AUTOMATED WEATHER MONITORING STATION USING EMBEDDED SYSTEM

MINOR PROJECT-2 REPORT

Submitted by

SURYA.P

MANIKANTA.B

DURGA SRINIVAS PRASAD.S

Under the Guidance of

Dr. M.SATHESH

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS & COMMUNICATION ENGINEERING

Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)

OCTOBER 2024

BONAFIDE CERTIFICATE

Certified that this Minor project-2 report entitled **"AUTOMATED WEATHER MONITORING STATION USING EMBEDDED SYSTEM"** is the bonafide work of **" MANIKANTA.B (21UEEC0033), DURGA SRINIVAS PRASAD.S (21UEEC0298) and SURYA.P (21UEEC0245)"** who carried out the project work under my supervision.

SUPERVISOR	**HEAD OF THE DEPARTMENT**
Dr. M.SATHESH	**Dr.A. SELWIN MICH PRIYADHARSON**
Assistant Professor	Professor
Department of ECE	Department of ECE

Submitted for Minor project-2 work viva-voce examination held on:______________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deepest gratitude to our Respected Founder President and Chancellor **Col. Prof. Dr. R. Rangarajan**, Foundress President **Dr. R. Sagunthala Rangarajan**, Chairperson and Managing Trustee and Vice President.

We are very thankful to our beloved Vice Chancellor **Prof. Dr. Rajat Gupta** for providing us with an environment to complete the work successfully.

We are obligated to our beloved Registrar **Dr. E. Kannan** for providing immense support in all our endeavours. We are thankful to our esteemed Dean Academics **Dr. S . RAJU** for providing a wonderful environment to complete our work successfully.

We are extremely thankful and pay my gratitude to our Dean SoEC **Dr. R. S. Valarmathi** for her valuable guidance and support on completion of this project.

It is a great pleasure for us to acknowledge the assistance and contributions of our Head of the Department **Dr. A. Selwin Mich Priyadharson**, Professor for his useful suggestions, which helped us in completing the work in time .

We are grateful to our supervisor **Dr. M.SATHESH**, Associate Professor ECE for his/her guidance and valuable suggestion to carry out our project work successfully.

We thank our department faculty, supporting staffs and our family and friends for encouraging and supporting us throughout the project.

SURYA.P

MANIKANTA.B

DURGA SRINIVAS PRASAD.S

TABLE OF CONTENTS

ABSTRACT

Weather monitoring is critical for agriculture, environmental research, smart infrastructure, and disaster management. Traditional methods of weather data collection involve manual processes, which are time-consuming, prone to errors, and require continuous human involvement. In response to these limitations, this project proposes the development of an **automated weather monitoring station using embedded systems** to gather and display environmental parameters efficiently. The system utilizes the **ESP8266 microcontroller** to collect data from multiple sensors and transmit it wirelessly to the **Blynk app** for remote monitoring.

Key components include the **DHT11 sensor** for measuring temperature and humidity, a **rain sensor** to detect rainfall levels, and an **LDR sensor** to measure ambient light intensity. The collected data is displayed locally on an **LCD screen with I2C support** for easy visualization, reducing hardware complexity. The use of the ESP8266 enables real-time transmission of sensor data to the Blynk app, allowing users to monitor environmental conditions from anywhere with internet access.

This automated weather station offers a **reliable, cost-effective, and efficient solution** to continuously track environmental conditions without the need for manual intervention. Alerts can be configured in the Blynk app to notify users of critical events, such as sudden rainfall, extreme temperatures, or humidity levels that require immediate attention. These real-time insights support better decision-making in agriculture, disaster preparedness, and resource management. Furthermore, the system's **modular design and IoT integration** make it easy to scale and customize based on specific use cases. The proposed solution is not only practical for agricultural fields but can also be deployed in remote areas, urban infrastructure projects, and research institutions to gather weather data continuously.

By automating the weather monitoring process, this system reduces human error, improves the accuracy of data collection, and ensures **real-time access to critical environmental information**. Its application in agriculture helps farmers plan irrigation schedules, predict weather changes, and manage resources effectively, while urban deployments can assist in traffic management and disaster mitigation efforts. This project demonstrates how **embedded systems combined with IoT technology** can enhance environmental monitoring and promote sustainable practices.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 A COMPREHENSIVE OVERVIEW

Monitoring weather conditions in real-time is crucial for agriculture, environmental management, and public safety. Traditional weather stations can be expensive and complex, limiting their accessibility. With the advancement of IoT and embedded systems, **automated weather monitoring stations** offer a more practical, low-cost, and scalable solution. This project aims to develop a **weather station using an ESP8266 microcontroller** to measure temperature, humidity, light intensity, and rainfall, and display the data both on an LCD screen and remotely via the Blynk app.

1.2 HOW IT WORKS

The weather station integrates several sensors with the ESP8266 microcontroller to collect environmental data and transmit it wirelessly.

DHT11 sensor: Measures temperature and humidity.

Rain sensor: Detects the presence and intensity of rainfall.

LDR sensor: Monitors ambient light intensity.

The data is displayed on an **LCD with I2C support** for easy readability and minimal wiring. Simultaneously, the **ESP8266 transmits data to the Blynk app**, allowing users to monitor weather conditions remotely from their smartphones.

1.3 WHY THIS DEVICE IS REQUIRED

1. Accurate Weather Data: In agriculture, knowing the real-time weather helps farmers optimize irrigation and field activities.

2. Disaster Management: Early detection of changes in rainfall or temperature enables prompt action during emergencies, such as floods or heatwaves.

3. Cost-Effective Monitoring: Traditional weather stations are expensive and bulky, while the proposed system offers an affordable solution.

4. Remote Access: Through the Blynk app, users can monitor environmental data from anywhere, improving responsiveness and planning.

5. Automation and Convenience: Automated alerts for specific weather conditions enhance decision-making and reduce human error.

6. Adaptability to Different Environments: The device can be deployed in various settings, such as farms, urban areas, greenhouses, or remote regions, making it a versatile solution for diverse weather monitoring needs.

