

Bio Medical & Instrumentations

BME - 309

B. Tech – 5th Semester

Section – D

Report File

Voice controlled Home Automation for Elderly and Disabled Person



DELHI TECHNOLOGICAL UNIVERSITY

Submitted To:
Prof. Anukul Pandey

Submitted By:
Lokesh 2K19/EC/101
Mohammad Sameer 2K19/EC/110

DECLARATION

WE, **LOKESH AND MOHAMMAD SAMEER**, hereby solemnly declare that project report “ **Voice Controlled Home Automation for elderly and Disabled Person**” is based on my work carried out during the course of study under the supervision of **PROF. Anukul Pandey**.

We assert the statements made and conclusions drawn are an outcome of my research work.

We further certify that:

- I. The work contained in the report is original and has been done by me under the general supervision of my supervisor.
- II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
- III. I have followed the guidelines provided by the university in writing the report.
- IV. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them in the text of the report and giving their details in the references.

SUPERVISOR

Anukul Pandey Sir

MADE BY

Lokesh (2K19/EC/101)

Mohammad Sameer (2K19/EC/110)

CERTIFICATE

This is to certify that the project report entitled **“Voice Controlled Home Automation for elderly and Disabled Person”**, submitted to the Department of Electronics and Communication Engineering, Delhi Technological University, in partial fulfilment for the award of the degree of B.Tech , is a record of work carried out by **Lokesh (2K19/EC/101) and Mohammad Sameer (2K19/EC/110)**.

No part of this report has been submitted elsewhere for award of any other degree.

SUPERVISOR

Prof. Anukul Pandey

SUBMITTED BY

Lokesh (2K19/EC/101)

Mohammad Sameer (2K19/EC/110)

ACKNOWLEDGEMENT

We would like to take this opportunity to give special thanks to Mr. Anukul Pandey Sir for giving me the golden opportunity to perform this project. We are so thankful for all the new things that we got to learn through this project. All the research papers and documents that we went through in order to complete this project were extremely helpful. We are sure that these things will prove to be of great use in the future as well. Once again, we would like to thank sir for always being so supportive and helpful. None of this would have been possible without his guidance.

A special thanks to all our colleagues who have been so supportive during this time. We feel that words will fall short in expressing how grateful we that we got to perform this project and that we were given the chance to explore and expand my horizons.

Thank you everyone.

Regards

Lokesh (2K19/EC/101)

Mohammad Sameer (2K19/EC/110)

CONTENTS

- 1. Declaration**
- 2. Certificate**
- 3. Acknowledgment**
- 4. Contents**
- 5. Introduction**
- 6. Components Required**
- 7. Components Discription**
- 8. Proposed Model of Project**
- 9. Circuit Design**
- 10. Working**
- 11. Arduino Code**
- 12. Hardware Model**
- 13. Applications**
- 14. Conclusion**
- 15. Future Scope**

Introduction:

- ❖ In recent years there has been an increase in smart home technology. Home automation has become the new trend. It allows us to control all our home appliances (lights, fans, thermostat, TV, security cameras) by connecting them to a common remotely controllable network making it much more accessible and convenient. And, while it brings home management to a whole new level, it also helps to maximize home security.
- ❖ Here we will discuss a Voice Activated Home Automation System, how to design a voice activated home automation system that uses voice command to control your appliances. This project is designed by incorporating Arduino UNO, Bluetooth, and a smartphone device, along with other components.

How it is useful for disabled person :

- We can monitor our home appliances without moving here and there . It works on Bluetooth range .It is very useful for persons who are not able to move .
- This problem is motivation to numerous studies also in robotics and automation. Instead of moving to care institutions, it is more desirable and often economically viable to live at home as long as possible. We are developing technology for the elderly to help them to live longer at their own homes.

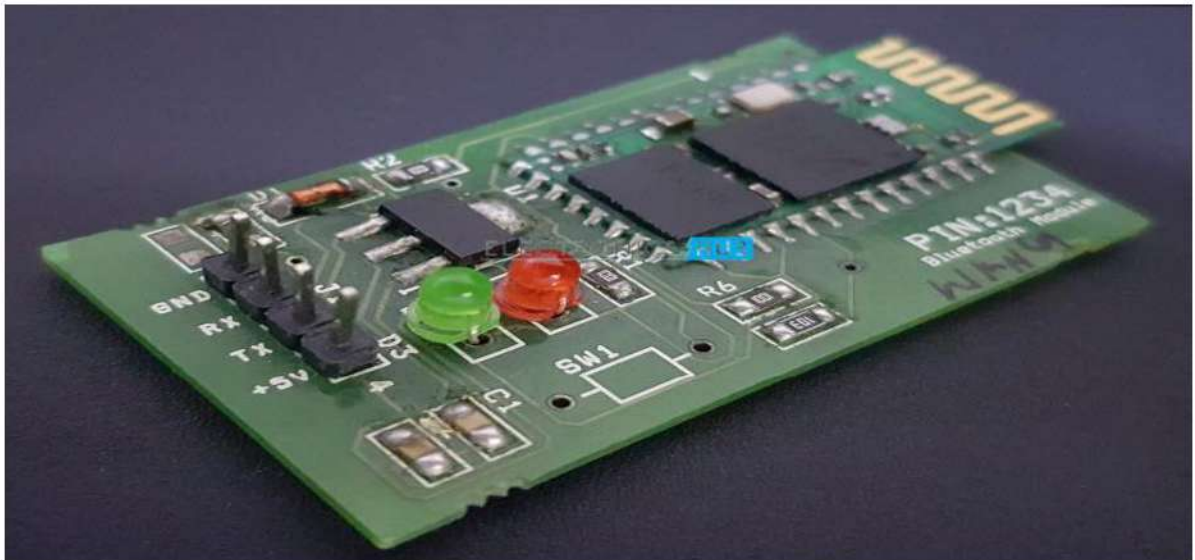
Components Required :

