

# International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)  
Impact Factor: 5.164



**Chief Editor**

**Dr. J.B. Helonde**

**Executive Editor**

**Mr. Somil Mayur Shah**

## ABSTRACT

The main objective of this project is to develop a Home automation system using an Arduino board with Bluetooth being remotely controlled by any android OS smart phone. As technology is advancing so houses are also getting smarter. Modern houses are gradually shifting from conventional switches to centralized control system, Involving remote control switches. Presently, conventional wall switches located in different parts of the houses makes it difficult for the user to go near them to operate. Even more it becomes more difficult for the elderly or physically handicapped people to do so. Remote controlled home automation system provides a most modern solution with smart phone.

In order to achieve this, Bluetooth module is interfaced to Arduino board at the receiver end while on the transmitter end, a GUI application on the cell phone sends ON/OFF, commands receiver where loads are connected. By touching the specified location on the GUI a prompt will appear which will seek a command. Thus switching ON/OFF the connected loads.

**KEYWORDS:** surface foundations, early degradation, shear strength, prediction model.

## 1. INTRODUCTION

Nowadays, we have remote controls for our television sets and other electronic systems, which have made our lives real easy. Have you ever wondered about home automation which would give the facility of controlling tube lights, fans and other electrical appliances at home using a remote control? Off-course, Yes! But, are the available options cost-effective? If the answer is No, we have found a solution to it. We have come up with a new system called Arduino based home automation using Bluetooth. This system is super-cost effective and can give the user, the ability to control any electronic device without even spending for a remote control. This project helps the user to control all the electronic devices using his/her smartphone.

With the help of this system you can control your home appliances from your mobile phone. You can turn on/off your home appliances within the range of Bluetooth.

**Project aim**

The aim of the project is to design and construct a home automation system that will remotely switch on or off any household appliances connected to it, using a Bluetooth based android application using voice control.

**Project objective**

The objective of this project is to implement a low cost, reliable and scalable home automation system that can be used to remotely switch on or off any household appliance, using a microcontroller to achieve hardware simplicity, low cost short messaging service for feedback and voice dial from any phone to toggle the switch state.

**Project scope and limitation**

This project work is complete on its own in remotely and automatically switching on or off of an electrical appliance not limited to household appliances and sends a feedback message indicating the new present state of the appliance.

**Description of the project**

This project is one of the important Arduino Projects. Arduino based home automation using Bluetooth project helps the user to control any electronic device using Device Control app on their Android Smartphone. The android app sends commands to the controller – Arduino, through wireless communication, namely, Bluetooth. The Arduino is connected to the main PCB which has five relays as shown in the block diagram. These relays can be connected to different electronic devices. As per the block diagram, Device 1 – Buzzer, Device 2- Fan, Device 3 – Lights , Device 4 – temperature sensor.

When the user speaks the respective command to the respective device, the device can turn 'on' and 'off'.

This project of home automation using Bluetooth and Arduino can be used for controlling any AC or DC devices. In the demonstration, we have used DC Fan and 5mm led lights. To drive this DC Fan and Light, a 9V battery is connected.

#### HARDWARE REQUIRMENTS

#### SOFTWARE REQUIRMENTS

Sr No.	Name	Specification
1	Arduino UNO	1) Digital I/O Pins: 14 (of which 6 provide PWM output) 2) Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader. 3) SRAM: 2 KB(ATmega328P), Operating Voltage 5V
2	5 mm visible LED's - 1 Red,1 Green,1 White	1) LED voltage = 2V 2) Maximum current = 20mA
3	Jumper Cables	Each cable length about 20cm
4	Temperature Sensor- LM35	1) Minimum and Maximum Input Voltage is 35V and -2V respectively. 2) Temperature range from -55°C to 150°C. 3) Rise of 10mV (0.01V) for every 1°C rise in temperature.
5	Buzzer	1) Rated Voltage: 6V DC 2) Operating Voltage: 4-8V DC. 3) Rated current: <30mA. 4) Sound Type: Continuous Beep 5) Resonant Frequency: 2300 Hz.
6	Battery	1) Voltage: 9 volts 2) Capacity: 500 (mAh)
7	Bluetooth Module- HC05 FC 114	1) Power Supply: +3.3VDC 50mA 2) Frequency: 2.4GHz ISM band 3) Bluetooth Protocol: Bluetooth specification v2.0+EDR
8	IR sensor	Supply input: 3.3 to 5 V [DC]
9	Electric Motor	Rated Voltage 5V DC
10	Resistors	1) Resistor type 1: 220 Ohms 5%, 5 pieces 2) Resistor type 2: 100 Ohms 5%, 1 piece
11	9V Battery Connector	Wire length 10 cm
12	Battery Snap Power Cable	Size: 2.1 x 5.5mm
13	Bread board	Size: 66x 174 x 8.5mm

- Arduino IDE 1.8.12
- Android application
- MIT app inventor 2

**DESCRIPTION OF HARDWARE REQUIRED**

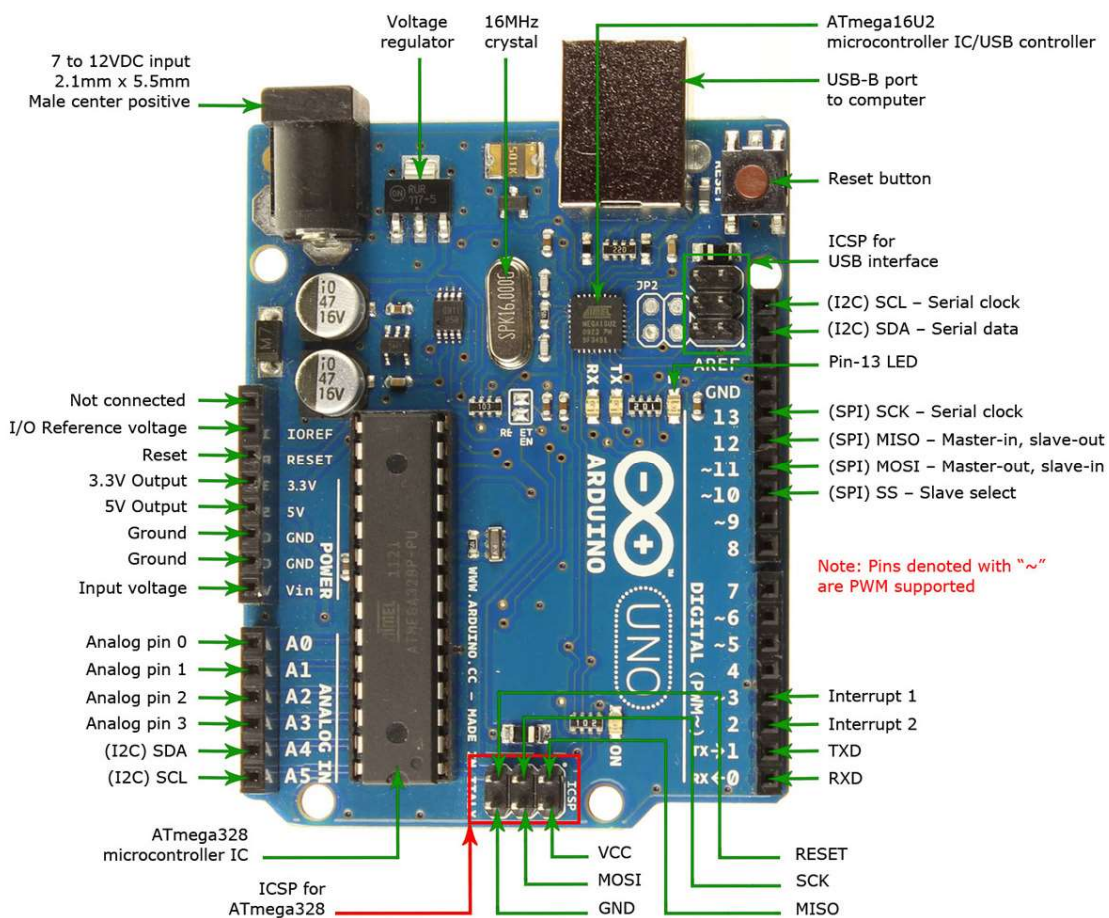
**Arduino UNO –**

The Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

❖ Some technical specification of Arduino UNO :-

1.	Microcontroller	ATmega328P
2.	Operating voltage	5V
3.	Digital I/O pins	14
4.	Flash memory	32KB of which 0.5KB used by boot loader
5.	SRAM	2KB
6.	Clock speed	16Mhz

