Major Project Report

On

“Voice Based Home Automation”

By

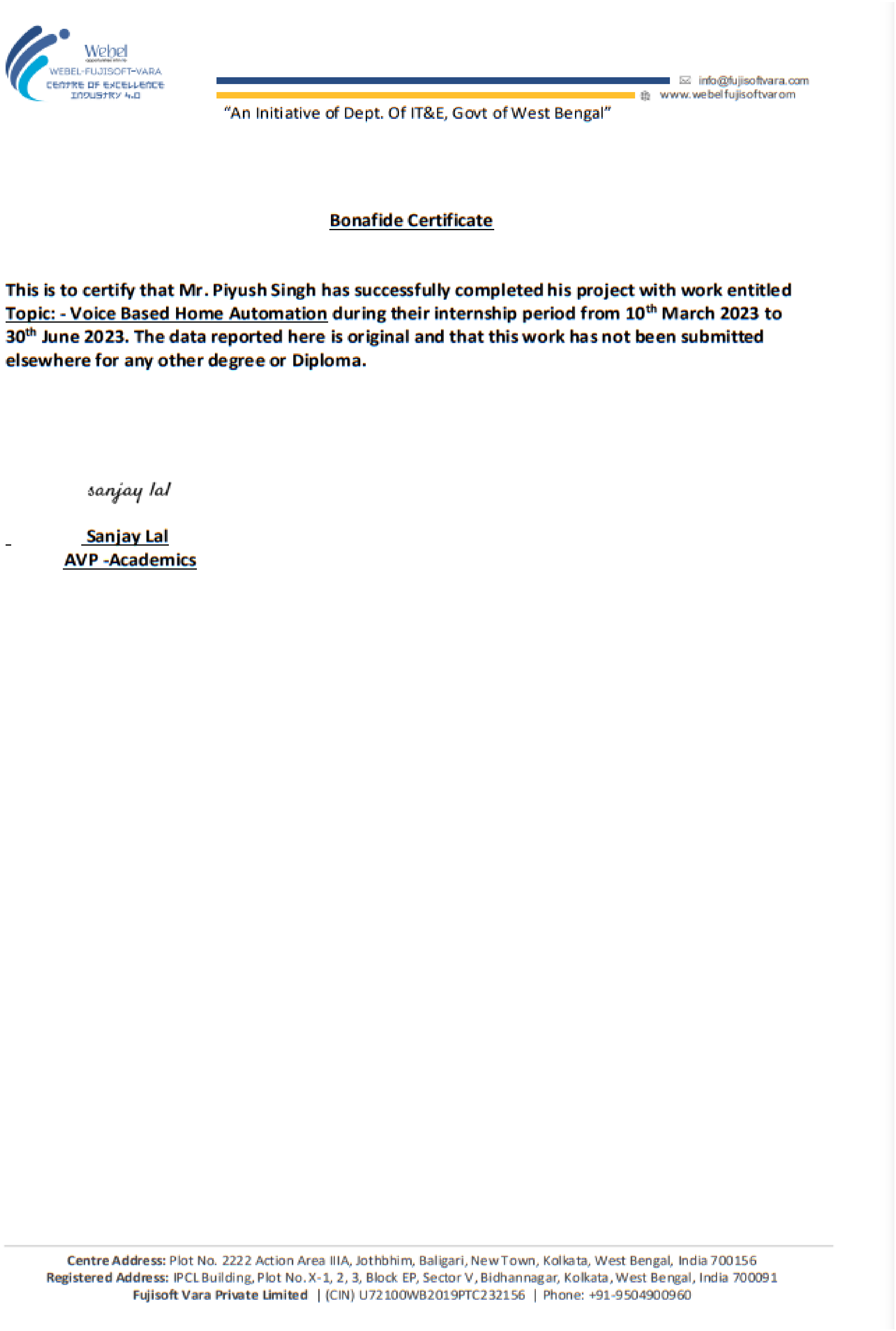
Piyush Singh (201900151)
Anu Kumari Gupta (201900098)
(Group ID: TNP82)

In partial fulfillment of requirements for the award of degree in
Bachelor of Technology in Computer Science and Engineering
(2023)

Under the Project Guidance of
Mrs. Riti Chatterjee (External Guide, Webel Fuji Soft Vara)

Mr. Tawal Kumar Koirala (Internal Reviewer, Assistant Professor I)
&
Ms. Tanuja Subba (Internal Reviewer,Assistant professor II)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY
(A constituent college of Sikkim Manipal University) MAJITAR,
RANGPO, EAST SIKKIM – 737136



Webel

WEBEL-FUJISOFT-VARA
CENTRE OF EXCELLENCE
INDUSTRY 4.0

✉ info@fujisoftvara.com
www.webelfujisoftvarom

"An Initiative of Dept. Of IT&E, Govt of West Bengal"

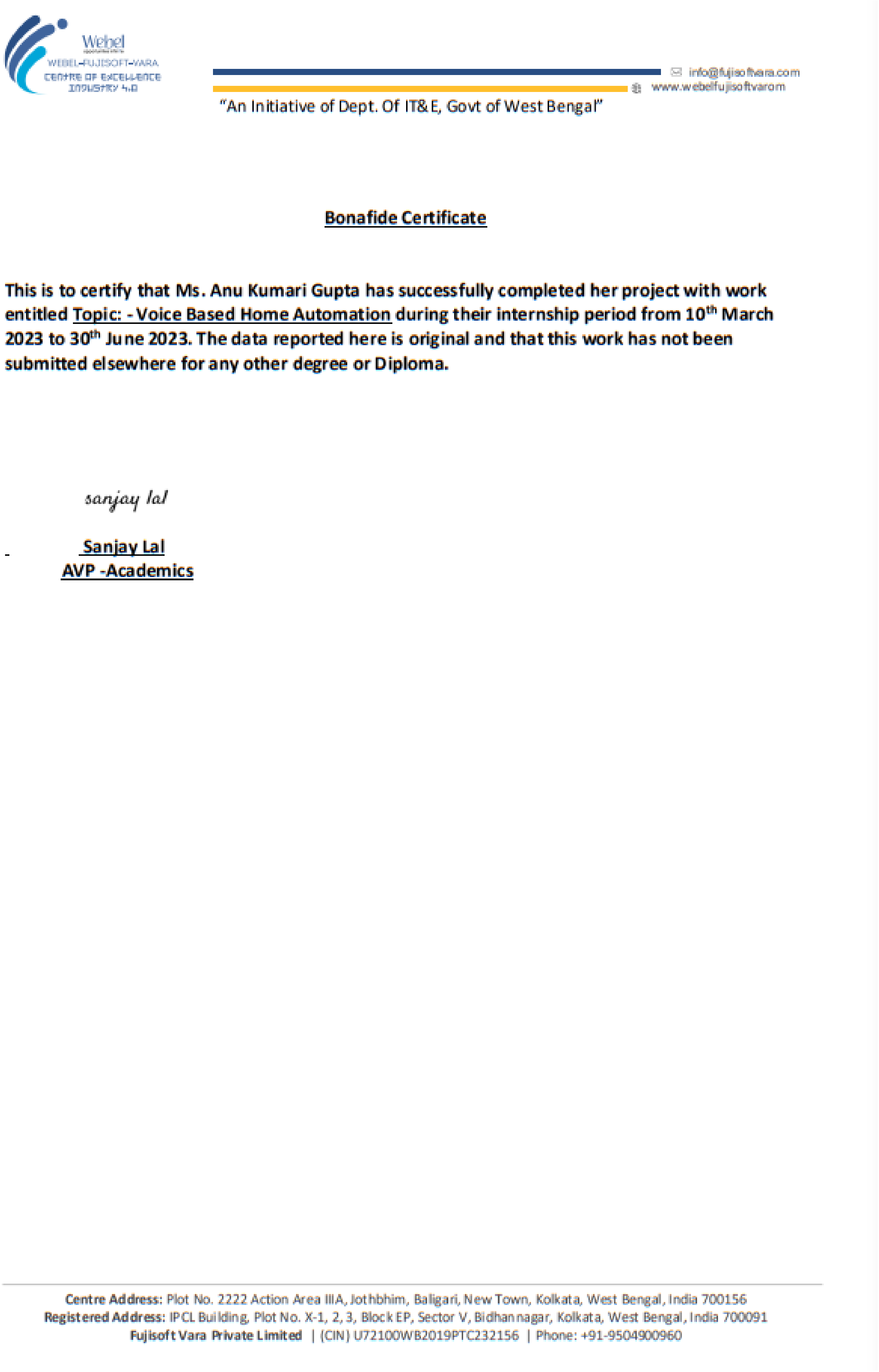
Bonafide Certificate

This is to certify that Mr. Piyush Singh has successfully completed his project with work entitled Topic: - Voice Based Home Automation during their internship period from 10th March 2023 to 30th June 2023. The data reported here is original and that this work has not been submitted elsewhere for any other degree or Diploma.

sanjay lal

Sanjay Lal
AVP -Academics

Centre Address: Plot No. 2222 Action Area IIIA, Jothbhim, Baligari, New Town, Kolkata, West Bengal, India 700156
Registered Address: IPCL Building, Plot No. X-1, 2, 3, Block EP, Sector V, Bidhannagar, Kolkata, West Bengal, India 700091
Fujisoft Vara Private Limited | (CIN) U72100WB2019PTC232156 | Phone: +91-9504900960



Webel

WEBEL-FUJISOFT-VARA
CENTRE OF EXCELLENCE
INDUSTRY 4.0

✉ info@fujisoftvara.com
www.webelfujisoftvarom

"An Initiative of Dept. Of IT&E, Govt of West Bengal"

<u>Bonafide Certificate</u>

This is to certify that Ms. Anu Kumari Gupta has successfully completed her project with work entitled <u>Topic: - Voice Based Home Automation</u> during their internship period from 10th March 2023 to 30th June 2023. The data reported here is original and that this work has not been submitted elsewhere for any other degree or Diploma.

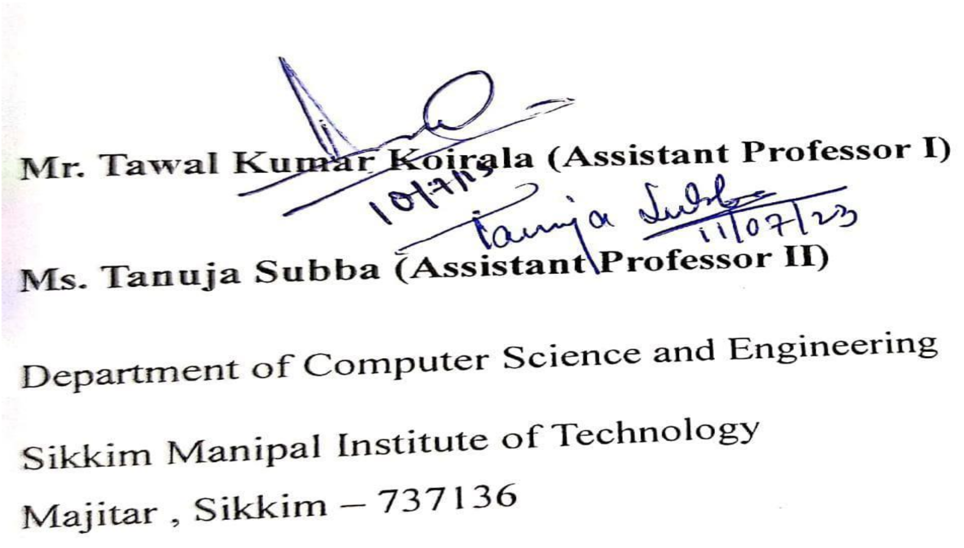
sanjay lal

<u>Sanjay Lal</u>
<u>AVP -Academics</u>

Centre Address: Plot No. 2222 Action Area IIIA, Jothbhim, Baligari, New Town, Kolkata, West Bengal, India 700156
Registered Address: IPCL Building, Plot No. X-1, 2, 3, Block EP, Sector V, Bidhannagar, Kolkata, West Bengal, India 700091
Fujisoft Vara Private Limited | (CIN) U72100WB2019PTC232156 | Phone: +91-9504900960

Project Review Certificate

This is to certify that the work recorded in this project report entitled **“Voice Based Home Automation”** has been jointly carried out by **Anu Kumari Gupta (201900098) and Piyush Singh (201900151)** of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Major Project Viva Examination.



Mr. Tawal Kumar Koirala (Assistant Professor I)

10/7/23

Tanuja Subba 11/07/23

Ms. Tanuja Subba (Assistant Professor II)

Department of Computer Science and Engineering

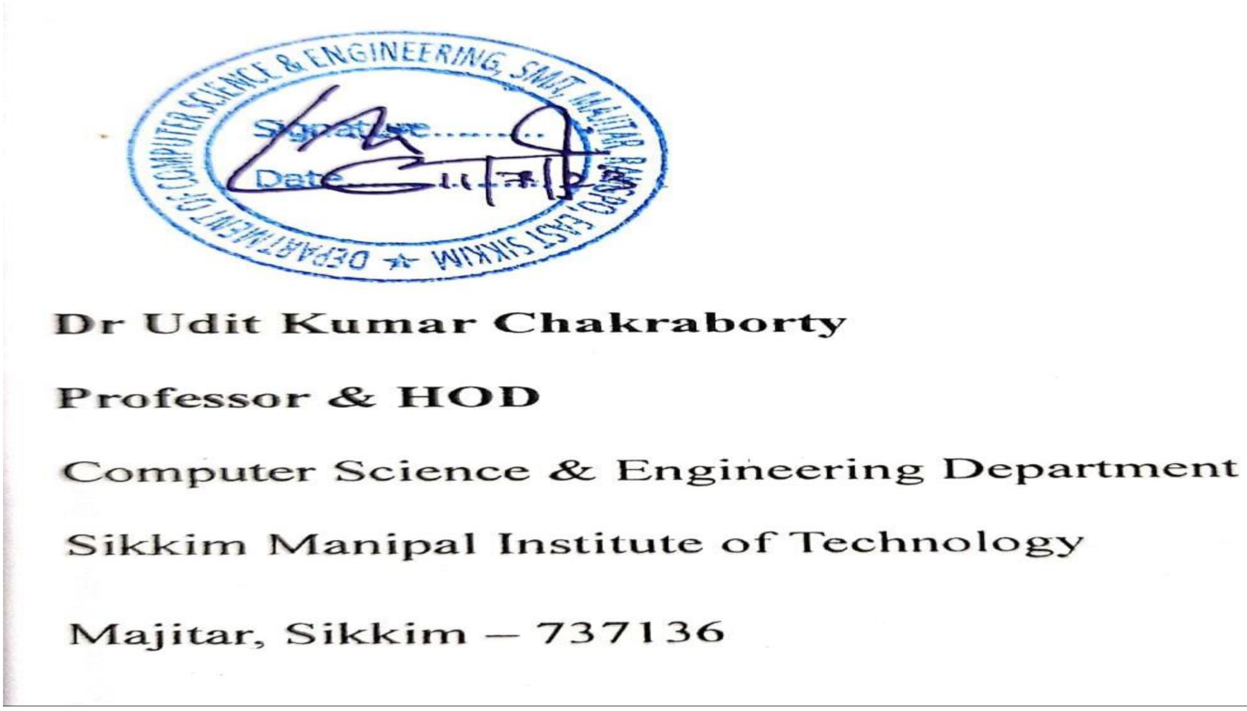
Sikkim Manipal Institute of Technology

Majitar , Sikkim – 737136

Certificate of Acceptance

This is to certify that the below mentioned student(s) of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) has/have worked under the supervision of **Mr. Tawal Kumar Koirala (Assistant Professor I) & Ms. Tanuja Subba (Assistant Professor II)** , of Department of Computer Science and Engineering from **10th March 2023 to 30th June 2023** on the project entitled **"Voice Based Home Automation".** The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student(s)	Project Venue
201900151	Piyush Singh	SMIT
201900098	Anu Kumari Gupta	SMIT



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING, SMIT, MAJITAR, RANGPO, EAST SIKKIM ★

Signature............

