your roots to success.
Estd.2007

NARSIMHA REDDY ENGINEERING COLLEGE

An Autonomous Institution Affiliated to JNTUH | Approved by AICTE
Accredited by NBA & NAAC with 'A' Grade

IoT INNOVATION AND PROTOTYPE DEVELOPMENT

USING BLYNK APP

USING ESP Wi-Fi MODULE AND ARDUINO

FOR NARSIMHA REDDY ELECTRONICS AND COMMUNICATION ENGINEERING STUDENTS

- Real-time Monitoring and Control
- Wireless Connectivity with ESP Wi-Fi Module
- IoT Platform Integration
- Data Visualization and Analytics
- Innovative Projects for Real-world Applications

Author
Dr V PURANDHAR REDDY

PROJECT MANUAL

PROJECT–1

IoT HOME LIGHT CONTROL USING BLYNK APP

Using ESP8266/ESP32, Relay Module and Arduino IDE

1. AIM

To design and implement an Internet of Things (IoT) based home automation system that allows users to remotely control an AC light bulb through the Blynk IoT platform using an ESP8266/ESP32 Wi-Fi module.

2. OBJECTIVES

- Understand IoT-based home automation.
- Interface ESP8266/ESP32 with Relay Module.
- Configure Blynk IoT Cloud.
- Control appliances over the Internet.
- Learn GPIO programming using Arduino IDE.
- Build a scalable smart home prototype.

3. COMPONENTS REQUIRED

S.No	Component	Quantity
1	ESP8266 NodeMCU / ESP32	1
2	1-Channel Relay Module	1
3	230V AC Bulb with Holder	1
4	Breadboard	1
5	Jumper Wires	Several
6	USB Cable	1
7	Wi-Fi Internet Connection	1
8	Blynk Mobile App	1

4. THEORY

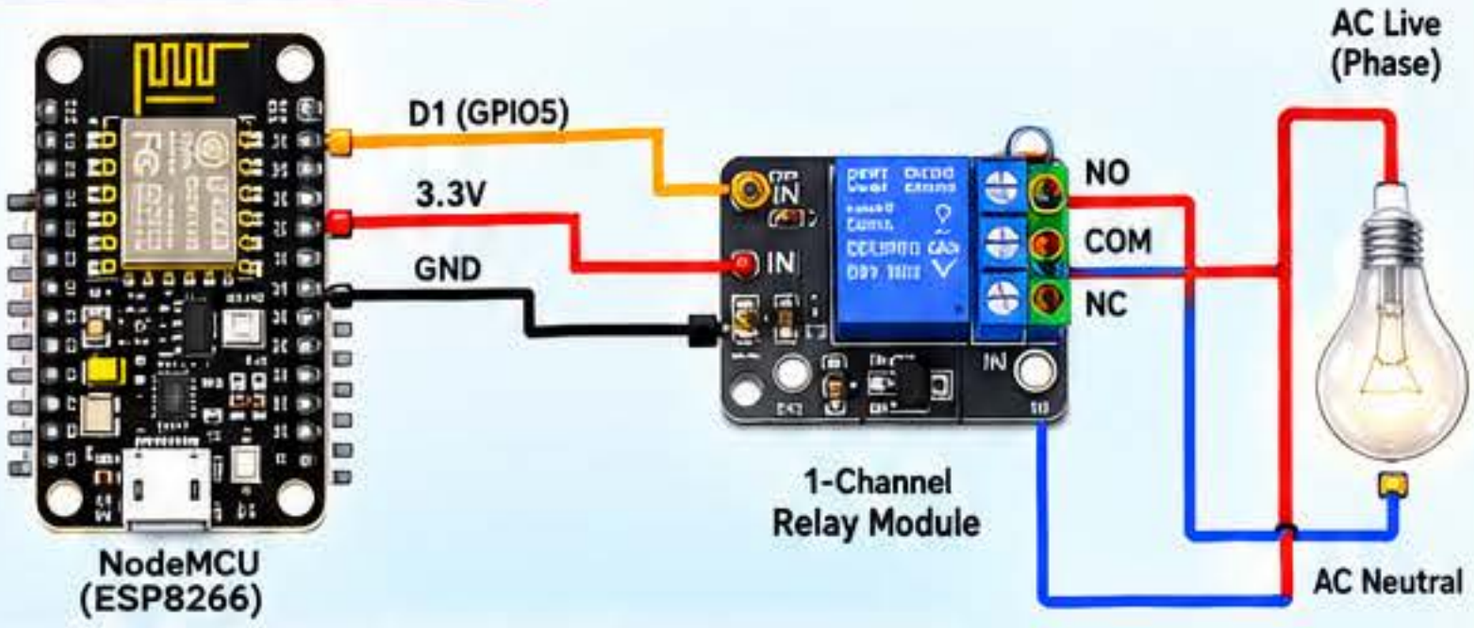
Home automation is one of the most popular applications of the Internet of Things (IoT). In this project, an ESP8266/ESP32 module connects to the Wi-Fi network and communicates with the Blynk Cloud Server. The Blynk mobile application sends commands through the cloud to the ESP board, which activates or deactivates a relay module. The relay behaves like an electronic switch that controls an AC bulb.

The project demonstrates:

- Internet-based switching
- Cloud communication
- Remote monitoring
- Relay interfacing
- Mobile App Control

5. BLOCK DIAGRAM

6. CIRCUIT DIAGRAM



ESP8266 Connections		
ESP8266 Pin	Relay Module Pin	Description
D1 (GPIO5)	IN	Digital Input (Control Pin)
3.3V	VCC	Power Supply (3.3V)
GND	GND	Ground

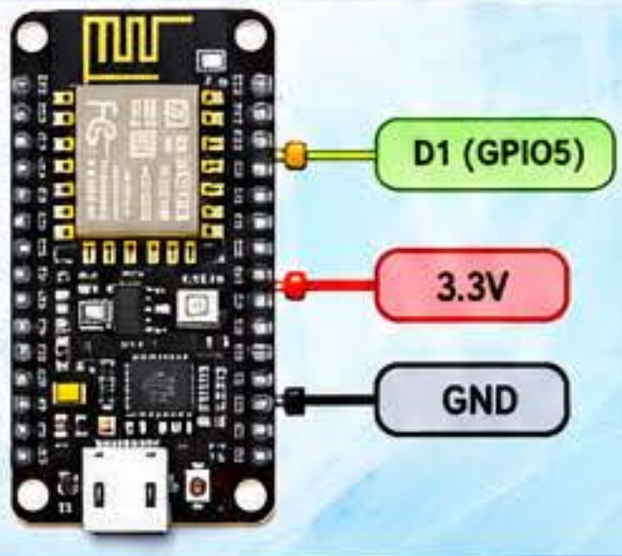
7. WORKING PRINCIPLE

1. ESP8266 connects to Wi-Fi.
2. ESP establishes communication with Blynk Cloud.
3. User presses switch in Blynk App.
4. Command reaches ESP through internet.
5. ESP changes GPIO output.
6. Relay energizes.
7. Bulb turns ON/OFF.

8. SOFTWARE REQUIREMENTS

- Arduino IDE
- ESP8266 Board Package
- Blynk IoT Library
- Blynk Mobile App
- Wi-Fi Network

9. ESP8266 PIN OUT (Key Pins Used)



7. WORKING PRINCIPLE

1. ESP8266 connects to Wi-Fi.
2. ESP establishes communication with Blynk Cloud.
3. User presses switch in Blynk App.
4. Command reaches ESP through internet.
5. ESP changes GPIO output.
6. Relay energizes.
7. Bulb turns ON/OFF.

8. SOFTWARE REQUIREMENTS

- Arduino IDE
- ESP8266 Board Package
- Blynk IoT Library
- Blynk Mobile App
- Wi-Fi Network

9. BLYNK DASHBOARD CONFIGURATION

Step 1: Create Template

Template Name: ✓
Home Automation

Hardware:
ESP8266

Connection:
Wi-Fi

Step 2: Create Datastream

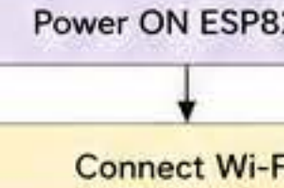
Virtual Pin:
V0

Data Type:
Integer

Range:
0 to 1

Step 3: Add Widget

Widget:
Button (Switch)



Datastream: **V0**
Mode: **Switch**

Step 4: Copy Credentials

Copy the following from Blynk Console:

- BLYNK_TEMPLATE_ID
- BLYNK_TEMPLATE_NAME
- BLYNK_AUTH_TOKEN

Paste them in Arduino Code.

10. ARDUINO PROGRAM

```
#define  BLYNK_TEMPLATE_ID     "YOUR_TEMPLATE_ID"
#define  BLYNK_TEMPLATE_NAME   "Home Automation"
#define  BLYNK_AUTH_TOKEN      "YOUR_AUTH_TOKEN"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] =  "YOUR_WIFI_NAME";
char pass[] =  "YOUR_WIFI_PASSWORD";

#define RELAY_PIN D1

BLYNK_WRITE(V0)
{
  int value = param.asInt();

  if(value == 1)
  {
    digitalWrite(RELAY_PIN, LOW);   // Relay ON (Bulb ON)
  }
  else
  {
    digitalWrite(RELAY_PIN, HIGH);  // Relay OFF (Bulb OFF)
  }
}

void setup()
{
  Serial.begin(115200);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH); // Relay OFF initially
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}
void loop()
{
  Blynk.run();
}
```

11. FLOWCHART

12. HARDWARE CONNECTIONS SUMMARY

ESP8266 (NodeMCU)	Relay Module
D1 (GPIO5)	IN
3.3V	VCC
GND	GND

AC SIDE CONNECTIONS (CAUTION)

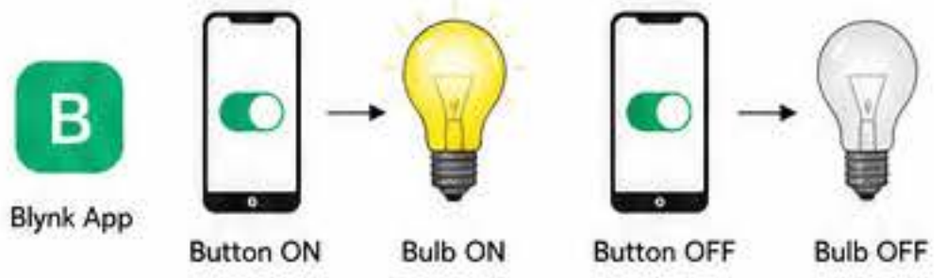
- AC Live → COM of Relay
- AC Live → NO of Relay → Bulb Live
- AC Neutral → Bulb Neutral

High Voltage
Use Caution!

13. SERIAL MONITOR OUTPUT (EXPECTED)

14. RESULT

When the button in the Blynk App is pressed, the relay switches ON and the connected AC bulb glows. When the button is pressed again, the relay turns OFF and the bulb turns OFF. The system can be controlled remotely from anywhere in the world using internet connection.



12. PROCEDURE (STEP-BY-STEP)

Step 1 : Install Arduino IDE.

Step 2 : Install ESP8266 Board Package.
(File → Preferences → Additional Board URL →
http://arduino.esp8266.com/stable/package_esp8266com_index.json)

Step 3 : Install Blynk Library.
(Sketch → Include Library → Manage Libraries → Search "Blynk")

Step 4 : Create Blynk Template.

Step 5 : Create Datastream V0.

Step 6 : Create Button Widget (Switch).

Step 7 : Connect Relay with ESP8266 as per circuit diagram.

Step 8 : Enter Wi-Fi Name, Password, Template ID and Auth Token in the Arduino code.

Step 9 : Upload Code to ESP8266.

Step 10 : Open Serial Monitor. Expected output:
Connecting WiFi...
WiFi Connected
Connecting Blynk...
Connected

Step 11 : Press Button in Blynk App → Relay clicks → Bulb turns ON.
Press again → Bulb turns OFF.

13. BLYNK APP INTERFACE (EXAMPLE)

BLYNK WIDGET DETAILS

Widget Type	: Button
Datastream	: V0
Mode	: Switch
Label	: Smart Light
Button Text (ON)	: ON
Button Text (OFF)	: OFF
Color	: Green

NOTE:

- Ensure stable Wi-Fi connection.
- Keep the device powered continuously.
- Relay module should be opto-isolated for safety.
- Do not touch AC connections while power is ON.

14. APPLICATIONS

- Smart Home Automation
- Office Automation
- Remote Appliance Control
- Hostel Room Automation
- Smart Classrooms
- Industrial Equipment Control
- Smart Energy Management

15. ADVANTAGES

- Low Cost
- Easy Installation
- Internet-based Control
- Energy Saving
- Expandable System
- User Friendly
- Supports Multiple Devices

16. PRECAUTIONS

- Never touch AC wiring while powered.
- Use an opto-isolated relay module.
- Ensure proper insulation of AC connections.
- Verify relay ratings before connecting loads.
- Use a stable Wi-Fi connection.
- Disconnect power before making hardware changes.

17. LEARNING OUTCOMES

After completing this project, students will be able to:

- Understand IoT architecture.
- Configure Blynk IoT Cloud.
- Interface ESP8266/ESP32 with relay modules.
- Develop Arduino programs for IoT applications.
- Control electrical appliances remotely via the Internet.
- Design basic smart home automation systems.

18. VIVA QUESTIONS

1. What is IoT?
2. What is ESP8266?
3. Why is a relay used?
4. What is the purpose of Blynk Cloud?
5. What is a Virtual Pin in Blynk?
6. What is GPIO?
7. Difference between ESP8266 and ESP32?
8. Why is Wi-Fi required?
9. What happens if the Internet connection is lost?
10. Name two applications of home automation.

19. TROUBLESHOOTING GUIDE

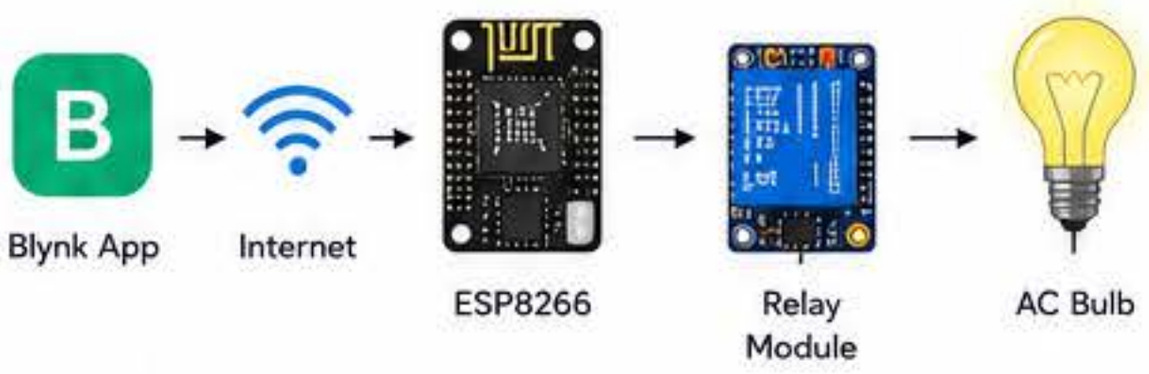
Problem	Possible Cause	Solution
ESP not connecting to Wi-Fi	Wrong Wi-Fi credentials Weak Wi-Fi signal	Check SSID and Password Ensure strong signal
Not connected to Blynk Cloud	Wrong Auth Token No Internet	Check Auth Token Check Internet connectivity
Relay not working	Wrong wiring Insufficient power	Check connections Provide 3.3V and GND properly
Bulb not turning ON	AC wiring issue Relay rating low	Check AC connections Use suitable relay module

20. EXPANSION IDEAS

- Add multiple relays to control multiple appliances.
- Add real-time energy monitoring using PZEM-004T.
- Add timer function to automate ON/OFF.
- Integrate with Google Assistant / Alexa.
- Add scheduling feature in Blynk.
- Create smart notification for device status.

21. CONCLUSION

This project demonstrates a simple and effective IoT based home automation system using ESP8266 and Blynk App. It allows users to control AC appliances remotely from anywhere in the world through internet. This system is low cost, easy to build and very useful in daily life. The same concept can be extended to multiple appliances to create a complete smart home automation system.



PROJECT MANUAL

IoT INNOVATION AND PROTOTYPE DEVELOPMENT

USING BLYNK APP

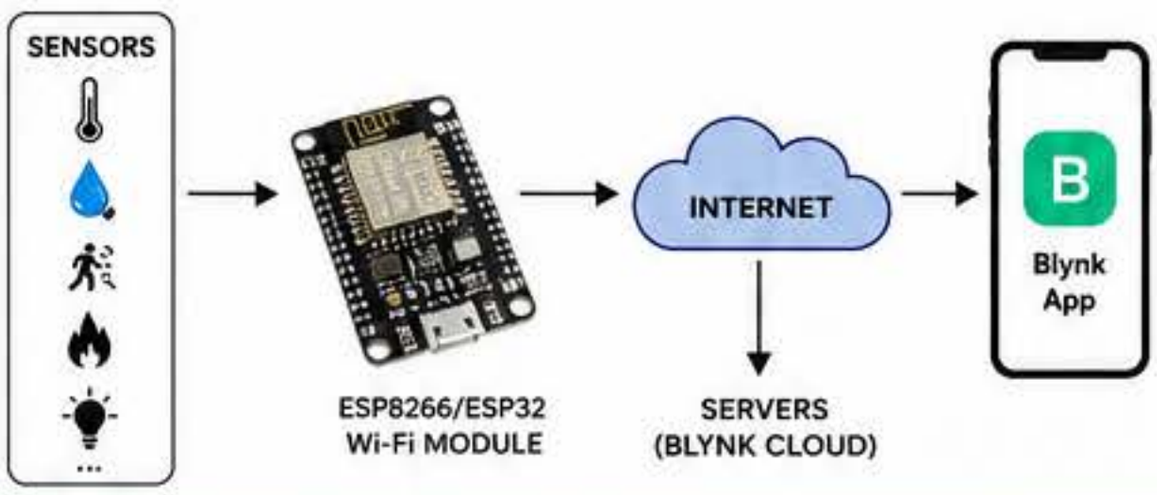
USING ESP Wi-Fi MODULE AND ARDUINO

For II B.Tech I Semester Students

ABOUT THE PROJECT MANUAL

This manual is designed to help students understand and implement basic IoT projects using ESP Wi-Fi modules, Arduino and Blynk IoT platform. Each project introduces real-time monitoring and control through the internet using sensors and actuators, building strong foundational skills in IoT innovation and prototype development.

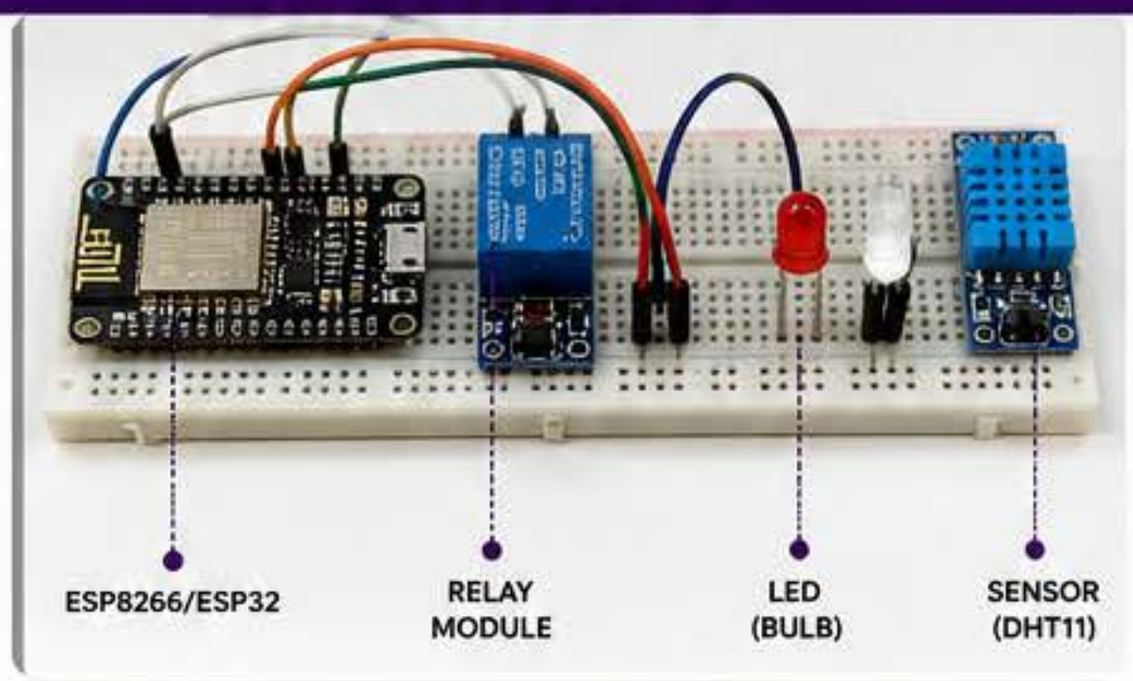
TYPICAL IoT SYSTEM ARCHITECTURE



LIST OF PROJECTS

S.No	Project Title	Main Components
1.	IoT Home Light Control Using Blynk	ESP8266/ESP32, Relay, Bulb
2.	IoT Smart Temperature Monitoring System	ESP8266/ESP32, DHT11/LM35
3.	IoT Smart Water Level Monitoring	ESP8266/ESP32, Water Level Sensor
4.	IoT Rain Alert System	ESP8266/ESP32, Rain Sensor, Buzzer
5.	IoT Gas Leakage Detection and Alert	ESP8266/ESP32, MQ-2 Sensor, Buzzer
6.	IoT Smart Door Lock Monitoring	ESP8266/ESP32, Reed Switch
7.	IoT Motion Detection and Security Alarm	ESP8266/ESP32, PIR Sensor, Buzzer
8.	IoT Smart Garden Irrigation System	ESP8266/ESP32, Soil Moisture, Relay, Pump
9.	IoT Smart Street Light System	ESP8266/ESP32, LDR, Relay, LED
10.	IoT Weather Monitoring Station	ESP8266/ESP32, DHT11, LDR, Rain Sensor
11.	IoT Smart Energy Meter Monitoring	ESP8266/ESP32, ACS712 Current Sensor
12.	IoT Fire Detection and Alert System	ESP8266/ESP32, Flame Sensor, Buzzer
13.	IoT Smart Parking Slot Monitoring	ESP8266/ESP32, IR Sensor, LEDs
14.	IoT Smart Plant Health Monitoring	ESP8266/ESP32, Soil Moisture, DHT11
15.	IoT Patient Health Monitoring System	ESP8266/ESP32, Pulse Sensor, LM35

TYPICAL HARDWARE SETUP



SOFTWARE REQUIREMENTS

- Arduino IDE
- Blynk IoT App (Android / iOS)
- Blynk Cloud Platform
- ESP8266/ESP32 Board Package
- USB Driver (CH340 / CP2102)

Blynk IoT

OBJECTIVES

- To understand IoT fundamentals and architecture.
- To interface sensors and actuators with ESP Wi-Fi modules.
- To send and receive data through internet using Blynk platform.
- To develop real-time monitoring and control applications.
- To build innovative IoT prototypes for real-world problems.

HARDWARE REQUIREMENTS

- ESP8266 NodeMCU / ESP32 Dev Board
- Arduino Uno (optional)
- Breadboard and Jumper Wires
- Relay Module
- LEDs, Buzzer
- Power Supply (5V / 12V)
- DHT11 / DHT22 Sensor
- LM35 Temperature Sensor
- PIR Motion Sensor
- LDR Sensor
- MQ-2 Gas Sensor
- Other Sensors as per projects

METHODOLOGY

TABLE OF CONTENTS

PREPARED BY

Dr V PURANDHAR REDDY

Head of the Department

Department of Electronics and Communication Engineering

PROJECT-2 MANUAL

IoT SMART TEMPERATURE MONITORING SYSTEM

USING BLYNK APP

USING ESP8266/ESP32, DHT11/LM35 SENSOR AND ARDUINO IDE

1. AIM

To design and implement an IoT based smart temperature monitoring system that measures ambient temperature using DHT11/LM35 sensor and displays the real-time temperature on Blynk App.

2. OBJECTIVES

- Understand IoT based monitoring.
- Interface DHT11/LM35 with ESP8266/ESP32.
- Send temperature data to Blynk Cloud.
- Display temperature on mobile app in real-time.
- Learn sensor interfacing and data visualization.
- Build a scalable IoT monitoring prototype.

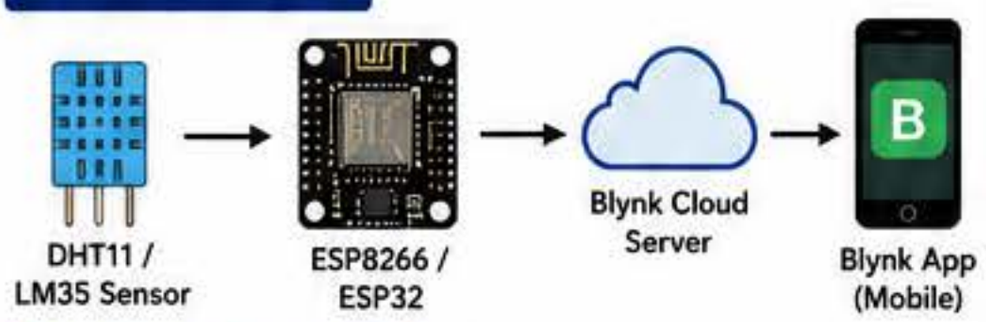
3. COMPONENTS REQUIRED

S.No	Component	Quantity
1	ESP8266 NodeMCU / ESP32	1
2	DHT11 / LM35 Temperature Sensor	1
3	Breadboard	1
4	Jumper Wires	Several
5	USB Cable	1
6	Wi-Fi Internet Connection	1
7	Blynk Mobile App	1

4. THEORY

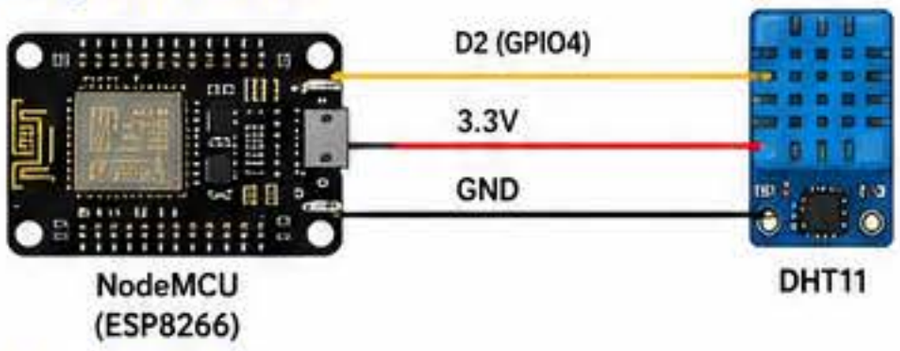
The DHT11 or LM35 sensor senses the surrounding temperature and sends analog/digital output to the ESP8266/ESP32. The microcontroller reads the temperature values and sends them to the Blynk Cloud server using Wi-Fi. The Blynk app then receives the data and displays the temperature in real-time on the mobile dashboard using a Gauge widget.

5. BLOCK DIAGRAM

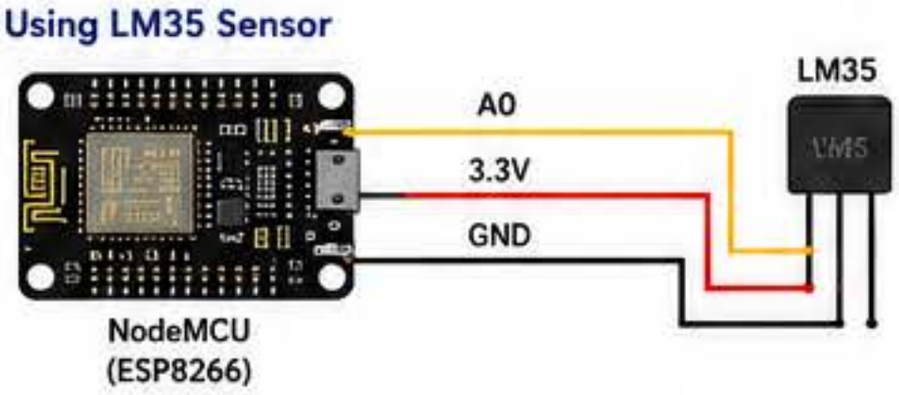


6. CIRCUIT DIAGRAM

Using DHT11 Sensor



Using LM35 Sensor



DHT11 Connections	
ESP8266	DHT11
D2 (GPIO4)	DATA
3.3V	VCC
GND	GND

LM35 Connections	
ESP8266	LM35
A0	OUT
3.3V	VCC
GND	GND

7. WORKING PRINCIPLE

1. ESP8266 connects to Wi-Fi.
2. Sensor reads the temperature.
3. Temperature value is processed.
4. Data is sent to Blynk Cloud.
5. Blynk App receives data and shows on Gauge.
6. Data updates in real-time.

8. SOFTWARE REQUIREMENTS

- Arduino IDE
- ESP8266 Board Package
- Blynk IoT Library
- Blynk Mobile App
- Wi-Fi Network

9. BLYNK DASHBOARD SETUP

Step 1: Create Template

Template Name: **Temp Monitor**

Hardware: **ESP8266**

Connection: **Wi-Fi**

Step 2: Create Datastream

Virtual Pin: **V0**

Data Type: **Double**

Units: **°C**

Min: **-40**

Max: **80**

Step 3: Add Widget

Gauge

Datastream: **V0**
Min: **-40**
Max: **80**

Step 4: Copy Credentials

Copy the following from Blynk Console:

- BLYNK_TEMPLATE_ID
- BLYNK_TEMPLATE_NAME
- BLYNK_AUTH_TOKEN

Paste them in Arduino Code.

10. ARDUINO PROGRAM (Using DHT11)

```
#define BLYNK_TEMPLATE_ID     "YOUR_TEMPLATE_ID"
#define BLYNK_TEMPLATE_NAME   "Temp Monitor"
#define BLYNK_AUTH_TOKEN      "YOUR_AUTH_TOKEN"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

char ssid[] =  "YOUR_WIFI_NAME";
char pass[] =  "YOUR_WIFI_PASSWORD";

#define DHTPIN D2      // GPIO4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

BLYNK_WRITE(V0)
{
   // Not used in this project (data from device to app)
}

void setup()
{
   Serial.begin(115200);
   dht.begin();
   Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

void loop()
{
   float t = dht.readTemperature(); // Read temperature in °C

   if (isnan(t)) {
     Serial.println("Failed to read from DHT sensor!");
     return;
   }

   Blynk.virtualWrite(V0, t);  // Send temperature to Blynk
   delay(2000);  // Update every 2 seconds
   Blynk.run();
}
```

Symbols Used

- VCC (3.3V)
- GND
- Signal (Data)
- Analog

Note:

- Use DHT11 or LM35 only one at a time.
- Ensure stable Wi-Fi connection for real-time updates.

11. PROCEDURE (OVERVIEW)

1. Install Arduino IDE.
2. Install ESP8266 Board Package.
3. Install Blynk Library.
4. Create Blynk Template.
5. Create Datastream V0 (Double).
6. Add Gauge Widget and link to V0.
7. Connect DHT11/LM35 to ESP8266 as per circuit.
8. Enter Wi-Fi credentials, Template ID and Auth Token in code.
9. Upload code to ESP8266.
10. Open Blynk App and check temperature in Gauge.

12. BLYNK APP INTERFACE (EXAMPLE)

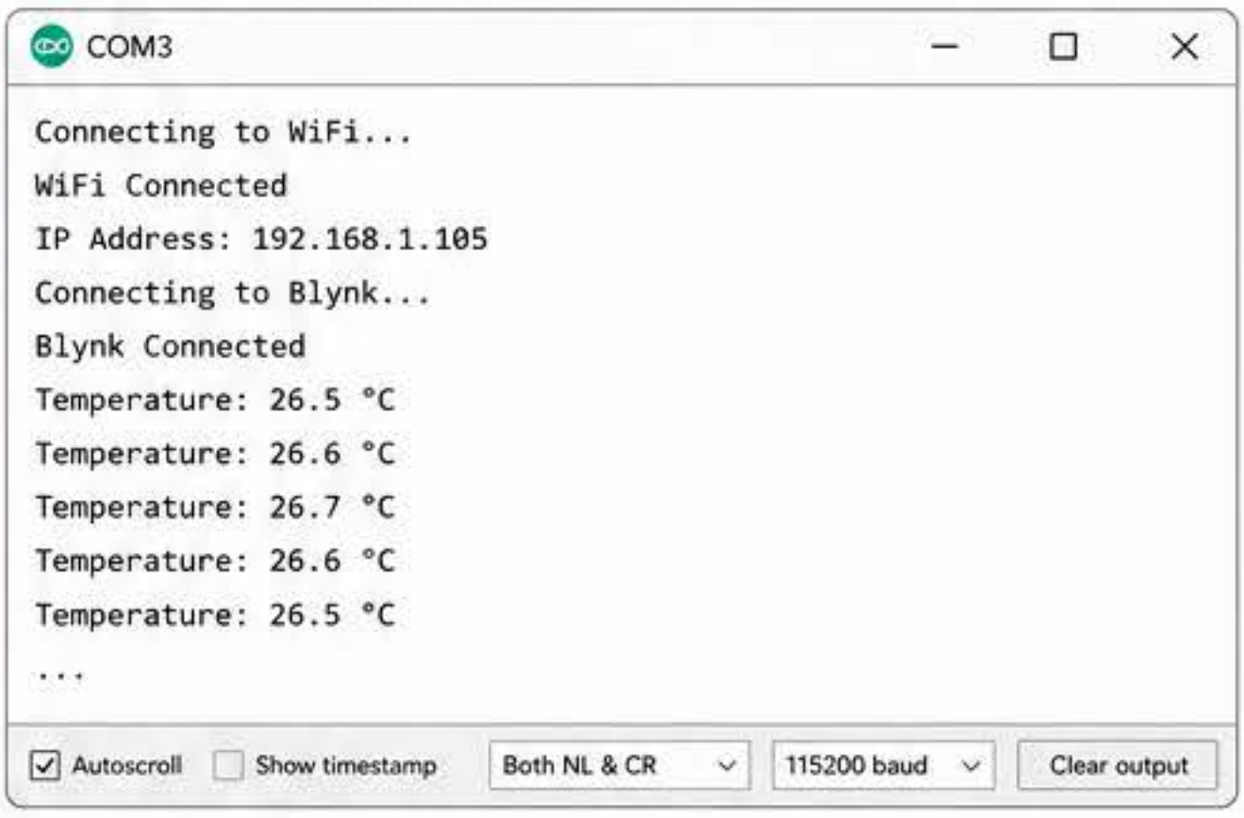
WIDGET DETAILS

Widget Type	: Gauge
Datastream	: V0
Data Type	: Double
Units	: °C
Min Value	: -40
Max Value	: 80
Decimals	: 1
Title	: Temperature
Color	: Green

NOTE:

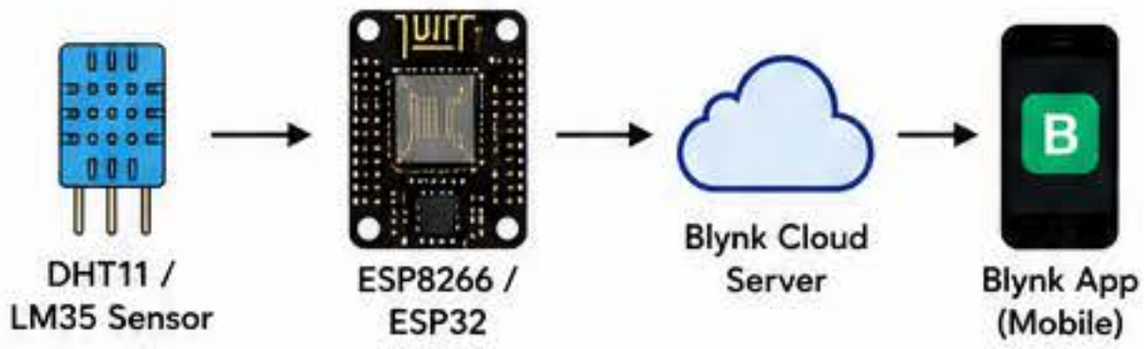
- Gauge displays real-time temperature.
- Data updates automatically every 2 seconds.
- Internet must be available for live monitoring.

13. SERIAL MONITOR OUTPUT (EXPECTED)



14. RESULT

The system successfully reads the temperature from DHT11/LM35 sensor and displays it on the Blynk App Gauge widget in real-time. The values are updated every 2 seconds. The project demonstrates IoT based temperature monitoring which can be accessed from anywhere in the world.



15. APPLICATIONS

- Smart Home Temperature Monitoring
- Office / Factory Environment Monitoring
- Greenhouse Monitoring
- Cold Storage Monitoring
- HVAC Systems
- Weather Monitoring Stations
- Remote Industrial Monitoring

16. ADVANTAGES

- Low Cost
- Real-time Monitoring
- Internet-based Access
- Easy to Install and Use
- Scalable and Flexible
- Data Visualization on Mobile
- Energy Efficient

17. PRECAUTIONS

- Use DHT11 or LM35 only one at a time.
- Ensure stable Wi-Fi connection.
- Do not exceed sensor operating voltage.
- Keep sensors away from direct heat sources.
- Use proper wiring and connections.
- Use 3.3V supply for ESP8266.

18. LEARNING OUTCOMES

After completing this project, students will be able to:

- Understand IoT based monitoring system.
- Interface DHT11/LM35 sensor with ESP8266/ESP32.
- Send sensor data to Blynk Cloud.
- Visualize real-time data on mobile dashboard.
- Develop Arduino programs for IoT applications.
- Build and deploy smart monitoring prototypes.

19. VIVA QUESTIONS

1. What is IoT?
2. What is DHT11?
3. What is LM35?
4. What is the range of DHT11?
5. What is the output of LM35?
6. What is Blynk Cloud?
7. What is a Datastream in Blynk?
8. Why is Wi-Fi required?
9. How often is data updated in this project?
10. Name two applications of this project.

20. TROUBLESHOOTING GUIDE

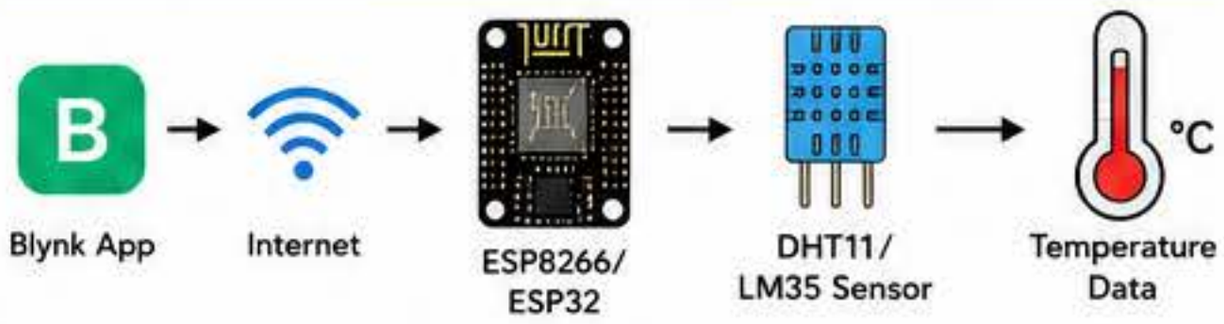
Problem	Possible Cause	Solution
ESP not connecting to Wi-Fi	Wrong Wi-Fi credentials Weak Wi-Fi signal	Check SSID & Password Ensure strong signal
Blynk not connecting	Wrong Auth Token No Internet	Check Auth Token Check Internet connection
No temperature displayed	Wrong wiring Sensor not connected	Check connections Recheck sensor wiring
Incorrect temperature reading	Sensor near heat source Improper power supply	Place sensor in normal area Use stable 3.3V supply

21. EXPANSION IDEAS

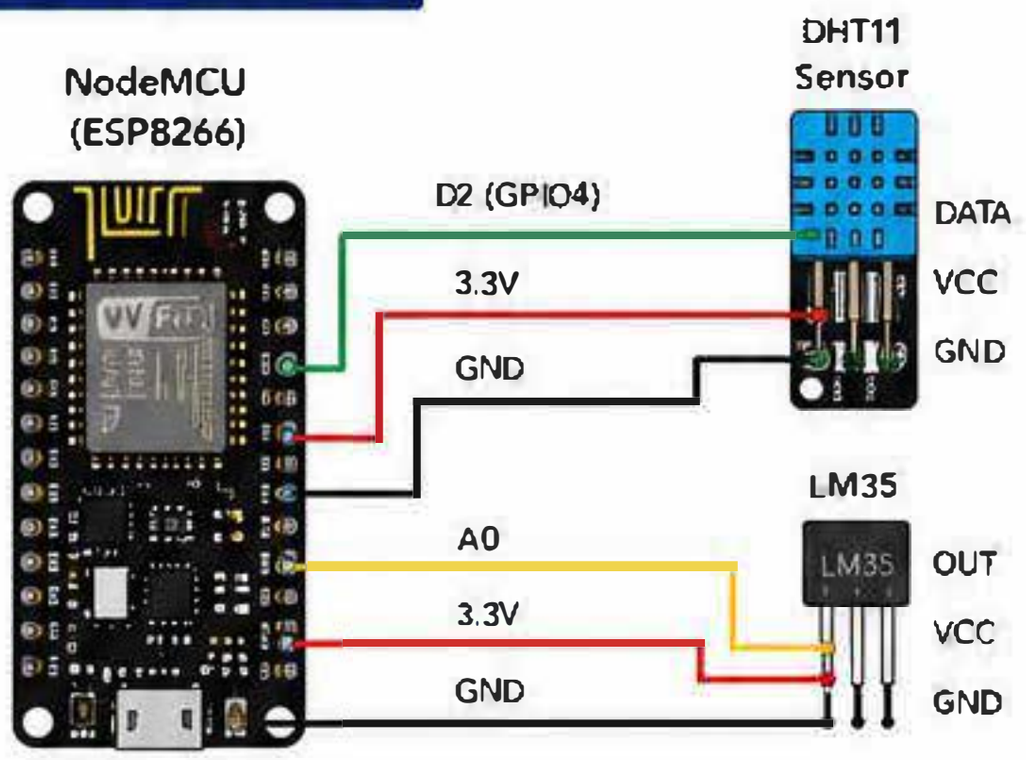
- Add Humidity monitoring using DHT11.
- Store temperature data in Google Sheets / Firebase.
- Add alert notification when temperature exceeds limit.
- Add OLED display for local display of temperature.
- Create web dashboard for advanced analytics.
- Add multiple sensors and monitoring points.

22. CONCLUSION

This project demonstrates a simple and effective IoT based temperature monitoring system using ESP8266/ESP32 and DHT11/LM35 sensor. It allows real-time monitoring of temperature from anywhere in the world using Blynk App. The system is low cost, easy to implement, and can be extended for various industrial, agricultural, and environmental applications.



23. CIRCUIT DIAGRAM



24. DHT11 & LM35 DETAILS

DHT11 Sensor

- Supply Voltage: 3.3V – 5V
- Measures Temperature & Humidity
- Temperature Range: 0 – 50°C
- Humidity Range: 20 – 90% RH
- Digital Output

LM35 Sensor

- Supply Voltage: 4V – 30V
- Temperature Range: -55°C to 150°C
- Output: 10mV / °C
- Analog Output
- High Accuracy

25. LM35 CONNECTION & OUTPUT

Pin No.	Pin Name	Description
1	VCC	+4V to +30V
2	OUT	Analog Output (10mV/°C)
3	GND	Ground

LM35 Output Calculation

Temperature (°C) = Voltage (mV) at A0 × 100

Example: If analogRead(A0) = 512

Voltage = (512 / 1023) × 3.3 = 1.652 V

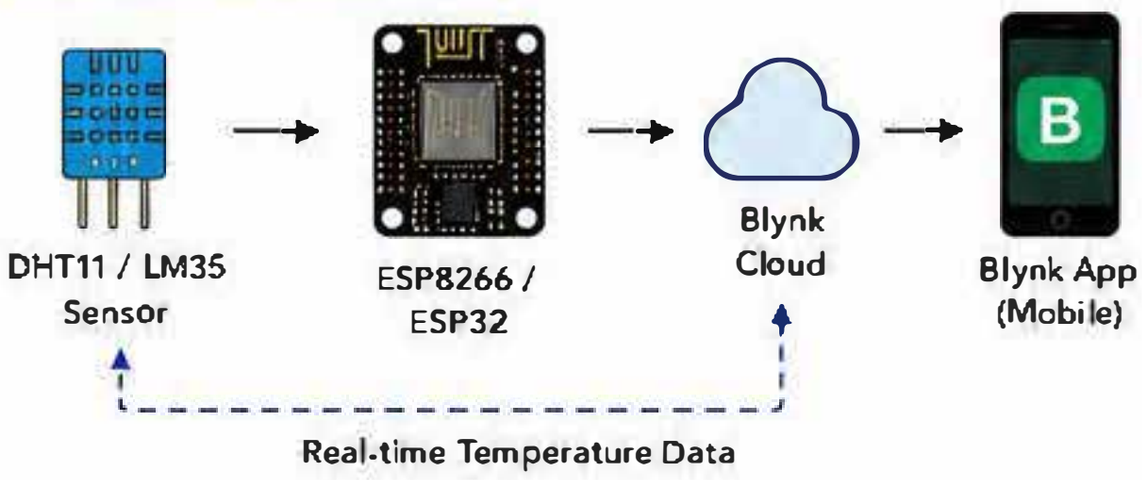
Temperature = 1.652 × 100 = 165.2 °C (Formula use in code)

26. ARDUINO CODE EXPLANATION

- Program connects ESP8266 to Wi-Fi and Blynk Cloud.
- Reads temperature from DHT11 or LM35 sensor.
- Sends data to Blynk Virtual Pin V0 every 2 seconds.
- Blynk Gauge displays temperature in realtime.

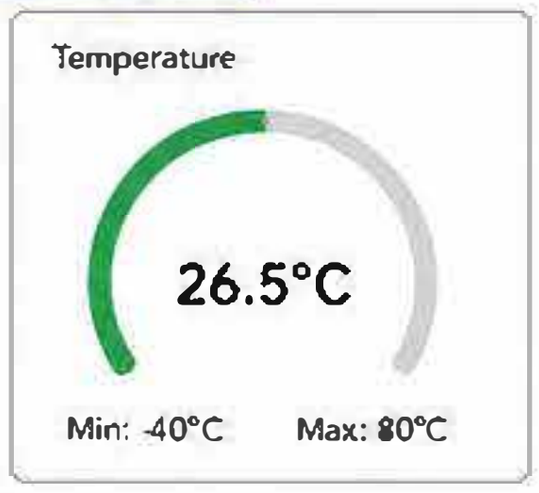
Code Section	Function
Libraries	Include ESP8266, Blynk and DHT libraries
Wi-Fi Credentials	Connects ESP8266 to Wi-Fi network
Blynk Auth Token	Connects to Blynk Cloud using Auth Token
Sensor Read	Reads temperature from DHT11 or LM35
Send to Blynk	Sends temperature to Virtual Pin V0
Delay	Updates data every 2 seconds

27. DATA FLOW DIAGRAM



28. BLYNK GAUGE CONFIGURATION GUIDE

1. Add Gauge Widget
2. Select Datastream V0
3. Set Min Value: -40
4. Set Max Value: 80
5. Units: °C
6. Title: Temperature
7. Choose Color and Save



29. PROGRAM OUTPUT ON BLYNK APP

30. SAMPLE TEMPERATURE READINGS

S.No	DHT11 Temp (°C)	LM35 Temp (°C)	Status
1	18.4	18.2	Normal
2	22.7	22.5	Normal
3	26.5	26.4	Normal
4	29.8	29.6	Normal
5	32.7	32.6	Normal
6	35.0	34.9	Normal

Note: Values may slightly vary due to sensor accuracy and environment.

31. FUTURE SCOPE

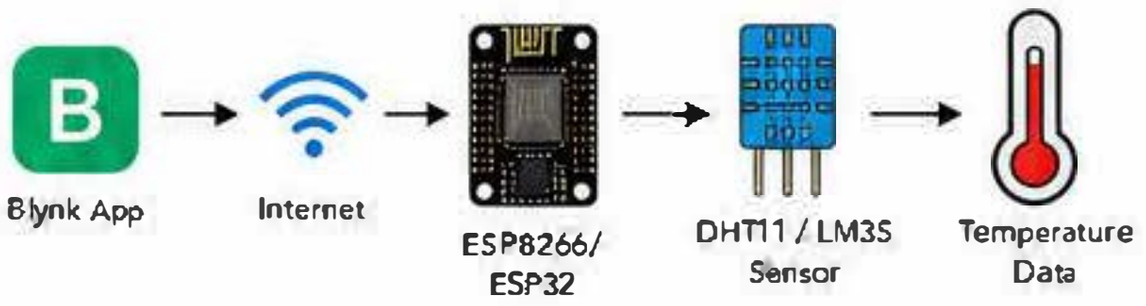
- Add humidity monitoring using DHT11.
- Store data in cloud (Google Sheets / Firebase).
- Add alert system for high temperature.
- Create web dashboard for data visualization.
- Battery powered IoT temperature logger.
- Add multiple sensors for multi-location monitoring.

32. REFERENCES

- ESP8266 Wi-Fi Module Datasheet
- DHT11 Sensor Datasheet
- LM35 Sensor Datasheet
- Blynk Documentation
- Arduino Reference
- IoT Basics – Various Online Resources

33. CONCLUSION

This project demonstrates an IoT based Temperature Monitoring System using ESP8266/ESP32 and DHT11/LM35 sensor. It successfully reads the temperature and displays real-time data on the Blynk App. The system is simple, low cost, and highly useful for home, industrial, agricultural and environmental monitoring applications.



PROJECT-3 MANUAL

IoT SMART WATER LEVEL MONITORING SYSTEM

USING BLYNK APP

USING ESP8266/ESP32, ULTRASONIC SENSOR AND ARDUINO IDE

1. AIM

To design and implement an IoT based smart water level monitoring system using an Ultrasonic sensor and display the real-time water level on Blynk App.

2. OBJECTIVES

- Understand IoT based monitoring.
- Interface Ultrasonic sensor with ESP8266/ESP32.
- Measure water level in a tank.
- Send level data to Blynk Cloud.
- Display water level on mobile app.
- Create a scalable tank monitoring system.

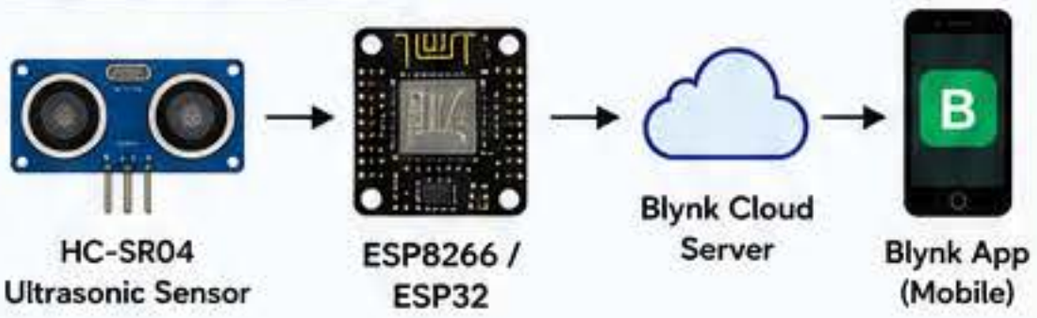
3. COMPONENTS REQUIRED

S.No	Component	Quantity
1	ESP8266 NodeMCU / ESP32	1
2	HC-SR04 Ultrasonic Sensor	1
3	Breadboard	1
4	Jumper Wires	Several
5	USB Cable	1
6	Wi-Fi Internet Connection	1
7	Blynk Mobile App	1
8	Water Tank (Demo Model)	1

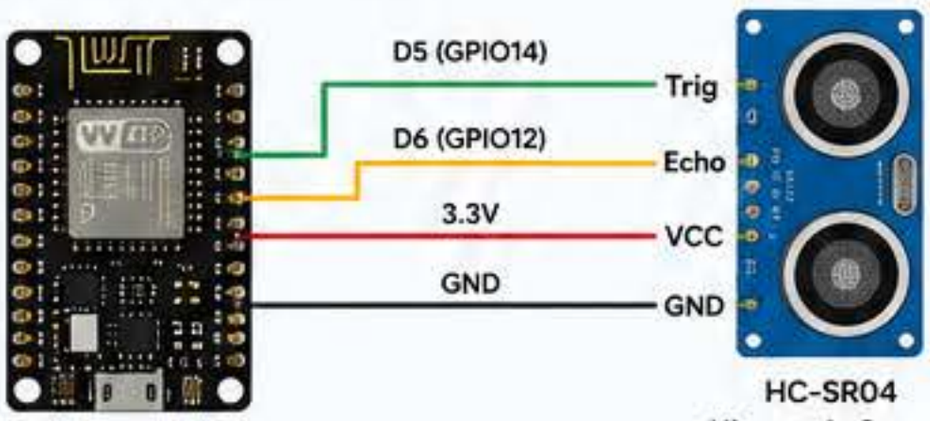
4. THEORY

The HC-SR04 Ultrasonic sensor emits ultrasonic waves and measures the time taken for the echo to return after hitting the water surface. This time is used to calculate the distance between the sensor and the water surface. The water level is then calculated based on the tank height and displayed on the Blynk App in real-time.

5. BLOCK DIAGRAM



6. CIRCUIT DIAGRAM



Ultrasonic Sensor Connections		
HC-SR04 Pin	ESP8266 Pin	Description
VCC	3.3V	Power Supply
GND	GND	Ground
Trig	D5 (GPIO14)	Trigger Pin
Echo	D6 (GPIO12)	Echo Pin

7. WORKING PRINCIPLE

1. ESP8266 connects to Wi-Fi.
2. Ultrasonic sensor measures distance to water surface.
3. Distance is converted to water level percentage.
4. Data is sent to Blynk Cloud.
5. Blynk App shows water level in real-time.
6. Data updates every 2 seconds.