**Circuit diagram**

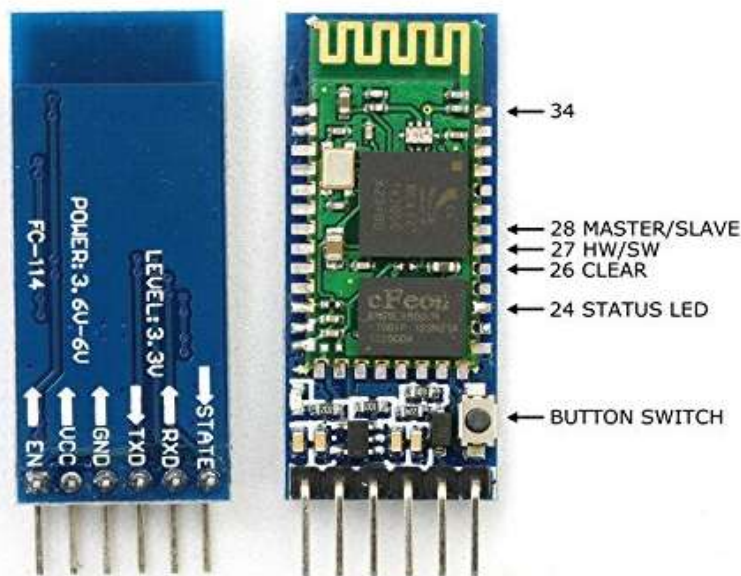




### HC-05 Bluetooth module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development.

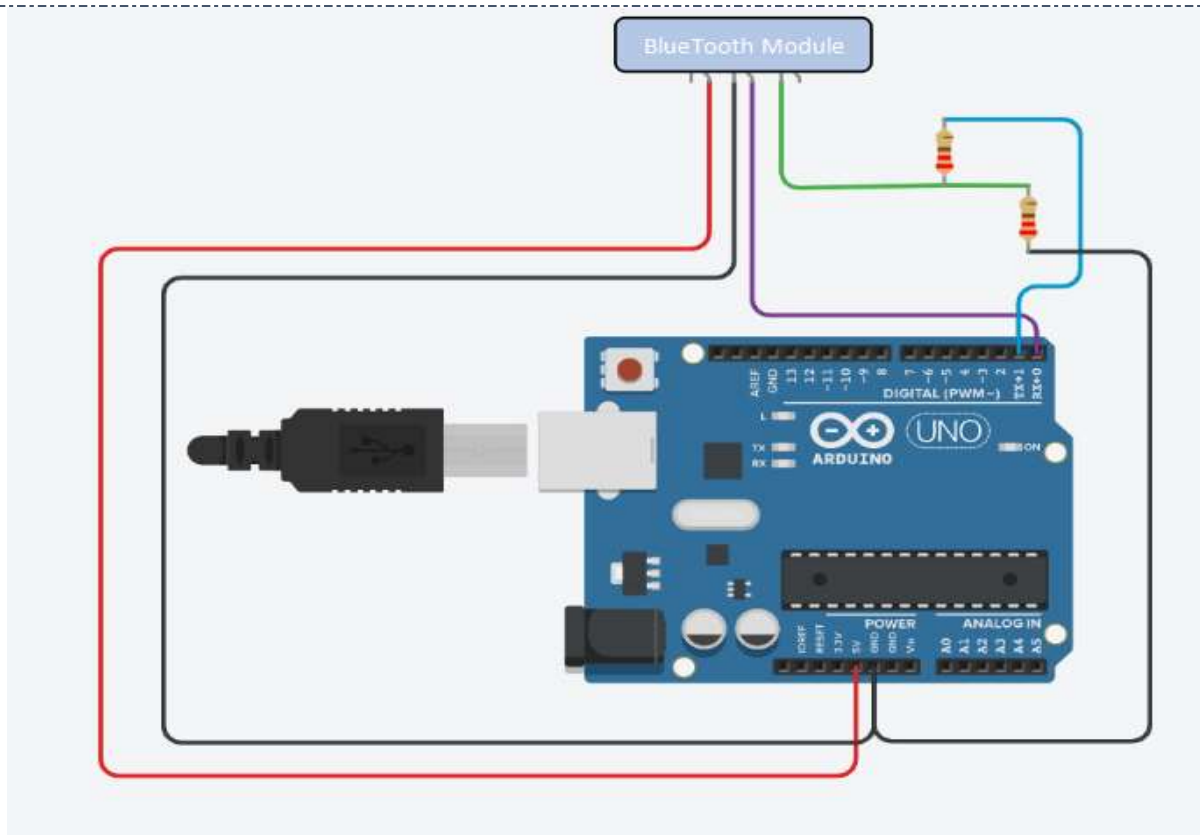


### HC-05 Bluetooth module interfacing with Arduino UNO

HC-05 is a Bluetooth device used for wireless communication with Bluetooth enabled devices (like smartphone). It communicates with microcontrollers using serial communication (USART).

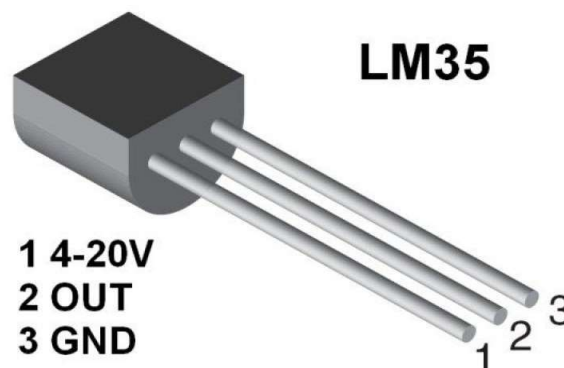
Default settings of HC-05 Bluetooth module can be changed using certain AT commands.

As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, there is no need to shift TX voltage level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

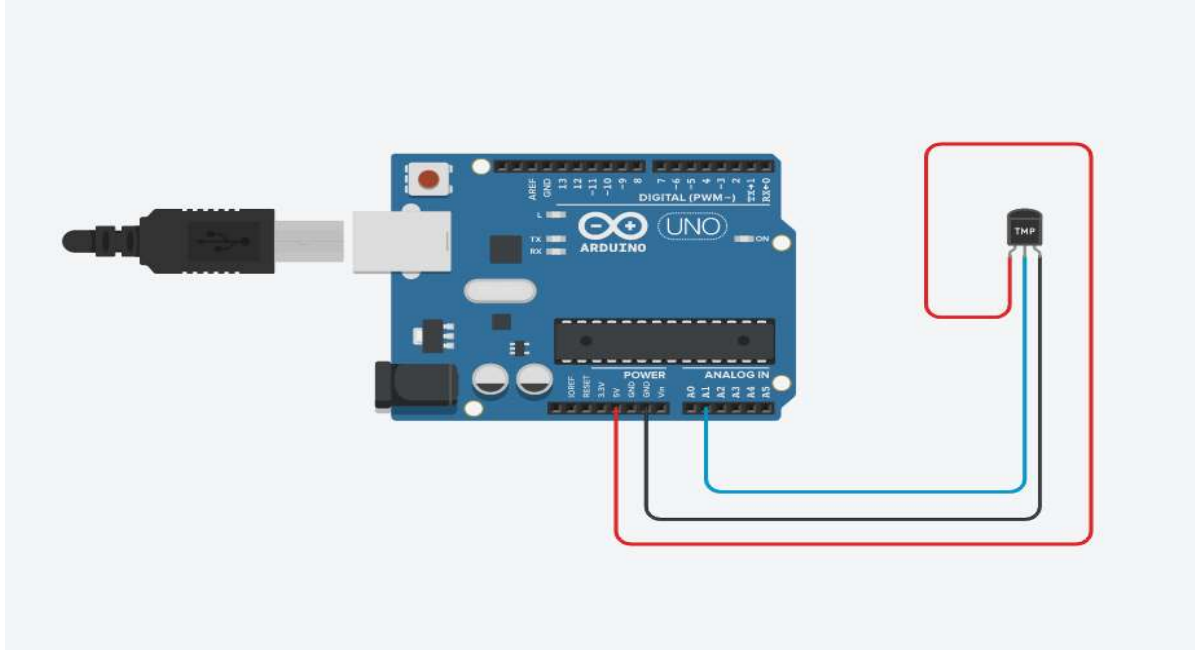


### Temperature sensor IM35

LM35 is a temperature measuring device having an analog output voltage proportional to the temperature. It provides output voltage in Centigrade (Celsius). ... The sensitivity of LM35 is 10 mV/degree Celsius. As temperature increases, output voltage also increases. Main purpose of temperature sensor is to show the current room temperature.