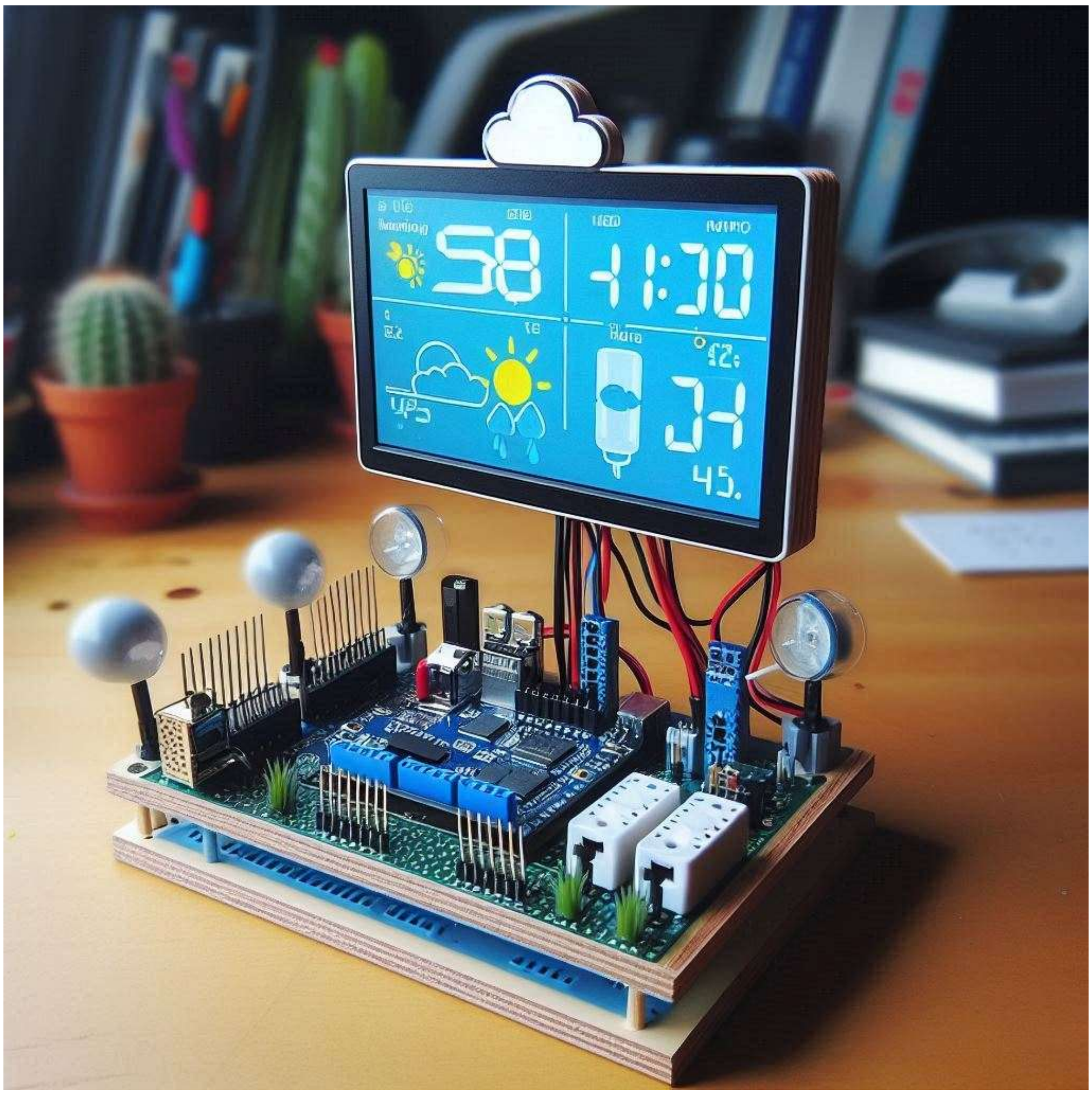


Figure 1.1: DISPLAYING ON LCD

1.4 ATTRIBUTES

The proposed automated weather monitoring system is designed with several key attributes that make it efficient, easy to use, and highly practical for various applications. Below is a detailed breakdown of its core features:

1. Real-Time Data Collection: The ESP8266 microcontroller ensures continuous monitoring of environmental parameters. It collects data in real-time from the DHT11 (temperature and humidity), rain sensor (rain detection), and LDR sensor (light intensity), providing immediate insights for users. Real-time data helps farmers, urban planners, and researchers respond promptly to changing weather conditions.

2. Local and Remote Display of Data: The system displays weather information on an LCD screen with I2C support, making it easy for users to check conditions on-site without needing additional equipment. Simultaneously, the data is transmitted via Wi-Fi to the Blynk app, allowing users to access the information remotely from any smartphone or internet-enabled device. This dual-display feature ensures flexibility, whether on-site or remote.

3. Minimalist and Compact Design: The use of an I2C-compatible LCD minimizes wiring, making the hardware setup simple and clutter-free. The sensors and ESP8266 module are compact, ensuring the system is lightweight and portable. This design makes the station ideal for installation in both rural and urban areas, including greenhouses, farms, or even rooftops.

4. Energy-Efficient Operation: The ESP8266 microcontroller is known for its low-power consumption, ensuring that the system can run continuously for long periods. It can also be powered by batteries or solar panels, enhancing its suitability for remote locations where access to electricity may be limited.

5. Reliable Wireless Communication: With built-in Wi-Fi capability, the ESP8266 ensures smooth communication with the Blynk app, even in areas with limited network coverage. The system sends data packets at regular intervals, ensuring that users always have access to the latest weather information. In case of network failure, the local LCD display continues to show data, ensuring uninterrupted monitoring.

6. Customizable Alerts and Automation: Users can set threshold values for weather parameters, such as temperature, humidity, or rainfall, and receive automated alerts via the Blynk app. This feature helps users respond proactively, such as by closing greenhouse windows during rain or adjusting irrigation schedules based on humidity levels.

Figure 1.2: MONITORING THROUGH MOBILE

7. Scalability and Modularity: The weather station is designed to be easily expandable. Additional sensors, such as wind speed, air quality, or soil moisture sensors, can be integrated without changing the core structure. This flexibility allows the system to adapt to various use cases, such as agriculture, urban monitoring, or environmental research.

8. User-Friendly Interface: The Blynk app provides an intuitive interface, allowing users to view real-time data and customize settings effortlessly. Caregivers and farmers can easily configure notification preferences, set geofencing limits for alerts, and track environmental data trends over time. The app also allows easy calibration of sensors through a user-friendly dashboard.

9. Minimal Maintenance Requirements: The system is designed to operate with minimal maintenance. The sensors are robust and reliable, and the modular setup makes it easy to replace or upgrade individual components as needed. Regular firmware updates can be made wirelessly through the ESP8266, ensuring the system remains up-to-date with the latest features and security protocols.

10. Cost-Effective and Sustainable Solution: Compared to traditional weather stations, this system is more affordable and accessible. Its energy-efficient design reduces operational costs, and it can run on solar power in remote areas. The low-cost sensors and modular approach make it easy to deploy and maintain without requiring specialized equipment or expertise.