8. SOFTWARE REQUIREMENTS

- Arduino IDE
- ESP8266 Board Package
- Blynk IoT Library
- Blynk Mobile App
- Wi-Fi Network

9. BLYNK DASHBOARD SETUP

Step 1: Create Template

Template Name: ✓
Water Level Monitor

Hardware:
ESP8266

Connection:
Wi-Fi

Step 2: Create Datastream

Virtual Pin: **V0**

Data Type: **Double**

Units: **%**

Min: **0**

Max: **100**

Step 3: Add Widget

Gauge Widget

Datastream: **V0**
Min: **0**
Max: **100**
Units: **%**

Step 4: Copy Credentials

Copy the following from Blynk Console:

BLYNK_TEMPLATE_ID

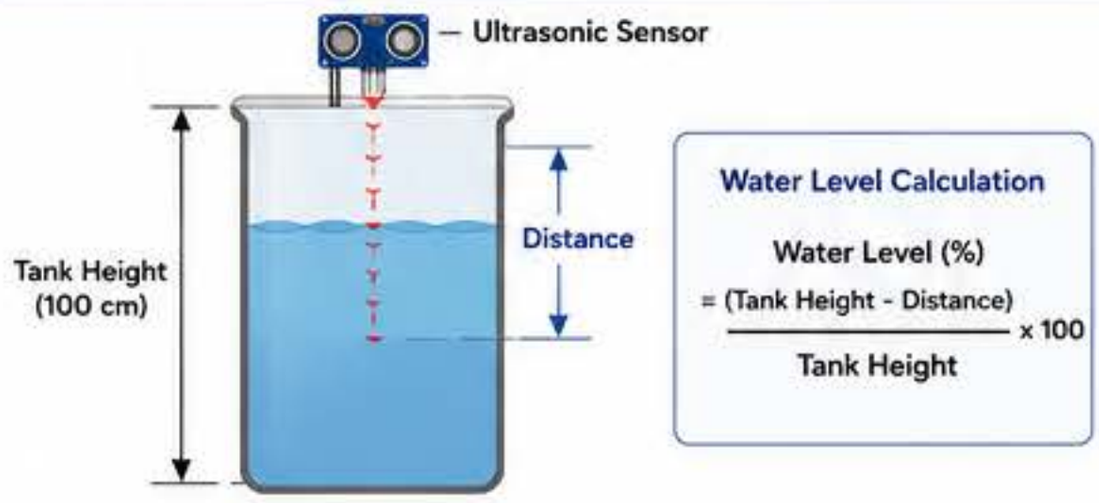
BLYNK_TEMPLATE_NAME

BLYNK_AUTH_TOKEN

Paste them in Arduino Code.

10. ARDUINO PROGRAM (Using HC-SR04)

```
#define  BLYNK_TEMPLATE_ID     "YOUR_TEMPLATE_ID"
#define  BLYNK_TEMPLATE_NAME   "Water Level Monitor"
#define  BLYNK_AUTH_TOKEN      "YOUR_AUTH_TOKEN"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_NAME";
char pass[] = "YOUR_WIFI_PASSWORD";

#define TRIG_PIN D5     // GPIO14
#define ECHO_PIN D6     // GPIO12
#define TANK_HEIGHT 100.0   // Total tank height in cm

float getDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH, 30000);
  float distance = duration * 0.034 / 2; // in cm
  return distance;
}

void setup() {
  Serial.begin(115200);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

void loop() {
  float distance = getDistance();
  float level = (TANK_HEIGHT - distance) / TANK_HEIGHT * 100.0;
  if(level < 0) level = 0;
  if(level > 100) level = 100;

  Blynk.virtualWrite(V0, level); // Send to Blynk
  delay(2000); // Update every 2 seconds
}
```



11. PROCEDURE (OVERVIEW)

1. Install Arduino IDE.
2. Install ESP8266 Board Package.
3. Install Blynk Library.
4. Create Blynk Template.
5. Create Datastream V0 (Double).
6. Add Gauge Widget and link to V0.
7. Connect HC-SR04 to ESP8266 as per circuit.
8. Enter Wi-Fi credentials, Template ID and Auth Token in code.
9. Upload code to ESP8266.
10. Open Blynk App and monitor water level.

12. BLYNK APP INTERFACE (EXAMPLE)

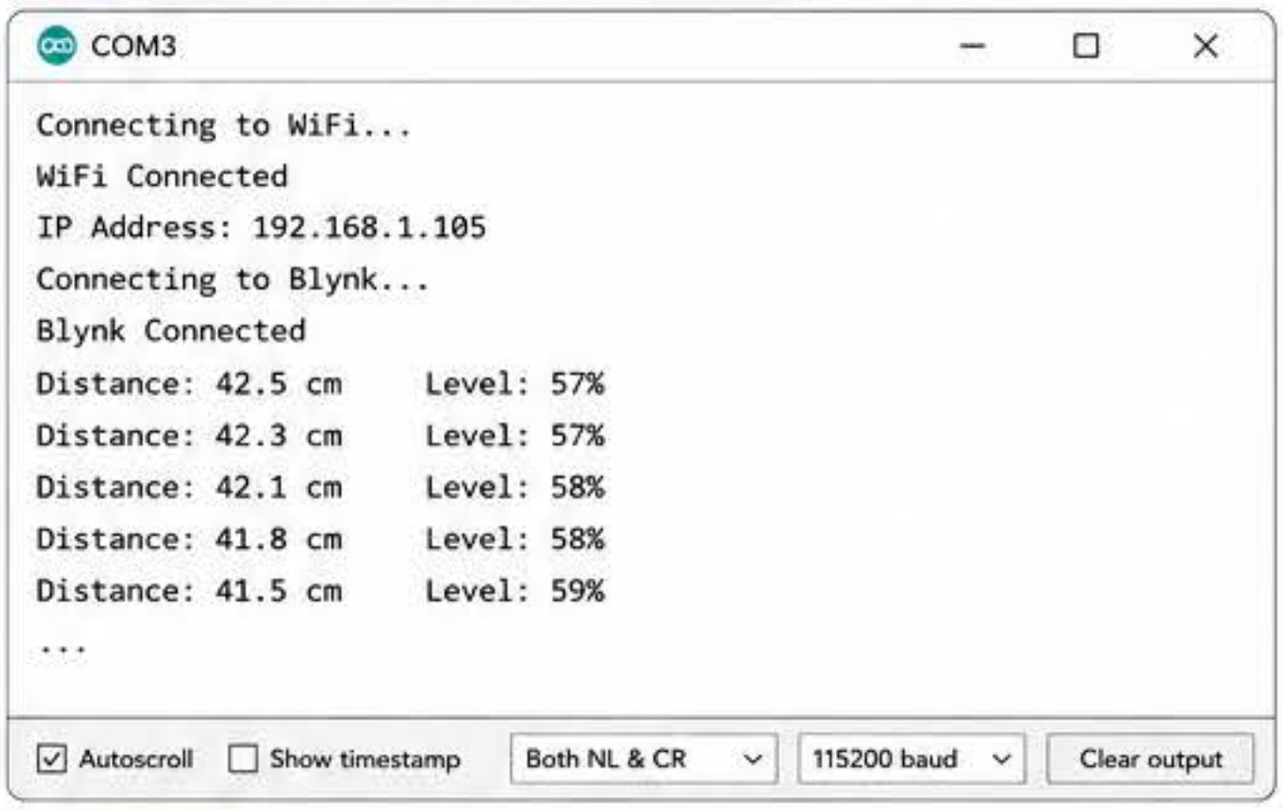
WIDGET DETAILS

Widget Type	:	Gauge
Datastream	:	V0
Data Type	:	Double
Units	:	%
Min Value	:	0
Max Value	:	100
Decimals	:	0
Title	:	Tank Level
Color	:	Blue

NOTE:

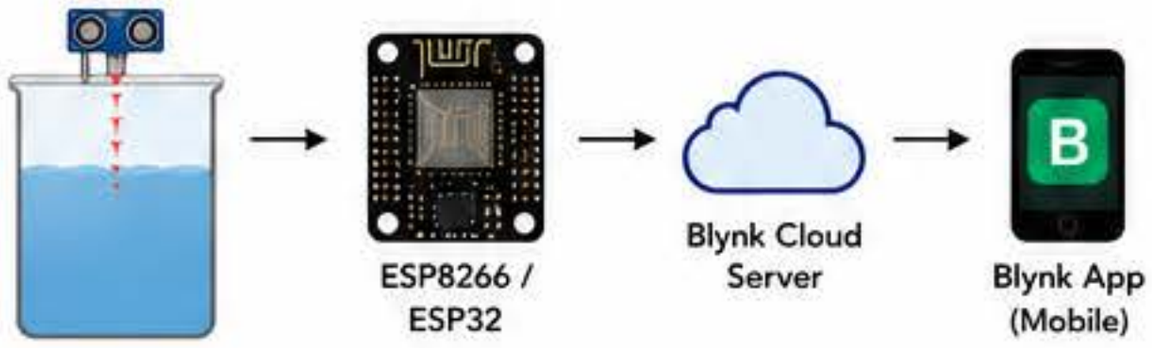
- Gauge shows water level in percentage.
- Data updates automatically every 2 seconds.
- Internet must be available for live monitoring.

13. SERIAL MONITOR OUTPUT (EXAMPLE)



14. RESULT

The system successfully measures the distance to water surface using the Ultrasonic sensor, calculates the water level in percentage, and displays it in real-time on the Blynk App Gauge widget. The data is updated every 2 seconds and can be accessed from anywhere in the world.



15. APPLICATIONS

- Overhead Water Tank Monitoring
- Underground Water Tank Monitoring
- Borewell / Sump Monitoring
- Industrial Tank Level Monitoring
- Smart Irrigation Systems
- Water Management Systems
- Home Automation
- Water Conservation Systems

16. ADVANTAGES

- Real-time Monitoring
- Low Cost
- Easy to Install
- Internet-based Access
- Reduces Water Overflow
- Prevents Water Shortage
- User Friendly
- Scalable for Multiple Tanks

17. PRECAUTIONS

- Mount sensor properly at top of tank.
- Ensure sensor is not exposed to direct sunlight/rain.
- Keep sensor surface clean and dry.
- Do not exceed maximum range of sensor (4m).
- Ensure stable Wi-Fi connection.
- Use regulated 5V or 3.3V power supply.

18. LEARNING OUTCOMES

After completing this project, students will be able to:

- Understand IoT based monitoring system.
- Interface Ultrasonic sensor with ESP8266/ESP32.
- Measure distance and calculate water level.
- Send data to Blynk Cloud and visualize on mobile.
- Develop Arduino programs for IoT applications.
- Design and implement smart water level monitoring systems.

19. VIVA QUESTIONS

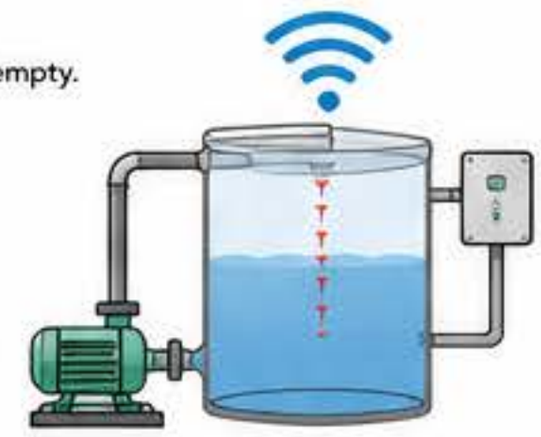
1. What is IoT?
2. What is HC-SR04?
3. How does an ultrasonic sensor work?
4. What is the range of HC-SR04?
5. How is water level calculated?
6. What is Blynk Cloud?
7. What is Datastream in Blynk?
8. Why is Wi-Fi required?
9. What is the purpose of Virtual Pin?
10. Name two applications of this project.

20. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
ESP not connecting to Wi-Fi	Wrong Wi-Fi credentials Weak Wi-Fi signal	Check SSID & Password Ensure strong signal
Blynk not connecting	Wrong Auth Token No Internet	Check Auth Taken Check Internet connection
Incorrect distance reading	Sensor not mounted properly Obstacle near sensor	Mount sensor properly Keep area clear
Water level not displayed	Wrong formula Tank height not set correctly	Check formula Set correct tank height
No data on Blynk App	Wi-Fi disconnected Datastream not correct	Check Wi-Fi connection Use correct Virtual Pin

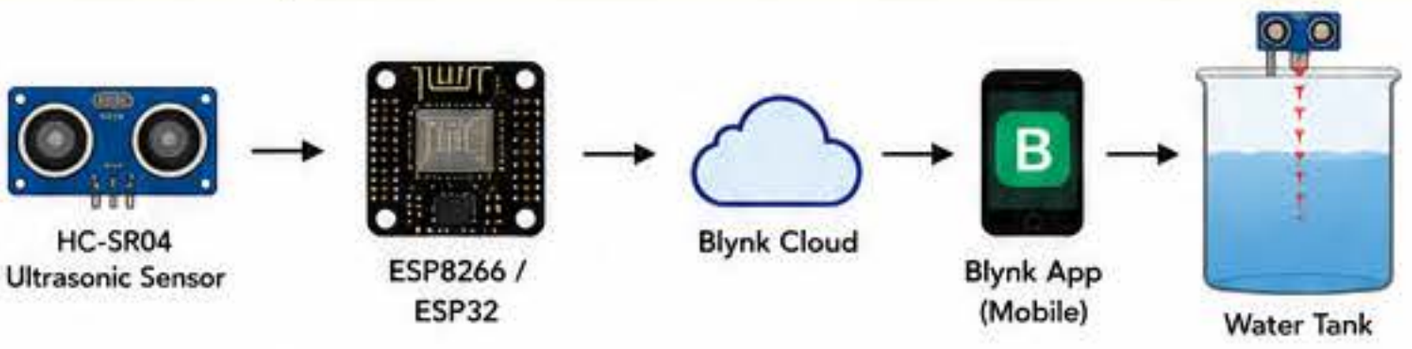
21. EXPANSION IDEAS

- Add multiple tanks monitoring.
- Add alert/notification when tank is full or empty.
- Store data in Google Sheets / Firebase.
- Add pump control based on water level.
- Add flow rate measurement.
- Create web dashboard for analytics.
- Use solar power for remote locations.
- Add historical graph and statistics.

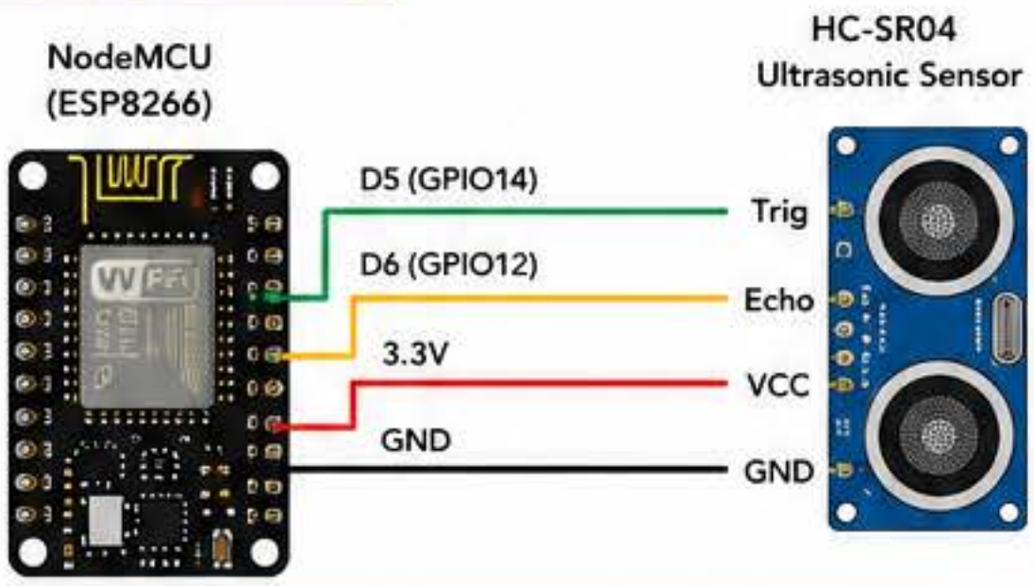


22. CONCLUSION

This project demonstrates an IoT based smart water level monitoring system using ESP8266/ESP32 and HC-SR04 Ultrasonic sensor. It measures the water level in a tank and displays it in real-time on the Blynk App. The system is low cost, easy to implement, and highly useful for water management applications in homes, industries, agriculture, and many other areas.

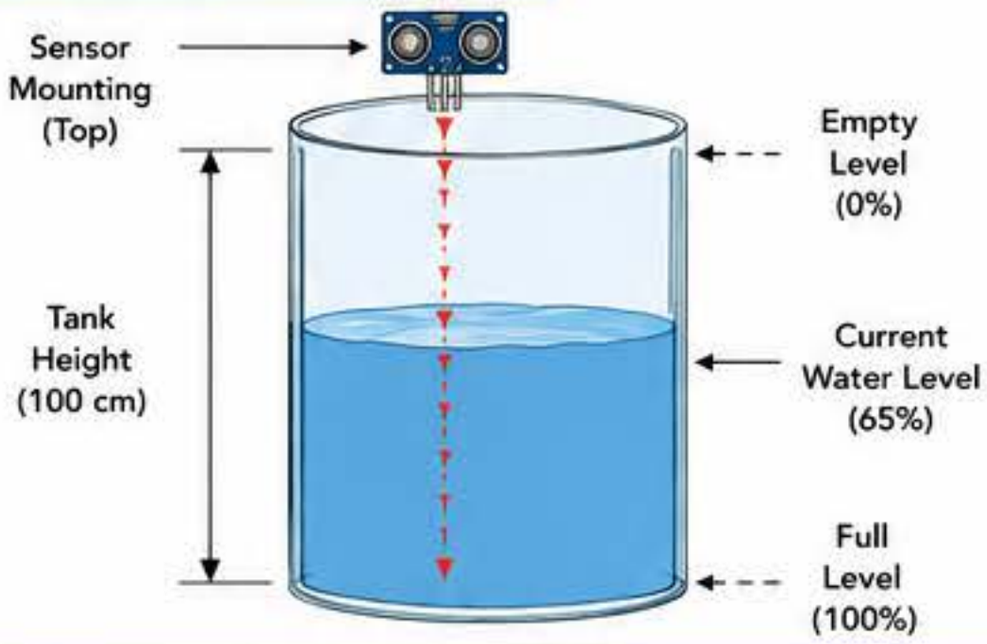


7. CIRCUIT DIAGRAM

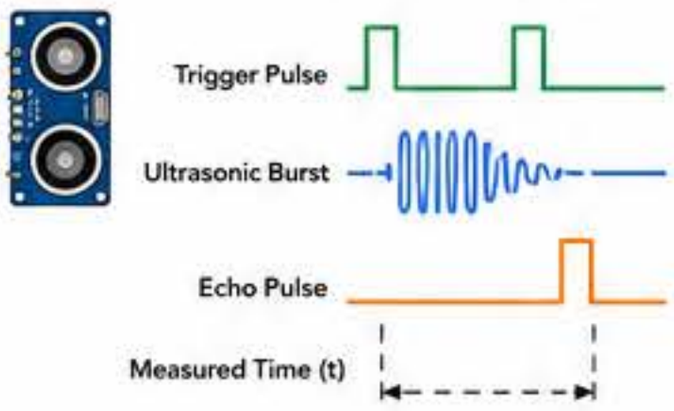


HC-SR04 Pin Description		
Pin No.	Pin Name	Description
1	VCC	+5V Power Supply
2	Trig	Trigger Input (from ESP8266 D5)
3	Echo	Echo Output (to ESP8266 D6)
4	GND	Ground

8. TANK MODEL (DEMO SETUP)



9. ULTRASONIC SENSOR WORKING



Distance Calculation

$$\text{Distance (cm)} = \frac{\text{Time (µs)} \times 0.034}{2}$$

Where,

0.034 = Speed of sound in air (cm/µs)

t = Echo time (µs)

10. WATER LEVEL CALCULATION

Formula

Distance (d) = (Time × 0.034) / 2

Water Level (cm) = Tank Height – Distance

Water Level (%) = (Water Level / Tank Height) × 100

Example:

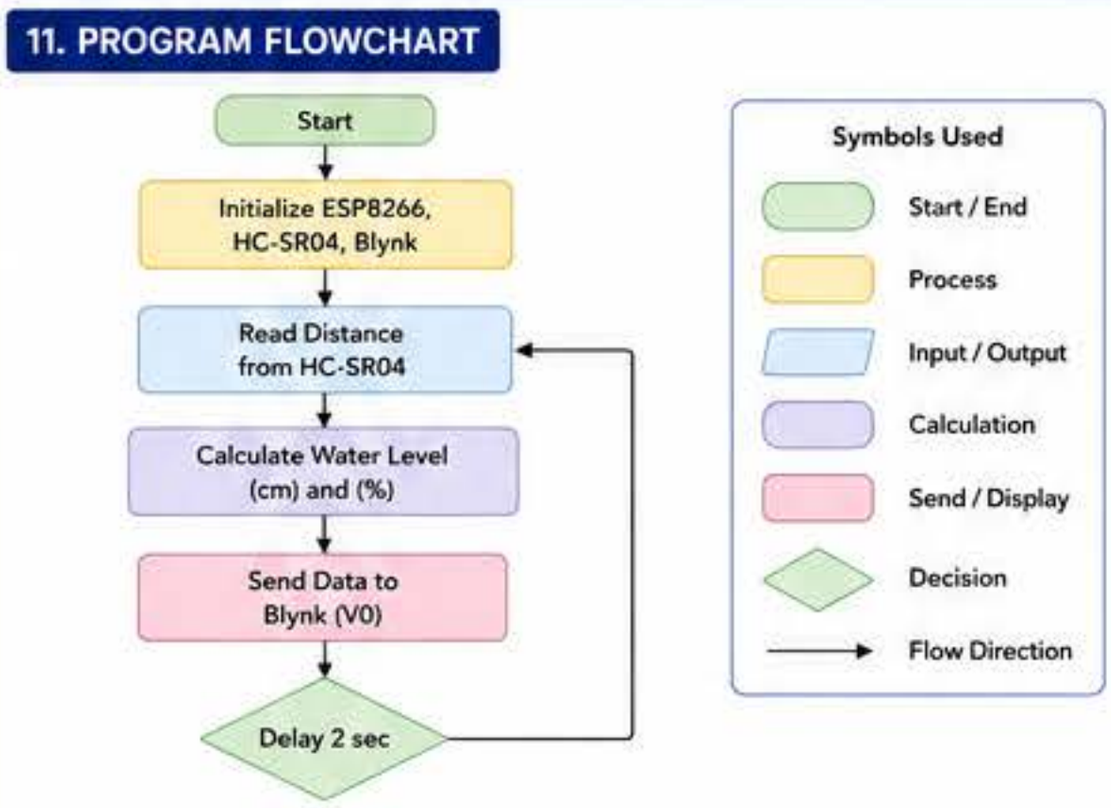
Tank Height = 100 cm

Measured Distance = 35 cm

Water Level = 100 – 35 = 65 cm

Water Level (%) = (65 / 100) × 100 = 65%

11. PROGRAM FLOWCHART



12. HC-SR04 & SPECIFICATIONS

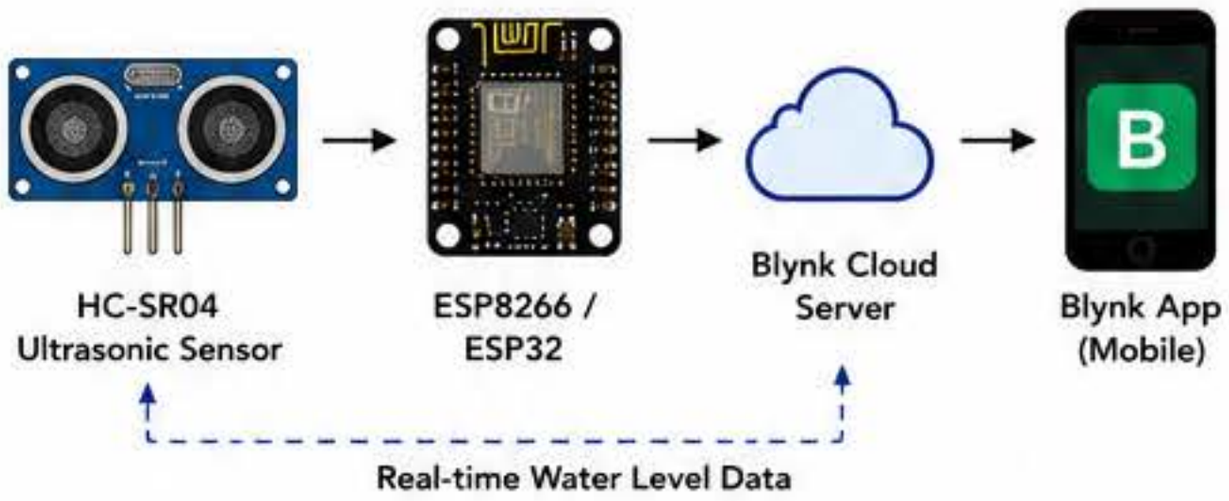
HC-SR04 Sensor

- Operating Voltage: 5V DC
- Operating Current: 15mA
- Measuring Angle: < 15°
- Max Range: 4m
- Min Range: 2cm
- Accuracy: ±3mm
- Trigger Pulse: 10µs TTL
- Echo Pulse: TTL level

13. ARDUINO CODE EXPLANATION

Code Section	Function
Include Libraries	Include ESP8266 and Blynk libraries
Wi-Fi Credentials	Connects ESP8266 to Wi-Fi
Blynk Auth Token	Connects to Blynk Cloud
Read Distance	Reads time from HC-SR04
Calculate Level	Calculates level in cm and %
Send Data	Sends data to Blynk Virtual Pin V0
Delay	Waits for 2 seconds and repeats

14. DATA FLOW DIAGRAM



15. SAMPLE READINGS

S.No.	Distance (cm)	Water Level (cm)	Water Level (%)	Status
1	15	85	85%	High
2	25	75	75%	High
3	35	65	65%	Medium
4	60	40	40%	Low
5	85	15	15%	Very Low
6	100	0	0%	Empty

Note: Tank Height = 100 cm

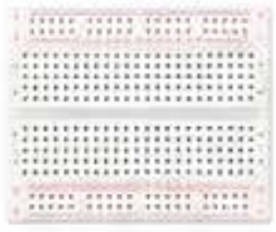
16. BLYNK GAUGE CONFIGURATION

1. Add Gauge Widget
2. Select Datastream V0
3. Set Min Value: 0
4. Set Max Value: 100
5. Units: %
6. Title: Tank Level
7. Choose Color and Save

17. COMPONENTS IMAGES

ESP8266 (NodeMCU)

HC-SR04 Ultrasonic Sensor



Breadboard

Jumper Wires

PROJECT-4 MANUAL

IOT SMART ENERGY METER

USING BLYNK APP

USING PZEM-004T SENSOR, ESP8266 AND ARDUINO IDE

1. AIM

To design and implement an IoT based smart energy meter using PZEM-004T sensor that measures voltage, current, power, energy and frequency and displays the data in real-time on Blynk App.

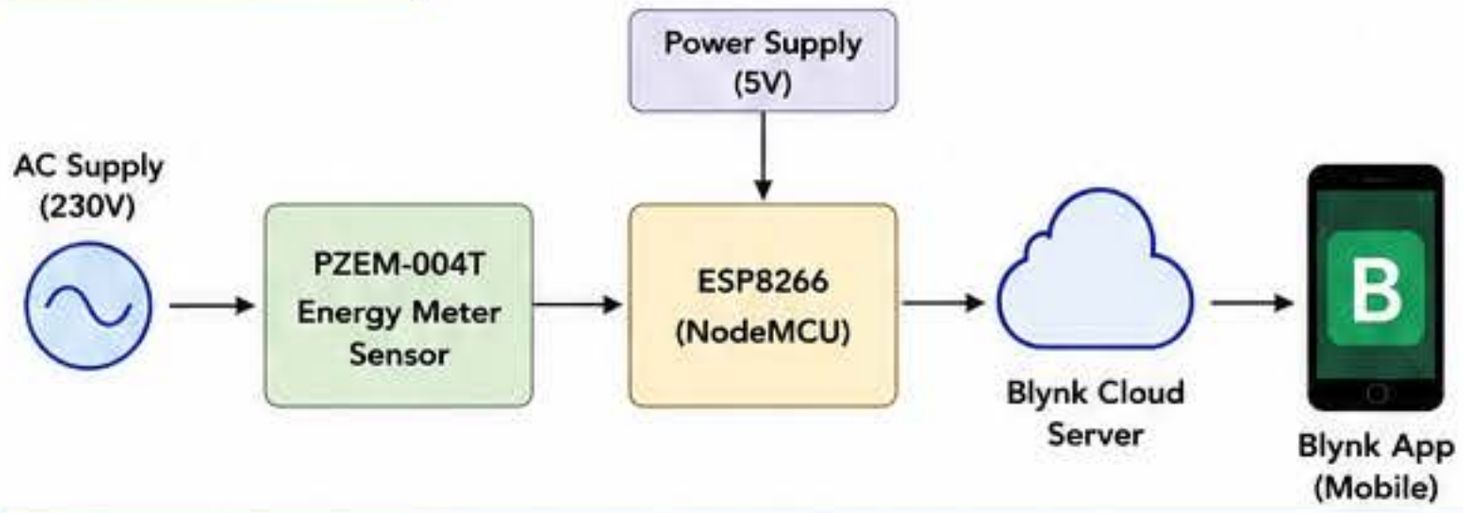
2. OBJECTIVES

- Measure voltage, current, power, energy and frequency.
- Interface PZEM-004T with ESP8266.
- Send data to Blynk Cloud.
- Display real-time parameters on Blynk App.
- Store and analyze energy consumption.
- Promote energy awareness and conservation.

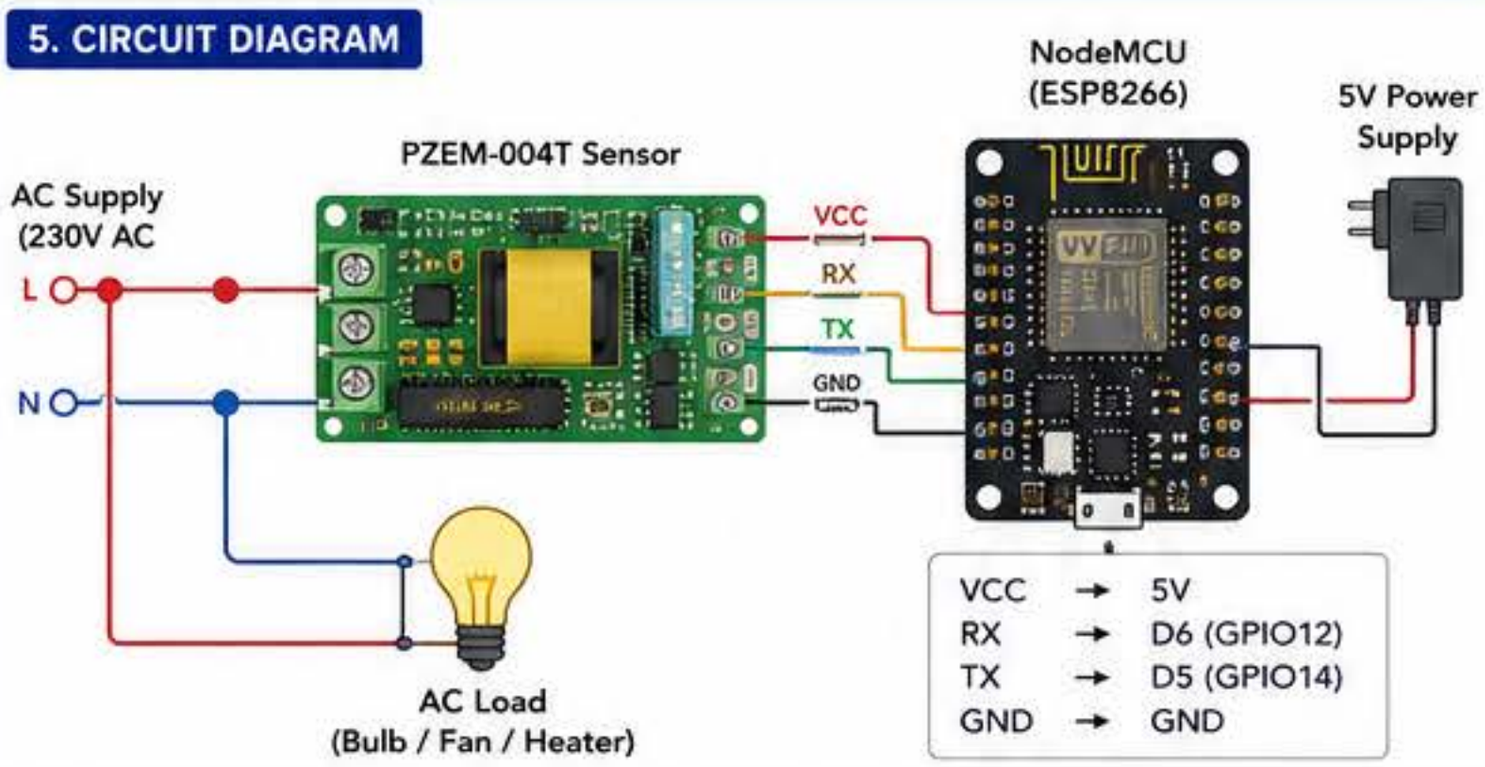
3. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	PZEM-004T Energy Meter Sensor	1
3	AC Load (Bulb / Fan / Heater)	1
4	AC Source (230V AC)	1
5	Jumper Wires	Several
6	USB Cable	1
7	Wi-Fi Internet Connection	1
8	Blynk Mobile App	1

4. BLOCK DIAGRAM



5. CIRCUIT DIAGRAM



6. THEORY

The PZEM-004T is an AC multifunction energy meter module that can measure voltage, current, power, energy, frequency and power factor. It communicates with microcontrollers using UART protocol. The ESP8266 sends the data to Blynk Cloud through Wi-Fi and the user can monitor it on the Blynk App in real-time from anywhere in the world.

7. WORKING PRINCIPLE

1. The PZEM-004T measures electrical parameters from the AC supply.
2. It sends the data to ESP8266 via UART.
3. ESP8266 connects to Wi-Fi and uploads the data to Blynk Cloud.
4. Blynk App fetches the data and displays it in real-time on gauges and value displays.
5. The user can monitor and analyze energy consumption.

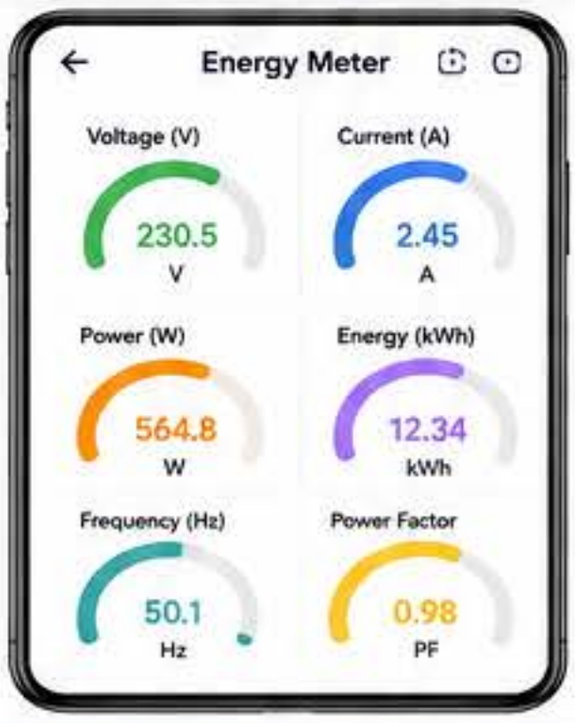
8. PZEM-004T SPECIFICATIONS

Parameter	Measurement Range
Voltage	80 – 260V AC
Current	0 – 100A AC
Power	0 – 23kW
Energy	0 – 99999 kWh
Frequency	45 – 65Hz
Power Factor	0.00 – 1.00
Accuracy	1.0%

9. FEATURES

- Measures multiple electrical parameters
- High accuracy and reliability
- Real-time monitoring from anywhere
- Data visualization on Blynk App
- Low cost and easy to implement
- Helps in energy saving
- Supports historical data storage

10. BLYNK DASHBOARD (EXAMPLE)



11. PARAMETER DESCRIPTION

Parameter	Description
Voltage (V)	Instantaneous AC voltage
Current (A)	Instantaneous AC current
Power (W)	Instantaneous power consumption
Energy (kWh)	Total energy consumed
Frequency (Hz)	Frequency of the AC supply
Power Factor	Ratio of real power to apparent power

NOTE:

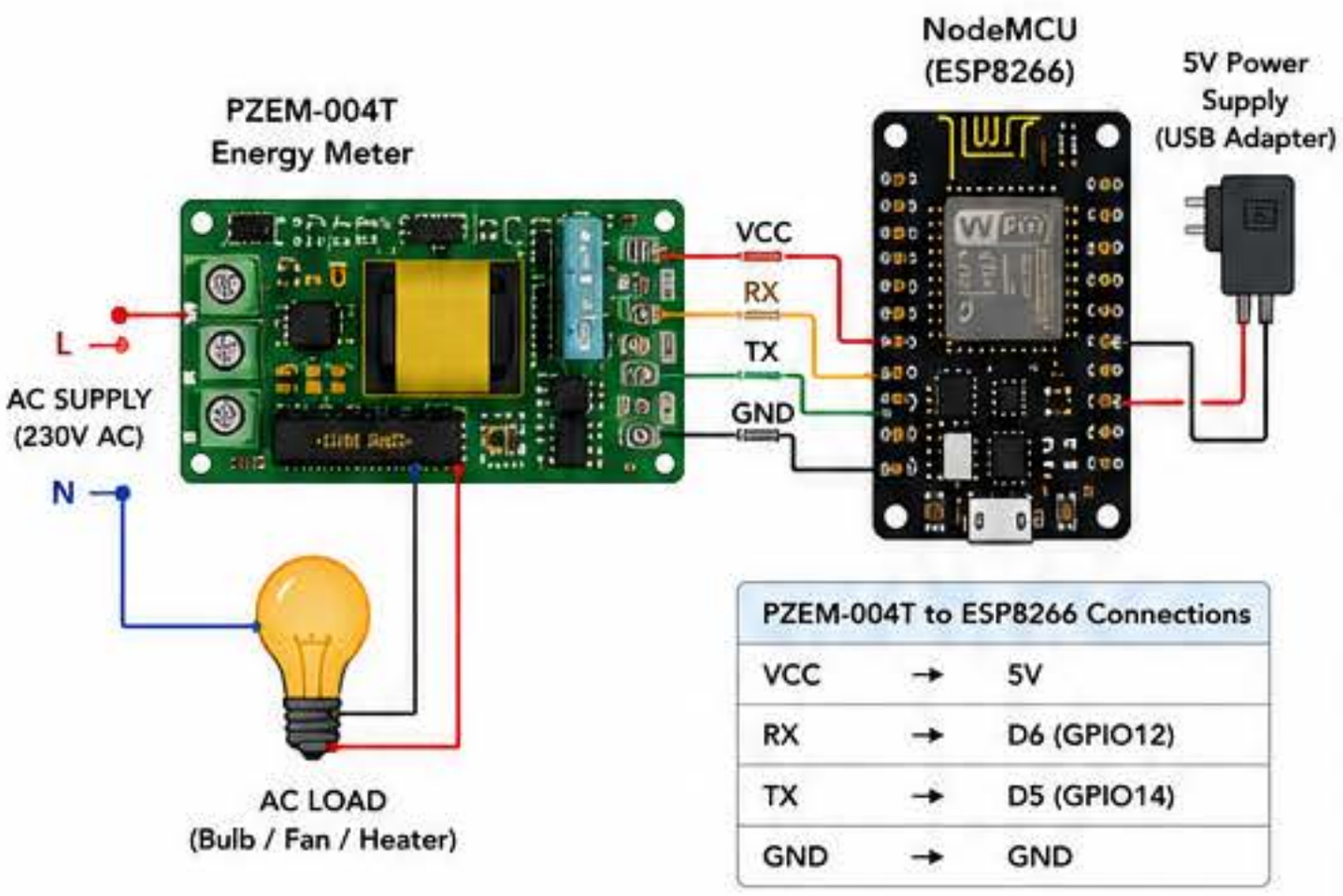
- Ensure proper connections and isolation while working with 230V AC.
- Do not touch the circuit when power is ON.
- Use enclosed box for the circuit for safety.
- Disconnect power before making any connections.

12. PROCEDURE (OVERVIEW)

1. Install Arduino IDE.
2. Install ESP8266 Board Package.
3. Install Blynk Library.
4. Wire PZEM-004T with ESP8266 as per circuit diagram.
5. Connect ESP8266 to USB and upload the code.
6. Create a new template in Blynk App.
7. Add gauges/value displays and note Virtual Pins.
8. Enter Wi-Fi credentials, Auth Token and upload code.
9. Open Blynk App and check real-time data.
10. Monitor and analyze energy consumption.

APPLICATIONS

13. WIRING CONNECTIONS



PZEM-004T to ESP8266 Connections		
VCC	→	5V
RX	→	D6 (GPIO12)
TX	→	D5 (GPIO14)
GND	→	GND

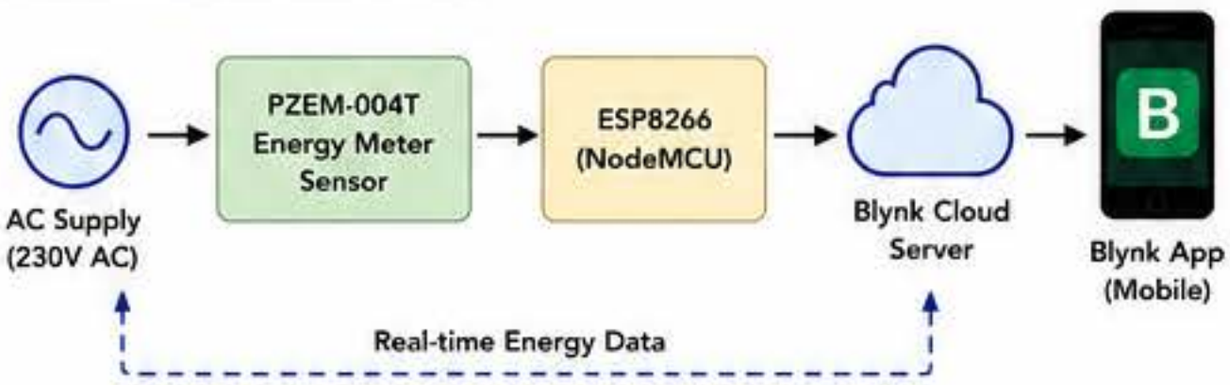
NOTE:

- Use 5V regulated adapter for ESP8266 power.
- Ensure proper isolation and safety while working with 230V AC.
- Do not touch the circuit when power is ON.
- Use proper enclosure for the circuit.

14. ARDUINO CODE (Using PZEM-004T Library)

```
#include <PZEM004Tv30.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "YOUR_BLYNK_AUTH_TOKEN";
char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

#define RX_PIN D6
#define TX_PIN D5

PZEM004Tv30 pzem(TX_PIN, RX_PIN);
BlynkTimer timer;

BLYNK_WRITE(V0) { }  // Virtual pin V0 (not used for input)

void sendData() {
  float voltage = pzem.voltage();     // Voltage (V)
  float current = pzem.current();     // Current (A)
  float power   = pzem.power();       // Power (W)
  float energy  = pzem.energy();      // Energy (kWh)
  float freq    = pzem.frequency();   // Frequency (Hz)
  float pf      = pzem.pf();          // Power Factor

  // Send data to Blynk
  Blynk.virtualWrite(V0, voltage);
  Blynk.virtualWrite(V1, current);
  Blynk.virtualWrite(V2, power);
  Blynk.virtualWrite(V3, energy);
  Blynk.virtualWrite(V4, freq);
  Blynk.virtualWrite(V5, pf);
}

void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(2000L, sendData); // Update every 2 sec
}
void loop() {
  Blynk.run();
  timer.run();
}
```

15. DATA FLOW DIAGRAM



16. BLYNK APP INTERFACE (EXAMPLE)

Widget Details	
Widget	Virtual Pin
Voltage	V0
Current	V1
Power	V2
Energy	V3
Frequency	V4
Power Factor	V5

NOTE:

- All values are updated automatically every 2 seconds.
- You can customize range, color and title for each widget.

17. SAMPLE READINGS

S.No.	Voltage (V)	Current (A)	Power (W)	Energy (kWh)	Frequency (Hz)	PF
1	229.8	1.20	276.0	5.12	50.0	0.99
2	230.1	2.10	483.0	5.14	50.1	0.98
3	230.5	2.45	564.8	5.16	50.1	0.98
4	230.7	1.85	426.9	5.18	50.0	0.99
5	229.9	0.95	218.0	5.19	50.0	1.00

Note: Readings may vary depending on the load.

18. ADVANTAGES

- Real-time monitoring of electrical parameters.
- Helps in energy conservation and cost saving.
- User can monitor from anywhere in the world.
- Data logging and historical analysis.
- Easy to install and low maintenance.
- Improves energy awareness.

19. APPLICATIONS

- Home energy monitoring.
- Industrial energy management.
- Commercial building monitoring.
- Solar energy monitoring systems.
- Smart home automation.
- Load management and analysis.

20. PRECAUTIONS

- Ensure proper wiring and connections.
- Do not touch the circuit during operation.
- Use proper enclosure for safety.
- Keep away from water and moisture.
- Use only regulated 5V power supply.
- Check all connections before powering ON.

21. TROUBLESHOOTING GUIDE

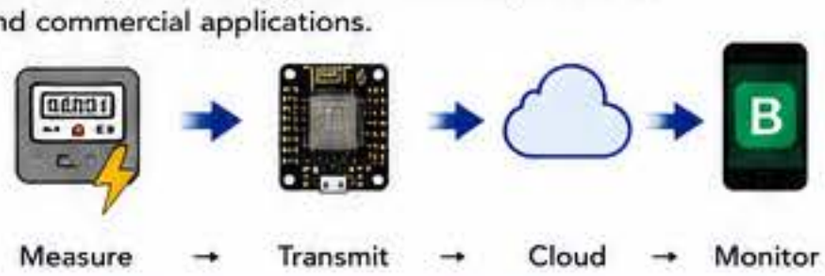
Problem	Possible Cause	Solution
No data on Blynk App	Wrong Wi-Fi credentials or internet issue	Check Wi-Fi, SSID, password and internet
Incorrect readings	Loose connections or wrong wiring	Check all connections and wiring
Sensor not detected	RX/TX pins connected incorrectly	Check RX to D6, TX to D5 connections
ESP8266 not powering	Power supply issue	Use proper 5V power supply (>=1A)
App not updating values	Blynk Auth Token wrong or expired	Check Auth Token and update code

22. EXPANSION IDEAS

- Add energy cost calculation.
- Add daily/monthly energy statistics.
- Store data in cloud (Google Sheets / Firebase).
- Add OLED display for local monitoring.
- Add relay control for load automation.
- Add alerts for high energy usage.

23. CONCLUSION

This project demonstrates an IoT based Smart Energy Meter using PZEM-004T sensor and ESP8266. It measures voltage, current, power, energy, frequency and power factor and sends the data to Blynk App in real-time. The system is simple, cost-effective, and useful for energy monitoring and management in homes, industries and commercial applications.



24. PARAMETER EXPLANATION

Parameter	Description	Unit
Voltage (V)	Instantaneous AC voltage	V
Current (A)	Instantaneous AC current	A
Power (W)	Instantaneous power consumption	W
Energy (kWh)	Total energy consumed	kWh
Frequency (Hz)	AC supply frequency	Hz
Power Factor (PF)	Ratio of real power to apparent power	–

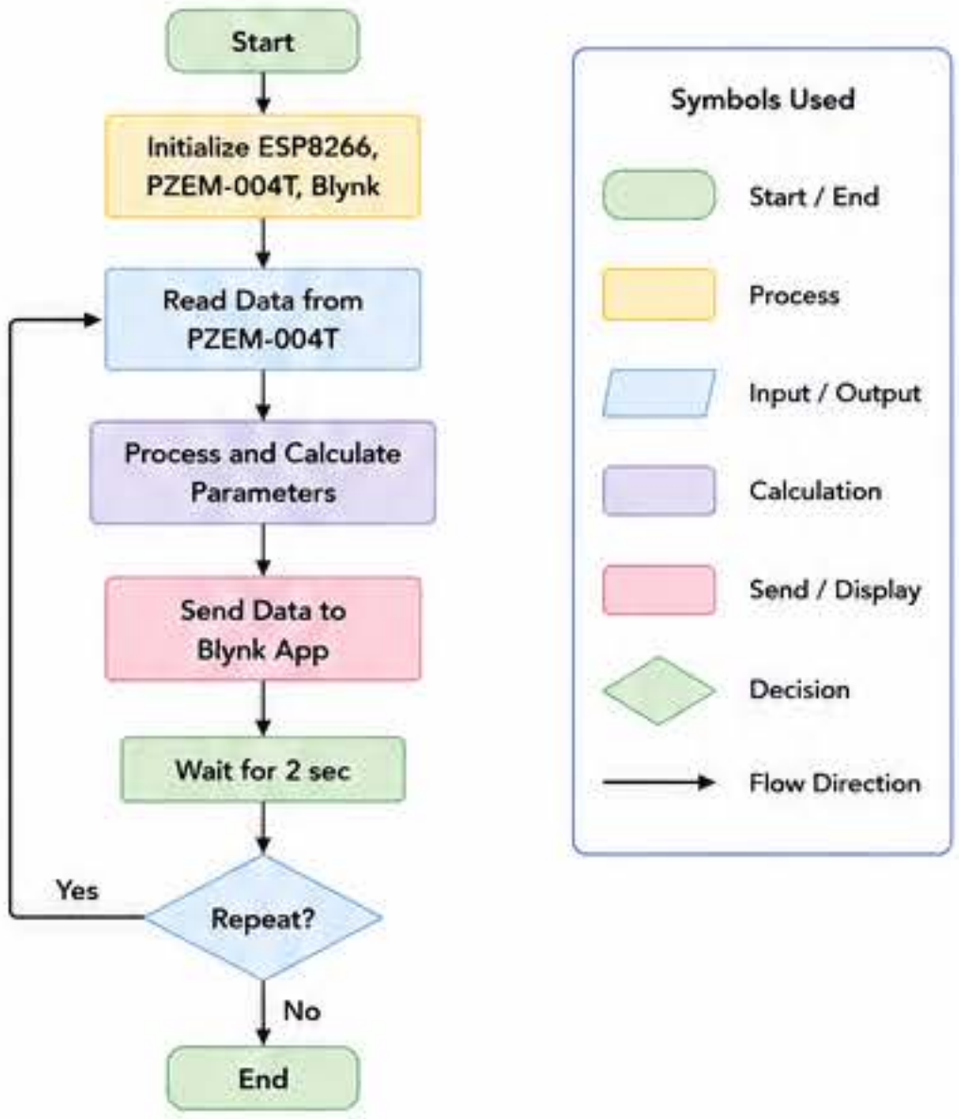
25. POWER CALCULATION

$$\text{Power (W)} = \text{Voltage (V)} \times \text{Current (A)} \times \text{Power Factor (PF)}$$

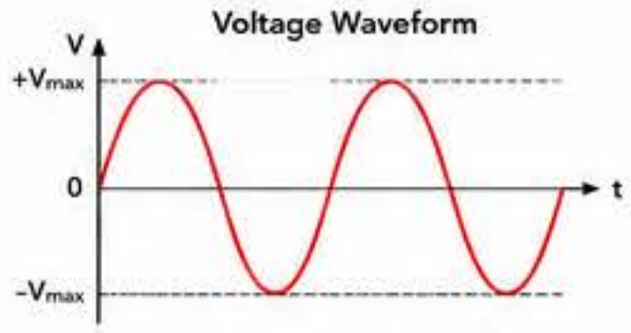
$$\text{Energy (kWh)} = \text{Power (W)} \times \frac{\text{Time (h)}}{1000}$$

$$\text{Power Factor (PF)} = \frac{\text{Real Power (W)}}{\text{Apparent Power (VA)}}$$

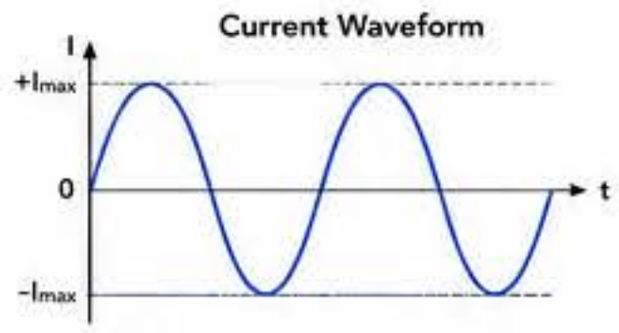
26. PROGRAM FLOWCHART



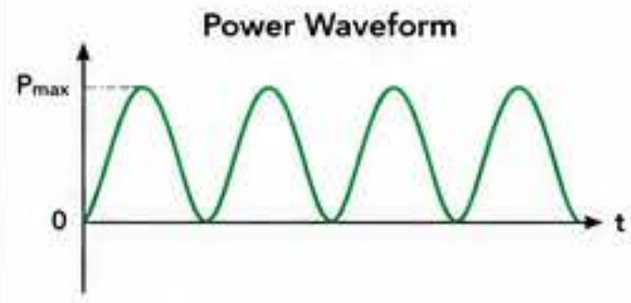
27. WAVEFORM REPRESENTATION



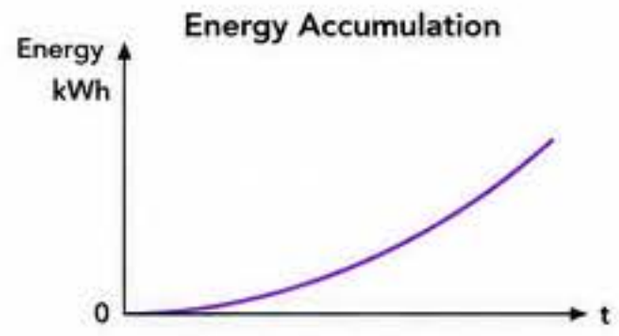
AC Voltage varies sinusoidally between +Vmax and -Vmax



AC Current varies sinusoidally between +Imax and -Imax



Instantaneous Power (P) = V × I × PF
Always positive for AC load



Energy increases over time based on power consumption

28. BLYNK VIRTUAL PIN CONFIGURATION

Parameter	Widget Type	Virtual Pin	Units
Voltage	Gauge	V0	V
Current	Gauge	V1	A
Power	Gauge	V2	W
Energy	Gauge	V3	kWh
Frequency	Gauge	V4	Hz
Power Factor	Gauge	V5	PF

29. BLYNK AUTH TOKEN

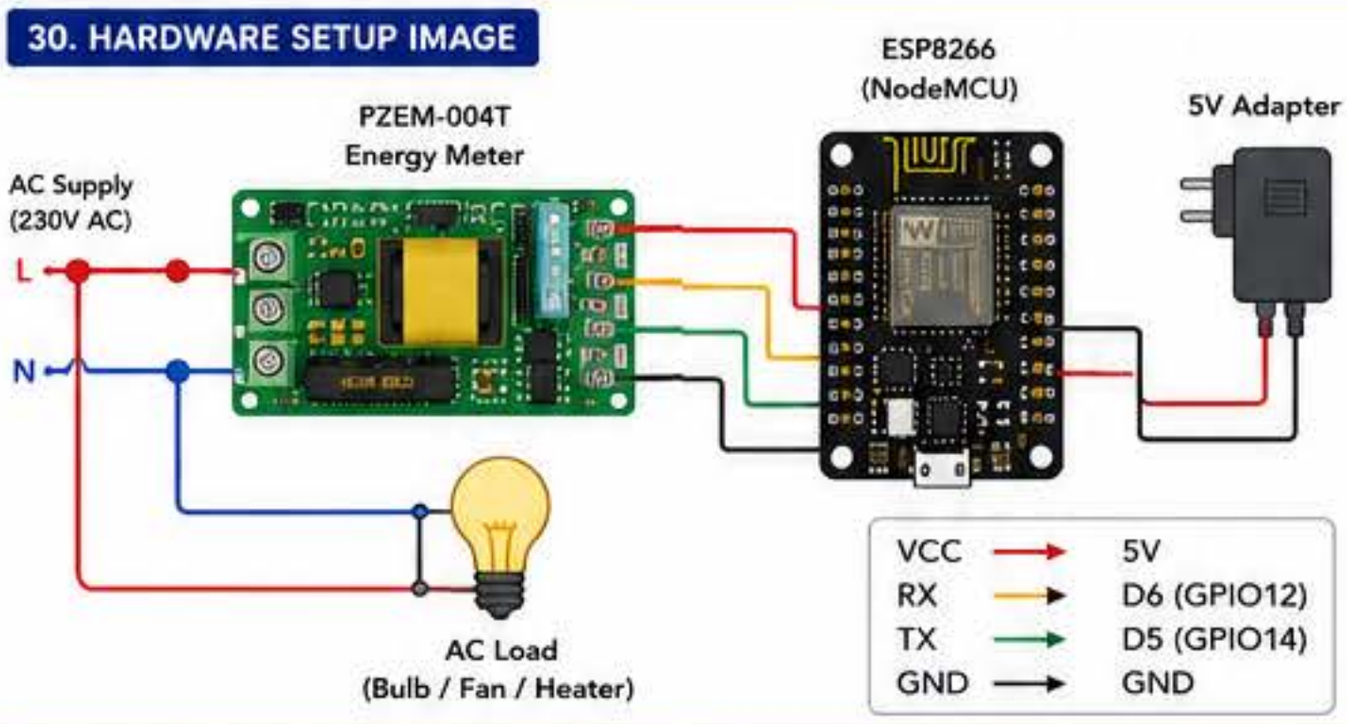
Find the Auth Token in Blynk App:

Project Settings
→ Device Info
→ Auth Token

Copy and paste this token in the Arduino code.

```
Ex: YOUR_BLYNK_AUTH_TOKEN
```

30. HARDWARE SETUP IMAGE



31. COMPONENTS USED

- ESP8266 (NodeMCU)
- PZEM-004T Energy Meter Sensor
- AC Load (Bulb / Fan / Heater)
- AC Source (230V AC)
- 5V Power Adapter
- Jumper Wires
- USB Cable

32. LEARNING OUTCOMES

After completing this project, students will be able to:

- Understand energy measurement concepts.
- Interface PZEM-004T sensor with ESP8266.
- Measure voltage, current, power, energy, frequency and power factor.
- Send real-time data to Blynk Cloud.
- Visualize and analyze data on mobile app.
- Promote energy conservation and awareness.

