A Smart Home Automation Model for Remote Light Control Using IOT and Node MCU ESP8266

Shipra Moharana[1], Varsha Gola[2]

Dept. of Computer science and technology (of Affiliation) Sharda University Greater Noida , Uttar Pradesh

moharanashipra@gmail.com

varshagola1979@gmail.com

ABSTRACT

This research paper deals with a basic and effective smart home automation model developed to switch on or off a light using a button with the help of Node MCU ESP8266, Arduino Uno, and relay module. Due to increasing IOT technology in the world today, home automation systems have enhanced in flexibility and ease of use to manage the home appliances through remote methods. In this project, the Arduino Uno provides power to the circuit but Node MCU ESP8266 controls the connectivity aspect as well as the logic part. Coup is made by a relay module which controls the light to turn on and off each time the button input is received by the Node MCU to act upon.

KEYWORDS: Smart Home Automation, IOT (Internet of Things), Node MCU ESP8266, Relay Control System and Remote Light Control

1. OBJECTIVES

The above model aims at showing a cheapest, efficient and easy method of implementing home automation essentials. This setup could be extended to operate various home appliances so as to increase energy efficiency and comfort of the contemporary living environment. This is also true of the system design where it is made up of components that are ordinarily available in the market and those easy to interface. The testing shows that the model works well with immediate response to the button presses to adjust the light. Lastly this research unevenly stresses on the microcontroller based automation models in terms status of realizing the practical and manageable smart home solutions.

2. INTRODUCTION

Smart home technology is often used interchangeably with home automation- the management, monitoring and automation of systems and home devices normally through the internet connection. Smart homes, as an idea, have gone through some development processes over the course of years and now influence people's experience of commensal space. The demand for IOT technology has risen from the luxury or middle-class home automation systems and later it expanded to several homes across the globe. From isolated operations like lighting, and temperature adjustment to advanced systems such as safety and energy control, smart home technology enhances the level of comfort, security and efficiency.

These are the reasons that make people to prefer automated systems so much in the current world. Home automation not only improves the comfort of the inhabitants, but also has the potential of realizing great energy savings which will result in environmental consciousness in the world. Smart homes can save energy, cut costs, and have a lighter ecological footprint because they can automatically fine-tune the use of home electronics and appliances. Besides, these systems enhance security by providing real-time monitoring, producing alerts and allowing the user to maintain their home safety via the internet.

This paper starts by presenting a simple , practical guide on how to implement a home automation system that is intended to turn on a light using a button. Using easily obtained sources such as Node MCU ESP8266, Arduino Uno, and a relay module, this model illustrates a way in which basic devices can effectively be incorporated in an efficient but cheap adoption. Such configuration can be employed as a support for more complicated scenarios, and the goal of the present work is to demonstrate that home automation using microcontrollers is viable for typical homeowners.

3. LITERATURE SURVEY

Earlier last year, in 2023, Vaidya and Vishwakarma had focused on the comparison of technologies used in smart home system, such as GSM and Bluetooth, IoT and PIC microcontrollers using ZigBee modulation. They compared the advantages and disadvantages for each technology and concluded that IoT is the most impactful approach to remote controlling and monitoring home appliances. They also envisaged that with increase in popularity of smart homes in the future, more innovations will be focused on automation and home security.

In their article, Lin et al. (2022) developed a neurocomputing-based Smart Home Energy Management System (SHEMS) wherein various load forecasting models, including autoregressive MLP and LSTM, were introduced. They focused on energy decomposition for accurate appliance level monitoring and pointed teeth towards accuracy depiction through smart meters. According to their results they found that several improvements in SHEMS could help foster efficiency and increase product responsiveness to energy fluctuations.

Smart Residential Load Control Technique for Appliance Identification in Smart Homes Using IoT and AI was proposed in the study by Datta et al. published in 2023. Based on applications of photovoltaic power, this research focuses on the synchronization issues in home

automation while demonstrating IoT and pattern recognition in enhancing load management in homes.
Sankar et al. (2021) examined smart home advancement for elderly and disabled people with regard to artificial home intelligence, hand movement detection, and brain control systems. According to their studies, control systems should be easy to access and preferably smartphone-based or voice command operated in order to enhance the independent living of those communities, while they also suggested some ‘’application simulations’’ for practical implementation.

Ramkumar et al., 2023 surveyed the developments of smart home systems with smart phone control, emphasizing security and alertness on emergencies. Their system integrated Arduino and Android Studio, which included an ‘Ultra Panic Mode’ in emergency situations with the input of authority and neighbours. It also helped in improving home security and ease of using the security measures in a home.

To develop new knowledge, Sinha (2022) proposed the use of AutoHome – the Decentralized voice-controlled smart home automation system; It incorporated the use of technologies like ESP8266, REST, API, MQTT, and Bluetooth. Such features were acknowledged with AutoHome scalability and environmental monitoring; Miller and Rao mentioned that voice-controlled decentralization provides large flexibility of home automation.

