International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

IoT-Based Real-Time Rainfall Quantification and Automated Response System using ESP32 and MQTT Protocol

[1]Palak Shrivastava, [2]Kartik Krishna, [3]Saee Kshirsagar, [4]Prajakta Umrikar
[1/2/3/4]Dept. of ECE, LNCT Group of College, Bhopal, India
[1]palak3261@gmail.com, [2]kartikkrishna321@gmail.com, [3]saee0419@gmail.com,
[4]sumrikar080@gmail.com

ABSTRACT
Communication between environmental sensors and automated response systems is critical for precision agriculture and smart home infrastructure. This paper presents "RAINSENSE," an IoT-based environmental monitoring system designed to detect rainfall onset and quantify intensity in real-time. The system leverages the ESP32 microcontroller and the MQTT (Message Queuing Telemetry Transport) protocol to ensure low-latency data transmission to a cloud-based dashboard. A resistive rain-sensing module provides dual-output signals (analog and digital), allowing the system to categorize precipitation into light, moderate, and heavy rainfall. To translate this data into physical action, the system integrates servo motors for automated cover deployment (e.g., closing windows or roofs) and relays for triggering external alarms or pumps. Furthermore, a software-driven power-gating mechanism is implemented to prevent the electrochemical corrosion of the sensor pad, significantly increasing device longevity. The results demonstrate a responsive, scalable, and cost-effective solution for automated environmental management.

Keywords— ESP32, MQTT, IoT, Rainfall Detection, Servo Motor, Relay, Cloud Dashboard, Power Gating, Smart Agriculture.

1.INTRODUCTION

A. Background

Environmental monitoring has evolved from passive observation to active, real-time automation. In sectors such as agriculture and residential architecture, the ability to react instantly to rainfall is essential. Sudden precipitation can lead to crop spoilage or interior water damage. While professional weather stations provide high-accuracy data, they are often too expensive and bulky for localized, rapid-response automation.

B. Problem Statement

Most low-cost rain sensors are binary (Rain/No Rain), failing to provide the intensity data needed for nuanced responses. Moreover, the use of DC current on exposed copper sensing pads leads to rapid electrolysis and corrosion, causing sensor failure. Additionally, traditional Wi-Fi-based HTTP requests can be slow.

C. Objectives

The primary objective of this research is to design an end to- end IoT system that:

1. Quantifies rainfall intensity using 12-bit ADC resolution.
2. Implements a low-latency communication pipeline via the MQTT protocol.
3. Automates physical responses using servo motors and relays.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Extends sensor lifespan through software-controlled power gating.

2. SYSTEM DESIGN AND WORKING

A. Working Principle

The RAINSENSE system operates on the principle of electrical conductivity. The sensing pad consists of interleaved copper traces. In a dry state, the resistance is near-infinite. As raindrops land on the pad, they create a conductive path, reducing the resistance. The ESP32 microcontroller samples this resistance. If the value falls below a predefined threshold, the system triggers a "Rain Event."

B. System Architecture

The system is divided into three functional layers:

1. Perception Layer:

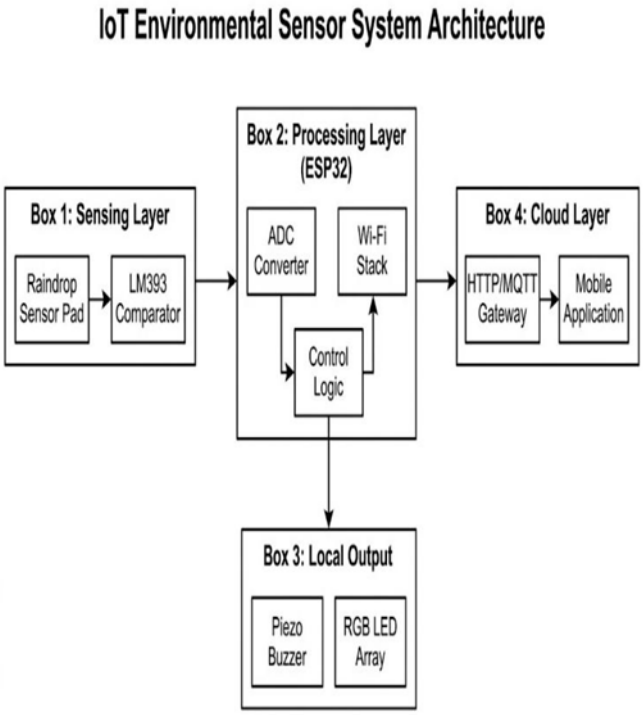
Rain sensor pad →→ LM393 Comparator.

2.Control Layer:

ESP32 →→MQTT Client →→ Power Gating Logic.

3.Actuation Layer:

Servo Motors (Position Control) →→ Relays (Switching) →→Cloud Dashboard.



