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

CSP391: PROJECT BASED LEARNING REPORT

by

Shipra Moharana

and

Varsha Gola

Under the supervision of

Mr. Dharam Raj

Assistant Professor,

Department of CSE, SSET

Department of Computer Science & Technology

Sharda School of Engineering & Technology

Sharda University, Greater Noida, INDIA

BONAFIDE CERTIFICATE

This is to certify that this project report entitled **"A Smart Home Automation Model for Remote Light Control Using IOT and Node MCU ESP8266"** submitted to **School of Engineering and Technology, Sharda University**, is a bonafide record of work done by "**Shipra Moharana" and "Varsha Gola"** under my supervision.

<<Signature of the Supervisor>>

Dharam Raj

Assistant Professor

Declaration by Author(s)

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We aware that if any part of the report is found to be plagiarized, we are shall take full responsibility for it.

Signature of author 1

Shipra Moharana

Roll number:2201010666

Signature of author 2

Varsha Gola

Roll number:220101077

ABSTRACT

This paper simply demonstrates a smart home automation model for remote light control, using the Node MCU ESP8266, Arduino Uno, and a relay module. The system provides users with the ability to switch off and on a light using a virtual button located within the Blynk IoT application due to the increased flexibility and accessibility provided by the IoT technology. The Arduino Uno powers the circuit, with connectivity and logic provided by Node MCU ESP8266. Now, this model executes light control through a relay module governed by the app regarding the button status. This model represents a cost-effective and efficient model for home automation with potential scope to be scalable enough to regulate other electric appliances, thereby providing greater energy efficiency and more ease for users.

Table of Contents

INTRODUCTION

1.1. Problem Statement:

These are combined requirements of comfort, energy efficiency, and safety for an indoor environment. Home automation solutions that are affordable and user-friendly to meet these needs must be realized. Traditional lighting systems are operated manually, and hence have no accessibility from a distance. Therefore, it has very little to offer in terms of convenience and ease of control. A developed low-cost smart home automation system enables its users to remotely control the lighting of their houses using IoT technology based on a Node MCU ESP8266 and an Arduino Uno module. This solution offers an easy model that uses widely available components to reflect an accessible approach towards home automation. In this respect, it specifies efficient control of lights and promotes further expansion into other appliances.

1.2. Objective:

- **Design a Low Cost Home Automation Model:** Describing the System: This should allow the users to remotely control the home lighting system using products like Node MCU ESP8266, Arduino Uno, and relay module at a very affordable cost.

- **Ease of User:** This will make it easy for users to gain access through Blynk IoT application in controlling their lighting system from any place where they have an internet connection.

- **Applying IoT in Home Automation:** As stated above, IoT can be applied into home automation, for example, in the case of building a smart responsive and efficient remote control for light control, which can further be extended to remote management of other home appliances.

- **Energy Efficiency:** The remote controlling of Lighting will reduce unnecessary power usage and help save energy.

- **Lay the foundation for scalable smart home systems:** Establish a simple automation setup that can be built on and serves as a basis for further growth in the inclusion of higher level features, such as control over multiple devices, sensor interfacing and voice command compatibility.

1.3. Challenges:

- **Network Dependency and Stability :** This system is dependent on the internet for stability. A failure in Wi-Fi or server connectivity, such as with the Blynk app, will make the system unusable until the issue is resolved, reducing the reliability of the system.

- **Security Risks :** Data transport between the IoT device and the app allows a signal to be intercepted, which raises security issues if encryption and authentication are not appropriately utilized.

- **There is no Physical Backup Control:** Since the system functions only through a virtual application interface, users cannot obtain control of the lights if the app or internet connection fails. It has some potential usability issues.

- **Single Scalability:** The model designed currently works to control just one light. The system's scale to have multiple appliances will require extra programming and components and may also require an advanced microcontroller.

- **Response Time and Latency**: The system will depend on minimal response time, that could possibly be influenced by the network delays or by processing lag of the Node MCU ESP8266, particularly under highly trafficked conditions for the network.

- **Environmental and Operational Limitations:** The system hasn't been fully subjected to mixed environmental conditions (such as high humidity and extreme temperatures) that may impact the operations of the hardware, which can actually impair its reliability in real-world different scenarios.

- **Power Source Stability**: The Arduino Uno provides the source of power for the whole system; any instability in its power source will then impact the performance of the system and the dependability of the device operations.

1.4. Contributions:

- **Practical IoT-Based Model for Light Control:** This project contributes a smart home automation model that is low-cost and practical, based on the Internet of Things technology that will control lights wirelessly from anywhere. It illustrates how easily components like Node MCU ESP8266, an Arduino Uno, and relay module can create an easy access of smart home.

- **User-friendly interface for remote control:** The system incorporates a basic and user-friendly interface to the users with the help of Blynk IoT application, enabling users to have full control over lighting remotely. This is a great example of mobile IoT applications in home automation applications.

- **Future Automation Scalability Framework:** The foundation of the project itself is expandable to control other household appliances, thus promoting smart home systems with more sophisticated automation features such as multiple control, sensors, and scheduled operations.