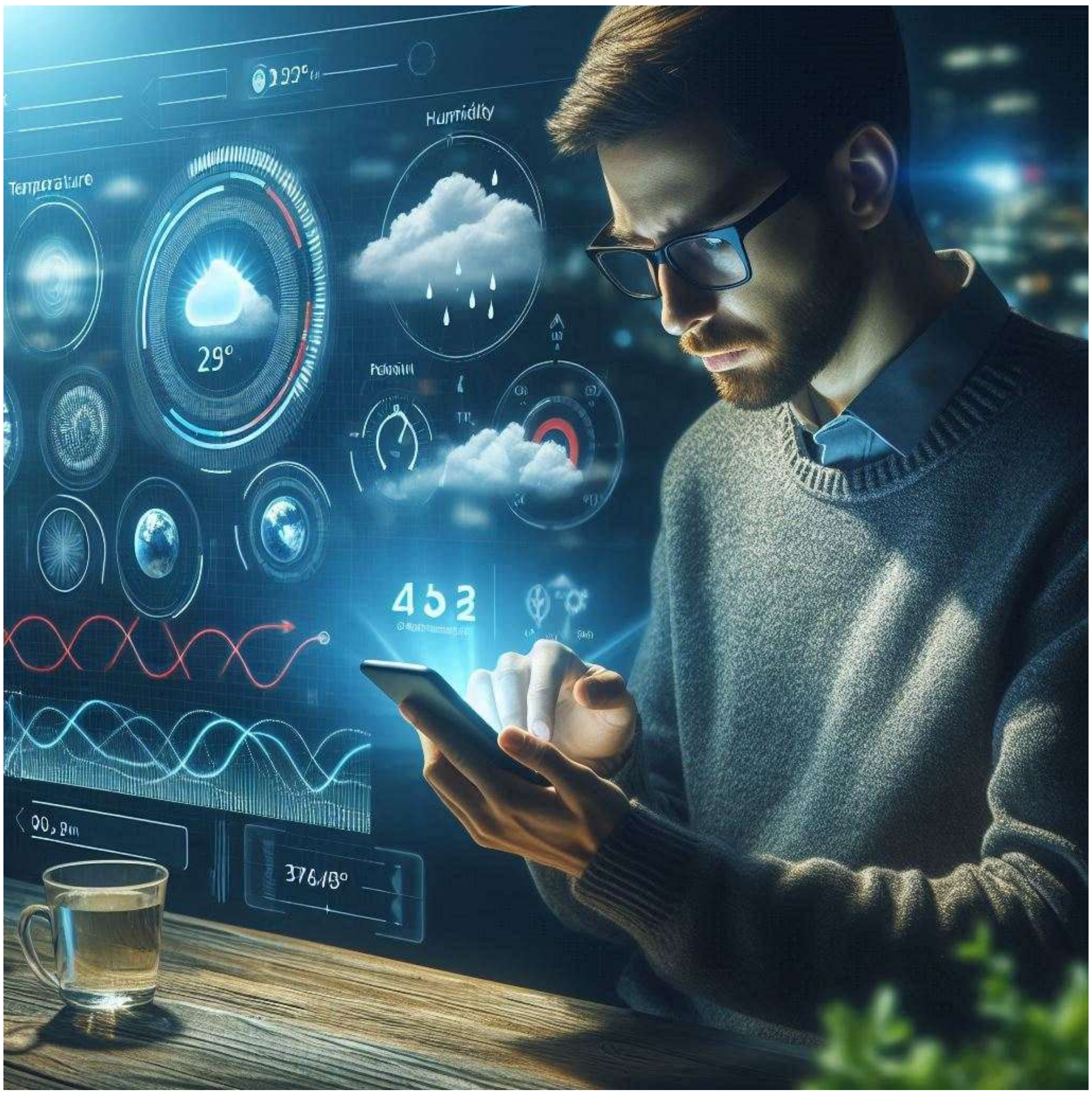


Figure 1.3: WEATHER WATCH

CHAPTER 2

COMPONENTS REQUIRED

2.1 ESP8266

The **ESP8266** is a low-cost Wi-Fi microcontroller module widely used in IoT applications. It features built-in Wi-Fi capability, allowing devices to connect to the internet for real-time data transmission. With its compact size and low power consumption, it's ideal for smart devices, home automation, and remote monitoring systems. The module supports both **client and server modes**, enabling it to send or receive data from web servers and cloud platforms like Blynk. It offers **GPIO pins** for interfacing with sensors and other components. The ESP8266 can be programmed using **Arduino IDE** or other platforms, making it beginner-friendly. Its versatility and affordability have made it popular in DIY projects and commercial IoT solutions. Additionally, the chip supports Over-the-Air (OTA) firmware updates, ensuring easy maintenance and upgrades.

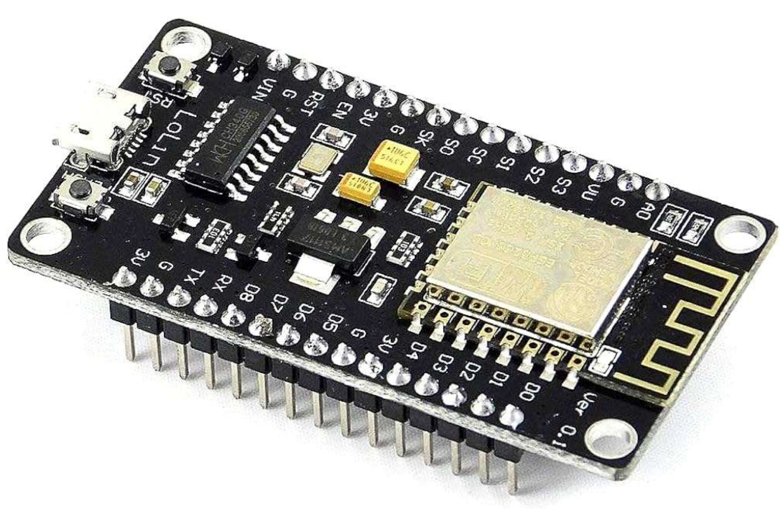


Figure 2.1: ESP8266

2.2 LDR

The **LDR (Light Dependent Resistor)** is a sensor that changes its resistance based on the intensity of light it receives. In **bright light**, the resistance decreases, allowing more current to flow through, while in **low light or darkness**, the resistance increases, reducing current flow. This property makes LDRs ideal for applications like **automatic lighting systems, streetlights, and light-sensing circuits**. They are commonly used to detect day-night changes or adjust brightness in devices. LDRs are inexpensive, easy to use, and have a **slow response time** compared to more advanced light sensors like photodiodes. They are non-polar, meaning they can be connected in either direction in a circuit. However, their sensitivity may vary depending on environmental factors like temperature. LDRs are perfect for **simple light-based automation tasks** in embedded systems and IoT projects.

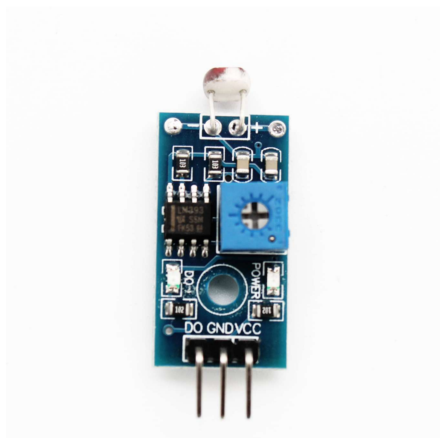


Figure 2.2: LDR SENSOR

2.3 DHT11

The **DHT11 sensor** is a low-cost digital sensor used to measure **temperature and humidity**. It features a capacitive humidity sensor and a thermistor to provide precise readings of both parameters. The DHT11 outputs data in a digital format, making it easy to interface with microcontrollers like **Arduino and ESP8266**. It offers a **temperature range** of 0–50°C and **humidity range** of 20–90% RH, with basic accuracy suitable for most hobbyist and IoT projects. Though slower in response and less precise than advanced sensors, it is highly popular due to its **affordability and ease of use**. The sensor operates on **3.3–5V** power, making it compatible with various systems. It is commonly used in weather monitoring, home automation, and greenhouse management. However, frequent polling can affect its performance due to a **2-second sampling rate limit**.

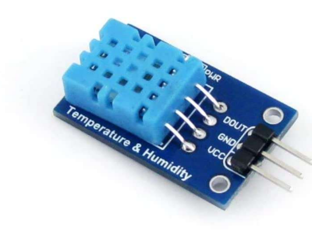


Figure 2.3: DHT11 SENSOR

2.4 RAIN SENSOR

A **rain sensor** detects the presence and intensity of rain by measuring the conductivity of water on its surface. It typically consists of a sensing plate with exposed conductive traces that form a grid. When rainwater touches the plate, the conductivity increases, triggering a signal. The sensor outputs either analog or digital data, making it easy to interface with microcontrollers like **ESP8266**. It is commonly used in **weather stations, irrigation systems**, and **automatic wipers** for vehicles.

Figure 2.4: RAIN SENSOR

2.5 LCD

An **LCD (Liquid Crystal Display)** is a flat-panel display that uses liquid crystals and backlighting to show information. It is commonly used in embedded systems for displaying **text, numbers, and basic symbols**. An **I2C module** can be added to reduce wiring, making it easier to interface with microcontrollers like **ESP8266 or Arduino**. LCDs are widely used in **weather stations, appliances, and digital meters** for clear and low-power visual output.

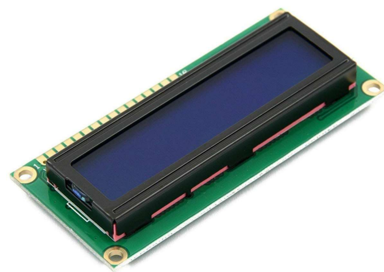


Figure 2.5: LCD

2.6 I2C

I2C (Inter-Integrated Circuit) is a communication protocol that enables multiple devices to communicate using just **two wires**: **SDA (data line)** and **SCL (clock line)**. It allows microcontrollers, sensors, and peripherals like **LCDs** to share data efficiently with unique addresses for each device. I2C supports both **master-slave** communication, making it ideal for embedded systems with multiple components. Its simplicity and low pin usage make it popular in **IoT projects and small electronics**.

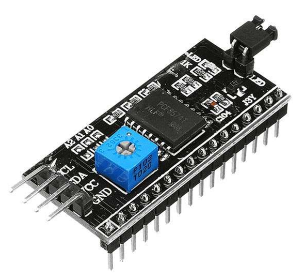


Figure 2.6: I2C

2.7 BREADBOARD

A **breadboard** is a reusable platform for building and testing electronic circuits without soldering. It consists of rows and columns of interconnected holes where components like **resistors, sensors, and microcontrollers** can be inserted. Breadboards are ideal for **prototyping** because they allow easy modifications and troubleshooting. They are widely used in **educational projects** and by hobbyists to quickly assemble circuits.

