

# **CONSTRUCTION OF VOICE CONTROL HOME AUTOMATION**

**A Project and Thesis submitted in partial fulfillment of the  
Requirements for the Award of Degree of  
Bachelor of Science in Electrical and Electronic Engineering**

**Submitted By  
Azijul Hakim  
ID: 133-33-1611**

**Supervised By  
Mr. Abu Shahir Md. Khalid Hasan  
Lecturer, Department of EEE  
Faculty of Engineering**



**DEPARTMENT OF ELECTRICAL AND ELECTRONIC  
ENGINEERING  
FACULTY OF ENGINEERING**

**DAFFODIL INTERNATIONAL UNIVERSITY**

**December 2018**

***TO***  
***MY BELOVED PARENTS***  
***&***  
***HONOURABLE SUPERVISER***  
***MR. ABU SHAHIR MD. KHALID HASAN***

# CERTIFICATION

This is to certify that this project and thesis entitled “Construction of Voice Control Home Automation System” is done by the following student under my direct supervision and this work has been carried out by them in the Department of Electrical and Electronic Engineering under the Faculty of Engineering of Daffodil International University in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering. The presentation of the work was held on December 2018.

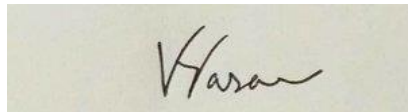
**Signature of the candidate**



---

**Azijul Hakim**  
**ID 133-33-1611**  
Department of EEE  
Daffodil International University

**Signature of the supervisor**



---

**Mr. Abu Shahir Md. Khalid Hasan**  
Lecturer, Faculty of Engineering  
Department of EEE  
Daffodil International University

The project and thesis entitled “Construction of Voice Control Home Automation System,” submitted by Azijul Hakim, ID No:133-33-1611, Session Fall 2013 has been accepted as satisfactory in partial fulfillment of the requirements for the degree of **Bachelor of Science in Electrical and Electronic Engineering** on December 2018.

## **BOARD OF EXAMINERS**

---

**Dr.Engr. ---**

Professor

Department of EEE, DIU

Chairman

---

**Dr.Engr. ---**

Professor

Department of EEE, DIU

Internal Member

---

**Dr.Engr. ---**

Professor

Department of EEE, DIU

Internal Member

# CONTENTS

<b>LIST OF FIGURES</b>	<b>viii</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>ACKNOWLEDGMENT</b>	<b>x</b>
<b>ABSTRACT</b>	<b>xi</b>
<b>Chapter 1 INTRODUCTION</b>	<b>01-02</b>
1.1 Introduction	01
1.2 Aim Of The Project	01
1.3 Scopes	02
1.4 Methodology	02
1.5 Organization Of The Report	02
<b>Chapter 2 SYSTEM REVIEWS</b>	<b>03-05</b>
2.1 Introduction	03
2.2 General Block Diagram	03
2.2.1 Block Diagram Description	04
2.3 Circuit Diagram	04
2.3.1 Working Process Of Our Circuit	05
2.4 List Of Component Used In Circuit	05
2.5 Conclusion	05
<b>Chapter 3 COMPONENT DESCRIPTION</b>	<b>06-30</b>
3.1 Introduction	06
3.2 Description Of The Bluetooth Module(HC-05)	06
3.2.1 Features Of The Bluetooth Module(HC-05)	08
3.2.2 Pin Description Of HC-05	08
3.2.3 Summary	09
3.3 Basic Description Of Android Application	09
3.3.1 Arduino Voice Application	10
3.4 Basic Description Of Controller Unit	10
3.4.1 Technical Specification Of Arduino UNO	11
3.4.2 Description Of Microcontroller Atmega328	12
3.4.3 Block Diagram Of Microcontroller Atmega328p	13
3.4.4 Pin Configuration Of Microcontroller Atmega328p	14
3.4.5 Pin Description	14

3.5	Description Of Relay Module	16
3.5.1	Basic Working Procedure Of Relays	17
3.5.2	Arduino Relay Module	18
3.5.3	Features Of 4 Channel 5v Relay	18
3.5.4	Pin Configuration Of Relay	19
3.6	Basic Arduino Software	20
3.6.1	Features Of Arduino Software	21
3.7	Description LCD Display16x2	23
3.7.1	Shapes And Sizes	24
3.7.2	Pin Descriptions	24
3.7.3	Control Lines	25
3.7.4	Initialization And Instructions	26
3.8	Wires	27
3.9	Power Supply	28
3.10	Loads	29
3.11	Cost Analysis	30
3.11.1	Cost Sheet	30
3.12	Conclusion	30
<b>Chapter 4</b>	<b>SOFTWARE ANALYSIS</b>	<b>31-34</b>
4.1	Introduction	31
4.2	Description Of Our Software	31
4.3	Flow Chart Diagram	33
4.4	Conclusion	34
<b>Chapter 5</b>	<b>HARDWARE IMLEMENTATION</b>	<b>35-39</b>
5.1	Introduction	35
5.2	Interfacing Of Bluetooth Module(Hc-05)	35
5.3	Interfacing With Application	36
5.4	Interfacing Of Relay Module	37
5.5	Load Interfacing	38
5.6	System Operation	39
5.7	Conclusion	39

<b>Chapter 6</b>	<b>RESULTS AND DISCUSSION</b>	<b>40-42</b>
6.1	Introduction	40
6.2	Experimental Setup	40
6.3	Result	41
6.4	Advantages	42
6.5	Disadvantages	42
6.6	Conclusion	42
<b>Chapter 7</b>	<b>CONCLUSION</b>	<b>43-44</b>
7.1	Conclusion	43
7.2	Application	43
7.3	Limitations Of The Works	44
7.4	Future Work	44
	<b>REFERENCES</b>	<b>45</b>
	<b>APPENDIX A</b>	<b>46-60</b>

# LIST OF FIGURES

FIGURE	FIGURE CAPTION	PAGE
2.1	General Block Diagram	3
2.2	Circuit Diagram	4
3.1	Logo Of Bluetooth	6
3.1.1	Bluetooth Module(HC-05)	7
3.1.2	Pins Of HC-05	8
3.2	Home Page Of Android Apps	10
3.3(a)	Arduino UNO	11
3.3(b)	Arduino UNO	12
3.3.1	Block Diagram Of ATmega328p	13
3.3.2	Pin Configurations Of ATmega328p	14
3.4	Relay	16
3.4.1	Construction Of Relay	17
3.4.2	Arduino Relay Module	18
3.4.3	Pin Configuration Of 4 Channel Relay Module	19
3.5	Arduino Software	20
3.6	LCD Different Models	24
3.7	Pin Diagram Of 16x2 Lines LCD	24
3.8	Flowchart Of Installing	26
3.9(a)	Ftom Jumper Wire	27
3.9(b)	Mtof Jumper Wire	27
3.9.1	220v Carried Wire	27
3.10	12v Power Adapter	28
3.11	Loads	29
4.1	Software Platform	31
4.2	Compiling Window	32
4.3	Flow Chart Of Our System	33
5.1	Interfacing Of Hc-05	35
5.2(a)	Interfacing With Application	36
5.2(b)	Interfacing With Application	36
5.3	Interfacing With Relay	37
5.4	Interfacing With Loads	38
6.1	Our Project	40
6.1.1	Setup Of Our System	41



# LIST OF TABLES

<b>TABLE</b>	<b>TABLE CAPTION</b>	<b>PAGE</b>
2.1	List Of Components Used In Circuit	5
3.1	Pin Description Of Bluetooth Module	9
3.2	Technical Specification Arduino Uno	11
3.3	Address Locations For A 1x16 Line LCD	23
3.4	Pin Descriptions Of LCD	25
3.5	Cost Sheet	30

# ACKNOWLEDGEMENT

Firstly I give thanks to Almighty Allah from the deep of my hearts. I would like to express my sincere gratitude to my honorable supervisor, Mr. Abu Shahir Md. Khalid Hasan, Lecturer, Department of Electrical and Electronic Engineering, Daffodil International University, who inspired me in every moment. I am thankful to him for his continuous encouragement, kind co-operation, and scholastic guidance all along the project work. He has always been extremely generous with his time, knowledge and ideas and allowed us great freedom in this research. I also want to convey our thankfulness to Prof. Dr. MShamsulAlam, Dean, Department of Electrical and Electronic Engineering for his help, support and constant encouragement. I am express my humble gratitude to all teachers of Department of Electrical and Electronic Engineering for their support in numerous ways throughout this project work. I am also grateful to the authors whose valuable research papers and books I have considered as reference in this project paper. Apart from that, I would like to thank my entire friends for sharing knowledge; information and helping to making this project a success. Also thanks for lending me some tools and equipment. Finally I would like to thanks my parents who have given me tremendous inspirations and supports. Without their mental and financial supports, I would not able to complete my project.

# ABSTRACT

Voice is utilized in this project for control switches. Reason for selecting voice is because it's simply being reproduced by human. Besides that, usage of voice provides a bearing system which will be effective and convenient to be used. The applying of this technique involve modifying the switch system from the normal approach that is physical contact with the switch to a safer approach wherever the usage of voice to exchange all the physical contact. Home automation is building automation for a home, called a smart home or smart house. It involves the management and automation of lighting, heating, ventilation, air con, and security, also as home appliances like ovens or refrigerators/freezers. There square measure completely different strategies in wireless technology like Bluetooth, WIFI, and GSM. During this paper, new style and completely different home appliances square measure bestowed. The home automation application that has presented in this paper is the ability to control the bulb using a smart phone application with Bluetooth wireless technology. The relays is used to connect these appliances to the input/ output ports of the board .The design is a low cost, flexible and using a modern technology and devices for this application. Application of wireless Bluetooth connection in control board enables a simplified way to system installation. The system has been built and operated successfully.

# CHAPTER 1

## INTRODUCTION

### 1.1 INTRODUCTION

Home automation permits controlling house appliances like door, light, fan, oven etc. It conjointly provides associate emergency system and residential security. It allows the consumer additional control of his home it facilitates several conditions, as an example, if the consumer is permit to on his home appliance to controlled light, fan or any other load, or pre-heating kitchen appliance once he got home, therefore, several manual actions is replaced by home automation that reduces human efforts and time-saving. During this paper, new style and totally different home appliances are given. Many different electric applications is controlled by using Arduino UNO Microcontroller Based Home Automation System. The system is used to bulb using a smart phone application with Bluetooth wireless technology.

### 1.2 AIM OF THE PROJECT

- For the house automation system by associate Android application.
- Setup a primarily based microcontroller automation system.
- Change devices and appliances through the electrical device module.
- Style a correct circuit diagram.
- Implement the planning.

### 1.3 SCOPES

- Without cost home automation system project to make easy daily life works in millions of houses.
- It can be used commercially in industries.
- Portable Android application to use it frequently in hand.
- Bluetooth technology which might be utilized in wireless affiliation for cellular phones and residential appliances. Bluetooth technology offers associate

economical methodology for dominant home automation. It's an occasional value and a secured technology.

- Several relays use associate magnet to automatically operate the switch and supply electrical isolation between 2 circuits. During this project there ought to isolate one circuit from the opposite, however we are going to use associate ArduinoUNO to regulate the relay.

## **1.4 METHODOLOGY**

This system is used to control all the appliances that connected to the microcontroller. The methodology of the proposed system is mainly divided into the three steps. In the initial step an Android application interfaces with the Bluetooth module. After that within the second step, the microcontroller receives the signal that has been causing by the Bluetooth module. Then microcontroller sends the activation signal to the relay module. In the last step relay module switches the devices that connected to the relay.

## **1.5 ORGANIZATION OF THE REPORT**

This project report has seven chapters in total. The first chapter describes an idea about our project "Voice Control Home Automation", Brief description of the project, scopes and methodology. The second chapter about history, block diagram, circuit diagram, list of components. The chapter third about component description, cost analysis of our system. The chapter fourth software analysis & program explanation. The chapter five hardware implementation. Then chapter six describes result & discussion properly. Finally, chapter seven gives the concluding remarks, limitation of our system and suggestion for the future works.

# CHAPTER 2

## SYSTEM REVIEWS

### 2.1 INTRODUCTION

The home automation system permits people to regulate home appliances by using a sensible phone application. It's necessary to appear on hardware and users good phone software for developing a home automation system. The product can deal with many home appliances such as lights, door lock etc. In this chapter we will discuss about Bluetooth module (HC-05), Relay module and Arduino UNO, 'Automation Smart Home' apps, Arduino compiler, Block diagram & circuit diagram of our System.

