

SMART HOME AUTOMATION USING ARDUINO, ANDROID AND BLUETOOTH EEE F376

BY

AKASH CHINTHA

2014A3PS241P

BHANU TEJA

2014A3PS279P

INSTRUCTOR

ABHIJIT R ASATI



**ELECTRICAL AND ELECTRONICS ENGINEERING
BITS-PILANI**

TABLE OF CONTENTS

ABSTRACT	3
KEYWORDS.....	3
INTRODUCTION	3
COMPONENTS	4
ARDUINO UNO	4
BLUETOOTH MODULE	5
RELAYS.....	6
ANDROID APPLICATION	7
CONNECTIONS	9
FLOW CHART FOR DESIGN MODEL	11
CODE FOR ARDUINO UNO.....	12
CONCLUSION	15
LINKS FOR COMPONENTS	16

I. Abstract

While the cost of living is going up, there is a growing focus to involve technology to lower those prices. Automation lowers the human judgment to the lowest degree possible but does not completely eliminate it. The project provides a low cost-effective and flexible home control and monitoring system with the aid of Bluetooth and Wireless Fidelity to access and control electrical and electronic equipment and devices remotely using Android-based smartphone application. The proposed system doesn't require a dedicated server with respect to similar systems and offers a direct communication protocol for monitoring and the home environment with more than just switching functionality. Depending on the number of devices to control within their specifications provided, we design the structure of connections accordingly.

Keywords

Arduino UNO, Android, Bluetooth, Relay, Voice-recognizer.

II. Introduction

Home automation is computerization of the home, housework or household action. Home automation may incorporate a control unit for controlling of lighting, HVAC (warming, ventilation and aerating and cooling), machines, and different frameworks, to give enhanced accommodation, solace, better energy saving ,productivity and security.

Home automation for the elderly and debilitated can give expanded personal satisfaction to persons who may generally need parental figures or institutional consideration. It can likewise give a remote interface to home apparatuses or the automation system itself, through phone line, remote transmission or the web, to give control and observe and monitor by means of a smart phones or a web explorer program.

By empowering consumers with near real-time information of their energy usage through an array of products and services, the intent is to help consumers use energy more efficiently.

III. Components

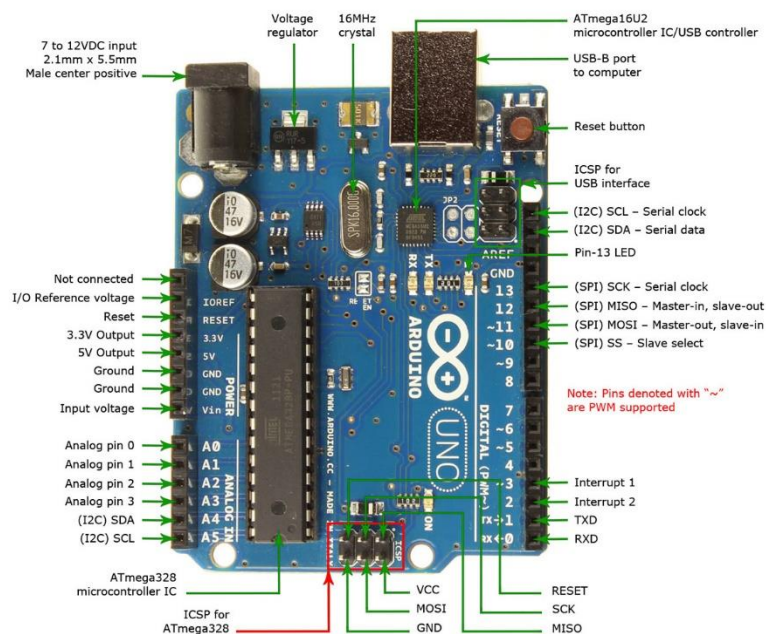
1) Arduino UNO

The Arduino board can be programmed to do anything by simply programming the microcontroller on board using a set of instructions for which, the Arduino board consists of a USB plug to communicate with your computer and a bunch of connection sockets that can be wired to external devices like motors, LEDs etc.

UNO is based on ATmega328P microcontroller. Arduino UNO comes with different features and capabilities. As mentioned earlier, the microcontroller used in UNO is ATmega328P, which is an 8-bit microcontroller based on the AVR architecture.

