**GANDAKI UNIVERSITY**

**Bachelor of Information Technology**



**A Project Report on**

**“IoT Based Smart Home Automation System”**

**SUBMITTED BY**

DHANU GURUNG (21021005)

PRABESH SUBEDI (21021016)

SRISTI LAMICHHANE (21021027)

SWASTIKA THAPA (21021030)

**SUPERVISOR**

Er. Saroj Giri

**SUBMITTED TO**

**BIT Program**

Rajchautara, Pokhara-32

2024

**GANDAKI UNIVERSITY**

**Bachelor of Information Technology**

**A Project Report on**

**“IoT Based Smart Home Automation System”**

**SUBMITTED BY**

DHANU GURUNG (21021005)

PRABESH SUBEDI (21021016)

SRISTI LAMICHHANE (21021027)

SWASTIKA THAPA (21021030)

**SUPERVISOR**

Er. Saroj Giri

**SUBMITTED TO**

BIT Program

Rajchautara, Pokhara-32

2024

Date: 26th Nov, 2024

**CERTIFICATE**

A Project Report

On

**“IoT Based Online Home Automation System”**

**Submitted by:**

Dhanu Gurung (021021005)

Prabesh Subedi (021021016)

Sristi Lamichhane (021021027)

Swastika Thapa (021021030)

Approved as to Feasibility, content and Style of Minor Project by Committee, BIT program, Gandaki University.

…..……………………

Er. Saroj Giri

Program Coordinator

………………………..

Er. Saroj Giri

Supervisor

……………………….

Er. Sandeep Gupta

ACKNOWLEDGEMENT

This project was done as per the requirement of 3rd year minor project in Bachelor of Information Technology, Gandaki University, Nepal. We would like to express our deepest gratitude to Er. Saroj Giri for supervising this project with full dedication, support and guidance. Without him, this project would take twice as long to complete.

We would like to thank the Department of Computer Science and Technology for carrying out a minor project that will help us shape our career among all the members of the department.

We would like to acknowledge all the research papers that helped us understand the required concepts as well as algorithm better, and that we utilized to prepare this project.

Every attempt has been made to include each and every aspect of the project in this report so that the reader can clearly understand our project. We would be pleased to get feedback on this project.

Sincerely,

Dhanu Gurung (021021005)

Prabesh Subedi (021021016)

Sristi Lamichhane (0210210027)

Swastika Thapa (021021030)

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **TITLE** | **PAGE NO.** |
|  | **Abstract** | **1** |
| **1.** | **INTRODUCTION** | **2-5** |
|  | 1.1 Background  1.1.1 Need for the New System  1.1.2 Objective of the Project  1.2 Detailed Problem Definition | 2  2  2  3 |
|  | 1.3 Viability of the System | 3 |
|  | 1.4 Presently Available System for the Same  1.5 Future Prospects  1.6 Organization of the Report | 4  4  4 |
| **2.** | **LITERATURE REVIEW**  2.1 Literature Review | **6**  6 |
| **3.** | **ANALYSIS** | **7-8** |
|  | 3.1 Project Management  3.1.1 Initial Investigation  3.1.2 Cost Benefit Analysis  3.2 Requirement Analysis  3.2.1 Hardware Requirements  3.2.2 Software Requirement  3.2.3 Connectivity Requirement  3.2.4 Power Supply Requirement | 7  7  7  7  8  8  8  8 |
| **4.** | **DESIGN** | **9-10** |
|  | 4.1. Software Requirements Specification  4.1.1 Functional Requirement  4.1.2 Non-functional Requirement  4.2 Risk Assessment | 9  9  9  10 |
| **5.** | **SYSTEM MODELING** | **11-12** |
|  | 5.1 Block Diagram | 11 |
|  | 5.2 Circuit Diagram | 12 |
| **6.** | **CODING**  6.1 Hardware Specification  6.2 Platform  6.2.1 Hardware Platform  6.2.2 Software Platform | **13-16**  13  13  13  13 |
|  | 6.3 Programming Language Uses  6.3.1 C Programming Language | 14  14 |
|  | 6.4 Software Tools Used  6.4.1 Arduino IDE  6.4.2 Blynk App | 14  14  14 |
|  | 6.5 Coding Style Followed | 14 |
| **7.** | **TESTING**  7.1 Testing  7.1.1 Formal Technical Review  7.1.2 Test Plan  7.1.3 Test Cases and Results | **16-17**  16  16  17  17 |
| **8.** | **LIMITATION**  8.1 Limitation | **18**  18 |
| **9.** | **CONCLUSION**  9.1 Conclusion | **19**  19 |
|  | **Bibliography** | **20** |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Figure** |  | **PAGE NO.** |
| 1. | Figure 5.1 | Block Diagram | 11 |
| 2. | Figure 5.2 | Circuit Diagram | 12 |