Date......11.7.23

Dr Udit Kumar Chakraborty

Professor & HOD

Computer Science & Engineering Department

Sikkim Manipal Institute of Technology

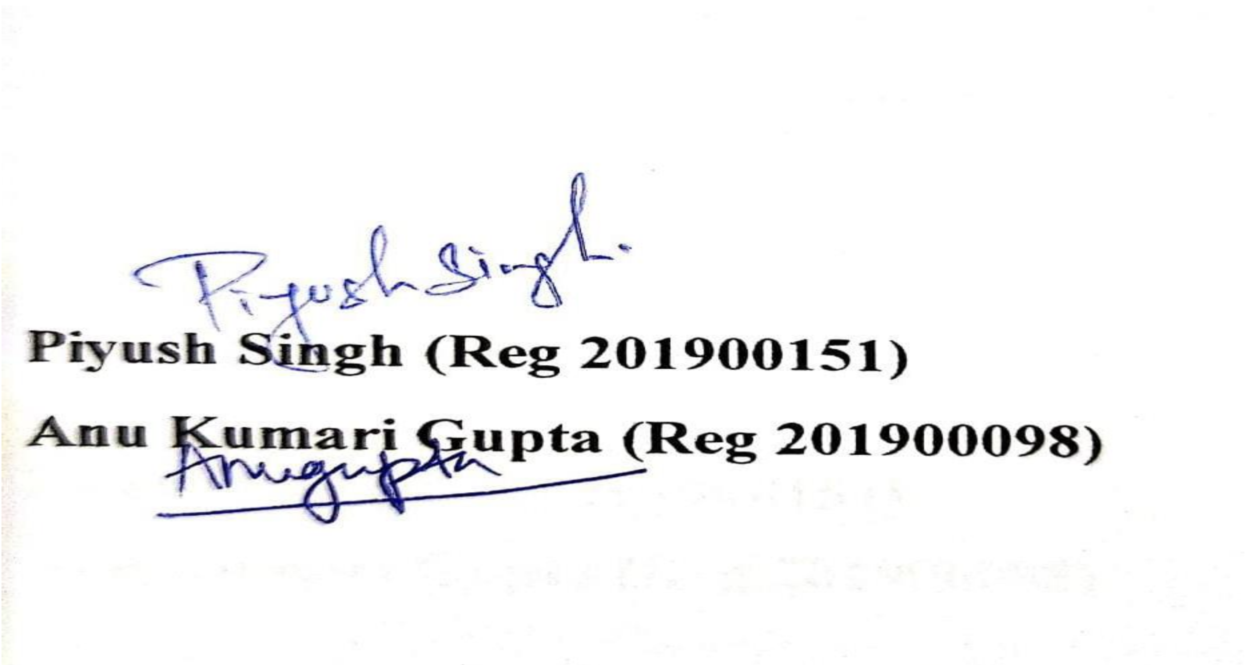
Majitar, Sikkim – 737136

Declaration

We the undersigned, hereby declare that the work recorded in this project report entitled **"Voice Based Home Automation"** in partial fulfilment for the requirements of award of B.Tech. (CSE) from "**SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY"** (A constituent college of Sikkim Manipal University) is a faithful and bonafide project work carried out at **"Webel Fujisoft Vara, Kolkata"** under the supervision and guidance of **Mr. Tawal Kumar Koirala (Assistant Professor I) & Ms. Tanuja Subba (Assistant Professor II)** of Department of Computer Science and Engineering.

The results of this investigation reported in this project have so far not been reported for any other Degree or any other technical forum.

The assistance and help received during the investigation have been duly acknowledged



Piyush Singh.

Piyush Singh (Reg 201900151)

Anu Kumari Gupta (Reg 201900098)

Anugupta

Acknowledgement

We wish to express our sincere thanks to External reviewer **Mrs. Riti Chatterjee, Senior Faculty of Webel Fujisoft Vara** for providing us an opportunity to carry out project work on **"Voice Based Home Automation"** and their unlisted encouragement and guidance carrying out this project work.

We sincerely thank **Mr. Tawal Kumar Koirala (Assistant Professor I) & Ms. Tanuja Subba (Assistant Professor II),** of Department of Computer Science and Engineering., Sikkim Manipal Institute of Technology for her guidance and encouragement in carrying out this project till the completion.

We would like to express our sincere thanks to **(Prof.) Dr. Udit Kumar Chakraborty, HOD, Computer Science and Engineering Department** for allowing us to carry out our project from and valuable support and guidance during the project period.

We would like to express our humble gratitude to **Dr. Biswaraj Sen(Professor), Mr. Santanu Kumar Misra(Associate Professor), Mr. Saurav Paul(Assistant Professor), and Mrs. Chitrapriya N.(Assistant Professor)** Project Coordinators, Computer Science and Engineering Department, Sikkim Manipal Institute of Technology for their unlisted encouragement and their timely support and guidance till the completion of the project work.

Lastly, we wish to avail myself of this opportunity, express a sense of gratitude and love to all our teachers, staff of the department of Computer Science and Engineering Department, friends and fellow for their support and help.

Piyush Singh (Reg 201900151)
Anu Kumari Gupta (Reg 201900098)

Document Control Sheet

1	Report No	CSE/Major Project/Internal /B.Tech/SA6/2023
2	Title of the Report	Voice Based Home Automation
3	Type of Report	Technical
4	Author(s)	Anu Kumari Gupta (Reg. No.-201900098) Piyush Singh (Reg.No.-201900151)
5	Organizing Unit	Sikkim Manipal Institute of Technology
6	Language of the Document	English
7	Abstract	In this project titled, Voice Based Home Automation, the system uses voice commands to operate and manage various gadgets inside the home.
8	Security Classification	General
9	Distribution Statement	General

List of Contents

List of Figures

List of Tables

ABSTRACT

The Voice Controlled Wireless Home Automation project is a comprehensive system created to give the elderly and disabled people an easy and simple way to manage their home's utilities using voice commands through a mobile phone application. The goal of the project is to develop a portable device that is simple for even non-technical users to install, configure, use, and maintain.

Connecting numerous electrical appliances frequently found in homes to a centralized control system is known as home automation. The home automation system and the user's mobile phone can communicate with one another thanks to this project's use of internet connectivity, Bluetooth, or Wi-Fi.

The control interface for the home automation system is a mobile phone app that lets users communicate with it by speaking commands. This method is more suitable for people with special needs or restricted mobility because it does not require physical switches or manual control.

By enabling the elderly and disabled to easily handle household utilities, the project seeks to improve their quality of life. Users may operate electronic appliances, change room temperature, turn on or off lights, and carry out other necessary chores around their home with just a few easy voice commands.

The design of the system takes user-friendliness, dependability, and ease of maintenance into account. Users may simply manage and customize their home automation settings using the mobile application's user-friendly interface. The system can be deployed and configured without requiring a high level of technical expertise, allowing non-technical people to take advantage of its capability.

The Voice Controlled Wireless Home Automation project's overall goal is to improve accessibility and practicality for the elderly and disabled by giving them the ability to independently control and manage their home utilities using voice commands through a mobile phone application.

1.INTRODUCTION

1.1 General overview

Physical switches, remote controls, and smartphone apps were necessary for manual control of smart home appliances. Users could not control their devices remotely; they needed to be present physically or have access to their smartphones. The convenience and flexibility of controlling a smart home were hampered by limited remote access to manage devices when away from home or the need for complicated setup processes.

With the help of speech commands, users of voice-based home automation can control many parts of their residence. With voice-based home automation, one may talk to a voice-enabled device to manage many different aspects of their house, including the lighting, temperature, security systems, entertainment systems, and more. Due to the comfort, it provides and its capacity to make daily tasks simpler, this technology has experienced a substantial increase in popularity in recent years.

Home automation that uses voice commands has many benefits. First off, it offers a simple and hands-free way to operate a variety of appliances and systems in your house. one does not need to use switches or remote controls by hand; they can only give commands to make the necessary activities happen. People with disabilities or restricted mobility will especially appreciate this convenience. Voice-based home automation also encourages money and energy savings. one can quickly manage the energy usage of various devices with voice commands, making sure they are only in use when necessary.

1.2 Literature survey

SL. no	Author(s)	Paper and publication details	Findings	Relevance to the project
1	Akash Agrawal and Ayush Agrawal.	"A Voice Enabled Home Automation System Using Raspberry Pi" (Journal of Home Automation, 2018).	A Raspberry Pi-based voice-activated home automation system is described in this paper. It discusses how to voicecontrol several home appliances by combining speech recognition and natural language processing techniques	While the study focuses on a Raspberry Pi-based system, it shares common goals and concepts with the proposed system.
2	Manjunatha Reddy and C. Dinesh Kumar	The goal of this study is to create an Android app and Arduino-based home automation system that can be voice-controlled. It investigates the application of Bluetooth connectivity between an Android device and an Arduino board to voice-activated appliance control.	The paper proposed a hierarchical method that uses clustering and deep learning to accurately forecast sales in the retail industry.	The Android handset and Arduino board are connected via Bluetooth for the study. This fits with how a Bluetooth module is used in our project. To learn how to connect Bluetooth devices, manage data transmission, and integrate them with our voiceactivated home automation system.
3	Shang Gao, Hong-Zhi Wang, and JianWu Zhang	"Smart Home Automation System Based on Intelligent Voice Recognition" (IEEE Transactions on Consumer Electronics, 2018).	This research suggests an intelligent voice recognition-based smart home automation system. In order to provide voicebased control of household appliances, it describes the integration of voice recognition algorithms, IoT technology, and a mobile application. The integration of voice recognition algorithms to enable voice-based control of home appliances is the article's main topic.	This fits with the goal of our project, which is to use Arduino to create voice control. For information on various voice recognition algorithms and strategies that can be customised for our project, we can refer to the study article..
4	Muhammad Salman Razzaq and Nadia Kanwal	"Voice Control of Home Automation Systems Using Deep Learning" (ACM Transactions on Sensor Networks, 2019)	This study focuses on deep learning-based voice control for home automation systems. It investigates the integration of voice-controlled devices in a home automation setup as well as the usage of deep neural networks for speech recognition.	This is relevant to our project as we aim to control appliances using voice commands. The paper may discuss techniques for connecting and controlling devices, which can provide guidance when integrating our Arduino-based system with voice-controlled appliances.

1.2 Literature survey (Contd..)

5	P. L. Mok and S. K. Kwok	“Smart home and security increasing security issues,” Springer,2017.	This paper provides a comprehensive survey of smart home automation systems based on the Internet of Things (IoT) and discusses the challenges and opportunities of voicebased home Automation.	Challenges and Future scope of smart home automation systems.

Table 1.1 : Literature Survey

1.3.Problem definition

- Manual operation of traditional home automation systems using remote controllers or mobile applications can be cumbersome and time-consuming.
- Current voice control systems might not offer a user-friendly interface, which makes it challenging for customers to engage and successfully operate their home appliances. The goal of this project is to develop a voice control system that is simple to understand and use for users of all technical backgrounds.
- Using conventional home automation systems might be challenging for people with disabilities or physical restrictions. This issue aims to address the need for a voice-based, inclusive home automation system that is simple to use and accessible to people of all abilities.
- Due to compatibility concerns and the requirement for continuous communication between the components, integrating speech recognition technology with an Arduino board and a Bluetooth module might be difficult. These issues are addressed in order to provide a unified voice-based home automation system.
- Modern voice control installations for home automation can entail pricey gear and in-depth programming skills. With Arduino and a Bluetooth module, this issue aims to produce an affordable and widely available voice control system.

1.4. Software Requirements Specifications

1.4.1 Software Requirements

- Arduino IDE: The Arduino board must be programmed and loaded with the necessary code using the Arduino Integrated Development Environment (IDE).
- Bluetooth Library: For the Arduino and Bluetooth module to communicate, a compatible Bluetooth library, such as the "Software Serial" or "Bluetooth Serial" library, is required.
- MIT App INVENTOR: for mobile application

1.4.2 Hardware Requirements

- Arduino Board: Arduino Uno
- Bluetooth Module: HC-05
- Power Supply
- Mobile Device
- Wiring and Connectors: jumper wires, breadboard

Functional Requirements:

Functional Requirements for Voice-Based Home Automation using Arduino and Bluetooth Module:

- Voice Recognition: The user's voice commands should be accurately recognised and understood by the system. A wide variety of voice commands for managing household utilities and appliances should be supported.
- Bluetooth Connectivity: The Arduino and the user's mobile device must to connect over Bluetooth in a steady and dependable manner. For seamless engagement, it should enable Bluetooth pairing and communication protocols.
- Appliance Control: The system should enable voice commands to operate a variety of home equipment, such as fans, lights, bulbs, switches, etc. It ought to offer commands for turning devices on and off, altering brightness and intensity, switching between states, and customising settings.
- Voice Feedback: To acknowledge user commands and provide the status of the controlled appliances, the system should offer voice feedback.

Non - Functional Requirements

- Performance: For a seamless user experience, the system should have low latency when recognising and carrying out voice requests.
- Reliability: The system should be extremely dependable, guaranteeing that voice commands are correctly understood and carried out.
- Scalability: The system ought to be expandable, making it simple to incorporate new hardware and features in the future. It should enable the home automation network to grow without sacrificing performance.
- Compatibility: The system ought to work with a variety of market-available home appliances, Arduino boards, and Bluetooth modules.
- Usability: The system must to have an intuitive user interface that is simple to use and comprehend.
- Portability: Users should be able to set up the system in various situations because it should be portable and simple to deploy.
- Energy Efficiency: In order to ensure effective functioning and increase battery life, particularly for mobile devices, the system should optimise power consumption.
- Accessibility: The system should take accessibility standards into account to make sure people with impairments can use it.