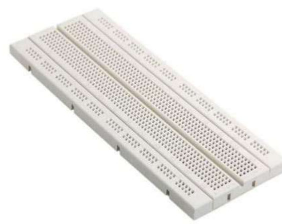


Figure 2.7: BREAD BOARD

2.8 JUMPER WIRES

A **breadboard** is a reusable platform for building and testing electronic circuits without soldering. It consists of rows and columns of interconnected holes where components like **resistors, sensors, and microcontrollers** can be inserted. Breadboards are ideal for **prototyping** because they allow easy modifications and troubleshooting. They are widely used in **educational projects** and by hobbyists to quickly assemble circuits.

Figure 2.8: JUMPER WIRES

CHAPTER 3

WORKING

The **Automated Weather Monitoring Station** is designed to continuously monitor environmental conditions, such as temperature, humidity, rainfall, and light intensity. It utilizes an **ESP8266** microcontroller for data processing and wireless communication, along with several sensors to gather weather-related data. The processed information is displayed on an **LCD with I2C** for local viewing and sent to the **Blynk app** for remote monitoring. Below is a detailed explanation of its components, working mechanism, and data flow.

3.1 WORKING MECHANISM

1. **Power Supply:** The entire system is powered using a stable power supply, typically through a USB or battery source suitable for the ESP8266 and sensors.

2. **Data Collection:** The ESP8266 initializes all sensors upon startup:

   - **DHT11 Sensor:** The microcontroller reads temperature and humidity values every few seconds. The DHT11 provides digital output, which is easy to interpret.
   - **Rain Sensor:** The rain sensor continuously monitors for rain. When rain is detected, the sensor's output changes, indicating moisture presence.
   - **LDR Sensor:** The light-dependent resistor measures ambient light levels. It varies its resistance based on light intensity, providing an analog signal to the ESP8266.

3. **Data Processing:** The ESP8266 processes the data received from the sensors:

   - Temperature and humidity readings from the DHT11 are converted into meaningful data formats.
   - The rain sensor output is interpreted to determine if it is raining (active) or not (inactive).
   - The LDR sensor's output is read and converted to a light intensity value.

4. **Display on LCD:**
   - The processed data is sent to the **LCD with I2C** for local display. The LCD is updated every few seconds to show the latest readings for:
     - Temperature (°C)
     - Humidity (%)
     - Rainfall (detected or not)
     - Light intensity (in lux or a relative scale)
   - The I2C interface simplifies wiring and allows for multiple devices to communicate over two wires (SDA and SCL).
5. **Data Transmission to Blynk App:**
   - The ESP8266 establishes a Wi-Fi connection to the internet.
   - Using the Blynk library, the microcontroller sends the collected data to the Blynk server.
   - The Blynk app on a smartphone or tablet is configured to display real-time weather data, providing a user-friendly interface for monitoring:
     - Temperature
     - Humidity
     - Rain status
     - Light intensity
   - Users can view these values remotely, enabling effective tracking of weather conditions from anywhere.

3.2 HARDWARE CONNECTIONS

1. **Connecting the DHT11 Sensor:**
   - **VCC Pin:** Connect to the **3.3V** or **5V** pin of the ESP8266 (depends on the model).
   - **GND Pin:** Connect to the **GND** pin of the ESP8266.
   - **DATA Pin:** Connect to a digital GPIO pin on the ESP8266 (e.g., GPIO4).
2. **Connecting the Rain Sensor:**
   - **VCC Pin:** Connect to the **3.3V** or **5V** pin of the ESP8266.
   - **GND Pin:** Connect to the **GND** pin of the ESP8266.
   - **AO Pin (Analog Output):** Connect to an analog GPIO pin on the ESP8266 (e.g., A0) to read rainfall intensity. Alternatively, if using a digital output, connect the digital output pin to a digital GPIO pin (e.g., GPIO5).

3. **Connecting the LDR Sensor:**
    - Connect one end of the LDR to **VCC** (3.3V or 5V).
    - Connect the other end of the LDR to one end of a resistor (typically 10k).
    - Connect the other end of the resistor to **GND**.
    - Connect the junction between the LDR and the resistor to an analog GPIO pin on the ESP8266 (e.g., A0). This setup forms a voltage divider, allowing the ESP8266 to read varying resistance as varying voltage.
4. **Connecting the LCD Display with I2C Module:**
    - **VCC Pin:** Connect to the **3.3V** or **5V** pin of the ESP8266.
    - **GND Pin:** Connect to the **GND** pin of the ESP8266.
    - **SDA Pin:** Connect to the SDA pin of the ESP8266 (typically GPIO4).
    - **SCL Pin:** Connect to the SCL pin of the ESP8266 (typically GPIO5).
5. **Power Supply:** Ensure that the ESP8266 is powered either via a USB cable or a suitable battery pack. If using a battery pack, make sure it provides adequate voltage for the ESP8266.

3.3 PROGRAMMING LOGIC

Initialization: The program begins by including necessary libraries and initializing components like the DHT11 sensor, rain sensor, I2C LCD, and the ESP8266 for Wi-Fi connectivity. It sets the Wi-Fi credentials and Blynk authentication token for remote monitoring.

Startup Display: In the setup() function, a startup message is displayed on the LCD for 3 seconds, indicating that the weather monitoring system is active. The system also establishes a connection to the Blynk server.

Sensor Reading Interval: A timer is set to trigger the readSensors() function every second, allowing the program to continuously gather data from the sensors.

Data Acquisition: The readSensors() function reads temperature and humidity from the DHT11 sensor and checks for valid readings. It also reads the analog value from the rain sensor, mapping it to a percentage for easier interpretation.

Data Display and Transmission: The sensor values are sent to the Blynk app for remote access, while simultaneously updating the LCD display with the latest temperature, humidity, and rain level, ensuring that both local and remote users can monitor the weather conditions effectively.

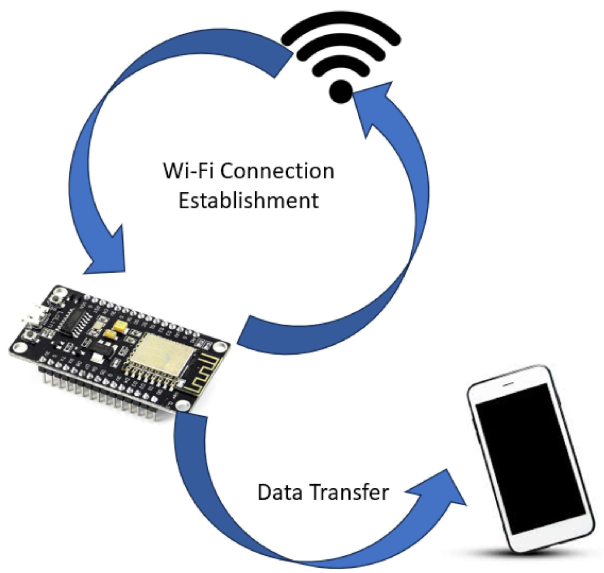


Figure 3.1: WORKING

3.4 SAFETY CONSIDERATIONS

Electrical Safety

Ensure stable and appropriate power supply to prevent damage.

Environmental Protection

Use a weatherproof enclosure for all components.

Data Security

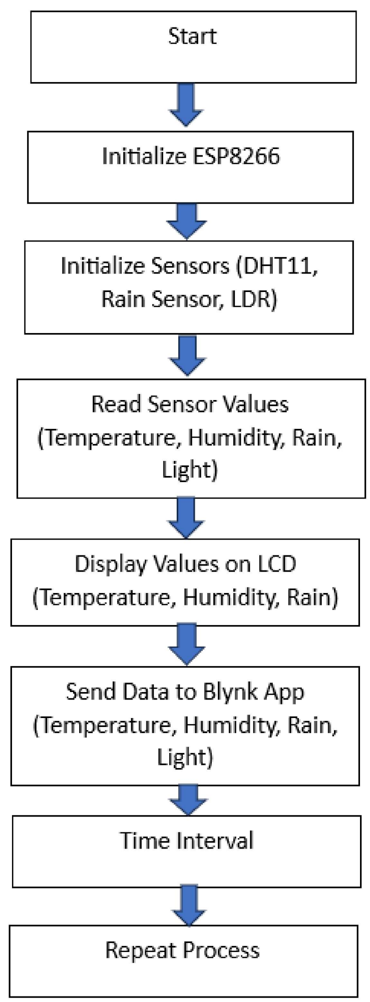
Use strong passwords and encryption for Wi-Fi networks.