### 2.2 GENERAL BLOCK DIAGRAM

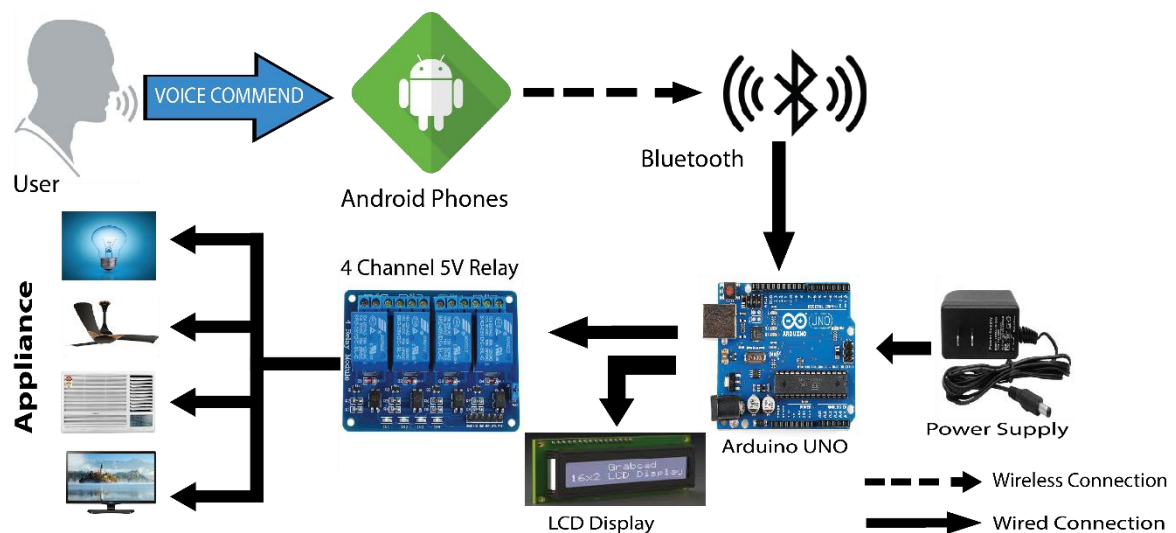


Fig. 2.1 General Block Diagram

#### 2.2.1 BLOCK DIAGRAM DESCRIPTION

The block diagram of home automation system as shown in fig.2.1, is using the Arduino UNO, we can operate the Bluetooth module, relay module. When the dedicated command is available, then Bluetooth module (HC-05) sends a digital output to control unit. At this time the control units (Arduino UNO) will send commands to relay module to switching ON/OFF.

## 2.3 CIRCUIT DIAGRAM

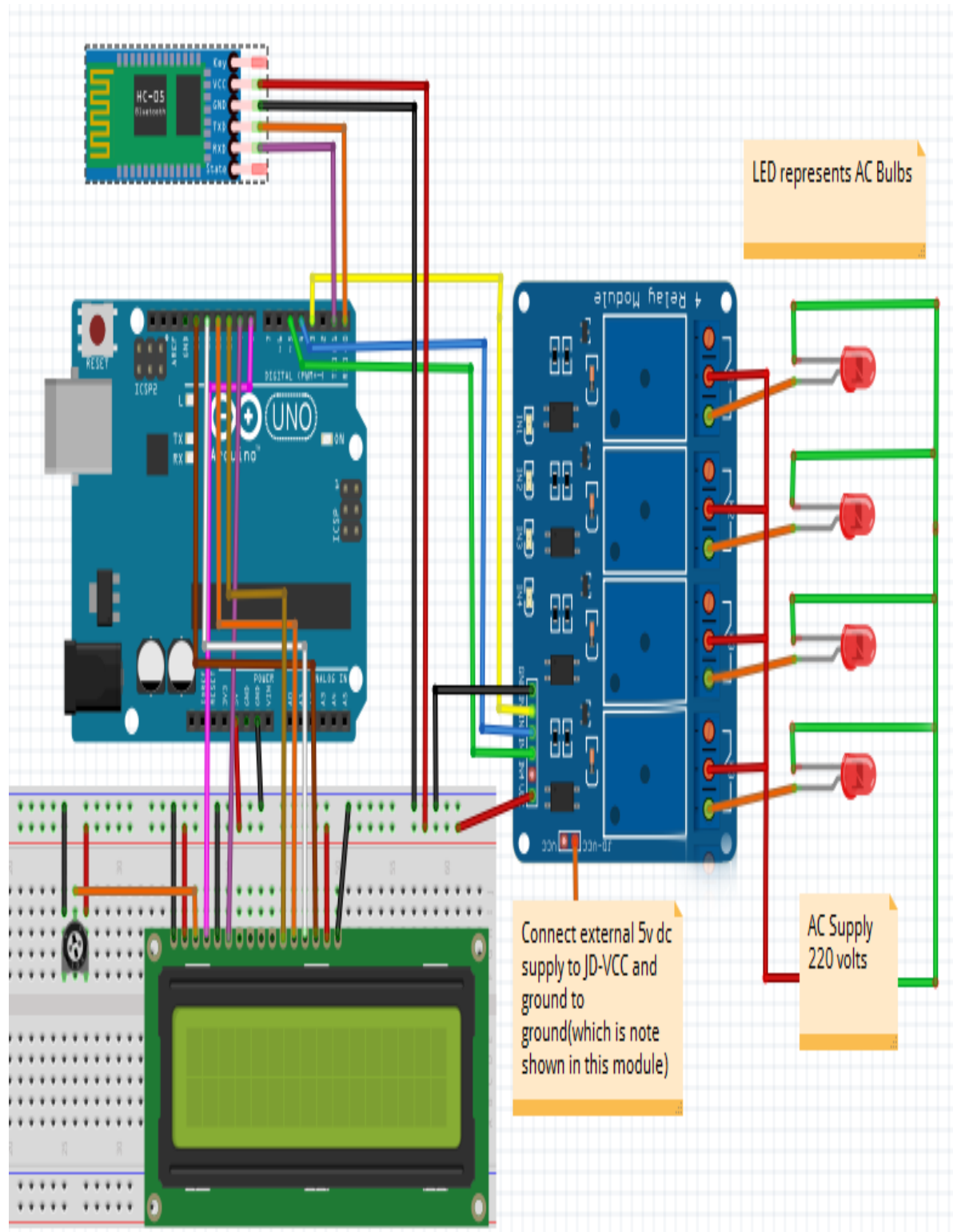


Fig. 2.2 Circuit Diagram

### 2.3.1 WORKING PROCESS OF OUR CIRCUIT

The Voice Control Home Automation project operated by a free application known as “ARDUINO BLUETOOTH CONTROLLER”. This application operate the various appliances attached to our ArduinoUNO and relays. Once the toggle switch on the applying are press, corresponding Bluetooth signals are transfer data from our android cell phone to the Bluetooth module we've attached to our Arduino. The Arduino finds out that the signal was sent and compares it to the predefined signals appointed for every appliance. Once it recognize that signal, the Arduino turn on the relay connected to its digital pin by carrying out 5V through it. Thus the relay is turn on and also the similar appliance connected to the relay is turned on exactly same way. To turn it OFF, Arduino UNO transfer 0V or logic low to its digital pin.

### 2.4 LIST OF COMPONENTS USED IN CIRCUIT

NO	COMPONENT NAME	QUANTITY	USED
01	Bluetooth Module (HC-05)	01	To direct communication.
02	Microcontroller- ATmega328P(ArduinoUNO)	01	To Control the System.
03	Relay Module	01	To switching.
04	Arduino compiler		To compile code.
04	Arduino voice control app	01	To giving command.
06	Ac to dc power supply	02	Power Supply
07	LCD Display	19	Showing Text
09	Load	03	Showing Output.

Table 2.1 list of component we used

### 2.5 CONCLUSION

The home automation system which used in many applications because of its desirable properties like homes, hostels, industries, vehicles.



# CHAPTER 3

## COMPONENT DESCRIPTION

### 3.1 INTRODUCTION

System hardware design composed of Arduino UNO, 4 Channel 5V Relay Module, Bluetooth Module (HC-05), Wires, Arduino compiler, Android application. In this chapter we will discuss about component description, features, working procedure and cost analysis of our all component.

### 3.2 DESCRIPTION OF BLUETOOTH MODULE (HC-05)

Nowadays, everything goes wireless. Phones, computers, game controllers, everything. Wireless technology has allowed us to use electronic devices with AN unbound freedom that simply is not possible with awkward cables hanging off our device. There are several kinds of wireless property like Wi-Fi, Cellular Data, Zigbee, however, one among the foremost in style and wide used wireless protocols is Bluetooth. Whilst the Bluetooth HC-5 standard was declared earlier this year (2018), we'll be taking a look at using the Bluetooth 4.2 standard because it is most typically on the market at the instant.

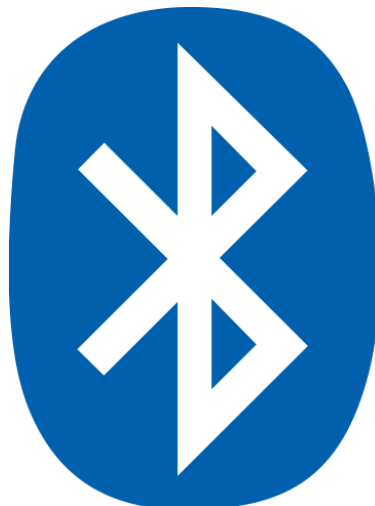


Fig 3.1 Logo of Bluetooth.

HC-05 module is a simple to utilize Bluetooth SPP (Serial Port Protocol) module, intended for straightforward remote sequential association setup. The HC-05 Bluetooth Module can be utilized in a Master or Slave setup, making it an incredible answer for remote correspondence. This sequential port Bluetooth module is completely qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio handset and base band. It utilizes CSR Blue-center 04-External single chip Bluetooth framework with CMOS innovation and with AFH (Adaptive Frequency Hopping Feature).The Bluetooth module HC-05 is a MASTER/SLAVE module. As a matter of course the processing plant setting is SLAVE. The Role of the module (Master or Slave) can be designed just by AT COMMANDS. The slave modules can't start an association with another Bluetooth gadget, however can acknowledge associations. Ace module can start a connection with different gadgets.

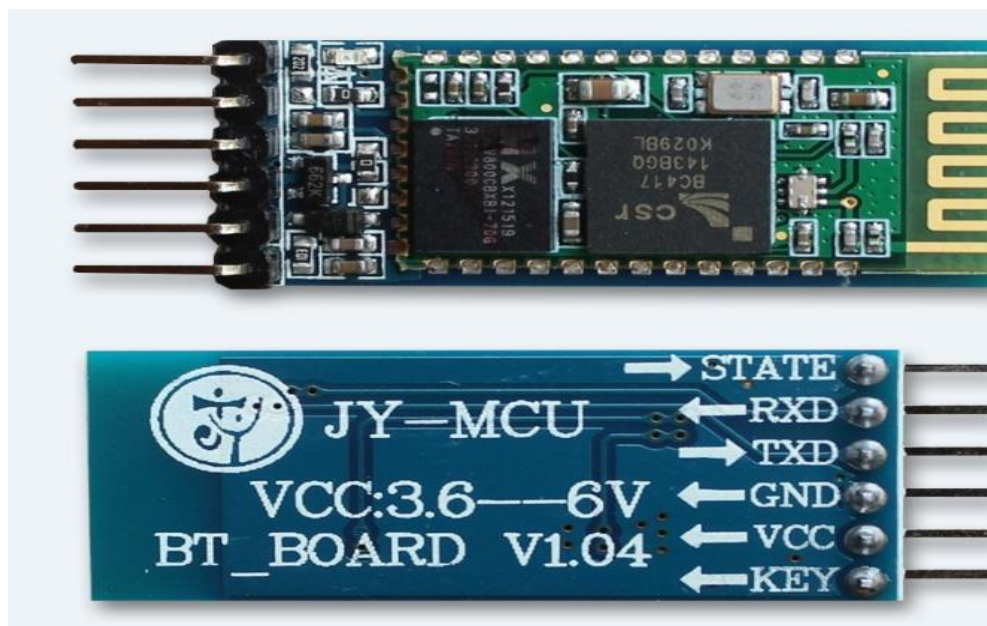


Fig. 3.1.1 Bluetooth Module (HC-05)

### 3.2.1 FEATURES OF BLUETOOTH MODULE (HC-05)

#### FEATURE OF HARDWARE

- Typical -80dBm sensitivity.
- Up to +4dBm RF transmit power.
- 3.3 to 5 V Input/Output.
- PIO (Programmable Input/Output) control.
- With integrated antenna.
- With edge connector.

#### FEATURE OF SOFTWARE

- Default Data bits 8, Stop bit 1, Baud rate 9600, Parity No parity.
- Automatically connect to the last device when power on.
- Automatically connect with paired device
- Default PINCODE is “1234” or auto-pairing.

