



**UNIVERSITY MALAYSIA TERENGGANU
FACULTY OF OCEAN ENGINEERING TECHNOLOGY &
INFORMATICS**

**[CSM3313]
INTERNET OF THINGS**

GROUP PROJECT REPORT

Prepared by:
[GROUP 10]

Prepared for:
DR AHMAD SHUKRI BIN MOHD NOOR

[*MOBILE COMPUTING*]
SEMESTER II 2022/2023

Group Members

No	Matric No	Name	Role
1	S63190	WAN ABDUL AZIM BIN WAN MALEK	Group leader
2	S62993	MUHAMMAD AFIQ HANIF BIN SUHAIMI	Supplier
3	S62734	MUHAMMAD AKMAL BIN ZAINAL ABDIN	Researcher
4	S62113	MUHAMMAD IRFAN HAKIM BIN SAHARUL REDZUAN	Programmer

Table Of Contents

Executive Summary	4
Introduction.....	6
Problem Statement	7
Objectives	8
Proposed Solution	9
Methodology/Flow Chart Result and Discussion Conclusion	10
Appendix 1: Source Code	11
Appendix 2: Screenshots of output	12
Appendix 3: PowerPoint presentation Slide	13
References.....	14

Executive Summary

The executive summary provides an overview of the key features and benefits of the ESP32 Raindrop Sensor, a device designed to detect and measure rainfall levels.

The ESP32 Raindrop Sensor is a state-of-the-art sensor that utilizes advanced technology to accurately measure rainfall intensity and provide real-time data. It offers several significant advantages over traditional rain gauges, making it a valuable tool for various applications, including weather monitoring, agriculture, and environmental research.

Key Features:

1. **High Precision:** The ESP32 Raindrop Sensor employs precise measurement techniques to deliver accurate and reliable rainfall data, enabling users to make informed decisions based on real-time information.
2. **Wireless Connectivity:** The sensor is equipped with built-in ESP32 wireless connectivity, allowing seamless integration with other devices or systems for data collection and analysis. This feature enables remote monitoring and eliminates the need for physical data retrieval.
3. **Easy Installation and Setup:** The sensor's compact and user-friendly design simplifies installation and setup processes. It can be easily deployed in various environments, including gardens, farms, and meteorological stations, without requiring extensive technical expertise.
4. **Real-Time Data Monitoring:** The ESP32 Raindrop Sensor provides continuous real-time monitoring of rainfall levels, allowing users to monitor rainfall patterns, detect anomalies, and make timely decisions in response to changing weather conditions.

Benefits:

1. **Improved Accuracy:** The sensor's advanced technology ensures highly accurate rainfall measurements, minimizing errors and providing reliable data for better decision-making.
2. **Cost-Effective Solution:** The ESP32 Raindrop Sensor offers a cost-effective alternative to traditional rain gauges, eliminating the need for manual data collection and reducing maintenance requirements.
3. **Versatile Applications:** The sensor's compatibility with various systems and its wireless connectivity open up a wide range of applications, including weather forecasting, irrigation management, and flood detection.
4. **Data Integration and Analysis:** The wireless connectivity of the sensor facilitates easy integration with data management systems, enabling comprehensive data analysis, visualization, and integration with other environmental variables.

In summary, the ESP32 Raindrop Sensor is a cutting-edge device that provides accurate rainfall measurement, wireless connectivity, and real-time monitoring capabilities. With its ease of installation, cost-effectiveness, and versatile applications, the sensor offers an invaluable solution for individuals, businesses, and organizations involved in weather monitoring, agriculture, and environmental research.

Introduction

The ESP32 Raindrop Sensor is a specialized device that is designed to accurately detect and measure rainfall levels. It utilizes advanced technology and wireless connectivity to provide real-time data on the intensity of rainfall. The sensor incorporates an ESP32 microcontroller, which enables precise measurements and seamless integration with other devices or systems for data collection and analysis.