User Interface and Accessibility

Securely mount the LCD display to avoid breakage.

CHAPTER 4

FLOW CHART



CHAPTER 5

BLYNK SETUP

Step 1: Sign Up / Log In

1. **Visit the Blynk Website**: Go to the https://blynk.io/Blynk website.
2. **Create an Account**: If you don't have an account, sign up by following the prompts. If you already have an account, simply log in.

Step 2: Create a New Project Template

1. **Access the Dashboard**: Once logged in, navigate to the dashboard.
2. **Create a Template**:
   - Click on **"Templates"** in the sidebar.
   - Select **"Create New Template"**.
   - **Name the Template**: Provide a descriptive name for your template (e.g., "Weather Monitoring Station").
   - **Select Device Type**: Choose the appropriate device type from the dropdown (e.g., ESP8266).
   - **Choose Connection Type**: Select the connection type (e.g., Wi-Fi).
3. **Add Variables**:
   - Define virtual pins for the data you want to send (e.g., temperature, humidity, rain, light).
   - For each variable, select the type (e.g., **"Gauge"**, **"Value Display"**) and configure any additional settings.
4. **Save the Template**: After configuring the template, click **"Save"**.

Step 3: Add Device to the Template

1. **Select the Template**: From the Templates list, select the one you just created.
2. **Add Device**:
   - Click on **"Add Device"** or a similar button.
   - **Device Name**: Enter a name for your device (e.g., "Weather Station 01").
   - **Auth Token**: The platform will generate an authentication token for the device. Save this token, as it will be needed in your code.
3. **Link the Device**: The device will now be linked to the template, and you can manage it through the Blynk app.

Step 4: Configure Device in Code

1. **Install Blynk Library**: Make sure you have the Blynk library installed in your Arduino IDE.
2. **Modify Code**: Update your device code with the new authentication token and template ID. Here's a modified version of the earlier code:

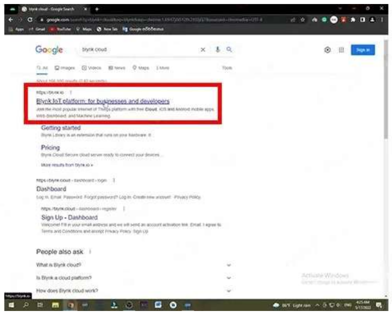
Step 5: Access the Blynk App

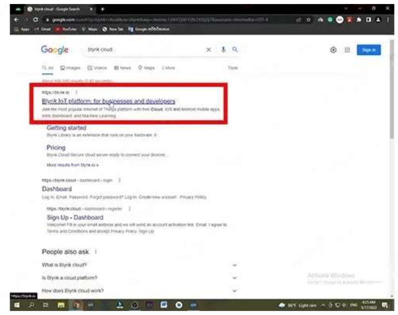
1. **Download the Blynk App**: If you haven't already, install the Blynk app on your mobile device.
2. **Log In**: Use the same account credentials to log in.
3. **Access Your Template**:
   - Navigate to **"Devices"** or **"Projects"** in the app.
   - Find your template and select it to view data.
4. **Customize Your Dashboard**: You can add widgets to your mobile dashboard that correspond to the variables defined in your template.

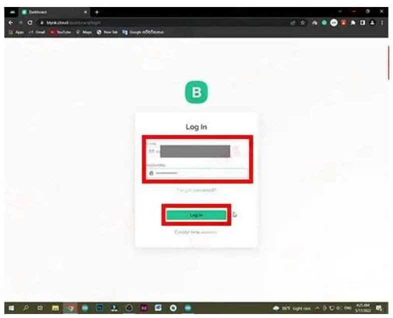
Step 6: Monitor and Control

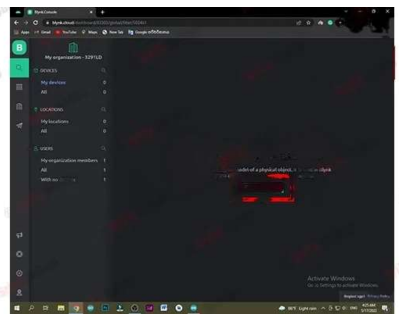
Once your device is connected and running, you can monitor the sensor data in real-time through both the web dashboard and the mobile app. You can also send commands back to the device if needed.

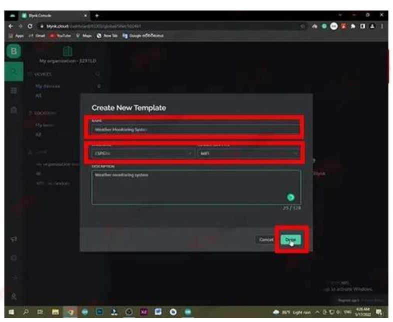
5.1 PROCESS

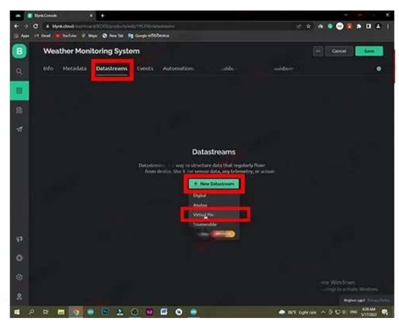

Blynk IoT platform: for businesses and developers
Getting started
Pricing
Dashboard
Sign Up - Dashboard
People also ask

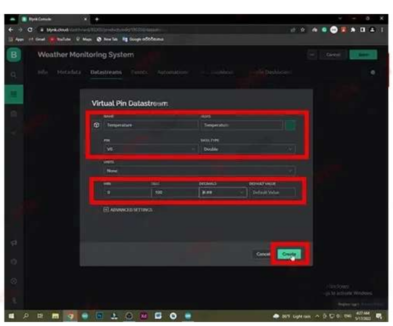

Blynk IoT platform: for businesses and developers
Getting started
Pricing
Dashboard
Sign Up - Dashboard
People also ask