33. VIVA QUESTIONS

1. What is PZEM-004T?
2. What parameters can be measured using PZEM-004T?
3. How does PZEM-004T communicate with ESP8266?
4. What is the working voltage range of PZEM-004T?
5. What is power factor?
6. What is the use of Blynk App?
7. What is Virtual Pin in Blynk?
8. How is energy (kWh) calculated?
9. Why is power factor important?
10. List two applications of Smart Energy Meter.

34. APPLICATIONS

- Home Energy Monitoring
- Industrial Energy Monitoring
- Commercial Building Monitoring
- Solar Power System Monitoring
- Energy Management System (EMS)
- Smart Home Automation

35. ADVANTAGES

- ✓ Real-time monitoring of electrical parameters.
- ✓ Accurate measurement and high reliability.
- ✓ Remote monitoring from anywhere.
- ✓ Helps in energy saving and cost reduction.
- ✓ Easy installation and low maintenance.
- ✓ Data logging and historical analysis.
- ✓ Supports IoT based smart applications.

36. FUTURE SCOPE

- → Integration with Home Automation.
- → Add load control and scheduling.
- → Solav + Grid energy monitoring system.
- → AI/ML based energy consumption prediction.
- → Alerts and notifications for abnormal usage.
- → Integration with voice assistants.
- → Multi-phase energy monitoring.

NOTE:

- Ensure proper wiring and safety while working with 230V AC.
- Use regulated 5V power supply for ESP8266.
- Never touch the circuit when power is ON.
- Use proper enclosure for the circuit.

PROJECT-5 MANUAL

IOT SMART HOME AUTOMATION SYSTEM

USING BLYNK APP

USING ESP8266 (NODEMCU) AND RELAY MODULE

1. AIM

To design and implement an IoT based smart home automation system using ESP8266 that allows users to control home appliances remotely through Blynk App.

2. OBJECTIVES

- Control home appliances (Light, Fan, AC, etc.) remotely.
- Interface ESP8266 with relay module.
- Send and receive data through Blynk Cloud.
- Display real-time status of devices in the app.
- Enhance comfort, safety and energy efficiency.

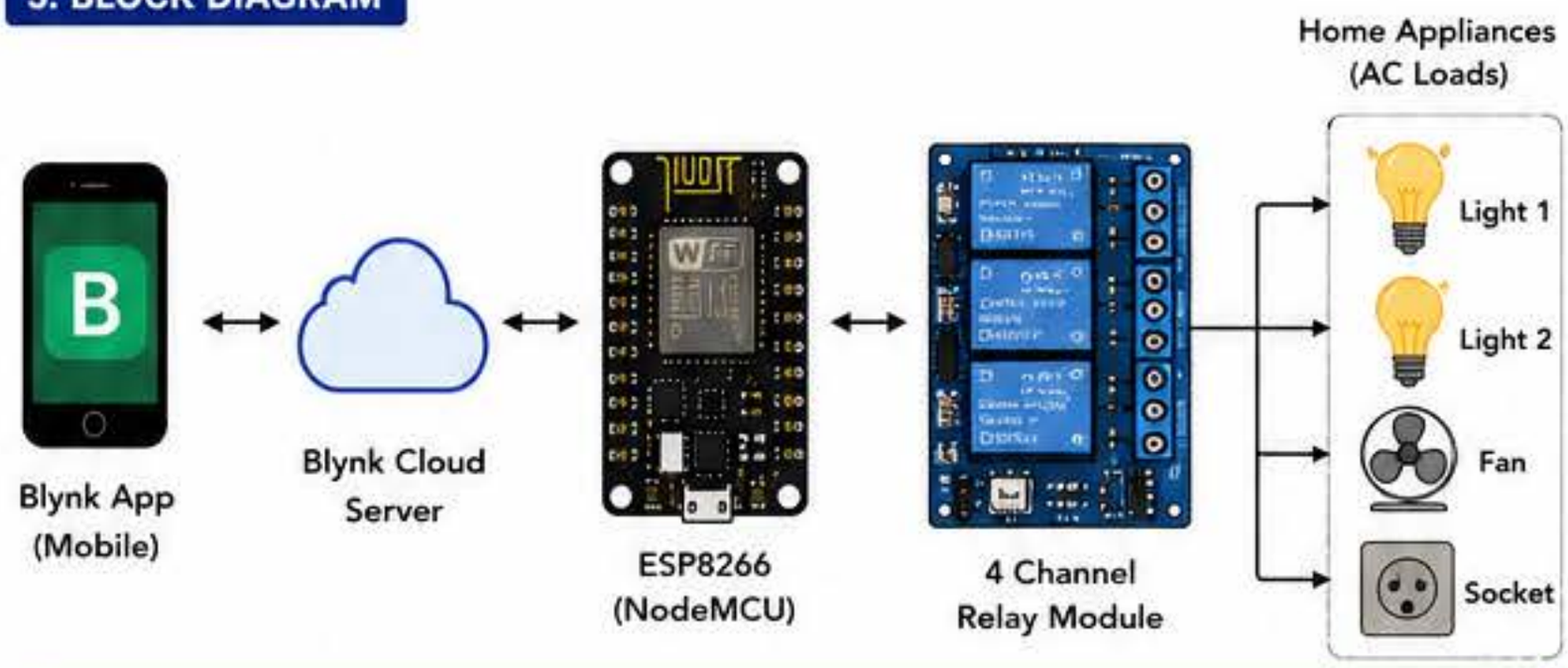
3. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	4 Channel Relay Module	1
3	AC Bulb / Lamp	2
4	AC Fan	1
5	5V Power Supply (for ESP8266)	1
6	Jumper Wires	Several
7	Wi-Fi Internet Connection	1
8	Blynk Mobile App	1

4. FEATURES

- Control multiple home appliances remotely.
- Real-time ON/OFF status on Blynk App.
- Easy to install and low cost.
- Internet based monitoring and control.
- Supports scheduling and automation (via Blynk).
- Safe, reliable and user friendly.

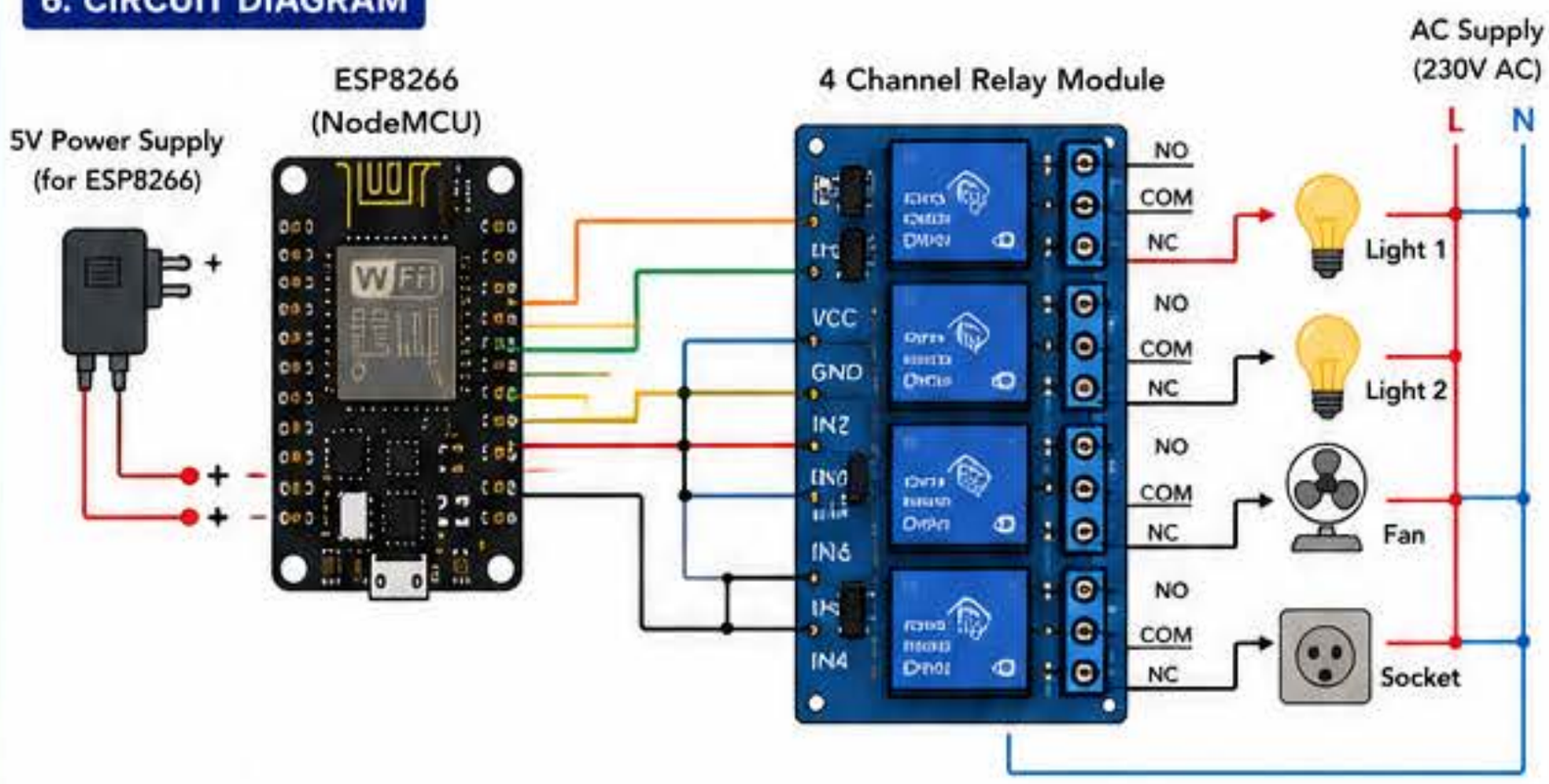
5. BLOCK DIAGRAM



WORKING:

The user controls appliances from the Blynk App. The command is sent to Blynk Cloud, which is received by ESP8266. The ESP8266 drives the relay module to switch ON/OFF the connected home appliances.

6. CIRCUIT DIAGRAM



ESP8266 (NodeMCU) to Relay Module Connections						
Relay Module Pin	VCC	GND	IN1	IN2	IN3	IN4
NodeMCU Pin	5V	GND	D1 (GPIO5)	D2 (GPIO4)	D5 (GPIO14)	D6 (GPIO12)

7. RELAY MODULE SPECIFICATIONS

- Operating Voltage : 5V DC
- Number of Channels : 4
- Trigger Level : Low Level
- Max AC Load : 250V AC / 10A
- Indicator LED for each relay
- Opto-isolation for safety

8. ESP8266 (NODEMCU) SPECIFICATIONS

- Microcontroller : Tensilica 8266
- Operating Voltage : 3.3V (Logic)
- Input Voltage : 5V (via USB)
- Wi-Fi : 2.4 GHz 802.11 b/g/n
- Digital I/O Pins : 11
- ADC : 1 (10-bit)
- Flash Memory : 4MB

9. BLYNK APP OVERVIEW

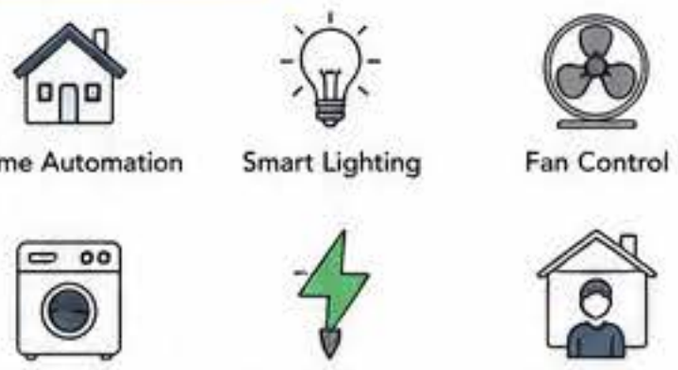
- Blynk is an IoT platform with iOS & Android apps.
- It allows you to control hardware remotely.
- Uses Internet to communicate with hardware.
- Supports widgets like Button, Switch, Slider, etc.
- Requires Auth Token for device connection.

Blynk

10. HOW IT WORKS

1. ESP8266 connects to Wi-Fi.
2. It connects to Blynk Cloud using Auth Token.
3. User presses buttons in Blynk App.
4. Commands are sent from Blynk Cloud to ESP8266.
5. ESP8266 activates the relay.
6. Appliance turns ON or OFF.
7. Status is updated in the Blynk App.

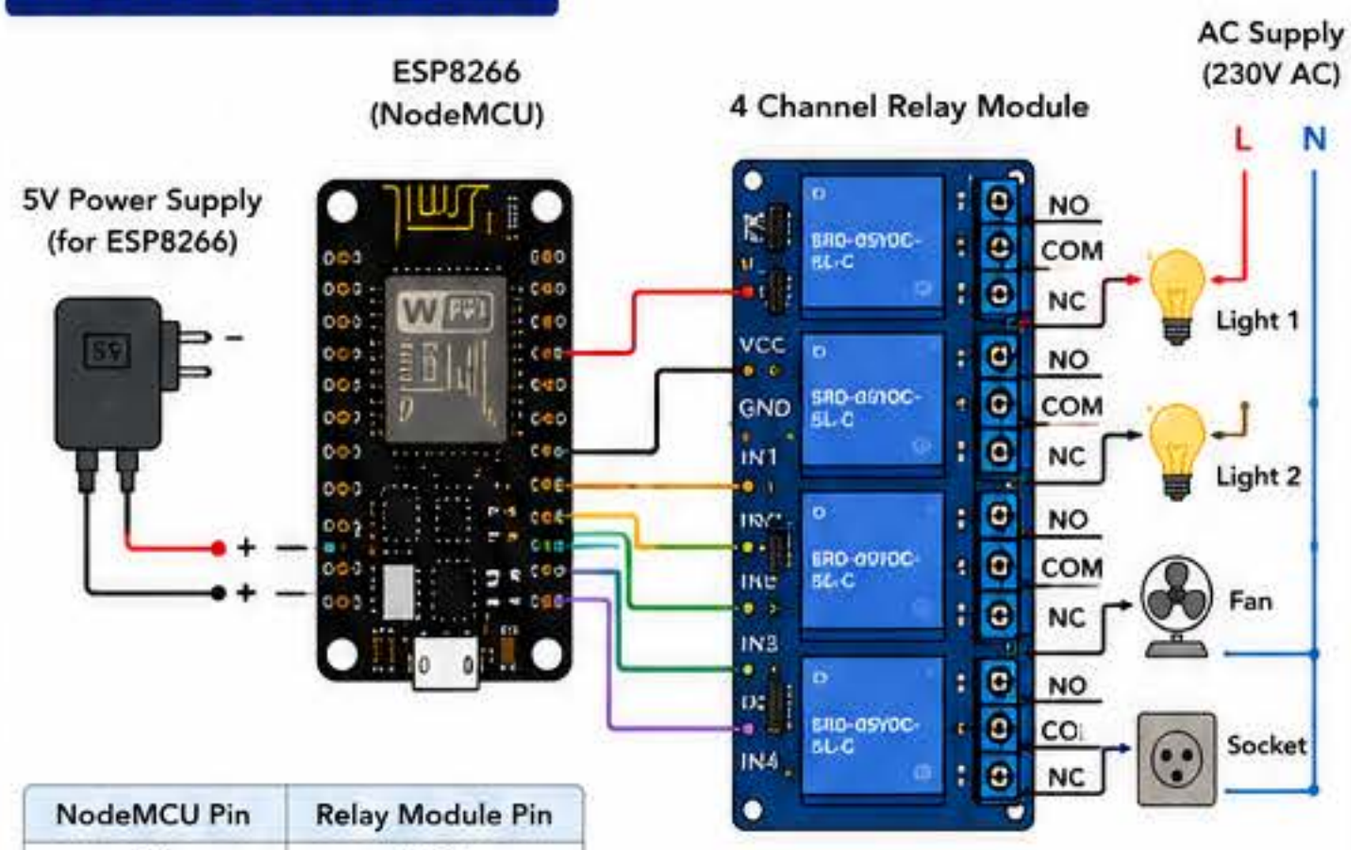
11. APPLICATIONS



12. ADVANTAGES

- ✓ Remote control of home appliances.
- ✓ Low cost and easy to implement.
- ✓ Reduces power consumption.
- ✓ Enhances convenience and comfort.
- ✓ Can be extended for more devices.

13. WIRING CONNECTIONS



NodeMCU Pin	Relay Module Pin
5V	VCC
GND	GND
D1 (GPIO5)	IN1
D2 (GPIO4)	IN2
D5 (GPIO14)	IN3
D6 (GPIO12)	IN4

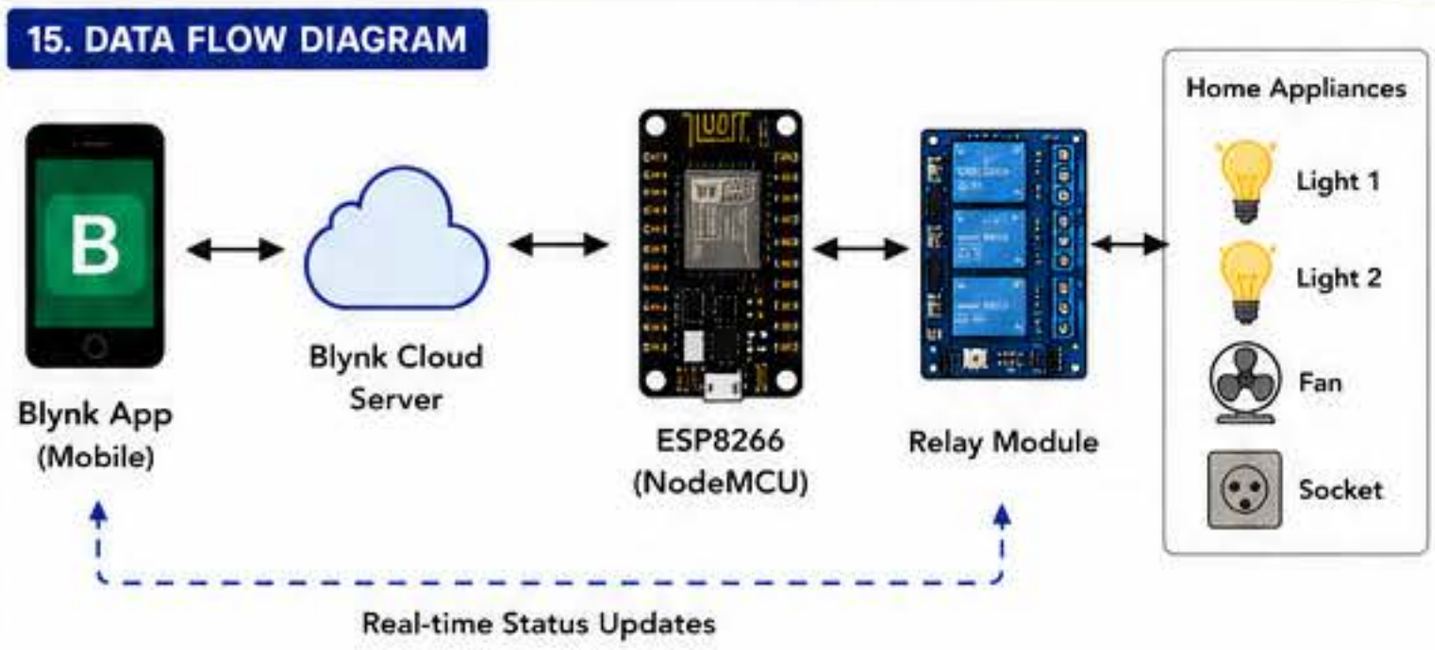
NOTE:

- All relay COM terminals are connected to Live (L).
- NO (Normally Open) terminals are connected to the appliances.
- Neutral (N) is connected to all appliances.
- Ensure proper insulation and safety while working with 230V AC.

14. ARDUINO CODE (USING BLYNK LIBRARY)

```
#define BLYNK_TEMPLATE_ID "TMPLxxxxxxx"
#define BLYNK_TEMPLATE_NAME "Smart Home Automation"
#define BLYNK_AUTH_TOKEN "Your_Blynk_Auth_Token"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

// Relay Pins
#define RELAY1 D1    // GPIO5
#define RELAY2 D2    // GPIO4
#define RELAY3 D5    // GPIO14
#define RELAY4 D6    // GPIO12

BlynkTimer timer;

void setup() {
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  pinMode(RELAY1, OUTPUT);
  pinMode(RELAY2, OUTPUT);
  pinMode(RELAY3, OUTPUT);
  pinMode(RELAY4, OUTPUT);

  // All relays OFF initially (HIGH)
  digitalWrite(RELAY1, HIGH);
  digitalWrite(RELAY2, HIGH);
  digitalWrite(RELAY3, HIGH);
  digitalWrite(RELAY4, HIGH);
}

void loop() {
  Blynk.run();
  timer.run();
}
// Blynk Handlers
BLYNK_WRITE(V1) { digitalWrite(RELAY1, param.asInt() ? LOW : HIGH); }
BLYNK_WRITE(V2) { digitalWrite(RELAY2, param.asInt() ? LOW : HIGH); }
BLYNK_WRITE(V3) { digitalWrite(RELAY3, param.asInt() ? LOW : HIGH); }
BLYNK_WRITE(V4) { digitalWrite(RELAY4, param.asInt() ? LOW : HIGH); }
```

15. DATA FLOW DIAGRAM



16. BLYNK APP INTERFACE (EXAMPLE)

WIDGETS USED	
Light 1	Button (V1)
Light 2	Button (V2)
Fan	Button (V3)
Socket	Button (V4)

NOTE:

- Green button = ON
- Gray button = OFF
- You can customize the button style and dashboard theme.

17. RELAY STATUS INDICATION

Relay	NodeMCU Pin	Blynk Virtual Pin	Status ON	Status OFF	LED Indicator
Relay 1	D1 (GPIO5)	V1	LOW	HIGH	LED1 ON
Relay 2	D2 (GPIO4)	V2	LOW	HIGH	LED2 ON
Relay 3	D5 (GPIO14)	V3	LOW	HIGH	LED3 ON
Relay 4	D6 (GPIO12)	V4	LOW	HIGH	LED4 ON

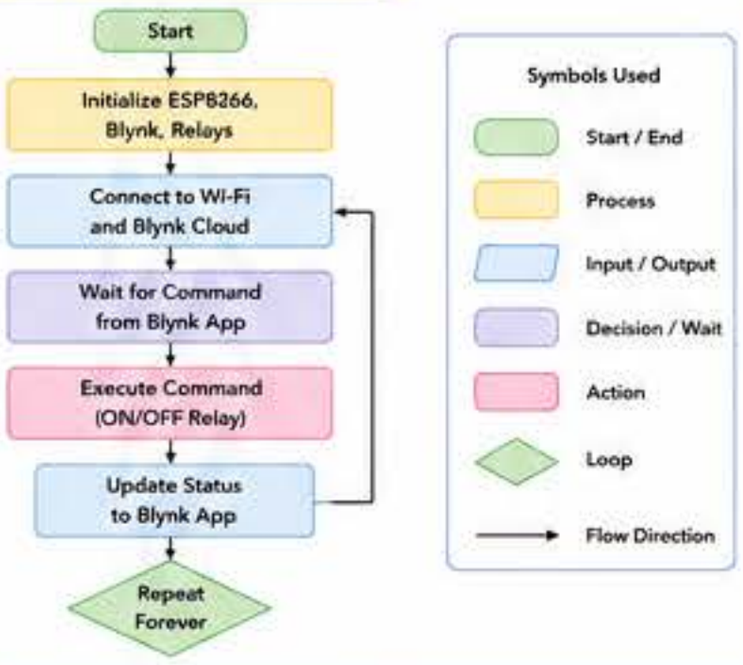
18. SAMPLE OPERATION

1. Open the Blynk App and login.
2. Tap the ON button for any appliance.
3. The command is sent to Blynk Cloud.
4. ESP8266 receives the command.
5. Corresponding relay is activated.
6. Appliance turns ON.
7. Tap OFF button to turn appliance OFF.
8. Status is updated in real-time.

19. IMPORTANT NOTES

- Use a regulated 5V power supply for ESP8266.
- Do not touch the circuit when 230V AC is ON.
- Ensure proper enclosure for safety.
- Keep the system away from water/moisture.
- Use high quality relay module.
- Always test the circuit before final installation.

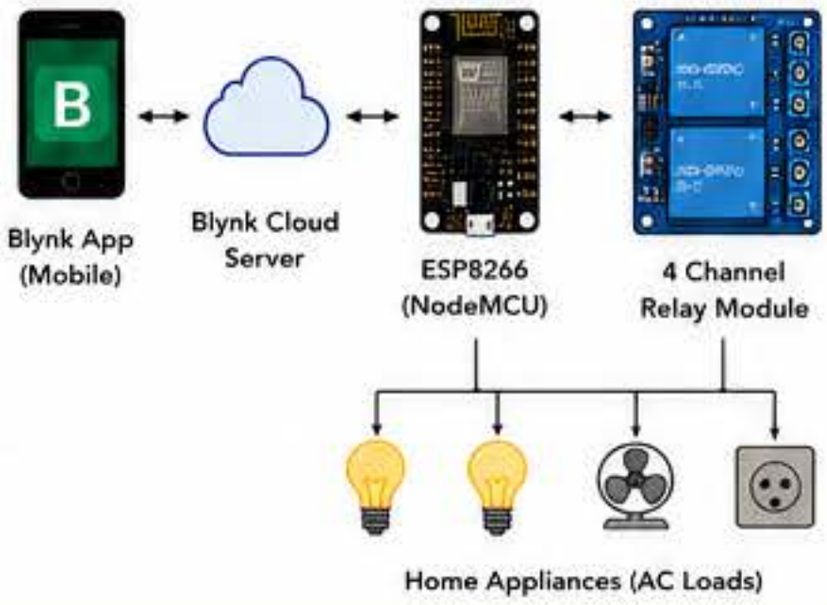
20. PROGRAM FLOWCHART



21. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
ESP8266 not connecting to Wi-Fi	Wrong SSID/Password Wi-Fi out of range	Check SSID & Password Move closer to router
Devices not turning ON/OFF	Wrong wiring Relay not working	Check all connections Replace relay module
App shows offline	Internet connection issue Blynk Auth Token wrong	Check Wi-Fi/Internet Verify Auth Token
App control delay	Weak internet Server busy	Improve internet Try again later

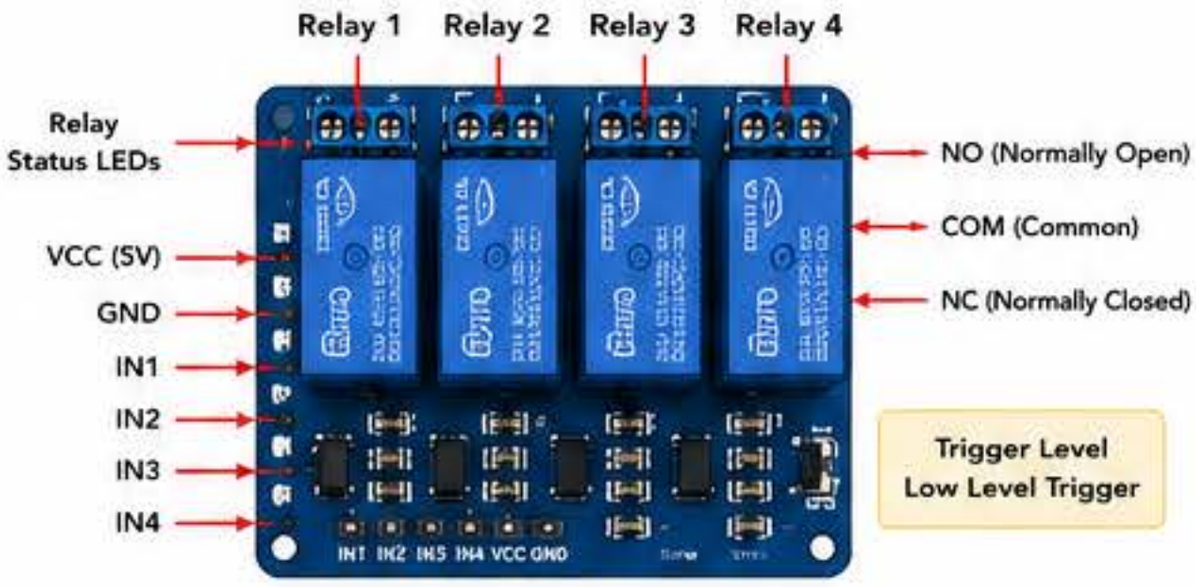
22. SYNOPTIC DIAGRAM



SAFETY FIRST:

- Work with caution while handling 230V AC.
- Use proper insulation and protective enclosure.
- Make sure all connections are tight.
- Test the system step by step.

23. RELAY MODULE DETAILS



Trigger Level
Low Level Trigger

Relay Terminals (Per Channel)	
Terminal	**Description**
NO (Normally Open)	Connected to COM when relay is OFF. Open when relay is ON.
COM (Common)	Common terminal connected to Live (L) for AC loads.
NC (Normally Closed)	Connected to COM when relay is ON. Open when relay is OFF.

24. COMMANDS SENT FROM BLYNK APP

Action in App	Command Sent	Description
Turn ON (Any Device)	HIGH (1)	Sends HIGH to the respective relay pin via Blynk Cloud. Relay gets activated and device turns ON.
Turn OFF (Any Device)	LOW (0)	Sends LOW to the respective relay pin via Blynk Cloud. Relay gets deactivated and device turns OFF.
Check Status	Status Update	ESP8266 sends current status of all relays to Blynk App in real-time.

25. WORKING EXPLANATION

1. The ESP8266 connects to Wi-Fi and Blynk Cloud.
2. When the user presses ON/OFF button on Blynk App, the command is sent to Blynk Cloud.
3. Blynk Cloud forwards the command to ESP8266.
4. ESP8266 activates or deactivates the corresponding relay.
5. Relay switches the connected appliance ON or OFF.
6. The status is sent back to Blynk App and displayed in real-time.

26. SAFETY PRECAUTIONS

- Work with caution while handling 230V AC.
- Ensure proper insulation and protective enclosure.
- Do not touch the circuit when power is ON.
- Use proper wire gauge for AC connections.
- Keep the system away from water and moisture.
- Use regulated 5V power supply for ESP8266.

27. ESP8266 PIN DESCRIPTION

Pin	Function	Description
5V	Power	5V input from regulated supply
GND	Ground	Ground connection
D1 (GPIO5)	Relay 1	Controls Relay 1
D2 (GPIO4)	Relay 2	Controls Relay 2
D5 (GPIO14)	Relay 3	Controls Relay 3
D6 (GPIO12)	Relay 4	Controls Relay 4
TX (GPIO1)	UART TX	Serial Transmit (for programming)
RX (GPIO3)	UART RX	Serial Receive (for programming)

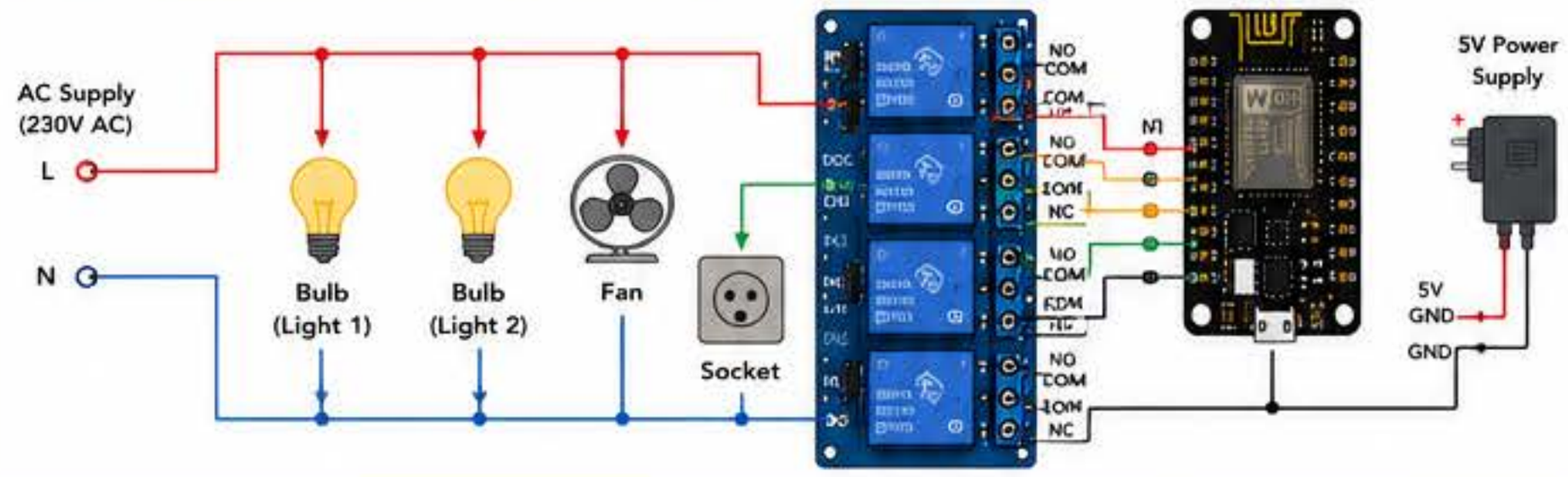
28. RELAY CONTROL LOGIC

GPIO Pin	Output	Relay Status	Appliance
D1 (GPIO5)	HIGH	ON	Light 1 ON
	LOW	OFF	Light 1 OFF
D2 (GPIO4)	HIGH	ON	Light 2 ON
	LOW	OFF	Light 2 OFF
D5 (GPIO14)	HIGH	ON	Fan ON
	LOW	OFF	Fan OFF
D6 (GPIO12)	HIGH	ON	Socket ON
	LOW	OFF	Socket OFF

29. LED INDICATIONS

LED	Status	Indication
Power LED (PWR)	ON	Module is powered
Relay 1 LED	ON	Relay 1 Active (ON)
Relay 2 LED	ON	Relay 2 Active (ON)
Relay 3 LED	ON	Relay 3 Active (ON)
Relay 4 LED	ON	Relay 4 Active (ON)
All LEDs OFF	System OFF / No activity	No relays are active

30. HOME AUTOMATION CONNECTION EXAMPLE



31. PARAMETER SUMMARY

Parameter	Value
Operating Voltage (NodeMCU)	5V DC
Operating Voltage (Relays)	5V DC
Number of Channels	4
Max AC Load per Channel	250V AC / 10A
Communication	Wi-Fi (2.4GHz)
Platform	Blynk IoT
Control Range	Anywhere (Internet)
Application	Home Automation

32. SAMPLE USE CASES

Smart Lighting
Control room lights from anywhere.

Fan Control
Switch ON/OFF based on temperature or schedule.

Appliance Automation
Control any home appliance remotely.

Security & Safety
Turn ON lights when motion detected or control devices during emergency.

Energy Saving
Turn OFF devices when not in use.

33. LEARNING OUTCOMES

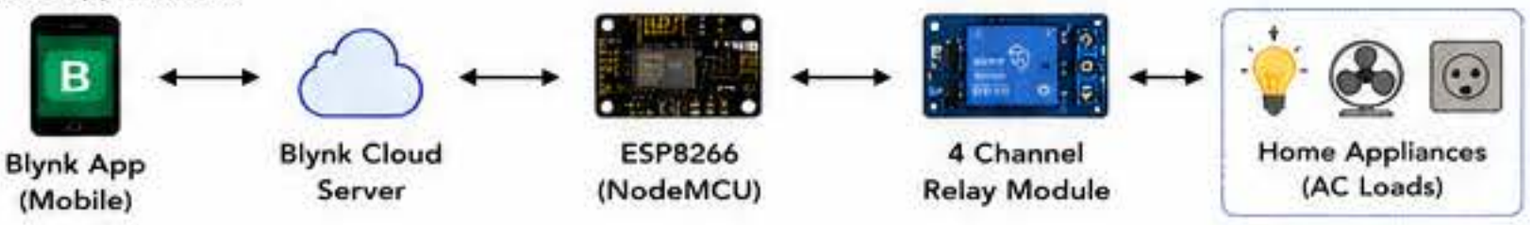
- Interface ESP8266 with relay module.
- Control multiple AC appliances via Internet.
- Use Blynk App for IoT based control.
- Understand cloud communication.
- Implement real-time monitoring and control.
- Apply IoT in home automation applications.

34. EXPANSION IDEAS

- Add temperature & humidity sensor.
- Add motion sensor for automation.
- Add voice control using Google Assistant / Alexa.
- Add energy meter (PZEM-004T) for power monitoring.
- Create schedules and timers in Blynk.
- Add battery backup and solar integration.
- Make it compatible with IFTTT for smart triggers.

35. CONCLUSION

This project demonstrates an IoT based Smart Home Automation System using ESP8266 and Relay Module. It allows users to control home appliances remotely from anywhere in the world using Blynk App. The system is cost-effective, flexible, and easy to expand for various smart home applications.



36. REFERENCES

- ESP8266 NodeMCU Datasheet
- Blynk Documentation
- 4 Channel Relay Module Datasheet
- Arduino & ESP8266 Programming Guide
- IoT with Blynk – IoT Projects Examples

NOTE:

- Ensure proper wiring and insulation while working with 230V AC.
- Use regulated 5V power supply for ESP8266 and Relay Module.
- Do not touch the circuit when power is ON.
- Test the system step by step before final installation.

PROJECT-6 MANUAL

IOT SMART DOOR LOCK MONITORING SYSTEM

USING BLYNK APP

USING ESP8266 (NODEMCU) AND REED SWITCH

1. AIM

To design and implement an IoT based smart door lock monitoring system using ESP8266 and Reed Switch to monitor door status and control the lock from anywhere through Blynk App.

2. OBJECTIVES

- Monitor door open/close status using Reed Switch.
- Lock and unlock the door remotely using Blynk App.
- Display real-time status on Blynk App.
- Send alert notifications for unauthorized access.
- Enhance security and convenience.

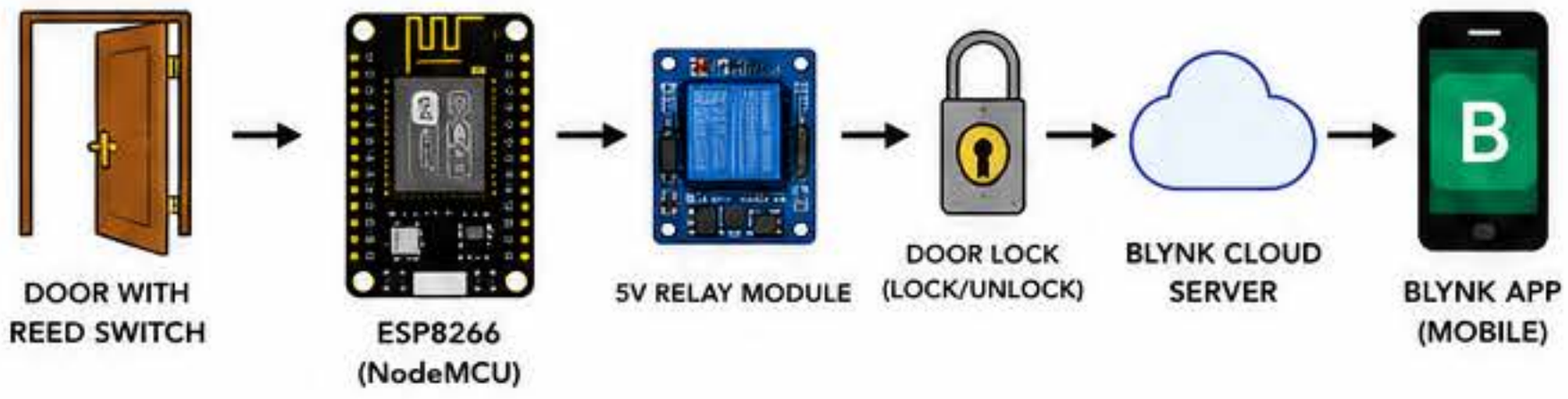
3. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	Reed Switch	1
3	Servo Motor (SG90)	1
4	5V Relay Module (1 Channel)	1
5	5V Power Supply (for NodeMCU)	1
6	Jumper Wires	Several

4. FEATURES

- Real-time door status monitoring.
- Remote door lock/unlock control.
- Instant notifications and alerts.
- Easy to install and user friendly.
- Low power consumption.
- Internet based remote access.

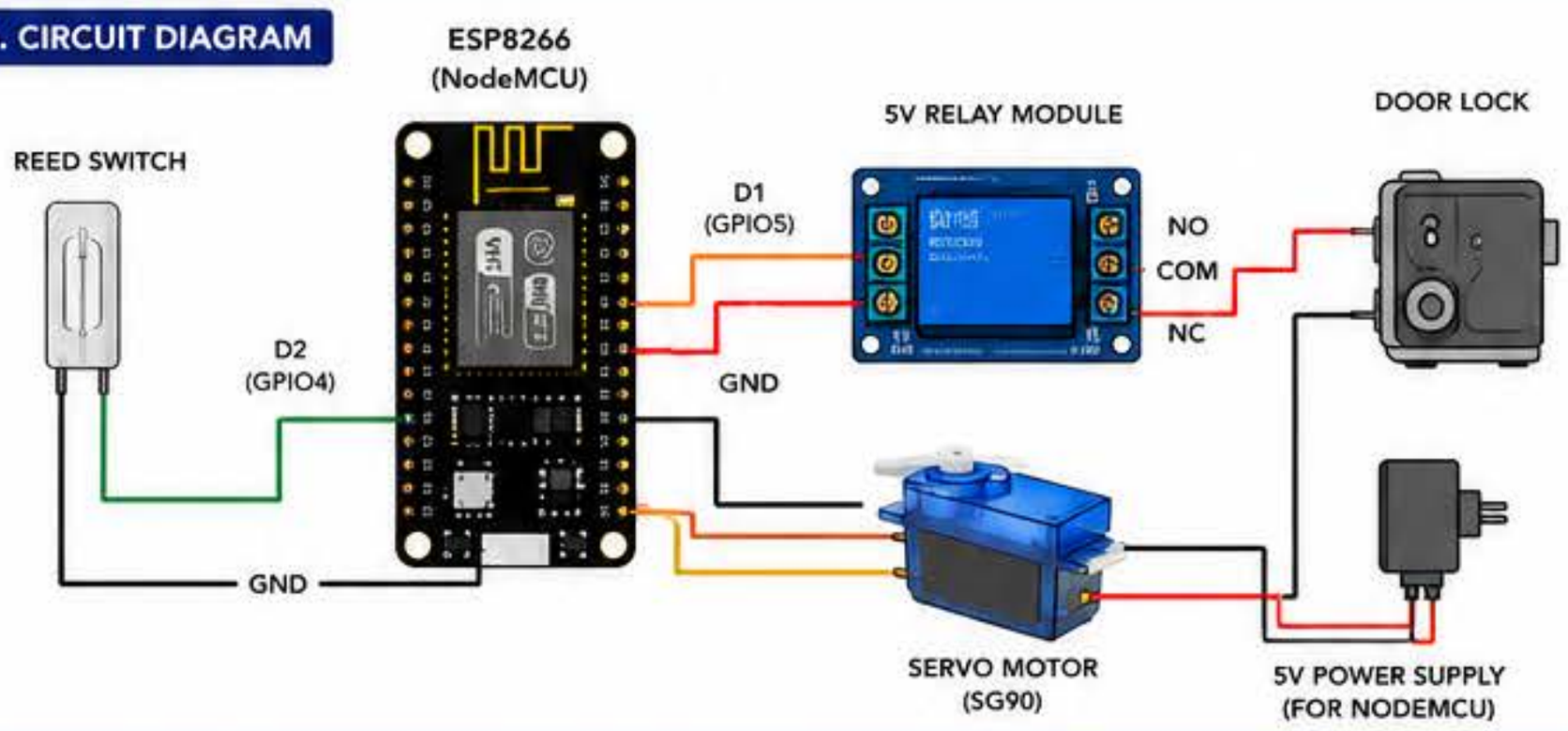
5. BLOCK DIAGRAM



WORKING:

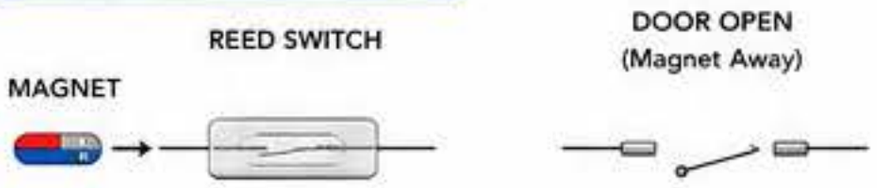
The Reed Switch detects the door open or close status and sends the signal to ESP8266. The user can lock or unlock the door from the Blynk App. ESP8266 controls the relay which drives the door lock mechanism. Status updates are sent to Blynk Cloud and displayed on the App.

6. CIRCUIT DIAGRAM



Connections					
Reed Switch	NodeMCU (ESP8266)	Relay Module	NodeMCU (ESP8266)	Servo Motor (SG90)	NodeMCU (ESP8266)
One Terminal	D2 (GPIO4)	VCC	5V	Signal (Orange)	D5 (GPIO14)
Other Terminal	GND	GND	GND	VCC (Red)	5V
-	-	IN	D1 (GPIO5)	GND (Brown)	GND

7. REED SWITCH WORKING



When the door is closed, magnet is near to reed switch and the circuit is closed.

When the door is opened, magnet moves away and the circuit is open.

8. SERVO MOTOR (SG90)

- Operating Voltage : 4.8V – 6V
- Rotation Angle : 0° – 180°
- Used to lock and unlock the door.

9. RELAY MODULE

- Operating Voltage : 5V DC
- Type : 1 Channel
- Contact Rating : 10A/250VAC
- Used to drive door lock mechanism.

10. BLYNK APP OVERVIEW

- Shows door status (Open/Closed).
- Lock and unlock door remotely.
- Real-time notifications.
- History of access logs.
- Works from anywhere via Internet.

11. HOW IT WORKS

1. Reed Switch detects door open/close status.
2. ESP8266 reads the status and sends to Blynk Cloud.
3. Status is displayed on Blynk App in real-time.
4. User can tap LOCK or UNLOCK button on the App.
5. ESP8266 controls the relay and servo motor to lock or unlock the door.
6. Notifications are sent for any status change.

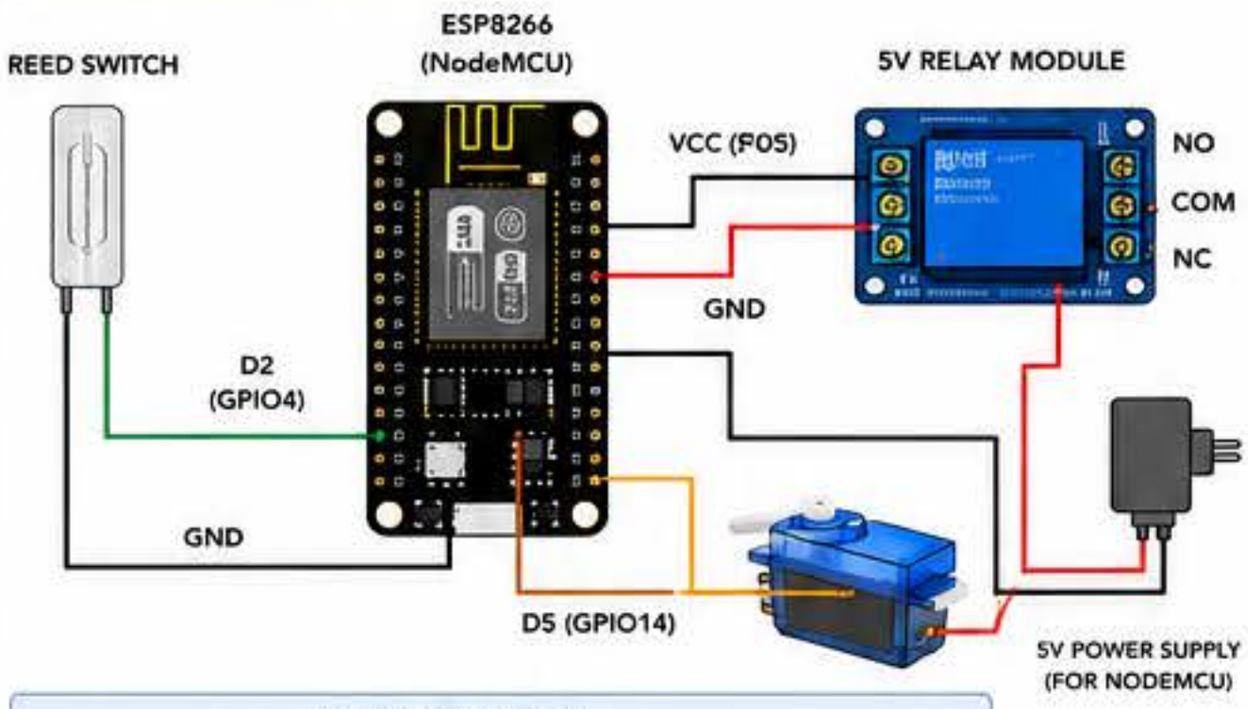
12. APPLICATIONS

- Home Security Systems
- Office Door Monitoring
- Smart Access Control
- Hotels and Guest Rooms
- Industrial Access Control

NOTE:

- Use regulated 5V power supply for ESP8266 and Relay Module.
- Ensure proper wiring and insulation.
- Do not keep the door lock under high load.
- Test the system step by step before final installation.