UNO has 14 digital input – output (I/O) pins which can be used as either input or output by connecting them with different external devices and components. Out of these 14 pins, 6 pins are capable of producing PWM signal. All the digital pins operate at 5V and can output a current of 20mA.

Pins 0 and 1 are used for serial communication. Pins 2 and 3 are used for external interrupts. Pins 3, 5, 6, 9, 10 and 11 can provide PWM output. Pins 10-13 are used for SPI communication. Pin 13 has a built-in LED connected to it. Arduino Uno has 6 analog input pins which can provide 10 bits of resolution.

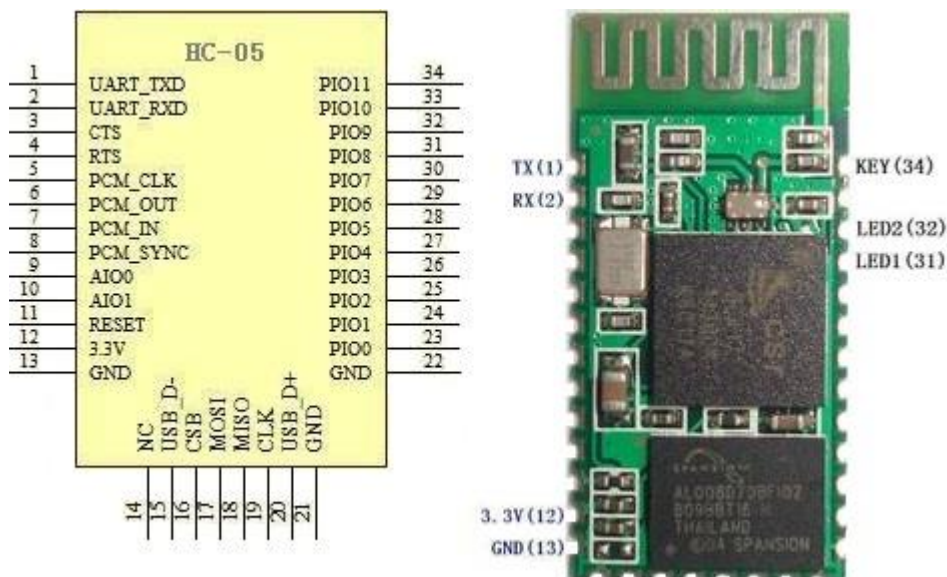


Arduino UNO is chosen due to the following:-

- Inexpensive
- Runs on multiple platforms
- C-programming
- Open Source
- Extensible programming and hardware

