INTERNET OF THINGS (IoT) BASED VOICE CONTROLLED FREQUENCY MODULATED RADIO RECEIVER

BY

ASHINZE IFECHUKWUDE ANTHONY

17/EEN/027

SUBMITTED TO

DEPARTMENT OF ELECTRICAL/ELECTRONIC ENGINEERING,

FACULTY OF ENGINEERING,

CROSS RIVER UNIVERSITY OF TECHNOLOGY (CRUTECH), CALABAR.

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF ENGINEERING (B.ENG.) IN ELECTRICAL/ELECTRONIC ENGINEERING.

NOVEMBER, 2023.

CERTIFICATION

This is to certify that this project titled "Internet of Things (IoT) Based Voice Controlled FM Radio Using Arduino and Google Assistant" was written by **ASHINNZE IFECHUKWUDE ANTHONY (17/EEN/027)** in partial fulfillment of the requirement for the award of Bachelor of Engineering (B.Eng.) in electrical/electronic engineering.

Engr. E. O. Onoyom-Ita **Signature______________**
(Supervisor) **Date_________________**

Engr. S. A. Ojomu **Signature_____________**
(Head of Department) **Date________________**

DEDICATION

I dedicate this work to my loving family, whose unwavering support and encouragement has been the cornerstone of my academic journey. Your belief in my abilities and your constant encouragement has given me the strength and motivation to pursue excellence. I am forever grateful for the sacrifices you have made and the countless assets you have invested in my education.

I would also like to extend my heartfelt appreciation to my dedicated supervisor, whose guidance and expertise have shaped this academic endeavor. Your patience, insights, and mentorship have been instrumental in shaping my research and pushing me to explore new boundaries.

Lastly, I would like to express my gratitude to the countless individuals who have contributed to this field of study through their research, publications, and insights. Your dedication to advancing knowledge has paved the way for my own intellectual growth, and I am honored to be a part of this scholarly community.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to God, and appreciation to all those who have contributed to the completion of this academic work. Their support, guidance, and encouragement have been invaluable in shaping the outcome of this research.

First and foremost, I would like to extend my deepest appreciation to my supervisor, Engr. E. Onoyom-Ita, for his guidance and mentorship throughout this journey. His expertise, insightful feedback, and constructive criticism have played a pivotal role in refining the research and pushing me to new heights.

My gratitude extends to my family and friends, whose unwavering support, encouragement, and understanding have been a constant source of strength throughout this academic endeavor. Their belief in my abilities and their unconditional love has provided the motivation and inspiration to persevere during challenging times.

Lastly, I would like to acknowledge the esteemed lecturers of Electrical/Electronic engineering department, and other academia whose work has laid the foundation for this study.

ABSTRACT

The Internet of Things (IoT) has revolutionized the way we interact with everyday objects. This project presents the design and implementation of an IoT-based voice-controlled mechanism for an FM radio receiver. The proposed system integrates the hardware and software components that enable the users to interact with the radio receiver using voice commands. The system consists of an Arduino microcontroller board, FM receiver module, a speaker, and a Wi-Fi module for connecting to the internet. Google Assistant is used as a voice assistant to control the radio receiver through voice commands. The voice commands are processed through the Google Assistant API, which sends the commands to the Arduino board. The designed system provides various features such as volume control, station selection, and tuning. The users can control the radio using simple voice commands, such as "Hey Google, activate turn the radio on," "Hey Google, activate set the radio station to Hit FM," and "Hey Google, activate set the radio volume to 90." The result shows that the proposed system is effective and efficient in controlling the radio receivers' volume and channels using voice commands. The system provides a hands-free and intuitive way of controlling the radio, which is especially useful for people with physical disabilities. This project contributes to the IoT-based radio industry by providing a practical solution for voice-controlled FM radios using Arduino and Google Assistant. The proposed system can be extended to support other voice assistants and can be applied in various IoT-based applications.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1. Background of study

The Internet of Things (IoT) has emerged as a revolutionary concept that connects various physical objects to the internet, allowing them to communicate with each other and with humans [1]. Voice-controlled systems, on the other hand, have gained immense popularity due to their ease of use and hands-free operation [2]. FM radio, a well-established technology, has been widely used for entertainment and information dissemination for decades [3]. Combining these three technologies can create a new generation of smart and user-friendly devices that offer a seamless experience to the users.

This project presents a novel application of IoT-based voice-controlled systems, specifically the development of an FM radio that can be controlled using voice commands. The system is built using Arduino, a popular open-source platform for building electronic projects, and Google Assistant, a voice-controlled intelligent personal assistant developed by Google. The aim is to develop a system that can be easily used by anyone, without requiring any technical expertise.

The Internet of Things (IoT) is a term first used by Kevin Ashton in 1999 while working at MIT's Media Center. He meant the concept of computers and machines with sensors that connect to the internet to report status and receive control commands [4]. The Internet of Things (IoT) is a network of devices that are connected to the internet and can communicate with each other. IoT has brought about a new wave of innovation in various fields, including healthcare, transportation, and smart homes [4]. IoT has made it possible to control everyday devices remotely, and voice assistants have made it easier to interact with these devices without the need for physical interaction [5].

According to [6], The Internet of Things (IoT) describes the network of physical objects in which sensors, software and other technologies are embedded in order to connect and exchange data with other devices and systems via the Internet. These devices range from ordinary household items to sophisticated industrial tools. With more than 7 billion connected IoT devices today, experts expect that number to grow to 10 billion by 2020 and 22 billion by 2025.

The Internet of Things, or IoT, can also be described as a system of interconnected computing devices, mechanical and digital machines, objects, animals, or people, each tagged with unique identifiers (UIDs) and the ability to transmit data over a network without this requires human-to-human or human-to-computer interaction [7]. A thing on the Internet of Things can be a person with a heart monitor implant, a farm animal with a biochip transponder, a car with built-in sensors that warn the driver when tire pressure is low, or any other natural or man-made object that an Internet Protocol (IP) address can be assigned and that can transmit data over a network [7].

In short, IoT refers to the collective network of connected devices and the technology that facilitates communication between devices and the cloud, as well as between the devices themselves [8].

Over the years, there have been several advances in IoT technology, which have made it easier to connect devices and create smarter, more efficient systems [9]. Some of these advances include:

Enhanced Connectivity: IoT devices now have improved connectivity options, including low-power, long-range wireless protocols like LoRa and Sigfox, as well as the latest cellular networks like 5G, which offer faster speeds and lower latency [9].

Improved Data Analytics: IoT platforms can now process large amounts of data generated by devices in real-time, thanks to advanced data analytics tools and algorithms. This enables businesses to make better decisions based on insights gained from IoT data [10].

Smarter Sensors: Advances in sensor technology have made it possible to create smaller, more efficient sensors that can detect and measure a wider range of environmental factors. This has enabled the development of more sophisticated IoT applications [11].

Increased Security: IoT security has improved significantly, with better encryption and authentication protocols, as well as advanced security features like secure boot and over-the-air firmware updates [11].

Edge Computing: Edge computing allows IoT devices to process data locally, reducing the need for large amounts of data to be transmitted to the cloud. This results in faster response times and lower latency, making IoT systems more efficient [12].

AI and Machine Learning: AI and machine learning are being integrated into IoT systems to enable them to learn and adapt to changing conditions, making them more efficient and effective [12].

Overall, these advances in IoT technology have made it possible to create smarter, more efficient systems that can help businesses and individuals save time, reduce costs, and improve productivity. As IoT technology continues to evolve, we can expect to see even more advanced applications and use cases in the future.

Voice-controlled systems, offer several advantages over traditional systems. They allow for hands-free operation, which is particularly useful while driving or performing other tasks. They also provide greater flexibility in terms of controlling the system, as users can use natural language commands rather than having to navigate through menus [13].

Digital voice assistants like Google Assistant, first became popular in mainstream computing when they were introduced to smartphones and then tablets. This allowed users to use the mobile devices to ask a series of questions without having to physically type in the request. This is achieved by using data stored in the connected digital voice assistant cloud service, e.g., information stored on a computer server that can be accessed by the digital language assistant. So, if a voice query like “What is the weather like in Kathmandu today?” occurs, the response is generated from information stored on the internet, just like typing a query into a search engine on a PC, laptop, smartphone or tablet [2].

Internet of Things (IoT) has been utilized recently to integrate voice control to various static devices, making them more interactive, convenient and easy to use. With the power of voice commands, users can control and automate various device tasks hands-free, making their lives more convenient, efficient, and enjoyable. Some of the areas where IoT has been used to integrate voice control include Smart homes, Retail and Hospitality, Industrial Automation, Healthcare, etc. [14]

FM (Frequency Modulation) radio is a popular broadcasting technology that utilizes frequency modulation to transmit audio signals. It has a long and significant history, starting from its inception in the early 20th century [3]. FM radio was invented by Edwin H. Armstrong, an American electrical engineer, in the 1930s. Armstrong's breakthrough was the development of a reliable method of frequency modulation, which allowed for high-quality audio transmission with reduced noise and interference compared to the existing AM (Amplitude Modulation) radio technology [15].

On June 9, 1939, Armstrong made the first public demonstration of FM radio broadcasting, transmitting a performance of the Philadelphia Symphony Orchestra over an FM station located in Alpine, New Jersey. This event marked the beginning of FM radio's journey towards becoming a widely used broadcasting technology [16]. The success of FM radio sparked further innovations in the field of radio broadcasting, including the development of satellite radio, internet radio, and digital audio broadcasting (DAB) systems. These advancements have expanded the options available to listeners and diversified the radio broadcasting landscape [17].

1.2. Problem statement

The current FM radio technology lacks voice control, which limits the convenience and ease of operation of the device. The traditional FM radio receivers requires manual adjustments, such as changing the channel or adjusting the volume, which can be inconvenient for users who are busy or have physical limitations. Furthermore, with the advent of IoT, users are increasingly expecting their devices to be connected to the internet and have the ability to interact with other devices [18]. The proposed project addresses these limitations by developing an IoT-based voice-controlled FM radio using Arduino and Google Assistant. This device will enable users to operate the FM radio receiver remotely using voice commands, providing convenience and ease of operation.

1.3. Aim and objectives of the study

The aim of the project is to design and implement a mechanism used in controlling an FM radio receiver's volume and channel selection using Internet of Things (IoT)-based voice-control

The specific objectives of this project are:

- To investigate the feasibility of using the Google Assistant platform for voice control and explore the potential of using Arduino for IoT-based applications.
- The project will involve designing and building the hardware and software components of the IoT-based voice-controlled Mechanism.
- To evaluate the system performance in terms of accuracy, response time, and user satisfaction.
- To compare the proposed system with existing IoT based voice-controlled systems.

1.4. Significance of the study

This study addresses the growing demand for smart home devices that can be controlled using voice commands, thereby enhancing user experience and convenience [19]. It also utilizes the Internet of Things (IoT) technology, which is rapidly gaining popularity due to its ability to connect various devices and enable remote access and control [20]. Furthermore, this project highlights the need for further research into improving the voice recognition accuracy of Google Assistant and developing more sophisticated voice-controlled devices.

1.5. Methodology

The methodology involves using Arduino and Google Assistant to develop the mechanism for control using voice commands. Thus, these key phases are significant; the system design phase; software development phase; integration phase; and the testing/evaluation phase.

In the system design phase, the project highlights the necessary hardware components and their integration. It explains the selection and setup of the Arduino board, FM radio module, microphone, and speaker. The circuit design and connections between these components are explained in detail, providing a clear understanding of the hardware setup.

The software development phase delves into the programming aspects of the project. It outlines the programming language used, such as the Arduino programming language, and the IDE (Integrated Development Environment) utilized for coding. The project also explains the code structure and algorithms required for the system's operation. Additionally, it discusses the libraries or APIs used for integrating Arduino with Google Assistant, enabling voice control functionalities.

The integration phase focuses on the integration with Google Assistant which a crucial aspect of the project. The project explains the configuration steps required to set up a Google Assistant project and obtain the necessary API credentials. It discusses the methods used to establish communication between the IoT device, represented by the Arduino-based FM radio, and the Google Assistant platform. This integration allows users to control the FM radio device using voice commands issued through Google Assistant.

The testing and evaluation phase focuses on assessing the functionality, performance, and user experience of the voice-controlled FM radio system. The project outlines the methodology for conducting tests, including test scenarios and criteria for evaluating the system's accuracy and response time. It analyzes the results obtained and provides insights into the system's strengths and limitations. The findings from the testing phase are then discussed, with a comparison to the project's initial objectives. Additionally, potential improvements, future directions, and areas for further research are highlighted.

1.6. Scope of the study

The scope of the study involves the development, and implementation of an IoT-based voice-controlled FM radio system using Arduino and Google Assistant. The study will involve the selection and integration of appropriate hardware components, such as the Arduino board, FM radio module, microphone, and speaker. The circuit design and connections between these components will be established to ensure proper functionality and communication.

Software development will be a key aspect of the study, including programming the Arduino board and developing algorithms for voice recognition and control. The system will utilize the Arduino programming language, and the coding process will involve the use of libraries or APIs for seamless integration with Google Assistant [21].

Integration with Google Assistant will be a crucial phase of the study. This will involve configuring a Google Assistant project, obtaining the necessary API credentials, and establishing a communication interface between the IoT device and the Google Assistant platform. The integration will enable the voice commands received by Google Assistant to be translated and executed on the FM radio device [22].

1.7. Limitations of the study

This section discusses the limitations encountered during the research conducted on the development of an IoT-based voice-controlled FM radio using Arduino and Google Assistant. By acknowledging these limitations, researchers can gain a comprehensive understanding of the study's scope and potential areas for future improvements.

i. Hardware Limitations: The study employs Arduino as the hardware platform for implementing the IoT-based voice-controlled FM radio system. However, Arduino boards have specific limitations, such as limited processing power and memory capacity. These limitations might constrain the complexity and responsiveness of the implemented system, potentially impacting the overall user experience [23].

ii. Voice Recognition Accuracy: The accuracy of voice recognition systems, including Google Assistant, is a crucial factor affecting the performance of the IoT-based voice-controlled FM radio. Although Google Assistant offers advanced natural language processing capabilities, it may still encounter challenges in accurately recognizing voice commands, particularly in noisy environments or when dealing with accents or variations in speech patterns. Inaccurate voice recognition could lead to unintended actions or system malfunctions [24].

iii. Network Connectivity and Latency: IoT devices rely on network connectivity for seamless communication and operation. In the case of the IoT-based voice-controlled FM radio, a stable and robust internet connection is essential for transmitting voice commands to the Google Assistant platform and receiving appropriate responses. However, network instability or high latency could introduce delays in command execution, leading to a suboptimal user experience [25].

iv. FM Radio Frequency Limitations: FM radio frequencies are regulated and allocated by governmental bodies in different regions. The availability and allocation of frequencies may vary, leading to differences in signal strength, quality, and coverage. The study's findings related to FM radio reception and performance may be influenced by the specific geographical location or regulatory environment, limiting the generalizability of the results [26].

v. User Acceptance and Usability: The success of an IoT-based voice-controlled FM radio system relies on user acceptance and usability. Factors such as user familiarity with voice-controlled interfaces ease of use, and overall user experience can significantly influence the adoption and satisfaction levels. The study might not capture the diverse user perspectives and preferences that could impact the overall assessment of the system's feasibility and effectiveness [27].

1.8. Project outline

The introduction section provides an overview of the topic and sets the context for the study. It typically begins with a brief explanation called the background of the study it then precedes to the problem statement, the aims and objectives of the study, the significance of the study, the methodology, and scope of the study. The introduction serves to capture the reader's attention and provide a clear understanding of the purpose of the research.

The literature review is a critical evaluation of existing scholarly works and research relevant to the topic of study. It involves reviewing and summarizing previously published literature, such as journal articles, books, and conference papers. The purpose of the literature review is to identify gaps, debates, or trends in the existing body of knowledge and to establish the theoretical and conceptual framework for the current research. It helps to situate the research within the broader academic discourse and demonstrates the researcher's understanding of the existing literature in the field.

The materials and methods section details the materials, tools, and techniques used to achieve the objectives of the study. It describes the experimental design, and procedures employed to address the research objectives. This section should provide sufficient detail to allow other researchers to replicate the study if desired. It may also include information on data analysis methods, and any ethical considerations or limitations.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview of FM radio technology

FM radio technology is one of the oldest wireless communication technologies, dating back to the early 1930s. During the twentieth century, two types of modulation dominated radio broadcasting: amplitude modulation (AM) and frequency modulation (FM). They both begin with a continuous radio-frequency sine wave called a carrier that is, a wave of constant amplitude and frequency that oscillates above approximately 100,000 cycles per second (cps) [16].

American engineer Edwin Howard Armstrong was a pioneer in frequency modulation (FM) radio technology, which revolutionized radio broadcasting. The technology was an improvement over amplitude modulation (AM), which was the standard radio technology at the time. FM radio had better sound quality, a more extended range, and fewer noise problems than AM radio. However, the technology was not widely adopted initially, and it was not until the 1940s that FM radio started to gain popularity [16].