[Functional Block Diagram of RAINSENSE]

3. HARDWARE COMPONENTS

A. ESP32 Microcontroller

The ESP32 was selected due to its dual-core processor and integrated Wi-Fi/Bluetooth capabilities. Its high-speed ADC is essential for quantifying rain intensity.

B. Raindrop Sensor Module

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

The module utilizes a nickel-plated pad and an LM393 comparator. It provides a Digital Output (DO) for instant triggers and an Analog Output (AO) for intensity measurement.

C. Servo Motors

High-torque servo motors are used to actuate physical barriers. Upon detecting heavy rain, the ESP32 sends a Pulse Width Modulation (PWM) signal to rotate the servo from 0 to 180, effectively closing a protective cover.

D. Relay Module

A 5V electromagnetic relay is used as a switch for high-power devices. In this system, the relay triggers a buzzer for
local alarms or a water pump for drainage management.

E. Cloud Dashboard & MQTT Broker

Instead of standard HTTP, this system uses MQTT. A broker (such as Mosquitto or Adafruit IO) manages the "Publish/Subscribe" mechanism. The ESP32 publishes rain data, and the Dashboard subscribes to it, ensuring near-instantaneous updates.

IOT WEATHER STATION HARDWARE PIN MAPPING TABLE

ESP32 Pin	Interface	Function
GPIO 27 (Analog ADC1_6) →	Analog/Digital Input (Sensor Power Control)	Reads analog value from Raindrop Sensor AO
GPIO 35 (Digital Input) →	Digital Input	Reads digital threshold from Raindrop Sensor DO
GPIO 25 (Digital Output/DAC1) →	Digital Output	Activates Piezo Buzzer
GPIO 26 (Digital Output) →	Digital Output	Controls Red LED (Warning)
GPIO 27 (Digital Output) →	Digital Output	Controls Yellow LED (Alert)
GPIO 14 (Digital Output) →	Digital Output	Controls Green LED (Normal)
ESP32 Wi-Fi Stack →	Internal (Software Stack)	Cloud Communication via HTTP/MQTT Gateway

[Hardware Pin Mapping Table]

4. SOFTWARE IMPLEMENTATION

A. MQTT Protocol Integration

MQTT is a lightweight messaging protocol ideal for IoT. The ESP32 connects to the broker and publishes data to a specific topic (e.g., home/garden/rain). This reduces overhead and power consumption compared to traditional API calls.

B. Anti-Corrosion Power Gating

To prevent electrolysis, the sensor is not powered continuously. The software implements the following logic:

1. Set GPIO 27 to HIGH (Sensor Power ON).

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

1. Delay 10ms for signal stabilization.
2. Read Analog Value →→Read Digital Value.
3. Set GPIO 27 to LOW (Sensor Power OFF).
4. Enter Sleep Mode for 5 seconds.

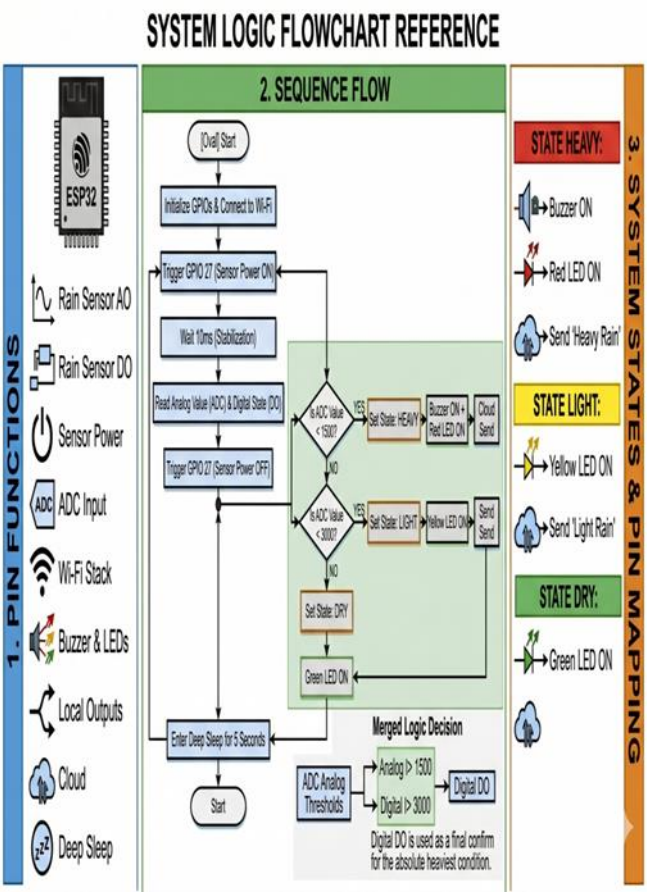
C. Automation Logic Flow

The system follows a threshold-based decision tree:

Dry (ADC>3000ADC>3000): Servo at 0∘, Relay OFF, LED Green.