- Arduino UNO – 1
- HC – 05 Bluetooth Module – 1
- Smart Phone or Tablet – 1
- 2N2222 NPN Transistor – 4
- 12V Relay – 4
- 1 K Ω Resistor – 4
- 1N4007 PN Junction Diode – 4
- Power Supply
- Connecting Wires
- Breadboard (Prototyping Board)
- App for transmitting voice to Bluetooth

Components Description :

1. Bluetooth HC-05:

- For this wireless home automation, we will use Bluetooth module HC-05. HC-05 is an easy to use Bluetooth SPP module (Serial Port Protocol) for wireless connection setup. It can be used as either Master or Slave configuration. The default settings are SLAVE. The slave module can only accept connections from another device but cannot connect to another Bluetooth device. On the other hand, a MASTER module can connect to other devices. The module uses UART interface to communicate and uses a bit rate of 9600 bps when at default mode. In the Bluetooth HC-05 module, there are pins for VCC (5V), GND, TX, and RX.



What is an HC 05 Bluetooth Module?

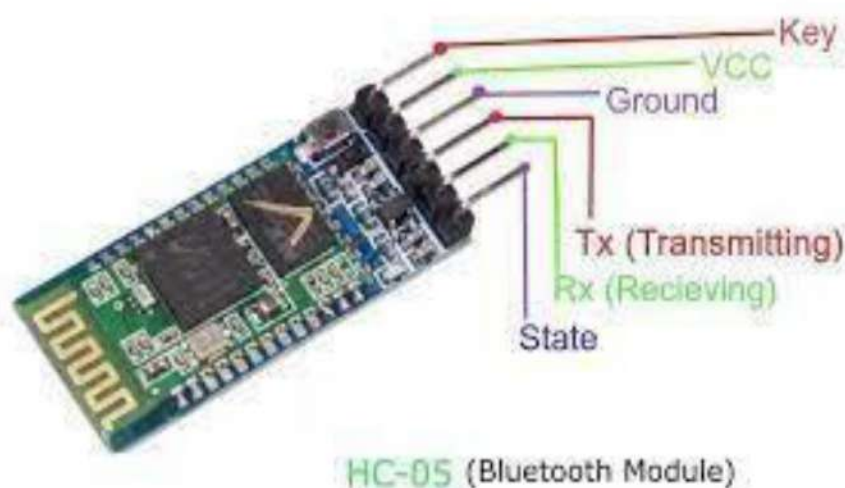
- HC 05 Bluetooth is a wireless communication protocol; it is used in two devices as a sending and receiving the information. The Bluetooth is free to use in the wireless communication protocol as the range of the Bluetooth is less than the other wireless communication protocols like WiFi and Zigbee. The Bluetooth operates at the frequency of the 2.41 GHz and also used in many small ranges of applications.

The following are some specifications of the H05 Bluetooth module:

- The frequency is about the ISM band of 2.4GHz.
- The emission power is about the <4db
- The input power supply is from the 3.6V To 6v.
- The operating voltage of the HC 05 Bluetooth module is 3.3V.
- The HC 05 Bluetooth module works at the temperature of -200C ~ +750C.
- The security of this Bluetooth module is in authentication and encryption.

Description of pins

- **Enable** - This pin is used to set the Data Mode or and AT command mode (set high).
- **VCC** - This is connected to +5V power supply.
- **Ground** - Connected to ground of powering system.
- **Tx (Transmitter)** - This pin transmits the received data Serially.
- **Rx (Receiver)** - Used for broadcasting data serially over bluetooth.
- **State** -Used to check if the bluetooth is working properly.



Modes of Operation

The HC-05 Bluetooth Module can be used in two modes of operation: Command Mode and Data Mode.

