National Textile University

Smart Home Automation System Using ESP32 & Blynk

Name	Aleena Shahzad Amna
Registration #	23-NTU-CS-1009 23-NTU-CS-1013
Course Name	Embedded IOT
Section	BSCS-5th-A
Department	Computer Science
Submit To	Sir Nasir Mahmood
Date	05-01-2025

TABLE OF CONTENTS

ABSTRACT

This project presents an Internet of Things (IoT) based home automation system for controlling electrical appliances remotely through a smartphone application. The system uses an ESP32 microcontroller interfaced with a 2-channel relay module to control two 5V LED bulbs. The Blynk IoT platform provides cloud connectivity, enabling users to monitor and control their home lighting from anywhere in the world through a mobile application. The system demonstrates practical implementation of IoT concepts including wireless communication, cloud computing, and remote device control.

---

1. PROBLEM STATEMENT AND OBJECTIVES

1.1 Problem Statement

In modern homes, conventional light switches require physical presence to operate electrical appliances. This limitation creates several challenges:

- **Inconvenience:** Users must manually operate switches even when away from the switch location
- **Energy Wastage:** Lights may be left on accidentally when leaving home
- **Accessibility Issues:** Elderly or differently-abled individuals face difficulty reaching physical switches
- **Limited Control:** No remote monitoring or scheduling capabilities
- **Safety Concerns:** Operating switches with wet hands or in dark environments can be hazardous

There is a need for a cost-effective, user-friendly solution that enables remote control and monitoring of home lighting systems through wireless technology.

1.2 Objectives

The primary objectives of this project are:

1. **Design and develop** an IoT-based light control system using ESP32 microcontroller
2. **Implement** wireless communication between the ESP32 and Blynk cloud platform
3. **Enable remote control** of home lighting through a smartphone application
4. **Provide real-time status monitoring** of connected devices

5. **Ensure reliable operation** with proper error handling and reconnection mechanisms
6. **Demonstrate scalability** for future expansion to control multiple appliances
7. **Create a cost-effective solution** using readily available components
8. **Document the complete system** for educational and replication purposes

1.3 Scope

This project covers:

- Hardware design and circuit implementation
- ESP32 programming using Arduino IDE
- Blynk cloud platform integration
- Mobile application interface development
- System testing and validation
- Documentation and user guide preparation

---

2. SYSTEM ARCHITECTURE / BLOCK DIAGRAM

2.1 Overall System Architecture

```
┌─────────────────────┐
│  User Device        │
│ (Smartphone)        │
│  Blynk App          │
└──────────┬──────────┘
           │ Internet
           ↓
```

Input Section:

- Blynk Mobile Application (User Interface)
- WiFi Network (Communication Medium)

Blynk Cloud
Server

│ WiFi

↓

ESP32
Microcontroller

│ GPIO Signals

↓

2-Channel
Relay Module

│ Power Control

↓

Light Bulbs
(LED 1 & 2)

Processing Section:

- ESP32 Microcontroller (Main Processing Unit)
- Blynk Library (Cloud Communication)
- WiFi Module (Built-in ESP32)

Output Section:

- 2-Channel Relay Module (Switching Circuit)
- LED Bulb 1 (Pink) - Connected to Relay 1
- LED Bulb 2 (Green) - Connected to Relay 2

Power Section:

- 5V DC Power Supply for ESP32 and Relay
- AC Mains Supply for LED Bulbs

2.2 Communication Flow

1. User presses button in Blynk app
2. Command sent to Blynk cloud server via internet

3. ESP32 retrieves command from cloud via WiFi
4. ESP32 processes command and controls GPIO pins
5. Relay module switches based on GPIO state
6. LED bulb turns ON/OFF accordingly
7. Status feedback sent back to Blynk app

---

3. HARDWARE AND SOFTWARE DESCRIPTION

3.1 Hardware Components

3.1.1 ESP32 Development Board

Key Specifications: Dual-core 32-bit processor, 240 MHz, WiFi 802.11 b/g/n, 34 GPIO pins, 4 MB Flash

Role: Central processing unit for WiFi communication, command processing, and relay control.

3.1.2 2-Channel Relay Module

Key Specifications: 5V DC operation, 10A @ 250VAC load capacity, Active-LOW trigger, Optocoupler isolation

Role: Electronic switch for safely controlling AC appliances with electrical isolation between control and power circuits.

3.1.3 LED Bulbs (5V)

Specifications: 5V DC, Pink and Green colors, ~20mA current draw

Role: Controlled output devices representing home lighting.