**ABSTRACT**

This project focuses on the development of an IoT-based smart home automation system utilizing the Blynk app and a local server. The primary goal is to enable users to remotely monitor and control household appliances through a smartphone. By integrating the Blynk app with a local server and an Arduino-based system, the project ensures real-time communication between devices and users.

Manual operation of home appliances is time-consuming and inconvenient, especially for the elderly or individuals with physical limitations. The developed system provides an efficient solution by allowing users to send ON/OFF commands to connected appliances via the Blynk app. These commands are processed through a local server, ensuring reliable and secure communication without the need for internet dependency.

In the implementation phase, the Arduino board acts as the core controller, receiving commands from the Blynk app and activating appliances through mechanical relays. Real-time updates and status monitoring are displayed within the app, enhancing user experience. This system currently offers a cost-effective, user-friendly approach to modern home automation while reducing energy consumption and improving convenience. It represents a significant step toward smarter and more sustainable living environments.

**Keywords: Home automation, Remotely Monitor, Arduino board, Blynk App, Real-time Update**

**CHAPTER-1**

**INTRODUCTION**

## 1.1 Background

A smart home automation system uses IoT technology to monitor, control, and automate home devices through a centralized system like a smartphone app or voice assistant. It enhances convenience, security, and energy efficiency by enabling devices to communicate and optimize their use.

For instance, users can remotely turn off lights or lock doors, while energy consumption is monitored for efficiency. As IoT advances, smart homes are becoming more affordable, accessible, and intelligent, adapting to user preferences to simplify daily life.

## 1.1.1 Need for the new system

In today’s fast-paced world, managing household tasks efficiently is crucial. Traditional systems often require manual control, which can be time-consuming and less efficient. An IoT-based smart home automation system addresses these challenges by offering centralized, remote control of home appliances and devices through smartphones.

This system enhances convenience, saves energy, and improves security. For example, users can remotely control lights, or security cameras, reducing energy waste and ensuring safety. It also provides real-time monitoring, offering peace of mind.

## With the growing adoption of smart devices, an IoT-based system seamlessly integrates them, making homes smarter, more efficient, and adaptable to modern lifestyles.

## 1.1.2 Objective of the project

The main objective is mentioned below:

To develop the Arduino based home appliance control using application like Blynk and local server.

# 1.2 Detailed Problem Study

The problem with traditional home systems was the lack of automation and control, especially when users were away from home. Conventional methods relied heavily on manual operations, which often led to inefficiencies, energy wastage, and limited security. For instance, lights or appliances could accidentally be left on, leading to higher electricity bills, or there was no way to monitor and control home security remotely.

Additionally, the absence of interconnected devices made it difficult to integrate various home functions, such as lighting, temperature control, and security systems, into a single, cohesive system. This created inconvenience and required users to manage multiple systems separately.

**1.3 Viability of the System**

The viability of the IoT-based smart home automation system is assessed based on its ability to enhance convenience, security, and energy efficiency. The system demonstrates its effectiveness in connecting and controlling various household devices through a centralized platform, such as a smartphone.

The system proves feasible due to the increasing affordability of IoT devices and widespread internet connectivity. Users find it easy to manage appliances remotely, monitor energy usage.

However, challenges such as data privacy, system vulnerabilities, and the need for technical maintenance are identified. Despite these issues, the system is considered a viable solution, with potential improvements ensuring broader adoption in the future.