1.5 Proposed Solution Strategy

- Voice Recognition Integration: Used a voice recognition module, such as a speech recognition shield or module, that can accurately translate spoken commands into text or executable actions.
- Created a mobile phone application: To communicate with the Bluetooth receiver HC05, a mobile phone application must be created.
- Identified and chose Bluetooth modules and Arduino microcontroller boards that are compatible with one another, capable of handling voice recognition, and wireless communication.
- : Created a flexible solution that makes it simple to add new devices and integrate them with existing home automation systems. Made the system adaptable so that users can alter and enhance functionality in accordance with their unique requirements and preferences.

1.6 Preliminary User's Manual

- Launch the computer programme or smartphone application created with MIT App Inventor.
- Bluetooth Pairing: your mobile device's Bluetooth should be enabled.
- Launch the mobile programme you developed with MIT App Inventor.
- Search for available devices in the Bluetooth settings on your mobile device.
- Track down the Arduino's Bluetooth module and pair with it.

  Voice Commands and Control: Open the mobile application, then make sure the Bluetooth module is connected. Speak voice commands in a loud, clear voice into the microphone of the mobile device.

  Use voice commands that are particular to each appliance (such as "Turn on the light," "Turn off the fan") to control them.

  The signals are decoded by the Arduino, which then uses that information to control the appropriate appliances.
- Troubleshooting: If the voice instructions are not recognised, examine the sensitivity settings for the speech recognition module and make any necessary adjustments.

Make sure the mobile device is paired with the Arduino and the Bluetooth module is connected to it properly.

Make that the Bluetooth module is connected to the mobile application and that there are no connectivity problems.

To make sure the Arduino and attached components are operating properly, check the power supply.

1.7 Organization of the report

Chapter 1 : Introduction

- Chapter 1.1 : ‘General Overview of the problem’ gives Introduction about the basic idea about the approach on how we solve the problems and the basic problem that we faced.
- Chapter 1.2 : ‘Literature survey’ gives the research that we conducted to know how to solve and understand the problem and validate our approach of solving the same.
- Chapter 1.3 : ‘Problem statement’ talk about the problem that we are trying to solve and approaches that we are taking. ○ Chapter 1.4 : ‘Analysis of the problem’ gives the detailed description of the difficulties associated and encountered .
- Chapter 1.5 : ‘Software Requirement Specification’ elaborates the system , functional and non-functional requirements associated with the project.
- Chapter 1.6 : ‘Proposed Solution Strategy’ gives the explanation on solution for the problems associated with the project. ○ Chapter 1.7 : ‘Preliminary user’s manual’ provides the glimpse of how the deployed software works and how it is to be used by the users.

Chapter 2 : ‘Design Strategy’ contains the block diagram, use case diagrams , proposed architecture diagram , activity diagram ,process flow diagram and flowchart.

Chapter 3 : ‘Testing’ contains the various test cases performed for different scenarios to check whether the deployed software matches the requirements of the users.

Chapter 4 : ‘Implementation Details’ explains the incorporated algorithms and various modules .

Chapter 5: ‘Results and Discussions’ gives a comparative view on the obtained results in the deployed application.

Chapter 6: ‘Summary and Conclusion’

- Chapter 6.1 : ‘Summary of achievements’ gives a description on the finally achieved goals within the project.
- Chapter 6.2 : ‘Difficulties encountered’ provides a description on the various difficulties that were encountered while accomplishing the project ○ Chapter 6.3 : ‘Limitation of the project’ highlights the project’s functionality restrictions. ○ Chapter 6.4: ‘Future Scope of

Work' shows the works associated with the proposed project which can be accomplished in future to improve the project's functionality.

Chapter 7: 'Gantt Chart' illustrate the timeline of tasks and activities.

Chapter 8: 'References' depicts the sources of information that were used in the project.

2.Design

2.1 Block Diagram

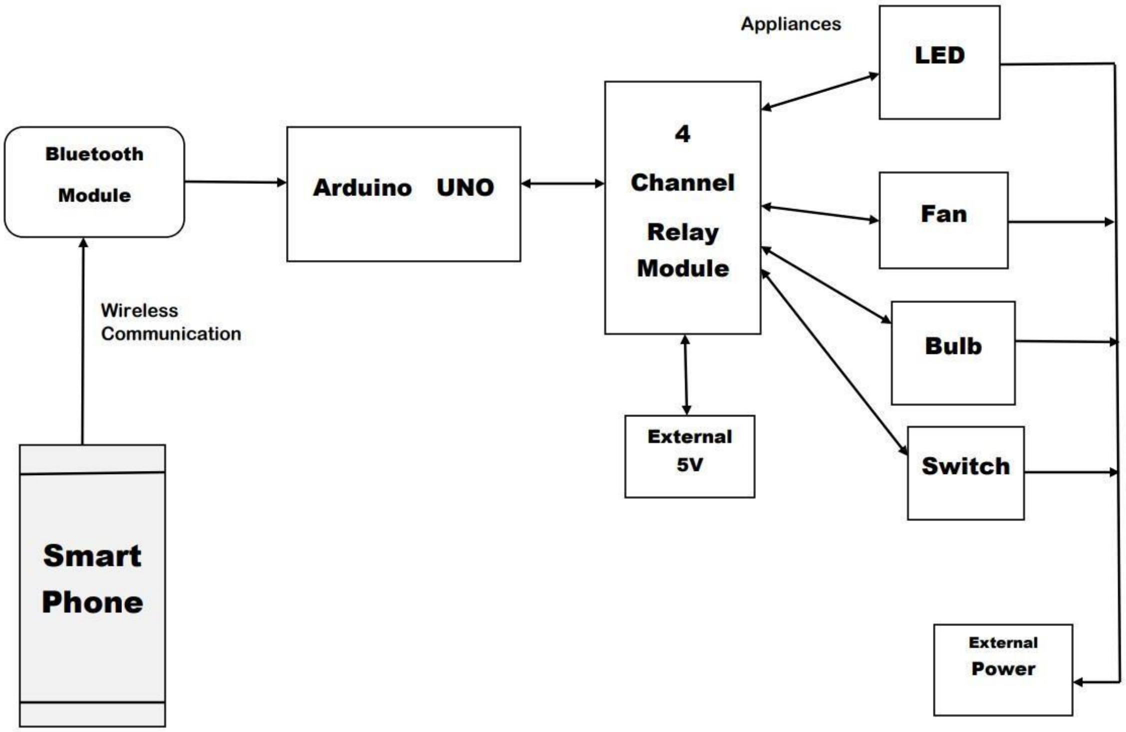


Fig 2.1: Block Diagram

The project's key component is the microcontroller. The diagram shows how voice commands are obtained from a computer or mobile device's microphone and then sent on to the Arduino. Following the Arduino's execution of the command, the relay board receives the instructions to turn on or off the utilities in accordance. When the relay board is turned on or off, the light, fan, or air conditioner turn on or off. C Language is used by Arduino Uno. "Sketch" is how the Arduino firm termed it.

2.2 Flowchart

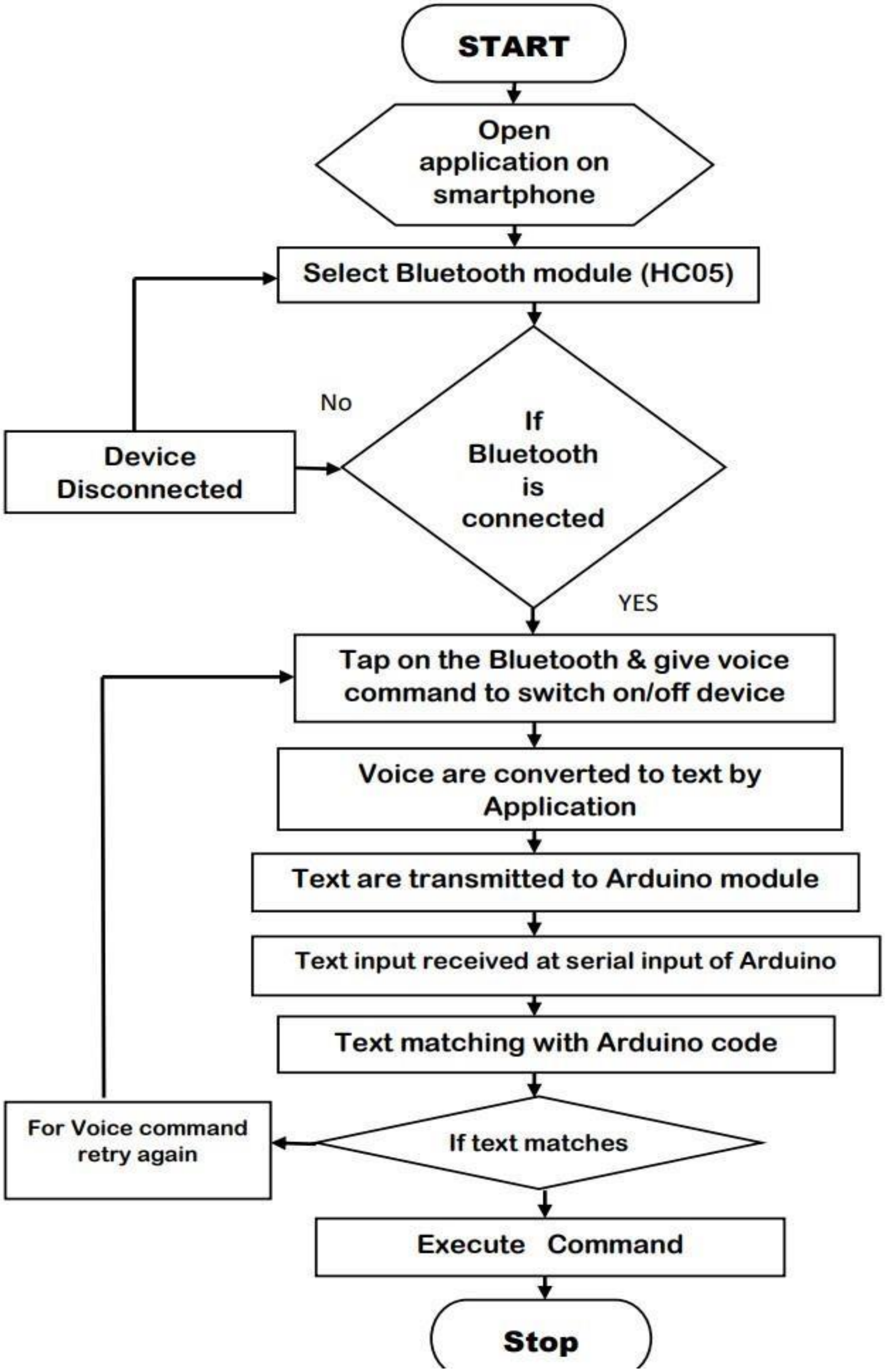

START
Open application on smartphone
Select Bluetooth module (HC05)
No
If Bluetooth is connected
Device Disconnected
YES
Tap on the Bluetooth & give voice command to switch on/off device
Voice are converted to text by Application
Text are transmitted to Arduino module
Text input received at serial input of Arduino
Text matching with Arduino code
For Voice command retry again
If text matches
Execute Command
Stop

Fig 2.2: Flowchart

2.3 Use Case Diagram

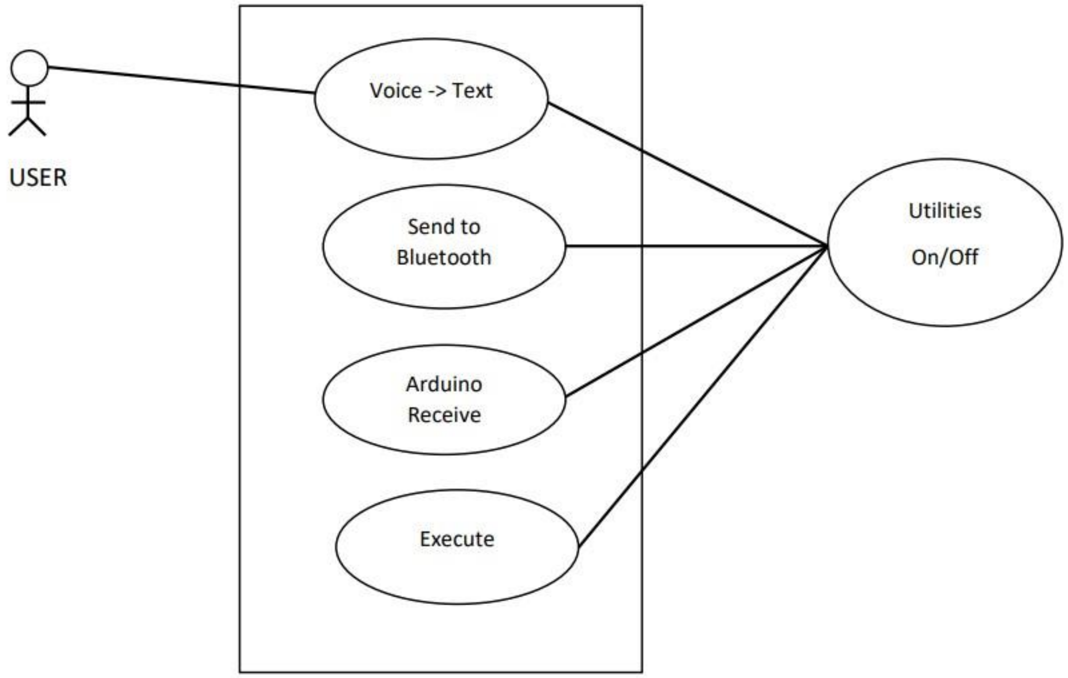
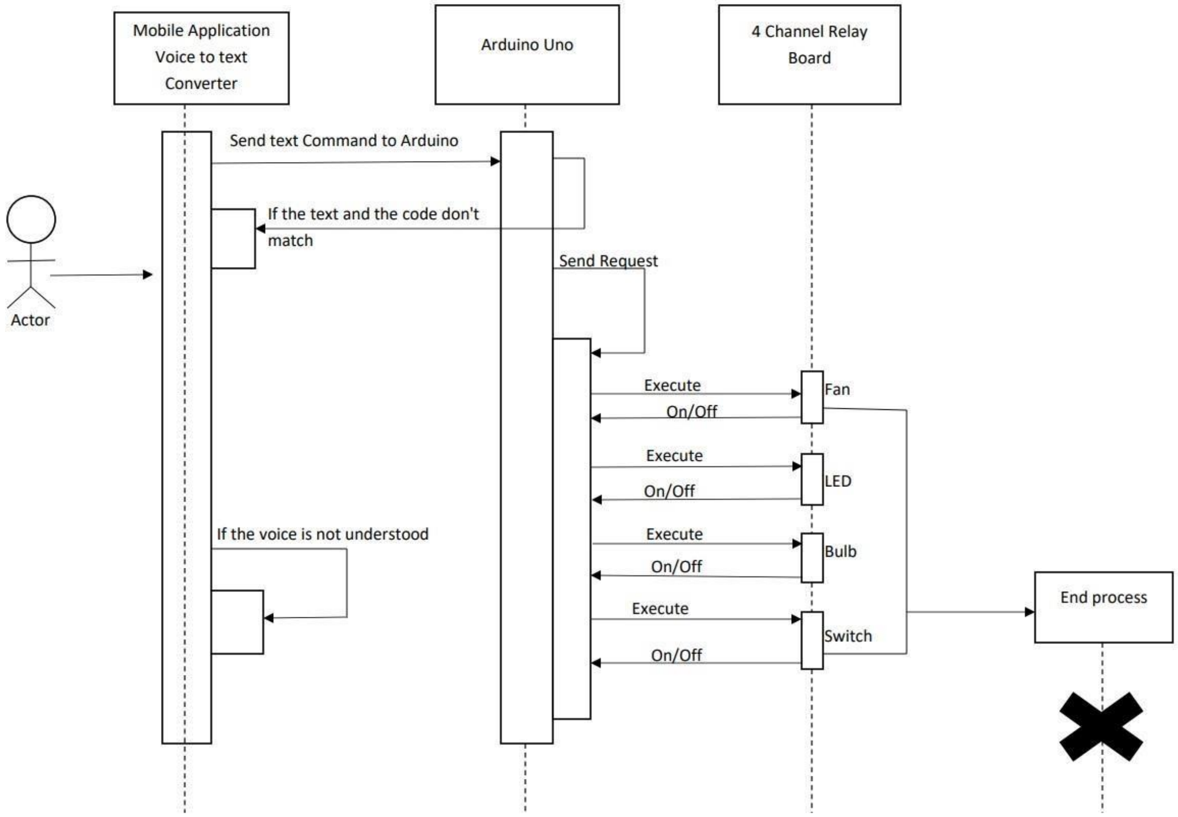


Fig 2.3: Use Case Diagram

As shown in the figure above every aspect of the system's operation, from user interaction to programme execution. Voice commands such as ON/OFF are recognised, translated into text, and delivered to the Arduino Uno over Bluetooth or the internet to be executed as input commands on the screen.