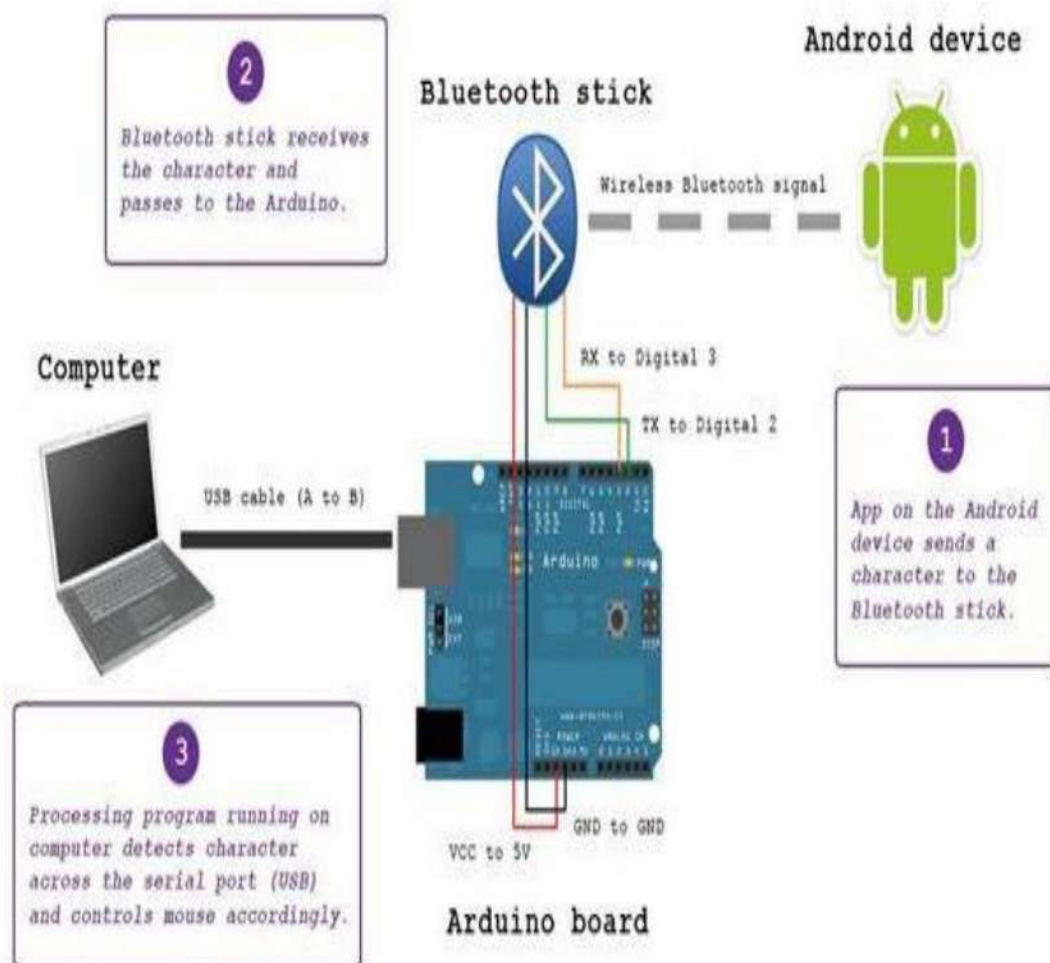
Command Mode

In Command Mode, you can communicate with the Bluetooth module through AT Commands for configuring various settings and parameters of the Module like get the firmware information, changing Bit Rate, changing module name, it can be used to set it as master or slave.

A point about HC-05 Module is that it can be configured as Master or Slave in a communication pair. In order to select either of the modes, you need to activate the Command Mode and sent appropriate AT Commands.

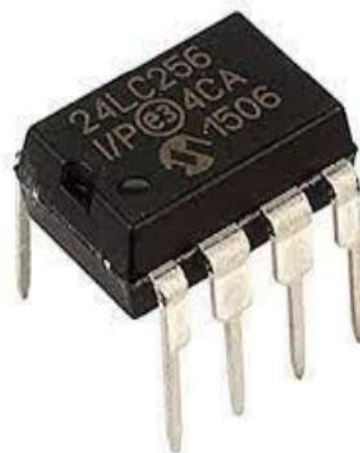
Data Mode

Coming to the Data Mode, in this mode, the module is used for communicating with other Bluetooth device i.e. data transfer happens in this mode.



2. EEPROM :

- EEPROM is the acronym for Electronically Erasable Programmable Read Only Memory, which is used here to store the status of the relay (ON/OFF), so that in case of a power failure when the controller resets, all the relays which were kept ON will come back to their ON position after the power has come up. So whenever a relay is turned on, a variable stored in an address in the EEPROM changes its value to 1 and whenever it's turned OFF the same variable changes to 0. Each relay has its own variable assigned to store its status in the EEPROM. So at the beginning of the code the first thing we did was to initialize the relays according to the values stored in the EEPROM.
- It is a non-volatile flash memory device, that is, stored information is retained when the power is removed. EEPROM generally offers excellent capabilities and performance. In EEPROM we can write and program the IC for many times and these are acting as EPROM (UV erasable programming ROM).



Principle of operation of EEPROM

- The EEPROM uses the principle same as that of the UV-EPROM. The electrons which are trapped in a floating gate will modify the characteristics of the cell, so instead of that logic "0" or logic "1" will be stored.
- EEPROM is the memory device which implements the fewest standards in cell design. Most of the common cells are composed of two transistors. In this the storage transistor has the floating gate that will trap the electrons. Apart from that there is an access transistor which is used in the

operation. In EPROM , cell is erased when electrons are removed from the floating gate, whereas in EEPROM , cell is erased when electrons are trapped in the floating cell.

Features of EEPROM:

- Low and standard voltage operation (100 kHz (1.8V) and 400 kHz (2.7V, 5V) compatibility)
- Two-wire serial interface (Two wires are used to read and write data into it)
- Bidirectional data transfer protocol
- Internally Organized 128 x 8 (1K), 256 x 8 (2K), 512 x 8 (4K), 1024 x 8 (8K) or 2048 x 8 (16K)