### 3.2.2 PIN DESCRIPTION OF HC-05

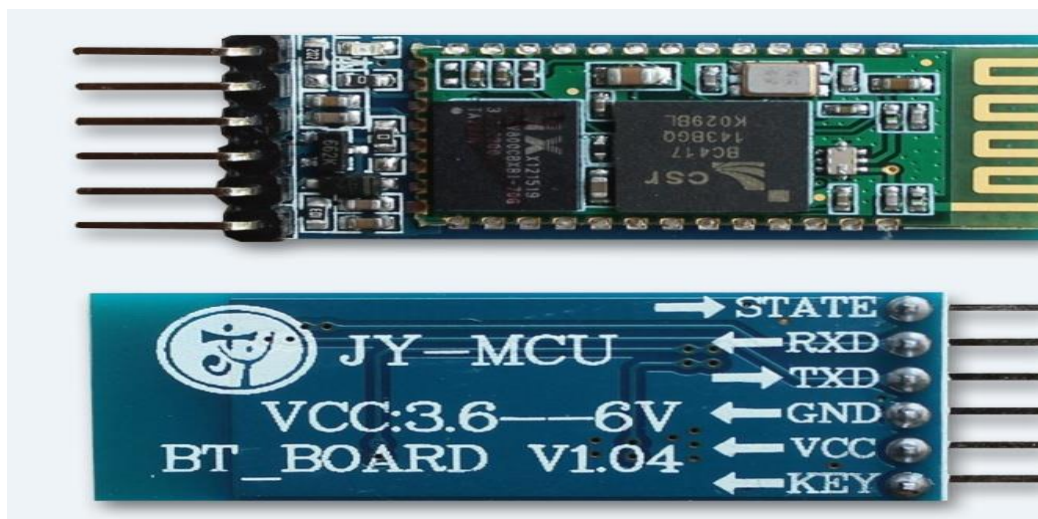


Fig 3.1.2 Pins of HC-05.

SL.	PIN NAME	DESCRIPTION
1	+5V or 3.3V	+5V supply or Power Pin.
2	TX	The Data/Command to be transmitted is sent through this pin.
3	RX	The Received data is read from this pin.
4	KEY/EN	Input pin, which alters module between the Data mode and the AT Command mode.
5	STATE	Output pin, the state of the module is indicated through this pin.
6	GROUND	0V / GND or Power Pin

Table 3.1 Pins Description of Bluetooth Module

### 3.2.3SUMMARY

In the end this is a very good device to use in home or industry. This is a low cost project so that it is very cost effective and efficient. In short term uses of communication it is better than other wireless communication.

### 3.3 Basic Description of Android Application

An android application could be a product framework application running on the android stage. Since the android stage is developed for cell phones, a run of the mill android application is intended for an advanced mobile phone or a tablet PC running on the android OS. Despite the fact that an android application will be made available by designers through their sites, most android applications are transferred and distributed on the android Market, a web store devoted to those applications. The android Market choices each free and estimated applications. Golem applications are composed inside the Java programing dialect and use Java center libraries. They're first assembled to Dalvik executable to keep running on the Dalvik virtual machine that is a virtual machine exceptionally intended for cell phones. Designers may exchange the golem PC code improvement pack (SDK) from the android site. The SDK incorporates apparatuses, test code and important records for making android applications.

### 3.3.1 ARDUINO VOICE APPLICATION

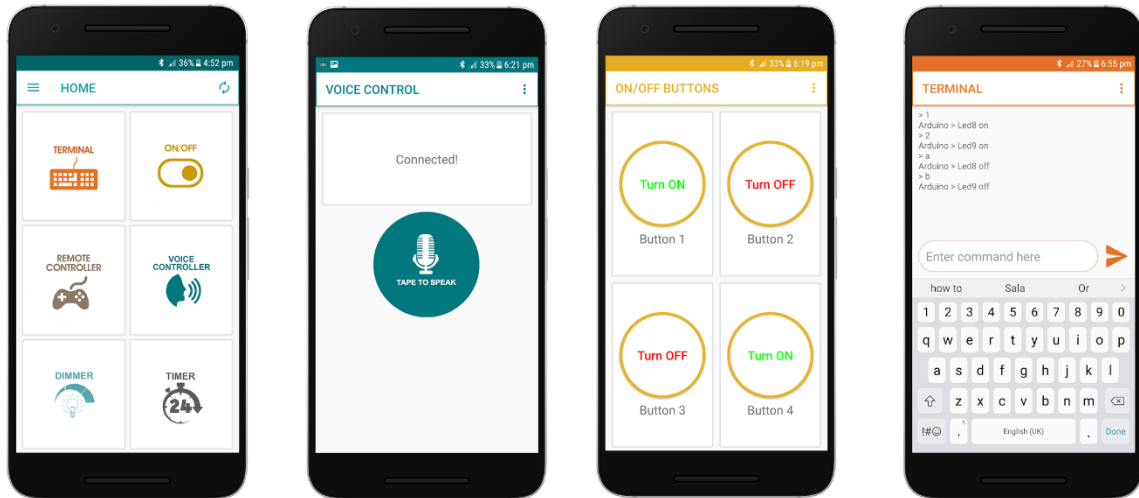


Fig 3.2 Home page of Android Apps.

### 3.4 BASIC DESCRIPTION OF COMMEND UNIT

In the commend unit, we are using Arduino equipment board (with AVR microcontroller). With the assistance of Arduino 1.6.8 software system we are able to simply program AVR IC, as our requirement. Arduino is an open-source electronics prototyping platform supported versatile, easy-to-use hardware and software system. It's meant for artists, designers, hobbyists, and anyone fascinated by making interactive objects or environments. The microcontroller on the board is programmed exploitation the Arduino artificial language (based on Wiring) and also the Arduino development setting (based on Processing). The Arduino UNO could be a microcontroller board supported the ATmega-328. Fourteen digital I/O pins in this device (which half-dozen will be utilized as Pulse width modulation outputs), Six analog inputs, A 16 MHz ceramic resonator, a USB port, an impact jack, an ICSP header, and a reset push. It contains everything required to support the microcontroller, easily connect it to a laptop with a USB link or power it with an AC-to-DC adapter or battery to begin.

### 3.4.1 TECHNICAL SPECIFICATION ARDUINO UNO



Fig 3.3(a) ArduinoUNO.

Microcontroller	ATmega328
Operating Voltage	5V
Supply Voltage (recommended)	7-12V
Maximum supply voltage (not recommended)	20V
Digital INPUT/OUTPUT Pins	Fourteen (of which six provide PW output)
Analog Input Pins	Six
DC Current per INPUT/OUTPUT Pin	40 Ma
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 utilized by boot loader
SRAM	2 KB (ATmega328)

Table 3.2 Technical specification of ArduinoUNO

### 3.4.2 DESCRIPTION OF MICROCONTROLLER ATmega328

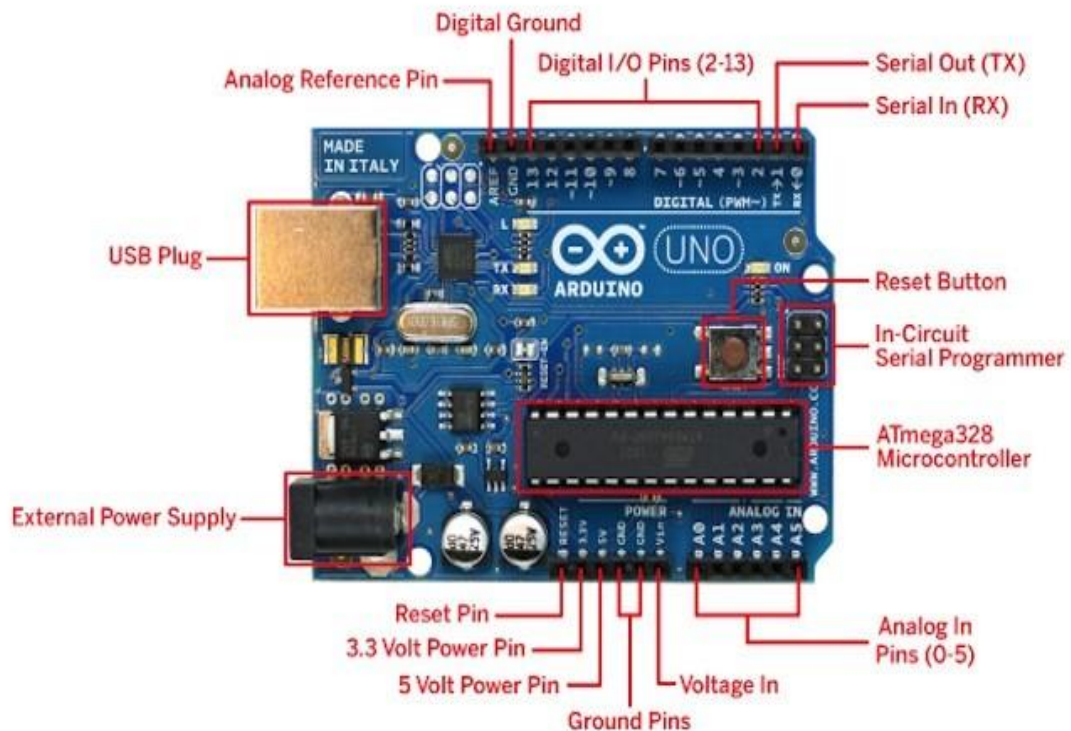


Fig 3.3(b) ArduinoUNO.

The Arduino UNO could be a microcontroller board supported the ATmega-328 (Datasheet). Fourteen digital I/O pins in this device (which half-dozen will be utilized as Pulse width modulation outputs), Six analog inputs, A 16 MHz ceramic resonator, a USB port, an impact jack, an ICSP header, and a reset push. It contains everything required to support the microcontroller, easily connect it to a laptop with a USB link or power it with an AC-to-DC adapter or battery to begin. The UNO varies from every previous board in that it doesn't utilize the FTDI USB-to-serial driver chip. Rather, it includes the Atmega328 modified as a USB-to-sequential converter. "UNO" means "One" in Italian and is called to mark the upcoming release of Arduino 1.0. The UNO and version 1.0 are going to be the reference from of Arduino, moving forward. The UNO is that the latest in an exceedingly series of USB Arduino boards and also the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino board.

### 3.4.3 BLOCK DIAGRAM OF MICROCONTROLLER – (ATmega328)

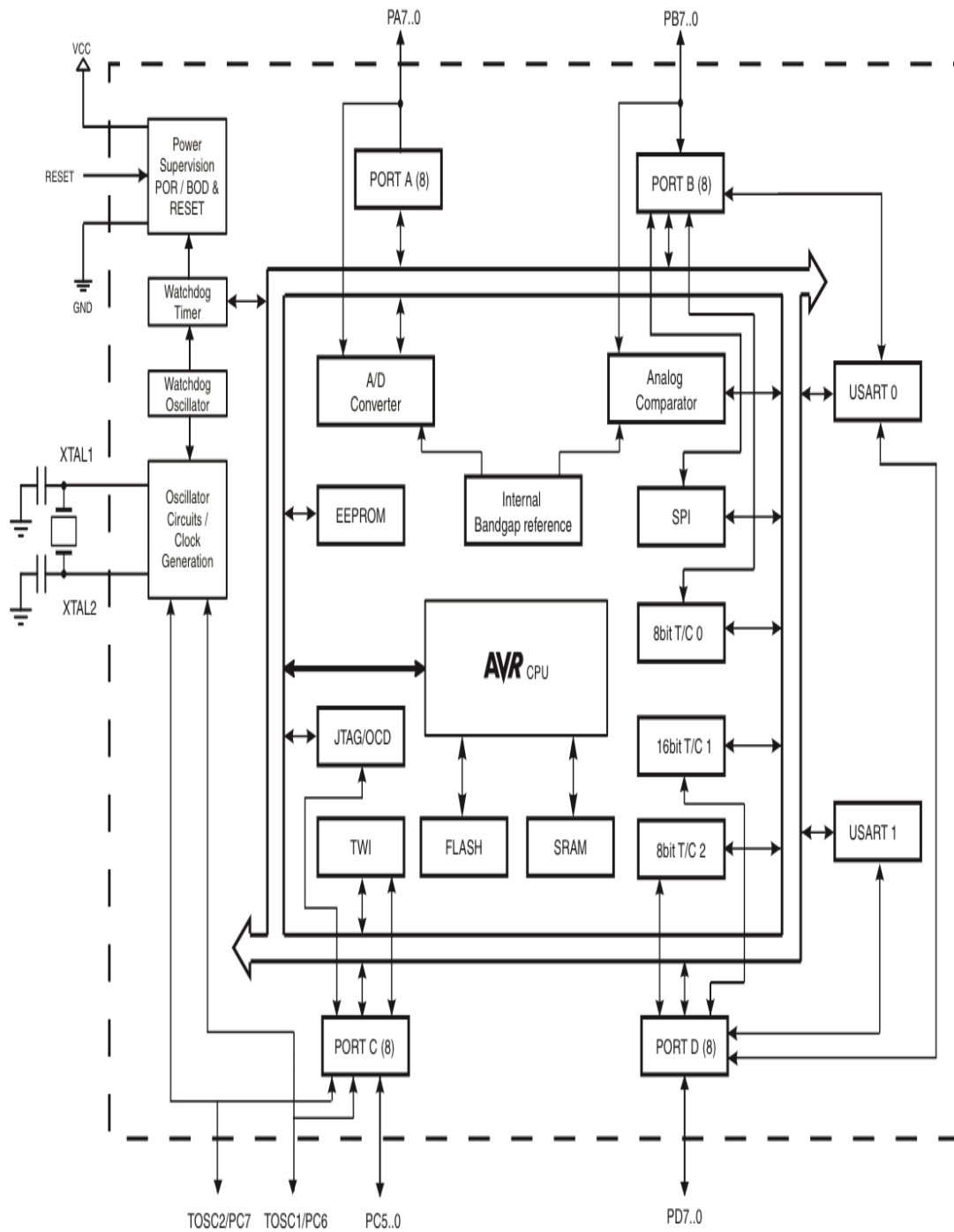


Fig. 3.3.1 Block Diagram of Microcontroller – (Atmega328)