2.4 Sequence Diagram



Mobile Application Voice to text Converter
Arduino Uno
4 Channel Relay Board
Send text Command to Arduino
If the text and the code don't match
Send Request
Actor
Execute
On/Off
Fan
Execute
LED
On/Off
If the voice is not understood
Execute
Bulb
On/Off
End process
Execute
Switch
On/Off

Fig 2.4: Sequence Diagram

By showing the relationships between the various elements in chronological order, this type of diagram demonstrates how the system and the user interact. It also surfaces during the code identification process . If the code is not recognized by the Arduino or application, it proceeds in the order depicted in the picture below. Additionally, it shows how the procedure is complete when the user enters the correct command, which causes the relay board-connected lights or fans to turn on or off.

2.5 Circuit Diagram

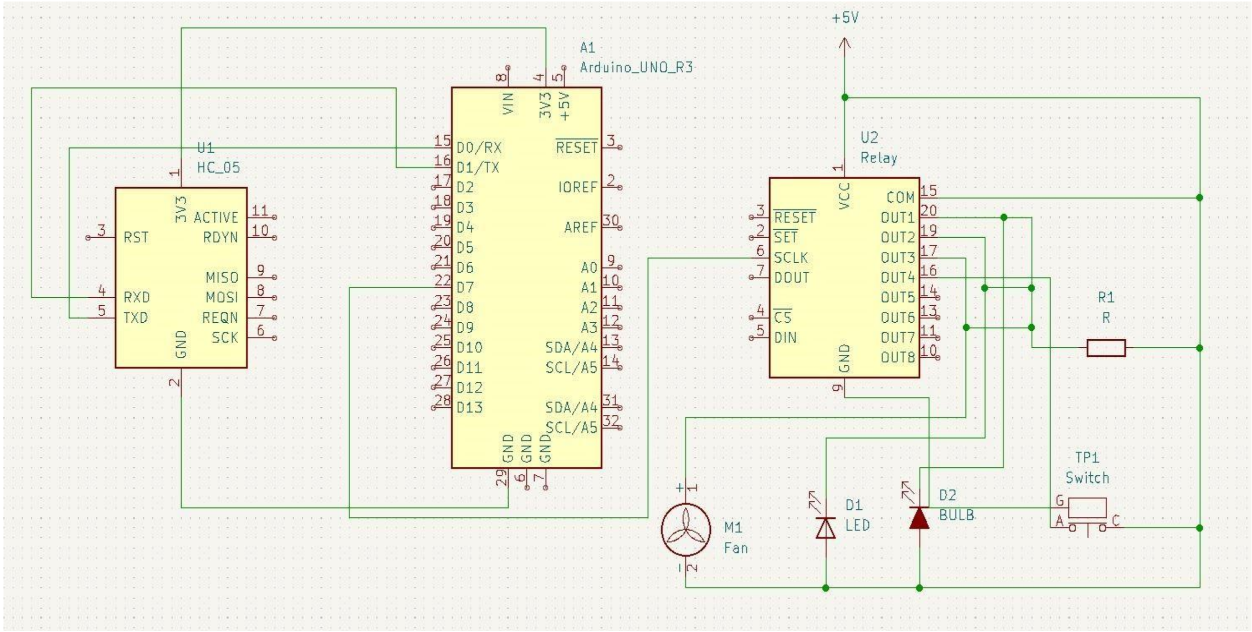


Fig 2.5 Circuit Diagram

In the circuit diagram it shows that, The Bluetooth module receives voice commands from a connected device. The Arduino reads the voice commands through the RX pin from the Bluetooth module. The relay module acts as a switch, allowing the Arduino to control the power supply of devices. The fan, bulb, LED, and switch are connected to separate channels of the relay module. When a command is received to turn on a device, the respective relay channel is activated, allowing current to flow and power on the device. When a command is received to turn off a device, the respective relay channel is deactivated, cutting off the power supply to the device. The Arduino controls the relay module by setting the respective digital pins high or low. Ground connections are made to complete the circuit for each device. This setup enables voice-based control of the fan, bulb, LED, and switch using the Arduino and Bluetooth module.

2.6 Process Flow Diagram

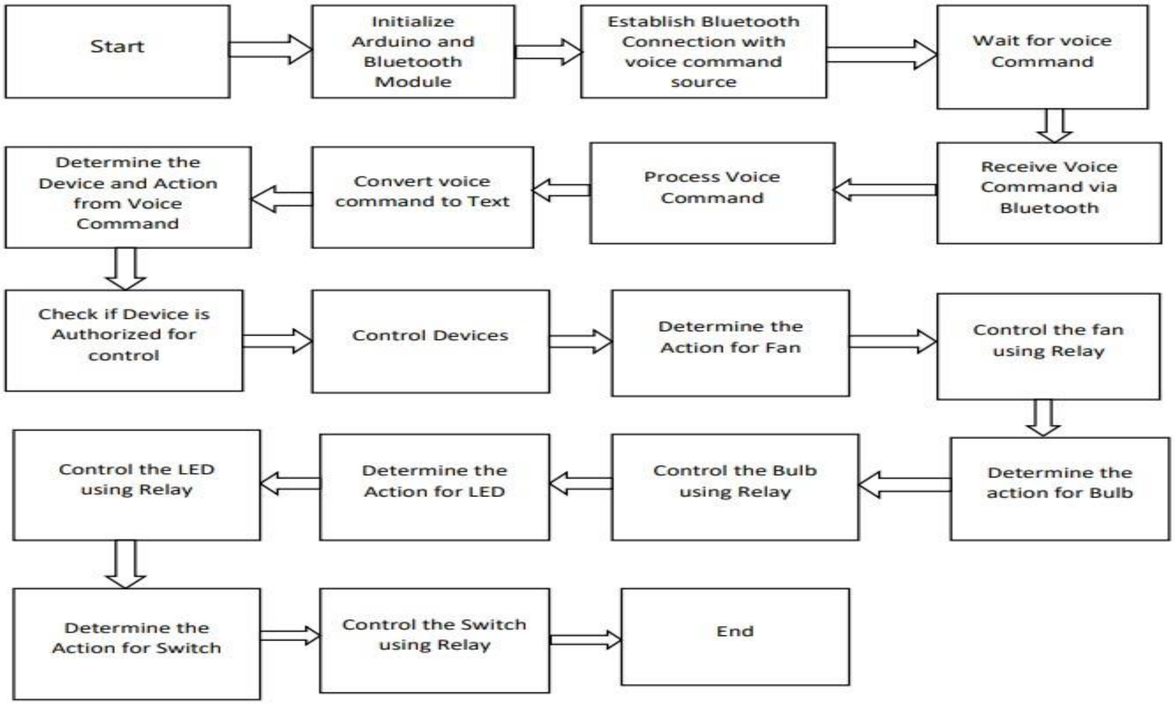

Start
Initialize Arduino and Bluetooth Module
Establish Bluetooth Connection with voice command source
Wait for voice Command
Receive Voice Command via Bluetooth
Process Voice Command
Convert voice command to Text
Determine the Device and Action from Voice Command
Check if Device is Authorized for control
Control Devices
Determine the Action for Fan
Control the fan using Relay
Determine the action for Bulb
Control the Bulb using Relay
Determine the Action for LED
Control the LED using Relay
Determine the Action for Switch
Control the Switch using Relay
End

Fig 2.6 Process Flow Diagram

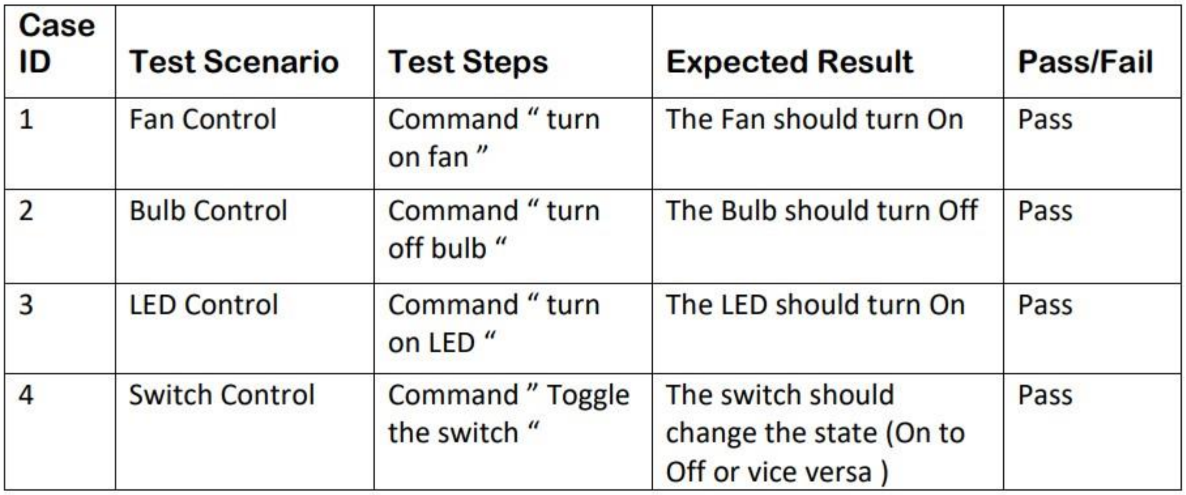
In the above diagram of the process flow diagram it shows the flow of the system, first the user needs to initialize Arduino and Bluetooth module and then establish Bluetooth connection with voice command source , the voice command will be processed and the voice command will be converted to text and it determines action from voice command, then it checks if the voice command is authorized for control, determines the action for the command given by the user ,controls the fan using the relay, likewise all the other appliances connected is controlled by the relay.

3. Testing

One test case is a document that details the input action or event and an anticipated response in order to assess whether the application feature is appropriately functioning. The expected output, test case identifier, test case name, and objective name should all be included in a test case. Because creating test cases entails thoroughly considering how the application will operate, it is crucial to do so as early in the development life cycle as is practical. Test cases can be used to identify issues with requirements or application design.

Test case specifications

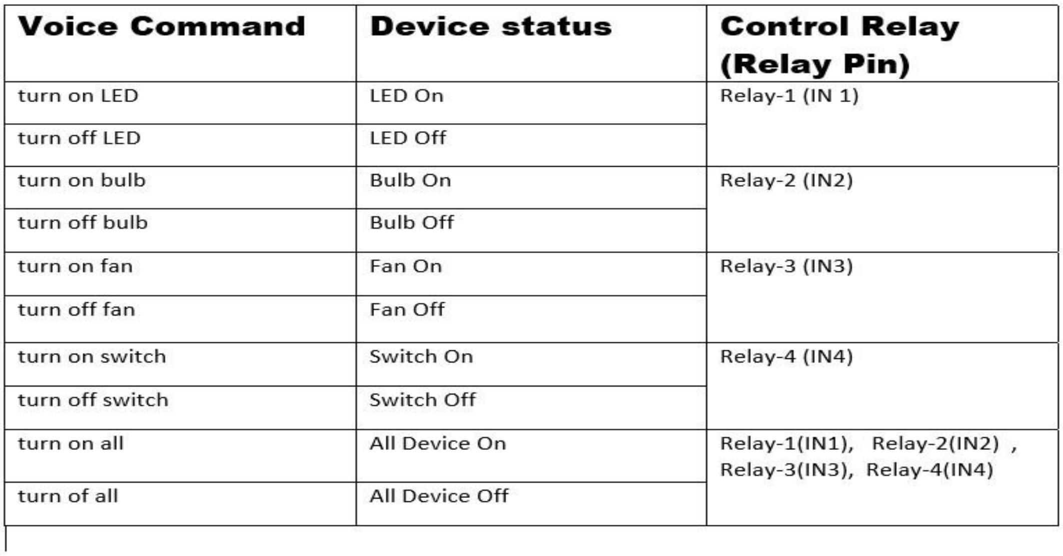
The following are the test case specifications included:



Case ID	Test Scenario	Test Steps	Expected Result	Pass/Fail
1	Fan Control	Command “ turn on fan ”	The Fan should turn On	Pass
2	Bulb Control	Command “ turn off bulb “	The Bulb should turn Off	Pass
3	LED Control	Command “ turn on LED “	The LED should turn On	Pass
4	Switch Control	Command ” Toggle the switch “	The switch should change the state (On to Off or vice versa)	Pass

Table 3.1:Testing 1

3. Testing Contd…



Voice Command	Device status	Control Relay (Relay Pin)
turn on LED	LED On	Relay-1 (IN 1)
turn off LED	LED Off	
turn on bulb	Bulb On	Relay-2 (IN2)
turn off bulb	Bulb Off	
turn on fan	Fan On	Relay-3 (IN3)
turn off fan	Fan Off	
turn on switch	Switch On	Relay-4 (IN4)
turn off switch	Switch Off	
turn on all	All Device On	Relay-1(IN1), Relay-2(IN2) , Relay-3(IN3), Relay-4(IN4)
turn of all	All Device Off	

Table 3.2 Testing II

4. Implementation Details

4.1 Technology Background

- **Arduino board**

Popular microcontroller board called the Arduino Uno is based on the ATmega328P microcontroller. One of the Arduino family's most popular boards, it is made for novices and amateurs to quickly prototype and develop their electronic creations.