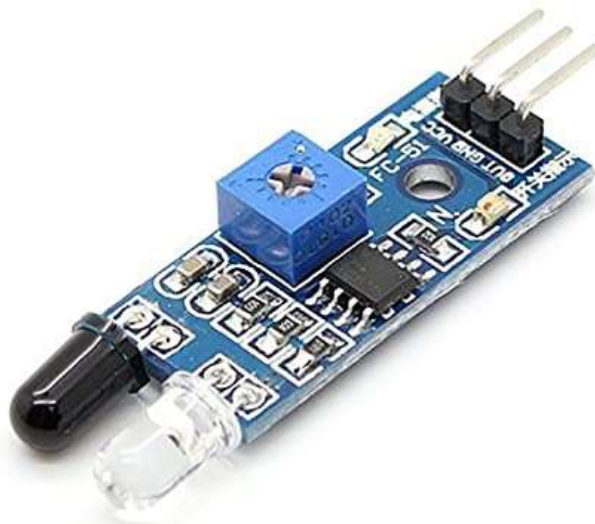


Temperature sensor im35 interfacing with Arduino UNO

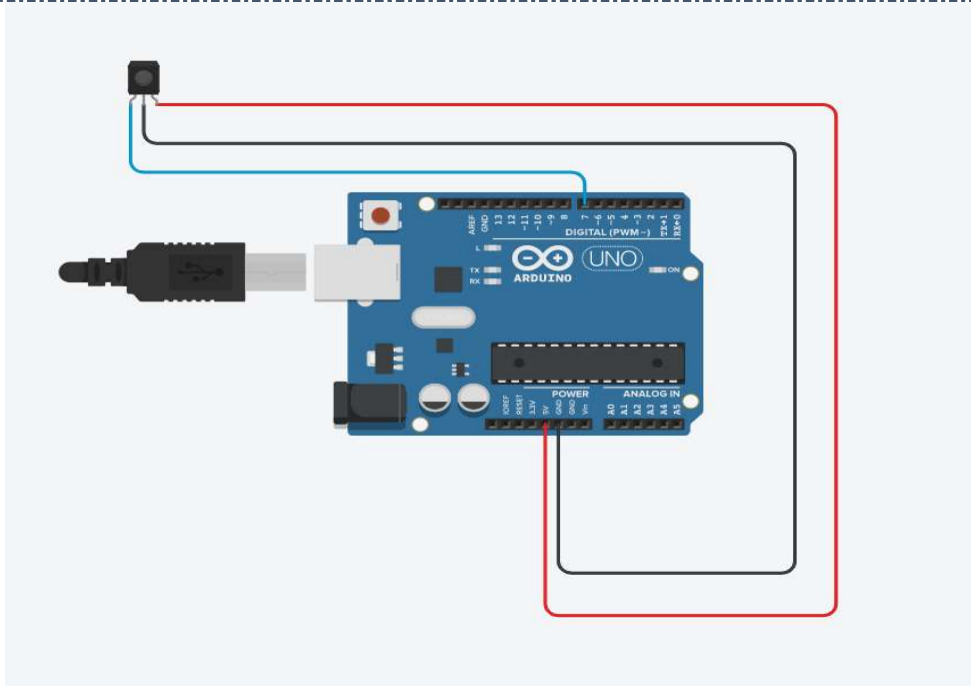


### IR sensor

The IR sensor module consists mainly of the IR Transmitter and Receiver, Variable Resistor (Trimmer pot), output LED in brief. IR LED Transmitter. IR LED emits light, in the range of Infrared frequency. IR light is invisible to us as its wavelength (700nm – 1mm) is much higher than the visible light range. The main purpose of IR sensor is to detect any intruder entering the house.



### IR sensor interfacing with Arduino UNO



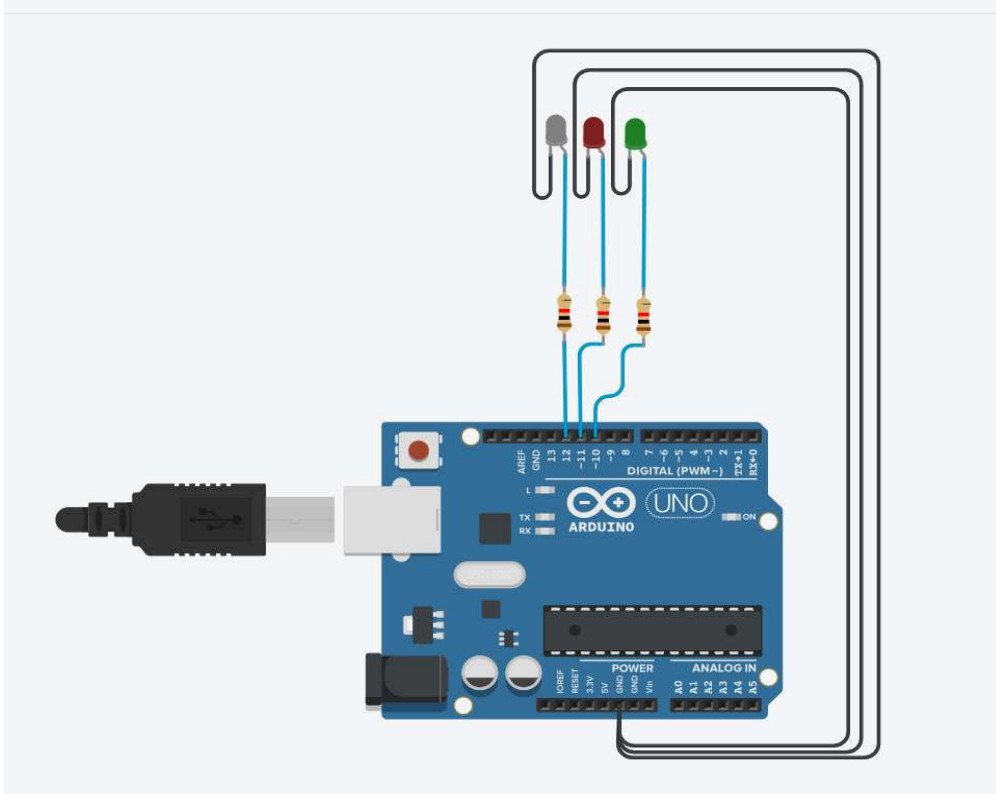
### 5mm LED lights

These are small LED lights which contains 2 pin of anode (+) and cathode (-).the anode pin is longer than the cathode pin. Three different colored led lights are used. Red, green and white.



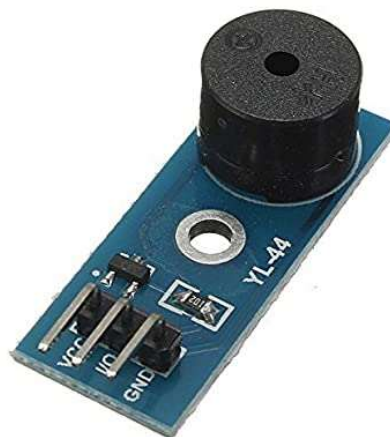
### LED lights interfacing with Arduino UNO