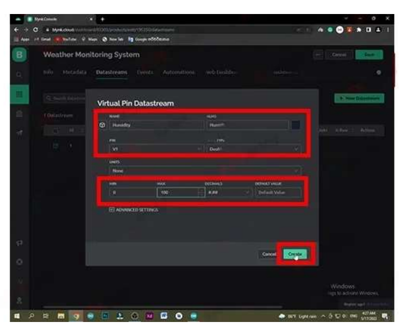

Log In

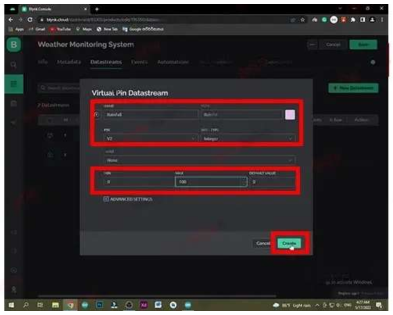


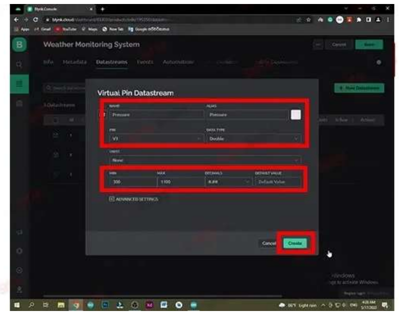

Create New Template

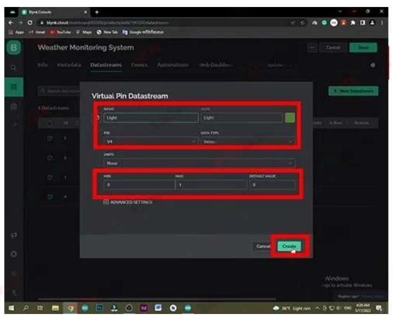

Weather Monitoring System
Datastreams

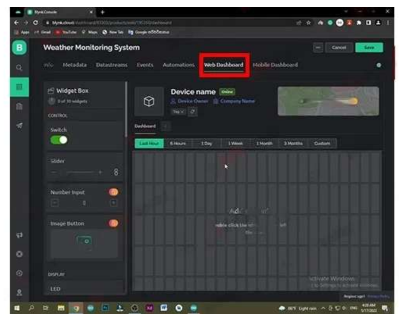

Weather Monitoring System
Virtual Pin Datastream

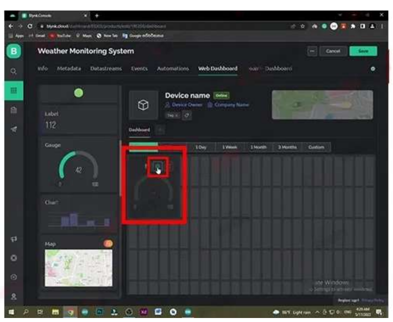

Weather Monitoring System
Virtual Pin Datastream

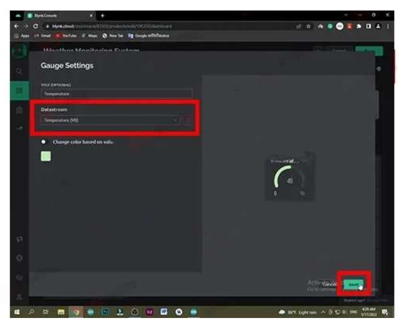

Weather Monitoring System
Virtual Pin Datastream


Weather Monitoring System
Virtual Pin Datastream

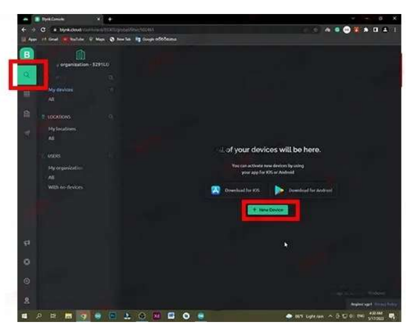

Weather Monitoring System
Virtual Pin Datastream

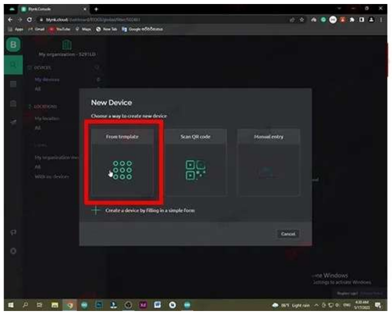
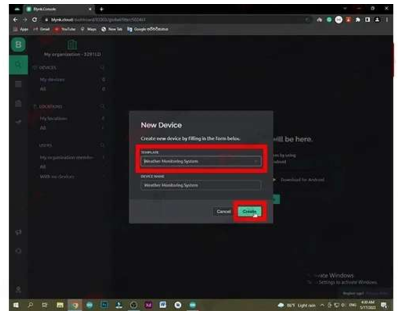

Weather Monitoring System
Device name


Weather Monitoring System
Device name

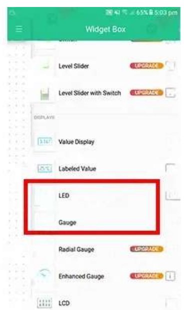
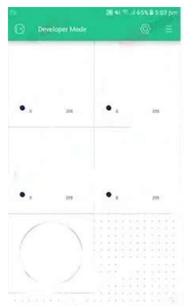
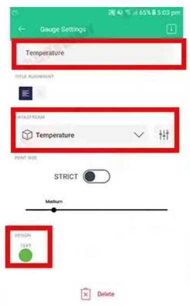

Gauge Settings


Weather Monitoring System
Device name



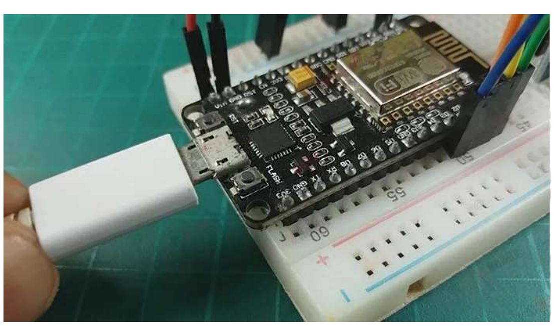
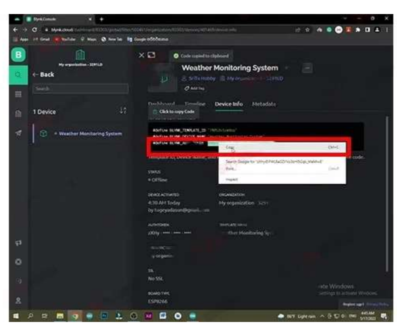