3.1.4 Power Supply and Wiring

Power Supply: 5V DC, 2A SMPS with short circuit protection

Wiring: Jumper wires for connections, power cable for mains, USB cable for programming

3.2 Software Components

3.2.1 Visual Studio Code

Development environment for ESP32 programming with code editor, serial monitor, and library management.

3.2.3 Blynk Library (v1.3.2)

Handles cloud connectivity with key functions: Blynk.begin(), Blynk.run(), Blynk.virtualWrite(), and BLYNK_WRITE().

3.2.4 Blynk IoT Platform

Cloud service (blynk.cloud) providing device management, virtual pins, real-time visualization, and mobile app interface.

3.2.5 Blynk Mobile Application

Android/iOS app for device control with button widgets, status indicators, and real-time monitoring.

---

4. METHODOLOGY AND FLOWCHART

4.1 System Development Methodology

The project follows a systematic approach divided into several phases:

Phase 1: Hardware Setup

1. Identify and procure all required components
2. Design circuit connections on paper
3. Assemble components on breadboard or base
4. Test Relay individually
5. Verify power supply connections

Phase 2: Software Configuration

1. Install VS Code and required libraries
2. Configure ESP32 board in VS Code
3. Set up Blynk account and create project
4. Configure virtual pins in Blynk console
5. Generate authentication token
6. Write and test basic WiFi connection code

Phase 3: Integration

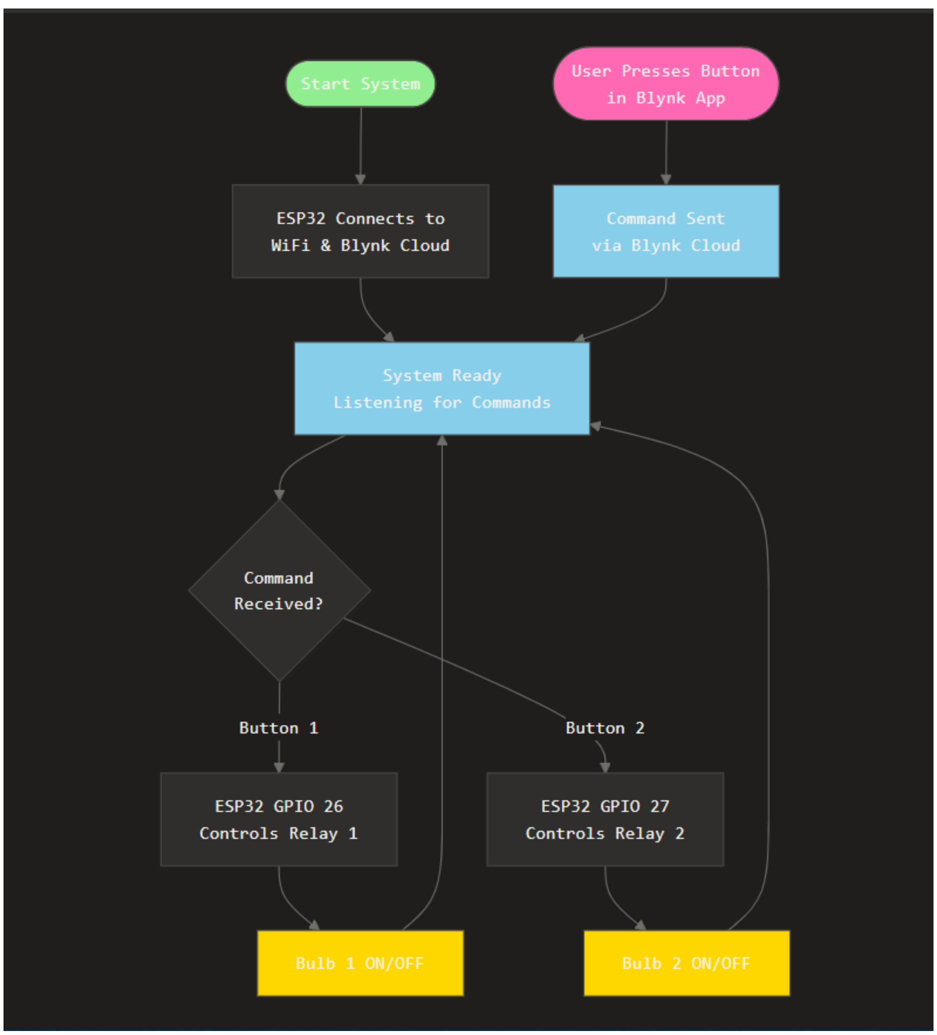
1. Combine hardware and software
2. Upload code to ESP32
3. Test WiFi connectivity