Examining the smart home system and the application of Google Assistant in particular, Hussain et al. (2023) considered aspects of energy effectiveness and user convenience. They also tried to find inexpensive ways of achieving automation, allowing smart home technology to reach additional people without breaking the bank.

Low-cost and flexible IoT-based smart home automation system with AI inclusion has also been introduced by Tsankov et al. (2022). Their system was meant to cut the energies and enhance the security and thus they proved their idea that IoT smart home could make the home more accessible and multifunctional.

Sridhar et al. (2023) proposed the smart home automation using IoT technology comparing with GSM and Bluetooth. Their study indicated that IoT was stable and convenient especially for elderly and disable users and described a low cost microcontroller based home automation system.

Kumar and Chaudhury (2023) designed a versatile smart home automation using IoT with ESP8266 and Raspberry Pi. APC’s design focused on real-time control through the IoT and low-cost impact incorporating sensors, actuators, and Android control, augmenting comfort and security approval for users.

Srinivasan et al. (2021) explored the use of Arduino-based voice control systems for elderly medication management. They highlighted the potential of voice recognition technology to improve adherence and health outcomes for elderly individuals, emphasizing the role of smart technology in daily life assistance.

Shabber et al. (2023) have proposed an IoT home automation system called iHAS for smart city requirement with GSM and RF for appliances coordination. They also presented, how iHAS supports real time home health care, especially for elder and disabled ones, and how IoT plays a crucial rule for Smart Urban Home Solutions.

Akhmetzhanov et al. (2023) analyzed the deployment of the video surveillance system in smart home districts employing IoT devices such as Raspberry Pi and Home Assistant. The authors focused on the cost of IoT surveillance systems and how they can be operated through a smartphone; offering a better security option than conventional security measures.

Further, Pujara et al, in 2022, presented an innovative home automation system which incorporates smart lights, smart temperature and moreover smart security systems. Their research was focusing on improvement of systems with the help of RFID and face recognition, which creates the basis of future smart home.

Nandhini et al. (2023) discussed design of IoT based smart home automation system with voice operated appliances using Raspberry Pi board with main concerns on user authentication and ease of use for elderly and disable person. He proved that the integration of the system with Google Cloud for voice recognition is as straightforward, efficient and real-time for the users.

Convenient, safe, and energy-efficient home automation by implementing IoT technologies has interested Joha et al. (2023). Their system that employed sensors, NodeMCU and the said Blynk app made for real-time monitoring and safe guarding against any hazards which helped enhancing the worthiness and safety of homes and users’satisfaction.

Verma et al. (2022) explore cloud-based smart home security system using ESP32, motion detector and face recognition and real-time alert system. They explained the viability of this system and show how it can switch off appliances, thus encouraging energy conservation and security.

4.MATERIALS AND COMPONENTS

This project employs a common and easily sourced IOT components to develop a home automation system that turns on a light through a button press Control and Monitoring done through the Blynk IOT app. These are the Node MCU ESP8266, Arduino Uno, Relay module and a virtual button which is an icon in the Blyn’ app. All these components have a very important role to play in the overall working the system.

1. Node MCU ESP ho8266

Node MCU ESP8266 is a wireless microcontroller for Internet of Things application that incorporates a Wi-Fi module and is easy to program. In this project the Node MCU handles the control logic and shipments of the Blynk IOT application, remotely. By connecting it with the internet, the Node MCU makes it possible for the system to be operated through the Blynk app from any device. When a button is pressed inside an app, a signal is transferred to Node MCU that in turn processes this signal and turns on or off a relay module, therefore a light. This microcontroller receives input from the app and sends output to the relay module which is the control of the system.

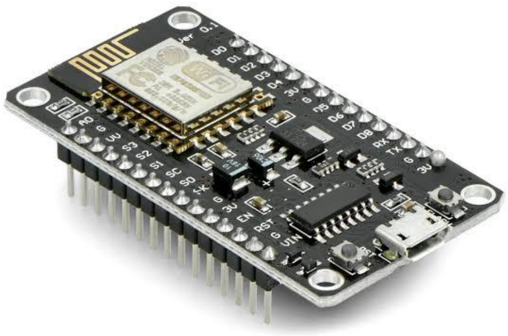

FIGURE 1. NODE MCU ESP8266

2. Arduino Uno

The Arduino Uno is an open source microcontroller board used in prototype development and most electronic applications. In this configuration, its chief purpose is to provide stable power supply to the Node MCU as well as the relay module in order to supply the necessary voltage and current for each of the devices. The power supply hereby used through the Arduino Uno ensures the constant and stable performance of the system.. However, it does not participate in the control logic directly; it lights up the system and contributes significantly toward maintaining the functionality of the system through time as it is powered by the Arduino Uno.

FIGURE 2. ARDUINO UNO

3. Relay Module

The relay is a bi-stable device which switches the flow of current on and off to the particular light. It is an electrically operated switch that has a low power circuit input from the Node MCU controlling a larger power circuit output such as a residential light. As for this project, the relay module is joined with the light and receives commands from the Node MCU. Depending on whether the Blynk app's button is pressed, the Node MCU sends a signal to the relay module and this switches the current either making the light work or stop working. Such arrangement entails secure and effective operation of electrical equipment from a distance without physical input.