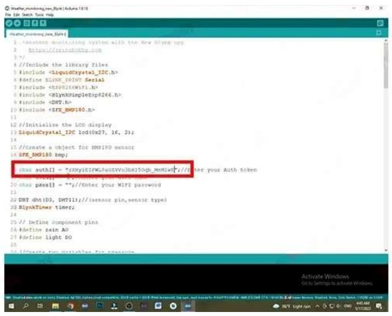


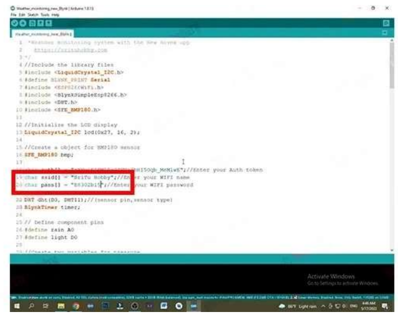




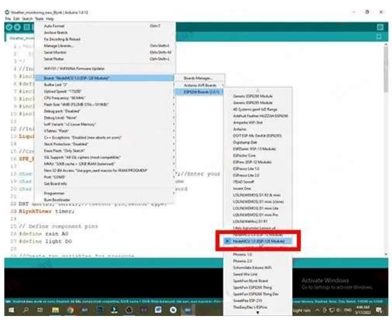


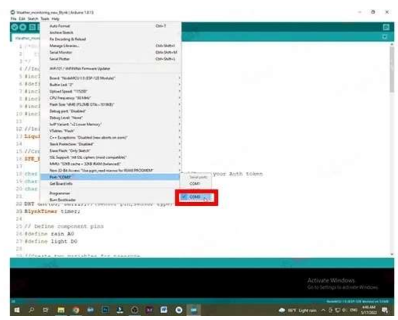


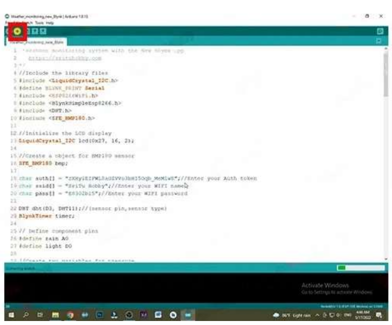













CHAPTER 6

ARDUINO CODE

```
// Include the required libraries
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

// Initialize the LCD display (I2C Address: 0x27, 16 columns, 2 rows)
LiquidCrystal_I2C lcd(0x27, 16, 2);

// DHT11 Sensor Setup
#define DHTPIN D3       // Pin connected to DHT11 data pin
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Rain and LDR sensor pins
#define RAIN_SENSOR_PIN A0  // Analog pin for rain sensor
#define LDR_SENSOR_PIN D0   // LDR sensor connected to D0

// Blynk credentials
char auth[] = "[redacted]";  // Enter your Blynk Auth Token
char ssid[] = "DSP";  // Enter your WiFi Name (SSID)
char pass[] = "444455555";        // Enter your WiFi Password

// Initialize Blynk Timer
BlynkTimer timer;
```

```
// Variables for sensor values
float temperature, humidity;
int rainLevel;
int ldrReading;  // Variable to hold LDR sensor reading

// Thresholds for light detection
const int lightThreshold = 500; // Adjust this value based on your testing

void setup() {
  // Start serial communication for debugging
  Serial.begin(9600);

  // Initialize Blynk connection
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  // Initialize DHT11 sensor
  dht.begin();

  // Initialize I2C LCD
  lcd.init();
  lcd.backlight();

  // Display a startup message on LCD
  lcd.setCursor(0, 0);
  lcd.print("Weather Monitor");
  lcd.setCursor(4, 1);
  lcd.print("System");
  delay(3000);  // Hold message for 3 seconds
  lcd.clear();

  // Set LDR pin as input
  pinMode(LDR_SENSOR_PIN, INPUT);

  // Set timer to call sensor reading functions every second (1000ms)
  timer.setInterval(1000L, readSensors);
}
```

```
// Function to read sensors and update values
void readSensors() {
  // Read temperature and humidity from DHT11
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();

  // Check if readings are valid
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Read rain sensor values (mapped to 0-100)
  rainLevel = analogRead(RAIN_SENSOR_PIN);
  rainLevel = map(rainLevel, 0, 1024, 0, 100);

  // Read LDR sensor value
  ldrReading = analogRead(LDR_SENSOR_PIN);
  Serial.print("LDR Reading: ");
  Serial.println(ldrReading);  // Debugging output

  // Determine if light is detected based on LDR reading
  if (ldrReading < lightThreshold) {  // Light detected
    Serial.println("Light Level: HIGH");
    Blynk.virtualWrite(V4, 1);  // Set virtual pin V4 to 1 for light detected (HIGH
  } else {  // Light not detected
    Serial.println("Light Level: LOW");
    Blynk.virtualWrite(V4, 0);    // Set virtual pin V4 to 0 for no light detected
  }

  // Send data to Blynk app
  Blynk.virtualWrite(V0, temperature);  // Send temperature to Virtual Pin V0
  Blynk.virtualWrite(V1, humidity);     // Send humidity to Virtual Pin V1
  Blynk.virtualWrite(V2, rainLevel);    // Send rain level to Virtual Pin V2

  // Display values on LCD
  lcd.setCursor(0, 0);
  lcd.print("T:");
```

```
  lcd.print(temperature);
  lcd.print("C ");    // Space after temperature
  lcd.print("H:");
  lcd.print(humidity);
  lcd.print("% ");     // Space after humidity

  lcd.setCursor(0, 1);
  lcd.print("R:");
  lcd.print(rainLevel);
  lcd.print("%  ");     // Space after rain level

  // Display light status on LCD
  lcd.setCursor(8, 1);
  if (ldrReading < lightThreshold) {
    lcd.print("L:HIGH   ");  // Display "L:HIGH" for light detected
  } else {
    lcd.print("L:LOW    ");   // Display "L:LOW" for light not detected
  }
}

void loop() {
  Blynk.run();  // Run Blynk library
  timer.run();  // Run Blynk timer
}
```

CHAPTER 7

OUTPUT

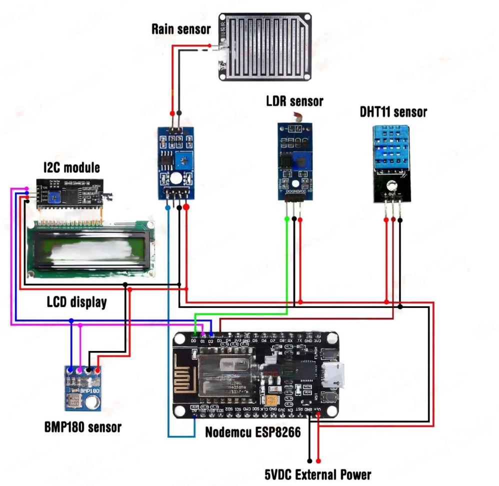


Figure 7.1: CIRCUIT

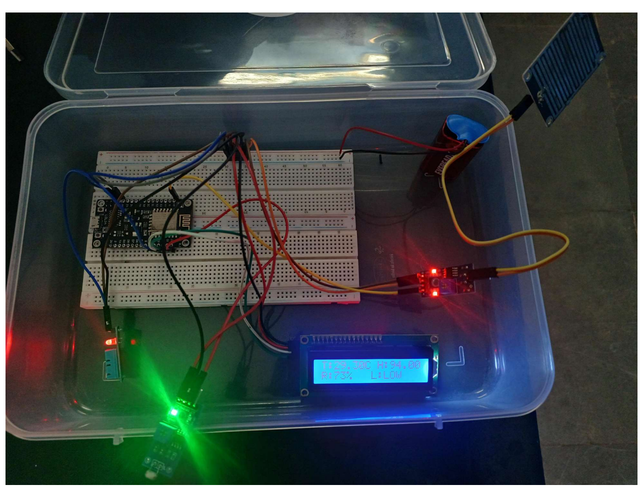


Figure 7.2: PROJECT

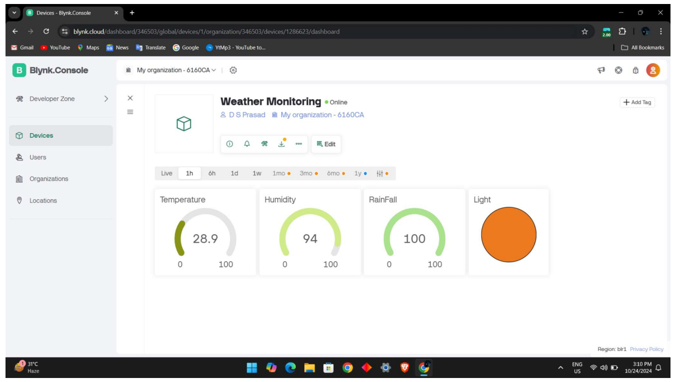


Figure 7.3: Monitoring Through Website

Figure 7.4: Monitoring Through Mobile

CHAPTER 8

LITERATURE SURVEY

8.1 OVERVIEW

An **automated weather monitoring system** leveraging IoT technology provides real-time environmental data, enabling users to monitor temperature, humidity, rainfall, and light intensity remotely. Using inexpensive and accessible components like the **ESP8266 microcontroller**, **DHT11 sensor for temperature and humidity**, **LDR for light intensity**, and **rain sensors**, the system displays this data on an **I2C-enabled LCD** and sends it to the **Blynk app** for remote access. Such a setup offers several advantages, including **automation, reduced human intervention, and remote accessibility through mobile apps**. The ability to visualize data on both a **physical display and a mobile app dashboard** ensures a practical solution for weather tracking.

In particular, IoT platforms like **Blynk 2.0** simplify connectivity by integrating cloud services, allowing users to control and monitor devices in real-time. The **ESP8266** plays a key role by connecting the sensors to the internet, uploading sensor data to the cloud, and enabling communication with the mobile app. The **Blynk app** also allows users to configure alerts and track historical data, making it useful for forecasting trends and environmental analysis. This combination of **hardware, software, and cloud integration** provides **a cost-effective, modular, and scalable solution** for weather monitoring systems.

8.2 LITERATURE SURVEY DETAILS

1. **Title:** Smart Weather Monitoring System using IoT
   Description: This study explores the design of a real-time weather monitoring system with ESP8266, DHT11, and rain sensors, integrated with Blynk for remote data analysis.
   Link: https://github.com/desilva2017/Smart-Weather-Monitoring-System-using-IOT

2. **Title:** IoT-Based Weather Station using ESP8266 and BME280

Description: This project focuses on monitoring temperature, humidity, and pressure in real-time using BME280 sensors, connected via Blynk. It offers practical implementation for internal environments.

Link: https://iotprojectsideas.com/esp8266-bme280-weather-station

3. **Title:** IoT Weather Monitoring System for Smart Agriculture

   Description: This research targets agricultural applications, using IoT to monitor weather patterns to help farmers make informed decisions. It employs sensors like DHT11 for temperature and humidity.