13. WIRING CONNECTIONS



Connection Summary		
From (ESP8266 Pin)	To	Purpose
D1 (GPIO5)	Relay IN	Control Relay (Door Lock)
D2 (GPIO4)	Reed Switch	Read Door Status
D5 (GPIO14)	Servo Signal	Control Servo Motor
5V	Relay VCC, Servo VCC	Power Supply
GND	All GND	Common Ground

NOTE:

- Make sure to connect grounds (GND) of all components together.
- Use external 5V supply if current requirement is high.

14. ARDUINO CODE (USING BLYNK LIBRARY)

```
// IoT Smart Door Lock using Blynk
#define BLYNK_TEMPLATE_ID "TMPLxxxxxxx"
#define BLYNK_TEMPLATE_NAME "Smart Door Lock"
#define BLYNK_AUTH_TOKEN "Your_Blynk_Auth_Token"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

#define RELAY_PIN   D1    // GPIO5
#define REED_PIN    D2    // GPIO4
#define SERVO_PIN   D5    // GPIO14

#include <Servo.h>
Servo doorServo;

BlynkTimer timer;

void setup() {
 Serial.begin(115200);
 Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
 pinMode(RELAY_PIN, OUTPUT);
 pinMode(REED_PIN, INPUT_PULLUP); // Reed switch
 doorServo.attach(SERVO_PIN);
 doorServo.write(0);  // Locked position
 digitalWrite(RELAY_PIN, LOW);  // Lock OFF
 timer.setInterval(1000L, sendStatus);
}
void loop() {
 Blynk.run();
 timer.run();
}
// Lock Door
BLYNK_WRITE(V1) {
 int value = parom.asInt();
 if(value == 1){
   unlockDoor();
 }
```

WIDGETS USED	
V1	Button (Unlock)
V2	Button (Lock)
V3	Value Display (Door Status)
V4	LED (Online Status)
Functions	
unlockDoor()	
lockDoor()	
sendStatus()	

NOTE:

- Change the Blynk Template ID and Auth Token.
- Switch positions of zero and 180 in Servo if required.
- Ensure stable power supply.

15. BLYNK APP INTERFACE (EXAMPLE)

Widgets Used	
Widget	Purpose
V1 Button (Unlock)	Unlock the door
V2 Button (Lock)	Lock the door
V3 Value Display	Show door status
V4 LED	Indicate online status

NOTE:

- Green text shows door is closed.
- Red text shows door is open.
- Buttons control the lock/unlock function.

16. DOOR STATUS LOGIC (USING REED SWITCH)

When Door is CLOSED

Reed Switch is ON (LOW)

Status = CLOSED

When Door is OPEN

Reed Switch is OFF (HIGH)

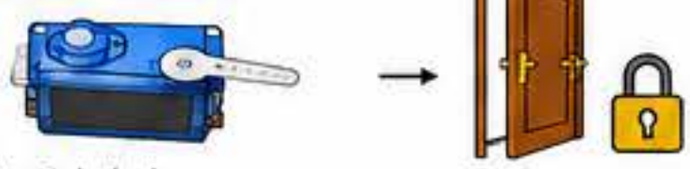
Status = OPEN

NOTE:

- Reed Switch gives digital output based on magnet position.
- Keep magnet properly aligned with reed switch.

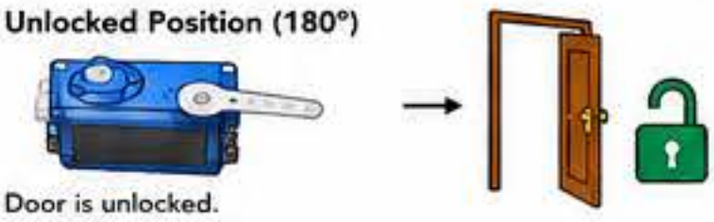
17. SERVO MOTOR POSITIONS

Locked Position (0°)



Door is locked.
Servo at 0 degrees.

Unlocked Position (180°)



Door is unlocked.
Servo at 180 degrees.

NOTE:

- Adjust servo arm as per your door lock mechanism.
- Test both positions before final installation.

18. RELAY STATUS INDICATION

Relay State	Relay Pin (D1)	Door Lock	Description
LOW (0)	Relay OFF	LOCKED	Power to lock is OFF
HIGH (1)	Relay ON	UNLOCKED	Power to lock is ON

19. DOOR LOCK OPERATIONS

Action	Blynk Button	Relay	Servo	Result
Unlock Door	Unlock (V1)	ON	180°	Door Unlocked
Lock Door	Lock (V2)	OFF	0°	Door Locked

20. SAMPLE TEST CASES

Test Case	Action	Expected Result
TC-1	Press Unlock	Door Unlocks, Status = OPEN
TC-2	Press Lock	Door Locks, Status = CLOSED
TC-3	Open Door Manually	Status = OPEN
TC-4	Close Door Manually	Status = CLOSED

21. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
Blynk App not connecting	Wrong Wi-Fi / Auth Token	Check Wi-Fi and Blynk Token
Door not unlocking	Relay wiring / Power issue	Check relay connections & power
Wrong door status	Reed switch alignment	Align magnet with reed switch
Servo not working	Wrong pin / power issue	Check servo connections
Frequent disconnections	Weak Wi-Fi	Use strong Wi-Fi or antenna

22. SAFETY PRECAUTIONS

- Use a regulated 5V power supply.
- Ensure proper insulation and wiring.
- Do not expose to water or moisture.
- Keep the lock mechanism under proper load.
- Test the system thoroughly before use.
- Do not share lock control with unknown users.

23. SPECIFICATIONS

Parameter	Details
• Microcontroller	ESP8266 (NodeMCU)
• Wi-Fi Standard	802.11 b/g/n (2.4 GHz)
• Operating Voltage	5V DC
• Control Range	Anywhere (Internet)
• Relay Output	1 Channel, 5V Relay
• Lock Mechanism	Servo + Relay
• Sensor	Reed Switch
• App	Blynk IoT

24. ADVANTAGES

- Remote door lock/unlock from anywhere.
- Real-time door status monitoring.
- Easy to install and low cost.
- Instant alerts and notifications.
- Enhances security and convenience.
- Works with any smart device.

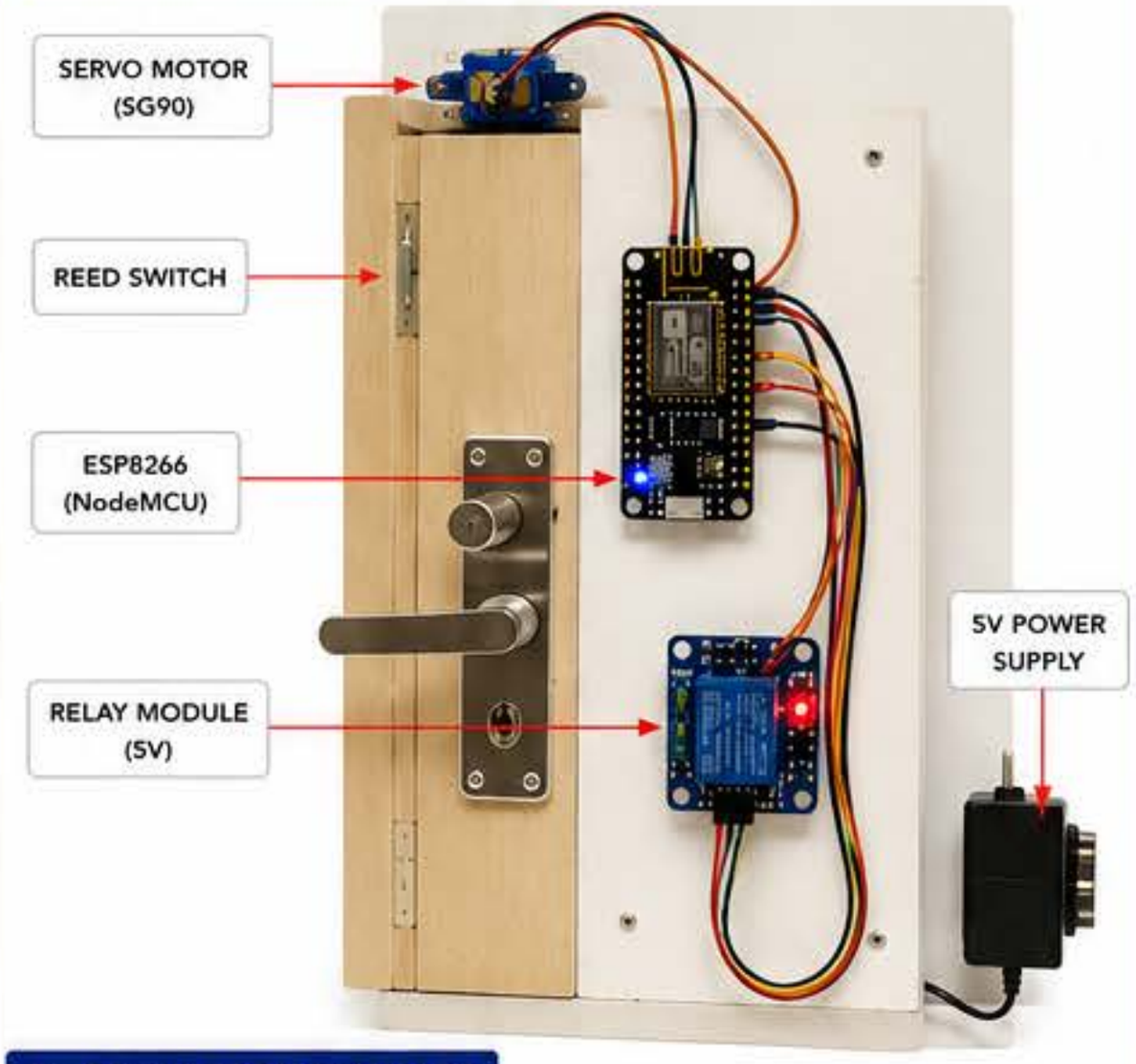
25. APPLICATIONS

26. CONCLUSIONS

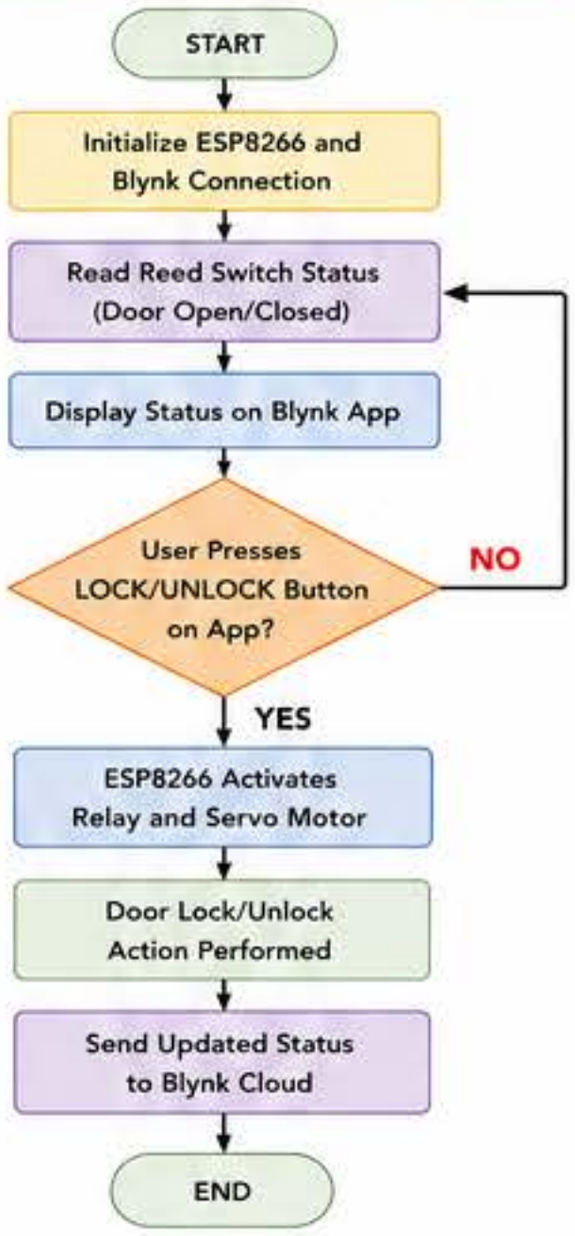
This project demonstrates an IoT based Smart Door Lock Monitoring System using ESP8266, Reed Switch, Relay, and Servo Motor. It allows users to lock/unlock the door remotely and monitor the door status in real-time through Blynk App. The system is cost-effective, reliable, and can be used in various security applications.

- Use regulated 5V power supply for ESP8266, Relay & Servo.
- Ensure proper wiring and insulation.
- Do not keep the door lock under high load.
- Test the system step by step before final installation.

24. IMPLEMENTATION SETUP



25. WORKING PROCESS FLOWCHART



PROCESS EXPLANATION

- Read Switch senses the door status (Open/Closed).
- Status is sent to ESP8266 and displayed on Blynk App.
- User can lock or unlock the door using buttons on the App.
- ESP8266 controls Relay Module to power the Servo Motor.
- Servo rotates to lock (0°) or unlock (180°).
- Updated status is sent to Blynk Cloud and shown on the App.

26. CONNECTION SUMMARY

From (ESP8266 Pin)	To	Purpose
D1 (GPIO5)	Relay IN	Control Relay (Door Lock)
D2 (GPIO4)	Reed Switch	Read Door Status
D5 (GPIO14)	Servo Signal	Control Servo Motor
5V	Relay VCC, Servo VCC	Power Supply
GND	All GND	Common Ground

NOTE:

- Make sure to connect grounds (GND) of all components together.
- Use external 5V supply if current requirement is high.

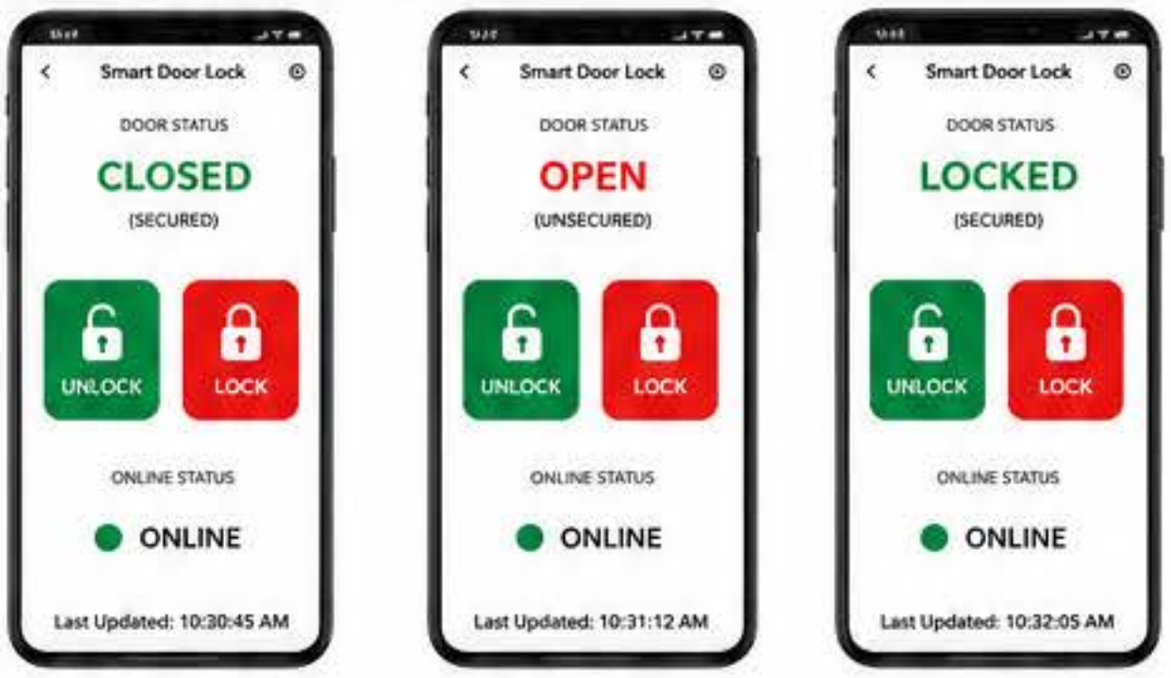
27. RESULTS AND OBSERVATIONS

Test Case	Action	Expected Result	Actual Result	Status
TC-1	Door Closed	Status = CLOSED	Status = CLOSED	PASS
TC-2	Unlock from App	Door Unlocks	Door Unlocked	PASS
TC-3	Lock from App	Door Locks	Door Locked	PASS
TC-4	Internet Disconnected	Last Status Shown	Last Status Shown	PASS
TC-5	Power ON	System Starts	System Started	PASS

OBSERVATION:

- The system works reliably for both locking and unlocking the door.
- Real-time status updates are fast and accurate on Blynk App.
- Servo motor positions are precise for 0° and 180°.
- The system remains stable with continuous operation.

28. SCREENSHOTS OF BLYNK APP



1. Door Closed Status
2. Door Open Status
3. Door Locked Status

29. FUTURE ENHANCEMENTS

- Biometric Authentication (Fingerprint / Face Recognition)
- RFID Card Based Access Control
- Door Camera Integration for Live Streaming
- Push Notifications and Email Alerts
- Battery Backup for Power Failure
- Solar Power Integration
- Voice Control using Google Assistant / Alexa
- Access Log and History with Web Dashboard

30. APPLICATIONS

- Home Security
- Office Access Control
- Hotels and Guest Rooms
- Industrial Security
- Smart Apartments
- Bank Locker Rooms
- Server Room Access
- Laboratories
- Institutions and Colleges
- Restricted Areas

31. CONCLUSION

The IoT Smart Door Lock Monitoring System using ESP8266 and Reed Switch provides a secure, efficient and cost-effective solution for remote door access control. The system allows users to lock/unlock the door and monitor its status in real-time from anywhere in the world through the Blynk App. The integration of ESP8266, Relay, Servo Motor and Reed Switch ensures reliable operation and instant status updates. This system can be widely used in homes, offices, industries and other restricted areas to enhance security and convenience.

32. ADVANTAGES

- Remote door lock/unlock from anywhere.
- Real-time door status monitoring.
- Easy to install and low cost.
- Instant alerts and notifications.
- Enhances security and convenience.
- Works with any smart device.

33. SAFETY PRECAUTIONS

- Use a regulated 5V power supply.
- Ensure proper insulation and wiring.
- Do not expose to water or moisture.
- Keep the lock mechanism under proper load.
- Test the system thoroughly before use.
- Do not share lock control with unknown users.

34. SPECIFICATIONS

Parameter	Details
Microcontroller	ESP8266 (NodeMCU)
Wi-Fi Standard	802.11 b/g/n (2.4 GHz)
Operating Voltage	5V DC
Control Range	Anywhere (Internet)
Relay Output	1 Channel, 5V Relay
Lock Mechanism	Servo + Relay
Sensor	Reed Switch
App	Blynk IoT

NOTE:

- Use regulated 5V power supply for ESP8266, Relay & Servo.
- Ensure proper wiring and insulation.
- Do not keep the door lock under high load.
- Test the system step by step before final installation.

PROJECT-7

IOT MOTION DETECTION AND SECURITY ALARM USING BLYNK APP

USING ESP8266 (NodeMCU), PIR SENSOR AND BUZZER

1. AIM

To design and implement an IoT based motion detection security system that detects human movement using PIR sensor, triggers an alarm (buzzer) and sends real-time alerts to the user through Blynk App.

2. OBJECTIVES

- Detect motion using PIR sensor.
- Trigger buzzer and LED on motion detection.
- Send real-time notifications to Blynk App.
- Allow user to arm/disarm the system from the Blynk App.
- Enhance security and monitoring.

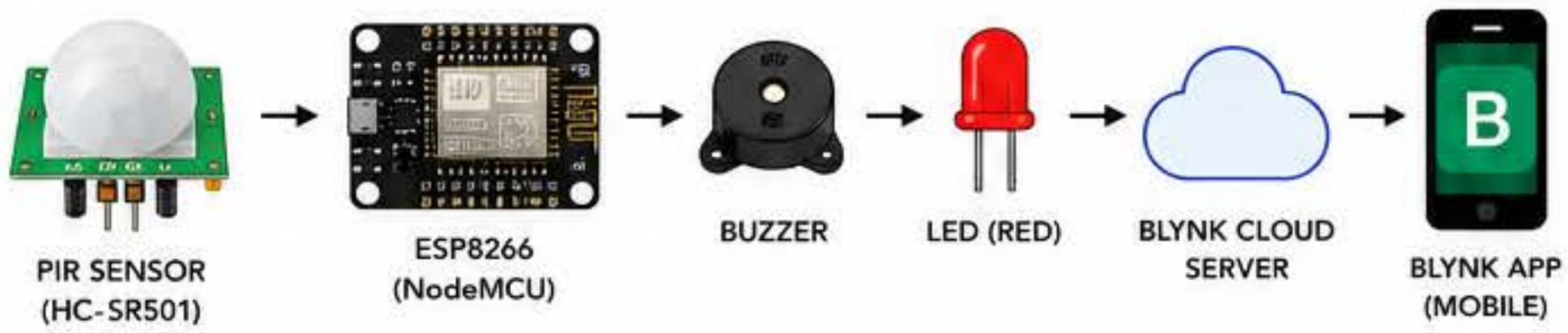
3. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	PIR Motion Sensor (HC-SR501)	1
3	Active Buzzer	1
4	LED (Red)	1
5	1kΩ Resistor	1
6	Jumper Wires	Several
7	5V Power Supply / USB Cable	1

4. FEATURES

- Detects motion in the monitored area.
- Triggers buzzer and LED alarm.
- Sends instant alerts on Blynk App.
- Arm/Disarm control from anywhere.
- Real-time status monitoring.
- Easy to install and low cost.
- Internet based remote monitoring.

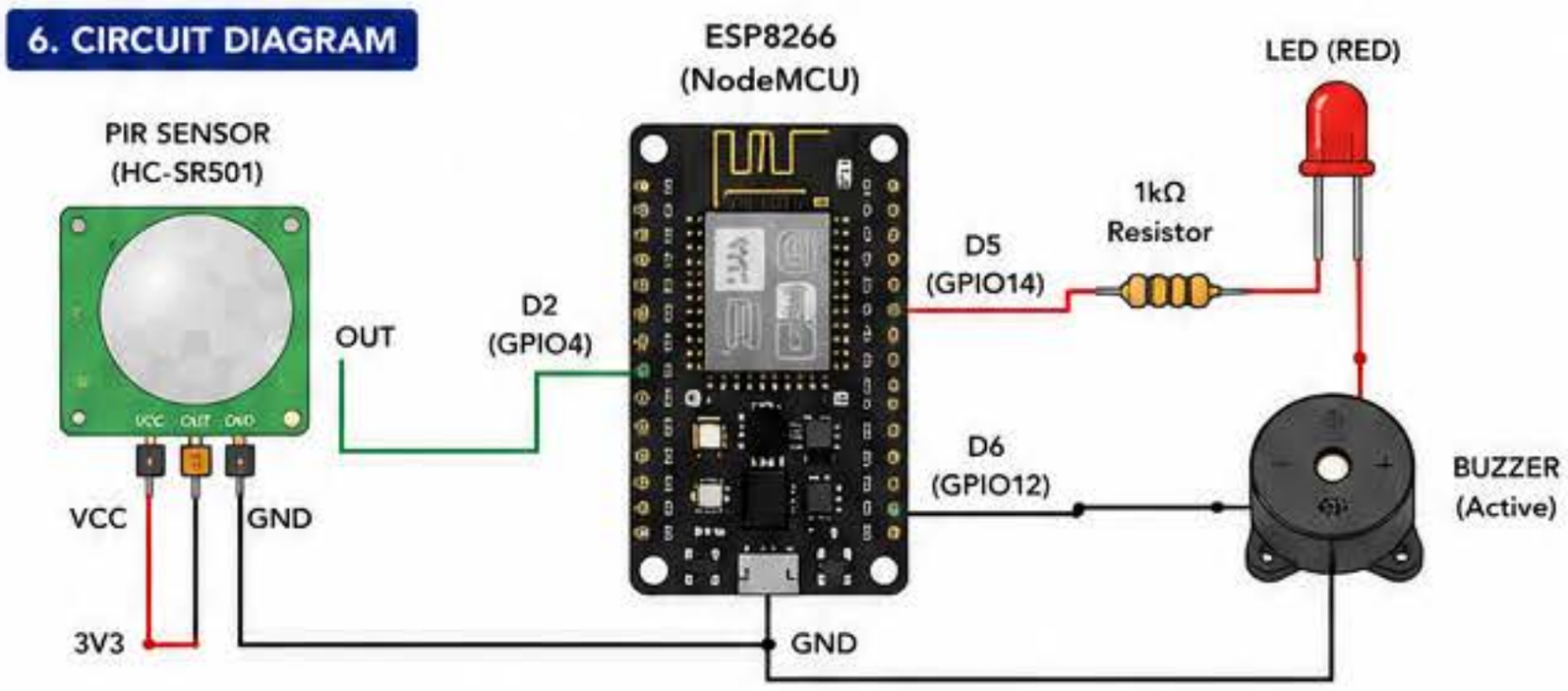
5. BLOCK DIAGRAM



WORKING:

The PIR sensor detects human motion and sends a HIGH signal to ESP8266.
ESP8266 activates the buzzer and LED as alarm.
At the same time, ESP8266 sends a notification to Blynk Cloud.
The user can monitor the status and control the system (Arm/Disarm) from Blynk App.

6. CIRCUIT DIAGRAM



Connections				
PIR Sensor	NodeMCU (ESP8266)	LED (Red)	Buzzer	Power Supply
VCC	3V3	Anode → D5 (GPIO14) through 1kΩ Resistor	Positive → D6 (GPIO12)	5V (USB / Adapter)
OUT	D2 (GPIO4)	-	-	-
GND	GND	Cathode → GND	Negative → GND	GND

7. PIR SENSOR (HC-SR501)

- Operating Voltage : 4.5V – 20V DC
- Detection Range : 3 – 7 meters
- Detection Angle : ~120°
- Output : HIGH on motion detected

8. ACTIVE BUZZER

- Operating Voltage : 3V – 12V DC
- Produces alert sound when activated.

9. BLYNK APP OVERVIEW

- Displays current system status (Armed / Disarmed).
- Shows motion detection status (Detected / Clear).
- Sends instant push notifications.
- Allows user to Arm or Disarm the system remotely.
- Works from anywhere via Internet.

10. HOW IT WORKS

1. System is armed using Blynk App.
2. PIR sensor continuously monitors the area.
3. When motion is detected:
   - ➢ PIR sends HIGH signal to ESP8266.
   - ➢ ESP8266 triggers buzzer and LED.
   - ➢ ESP8266 sends alert to Blynk Cloud.
4. User receives instant notification on Blynk App.
5. User can disarm the system from the App.

11. APPLICATIONS

- Home Security Systems
- Office / Shop Surveillance
- Warehouse Monitoring
- Smart Alarm Systems
- Intrusion Detection

12. ADVANTAGES

- ✔ Real-time motion detection and alert.
- ✔ Remote monitoring from anywhere.
- ✔ Instant notifications on mobile.
- ✔ Low cost and easy to implement.
- ✔ Improves security and safety.
- ✔ Energy efficient and reliable.

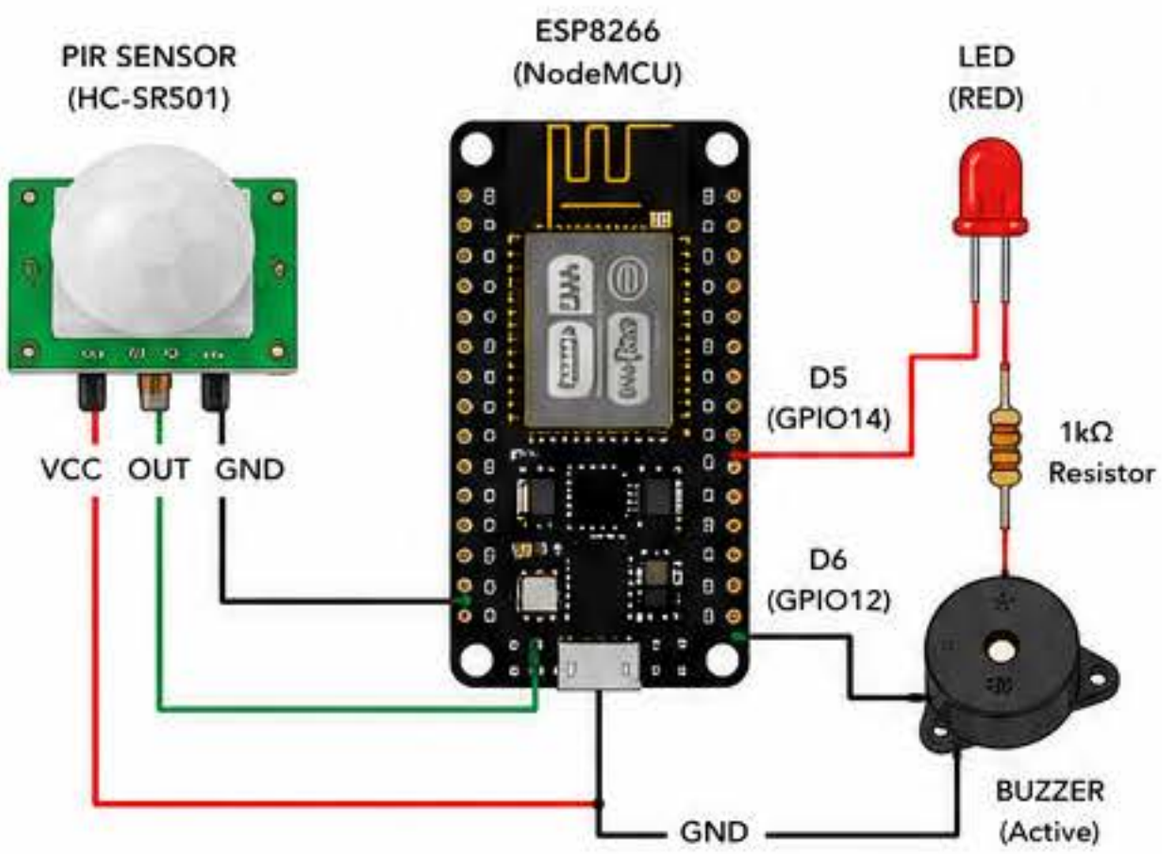
12. SAFETY PRECAUTIONS

- ⚠ Use regulated 5V power supply.
- ⚠ Ensure proper wiring connections.
- ⚠ Do not expose the circuit to moisture or high temperature.
- ⚠ Keep the device out of reach of children.
- ⚠ Test the system before final installation.

14. FUTURE SCOPE

- ➔ Add camera module for live streaming.
- ➔ Add SD card storage for event logs.
- ➔ Integrate with voice assistants.
- ➔ Add GSM module for SMS alerts.
- ➔ Solar powered security system.

15. WIRING DIAGRAM (CIRCUIT CONNECTIONS)



NOTE:

- Connect all GND of components to NodeMCU GND.
- Use external 5V power supply if current requirement is high.

16. ARDUINO CODE (USING BLYNK LIBRARY)

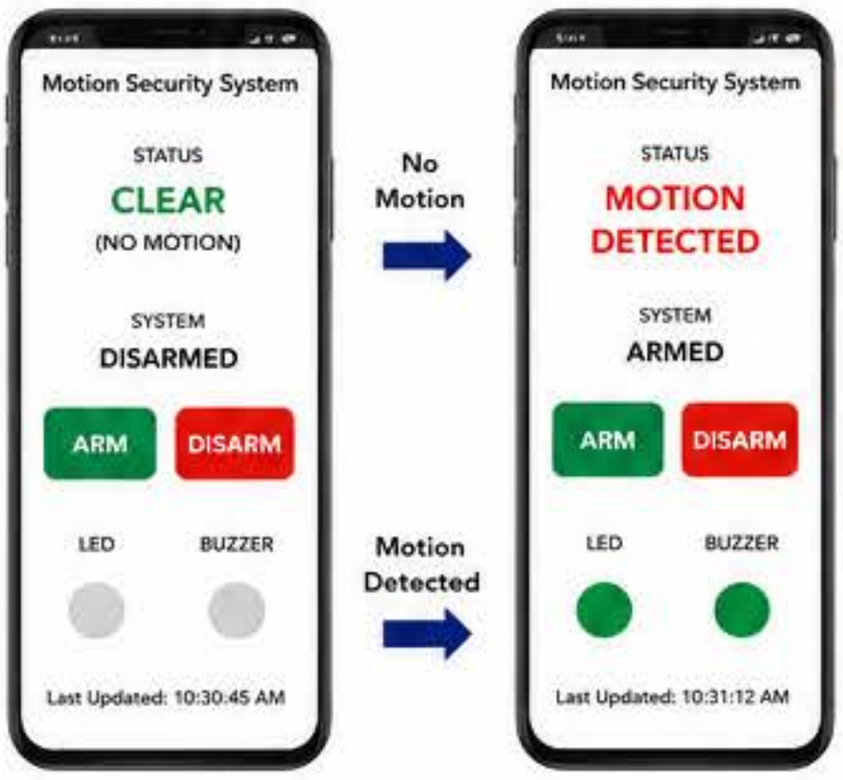
```
#define BLYNK_TEMPLATE_ID "TMPLxxxxxx"
#define BLYNK_TEMPLATE_NAME "Motion Security System"
#define BLYNK_AUTH_TOKEN "Your_Blynk_Auth_Token"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

#define PIR_PIN    D2    // GPIO4
#define LED_PIN    D5    // GPIO14
#define BUZZER_PIN D6    // GPIO12

void setup() {
 Serial.begin(115200);
 Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
 pinMode(PIR_PIN, INPUT);
 pinMode(LED_PIN, OUTPUT);
 pinMode(BUZZER_PIN, OUTPUT);
 digitalWrite(LED_PIN, LOW);
 digitalWrite(BUZZER_PIN, LOW);
}
void loop() {
 Blynk.run();
 int motion = digitalRead(PIR_PIN);
 if (motion == HIGH) {
   digitalWrite(LED_PIN, HIGH);
   digitalWrite(BUZZER_PIN, HIGH);
   Blynk.virtualWrite(V0, 1); // Motion Detected
 } else {
   digitalWrite(LED_PIN, LOW);
   digitalWrite(BUZZER_PIN, LOW);
   Blynk.virtualWrite(V0, 0); // No Motion
 }
}
```

BLYNK WIDGETS USED	
Widget	**Purpose**
V0 (Value Display)	Motion Status
V1 (LED Button)	Arm / Disarm
V2 (LED Indicator)	LED Status
V3 (LED Indicator)	Buzzer Status

NOTE:

- Change Blynk Template ID and Auth Token.
- Make sure Wi-Fi is stable.
- PIR output is HIGH when motion is detected.

17. BLYNK APP DASHBOARD (EXAMPLE)



Widgets Explanation

- **STATUS (V0):** Shows motion status.
- **ARM / DISARM (V1):** Arm or disarm the system.
- **LED (V2):** Indicates LED ON/OFF status.
- **BUZZER (V3):** Indicates Buzzer ON/OFF status.

18. TEST CASES

Test Case No.	Condition	Expected Result	Actual Result	Status
TC-1	No motion in area	Status = CLEAR LED OFF, Buzzer OFF	Same as Expected	PASS
TC-2	Motion detected	Status = MOTION DETECTED LED ON, Buzzer ON	Same as Expected	PASS
TC-3	Arm system from App	System ARMED (Controls Enabled)	Same as Expected	PASS
TC-4	Disarm system from App	System DISARMED (Controls Disabled)	Same as Expected	PASS
TC-5	Wi-Fi disconnected	Cannot update status on Blynk App	Same as Expected	PASS
TC-6	Reconnect Wi-Fi	Status updates normally on App	Same as Expected	PASS

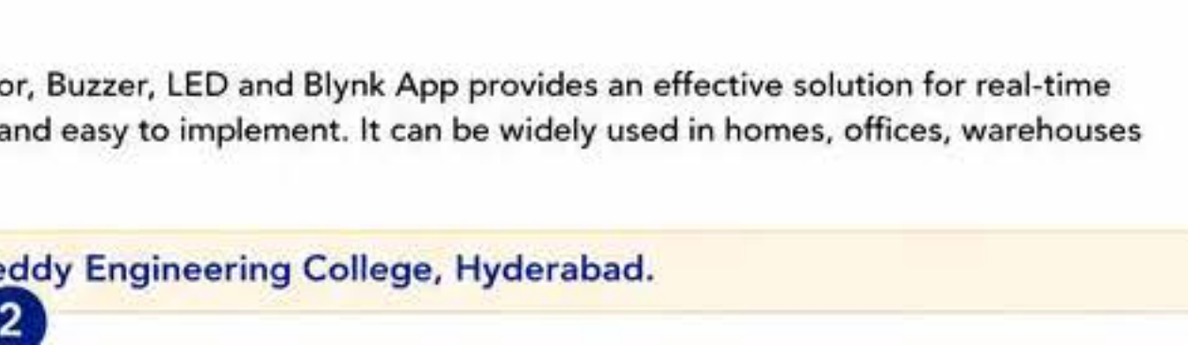
OBSERVATION:

- The system accurately detects motion and triggers alarm (LED + Buzzer).
- Instant notifications are received on Blynk App.
- System works reliably with stable Wi-Fi connection.

19. COMPONENT FUNCTIONS

Component	Function
ESP8266 (NodeMCU)	Connects to Wi-Fi and communicates with Blynk Cloud.
PIR Sensor (HC-SR501)	Detects human motion using infrared signals.
LED (Red)	Visual indicator for motion detection.
Active Buzzer	Audio alert when motion is detected.
Blynk App	Monitors status and controls the system remotely.

20. HARDWARE IMPLEMENTATION (ACTUAL SETUP)



21. ADVANTAGES

- ✔ Real-time motion detection and alerts.
- ✔ Remote monitoring and control via Blynk App.
- ✔ Cost-effective and easy to implement.
- ✔ Enhances security of home, office, and industries.
- ✔ Low power consumption and reliable performance.

22. APPLICATIONS

- Home Security Systems
- Office / Shop Surveillance
- Warehouse Monitoring
- Smart Alarm Systems
- Intrusion Detection
- Retail Store Security

23. SAFETY PRECAUTIONS

- ⚠ Use regulated 5V power supply.
- ⚠ Ensure proper wiring and insulation.
- ⚠ Do not expose the circuit to water.
- ⚠ Keep the device out of reach of children.
- ⚠ Test the system before final installation.

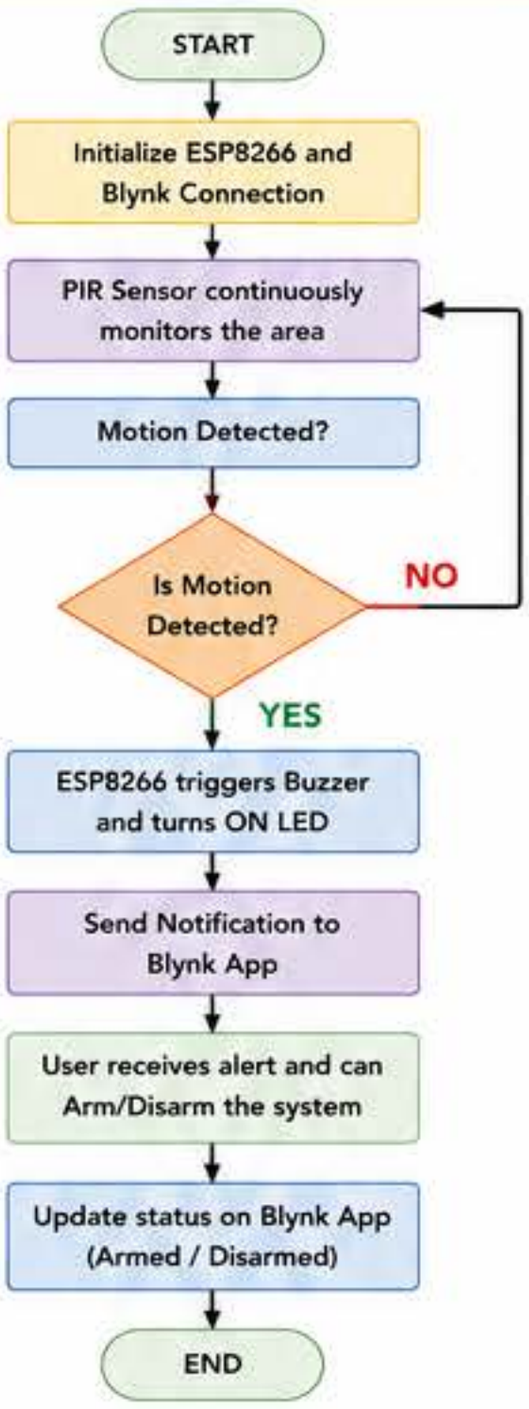
24. FUTURE SCOPE

- → Add camera for live streaming.
- → Add GSM module for SMS alerts.
- → Store event logs on cloud / SD card.
- → Integrate with voice assistants.
- → Add AI-based motion classification.
- → Solar powered security system.

25. CONCLUSION

The IoT Motion Detection and Security Alarm System using ESP8266, PIR sensor, Buzzer, LED and Blynk App provides an effective solution for real-time motion detection and remote monitoring. The system is reliable, cost-effective and easy to implement. It can be widely used in homes, offices, warehouses and other security applications to enhance safety and security.

26. WORKING PROCESS FLOWCHART



PROCESS EXPLANATION

- System is armed using Blynk App.
- PIR sensor keeps monitoring the surroundings.
- When motion is detected, ESP8266 activates Buzzer and LED.
- Simultaneously, an alert is sent to Blynk Cloud.
- User receives instant notification on the Blynk App.
- User can disarm the system from the App.
- All status updates are displayed in real-time on the App.

27. RESULTS AND OBSERVATIONS

Test Case	Condition	Expected Result	Actual Result	Status
TC-1	No motion in area	LED OFF, Buzzer OFF No Notification	LED OFF, Buzzer OFF No Notification	PASS
TC-2	Motion detected	LED ON, Buzzer ON Notification Sent	LED ON, Buzzer ON Notification Sent	PASS
TC-3	Multiple motion detections	Continuous alerts (No system crash)	Continuous alerts (System Stable)	PASS
TC-4	Disarm from App	Buzzer OFF, LED OFF System Disarmed	Buzzer OFF, LED OFF System Disarmed	PASS
TC-5	Wi-Fi disconnected	No update on App (Offline)	No update on App (Offline)	PASS
TC-6	Reconnect Wi-Fi	Status updates normally	Status updates normally	PASS

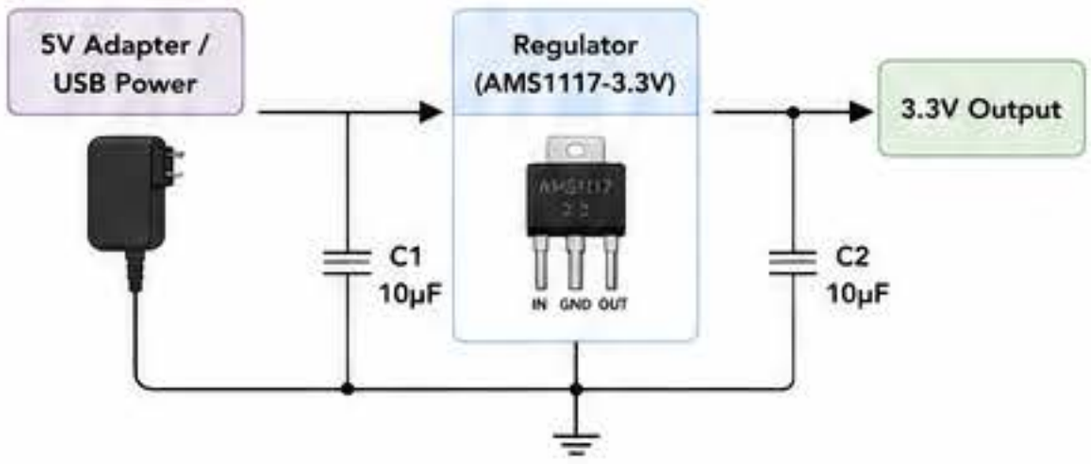
OBSERVATION:

- The PIR sensor accurately detects motion in the monitored area.
- Buzzer and LED are reliably triggered on motion detection.
- Instant notifications are received on Blynk App.
- System remains stable with continuous monitoring.

28. BLYNK APP INTERFACE (SAMPLE SCREENS)

1. System Disarmed (No Motion)
2. System Armed (No Motion)
3. Motion Detected (Alert Triggered)
4. System Disarmed (After Alert)

29. POWER SUPPLY CIRCUIT



NOTE:
ESP8266 works on 3.3V. Use a stable power supply for reliable operation.

30. PERFORMANCE SUMMARY

System Performance	
Motion Detection Range	3 – 7 meters
Detection Angle	≈ 120°
Response Time	≈ 1 – 2 seconds
Notification Delay	≈ 2 – 3 seconds
Wi-Fi Range	Up to 30 meters (Indoor)
Operating Voltage	5V (Power Supply) 3.3V (ESP8266)
Current Consumption	≈ 250 – 300 mA
System Stability	High

Test Environment	
Location	Indoor
Temperature	25 – 32 °C
Wi-Fi Network	2.4 GHz
Blynk App Version	Latest
Test Duration	Multiple Trials
Result	Successful

31. ADVANTAGES

- ✔ Real-time motion detection and instant alerts.
- ✔ Remote monitoring and control using Blynk App.
- ✔ Cost-effective and easy to implement.
- ✔ Enhances security for homes and offices.
- ✔ Low power consumption.
- ✔ Scalable and can be extended with camera, GSM, etc.

32. APPLICATIONS

- Home Security Systems
- Office / Shop Surveillance
- Warehouses and Godowns
- Smart Alarm Systems
- Intrusion Detection
- Retail Store Security

33. SAFETY PRECAUTIONS

- ⚠ Use a regulated 5V power supply.
- ⚠ Ensure proper wiring and insulation.
- ⚠ Do not expose the circuit to moisture.
- ⚠ Keep the device out of reach of children.
- ⚠ Test the system before final installation.

34. FUTURE SCOPE

- ➔ Integrate camera for live streaming.
- ➔ Add GSM module for SMS alerts.
- ➔ Store events and logs on SD card.
- ➔ Integrate with voice assistants.
- ➔ Add AI-based motion classification.
- ➔ Solar powered security system.

35. CONCLUSION

The IoT Motion Detection and Security Alarm System using ESP8266, PIR Sensor, Buzzer, LED and Blynk App provides an efficient, reliable and affordable solution for real-time security monitoring. The system detects motion accurately, triggers alarm and sends instant notifications to the user's mobile. It can be easily deployed in homes, offices, warehouses and other security-sensitive areas. The system is stable, user-friendly and can be enhanced with additional features for more advanced security applications.

PROJECT OUTCOME

- Successfully designed and tested an IoT based motion detection alarm system.
- Achieved real-time monitoring and instant notifications on Blynk App.
- System is reliable, cost-effective and easy to implement.

SKILLS LEARNED

- IoT integration with ESP8266 and Blynk App.
- Interfacing sensors and actuators.
- Real-time data monitoring and control.
- Embedded system design and debugging.

PROJECT-8

IOT SMART GARDEN IRRIGATION SYSTEM
USING BLYNK APP

USING ESP8266 (NodeMCU), SOIL MOISTURE SENSOR, RELAY AND PUMP

1. AIM

To design and implement an IoT based smart irrigation system that monitors soil moisture and automatically controls water pump for plants. The system sends real-time data and notifications to the user through Blynk App.

2. OBJECTIVES

- Monitor soil moisture level in real-time.
- Automatically ON/OFF the water pump based on threshold.
- Control irrigation manually from Blynk App.
- Send real-time updates and alerts on Blynk App.
- Water plants efficiently and save water.

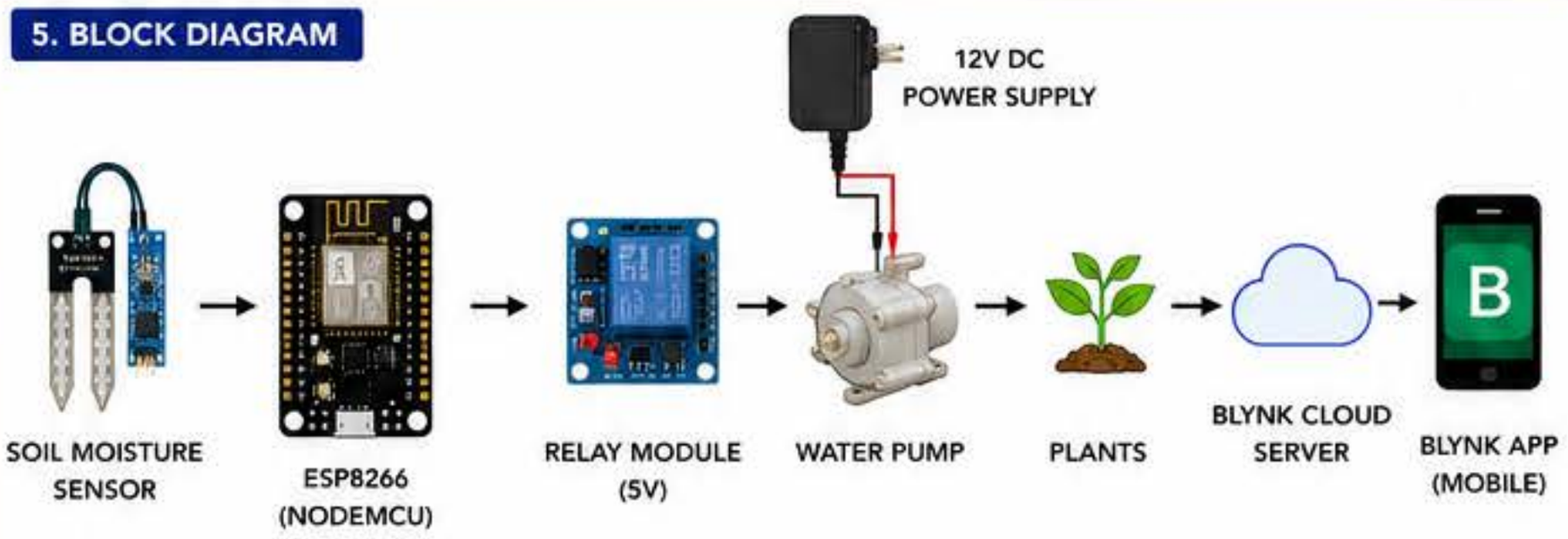
3. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	Soil Moisture Sensor	1
3	Relay Module (5V)	1
4	Submersible Water Pump	1
5	12V DC Power Supply	1
6	Jumper Wires	Several
7	Connecting Wires / Tubing	As Required

4. FEATURES

- Real-time soil moisture monitoring.
- Automatic and manual control of water pump.
- Real-time data display on Blynk App.
- Alerts and notifications on moisture level.
- Saves water and improves plant health.
- Easy to install and low cost.
- Remote monitoring and control via Internet.

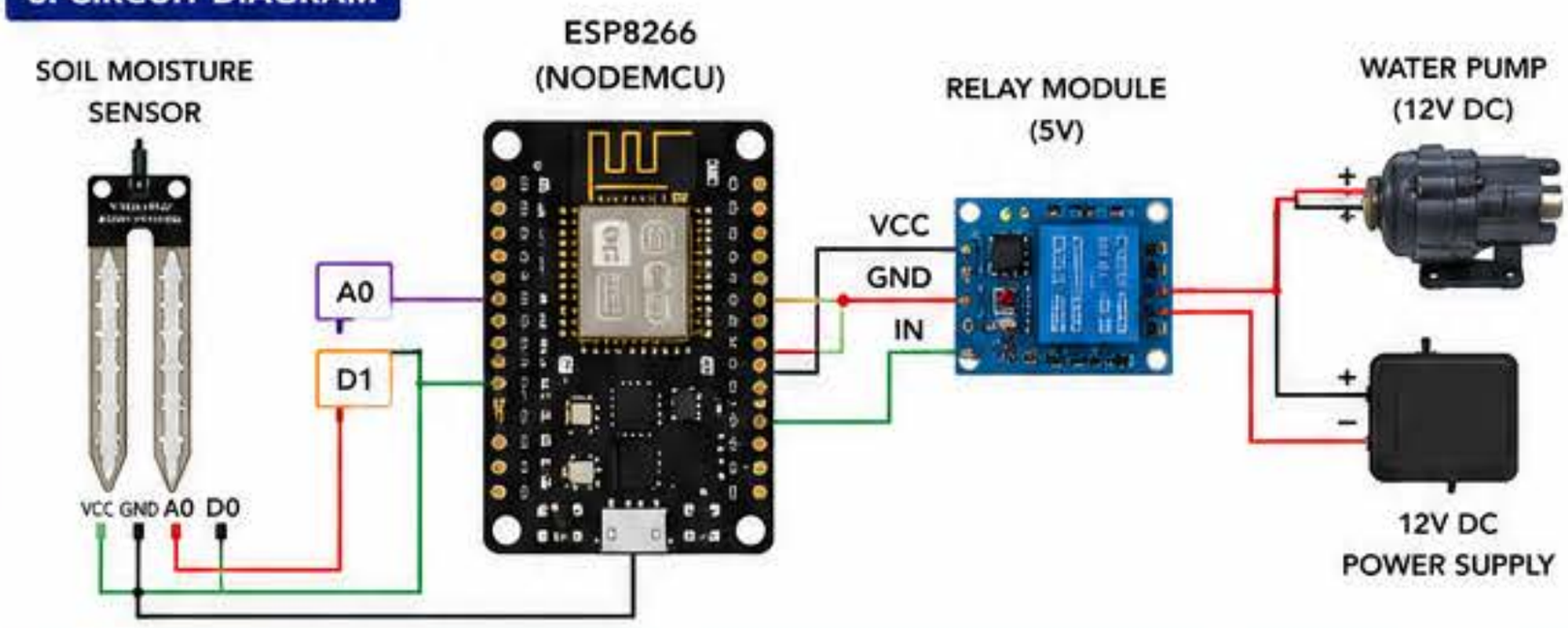
5. BLOCK DIAGRAM



WORKING:

The soil moisture sensor measures the moisture content in the soil and sends the data to ESP8266.
If the moisture level is below the set threshold, ESP8266 activates the relay to turn ON the water pump.
Once the moisture level reaches the threshold, the pump is turned OFF.
The system updates the moisture level and pump status on Blynk App in real-time.

