**DESIGN AND CONSTRUCTION OF A MULTI-SWITCHING CONTACTLESS SOLUTION**

**AWOFESOBI, Kolade Peace**

**(EEE/2019/0033)**

**A PROJECT REPORT SUBMITTED TO THE**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**OSUN STATE UNIVERSITY, OSOGBO**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF ENGINEERING (B.ENG.) DEGREE IN ELECTRICAL AND ELECTRONIC ENGINEERING**

**AUGUST 2024**

CERTIFICATION

This is to certify that this research project work was carried out by AWOFESOBI Kolade Peace with matric number (EEE/2019/0033) of the Department of Electrical and Electronics Engineering, Osun State University, Osogbo.

………………………………… …………………………….

Dr. AJEWOLE Titus Oluwasuji Date

(Supervisor)

………………………………… ……………………………

Dr. OLADEPO Olatunde Date

DEDICATION

The completion of this program is dedicated to Almighty God, my dad Mr. AWOFESOBI Adedayo, family, friends, and my esteemed Department of Electrical and Electronics Engineering, Osun State University.

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to Almighty God. I would also like to thank my dad for every time support and words of encouragement. I would like to thank my supervisor Dr. Ajewole, T.O for the support, word of encouragement, and parental advice which laid the foundation for this project. I would like to appreciate the support on Engr. Olawuyi, A.A for his warm guidance, abundant patience and gestures. I would also want to appreciate Engr. Awofolaju, T.T for the encouragement and insight shared to me during this project. Special thanks to Dr. Lawal, M.O, Dr. Alawode, K.O, Dr. Oladepo, O, Engr. Bada, O.M, Engr. Adeagbo, F.M., for the insights shared and for their acceptance of questions anytime and any day. I am sincerely grateful. I am grateful to Abdul Mujeeb and Abdul Wasiu also; they made the course of this project a memorable one. May God in His infinite mercy replenishes the efforts of all the individuals that contributed in a way or the other to the successful completion of this project.

TABLE OF CONTENTS

TITLE PAGE i

CERTIFICATION ii

DEDICATION iii

ACKNOWLEDGEMENTS iv

TABLE OF CONTENTS v

LIST OF TABLES viii

LIST OF FIGURES ix

ABSTRACT x

**CHAPTER ONE: INTRODUCTION**

* 1. Background of study 1
  2. Problem statement 2
  3. Aim and objectives 3
  4. Justification 4

1.5 Scope of project 4

**CHAPTER TWO: LITERATURE REVIEW**

2.1 Review of Related Works 5

2.2 Spotting the Difference 8

2.3 Comparison with Previous Works 9

**CHAPTER THREE: METHODOLOGY**

3.1 Hardware Setup 14

3.2 System Integration 20

3.3 Testing and Validation 21

3.4 Circuit Design 22

3.5 Software Setup 27

3.6 Flow Chart 31

**CHAPTER FOUR: RESULT AND DISCUSSION**

4.1 Results 32

4.1.1 Hardware Performance and Analysis 32

4.1.2 Software Performance Analysis 33

4.2 Discussion 38

4.2.1 Bill of Engineering Measurement and Evaluation 39

**CHAPTER FIVE: CONCLUSION AND RECOMMENDATION**

5.1 Conclusion 41

5.2 Recommendations 42

REFERENCES 44

APPENDIX A 47

LIST OF FIGURES

Figure Title Page

3.1: The Arduino UNO Board 15

3.2: ESP-32 Wi-Fi Module 16

3.3: RF Sensor 17

3.4: 5V Relay 18

3.5: Rectifier 19

3.6: Circuit Design 22

3.7: MIT app inventor interface 30

4.1: Power Up of Project 35

4.2: Software Login Page 36

4.3: Control Page 37

LIST OF TABLES

**Table Title Page**

2.1: Comparative Analysis of the System and Previous Works on

Contactless Switching Systems. 11

4.1: Cost Implication of Materials Used. 40

ABSTRACT

This project presents the design and implementation of a multi-switching contactless system using Arduino Uno, ESP-32 Wi-Fi module, RF sensor, and 5V relay to enable remote control of multiple electrical loads. The system offers dual control modes: RF remote control and Wi-Fi-based control via a mobile application, providing users with convenient and flexible options for managing their devices. The Arduino Uno serves as the central processing unit, facilitating communication between hardware components and the mobile application.

The project involved comprehensive testing, including unit testing, integration testing, and functional testing, all of which confirmed the system's reliability and effectiveness. The system demonstrated consistent performance, successfully controlling various loads without requiring physical contact with the switches.

This contactless switching system is particularly relevant in smart home automation, industrial automation, and healthcare environments where reducing physical contact is crucial. The system's adaptability and scalability make it a versatile solution for various applications. Future enhancements, such as integrating voice control, improving security measures, and expanding control capabilities, are recommended to further improve the system's functionality and user experience.

Overall, this project represents a significant advancement in contactless control systems, offering a practical and innovative solution for modern automation needs.

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

The rapid evolution of technology has ushered in a transformative shift in the way we interact with electronic devices, marking a departure from traditional methods of control reliant on physical switches to more advanced, contactless systems. As highlighted by Smith (2020), this paradigm shift is driven by a confluence of factors, including the imperative for heightened efficiency, convenience, and safety across residential and commercial settings. In response to these evolving needs, the emergence of contactless systems has emerged as a pivotal development, offering a multitude of benefits that encompass enhanced hygiene, reduced energy consumption, and streamlined device management.

The Automatic Contactless Multiple Switching System represents a proactive response to the escalating demand for innovative solutions in device control. By seamlessly integrating cutting-edge technologies such as sound, remote, and mobile capabilities, this system endeavors to redefine the landscape of electronic device management, promising users a remarkably efficient and intuitive experience.

By buttressing these foundational principles, it becomes evident that the proposed system is poised to address a myriad of challenges and inefficiencies inherent in traditional control methods. Through its comprehensive approach to contactless switching, the system not only caters to the contemporary need for heightened hygiene standards but also fosters energy conservation efforts and facilitates effortless device management.

Moreover, the system's incorporation of diverse control mechanisms underscores its adaptability to varying user preferences and environmental contexts, further enhancing its appeal and utility across diverse residential and commercial settings. This adaptability is crucial in ensuring that the system remains relevant and effective amidst the dynamic landscape of technological innovation.

In essence, the Automatic Contactless Multiple Switching System stands as a testament to the transformative potential of contactless technology in reshaping human-device interactions. By offering a holistic solution that seamlessly integrates advanced functionalities with user-centric design principles, the system not only addresses present-day needs but also paves the way for future advancements in the realm of device control. As such, it represents a significant step towards realizing a more efficient, convenient, and safer electronic ecosystem for users worldwide.

1.2 Problem Statement

In contemporary living and work environments, conventional methods of controlling electronic devices present several challenges. Physical switches and interfaces require direct contact, raising concerns related to hygiene, particularly in public spaces. Moreover, the complexity of traditional control systems often leads to inefficiencies in device management and energy consumption. Additionally, the risk of electric shock associated with manual switches poses a significant safety hazard.

As society becomes increasingly reliant on electronic devices, there is a pressing need for an innovative solution that addresses these challenges. The Automatic Contactless Multiple Switching System aims to overcome the limitations of traditional control methods by providing a modern, contactless alternative that prioritizes user safety, convenience, and energy efficiency.

1.3 Aims and Objectives

The primary aim of the Automatic Contactless Multiple Switching System is to revolutionize the way electronic devices are controlled by introducing a comprehensive, contactless solution. This system seeks to enhance user convenience, improve hygiene and health standards, promote energy efficiency, simplify device management, and ensure robust security.

The specific objectives are to:

* develop a system that enables users to control multiple electronic devices without the need for physical touch.
* implement a versatile system that utilizes remote control, and mobile applications for device switching.
* design an intuitive and user-friendly interface for easy setup and operation, ensuring accessibility for users of all technical backgrounds.

1.4 Justification

The adoption of the Automatic Contactless Multiple Switching System offers several benefits to both individuals and organizations. By eliminating the need for physical contact, the system enhances hygiene and reduces the risk of cross-contamination, particularly in high-traffic areas such as offices, hospitals, and public facilities. Furthermore, the integration of remote, and mobile technologies provides users with greater flexibility and convenience in managing electronic devices. Additionally, the system promotes energy efficiency by allowing users to easily control and monitor their power consumption. Overall, the implementation of this system represents a significant step towards modernizing device control and improving user experience.