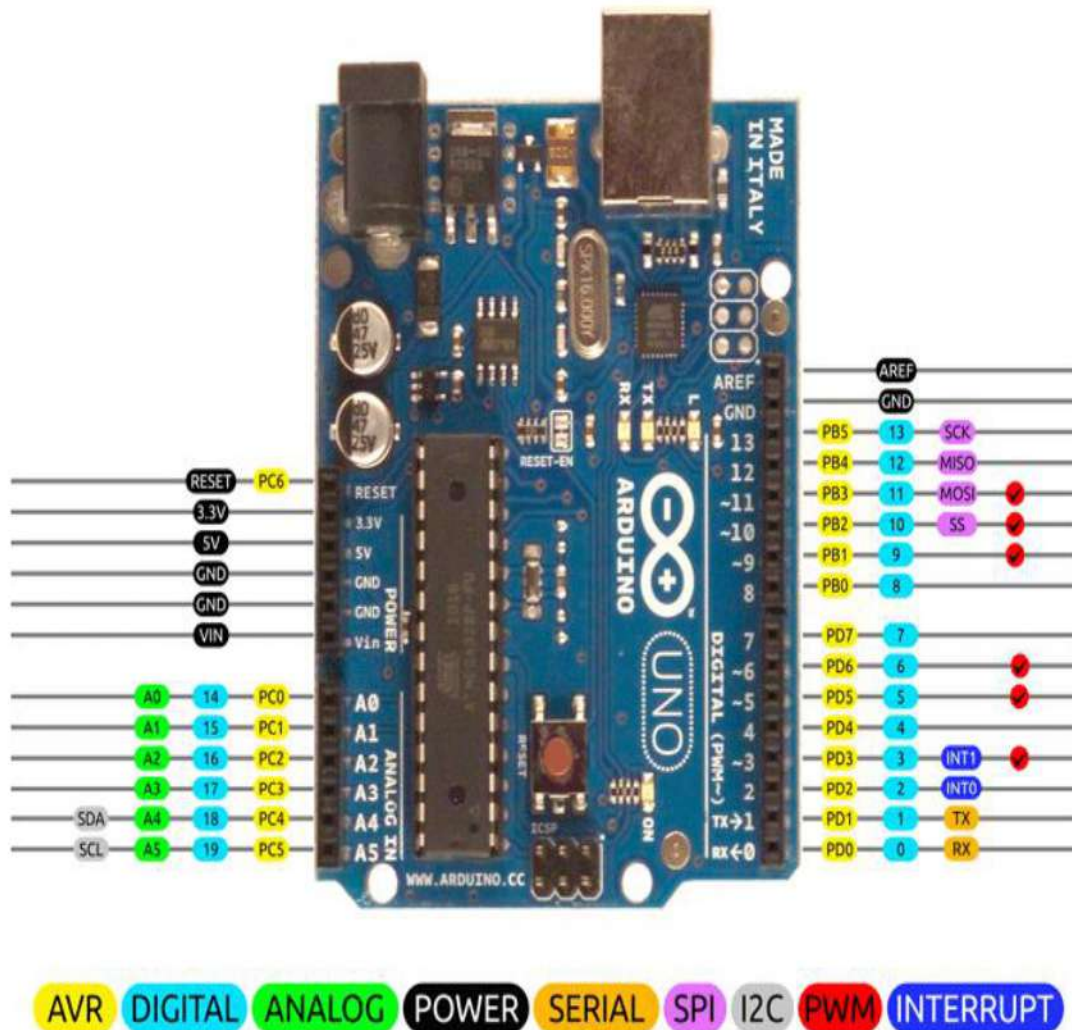
3. Arduino Uno

Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

Specifications of Arduino UNO:

- **Microcontroller:** ATmega328p
- **Operating Voltage:** 5V
- **Input Voltage (recommended):** 7-12V
- **Input Voltage (limits):** 6-20V
- **Digital I/O Pins:** 14 pins (of which 6 are PWM output pins)
- **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 is taken by bootloader)

- **EEPROM:** 1 KB (ATmega328)
- **Clock Speed:** 16 MHz



Voltage Regulator-The voltage regulator converts the input voltage to 5V. The primary use of a voltage regulator is to control the voltage level in the Arduino Board. Even if there are any fluctuations in the input supply voltage of the regulator, the output voltage remains constant and near 5 volts.

Barrel Jack – The Barrel jack or DC Power Jack is used to power the Arduino board using an external power supply. The barrel jack is usually connected to an adapter. The board can be powered by an adapter that ranges between 5-20 volts but the manufacturer recommends keeping it between 7-12 volts.

Digital Pins:

On the Arduino UNO board, pins 0-13 are digital input/output pins.

The Arduino digital pins can read only two states: when there is a voltage signal and when there is no signal. This kind of input is usually called digital (or binary) and these states are referred to as HIGH and LOW or 1 and 0.

Analog Pins:

The Arduino Uno consists of 6 analog pins, which use ADC (Analog to Digital converter). These pins can serve as analog inputs but can also function as digital inputs or digital outputs. These pins accept inputs in the form of Analog signals and return values that range between 0 and 1023 (because the Arduino Uno has a 10-bit Analog to Digital converter or 2^{10} resolution).

Communication :

Arduino can be used to communicate with a computer, another Arduino board or other microcontrollers. The ATmega328P microcontroller provides UART TTL (5V) serial communication which can be done using digital pin 0 (Rx) and digital pin 1 (Tx). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The ATmega16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. There are two RX and TX LEDs on the arduino board which will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328P also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

4. Relay Module :

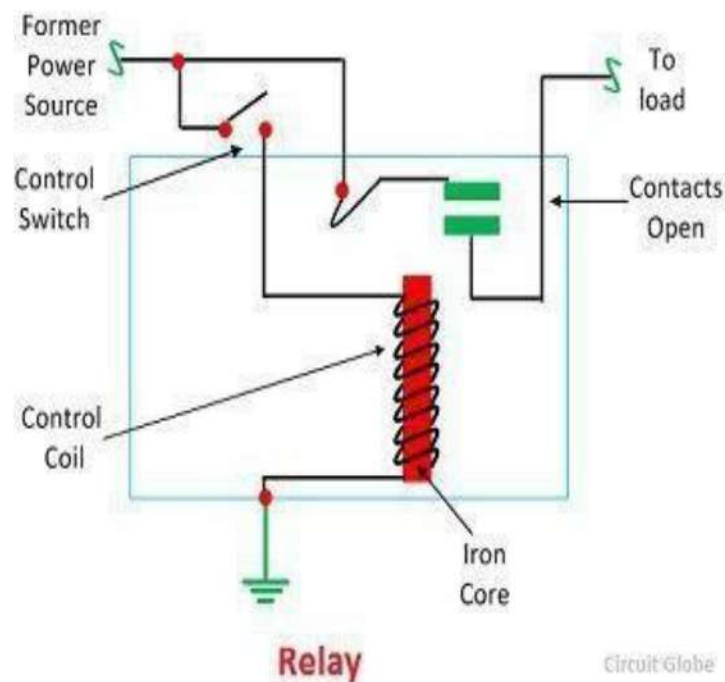
It works on the principle of an electromagnetic attraction. When the circuit of the relay senses the fault current, it energises the electromagnetic field which produces the temporary magnetic field.

This magnetic field moves the relay armature for opening or closing the connections. The small power relay has only one contacts, and the high power relay has two contacts for opening the switch.