Frequency Modulation or FM is a form of modulation that conveys information by varying the frequency of a carrier wave; the older amplitude modulation, or AM, varies the amplitude of the carrier while keeping its frequency constant [28]. With FM, the frequency deviation from the assigned carrier frequency at any point in time is directly proportional to the amplitude of the (audio) input signal and determines the instantaneous frequency of the transmitted signal. Because transmitted FM signals use significantly more bandwidth than AM signals, this form of modulation is commonly used with the higher frequencies (VHF or UHF) used in television, the FM broadcast band, and land mobile radio systems [28].

The word radio derives from the Latin word radius and means wheel spoke, ray of light, ray. It was first applied to communications in 1881 when Alexander Graham Bell, at the suggestion of French scientist Ernest Mercadier [fr], adopted radiophone (meaning radiated sound) as an alternative name for his optical photophone transmission system [29]. Radio is the technology of signaling and communicating using radio waves. Radio waves are electromagnetic waves of frequency between 3 hertz (Hz) and 300 gigahertz (GHz). They are generated by an electronic device called a transmitter connected to an antenna which radiates the waves and received by another antenna connected to a radio receiver [29]. Electromagnetic waves were predicted by James Clerk Maxwell in his 1873 concept of electromagnetism, now called Maxwell's equations, who proposed that a coupled oscillating electric field and magnetic field could travel through space as a wave and proposed that light consisted of electromagnetic waves of short wavelength. On November 11, 1886, German physicist Heinrich Hertz, attempting to confirm Maxwell's theory, first observed radio waves he generated using a primitive spark gap transmitter. Experiments by Hertz and physicists Jagadish Chandra Bose, Oliver Lodge, Lord Rayleigh, and Augusto Righi, among others, showed that radio waves like light demonstrated reflection, refraction, diffraction, polarization, standing waves, and traveled at the same speed as light, confirming that both light and radio waves were electromagnetic waves, differing only in frequency. In 1895, Guglielmo Marconi developed the first radio communication system, using a spark gap transmitter to send Morse code over long distances [29].

FM radio has multiple uses and applications. It serves as a source of entertainment, broadcasting music, talk shows, and various audio content. FM radio also provides news updates, weather reports, and traffic information, making it a reliable medium for staying informed. It is used for public service announcements, educational programming, advertising, and marketing purposes. FM radio facilitates communication during special events and emergencies, offering live coverage and critical information to listeners. Its versatility and wide reach make it an essential medium for entertainment, information dissemination, and community engagement [30].

FM radio broadcasting stations transmit audio signals by modulating a carrier wave with the original audio source. The audio signal is processed to optimize its quality and undergoes frequency modulation, where the carrier wave's frequency is varied according to the changes in the audio signal [31]. The modulated signal is then amplified and broadcasted through an antenna, converting it into radio waves that propagate through the air. FM radio receivers equipped with antennas capture these radio waves and demodulate them, extracting the original audio signal [32]. The demodulated audio is then amplified and played back through speakers or headphones, allowing listeners to enjoy the broadcasted content. By tuning their receivers to specific frequencies, listeners can select and tune in to different FM radio stations in their area [31].

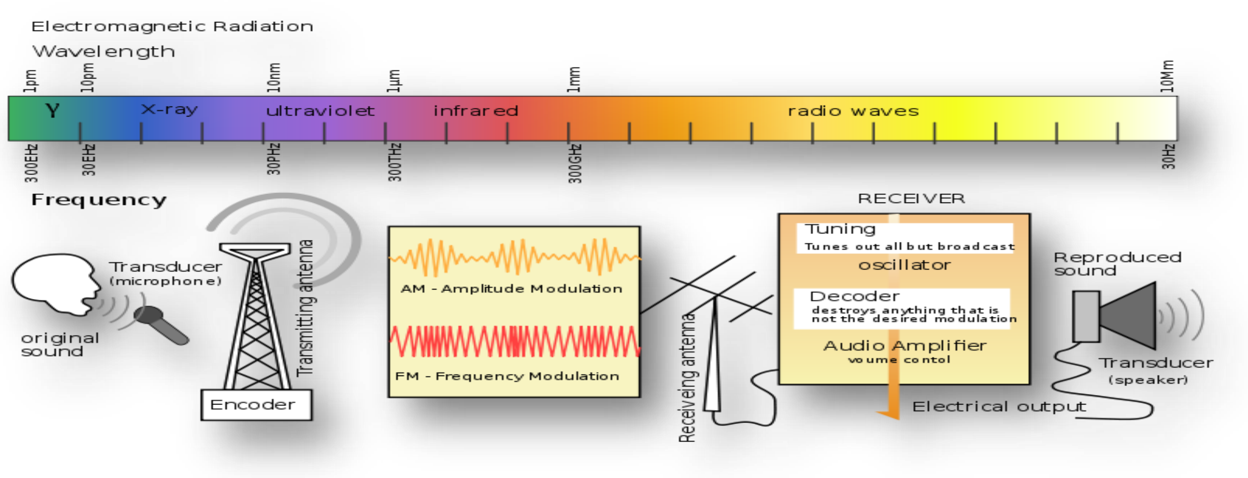


Figure 2.1 shows electromagnetic spectrum with radio frequency spectrum. It also shows the use of AM and FM modulation to transmit audio information [31]

2.2. The concept of IoT and voice-controlled systems

The Internet of Things (IoT) refers to the network of physical objects or "things" that are embedded with sensors, software, and other technologies that enable them to connect and exchange data with other devices and systems over the internet [33]. The data collected can be analyzed to extract valuable insights, which can be used to improve decision-making, optimize processes, and enhance user experiences [33]. The integration of IoT with voice-controlled systems has revolutionized the way we interact with devices and appliances, allowing us to control them using simple voice commands.

Voice-controlled systems are an example of how IoT can be leveraged to enhance user experiences. They enable users to interact with devices using natural language, which makes the interaction more intuitive and user-friendly. Voice-controlled systems use speech recognition technology to convert spoken words into digital signals, which are then processed and interpreted by the device [34]. The device can then perform the desired action, such as playing a song, turning on a light, or setting a reminder [34]. Voice-controlled systems are becoming increasingly popular, with products such as Amazon Echo, Google Home, and Apple HomePod leading the market [2].

IoT and voice-controlled systems can be coupled to produce a distinctive and novel product, as shown by the development of an IoT-based voice-controlled FM radio utilizing Arduino and Google Assistant. An FM receiver module and a speaker are attached to an Arduino board, which is used to construct the radio. Using the Google Assistant API, the Arduino board is configured to accept voice commands from Google Assistant. The desired FM frequency, which is set on the FM receiver module, is then determined by processing the spoken commands. The audio is then played over the speaker when the radio has been adjusted to the proper frequency.

Fig 2.2 shows the block diagram of IoT based Home-Automation using Google assistant [5].

The Internet of Things (IoT) and voice-controlled devices have completely changed how we interact with technology. The integration of Google Assistant and Arduino to develop an IoT-based voice-controlled FM radio shows how these technologies could produce cutting-edge and user-friendly products. We may anticipate seeing more exciting and cutting-edge uses in the future as IoT and voice-controlled technologies continue to develop.

2.3. Overview of Arduino in IoT devices

Arduino is popular open-source hardware and software platform widely used in the Internet of Things (IoT) devices [35]. It provides an easy-to-use interface for building and prototyping electronic devices, enabling developers to quickly and easily create IoT solutions that can connect to the internet, interact with other devices, and collect data [35].

Arduino is made up of both hardware and software. The Arduino board is a printed circuit board (PCB) with additional inputs and outputs in addition to a microcontroller chip. Numerous other electronic components on the board are necessary for the microcontroller to operate or to increase its functionalities [36]. The Arduino Integrated Development Environment (IDE), a piece of software that offers a straightforward and user-friendly interface for creating and uploading code to the board, can be used to program these boards [35].

A microcontroller is a tiny computer that is housed on a single integrated circuit. Electronics programming and control are made possible by microcontrollers. Inputs and outputs can be connected to microcontroller boards, which also contain other practical connectors and parts. The Basic Stamp, the PIC, and the Wiring board are a few examples of devices with microcontroller boards [36].

Arduino started its life in Italy, at Interaction Design Institute Ivrea (IDII), a graduate school for interaction design that focuses on how people interact with digital products, systems, and environments and how they in turn influence us [36].

There isn't a single, universal Arduino board; instead, there are numerous varieties, each with a unique design to accommodate a range of uses. The decision of which board to use might be difficult because there are an expanding number of boards, each with intriguing new possibilities. However, the Arduino Uno can be referred to as the one board that practically everyone starts with and is appropriate for the majority of applications—can be called the archetype of the Arduino hardware [36].

For this project, the NodeMCU (Node MicroController Unit) an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266 [37]. The NodeMCU offers a variety of development environments, including compatibility with the Arduino IDE (Integrated Development Environment) [37].

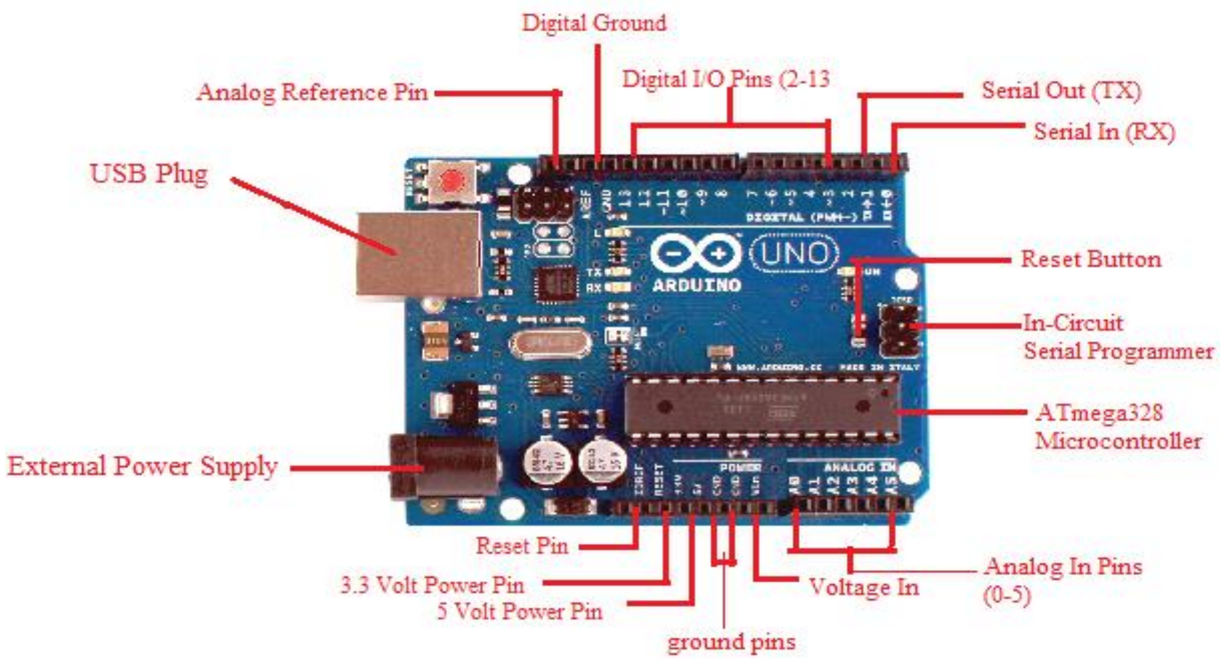


Fig 2.3 shows the layouts of a typical Arduino Uno board [21], [36]

IoT devices powered by Arduino boards can perform a wide range of functions, such as controlling lights, monitoring environmental conditions, collecting data from sensors, and even running small web servers [38]. Arduino's flexibility and versatility make it an ideal platform for rapid prototyping and development of IoT devices.

One of the advantages of using Arduino in IoT devices is the large community of developers and enthusiasts who share their knowledge, code, and projects [39]. This community has created a vast ecosystem of libraries, tools, and resources that enable developers to easily integrate sensors, modules, and other components into their projects [39]. Additionally, the Arduino platform is compatible with a variety of communication protocols such as Wi-Fi, Bluetooth, and Zigbee, making it easy to connect to other devices and services [21].

Here are some examples of IoT devices that use Arduino:

- Smart home devices such as thermostats, lighting systems, and security systems.
- Environmental monitoring systems that measure temperature, humidity, air quality, and other environmental parameters.
- Wearable devices such as fitness trackers and smart watches.
- Industrial monitoring and control systems for factories and manufacturing plants.
- Agricultural monitoring systems for monitoring crop health and soil conditions.

In general, the way developers' approach IoT development has changed because of the use of Arduino in IoT devices. It is a popular choice for experimenting and developing IoT solutions due to its user-friendliness, adaptability, and large open-source community.

2.4. Related works

The Internet of Things (IoT) is an emerging technology that allows devices to communicate with each other via the internet. IoT has already revolutionized many industries, and the radio industry is no exception. The IoT-based voice-controlled FM radio is an innovative solution that combines the power of IoT and voice control to provide a user-friendly and hands-free experience for the users. In recent years, there have been numerous studies exploring the use of voice control in conjunction with devices like Arduino and Google Assistant.

Voice-controlled devices have been an area of interest for researchers for several years. Here are some of the previous studies conducted on voice-controlled devices:

[40] In this research article, the authors have presented the current status of IoT standards and their applicability to human life. The authors have also highlighted the potential of IoT technology to revolutionize human life with its ability to recognize communication between articles, gadgets, and other belongings of human beings to provide intelligent services. Furthermore, the authors offer an overview of IoT architecture, applications, and future directions.

[41] This research article presents the construction and implementation of a smart home system using Raspberry Pi microcontroller and a GSM module, enabling automatic or semi-automatic control and monitoring of household appliances and other house features. The authors have demonstrated the use of a microcontroller to build smart electronic devices that can automatically control electrical appliances with minimal circuitry complexities and components.

[42] In this article, the authors have presented an IoT-based wireless wearable device for an early warning system to detect cytokine storms from former COVID-19 patients even after recovering from COVID-19. The wearable device can monitor former COVID-19 patients remotely from anywhere by connecting to the internet by the use of dedicated mobile apps for them.

[43] This research article presents a proposed framework for IoT security against malware detection, with the aim of ensuring a secure IoT environment. The author

explores various factors affecting malware detection and proposes a framework to mitigate IoT security threats.

[44] The authors present a research article presenting a control mechanism for smart home automation using natural language processing (NLP), voice recognition, and IoT technologies in a multi-tier architecture. Their proposed system exploits several available APIs in order to make the home devices more user-friendly, which make it easier to control them.

[45] The authors in this article discussed cognitive Internet of Things (CIoT) and its implementation through Software-Defined Radio (SDR). They have also presented a preliminary step towards CIoT by exploring the requirements of a CIoT-based application, feasibility of the application, and some research insights.

[46] This research article presents the concept of smart energy consumption of IoT with millimeter-wave cognitive radio for 5G cellular network. The authors propose a smart energy consumption solution for IoT devices using millimeter-wave cognitive radio. The solution is envisaged to be an energy-saving utility in the flexible deployment of IoT architecture.

[47] The authors in this research article present a smart parking solution by integrating NB-IoT radio communication technology into the core IoT platform. The article proposes the integration of NB-IoT into the core IoT platform, enabling direct sensor data navigation to the IoT radio stations for processing, after which they are forwarded to the user application programming interface (API).

[48] This research article presents a review on IoT techniques for automating devices. The author offers an overview of the IoT architecture and technologies used in IoT. The article also highlights the potential of IoT technology to revolutionize human life with its ability to recognize communication between articles, gadgets, and every belonging of human beings to provide intelligent services.

[49] This research article provides a comprehensive review on the Internet of Things (IoT) based wireless sensor network. The authors explore different IoT sensor network related concepts, applications, and existing IoT technologies. The review also includes future areas of application for IoT.

[50] The authors in this research article presented the design and development of voice enabled IoT devices for smart home automation. The proposed system uses speech recognition, natural language processing, and other IoT technologies to control home appliances, lighting, and other smart-home devices.

[51] This research article presents the design and implementation of a voice-based home automation and security system using the Internet of Things (IoT). The authors use different IoT technologies such as voice recognition, home appliances control, and smart sensors to control various household gadgets and security systems.

[52] In this research article, authors present an IoT-based home automation system that uses voice control to control different household gadgets. The proposed system uses different IoT technologies such as Raspberry pi, Bluetooth, home appliances control, and smart sensors to control different home appliances.

[53] This research article presents the design and implementation of a voice-controlled smart wheelchair for physically disabled people. The system uses voice recognition and other IoT technologies to control the wheelchair's movement and provide a more natural way of controlling the wheelchair for physically disabled individuals. [54] The authors of this research article present a Voice Controlled Wireless Wheelchair System for physically disabled people. This wheelchair system aims to offer an intelligent system that allows physically disabled people to use voice commands for moving their wheelchair.