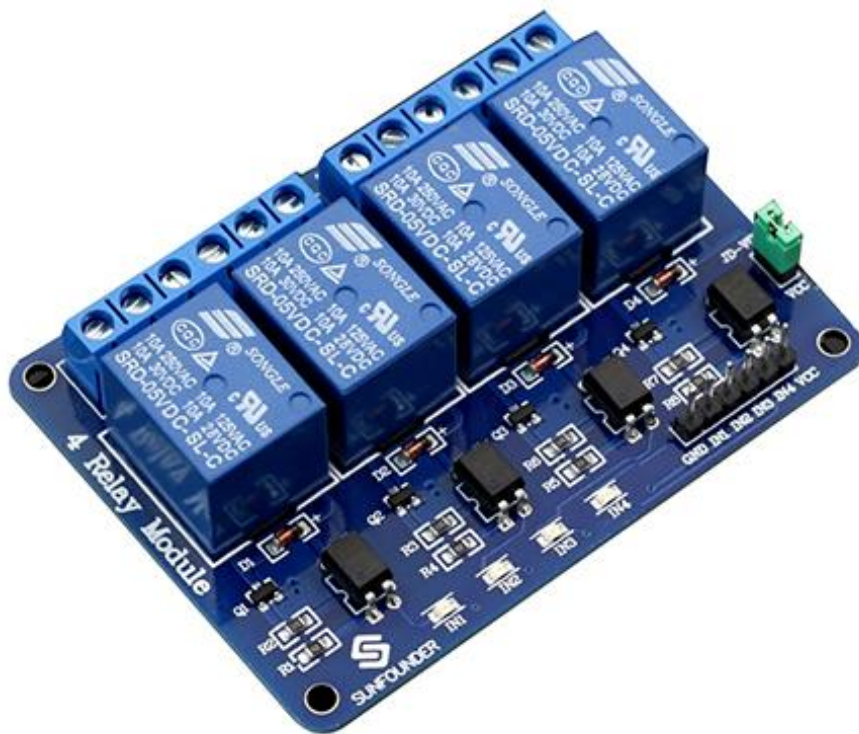
2) Bluetooth

Arduino-Uno board doesn't support Bluetooth connection on its own, which prevents the idea of connecting it wirelessly to an Android device. So, a medium between the Arduino-Uno board and android device is needed and here, a HC-05 Bluetooth module is used to serve this purpose. It is operated using AT commands.



3) Relays

A relay is an electromagnetic switch operated by a relatively small electric current that can turn on or off a much larger electric current. The heart of a relay is an electromagnet (a coil of wire that becomes a temporary magnet when electricity flows through it). We use a 4 channel 5V relay module to operate electrical appliances that run on low voltages.



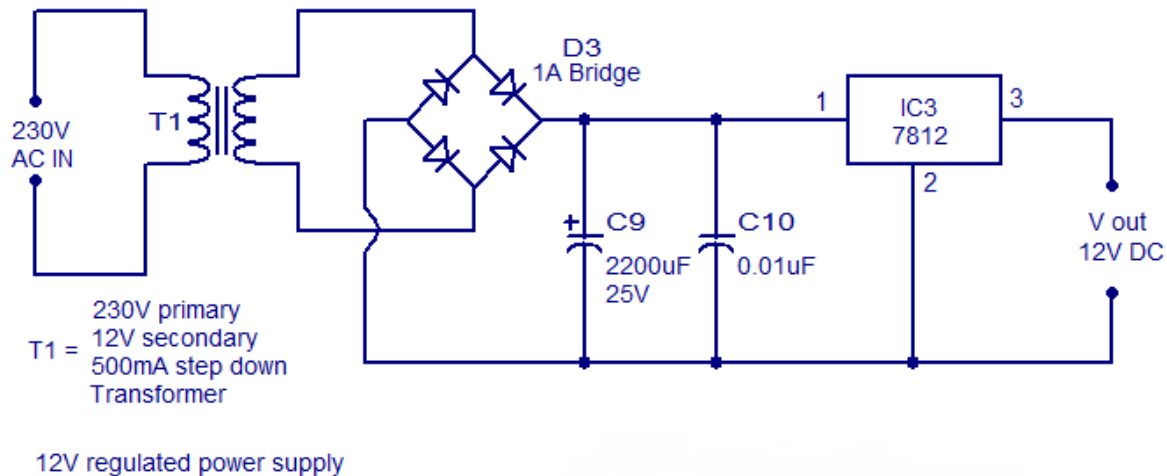
To run electrical appliances of higher voltages like 12V, we need to configure the design using 4 channel 12V relay module instead of 5V module. Unfortunately, Arduino UNO can't run higher relay switches. Hence, we need to establish an external power supply to drive the relay coil.

The usage of 12V relay channel is optional as it depends on the voltage specifications of the electrical appliances that we need to run using the microcontroller. **NOT USED IN THIS MODEL BECAUSE OF NO HIGH VOLTAGE IMPLEMENTATION.**

Design of 12V D.C. power supply:

Designing of a 12V D.C power supply takes four steps namely

- i) Transformer (Stepping Down)
- ii) Full wave Rectifier (AC to DC)
- iii) Filter (Removing ripples)
- iv) Voltage Regulation (To set regulated voltage)