6. CIRCUIT DIAGRAM



CONNECTIONS			
Soil Moisture Sensor	**NodeMCU (ESP8266)**	**Relay Module (5V)**	**Power Supply (12V)**
VCC	3V3	VCC	+12V (to Pump)
GND	GND	GND	GND (to Pump)
A0	A0	-	-
D0	D1	IN	-

7. SOIL MOISTURE SENSOR

- Operating Voltage : 3.3V – 5V DC
- Detection Range : 0% – 100% (Dry to Wet)
- Output : Analog (A0) & Digital (D0)
- Used to measure soil moisture level.

8. RELAY MODULE (5V)

- Operating Voltage : 5V DC
- Relay Type : SPDT
- Can switch high voltage load (up to 10A).
- Used to control water pump.

9. WATER PUMP (12V DC)

- Type : Submersible Pump
- Voltage : 12V DC
- Current : As per pump rating
- Used to irrigate plants.

10. BLYNK APP OVERVIEW

- Displays soil moisture level in real-time.
- Shows pump status (ON/OFF).
- Allows manual control of pump ON/OFF.
- Sends alerts and notifications.
- Works from anywhere via Internet.

11. HOW IT WORKS

1. Soil moisture sensor reads the moisture level in soil.
2. Data is sent to ESP8266 controller.
3. If moisture level < set threshold, ESP8266 turns ON the pump.
4. If moisture level ≥ set threshold, ESP8266 turns OFF the pump.
5. Real-time data and status are updated on Blynk App.
6. User can also control pump manually from the App.

12. APPLICATIONS

- Home Gardens
- Terrace Gardens
- Greenhouses
- Agriculture Fields
- Potted Plants

13. ADVANTAGES

- Real-time monitoring and control.
- Automatic irrigation saves water.
- Remote access from anywhere.
- Improves plant health.
- Low cost and easy to implement.
- Reduces manual effort.

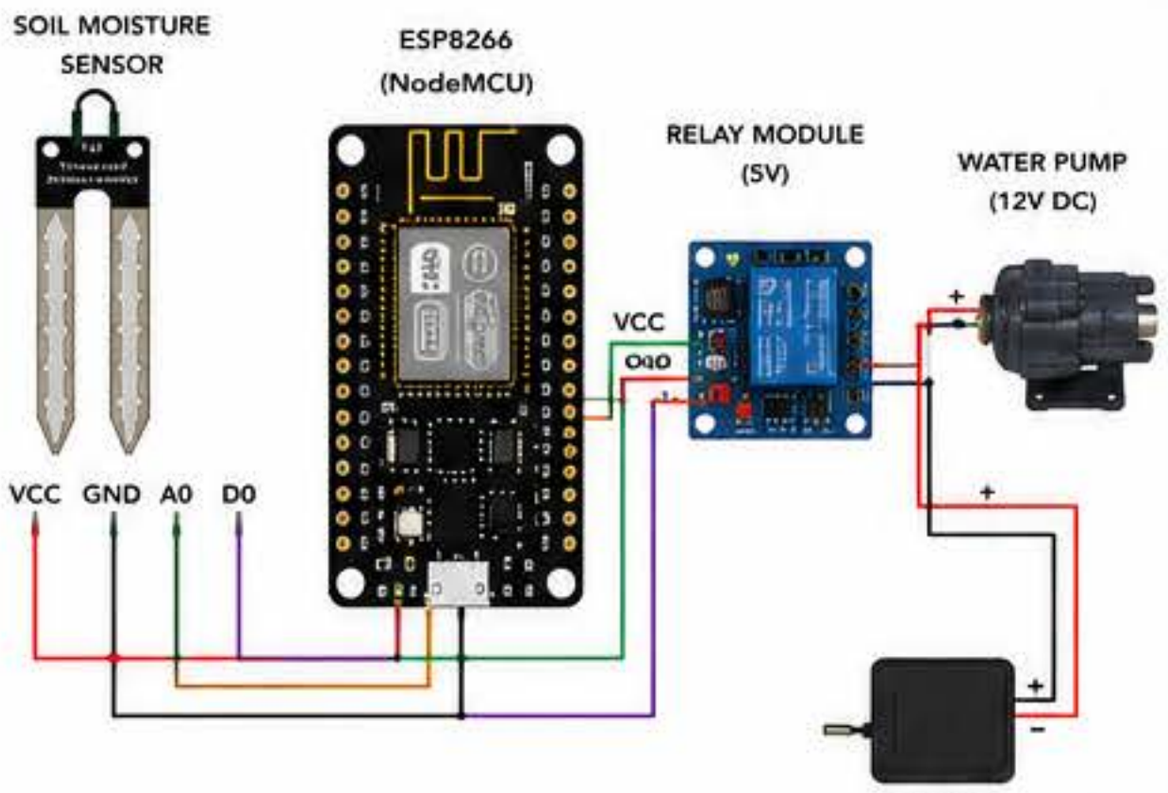
14. SAFETY PRECAUTIONS

- ⚠ Use proper 12V power supply.
- ⚠ Ensure correct wiring connections.
- ⚠ Do not expose electronics to water.
- ⚠ Keep the device in dry place.
- ⚠ Test the system before final installation.

15. FUTURE SCOPE

- → Add rain sensor for automation.
- → Integrate with weather forecasting.
- → Add solar power for energy saving.
- → Add data logging and charts.
- → Integrate with voice assistants.

16. WIRING DIAGRAM (CIRCUIT CONNECTIONS)



NOTE:

- Connect all GNDs together (Common Ground).
- Use separate 12V power supply for pump.

17. ARDUINO CODE (USING BLYNK LIBRARY)

```
#define BLYNK_TEMPLATE_ID   "TMPLxxxxxx"
#define BLYNK_TEMPLATE_NAME "Smart Garden Irrigation"
#define BLYNK_AUTH_TOKEN    "Your_Blynk_Auth_Token"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

#define SOIL_MOISTURE_PIN  A0
#define RELAY_PIN  D1
#define MOISTURE_THRESHOLD  450  // Adjust as required

BlynkTimer timer;

void setup() {
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  pinMode(RELAY_PIN,  OUTPUT);
  digitalWrite(RELAY_PIN, HIGH); // Pump OFF
  timer.setInterval(2000L, sendData);
}

void sendData() {
  int moistureValue = analogRead(SOIL_MOISTURE_PIN);
  Blynk.virtualWrite(V0, moistureValue);
  if (moistureValue < MOISTURE_THRESHOLD) {
    digitalWrite(RELAY_PIN, LOW);   // Pump ON
    Blynk.virtualWrite(V1, 1);      // Pump Status ON
  } else {
    digitalWrite(RELAY_PIN, HIGH); // Pump OFF
    Blynk.virtualWrite(V1, 0);      // Pump Status OFF
  }
}

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

BLYNK VIRTUAL PINS	
V Pin	Purpose
V0	Soil Moisture Value
V1	Pump Status (ON/OFF)
V2	Manual Pump Control
V3	Moisture Threshold

NOTE:

- Change Wi-Fi SSID, Password and Blynk Auth Token.
- Adjust MOISTURE_THRESHOLD as per soil type and sensor.

18. BLYNK APP DASHBOARD (EXAMPLE)

Widgets Explanation	
V0 (Gauge)	Displays real-time soil moisture level (0% – 100%).
V1 (LED)	Shows pump status ON or OFF.
V2 (Button)	Manual control to turn pump ON / OFF.
V3 (Slider)	Set moisture threshold level.

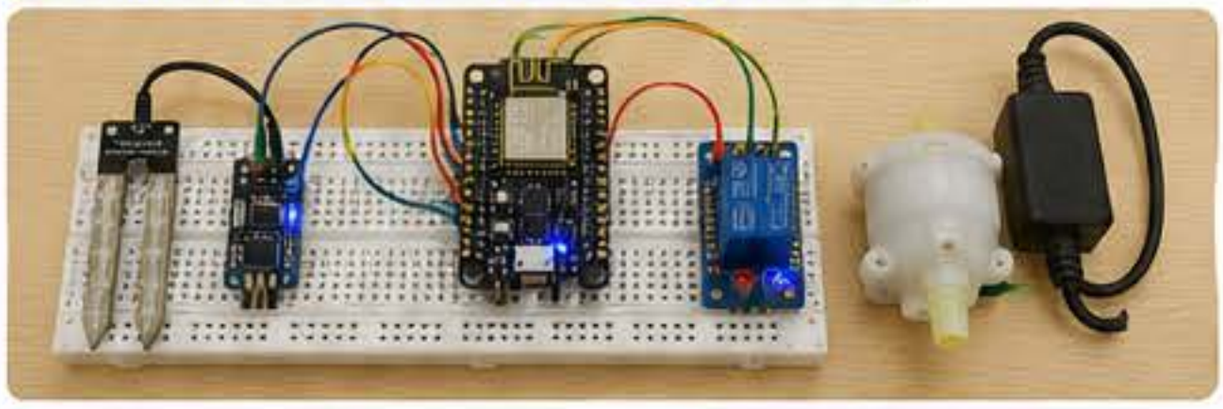
19. TEST CASES

Test Case No.	Condition	Expected Result	Actual Result	Status
TC-1	Moisture level > Threshold	Pump OFF	Same as Expected	PASS
TC-2	Moisture level < Threshold	Pump ON	Same as Expected	PASS
TC-3	Manual Pump ON from Blynk App	Pump turns ON	Same as Expected	PASS
TC-4	Manual Pump OFF from Blynk App	Pump turns OFF	Same as Expected	PASS
TC-5	Wi-Fi Disconnected	No update on App (Offline)	Same as Expected	PASS
TC-6	Reconnect Wi-Fi	Status updates normally	Same as Expected	PASS

OBSERVATION:

- The system monitors soil moisture and controls the pump automatically.
- Data is updated on Blynk App in real-time.
- Manual control works perfectly from the App.
- System is stable and reliable for garden irrigation.

20. HARDWARE IMPLEMENTATION (ACTUAL SETUP)



21. COMPONENT FUNCTIONS

Component	Function
ESP8266 (NodeMCU)	Reads sensor data, processes and sends to Blynk Cloud.
Soil Moisture Sensor	Measures soil moisture level.
Relay Module (5V)	Acts as switch to control water pump.
Water Pump (12V DC)	Pumps water to plants when soil is dry.
Blynk App	Displays data, sends alerts and allows manual control.
12V Power Supply	Provides power to water pump.

22. ADVANTAGES

- ✔ Real-time soil moisture monitoring.
- ✔ Automatic irrigation control.
- ✔ Remote monitoring and control.
- ✔ Saves water and improves efficiency.
- ✔ Easy to install and low cost.
- ✔ Enhances plant health.
- ✔ Can be extended with weather data.

23. APPLICATIONS

- Home Gardens
- Terrace Gardens
- Agriculture Fields
- Greenhouses
- Potted Plants
- Nurseries

24. SAFETY PRECAUTIONS

- ⚠ Use regulated 12V power supply.
- ⚠ Ensure correct wiring connections.
- ⚠ Do not operate pump without water.
- ⚠ Keep the device in dry place.
- ⚠ Do not expose electronics to moisture.
- ⚠ Test the system before final installation.

25. FUTURE SCOPE

- ➡ Add rain sensor for smart control.
- ➡ Integrate with weather forecasting.
- ➡ Add solar panel for power savings.
- ➡ Data logging on SD card.
- ➡ AI/ML based irrigation prediction.
- ➡ Integrate with voice assistants.

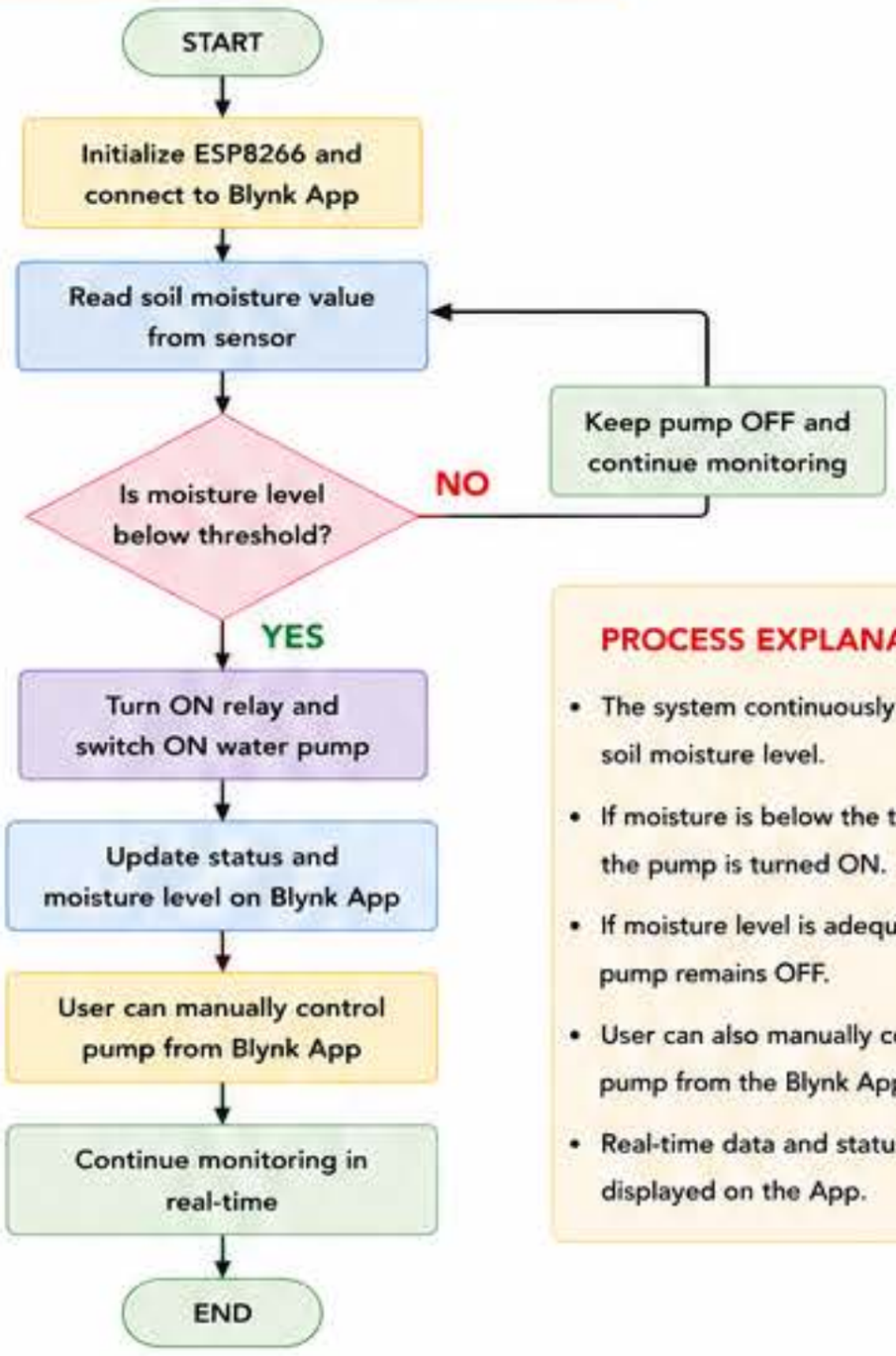
26. CONCLUSION

The IoT Smart Garden Irrigation System using ESP8266, Soil Moisture Sensor, Relay, Pump and Blynk App provides an efficient and automatic solution for irrigation management. The system monitors soil moisture in real-time and controls the water pump automatically to maintain optimal moisture level. It helps in saving water, improving plant health and can be monitored and controlled remotely through the Blynk App. This system is cost-effective, easy to implement and suitable for home gardens, agriculture fields, greenhouses and nurseries.

PROJECT OUTCOME

- Successfully designed and implemented IoT based Smart Garden Irrigation System.
- Achieved real-time monitoring and automatic pump control.
- Saves water, time and manual effort.
- System is reliable, efficient and user-friendly.

27. WORKING PROCESS FLOWCHART



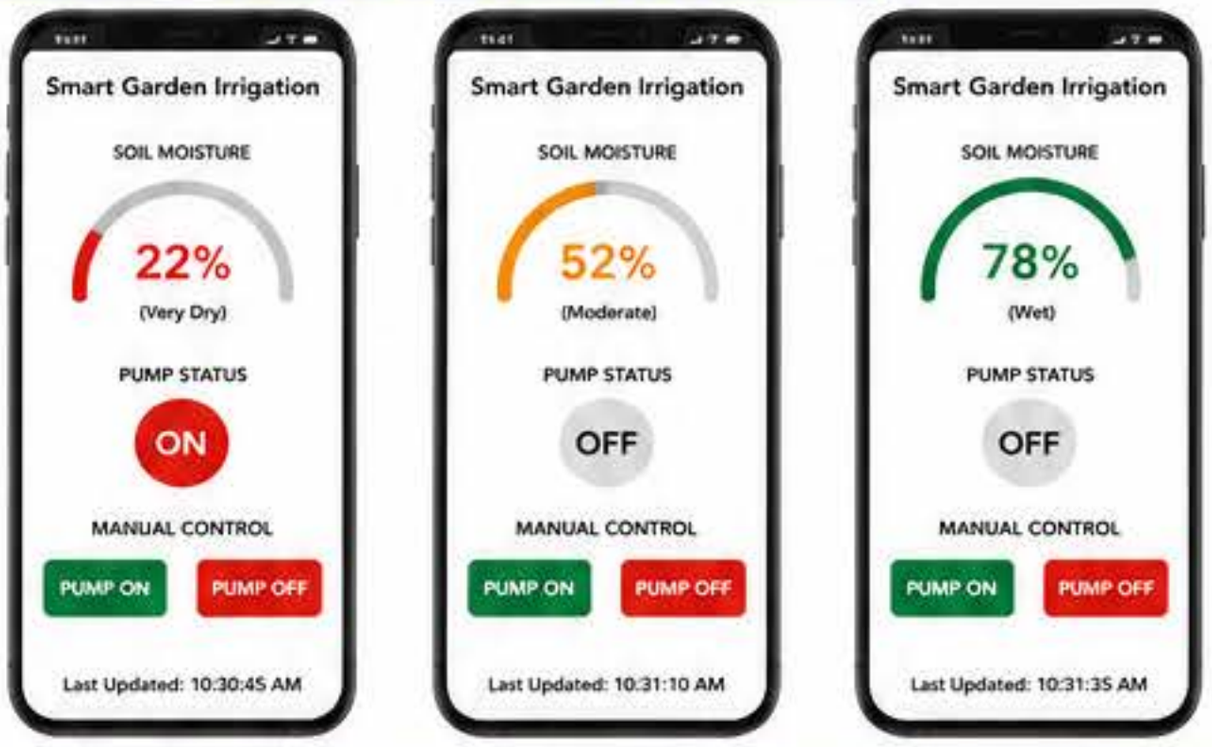
PROCESS EXPLANATION

- The system continuously checks soil moisture level.
- If moisture is below the threshold, the pump is turned ON.
- If moisture level is adequate, the pump remains OFF.
- User can also manually control the pump from the Blynk App.
- Real-time data and status are displayed on the App.

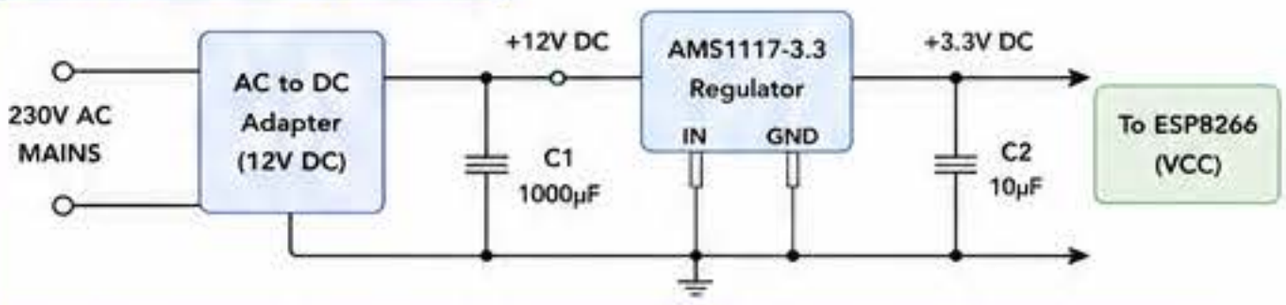
28. SOIL MOISTURE LEVEL GUIDE

Moisture Level (%)	Soil Condition	Status	Recommended Action
0 – 30 %	Very Dry	LOW	Pump ON (Irrigation Required)
30 – 60 %	Moderate	MEDIUM	Monitor (No Action)
60 – 100 %	Wet	HIGH	Pump OFF (Sufficient Moisture)

29. EXAMPLE – REAL TIME DATA ON BLYNK APP



30. POWER SUPPLY CIRCUIT



NOTE: Use regulated 12V for pump and relay. Use 3.3V output for ESP8266.

31. RELAY MODULE CONNECTION DETAILS

Relay Pin	Connection
VCC	5V from ESP8266 (VU Pin)
GND	GND
IN	Any Digital Pin (D1 in this project)
COM	+12V from Power Supply
NO	+12V to Pump Positive
Pump Negative	To Power Supply GND

NOTE: Use NO (Normally Open) for automatic irrigation control.

32. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
ESP8266 not connecting to Wi-Fi	Wrong Wi-Fi credentials	Check SSID and Password
Moisture value always 0%	Sensor not inserted properly / Dry soil	Insert sensor properly / Water the soil lightly
Moisture value always 100%	Sensor shorted / Wet soil	Check sensor pins / Use in normal soil
Pump not turning ON	Relay wiring incorrect / No 12V supply	Check wiring & 12V supply to pump
Data not updating on App	Internet issue / Blynk Auth Token incorrect	Check internet / Verify Auth Token
Manual control not working	Button widget not configured properly	Check V2 widget and pin mapping

33. SENSOR CALIBRATION (SOIL MOISTURE SENSOR)

1. Insert the sensor in completely dry soil. Note the value (should be near 0%).
2. Insert the sensor in waterlogged soil. Note the value (should be near 100%).
3. Adjust the threshold value in the code based on observed values.
4. Typical Threshold Range: 300 – 600 (Adjust as per soil type and crop).

Calibration Tip:
Different soil types give slightly different readings.
Always calibrate for accurate results.

34. ADVANTAGES

- ✔ Real-time soil moisture monitoring.
- ✔ Automatic irrigation control.
- ✔ Manual control option.
- ✔ Saves water and reduces wastage.
- ✔ Improves plant health and yield.
- ✔ Remote monitoring from anywhere.
- ✔ Easy to install and low maintenance.
- ✔ Cost-effective and energy efficient.

35. APPLICATIONS

- Home Gardens
- Terrace Gardens
- Agriculture Fields
- Greenhouses
- Nurseries
- Landscape Irrigation
- Potted Plants

36. SAFETY PRECAUTIONS

- ⚠ Use proper 12V power supply.
- ⚠ Ensure correct wiring connections.
- ⚠ Do not operate pump without water.
- ⚠ Keep the device in dry place.
- ⚠ Do not expose electronics to moisture.
- ⚠ Test the system before field use.
- ⚠ Keep the device out of reach of children.

37. FUTURE SCOPE

- Integrate rain sensor to avoid watering during rainfall.
- Add solar power supply for energy efficiency.
- Data logging and analytics for crop growth analysis.
- AI/ML based irrigation prediction based on weather and soil data.
- Add WhatsApp / SMS alerts for pump status and moisture level.
- Integrate camera for crop monitoring and disease detection.

PROJECT OUTCOME

- Successfully designed and implemented an IoT based Smart Garden Irrigation System.
- Achieved real-time monitoring of soil moisture and automated irrigation.
- System provides water conservation and improves plant health.
- System is reliable, efficient and user-friendly.

SKILLS LEARNED

- IoT integration with ESP8266 and Blynk App.
- Interfacing sensors, relays and pumps.
- Real-time data monitoring and control.
- Embedded system design and programming.

PROJECT-9

IOT SMART STREET LIGHTING SYSTEM USING BLYNK APP

USING ESP8266 (NodeMCU), LDR SENSOR, RELAY AND LED STREET LIGHT

1. AIM

To design and implement an IoT based smart street lighting system that automatically turns ON street lights at night and OFF during the day based on ambient light intensity. The system allows remote monitoring and manual control via Blynk App.

2. OBJECTIVES

- Automatically control street lights based on ambient light.
- Remotely monitor and control street lights using Blynk App.
- Reduce energy consumption.
- Provide real-time status updates.
- Improve maintenance and operational efficiency.

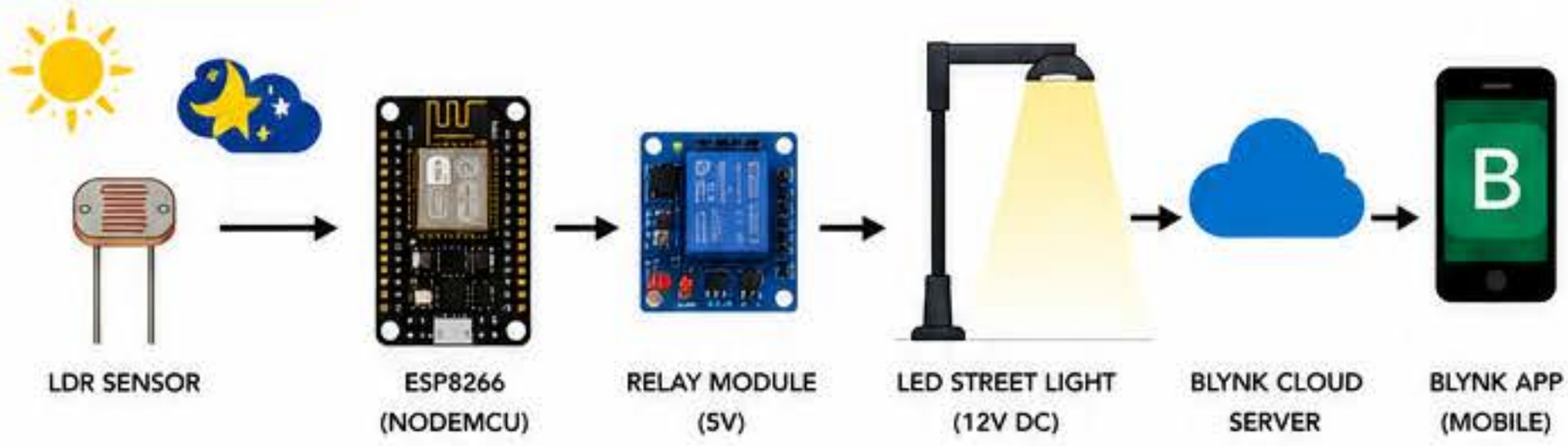
3. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	LDR Sensor (Light Dependent Resistor)	1
3	Relay Module (5V, 1 Channel)	1
4	LED Street Light (12V DC)	1
5	12V DC Power Supply	1
6	Jumper Wires	Several
7	Resistors (10kΩ, 1kΩ)	As Required

4. FEATURES

- Automatic ON/OFF based on light intensity.
- Manual ON/OFF control from Blynk App.
- Real-time monitoring of light status.
- Energy saving and cost effective.
- Remote access from anywhere.
- Easy to install and maintain.

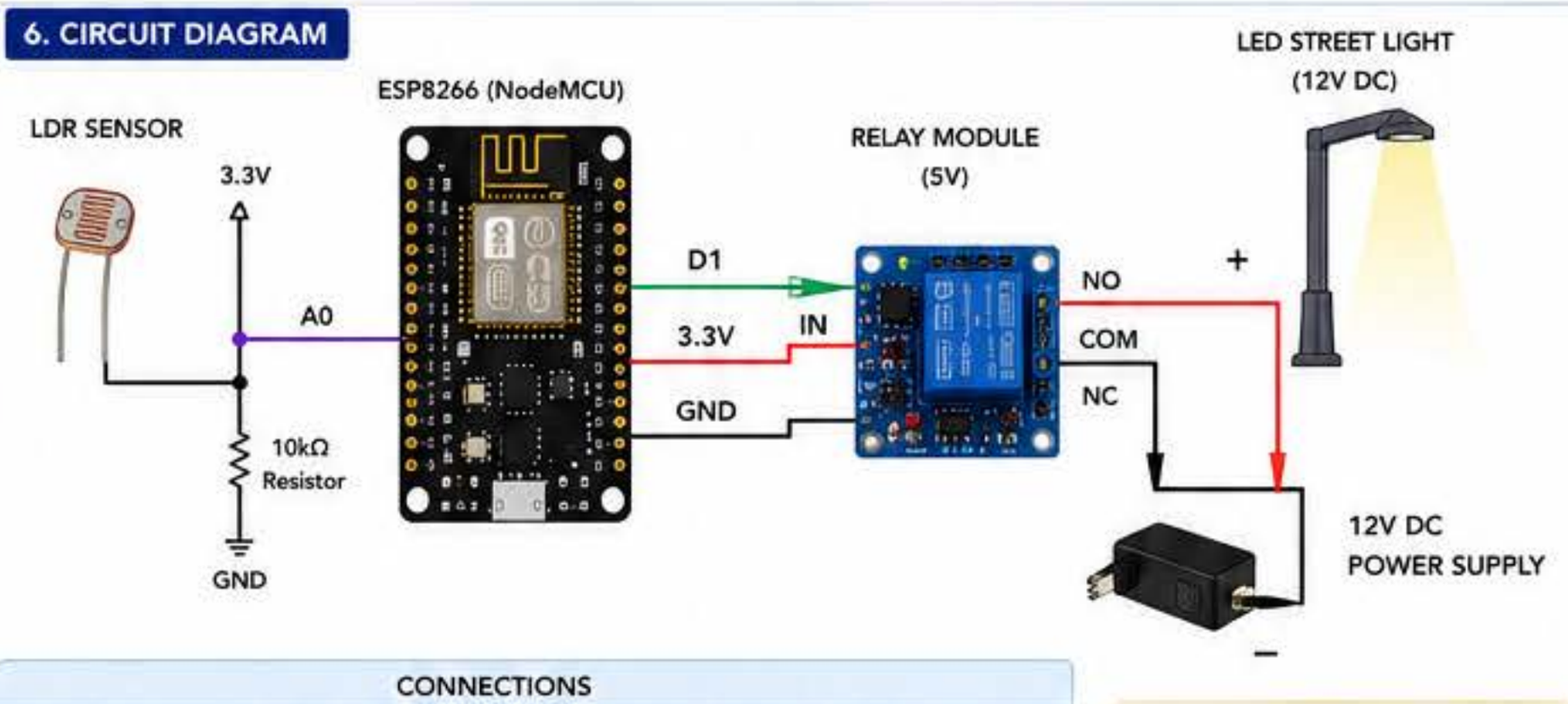
5. BLOCK DIAGRAM



WORKING:

The LDR sensor senses the ambient light intensity and sends data to ESP8266. If it is dark (below threshold), ESP8266 turns ON the street light through the relay module. During daylight, it turns OFF the light automatically. The user can also manually control the street light and monitor its status through Blynk App in real-time.

6. CIRCUIT DIAGRAM



CONNECTIONS			
LDR Sensor	NodeMCU (ESP8266)	Relay Module (5V)	Power Supply (12V)
VCC (3.3V)	3V3	VCC	+12V
GND	GND	GND	GND
A0	A0	-	-
D1	D1	IN	-
NO	-	NO	+12V (to Street Light +)
COM	-	COM	+12V (from Power Supply)
NC	-	NC	Not Connected

NOTE:

- Use common GND between ESP8266, Relay Module and Power Supply.
- Adjust LDR threshold in code as per environment.
- Ensure proper insulation and wiring for street light.

7. ESP8266 (NODEMCU)

- Wi-Fi enabled microcontroller
- 3.3V Operation
- Easy to program with Arduino IDE
- Connects to Blynk Cloud

8. LDR SENSOR

- Detects ambient light intensity
- Output voltage varies with light
- Low cost and easy to use

9. RELAY MODULE (5V)

- 1 Channel 5V Relay Module
- Isolates high voltage load
- Controls street light ON/OFF

10. LED STREET LIGHT (12V DC)

- Energy efficient LED light
- 12V DC Operation
- Long life and low power consumption

11. BLYNK APP OVERVIEW

- Real-time status (ON / OFF)
- Manual ON / OFF control
- Instant notifications
- Accessible anywhere

12. APPLICATIONS

- Street Lighting
- Smart Cities
- Highways & Roads
- Parks and Gardens
- Residential Areas
- Industrial Areas

13. ADVANTAGES

- Automatic and manual control
- Saves energy and reduces cost
- Remote monitoring from anywhere
- Real-time status and alerts
- Easy installation and maintenance
- Scalable for large areas

14. SAFETY PRECAUTIONS

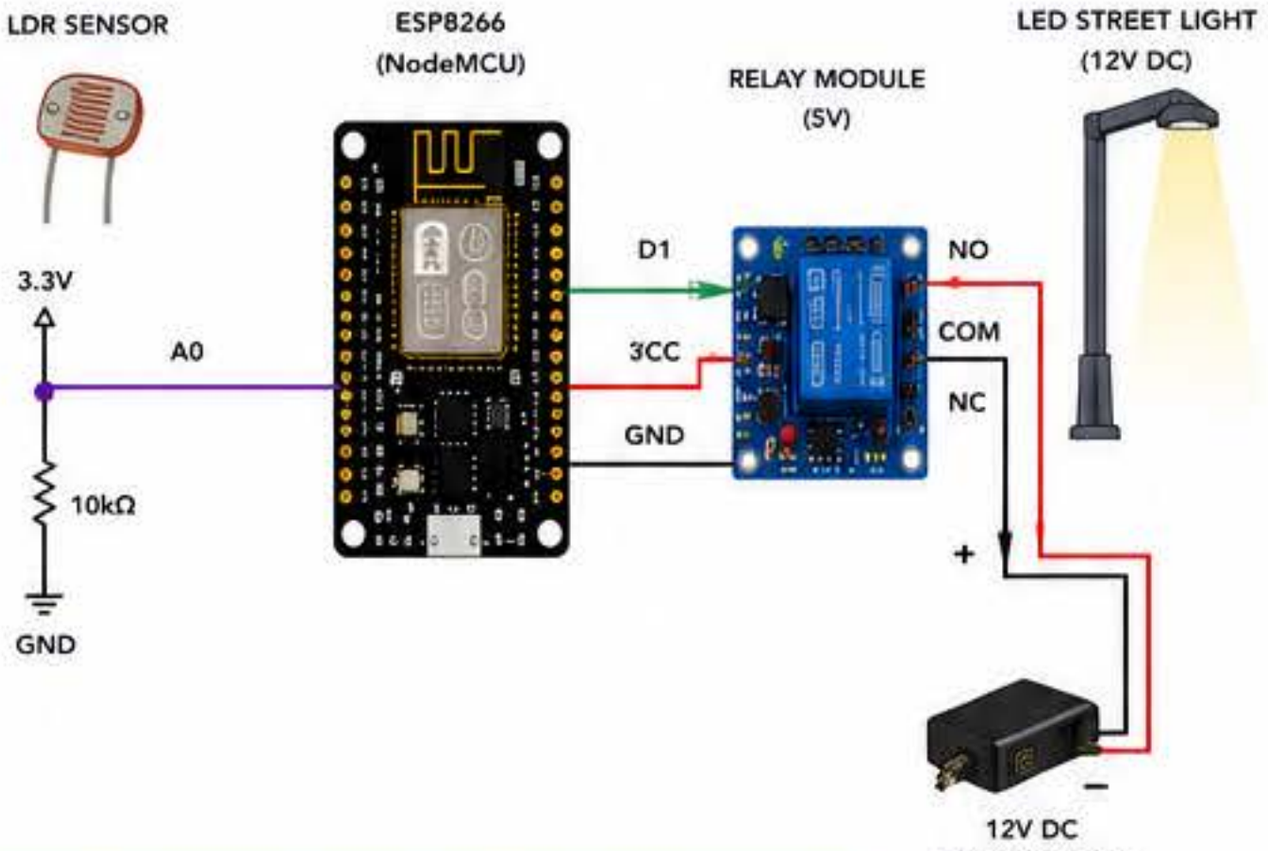
- Use regulated 12V power supply.
- Ensure correct wiring connections.
- Do not expose electronics to rain.
- Use proper insulation and enclosures.
- Keep the device out of reach of children.
- Test the system before final installation.

15. FUTURE SCOPE

- Add dimming control for brightness adjustment.
- Integrate with solar power system.
- Add fault detection and reporting.
- Integrate with Smart City IoT platforms.
- Add GSM/SMS alerts for failure.
- Add AI based failure prediction.

Dr V Purandhar Reddy, Narsimha Reddy Engineering College, Hyderabad.

16. WIRING DIAGRAM (CIRCUIT CONNECTIONS)



NOTE:

- Use common GND between ESP8266, Relay Module and Power Supply.
- LDR and voltage divider output is connected to A0 of ESP8266.
- Relay NO (Normally Open) is used to switch ON the street light.

17. ARDUINO CODE (USING BLYNK LIBRARY)

```
#define BLYNK_TEMPLATE_ID     "TMPLxxxxxx"
#define BLYNK_TEMPLATE_NAME   "Smart Street Lighting"
#define BLYNK_AUTH_TOKEN      "Your_Blynk_Auth_Token"

#define LDR_PIN   A0
#define RELAY_PIN D1
#define LDR_THRESHOLD  600  // Adjust as per environment

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

BlynkTimer timer;

void setup() {
  Serial.begin(115200);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);  // Light OFF initially
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid,pass);
  timer.setInterval(2000L, sendData);
}

void sendData() {
  int ldrValue = analogRead(LDR_PIN);
  Blynk.virtualWrite(V0, ldrValue);
  if (ldrValue < LDR_THRESHOLD) {
    digitalWrite(RELAY_PIN, HIGH); // Street Light ON
    Blynk.virtualWrite(V1, 1);     // Status ON
  } else {
    digitalWrite(RELAY_PIN, LOW);  // Street Light OFF
    Blynk.virtualWrite(V1, 0);     // Status OFF
  }
}
void loop() {
  Blynk.run();
  timer.run();
}
```

BLYNK VIRTUAL PINS	
V Pin	Purpose
V0	Ambient Light (LDR Value)
V1	Street Light Status (ON/OFF)
V2	Manual Control (ON/OFF)
V3	Light Intensity (for future use)

NOTE:

- Change Wi-Fi SSID, Password and Blynk Auth Token in the code.
- Adjust LDR_THRESHOLD based on ambient light conditions.
- Upload code to ESP8266 using Arduino IDE.

18. BLYNK APP DASHBOARD (EXAMPLE)

Widgets Explanation	
V0 (Gauge)	Displays real-time ambient light intensity from LDR.
V1 (LED)	Shows street light status ON or OFF.
V2 (Button)	Allows manual control of street light.
V3 (Slider)	(Optional) Reserved for brightness control in future.

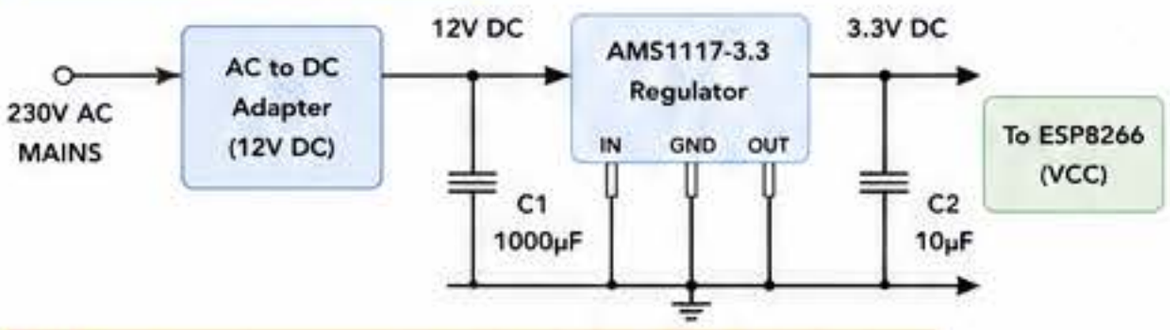
19. TEST CASES

Test Case No.	Condition	Expected Result	Actual Result	Status
TC-1	Bright Light (Day Time)	Street Light OFF	Same as Expected	PASS
TC-2	Dark Light (Night Time)	Street Light ON	Same as Expected	PASS
TC-3	Manual ON from App	Street Light ON	Same as Expected	PASS
TC-4	Manual OFF from App	Street Light OFF	Same as Expected	PASS
TC-5	Wi-Fi Disconnected	No update on App (Offline)	Same as Expected	PASS
TC-6	Reconnect Wi-Fi	Status updates normally	Same as Expected	PASS

OBSERVATION:

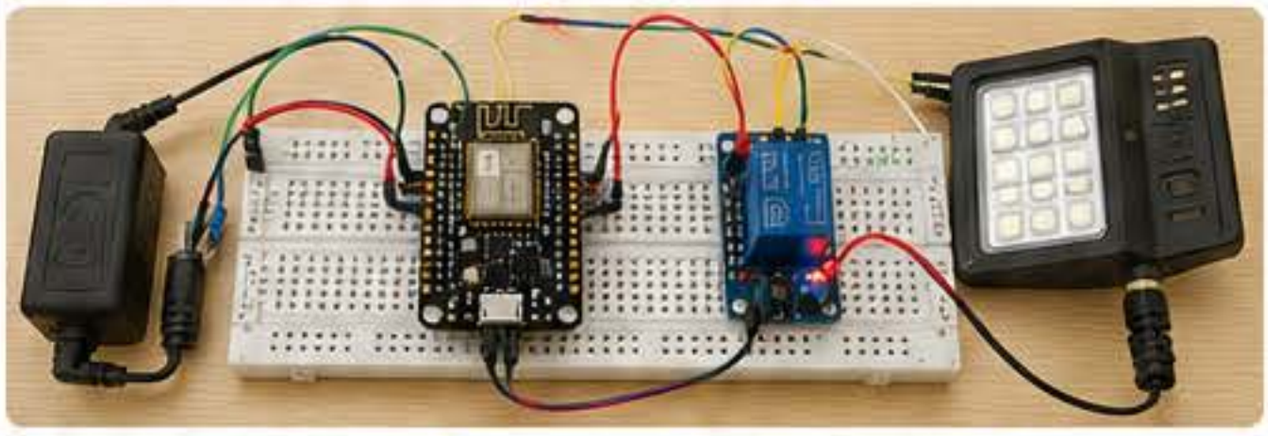
- The system automatically controls street light based on ambient light intensity.
- Manual override works from Blynk App.
- Real-time data is updated on Blynk App.
- System is stable and reliable for street lighting automation.

20. POWER SUPPLY CIRCUIT



NOTE: Use regulated 12V for street light and 3.3V output for ESP8266.

21. HARDWARE IMPLEMENTATION (ACTUAL SETUP)



22. ADVANTAGES

- ✔ Automatic ON/OFF based on ambient light.
- ✔ Remote monitoring and control via Blynk App.
- ✔ Energy saving and cost effective.
- ✔ Real-time notifications and updates.
- ✔ Easy installation and maintenance.
- ✔ Scalable for smart city applications.

23. APPLICATIONS

- Street Lighting Systems
- Smart Cities
- Highways & Parking Areas
- Parks & Gardens
- Industrial Areas
- Residential Communities

24. SAFETY PRECAUTIONS

- ⚠ Use regulated 12V power supply.
- ⚠ Ensure correct wiring connections.
- ⚠ Do not expose electronics to rain.
- ⚠ Use proper insulation and enclosures.
- ⚠ Keep the device out of reach of children.
- ⚠ Test the system before final installation.

25. FUTURE SCOPE

- ➡ Add brightness control (dimming).
- ➡ Integrate solar power system.
- ➡ Add fault detection and alerts.
- ➡ Add motion sensor for adaptive lighting.
- ➡ Integrate with Smart City IoT platforms.
- ➡ Add AI based energy optimization.

26. CONCLUSION

The IoT Smart Street Lighting System using ESP8266, LDR Sensor, Relay and LED Street Light provides an efficient, reliable and intelligent solution for modern street lighting automation. The system automatically controls street lights based on ambient light conditions and can also be controlled manually via Blynk App. It helps in saving energy, reducing operational cost and supports the development of smart cities.

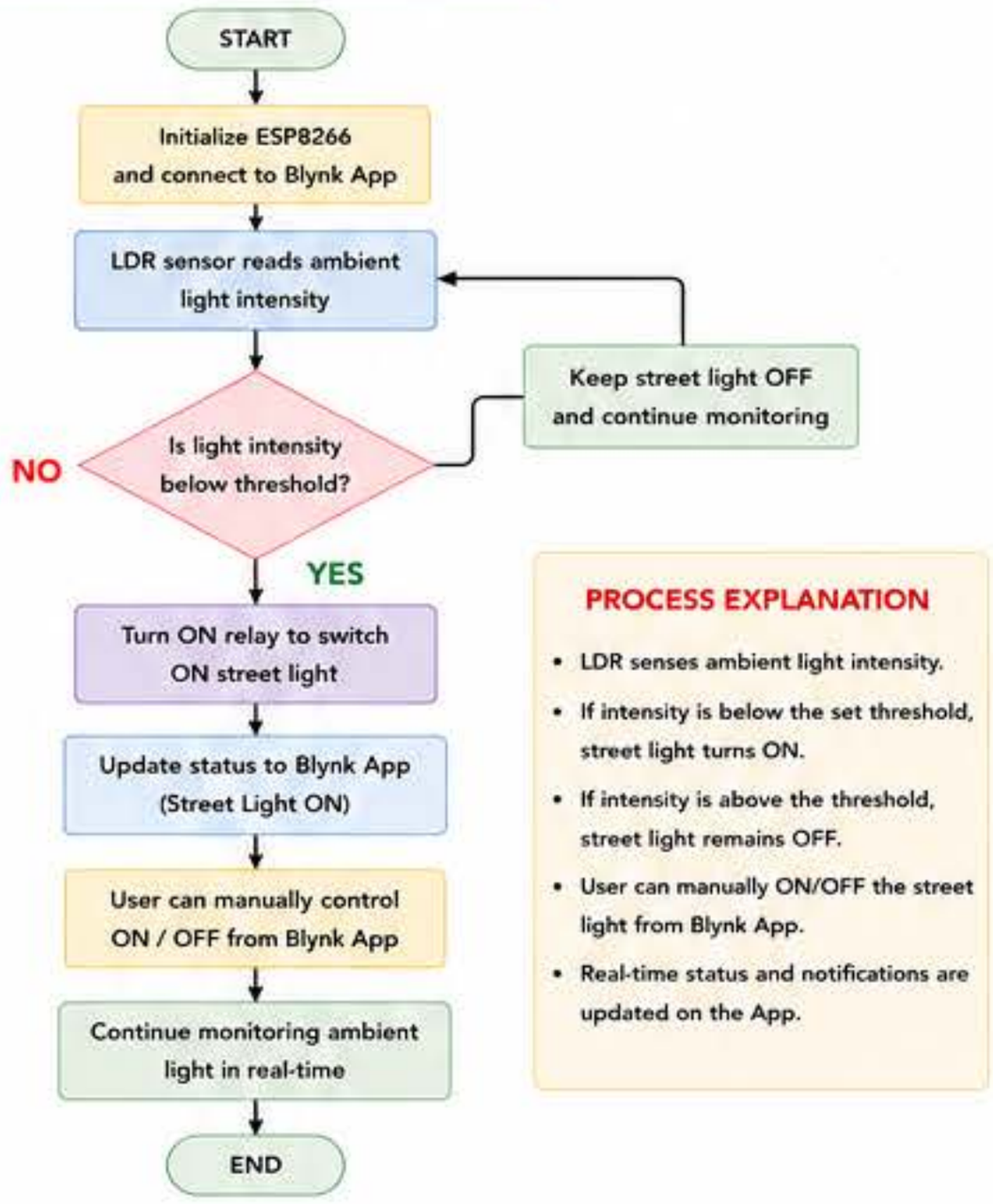
PROJECT OUTCOME

- Successfully designed and implemented IoT based Smart Street Lighting System.
- Achieved automatic ON/OFF control based on ambient light.
- Real-time monitoring and manual control via Blynk App.
- System is reliable, energy efficient and user-friendly.

SKILLS LEARNED

- IoT integration with ESP8266 and Blynk App.
- Interfacing LDR sensor and relay module.
- Real-time data monitoring and control.
- Embedded system design and programming.

27. WORKING PROCESS FLOWCHART



PROCESS EXPLANATION

- LDR senses ambient light intensity.
- If intensity is below the set threshold, street light turns ON.
- If intensity is above the threshold, street light remains OFF.
- User can manually ON/OFF the street light from Blynk App.
- Real-time status and notifications are updated on the App.

28. LIGHT INTENSITY LEVEL GUIDE

Ambient Light (Lux Approx.)	Condition	System Action	Street Light Status
0 – 30 Lux	Dark (Night)	Turn ON	ON
30 – 150 Lux	Low Light (Evening)	Turn ON	ON
150 – 300 Lux	Dusk / Dawn	May Turn OFF/ON (Based on Threshold)	Depends
> 300 Lux	Bright (Day Time)	Turn OFF	OFF

29. EXAMPLE – REAL TIME DATA ON BLYNK APP

30. RELAY MODULE CONNECTION DETAILS

Relay Pin	Connection
VCC	5V from ESP8266 (VU Pin)
GND	GND
IN	Digital Output Pin (D1)
COM	+12V from Power Supply
NO	To + of LED Street Light
NC	Not Connected
GND (Supply)	To – of Power Supply

NOTE: LED Street Light is connected between +12V (via Relay NO) and GND of the Power Supply.

31. TEST CASES

Test Case No.	Condition	Expected Result	Actual Result	Status
TC-1	Dark Environment (< 30 Lux)	Street Light ON	Same as Expected	PASS
TC-2	Low Light Environment (30 – 150 Lux)	Street Light ON	Same as Expected	PASS
TC-3	Dusk Environment (150 – 300 Lux)	As per Threshold	Same as Expected	PASS
TC-4	Bright Environment (> 300 Lux)	Street Light OFF	Same as Expected	PASS
TC-5	Manual ON from App	Street Light ON	Same as Expected	PASS
TC-6	Manual OFF from App	Street Light OFF	Same as Expected	PASS
TC-7	Wi-Fi Disconnected	No update on App (Offline)	Same as Expected	PASS

OBSERVATION:

- The system automatically controls street light based on ambient light intensity.
- Manual override through Blynk App works perfectly.
- Real-time status is updated on the App.
- System is stable, reliable and effective for smart city lighting.

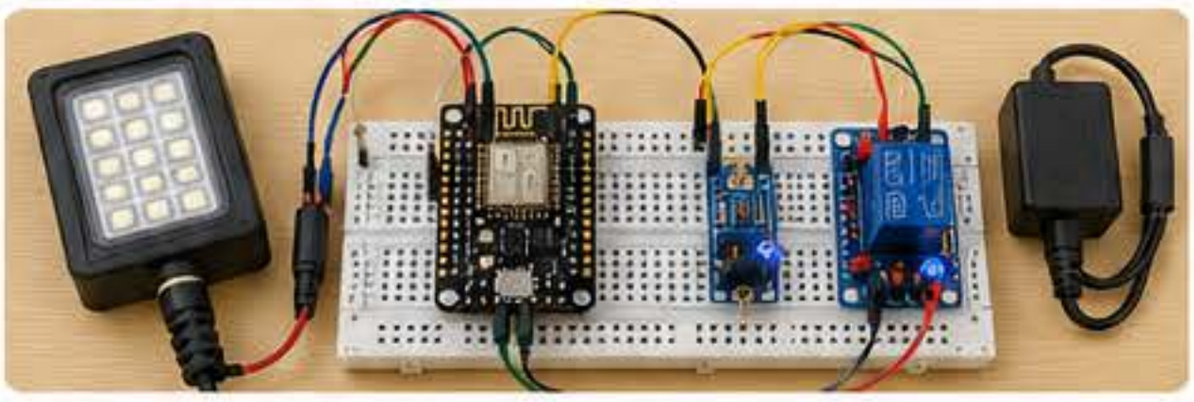
32. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
ESP8266 not connecting to Wi-Fi	Wrong Wi-Fi credentials	Check SSID and Password
LDR value always 0 or 1023	Loose connection / Wrong wiring	Check wiring and connections
Street light always ON	LDR not detecting / Threshold too high	Check LDR, adjust threshold
Street light always OFF	Threshold too low	Increase LDR threshold value
Relay not switching	Wrong wiring / No 12V supply	Check wiring and 12V supply
No update on Blynk App	Wi-Fi issue / Auth Token incorrect	Check internet and Auth Token
Manual control not working	Widget not configured properly	Check Blynk widget and pin mapping

33. COMPONENT FUNCTIONS

Component	Function
ESP8266 (NodeMCU)	Wi-Fi enabled microcontroller, processes data and connects to Blynk
LDR Sensor	Detects ambient light intensity
Relay Module (5V)	Switches ON/OFF the street light
LED Street Light (12V DC)	Provides illumination
12V DC Power Supply	Powers the street light and relay module
Blynk App	Monitors and controls street light remotely

34. HARDWARE SNAPSHOT



35. APPLICATIONS

- Smart Street Lighting Systems
- Smart Cities & Municipal Corporations
- Highways, Roads & Expressways
- College & University Campuses
- Residential Communities
- Parks, Gardens & Public Places

36. ADVANTAGES

- Automatic control based on ambient light.
- Remote monitoring and control.
- Energy saving and cost effective.
- Real-time alerts and notifications.
- Reduces manual effort.
- Easy installation and maintenance.
- Improves operational efficiency.
- Environment friendly.

37. FUTURE SCOPE

- Integrate with solar power system for energy saving.
- Add dimming control using PWM for brightness adjustment.
- AI based adaptive lighting using traffic and weather data.
- Add CCTV and security monitoring integration.
- Data analytics and prediction for maintenance and energy optimization.
- Integrate with Smart City IoT platforms and cloud dashboards.

PROJECT OUTCOME

- Successfully designed and implemented an IoT based Smart Street Lighting System.
- Achieved automatic lighting control based on ambient light intensity.
- Enabled remote monitoring and manual control via Blynk App.
- Reduced energy consumption and improved operational efficiency.
- System is reliable, scalable and ready for real-world smart city applications.

SKILLS LEARNED

- IoT integration with ESP8266 and Blynk App.
- Sensor interfacing and data processing.
- Relay control and power management.
- Real-time monitoring and automation.
- Embedded system design and programming.

Dr V Purandhar Reddy, Narsimha Reddy Engineering College, Hyderabad.

PROJECT-10

IOT SMART PARKING SYSTEM
USING BLYNK APP

USING ESP8266 (NodeMCU), IR SENSORS, SERVO MOTOR AND LCD DISPLAY

1. AIM

To design and implement an IoT based Smart Parking System that helps in detecting available parking slots and controlling entry/exit using a servo barrier. The system provides real-time parking status on LCD display and Blynk App.

2. OBJECTIVES

- Detect vehicle presence in each slot using IR sensors.
- Display available slots count on LCD and Blynk App.
- Control entry barrier using servo motor.
- Provide real-time updates and alerts.
- Improve parking efficiency and reduce congestion.

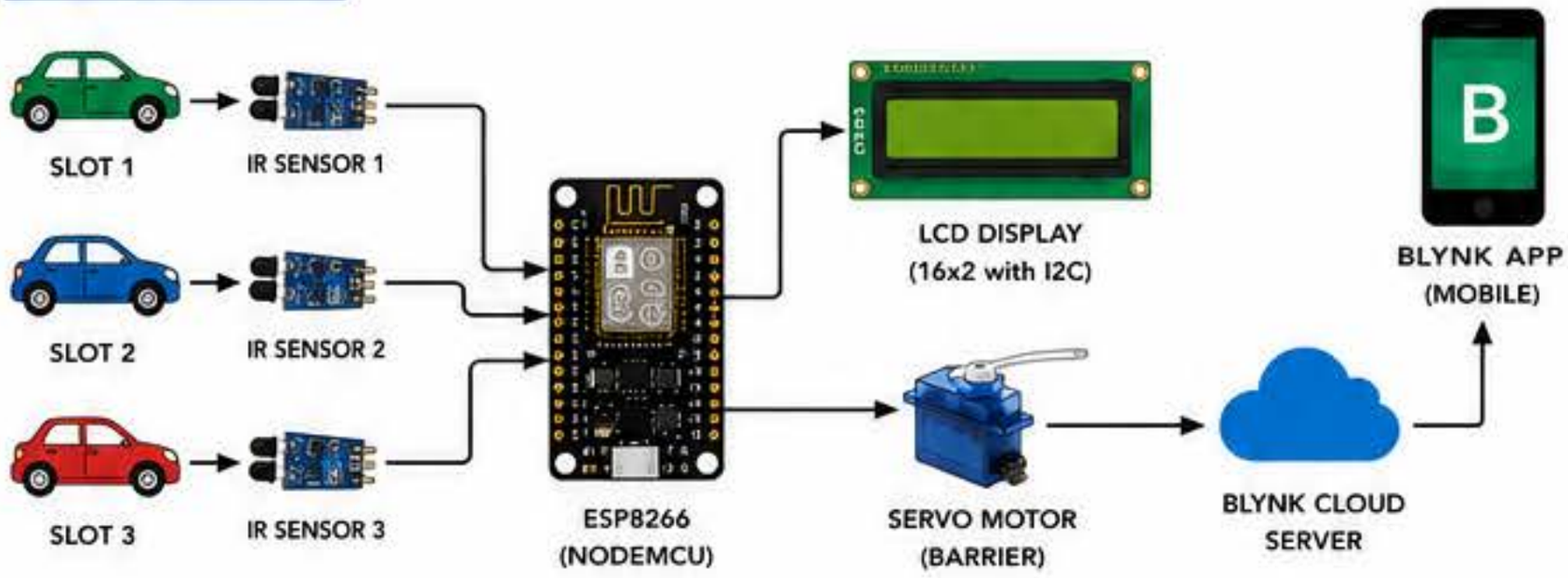
3. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	IR Sensor (Obstacle Sensor)	3
3	Servo Motor (SG90)	1
4	16x2 LCD Display	1
5	I2C LCD Module (PCF8574)	1
6	5V Power Supply	1
7	Jumper Wires	Several

4. FEATURES

- Real-time parking slot detection.
- Automatic barrier control.
- LCD display for local viewing.
- Blynk App for remote monitoring.
- Alerts and notifications.
- Easy to install and user-friendly.

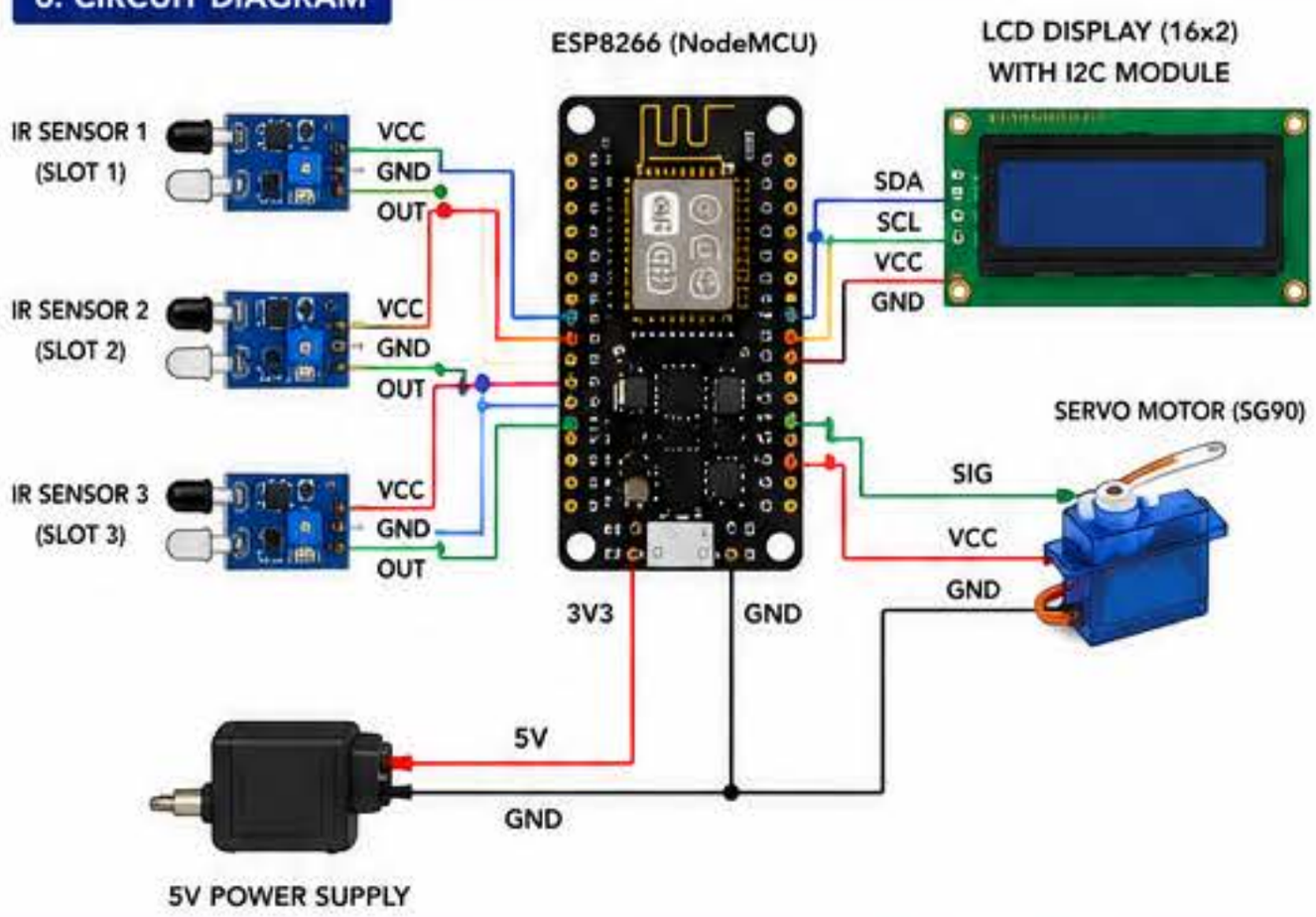
5. BLOCK DIAGRAM



WORKING:

IR sensors detect vehicles in each parking slot and send signals to ESP8266. The ESP8266 counts the available slots and displays the information on LCD and Blynk App. When a vehicle approaches, the servo motor opens the barrier if slots are available; otherwise, access is denied.

6. CIRCUIT DIAGRAM



NOTE:

- All IR sensor VCC to 3.3V of NodeMCU.
- All GNDs must be common.
- Servo motor should be powered from external 5V supply.
- Connect I2C LCD to D1 (SCL) and D2 (SDA).

7. ESP8266 (NODEMCU)

- Wi-Fi enabled microcontroller
- 3.3V Operation
- Easy to program using Arduino IDE
- Connects to Blynk Cloud

8. IR SENSOR (OBSTACLE SENSOR)

- Detects obstacle/vehicle presence
- Digital output (HIGH/LOW)
- Adjustable sensitivity
- 3.3V to 5V Operation

9. SERVO MOTOR (SG90)

- 180° rotation
- Control barrier gate
- Lightweight and reliable
- 4.8V – 6V Operation

10. LCD DISPLAY (16x2 with I2C)

- 16x2 Character Display
- I2C Interface (2 pins only)
- Displays available slots count and status

11. BLYNK APP OVERVIEW

- Real-time slot status
- Available slots count
- Barrier status
- Instant notifications
- Remote monitoring

12. APPLICATIONS

- Shopping Malls
- Office Buildings
- Hospitals
- Airports
- Educational Institutions
- Residential Complexes

13. ADVANTAGES

- Real-time parking availability
- Reduces waiting time
- Efficient space utilization
- Remote monitoring
- Improves security
- User-friendly and cost effective

14. SAFETY PRECAUTIONS

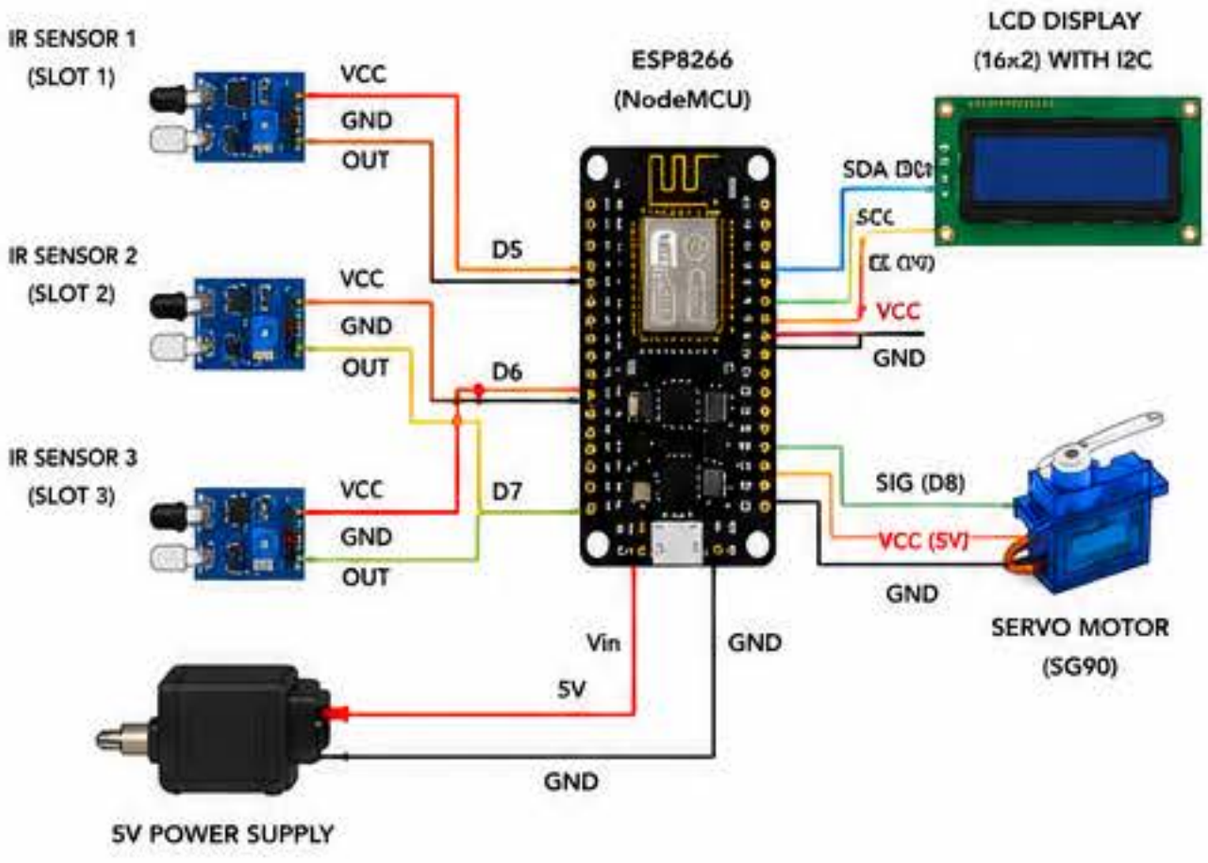
- Use regulated 5V power supply.
- Ensure proper wiring connections.
- Do not expose electronics to water.
- Keep the device in dry place.
- Secure the barrier mechanism.
- Test the system before final installation.

15. FUTURE SCOPE

- Add camera for vehicle recognition.
- Integrate RFID for access control.
- Add payment gateway for parking fee.
- Use ultrasonic sensors for better accuracy.
- Develop web dashboard for admin.
- AI based parking slot prediction.

Dr V Purandhar Reddy, Narsimha Reddy Engineering College, Hyderabad.

16. WIRING DIAGRAM (CIRCUIT CONNECTIONS)



NOTE:

- All IR sensors VCC to 3.3V of NodeMCU.
- All GNDs must be common.
- Servo motor powered by external 5V supply (Do not power from NodeMCU).
- LCD I2C connections: D1 (SCL), D2 (SDA).

17. ARDUINO CODE (USING BLYNK LIBRARY)