The ESP32 Raindrop Sensor offers several advantages over traditional rain gauges. It eliminates the need for manual data collection by enabling wireless connectivity, allowing for remote monitoring and reducing the need for physical retrieval of data. The sensor's accuracy ensures reliable measurements, minimizing errors and providing trustworthy data for decision-making purposes.

With its compact and user-friendly design, the ESP32 Raindrop Sensor is easy to install and set up in various environments such as gardens, farms, or meteorological stations. It is a cost-effective solution, reducing expenses associated with manual data collection and maintenance.

By providing continuous real-time monitoring of rainfall patterns, the ESP32 Raindrop Sensor enables users to stay informed about changing weather conditions. This information allows for timely decision-making in areas such as agriculture, irrigation management, and flood detection.

Furthermore, the ESP32 Raindrop Sensor is compatible with different systems and can integrate with data management platforms. This feature enables comprehensive data analysis, visualization, and correlation with other environmental variables, leading to a deeper understanding of rainfall patterns and their impact.

Overall, the ESP32 Raindrop Sensor is a cutting-edge device that combines accuracy, wireless connectivity, and ease of use. Its capabilities make it a valuable tool for various applications in weather monitoring, agriculture, and environmental research.

Problem Statement

1. **Manual Data Collection:** Traditional rain gauges require manual collection of rainfall data, which is time-consuming, labor-intensive, and prone to errors or delays.
2. **Limited Accuracy:** Traditional rain gauges may have limitations in accuracy, leading to unreliable measurements and hindering precise rainfall monitoring.
3. **Maintenance Requirements:** Traditional rain gauges often require regular cleaning, calibration, and physical retrieval of data, resulting in additional costs and inconvenience.
4. **Lack of Real-Time Monitoring:** The absence of wireless connectivity in traditional rain gauges hampers real-time monitoring and timely decision-making.
5. **Integration Challenges:** Traditional rain gauges may lack compatibility with other systems or platforms, making it difficult to integrate rainfall data with broader environmental monitoring or data analysis.

The ESP32 Raindrop Sensor addresses these problems by offering accurate measurements, real-time data, wireless connectivity, ease of installation, and compatibility with data management systems.

Objectives

1. **Accurate Rainfall Measurement:** The primary objective of the ESP32 Raindrop Sensor is to provide highly accurate measurements of rainfall intensity. It aims to ensure reliable data collection, minimizing errors and providing precise information for decision-making purposes.
2. **Real-Time Monitoring:** The sensor aims to enable real-time monitoring of rainfall patterns. By continuously updating the rainfall data, it allows users to stay informed about changing weather conditions and make timely decisions in response to rainfall events.
3. **Wireless Connectivity:** The ESP32 Raindrop Sensor aims to provide wireless connectivity capabilities. This objective allows for seamless integration with other devices or systems, enabling remote monitoring, data transmission, and eliminating the need for physical data retrieval.
4. **Ease of Installation and Use:** The sensor is designed with the objective of being easy to install and user-friendly. It should have a straightforward setup process, making it accessible to users with varying technical expertise.
5. **Cost-Effectiveness:** The ESP32 Raindrop Sensor aims to offer a cost-effective solution for rainfall measurement. By reducing the need for manual data collection, minimizing maintenance requirements, and providing long-term reliability, it helps reduce expenses associated with traditional rain gauges.
6. **Integration with Data Management Systems:** Another objective of the sensor is to facilitate integration with data management systems or platforms. This feature allows for comprehensive data analysis, visualization, and correlation with other environmental variables, enhancing the understanding of rainfall patterns and their impacts.