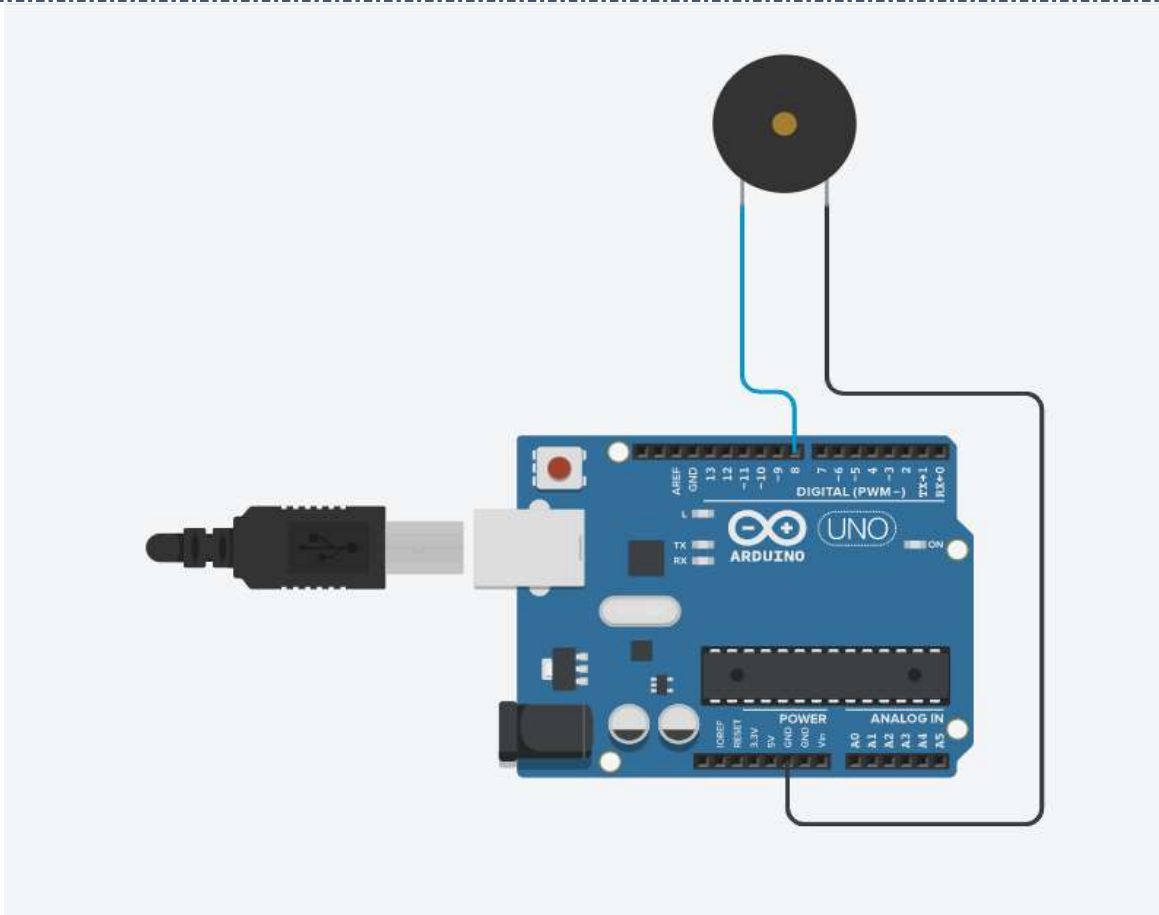


### Buzzer

It's simple, `tone(buzzer, 1000)` sends a 1 KHz sound signal to pin 9, `delay(1000)` pause the program for one second and `noTone(buzzer)` stops the signal sound. The loop routine will make this run again and again making a short beeping sound.



### Buzzer interfacing with Arduino UNO



### DC Motor

By connecting an L298 bridge IC to an Arduino, you can control a DC motor. By connecting an L298 bridge IC to an Arduino, you can control a DC motor. A direct current, or DC, motor is the most common type of motor. DC motors normally have just two leads, one positive and one negative.

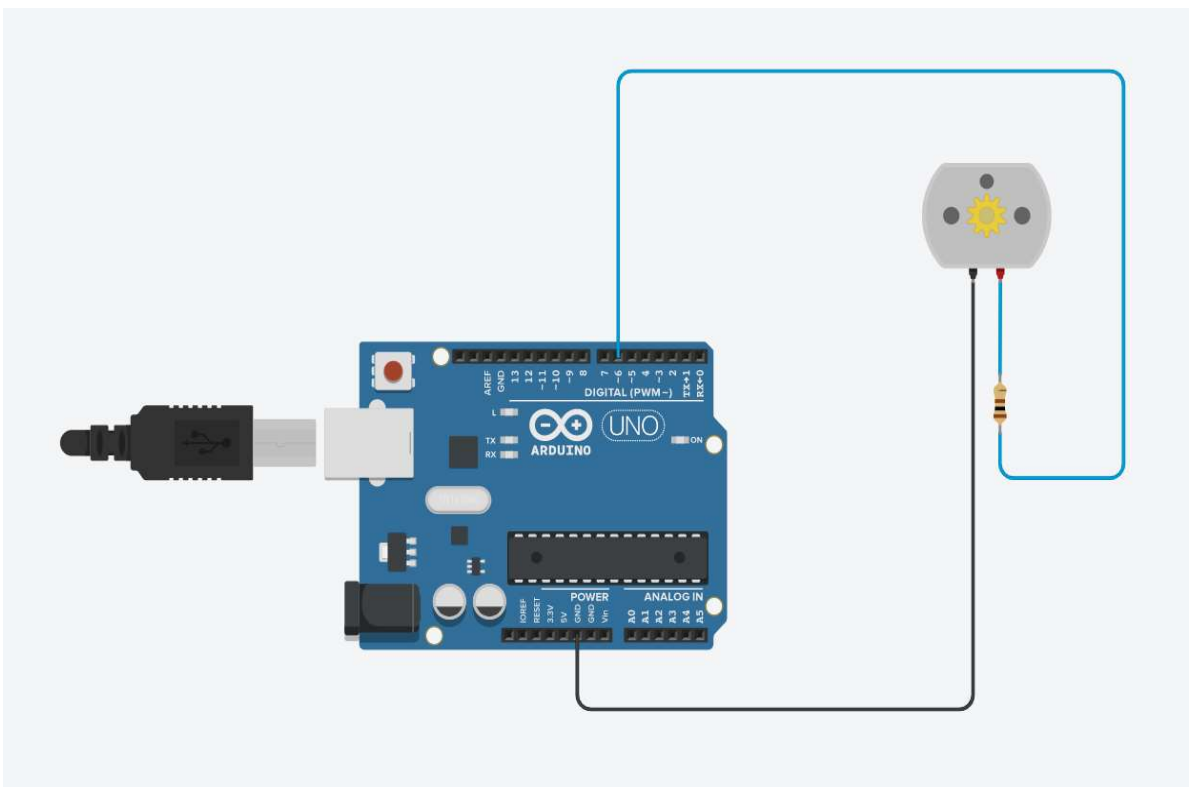


### DC Motor interfacing with Arduino UNO

htytp: // [www.ijesrt.com](http://www.ijesrt.com) © International Journal of Engineering Sciences & Research Technology

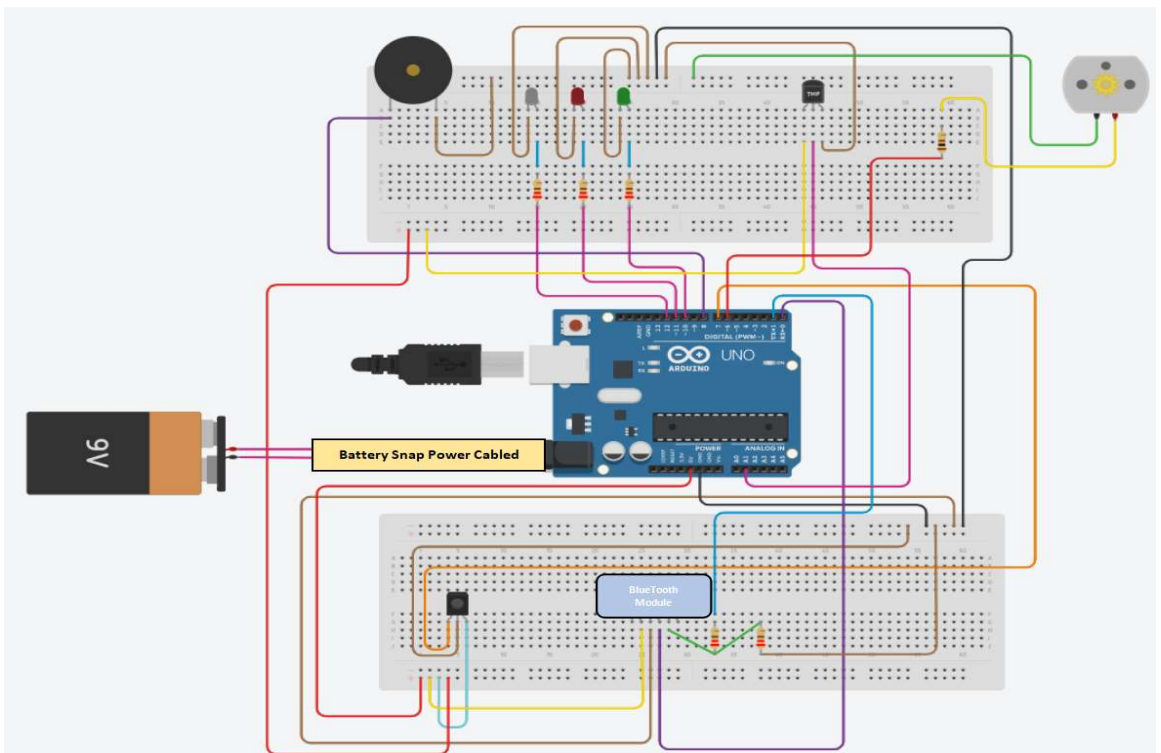
[24]