The inner section of the relay is shown in the figure below. It has an iron core which is wound by a control coil. The power supply is given to the coil through

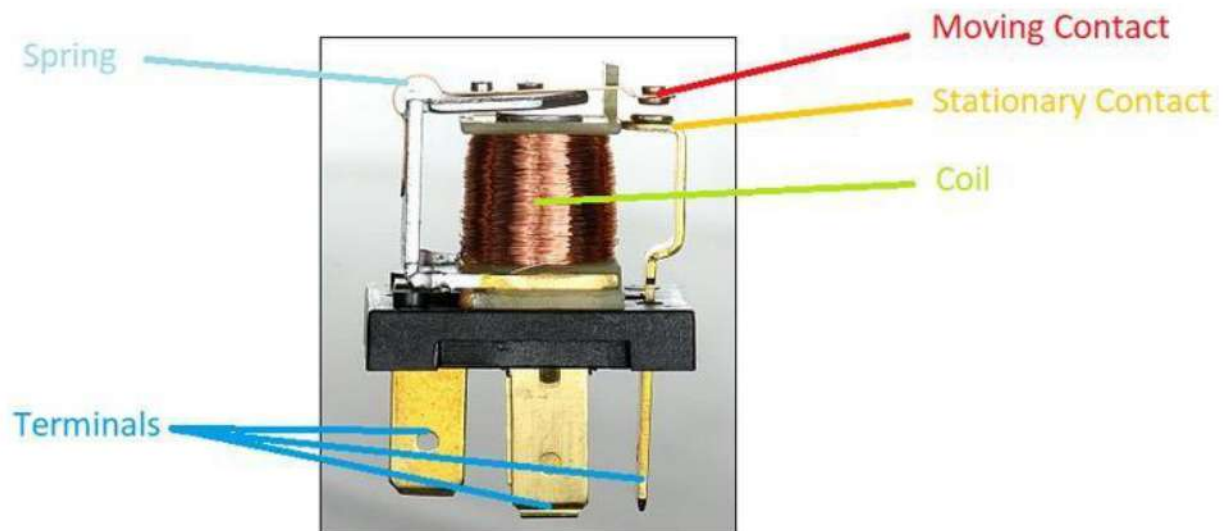
the contacts of the load and the control switch. The current flows through the coil produces the magnetic field around it.

Due to this magnetic field, the upper arm of the magnet attracts the lower arm. Hence close the circuit, which makes the current flow through the load. If the contact is already closed, then it moves oppositely and hence open the contacts.



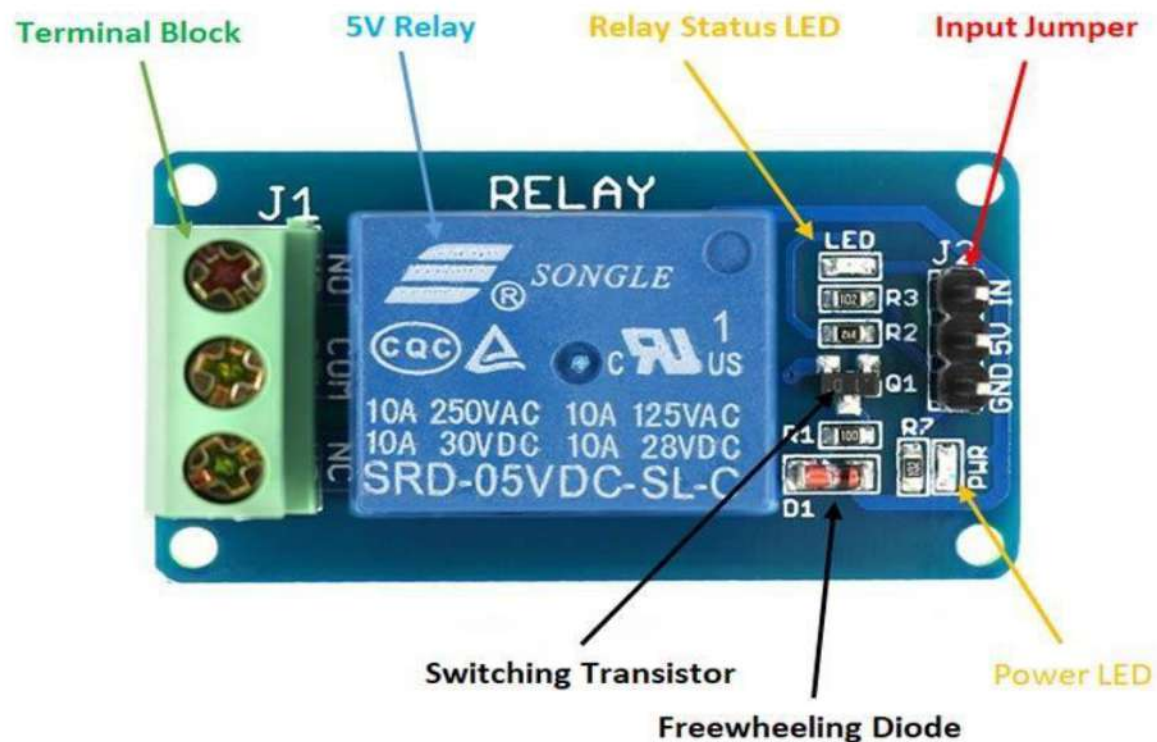
How Does A Relay Work?

- The relay uses an electric current to open or close the contacts of a switch. This is usually done using the help of a coil that attracts the contacts of a switch and pulls them together when activated, and a spring pushes them apart when the coil is not energized.
- There are two advantages of this system – First, the current required to activate the relay is much smaller than the current that relay contacts are capable of switching, and second, the coil and the contacts are galvanically isolated, meaning there is no electrical connection between them. This means that the relay can be used to switch mains current through an isolated low voltage digital system like a microcontroller.