### 3.4.4 PIN CONFIGURATIONS OF MICROCONTROLLER (ATmega328)

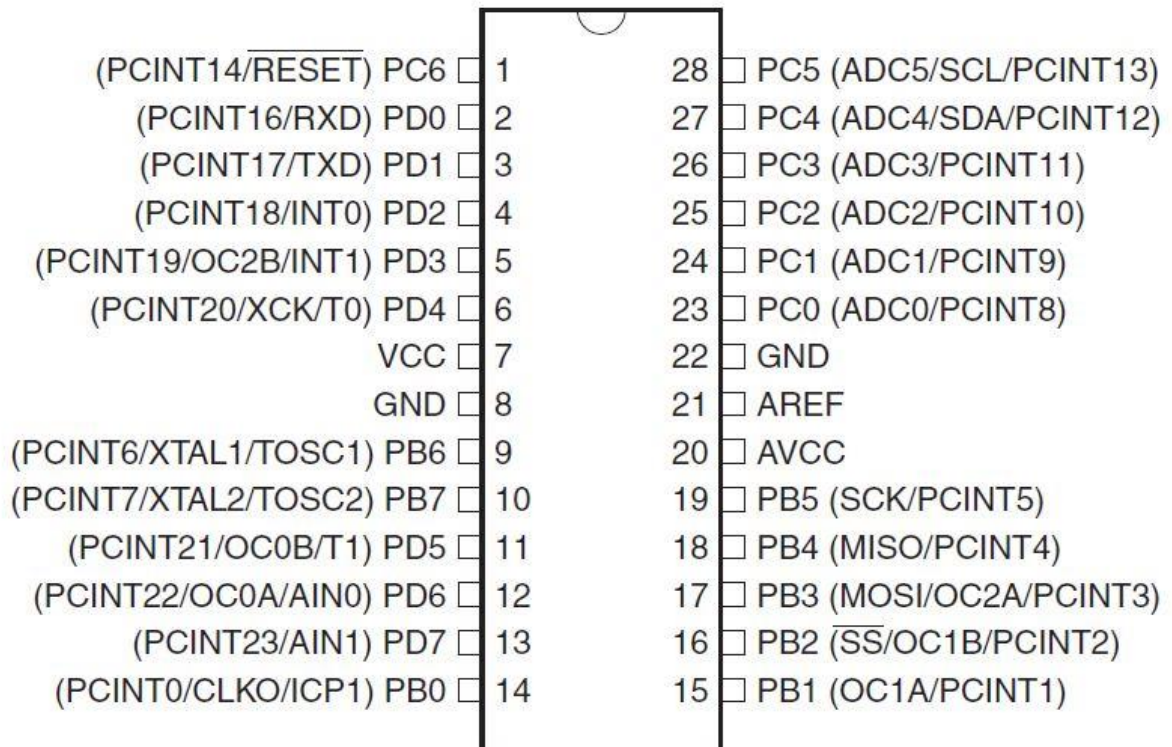


Fig. 3.3.2 Pin Configurations of Microcontroller – (Atmega328)

### 3.4.5 PIN DESCRIPTIONS

The underneath gives a depiction for every one of the pins, alongside their capacity-

The four pins are as below:

- **VIN:** The input voltage to the Arduino board when it's utilizing an external power supply (as against five volts from the USB association or another regulated power source). We are able to provide voltage through this pin, or, if supply voltage via the power jack, access it through this pin.
- **5V:** The regulated power supply utilize to power the microcontroller and different elements on the board. This will come back either from VIN via an on-board regulator, or be provided by USB or another regulated 5V supply.
- **3.3V:** A 3.3V supply generated by the on-board regulator. Highest current draw is 50 mA.
- **GND (Ground pins)**

## Input & Output of Arduino UNO

Each of the fourteen digital pins on the Mega can be utilized as an input or output, using pin Mode, digital Write, and digital Read functions. They operate at 5 volts. Every pin will be provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 ohms. Additionally, some pins have specialized functions.

- **Serial:**Serial 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega328 USB-to-TTL Serial chip.
- **External Interrupts:**2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM:**3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function.
- **SPI:**10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **LED:**There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I 2C 20 (SDA) and 21 (SCL):**The ATmega328 additionally supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus.

There are a couple of other pins on the board

- **AREF:**The reference voltage for the simple sources of info (analog Inputs). Utilized with simple Reference.
- **Reset:**Bring this line LOW to reset the microcontroller. Normally used to add a reset button to shields which block the one on the board.

**Memory:**The ATmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is utilized for the boot loader), 2 KB of SRAM and 1 KB of EEPROM (which will be read and written with the EEPROM library).

### 3.5 DESCRIPTION OF RELAY MODULE

The Relay is an attraction switch operated by a comparatively very little physical phenomenon that is ready to activate or off away larger current. The middle of a relay is an magnet (a coil of wire that becomes a transitory magnet once electricity flows through it).we tend to able to suppose a relay as a form of electrical lever switch it on with a small amount current and it switches on ("leverages") another appliance using a heap of larger current. Why is that useful? Because the name suggests, many sensors unit of measurement implausibly sensitive things of kit and switch out alone little electrical currents. However, we'd like them to drive larger things of kit that use larger currents. Relays bridge the gap, creating it potential for tiny currents to activate larger ones. Which implies relays can work either as switches (turning things on and off) or as amplifiers (converting little currents into larger ones).



Fig 3.4 Relay

### 3.5.1 BASIC WORKING PROCEDURE OF RELAYS

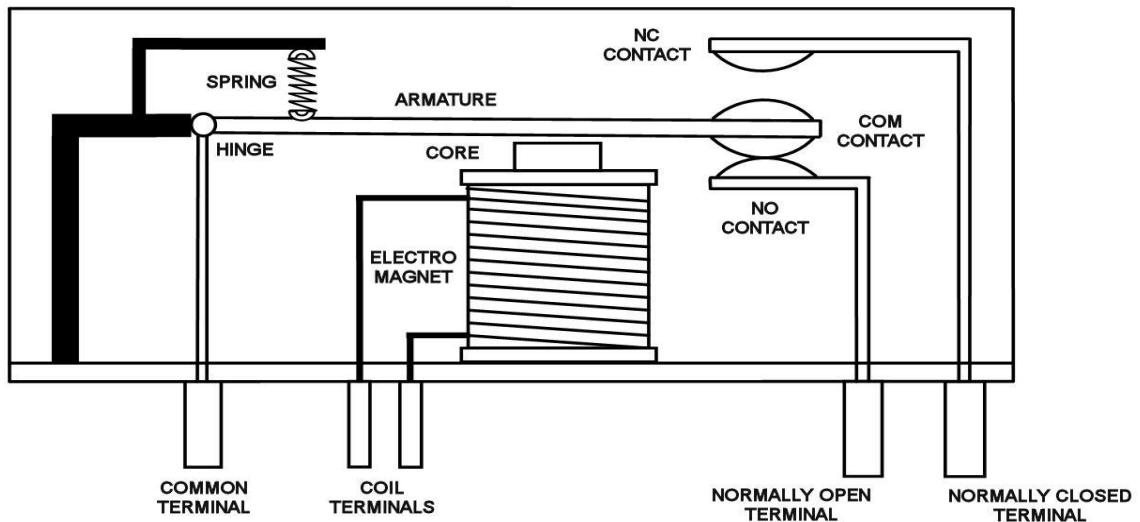


Fig 3.4.1 Construction of relay.

The diagram shows an internal section diagram of a relay. An iron core is enclosed by a solenoid coil. As shown, the power supply is given to the magnet through a switch and through contacts to the load. Once current starts flowing through the management coil, the magnet starts energizing and therefore intensifies the magnetic flux. Therefore the higher wiper arm starts to be attracted to the lower mounted arm and therefore closes the contacts forming a circuit for the power to the load. On the opposite hand, if the relay was already de-energized once the contacts are closed, then the contact moves oppositely and breaks a circuit. As soon as the coil current is off, the movable coil goes back to its initial position. This force is nearly up to 0.5 the strength of the magnetic attraction. This force is principally provided by a pair of factors. They're the spring and put together gravity. Relays are within the main created for two basic operations. One is low voltage application and therefore the distinction is high voltage. For low voltage applications, extra preference goes to lean to scale back the noise of the complete circuit. For high voltage applications, they are within the main designed to chop back a development referred to as arcing.

### 3.5.2 ARDUINO RELAY MODULE

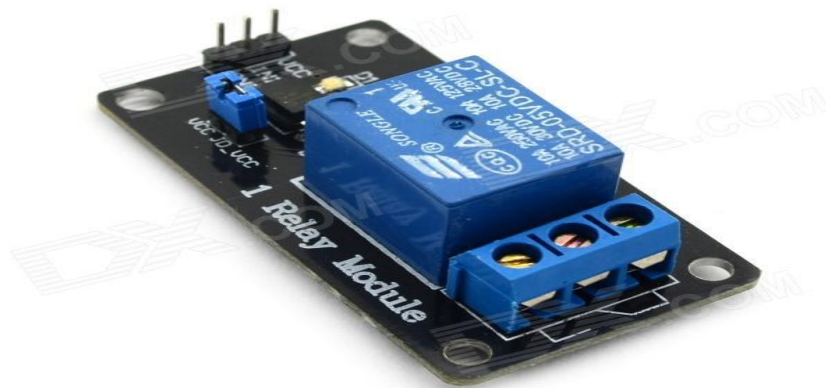


Fig3.4.2 ArduinoRelay Module.

The relay module is an electrically operated switch that permits us to turn on or off a circuit using voltage or current a lot of more than a microcontroller might handle. There is no connection between the low voltage circuit operated by the microcontroller and also the high power circuit. The relay protects each circuit from each other. The relay protects every circuit from one another. Every channel within the module has three connections named NC (normally closed), COM, and NO (normally open). Depending on the input signal trigger mode, the jumper cap is placed at a high-level effective mode that ‘closes’ the normally open (NO) switch at high-level input and at a low-level effective mode that operates a similar however at low-level input.

### 3.5.3 FEATURES OF 4 CHANNEL 5V RELAY

- Size: 75mm (Length) \* 55mm (Width) \* 19.3mm (Height)
- Weight: 61g
- PCB Color: Blue
- There are four fixed screw holes at each corner of the board, easy to install and fix. The diameter of the hole is 3.1mm
- High-quality Single relay is used with single pole double throw, a common terminal, a normally open terminal, and a normally closed terminal
- Optical coupling isolation, good anti-interference.
- VCC is a system power source, and JD\_VCC is a relay power source. Ship 5V relay by default. Plug jumper cap to use
- The maximum output of the relay: DC 30V/10A, AC 250V/10A

### 3.5.4 PIN CONFIGURATION OF RELAY

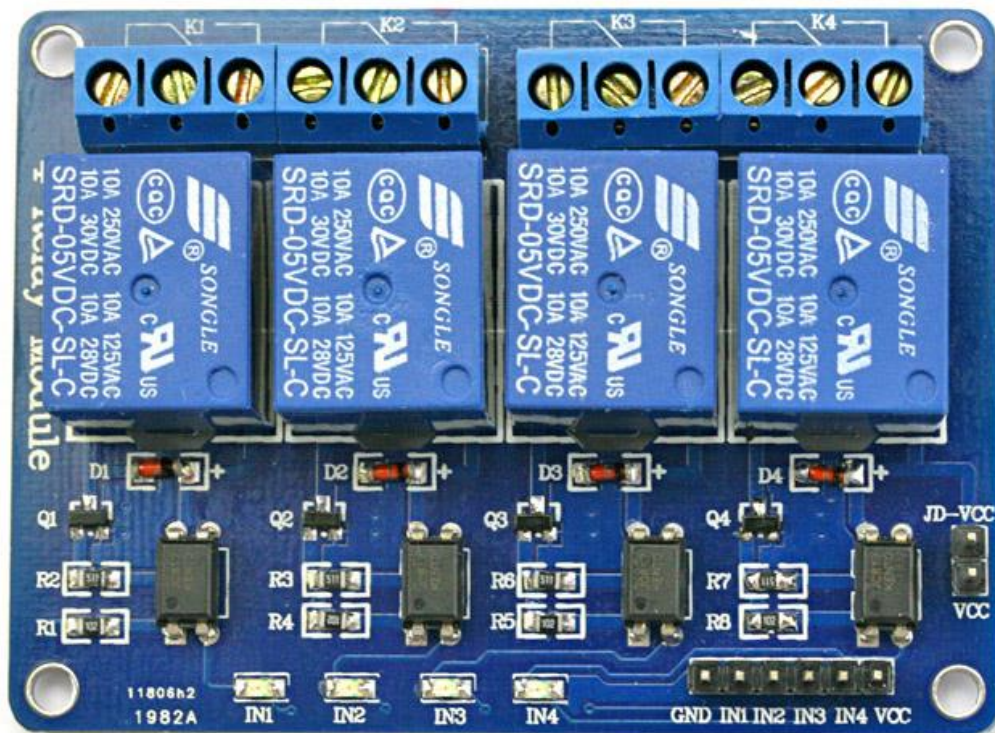


Fig 3.4.3 Pin configuration of relay module.