FIGURE 3. RELAY MODULE

4. Button/Switch

This control option is connected with the Blynk IOT Application. The button in this project is an artificial button that is generated by Blynk IOT application. Blynk is one more well-known IOT platform that allows users to control hardware remotely by a smartphone application, so it will be optimal for this project. A button in the Blynk app is created to toggle on and off signals between the Blynk app and Node MCU. When the user touches the virtual button shown in the android application, the android applications sends a string over Wi-Fi to the Node MCU to switch the state of the relay module thereby controlling the light. Availability of this virtual button offers ease of remote control and displaces the conventional switches and enhances the uniqueness of controlling the light from any place of the world through the internet.

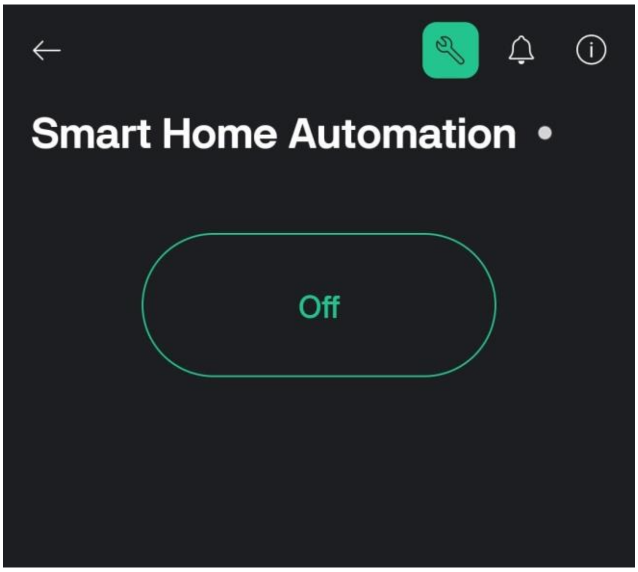

FIGURE 4. BYNK APP BUTTON/SWITCH

5. SYSTEM DESIGN AND ARCHITECTURE

BLOCK DIAGRAM

1. **Blynk IOT Application**

- The light is controlled through the Blynk application which is an android and IOS mobile application. In this case, a virtual button on the app sends a control signal to the Node MCU ESP8266 through Wi-Fi in terms of the state of the light: on/off.

2. **Node MCU ESP8266**

- Gets control signals through Wi-Fi from the Blynk app.
- Analyzes the received signal and determines whether it is necessary to use or turn off the relay module following the button's status.
- Sends control commands to the relay module through the proposed architecture.

3. **Arduino Uno (Power Source)**

- Provides constant voltage to the Node MCU ESP8266 and to the relay module as well.
- Helps the system to maintain the same flow of operation to every piece of the equipment particularly during the wireless communication.

4. **Relay Module**

- Receives control signal from the Node MCU ESP8266.
- It has a function of electrical breaker which can either allow or block current flow to the particular light.
- Turns on the light if the signal received is from the Node MCU.

5. **Light (Load)**

- Connected with relay module and gets power supply from the relay module itself.
- Controlled by the relay module, which is undergoing the command from Node MCU and switch 'ON/OFF'.

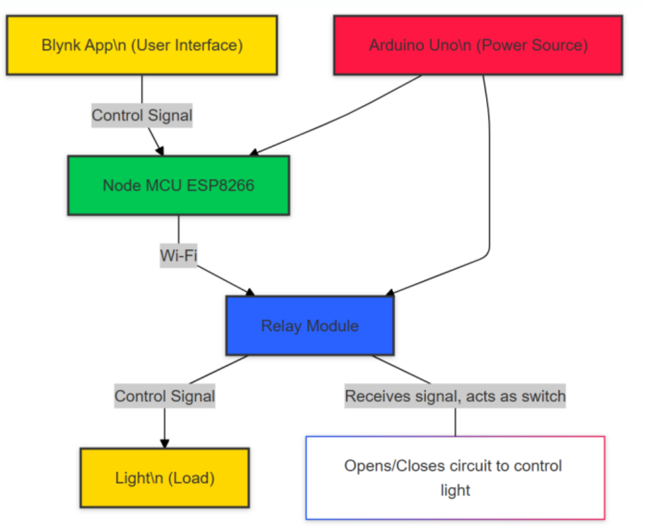

FIGURE 5. BLOCK DIAGRAM

6. METHODOLOGY

This section contains step by step procedure on various aspects including the physical connections and configurations of the hardware components, programming Node MCU ESP8266, the relay control technique employed to control the light. The system uses the Blynk IOT for control of the light using a button in the Blynk application interface.

a. Detailed Procedure in Setting and Configuring the System

Hardware Setup:

1. **Node MCU ESP8266:**

In this setup, the Node MCU ESP8266 is the main microcontroller to drive the entire system. It establishes connection with Wi-Fi network and wirelessly transmits data to and from Blynk application through the internet. Node MCU simply scans through signals received from the Blynk app and in turn triggers the relay module to switch on/off the light.
The current value of the GPIO pin D1 is used in switching of the relay. However, if a different pin is wanted, then it can be altered in the code as is seen in the following section.