```
#define BLYNK_TEMPLATE_ID       "TMPLxxxxxx"
#define BLYNK_TEMPLATE_NAME     "Smart Parking System"
#define BLYNK_AUTH_TOKEN        "Your_Blynk_Auth_Token"

#define IR1  D5     // Slot 1
#define IR2  D6     // Slot 2
#define IR3  D7     // Slot 3
#define SERVO_PIN  D8
#define TRIG_PIN D4     // (optional for future use)
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>
char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";
BlynkTimer timer;
Servo gateServo;
void setup() {
  Serial.begin(115200);
  pinMode(IR1, INPUT);
  pinMode(IR2, INPUT);
  pinMode(IR3, INPUT);
  gateServo.attach(SERVO_PIN);
  gateServo.write(0);     // Barrier Closed
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval (1000L, sendData);
}
void sendData() {
  int s1 = digitalRead(IR1);  // LOW = vehicle present
  int s2 = digitalRead(IR2);
  int s3 = digitalRead(IR3);
  int available = 3 - ((s1==LOW) + (s2==LOW) + (s3==LOW));
  Blynk.virtualWrite(V0, available);    // Available Slots
  Blynk.virtualWrite(V1, s1==LOW);      // Slot 1 Status
  Blynk.virtualWrite(V2, s2==LOW);      // Slot 2 Status
  Blynk.virtualWrite(V3, s3==LOW);      // Slot 3 Status
}
BLYNK_WRITE(V4) {          // Manual Gate Control
  int val = param.asInt();
  if(val == 1) gateServo.write(90);   // Open
  else gateServo.write(0);            // Close
}
void loop() {
  Blynk.run();
  timer.run();
}
```

BLYNK VIRTUAL PINS	
V Pin	Purpose
V0	Available Slots Count
V1	Slot 1 Status (1=Occupied)
V2	Slot 2 Status (1=Occupied)
V3	Slot 3 Status (1=Occupied)
V4	Gate Control (1=Open, 0=Close)

NOTE:

- Update Wi-Fi SSID, Password and Blynk Auth Token in the code.
- Upload code to NodeMCU using Arduino IDE.
- Ensure all connections are correct before powering.

18. BLYNK APP DASHBOARD (EXAMPLE)

Widgets Explanation	
V0 (Gauge)	Displays the number of available slots.
V1 (LED)	Shows status of Slot 1.
V2 (LED)	Shows status of Slot 2.
V3 (LED)	Shows status of Slot 3.
V4 (Button)	Manual open/close control for barrier.

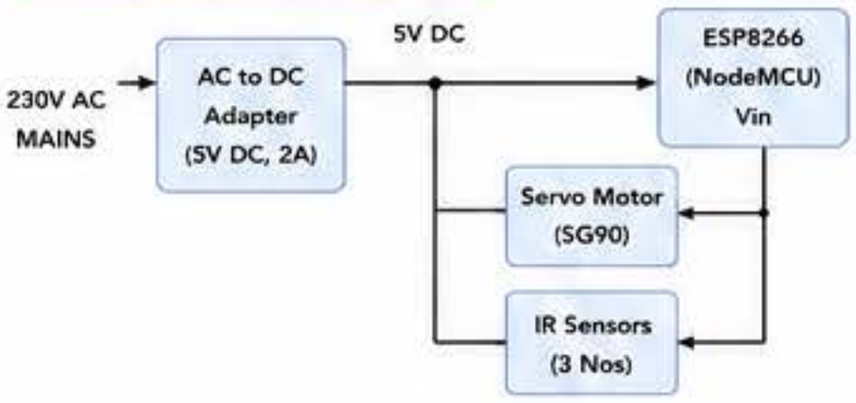
19. TEST CASES

Test Case No.	Condition	Expected Result	Actual Result	Status
TC-1	All slots empty	Available slots = 3 All LEDs = Green	Same as Expected	PASS
TC-2	One slot occupied	Available slots = 2 One LED = Red	Same as Expected	PASS
TC-3	Two slots occupied	Available slots = 1 Two LEDs = Red	Same as Expected	PASS
TC-4	All slots occupied	Available slots = 0 All LEDs = Red	Same as Expected	PASS
TC-5	Vehicle enters (Gate closed)	Gate opens automatically	Same as Expected	PASS
TC-6	Vehicle exits	Gate closes automatically	Same as Expected	PASS
TC-7	Manual open from Blynk	Gate opens	Same as Expected	PASS
TC-8	Manual close from Blynk	Gate closes	Same as Expected	PASS

OBSERVATION:

- The system correctly detects vehicle presence and updates slot availability.
- Gate operation is reliable and controlled by servo motor.
- Real-time data is updated on LCD and Blynk App without delay.

20. POWER SUPPLY CIRCUIT



NOTE: Use external 5V supply with sufficient current for Servo Motor.

21. HARDWARE IMPLEMENTATION (ACTUAL SETUP)

20. COMPONENT FUNCTIONS

Component	Function
IR Sensor (Obstacle)	Detects vehicle presence in slot.
ESP8266 (NodeMCU)	Processes data and connects to Wi-Fi.
Servo Motor (SG90)	Controls the parking barrier.
LCD Display (16x2, I2C)	Displays slot availability and status.
Blynk App	Provides remote monitoring and control.
Power Supply (5V)	Powers the system components.

23. ADVANTAGES

- ✔ Real-time parking slot monitoring.
- ✔ Reduces traffic and searching time.
- ✔ Improves parking space utilization.
- ✔ Remote control of barrier.
- ✔ Cost-effective and easy to implement.
- ✔ Scalable for large parking areas.

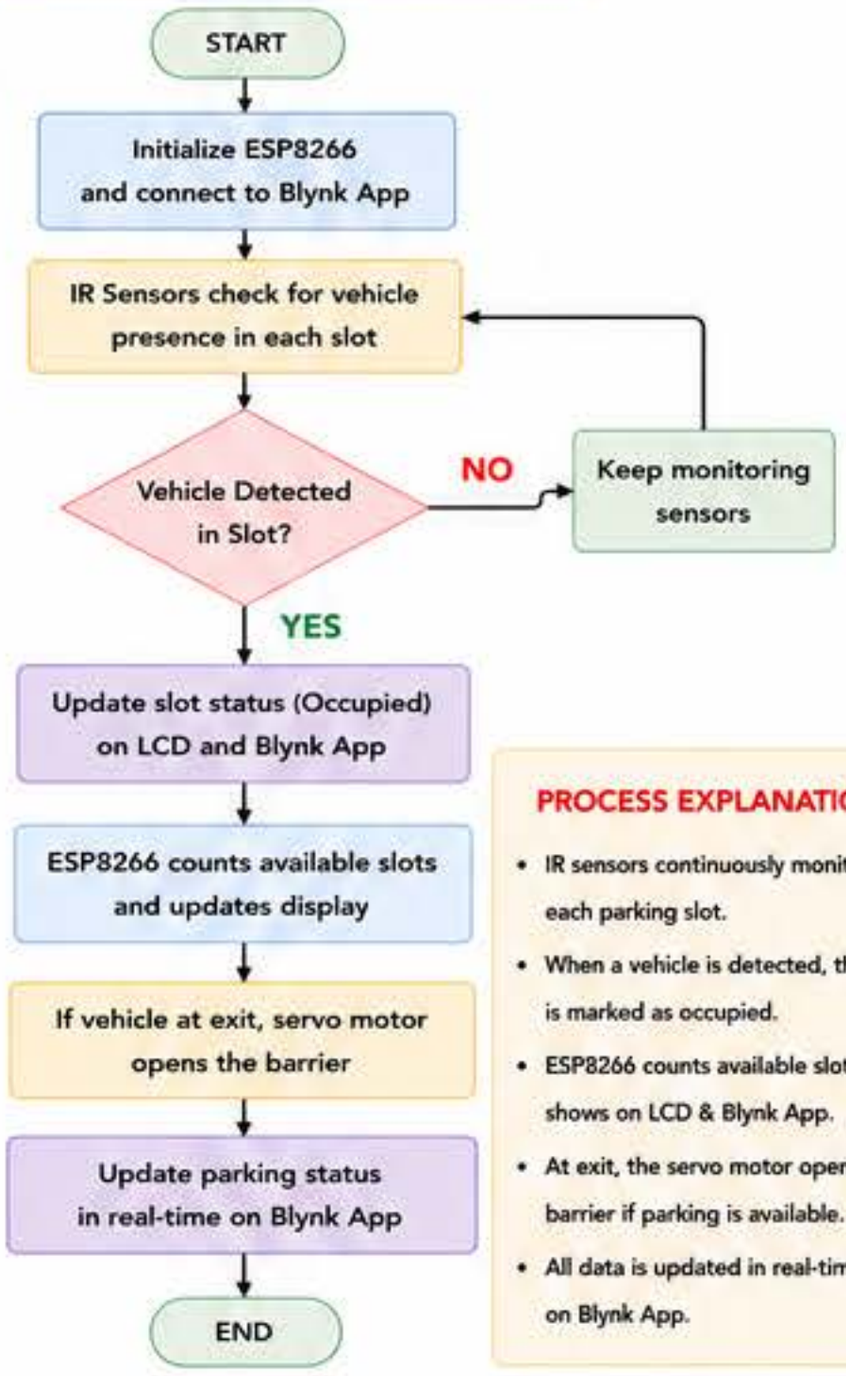
24. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
IR sensor not detecting	Wrong wiring / Sensitivity low	Check connections / Adjust position
LCD not displaying	I2C address wrong / Wiring issue	Check SDA (D2) and SCL (D1)
Blynk not connecting	Wrong Wi-Fi / Auth Token	Check internet / Auth Token
Gate not opening	Servo connection / Power issue	Check SIG pin and 5V supply
Slot count not correct	Sensor logic inverted	Check code and sensor alignment

24. FUTURE SCOPE

- Integrate CCTV camera for security surveillance.
- Add QR code based entry for registered users.
- Integrate digital payment system for parking fees.
- Data analytics and report generation for admin.
- Add license plate recognition for automation.
- Cloud storage for historical data and analytics.
- Multi-level parking support with map view.
- AI based prediction for parking space availability.

25. WORKING PROCESS FLOWCHART



PROCESS EXPLANATION

- IR sensors continuously monitor each parking slot.
- When a vehicle is detected, the slot is marked as occupied.
- ESP8266 counts available slots and shows on LCD & Blynk App.
- At exit, the servo motor opens the barrier if parking is available.
- All data is updated in real-time on Blynk App.

26. PARKING SLOT STATUS LOGIC

IR1 (Slot 1)	IR2 (Slot 2)	IR3 (Slot 3)	Slot Status	LCD Display	Blynk App Display
HIGH (No Vehicle)	HIGH (No Vehicle)	HIGH (No Vehicle)	All Empty	Available Slots: 3	3
LOW (Vehicle)	HIGH (No Vehicle)	HIGH (No Vehicle)	Slot 1 Occupied	Available Slots: 2	2
HIGH (No Vehicle)	LOW (Vehicle)	HIGH (No Vehicle)	Slot 2 Occupied	Available Slots: 2	2
HIGH (No Vehicle)	HIGH (No Vehicle)	LOW (Vehicle)	Slot 3 Occupied	Available Slots: 2	2
LOW (Vehicle)	LOW (Vehicle)	HIGH (No Vehicle)	Slots 1 & 2 Occupied	Available Slots: 1	1
LOW (Vehicle)	HIGH (No Vehicle)	LOW (Vehicle)	Slots 1 & 3 Occupied	Available Slots: 1	1
HIGH (No Vehicle)	LOW (Vehicle)	LOW (Vehicle)	Slots 2 & 3 Occupied	Available Slots: 1	1
LOW (Vehicle)	LOW (Vehicle)	LOW (Vehicle)	All Slots Occupied	No Parking	0

NOTE: IR Sensor Output → HIGH (No vehicle), LOW (Vehicle detected)

27. EXAMPLE – REAL TIME DATA ON BLYNK APP

28. LCD DISPLAY EXAMPLES

(a) All Slots Empty

(b) Slot 1 Occupied

(c) Slots 1 & 2 Occupied

(d) All Slots Occupied

Note: 0 = Empty, 1 = Occupied

29. TEST CASES

Test Case No.	Condition	Expected Result (LCD)	Expected Result (Blynk App)	Actual Result	Status
TC-1	All slots empty	Available Slots: 3	3	Same as Expected	PASS
TC-2	Slot 1 occupied	Available Slots: 2	2	Same as Expected	PASS
TC-3	Slot 2 occupied	Available Slots: 2	2	Same as Expected	PASS
TC-4	Slot 3 occupied	Available Slots: 2	2	Same as Expected	PASS
TC-5	Slots 1 & 2 occupied	Available Slots: 1	1	Same as Expected	PASS
TC-6	Slots 1 & 3 occupied	Available Slots: 1	1	Same as Expected	PASS
TC-7	Slots 1 & 3 occupied	Available Slots: 1	1	Same as Expected	PASS
TC-8	All slots occupied	No Parking	0	Same as Expected	PASS
TC-9	Exit gate open	Barrier OPEN	OPEN	Same as Expected	PASS
TC-10	Exit gate close	Barrier CLOSED	CLOSED	Same as Expected	PASS

OBSERVATION:

- System accurately detects vehicle presence in each slot using IR sensors.
- Available slots count is correctly displayed on LCD and Blynk App.
- Barrier control using servo motor works as expected.
- Real-time updates are fast and reliable.

30. APPLICATIONS & ADVANTAGES

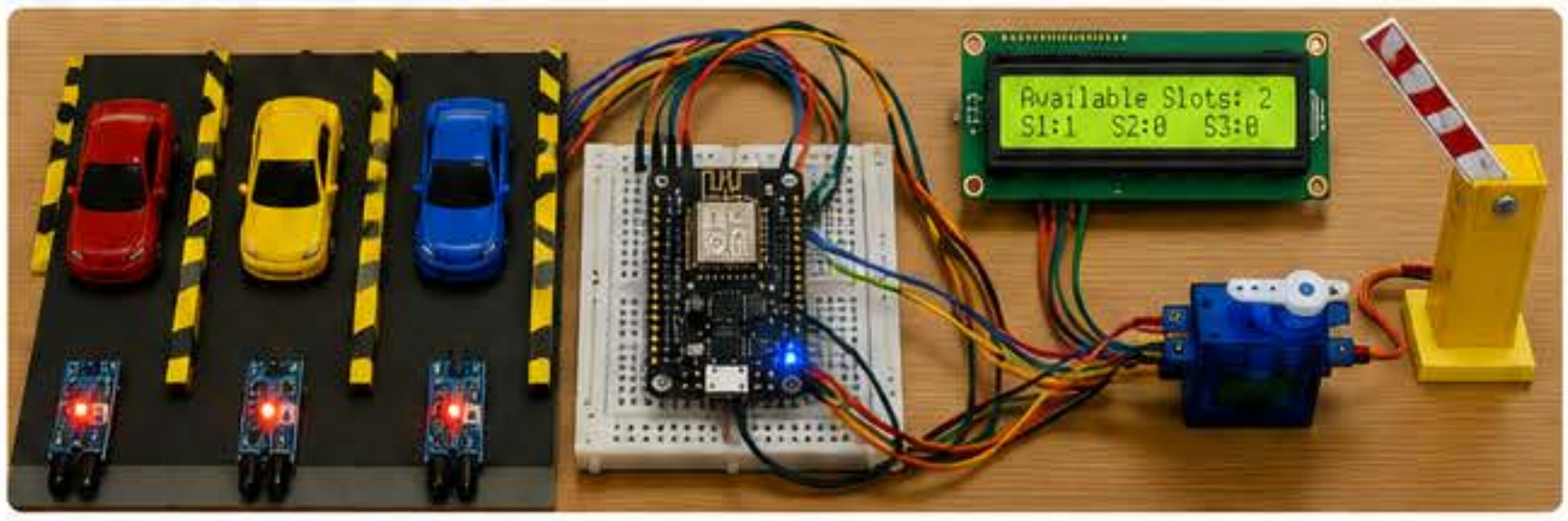
APPLICATIONS

- Shopping Malls
- Office Buildings
- Hospitals
- Airports
- Educational Institutions
- Residential Complexes
- Public Parking Areas

ADVANTAGES

- Real-time slot monitoring.
- Reduces traffic congestion.
- Saves time and fuel.
- Improves parking space utilization.
- Remote access and monitoring.
- Cost-effective and scalable.
- Enhances user convenience.

31. HARDWARE SNAPSHOT



FUTURE SCOPE

- Mobile payment integration for parking fee.
- Automatic number plate recognition (ANPR).
- Cloud based centralized parking management.
- Multi-level parking support.
- Solar powered parking system.
- AI based prediction of parking availability.
- Integration with smart city infrastructure.

PROJECT-11

IOT SMART ENERGY METER
USING BLYNK APP

USING ESP8266 (NodeMCU), PZEM-004T SENSOR AND BLYNK CLOUD

1. AIM

To design and implement an IoT based Smart Energy Meter that measures voltage, current, power, energy and cost, and monitors the data in real-time on Blynk App to promote energy awareness and efficient usage.

2. OBJECTIVES

- Measure voltage, current, power factor, frequency and energy consumption.
- Display parameters on Blynk App in real-time.
- Store energy consumption (kWh) and calculate cost.
- Provide alerts for high power consumption.
- Enable remote monitoring and historical data tracking.

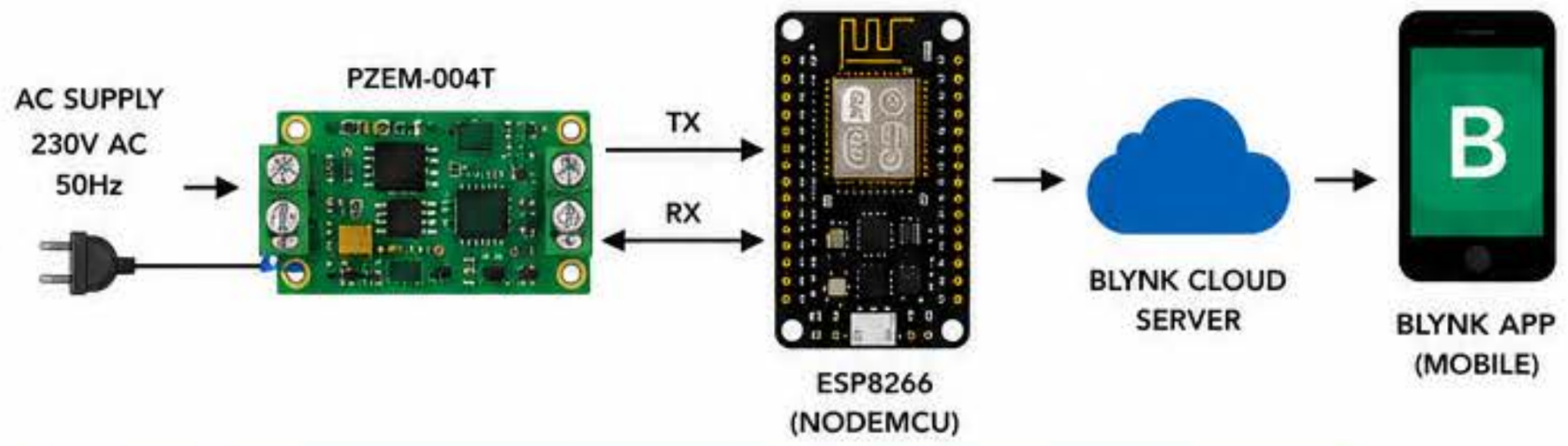
3. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	PZEM-004T Energy Meter Module	1
3	5V DC Power Supply	1
4	Connecting Wires	Several

4. FEATURES

- Real-time measurement of electrical parameters.
- Automatic calculation of energy (kWh) and cost.
- Live data display on Blynk App.
- Alerts when power or energy exceeds limit.
- User-friendly and low-cost solution.
- Easy to install and scalable.

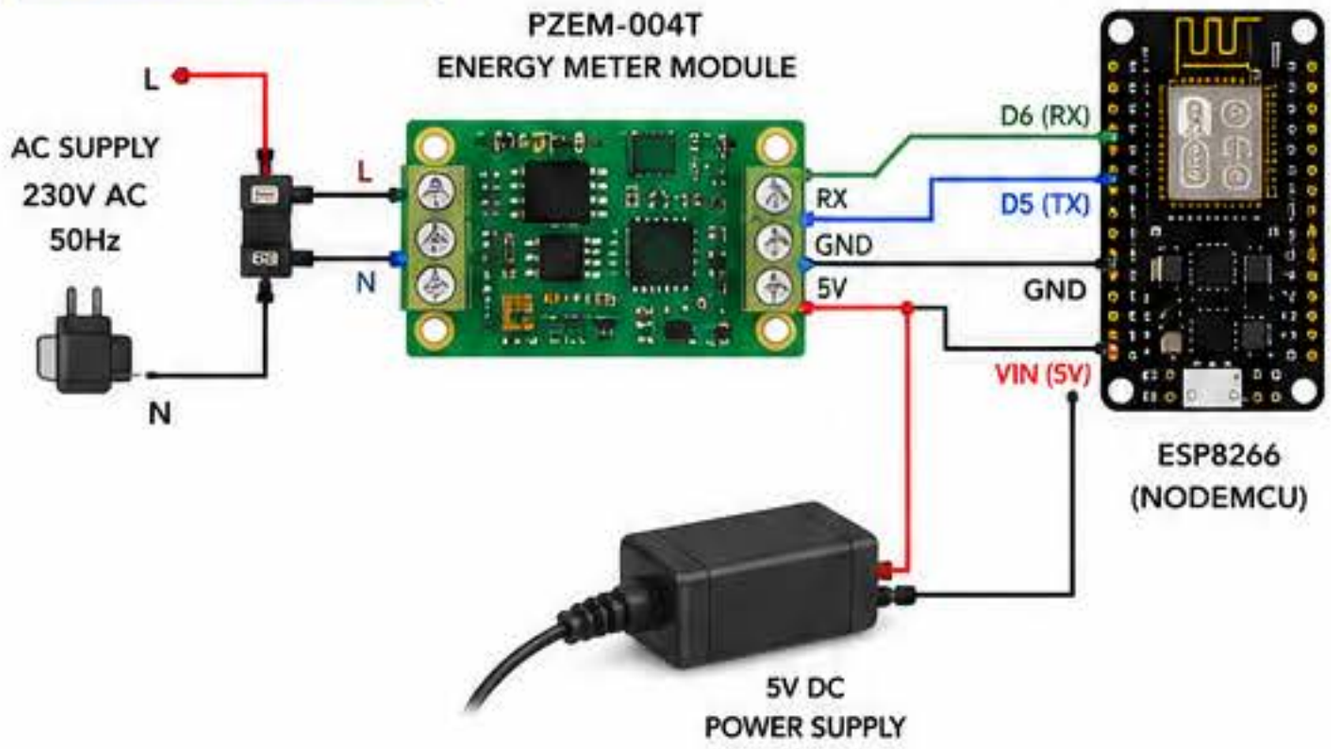
5. BLOCK DIAGRAM



WORKING:

The PZEM-004T module measures voltage, current, power, energy and other parameters from the AC supply and sends the data to ESP8266 via UART. ESP8266 connects to Blynk Cloud through Wi-Fi and uploads the data. The user can view all parameters in real-time on Blynk App and set alerts for high power or cost.

6. CIRCUIT DIAGRAM



NOTE:

- PZEM-004T works with 80–260V AC.
- Connect L and N through load.
- Ensure proper insulation and safety.
- Common GND between PZEM and NodeMCU.

7. ESP8266 (NODEMCU)

- Wi-Fi enabled microcontroller
- 3.3V Operating
- Easy programming (Arduino IDE)
- Connects to Blynk Cloud

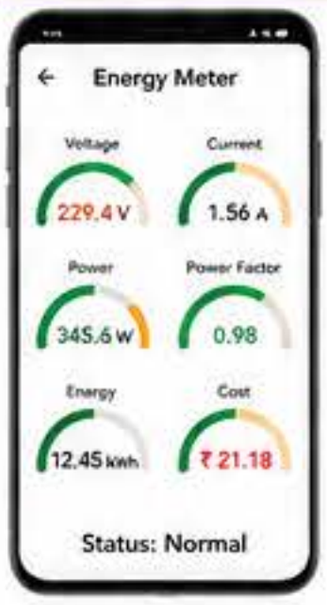
8. PZEM-004T MODULE

- Measures Voltage, Current, Power, Energy, PF, Frequency
- Built-in CT and Voltage circuit
- UART Output (TTL)
- Wide Voltage: 80–260V AC

9. 5V DC POWER SUPPLY

- Provides 5V DC to ESP8266 and PZEM
- 1A or above recommended
- Stable regulated supply

10. BLYNK APP OVERVIEW



- Real-time data monitoring
- Historical graph view
- Alerts and notifications
- Energy and cost calculation

11. PARAMETER MEASURED

- Voltage (V)
- Current (A)
- Power (W)
- Power Factor
- Frequency (Hz)
- Energy (kWh)
- Cost (₹)

12. APPLICATIONS

- Residential Energy Monitoring
- Industrial Energy Monitoring
- Smart Home Automation
- Energy Management Systems
- Hostels and PG Monitoring
- Solar and Renewable Systems

13. ADVANTAGES

- Real-time monitoring anywhere
- Low cost and easy to implement
- Helps in energy saving
- Reduces electricity bills
- Historical data and analysis
- Alerts for over usage

14. SAFETY PRECAUTIONS

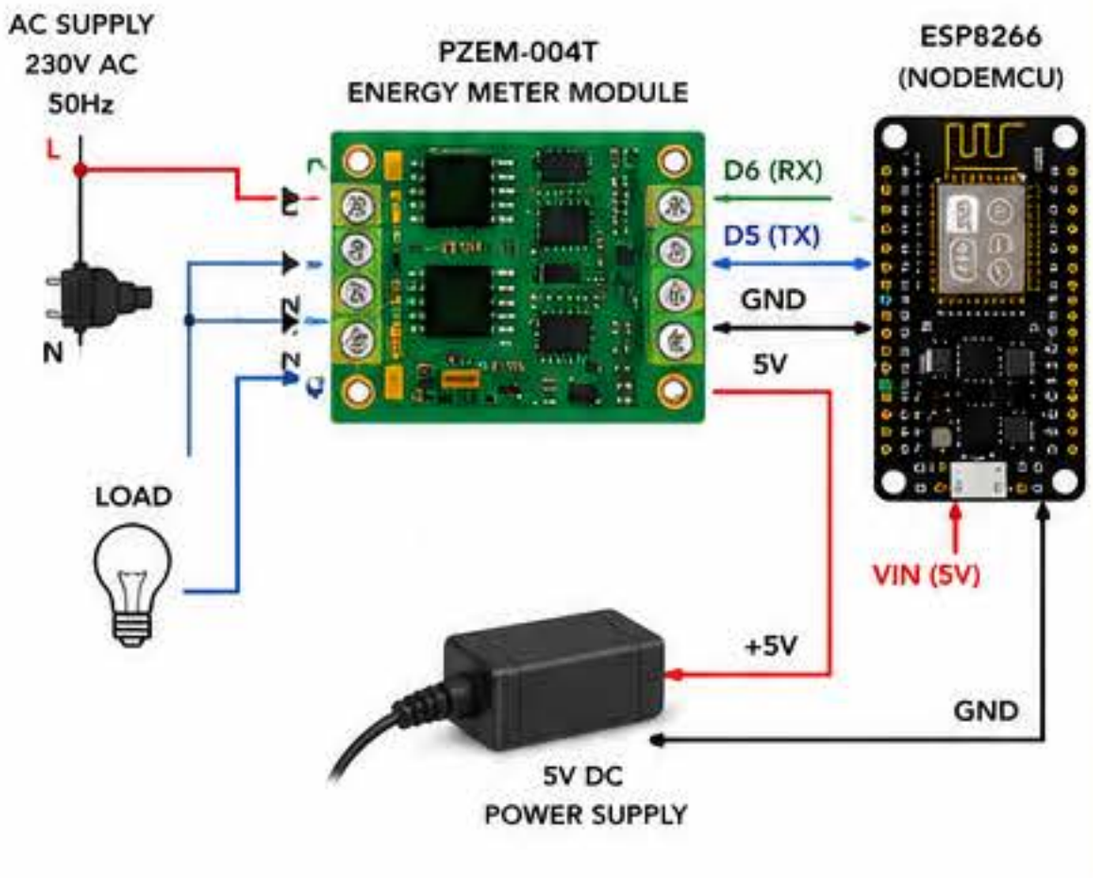
- Work with AC only if trained.
- Ensure proper insulation.
- Use quality connectors and wires.
- Do not touch live connections.
- Keep the device in dry place.
- Test the system before final use.

15. FUTURE SCOPE

- Add load control and automation.
- Integrate with home appliances.
- Add solar generation monitoring.
- Integrate with voice assistants.
- Data logging to cloud database.
- AI based consumption prediction.

Dr V Purandhar Reddy, Narsimha Reddy Engineering College, Hyderabad.

16. CIRCUIT CONNECTIONS (DETAILED)



NOTE:

- Connect L through the load.
- Ensure proper insulation and safety while working with high voltage.
- Use a regulated 5V supply (1A or above) for stable operation.

17. ARDUINO CODE (USING BLYNK LIBRARY)

```
#define BLYNK_TEMPLATE_ID       "TMPLxxxxxxxx"
#define BLYNK_TEMPLATE_NAME     "Smart Energy Meter"
#define BLYNK_AUTH_TOKEN        "Your_Blynk_Auth_Token"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <PZEM004T.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";
PZEM004T pzem(D5, D6);   // RX, TX
BlynkTimer timer;