1. VCC 5V DC
3. IN1 low/high output
3. IN2 low/high output
4. IN3 low/high output
5. IN4 low / high output
6. GND (Ground)

### 3.6 BASIC ARDUINO SOFTWARE



Fig 3.5 Arduino Software.

The Arduino software is published as open source tools, available for extension by experienced programmers. The language may be distended through C++ libraries, and people needing to perceive the technical details will build the leap from Arduino to the AVR C programming language on which it's based. Similarly, we can add AVR-C code directly into our Arduino programs that it's primarily based. The Arduino Integrated Development Environment - or Arduino Software package (IDE) - contains a text editor for writing code, a message space, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and real hardware to transfer programs and communicate with them. Programs are written utilized Arduino Software (IDE) are known as sketches. These sketches are written text editor and are saved with the file extension .ino. The editor has options for cutting/pasting and for searching/replacing text. The message area gives feedback whereas saving and exporting and conjointly displays errors. The console displays text output by the Arduino software (IDE), together with complete error messages and alternative info. The bottom right-hand corner of the window displays the organized board and port. The toolbar buttons allow us to verify and transfer programs, create, open, and save sketches, and open the serial monitor.

### 3.6.1 Features of Arduino Software

- **File**
- **Edit**
- **Sketch**
- **Tools**
- **Help**

#### **Sketch book**

The Arduino Software (IDE) uses the concept of a sketchbook a standard place to store our programs (or sketches). The sketches in our sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time we run the Arduino software, it will automatically create a directory for our sketchbook. We can view or change the location of the sketchbook location from with the Preferences dialog. Beginning with version 1.0, files are saved with an .ino file extension. Previous versions use the .pde extension. We may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

#### **Tabs, Multiple Files, and Compilation**

Allows we to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

#### **Uploading**

Before uploading our sketch, we need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for a Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, we look for USB serial device in the port section of the Windows Device Manager. On Linux, Once we've



selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Decimal) that lack auto-reset, we'll need to press the reset button on the board just before starting the upload. On most boards, we'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error. When we upload a sketch, we're using the Arduino boot loader, a small program that has been loaded on to the microcontroller on our board. It allows us to upload code without using any additional hardware. The boot loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The boot loader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

### **Libraries**

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more `#include` statements at the top of the sketch and compile the library with our sketch. Because libraries are uploaded to the board with our sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its `#include` statements from the top of our code. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, we do can import a library from a zip file and use it in an open sketch.

### **Third-Party Hardware**

Support for third-party hardware can be added to the hardware directory of our sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, boot loaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "Arduino" as the sub-directory name or we'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

## Serial Monitor

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to Serial. Begin in our sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun our sketch execution to the beginning) when we connect with the serial monitor. We can also talk to the board from Processing, Flash, MaxMSP, etc.

## Preferences

Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

## 3.7 DESCRIPTION OF LCD DISPLAY 16X2

A liquid crystal display (LCD) could be a thin, flat display device created from any range of color or monochrome pixels arrayed ahead of a light supply or reflector. every pixel consists of a column of liquid molecules suspended between 2 clear electrodes, and 2 polarizing filters, the axes of polarity of which are perpendicular to each other. LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller. Data can be placed at any location on the LCD. For 16×2 LCD, the address locations are

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Table 3.3 Address locations for a 1x16 line LCD

### 3.7.1 SHAPE AND SIZES

Even restricted to character-based modules, there's still a good style of shapes and sizes on the market. Line lengths of 8, 16,20,24,32 and 40 characters are all standard, in one, two and four line versions.

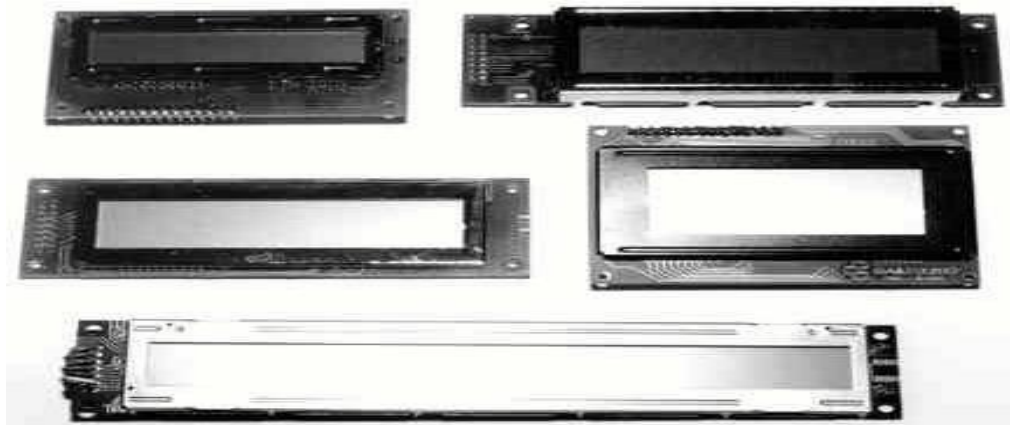


Fig 3.6 LCD Different Models.

Totally different LC technologies exist. “Supertwist” types, for example, offer improved contrast and viewing angle over the older “twisted nematic” types. Some modules are on the market with backlighting, in order that they'll be viewed in dimly-lit conditions. The rear lighting is also either “electro-luminescent”, requiring a high voltage electrical converter circuit, or easy LED illumination.

### 3.7.2 PIN DESCRIPTION

Most of the LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two extra pins in each for back-light LCD connections).

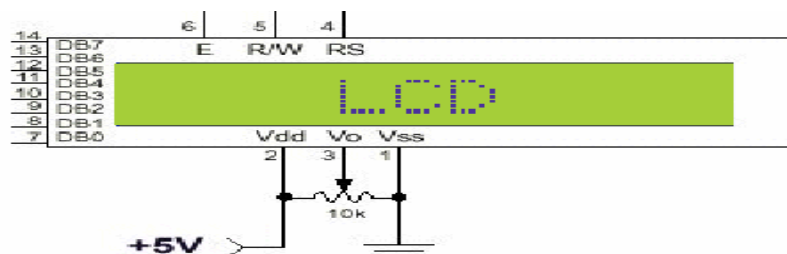


Fig 3.7 Pin Diagram of 16x2 line LCD.

PIN	SYMBOL	FUNCTION
1	GND	Power Supply(GND)
2	Vcc	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	RW	Data Bus Line
6	E	Enable
7-14	DB0-DB7	8-bit data pins
15	A	Power supply for LED (+)
16	K	Power supply for LED (-)

Table 3.4 Pin Description of LCD

### 3.7.3 CONTROL LINES

**EN:**Line is named "Enable." This control line is used to inform the LCD that you are sending it information. To send information to the LCD, our program should make sure this line is low (0) and then set the other two control lines and/or place information on the data bus. Once the opposite lines are fully prepared, bring EN high (1) and sit up for the minimum quantity of time needed by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

**RS:**Line is named "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

**RW:**Line is named "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low. Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the

user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

### 3.7.4 INITIALIZATION AND INSTRUCTIONS

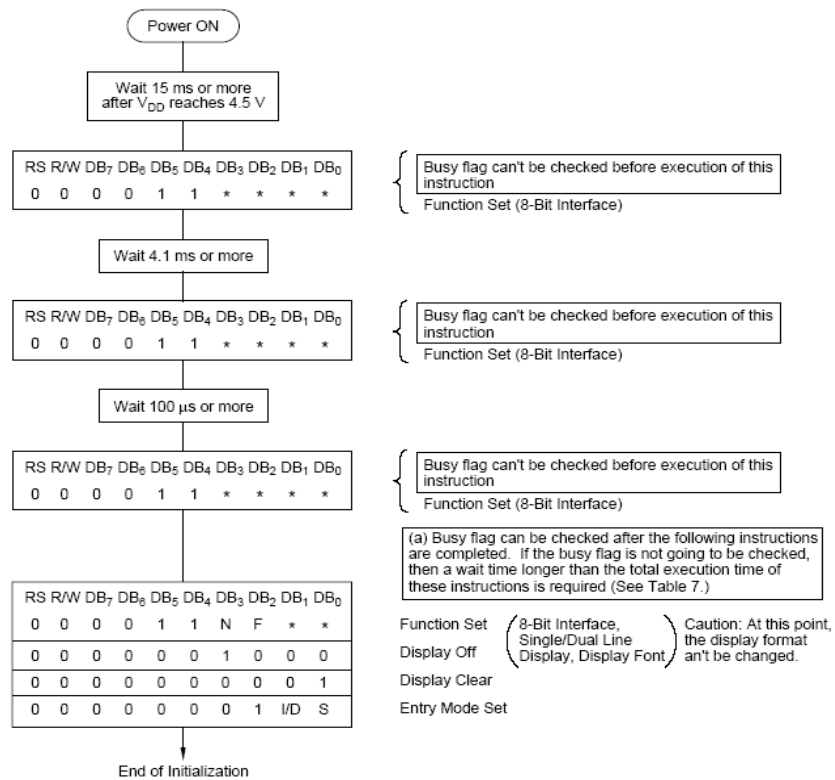


Fig 3.8. Flow Chart of Installation

If the power conditions for the normal operation of the internal reset circuit don't seem to be glad, then executing a series of instructions must initialize LCD unit. The procedure for this initialization method is as above show.

### 3.8 WIRES

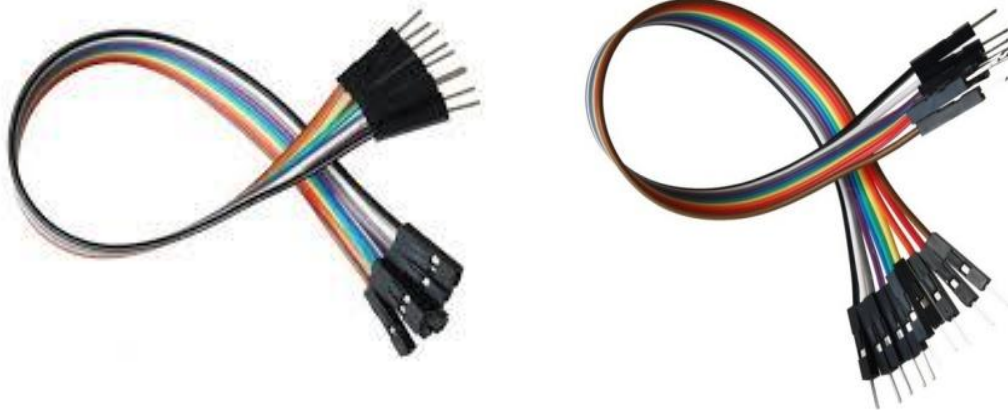


Fig 3.9(a) Female to Male Jumper Wire.      Fig 3.9(b) Male to Male jumper wire.

In this project we need couples of Female to Male and Male to Male jumper wire. That is for connecting Bluetooth module and relay module to the ArduinoUNO board.



Fig 3.9.1 220v carried wire.

And we need some 220v carried wire to connect load in the relay module.

### 3.9 POWER SUPPLY

An AC adapter, AC/DC adapter, or AC/DC converter is a type of external power supply, typically enclosed during a case like to an AC plug. In the power supply section, we use one step down transformer to step down the voltage from 220-volt ac to 9-volt dc. The output of the transformer is further connected to the two diode circuit. Here two diode work as a full wave rectifier circuit. The output of the full wave rectifier is now filtered by the capacitor. Capacitor converts the pulsating dc into smooth dc with the help of charging and discharging effect. The output of the capacitor is now regulated by the IC 7805 regulator. IC 7805 provides a 5-volt regulation to the circuit and provides a regulated 5-volt power supply. The output of the regulator is currently once more filter by the capacitor. In the output of the capacitor, we use one resistor and one led in series to produce a visible indication to the circuit.



Fig 3.10 12V Power supply.