[55] In this research article, the authors present an application of IoT and voice control technology in smart campus. The proposed system uses IoT sensors to monitor and control different campus facilities using voice commands [56] Conducted a comprehensive analysis of the profile traces from the BG96 NB-IoT module operating in various states of the Radio Resource Control protocol of 2G, 3G, and 4G network technologies for smart cities. [57] Reviewed the role of IoT in designing smart home automation and highlighted privacy issues associated with the technology. [58] Developed a smart air conditioner equipped with air quality filters, human sensors, temperature control, voice control, cloud storage services, and solar panel services that can be controlled through smartphones devices. [59] Presented a proposed framework that integrates deep neural networks and support vector machines to improve the quality of voice processing in IoT devices. [60] Highlighted the role of open-source hardware as an example of an FM radio.

These studies show that voice-controlled devices have been an area of active research for several years, and that researchers have explored a variety of approaches to designing these devices, including microprocessor-based designs, neural networks, deep learning, and natural language processing.

CHAPTER THREE

MATERIALS AND METHOD

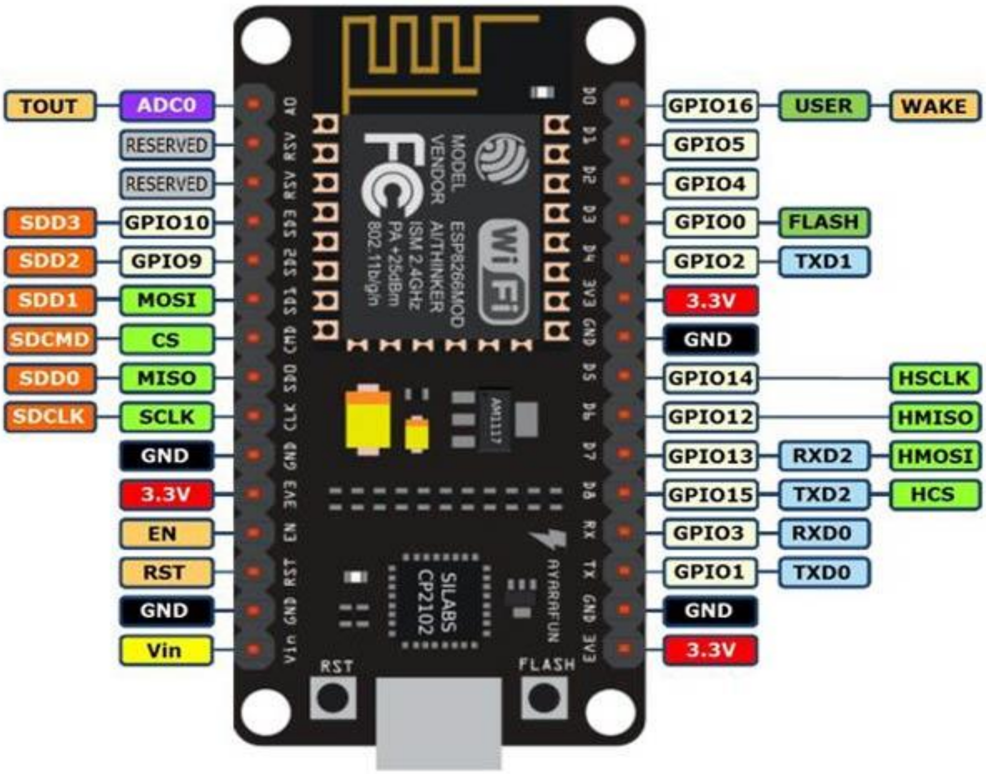
3.1. Description of the hardware and software components

For an IoT based voice-controlled FM radio using Arduino and Google Assistant project, the following hardware and software components may be used:

Hardware Components:

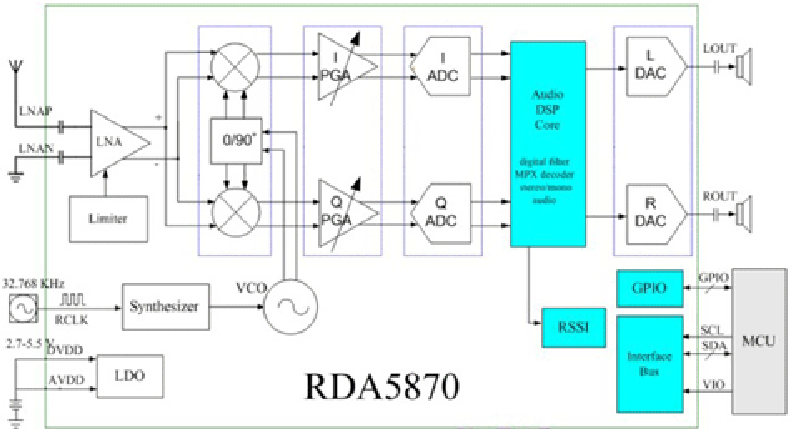
1. NodeMCU Microcontroller: The NodeMCU Microcontroller [61] is a popular and versatile development board specifically designed for Internet of Things (IoT) projects. It is based on the ESP8266 Wi-Fi module and combines the capabilities of a microcontroller with built-in Wi-Fi connectivity [61]. The NodeMCU board provides an easy-to-use platform for prototyping and developing IoT-based devices. Key features of the NodeMCU microcontroller include:
    i. ESP8266 Wi-Fi module: The NodeMCU is built around the ESP8266 Wi-Fi module, which allows for seamless wireless connectivity to the internet. This feature is essential for IoT applications, enabling devices to connect to the internet and communicate with other devices and services.
    ii. Programmable: The NodeMCU microcontroller can be programmed using the Arduino IDE (Integrated Development Environment), which is a popular and user-friendly programming platform. This makes it accessible to developers familiar with Arduino programming.
    iii. Lua-based firmware: The NodeMCU firmware is based on the Lua scripting language, which simplifies the process of programming and developing IoT applications. Lua offers a concise and efficient syntax, making it easier for developers to write code for the NodeMCU.
    iv. GPIO Pins: The NodeMCU board features a range of general-purpose input/output (GPIO) pins, allowing for the connection of various sensors, actuators, and other peripheral devices. These pins enable the board to interact with the physical world and interface with external components.
    v. Built-in USB interface: The NodeMCU board includes a built-in USB interface, which simplifies the process of programming and powering the board. It can be connected directly to a computer, eliminating the need for additional external programming hardware.

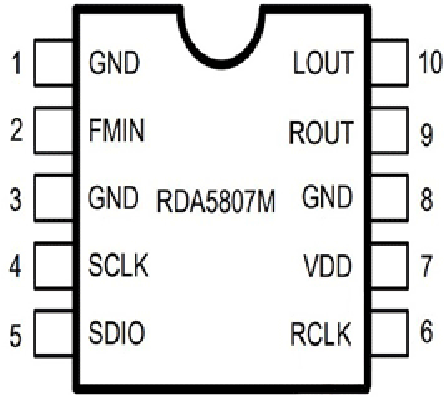
Fig 3.1 below shows the pinout of the NodeMCU ESP8266 microcontroller [61]



2. RDA5807 FM Radio Module: The RDA5807 FM Radio Module [62] is a compact and highly integrated circuit designed for FM radio applications. It provides a convenient and easy-to-use solution for adding FM radio functionality to electronic devices or projects. It features a built-in FM tuner with a frequency range of 50MHz to 115MHz and supports stereo audio output. The module communicates with the controlling device using the I2C protocol, enabling easy integration into microcontroller-based projects. With features such as auto search, frequency control, and low power consumption, the RDA5807 module provides a cost-effective solution for adding FM radio functionality to electronic devices. Its compact size and affordable price make it a popular choice for various applications, including portable radios, music players, and wireless speakers.

   Fig 3.2 below shows side by side the block diagram and top view of the RDA5807 FM Radio Module [62]





3. PT2258 Digital Volume Controller: The PT2258 [63] is a digital volume controller integrated circuit chip designed for audio applications. It provides a convenient and precise solution for controlling volume levels in electronic devices or audio systems. The PT2258 Digital Volume Controller is a versatile chip designed for audio applications, offering precise and accurate digital volume control. With support for multiple channels, a serial interface for easy integration, and a wide range of attenuation, it provides fine-grained adjustment of audio volume levels. The chip includes an integrated mute function for instant silencing, while maintaining low distortion and noise levels. With low power consumption and compact package options, the PT2258 is a reliable choice for various audio systems, amplifiers, and multimedia devices.

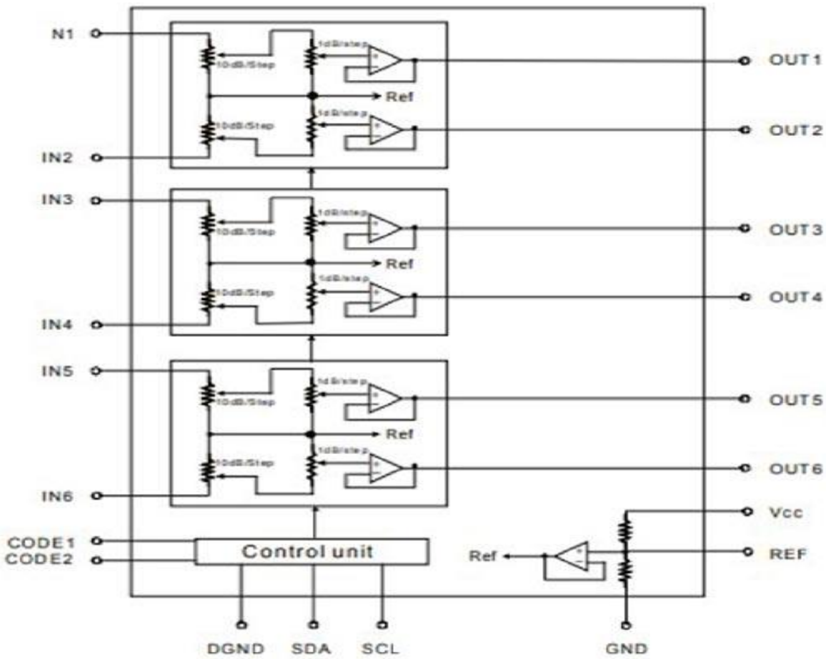


Fig 3.3 above shows the schematic diagram of the PT2258 Digital Volume Controller IC [63]

4. Logic Level Converter: The Logic Level Converter is a crucial component in IoT-based voice-controlled devices, enabling seamless communication between devices operating at different logic voltage levels. It facilitates bidirectional voltage level translation, ensuring compatibility and reliable data transfer. With support for multiple channels, it can handle simultaneous signal conversion, while maintaining signal integrity and accuracy. The converter's compact and user-friendly design makes it easy to integrate into IoT projects, serving as a vital interface between microcontrollers, sensors, voice recognition modules, and other components. Overall, the Logic Level Converter plays a critical role in enabling efficient and reliable communication in voice controlled IoT applications.
5. SPDT Relay 6V: The Single-Pole Double-Throw (SPDT) Relay, specifically a 6V variant, is a commonly used component in IoT-based voice-controlled devices. It serves as an electrically operated switch that can control the flow of current between two separate circuits. The SPDT Relay 6V is a crucial component for this project, serving as an electrically operated switch. With its three terminals and ability to toggle between circuits, it allows for versatile control of electrical devices or circuits. Operating at 6V, it can be controlled remotely through a microcontroller or control system, enabling voice-activated switching. The relay offers signal isolation, ensuring safety and protecting sensitive components. Its low power consumption, compact design, and durability make it an integral part of voice controlled IoT applications, providing efficient and convenient control over various electrical loads.
6. 1n4007 Diode: The 1N4007 diode is a widely used component in IoT-based voice-controlled devices, primarily serving as a rectifier diode for converting AC to DC. With its maximum reverse voltage rating of 1000V and average forward current of 1A, it can handle a range of voltage and current requirements. The diode provides signal protection by clamping voltage spikes, ensuring the safety and smooth operation of sensitive components. It is available in various package options and is cost-effective, making it a popular choice for voice controlled IoT applications. The 1N4007 diode's reliability, polarized nature, and affordability contribute to its versatility and widespread use in IoT devices.
7. Resistors (10K, 100K, and 150K): The 10K, 100K, and 150K resistors are vital components for this project. These resistors have different resistance values and serve multiple functions in the circuits. They are commonly used in voltage divider circuits, current limiting applications, and as pull-up or pull-down resistors. These resistors

play a crucial role in analog circuits, ensuring proper gain, stability, and protection of sensitive components. Their selection and integration into the circuit design are essential for accurate and reliable operation of voice controlled IoT devices, contributing to their functionality.

8. Capacitors (0.1uF and 10uF): The 0.1uF and 10uF capacitors are other integral components, serving multiple functions. With their respective capacitance values, they contribute to filtering, decoupling, energy storage, voltage regulation, and signal coupling. These capacitors help ensure stable power supply, reduce noise, and maintain clean signal transmission. They play a crucial role in regulating voltage levels, storing electrical energy, and adjusting circuit stability and timing. Available in various package sizes and types, their proper placement within the circuit layout is important for optimal performance.
9. Speaker: A speaker is required to output the audio signals from the audio amplifier.
10. Smartphone: This can be used to communicate with the Google Assistant and control the FM radio.

Other hardware components include Breadboard, 3.5mm headphone jack, 5mm screw terminal, and jumper wires.

Software Components:

1. Arduino IDE: The Arduino Integrated Development Environment (IDE) [64] is a software platform commonly used to program and develop IoT-based devices. It offers a simplified version of the C++ programming language, making it accessible to beginners and experienced programmers alike. The IDE provides compatibility with a range of Arduino boards, extensive library support, and hardware abstraction, simplifying the coding process and enabling rapid prototyping. With a large community of developers, the Arduino platform offers extensive support and resources. The IDE is cross-platform compatible and highly extensible, allowing developers to customize and enhance their projects. Overall, the Arduino IDE is a versatile tool that streamlines the development of IoT devices, making it a popular choice for programmers in this field.
2. Google Assistant API: The Google Assistant API [65] is a versatile and powerful tool that allows developers to integrate Google Assistant into IoT-based voice-controlled devices. By leveraging the API, devices can understand and respond to voice commands, enabling users to interact with their devices using natural language. The API incorporates advanced natural language processing

capabilities, ensuring accurate understanding of user queries and context. It also provides control over IoT devices, allowing users to perform various actions, such as turning devices on or off and adjusting settings, through voice commands. The integration with the Google Smart Home platform facilitates seamless incorporation of IoT devices into a user's smart home ecosystem, enabling centralized control and automation. Furthermore, the Google Assistant API supports third-party app integration, expanding the functionalities and possibilities of IoT devices. Google offers comprehensive developer tools, documentation, and resources to assist developers in integrating the API effectively. Privacy and data security are prioritized, with strict guidelines in place to protect user data. Overall, the Google Assistant API empowers developers to create intuitive and interactive voice-controlled experiences in their IoT devices, enhancing user convenience and satisfaction.

3. Adafruit IO Library: The Adafruit IO Library [66] is a comprehensive toolset that simplifies the development of IoT devices by providing a cloud-based platform and a set of powerful features. With Adafruit IO, developers can seamlessly connect their devices to the cloud, enabling easy data monitoring, storage, and retrieval. The library offers customizable dashboards and data visualization tools, allowing real-time monitoring and visualization of device metrics and sensor readings. It integrates smoothly with popular IoT hardware platforms like Arduino and Raspberry Pi, ensuring compatibility and ease of use. The bi-directional communication capabilities of Adafruit IO enable devices to receive commands or notifications from the cloud, enabling remote control and triggering of device actions. Additionally, the library provides webhooks and automation features that allow developers to create rules and triggers based on specific conditions, enhancing automation and efficiency. Adafruit IO prioritizes data security and reliability, ensuring secure transmission and storage of data. Adafruit IO library can be used to control the RDA5807 FM Radio Module.
4. Arduino SoftwareSerial Library: The Arduino SoftwareSerial Library [67] is a versatile tool that enhances the capabilities of IoT devices. It allows developers to create additional software-based serial communication ports, offering flexibility when hardware serial ports are limited. The library supports different baud rates and provides a simple API for transmitting and receiving serial data. It is compatible with various Arduino boards and optimizes memory usage for efficient

resource utilization. Although it has limitations compared to hardware serial ports, such as lower maximum baud rates, the SoftwareSerial Library remains a valuable resource for developers seeking to integrate reliable and flexible serial communication in their voice controlled IoT devices.

5. IFTTT: IFTTT (If This Then That) [68] is a versatile web-based service that will play a crucial role in the development of an IoT-based voice-controlled FM radio using Google Assistant. With IFTTT, users can create custom connections and automations between Google Assistant and the FM radio device, enabling seamless voice command integration. Using applets, which consist of triggers and actions, IFTTT allows users to define specific voice commands that trigger corresponding actions on the FM radio device, such as turning it on, tuning to a specific frequency, or adjusting volume. IFTTT also supports integration with FM radio APIs, facilitating functionalities like station switching or volume control via voice commands. The service's cross-device compatibility ensures smooth communication between Google Assistant, the FM radio device, and other IoT devices within the ecosystem. Users can take advantage of IFTTT's customization options to tailor the voice-controlled FM radio experience to their specific preferences and extend its functionality by integrating with other popular services like music streaming platforms or weather services. Overall, IFTTT empowers users to create a more intuitive and automated voice-controlled system, enhancing convenience and user experience.
6. Google Cloud Platform: Google Cloud Platform (GCP) enhances IoT-based voice-controlled devices that utilize Google Assistant by providing a comprehensive set of capabilities. GCP offers cloud integration for secure communication and data exchange, advanced voice recognition and natural language processing for accurate interpretation of voice commands, machine learning and AI services to improve intelligence and personalization, data analytics for insights and decision-making, scalability and reliability to handle growing user bases, robust security measures for data protection, and seamless integration with other Google services. Together, these features empower voice-controlled devices to deliver intelligent, efficient, and secure experiences to users.