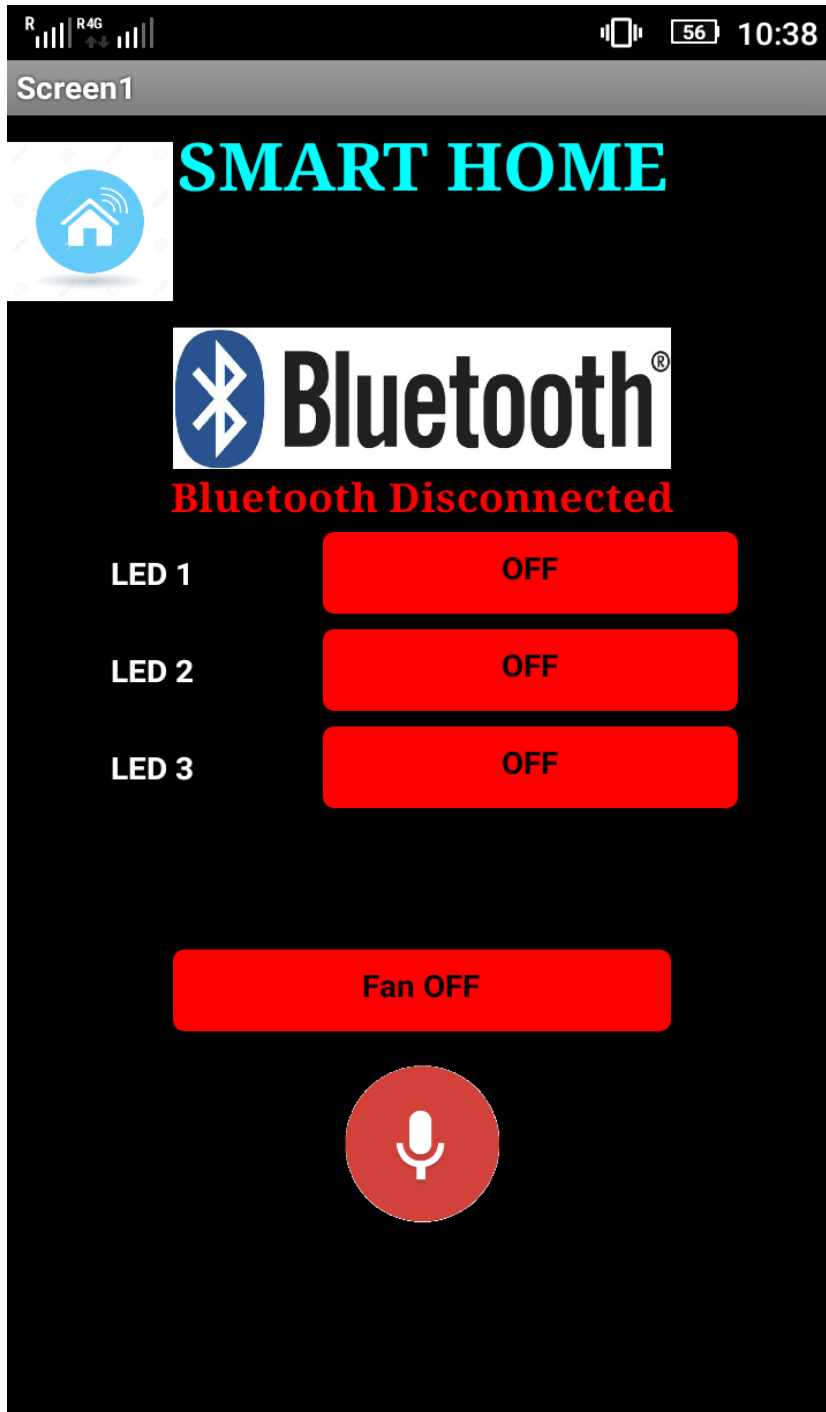
NOT USED IN THIS MODEL BECAUSE 12V RELAY ISN'T USED.

4) Android Application

An android application is created with all the switches and regulators embedded into it to access and control the electrical devices we need to operate. This app indeed works as a remote controlled devices to keep switching on and off the devices. This application directs the controls to Arduino UNO using the Bluetooth as a wireless interface which forwards the instructions to the components that need to be operated through the relay switches.