### 3.10LOADS

Bulbs are connected as a load. We are using three bulbs and one socket for charging or plug in any ac device to define how to control appliances like fan, TV switch, Door lock, Main Switch etc.This is not a fixed load, it can be vary in different places and different areas of use.



Fig 3.11 Loads



### 3.11 COST ANALYSIS

In this section we will show cost of our project that means cost sheet representation of our project.

#### 3.11.1 COST SHEET

No	Component Name	Quantity	Purchase Price TK
01	Bluetooth Module(HC-05)	1	350.00
02	ArduinoUNO	1	400.00
03	4 Channel 5V Relay Module	1	400.00
06	Arduino voice app	1	Free
07	Power Supply	1	350.00
08	Arduino Software	1	Free
09	Breadboard	1	50.00
10	Wires		100.00
11	Bulbs with holders		300.00
	<b>Total Cost</b>		<b>=1950.00</b>

Table 3.5 Cost Sheet

#### Comparison

Our all components are available in market. We get all components are very reasonable price. So that we make this project more cost efficient.

### 3.12 CONCLUSION

Five main Component & some tools are used in this system to makes it .This Project is used to control the house appliances. Our all component are very simple &available in our country market.

# CHAPTER 4

## SOFTWARE ANALYSIS

### 4.1 INTRODUCTION

In this chapter the software system used and also the language within which the program code is outlined is mentioned and also the program code dumping tools are explained. The chapter conjointly documents the development of the program for the application.

### 4.2 DESCRIPTION OF OUR SOFTWARE

The Arduino is open source environment makes it simple to write, re-write the code and upload it to the INPUT/OUTPUT board. It runs on Windows, Mac OS X, and Linux. The setting is written in Java and based on Processing, avr-gcc, and different open source software. The screen shot of Arduino1.6.8 is shown below...

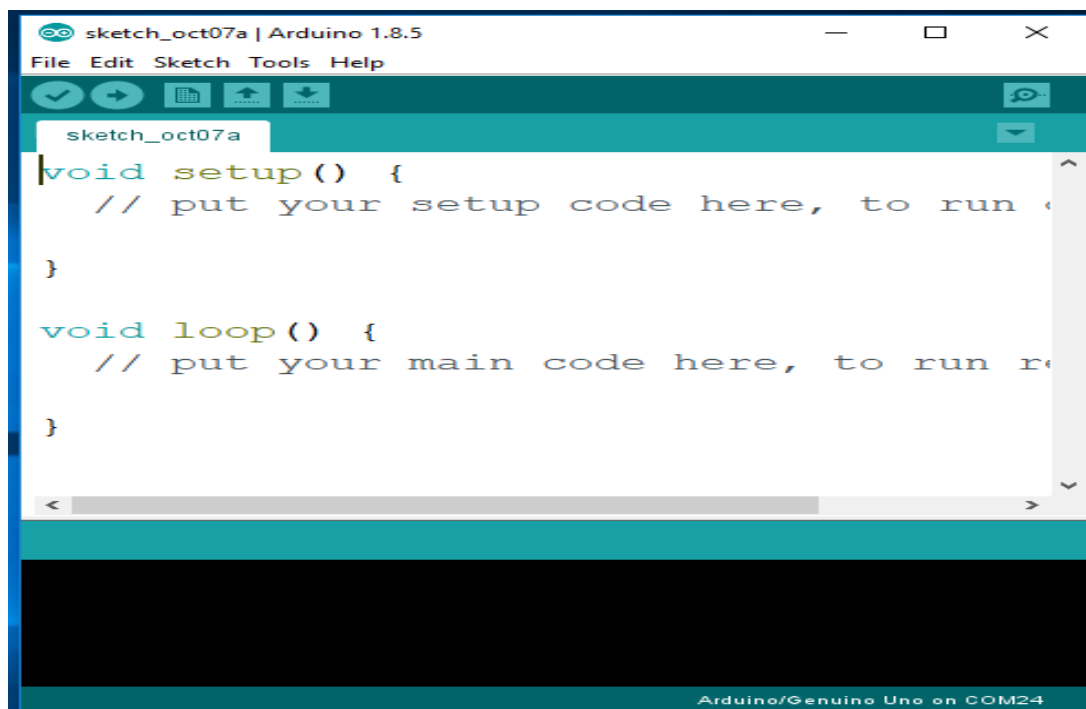


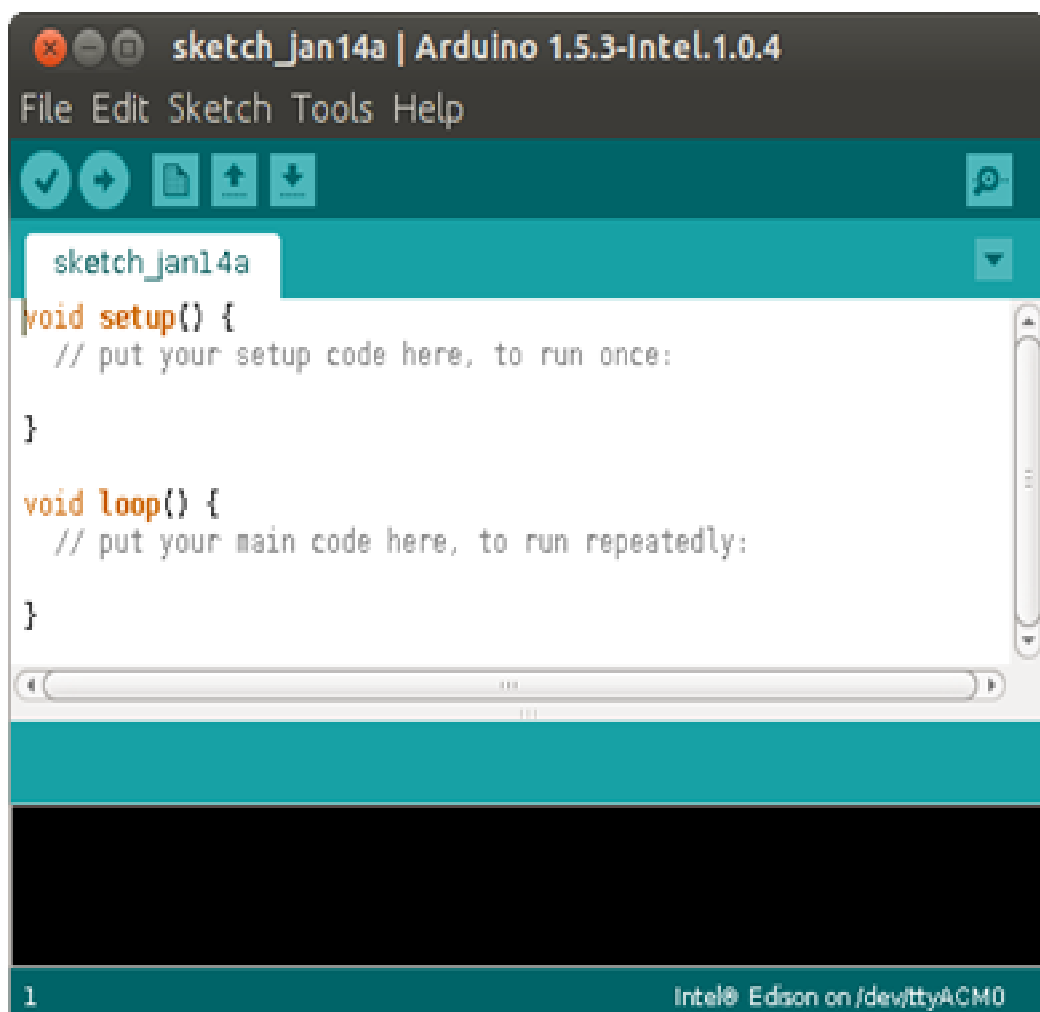
Fig. 4.1 Software Platform

It is conjointly capable of collecting and uploading programs to the board with a single click. There's usually no need to edit build files or run programs on a command-line interface. Although building on command-line is possible if needed with some third-party tools such as Ino.TheArduino IDE comes with a C/C++ library known as "Wiring" (from the project of the same name), that makes several common input/output operations much easier. Arduino programs are written in C/C++, although users only needed define two functions to make a runnable program

setup() – a function run once at the beginning of a program that can initialize settings

loop() – a function known as repeatedly till the board power off

**The compiled window of our code is shown below.**



```
sketch_jan14a | Arduino 1.5.3-Intel.1.0.4
File Edit Sketch Tools Help
sketch_jan14a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
1 Intel® Edison on /dev/ttyACM0
```

Fig. 4.2 Compiling window

### 4.3 FLOW CHART DIAGRAM

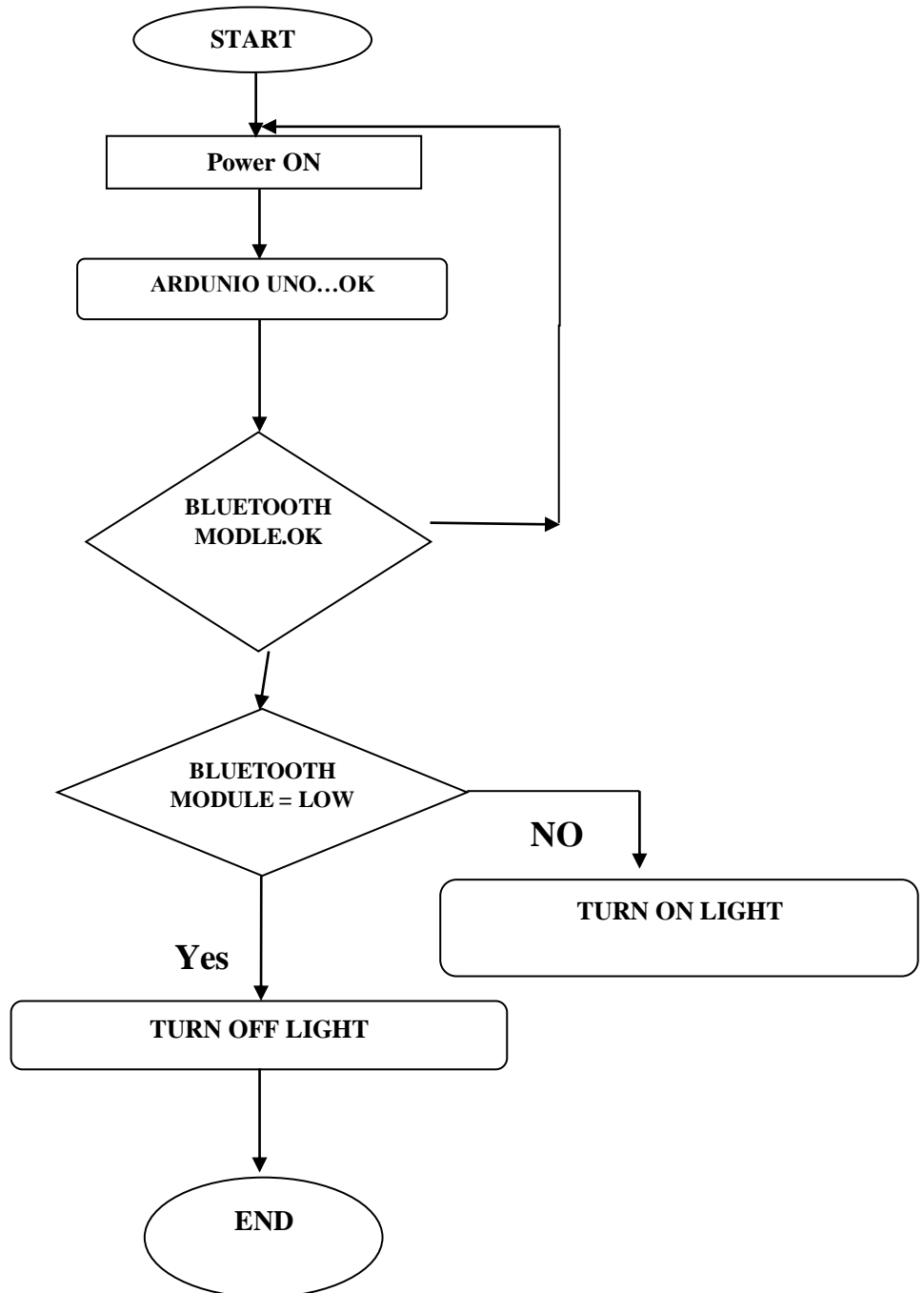


Fig. 4.3 Flow chart of our system

## **4.4CONCLUSION**

The system software system developed in Embedded C, C++ language that has the flexibility to receive the information from the sensing element and sending the information, and controls all the appliances that connected. As a result of its low power consumption, easy usage, reliability it's used in different fields. Software analysis is a very important a part of our system. A software analysis makes the sure sensible design. A proper software analysis and its burn into ArduinoUNO the project to a smooth end.