By using the above hardware and software components, it is possible to create a voice-controlled FM radio that can be controlled using voice commands, leveraging the capabilities of IoT powered by Arduino and Google Assistant.

3.2. Design of the voice-controlled fm radio system

The following is a possible setup procedure for an IoT-based voice-controlled FM radio using Arduino and Google Assistant. The first step in developing this system is to gather all the necessary hardware components and software components, then thoroughly follow each step enumerated below.

1. First, gather the required components; you will need a NodeMCU ESP8266-based microcontroller, an RDA5807 FM radio module, a PT2258 digital volume controller, a logic level converter, resistors (such as 10K, 100K, and 150K), capacitors (such as 0.1uF and 10uF), a 3.5mm headphone jack, connecting wires, and power supply.
2. Connect the NodeMCU to the Arduino IDE: Connect the NodeMCU to your computer and open the Arduino IDE. Install the necessary board libraries and select the appropriate board (NodeMCU) from the Tools menu.
3. Connect the components: assemble the circuit on a breadboard, following the schematic diagram in Fig 3.4 below. Connect the RDA5807 FM radio module to the NodeMCU using the necessary wiring for power, ground, and control signals. Connect the PT2258 digital volume controller to the NodeMCU for volume control. Use the logic level converter to adjust the voltage levels if required. Connect the resistors, capacitors, and diodes as per the schematic diagram.

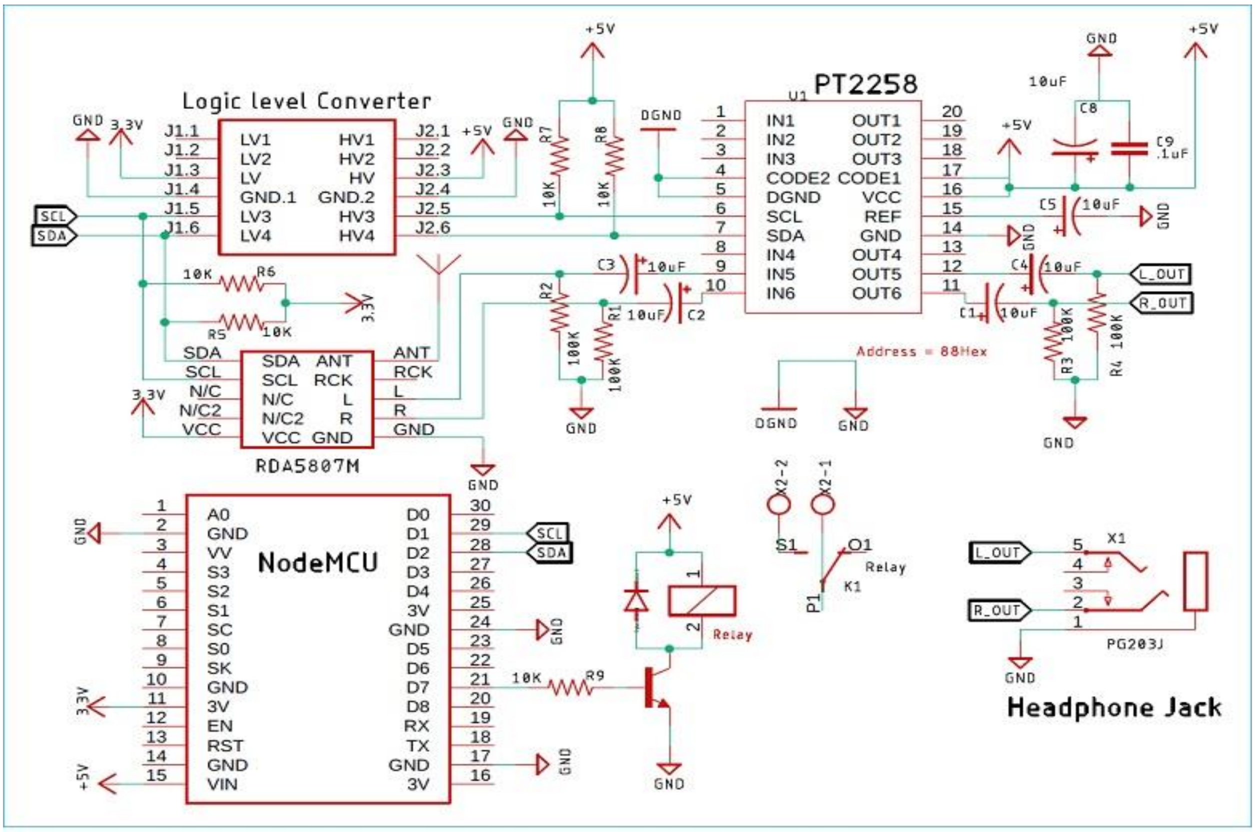


Fig 3.4 shows the schematic diagram of the voice controlled FM radio system

4. Write the firmware code: In the Arduino IDE, write the firmware code to control the FM radio module, volume controller, and other components. Use the appropriate libraries (such as Adafruit IO library) to interface with the modules. Implement the voice control functionality using the Google Assistant API and IFTTT. This code should include functions to receive voice commands, control the FM radio frequency, adjust the volume, and interact with the Google Assistant API.
5. Upload the firmware: Connect the NodeMCU to your computer and upload the firmware code to the microcontroller using the Arduino IDE. Ensure that the upload is successful without any errors.
6. Set up the Google Assistant API and IFTTT: Create a project in the Google Cloud Platform and enable the Google Assistant API. Configure IFTTT to link the voice commands from Google Assistant to trigger specific actions in your IoT-based FM radio system.
7. Test the system: Power on the setup, test the system and code by giving a voice command to a Google Assistant-enabled device e.g., an Android smartphone. For example, say "Hey Google, play 92.5 FM" or "Hey Google, increase the volume". Verify that the FM radio frequency changes, volume adjusts, and other functionalities work as intended.

   Connect the 3.5mm headphone jack: Connect the output of the FM radio module to the 3.5mm headphone jack to enable audio output. Ensure that the connection is secure, and the audio output is clear.
8. Once the system and code are working, perform experiments to evaluate the performance of the voice-controlled FM radio. Some possible experiments include:

i. Measuring the response time between giving a voice command and the FM radio tuning to the requested frequency.

ii. Measuring the accuracy of voice recognition by giving the Google Assistant different commands and evaluating how often it correctly interprets them.

iii. Evaluating the audio quality of the FM radio at different volume levels.

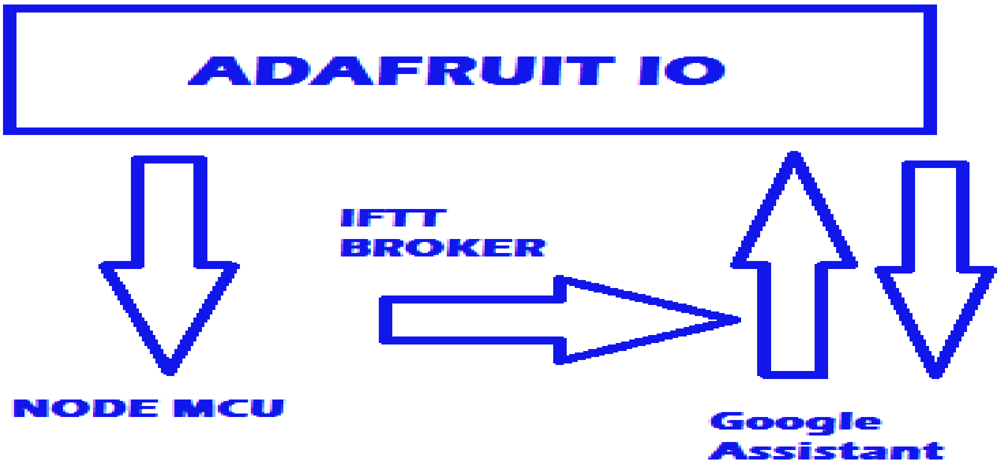


Fig 3.5 above shows the communication process between the Google Assistant and the NodeMCU.

CHAPTER FOUR
RESULT

4.1. Hardware implementation

The Implementation of the project design can be divided into two sections: Hardware and Software implementations. The hardware implementation consists of assembling the NodeMCU, RDA5807 FM Radio Module, PT2258 Digital Volume Controller, Logic Level Converter, SPDT Relay 6V, 1n4007 Diode, Resistors, and Capacitors on a bread board according to the schematic diagram shown in Fig 3.4.

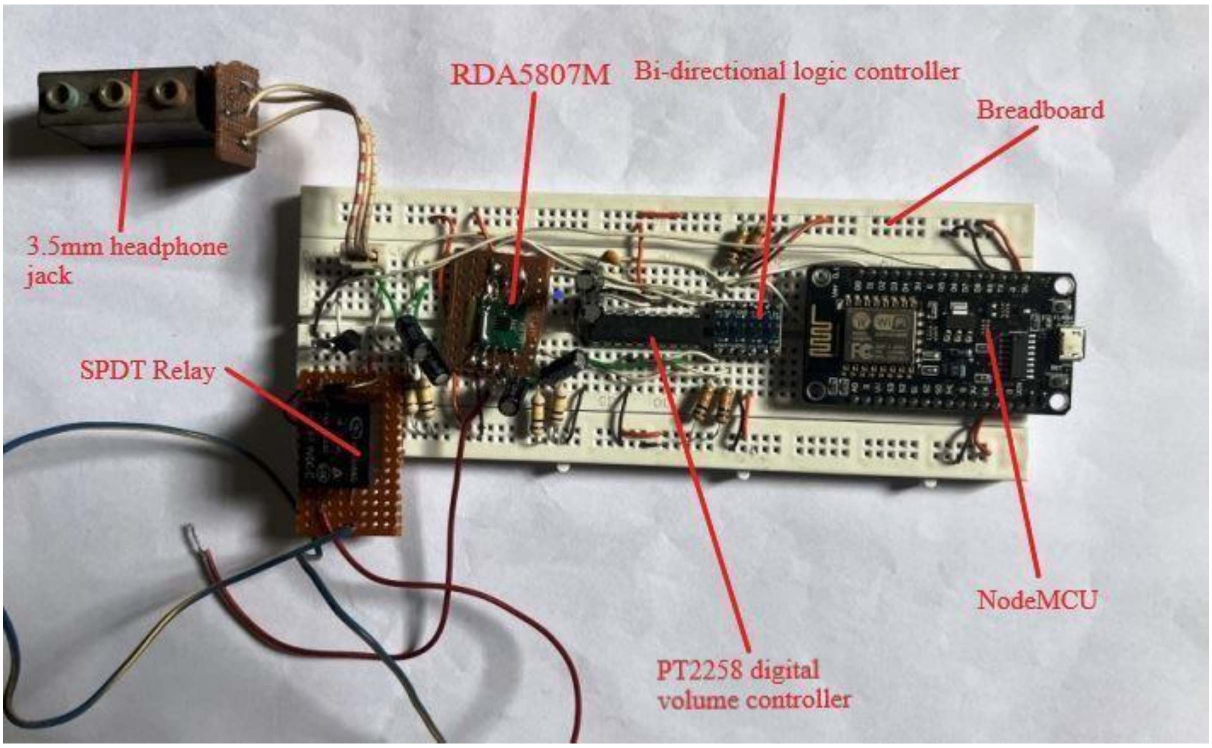


Fig 4.1 shows the assembled circuit

4.2. Software implementation

The software implementation focuses on the following.

- Setting up an Adafruit Account for Communication
- Setting up an IFTTT Broker for the FM Radio
- Writing code in Arduino IDE, to program the NodeMCU, control the FM radio module, capture audio from the microphone, and process voice commands.

4.2.1 Setting up an Adafruit account for communication

Create an Adafruit IO account on https://io.adafruit.com and login with your credentials.

Fig 4.2 shows the Adafruit homepage.

Click on *Dashboards*, then > *New Dashboard* after signing into your Adafruit account.

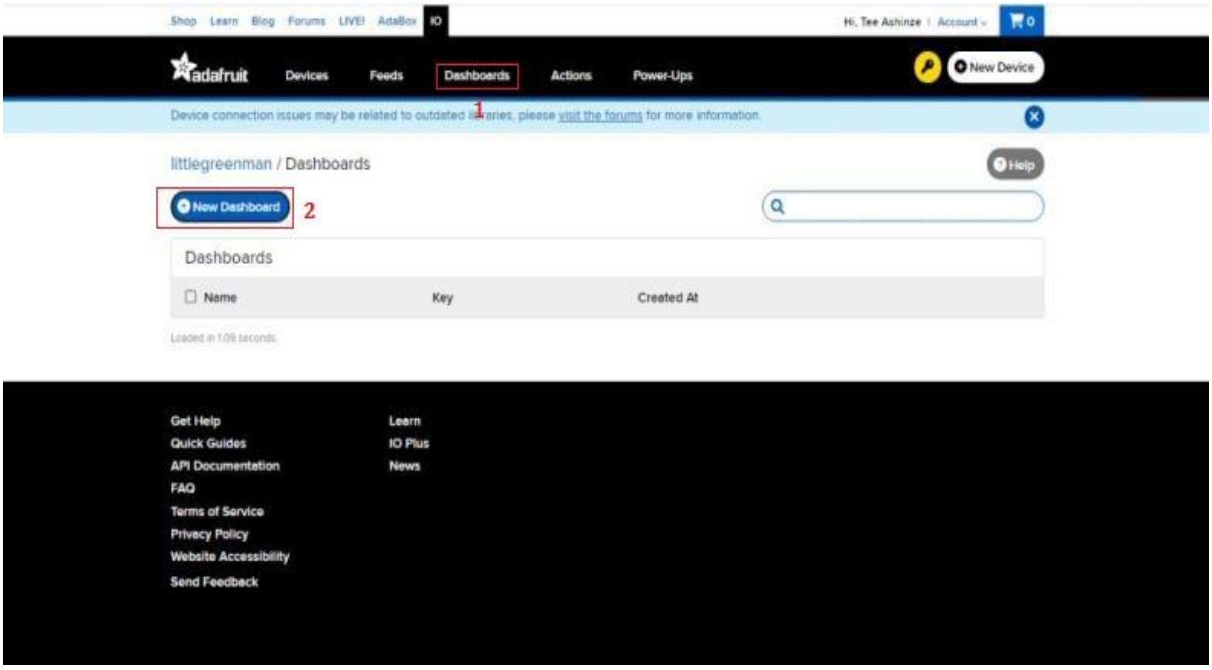


Fig 4.3 shows the steps to create a new Adafruit IO dashboard.

Next, we are going to add a new name & a short description of our new *Dashboard.*

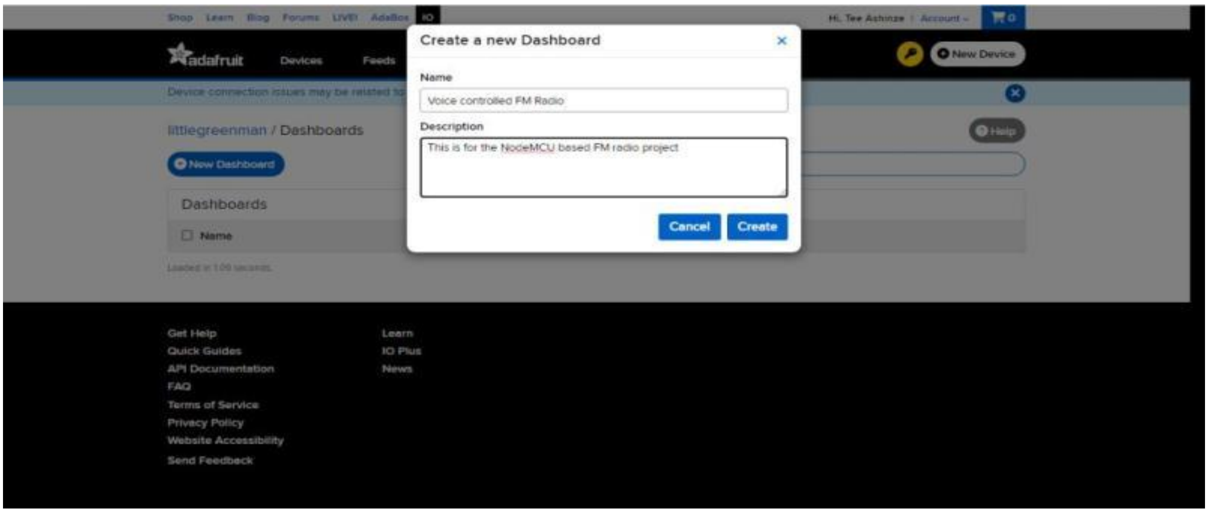


Fig 4.4 shows the step to give a name to the new Adafruit IO dashboard.