**1.4 Presently Available Systems for the Same**

**i. Home Assistant:** An open-source platform that supports numerous smart home devices and operating systems by integrating with voice assistants like Alexa and Google Assistant.

### ii. IoT-Centric Smart Devices: These are specific devices designed to enhance smart home functionality:

* **Smart Thermostats (e.g., Nest, Ecobee):** Learn user behavior to adjust temperature automatically, improving comfort and energy efficiency.
* **Smart Locks (e.g., August Smart Lock Pro):** Enable keyless entry and provide remote locking/unlocking for enhanced home security.
* **Smart Security Cameras (e.g., Ring, Arlo):** Allow real-time monitoring and send mobile alerts for better safety.

# 1.5 Future Prospects

* Adding sensors like temperature, humidity, motion, and light can automate tasks such as adjusting lighting based on ambient conditions or triggering security alarms when motion is detected.
* Compatibility with voice assistants like Amazon Alexa or Google Assistant can provide hands-free control of appliances.
* Incorporating energy monitoring modules can track appliance power consumption, enabling users to optimize energy use and reduce electricity bills.

**1.6 Organization of the System**

The organization of the system in the document is structured into eight main sections. It begins with an Introduction, outlining the need for a new system, defining the problem in detail, evaluating the system's viability, reviewing existing systems, exploring future possibilities, and summarizing the report's structure. Literature Review, that evaluates and summarizes existing research on a specific topic. Next, the Analysis section focuses on project management and requirements analysis. The Design section emphasizes creating software requirements and assessing risks. System Modeling includes creating block diagrams and circuit diagrams. Coding details hardware specification, platforms, programming languages, software tools, and coding styles. Testing involves conducting formal reviews, developing test plans, and documenting test cases with results. Limitation refers A limitation refers to a restriction, drawback, or constraint that affects the extent, scope, or functionality of something. Conclusion summarizes the project's achievements, findings, and overall impact and finally Bibliography provides a list of references and resources consulted during the project.

**CHAPTER-2**

**LITERATURE REVIEW**

**2.1 Literature Review**

Studies such as those by Alkar and Buhur (2005) emphasized the importance of connectivity and interoperability in home automation systems. These studies influenced the decision to integrate the Blynk platform, which provided a user-friendly IoT solution for remote device control. Research by Jadhav et al. (2018) demonstrated the effectiveness of the ESP8266 in managing IoT devices due to its low power consumption and robust performance. The Blynk platform had been extensively reviewed as a versatile tool for IoT device management. Researchers like Sharma and Chaudhary (2019) evaluated Blynk's effectiveness in providing a real-time user interface for remote control and monitoring of devices. Rashid and Farooq (2016) discussed the use of relay modules for safely controlling high-voltage appliances through microcontrollers. The study provided best practices for interfacing relays and mitigating issues such as electrical interference. Patil et al. (2017) detailed the implementation of a lightweight web server on the ESP8266, enabling local control of connected devices. Their work emphasized the flexibility and reliability of using HTTP-based communication for smart home applications.

**CHAPTER-3**

**ANALYSIS**

# 3.1 Project Management

For our project management of the project, we have performed several steps which are suitable for its accomplishment that include the initial investigation, various analysis, design, implementation, evaluation and modification of the projects. Some of the early steps that are taken are mentioned in these chapters below, while the later steps are discussed in the following chapters.

**3.1.2 Initial Investigation**

The first step involved understanding the needs of the users and the features required for the IoT-based home automation system. This included researching available technologies, such as relays and microcontrollers like Arduino, and evaluating how they could be integrated to automate devices like lights, fans, and security systems.

**3.1.2 Cost Benefit Analysis**

The project’s cost involved expenses for hardware (Wi-Fi modules, NodeMCU), software (Blynk app subscription), and installation. The benefits included energy savings, enhanced convenience, and improved security. Over time, the system paid for itself by reducing energy bills and providing users with remote control of home devices, making it a worthwhile investment.

**3.2 Requirement Analysis**

**3.2.1 Hardware Requirements**

* **Wi-Fi Module (e.g., ESP8266):** This is used to connect the Arduino to the internet, allowing remote control via the Blynk app.
* **Relay Modules:** For switching devices like lights, fans, and appliances on and off. Relays act as the interface between the low-voltage Arduino and high-voltage appliances.