2. **Arduino Uno (Power Source):**

Arduino Uno is used here to power up the system. It provides 5V to the Node MCU and the relay module in order that it is able to provide the voltage requirement to those two unit.

3. **Relay Module:**

The relay module is connected to the Node MCU through the GPIO pin which is number one as represented by D1 in this example. The relay is employed to control the high voltage circuit which incidentally in this case is the light. In this case the light comes on or off depending on whether the relay circuit is closed or opened when triggered by the Node MCU.

4. **Light (Load):**

The light is coupled to the relay's Normally Open (NO) and Common (COM) terminals. When the relay is on, the circuit depicts above gets complete and makes the particular light to come on.

5. **Wi-Fi and Blynk App Configuration:**

Burn this code into the Node MCU ESP8266 and create a project in the Blynk app, choosing the suitable hardware (Node MCU ESP8266) and creating a virtual button. They will be linked back to the Node MCU where the virtual button will then send instructions to switch the light on or off.
Once the Blynk project is established, get the auth token from the mobile app because the app connects to the Node MCU. This token is used in the code to complete the connection between the Node MCU and the Blynk app.

6. **Connection:**

The Wi-Fi information must be entered to connect Node MCU ESP8266 to the created network, SSID and password defined in the code.
Make sure that you input the auth token secured in the Blynk application to enable Node MCU and Blynk application interface.

b. General Description and Interaction of Arduino and Node MCU

1. Node MCU Programming:
Like other nodes, the Node MCU is controlled by the Arduino Integrated Development Environment (IDE). Libraries ESP8266 Censoring is included and Blynk simple Esp8266 is included in order to establish a connection between Node MCU and Blynk App.

The setup first sets the serial communication (Serial.begin(9600)) to capture and output info on the system through the console.

2. Pin Mode and Relay Control:
This particular pin is used to control the relay and thus the identify it as an output pin using (pinMode (D1, OUTPUT)). Firstly, the relay is set to HIGH (digitalWrite (D1, HIGH)) which means the light will be off when starting the program. Some of these can be turned on by default with the use of LOW in order to switch on the respective item.

3. Blynk Initialization:
The Blynk.begin(auth, ssid, pass) is a function used to open the Blynk app connection where one enters the auth token, the System in Built name, and password to help the Node MCU to access the internet and bridge the Blynk app.

4. Communication with the Blynk App:
The Node MCU can only connect with the Blynk app through the use of the internet. When the virtual button in the app is impressed then the Blynk server communicates the signal to the Node MCU. The loop() function contains the Blynk.run() function that monitors for any change in Blynk app, such as the pushing of a button.
When the virtual button is pressed then through a program Node MCU responds by switching on or off the state of a relay depending on the button type. When the button is pressed, the relay is squeezed which makes a connection and the light turns on. When the button is released, the relay gets deactivated and so the light source is switched off.

c. Control of the Light Using the Relay

1. Relay Control Logic:
The relay module is in fact a switching device which is managed by the pin D1 of the Node MCU. The relay has Normally Open (NO) contact for power control of the light.
In this project, on pressing the virtual button in the Blynk app, Node MCU pulls the control pin (D1) of the relay high (HIGH). This causes the relay to switch off the NO contact which mean that the circuit is complete, therefore current flows to the light and makes it on.
When the button is released, the Node MCU sends a low signal (LOW) to the relay and causes the NO contact to open and the circuit which powers the light is broken and hence the light goes off.

2. Relay and Light Operation:
The relay satisfies the specification for handling high-voltage loads keeping the Node MCU free from direct engagement with high current demand from the light.
The relay module after being activated by the Node MCU activates or deactivates the high voltage circuit which controls the light, though remotes with the help of Blynk app.