Here are some key features and specifications of the Arduino Uno:

1.Microprocessor: The ATmega328P microprocessor, which powers the Arduino Uno, operates at 16 MHz, and includes 32 KB of flash memory for storing your programme code. For data storage, it additionally features 1KB of EEPROM and 2KB of SRAM.

2.Digital and Analog I/O: The board has 14 digital input/output pins, 6 of which can be used as PWM (Pulse Width Modulation) outputs, and 8 analogue input/output pins. Sensors, LEDs, motors, and other electronic components can all be connected to and controlled by these pins. For reading analogue signals, it also includes 6 analogue input pins.

3.Interfaces: The Uno has a USB connection that makes it simple to connect it to a computer for serial communication and programming. Additionally, it contains a DC power jack, an ICSP (In-Circuit Serial Programming) header, and a reset button for resuming programming.

4. Power supply: The board can be powered either by an external power source or by a USB connection. A built-in voltage regulator produces a steady 5V output to power the microcontroller and other components, and the recommended input voltage range is 7–12V.

5.Programming: The C/C++-based Arduino programming language was used to create programmes for the Arduino Uno. A user-friendly interface is offered by the Arduino IDE (Integrated Development Environment) for authoring, compiling, and uploading code to the board.

The Arduino Uno is renowned for its ease of use, adaptability, and broad support from the community. It is a great option for both new and seasoned builders due to its big user base and extensive online library of open-source code and training.

Features of the Arduino UNO:

Microcontroller: ATmega328

Operating Voltage: 5V

Input Voltage (recommended): 7-12V

Input Voltage (limits): 6-20V

Digital I/O Pins: 14 (of which 6 provide PWM output)

Analog Input Pins: 6

DC Current per I/O Pin: 40 mA

DC Current for 3.3V Pin: 50 mA

Flash Memory: 32 KB of which 0.5 KB used by bootloader

SRAM: 2 KB (ATmega328)

EEPROM: 1 KB (ATmega328)

Clock Speed: 16 MHz

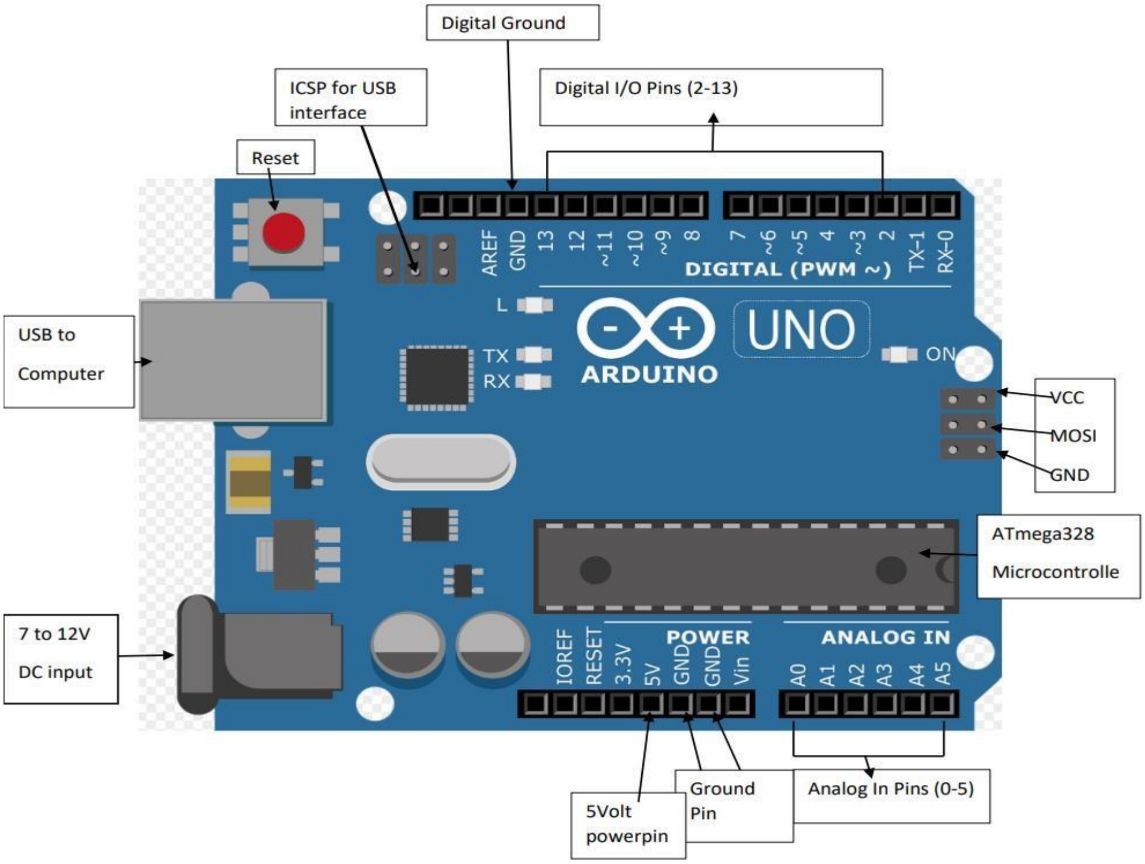


Fig 4.1:Arduino Uno

- **Arduino Hardware part:**

Hardware made with Arduino is open-source. The hardware reference designs are offered on the Arduino website and published under a Creative Commons Attribution Share-Alike 2.5 licence. There are additional layout and production files available for various hardware versions.

The developers have asked that the name Arduino to be reserved for the official product and not used for derived works without permission, even though the hardware and software blueprints are publicly available under copyleft licences. The project is open to integrate contributions from others into the official product, according to the official policy paper on the usage of the Arduino name. Many Arduino-compatible devices that were commercially produced had names that ended in "-Duino" to avoid utilizing the project name.

An Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with variable amounts of flash memory, pins, and features is the main component of most Arduino boards. In 2012, the 32-bit Arduino Due based on the Atmel SAM3X8E was released. Female headers or single- or double-row pins are used on the boards to provide connections for programming and circuit integration. These could link up with shield-style auxiliary modules. An I2C serial bus may allow for the independent addressability of many and possibly layered shields. A 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator are typically included on boards. Due to unique formfactor limitations, some designs, like the Lilypad, operate at 8 MHz without an inbuilt voltage regulator.

The boot loader that comes pre-installed on Arduino microcontrollers makes it easier to upload programmes to the on-chip flash memory. The optiboot bootloader is the Arduino UNO's standard bootloader. Programme code is loaded onto boards using a serial link to another computer. A level shifter circuit can convert between RS-232 logic levels and transistor-transistor logic (TTL) level signals on some serial Arduino boards. A USB-toserial adapter chip like the FTDI FT232 is used to implement the Universal Serial Bus (USB), which is the current programming protocol for Arduino boards. Some boards, like later-model Uno boards, replace the FTDI chip with a separate AVR chip that contains USBto-serial firmware and can be updated via its own ICSP interface.

The majority of the microcontroller's I/O pins are accessible on the Arduino board for usage by other circuits.

14 digital I/O pins, six of which may generate pulse-width modulated signals, and six analogue inputs, which can also be used as six digital inputs, are provided by the Diecimila, Duemilanove, and contemporary Uno.

pins for digital I/O. These pins are connected to female 0.1-inch (2.54 mm) headers on the top of the circuit board.

There are also a number of plug-in application shields that are bought commercially. The male header pins on the underside of the Bare Bones Board and Boarduino boards, as well as the Arduino Nano, may be compatible with solderless breadboards.

- **Arduino Software part:**

IDE

Java was used to create the cross-platform Arduino integrated development environment (IDE), which is available for Windows, macOS, and Linux. It came from the IDE for the programming languages Wiring and Processing. It has a code editor with tools for text copying and pasting, text replacement, automated indenting, brace matching, and syntax highlighting. It also offers straightforward one-click compiling and uploading tools for Arduino programmes. A hierarchy of operating menus, a message area, a text terminal, a toolbar with buttons for standard functions, and more are also included. The GNU General Public Licence, version 2, governs the publication of the IDE's source code.

The Arduino IDE has specific code organization guidelines to support the languages C and C++. A software library from the Wiring project, which offers numerous standard input and output operations, is provided by the Arduino IDE. For the sketch to start and the main programme loop, user-written code only needs two fundamental functions, which are combined with a programme stub main () to create an executable cyclic executive programme using the GNU toolchain, which is also distributed with the IDE. The executable code is transformed via the Arduino IDE's use of avrdude into a text file with hexadecimal encoding, which is then loaded into the Arduino board by a loader programme in the firmware.

- **Sketch**

A program written with the Arduino IDE is called a sketch.

Sketches are saved on the development computer as text files with the file extension. ino.

Arduino Software (IDE) pre-1.0 saved sketches with the extension. pde.

Two functions are all that a simple Arduino C/C++ programme needs to do:

After a power-up or reset, a sketch will only ever use setup () once. Variables, input, and output pin modes, as well as additional libraries required by the sketch are initialised using it. loop (): The main programme periodically calls the procedure loop () after calling setup ().

Until the board is turned off or reset, it is in control.

Blink Example

A load resistor and light-emitting diode (LED) are commonly found on Arduino boards, and they are connected between pin 13 and ground, which is a useful feature for many testing and programme operations. An LED is continually blinking in a normal Arduino programme for a beginner.

The internal libraries built into the IDE environment provide the pin Mode (), digital Write (), and delay () functions, all of which are used by this programme. A new Arduino board is often pre-programmed with this code by the maker.

- **Relay**

An electrically controlled switch is a relay. Solid-state relays are one type of working principle, although many relays employ an electromagnet to mechanically operate a switch. Relays are employed when multiple circuits need to be controlled by a single signal or when a separate low-power signal is required to control each circuit separately. The original relays served as amplifiers in long-distance telegraph circuits by repeating and retransmitting the signal from one circuit on another. To carry out logical processes, relays were widely utilised in early computers and telephone exchanges. A contactor is a particular kind of relay that can manage the high power necessary to directly control an electric motor or other loads. With no moving parts and switching performed by a semiconductor device, solid-state relays regulate power circuits. Relays that operate calibrated

Electrical circuits are protected from overload or defects using characteristics and occasionally several operating coils; in contemporary electric power systems, digital devices still known as "protective relays" carry out these responsibilities. In order to move the contacts of magnetic latching relays in one way, a coil power pulse must be applied, followed by another pulse that is redirected in the opposite direction. The same input's repeated pulses have no impact. Applications where interrupted power should not be able to transition the contacts benefit from the usage of magnetic latching relays.

- **Features of Relay**

Number of Relays: 4

Control signal: TTL level

Operating Voltage: 5V to 12V

- **Application Relay: -**

In situations where galvanic isolation is desired and it is necessary to manage a high power or high voltage circuit with a low power circuit, relays are employed. Relays were first used on long telegraph lines, where a contact may be controlled by a weak signal received at an intermediate station, renewing the signal for further transmission. Small, low voltage wire and pilot switches can be used to control high voltage or high current equipment. The high voltage circuit can be isolated from the operators.

The high voltage circuit can be isolated from the operators. Relays can be driven by lowpower devices like microprocessors to manage electrical loads beyond what they can do directly. In a vehicle, a starter relay enables the small wire and contacts in the ignition key to regulate the high current of the cranking motor.

Relays were widely used in supplementary control circuits in electromechanical switching systems like Stronger and Crossbar telephone exchanges. Additionally, telephone exchanges utilising just relay switching methods were produced by The Relay Automatic Telephone Company.

Gotthilf Ansgarius Betulander created the piece. Fleetwood received the first public relaybased telephone exchange in the UK on July 15 of that year, and it operated there until 1959. In A Symbolic Analysis of Relay and Switching Circuits, Claude Shannon formalised the use of Boolean algebra to relay circuit design. Shannon explored the use of relays for the logical control of complicated switching systems, such as telephone exchanges. The fundamental operations of Boolean combinatorial logic can be carried out using relays. For instance, connecting typically open relay contacts in series to get the boolean AND function and connecting them in parallel to achieve the boolean OR function. A typically closed contact can be used to invert a logical input. Automation systems for machine tools and manufacturing lines were managed by relays. Relay logic networks are frequently designed using the Ladder programming language.

Relays were utilised for logic and working registers in early electro-mechanical computers, including the ARRA, Harvard Mark II, Zuse Z2, and Zuse Z3. Electronic equipment, however, proved to be more efficient and user-friendly.

Relays are frequently employed in safety-critical logic, such as the control panels of machines used to handle radioactive waste, because they are far more resistant to nuclear radiation than semiconductors. By opening and closing circuit breakers, electromechanical protection relays can detect overload and other electrical line defects.

Relays are frequently employed in safety-critical logic, such as the control panels of machines used to handle radioactive waste, because they are far more resistant to nuclear radiation than semiconductors. By opening and closing circuit breakers, electromechanical protection relays are used to detect overload and other defects on electrical lines.