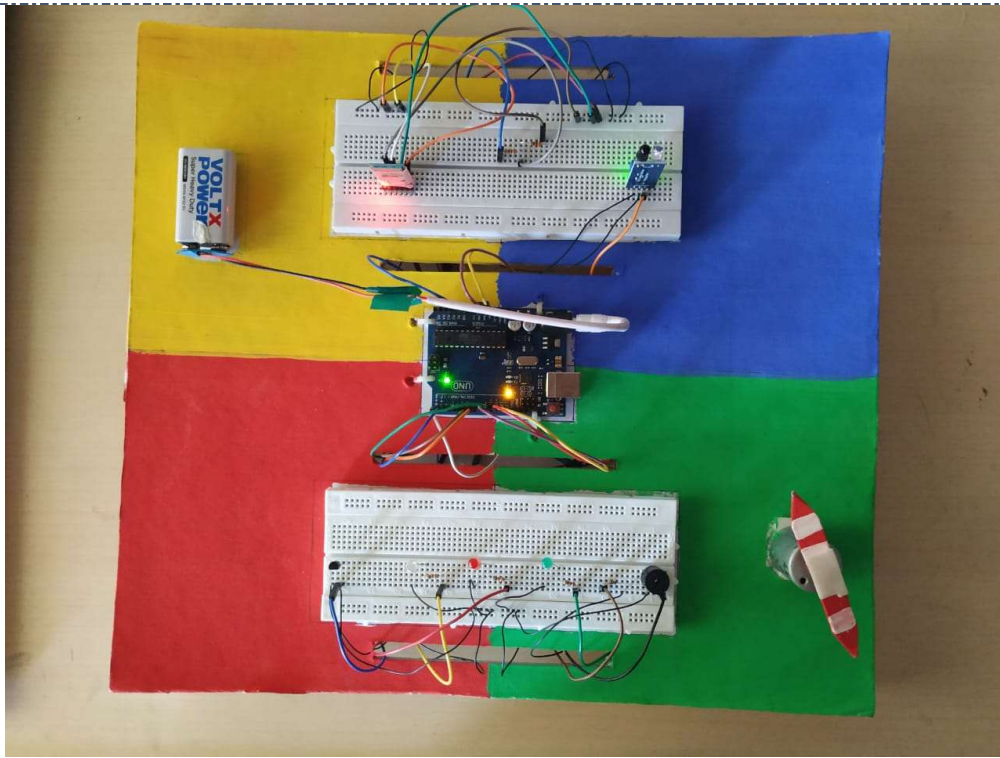




## 2. DESIGN AND CONTRUCTION

Overall circuit diagram of project

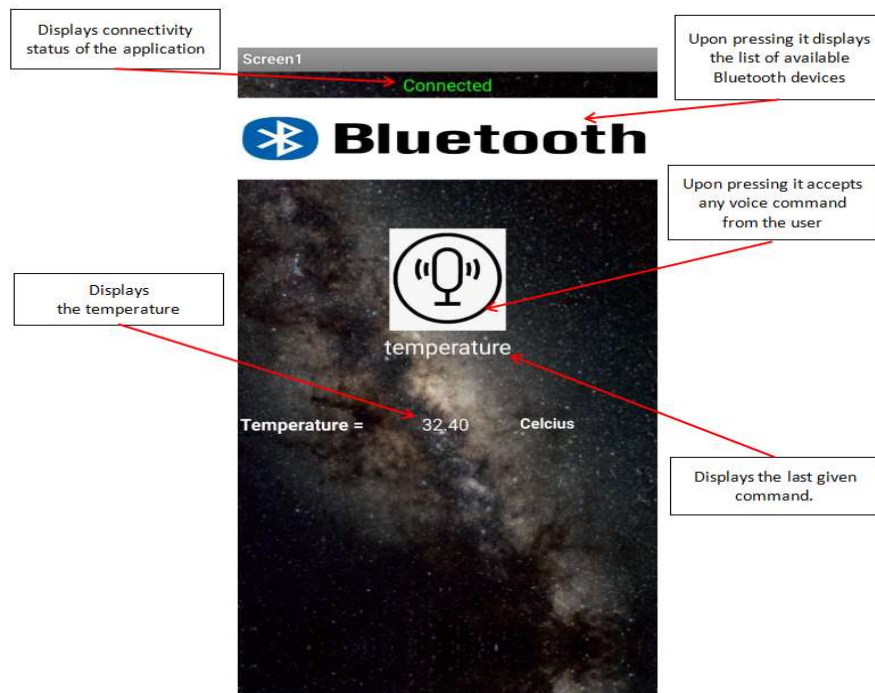




### 3. WORKING

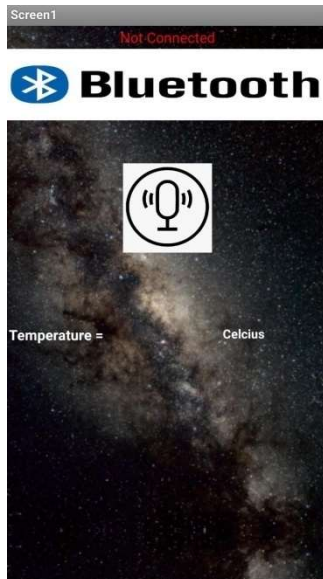
#### Layout of the application

The following image shows the format of the custom built application.

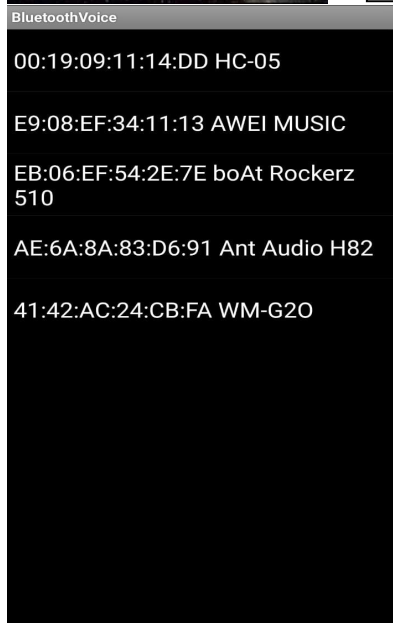


---

Connection to Bluetooth module



On pressing the white Bluetooth bar the list of Bluetooth devices will open.

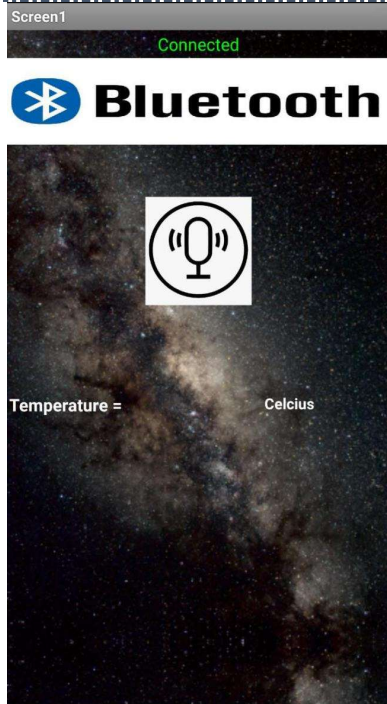


As shown in the fig. the list of the Bluetooth devices, click on the name of the Bluetooth module. As per this fig it is 1st on with HC-05 in end.