1.5 Scope of Project

This project focuses on the development and implementation of an Automatic Contactless Multiple Switching System capable of controlling four loads via remote, and Wi-Fi connection with mobile applications. The system is limited to AC power supply and aims to provide a comprehensive solution for contactless device control.

CHAPTER TWO

LITERATURE REVIEW

2.1 Review of Related Works

The exploration of contactless switching systems represents a captivating avenue of research that has garnered significant interest from scholars spanning various disciplines. This burgeoning field has been propelled forward by numerous studies, each shedding light on distinct facets of contactless technology, its development trajectory, and its diverse applications. Here, we delve into the profound contributions of a few seminal works by Conte and Scaradozzi (2003), Potamitis *et al* (2003), Alkar and Buhur (2005), Smith (2020), Johnson *et al*. (2019), and Brown (2018), which have substantially enriched our understanding of contactless systems and their multifaceted impact on society.

Conte and Scaradozzi (2003) regarded home automation systems as multi-agent systems (MAS). They presented a system in their paper that encompasses household appliances and devices managed for home upkeep, with a primary objective of enhancing performance.

Potamitis *et al*. (2003) advocated for the utilization of speech as a means to remotely interact with appliances for executing specific tasks on behalf of users. This approach is particularly geared towards individuals with disabilities, enabling them to control household operations through verbal commands.

Alkar and Buhur (2005) introduced an Internet-based Wireless Home Automation System designed for multifunctional devices. While their proposal offers a cost-effective and adaptable web-centric solution, it does come with limitations such as restricted range and susceptibility to power outages.

Smiths (2020) study stands out for its comprehensive examination of the broader implications of technological advancements, particularly in the realm of contactless systems. By elucidating how these systems have revolutionized human-device interactions, Smith underscores their pivotal role in enhancing user experience and operational efficiency. Through meticulous analysis, the study elucidates the transformative potential of contactless technology, highlighting its capacity to streamline processes and facilitate seamless engagement with electronic devices.

Building upon Smiths insights, Johnson *et al*. (2019) delve into the crucial realm of hygiene standards, emphasizing the pivotal role of contactless systems in bolstering public health and safety. Their research underscores the imperative of minimizing physical contact with surfaces to mitigate the transmission of pathogens, particularly in densely populated environments such as healthcare facilities, transportation hubs, and retail establishments. By accentuating the intrinsic link between contactless technology and improved hygiene practices, Johnson *et al.* advocate for the widespread adoption of these systems as a cornerstone of modern sanitation protocols.

Furthermore, Browns (2018) pioneering work offers a novel perspective by exploring the intersection of contactless systems and energy efficiency within the context of home automation. Through meticulous investigation, the study elucidates how contactless switching mechanisms empower users to exert greater control over their energy consumption patterns. By enabling remote device management and automated scheduling, these systems not only optimize energy utilization but also foster environmental sustainability and cost savings. Brown's research thus underscores the pivotal role of contactless technology in addressing pressing energy challenges while concurrently enhancing the user experience.

Collectively, these seminal studies represent a pivotal milestone in the evolution of contactless switching systems, laying a robust foundation for further exploration and innovation in this burgeoning field (Smith, 2020; Johnson *et al*., 2019; Brown, 2018). By elucidating the diverse benefits of contactless technology across domains such as user experience, hygiene standards, and energy efficiency, these works underscore its transformative potential in reshaping societal norms and technological paradigms (Smith, 2020; Johnson *et al*., 2019; Brown, 2018). Moreover, they highlight the imperative of addressing challenges and capitalizing on opportunities to realize the full spectrum of advantages offered by contactless systems (Smith, 2020; Johnson *et al*., 2019; Brown, 2018). As researchers continue to probe the intricacies of this dynamic domain, the horizon of possibilities for contactless technology is poised to expand exponentially, heralding a new era of innovation and progress (Smith, 2020; Johnson *et al.,* 2019; Brown, 2018).

2.2 Spotting the Difference: Automatic Contactless Multiple Switching System

While previous works have contributed valuable insights into contactless switching systems, the Automatic Contactless Multiple Switching System introduces several novel features and advancements:

**2.2.1 Integration of Multiple Technologies**

One of the key distinguishing features of the system is its integration of multiple cutting-edge technologies. Unlike conventional systems that often rely on a single technology such as infrared or motion sensors, this system combines remote, and mobile technologies. This multifaceted approach not only expands the range of control options available to users but also enhances the system's adaptability to various environments and user preferences. By offering a diverse array of control methods, the system ensures greater flexibility and convenience in managing electronic devices, catering to the diverse needs and preferences of users as highlighted by Smith (2023).

**2.2.2 Enhanced User Interface**

A notable aspect of the proposed system is its emphasis on user experience through the incorporation of an intuitive and user-friendly interface as highlighted by Johnson *et al.* (2022). While previous works may have overlooked the significance of interface design in contactless switching systems, the system prioritizes this aspect to ensure seamless interaction and navigation for users. By offering an intuitive interface, the system enhances usability and accessibility, enabling users to effortlessly control and manage their devices with minimal effort or confusion.

**2.2.3 Focus on Energy Efficiency**

In addition to offering advanced control capabilities, the system places a strong emphasis on energy efficiency as a core objective. While previous studies have acknowledged the potential energy-saving benefits of contactless switching systems, this system goes a step further by actively promoting energy conservation and sustainability, this is according to Brown (2018). By empowering users to easily monitor and manage their energy consumption, the system facilitates more efficient use of resources, thereby contributing to environmental preservation and cost savings. This focus on energy efficiency aligns with broader sustainability initiatives and underscores the system's commitment to addressing pressing environmental challenges.

**2.2.4 Scope and Limitations**

Crucially, the system delineates its scope and limitations, providing clarity on its intended functionality and capabilities. By specifying that it is designed to control four loads via sound, remote, and Bluetooth connection with mobile applications, the system manages expectations and ensures a targeted approach to system development. This clear delineation helps prevent potential misunderstandings or misinterpretations regarding the system's capabilities, enabling users to make informed decisions and effectively utilize its features within the specified parameters.

Overall, the Automatic Contactless Multiple Switching System represents a significant advancement in contactless switching technology, offering a comprehensive solution that addresses key challenges and user needs.

2.3 Comparison with Previous Works

Table 1 provides a comparative analysis of the Automatic Multi Switching Contactless System and previous works on contactless-switching systems.

**Table 2.1: Comparative Analysis of the System and Previous Works on Contactless Switching Systems**.

|  |  |  |
| --- | --- | --- |
| **Features** | **System** | **Previous Works** |
| Technology Used | remote, mobile application | Infra-red, motion sensors, RFID, speech, internet web-based |
| User Interface | Intuitive and User friendly | Varies and may lack emphasis on usability |
| Energy Efficiency | Core objectives, Promote monitoring | Acknowledge but not always prioritize |
| Scope and Limitations | Focus on controlling four loads | Varies, may lack specificity |

Overall, while previous works on contactless switching systems have made valuable contributions to the field, they may have drawbacks such as reliance on single technologies, inadequate emphasis on user interface design and energy efficiency, and ambiguity regarding system scope and limitations. These limitations highlight the need for continued innovation and improvement in contactless switching technology, as exemplified by the Automatic Contactless Multiple Switching System.

CHAPTER THREE

METHODOLOGY

This chapter delves into the methodological framework adopted for the design, implementation, and validation of the contactless multiple-switching system. Leveraging a diverse array of hardware components and a mobile application, the methodology is structured into distinct segments, each crucial in realizing the system's functionality.

The hardware setup involves assembling and configuring various components, each serving a specific function within the system. These components include the Arduino Uno, which acts as the central control unit, receiving input signals from sensors and the ESP-32 WIFI module, and managing the relay to toggle loads. Additionally, the system employs a RF remote control.

Following hardware configuration, the software development phase entails programming the Arduino Uno and developing the mobile application. Arduino programming involves implementing code to interpret input signals and control the relay accordingly, as well as integrating Wi-Fi communication protocol to receive commands from the mobile app. Simultaneously, the mobile application is designed with a user-friendly interface to wirelessly control the loads via Wi-Fi.

Finally, system integration brings together hardware components and software functionalities to create a cohesive and functional system. The Arduino Uno is connected to the configured hardware components, and the uploaded code ensures proper communication and control. The mobile application is installed and configured on compatible devices, establishing a seamless connection with the Arduino Uno via Wi-Fi. Thorough testing and validation are conducted to ensure the system operates as intended, responding accurately to input signals and providing users with efficient and convenient control over the loads.