# CHAPTER 5

## HARDWARE IMPLEMENTATION

### 5.1 INTRODUCTION

This is one of the most important chapters of this report. In this chapter we will show our completed project outlook that means Connection law & operation representation of our project. We will discuss about component's interfacing with Arduino UNO of our system.

### 5.2 INTERFACING OF BLUETOOTH (HC-05)

Module HC-05 is connected with Arduino, by attaching its one Data pin D0 and D1 with respectively and Module HC-05 operates on 5 volts.

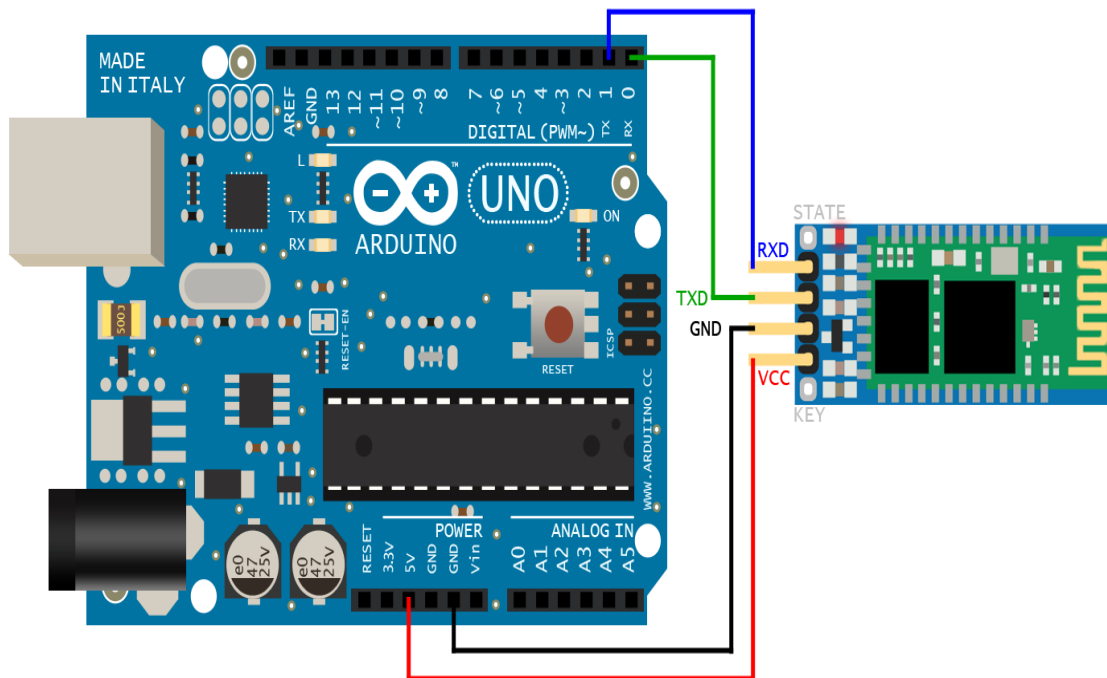


Fig. 5.1 Interfacing Of HC-05.

## 5.3 INTERFACING WITH APPLICATION

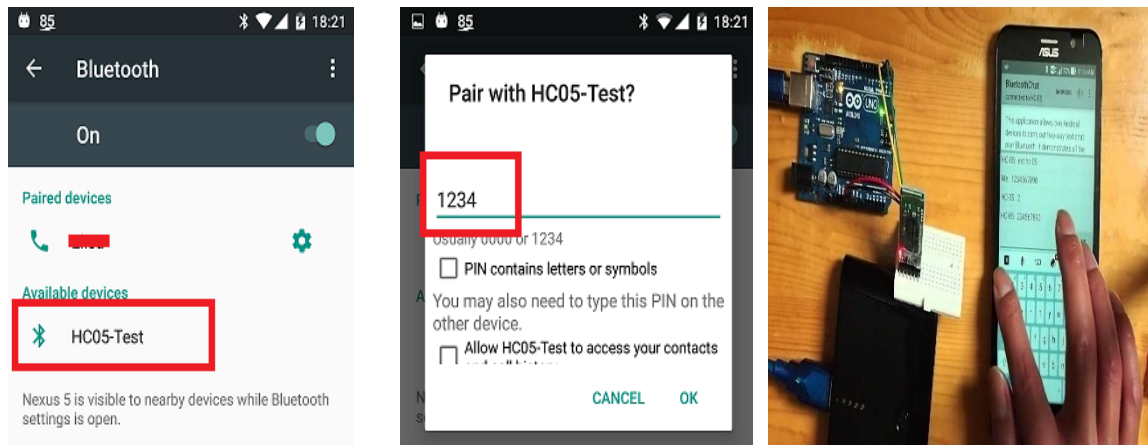


Fig 5.2(a) Interfacing with Application.

At first, we need to unlock your smartphone, then we need to turn ON our Bluetooth in the setting option. Then we need to connect our HC-05 with our smartphone. If any password or pin is asked for pair then we need to dial "1234" as like Fig 5.2(a). Then we need to open our Android application. Go to settings option and press connect robot as like Fig 5.2(a). Then we see our device is connected with our Android application. Now we are ready to use our device. If we want to connect a new smartphone then we need to disconnect our paired device and then we can connect our new device.

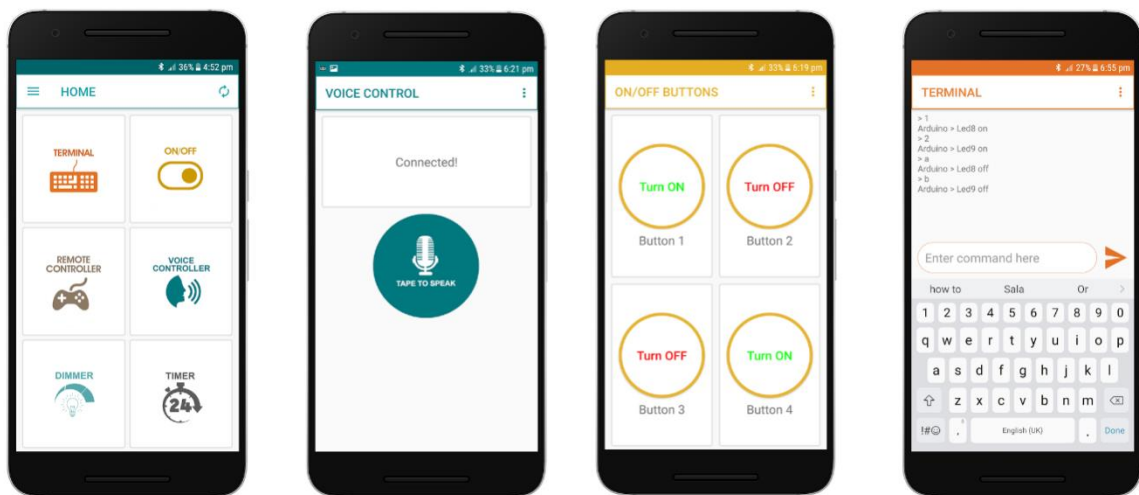


Fig 5.2(b) Interfacing of Application

After pairing we need to connect the Bluetooth with voice app, we need to give voice command to control the project, as example light (ON/OFF), TV (ON/OFF).

## 5.4 INTERFACING WITH RELAY MODULE

The below figure shows that how to connect a relay module with the ArduinoUNO

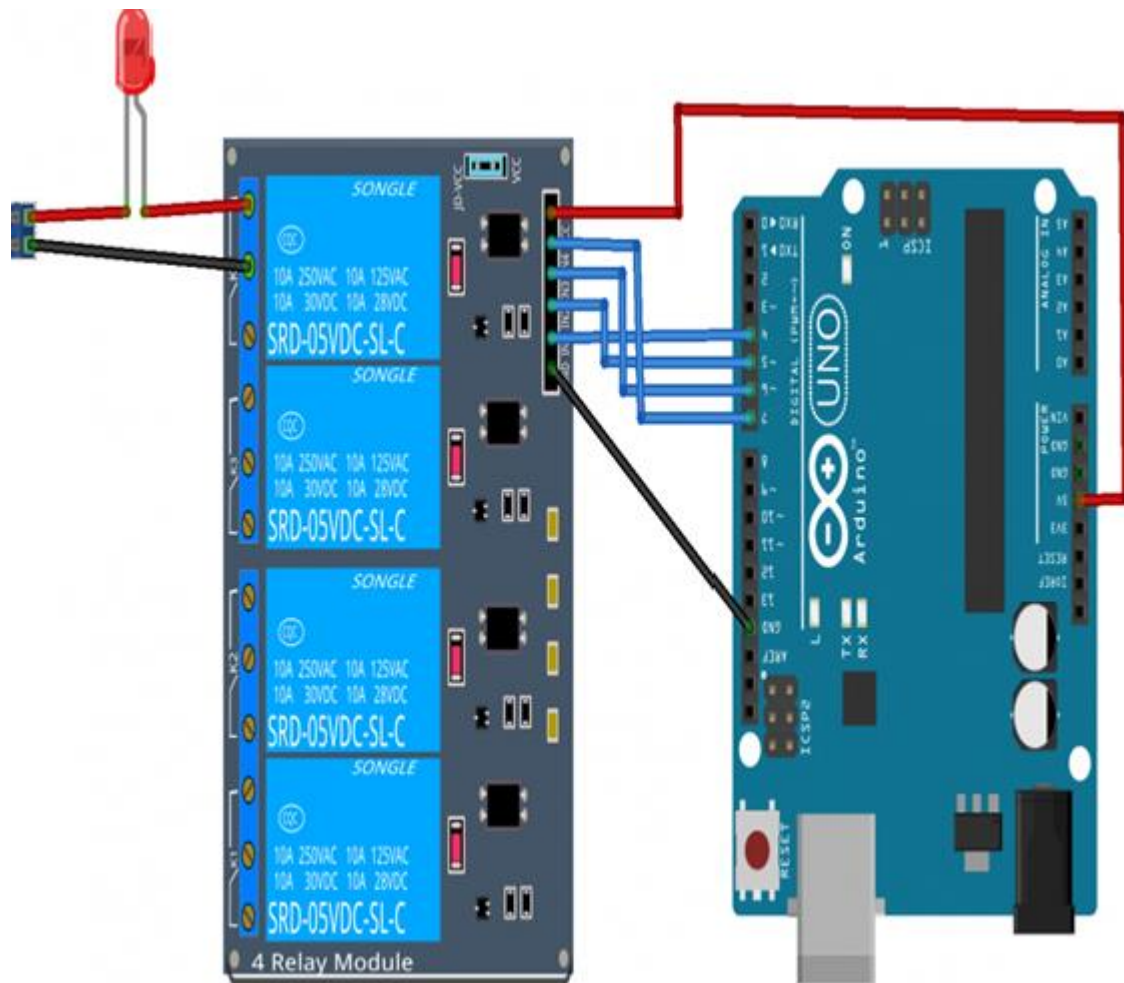


Fig 5.3 Interfacing with Relay.

Relay Module have 5pins. 1 pin is for deliver through the Arduino board and 4 other pins for 4 channel 5V relay module. In this project we are using four relays for switching the connected three devices. We are use only the four pin. That pins are the output of our project. As per code we use the four pins D4, D5, D6 and D7 in the Arduino Board.