void sendData() {
  float voltage = pzem.voltage();       // V
  float current = pzem.current();       // A
  float power   = pzem.power();         // W
  float energy  = pzem.energy();        // kWh
  float pf      = pzem.pf();            // Power Factor
  float freq    = pzem.frequency();     // Hz
  float cost    = energy * 8.0;         // Cost (Rs), assume 8 Rs/unit
  Blynk.virtualWrite(V0, voltage);
  Blynk.virtualWrite(V1, current);
  Blynk.virtualWrite(V2, power);
  Blynk.virtualWrite(V3, pf);
  Blynk.virtualWrite(V4, freq);
  Blynk.virtualWrite(V5, energy);
  Blynk.virtualWrite(V6, cost);
}
void setup(){
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(3000L, sendData);   // Send data every 3 sec
}
void loop(){
  Blynk.run();
  timer.run();
}
```

BLYNK VIRTUAL PINS	
V Pin	Parameter
V0	Voltage (V)
V1	Current (A)
V2	Power (W)
V3	Power Factor
V4	Frequency (Hz)
V5	Energy (kWh)
V6	Cost (₹)

NOTE:

- Update Wi-Fi SSID, Password and Blynk Auth Token.
- Ensure PZEM-004T connections are correct.
- Use quality power supply and wiring.

18. BLYNK APP DASHBOARD (EXAMPLE)

Widgets Used		
V0	Gauge	Voltage
V1	Gauge	Current
V2	Gauge	Power
V3	Gauge	Power Factor
V4	Gauge	Frequency
V5	Gauge	Energy
V6	Value Display	Cost

Features

- ✓ Real-time monitoring
- ✓ Historical graph (Blynk Console)
- ✓ Push notifications & alerts
- ✓ Remote monitoring from anywhere

19. TEST RESULTS

Test No.	Parameter	Expected Range	Measured Value	Status
TC-1	Voltage (V)	220 – 240 V	229.4 V	PASS
TC-2	Current (A)	0 – 10 A	1.56 A	PASS
TC-3	Power (W)	0 – 2500 W	345.6 W	PASS
TC-4	Power Factor	0.0 – 1.0	0.98	PASS
TC-5	Frequency (Hz)	45 – 55 Hz	50.0 Hz	PASS
TC-6	Energy (kWh)	0 – 9999 kWh	12.45 kWh	PASS
TC-7	Cost (₹)	As per tariff	21.18 ₹	PASS
TC-8	Data Upload	Real-time	Successful	PASS

OBSERVATIONS:

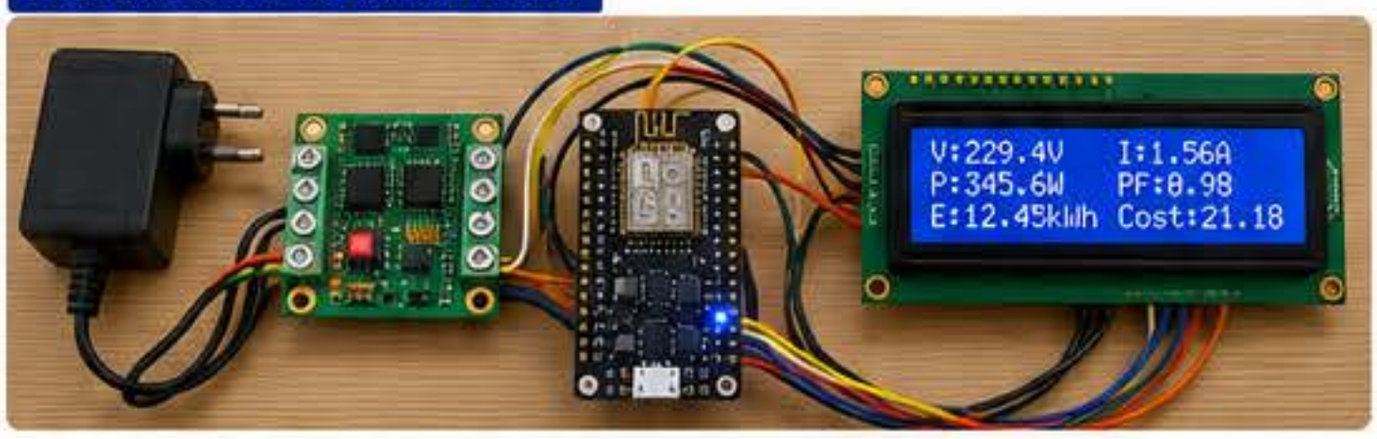
- All parameters are measured accurately and sent to Blynk Cloud.
- Data is updated in real-time with minimal delay.
- Alerts are triggered when power or energy exceeds set limits.

20. PARAMETER DISPLAYS ON LCD (OPTIONAL)

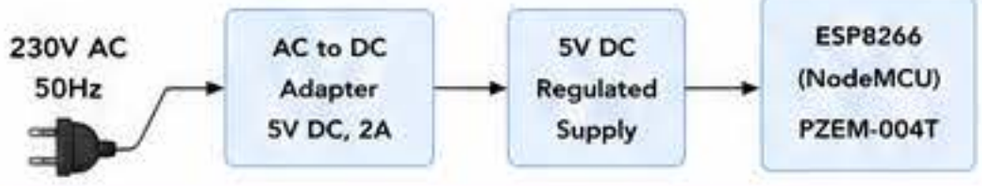
NOTE:

Connect 16x2 LCD with I2C module to ESP8266 (SDA - D1, SCL - D2, 5V, GND).

21. HARDWARE IMPLEMENTATION



22. POWER SUPPLY CIRCUIT



NOTE: Use a regulated 5V supply (1A or above) for stable and reliable operation.

23. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
No data on Blynk App	Wrong Wi-Fi / Auth Token	Check SSID, Password and Auth Token
Incorrect readings	Loose connections / Wrong wiring	Check wiring and tighten connections
Negative power/energy	CT or load wiring issue	Verify load connection through L terminal
Device not powering ON	Power supply issue	Check 5V adapter and connections
Data not updating	Poor Wi-Fi / Internet issue	Ensure stable Wi-Fi and internet

24. COMPONENT FUNCTIONS

Component	Function
ESP8266 (NodeMCU)	Wi-Fi enabled microcontroller, sends data to Blynk Cloud
PZEM-004T Module	Measures voltage, current, power, energy, PF, frequency
5V Power Supply	Provides regulated 5V DC power
Blynk Cloud	Stores data and displays on Blynk App
Blynk App	Displays live data, graphs and sends alerts

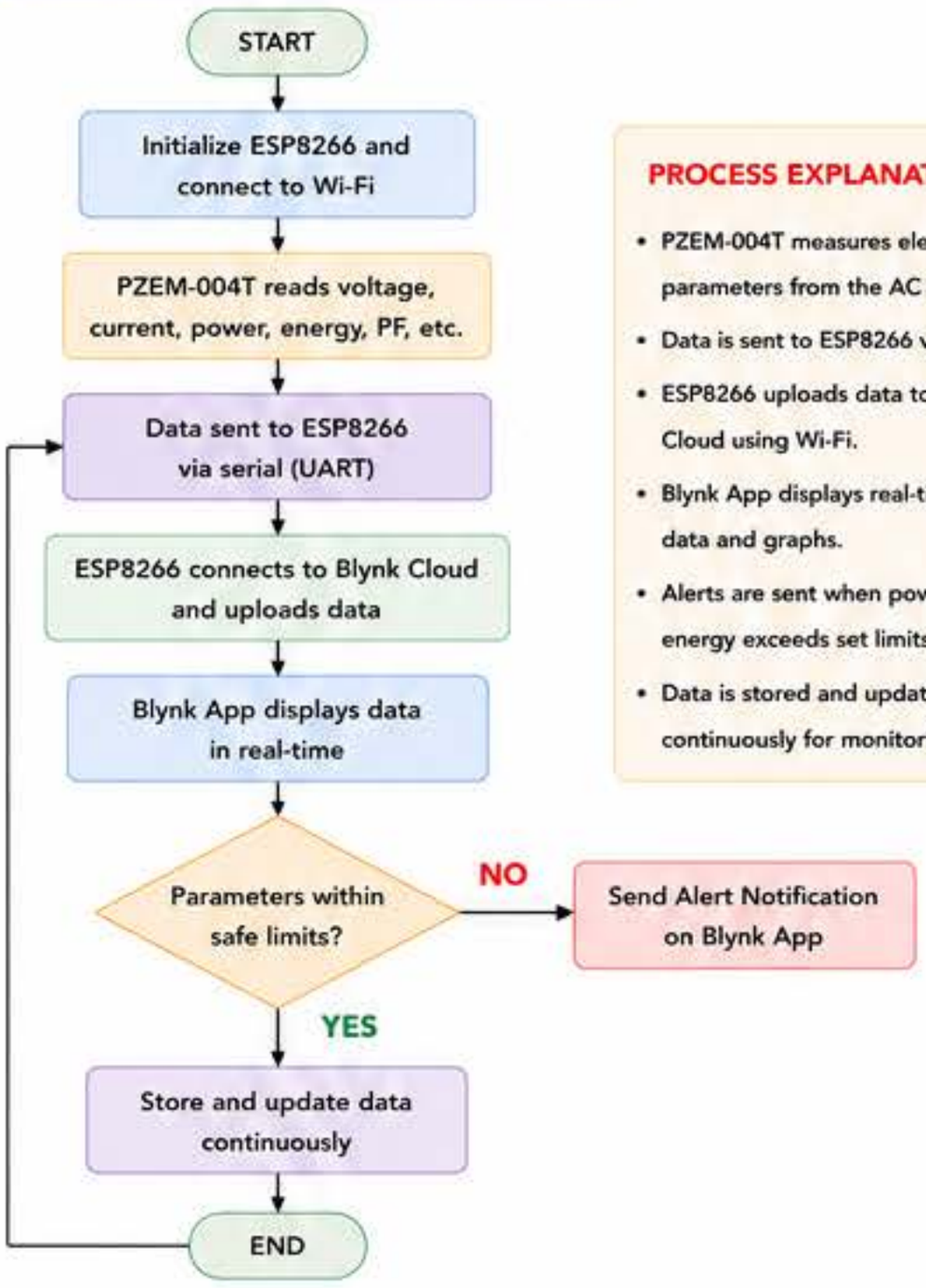
25. APPLICATIONS

- Home Energy Monitoring
- Industrial Energy Management
- Smart Buildings & Offices
- Solar & Renewable Energy Monitoring
- Energy Saving & Awareness Programs

26. FUTURE SCOPE

- ➡ Add relay for load control based on energy usage.
- ➡ Integrate solar generation and battery monitoring.
- ➡ Add voice assistant support (Alexa/Google Assistant).
- ➡ Implement AI based consumption prediction.
- ➡ Develop mobile app with advanced analytics.
- ➡ Add multi-phase energy monitoring.

27. WORKING PROCESS FLOWCHART



PROCESS EXPLANATION

- PZEM-004T measures electrical parameters from the AC load.
- Data is sent to ESP8266 via UART.
- ESP8266 uploads data to Blynk Cloud using Wi-Fi.
- Blynk App displays real-time data and graphs.
- Alerts are sent when power or energy exceeds set limits.
- Data is stored and updated continuously for monitoring.

28. REAL TIME DATA EXAMPLES (BLYNK APP)

Normal Load | High Load | Power Exceeded | Energy Exceeded

29. ALERT NOTIFICATIONS (EXAMPLES)

Blynk 10:23 AM
Smart Energy Meter
Alert: High Power Detected!
Power: 2410.5 W
Time: 10:23 AM

Blynk 02:15 PM
Smart Energy Meter
Alert: Energy Limit Exceeded!
Energy: 1000.5 kWh
Time: 02:15 PM

Blynk 06:42 PM
Smart Energy Meter
Alert: Power Factor Low!
PF: 0.62
Time: 06:42 PM

Blynk 08:10 PM
Smart Energy Meter
Alert: Voltage High!
Voltage: 242.5 V
Time: 08:10 PM

30. DATA LOGGING EXAMPLE (BLYNK CONSOLE)

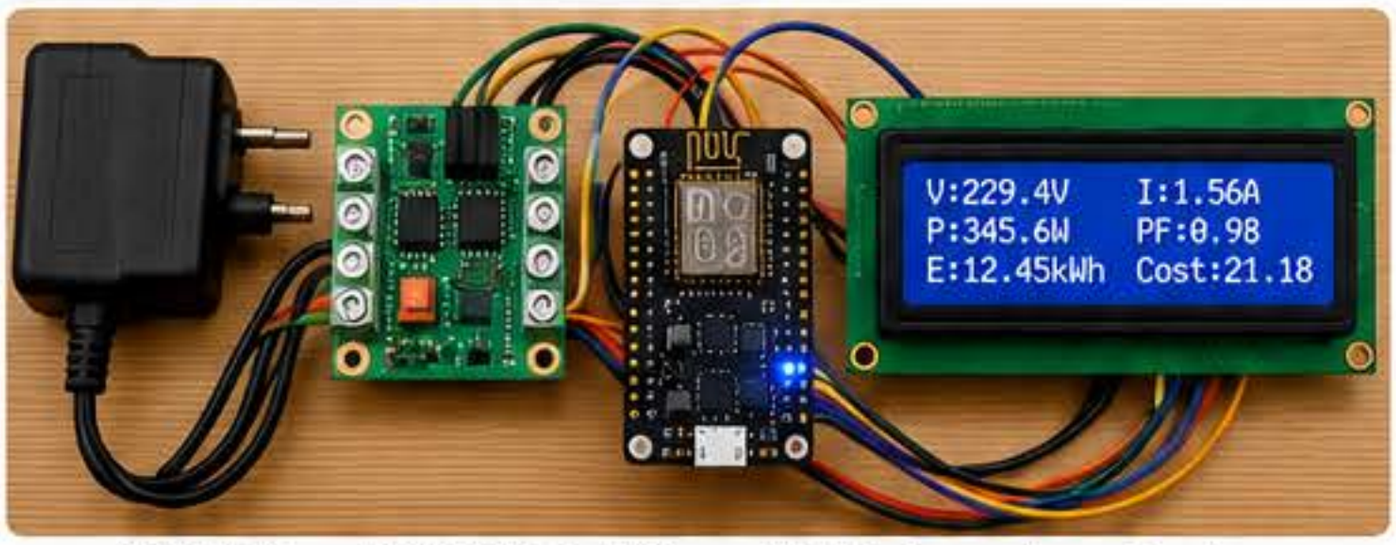
Time	Voltage (V)	Current (A)	Power (W)	PF	Energy (kWh)	Cost (₹)	Status
10:00:00	228.7	1.23	280.9	0.97	12.30	20.95	Normal
10:01:00	229.1	1.45	331.9	0.98	12.32	20.98	Normal
10:02:00	230.5	5.62	1283.6	0.88	12.40	21.11	High Load
10:03:00	241.2	10.12	2410.5	0.78	12.45	21.18	Power Exceeded
10:04:00	229.6	2.34	536.8	0.95	12.49	21.26	Normal
...	...	...	...	...	...	...	...

31. PERFORMANCE ANALYSIS

Parameter Accuracy

Parameter	Expected Value	Measured Value	Error (%)
Voltage (V)	230.0	229.4	0.26
Current (A)	2.00	1.98	1.00
Power (W)	460.0	458.5	0.33
PF	0.95	0.94	1.05
Energy (kWh)	10.00	9.96	0.40

32. HARDWARE SNAPSHOT (ACTUAL SETUP)



PZEM-004T + ESP8266 (NodeMCU) + LCD Display + Power Supply

33. COST ESTIMATION

S.No.	Component	Quantity	Unit Cost (₹)	Total Cost (₹)
1	ESP8266 (NodeMCU)	1	300	300
2	PZEM-004T Energy Meter Module	1	950	950
3	5V DC Power Supply (2A)	1	300	300
4	LCD Display (16x2 I2C)	1	150	150
5	Connecting Wires & Misc	1 Set	100	100
	Total Estimated Cost			**1,800/-**

34. SYSTEM SPECIFICATIONS

- Microcontroller : ESP8266 (NodeMCU)
- Energy Meter : PZEM-004T
- Supply Voltage : 80 – 260V AC (PZEM)
- Operating Voltage : 5V DC (ESP8266)
- Communication : UART, Wi-Fi
- Cloud Platform : Blynk Cloud
- App Platform : Blynk IoT (Android / iOS)
- Data Update Rate : 3 Seconds (Configurable)

35. ADVANTAGES

- Real-time monitoring from anywhere
- Easy installation and low cost
- Helps in energy saving and cost reduction
- Alerts for high consumption or fault
- Data logging and historical analysis
- Scalable and user-friendly solution

36. SAFETY PRECAUTIONS

- Do not touch live AC connections.
- Ensure proper insulation and wiring.
- Use regulated 5V power supply.
- Mount in dry and well-ventilated area.
- Check connections before powering ON.

37. FUTURE SCOPE

- Integration with solar and battery systems.
- AI based energy consumption prediction.
- Voice assistant integration (Alexa/Google).
- Automatic load control based on usage.
- Web dashboard for advanced analytics.

PROJECT-12

IOT FIRE DETECTION AND ALERT SYSTEM USING BLYNK APP

USING ESP8266 (NodeMCU), FLAME SENSOR AND BUZZER

1. AIM

To design and implement an IoT based Fire Detection and Alert System that detects fire using flame sensor and sends instant alerts and notifications to users through Blynk App.

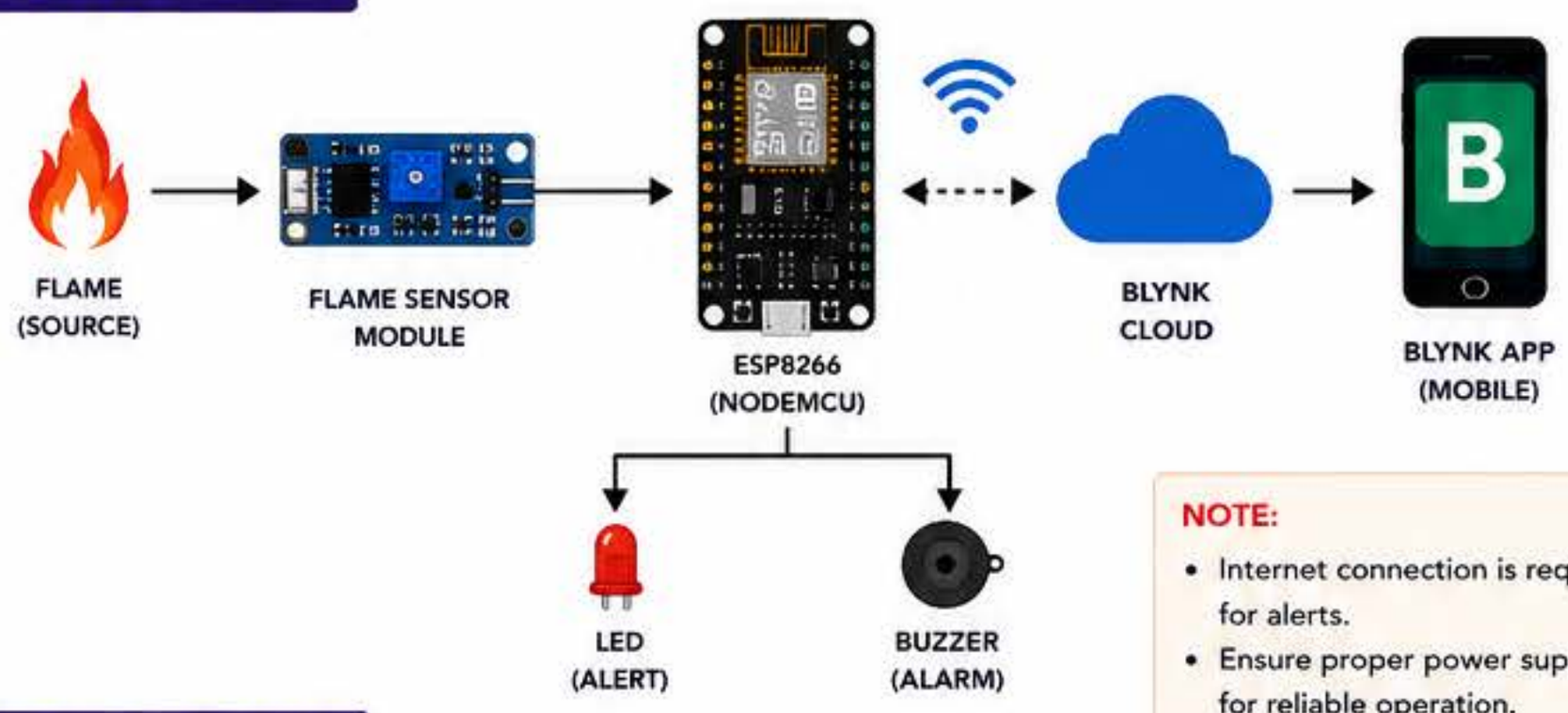
2. OBJECTIVES

- Detect fire using flame sensor.
- Trigger buzzer and LED for local alert.
- Send real-time notification on Blynk App.
- Monitor and control the system remotely.
- Enhance safety and minimize damage due to fire.

3. SYSTEM OVERVIEW

The flame sensor detects fire. ESP8266 (NodeMCU) processes the signal and triggers buzzer and LED. It also sends alert notifications to the user through Blynk Cloud. The user can monitor the status in real-time on Blynk App.

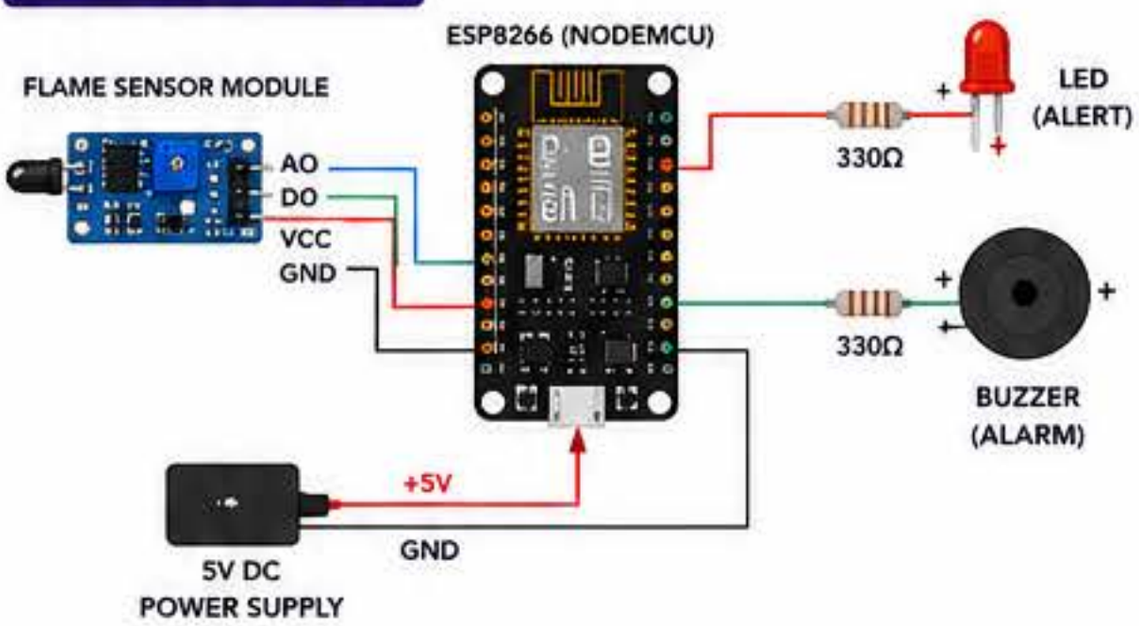
4. BLOCK DIAGRAM



NOTE:

- Internet connection is required for alerts.
- Ensure proper power supply for reliable operation.

5. CIRCUIT DIAGRAM



6. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	Flame Sensor Module	1
3	Buzzer	1
4	LED (Red)	1
5	330Ω Resistor	2
6	5V DC Power Supply	1
7	Connecting Wires	Several

7. LIST OF IOT PROJECTS

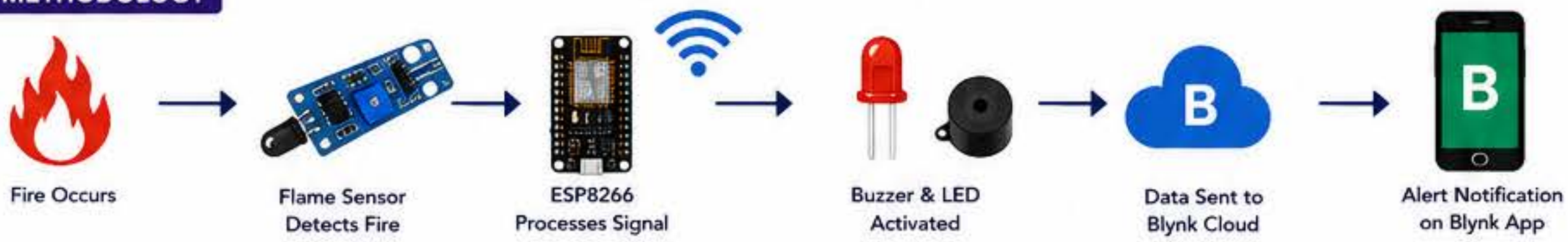
S.No.	Project Title	Main Components
1.	IoT Home Light Control Using Blynk	ESP8266/ESP32, Relay, Bulb
2.	IoT Smart Temperature Monitoring System	ESP8266/ESP32, DHT11/LM35
3.	IoT Smart Water Level Monitoring	ESP8266/ESP32, Water Level Sensor
4.	IoT Rain Alert System	ESP8266/ESP32, Rain Sensor, Buzzer
5.	IoT Gas Leakage Detection and Alert	ESP8266/ESP32, MQ-2 Sensor, Buzzer
6.	IoT Smart Door Lock Monitoring	ESP8266/ESP32, Reed Switch
7.	IoT Motion Detection and Security Alarm	ESP8266/ESP32, PIR Sensor, Buzzer
8.	IoT Smart Garden Irrigation System	ESP8266/ESP32, Soil Moisture, Relay, Pump
9.	IoT Smart Street Light System	ESP8266/ESP32, LDR, Relay, LED
10.	IoT Weather Monitoring Station	ESP8266/ESP32, DHT11, LDR, Rain Sensor
11.	IoT Smart Energy Meter Monitoring	ESP8266/ESP32, ACS712 Current Sensor
12.	IoT Fire Detection and Alert System	ESP8266/ESP32, Flame Sensor, Buzzer
13.	IoT Smart Parking Slot Monitoring	ESP8266/ESP32, IR Sensor, LEDs
14.	IoT Smart Plant Health Monitoring	ESP8266/ESP32, Soil Moisture, DHT11
15.	IoT Patient Health Monitoring System	ESP8266/ESP32, Pulse Sensor, LM35

8. FEATURES

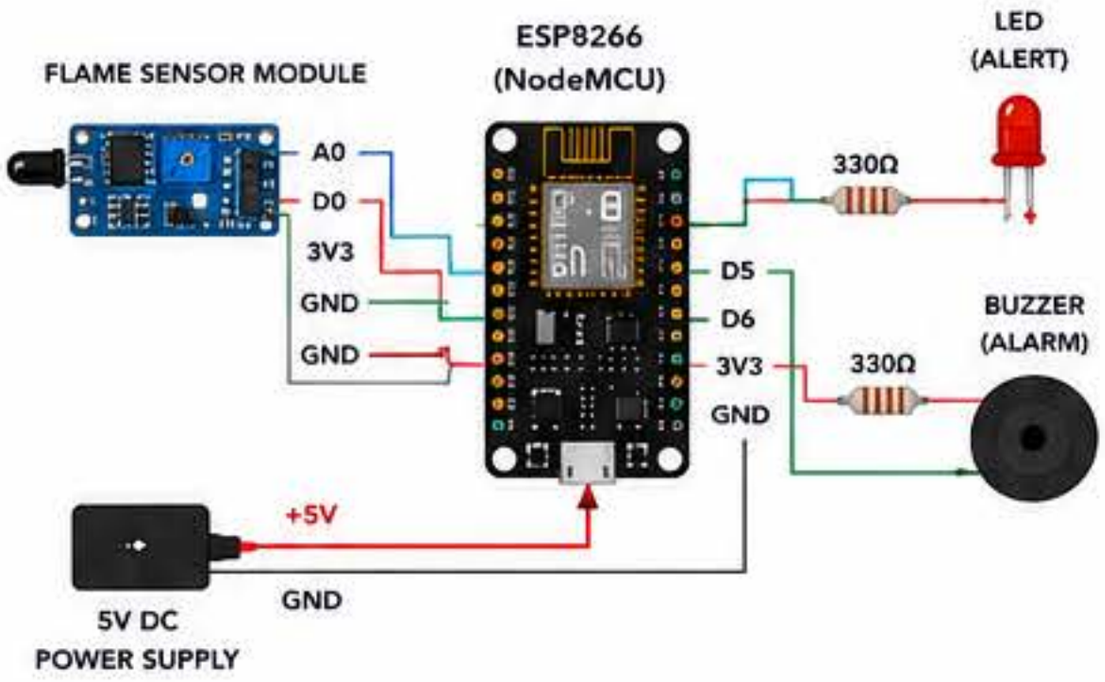
- ✓ Real-time fire detection using flame sensor.
- ✓ Instant alert with buzzer and LED.
- ✓ Push notification on Blynk App.
- ✓ Real-time status monitoring.
- ✓ Works from anywhere with internet.
- ✓ Low cost, easy to install and use.

9. BLYNK APP OVERVIEW

10. METHODOLOGY



11. CIRCUIT CONNECTIONS (DETAILED)



NOTE:

- Connect all GND pins together.
- Use 5V power supply (1A or above) for stable operation.
- Ensure the flame sensor is positioned to detect the fire properly.

12. ARDUINO CODE (USING BLYNK LIBRARY)

```
#define BLYNK_TEMPLATE_ID       "TMPLxxxxxx"
#define BLYNK_TEMPLATE_NAME     "Fire Detection System"
#define BLYNK_AUTH_TOKEN        "Your_Blynk_Auth_Token"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

#define FLAME_PIN D1
#define BUZZER_PIN D6
#define LED_PIN D5

BlynkTimer timer;

void sendData() {
  int flameValue = digitalRead(FLAME_PIN);   // 1 = Fire, 0 = No Fire
  Blynk.virtualWrite(V0, flameValue);        // Send status to Blynk
}

void setup() {
  Serial.begin(115200);
  pinMode(FLAME_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(1000L, sendData);   // Send data every 1 sec
}

void loop() {
  Blynk.run();
  timer.run();
  int flameValue = digitalRead(FLAME_PIN);
  if (flameValue == HIGH) {       // Fire Detected
    digitalWrite(LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH);
    Blynk.notify("⚠ FIRE DETECTED! Check Immediately.");
  } else {                                  // No Fire
    digitalWrite(LED_PIN, LOW);
    digitalWrite(BUZZER_PIN, LOW);
  }
}
```

BLYNK VIRTUAL PINS

V Pin	Function
V0	Fire Status (0/1)
V1	Alert Notification
V2	System Status

NOTE:

- Update Wi-Fi SSID, Password and Blynk Auth Token.
- Blynk will show real-time status and send notifications.

13. FLAME SENSOR MODULE

Pin	Description
A0	Analog Output (Fire Intensity)
D0	Digital Output (Fire Detected)
VCC	3.3V – 5V
GND	Ground

Working:

The flame sensor detects infrared wavelength of fire (760nm – 1100nm).
When fire is detected, D0 pin becomes HIGH.

14. BLYNK DASHBOARD (EXAMPLE)

Widget	Function
LED	Indicator – Shows LED Status (ON/OFF)
Buzzer	Indicator – Shows Buzzer Status (ON/OFF)
Fire Status	Value Display – 0 = No Fire, 1 = Fire Detected
Notification	Push Notification – Alert Message

Features on App:

- Real-time fire status monitoring.
- Instant alert notification.
- Remote monitoring from anywhere.

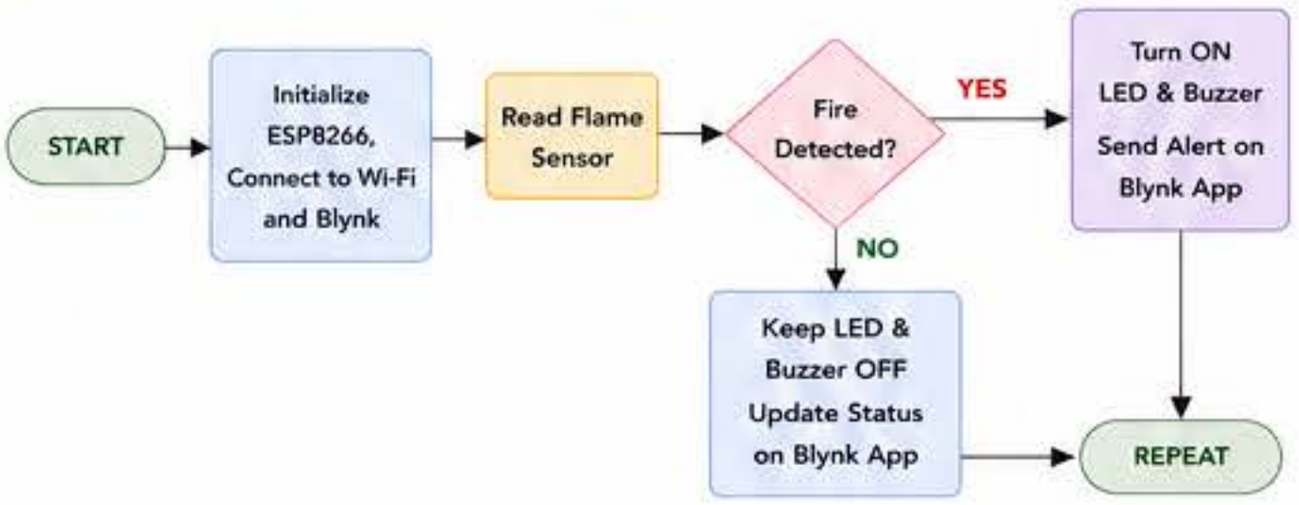
15. TEST RESULTS

Test No.	Condition	Flame Status (D0)	LED	Buzzer	Blynk App	Result
1	No Fire	LOW (0)	OFF	OFF	No Alert	PASS
2	Fire Detected	HIGH (1)	ON	ON	Alert Sent	PASS
3	Re-Test	HIGH (1)	ON	ON	Alert Sent	PASS
4	Fire Removed	LOW (0)	OFF	OFF	No Alert	PASS

OBSERVATIONS:

- Flame sensor detects fire accurately.
- LED and Buzzer are activated on fire detection.
- Blynk app receives real-time status and sends alert notification.

16. SYSTEM OPERATION FLOWCHART



17. APPLICATIONS

- Home and Building Fire Safety
- Industrial Fire Monitoring
- Server Room Fire Alert
- Hospitals and Schools
- Forest Fire Early Warning
- Shopping Malls and Offices

18. ADVANTAGES

- Real-time fire detection and alert.
- Instant notification on mobile.
- Low cost and energy efficient.
- Easy to install and use.
- Remote monitoring from anywhere.
- Enhances safety and reduces fire damage.

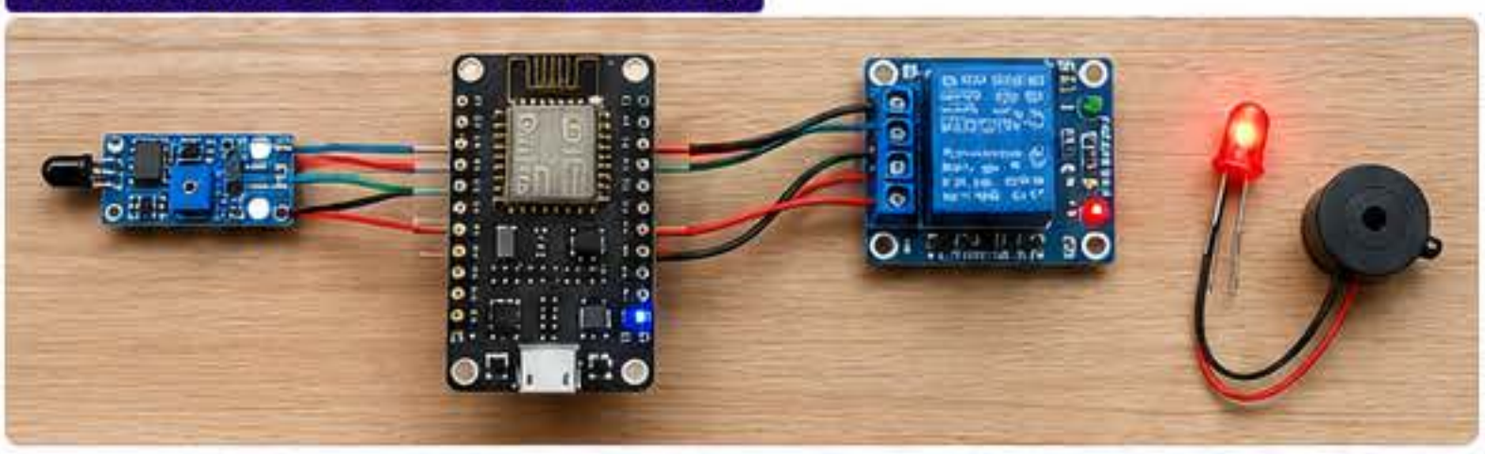
19. SAFETY PRECAUTIONS

- Use proper power supply.
- Do not place flame sensor near high temperature sources.
- Ensure good ventilation.
- Check wiring connections before powering ON.
- Keep the system away from water and moisture.

20. FUTURE SCOPE

- Integrate with smoke sensor.
- Add water sprinkler system.
- Send SMS / Email alerts.
- Use camera for live streaming.
- AI based fire prediction model.
- Battery backup for uninterrupted operation.

21. HARDWARE SETUP PHOTO (REFERENCE)



22. COST ESTIMATION

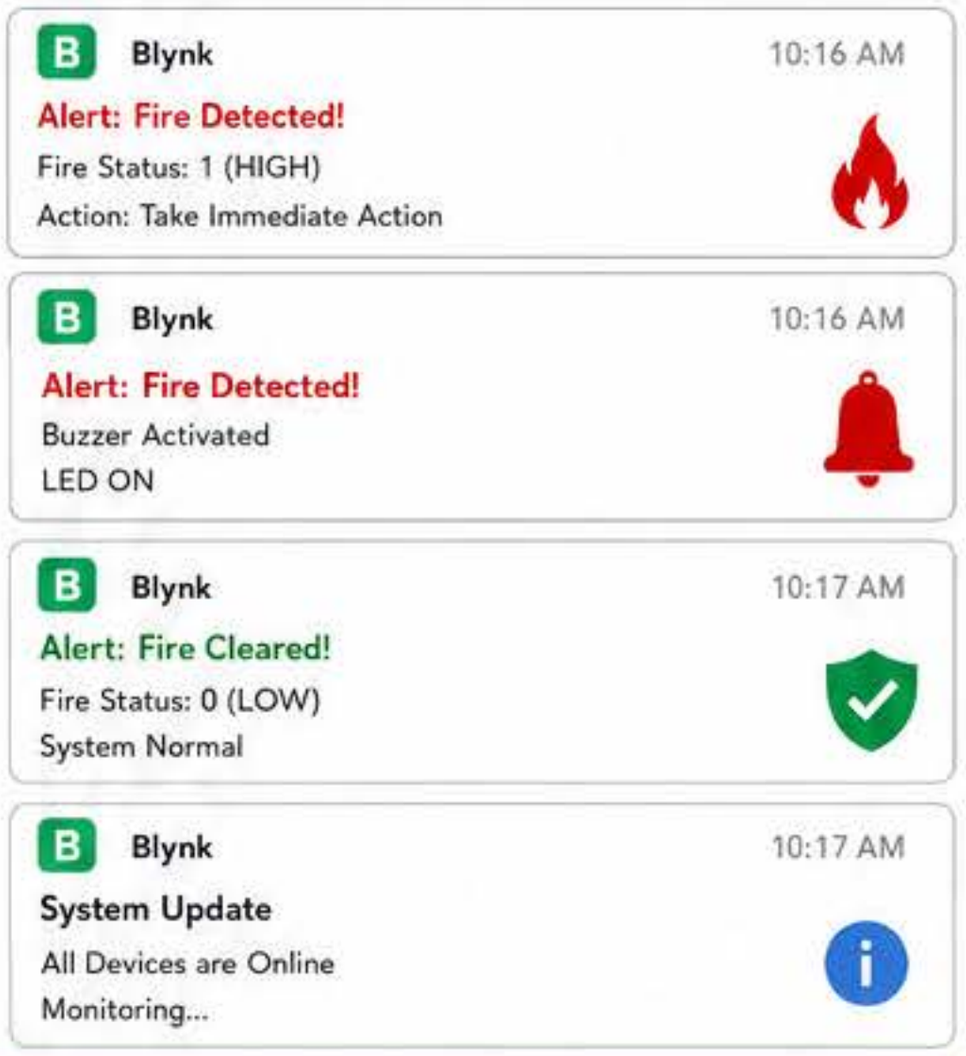
S.No.	Component	Quantity	Unit Cost (₹)	Total Cost (₹)
1	ESP8266 (NodeMCU)	1	300	300
2	Flame Sensor Module	1	120	120
3	Buzzer	1	30	30
4	LED (Red)	1	10	10
5	330Ω Resistor	2	2	4
6	5V DC Power Supply	1	300	300
7	Connecting Wires	1 Set	50	50
	Total Estimated Cost			814/-

23. REAL TIME ALERT EXAMPLES (BLYNK APP)

FEATURES ON APP

- Real-time fire status display.
- Instant LED and Buzzer indication.
- Push notification on fire detection.
- Monitor system from anywhere.
- Easy and user friendly interface.

24. BLYNK NOTIFICATION EXAMPLES



NOTE:
Blynk will send instant notifications to mobile when fire is detected and when system returns to normal.

25. DATA LOGGING EXAMPLE (BLYNK CONSOLE)

Time	Fire Status (DO)	LED	Buzzer	Notification	Remarks
10:14:50	LOW (0)	OFF	OFF	No Alert	No Fire
10:16:05	HIGH (1)	ON	ON	Alert Sent	Fire Detected
10:16:06	HIGH (1)	ON	ON	Alert Sent	Alert Repeated
10:17:45	LOW (0)	OFF	OFF	Clear Sent	Fire Cleared
10:18:10	LOW (0)	OFF	OFF	No Alert	System Normal

OBSERVATIONS:

- Fire sensor output becomes HIGH when fire is detected.
- LED and Buzzer turn ON instantly for local alert.
- Blynk app receives alert notifications in real time.
- System returns to normal when fire is removed.

26. PERFORMANCE SUMMARY

Parameter	Result
Fire Detection Time	1 – 2 Seconds
Alert Notification Time	1 – 2 Seconds
LED Response	Instant
Buzzer Response	Instant
Internet Connectivity	Wi-Fi (Stable)
Range	Up to Wi-Fi Range
System Reliability	High
Power Consumption	Low
Cost	Low

SYSTEM STATUS INDICATORS

NO FIRE
System Normal

FIRE DETECTED
Alert Active

CLEAR
System Normal

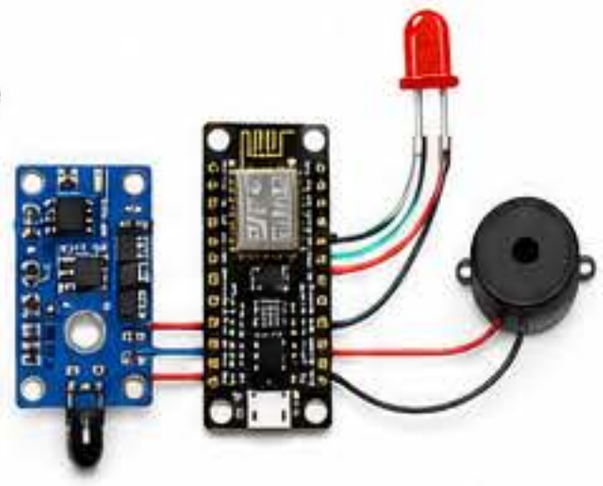
NOTE:
Results may vary depending on Wi-Fi strength and environment.

27. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
Flame not detected	Flame sensor not positioned correctly	Adjust sensor direction towards flame source
LED/Buzzer not working	Wrong wiring or loose connection	Check connections and component polarity
No notification on Blynk	Wi-Fi / Token issue	Check Wi-Fi, Auth Token and internet connection
LED always ON	Sensor output wire shorted	Check wiring and sensor output
Buzzer not sounding	Faulty buzzer or module	Check buzzer wiring and test separately
App shows offline	Internet connection lost	Ensure stable Wi-Fi connection

28. SYSTEM SPECIFICATIONS

- Microcontroller : ESP8266 (NodeMCU)
- Flame Sensor : Infrared Flame Sensor Module
- Alarm : Buzzer (Active), LED (Red)
- Communication : Wi-Fi (2.4 GHz)
- Platform : Blynk IoT Cloud
- Operating Voltage : 5V DC
- Power Supply : 5V DC (USB Adapter/Power Bank)
- App Support : Android & iOS
- Range : Up to Wi-Fi Coverage



29. CONCLUSION

- ✓ The IoT based Fire Detection and Alert System successfully detects fire using flame sensor and sends instant alerts.
- ✓ The system triggers local alarm using LED and buzzer for immediate warning.
- ✓ Real-time notifications are received on mobile through Blynk App.
- ✓ It can be used for home, office, industries and forest safety.
- ✓ The project is cost effective, easy to implement and reliable.
- ✓ This system helps in early fire detection and minimizes damage.

30. REFERENCES

- Blynk Documentation – docs.blynk.io
- ESP8266 (NodeMCU) Datasheet – espressif.com
- Flame Sensor Module Datasheet
- Arduino IDE Documentation – arduino.cc
- IoT with ESP8266 – Internet of Things, Academic Books
- Various IoT Projects and Research Papers

NOTE:

- Ensure proper ventilation and safety during testing.
- Do not expose flame sensor to excessive heat for long time.
- Use 5V regulated power supply for stable operation.
- This project is for educational and demonstration purpose only.

PROJECT-13

IOT SMART PARKING SLOT MONITORING SYSTEM

Smart Parking for Smart Cities

1. ABSTRACT

The IoT Smart Parking Slot Monitoring System helps drivers find available parking slots in real-time and reduces traffic congestion in crowded areas.
It uses IR/Ultrasonic sensors to detect vehicle presence in each slot and updates the data on a cloud platform. Users can view live slot availability on a mobile app (Blynk) and navigate to free slots quickly, saving time, fuel and improving urban parking management.

2. OBJECTIVES

- To monitor parking slot availability in real-time.
- To display the slot status on a mobile application.
- To reduce traffic and time spent in searching for parking.
- To optimize parking space utilization.
- To provide a low-cost and scalable solution for smart cities.

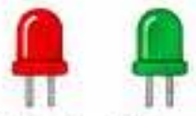
3. COMPONENTS USED

ESP32

IR / Ultrasonic Sensors

I2C LCD Display (16x2)

Wi-Fi Module (Built-in ESP32)



LEDs (Red/Green)

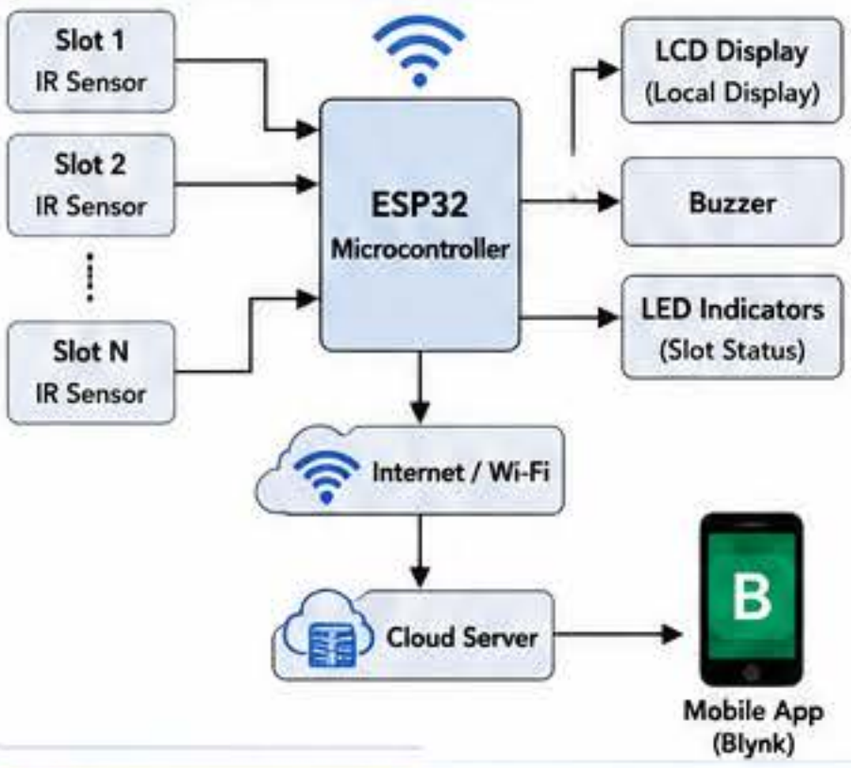
Buzzer

Power Supply (5V DC)

Jumper Wires

PCB / Breadboard

4. SYSTEM BLOCK DIAGRAM

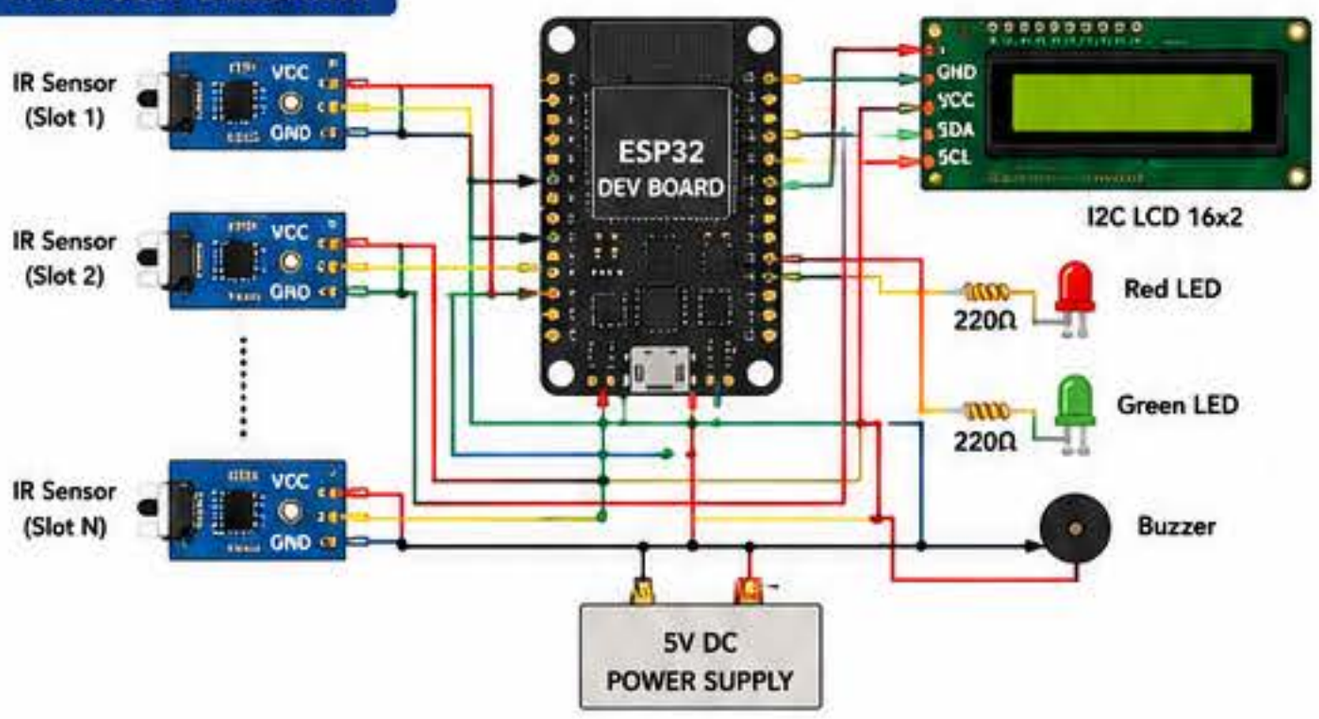


5. WORKING PRINCIPLE

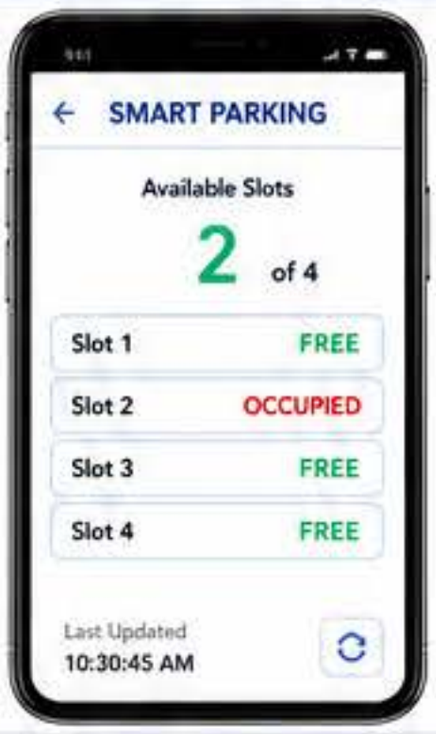
1. IR/Ultrasonic sensors are installed at each parking slot to detect the presence or absence of vehicles.
2. When a vehicle is detected, the sensor sends a signal to the ESP32.
3. ESP32 processes the data and updates the slot status.
4. The status is displayed on the LCD screen and LEDs (Green = Free, Red = Occupied).
5. Data is sent to the cloud via Wi-Fi.
6. Users can view real-time slot availability on the Blynk mobile app.
7. If all slots are full, buzzer alert can be activated at the entrance.

6. PARKING LOT OVERVIEW

7. CIRCUIT DIAGRAM



8. MOBILE APP DASHBOARD (BLYNK)



- Real-time Slot Status
- Live Count of Available Slots
- History (Optional)
- Alerts & Notifications
- User Friendly Interface

9. APPLICATIONS

- Shopping Malls
- Office Buildings
- Airports
- Hospitals
- Universities / Colleges
- Smart Cities
- Residential Complexes
- Public Parking Areas

10. ADVANTAGES

- ✔ Real-time monitoring of parking slots
- ✔ Reduces traffic congestion
- ✔ Saves time and fuel
- ✔ Easy to install and maintain
- ✔ Improves parking space utilization
- ✔ Low cost and scalable
- ✔ Environment friendly

11. PARAMETERS MONITORED

- Slot Occupancy Status (Occupied / Free)
- Total Available Slots
- Total Occupied Slots
- Last Updated Time
- Alerts (All Slots Full)

12. SOFTWARE & TECHNOLOGIES

• Hardware	:	ESP32, IR Sensors
• Software	:	Arduino IDE
• IoT Platform	:	Blynk IoT Cloud
• Mobile App	:	Blynk (Android / iOS)
• Language	:	C / C++ (Arduino)
• Communication	:	Wi-Fi (TCP/IP)
• Database	:	Blynk Cloud

13. FUTURE SCOPE

- ▶ Integration with digital payment gateways for automated parking.
- ▶ License Plate Recongition (ANPR) for vehicle identification.
- ▶ Dynamic pricing based on demand.
- ▶ Integration with navigation apps (Google Maps, Waze).
- ▶ Solar powered smart parking system.

14. SYSTEM FEATURES

IoT Based Monitoring · Real-time Alerts · Cloud Data Storage · Mobile App Access · Energy Efficient

15. CONCLUSION

The IoT Smart Parking Slot Monitoring System provides an efficient, reliable and low-cost solution to modern parking challenges. It helps in better management of parking resources, reduces urban congestion and enhances user experience.

"Smart Parking, Smart Living!"

Dr V Purandhar Reddy, Narsimha Reddy Engineering College, Hyderabad.

PROJECT-13

IOT SMART GARDEN IRRIGATION SYSTEM USING BLYNK APP

USING ESP8266 (NodeMCU), SOIL MOISTURE SENSOR AND PUMP

1. AIM

To design and implement an IoT based Smart Garden Irrigation System that monitors soil moisture and automatically controls the water pump to maintain optimal soil moisture level using Blynk App.

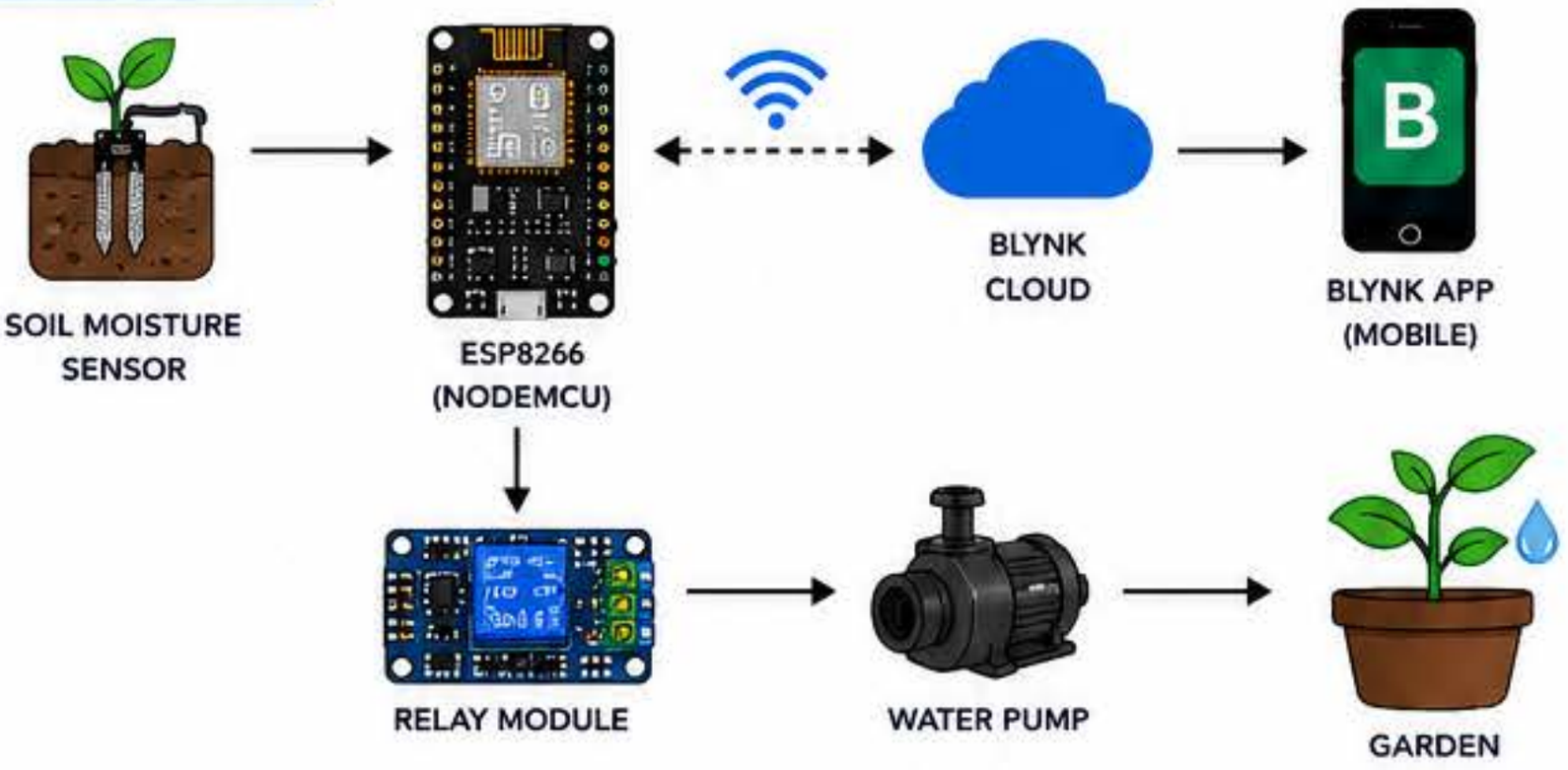
2. OBJECTIVES

- Monitor soil moisture in real-time.
- Automatically control water pump based on moisture level.
- Provide remote monitoring and manual control using Blynk App.
- Save water and improve plant health.
- Make the system low cost, efficient and easy to use.

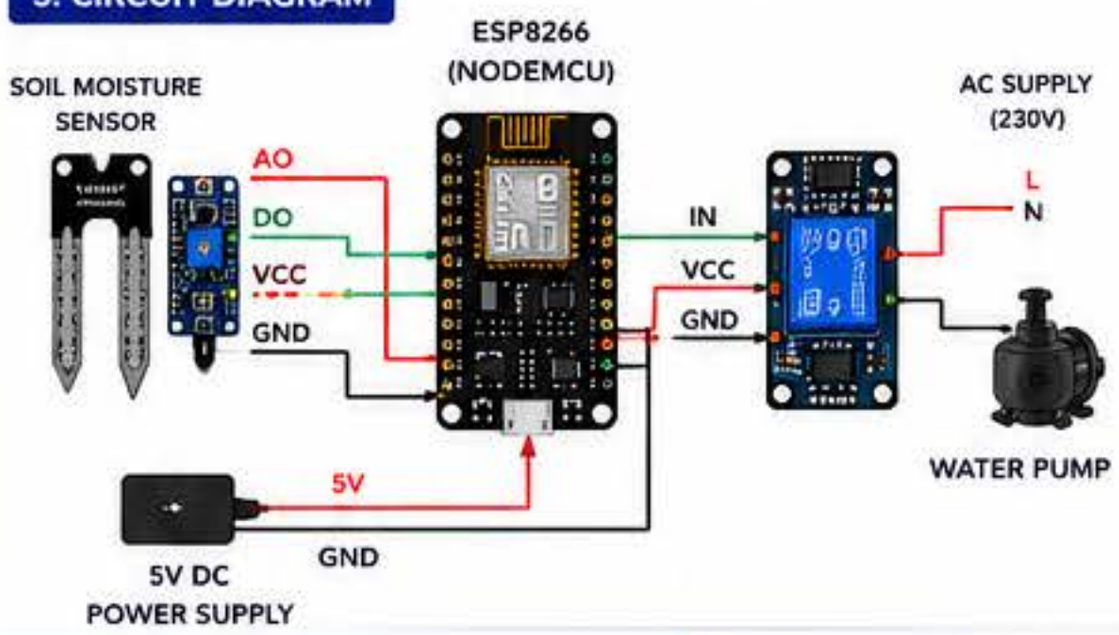
3. SYSTEM OVERVIEW

The system uses a Soil Moisture Sensor to measure the moisture level of soil. ESP8266 (NodeMCU) reads the sensor value and sends data to Blynk Cloud. Based on the moisture level, the water pump is automatically turned ON or OFF. The user can also monitor and control the pump manually through Blynk App from anywhere.

4. BLOCK DIAGRAM



5. CIRCUIT DIAGRAM



6. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	Soil Moisture Sensor	1
3	1 Channel Relay Module	1
4	Water Pump (5V / 12V)	1
5	5V Power Supply	1
6	Connecting Wires	Several

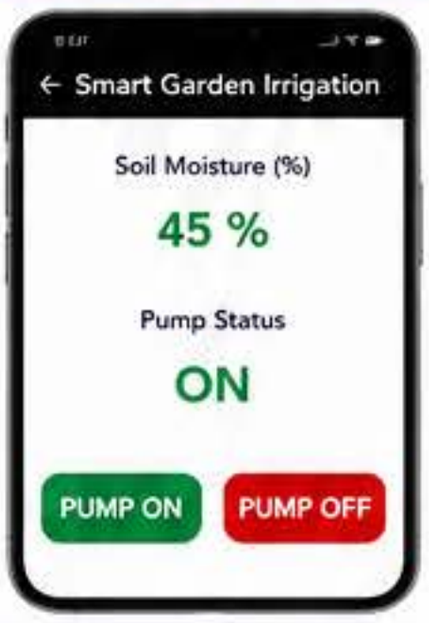
7. LIST OF IOT PROJECTS

S.No.	Project Title	Main Components
1.	IoT Home Light Control Using Blynk	ESP8266/ESP32, Relay, Bulb
2.	IoT Smart Temperature Monitoring System	ESP8266/ESP32, DHT11/LM35
3.	IoT Smart Water Level Monitoring	ESP8266/ESP32, Water Level Sensor
4.	IoT Rain Alert System	ESP8266/ESP32, Rain Sensor, Buzzer
5.	IoT Gas Leakage Detection and Alert	ESP8266/ESP32, MQ-2 Sensor, Buzzer
6.	IoT Smart Door Lock Monitoring	ESP8266/ESP32, Reed Switch
7.	IoT Motion Detection and Security Alarm	ESP8266/ESP32, PIR Sensor, Buzzer
8.	**IoT Smart Garden Irrigation System**	**ESP8266/ESP32, Soil Moisture, Relay, Pump**
9.	IoT Smart Street Light System	ESP8266/ESP32, LDR, Relay, LED
10.	IoT Weather Monitoring Station	ESP8266/ESP32, DHT11, LDR, Rain Sensor
11.	IoT Smart Energy Meter Monitoring	ESP8266/ESP32, ACS712 Current Sensor
12.	IoT Fire Detection and Alert System	ESP8266/ESP32, Flame Sensor, Buzzer
13.	IoT Smart Parking Slot Monitoring	ESP8266/ESP32, IR Sensor, LEDs
14.	IoT Smart Plant Health Monitoring	ESP8266/ESP32, Soil Moisture, DHT11
15.	IoT Patient Health Monitoring System	ESP8266/ESP32, Pulse Sensor, LM35

8. FEATURES

- Real-time soil moisture monitoring.
- Automatic pump ON/OFF based on moisture level.
- Remote monitoring and manual control via Blynk App.
- Saves water and improves plant health.
- Works from anywhere with internet.
- Low power consumption and easy to install.

9. BLYNK APP OVERVIEW (EXAMPLE)



DISPLAYED DATA

- Soil Moisture (%)
- Pump Status (ON/OFF)
- Manual Control Buttons (PUMP ON / PUMP OFF)

NOTE:

- Moisture < Set Level → Pump ON
- Moisture ≥ Set Level → Pump OFF

10. APPLICATIONS

- Home Gardens
- Terrace Gardens
- Greenhouses
- Agriculture
- Nurseries

11. ADVANTAGES

- Automatic irrigation control.
- Reduces water wastage.
- Remote access and monitoring.
- Improves plant growth.
- Cost effective and efficient solution.

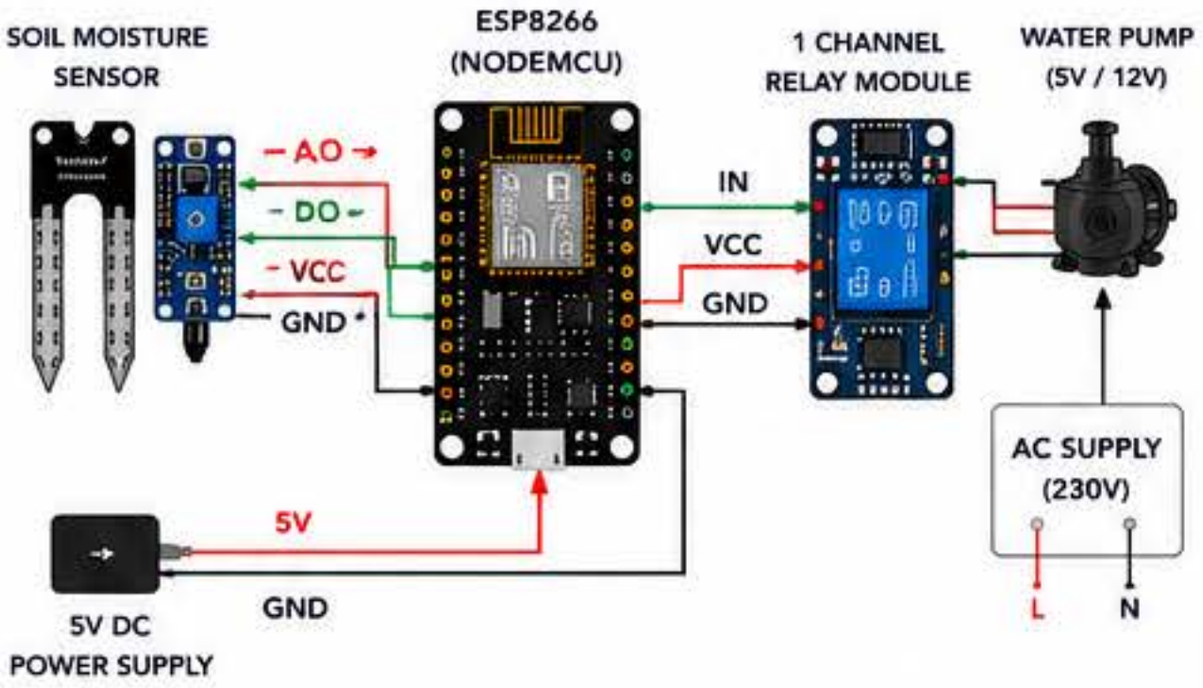
12. SAFETY PRECAUTIONS

- Use proper power supply.
- Ensure correct connections before switching ON.
- Keep the system away from water splashes.
- Use fuse or protection for pump and circuits.

13. FUTURE SCOPE

- Add rain sensor for smart control.
- Add water flow sensor.
- Integrate with solar power.
- Data logging and analytics.
- Multi zone irrigation system.

14. CIRCUIT CONNECTIONS (DETAILED)



NOTE:

- Connect all GND pins together.
- Use 5V power supply (1A or above) for stable operation.
- Relays and pump should be connected as per safety guidelines.

15. ARDUINO CODE (USING BLYNK LIBRARY)

```
#define BLYNK_TEMPLATE_ID      "TMPLxxxxxxx"
#define BLYNK_TEMPLATE_NAME    "Smart Garden Irrigation"
#define BLYNK_AUTH_TOKEN       "Your_Blynk_Auth_Token"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

#define MOISTURE_PIN A0       // Analog pin for soil moisture
#define PUMP_PIN     D1       // D1 to Relay IN
#define MOISTURE_THRESHOLD 45  // Threshold percentage

BlynkTimer timer;

void sendData() {
  int moistureValue = analogRead(MOISTURE_PIN);
  int moisturePercent = map(moistureValue, 1023, 0, 0, 100);
  Blynk.virtualWrite(V0, moisturePercent);

  if (moisturePercent < MOISTURE_THRESHOLD) {
    digitalWrite(PUMP_PIN, LOW);   // Pump ON
    Blynk.virtualWrite(V1, 1);
  } else {
    digitalWrite(PUMP_PIN, HIGH); // Pump OFF
    Blynk.virtualWrite(V1, 0);
  }
}

void setup() {
  Serial.begin(115200);
  pinMode(PUMP_PIN, OUTPUT);
  digitalWrite(PUMP_PIN, HIGH);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(2000L, sendData); // Send data every 2 sec
}

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

BLYNK VIRTUAL PINS

V Pin	Function
V0	Soil Moisture %
V1	Pump Status (0 = OFF, 1 = ON)

NOTE:

- Update Wi-Fi SSID, Password and Blynk Auth Token.
- Moisture threshold can be modified in code.
- Blynk will show real-time data and allow manual control.