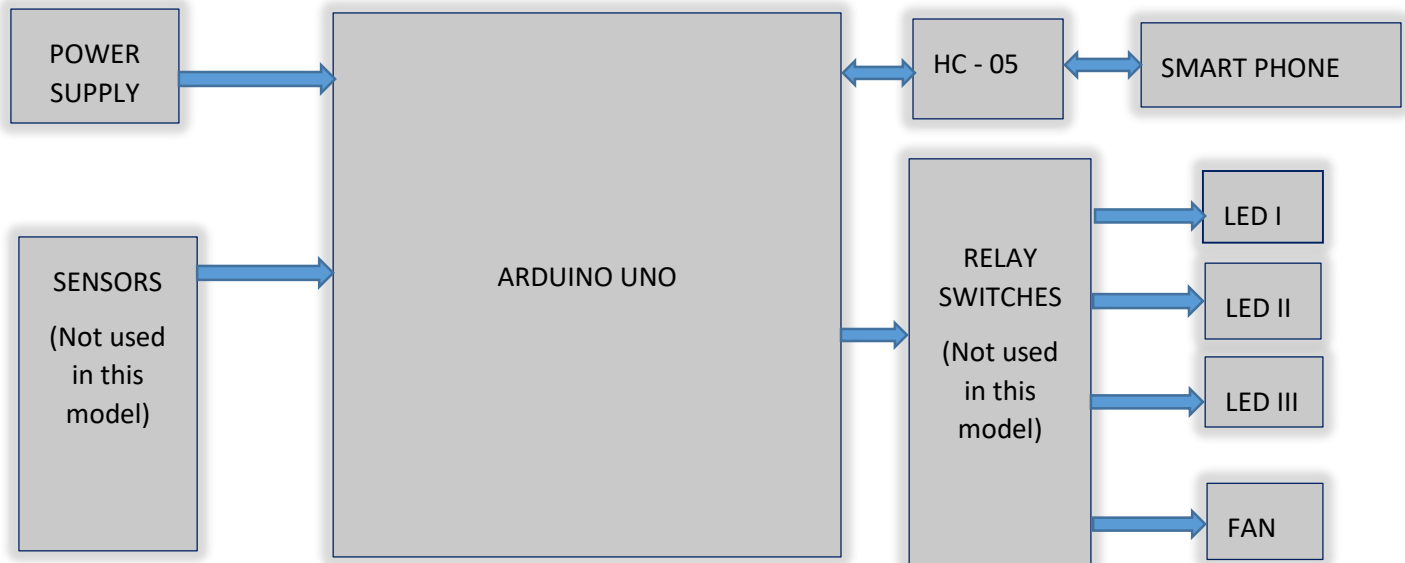
Apart from buttons to toggle the appliances, voice control has also been implemented and verified. Voice is converted to text within the app and that text is used by Arduino to decode and switch accordingly.

Screenshot of the Android Application:

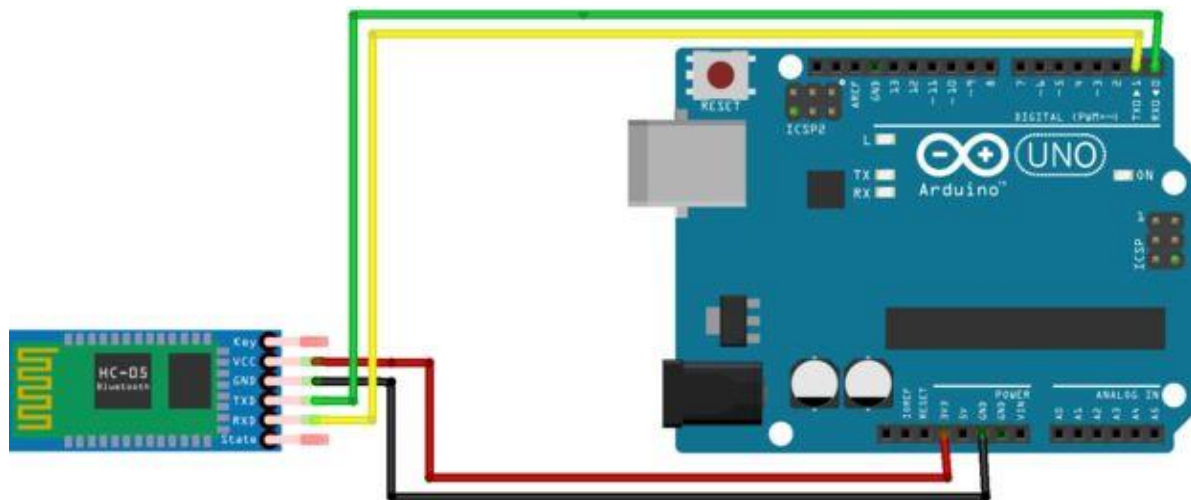


IV. CONNECTIONS

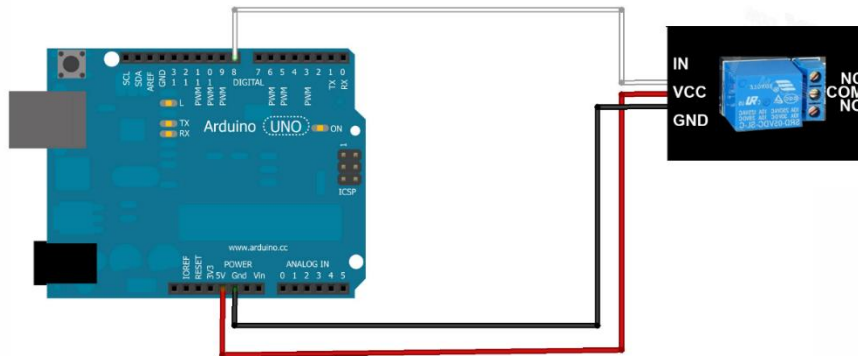
The block diagram of this project has been presented below:



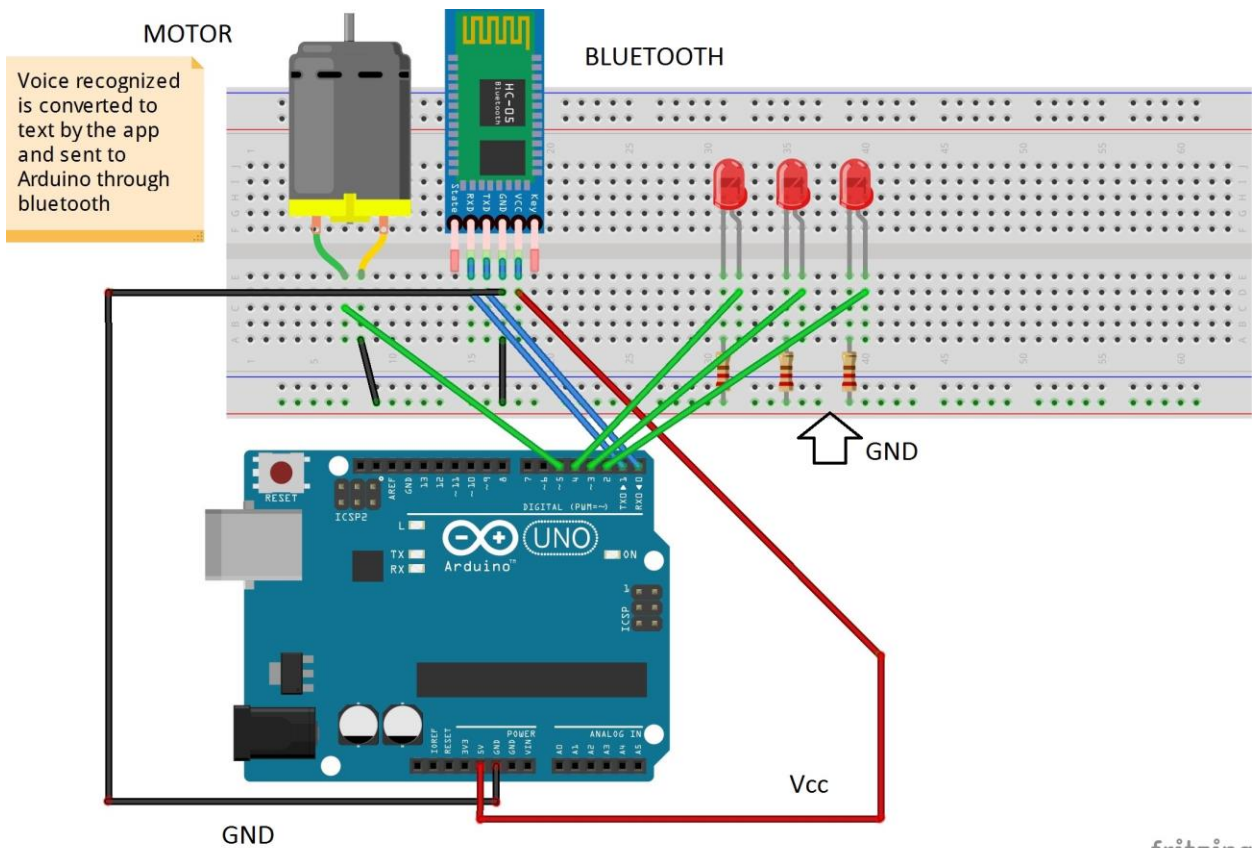
i) Connections between HC-05 and Arduino UNO