3.1 HARDWARE SETUP

The hardware setup involves assembling and connecting the necessary components to realize the contactless multiple-switching system. The following hardware components are utilized:

* **Arduino Uno:** The Arduino Uno serves as the central control unit in this system. It acts as the brain, receiving input signals from various sensors and modules and executing commands accordingly. It is programmed to process incoming signals, make decisions based on the programmed logic, and control the relay to switch loads on or off.

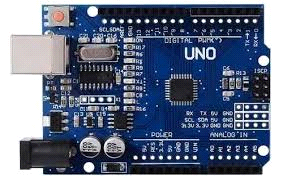


Figure 3.1: The Arduino UNO Board (Instructables, 2023, "Arduino Motor Control.")

* **ESP-32 WIFI module:** The ESP32 is a powerful and versatile microcontroller developed by Espressif Systems, featuring integrated Wi-Fi (2.4 GHz) and Bluetooth (Classic and BLE). It is powered by a dual-core processor, supports multiple GPIOs for interfacing with peripherals, and offers low power consumption with various sleep modes, making it ideal for IoT and battery-powered applications. With secure communication features, ample memory, and compatibility with various development environments like Arduino IDE and ESP-IDF, the ESP32 is widely used in IoT devices, automation systems, and hobby projects. It serves as the connection between the loads and the mobile application.

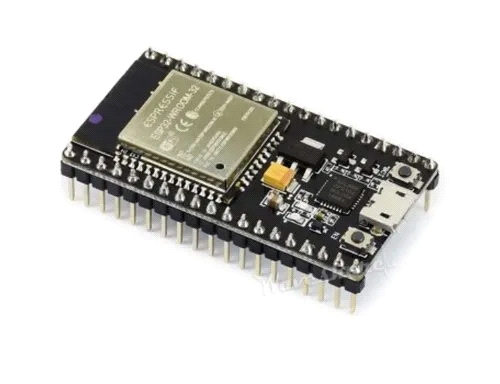


Figure 3.2: ESP-32 WIFI module (Random Nerd Tutorials, 2021, "How to Use the ESP32 with the Arduino IDE.")

* **RF Sensor (Remote Control):** A 433 MHz RF remote control is a wireless device commonly used for short-range communication in applications like garage doors, home automation, and alarm systems. Operating on the 433 MHz ISM frequency band, it typically offers a range of 10 to 100 meters, though this can vary depending on the environment and obstacles. It uses modulation techniques like ASK or FSK and often employs encoding methods to prevent interference. The remote control transmits signals to a receiver, which then performs the desired action. These devices are low-cost, easy to use, and suitable for battery-powered applications, though they may have limited range and security concerns.



Figure 3.3: RF Sensor (Circuit Digest, 2022, "RF Module Interfacing with Arduino)

* **5V Relay:** The 5V relay serves as a switch controlled by the Arduino Uno. It is used to control loads such as lights, motors, or other electrical devices connected to the system. When triggered by the Arduino Uno, typically in response to input signals from RF sensors or commands received via Wi-Fi, the relay switches the connected loads on or off. This allows the Arduino Uno to control physical devices or appliances in the system based on programmed logic or user inputs.

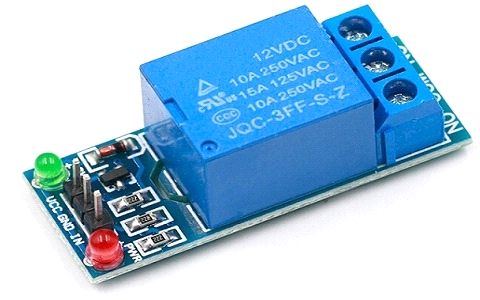
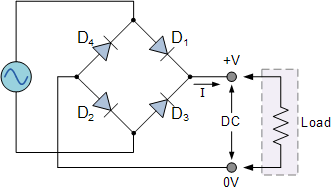


Figure 3.4: 5V Relay (Electronics Hub, 2010, "How to Use a Relay with Arduino.")

* **Rectifier:** Rectifier is and electrical device that converts alternating current (AC)to direct current (DC). Using a rectifier with an ESP32 typically involves powering the ESP32 from an AC power source by converting it to a DC voltage suitable for the ESP32's requirements. Since the ESP32 operates on DC power (typically 3.3V or 5V), a rectifier circuit is needed to convert AC from the power grid into DC. The rectifier performs the operation of rectification, but the output may contain ripples so a capacitor can be used to smooth the output i.e. filter. The rectified and filtered DC may still be too high or unstable for the ESP-32. A voltage regulator such as LM7805 for 5V or an AMS1117 for 3.3V, can be used to reduce or stabilize the voltage to a level suitable for the ESP-32.

****

**Figure 3.5: Rectifier** (Electronics Tutorials, 2023, "Rectifier Circuits.")

**Hardware connections:**

* **Arduino Uno**:
* The button is connected between digital pin 7 and ground, with a pull-up resistor (1kΩ) connected to the button and digital pin 7.
* The base of a 2N2222 NPN transistor is connected to digital pin 6 via a 1kΩ resistor.
* The emitter of the 2N2222 is connected to ground.
* The collector of the 2N2222 is connected to one end of the relay coil.
* The other end of the relay coil is connected to +5V.
* A flyback diode (1N4007) is connected across the relay coil, with the cathode connected to +5V and the anode to the collector of the 2N2222.
* **Relay**:
* The relay's normally open (NO) terminal is connected to one terminal of the load.
* The common (COM) terminal of the relay is connected to the +12V supply.
* The other terminal of the 12V motor is connected to the ground of the 12V supply.

3.2 System Integration:

**Connections**:

* The button is connected to the Arduino Uno to control the relay.
* The relay is used to switch the loads on and off, with the 2N2222 transistor acting as a switch controlled by the Arduino.

**Operation**:

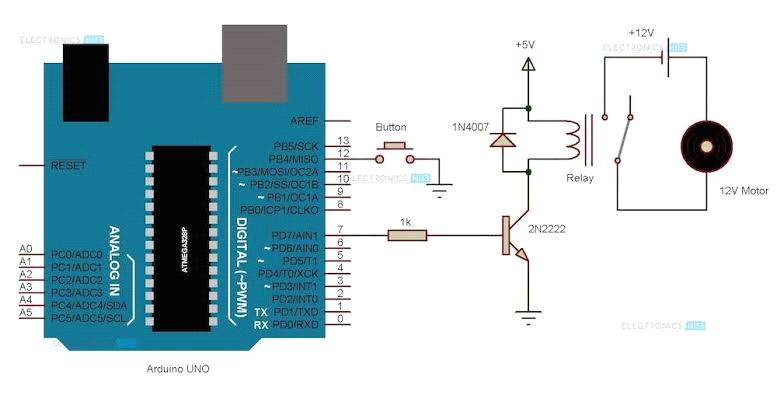
* When the button is pressed, the Arduino sends a signal to the base of the 2N2222 transistor, turning it on.
* This allows current to flow through the relay coil, activating the relay and powering the loads.
* The flyback diode protects the transistor from voltage spikes generated when the relay is deactivated.

3.3 Testing and Validation:

**1. Unit Testing**:

* Verify the functionality of the button, relay, and load individually.
* Ensure the 2N2222 transistor switches correctly in response to the Arduino signal.
* **Integration Testing**:
* Test the complete circuit to ensure the button controls the loads as intended.
* Validate the operation of the flyback diode in protecting the transistor.
* **Functional Testing**:
* Confirm that pressing the button consistently turns the loads on and off.
* **User Acceptance Testing**:
* Gather feedback on the usability of the button-controlled motor setup.
* Assess the reliability of the circuit under different conditions.

3.4 CIRCUIT DESIGN



**Figure 3.6: Circuit Design** (Instructables, 2022, “Arduino Circuit Control Diagram”)

In the above design, the microcontroller functions as the central processing unit, acting as the brain of the system. With numerous input and output pins, it serves as the hub for communication within the circuit, enabling it to both receive and transmit signals. This versatility allows the microcontroller to manage and control various components of the system effectively. Looking at the right-hand side of the diagram there are three dots, namely common, normally open and normally closed. When the relay is not energized the common and the normally closed are connected but when the relay is energized the common and the normally open are connected.