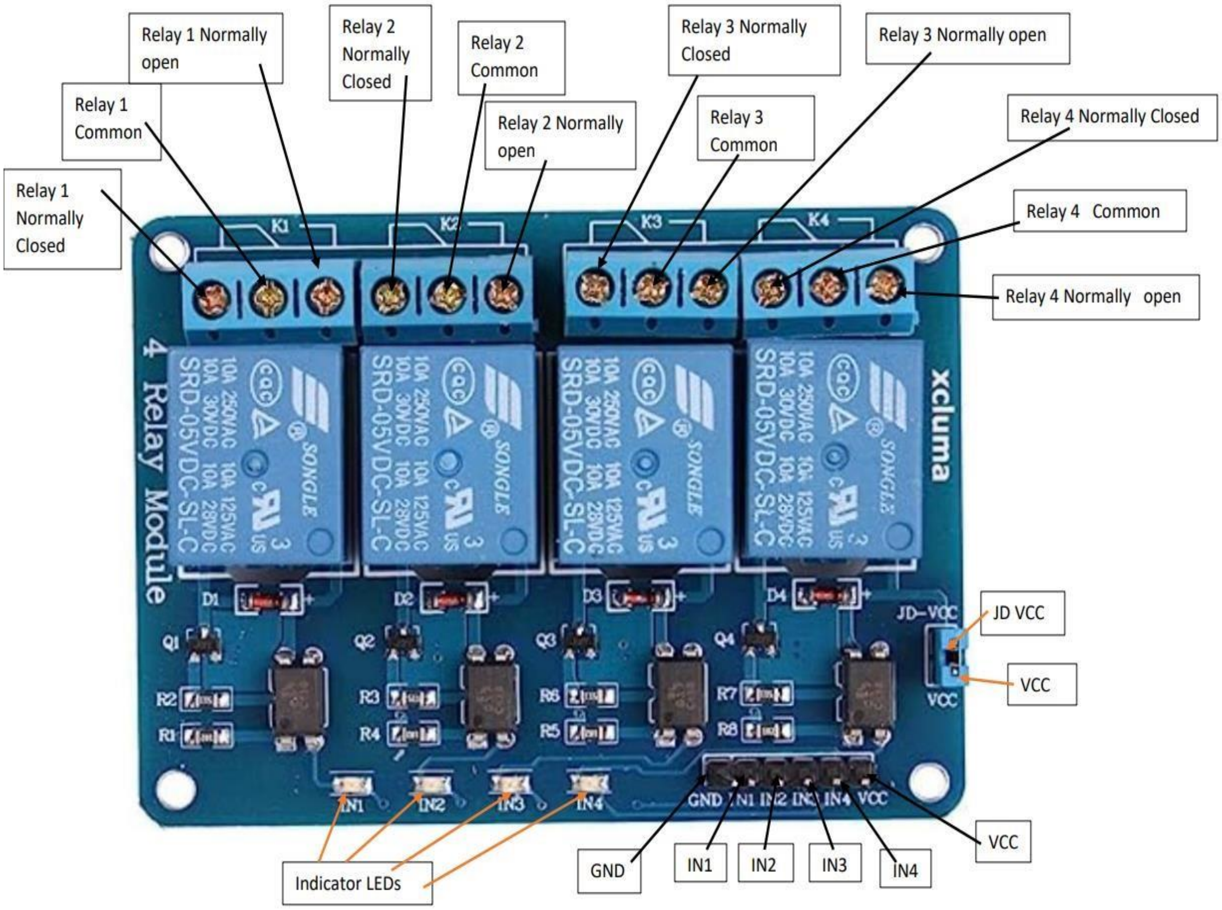


Fig 4.2: 4 channel relay modules

- **Bluetooth Module (HC-05 Bluetooth Module):**

HC-05 Specification:
Bluetooth protocol: Bluetooth Specification v2.0+EDR
Frequency: 2.4GHz ISM band
Modulation: GFSK(Gaussian Frequency Shift Keying)
Emission power: ≤4dBm, Class 2
Sensitivity: ≤-84dBm at 0.1% BER
Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
Security: Authentication and encryption
Profiles: Bluetooth serial port
Power supply: +3.3VDC 50mA
Working temperature: -20 ~ +75Centigrade
Dimension: 26.9mm x 13mm x 2.2 mm

- **Overview**

A straightforward Bluetooth SPP (Serial Port Protocol) module, the HC-05 is made for setting up transparent wireless serial connections. The HC-05 Bluetooth Module is an excellent option for wireless communication because it can be used in either a Master or Slave configuration. This Bluetooth V2.0+EDR (Enhanced Data Rate)3Mbps Modulation serial port module has a complete 2.4GHz radio transceiver and baseband. It makes use of the CMOS-based CSR Bluecore 04 External Single Chip Bluetooth System with AFH (Adaptive Frequency Hopping Feature).

A MASTER/SLAVE module is what the Bluetooth HC-05 module is.The factory setting is SLAVE by default. Only AT COMMANDS have the ability to configure the module's Role (Master or Slave).Although they can accept connections, the slave modules are unable to establish a connection with another Bluetooth device. The master module has the ability to connect to other components. It can be used as a simple serial port substitute to link an MCU to a GPS, a PC to an embedded project, etc.

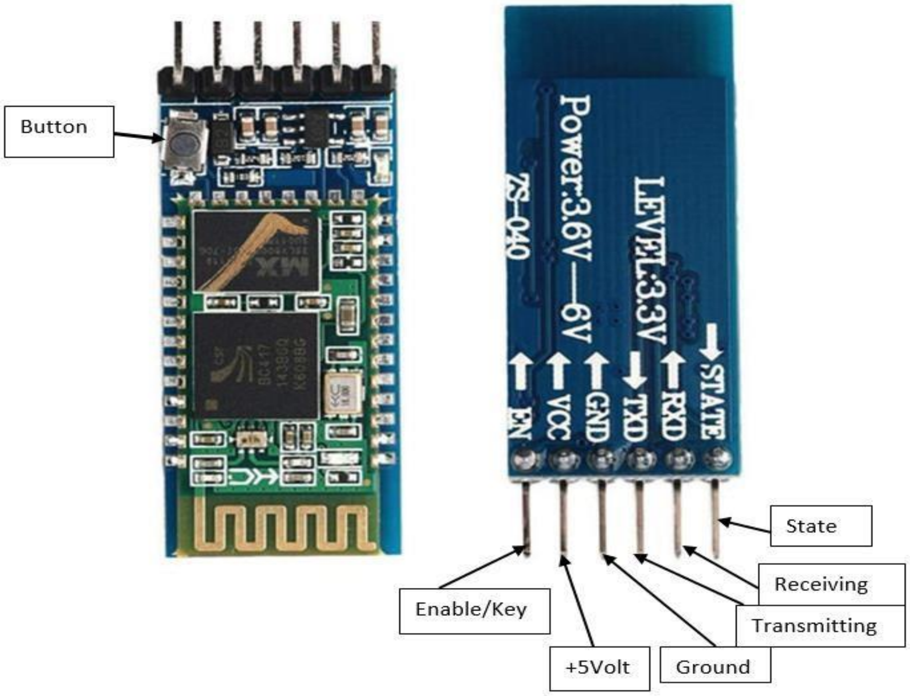


Fig 4.3 Bluetooth module-HC05

- **Pin Descriptions:**

The HC-05 Bluetooth module has 6 pins. They are as follows:

ENABLE: When enable is pulled LOW, the module is disabled, which prevents it from turning on and preventing it from communicating. The module is enabled when enable is left open or connected to 3.3V, meaning that the module is always on and that communication is possible.

Vcc : Supply Voltage 3.3V to 5V.

GND:Ground pin

TXD &RXD: An UART interface is provided via these two pins for communication. STATE: It serves as a status signal. Signal drops when the module is not paired or linked to any other Bluetooth device. When the module is in this low state, the led flashes continually, signalling that it is not paired with any other devices. The signal increases when this module is linked or connected to another Bluetooth device. When the module is paired, the led blinks at this high state with a constant delay, say a 2 second delay.

BUTTON SWITCH: The module is put into AT command mode using this. Press the button switch for one second to activate AT command mode. The parameters of this module can be changed by the user with the aid of AT commands, but only while the module is not paired with any other BT devices. The module stops functioning in AT command mode and begins communicating with any other Bluetooth device it is attached to.

Default settings for the HC-05 Bluetooth Device: HC-05
Password by default: 1234 or 0000
Default Language: Slave Data
Mode, default.
Analogue Mode 9600, 8, N, 1 Command Mode Baud Rate Standard firmware: LINVOR Baud Rate: 38400, 8, N, 1

- **JUMPING WIRES**

Fig 4.4:Jumping wires.

It was necessary to use the jumping wires in order to build the internal connection between the various parts of the Arduino Uno.

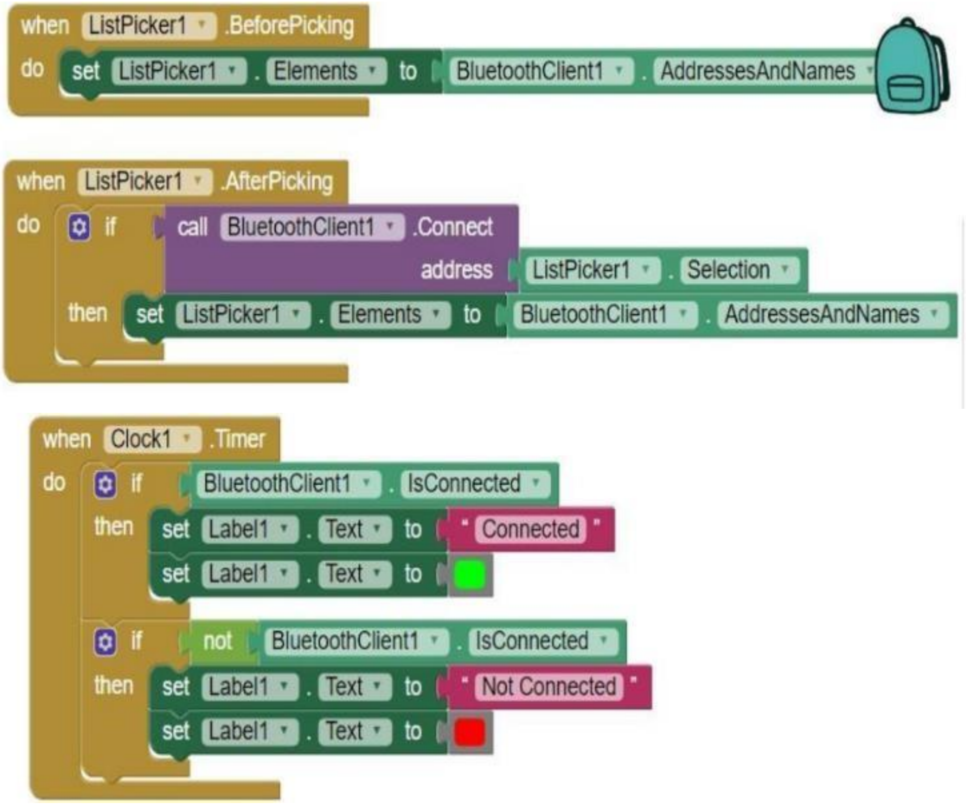
- **Software Application**

When examining the project's total system requirements, it becomes evident that software development topology is necessary to enable the mobile application's functionality via the Internet or Bluetooth. Therefore, MIT App Inventor is used for the mobile application.

To leverage Google speech recognition, transform voice to text, and send it to Arduino Uno via Bluetooth, MIT App Inventor is utilized.

The application's logical flow is depicted in the diagram below. A web-based GUI application can be quickly created using puzzle-style algorithms.

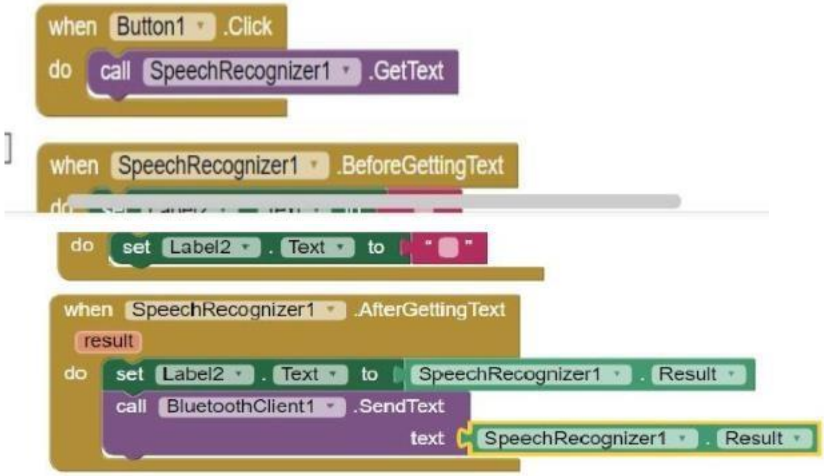
- The mobile's built-in Bluetooth software is selected initially to determine its status, whether turned on or off.
- If Bluetooth is "not connected," the text message is set to red. Conversely, if Bluetooth is active, the text's color is switched to green.
- Once Bluetooth is configured, Google's Speech Recognizer is chosen for voice recognition.
- The voice input is then converted into text, and the command is executed by Arduino.



```
when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to  BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do  if    call BluetoothClient1 .Connect
                              address  ListPicker1 . Selection
    then  set ListPicker1 . Elements to  BluetoothClient1 . AddressesAndNames

when Clock1 .Timer
do  if    BluetoothClient1 . IsConnected
    then  set Label1 . Text to  " Connected "
          set Label1 . Text to  🟩
    if    not  BluetoothClient1 . IsConnected
    then  set Label1 . Text to  " Not Connected "
          set Label1 . Text to  🟥
```



```
when Button1 .Click
do  call SpeechRecognizer1 .GetText

when SpeechRecognizer1 .BeforeGettingText
do  set Label2 . Text to " "

when SpeechRecognizer1 .AfterGettingText
 result
do  set Label2 . Text to SpeechRecognizer1 . Result
    call BluetoothClient1 .SendText
                     text SpeechRecognizer1 . Result
```

- **Hardware-Arduino Uno**

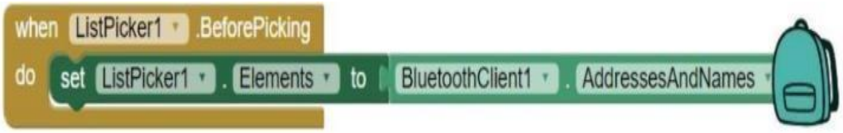
C programming language is employed to program the Arduino Uno. C is a highly robust language suitable for hardware programming. Loops are incorporated, and they are skipped if our voice doesn't meet a specific parameter, allowing the system to progress to the next loop. Eight loops are utilized to ensure smooth operation and maintain the code's flow.

The Arduino code receives a string command from the application and performs actions accordingly. It toggles the assigned pin on or off based on whether the string matches the predefined variables.