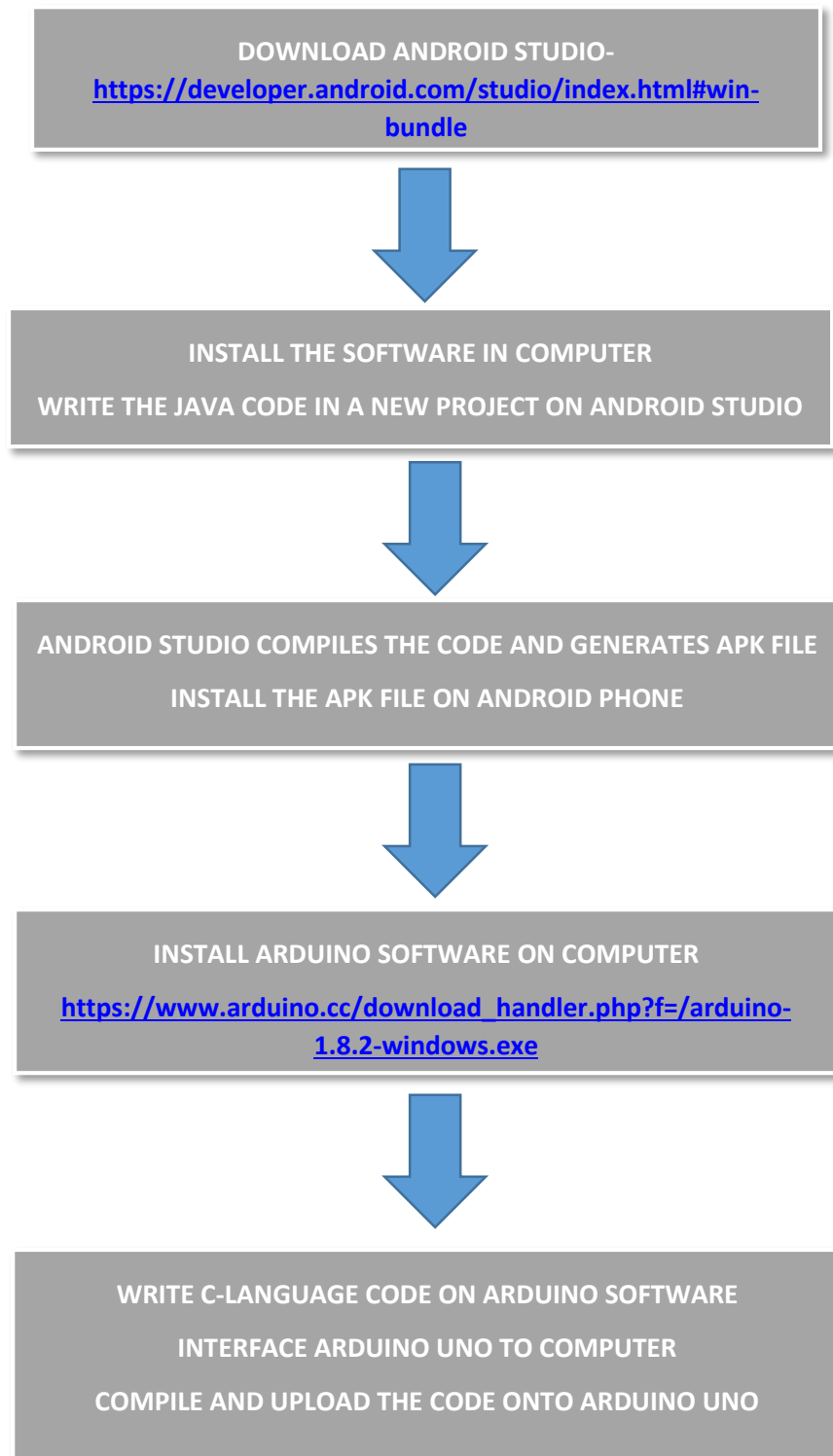
ii) *Connections between Relay module and Arduino UNO*