Proposed Solution

1. Implement an ESP32-based raindrop sensor that can automatically collect and transmit rainfall data wirelessly. The raindrop sensor should be capable of accurately measuring rainfall intensity and duration, eliminating the need for manual data collection.
2. Design the raindrop sensor with high sensitivity and precision to ensure accurate measurements. Implement advanced algorithms for signal processing techniques to filter out noise and minimize measurement errors, thereby improving the accuracy of rainfall monitoring.
3. Develop a self-cleaning mechanism for the raindrop sensor to minimize the accumulation of debris or contaminants. Implement an automated calibration feature to periodically verify and adjust the sensor's accuracy, reducing the need for manual calibration. Additionally, design the sensor to be durable and weather-resistant, minimizing the risk of damage and reducing maintenance requirements.
4. Integrate the ESP32 raindrop sensor with wireless connectivity capabilities, such as Wi-Fi or Bluetooth. Enable real-time data transmission to a cloud-based platform or a local server, allowing users to access and monitor rainfall data in real-time. Implement alerts or notifications to notify users of significant rainfall events or thresholds.
5. Design the ESP32 raindrop sensor with compatibility in mind, allowing seamless integration with other systems or platforms. Provide standardized data formats or APIs to facilitate easy integration with environmental monitoring systems or data analysis platforms. Consider popular protocols such as MQTT or RESTful APIs for data communication, ensuring compatibility with a wide range of applications and platforms.