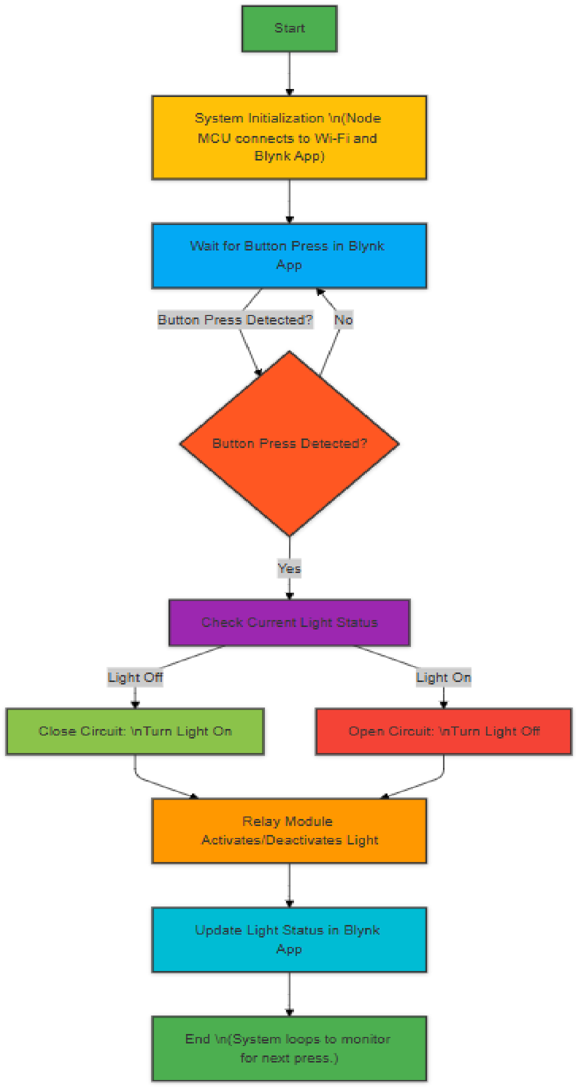

FIGURE 6. FLOWCHART RELATED TO THE WORKING OF THE MODEL

7. CIRCUIT DESIGN AND WIRING

In this section, the circuit diagram is provided and the detail description on the connection of these components to implement the Blynk IOT app for smart home automation model for controlling a light using a button.

Circuit Diagram:

The following components are connected as per the circuit diagram:

1. Node MCU ESP8266 Wi-Fi Controller
2. Arduino Uno (Power Source)
3. Relay Module (Switch for controlling light)
4. Light (Load)
5. Blynk App (Virtual Button)

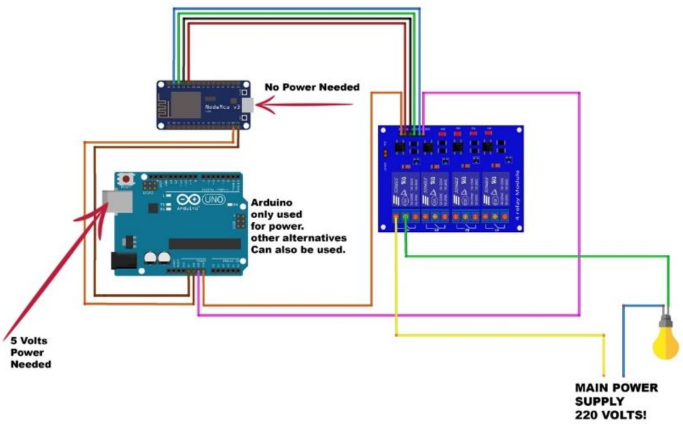

FIGURE 7. CIRCUIT DIAGRAM

1. Node MCU ESP8266 Connections:

- **GPIO Pin D1 of Node MCU goes to Input Pin (IN) of the Relay Module**

This pin turns on or off the relay by sending high or low signal to the relay module necessary for operating the light.

- **GND (Node MCU) to GND (Arduino Uno and Relay Module)**

In order to properly form the circuit, the Node MCU is commonly connected to the ground of both the Arduino Uno and the relay module.

- **Node MCU to VCC (Arduino Uno) to 5V**

It connects this node MCU board to the Arduino Uno where this system is being powered entirely from.

2. Relay Module Connections:

- **IN (Relay Module) to D1 pin in Node MCU**

IRQ of the Relay module is connected to the control pin IN, which is connected to the Node MCU's GPIO D1 to make the relay ON or OFF depending on the signal received.

- **COM (Relay Module) → Hot line (A.C. supply)**

The control terminal COM of the relay links the live wire of the AC power supply to the appropriate circuit board (for controlling the power of the light).

- **NO (Relay Module) → Light**

The Normally Open (NO) terminal of the relay is connected to one terminal of the light that you want to control. Light bulb is energized only if relay gets energized and NO is open.

- **RELAY MODULE GND → ARDUINO UNO GND**

The relay section GND is connected with the GND of the Arduino Uno to have the same common ground for the whole circuit.

3. Arduino Uno (Power Supply):

- 5V Pin (Arduino Uno) is connected to the VCC (Node MCU and Relay Module)
- Power voltage from the Arduino Uno to both the Node MCU and the relay module are powers of 5V.
- GND pin in the arduino uno → GND pin in Node MCU and Relay Module

  The ground of the Arduino is bonded to the ground of both, the Node MCU and the relay module to make the complete power loop closure.

4. Light (Load) Connections:

- It can be a lamp/bulb which is fixed between NO(Relay Module) and COM(Relay Module).
- When the relay is triggered by the Node MCU it makes the circuit between these two terminal to close and hence current is allowed to flow making the light to on.
- The other terminal of the light joins the Live wire (AC) from the power supply part of it.

a. Explanation of Circuit Design and Wiring Configuration

1. **Power Supply and Voltage Regulation:**
   The Arduino Uno is connected to supply the required power for the whole system; 5V in this case to Node MCU and the relay module. Internally Node MCU is working at 3.3V but can accept 5V power through VIN pin that is why connected Node MCU VIN pin with Arduino UNO 5V output.