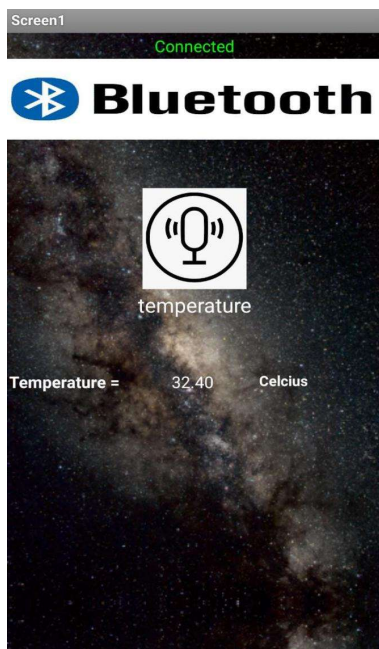
### How to obtain the temperature from temperature sensor

The temperature sensor has three terminals. 1st terminal is connected to the VCC, middle one is connected to A1 pin of Arduino and last one is connected to ground.



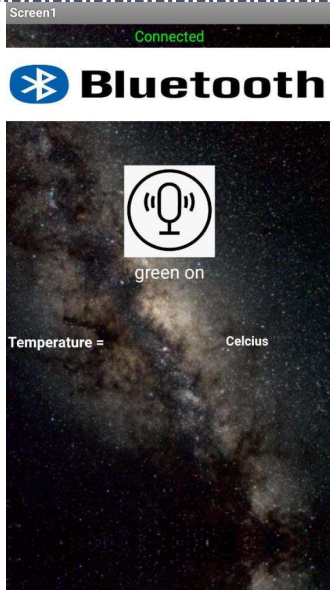


- To know the temperature, first click on the voice command button on screen as shown in fig.
- By click on the voice command button, it will allow to give voice command.
- To know the temperature say “ temperature”



In return JARVIS will reply or will send the current reading of temperature to the mobile through Bluetooth and will display the temperature on the screen of the mobile.

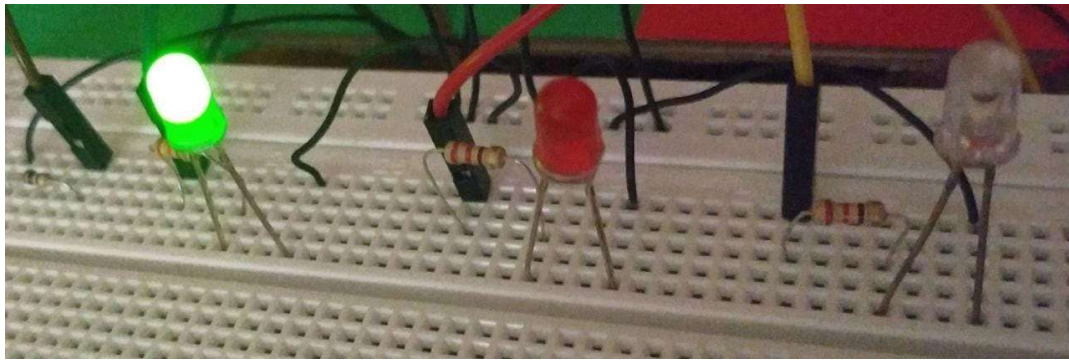
How to turn ON/OFF different lights

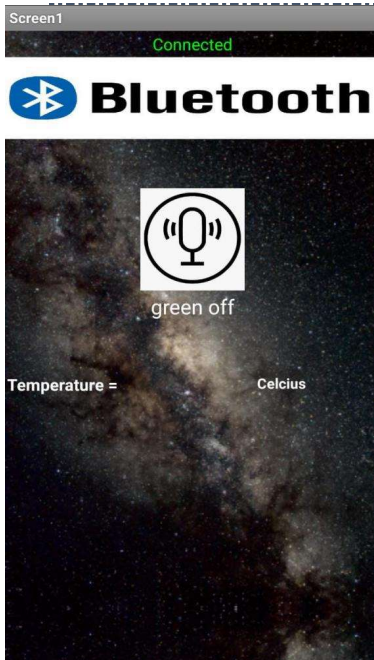


For turning on the light, the voice command button is to be pressed.

After pressing the command button, the command "GREEN ON "is to be said.

In return JARVIS replies "I turned on the green light" and will turn on the green light.

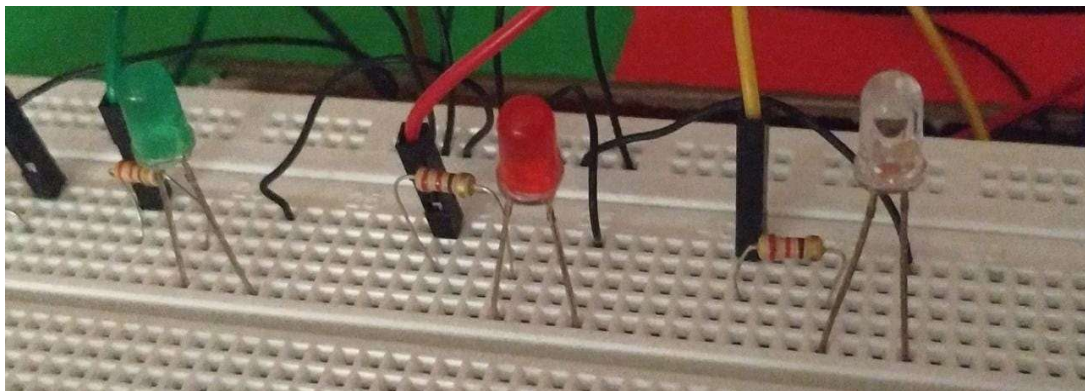




For turning off the light, the voice command button is to be pressed.

After pressing the command button, the command "GREEN OFF "is to be said.

In return JARVIS replies "I turned off the green light" and will turn off the green light.



The connections for the green light are given as follows.

The right(+ve) terminal is connected to resistor of 220 ohms and another leg to resistors is connected to the pin no 10 of digital side .The left (-ve) terminal of the green light is connected to ground .the same connection are given to white and red light also ,just the digital pins are changed. Pin no 12 and 11 are for white and red light respectively.

Similarly the commands for turning on and off the red and white lights are

For red lights:-

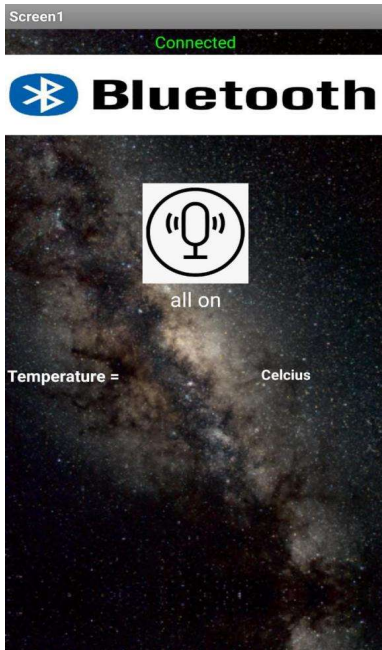
Turning on – "RED ON" and Turning off - "RED OFF"

For white light:-

Turning on - "WHITE ON" and Turning off - "WHITE OFF"

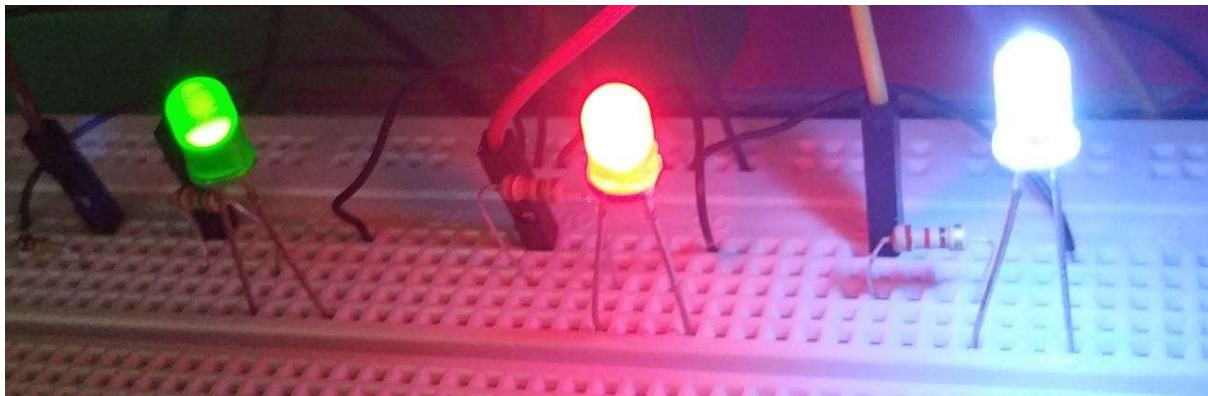


For tuning all lights ON together followed:

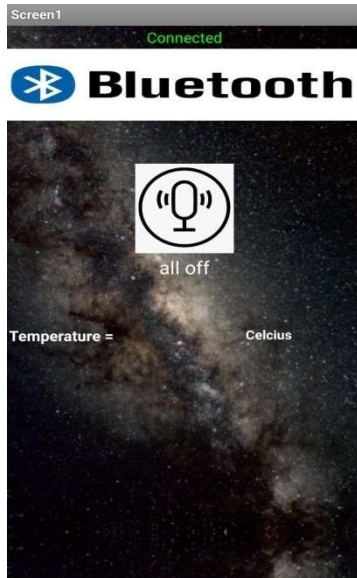


For turning all light on together, press the voice command button. Then it will allow the user to give voice command.

