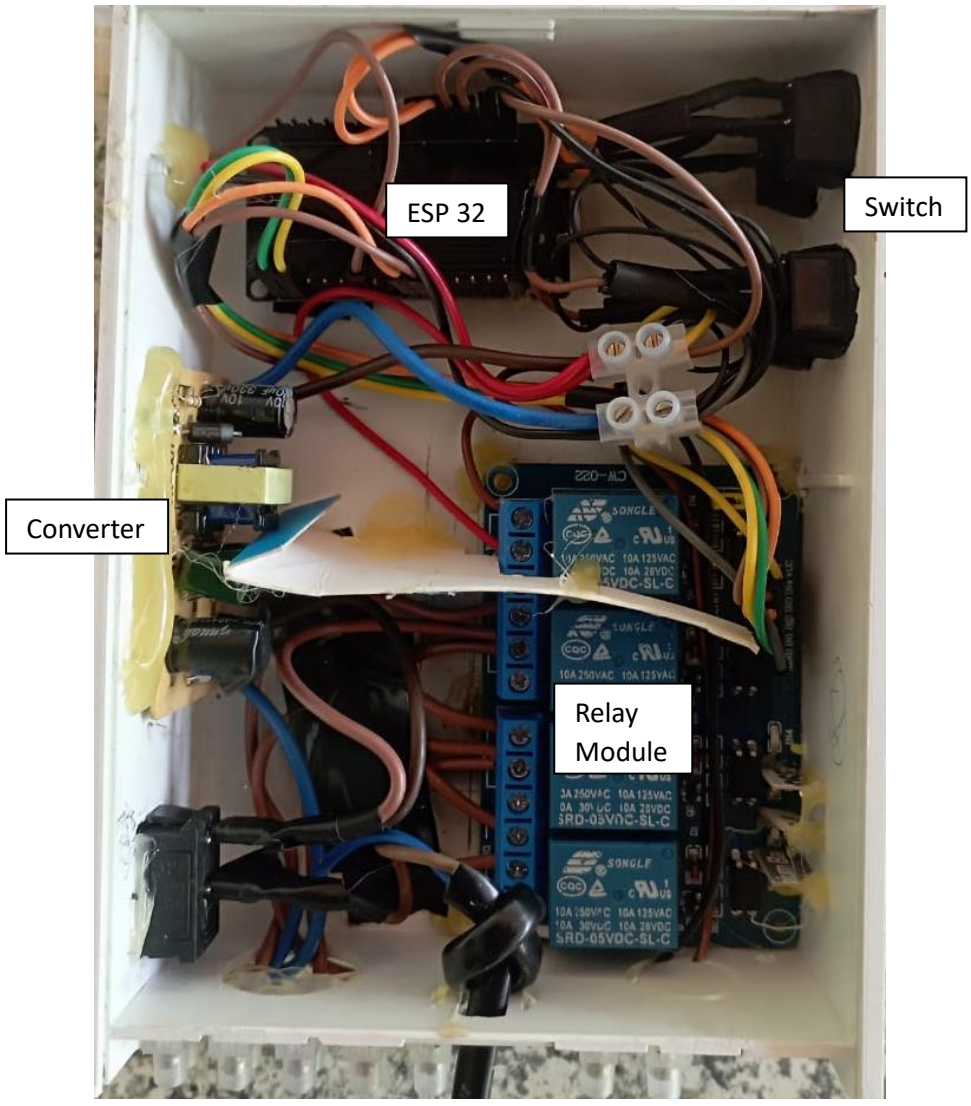

Figure 7: Controller Hardware Design

Future upgrade functions

a. Water flow control using a moisture sensor
   (problem arises: it is difficult to use for every plant)
b. Adding the moisture sensor, ph sensor and flow sensor
c. Using flow sensor automate the fertilizer quantity
d. Add the AI features to identify the behaviours of plants
e. Add the display for the controller