After the dashboard has been, get the Username and the Active Key from your account as it's required in the Arduino code. You can get that by clicking on the KEY icon.

After that, make three blocks: one Toggle Block, one Gauge Block, one Text Block.

The blocks are very important, as these blocks are responsible for the communication between Google assistant and the NodeMCU.

To make a block, you need to click on the gear sign in the upper right-hand corner.

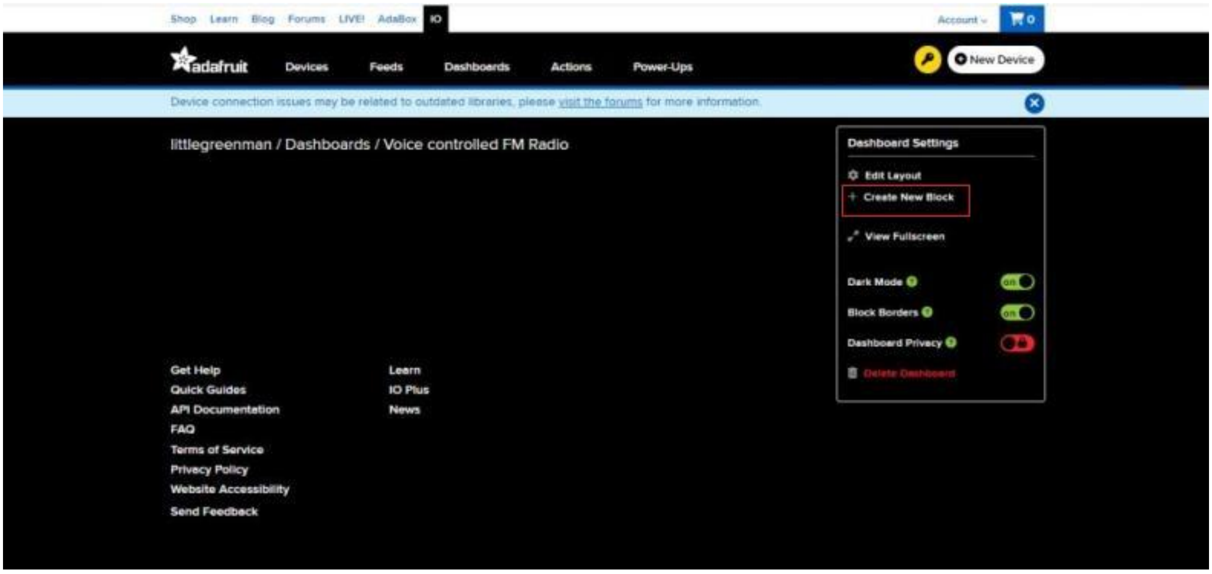


Fig 4.5 shows the steps to create a new Adafruit block.

Next, we are going to make the blocks.

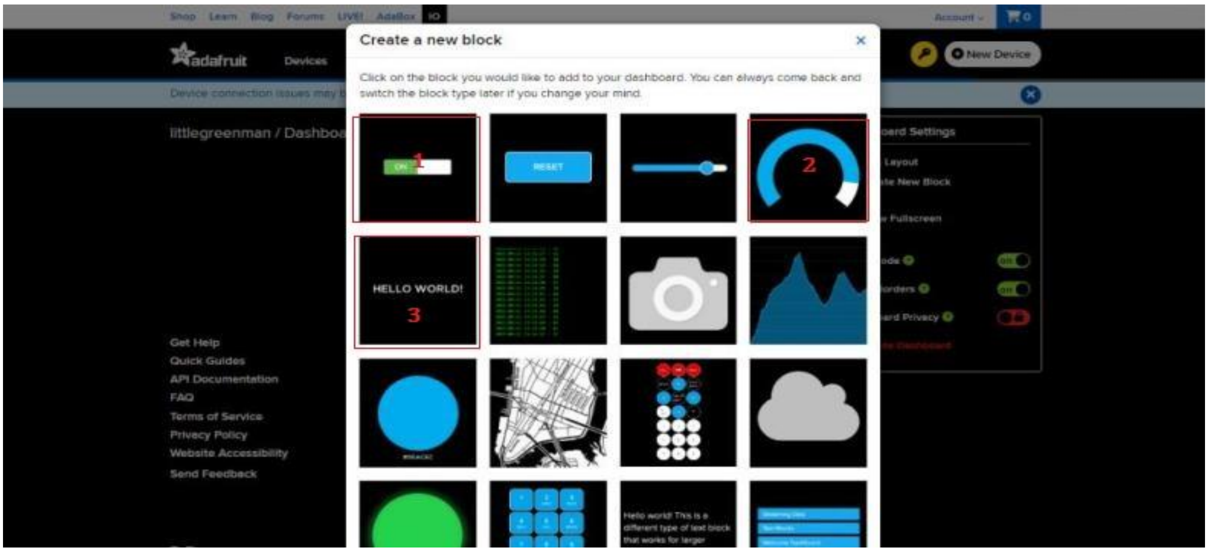


Fig 4.6 shows the specific Adafruit blocks to be used in this project.

Next, set up every block, for that, you need to tick on a particular block and click Next step.

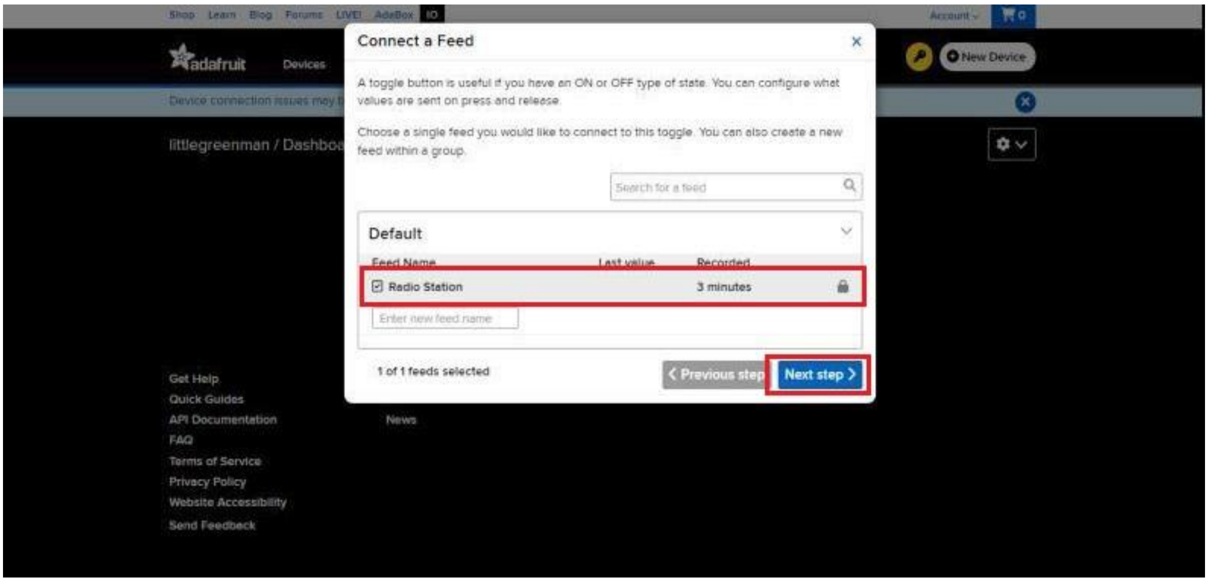


Fig 4.7 shows the setup process for a block.

For this project, there is no need to change any settings except for the toggle button. The text in the toggle button is in capital letterers, you need to make it small letterers and update the changes.

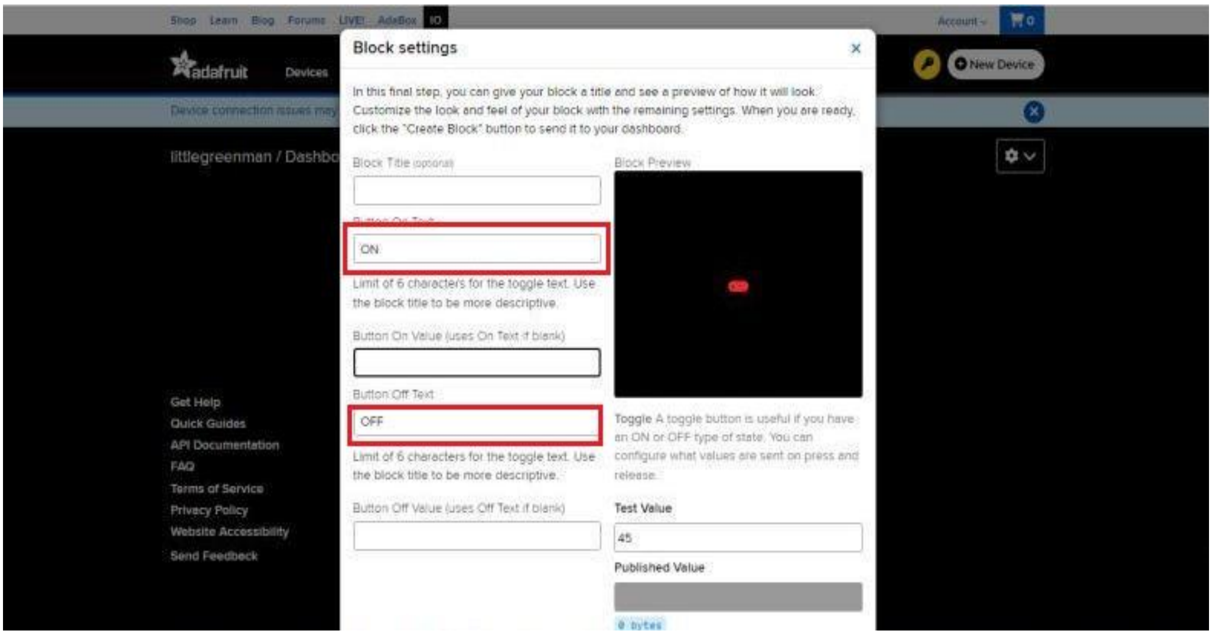


Fig 4.8 shows the setup process for a block.

The final screen looks like this-

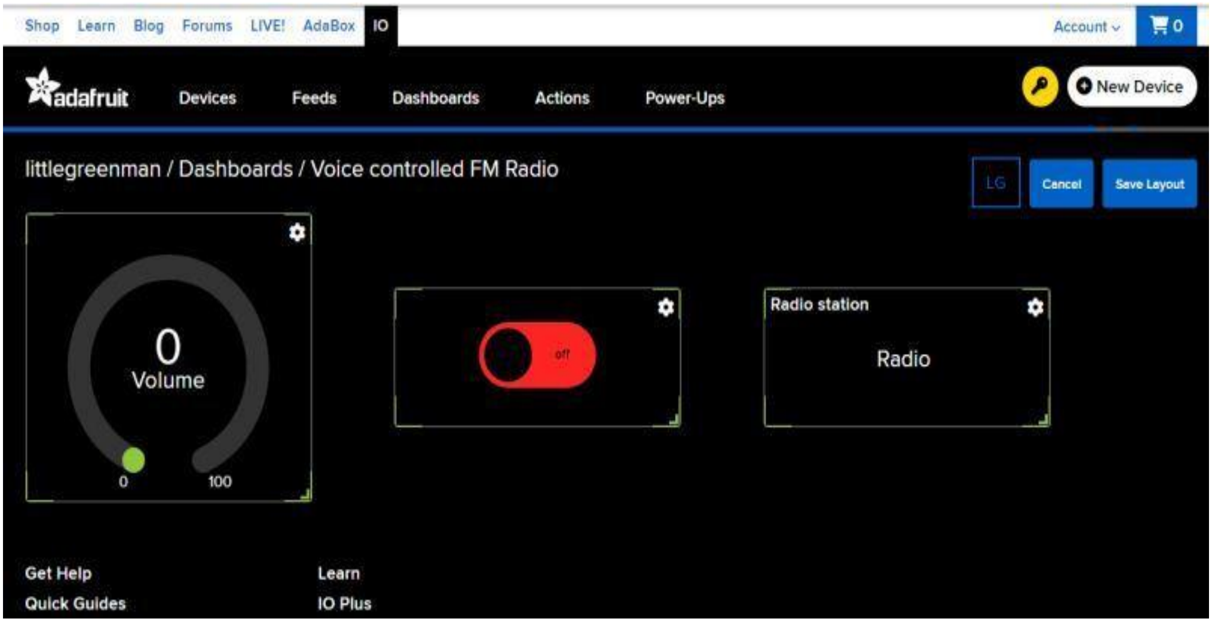


Fig 4.9 shows the Adafruit dashboard page, with the configured blocks.

4.2.2 Setting up an IFTTT Broker for the FM Radio

Create an IFTTT account on https://ifttt.com and login with your credentials.

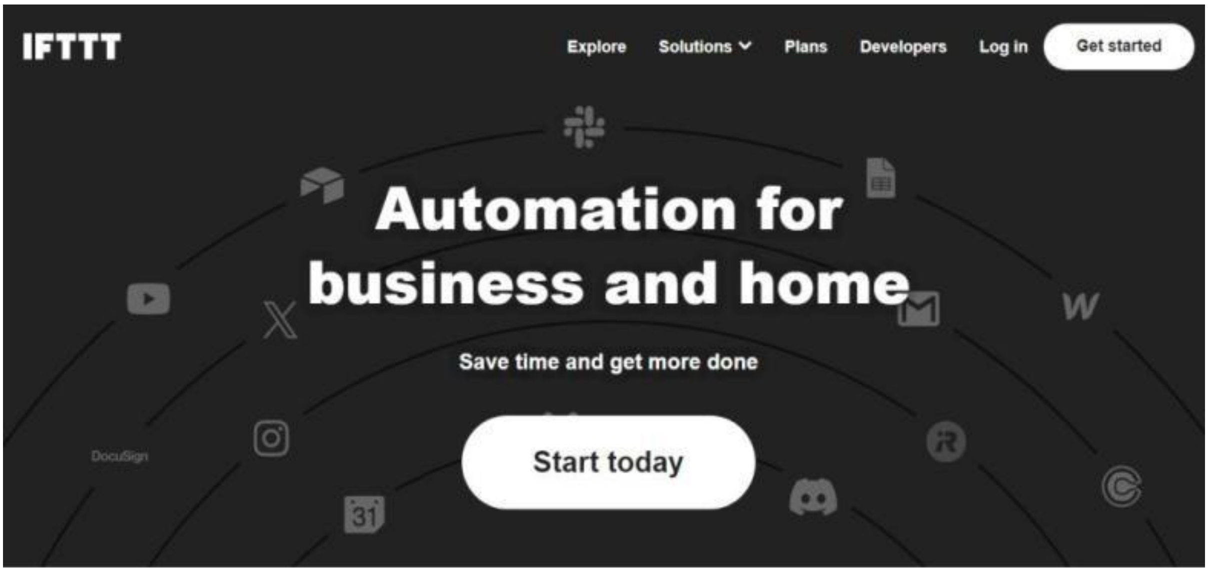


Fig 4.10 shows the IFTTT homepage.

Now, we create an Applet. For that, follow the steps below:

To make an applet, click on the *Create* icon.

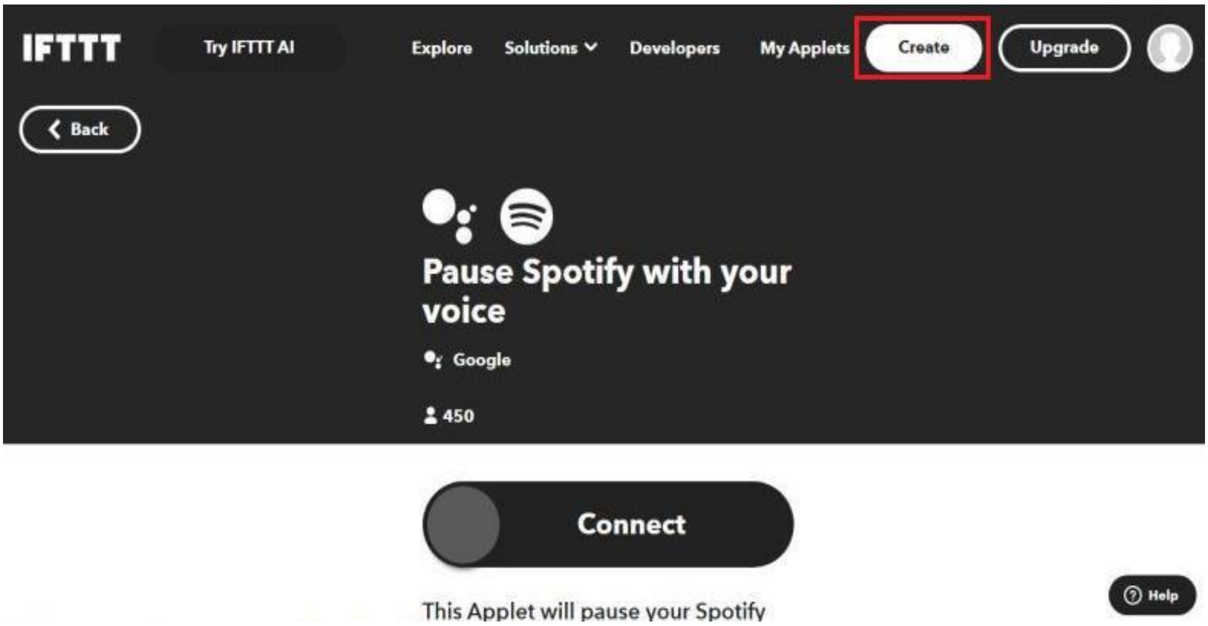


Fig 4.11 shows the screen to create an IFTTT applet.