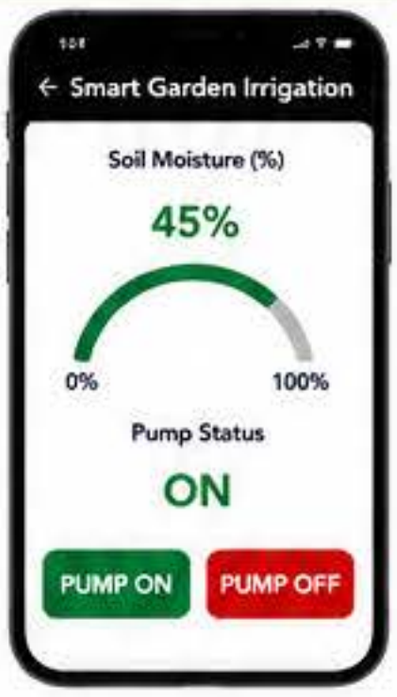
16. SOIL MOISTURE SENSOR MODULE

Pin	Description
AO	Analog Output (Moisture Level)
DO	Digital Output (Threshold)
VCC	3.3V – 5V
GND	Ground

Working:

The sensor detects moisture in soil. When soil is dry, output is LOW. When soil is wet, output is HIGH. Analog output gives moisture level.

17. BLYNK DASHBOARD (EXAMPLE)



Widget	Function
Gauge	Displays Soil Moisture (%)
LED	Shows Pump Status (ON/OFF)
Button	Manual Pump ON
Button	Manual Pump OFF

FEATURES ON APP

- ✔ Real-time soil moisture monitoring.
- ✔ Manual control of pump (ON/OFF).
- ✔ Automatic control based on threshold.
- ✔ Instant status update and notifications.

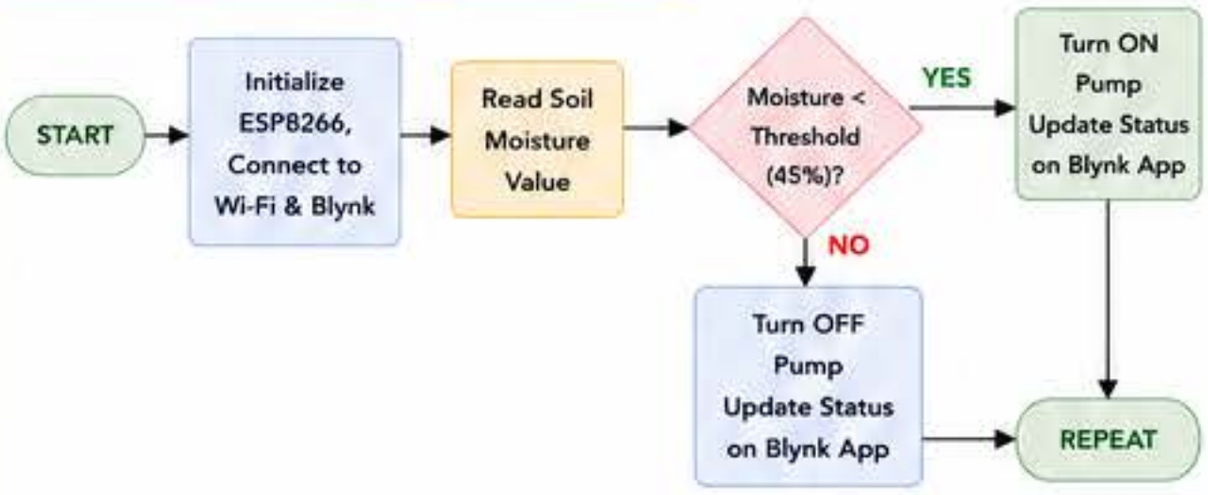
18. TEST RESULTS

Test No.	Soil Condition	Moisture (%)	Pump Status (Auto)	LED	Blynk App	Result
1	Very Dry	18%	ON	ON	Pump ON	PASS
2	Dry	32%	ON	ON	Pump ON	PASS
3	Moist	48%	OFF	OFF	Pump OFF	PASS
4	Wet	65%	OFF	OFF	Pump OFF	PASS
5	Very Wet	82%	OFF	OFF	Pump OFF	PASS

OBSERVATIONS:

- System automatically turned ON pump when moisture < threshold (45%).
- Pump turned OFF when moisture ≥ threshold.
- Blynk app displayed correct real-time values.

19. SYSTEM OPERATION FLOWCHART



20. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
No data on Blynk App	Wi-Fi not connected / Wrong Token	Check Wi-Fi and Blynk Auth Token
Moisture reading not correct	Sensor not inserted properly / Dry soil	Insert sensor properly and water the soil
Pump not switching ON	Relay wiring issue / Power issue	Check relay connections and power supply
Pump always ON	Threshold value too high	Adjust threshold in code and test
App shows offline	Internet connection lost	Ensure stable Wi-Fi connection

NOTE: Calibrate the moisture sensor for accurate results. Keep the system in protected area from rain and sun.

21. SYSTEM SPECIFICATIONS

- Microcontroller : ESP8266 (NodeMCU)
- Soil Moisture Sensor : Capacitive / Resistive Type
- Relay Module : 1 Channel (5V)
- Water Pump : 5V / 12V DC
- Communication : Wi-Fi (2.4 GHz)
- Platform : Blynk IoT Cloud
- Operating Voltage : 5V DC
- Power Supply : 5V DC (1A Adapter / Power Bank)
- App Support : Android & iOS
- Range : Up to Wi-Fi Coverage

22. CONCLUSION

- ✔ The IoT Smart Garden Irrigation System monitors soil moisture in real-time and automates the water pump based on the moisture level.
- ✔ It provides both automatic and manual control through the Blynk App.
- ✔ The system helps in saving water, time and improves plant health.
- ✔ It is cost effective, easy to implement and user friendly.
- ✔ This project can be extended for multi-zone irrigation and weather based control.

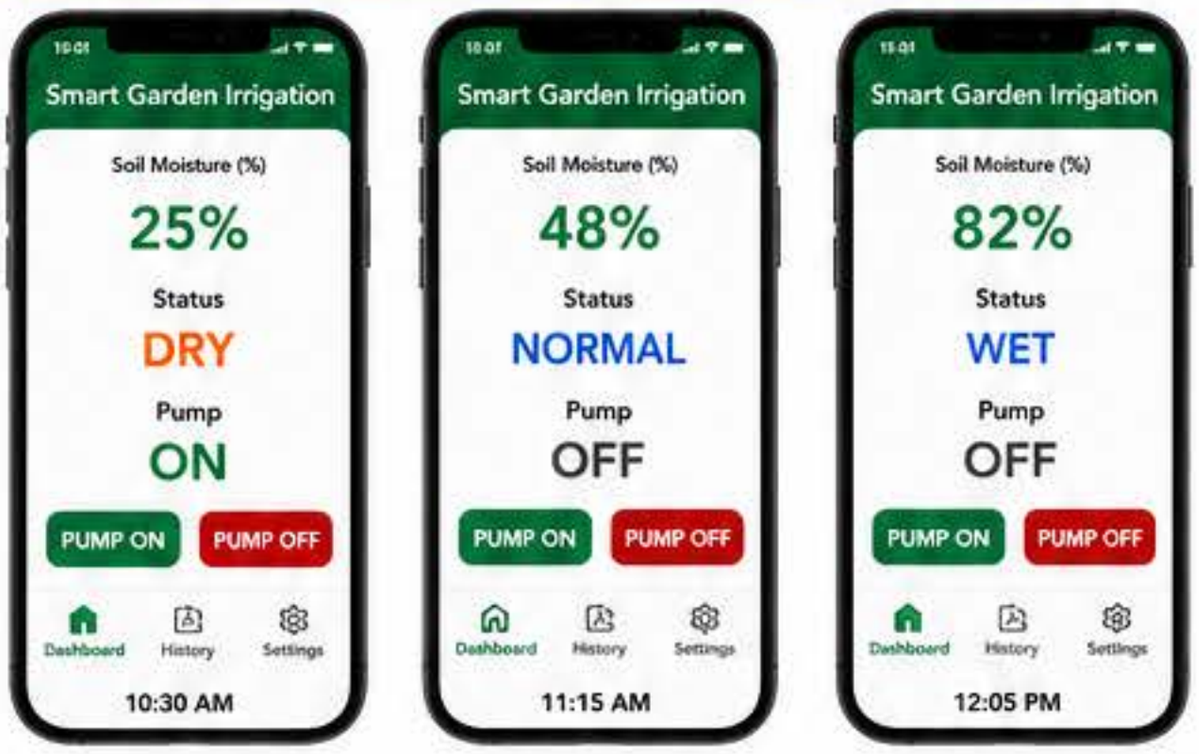
APPLICATIONS

NOTE:

- Use proper power supply.
- Ensure correct wiring before powering ON.
- Keep the system away from water and moisture.
- This project is for educational and demonstration purpose only.

Dr V Purandhar Reddy, Narsimha Reddy Engineering College, Hyderabad.

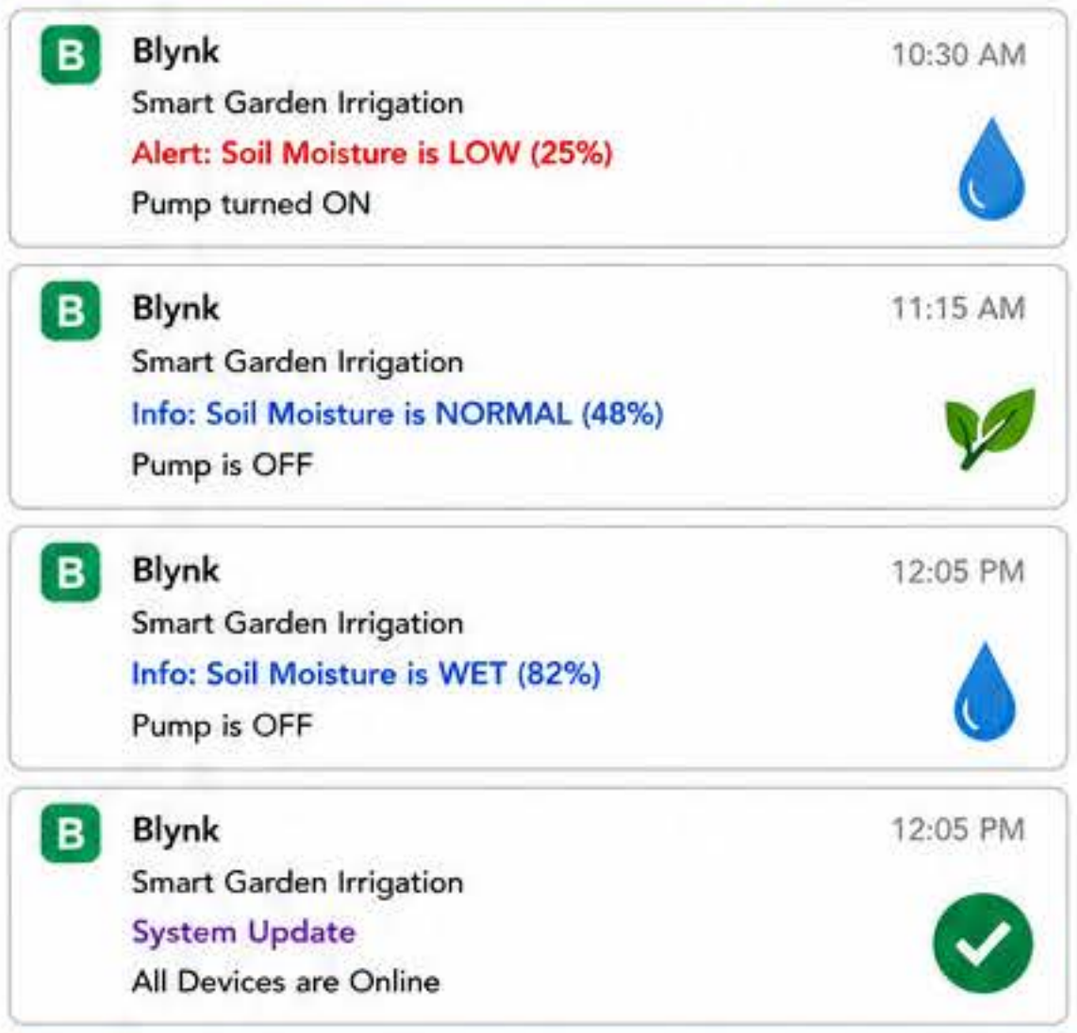
23. REAL TIME MONITORING EXAMPLES (BLYNK APP)



FEATURES ON APP

- Real-time soil moisture display.
- Automatic pump control.
- Manual pump control option.
- Instant status update.
- Data history and logs.
- Easy and user friendly interface.

24. BLYNK NOTIFICATION EXAMPLES



NOTE:
Blynk will send instant notifications to mobile when soil moisture changes or when pump status changes.

25. DATA LOGGING EXAMPLE (BLYNK CONSOLE)

Time	Moisture (%)	Threshold (%)	Pump Status	Blynk App	Remarks
10:15:20	15	45	ON	Pump ON	Dry Soil
10:30:10	25	45	ON	Pump ON	Pump Started
10:45:05	35	45	ON	Pump ON	Irrigating
11:00:12	48	45	OFF	Pump OFF	Reached Threshold
11:15:18	60	45	OFF	Pump OFF	Moist Soil
11:30:22	72	45	OFF	Pump OFF	Wet Soil
12:05:30	82	45	OFF	Pump OFF	Very Wet

OBSERVATIONS:

- Pump turns ON automatically when moisture < threshold (45%).
- Pump turns OFF when moisture >= threshold.
- Data is logged in Blynk console in real-time.
- Helps in analyzing soil behavior and water usage.

26. PERFORMANCE SUMMARY

Parameter	Result
Moisture Detection Range	0% – 100%
Threshold Accuracy	± 2%
Pump Response Time	1 – 2 Seconds
Data Update Interval	2 Seconds
Wi-Fi Connectivity	Stable
Blynk Data Sync	Real-time
Manual Control	Working
System Reliability	High
Power Consumption	Low
Cost	Low

SYSTEM STATUS INDICATORS

DRY (Low Moisture)
Moisture < Threshold
Pump ON

NORMAL (Optimal)
Moisture ≈ Threshold
Pump OFF

WET (High Moisture)
Moisture > Threshold
Pump OFF

NOTE:
System performance depends on Wi-Fi strength and sensor conditions.

27. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
No data on Blynk App	Wi-Fi not connected / Wrong Token	Check Wi-Fi and Auth Token
Moisture value not changing	Sensor not inserted properly / Dry soil	Insert sensor properly and water soil
Pump not turning ON	Relay wiring issue / Power issue	Check relay connections and power
Pump always ON	Threshold value too high / Code issue	Adjust threshold and check code
App shows offline	Internet connection lost	Ensure stable Wi-Fi connection
Incorrect readings	Sensor dirty / Corroded pins	Clean sensor and check connections

NOTE: Regularly clean the sensor and check the wiring connections to get accurate results.

28. SYSTEM SPECIFICATIONS

- Microcontroller : ESP8266 (NodeMCU)
- Soil Moisture Sensor : Capacitive / Resistive Type
- Relay Module : 1 Channel (5V)
- Water Pump : 5V / 12V DC
- Power Supply : 5V DC (USB Adapter / Power Bank)
- Communication : Wi-Fi (2.4 GHz)
- Platform : Blynk IoT Cloud
- Operating Voltage : 5V DC
- App Support : Android & iOS
- Range : Up to Wi-Fi Coverage

29. CONCLUSIONS

- ✓ The IoT Smart Garden Irrigation System efficiently monitors soil moisture in real-time.
- ✓ It automatically controls the water pump based on the moisture level.
- ✓ Real-time data is displayed on Blynk App with instant notifications.
- ✓ This system helps in saving water, time and improves plant health.
- ✓ It is cost effective, easy to implement and scalable for multiple zones.
- ✓ This project can be extended with weather data, solar power and AI based irrigation.

APPLICATIONS

30. FUTURE SCOPE

- ➡ Integrate with weather forecasting APIs.
- ➡ AI based irrigation scheduling.
- ➡ Solar powered irrigation system.
- ➡ Add pH and temperature sensors.
- ➡ Multi zone automatic irrigation.
- ➡ Data analytics and report generation.
- ➡ Voice assistant integration.
- ➡ Mobile app with advanced controls.

31. HARDWARE SETUP PHOTO (REFERENCE)

32. COST ESTIMATION

S.No.	Component	Quantity	Unit Cost (₹)	Total Cost (₹)
1	ESP8266 (NodeMCU)	1	300	300
2	Soil Moisture Sensor	1	120	120
3	1 Channel Relay Module	1	90	90
4	Water Pump (5V/12V)	1	250	250
5	5V DC Power Supply	1	300	300
6	Connecting Wires	1 Set	50	50
	Total Estimated Cost			1110/-

PROJECT-13

IOT SMART GARDEN IRRIGATION SYSTEM USING BLYNK APP

USING ESP8266 (NodeMCU), SOIL MOISTURE SENSOR AND PUMP

1. AIM

To design and implement an IoT based Smart Garden Irrigation System that monitors soil moisture and automatically controls the water pump to maintain optimal soil moisture level using Blynk App.

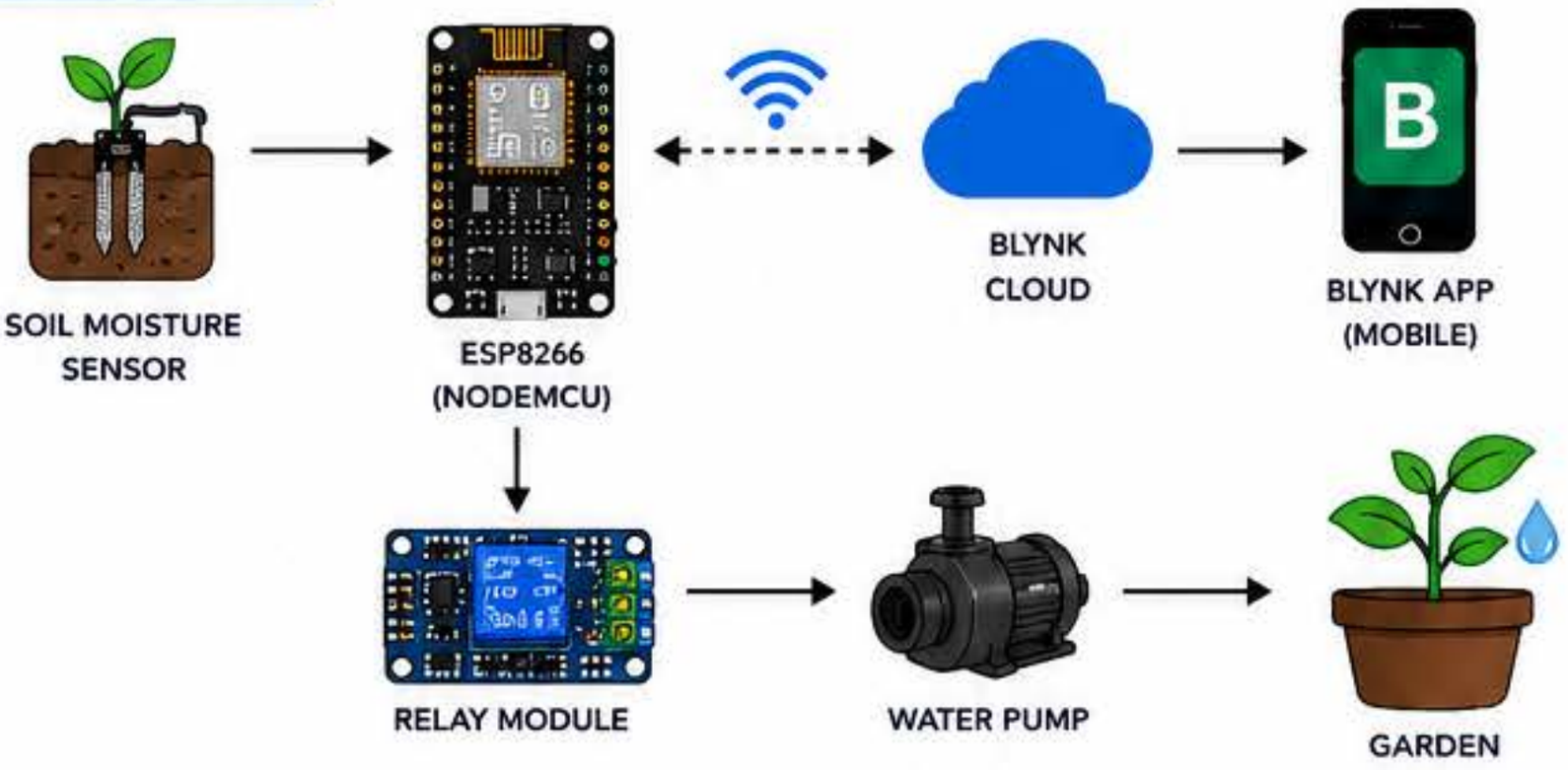
2. OBJECTIVES

- Monitor soil moisture in real-time.
- Automatically control water pump based on moisture level.
- Provide remote monitoring and manual control using Blynk App.
- Save water and improve plant health.
- Make the system low cost, efficient and easy to use.

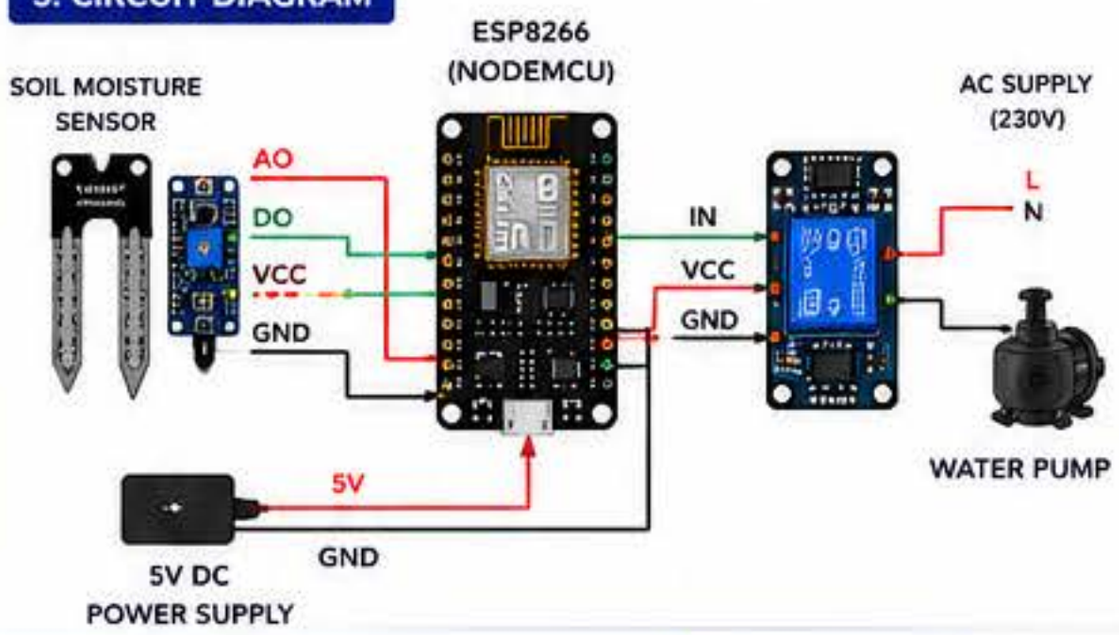
3. SYSTEM OVERVIEW

The system uses a Soil Moisture Sensor to measure the moisture level of soil. ESP8266 (NodeMCU) reads the sensor value and sends data to Blynk Cloud. Based on the moisture level, the water pump is automatically turned ON or OFF. The user can also monitor and control the pump manually through Blynk App from anywhere.

4. BLOCK DIAGRAM



5. CIRCUIT DIAGRAM



6. COMPONENTS REQUIRED

S.No.	Component	Quantity
1	ESP8266 (NodeMCU)	1
2	Soil Moisture Sensor	1
3	1 Channel Relay Module	1
4	Water Pump (5V / 12V)	1
5	5V Power Supply	1
6	Connecting Wires	Several

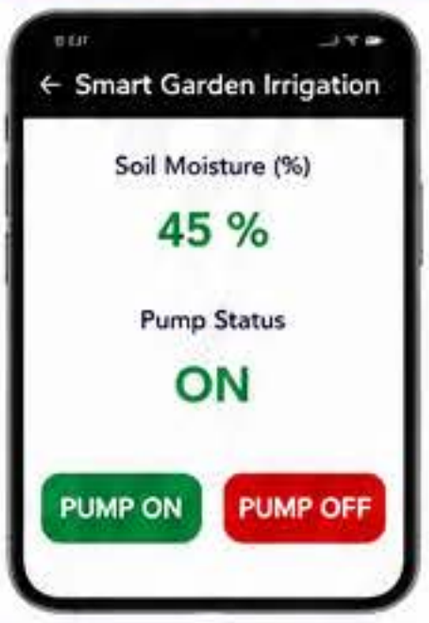
7. LIST OF IOT PROJECTS

S.No.	Project Title	Main Components
1.	IoT Home Light Control Using Blynk	ESP8266/ESP32, Relay, Bulb
2.	IoT Smart Temperature Monitoring System	ESP8266/ESP32, DHT11/LM35
3.	IoT Smart Water Level Monitoring	ESP8266/ESP32, Water Level Sensor
4.	IoT Rain Alert System	ESP8266/ESP32, Rain Sensor, Buzzer
5.	IoT Gas Leakage Detection and Alert	ESP8266/ESP32, MQ-2 Sensor, Buzzer
6.	IoT Smart Door Lock Monitoring	ESP8266/ESP32, Reed Switch
7.	IoT Motion Detection and Security Alarm	ESP8266/ESP32, PIR Sensor, Buzzer
8.	**IoT Smart Garden Irrigation System**	**ESP8266/ESP32, Soil Moisture, Relay, Pump**
9.	IoT Smart Street Light System	ESP8266/ESP32, LDR, Relay, LED
10.	IoT Weather Monitoring Station	ESP8266/ESP32, DHT11, LDR, Rain Sensor
11.	IoT Smart Energy Meter Monitoring	ESP8266/ESP32, ACS712 Current Sensor
12.	IoT Fire Detection and Alert System	ESP8266/ESP32, Flame Sensor, Buzzer
13.	IoT Smart Parking Slot Monitoring	ESP8266/ESP32, IR Sensor, LEDs
14.	IoT Smart Plant Health Monitoring	ESP8266/ESP32, Soil Moisture, DHT11
15.	IoT Patient Health Monitoring System	ESP8266/ESP32, Pulse Sensor, LM35

8. FEATURES

- Real-time soil moisture monitoring.
- Automatic pump ON/OFF based on moisture level.
- Remote monitoring and manual control via Blynk App.
- Saves water and improves plant health.
- Works from anywhere with internet.
- Low power consumption and easy to install.

9. BLYNK APP OVERVIEW (EXAMPLE)



DISPLAYED DATA

- Soil Moisture (%)
- Pump Status (ON/OFF)
- Manual Control Buttons (PUMP ON / PUMP OFF)

NOTE:

- Moisture < Set Level → Pump ON
- Moisture ≥ Set Level → Pump OFF

10. APPLICATIONS

- Home Gardens
- Terrace Gardens
- Greenhouses
- Agriculture
- Nurseries

11. ADVANTAGES

- Automatic irrigation control.
- Reduces water wastage.
- Remote access and monitoring.
- Improves plant growth.
- Cost effective and efficient solution.

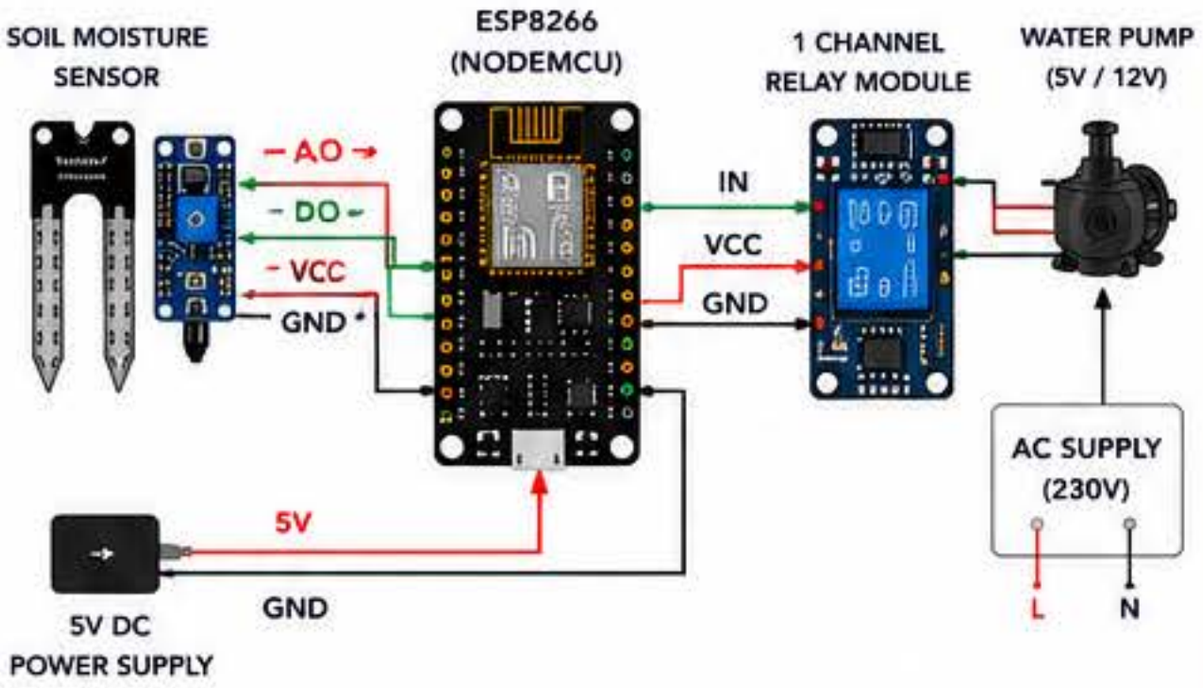
12. SAFETY PRECAUTIONS

- Use proper power supply.
- Ensure correct connections before switching ON.
- Keep the system away from water splashes.
- Use fuse or protection for pump and circuits.

13. FUTURE SCOPE

- Add rain sensor for smart control.
- Add water flow sensor.
- Integrate with solar power.
- Data logging and analytics.
- Multi zone irrigation system.

14. CIRCUIT CONNECTIONS (DETAILED)



NOTE:

- Connect all GND pins together.
- Use 5V power supply (1A or above) for stable operation.
- Relays and pump should be connected as per safety guidelines.

15. ARDUINO CODE (USING BLYNK LIBRARY)

```
#define BLYNK_TEMPLATE_ID      "TMPLxxxxxxx"
#define BLYNK_TEMPLATE_NAME    "Smart Garden Irrigation"
#define BLYNK_AUTH_TOKEN       "Your_Blynk_Auth_Token"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "YOUR_WIFI_SSID";
char pass[] = "YOUR_WIFI_PASSWORD";

#define MOISTURE_PIN A0       // Analog pin for soil moisture
#define PUMP_PIN     D1       // D1 to Relay IN
#define MOISTURE_THRESHOLD 45  // Threshold percentage

BlynkTimer timer;

void sendData() {
  int moistureValue = analogRead(MOISTURE_PIN);
  int moisturePercent = map(moistureValue, 1023, 0, 0, 100);
  Blynk.virtualWrite(V0, moisturePercent);

  if (moisturePercent < MOISTURE_THRESHOLD) {
    digitalWrite(PUMP_PIN, LOW);   // Pump ON
    Blynk.virtualWrite(V1, 1);
  } else {
    digitalWrite(PUMP_PIN, HIGH); // Pump OFF
    Blynk.virtualWrite(V1, 0);
  }
}

void setup() {
  Serial.begin(115200);
  pinMode(PUMP_PIN, OUTPUT);
  digitalWrite(PUMP_PIN, HIGH);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(2000L, sendData); // Send data every 2 sec
}

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

BLYNK VIRTUAL PINS

V Pin	Function
V0	Soil Moisture %
V1	Pump Status (0 = OFF, 1 = ON)

NOTE:

- Update Wi-Fi SSID, Password and Blynk Auth Token.
- Moisture threshold can be modified in code.
- Blynk will show real-time data and allow manual control.

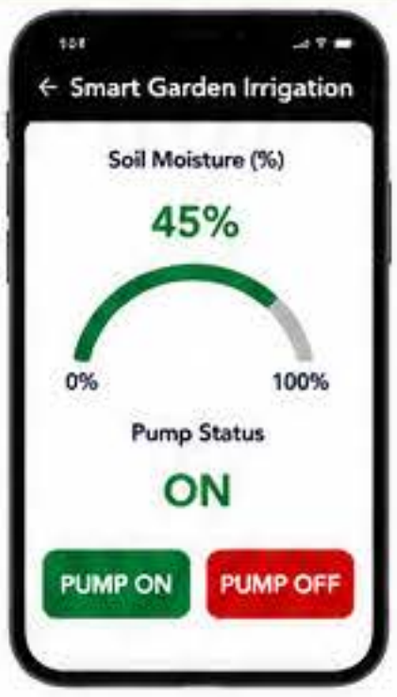
16. SOIL MOISTURE SENSOR MODULE

Pin	Description
AO	Analog Output (Moisture Level)
DO	Digital Output (Threshold)
VCC	3.3V – 5V
GND	Ground

Working:

The sensor detects moisture in soil. When soil is dry, output is LOW. When soil is wet, output is HIGH. Analog output gives moisture level.

17. BLYNK DASHBOARD (EXAMPLE)



Widget	Function
Gauge	Displays Soil Moisture (%)
LED	Shows Pump Status (ON/OFF)
Button	Manual Pump ON
Button	Manual Pump OFF

FEATURES ON APP

- ✔ Real-time soil moisture monitoring.
- ✔ Manual control of pump (ON/OFF).
- ✔ Automatic control based on threshold.
- ✔ Instant status update and notifications.

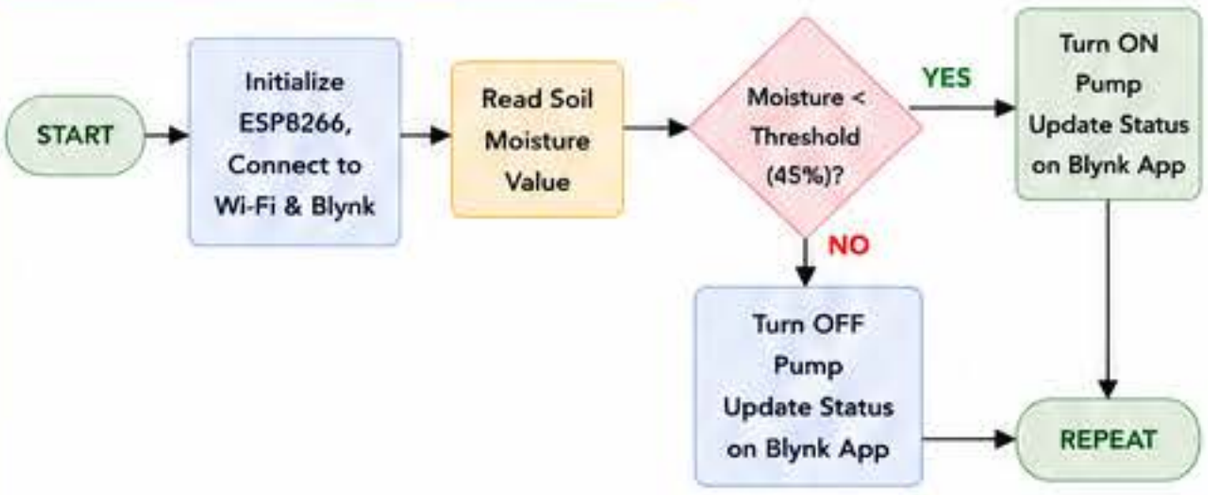
18. TEST RESULTS

Test No.	Soil Condition	Moisture (%)	Pump Status (Auto)	LED	Blynk App	Result
1	Very Dry	18%	ON	ON	Pump ON	PASS
2	Dry	32%	ON	ON	Pump ON	PASS
3	Moist	48%	OFF	OFF	Pump OFF	PASS
4	Wet	65%	OFF	OFF	Pump OFF	PASS
5	Very Wet	82%	OFF	OFF	Pump OFF	PASS

OBSERVATIONS:

- System automatically turned ON pump when moisture < threshold (45%).
- Pump turned OFF when moisture ≥ threshold.
- Blynk app displayed correct real-time values.

19. SYSTEM OPERATION FLOWCHART



20. TROUBLESHOOTING GUIDE

Problem	Possible Cause	Solution
No data on Blynk App	Wi-Fi not connected / Wrong Token	Check Wi-Fi and Blynk Auth Token
Moisture reading not correct	Sensor not inserted properly / Dry soil	Insert sensor properly and water the soil
Pump not switching ON	Relay wiring issue / Power issue	Check relay connections and power supply
Pump always ON	Threshold value too high	Adjust threshold in code and test
App shows offline	Internet connection lost	Ensure stable Wi-Fi connection

NOTE: Calibrate the moisture sensor for accurate results. Keep the system in protected area from rain and sun.

21. SYSTEM SPECIFICATIONS

- Microcontroller : ESP8266 (NodeMCU)
- Soil Moisture Sensor : Capacitive / Resistive Type
- Relay Module : 1 Channel (5V)
- Water Pump : 5V / 12V DC
- Communication : Wi-Fi (2.4 GHz)
- Platform : Blynk IoT Cloud
- Operating Voltage : 5V DC
- Power Supply : 5V DC (1A Adapter / Power Bank)
- App Support : Android & iOS
- Range : Up to Wi-Fi Coverage

22. CONCLUSION

- ✔ The IoT Smart Garden Irrigation System monitors soil moisture in real-time and automates the water pump based on the moisture level.
- ✔ It provides both automatic and manual control through the Blynk App.
- ✔ The system helps in saving water, time and improves plant health.
- ✔ It is cost effective, easy to implement and user friendly.
- ✔ This project can be extended for multi-zone irrigation and weather based control.

APPLICATIONS

NOTE:

- Use proper power supply.
- Ensure correct wiring before powering ON.
- Keep the system away from water and moisture.
- This project is for educational and demonstration purpose only.

Dr V Purandhar Reddy, Narsimha Reddy Engineering College, Hyderabad.

SMART HEALTH
SMART LIFE

IoT PATIENT HEALTH MONITORING SYSTEM

Real-time Health Monitoring | Anytime, Anywhere

1. ABSTRACT

The IoT Patient Health Monitoring System enables continuous monitoring of vital signs such as Heart Rate, SpO_2, Body Temperature and ECG (optional) in real-time. The data is transmitted to the cloud using ESP32 Wi-Fi module and can be accessed by doctors and caregivers through a mobile app (Blynk). The system helps in early detection of health issues and provides timely medical assistance.

2. OBJECTIVES

- To monitor vital health parameters in real-time.
- To upload patient data to cloud for remote access.
- To provide alerts in case of abnormal readings.
- To enable doctors/caregivers to monitor patients remotely.
- To build a low-cost, portable and accurate health monitoring system.

3. COMPONENTS USED


ESP32 Wi-Fi Module


MAX30102 (Heart Rate & SpO_2) Sensor)


LM35 (Temperature Sensor)


ECG AD8232 (Optional)

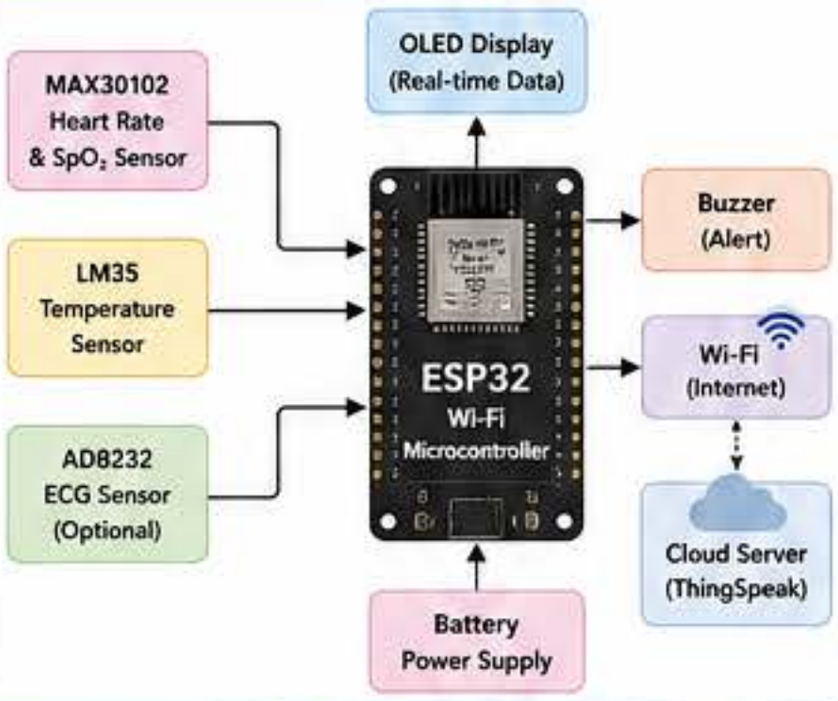

0.96" OLED Display


Buzzer


Li-ion Battery (3.7V)


Jumper Wires

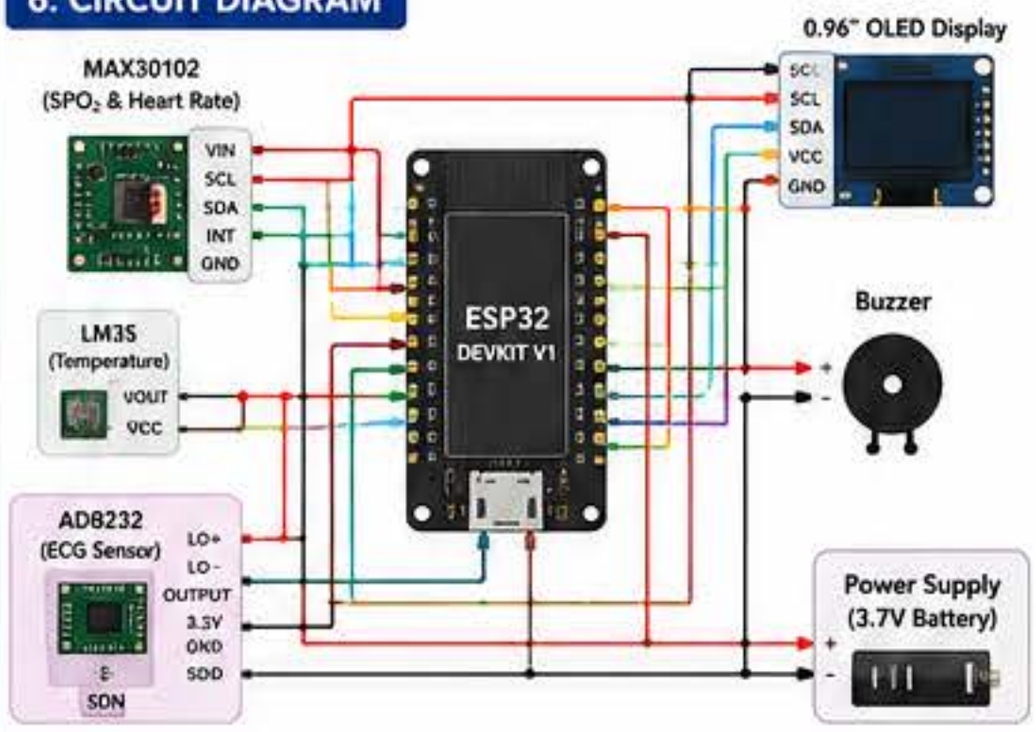
4. SYSTEM BLOCK DIAGRAM



5. WORKING PRINCIPLE

1. Sensors collect the patient's vital signs.
2. ESP32 reads the sensor data and processes it.
3. Data is displayed on OLED in real-time.
4. Data is sent to cloud server (ThingSpeak) using Wi-Fi.
5. Doctors/Caregivers can view data on Blynk mobile app.
6. If any parameter crosses the set threshold, an alert is triggered (buzzer + app notification).

6. CIRCUIT DIAGRAM



7. MOBILE APP DASHBOARD (BLYNK)

- Real-time Monitoring
- Historical Data Graphs
- Alerts & Notifications
- Remote Access (Anywhere)
- User Friendly Interface
- Multi-user Access

8. PIN CONNECTIONS

Component	ESP32 Pin	Description
MAX30102 VIN	3.3V	Power Supply
MAX30102 GND	GND	Ground
MAX30102 SCL	22	I2C Clock
MAX30102 SDA	21	I2C Data
MAX30102 INT	19	Interrupt
LM35 VOUT	34	Analog Input
LM35 VCC	3.3V	Power
LM35 GND	GND	Ground
AD8232 OUTPUT	32	Analog Input
AD8232 LO+	25	Lead Off Detect
AD8232 LO-	26	Lead Off Detect
AD8232 3.3V	3.3V	Power
AD8232 GND	GND	Ground
OLED SCL	22	I2C Clock
OLED SDA	21	I2C Data
OLED VCC	3.3V	Power
OLED GND	GND	Ground
Buzzer +	27	Digital Output
Buzzer -	GND	Ground

9. PARAMETERS MONITORED

Heart Rate (BPM)
Monitors the number of heart beats per minute.

SpO_2 (%)
Measures the oxygen saturation level in the blood.

Body Temperature (°C)
Monitors the body temperature of the patient.

ECG (Optional)
Monitors electrical activity of the heart.

10. SOFTWARE REQUIREMENTS

- Arduino IDE (Code Development)
- Blynk IoT Platform (Mobile App)
- ThingSpeak (Cloud Server)
- Libraries:
  - Wire.h (I2C Communication)
  - MAX30105.h
  - Adafruit_Sensor.h
  - Adafruit_MAX30105.h

11. APPLICATIONS

- Remote Patient Monitoring
- Elderly Care
- Hospitals & Clinics
- Emergency Services
- Home Healthcare
- Fitness & Wellness Tracking

12. ADVANTAGES

- ✓ Real-time health monitoring
- ✓ Remote access from anywhere
- ✓ Early detection of abnormalities
- ✓ Reduces hospital visits
- ✓ Cost-effective and portable
- ✓ Data storage & history analysis
- ✓ Alerts for critical conditions

13. FUTURE SCOPE

- Integration with AI for disease prediction
- Addition of more sensors (BP, Glucose, etc.)
- Integration with Electronic Health Records (EHR)
- Voice assistant & SMS alerts
- Wearable device with long battery life
- Edge computing for faster decision making

14. TESTING & RESULTS

- ✓ All sensors tested for accuracy.
- ✓ Data transmitted to cloud successfully.
- ✓ Real-time data viewed on Blynk app.
- ✓ Alerts triggered on abnormal values.
- ✓ System is stable and portable.

Parameter	Normal Range	Test Reading	Status
Heart Rate (BPM)	60 – 100	72	Normal
SpO_2 (%)	95 – 100	98	Normal
Temperature (°C)	36.0 – 37.5	36.6	Normal
ECG (mV)	-	Normal Wave	Normal

15. CONCLUSION

The IoT Patient Health Monitoring System provides a reliable, real-time and cost-effective solution for continuous monitoring of vital signs. It helps in early detection of health issues and enables timely medical intervention, improving patient care and saving lives.

"Better Monitoring, Better Care, Better Health."