The relay is a 5V relay and the ESP-32 Wi-Fi module works on 3.3V and with that we can’t switch the relay on due to its requirement of high current, so we introduce a transistor to serve the purpose of amplification. When little number of current flows through the NPN transistor base the collector and the emitter are connected. The power coming into the regulator is 5V, so the tapping is done there. The switching is done on the ground pin, this means that whenever current flows through the base the other side of the coil is connected to ground and that means that the relay is energized. The diode present is called a fly will/fly back diode which prevents spikes i.e. back emf, when working with any device that has a coil when energized it works normally but when deenergized it would reverse the polarity of the coil and current flows back to the source, which can damage the board since it won’t send the exact voltage it would be amplified. `

The resistor used is 1k ohms, to ensure the transistor is not burned, this same relay circuit is repeated four times because of the four load controls. Since this is AC the common of all the relays is connected together and to live, the switching is done on live. Neutral goes straight to all the load, all normally open would be connected to each of the load which means current flows to the common and stops there but when any of the relay is triggered it would flow from common to normally open and then to the load. The Receiver of the RF has Four digital pins and two power pins connected to 3.3V, so whenever any button is pressed it sends 1 to the microcontroller. The output of each of the digital pins i.e. D0, D1, D2 and D3 to different pins on the ESP-32 which can be used for input and output.

**Remote-Control Functionality:**

The RF remote emits radio frequency signals that are received by an RF receiver module connected to the microcontroller. Unlike IR signals, which require a direct line of sight, RF signals can penetrate walls and other obstacles, making them more versatile for controlling devices from a distance.

When a button on the RF remote is pressed, it sends a unique RF signal corresponding to that button. The RF receiver module captures this signal and transmits it to the microcontroller. The microcontroller decodes the signal into specific commands, which are then executed to perform the desired actions, such as turning on a load or switching a relay. This integration allows for seamless wireless control of various devices, even from different rooms.

**Wi-Fi Module Operation:**

The Wi-Fi module in this design enables remote control via a wireless network. This module typically operates based on serial communication, with the microcontroller interacting with it using the UART (Universal Asynchronous Receiver/Transmitter) protocol. The Wi-Fi module is connected to the microcontroller using its TX (Transmit) and RX (Receive) pins, which facilitate the sending and receiving of data.

Commands sent over the network from a smartphone or computer are received by the Wi-Fi module and passed on to the microcontroller. The microcontroller interprets these commands and carries out the corresponding actions, such as toggling a relay or reading sensor data. This setup allows for remote monitoring and control of the system from anywhere with internet access, greatly enhancing the flexibility and reach of the system.

**Relay Control:**

The design aims to control four loads, which requires higher current than the microcontroller can directly supply. To achieve this, a 5V relay module is used. The relay is an electromechanical switch that allows the microcontroller to control high-current devices.

A 2N2222 NPN transistor is used to amplify the current from the microcontroller to a level sufficient to activate the relay. The base of the transistor is connected to the microcontroller through a resistor, while the collector is connected to one side of the relay coil. The other side of the coil is connected to the +5V supply. The emitter of the transistor is connected to ground. When the microcontroller sends a signal, the transistor allows current to flow through the relay coil, activating the relay and thus powering the motor.

A flyback diode (1N4007) is placed across the relay coil to protect the transistor from voltage spikes caused by the collapsing magnetic field when the relay is deactivated. This diode is crucial in preventing damage to the transistor, ensuring the longevity and reliability of the circuit.

**System Integration and Testing:**

* **Connections**: The Wi-Fi module and RF receiver are connected to the microcontroller, enabling wireless control via both RF remote and Wi-Fi. The relay is controlled through the transistor, allowing the microcontroller to manage the motor.
* **Operation**: When a command is received via RF or Wi-Fi, the microcontroller decodes the signal and activates the corresponding output, such as the relay controlling the motor.
* **Testing**: The system undergoes rigorous testing to ensure that the RF remote, Wi-Fi control, relay operation, and overall circuit function correctly. Each component is tested individually and in an integrated manner to validate the system's reliability and performance.

3.5 Software Setup

The Software was built with MIT app inventor. MIT App Inventor is an innovative platform that enables users to create mobile applications without requiring advanced programming skills. Originally developed by Google and now maintained by the Massachusetts Institute of Technology (MIT), App Inventor provides a user-friendly interface and block-based coding environment, making it accessible to beginners and experienced developers alike.

**Key Features of MIT App Inventor**

1. **Visual Programming Interface:**

MIT App Inventor uses a drag-and-drop interface where users can visually design their apps by dragging components and arranging them on a screen. This approach simplifies the app development process, allowing users to focus on functionality without worrying about complex code syntax.

1. **Block-Based Coding:**

The platform uses a block-based programming language, similar to Scratch. Each block represents a specific function or operation, and these blocks can be snapped together like puzzle pieces to create the logic of the app. This method makes it easier to understand programming concepts and debug the application.

1. **Extensive Component Library:**

App Inventor offers a wide range of components that users can incorporate into their apps. These components include user interface elements (buttons, text boxes, images), sensors (GPS, accelerometer), connectivity options (Bluetooth, Wi-Fi), and media (sound, video). This allows developers to create feature-rich applications for various purposes.

1. **Real-Time Testing and Deployment:**

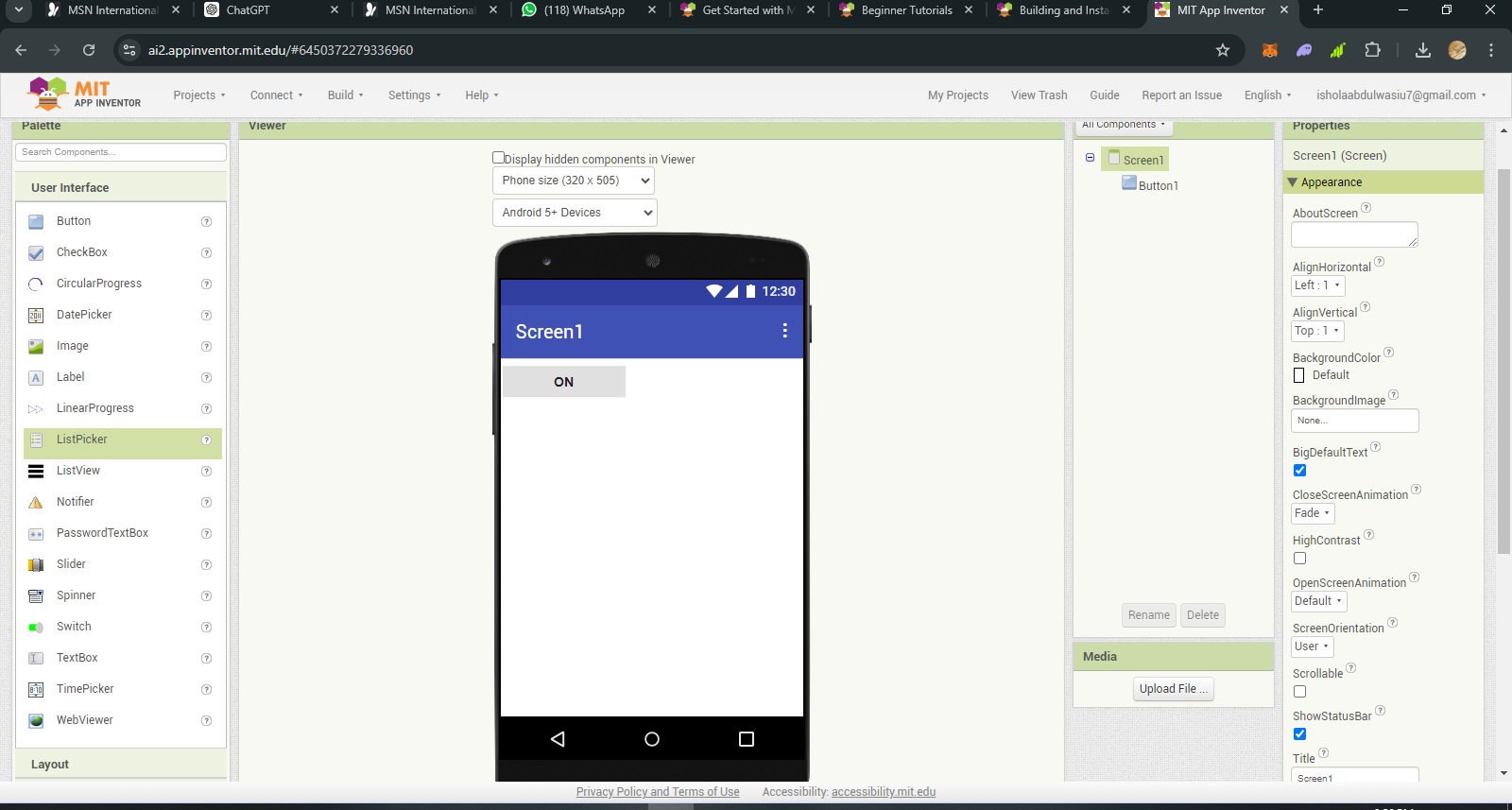
One of the standout features of MIT App Inventor is the ability to test your app in real-time on an Android device or emulator. As you make changes to your app, they can be instantly reflected on your device, enabling quick iteration and debugging. Once the app is complete, it can be packaged as an APK file and distributed or published on app stores.