   Link:https://create.arduino.cc/projecthub/techstudycell/weather-monitoring-system-bdbf44

4. **Title:** Weather Monitoring with NodeMCU and New Blynk App

   Description: Describes setting up an IoT-based weather system using NodeMCU and interfacing with sensors through the updated Blynk platform for easy monitoring.

   Link: https://github.com/Rakshita2003/Weather-Monitoring-System

5. **Title:** Environmental Monitoring System with ESP8266

   Description: Focuses on measuring key environmental parameters and transmitting them wirelessly using ESP8266 and various sensors integrated with the Blynk app.

   Link: https://iotprojectsideas.com/environmental-monitoring-system-esp8266-bme680

6. **Title:** Cloud-Connected Weather Station

   Description: This study examines the integration of cloud services with IoT for enhanced data logging and prediction capabilities, leveraging ESP8266 and Blynk for visualization.

   Link: https://github.com/desilva2017/Smart-Weather-Monitoring-System-using-IOT

7. **Title:** Remote Weather Station Monitoring Using DHT11 and ESP8266

   Description: This project involves real-time data transmission through Wi-Fi modules, collecting temperature and humidity information to the Blynk app.

   Link: https://iotprojectsideas.com/dht11-esp8266-weather-monitor

8. **Title:** Advanced IoT Weather System for Urban Use

   Description: This research emphasizes urban weather monitoring and its impact on local communities, utilizing ESP modules for public data sharing via Blynk.

   Link: https://iotprojectsideas.com/smart-weather-monitoring-urban

9. **Title:** IoT-Based Weather Station for Real-Time Alerts

   Description: Describes an alert-based system connected to the Blynk platform, offering notifications to users during significant weather changes.

   Link: https://github.com/Rakshita2003/Weather-Monitoring-System

10. **Title:** Weather Data Analysis Using IoT Sensors and Blynk App

    Description: Focuses on gathering and analyzing data over time, demonstrating how trends in temperature and humidity can support predictions.

    Link: https://iotprojectsideas.com/weather-analysis-iot-sensors

11. **Title:** IoT-Based Weather Monitoring Using ESP8266

    Description: This project uses ESP8266 with DHT11 and LDR sensors to monitor environmental parameters and display the data both on an LCD and remotely through the Blynk app.

    Link: https://how2electronics.com/iot-live-weather-station-monitor/

12. **Title:** IoT Weather Station Using Blynk Application

    Description: This station utilizes NodeMCU with DHT11 and Blynk to send real-time data wirelessly, making it accessible remotely.

    Link: https://www.instructables.com/IoT-Weather-Station-Using-Blynk-Application/

13. **Title:** IoT Atmospheric Monitoring System

    Description: The system uses ESP8266, DHT11, LDR, and rain sensors to collect environmental data and displays it via Blynk for precision monitoring.

    Link: https://github.com/Prathvihegde/IOT

14. **Title:** Weather Monitoring Using Blynk with DHT11 Sensor

    Description: A guide for building a low-cost weather station using ESP8266 and Blynk to monitor temperature and humidity remotely.

    Link: https://randomnerdtutorials.com/esp8266-weather-station-with-dht11-blynk/

15. **Title:** IoT-Based Environmental Monitoring System

    Description: This system employs ESP8266 and multiple sensors to collect data, which is displayed on the Blynk platform, aiding in environmental tracking.

    Link: https://www.electronicshub.org/iot-weather-station/

CHAPTER 9

CONCLUSION

The **automated weather monitoring station** designed with **ESP8266**, **DHT11**, **LDR sensor**, **rain sensor**, and **LCD with I2C** provides a **cost-effective, scalable, and efficient solution** for real-time environmental monitoring. By leveraging **embedded systems and IoT capabilities**, the project demonstrates how multiple environmental parameters like temperature, humidity, light intensity, and rainfall can be measured, analyzed, and monitored effectively.

The **integration with the Blynk platform** enhances the usability of the system by enabling remote access through a smartphone. This feature allows users to **visualize data on dashboards** or receive alerts via mobile apps, ensuring continuous awareness of environmental conditions from anywhere with internet connectivity. Additionally, the **LCD module with I2C** facilitates on-site monitoring, making the system suitable for areas where continuous mobile connectivity may be limited. Furthermore, **trend analysis over time** using stored weather data can support predictive models, leading to **better forecasting** and **decision-making**.

This project demonstrates how **IoT-enabled solutions** can transform traditional weather monitoring systems into **smart, connected systems**. With future upgrades—such as using additional sensors (e.g., air quality or wind speed sensors) or connecting to **cloud-based platforms for advanced analytics**—the system could evolve into a **comprehensive environmental monitoring solution**.

In summary, this automated weather monitoring station is **scalable, reliable, and practical** for a variety of applications, proving to be an ideal solution for individuals and organizations seeking **low-cost and easy-to-implement weather tracking systems**. The combination of **embedded hardware and IoT** showcases the potential of technology in **bridging gaps between environmental sensing and human interaction** in meaningful ways.

CHAPTER 10

REFERENCES

1. Patil, S., & Sutar, S. (2017). IoT Based Weather Monitoring System Using NodeMCU ESP8266. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 6*(4), 689-692.

2. Kaur, G., Sharma, A., & Sharma, V. (2018). Real-Time Weather Monitoring Using IoT and Cloud Platforms. In *Proceedings of 2018 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)* (pp. 1023-1026). IEEE.

3. Vishwakarma, V., & Singh, P. K. (2020). IoT-Based Environmental Monitoring Using ESP8266 and Blynk App. *International Journal of Scientific & Technology Research, 9*(4), 190-194.

4. Verma, A., Gupta, S., & Jain, M. (2019). A Smart Weather Monitoring System Using IoT. *International Journal of Engineering and Advanced Technology (IJEAT), 8*(6), 2320-2323.

5. Ramesh, M., & Prasad, K. (2021). Development of Real-Time IoT-Based Weather Station Using ESP8266. *International Conference on Smart Technologies for Smart Nation (SmartTechCon)* (pp. 1100-1105). IEEE.

6. Desai, P., Patil, V., & Shrivastava, R. (2022). IoT-Based Weather Forecasting System Using NodeMCU. *Journal of Electronics, Communication and Instrumentation Engineering Research and Development (JECIERD), 12*(1), 10-15.

7. Johnson, M., & Sharma, R. (2020). Real-Time Data Logging for Weather Monitoring Using IoT and Cloud Computing. *Journal of Emerging Trends in Engineering and Applied Sciences, 11*(2), 73-78.

8. Singh, D., & Verma, R. (2018). Wireless Weather Station Using ESP8266 and Cloud Platforms. *International Conference on Recent Advances in Technology and Engineering (RATE)* (pp. 235-238). IEEE.

9. Sharma, N., & Mehra, S. (2019). IoT Weather Monitoring System for Agriculture. *International Journal of Innovative Research in Science, Engineering and Technology, 8*(3), 542-545.

10. Ali, A., & Malik, M. (2021). Weather Monitoring and Prediction Using ESP8266 and Machine Learning Models. *Proceedings of the International Conference on Machine Learning and Data Science (ICMLDS)* (pp. 201-206). IEEE.

11. Karmakar, A., & Saha, S. (2020). A Smart Weather Monitoring System Using IoT and Cloud Computing. *International Journal of Recent Technology and Engineering (IJRTE), 8*(4), 478-481.

12. Jaiswal, P. K., & Soni, R. (2021). IoT Based Weather Monitoring and Control System Using ESP8266. *International Journal of Computer Applications, 975*, 13-18.

13. Khan, A., & Kausar, F. (2021). IoT Based Weather Monitoring System Using DHT11 and ESP8266. *International Journal of Emerging Technologies in Engineering Research (IJETER), 9*(4), 26-31.

14. Choudhary, R., & Singh, S. (2021). Real-Time Weather Monitoring System Using IoT. *International Journal of Computer Applications, 975*, 35-40.

15. Kumar, R., & Singh, R. (2020). IoT Based Weather Monitoring System Using Raspberry Pi and Sensors. *International Journal of Engineering Research & Technology (IJERT), 9*(1), 245-249.