In the create screen, Click the *Add* icon after If This.

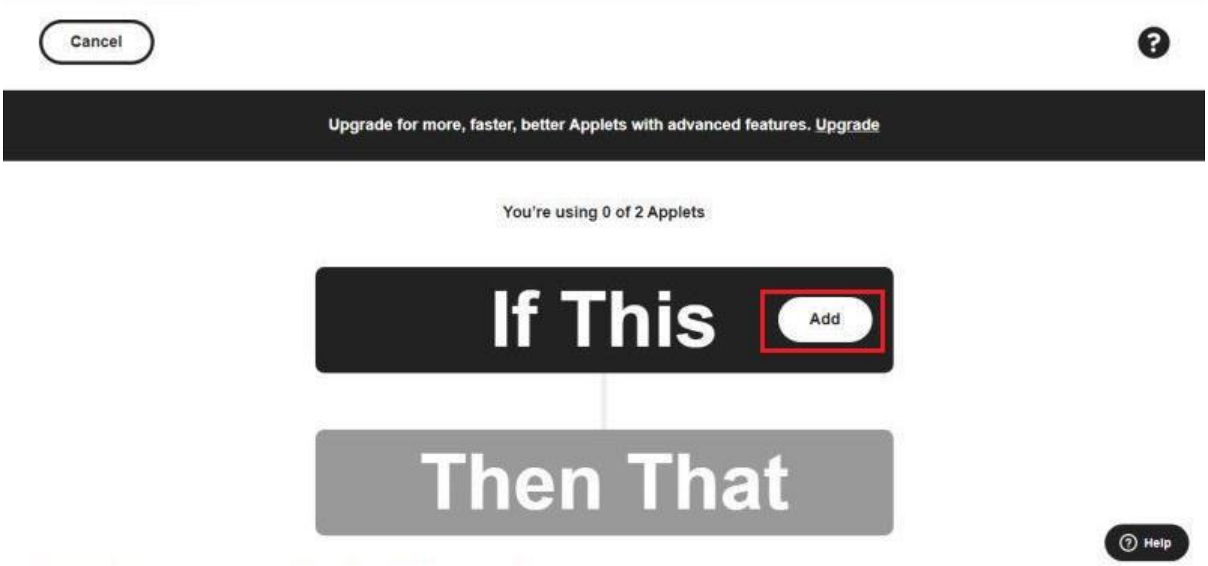


Fig 4.12 shows the applet setup process.

After that, you need to allow access to your Google account.

For that, you search for Google Assistant in the search bar and click on the Google Assistant icon.

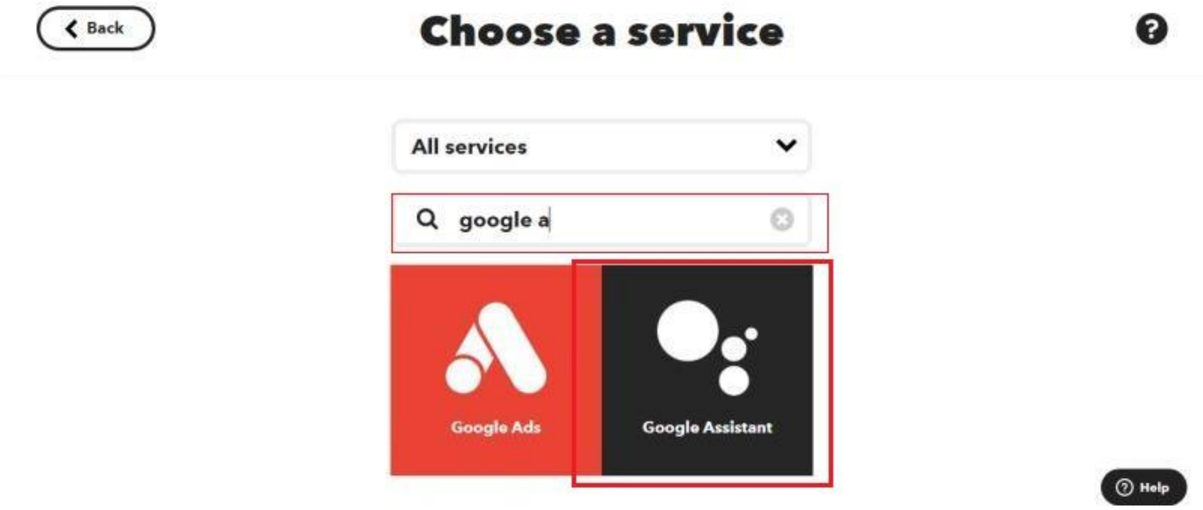


Fig 4.13 shows the applet setup process.

In the Next Screen, we must choose a trigger,

Remember, we made three blocks in the Adafruit IO Server; we need to make the triggers for those three blocks.

First, the Radio Station Block, for that, we select *Activate scene*.

.

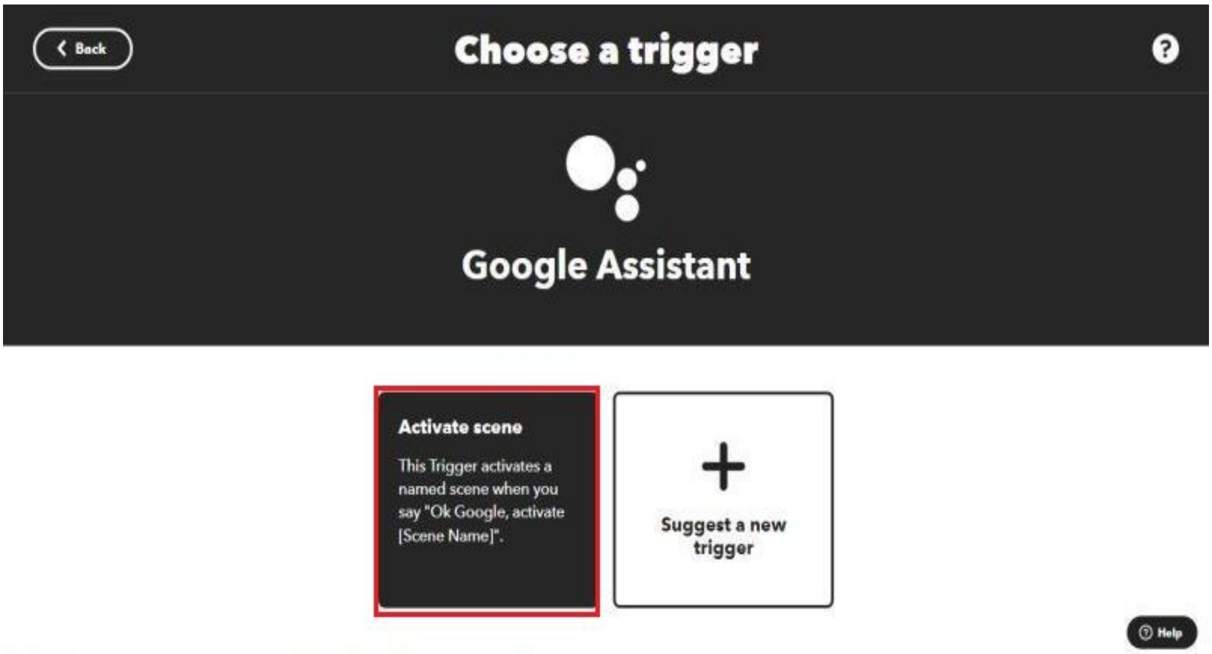


Fig 4.14 shows the applet setup process.

In the next screen, we type *Set the Radio Station to $* as scene name, then click *Create trigger*.

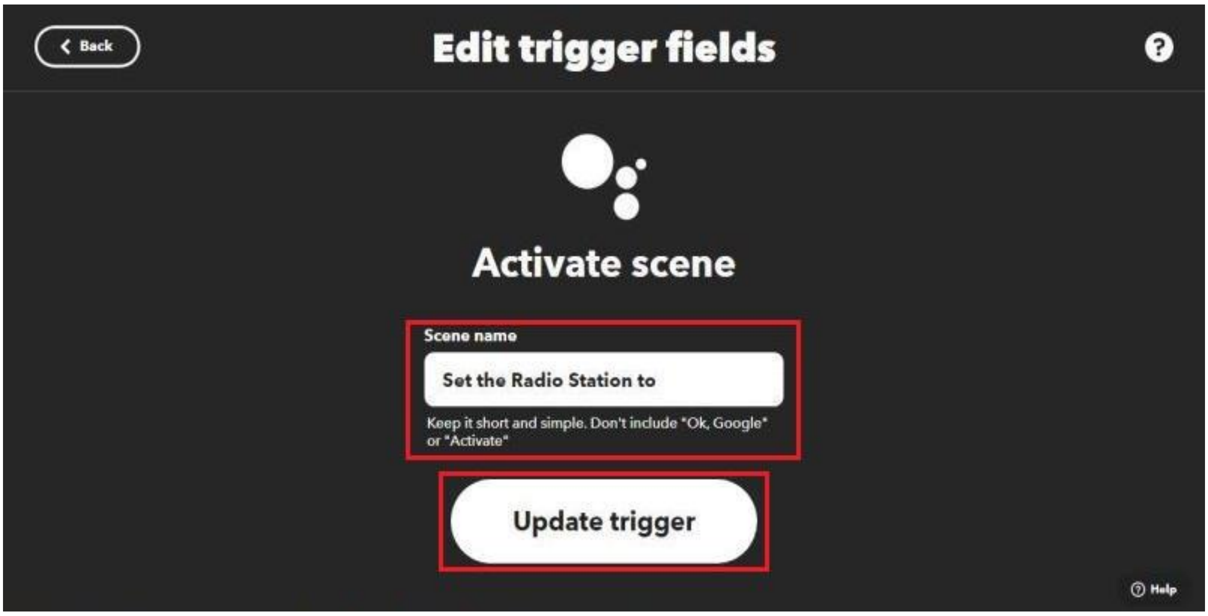


Fig 4.15 shows the applet setup process.

As we have completed the If part, we then proceed to the *Then* part. Click on Add to continue.

Fig 4.16 shows the applet setup process.

In the next screen presented, search for Adafruit, and click on the Adafruit icon.

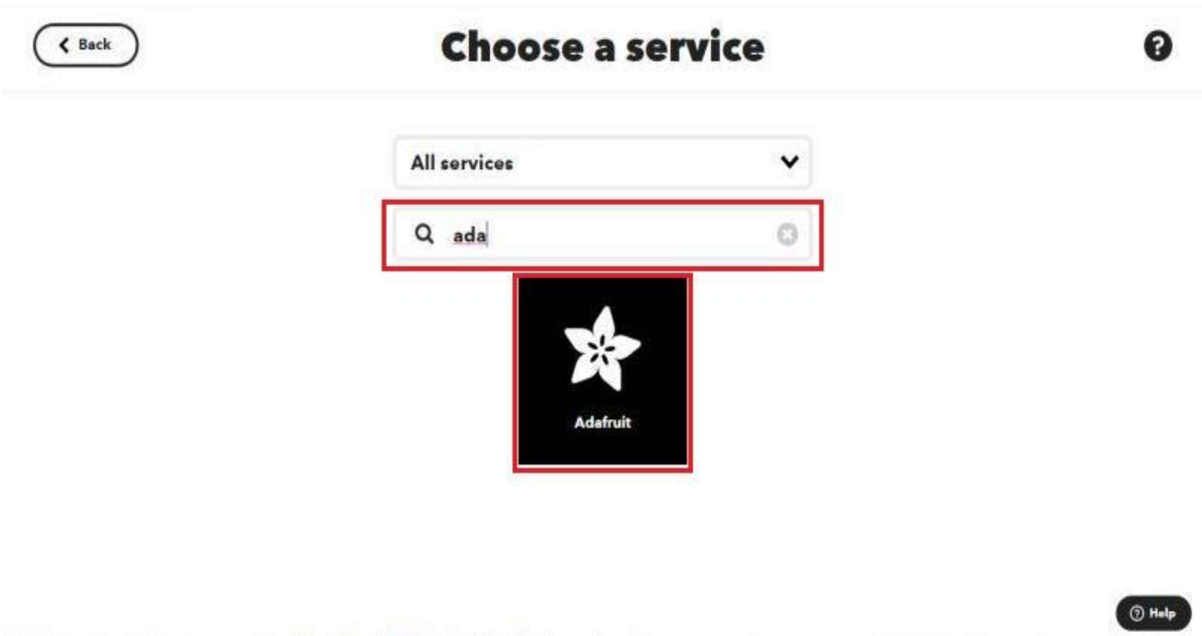


Fig 4.17 shows the applet setup process.

Next, click on *Send data to Adafruit IO*.

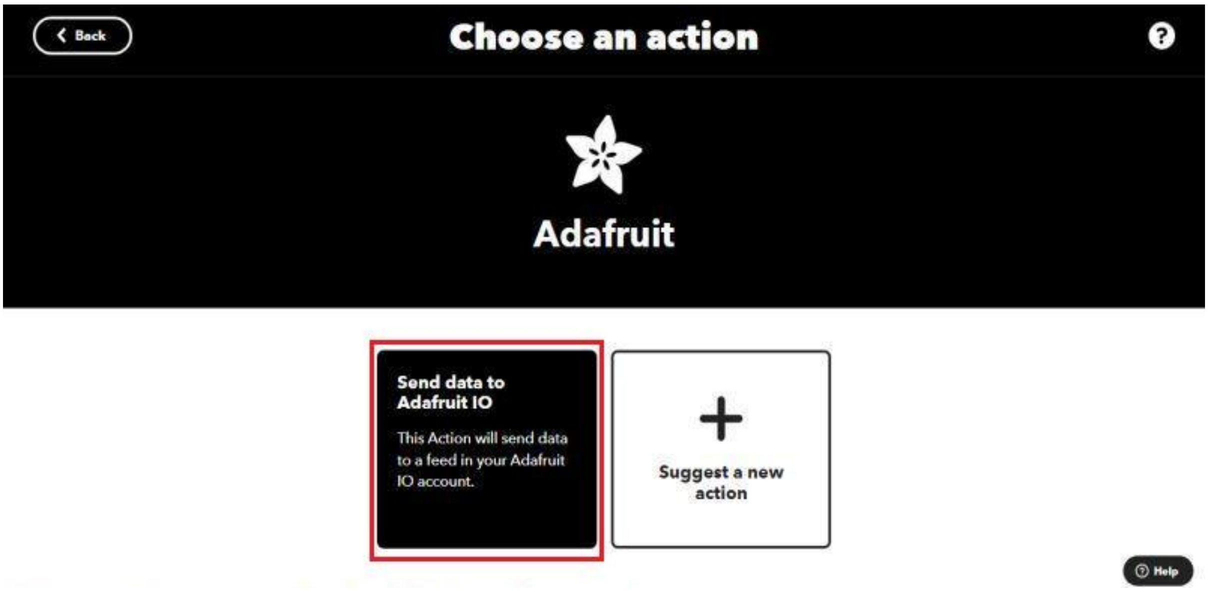


Fig 4.18 shows the applet setup process.

Next, click Connect to authorize your Adafruit account with IFTTT.