2. **Control Signal from Node MCU to Relay:**
   • The Node MCU ESP8266 switches on/off the relay by applying 5V on the relay module's input (IN) pin. This HIGH signal triggers the relay which turns the Normally Open (NO) contact to close a circuit and switch the light on.

- When a button in the Blynk app is pressed, the Node MCU changes the state of the relay sending a HIGH or LOW signal to the relay depending on the state of the virtual button.

3. **Relay Module and Light Control:**

- The relay module is a control switch that controls the flow of the high voltages current to the light. Because the Node MCU can only supply low voltage signals (3.3V), the relay module has been required to safely control the high voltage of the light.
- The relay has COM (Common), NO (Normally Open), and NC (Normally Closed) terminals. For this scenario, the NO terminal is applied; it causes

the current to energize the light only when the relay is triggered.

- When the relay is triggered by Node MCU through control pin D1 the NO terminal connects and the current flows from AC supply to the light and hence turns on the bulb. As the relay is turned off the NO contact breaks the circuit and therefore off the light is turned off.

4. **Grounding and Common Reference:**

- All of them namely Node MCU, relay module and Arduino Uno exhibit common ground (GND) that make the signal and power levels compatible. If there was no congruency, then the interactions between the components may not go as require.

5. **Safety Considerations for High-Voltage AC:**

- The relay module enables the system to interface with high voltage equipments such as the light without jeopardizing the Node MCU board.. The relay thus acts as a buffer through which the AC power circuits can be controlled through low voltage signals from the Node MCU.

8. SOFTWARE IMPLEMENTATIONS

a. SOFTWARE TOOLS:

- **Arduino IDE:**

To program the Node MCU ESP8266, The Arduino Integrated Development Environment (IDE) is used whereby users write, compile and upload the code to the board. The Node MCU is controlled by the Arduino IDE where users install board support and libraries required for Blynk app and Wi-Fi control.

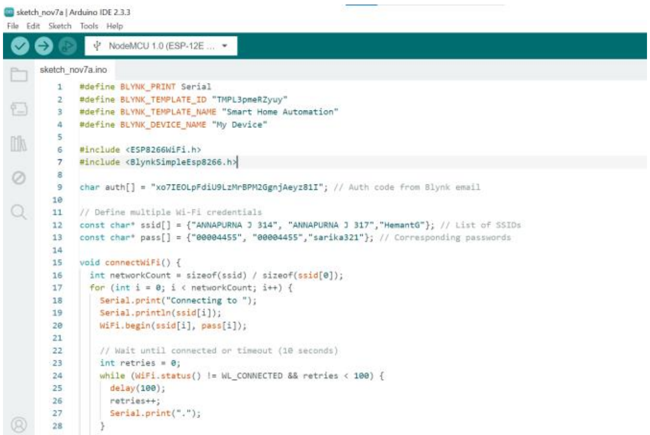

FIGURE 8. ARDUINO UNO

- **Blynk App:**

The Blynk IOT platform is used to implement the graphical User Interface for the smart home automation system. Light is also controlled through on screen buttons in the Blynk app which greatly demands less physical hard ware and is available in many portable devices like mobile phone. As such, through the internet, the Blynk app sends signals to the Node MCU to switch on or off the relay module.

FIGURE 9. BLYNK APP

9. TESTING AND RESULTS

Testing Section of the paper describes the testing methodology that was employed to check the functionality of the smart home automation system, outcome analysis of the testing process and comparison of the obtained results with the expected ones.

8.1. TESTING SCENARIOS:

1. Initial System Power-Up Test:

- **Objective**: To confirm if the system establishes connections with the Blynk app and powers up as needed.

- Procedure:

- Charge the system through the power supply by connecting Arduino Uno and Node MCU ESP8266.
- Make certain that there is no error when the Node MCU join to Wi-Fi and also the Blynk app.
- Make sure after launching the Blynk app the toggle button is visible and should work properly.

- Expected Outcome:

- It means that the system should start and the Node MCU should establish connection with Wi-Fi and Blynk application.
- It must give an impression in the light controlling in the Blynk App that the button is active and responsive.

2. Button Press Test (Turning Light On):

- **Objective:** To check that pressing the virtual button in the Blynk application turns on the light.

- Procedure:

- Press the virtual button in the Blynk app.

-Notice the relay is activated (D1 control pin of the relay should be set to LOW).

- Try turning the light on to make sure it lights up when you activate the relay.

- Expected Outcome:

- The relay will energize and close the NO contact, which in its turn, also forms a sequence that allows power to flow into the light.

-The light should come on instantly you press the button.

3. Button Release Test (Turning Light Off):

- Objective: To check if releasing the virtual button in the Blynk application turns the light off.

- Procedure:

- Release the virtual button in the Blynk app.

- Observe the relay being turned off (control pin D1 of the relay should be set to HIGH).