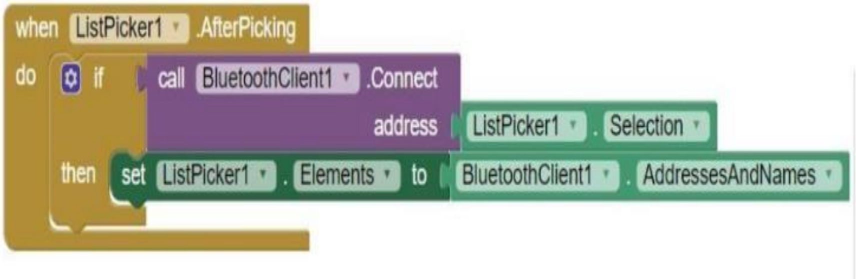
4.2 Pseudocode for Arduino

```
Step 1:BEGIN
Step 2: The code begins by incorporating the Software Serial library, which is required for
        Bluetooth connectivity.
        Step 2.1:Initialize Bluetooth module with RX and TX pins
        Step 2.2:Define pin numbers for fan, bulb, LED, and switch control
        Step 2.3:Setup pin modes for fan, bulb, LED, and switch as OUTPUT
        Step 2.4:In the setup function, the pin modes for the control pins are set as OUTPUT.
                 The devices are initially turned off by setting the corresponding pins to LOW.
        Step 2.5:Start Bluetooth communication with a baud rate of 9600.
Step 3 :LOOP forever the code checks if there is any data available from the Bluetooth module.
        Step 3.1:IF Bluetooth has available data
        Step 3.2:Read the received command from Bluetooth
        Step 3.3: IF command is ‘1’
                  Turn on the fan
        Step3.4:ELSE IF command is '2'
                  Turn off the fan
        Step3.4:ELSE IF command is '3'
                Turn on the bulb
        Step3.5:ELSE IF command is '4'
                  Turn off the bulb
        Step 3.6:ELSE IF command is '5'
                  Turn on the LED
        Step3.7:ELSE IF command is '6'
                  Turn off the LED
        Step3.8: ELSE IF command is '7'
                  Turn on the switch
        Step3.9:ELSE IF command is '8'
                  Turn off the switch
        Step3.10:END IF
                END LOOP
Step 4:END the program.
```

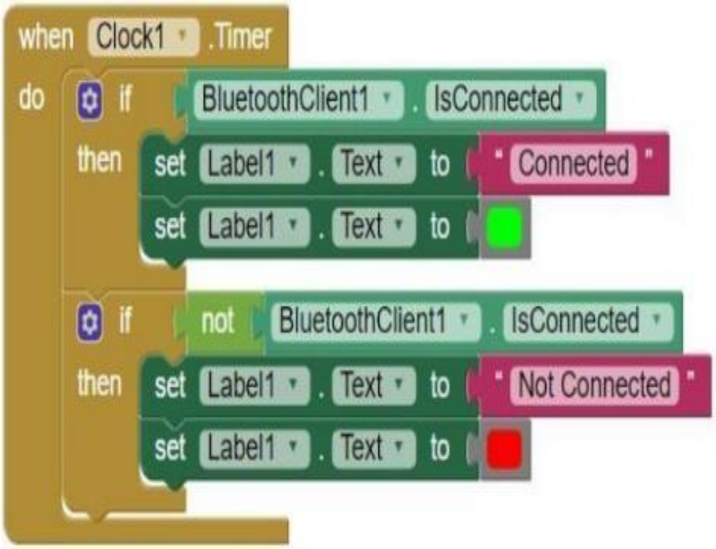
4.3.Blocks for the application



```
when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
```



```
when ListPicker1 .AfterPicking
do  if   call BluetoothClient1 .Connect
                           address  ListPicker1 . Selection
    then set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
```



```
when Clock1 .Timer
do  if   BluetoothClient1 . IsConnected
    then set Label1 . Text to " Connected "
         set Label1 . Text to 🟩
    if   not BluetoothClient1 . IsConnected
    then set Label1 . Text to " Not Connected "
         set Label1 . Text to 🟥
```

- MIT App Inventor is utilized for the objectives of the application. In this case, a graphical interface algorithm is employed. Bluetooth is selected and it is instructed to connect to Bluetooth as a result. When Bluetooth is activated, the text changes to green; otherwise, it appears red.
- Subsequently, when the program detects Google Speech Recognition, it captures speech input and transforms it simultaneously. The command is then transmitted over Bluetooth to Arduino Uno.

4.4 Circuit Connection steps:

Step 1: Arduino Connections

Step 1.1: Connect the Arduino's 5V pin to the relay module's VCC (power) pin.

Step 1.2: Connect the Arduino's GND (ground) pin to the relay module's GND pin.

Step 2: Bluetooth Module Connections

Step 2.1: Connect an Arduino digital pin to the Bluetooth module's RX (receive) pin.

Step 2.2: Connect an Arduino digital pin to the Bluetooth module's TX pin.

Step 3: Relay Module Connections

Step 3.1: Connect the Arduino's digital pin (Pin 4) to the IN1 pin of the relay module.

Step 3.2: Connect the Arduino's digital pin (Pin 5) to the IN2 pin of the relay module.

Step 3.3: Connect the Arduino's digital pin (Pin 6) to the IN3 pin of the relay module.

Step 3.4: Relay module IN4 pin should be connected to Arduino digital pin 7.

Step 4: Fan Control

Step 4.1: Connect the fan's positive wire to the relay channel 1's NO (normally open) terminal.

Step 4.2: Connect the fan's negative wire to the relay module's GND (ground) pin.

Step 5: Bulb Control

Step 5.1: Connect the bulb's positive wire to the relay channel 2's NO (normally open) terminal.

Step 5.2: Connect the bulb's negative wire to the relay module's GND (ground) pin.

Step 6: LED Control

Step 6.1: Connect the LED's positive wire to the relay channel 3's NO (normally open) terminal.

Step 6.2: Connect the LED's negative wire to the relay module's GND pin.

Step 7: Switch Control

Step 7.1: One switch terminal should be connected to the relay channel 4's COM (common) terminal.

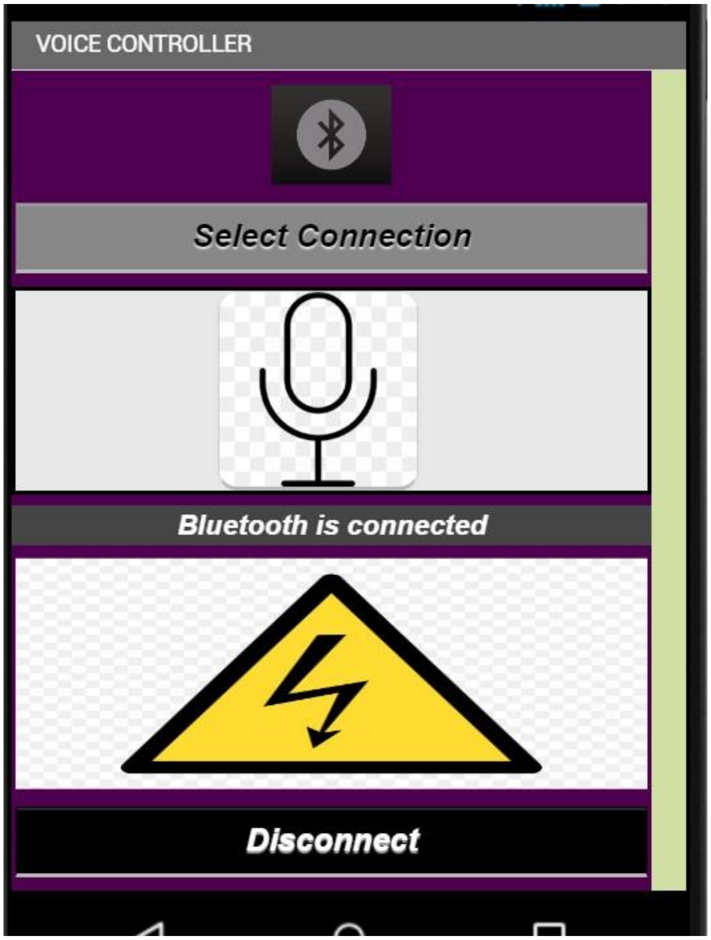
Step 7.2: Connect the switch's other terminal to the relay channel 4's NO (normally open) terminal.

Step 7.3: Relay module GND (ground) pin should be connected to Arduino GND pin.

5.Results and Discussion

This project involves designing a straightforward voice-activated home automation system. Different appliances can be controlled by voice commands.

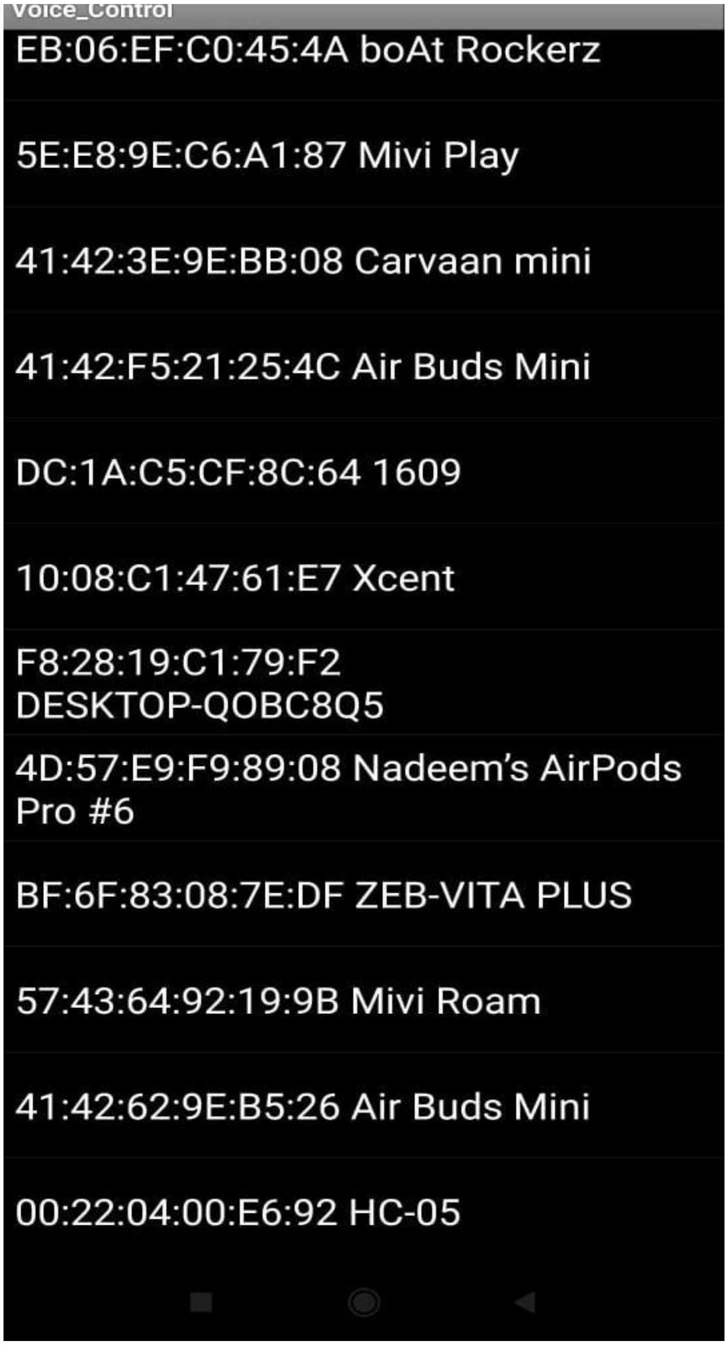
Every connection is done in accordance with the circuit design shown above.



Bluetooth is connected

Fig 5.1: Installation of app by the user.

The power source for the circuit must be turned on when the essential connections have been made. The HC-05 Bluetooth Module must be paired with the phone's Bluetooth. In order to do that, it should first install the mentioned app on the phone that have been created using MIT App Inventor. The Home screen of the app looks something like this.



Voice_Control

EB:06:EF:C0:45:4A boAt Rockerz

5E:E8:9E:C6:A1:87 Mivi Play

41:42:3E:9E:BB:08 Carvaan mini

41:42:F5:21:25:4C Air Buds Mini

DC:1A:C5:CF:8C:64 1609

10:08:C1:47:61:E7 Xcent

F8:28:19:C1:79:F2
DESKTOP-QOBC8Q5

4D:57:E9:F9:89:08 Nadeem’s AirPods
Pro #6

BF:6F:83:08:7E:DF ZEB-VITA PLUS

57:43:64:92:19:9B Mivi Roam

41:42:62:9E:B5:26 Air Buds Mini

00:22:04:00:E6:92 HC-05

Fig 5.2:Setting up the connection

Next step is to connect the phone with the Bluetooth module. For this, choose the option “Select Connection” and select the appropriate Bluetooth Device. If the devices aren’t paired earlier, we need to pair them now using the Pin of the HC – 05 Bluetooth Module.

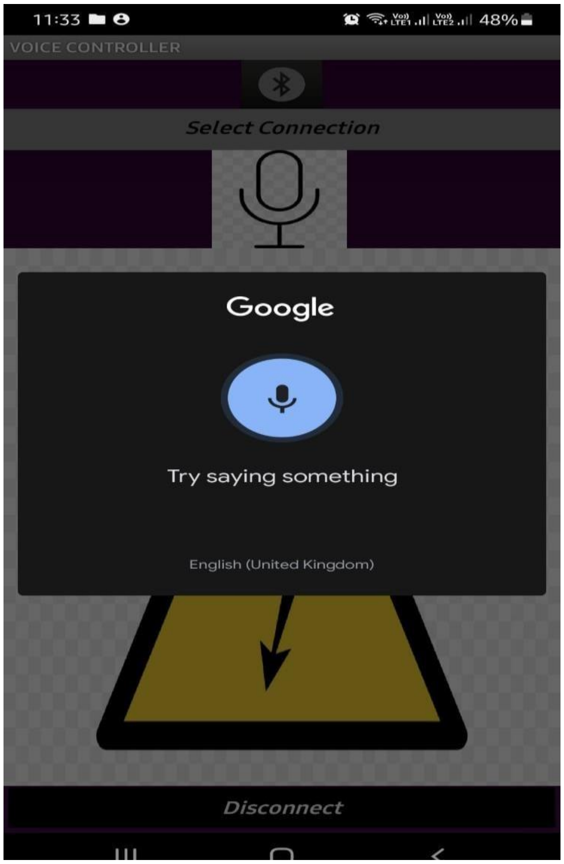
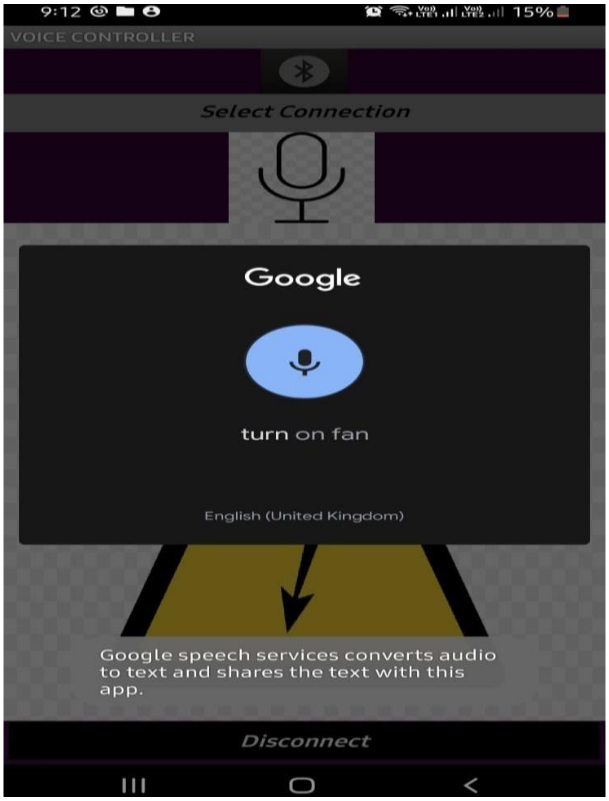


Fig 5.3:Press to speak button

Press to Speak' button: Basically, this button enables our mobile app to connect directly to Google, so that using the app, voice recognition might be achievable. The Google is used for speech recognition, thus while hitting this button it will cause the mobile application's voice sensors to be activated, and that message is transferred to the Google app, while the voice-to-text conversion process is ongoing occurs, then over Bluetooth or Ethernet it is finally delivered to Arduino. To carry out the action, a module and a program are assembled.



turn on fan

Google speech services converts audio to text and shares the text with this app.