Relay Module Specifications

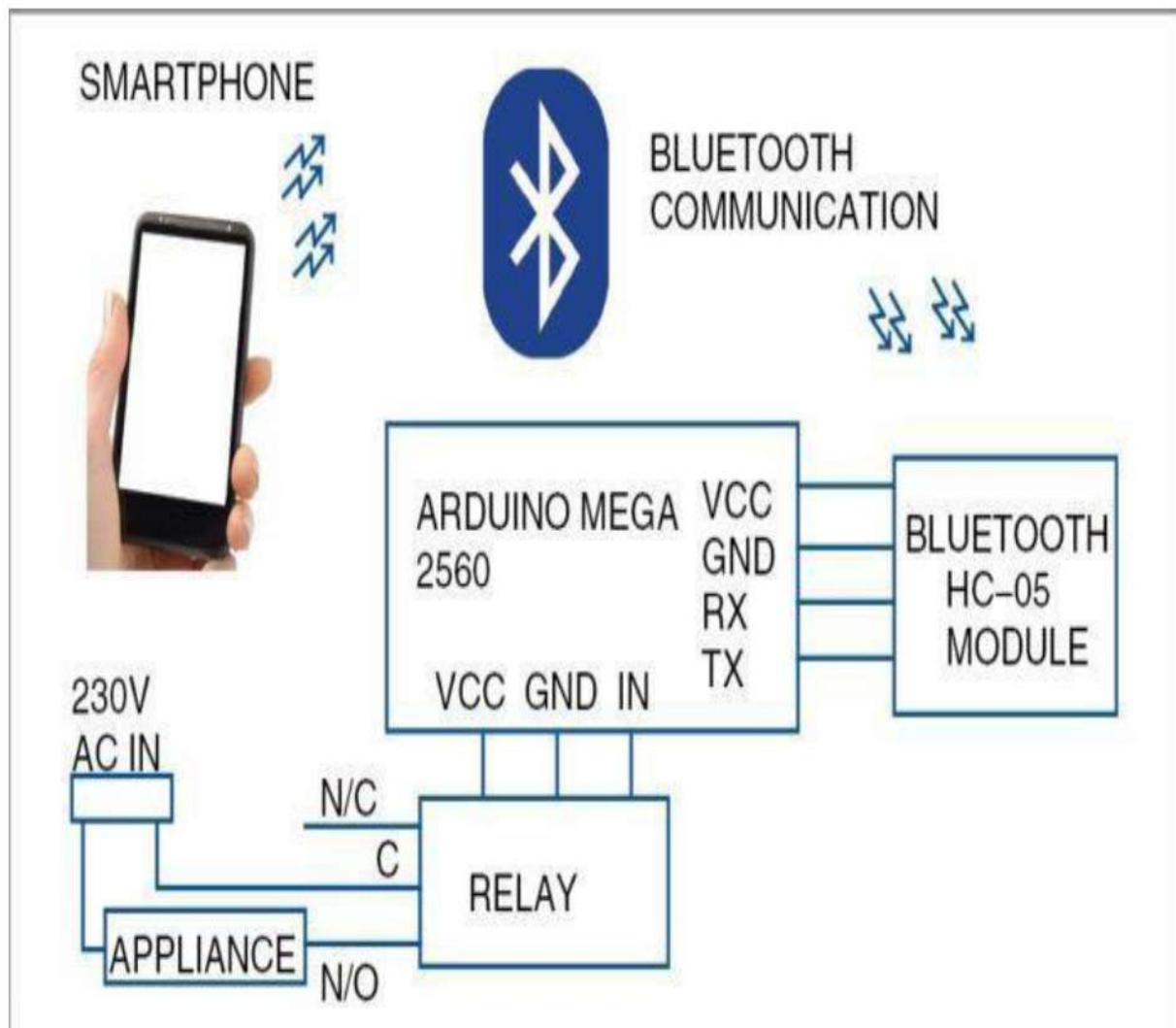
- Supply voltage – 3.75V to 6V
- Quiescent current: 2mA
- Current when the relay is active: ~70mA
- Relay maximum contact voltage – 250VAC or 30VDC



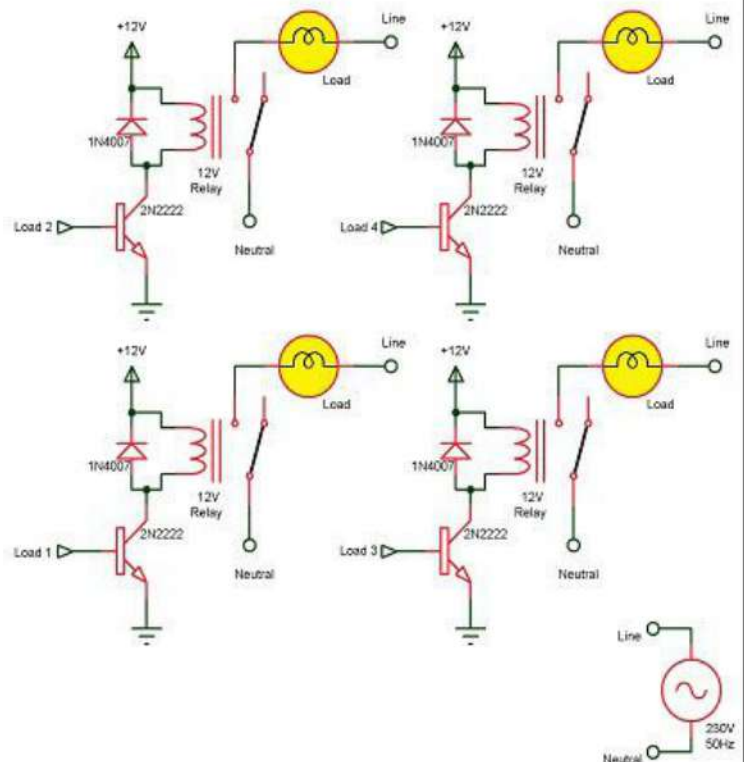
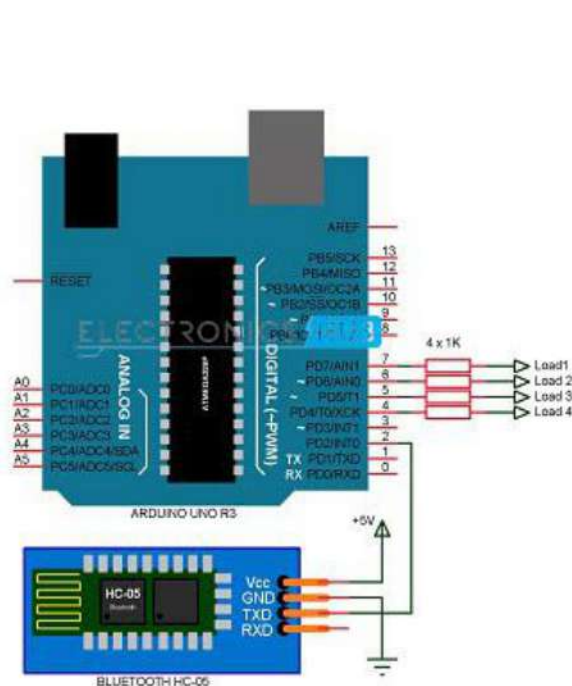
- Relay maximum current – 10A