Methodology/Flow Chart Result and Discussion Conclusion

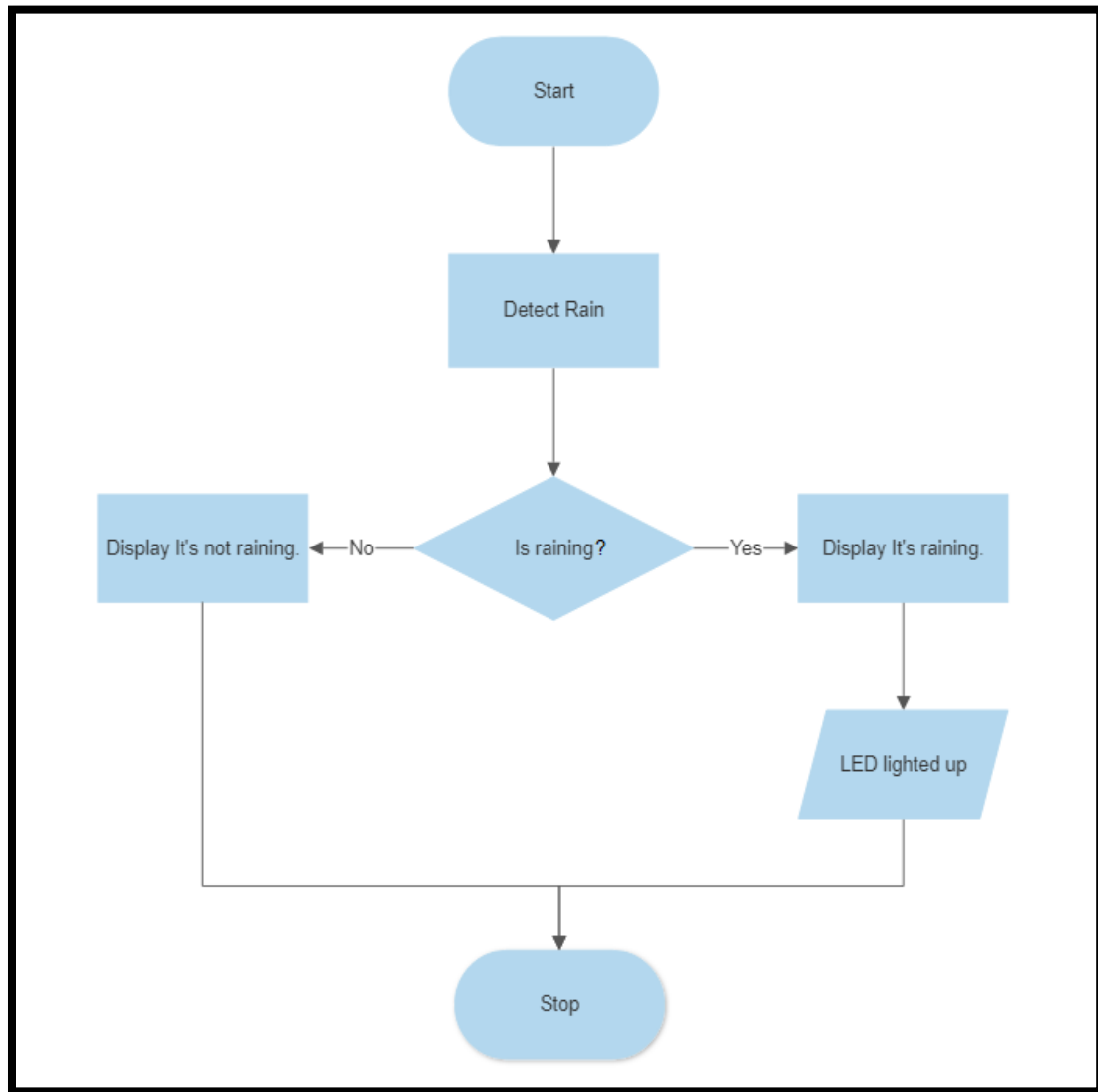


Figure 1 - Flow Chart Result

This figure shows how the rain sensor function, first the sensor detect the rain and the condition will go is raining or not, if it displays it's raining, the LED will be lighted up, if it displays it's not raining, the LED will not light up, then it will stop.

Appendix 1: Source Code

```
*main.py X
1 import network
2 import time
3 from umqtt.simple import MQTTClient
4 from machine import Pin, ADC
5
6 SERVER = "mqtt.favoniot.com"
7 client = MQTTClient("umqtt_client", SERVER, user="d0V0aeJpNy9ZH5xFyJuFT2qKyowTK8ph", password="d0V0aeJpNy9ZH5xFyJuFT2qKyowTK8ph")
8 wlan = network.WLAN(network.STA_IF)
9 wlan.active(True)
10 wlan.disconnect()
11 #mqtt required info
12
13 rain_sensor_adc_pin = ADC(Pin(32))
14 rain_sensor_digital_pin = Pin(33, Pin.IN)
15 led = Pin(18, Pin.OUT)
16 rainingStatus = ''
17 =if not wlan.isconnected():
18     print('connecting to network ...')
19     wlan.connect('Galaxy Note20 Ultra 5G4fc9','00000000')
20 - while not wlan.isconnected():
21     print("network config:", wlan.ifconfig())
22     time.sleep(2)
23
```

```
=while wlan.isconnected():
    client.connect()
    print("client connected")

    # Read the analog value from the rain sensor ADC pin
    rain_sensor_analog_value = rain_sensor_adc_pin.read()

    # Read the digital value from the rain sensor DO pin
    rain_sensor_digital_value = rain_sensor_digital_pin.value()

    # Check if it is raining based on the sensor values
    = if rain_sensor_analog_value < 1000:
        print("\nRaining analog value: " + str(rain_sensor_analog_value))
        print("Raining digital value: " + str(rain_sensor_digital_value))
        print("It is raining heavily!")
        led.on()
    = else:
        print("\nRaining analog value: " + str(rain_sensor_analog_value))
        print("Raining digital value: " + str(rain_sensor_digital_value))
        print("It is not raining.")
        led.off()
```

```
print("Raining:", rain_sensor_analog_value)
client.connect()
client.publish('d0V0aeJpNy9ZH5xFyJuFT2qKyowTK8ph/v2/streams', '{"device_developer_id": "Arduino@S63190", "data": {"Raining anal
client.disconnect()

time.sleep(5) # Delay for stability
```

Appendix 2: Screenshots of output

```
Raining analog value: 4095
Raining digital value: 1
It is not raining.
Raining: 4095
client connected

Raining analog value: 592
Raining digital value: 0
It is raining heavily!
Raining: 592
client connected

Raining analog value: 405
Raining digital value: 0
It is raining heavily!
Raining: 405
client connected
```

Import Export Show real-time data

The minimum, maximum and average readings of the data from specific devices are now included in the exported file.