Fig 5.4:command “Turn on fan”

Here is a picture of some sample example, here for turning on the fan.

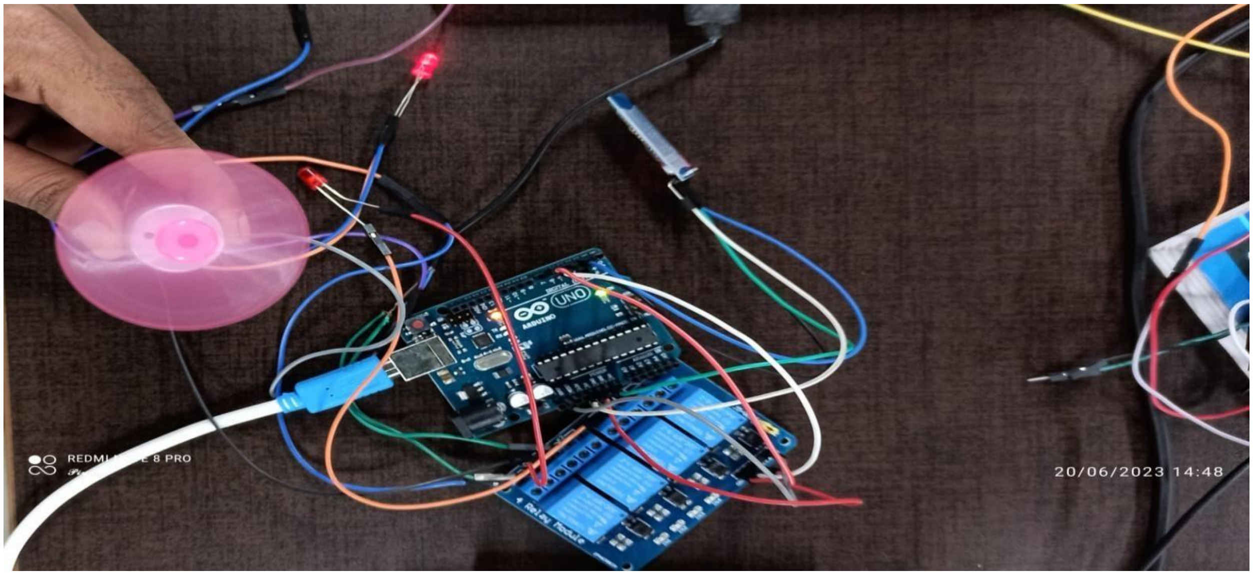
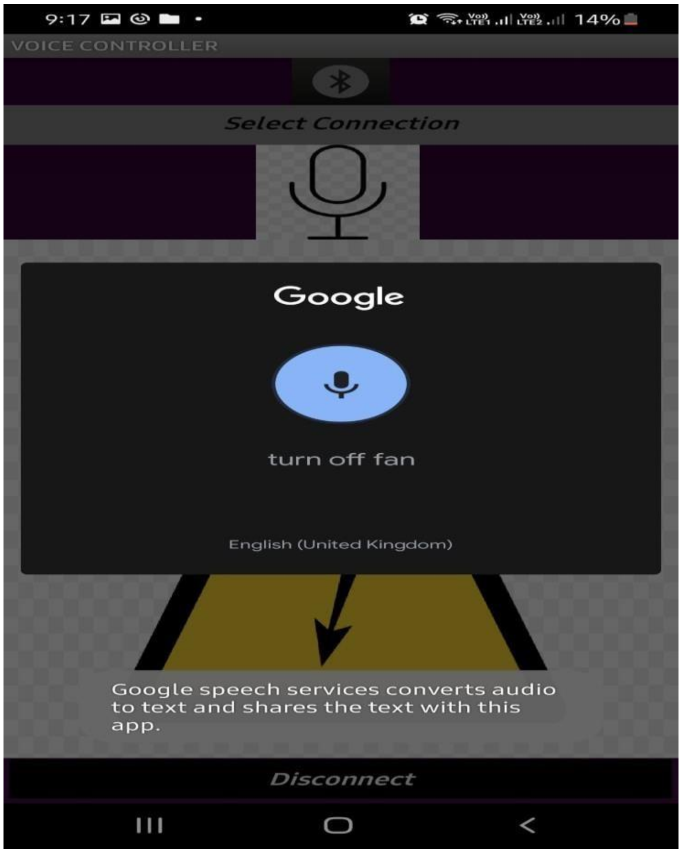


Fig 5.5:Fan Gets turned On



turn off fan

Google speech services converts audio to text and shares the text with this app.

Fig 5.6:command “turn off fan”

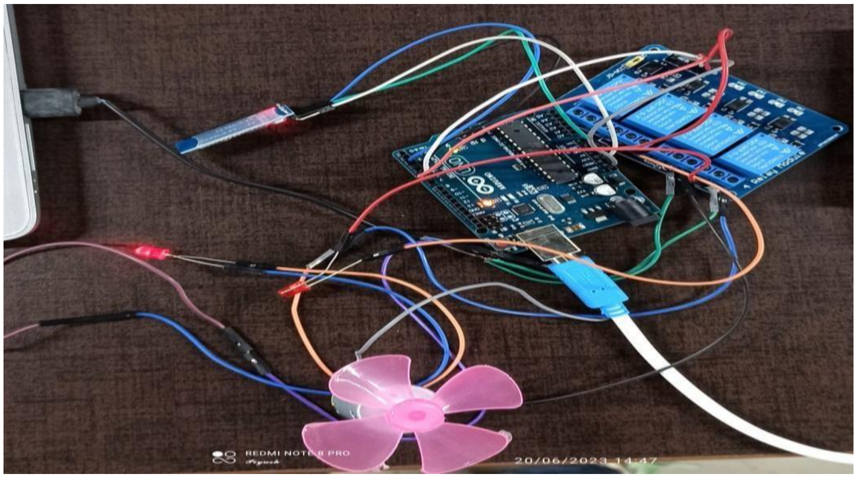
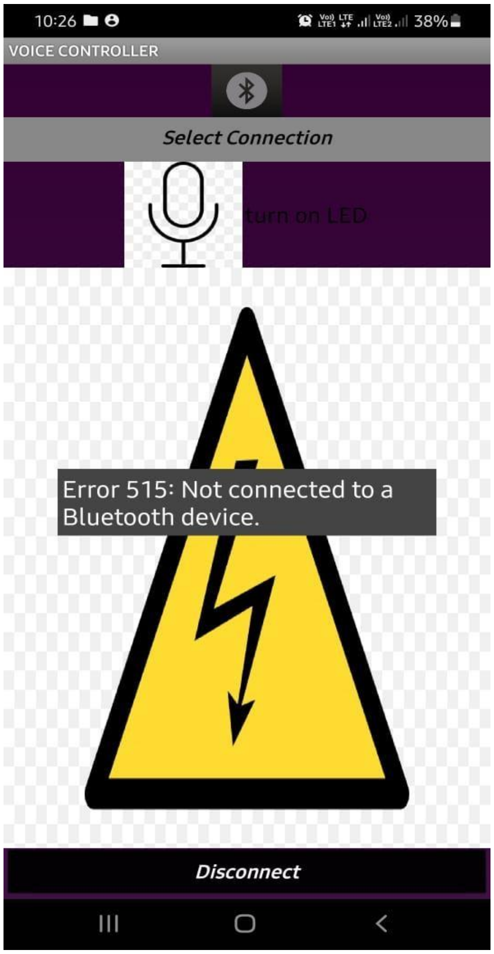


Fig 5.7:Fan Gets turned off Here the fan is turned off.

These screenshots are demo example of one of the working of the model.



Error 515: Not connected to a Bluetooth device.

Fig 5.8:No Bluetooth connection

If there is no Bluetooth connection, then the google voice recognition does not work. It will say that there it is not connected to a Bluetooth device.

6.Summary and Conclusions

6.1 Summary of achievements

Implementation of a voice recognition system that successfully interprets and understands user voice commands. Ability to control a wide range of devices and appliances using voice commands, such as lights, fans, switch, and bulbs. Communication between the voice recognition system and the linked devices is established through the seamless integration of Bluetooth modules and Arduino microcontrollers.

6.2 Main Difficulties Encountered and How they were Tackled

Accurate voice recognition is among the biggest challenges. Since Arduino boards often have low processing capacity, it can be difficult to build reliable voice recognition algorithms on an Arduino board. Also, it can be difficult to establish a steady and dependable Bluetooth connection between the Arduino and the controlling device (such as a smartphone). We must make sure to select a dependable Bluetooth module and adhere to the right wiring and connecting instructions offered by the module's manufacturer if we want to guarantee a steady connection. To handle any connection breaks, one should also think about adding error handling and reconnection capabilities to the Arduino code. Since Arduino boards have a limited amount of power, poor power management may cause them to operate inconsistently or require regular battery replacement. In order to combat this, we optimize the power usage by reducing pointless sensor or component activation and considering effective power sources like external power supplies or rechargeable batteries.

6.3 Limitation of the project

- Voice Recognition Accuracy: Due to the limited processing capacity of Arduino boards, accurate and reliable voice recognition can be difficult to implement. The effectiveness of the microphone and the speech recognition algorithm utilised have a significant impact on voice recognition accuracy. Higher-end technology or the use of third-party voice recognition services can be necessary to achieve high speech recognition accuracy.

- Limited Vocabulary and Command Syntax: The range of voice commands that Arduino-based systems can understand may be constrained by memory and processing restrictions. It may be difficult to implement instructions that are lengthy or complex with precise syntax.
- Hardware Restrictions: The number of input/output pins, memory, and processing power on Arduino boards is restricted. The number of devices that can be operated simultaneously may be limited as a result. It could be necessary to add extra hardware or use intricate wire arrangements to scale the system up or add more devices.
- Integration with Other Systems: Using Arduino and Bluetooth modules, voice-based home automation systems may encounter difficulties interacting with other smart home systems or protocols. Depending on the particular devices and protocols utilised, compatibility problems and the requirement for bespoke integration may occur.

6.4 Future Scope of Work

- Device Compatibility: Future innovations may concentrate on enhancing device compatibility as the Internet of Things (IoT) ecosystem develops. This involves combining Arduino-based systems with a larger selection of smart home appliances, door locks, security systems, and thermostats to provide voice control of a greater range of products.

- Voice Assistant Integration: A smooth user experience can be achieved by integrating voice-based home automation systems with well-known voice assistants like Amazon Alexa or Google Assistant. Through this interface, users can use the infrastructure and capabilities of voice assistant platforms to operate Arduino-based devices with voice commands.

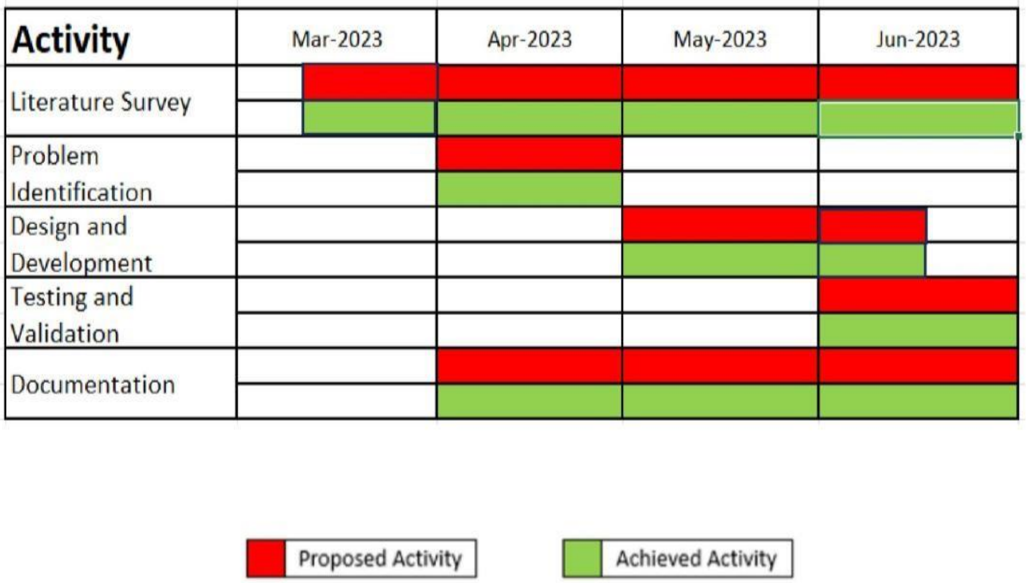
- Multi-Language Support: By adding more languages, users of voice-based home automation systems from various geographical and linguistic origins will be able to communicate with the system in their own tongue. In order for the system to recognise and react to voice instructions in many languages, language models must be created.

- Energy Efficiency and Sustainability: Upcoming improvements may put an emphasis on reducing energy use and incorporating energy-management tools. To make voicebased home automation systems more sustainable, this entails putting into practise intelligent algorithms that optimise device consumption based on user habits, energysaving techniques, and incorporating renewable energy sources.

- Remote connectivity and cloud connectivity allow for remote control of home automation systems, giving consumers access to their systems from anywhere. This entails combining cloud platforms with Arduino-based systems to enable users to access and manage their gadgets remotely using voice commands.

Using an Arduino and a Bluetooth module, voice-based home automation has the potential to expand device compatibility, enhance voice recognition accuracy, integrate with voice assistants, support multiple languages, improve security and personalization, maximise energy efficiency, and offer cutting-edge user interfaces and remote-control capabilities.

7.Gantt Chart



Activity	Mar-2023	Apr-2023	May-2023	Jun-2023
Literature Survey	🟥	🟥	🟥	🟥
	🟩	🟩	🟩	🟩
Problem Identification		🟥		
		🟩		
Design and Development			🟥	🟥
			🟩	🟩
Testing and Validation				🟥
				🟩
Documentation		🟥	🟥	🟥
		🟩	🟩	🟩

🟥 Proposed Activity 🟩 Achieved Activity

Fig 7.1 : Gantt Chart

8.References

[1].A. Ahmim, T. Le, E. Ososanya and S. Haghani, "Design and implementation of a home automation system for smart grid applications," 2016 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2016.

[2]M.Field DEC2017. [Online]. Available: http://www.telegraph.co.uk/technology/0/amazonecho-google-home-best-smart-home-devices-2018/.

[3]M. G. S. M. R. K. A. K. Kim Baraka, “Smart Power Management System for Home Appliances and Wellness Based on Wireless Sensors Network and Mobile Technology,” in XVIII AISEM Annual Conference, 2015.

[4] N. N. Akhila, N. B. Kanth, and K. Srinivasan, "Smart home automation using voice recognition and IoT," 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Noida, India, 2018.

[5] K. Dey, A. K. Nandi, and S. Das, "Voice controlled home automation using IoT," 2017 4th International Conference on Advances in Computing, Communications, and Informatics (ICACCI), Udupi, India, 2017.

[6] Y. Li, Y. Wang, and C. Zhang, "Voice recognition home automation control system based on Internet of Things," 2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE), Singapore, 2016