**3.2.2 Software Requirements**

* **Arduino IDE:** The development environment used to program the Arduino board. The code will include logic for interacting with relays, and Wi-Fi modules.
* **Blynk App (Mobile Application):** The app will allow users to monitor and control the smart home system remotely. The app interface is customizable, providing control over device, viewing data.
* **Blynk Library for Arduino:** This library will be integrated into the Arduino code to communicate with the Blynk app via the internet. It will send and receive data from the app, allowing users to interact with the system.

**3.2.3 Connectivity Requirements**

* **Internet Connection:** A stable internet connection is essential for remote control via the Blynk app.
* **Local Network:** The Arduino system should be able to communicate with the Wi-Fi module (e.g., ESP8266) to establish a local network for device control.

**3.2.4 Power Supply Requirements**

* **Arduino Board Power:** The Arduino board should be powered using a 12V DC adapter for development purposes.
* **Device Power Supply:** The appliances (lights, fans, etc.) will require an independent power supply, typically 110V or 220V, depending on the region.

**CHAPTER-4**

**DESIGN**

**4.1 Software Requirement Specification**

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfil all stakeholders needs. It is the description of a software system to be developed, laying out functional and non-functional requirement.

## 4.1.1 Functional Requirement

* The system must allow users to control appliances (e.g., lights, fans) via the Blynk app.
* The system must toggle relays to turn devices ON/OFF based on commands received from the app.
* The NodeMCU must connect to a Wi-Fi network for communication with the Blynk cloud server.
* The system must support control of at least four devices via GPIO pins.

## 4.1.2 Non-functional Requirement

* The system should process and execute commands from the Blynk app within 1-2 seconds.
* The system should maintain consistent operation for extended periods without crashes.
* The Blynk app interface must be user-friendly and intuitive for users with minimal technical knowledge.
* The NodeMCU and relays should consume minimal power during operation and standby modes.

**4.2 Risk Assessment**

* **Incorrect wiring:** Wiring the relays and NodeMCU incorrectly caused hardware malfunctions during initial testing.

**Solution:** The wiring was verified using datasheets and circuit diagrams.

* **Insufficient Power Supply:** The NodeMCU did not function reliably when powered through USB with multiple relays in operation.

**Solution:** A 12V regulated power adapter was used to provide adequate power for both the NodeMCU and the relay module.

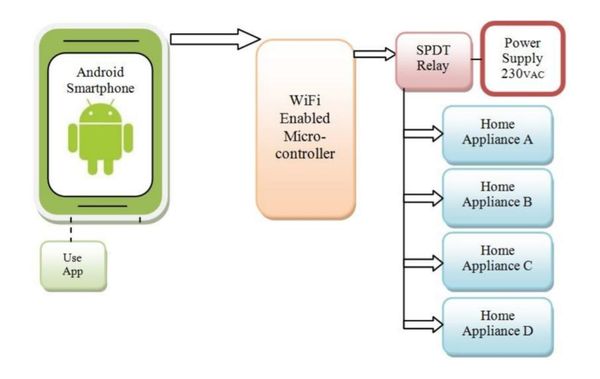
* **Difficulty Identify Issues:** Debugging connectivity and hardware issues was challenging without clear logs or indicators.