BT Voice Control for Arduino: This app is developed by SimpleLabsIN for voice based Arduino projects. This Android App will use the phone's voice recognition feature and will convert the voice commands to text and transfer the string via Bluetooth.

PROPOSED MODEL OF PROJECT :



Circuit Diagram :



Circuit Design :

We will now see the design of the Voice Activated Home Automation circuit. First, we will connect the Bluetooth Module to the Arduino. Since Bluetooth uses UART protocol, we need to use the RX and TX pins of the Arduino. We will be using “SoftwareSerial” library to define our own RX and TX pins (Pin 2 is RX and Pin 3 is TX).

We have left out the Bluetooth’s RX and Arduino’s TX connection as it is not used. Then, connected a voltage divider to convert the Arduino TX’s 5V signal to Bluetooth RX’s 3.3V.

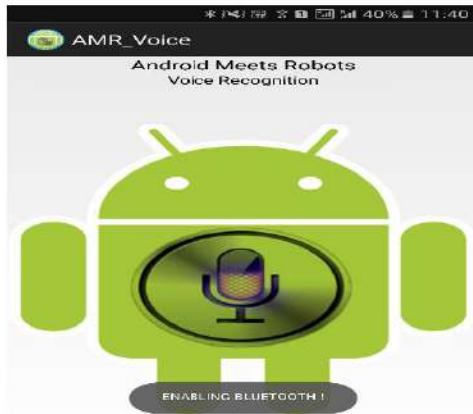
Next, we will connect the relays to the Arduino. Since we used a readymade relay board with 4 – channels, all we need to do is to connect the inputs of the individual relays to the Arduino. Then, we have connected some small loads and check out the functioning

All the necessary connections are explained in the circuit diagram.

Working :

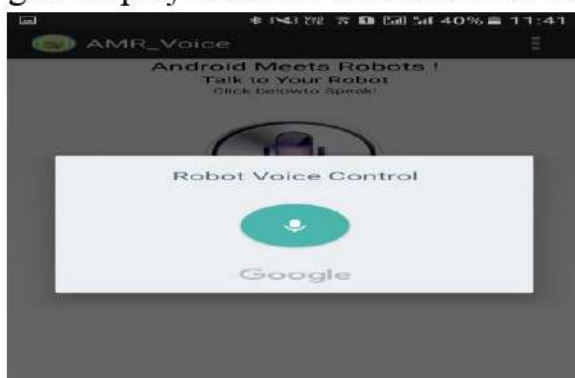
In this project, a simple Voice Activated Home Automation system is designed. Voice commands are used to control different appliances. We will now see the working of the project. All the connections are made as per the circuit diagram above.

After making the necessary connections, we have to switch on the power supply to the circuit. Now, we need to pair the Phone's Bluetooth to the HC – 05 Bluetooth Module. Before that, we have to install the App mentioned above in the phone. The home screen of the app looks something like this.



Next step is to connect the phone with the Bluetooth module. For this, choose the option “Connect Robot” 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.

After successful connection, the devices are ready to transmit data. For that, press the press microphone icon on the app and start giving voice commands. Make sure that the voice recognition feature is enabled on the phone (this is usually associated with Google app). For example, if we press the microphone icon and say “turn on light”, the app will recognise the command and the transfers it to the Bluetooth Module. Also, the command gets displayed on the screen for our reference.



When the string “turn on light” is detected by the app, it will send the string as “*turn on light#”. So, the actual message received by the Bluetooth Module is in the format of “*Message#”. The reason for padding the ‘*’ and ‘#’ at the beginning and end of the string is to identify the starting and ending of the message.

We are able to delete the ‘#’ from the string but left out the ‘*’ in order to identify the

starting of the string. The received message is compared with some predefined strings and if the message matches with any of them, then corresponding action like turning on or turning off the load happens.

We have used the following commands: “turn on AC”, “turn off AC”, “turn on light”, “turn off light”, “turn on TV”, “turn off TV”, “turn on fan”, “turn off fan”, “turn on all” and “turn off all”.

We have set conditions in the code of Arduino according to that our whole circuit will on and off . the basic part of our code is to set to voltage high and low according to on and off of circuit.