Entries 10 Device All Delete All Streams

Search

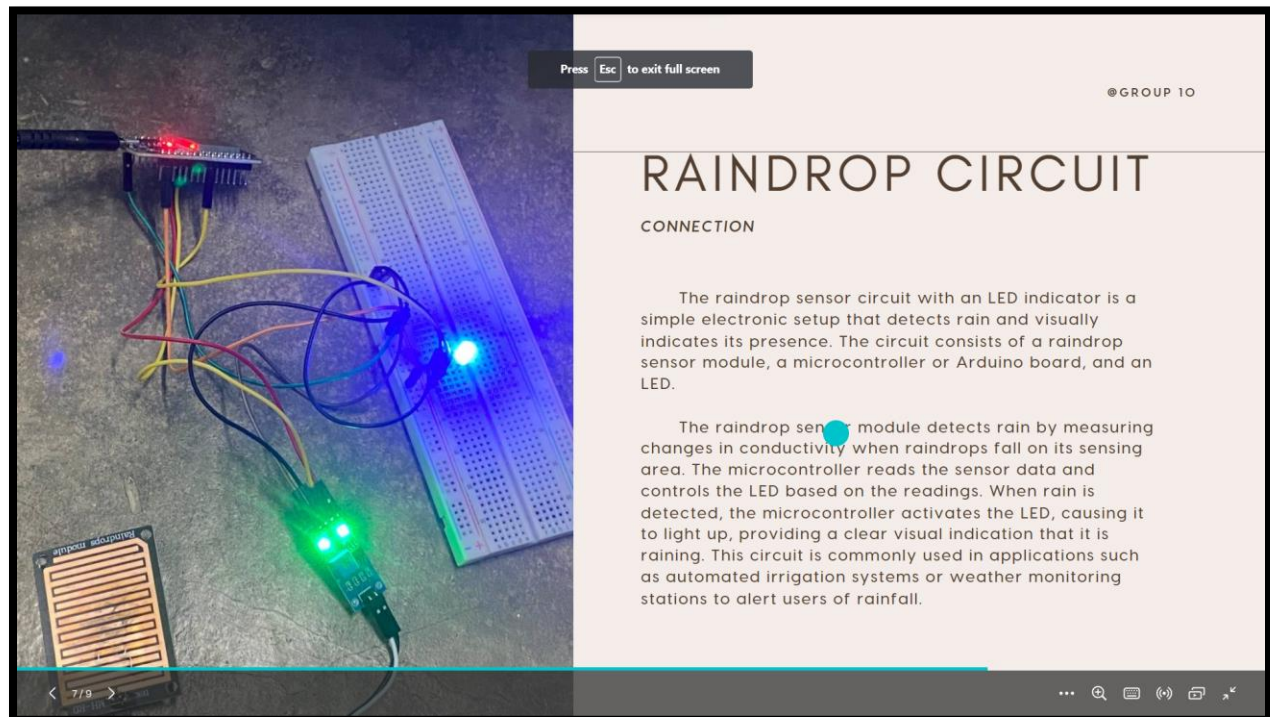
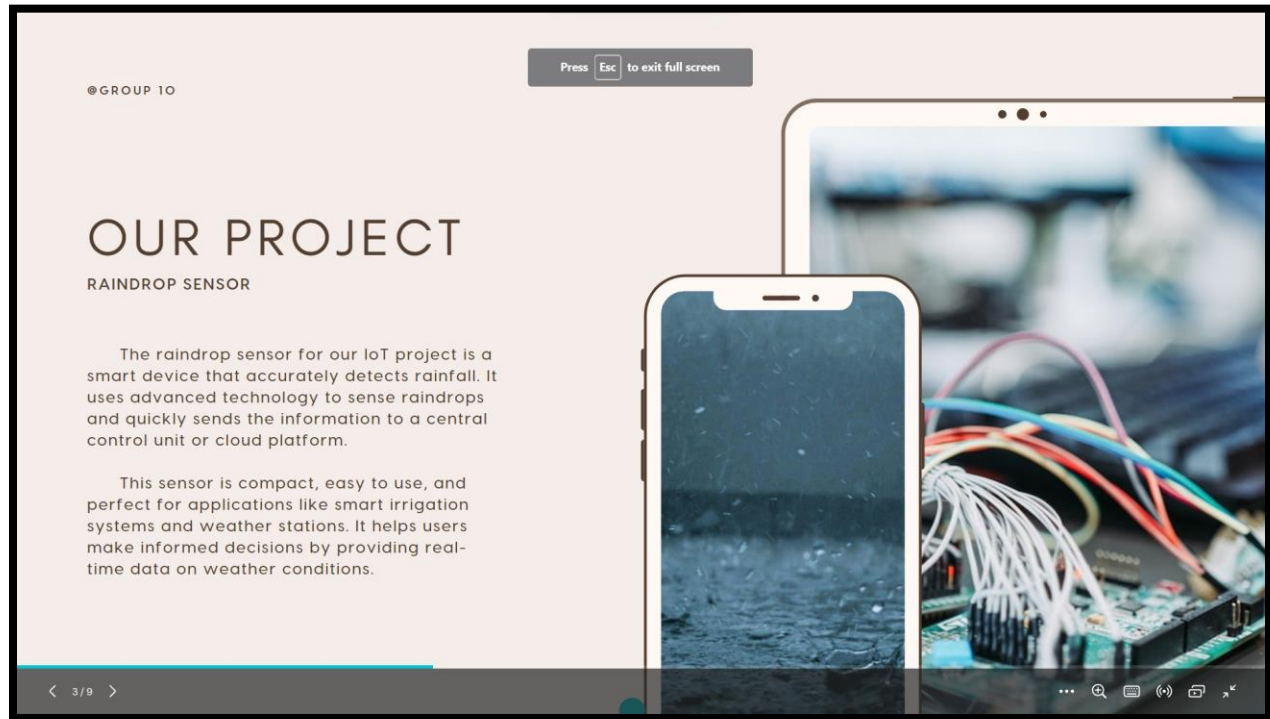
Select	Device Developer ID	Data	Uploaded
<input type="checkbox"/>	Arduino@S63190	"Raining_analog values":"400"	6/23/2023, 4:01:27 AM
<input type="checkbox"/>	Arduino@S63190	"Raining_analog values":"403"	6/23/2023, 4:01:22 AM
<input type="checkbox"/>	Arduino@S63190	"Raining_analog values":"400"	6/23/2023, 4:01:16 AM
<input type="checkbox"/>	Arduino@S63190	"Raining_analog values":"405"	6/23/2023, 4:01:11 AM
<input type="checkbox"/>	Arduino@S63190	"Raining_analog values":"592"	6/23/2023, 4:01:06 AM
<input type="checkbox"/>	Arduino@S63190	"Raining_analog values":"4095"	6/23/2023, 4:01:00 AM
<input type="checkbox"/>	Arduino@S63190	"Raining_analog values":"4095"	6/23/2023, 4:00:55 AM