To turn all lights ON say “all on”. It will turn all lights ON



### For tuning all lights OFF:

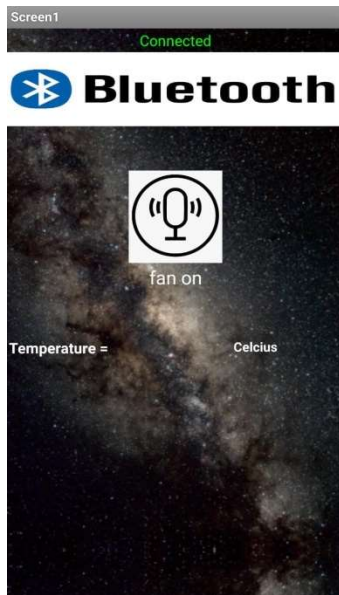


To turn OFF all lights, press the voice command button again and give the voice command “all OFF”. Jarvis will turn OFF all lights.

### How to control the fan / motor

The connection of fan is given as follows:

The right(+ve) terminal is connected to resistor of 100 ohms just to control the current. Another leg of resistor is connected to the pin no 6 of digital side. The left (-ve) terminal is grounded.

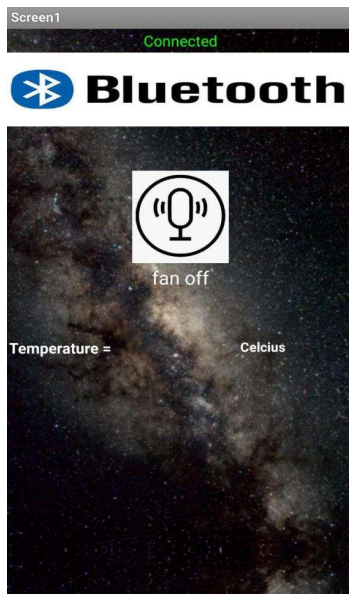


To start the fan, the voice command button on the mobile screen is to be pressed.

After pressing the button .the command "FAN ON "is to be said.

In return JARVIS will reply “I turned on the fan ” and JARVIS will turn on the fan by providing power to the motor.





Similarly for turning off the fan, the same procedure is to be followed:

To stop the fan, the voice command button on the mobile screen is to be pressed.

After pressing the button .The command "FAN OFF" is to be said.

In return JARVIS will reply "I turned off the fan" and JARVIS will turn off the fan.



**How to lock and unlock the house ( IR sensor)**

The left terminal of the Buzzer is connected to the ground. And right terminal is connected to pin no. 8 of digital side.

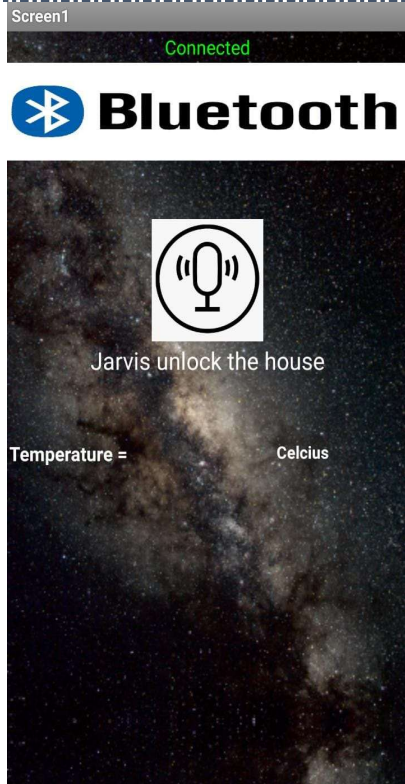


The working of buzzer is shown in following steps:

To lock the house for security purpose, if user wants to lock it press the voice command button as shown in the display, this will allow the user to give the voice command.

Then user says “Jarvis Lock the house”, due to this IR sensor is activated.

Thus if an intruder comes within the range of IR sensor the buzzer gets activated



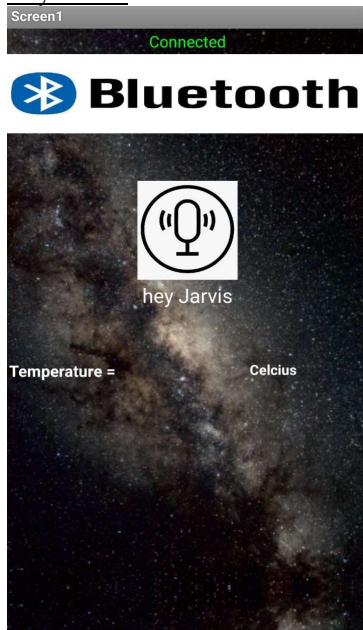
To deactivate the buzzer, then have to follow the same steps.

Press the voice command button and give the command “Unlock the house”. Due to this the IR sensor is deactivated and buzzer will not get activated.

**Miscellaneous (other voice commands)**

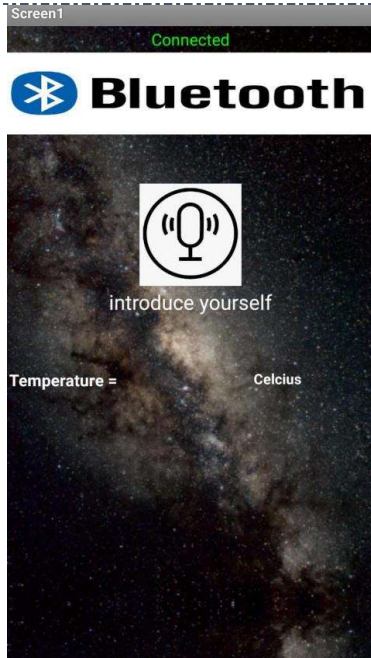
Apart from the above written commands, we have added three more commands for the purpose of increasing the interactivity of JARVIS and the user, this are:

Hey Jarvis:



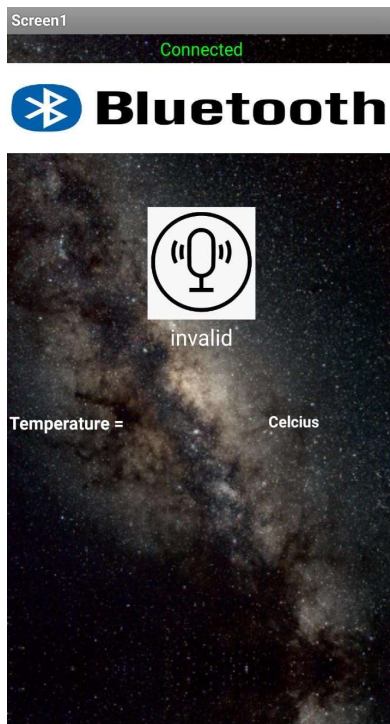
By pressing the voice command button on the screen; “Hey Jarvis” is to be said by the user. In return, JARVIS will reply by saying, “Yes sir, how can I help you?”

**Introduce yourself:**



By pressing the voice command button on the screen, “Introduce Yourself” is to be said by the user. In return, JARVIS will introduce itself by saying, “Hey everyone, I am Just a Rather Very Intelligent System, also known as JARVIS.”