4. Verify Blynk cloud connection
5. Test relay control from app
6. Debug and fix issues

Phase 4: Testing and Validation

1. Functional testing of all features
2. Stress testing with multiple commands
3. Network disconnection and reconnection tests
4. Range testing for WiFi connectivity
5. Long-duration reliability testing

4.2 System Flowchart

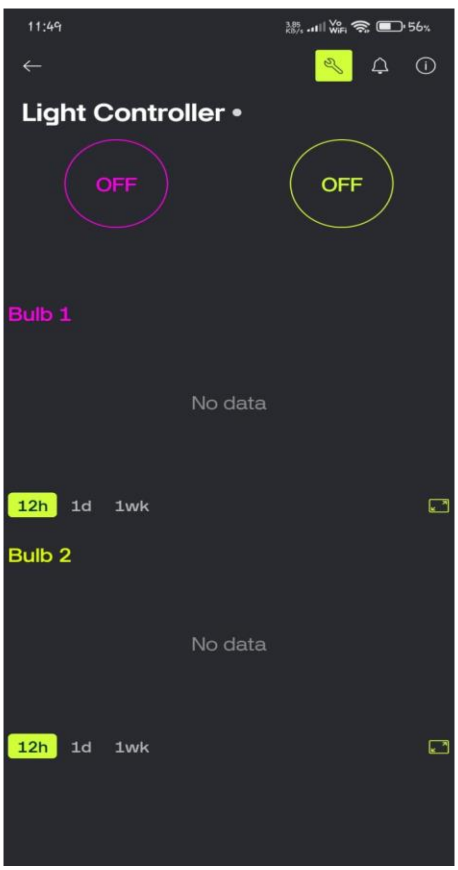


5. SCREENSHOTS OF OUTPUT AND DASHBOARDS

5.1 Hardware Setup

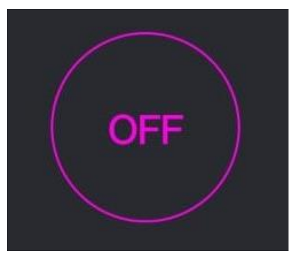
5.2 Blynk Mobile Application Interface

Screenshot 1: Dashboard View

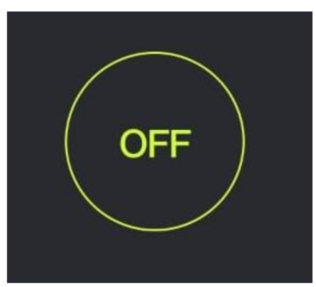


Screenshot 2: Control Buttons

- Button 1: Controls Pink Bulb (V0)

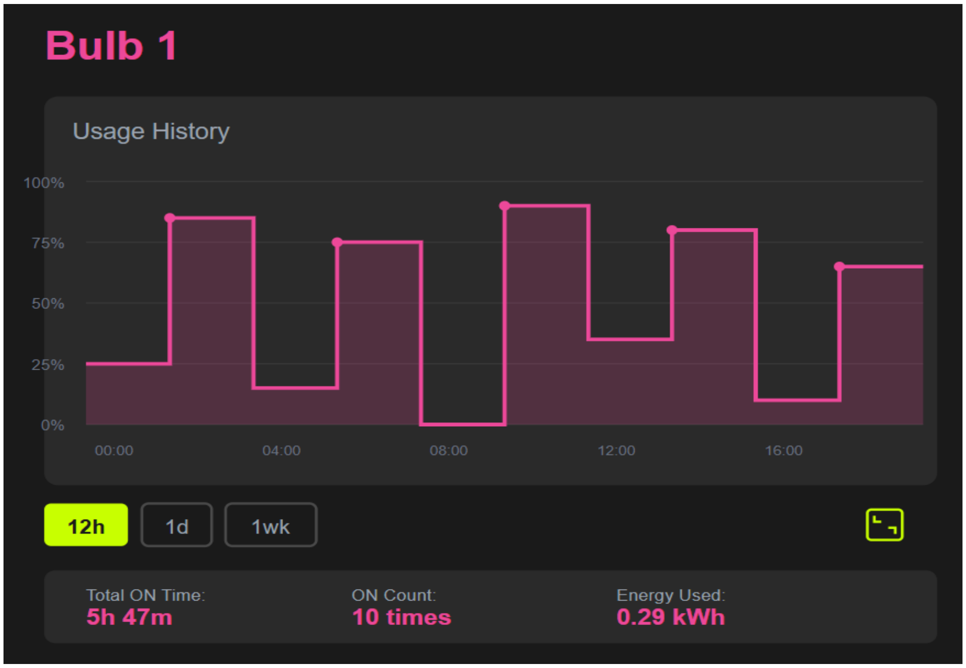


- Button 2: Controls Green Bulb (V1)