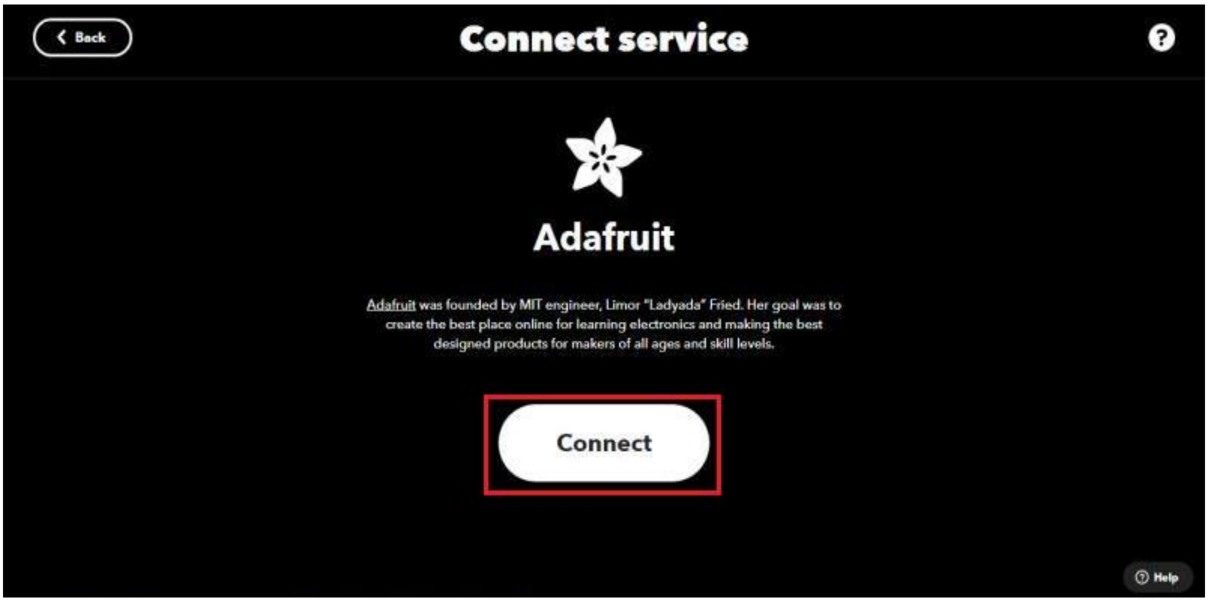


Fig 4.19 shows the applet setup process.

Then you will be presented with a dropdown of feeds created earlier in the Adafruit account. Choose any one of the feeds, and click on *Create action*, repeat this process for all three feeds (Radio_Station, Volume, Toggle_FM).

Fig 4.20 shows the applet setup process.

That concludes the IFTTT setup process; the final applet screens appear as follows.

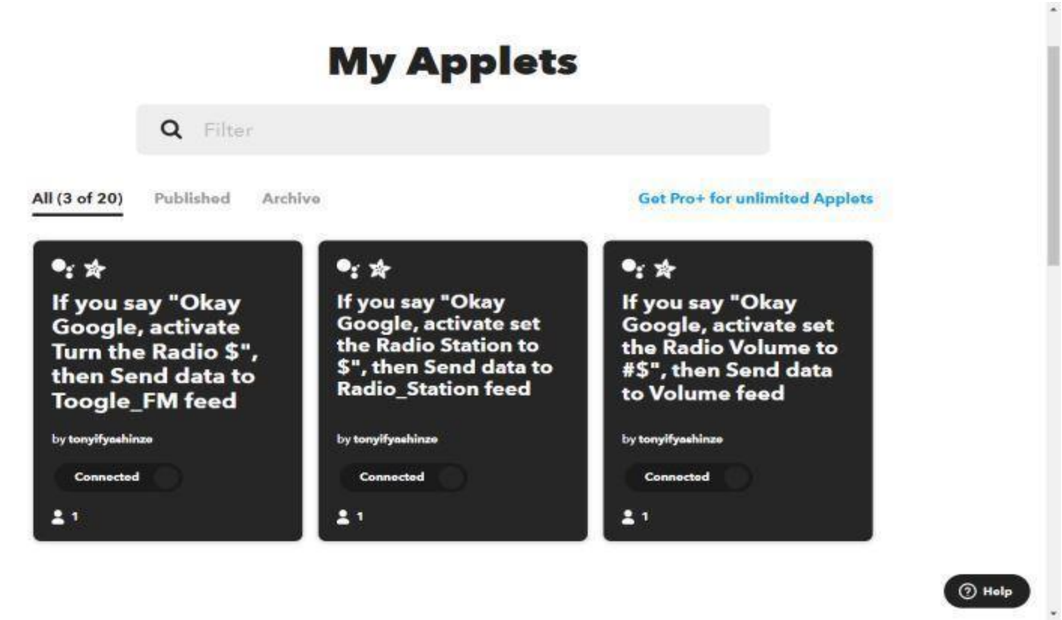


Fig 4.21 shows the final applet screen.

4.2.3 Writing code in Arduino IDE

The communication between the IC, Adafruit IO, IFTTT, and WIFI are all managed by the Arduino code. To write the code to program the NodeMCU, control the FM radio module, capture audio from the microphone, and process voice commands.

First, install and launch Arduino IDE.

Fig 4.22 shows the Arduino IDE start page.

Include all the required libraries, they are:

```
#include <ESP8266WiFi.h> //Library for the ESP8266 NodeMCU Board
#include <Wire.h> //Wire library is needed for I2C communication
#include <PT2258.h> //PT2258 Library is needed to communicate with the IC
#include <RDA5807M.h> // RDA5807M library is used to communicate with the RDA module
#include <Adafruit_MQTT.h> // Adafruit MQTT is need for MQTT
#include <Adafruit_MQTT_Client.h> // Adafruit MQTT is for the MQTT Client
```

Then, define the SSID and password for the WI-FI; this is the SSID and the PASSWORD of the router to be used.

```
const char* ssid = "CASTLE BLACK"; // SSID of your router
const char* password = "Blackpegasus"; // Password of Your Router
```

We define two Booleans and a variable. The Booleans are used to hold the communication status of the ICs, and the volume variable is used to set the volume level.

```
bool potStatus; // 1 when communication is established between the MCU and the IC
bool radioStatus; // 1 when communication is established between the MCU and the IC
int volume = 15; // default volume level with the IC starts with
```

Set up a GPIO pin named Relay_Pin to turn on or off the amplifier.

```
#define Relay_Pin 7 // This pin is used to turn on and off the radio
```

Next, we write all the necessary definitions to communicate with Adafruit IO.

```
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
#define AIO_USERNAME    "littlegreenman"    // Replace it with your username
#define AIO_KEY “aio_rWry521iHxbtKrhVlGc1JQZ8L4z5"
// Replace with your Project Auth Key
```

The below definitions FIX_BAND is a proprietary definition used by the library. The next defined statement sets the internal volume of the module.

```
#define FIX_BAND      RADIO_BAND_FM    //< The band will be tuned by this sketch is FM.
#define FIX_RADIO_VOLUME    6              ///< Default volume of the module.
```

Next, we make the required objects for the PT2258, the RDA5807M, and the Wi-Fi Client.

```
PT2258 digitalPot; // PT2258 Object
RDA5807M radio; //RDA5807M Object
WiFiClient client; //WiFiClient Object
```

Then set up the MQTT client class by passing in the Wi-Fi client and MQTT server and login details.

```
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
```

Then we need to subscribe to a Feed. If some values, some parameters change in the Adafruit server, the changes will be reflected here.

```
Adafruit_MQTT_Subscribe Radio_Station = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME"/feeds/Radio_Station"); // Methods used to subscribe to a Feed
Adafruit_MQTT_Subscribe Toggle_FM = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME"/feeds/Toggle_FM"); // Methods used to subscribe to a Feed
Adafruit_MQTT_Subscribe Volume = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME"/feeds/Volume"); // Methods used to subscribe to a Feed
```

Below is the function prototype for MQTT_connect() function.

```
void MQTT_connect();//Function Prototype for MQTT Connect
```

Then we begin our setup process. At first, we start the UART communication with the begin method.

```
Serial.begin(9600); //UART begin
Serial.println(); // adds an extra line for spacing
Serial.println();// adds an extra line for spacing
Next, we do all the usual thing to connect to WiFI
**************** all the usual things required for a WiFi connection***********************/
  Serial.print("connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
/**************** all the usual things required for a WiFi connection***********************/
```

Next, call the Wire.begin() method to instantiate an I2C connection & we call the Wire.setClock() method to fix the I2C frequency to 100KHz as it's the full speed of the PT2258 IC.

```
Wire.begin(); // begin the I2C starting sequence
Wire.setClock(100000); // setting the I2C clock to 100KHz
```

Next, call the init() method for both the PT2258 and the RDA5807 IC and hold the return status into the previously defined Booleans.

```
potStatus = digitalPot.init();
radioStatus = radio.init();
```

Next, we check if the MCU was able to communicate with the IC or not. We do this with two *if else* statements.

```
if (potStatus)  {
    Serial.println("Found PT2258 Device!");
  }
else{
    Serial.println("Failed to Initiate PT2258");
  }
  if (radioStatus)  {
    Serial.println("Found RDA5807M Device!");
  }
  else{
    Serial.println("Failed to Initiate RDA5807M");
  }
```

Next, call the subscribe method from the MQTT library. We will be notified by the MQTT server if any changes happened in our subscribed feeds.

```
mqtt.subscribe(&Radio_Station);  //Setup MQTT subscription for Radio_Station feed
mqtt.subscribe(&Toggle_FM); //Setup MQTT subscription for Toggle_FM feed
mqtt.subscribe(&Volume);  //Setup MQTT subscription for Volume feed
```

Next, we set the Relay_pin as output and the pin status to LOW.

```
pinMode(Relay_Pin, OUTPUT);
digitalWrite(Relay_Pin, LOW);
```

Next, set a predetermined radio volume, this parameter sets the internal volume of the RDA5807 IC, which marks the end of our setup process.

```
radio.setVolume(FIX_RADIO_VOLUME); //next we set the normalize radio volume
radio.setMono(false); // we do not want the chip to give mono output
radio.setMute(false); // we don't want the chip to go mute at start
```

We start the loop by calling the MQTT_connect() function which establishes a connection to the MQTT server.

In the MQTT connect function; we try three times to make a connection to the MQTT server. If it is successful, we get a Success message else we will get an Error message.

void MQTT_connect()

```
{
  int8_t ret; // 8 bit integer to store the retries
  // Stop if already connected.
  if (mqtt.connected()) {
    return;
  }
  Serial.print("Connecting to MQTT... ");
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);  // wait 5 seconds
    retries--;
    if (retries == 0) {
      // basically die and wait for WDT to reset me
      while (1);
    }
  }
  Serial.println("MQTT Connected!");
}
```

Next, start by creating a pointer to an Adafruit_MQTT_Subscribe object. We'll use this to determine which subscription was received.

```
Adafruit_MQTT_Subscribe *subscription;
```

Next, we wait for a subscription message.

mqtt.readSubscription(timeInMilliseconds) will listen to a certain time, for any messages coming from the MQTT server.

If it gets a message before timeout, it will reply with a pointer to the subscription or it will just time out and return 0. In that case, it will wait for 2 Sec.

```
while ((subscription = mqtt.readSubscription(20000)))
```

If a timeout occurs, the while loop fill fails. If not, we compare what subscription and will get our known subscriptions.

In this code, we do this for all three of our subscribed feeds.

```
if (subscription == &Toggle_FM)
if (subscription == &Radio_Station)
if (subscription == &Volume)
```

This section of the code is used to monitor & set the Toggle_FM feed.

```
if (subscription == &Toggle_FM) // is it a message from the Toggle_FM Feed
    {
      Serial.print(F("Got: "));
      Serial.println((char *)Toggle_FM.lastread); //print the Feed data just for debugging
      if (String((char *)Toggle_FM.lastread) == String("on")) // we compare the received data to a known parameter in this case we are expecting that "on" is coming from the sever
      {                                                          // but before we do that we have to make it a string which makes the comparison super easy
        digitalWrite(Relay_Pin, HIGH);// if we get a "on" string from the server we are making the D7 pin HIGH
      }
      if (String((char *)Toggle_FM.lastread) == String("off")) // again we are checking for the string off
      {
        digitalWrite(Relay_Pin, LOW);//if we get a "off" string from the server we are making the D7 pin LOW
      }
    }
```

This section of the code is used to monitor & set the Radio_Station feed.

```
if (subscription == &Radio_Station)
    {
      Serial.print(F("Got: "));
      Serial.println((char *)Radio_Station.lastread);
      if (String((char *)Radio_Station.lastread) == String("Hit FM")) // hear we are checking for the string Big FM
      {
        radio.setBandFrequency(FIX_BAND, 9590); // if the above condition is true we are setting the radio channel to 95.9MHz
      }
      // The above mentioned proces is continued below
      if (String((char *)Radio_Station.lastread) == String("Sparkling FM"))
      {
        radio.setBandFrequency(FIX_BAND, 9230);
      }
      if (String((char *)Radio_Station.lastread) == String("Correct FM"))
      {
        radio.setBandFrequency(FIX_BAND, 9730);
      }
    }
```

This section of the code is used to monitor & set the Volume feed.

```
if (subscription == &Volume) // // hear we are checking for the string Volume and it is an integer value in a string format
                                    // We must convert it back to an integer to change the volume with the help of the PT2258 IC
      Serial.print(F("Got: "));
      Serial.println((char *)Volume.lastread);
      volume = atoi((char *)Volume.lastread); // We are using the atoi() methode to convert a character pointer to a integer
     volume= map(volume,0,100,79,0); //map(value, fromLow, fromHigh, toLow, toHigh) // as the pt2258 only understand integer values in dB
                                      // we are maping the 0dB - 79dB value to 0% - 100%
      digitalPot.setChannelVolume(volume, 0); //after all that we are setting the volume for the channel 0 of the PT2258 IC
      digitalPot.setChannelVolume(volume, 1); //after all that we are setting the volume for the channel 1 of the PT2258 IC
    }
  }
```

4.3. Result

The Internet of things (IoT) based voice-controlled FM radio receiver project was successfully completed, creating a voice-control based mechanism using Arduino and Google Assistant. The project achieved all its objectives, including:

- To investigate the feasibility of using the Google Assistant platform for voice control and explore the potential of using Arduino for IoT-based applications.
- Designing and building the hardware and software components of the IoT-based voice-controlled FM radio.

To test the circuit, the following apparatus was used:

a) 240W ATX Power Supply Unit (PSU)
b) Two 2Ω 8W Speakers as a load
c) PAM8403 Dual channel amplifier
d) Phone to use Google Assistant

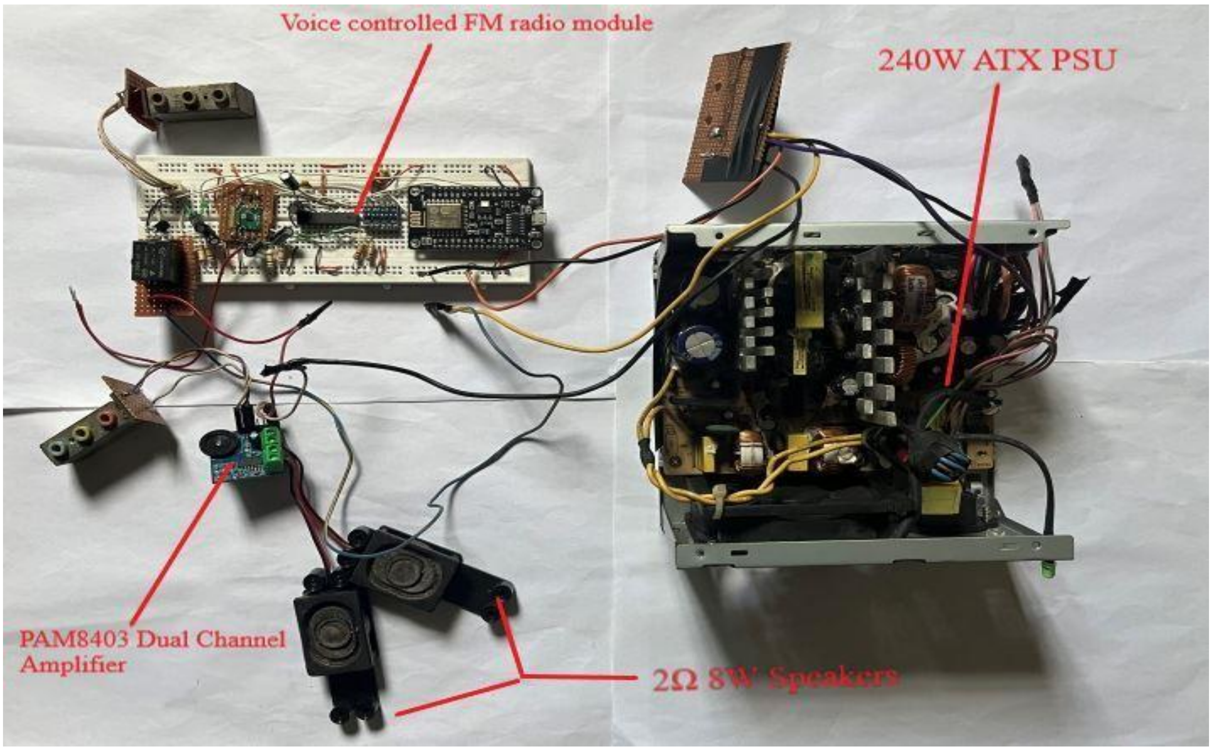


Fig 4.23 shows the circuit testing setup.

The system was tested with a variety of users and was found to be easy to use and reliable. The voice-controlled FM radio receiver system successfully allowed users to control the FM radio reciver using voice commands through Google Assistant. The key functionalities achieved were:

1. **Power Control:**
    - Voice command "Hey Google, turn on the radio" or "Hey Google, turn off the radio" toggled the power state of the FM radio.
2. **Frequency Control:**
    - Voice commands such as "Hey Google, set the frequency to 95.5" enabled users to tune the radio to a specific FM frequency.
3. **Volume Control:**
    - Users could adjust the volume using voice commands like "Hey Google, increase the volume" or "Hey Google, decrease the volume."