Any invalid command:



By pressing the voice command button on the screen if the user says any command which has not been enumerated, JARVIS will reply by saying “Invalid command, please try again” and not perform any action.



#### 4. PROGRAM

##### Arduino program:

```
Bluetooth:Final
String device;
int buzzer = 8;
int irPin = 7;
int sensorOut = HIGH;
int hou_lock = 0;
float temp = 1;
void setup() {
  BT.begin(9600);
  Serial.begin(9600);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(irPin, INPUT);
}
int temp_adc_val;
float temp_val;

void loop() {
  while (BT.available())
  { //Check if there is an available byte to read
    delay(10); //Delay added to make thing stable
    char c = BT.read(); //Conduct a serial read
    device += c; //build th0e string.
  }
  sensorOut = digitalRead(irPin);
  if (sensorOut == LOW && hou_lock == 1 )
  {
    BT.print(temp);

    digitalWrite(buzzer, HIGH);
  }
  if (hou_lock == 0)
  {
    digitalWrite(buzzer, LOW);
  }

  if (device.length() > 0)
  {

    if(device == "white on" || device == "White on")
    {
      digitalWrite(12, HIGH);
    }

    else if(device == "white off" || device == "White off")
    {
      digitalWrite(12, LOW);
    }

    if(device == "read on" || device == "Read on")
    {
      digitalWrite(11, HIGH);
    }

    else if(device == "read off" || device == "Read off")
    {
      digitalWrite(11, LOW);
    }
  }
}
```






```
if(device == "green on" || device == "Green on")
{
digitalWrite(10, HIGH);
}

else if(device == "green off" || device == "Green off")
{
digitalWrite(10, LOW);
}
if(device == "temperature" || device == "Temperature")
{
temp_adc_val = analogRead(lm35_pin); /* Read Temperature */
temp_val = (temp_adc_val * 4.88); /* Convert adc value to equivalent voltage */
temp_val = (temp_val / 10); /* LM35 gives output of 10mv/°C */
BT.print(temp_val);
Serial.println(temp_val);
}
if(device == "all off")
{
digitalWrite(11, LOW);
if(device == "all on")
{
digitalWrite(11, HIGH);
digitalWrite(10, HIGH);
digitalWrite(12, HIGH);
}
}
if(device == "fan on" || device == "Fan on")
{
digitalWrite(6, HIGH);
}
if(device == "fan off" || device == "Fan off")
{
digitalWrite(6, LOW);
}
if(device == "Jarvis lock the house" || device == "lock the house")
{
hou_lock = 1;
}
if(device == "Jarvis unlock the house" || device == "unlock the house")
{
hou_lock = 0;
}

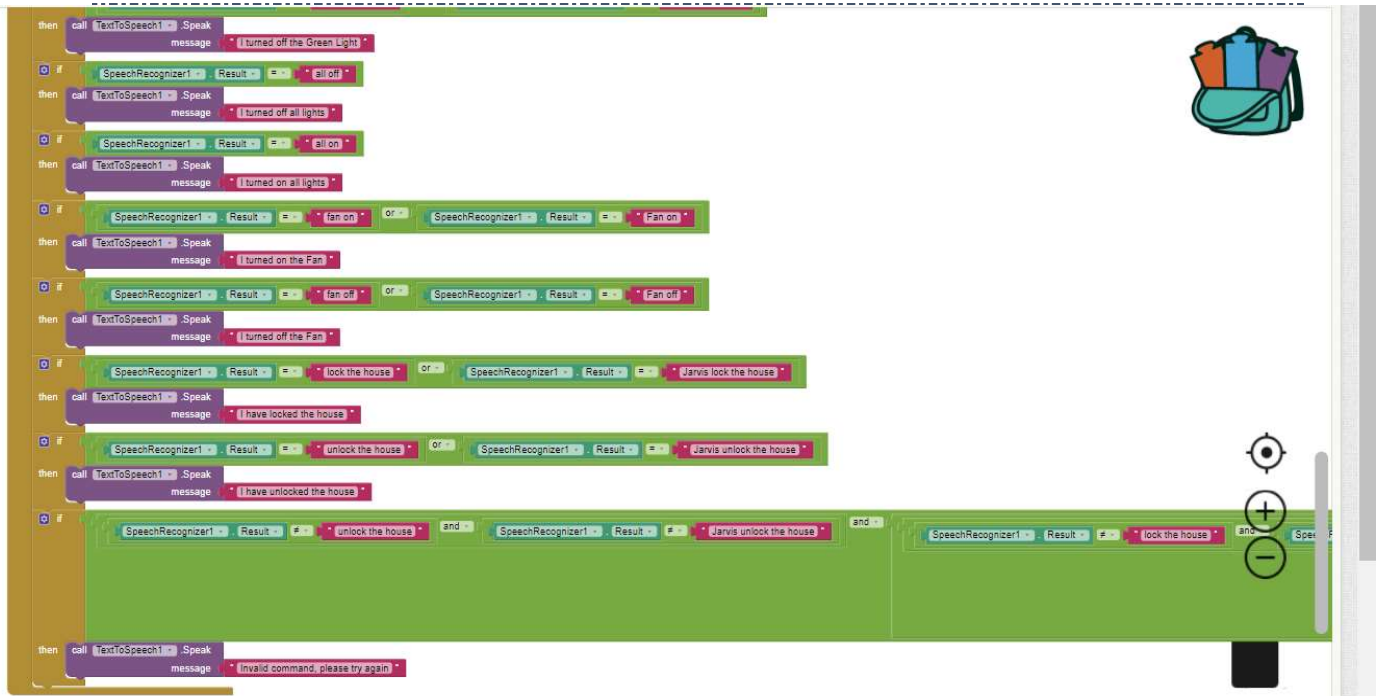
device="";
}
}
```



[MIT app inventor program:](#)



The image displays two panels of MIT App Inventor code blocks. The top panel includes logic for:   
- Initializing a list picker with Bluetooth client addresses.   
- Handling 'BeforePicking' and 'AfterPicking' events.   
- A timer that checks if the Bluetooth client is connected, updating a label's text and color.   
- A button click event that triggers a speech recognizer.   
- Logic for 'BeforeGettingText' and 'AfterGettingText' that sets a label's text and sends it to a Bluetooth client.   
- Conditional logic for voice commands: 'hello Jarvis', 'introduce yourself', 'I turned on the white Light', 'white off', 'White off', 'turned off the white Light', 'read on', 'Read on', 'turned on the red Light', 'read off', 'Read off', 'turned on the green Light', 'green on', 'Green on', 'turned off the Green Light', 'green off', 'Green off', 'temperature', and 'Temperature'.   
- A section for receiving data from the Bluetooth client, including receiving text and bytes, and speaking the received text.   
- A final condition for 'lock the door?' and 'Lock the door?'.   
The bottom panel continues the logic for receiving data and speaking messages.   
On the right side of each code panel, there are standard MIT App Inventor interface icons: a bag icon, a target icon, zoom in (+) and zoom out (-) icons, and a trash can icon.



## 5. APPLICATIONS OF HOME AUTOMATION

- Using this project, we can turn on or off appliances remotely i.e. using a phone or tablet.
- The project can be further expanded to a smart home automation system by including some sensors like light sensors, temperature sensors, safety sensors etc. and automatically adjust different parameters like room lighting, air conditioning (room temperature), door locks etc. and transmit the information to our phone.
- Additionally, we can connect to internet and control the home from remote location over internet and also monitor the safety.

## 6. FUTURE DEVELOPMENT OF THIS PROJECT

Arduino based device control using Bluetooth on Smartphone project can be enhanced to control the speed of the fan or volume of the buzzer etc. Home automation and Device controlling can be done using Internet of Things – IOT technology.

## 7. CONCLUSION

The home automation system has been experimentally proven to work satisfactorily by connecting sample appliances to it and the appliances were successfully controlled from a wireless mobile device. We learned many skills such as soldering, wiring the circuit and other tools that we use for this project and was able to work together as a team during this project. The Bluetooth client was successfully tested on a multitude of different mobile phones from different manufacturers, thus proving its portability and wide compatibility. Thus a low-cost home automation system was successfully designed, implemented and tested.

## REFERENCES

- [1] [www.components101.com](http://www.components101.com)
- [2] LinkedIn | [www.slideshare.net](http://www.slideshare.net).