iii) *Pin connections between Arduino UNO and components*



V. FLOW CHART OF DESIGNED MODEL



VI. CODE FOR ARDUINO UNO

```
String data;

int

led1 = 2, //Connect LED 1 To Pin #2

led2 = 3, //Connect LED 2 To Pin #3

led3 = 4, //Connect LED 3 To Pin #4

fan = 5; //Connect FAN To Pin #5

//-----Functions Declaration-----//

void allon(){

    digitalWrite(led1, HIGH);

    digitalWrite(led2, HIGH);

    digitalWrite(led3, HIGH);

    digitalWrite(fan, HIGH);

}

void alloff(){

    digitalWrite(led1, LOW);

    digitalWrite(led2, LOW);

    digitalWrite(led3, LOW);

    digitalWrite(fan, LOW);

}

//-----//

void setup() {

    Serial.begin(9600);
```

```
pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(fan, OUTPUT);

}

//-----//

void loop() {

    while (Serial.available()){ //Check if there is an available byte to read

        delay(3); //Delay added to make thing stable

        char c = Serial.read(); //Conduct a serial read

        data += c; //Shorthand for data = data + c

    }

    if (data.length() > 0) {

        Serial.println(data);

        //-----//

        //-----Control Multiple Pins In One Go-----//

        if(data == "all on") {allon();} //Turn On All Pins (Call Function)

        else if(data == "all off"||data== "all of"){alloff();} //Turn Off All Pins (Call Function)


        //-----Turn On One-By-One-----//

        else if(data == "one on" || data == "1 on") {digitalWrite(led1, HIGH);}

        else if(data == "two on" || data == "2 on") {digitalWrite(led2, HIGH);}

        else if(data == "three on" || data == "3 on") {digitalWrite(led3, HIGH);}

        else if(data == "fan on") {digitalWrite(fan, HIGH);}

        //-----Turn Off One-By-One-----//

        else if(data == "one off" || data == "1 off" || data == "1 of" || data == "one of") {digitalWrite(led1, LOW);}
```

```

else if(data == "two off" || data == "2 off" || data == "2 of" || data == "two of") {digitalWrite(led2, LOW);}

else if(data == "three off" || data == "3 off" || data == "3 of" || data == "three of") {digitalWrite(led3,
LOW);}

else if(data == "fan off" || data == "fan of") {digitalWrite(fan, LOW);}

//------(Off-Of) && (1-one) are included to encounter improper voice recognition-----//

//-----Toggling using Buttons in App-----//

else if(data == "J") {digitalWrite(led1, HIGH);}

else if(data == "U") {digitalWrite(led1, LOW);}

else if(data == "K") {digitalWrite(led2, HIGH);}

else if(data == "I") {digitalWrite(led2, LOW);}

else if(data == "L") {digitalWrite(led3, HIGH);}

else if(data == "O") {digitalWrite(led3, LOW);}

else if(data == "F") {digitalWrite(fan, HIGH);}

else if(data == "R") {digitalWrite(fan, LOW);}

//-----//

data="";} //Reset the variable after initiating

```

VII. CONCLUSION

Switching 3 LEDs and a fan have been tested through Android application and have been verified working. In case we need to adjust brightness of LEDs and alter the speed of the fan, instead of digital, analog signals to Arduino UNO from App need to be implemented.

Using all the pins of Arduino, we can control appliances in the whole house from the Android app via Bluetooth. Voice control to switch the working of electrical/electronic appliances is merely a technological success. Wireless implementation of this have made lives easier and is found to be a useful operation in every-day world.

LINKS FOR COMPONENTS

Arduino UNO	-----	http://amzn.in/8IRh8cn
Bluetooth(HC-05)	-----	http://amzn.in/gggODEj
4-channel relay	-----	http://amzn.in/eyM4oRW
LDR	-----	http://amzn.in/f0oOdH8