video demonstration

link : <https://youtu.be/c0JsBz0pR3M>

Appendix 3: PowerPoint presentation Slide

link : https://www.canva.com/design/DAFl_3_z33o/DFqVwg4-h2ihAjWm6OgZeQ/edit?utm_content=DAFl_3_z33o&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton



References

1. Lab, M. (2021). Rain Drop Sensor Module Interfacing with Arduino – Rain Detector Circuit. *Microcontrollers Lab*. <https://microcontrollerslab.com/raindrop-sensor-arduino-detector/>
2. Fahad, E. (2022b). Rain Drop Sensor with Arduino, Rain Detector Arduino code & Circuit. *Electronic Clinic*. <https://www.electronicclinic.com/rain-drop-sensor-with-arduino-rain-detector-arduino-code-circuit/>
3. Akbari, M. (2021). Interfacing Rain Drop Sensor with Arduino. *Electropeak*. <https://electropeak.com/learn/interfacing-rain-drop-sensor-with-arduino/>
4. Engineers, L. M. (2020, December 18). How Rain Sensor Works and Interface it with Arduino. *Last Minute Engineers*. <https://lastminuteengineers.com/rain-sensor-arduino-tutorial/>
5. Santos, R., & Santos, R. (2019). Guide for Rain Sensor FC-37 or YL-83 with Arduino | Random Nerd Tutorials. *Random Nerd Tutorials*. <https://randomnerdtutorials.com/guide-for-rain-sensor-fc-37-or-yl-83-with-arduino/>