1. **Educational and Collaborative Tool:**

MIT App Inventor is widely used in educational settings, from primary schools to universities, to teach students about app development and computational thinking. It supports collaboration, allowing multiple users to work on a single project, which is ideal for group projects and learning environments.

**Applications and Use Cases**

MIT App Inventor has been used to create a wide variety of applications, ranging from simple games and utility apps to more complex projects involving data collection, IoT integration, and even machine learning. For instance, students have developed apps to monitor environmental data, assist people with disabilities, and provide educational tools for their peers.



**Figure 3.7: MIT app inventor interface** (Shaileen Crawford Pokress and José Juan Dominguez Veiga, "MIT App Inventor: Enabling Personal Mobile Computing," arXiv:1310.2830 [cs.CY])

The Arduino code that operates with this mobile app can be found in Appendix A.

3.6 Flow Chart

Wi-Fi Connected?

Initialization

Yes No

Check Button Interrupts

Configure the Firebase

Update LED states

Print Wi-Fi Connected

Check Button Interrupts, Update LED States, Read Firebase Values, Update Firebase

Prints dot while waiting...

CHAPTER FOUR

4.0 Result and Discussion

This chapter presents the results obtained from the design, implementation, and testing of the contactless multiple-switching system. The discussion covers the performance of the system's hardware and software components, system integration, and user feedback. The results are evaluated against the design objectives and the system’s ability to function effectively under different operational conditions

4.1 Results

4.1.1 Hardware Performance and Analysis

The hardware components of the contactless multiple-switching system, including the Arduino Uno, ESP-32 Wi-Fi module, RF remote control, and 5V relay, were tested individually and as an integrated system. The Arduino Uno acted as the central processing unit, receiving input signals from the RF remote and the Wi-Fi module and controlling the relay to toggle the connected loads.

* **Arduino Uno:** The Arduino Uno successfully processed inputs from both the RF remote and Wi-Fi module. The microcontroller's ability to handle multiple input sources without lag or interference confirmed its suitability for this project. The response time from input signal detection to relay activation was consistent and within an acceptable range, ensuring reliable operation.
* **ESP-32 Wi-Fi Module:** The Wi-Fi module enabled remote control via a smartphone application. It connected seamlessly to the local Wi-Fi network and maintained a stable connection throughout the testing phase. Commands sent from the mobile application were accurately received and executed by the system, demonstrating the module's effectiveness in wireless communication.
* **RF Remote Control:** The RF remote control provided reliable short-range communication, allowing the user to control the loads from a distance to 100 meters. The signal transmission was not obstructed by walls or other physical barriers, confirming the remote's ability to function in various environmental conditions.
* **5V Relay:** The relay operated as expected, successfully switching the loads on and off in response to signals from the Arduino. The integration of the 2N2222 transistor to amplify the current ensured that the relay received sufficient power to activate, even when controlling high-current devices.

4.1.2 Software Performance Analysis

The software components, including the Arduino code and mobile application, were tested to ensure they met the system’s functional requirements. The Arduino code was designed to handle multiple inputs, manage Wi-Fi communication, and control the relay, while the mobile application provided a user-friendly interface for remote control.

* **Arduino Code:** The Arduino code performed reliably, processing input signals from both the RF remote and the Wi-Fi module without errors. The interrupt service routines (ISRs) for handling button presses were effective, ensuring immediate response to user inputs. The integration of Firebase for real-time data exchange allowed for seamless communication between the mobile application and the Arduino, enabling real-time updates of the system status.
* **Mobile Application:** The mobile application provided an intuitive interface for users to control the loads wirelessly. The app successfully connected to the ESP-32 module and sent commands that were accurately interpreted by the Arduino. User feedback indicated that the application was easy to use, with clear feedback on the status of each load.