**Solution:** Used the Blynk terminal widget for remote debugging of the device.

**CHAPTER-5**

**SYSTEM MODELING**

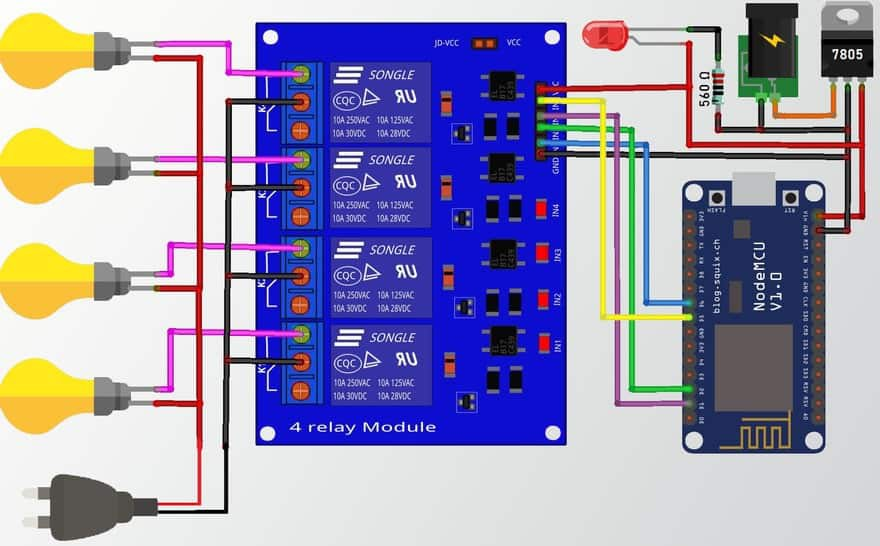
**5.1 Block Diagram**



**Figure 5.1: Block Diagram of the System**

The diagram shows a system for controlling home appliances using a smartphone. An Android app on the smartphone connects to a Wi-Fi-enabled microcontroller, which acts as the central unit. The microcontroller is linked to an SPDT (Single Pole Double Throw) relay that controls the flow of electricity from a 230V power supply to different home appliances (A, B, C, and D). Using the app, the user can remotely switch appliances on or off by sending commands to the microcontroller, which operates the relay to manage the appliances.

# 5.2 Circuit Diagram



**Figure 5.2.1: Circuit Diagram of System**

This diagram shows how to connect a NodeMCU microcontroller with a 4-channel relay module to control four electrical devices (like light bulbs). The NodeMCU provides signals to control the relays, which act as switches to turn the devices on or off. A 5V regulator (7805) is used to step down and stabilize the power supply for the circuit. The power source is connected to the relay module and NodeMCU through a combination of wires. The bulbs are connected to the relays using live and neutral wires, allowing them to be controlled via the NodeMCU (likely using a local web server or app). This setup can be used in IoT applications like smart home systems.

**CHAPTER-6**

**CODING**

# 6.1 Hardware Specification

The system was built using Node MCU (ESP8266), relays to control high-voltage appliances, and a Blynk mobile application for remote control. Below are the detailed hardware components:

**6.1.1 NodeMCU (ESO8266)**

* Microcontroller based on ESP8266 Wi-Fi chip.
* Provides Wi-Fi connectivity for remote control.

**6.1.2 Relay Module**

* Allows switching of high-voltage appliances using low-voltage signals from the Node MCU.

## 6.1.3 Bulb Holder

* Light bulb or any other appliance connected to the relay.

**6.2 Platform**

**6.2.1 Hardware Platform**

* **Node MCU (ESP8266):** This development board was ideal for IoT-based applications, providing built-in Wi-Fi for connectivity.

**6.2.2 Software Platform**

* **Arduino IDE** was used for coding the Node MCU.
* **Blynk App** was used to provide the mobile user interface to control the devices Remotely.

# 6.3 Programming Language Used

**6.3.1 C-Programming Language**

The system was developed using C with the Arduino IDE for Node MCU programming. The code uses the following:

* C programming was used for microcontroller programming to interact with relays, and Wi-Fi connectivity.
* Blynk API was used for integrating mobile application control of appliances.

# 6.4 Software Tools Used

# 6.4.1 Arduino IDE

* Used to write and upload C code to the Node MCU. The IDE also provides serial monitoring capabilities.
* **Libraries Used:** ESP8266WiFi.h: For Wi-Fi connection setup and BlynkSimpleEsp8266.h: For Blynk integration.

## 6.4.2 Blynk App

* Mobile application used to control the appliances remotely.
* Provides a graphical interface with buttons and sliders to interact with the Node MCU.
* Virtual Pin is used to control the relay.

# 6.5 Coding Style Followed

The coding style adopted for this project was structured and organized, prioritizing readability, maintainability, and functionality. The code was modularly divided into sections for Wi-Fi setup, Blynk integration, relay control, and local server creation. Descriptive naming conventions were used, with meaningful variable and function names that clearly indicated their purpose.