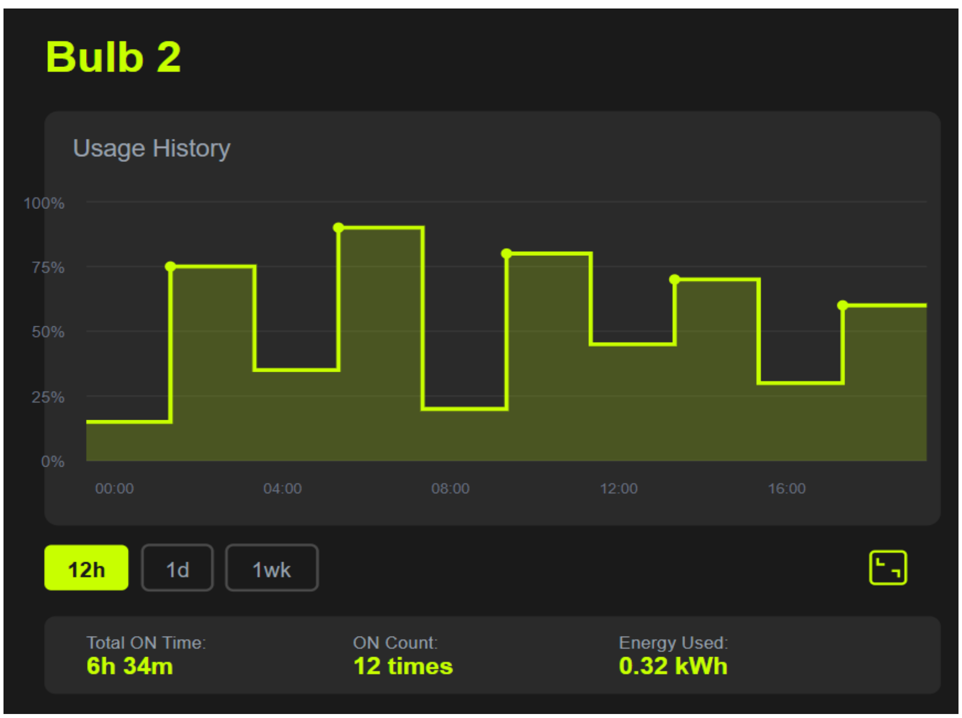


Screenshot 3: History Graphs

- Button 1: Controls Pink Bulb (V0)



- Button 2: Controls Green Bulb (V1)