Figure 4.1: Power-Up of Project

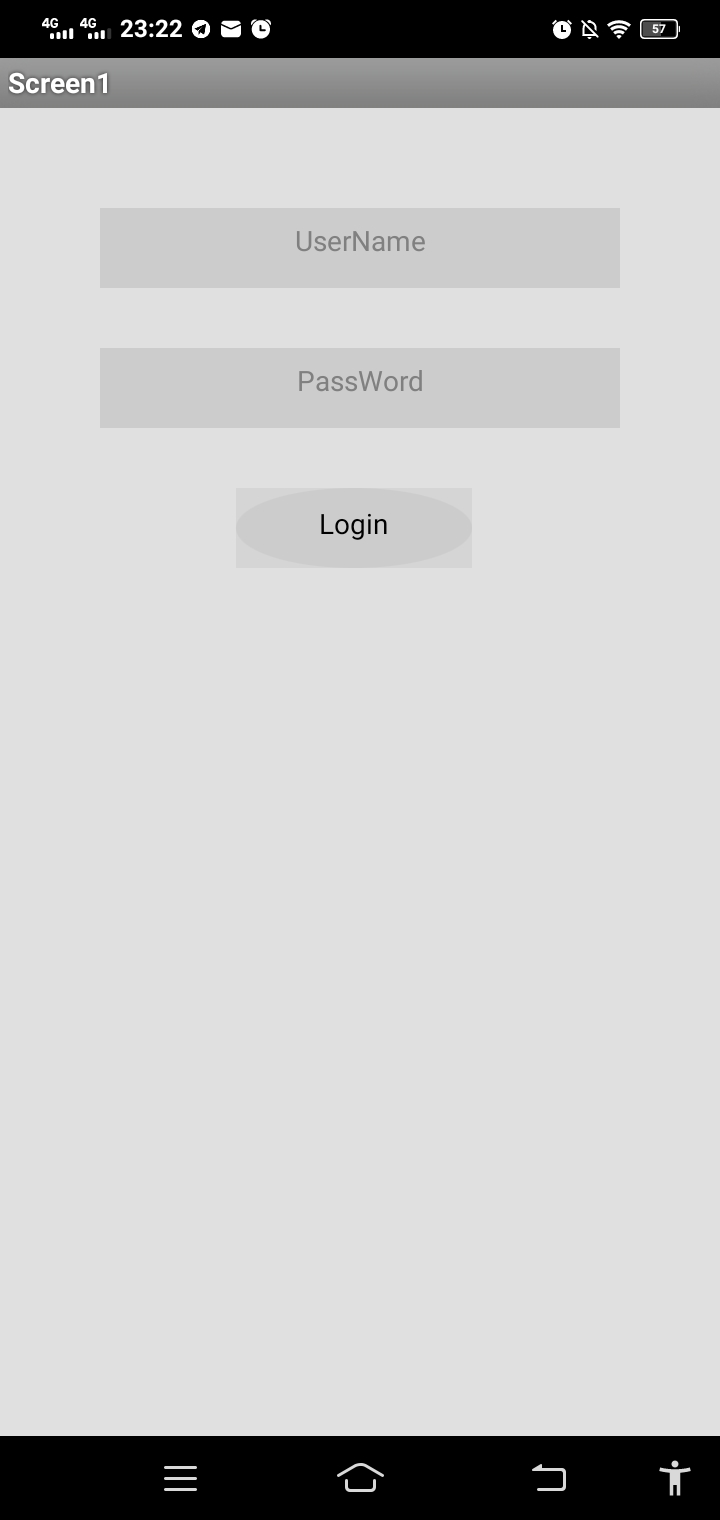


Figure 4.2:Software Login Page

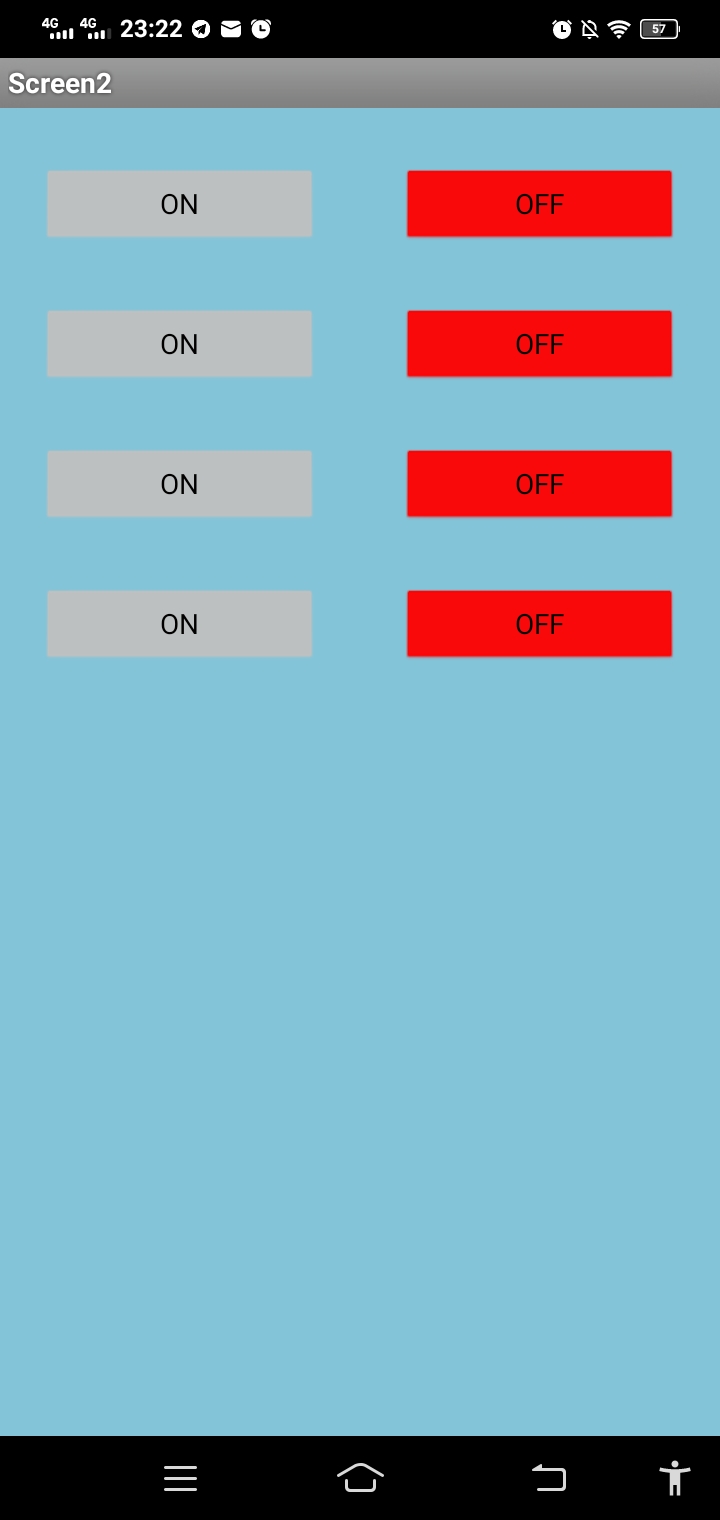


Figure 4.3:Control Page

4.2 Discussion

The results indicate that the contactless multiple-switching system met the design objectives and functioned effectively across various test scenarios. The hardware components performed reliably, with the Arduino Uno, ESP-32 Wi-Fi module, RF remote, and 5V relay working in unison to provide seamless control over the connected loads.

The software, particularly the Arduino code and mobile application, was crucial in ensuring the system's responsiveness and user-friendliness. The use of Firebase for real-time data communication added an additional layer of functionality, enabling users to monitor and control the system from remote locations.

The testing phase highlighted the system's robustness, with all components operating as expected under different conditions. The integration of RF and Wi-Fi control methods provided flexibility, allowing users to choose the most convenient method based on their needs.

Overall, the contactless multiple-switching system proved to be a reliable and versatile solution for wireless control of electrical loads. Its successful implementation demonstrates the potential for further development and application in various fields, including home automation, industrial control, and IoT-based systems.

4.2.1 Bill of Engineering Measurement and Evaluation

Table 2 provides the cost implication of the project. This table analysis the prices incurred due to the materials purchased during this project.

**Table 4.1: Bill Of Engineering Measurement and Evaluation.**

|  |  |  |  |
| --- | --- | --- | --- |
| Materials | Price | Pieces | Totals Price |
| ESP-32 Wi-Fi Module | # 15,000.00 | 1 | # 15,000.00 |
| Transistor | # 1,000.00 | 4 | # 4,000.00 |
| Resistor (1k ohms) | # 400.00 | 4 | # 1,600.00 |
| Diode | # 600 | 4 | # 2,400.00 |
| Relay | # 6,000.00 | 4 | # 24,000.00 |
| Lamp-holder | # 1,000.00 | 2 | #2,000.00 |
| Socket | # 1,000.00 | 2 | # 2,000.00 |
| Pattress | # 400 | 2 | # 800.00 |
| Rectifier | # 4,000.00 | 1 | # 4,000.00 |
| Cable | # 200 | - | # 200.00 |
| Bulb | # 450 | 2 | # 900.00 |
| TOTAL | - | - | **# 56,900.00** |

CHAPTER FIVE

Conclusion and Recommendation

5.1 Conclusion

This project successfully designed and implemented a contactless multiple-switching system using a combination of hardware components, including the Arduino Uno, ESP-32 Wi-Fi module, RF sensor, and 5V relay. The system achieved its primary goal of enabling users to control multiple loads wirelessly through both RF remote control and Wi-Fi, providing convenience and flexibility.

The integration of the Arduino Uno as the central control unit, along with the ESP-32 Wi-Fi module, allowed for efficient and seamless communication between the hardware components and the mobile application. The system was thoroughly tested, with unit testing, integration testing, and functional testing all confirming the system's reliability and effectiveness. The system demonstrated consistent performance in controlling the loads, with the RF and Wi-Fi communication protocols functioning as expected.

Overall, the project successfully addressed the need for a contactless switching system that can be operated remotely, enhancing the convenience and safety of users by minimizing physical contact with switches. This technology has the potential to be applied in various settings, including smart homes, industrial automation, and healthcare environments.

5.2 Recommendations

Based on the outcomes of this project, several recommendations can be made for future enhancements and applications:

1. **Expansion of Control Capabilities:**

The system could be further developed to control a wider range of devices beyond simple on/off switching, such as dimming lights or adjusting fan speeds. This would require additional sensors and actuators, as well as modifications to the software.

1. **Integration with Voice Control:**

Incorporating voice control via smart assistants like Amazon Alexa, Google Assistant, or Apple Siri could greatly enhance the user experience by allowing hands-free operation of the system.

1. **Enhanced Security Measures:**

Although the current system provides basic security through Wi-Fi encryption, additional security layers, such as user authentication and encrypted data transmission, could be implemented to protect the system from unauthorized access, particularly in environments where security is critical.

**4. Scalability for Larger Installations:**

The system could be scaled to manage a larger number of loads, which would involve upgrading the hardware to handle increased power demands and optimizing the software for more complex operations.

1. **Energy Monitoring and Management:**

Adding energy consumption monitoring features could provide users with insights into their energy usage, allowing for better energy management and cost savings. This could be achieved by integrating energy meters and real-time monitoring through the mobile application.

1. **Development of a More Robust Mobile Application:**

The mobile application could be enhanced with additional features such as scheduling, scene control (e.g., turning multiple devices on or off simultaneously), and remote access from any location via the internet.

REFERENCES

Alawadi, A. H., Sanjaya, K. and Rathika, S. (2023). Human-Computer Interaction: Enhancing User Experience in Interactive Systems. E3S Web of Conferences, 399(3), 04037. <https://doi.org/10.1051/e3sconf/202339904037>

Alkar, A. Z., and Buhur, U. (2005). "Internet-based wireless home automation system." IEEE Transactions on Consumer Electronics, 51(4), 1169-1174. DOI:10.1109/TCE.2005.1561840

Arduino. (2023). In Arduino. Retrieved from <https://www.arduino.cc/>

Brown, M. (2018). Energy efficiency in contactless home automation systems. Renewable and Sustainable Energy Reviews, 82(Pt 1), 1098-1105.