- It allows for the remote adjustment of lighting thus encouraging people to switch off their lights whenever they are not required thereby substantially contributing to reduced energy consumption and more environmental sustainability.

- It gives information about Low-Cost Solutions for IoT systems, as it illustrates an application of such feasible and easily available materials in home automation and could serve as a reference for future projects that look for economical approaches to smart home technology.

- **Educational Model for IoT and Automation Learning :** It is an introductory model that can be used as an educational tool for which students and enthusiasts learn the basics of IoT, microcontrollers, and home automation with the help of hands-on experience using a simple, practical system.

2. APPROACH USED

The following steps outline the system setup, hardware configuration, programming, and control mechanism used in the project:

2.1. Hardware Setup and Configuration:

Node MCU ESP8266: Under control of the Arduino board, connects the robot to the Wi-Fi and manages the Blynk application for switching the light on and off using a relay.

Arduino Uno: Power supply that supplies power to Node MCU as well as the relay module, with the 5V voltage.

Relay Module: Electrically linked to Node MCU's GPIO pin (D1) the relay which is controlled by signals turn the light on/off.

Light (Load): Connected to the relay module Normally Open (NO) and Common (COM) terminals to provide the connection as soon as the relay energisation.

Wi-Fi and Blynk App: When it comes to remotely making adjustments, Blynk gives the user a control button that will turn on or off the light. Node MCU after creating a project and getting an auth token is set to connect with the app of the project.

2.2. Programming and Control Logic

1. **Node MCU Programming:** The programming was done using Arduino IDE supported with the Blynk and ESP8266 library.

2. **Use Relay Control:** Connect relay through GPIO D1 in a configuration as an output pin to turn the light off/on.

3. **Blynk Initialization:** The code has the auth token, SSID, and password to initialize the communication between Node MCU and the Blynk app.

4. **Relay Activation:** When the virtual button is activated, Node MCU signals a relay to close NO contacts and power the light, and when the button is de-activated, the NO contact opens, cutting power to the light**.**

2.3. Circuit Design and Wiring

1.Power Supply: Arduino Uno provides a 5V supply to Node MCU and relay.

2.Signal Control: Node MCU will send high or low signals to control the relay and switch the light on or off.

3. Relay Protection: Relay is a protected high voltage switch for light control without affecting the low-voltage Node MCU.