However, it is important to check the light to ensure it goes off when the relay is deactivated.

- Expected Outcome:

- The machine should make the NC contact to close the circuit and allow current to flow, activating the light.

- They were expected to turn off as soon as the button was released.

4. Wi-Fi Disconnection and Reconnection Test:

- Objective: To verify the system's capability to handle Wi-Fi disconnections and reconnect automatically.

- Procedure:

- Turn off the router or disable Wi-Fi on the Node MCU to disconnect it from the Wi-Fi network.

- Observe the system's responses when you attempt to touch the virtual button.

- Reconnect the Node MCU to the Wi-Fi and check if it behaves normally.

- Expected Outcome:

- Thus, if Wi-Fi becomes disconnected, the button should not respond anymore.

- When the Wi-Fi is reconnected, the Node MCU should automatically connect to the Blynk app and the virtual button will become functional again.

5. Multiple Button Press Test (Simultaneous Control from Multiple Devices):

- Objective: To check whether several appliances are able to control the light at the same time using the Blynk application.

- Procedure:

- First, get two mobile devices, for example, a phone and a tablet and use them to toggle the virtual button in the Blynk app.

- When the button is pressed and released on both devices, the behavior of the light should be observed.

- Expected Outcome:

- The light should be activated or deactivated from any of the devices but only the button from the one device can be pressed; the same effect should also be kept on the other device.

- The light should be addressed in all devices so that the system is updated when the state changes on the light.

8.2. Analysis of Results and System Performance

1. System Stability:

- In the course of testing, the system showed satisfactory stability. The Node MCU was able to successfully connect to the Wi-Fi as well as the Blynk app, and all interactions between the devices had no noticeable lag or dropouts.
- The active button on the app was very responsive, and there were no faults with the relay switching from ON and OFF states.

2. Button Control Response Time:

- The time delay during which the virtual button is pressed and when the illuminator lights up was not considerable (usually less than two seconds). Such a response time is normal for a home automation system and as such enhances the efficacy of the system.
- Wi-Fi latency was the main cause of response time but it was also within the acceptable range and there was no delay between the Blynk app and the system.

3. Wi-Fi Stability and Reconnection:

- The system showed resilience and handled the disconnection and the reconnection to Wi-Fi quite efficiently. After a few seconds of no Wi-Fi, the Node MCU was able to connect with the network and the Blynk app on its own as required.
- The system light responded appropriately after the re-establishment of Wi-Fi, and normal service was restored without delay.

4. Simultaneous Control from Multiple Devices:

Without much hassle, the lights were controlled by several devices. When the virtual button was pressed or released on one of the devices, the status updated on the second device almost instantaneously, allowing for synchronized use of all Blynk app connected devices.

8.3. Comparison with Expected Outcomes

- **Power-Up and Wi-Fi Connection:** The system powered up and connected to the Blynk app as expected.
- **Button Control (Light On/Off):** Whenever the button was pressed and released in the Blynk app, the corresponding action of the light i.e. on and off, occurred in the manner expected.
- **Wi-Fi Handling:** The system managed Wi-Fi dropouts and restarts effectively without any significant period of downtime.
- **Multiple Device Control:** As expected, this was not a problem and the system worked as expected in this case.

The expectations were achieved during the testing of the system in each scenario. Control of the light through the Blynk app was satisfactory. Performance, response time, and reliability also did not present any major issues.

10. DISCUSSION

a. Limitations of the Current Model:

- Involves a third party Blynk app and server → The system loses stability if their server is down or Blynk has a problem with their app.
- Does not come with sophisticated control options like dimming or different modes of lighting, so less customization.
- No physical switch; users must control the lights strictly through Blynk app.
- Limited tests in diverse environments; extreme conditions (high humidity level) are not fully validated.
- There is no way to get feedback back to the user about the state of the light (on/off) if there are issues with app or network

b. Potential Improvements and Additional Features:

- **Control Multiple Devices:** Broaden control over different devices (like fans, thermostats) for a more complete smart home experience.
- **Local Control Option:** Switch to offline mode that enables controlling through local network in case of Wi-Fi outage.
- **Increased Security:** Implement encryption for the transmission of data, as well as user authentication, to provide a higher level of security.
- **Sensor Integration:** Add sensors (e.g. motion, temperature) for automated actions such as switching on lights based on room occupancy.
- **Voice Control Compatibility**: Integrate with a voice assistant (such as Google Assistant, Alexa) for hands-free mode.
- **Energy Monitoring:** Integration of an energy monitoring module providing the ability to monitor power consumption for improved energy management.
- **Scheduling Function:** User can set on/off in a scheduled manner through this feature that allows them to handle their devices remotely.

11. CONCLUSIONS

This project focuses on implementing a very primitive smart home automation system using the Node MCU ESP8266, Arduino Uno, relay module with the Blynk IOT Platform to control the light remotely. This system enables the user to control the light from a virtual button in Blynk app application showing how IOT can be used as a tool for improving the convenience and motion of functions in home environment.