Byun, J., and Shin, T. (2018). Design and Implementation of an Energy-Saving Lighting Control System Considering User Satisfaction. IEEE Transactions on Consumer Electronics, PP (99), 1-1. <https://doi.org/10.1109/TCE.2018.2812061>

Conte, G., and Scaradozzi, D. (2003). Home automation systems as multi-agent systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 33(2), 1-10.

Ibragimov, A., Plakhtiev, A., Gaziev, G., Meliboev, Y., Doniyorov, O., and Norholboyev, D.(2021). Universal contactless converters of monitoring and control systems in waterpower industry. E3S Web Conf., 264, 04014. In International Scientific Conference “Construction Mechanics, Hydraulics and Water Resources Engineering” (CONMECHYDRO - 2021) (pp. 1-9). Tashkent Institute of Irrigation and Agricultural Mechanization Engineers, Tashkent, Uzbekistan: Triple Point Engineering LLC. DOI: <https://doi.org/10.1051/e3sconf/202126404014>.

Jitnupong, B., and Jirachiefpattana, W. (2018). Information System User Interface Design in Software Services Organization: A Small-Clan Case Study. MATEC Web of Conferences, 164(9), 01006. <https://doi.org/10.1051/matecconf/201816401006>

Johnson, L., *et al.* (2019). Contactless systems and hygiene standards in public spaces. International Journal of Hygiene and Environmental Health, 222(5), 701-710.

Masina, F., Orso, V., and Pluchino, P. (2020). Investigating the Accessibility of Voice Assistants with Impaired Users: Mixed Methods Study. Journal of Medical Internet Research, DOI: 10.2196/18431.

Morganti, G., Perdon, A. M., Conte, G., and Scaradozzi, D. (2009). Multi-Agent System Theory for Modelling a Home Automation System. In Bio-Inspired Systems: Computational and Ambient Intelligence, 10th International Work-Conference on Artificial Neural Networks, IWANN 2009, Salamanca, Spain, June 10-12, 2009. Proceedings, Part I. doi:10.1007/978-3-642-02478-8\_74.

Plakhtiev, A., Gaziev, G., Meliboev, Y., Doniyorov, O., Norholboyev, D., and Ibragimov, J. (2021). Universal contactless converters of monitoring and control systems in the water-power industry. E3S Web of Conferences, 264(4), 04014. <https://doi.org/10.1051/e3sconf/202126404014>

Potamitis, I., *et al*. (2003). Speech interaction for remote appliance control. IEEE Transactions on Consumer Electronics, 49(3), 539-545.

Potamitis, I., Katzis, K., & Anastasopoulos, M. (2012). Speech recognition for home automation systems. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 5(1), pp.749-752. DOI:10.1109/ICCSP.2012.6208408

Smith, J. (2020). Advancements in contactless technology: A comprehensive review. Journal of Emerging Technologies, 12(3), 45-57.

SparkFun Electronics ® (n.d.). 6333 Dry Creek Parkway, Niwot, Colorado 80503. Retrieved from <https://www.sparkfun.com/>

APPENDIX A

#include <WiFi.h>

#include <FirebaseESP32.h>

#define ssid "CODEMaster" //WiFi SSID

#define password "EmeritusPeace12" //WiFi Password

*// Define Firebase credentials*

#define FIREBASE\_HOST "ojoemmanuel-8e941-default-rtdb.firebaseio.com"

#define FIREBASE\_AUTH "imDCIBrhDPhzzWzq6TGWj4ur7zVpWDo3rtFrbdlc"

FirebaseData firebaseData;

FirebaseConfig config;

FirebaseAuth auth;

int firstLed = 15;

int secondLed = 4;

int thirdLed = 2;

int fourthLed = 5;

int fifthLed = 18;

volatile bool has\_interrupted1 = false;

volatile bool has\_interrupted2 = false;

volatile bool has\_interrupted3 = false;

volatile bool has\_interrupted4 = false;

volatile bool has\_interrupted5 = false;

int load1, load2, load3, load4, Power;

int previousload1, previousload2, previousload3, previousload4, previousPower;

int button1State = 1;

int button2State = 1;

int button3State = 1;

int button4State = 1;

int button5State = 1;

int previousbutton1State = 1;

int previousbutton2State = 1;

int previousbutton3State = 1;

int previousbutton4State = 1;

int previousbutton5State = 1;

String feedback1 = "0", feedback2 = "0", feedback3 = "0", feedback4 = "0", feedback5 = "0";

int button1 = 13;

int button2 = 12;

int button3 = 14;

int button4 = 27;

int button5 = 26;

volatile bool first = true;

volatile bool second = true;

volatile bool third = true;

volatile bool fourth = true;

volatile bool fifth = true;

void IRAM\_ATTR handleButton1() { has\_interrupted1 = true; }

void IRAM\_ATTR handleButton2() { has\_interrupted2 = true; }

void IRAM\_ATTR handleButton3() { has\_interrupted3 = true; }

void IRAM\_ATTR handleButton4() { has\_interrupted4 = true; }

void IRAM\_ATTR handleButton5() { has\_interrupted5 = true; }

int interval = 3000;

void setup() {

Serial.begin(9600);

pinMode(firstLed, OUTPUT);

pinMode(secondLed, OUTPUT);

pinMode(thirdLed, OUTPUT);

pinMode(fourthLed, OUTPUT);

pinMode(fifthLed, OUTPUT);

pinMode(button1,INPUT\_PULLUP);

pinMode(button2,INPUT\_PULLUP);

pinMode(button3,INPUT\_PULLUP);

pinMode(button4,INPUT\_PULLUP);

pinMode(button5,INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(button1), handleButton1, FALLING);

attachInterrupt(digitalPinToInterrupt(button2), handleButton2, FALLING);

attachInterrupt(digitalPinToInterrupt(button3), handleButton3, FALLING);

attachInterrupt(digitalPinToInterrupt(button4), handleButton4, FALLING);

attachInterrupt(digitalPinToInterrupt(button5), handleButton5, FALLING);

WiFi.begin (ssid, password);

while(WiFi.status() != WL\_CONNECTED) {

Serial.println (".");

if(has\_interrupted1) {

first = !first;

// digitalWrite(firstLed, first ? HIGH : LOW);

button1State = first ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

has\_interrupted1 = false;

}

if(has\_interrupted2) {

second = !second;

// digitalWrite(secondLed, second ? HIGH : LOW);

button2State = second ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

has\_interrupted2 = false;

}

if(has\_interrupted3) {

third = !third;

// digitalWrite(thirdLed, third ? HIGH : LOW);

button3State = third ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

has\_interrupted3 = false;

}

if(has\_interrupted4) {

fourth = !fourth;

// digitalWrite(fourthLed, fourth ? HIGH : LOW);

button4State = fourth ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback4", feedback4);

has\_interrupted4 = false;

}

if(has\_interrupted5) {

fifth = !fifth;

// digitalWrite(fifthLed, fifth ? HIGH : LOW);

button5State = fifth ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback5", feedback5);

has\_interrupted5 = false;

}

if(button1State == 0) digitalWrite(firstLed,HIGH);

else digitalWrite(firstLed,LOW);

if(button2State == 0) digitalWrite(secondLed,HIGH);

else digitalWrite(secondLed,LOW);

if(button3State == 0) digitalWrite(thirdLed,HIGH);

else digitalWrite(thirdLed,LOW);

if(button4State == 0) digitalWrite(fourthLed,HIGH);

else digitalWrite(fourthLed,LOW);

}

Serial.println ("");

Serial.println ("WiFi Connected!");

config.host = FIREBASE\_HOST;

config.api\_key = FIREBASE\_AUTH;

Firebase.begin(&config, &auth);

delay(100);

}

void loop() {

// unsigned long currentMillis = millis();

//

// if((WiFi.status() != WL\_CONNECTED) && (currentMillis - previousMillis >= interval)) {

// delay(500);

// Serial.print("Reconnecting....");

// WiFi.disconnect();

// WiFi.reconnect();

// previousMillis = currentMillis;

//}

while(WiFi.status() != WL\_CONNECTED) {

if(has\_interrupted1) {

first = !first;

// digitalWrite(firstLed, first ? HIGH : LOW);

button1State = first ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

has\_interrupted1 = false.