Light Rain (1500<ADC≤30001500<ADC≤3000): Servo at 45∘, Relay OFF, LED Yellow, Cloud Notification sent.

Heavy Rain (ADC≤1500ADC≤1500): Servo at 180∘, Relay ON (Buzzer), LED Red, Critical Cloud Alert sent.



[System Logic Flowchart]

5. METHODOLOGY

A. Data Acquisition Stage

The process begins with the periodic activation of the sensing pad. The raw analog voltage is converted by the ESP32's 12-bit ADC into a value between 0 and 4095. To remove environmental noise, a Moving Average Filter is applied to five consecutive readings.

B. Cloud Communication Stage

Once the processed value is determined, the ESP32 packages the data into a JSON string. This packet is published to the MQTT broker. The cloud dashboard, acting as a subscriber, updates the gauges and switches in real-time.

International Journal of Research and Technology (IJRT)

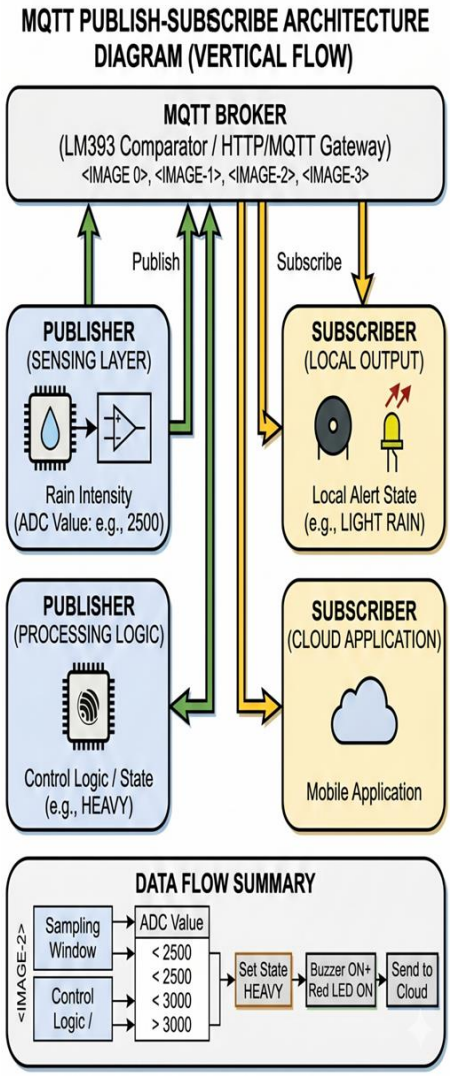
International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

C. Actuation Stage

The control signal is then dispatched to the hardware. The servo motor moves to the calculated angle to protect the assets, and the relay closes the circuit for the alarm system.



[MQTT Publish-Subscribe Architecture Diagram]

6. RESULTS AND ANALYSIS

A. Rainfall Intensity Calibration

The system was tested using various water densities to calibrate the ADC thresholds.

Rain Condition	Avg. ADC Value	Action Triggered	Dashboard Status
Dry	4095	No Action	"Clear"
Light	2400	Partial Servo Move	"Drizzle"

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

Moderate	1800	Servo 90∘	"Raining"
Heavy	600	Servo 180 + Relay	"Heavy Rain"

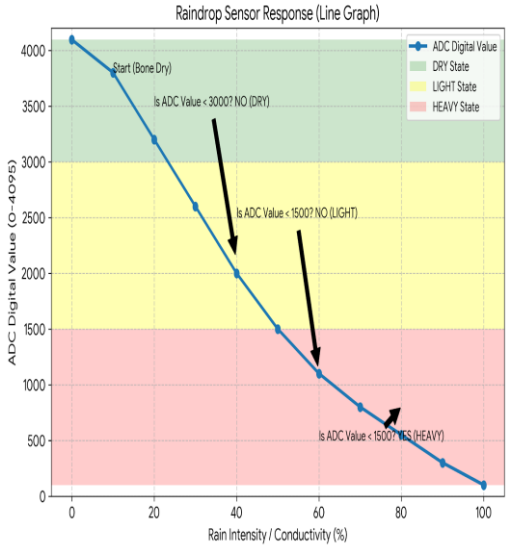
Table 1: Calibration of Rain Intensity

B. Latency Analysis

The response time was measured from the moment of rain detection to the completion of the servo movement and cloud update.