Code :

```
#include <SoftwareSerial.h>
const int rxPin = 2;
const int txPin = 3;
SoftwareSerial mySerial(rxPin, txPin);
```

```
int ac=4;
int light=5;
int fan=6;
int tv=7;
String data;
```

```
void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600);

  pinMode(ac, OUTPUT);
  pinMode(light, OUTPUT);
  pinMode(fan, OUTPUT);
  pinMode(tv, OUTPUT);

  digitalWrite(ac, LOW);
  digitalWrite(light, LOW);
  digitalWrite(fan, LOW);
  digitalWrite(tv, LOW);
}
void loop()
```

```

{
    int i=0;
    char ch=0;
    data="";
    while(1)
    {
        while(mySerial.available()<=0);
        ch = mySerial.read();
        if(ch=='#')
            break;
        data+=ch;
    }
    Serial.println(data);

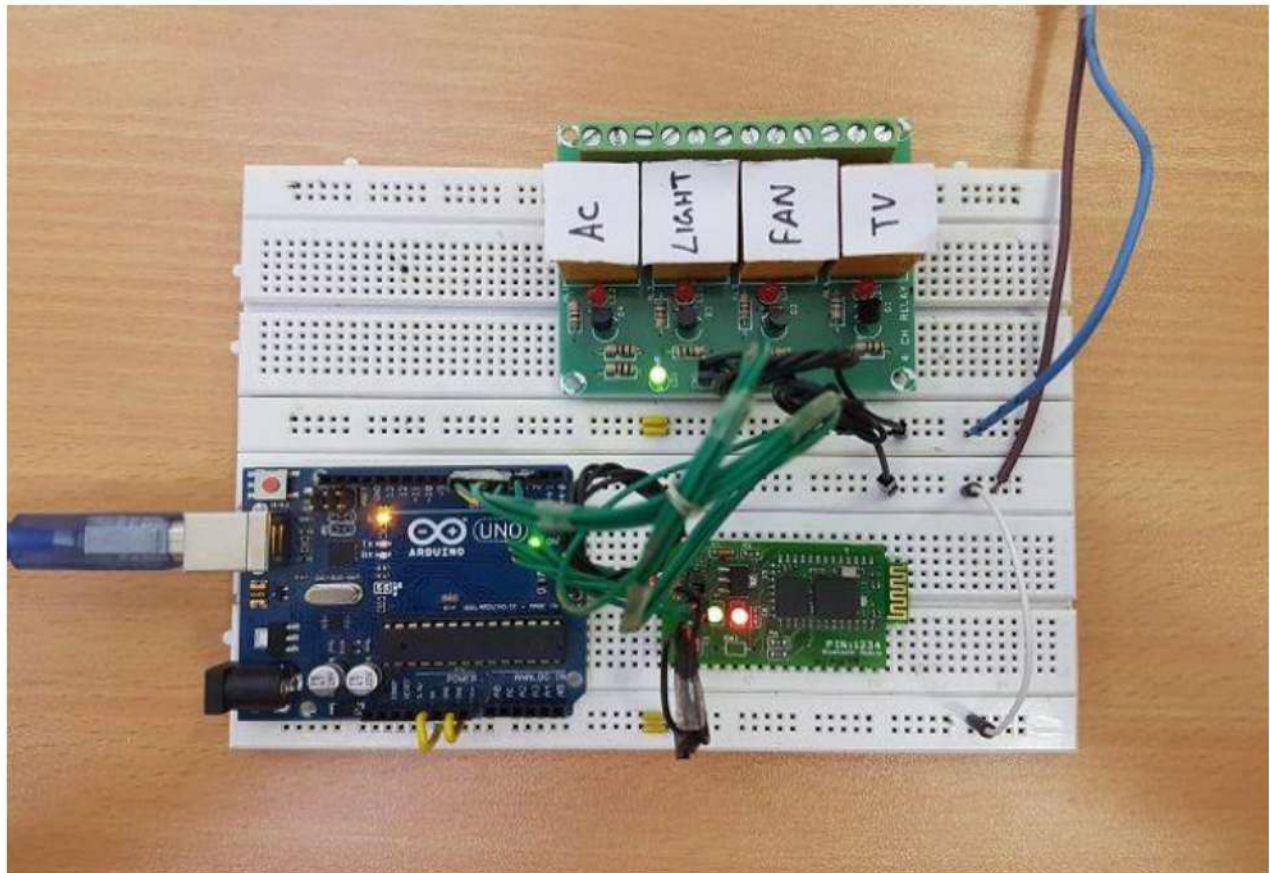
    if(data=="*turn on AC")
    {
        digitalWrite(ac,HIGH);
        Serial.println("ac on");
    }
    else if(data=="*turn off AC")
    {
        digitalWrite(ac,LOW);
        Serial.println("ac off");
    }
    else if(data=="*turn on light")
    {
        digitalWrite(light,HIGH);
        Serial.println("light on");
    }
    else if(data=="*turn off light")
    {
        digitalWrite(light,LOW);
        Serial.println("light off");
    }
    else if(data=="*turn on fan")
    {
        digitalWrite(fan,HIGH);
        Serial.println("fan on");
    }
    else if(data=="*turn off fan")
    {
        digitalWrite(fan,LOW);
    }
}

```



```
        Serial.println("fan off");
    }
    else if(data=="*turn on TV")
    {
        digitalWrite(tv,HIGH);
        Serial.println("tv on");
    }
    else if(data=="*turn on TV")
    {
        digitalWrite(tv,LOW);
        Serial.println("tv off");
    }
    else if(data=="*turn on all")
    {
        digitalWrite(ac,HIGH);
        digitalWrite(light,HIGH);
        digitalWrite(fan,HIGH);
        digitalWrite(tv,HIGH);
        Serial.println("all on");
    }
    else if(data=="*turn off all")
    {
        digitalWrite(ac,LOW);
        digitalWrite(light,LOW);
        digitalWrite(fan,LOW);
        digitalWrite(tv,LOW);
        Serial.println("all off");
    }
}
```

Hardware Model :



Applications of Home Automation System:

- It helps to control a number of appliances through a common network easily by simple voice commands. You can manage all your devices from one place. This system will prove to be highly useful for people with disabilities. With voice-activated home automation system, you can make your space more energy efficient and can improve your home management.
- Further, the project can be expanded by adding different sensors (light, smoke, etc.). So, that whenever there is a gas or anything leakage we can directly control through our mobile.

Conclusion and Future scope :

- As in this project we have work on Bluetooth module as a communication protocol to communicate with the appliances.
- The future scope of our project is that we can control our appliances on a cloud that has no distance limitation for ex: if we have gone to a market and we forget to switch off our fan or ac or any other appliances then we get a notification so, we can directly switch off that appliances by our mobile through cloud.