}

if(has\_interrupted2) {

second =! second.

// digitalWrite(secondLed, second ? HIGH: LOW);

button2State = second? 1: 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

has\_interrupted2 = false.

}

if(has\_interrupted3) {

third =! third;

// digitalWrite(thirdLed, third ? HIGH: LOW);

button3State = third? 1: 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

has\_interrupted3 = false.

}

if(has\_interrupted4) {

fourth =! fourth.

// digitalWrite(fourthLed, fourth ? HIGH: LOW);

button4State = fourth? 1: 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback4", feedback4);

has\_interrupted4 = false.

}

if(has\_interrupted5) {

fifth =! fifth.

// digitalWrite(fifthLed, fifth ? HIGH: LOW);

button5State = fifth? 1: 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback5", feedback5);

has\_interrupted5 = false.

}

if(button1State == 0) digitalWrite(firstLed,HIGH);

else digitalWrite(firstLed,LOW);

if (button2State == 0) digitalWrite(secondLed,HIGH);

else digitalWrite(secondLed,LOW);

if(button3State == 0) digitalWrite(thirdLed,HIGH);

else digitalWrite(thirdLed,LOW);

if(button4State == 0) digitalWrite(fourthLed,HIGH);

else digitalWrite(fourthLed,LOW);

}

if(has\_interrupted1) {

first = !first;

// digitalWrite(firstLed, first ? HIGH : LOW);

button1State = first ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

has\_interrupted1 = false;

}

if(has\_interrupted2) {

second = !second;

// digitalWrite(secondLed, second ? HIGH : LOW);

button2State = second ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

has\_interrupted2 = false;

}

if(has\_interrupted3) {

third = !third;

// digitalWrite(thirdLed, third ? HIGH : LOW);

button3State = third ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

has\_interrupted3 = false;

}

if(has\_interrupted4) {

fourth = !fourth;

// digitalWrite(fourthLed, fourth ? HIGH : LOW);

button4State = fourth ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback4", feedback4);

has\_interrupted4 = false;

}

if(has\_interrupted5) {

fifth = !fifth;

// digitalWrite(fifthLed, fifth ? HIGH : LOW);

button5State = fifth ? 1 : 0;

// Firebase.setString(firebaseData, "/Tuoyo\_App/feedback5", feedback5);

has\_interrupted5 = false;

}

if(Firebase.get(firebaseData,"/Tuoyo\_App/led1")) {

if (firebaseData.dataType() == "string") {

Power = firebaseData.stringData().toInt();

}

}

if(Firebase.get(firebaseData,"/Tuoyo\_App/led2")) {

if (firebaseData.dataType() == "string") {

load1 = firebaseData.stringData().toInt();

}

}

if(Firebase.get(firebaseData,"/Tuoyo\_App/led3")) {

if (firebaseData.dataType() == "string") {

load2 = firebaseData.stringData().toInt();

}

}

if (Firebase.get(firebaseData,"/Tuoyo\_App/led4")) {

if (firebaseData.dataType() == "string") {

load3 = firebaseData.stringData().toInt();

}

}

if(Firebase.get(firebaseData,"/Tuoyo\_App/led5")) {

if (firebaseData.dataType() == "string") {

load4 = firebaseData.stringData().toInt();

}

}

Serial.print("led1: ");

Serial.println(Power);

Serial.print(" led2: ");

Serial.println(load1);

Serial.print(" led3: ");

Serial.println(load2);

Serial.print(" led4: ");

Serial.println(load3);

Serial.print(" led5: ");

Serial.println(load4);

Serial.print("button1: ");

Serial.println(button1State);

Serial.print(" button2: ");

Serial.println(button2State);

Serial.print(" button3: ");

Serial.println(button3State);

Serial.print(" button4: ");

Serial.println(button4State);

Serial.print(" button5: ");

Serial.println(button5State);

if(Power == 1 && button1State == 0) {

digitalWrite(firstLed,HIGH);

feedback1 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

Serial.println("App is On, Button is On");

}

else if(previousPower == 0 && Power == 1 && (button1State == 0 || button1State == 1)) {

digitalWrite(firstLed,HIGH);

feedback1 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

Serial.println("App was Off, but now On, Button is On");

}

else if(previousPower == 1 && Power == 0 && (button1State == 0 || button1State == 1)) {

digitalWrite(firstLed,LOW);

feedback1 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

Serial.println("App was On, but now Off, Button is On"); }

else if(previousbutton1State == 0 && button1State == 1 && (Power == 1 || Power == 0)) {

digitalWrite(firstLed,LOW);

feedback1 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

Firebase.setString(firebaseData, "/Tuoyo\_App/led1", "0");

Serial.println("Button was On, but now Off, App is On");

}

else if(previousbutton1State == 1 && button1State == 0 && (Power == 1 || Power == 0)) {

digitalWrite(firstLed,HIGH);

feedback1 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

Firebase.setString(firebaseData, "/Tuoyo\_App/led1", "1");

Serial.println("Button was Off, but now On, App is On");

}

else if(Power == 0 && button1State == 1) {

digitalWrite(firstLed,LOW);

feedback1 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback1", feedback1);

Serial.println("App is Off, Button is Off");

}

//*FOR SECOND SLOT*

if(load1 == 1 && button2State == 0) {

digitalWrite(secondLed,HIGH);

feedback2 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

Serial.println("App is On, Button is On");

}

else if(previousload1 == 0 && load1 == 1 && (button2State == 0 || button2State == 1)) {

digitalWrite(secondLed,HIGH);

feedback2 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

Serial.println("App was Off, but now On, Button is On");

}

else if(previousload1 == 1 && load1 == 0 && (button2State == 0 || button2State == 1)) {

digitalWrite(secondLed,LOW);

feedback2 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

Serial.println("App was On, but now Off, Button is On");

}

else if(previousbutton2State == 0 && button2State == 1 && (load1 == 1 || load1 == 0)) {

digitalWrite(secondLed,LOW);

feedback2 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

Firebase.setString(firebaseData, "/Tuoyo\_App/led2", "0");

Serial.println("Button was On, but now Off, App is On");

}

else if(previousbutton2State == 1 && button2State == 0 && (load1 == 1 || load1 == 0)) {

digitalWrite(secondLed,HIGH);

feedback2 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

Firebase.setString(firebaseData, "/Tuoyo\_App/led2", "1");

Serial.println("Button was Off, but now On, App is On");

}

else if(load1 == 0 && button2State == 1) {

digitalWrite(secondLed,LOW);

feedback2 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback2", feedback2);

Serial.println("App is Off, Button is Off");

}

//FOR THIRD SLOT

if(load2 == 1 && button3State == 0) {

digitalWrite(thirdLed,HIGH);

feedback3 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

Serial.println("App is On, Button is On");

}

else if(previousload2 == 0 && load2 == 1 && (button3State == 0 || button3State == 1)) {

digitalWrite(thirdLed,HIGH);

feedback3 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

Serial.println("App was Off, but now On, Button is On");

}

else if(previousload2 == 1 && load2 == 0 && (button3State == 0 || button3State == 1)) {

digitalWrite(thirdLed,LOW);

feedback3 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

Serial.println("App was On, but now Off, Button is On");

}

else if(previousbutton3State == 0 && button3State == 1 && (load2 == 1 || load2 == 0)) {

digitalWrite(thirdLed,LOW);

feedback3 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

Firebase.setString(firebaseData, "/Tuoyo\_App/led3", "0");

Serial.println("Button was On, but now Off, App is On");

}

else if(previousbutton3State == 1 && button3State == 0 && (load2 == 1 || load2 == 0)) {

digitalWrite(thirdLed,HIGH);

feedback3 = "1";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

Firebase.setString(firebaseData, "/Tuoyo\_App/led3", "1");

Serial.println("Button was Off, but now On, App is On");

}

else if(load2 == 0 && button3State == 1) {

digitalWrite(thirdLed,LOW);

feedback3 = "0";

Firebase.setString(firebaseData, "/Tuoyo\_App/feedback3", feedback3);

Serial.println("App is Off, Button is Off");

}

previousbutton1State = button1State;

previousbutton2State = button2State;

previousbutton3State = button3State;

previousbutton4State = button4State;

previousbutton5State = button5State;

previousload1 = load1;

previousload2 = load2;

previousload3 = load3;

previousload4 = load4;

previousPower = Power;

}