Trial	**Sensor Detection (ms)**	**MQTT Transmission (ms)**	**Servo Actuation (ms)**	**Total (s)**
1.	10	320	400	0.73
2.	12	350	410	0.77
3.	11	310	390	0.71
Avg	11	326	400	0.74

Table 2: Latency Measurements



[Line Graph of ADC Value vs. Rain Intensity]

C. Corrosion Mitigation Results

A comparative study was conducted between a standard "Always-On" sensor and the RAINSENSE "Power-Gated" sensor over a period of 10 days of simulated rain.

Sensor Design	**Day 1 Resistance**	**Day 10 Resistance**	**Drift %**
Standard (DC)	10MΩ	3.5MΩ	65%
RAINSENSE	10MΩ	9.4MΩ	6%

Table 3: Baseline Resistance Drift

The results indicate that power-gating reduces hardware degradation by over 90%.

7. CHALLENGES AND LIMITATIONS

A. Sensitivity to Humidity

In conditions of extremely high humidity (above 90%), the sensor occasionally recorded a "Light Rain" state due to condensation. This was mitigated by increasing the analog threshold.

B. Power Consumption

Although MQTT is efficient, the Wi-Fi radio of the ESP32 consumes significant current. For fully autonomous deployment, a Li-Po battery with a solar charging circuit is required.

C. Mechanical Wear

Continuous movement of the servo motor can lead to mechanical wear over time. Implementing a "hysteresis" logic (waiting for a sustained change in rain value) prevents the motor from oscillating during light drizzle.

8. CONCLUSION

The RAINSENSE system successfully integrates IoT sensing, cloud communication via MQTT, and physical actuation. By combining a rain-sensing module with an ESP32, the system provides a real-time, granular measurement of rainfall intensity. The addition of servo motors and relays transforms the device from a simple monitor into an automated response system capable of protecting assets without human intervention.

The implementation of power-gating logic solves the critical industry problem of sensor corrosion, ensuring that the device remains operational for extended periods. With an average response time of 0.74 seconds, the system is highly efficient for real-time applications in smart agriculture and home automation.

9. FUTURE SCOPE

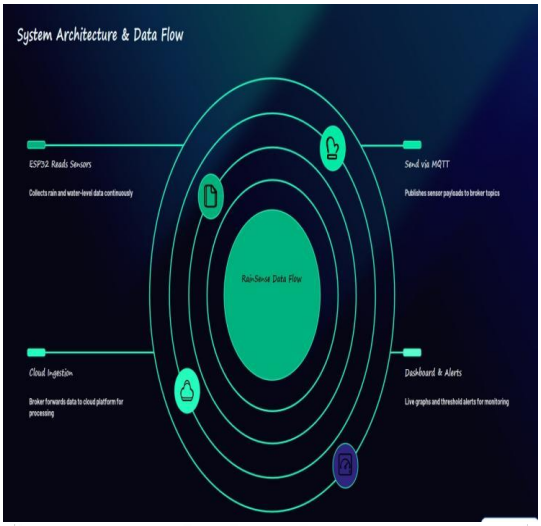
The system can be further enhanced through the following developments:

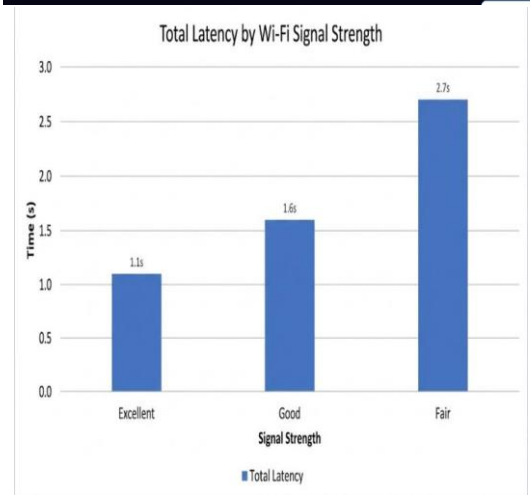
1.Machine Learning Integration: Implementing a LSTM (Long Short-Term Memory) network to predict rainfall based on historical data and humidity trends.

2.Mesh Networking: Deploying multiple RAINSENSE nodes across a large-scale farm using the ESP-NOW protocol for decentralized monitoring.

3.Dynamic Actuation: Integrating a motorized window system that adjusts the angle of closure based on wind speed and rain intensity.

4.Energy Harvesting: Implementing a Piezo-electric energy harvester that generates power from the impact of raindrops.





REFERENCES

1. Espressif Systems, "ESP32 Technical Reference Manual," [Online].
2. Texas Instruments, "LM393 Low-Power Voltage Comparator Datasheet."
3. IEEE Xplore, "IoT-Based Smart Irrigation and Weather Monitoring Systems," 2022.
4. Lakshmi Narain College Of Technology Library, "Embedded Systems and Sensor Fusion Guidelines."
5. MQTT.org, "MQTT Version 3.1.1 Specification," [Online].