The following are some of the key analyses from the project:

- The Arduino platform was found to be well-suited for developing the voice-controlled FM radio system. It is easy to use, affordable, and has a wide range of available libraries and components.
- The Google Assistant platform was also found to be well-suited for the project. It is a powerful and versatile voice assistant platform that is easy to integrate with other systems.
- The system was able to achieve accurate voice recognition and response. This was achieved by using a combination of Google Assistant's voice recognition capabilities and the Arduino's ability to decode and execute voice commands
- The system was also able to handle multiple users simultaneously. This was achieved by using a combination of Google Assistant's ability to identify individual users and the Arduino's ability to control multiple devices.

CHAPTER SIX
CONCLUSION

6.1. Conclusion

The Internet of things (IoT)-based voice-controlled FM radio project showcased the successful integration of Google Assistant and Arduino to provide a user-friendly and accessible interface for controlling an FM radio using voice commands. The system demonstrated the potential of IoT in enhancing consumer electronics, particularly in improving accessibility for individuals with mobility challenges. The achieved functionalities of power control, frequency tuning, and volume adjustment through voice commands signify a leap towards intuitive interaction and a more enjoyable user experience.

6.2. Recommendation

Based on the project outcomes, the following recommendations are proposed:

1. **User Experience Enhancement:**

- Conduct usability testing to gather user feedback on the voice-controlled interface, identifying areas for improvement in terms of intuitiveness, responsiveness, and overall user satisfaction.
- Collaborate with individuals with diverse abilities to ensure the system meets accessibility requirements and to gather insights for further enhancements catering to different user needs

2. **Performance Optimization:**

- Explore optimization techniques to reduce latency and improve the system's responsiveness in interpreting voice commands, ensuring a near-real-time control experience.
- Investigate the feasibility of implementing a more efficient communication protocol between the Arduino and Google Assistant-enabled devices to enhance the system's speed and efficiency. \

6.3. Suggestions for future work

The IoT-based voice-controlled FM radio system could be extended to support additional functions, such as the ability to save and recall favorite stations, create custom playlists, and set alarms. The system could also be integrated with other

IoT devices, such as smart speakers and smart displays. This would allow users to control the radio using voice commands from anywhere in their home.

The system could also be made more accessible to people with disabilities by integrating with assistive technologies such as screen readers and speech recognition software.

To further advance the capabilities and applications of the IoT-based voice-controlled FM radio project, the following avenues for future research and development are suggested:

1. **Multi-language support:**

- Extend the project to support multiple languages for voice recognition and commands, allowing a broader audience to interact with the system effectively.

2. **Integration of Machine Learning:**

- Integrate machine learning algorithms to improve voice recognition accuracy and adapt to individual users' speech patterns, enhancing the system's overall performance and personalization.

3. **Mobile application integration:**

- Develop a dedicated mobile application to enhance user interaction by providing an intuitive graphical interface for controlling the radio in addition to voice commands.

4. **Internet radio streaming:**

- Extend the project to support streaming of internet radio stations, allowing users to access a wider range of radio content and further enhancing the system's versatility.

REFRENCES

[1]Dimitrios Serpanos and M. Wolf, *INTERNET-OF-THINGS IOT SYSTEMS : architectures, algorithms, methodologies*. S.L.: Springer, 2019.

[2]N. Vandome, *Smart Homes in Easy Steps : Master Smart Technology for Your Home*. Computer Step, 2018.

[3]Wikipedia Contributors, “FM broadcasting,” *Wikipedia*, Nov. 18, 2019. https://en.wikipedia.org/wiki/FM_broadcasting

[4]D. Norris, *The Internet of things : do-it-yourself projects with Arduino, Raspberry Pi, and BeagleBone Black*. New York: Mcgraw-Hill Education, 2015.

[5]K. Chenumalla, S. Gottam, P. Kusuma, and P. B. Shri, “Google Assistant Controlled Home Automation,” IEEE Vaagdevi Engineering College Student Branch, Bollikunta, Warangal, Telangana, 506005, India, Nov. 2019.

[6]Various Contributiors, “What is the Internet of Things (IoT)?,” *Oracle.com*, 2020. https://www.oracle.com/ng/internet-of-things/what-is-iot/

[7]A. Gillis, “What is iot (internet of things) and how does it work?,” *IoT Agenda*, Mar. 2022. https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

[8]AWS, “What is IoT? - Internet of Things Beginner’s Guide - AWS,” *Amazon Web Services, Inc.*, 2022. https://aws.amazon.com/what-is/iot/

[9]J. Zulick, “Latest Advances and Applications in the IoT Technology,” *Compare the Cloud*, Dec. 02, 2019. https://www.comparethecloud.net/articles/latest-advances-and-applications-in-the-iot-technology/

[10]Anonymous, “55+ emerging IoT technologies you should have on your radar (2022 update),” *IoT Analytics*, Apr. 06, 2022. https://iot-analytics.com/iot-technologies/

[11]E. Panel®, “Council Post: 16 New And Potential IoT Developments Tech Leaders Are Excited About,” *Forbes*, Mar. 29, 2023. https://www.forbes.com/sites/forbestechcouncil/2023/03/29/16-new-and-potential-iot-developments-tech-leaders-are-excited-about/?sh=60fc26171e16 (accessed Jun. 04, 2023).

[12]R. James, “8 interesting developments in IoT technology,” *TechRadar*, May 17, 2019. https://www.techradar.com/news/8-interesting-developments-in-iot-technology

[13]A. Maharaja and N. Ansari, “Voice Controlled Automation Using Raspberry Pi,” *SSRN Electronic Journal*, 2019, doi: https://doi.org/10.2139/ssrn.3366895.

[14]P. J. Rani, J. Bakthakumar, B. P. Kumaar, U. P. Kumaar, and S. Kumar, “Voice

controlled home automation system using Natural Language Processing (NLP) and Internet of Things (IoT)," *IEEE Xplore*, Mar. 01, 2017. https://ieeexplore.ieee.org/abstract/document/8261311 (accessed Feb. 28, 2021).

[15]D. V. Erickson, *Armstrong's Fight for FM Broadcasting: One Man Vs Big Business and Bureaucracy*. The University of Alabama Press, 1973. Available: https://worldradiohistory.com/BOOKSHELF-ARH/Biography/Armstrong's-Fight-For-FM-Broadcasting-Erickson-1973.pdf

[16]G. L. Frost, *Early FM Radio*. JHU Press, 2010.

[17]S. Harvey, "The History of FM Radio: the Search for Sensational Sound," *Radio Fidelity*, May 2019. https://radiofidelity.com/the-history-of-fm-radio/

[18]R. Martin, "Impacts of Internet of Things on Society," *MYTECHMAG*, Mar. 01, 2022. https://www.mytechmag.com/impacts-of-iot-on-society/

[19]L. Sharma, "The Rise of Internet of Things and Smart Cities," pp. 3–14, Dec. 2020, doi: https://doi.org/10.1201/9781003056751-1.

[20]B. K. Tripathy and J. Anuradha, *INTERNET OF THINGS IOT : technologies, Applications Challenges and solutions.* S.L.: Crc Press, 2020.

[21]L. M. Satapathy, S. K. Bastia, and N. Mohanty, "Arduino Based Home Automation Using Internet of Things (IoT)," *International Journal of Pure and Applied Mathematics*, vol. Volume 118, no. No. 17, 2018, Accessed: Jun. 06, 2023. [Online]. Available: http://www.ijpam.eu

[22]K. Bhanuabhiram, L. Tharun Kumar, and N. Srinivasan, "Google Assistant Controlled Home Automation," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 8, pp. 3259–3264, Aug. 2019, doi: https://doi.org/10.1166/jctn.2019.8173.

[23]A. Hernández, A. Gonzalez, and R. Benítez-Pérez, "Arduino: a Low-Cost Hardware for Internet of Things (IoT) Projects," IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), 2018.

[24]S. Dutta, R. Sharma, and M. S. Ali, "State of the Art: Speech Recognition Using Google Assistant," International Conference on Automation, Computational and Technology Management (ICACTM), 2019.

[25]R. Kumar and J. Prasad, "Internet of Things (IoT): a Literature Review," International Conference on Computing, Communication and Automation (ICCCA), 2019.

[26]H. Zhang, L. Zhang, and H. M. Alnuweiri, "Cognitive Radio for Internet of Things: a Survey," IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems

(MASS), 2015.

[27]F. Cicirelli, A. Guerrieri, C. Mastroianni, G. Spezzano, and A. Vinci, *The Internet of things for Smart Urban Ecosystems*. Cham: Springer, 2018.

[28]M. Brain, "How Radio Works," *HowStuffWorks*, Feb. 11, 2021. https://electronics.howstuffworks.com/radio.htm

[29]Wikipedia Contributors, "Radio," *Wikipedia*, Nov. 12, 2019. https://en.wikipedia.org/wiki/Radio

[30]electronics notes, "What Is FM, Frequency Modulation | Electronics Notes," *Electronics-notes.com*, 2019. https://www.electronics-notes.com/articles/radio/modulation/frequency-modulation-fm.php

[31]Various Contributors, "Radio Broadcasting Basics | Mobile FM Radio Broadcasting," *www.rfwireless-world.com*, 2012. https://www.rfwireless-world.com/Articles/radio-broadcasting-basics.html

[32]C. Woodford, "Radio and digital radio | How it works | AM and FM compared," *Explain that Stuff*, Aug. 24, 2018. https://www.explainthatstuff.com/radio.html

[33]A. L. Colina, A. Vives, M. Zennaro , A. Bagula, and E. Pietrosemoli, *Internet of Things (IoT) in 5 Days*, 1.1 ed. University of Western Cape, 2016. Available: https://github.com/marcozennaro/IPv6-WSN-book

[34]D. Nelson, *GOOGLE NOW A Guide to World's Most Powerful Personal Digital Assistant*. Gold Medal Solutions, Inc, 2014.

[35]Adeel Javed, *Building Arduino Projects for the Internet of Things : Experiments with real-world Applications*. New York: Apress, New York, Ny, 2016.

[36]J. Nussey, *Arduino for Dummies*, 2nd ed. Hoboken, New Jersey: John Wiley and Sons, 2018.

[37]"NodeMCU ESP8266 Specifications, Overview and Setting up," *Make-It.ca*. https://www.make-it.ca/nodemcu-details-specifications/

[38]Massimo Alioto and Springerlink (Online Service, *Enabling the Internet of Things : from Integrated Circuits to Integrated Systems*. Cham: Springer International Publishing, 2017.

[39]opensource, "What is an Arduino?," *Opensource.com*, 2019. https://opensource.com/resources/what-arduino

[40]N. Vatan, A. Sharma, and Dr. Sandip. Goyal, "IoT Standards and Applicability to Human Life," *IJITEE*, vol. 8, no. 10, pp. 3147–3153, Aug. 2019, doi: https://doi.org/10.35940/ijitee.j9512.0881019.

[41]A. A. and Y. Gambo, “Construction and Implementation of Smart Home System,” *International Journal of Computer Applications*, vol. 176, no. 12, pp. 15–17, Apr. 2020, doi: https://doi.org/10.5120/ijca2020920031.

[42]Mokh. Sholihul Hadi, M. Fakih, Bhima Satria Rizki S, I. Ari, None Sujito, and M. Irvan, “IoT Based Cytokine Storm Symptoms Early Warning System for Covid-19 Self-Isolating Patient,” *2022 2nd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, Nov. 2022, doi: https://doi.org/10.1109/ice3is56585.2022.10010303.

[43]Chynna Bhciu and S. Nita, “IOT Security against Malware Detection: a Proposed Framework,” Apr. 2023, doi: https://doi.org/10.31219/osf.io/c9sur.

[44]G. Alexakis, S. Panagiotakis, A. Fragkakis, E. Markakis, and K. Vassilakis, “Control of Smart Home Operations Using Natural Language Processing, Voice Recognition and IoT Technologies in a Multi-Tier Architecture,” *Designs*, vol. 3, no. 3, p. 32, Jul. 2019, doi: https://doi.org/10.3390/designs3030032.

[45]M. Gummineni and Polipalli, Trinatha Rao, “Preliminary Step for Implementing Cognitive Internet of Things Through Software-Defined Radio,” *SN Computer Science*, vol. 2, Art. no. 4, 2021.

[46]D. Ye, “Smart Energy Consumption of IoT With Millimeter-Wave Cognitive Radiofor 5G Cellular Network,” Research Square Platform LLC, 2021.

[47]E. Kadusic, N. Zivic, C. Ruland, and N. Hadzajlic, “A Smart Parking Solution by Integrating NB-IoT Radio Communication Technology into the Core IoT Platform,” *Future Internet*, vol. 14, Art. no. 8, 2022.

[48]P. P, “A Review on IOT Techniques for Automating Devices,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, Art. no. 4, 2015.

[49]Nourildean, Shayma Wail, Hassib, Mustafa Dhia, and Y. A. Mohammed, “Internet of things based wireless sensor network: a review,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, Art. no. 1, 2022.

[50]A. M. A, A. M. M, A. A. O, and A.-Q. F. S, “Design and Development of Voice-Enabled IoT Devices for Smart Home Automation,” *IEEE Access*, vol. 16226–16239, 2017.

[51]Sabita, D. G. A and P. J. L, “Design and Implementation of Voice Based Home Automation and Security System Using Internet of Things (IoT),” *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1–7,

2021.

[52]A. Sharma and B. N. Y, “IoT based Home Automation and Voice Controlled Devices,” *2021 8th International Conference for Convergence in Technology (I2CT)*, vol. 1–4, 2021.

[53]R. S. M and H. M. S, “Design and Implementation of Voice Controlled Smart Wheel Chair for Physically Disable People,” *IEEE 5th International Conference on Computer and Communication Systems (ICCCS)*, vol. 98–101, 2021.

[54]P. V. K and T. P. R, “Voice Controlled Wireless Wheelchair System,” *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, vol. 974–979, 2020.

[55]J. Li, Y. Zhang, and S. Jiang, “Application of IoT and Voice Control Technology in Smart Campus,” *IEEE Internet of Things Journal*, vol. 19788–19798, Art. no. 8(24), 2021.

[56]Khan, M. M. Alam, L. Moullec, A. Kuusik, S. Parand, and C. Verikoukis, “An Empirical Modeling for the Baseline Energy Consumption of an NB-IoT Radio Transceiver,” *IEEE Internet of Things Journal*, vol. 8, Art. no. 19, 2021.

[57]Talal Alsharari, Dr, S. Saleh, Abdulaziz Fatani, Dr, and Ismail, “Significant role of internet of things (IoT) for designing smart home automation and privacy issues,” *International Journal of Engineering & Technology*, vol. 9, Art. no. 2, 2020.

[58]I. Surya, P. Dody, F. Mahardika, A. Wijanarko, and Handani, Sitaresmi Wahyu, “IoT-Based Smart Air Conditioner as a Preventive in the Post-COVID-19 Era: A Review,” *Journal of Robotics and Control (JRC)*, vol. 4, Art. no. 1, 2023.

[59]NGUYEN Nang An, TRAN Thanh Trung, TRAN Kim Hoan, NGUYEN Tuan Anh, and PHAM Minh Doanh, “Integrate Deep Neural Network and Support Vector Machine to Improve the Quality of Voice Processing in Internet of Things Devices,” *VUJS*, vol. 52, no. 1A, pp. 5–16, Mar. 2023, doi: https://doi.org/10.56824/vujs.2023a005.

[60]D. A. Mellis and L. Buechley, “Scaffolding Creativity with open-source Hardware,” Nov. 2011, doi: https://doi.org/10.1145/2069618.2069702.

[61]M. R. Thakur, *NodeMCU ESP8266 Communication Methods and Protocols Programming with Arduino IDE*. 2018.

[62]“RDA5807M Stereo FM Radio Module 50MHz-115MHz – Art of Circuits,” *Artofcircuits*. https://artofcircuits.com/product/rda5807m-stereo-fm-radio-module-50mhz-115mhz (accessed Jun. 10, 2023).

[63]D. Das, “Digital Audio Volume Control Circuit using PT2258 IC and Arduino,”

circuitdigest.com, Jun. 18, 2020. https://circuitdigest.com/microcontroller-projects/digital-audio-volume-control-circuit-using-pt2258-ic-and-arduino (accessed Jun. 10, 2023).

[64]Arduino, "Arduino Integrated Development Environment (IDE) v1 | Arduino Documentation," *docs.arduino.cc*. https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics

[65]D. Myers, "IoT & Google Assistant," *Medium*, Jan. 22, 2019. https://medium.com/google-developers/iot-google-assistant-f0908f354681

[66]"Welcome to Adafruit IO," *Adafruit Learning System*. https://learn.adafruit.com/welcome-to-adafruit-io/libraries

[67]Arduino, "SoftwareSerial Library | Arduino Documentation," *docs.arduino.cc*, May 16, 2023. https://docs.arduino.cc/learn/built-in-libraries/software-serial

[68]IFTTT, "Documentation - Build Your Integration," *IFTTT*. https://ifttt.com/docs