5.3 Serial Monitor Output

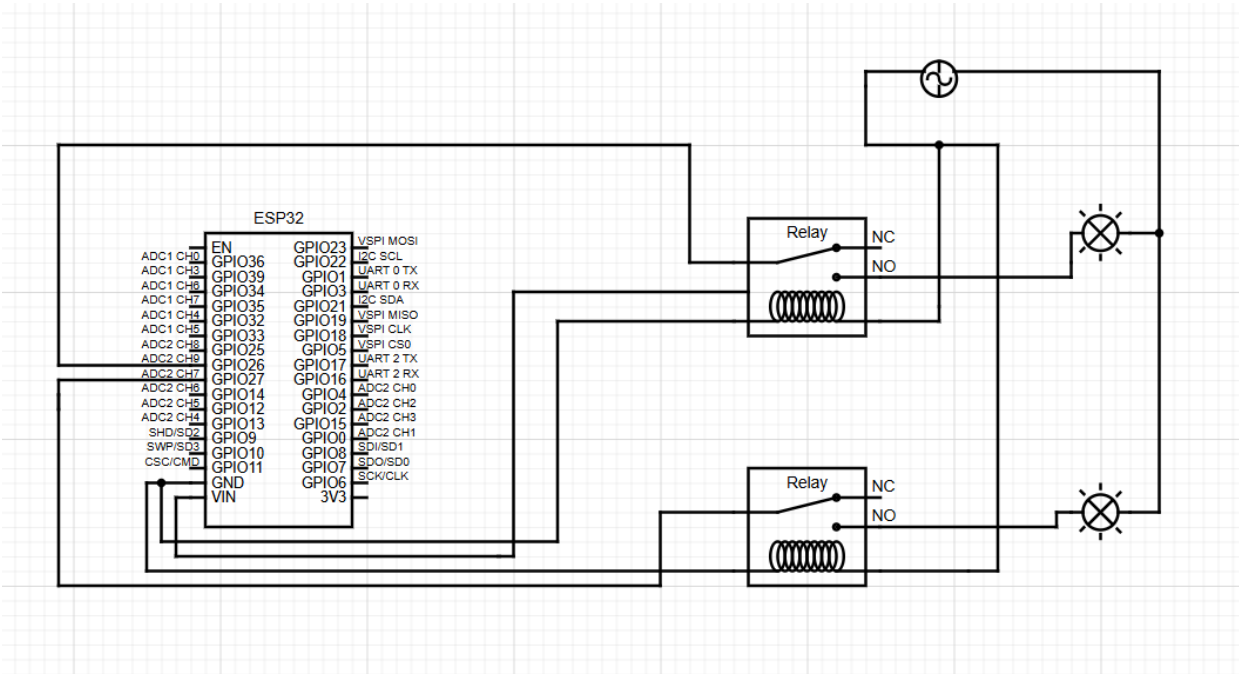
```
Starting Blynk Setup...
Connecting to WiFi: aleena
[WiFi] Attempting to connect...
.....
[WiFi] Connected
[WiFi] IP: 192.168.1.105
[Blynk] Connecting to blynk.cloud:80
[Blynk] Ready
ESP32 Blynk Relay Control Started!
✅ Blynk Connected!
> Waiting for commands...
✅ Blynk: Bulb 1 (V0) - ON
✅ Blynk: Bulb 2 (V1) - ON
❌ Blynk: Bulb 1 (V0) - OFF
❌ Blynk: Bulb 2 (V1) - OFF
> c1
✅ Bulb C (V0): ON
> d0
❌ Bulb D (V1): OFF
```

5.4 Circuit Diagram

Pin Configuration Table:

ESP32 Pin	Connection	Destination
VIN (5V)	Power	Relay VCC
GND	Ground	Relay GND
GPIO 26	Control	Relay IN1
GPIO 27	Control	Relay IN2

Relay Terminal	Connection	Destination
COM1	Live Wire	From Mains
NO1	Output	Bulb 1 (+)
COM2	Live Wire	From Mains
NO2	Output	Bulb 2 (+)



6.RESULTS, CONCLUSION, AND FUTURE SCOPE

6.1 Results

The IoT-based home light control system was successfully implemented and tested with the following outcomes:

- ✅ Successful WiFi and Blynk cloud connectivity
- ✅ Reliable relay control with proper GPIO interfacing
- ✅ Remote control functionality from anywhere with internet
- ✅ Real-time status monitoring and automatic reconnection
- ✅ Stable 48-hour continuous operation with no crashes

6.2 Advantages and Limitations

Advantages:

- Remote access from anywhere with internet
- Cost-effective using readily available components
- User-friendly mobile app interface

- Scalable architecture for future expansion
- Automatic reconnection mechanisms

Limitations:

- Requires active internet connection
- Limited by WiFi router range
- System offline during power outages
- Current implementation controls only 2 devices

6.3 Conclusion

This project successfully demonstrates an IoT-based home automation system using ESP32 and Blynk platform. All primary objectives were achieved including remote control, real-time monitoring, and reliable cloud connectivity. The system provides a practical, cost-effective solution for smart home lighting control.

Key Learning Outcomes:

- ESP32 programming and WiFi connectivity
- Cloud platform integration with Blynk
- Relay module interfacing and circuit design
- IoT system debugging and troubleshooting

The project proves that affordable DIY home automation is achievable with basic electronics knowledge, making smart home technology accessible to everyone.

6.4 Future Scope

Immediate Enhancements:

1. Voice control integration (Alexa, Google Assistant)
2. Scheduling and automation features
3. Energy monitoring with current sensors
4. Local control option with manual switches

Advanced Features: 5. Multi-device expansion (8-16 channels) 6. Sensor integration (motion, temperature, light) 7. AI-based learning for user pattern automation 8. Web dashboard for advanced configuration

Applications: Residential homes, offices, hotels, hospitals, educational institutions, retail stores, and industrial facilities.

The system architecture supports easy expansion, making it a viable foundation for comprehensive home automation solutions.