These results demonstrate how using low-cost and off-the-shelf components can provide dependable remote control of common household functions. This only allows us to control one light, but it covers the basis for further automation across multiple devices and is a starting point towards more sophisticated smart home systems.

In conclusion, this project shows the possible influence of IOT in home automation, i.e. it can make everyday tasks more efficient and change the user's needs smoothly. The evolution of the IOT technology will bring about more enhancements and security to smart home solutions, making them approachable and configurable, and spurring the future of connected life.

12. FUTURE WORK

1. Mobile App Integration Enhancements:
Consider Blynk's indigenous mobile app development in conjunction with others and discover unique features that can justify the use of the platform, as well as offline accessibility.

2. Voice Control Compatibility:
Pair with voice assistants like Google Assistant, Amazon Alexa, or Siri, ensuring hands-free operation and amplified accessibility for users.

3. Sensor Integration:
Utilize a combination of motion, light, and temperature sensors for automated responses, for instance, switching the lights on due to motion detection, or adjusting brightness according to the ambient light situation.

4. Energy Monitoring and Analytics:
Implement power tracking functions for energy consumption and provide the ability to view the visualized usage patterns, hence, energy efficiency becomes a leading point.

5. Broader Applications:
Incarnations of the model can be integrated into devices such as fans, thermostats, and smart locks to build a smart home that can handle several devices using a central control point.

6. Local Control Option:

Introduce a local control network mode to the system, enabling it to operate offline, thus boosting internet outage resilience and system reliability.

13. REFERENCES

1. Vaidya, V. D., & Vishwakarma, P. (2023). A comparative analysis on smart home system to control, monitor, and secure home based on technologies like GSM, IoT, Bluetooth, and PIC microcontroller with ZigBee modulation.

2. Lin, Y. H., Tang, H. S., Shen, T. Y., & Hsia, C. H. (2022). A smart home energy management system utilizing neurocomputing-based time-series load modeling and forecasting facilitated by energy decomposition for smart home automation.

3. Datta, S., Bhowmick, S., Datta, S., Chatterjee, S., Ghosh, S., Ganguly, B., & Chatterjee, A. (2023). A smart residential load control technique with Internet of Things and pattern recognition based load identification.

4. Sankar, S. M. U., Dhinakaran, D., Kavya, T., Priyanka, S., & Oviya, P. P. (2021). A way for smart home technology for disabled and elderly people.

5. Ramkumar, A., Saravanavel, S., Vaigaiselvam, T., Kamalesh, A., Rajendran, S., & Rajesh, K. (2023). Android controlled smart home automation with security system.

6. Sinha, A. (2022). AutoHom - A decentralized voice-controlled home automation system.

7. Hussain, I. S., Deepalakshmi, S., Benilla, R. J., & Nivetha, C. (2023). Automation of smart home using smartphone via Google Assistant.

8. Tsankov, V., Evstatie, B., & Valova, I. (2022). Concept for an IoT-based electronic system for smart home automation.

9. Sridhar, H. S., Somvanshi, S., Ramsinghan, M., & Nimishambha, H. C. (2023). Designing of smart home automation using IoT.

10. Kumar, K. M., & Chaudhury, S. (2023). Development of a smart home automation system using IoT-enabled devices.

11. Srinivasan, C., Sumanthkumar, C. H., Vardhan, H. C., & Chaitanya, S. (2021). Home automation using Arduino through voice control.

12. Shabber, S. M., Bansal, M., Devi, P. M., & Jain, P. (2023). iHAS: An intelligent home automation-based system for smart city.

13. Akhmetzhanov, B. K., Gazizuly, O. A., Nurlan, Z., & Zhakiyev, N. (2023). Integration of a video surveillance system into a smart home using the Home Assistant platform.

14. Pujara, D., Gautam, R., & Shah, D. (2022). Integrative home automation system - A multifaceted approach.

15. Nandhini, S., Rubla, H., Petchiammal, P., Sivalakshmi, K., Vaz, A. J., & Venkatasamy, B. (2023). IoT-based smart home with voice-controlled appliances using Raspberry Pi.

16. Joha, M. I., Islam, M. S., & Ahamed, S. (2023). IoT-based smart control and protection system for home appliances.

17. Verma, G., Pachauri, S., Kumar, A., Patel, D., Kumar, A., & Pandey, A. (2022). Smart home automation with smart security system over the cloud.

18. Arduino. (n.d.). Arduino Uno. https://www.arduino.cc/en/Guide/ArduinoUno

19. NodeMCU. (n.d.). NodeMCU Documentation. https://nodemcu.readthedocs.io/en/release/

20. Blynk. (n.d.). Blynk: The Most Popular IoT Platform. https://blynk.io/

21. Instructables. (n.d.). IoT Home Automation Using NodeMCU and Blynk. https://www.instructables.com/IOT-Home-Automation-Using-NodeMCU-and-Blynk/

22. Hackster.io. (n.d.). Home Automation Using NodeMCU and Blynk. https://www.hackster.io/