The code was optimized by eliminating redundancy and including only essential libraries to reduce memory usage. Debugging was made easier through serial monitoring, and sensitive information like Wi-Fi credentials and Blynk authentication tokens were segregated for clarity. The project adhered to Arduino standards for setup and loop structures, ensuring compatibility and predictable behaviour. Overall, this coding style resulted in a well-structured and robust implementation.

**CHAPTER-7**

**TESTING**

# 7.1 Testing

Testing of the application starts after the code has been generated. The testing logic concentrates on the internal logic of the software, ensuring that all the statements have been tested, and on the functional extends, testing to find errors and ensuring that defined input will give results that are accurate and in line with the desired results.

The testing of our project was done as follows:

**7.1.1 Formal Technical Review**

The goal of this formal technical review is to evaluate the system design, functionality, code quality, hardware-software integration, and overall performance of the Wi-Fi based home automation system with manual switch buttons using NodeMCU ESP8266 and Blynk app.

* **Wi-Fi Connectivity:** The Wi-Fi credentials were correctly configured, and the device successfully connected to the network without issues.
* **Blynk App Integration:** Communication with the Blynk app was verified to be seamless and functional, ensuring that all commands were transmitted correctly.
* **Code Quality:** The code was reviewed for efficiency, modularity, and readability, ensuring it was maintainable and scalable.
* **Error Handling:** Proper error handling was implemented for Wi-Fi connection issues and relay operations, ensuring stability during runtime.
* **Hardware-Software Interaction:** It was verified that the software successfully controlled the hardware components, including relays and connected devices.
* **User Interface Review:** The Blynk app interface was confirmed to function correctly, and commands sent from the app were successfully executed on the hardware.

**7.1.2 Test Plan**

* **Wi-Fi Connectivity Test:** Test if the Node MCU can successfully connect to the Wi-Fi network and verify by checking the IP address in the serial monitor.
* **Blynk App Control Test:** Use the Blynk app to control the relay by pressing the button widget and verify if the appliance responds correctly to the app commands.

**7.1.3 Test Case and Test Result**

* **Wi-Fi Connectivity Test:** Device connects to Wi-Fi with correct/incorrect credentials. Result: Connected successfully with correct credentials; retried with incorrect ones.
* **Blynk App Integration Test:** Toggle relays via Blynk app. Result: Relays toggled as expected.
* **Local Server Functionality Test:** Access and control relays via the server interface. Result: Webpage loaded, and controls worked.
* **User Interface Test:** Blynk app and web interface functionality. Result: Buttons worked, and states updated correctly.

**CHAPTER-8**

**LIMITATION**

**8.1. Limitation**

Limitation of the system are as follow:

* The system relied heavily on a stable Wi-Fi network.
* The system was designed for a limited number of devices (e.g., four relays).
* A failure in the microcontroller would render the entire system non-operational until repairs were made.

**CHAPTER-9**

**CONCLUSION**

# 9.1. Conclusion

The Wi-Fi-based home automation system was successfully implemented using NodeMCU ESP8266, the Blynk app, and a local web server. It enabled remote control of multiple appliances, offering convenience and cost-effectiveness. While the system was functional and reliable, limitations such as Wi-Fi dependence, limited range of operation, and security concerns highlighted areas for improvement. Overall, the project achieved its goals and provided a strong foundation for future advancements in smart home automation.

**Bibliography**

* Alkar, A. Z., &Buhur, U. (2005). An Internet-based wireless home automation system for multifunctional devices. IEEE Transactions on Consumer Electronics.
* Sharma, R., & Chaudhary, V. (2019). Blynk: IoT-based home automation system. International Journal of Emerging Technologies in Engineering Research.
* Rashid, M., & Farooq, U. (2016). Relay interfacing techniques for microcontroller-based systems. Proceedings of the International Conference on Embedded Systems and Applications.
* Patil, A., Shet, A., &Naik, M. (2017). Local web server for IoT applications using ESP8266. International Journal of Science and Research (IJSR).