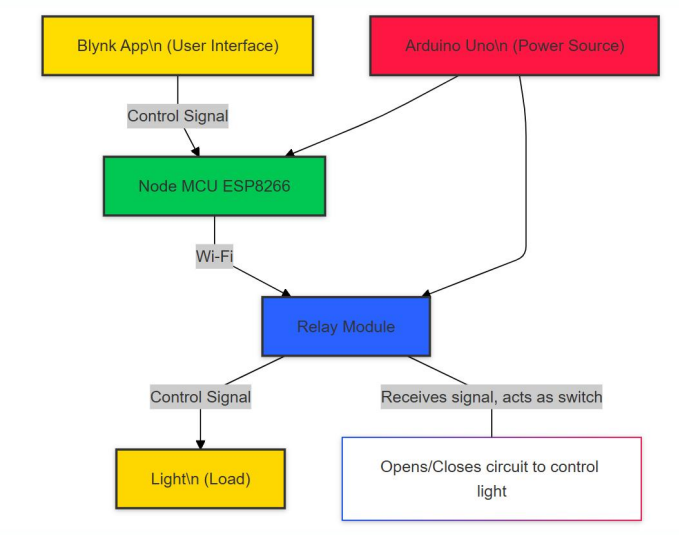

FIGURE 1. CIRCUIT DESIGN AND WIRING

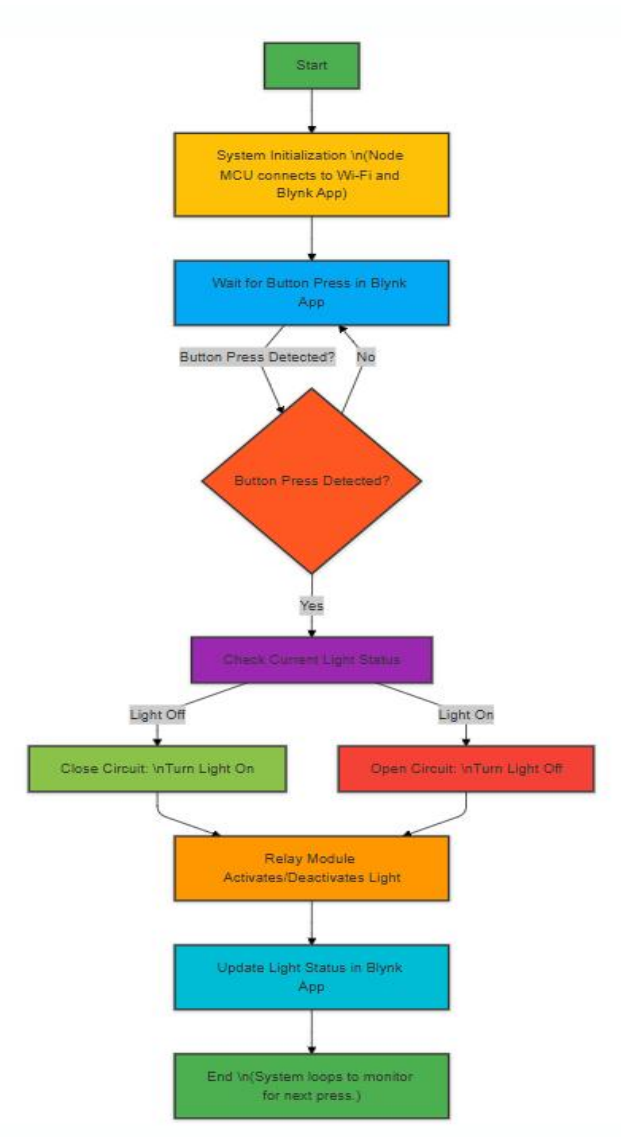

FIGURE 2. FLOWCHART RELATED TO THE WORKING OF THE MODEL

3. RESULT AND DISCUSSION

1. System Stability:

-In the testing procedure, the system proved quite stable. The Node MCU connected successfully to the Wi-Fi as well as Blynk app, with all interactions between the devices without any lag or dropping out.

- The switch button within the application was very sensitive, and there were no errors in the relay to change from the ON and OFF states.

2. Button Control Response Time:

- The time lag between the virtual button being pressed and when the illuminator lights up was not significant, 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.

- Latency due to Wi-Fi was the primary cause, though it was within the range and there was no delay between the Blynk app and the system.

3. Wi-Fi Stability and Reconnection:

- The system showed resiliency and managed to handle the disconnection and the reconnection to Wi-Fi quite efficiently. It took a few seconds for no Wi-Fi after which the Node MCU automatically established a connection with the network and the Blynk app as needed.

- The system light responded as well after the re - establishment of Wi-Fi and was back to normal without further delay.

4.Multi-device Control End

The lights were controlled through several devices without much hassle. Pressing or releasing the virtual button on one of the devices resulted in an almost instantaneous update in the status on the second device, allowing synchronized usage of all Blynk app connected devices.

3.1 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):** Every time the button was pressed and released in the Blynk app, the corresponding action of the light i.e. on and off, happened in the manner expected.

• **Wi-Fi Handling:** The system handled Wi-Fi dropouts and restarts quite effectively without any significant period of downtime.

• **Multiple Device Control:** As expected, it was not a problem, and the system worked as anticipated in this case. The expectations were met during the testing of the system in both cases. Control of the light using the Blynk application was satisfactory. Performance, response time, and reliability did not pose major issues in these aspects.

3.2. 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.

• Not enough tests on various environments; extreme conditions, i.e. high humidity level, not entirely tested.

• There is no way to provide feedback to the user on the status of the light (on/off) if there is an issue with the app or network

3.3. Potential Improvements and Additional Features:

• **Control Multiple Devices:** Expand the control of multiple devices such as fans and thermostats for a much more comprehensive smart home.

• **Local Control Option:** Go to local control mode where it is still possible to control it by using local network when the Wi-Fi is lost.

• **Improved Security:** Apply data encryption for the transfer, as well as user authentication for a better level of security.

• **Sensor Integration:** Add sensors (for example, motion, temperature) to automatically turn on lights depending on the occupancy of the room.

• **Voice Control Compatibility:** Integrate with a voice assistant (like Google Assistant, Alexa) for hands-free mode.

• **Energy Monitoring:** Integration of an energy monitoring module, which allows the monitoring of power consumption for better energy management.

• **Scheduling Function:** This features enable a user to set on/off in a scheduled manner so that one can deal with them remotely.

4. CONCLUSIONS

This project shows a smart home automation system that could easily automate the lights control by making use of a simple IoT-based system via Node MCU ESP8266, Arduino Uno, relay module. It uses the facilities of IoT to make life easier and power saving in home settings, by only controlling the lightings using the Blynk IoT application from anywhere in the world via internet. This was tested for confirmation and proved that the system had reliability, an economical price, and it was user-friendly so that it was practically viable for the homeowners who wanted to enhance their daily life by adding simple automation solutions to it.

The off-the-shelf, low-cost components allow for achieving the feasibility of IoT-based home automation by providing some practical advantages in the way of better energy management and convenience of usage. Some advances in the security attributes must be brought into offline controls, with the current vulnerability of the system being open to internet disconnection and the unavailability of the server.

It forms a prototyping base that can easily be scaled to control more devices and further be enhanced by integration features such as voice commands, sensors, and energy monitoring. These are going to drive a much more integrated smart home experience with probable future opportunities unfolding for IoT-driven automation solutions. Conclusion In short, this project gives evidence of the ever-growing applicability of IoT in home automation, and thus presents a realistic example on how smart technology can make the complexities of everyday life easy.

REFERENCES

1. Vaidya, V. D., & Vishwakarma, P. (2023). A comparative analysis on smart home systems to control, monitor, and secure homes 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). AutoHome - 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.