## 5.5 LOADS INTERFACING

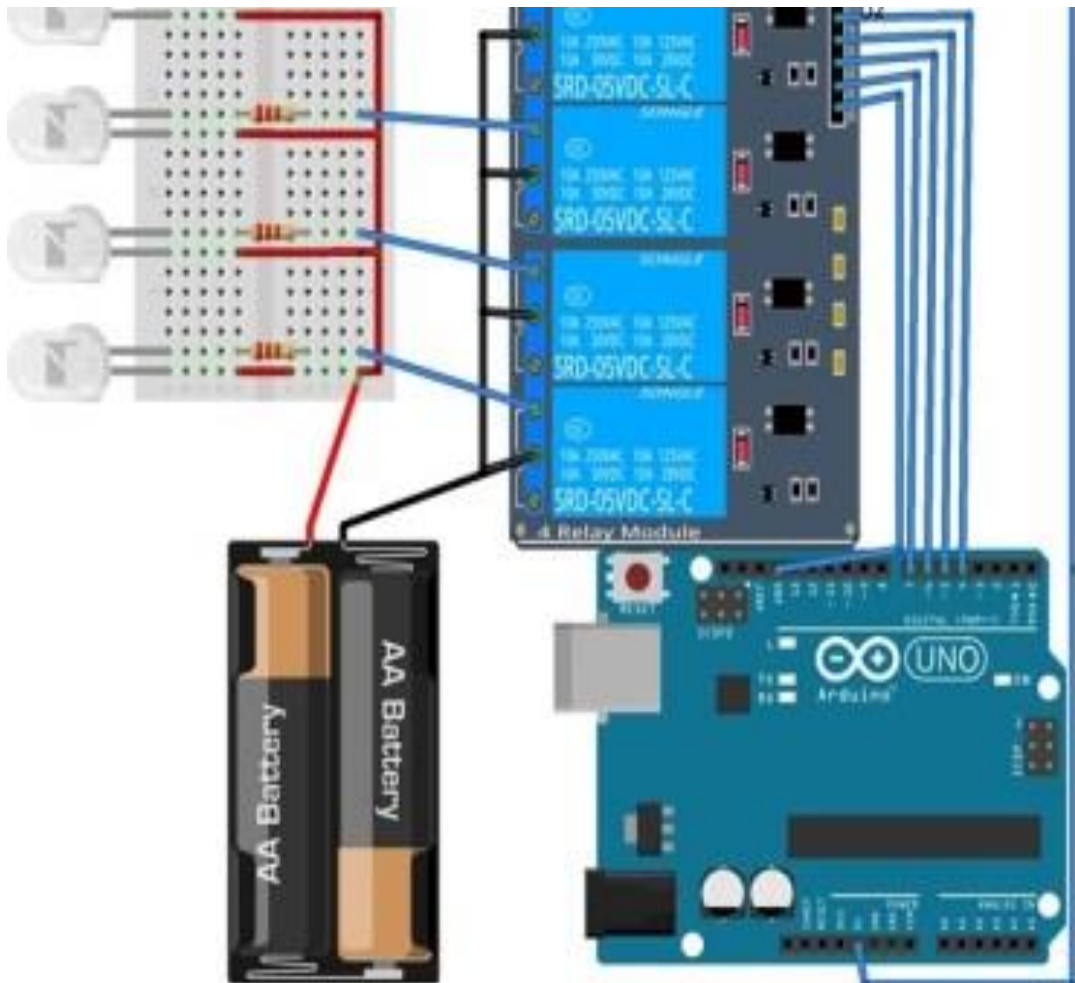


Fig 5.4 Interfacing with Loads.

The loads are connected to relay module. In this project we are using three loads as per example. There are three pins in each relay. NC (Normally Close), NO (Normally Open), and Common pin. All the common pins are going to line voltage (220v AC) negative side and all the neutrals of the loads go to positive side of line voltage. A wire is connected to NO pin to loads negative side.

## **5.6 SYSTEM OPERATION**

The base of the system is ATmega328. The Android App is acts as a remote. In the homework, we often use TV, AC, remotes. The apps are also like the remote. The inputs given to the microchip are the outputs of module HC-05. The module receives data from the app. The outputs of the controller are given to the relay module. The Output of Bluetooth module is given to the D0, D1 pin of ATmega328. When the microcontroller receives output from the HC-05, it gives a command to operate the relay module. The receiving and transmitter pins of Bluetooth module are connected to the TX and RX pins of ATmega328p respectively. For the protection circuitry exhaust fan is provided with a controller.

## **5.7 CONCLUSION**

To design, interfacing with Arduino UNO of the circuit is that the basic purpose of our project. Our Project work is already accomplished. The created circuit is incredibly working.

# CHAPTER 6

## RESULTS AND DISCUSSIONS

### 6.1 INTRODUCTION

The system projected consists of 4 major modules particularly, Bluetooth module, Relay module with the microcontroller, Android apps. It uses ATmega328 that is an 8-bit microcontroller. It finds wide application because of its features and low power. During this chapter, we are going to discuss the tests our project.

### 6.2 EXPERIMENTAL SETUP

ATmega328 is the base of the system. The inputs given to the microchip are the outputs of module HC-05. The output of the controller are given to the relay module. The Output of Bluetooth module is given to the D0, D1 pin of ATmega328.



Fig 6.1 Our Project

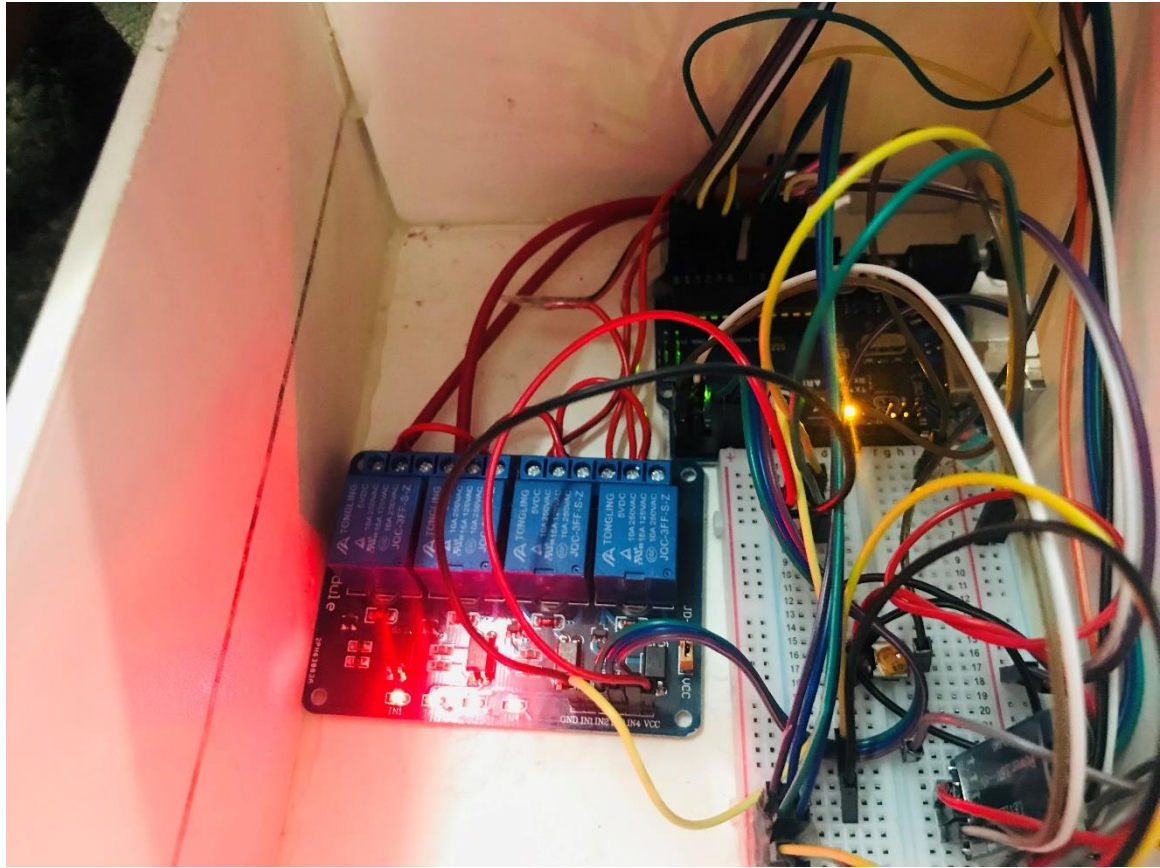


Fig 6.1.1 Setup of our system

We will connect our Bluetooth TX pin to RX pin, and RX pin to TX pin at ATmega328

### **6.3 RESULT**

Voice control home automation system is proposed and when the power is on, the relay switches the first light on and all are rest. Then we setup our communication with Android phone and given voice command to execute our controlled system by preset data. Overall system was designed and tested by introducing the home appliances controlling wirelessly. And the wireless system is possible by Bluetooth Module. Our project is completed a successful implementation & testing. It can be seen that output clearly. So we said that our system was effective fast response most secure and easy to construct.

## **6.4 ADVANTAGES**

- It is very easy to use.
- No additional coaching is needed for using it.
- All the control would be in your hands by using this voice control home automation system.
- This project will give the ability of observation all the appliances with within the communication limit through Bluetooth.
- By using this method the users will check the standing of the appliances at no matter time of the day.

## **6.5 LIMITATION**

- Bluetooth technology utilizes in this home automation system, which has a range ten to twenty meters so the control cannot be achieved from outside this compass.
- Application is connected after disconnecting of the Bluetooth.
- If we use voice technology oftentimes, we could experience some physical discomfort and vocal issues.

## **6.6 CONCLUSION**

We tested our system & it's worked properly. As beginning of the code section we use unit testing. When we made our modules we test it by unit testing. After complete development process of our full system, we test it by integration testing system. In every test section we found some problems and we solve those problems as soon as possible. After completing all process including testing we are assuring that it's ready for home and commercial use.

# CHAPTER 7

## CONCLUSION

### 7.1 CONCLUSION

This project is microcontroller based mostly project. A Voice Control Home Automation system is extremely effective during this time. During this project we wish execute the system to be used in varied places like home appliances, industrial automation etc. A Bluetooth module receives data through an Android apps like “Bro’s home automation”. The data of Bluetooth module delivers to the ArduinoUNO board. ArduinoUNO controls the switches of relay module. All the loads are connected to the relay. We have finally succeeded in creating the “Voice Control Home Automation” Satisfactorily. More of} information is gained and a lot of experiences are faced lot of information’s are collected ultimately, we've concluded with an excellent pleasure for achieving our aim. We’ve planned to satisfy my technical requirements. The knowledge we've earned with this project really would follow until the end of our career.

### 7.2 APPLICATIONS

- Applications are uses as remote control.
- This project used as security system in applications like homes, hostels, industries.
- We are used to be lazy and creative, the automation system works for us.
- For safety from main line plug-in, out.
- Children do not get any electric shock.
- Protection from all kind of human hazard.

### **7.3 LIMITATIONS OF WORK**

Bluetooth could be a short-range wireless data network originally planned as an alternate to the untidy tangle of add-on wires. The Bluetooth 4.0 Low Energy standard, at 26 megabits per second, is far quicker than Bluetooth used to supply and suitable for infrequent syncing and tiny backup operations. Though Bluetooth is associate energy-efficient technology, it will slowly drain the battery of your mobile phone or another mobile device. Once enabled, Bluetooth frequently scans for signals, searching for new devices to connect with, however using energy in the method.

### **7.4 FUTURE WORK**

This monitoring system will be more increased by using Bluetooth, which supports another period of time application. For industrial purposes, a mobile robot can be developed for automation. Within the future, all the devices that we tend to use on a daily basis that are connected to a microcontroller {and we tend to and that we} shall use them wirelessly not only Bluetooth control however conjointly we use approaching technologies. Within the coming back generation use of mechanical parts and advanced alarm, the method will be more useful for utilization.

# REFERENCES

- [1] <https://www.Arduino.cc>
- [2] [https://www.wetube.com/watch?v=BIf\\_mpnzZvY](https://www.wetube.com/watch?v=BIf_mpnzZvY)
- [3] <https://www.arrow.com/en/products/atmega328>
- [4] [https://en.wikipedia.org/wiki/AC\\_adapter](https://en.wikipedia.org/wiki/AC_adapter)
- [5] <http://www.computerhope.com/jargon/a/acadapte.htm>
- [6] <http://whatis.techtarget.com/definition/resistance>
- [7] <https://www.google.com.bd/#q=resistor+datasheet&start=10>
- [8] <https://www.google.com.bd/#q=2n2222-+transistor+picture>
- [9] <https://www.google.com.bd/#q=variable+resistor+10k+datasheet+pdf>
- [10] <http://www.electrical4u.com/variable-resistors/>
- [11] <https://www.Arduino.cc/en/Main/ArduinoBoardUNOSMD>
- [12] <http://www.learningaboutelectronics.com/Articles/Atmega328-pinout.php>
- [13] <https://en.wikipedia.org/wiki/Arduino>
- [14] <http://www.slideshare.net/AbhishekNeb/thesis-voice-control-home-automation>
- [15] [http://wiki.sgmk-ssam.ch/index.php/BabyGnusbuino\\_code\\_examples](http://wiki.sgmk-ssam.ch/index.php/BabyGnusbuino_code_examples)
- [16] <http://positrontech.in/eshop/product/hc-05-wireless-bluetooth/>
- [17] <https://www.elexp.com/PDFs/01ARDUNO.pdf>
- [18] [http://wiki.sunfounder.cc/index.php?title=4\\_Channel\\_5V\\_Relay\\_Module](http://wiki.sunfounder.cc/index.php?title=4_Channel_5V_Relay_Module)
- [19] <http://www.rroj.com>
- [20] <http://www.ermt.net>
- [21] <https://www.electronicshobby.com/2018/06/voice-activated-home-automation-by-using-arduino-iot-smart-home.html>



# APPENDIX A

Important Initialization & Setup-

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);
int light=6;
int fan=3;
int tv=4;
int ac=5;
int led=8;
String voice;
int i=0;
int pos;
void setup()
{
  Serial.begin(9600);
  pinMode(light, OUTPUT);
  pinMode(fan, OUTPUT);
  pinMode(tv, OUTPUT);
  pinMode(ac, OUTPUT);
  pinMode(led, OUTPUT);

  digitalWrite(light, LOW);
  digitalWrite(fan, LOW);
  digitalWrite(tv, LOW);
  digitalWrite(ac, LOW);

  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.print("VOICE ACTIVATED");
  lcd.setCursor(0,1);
  lcd.print("HOME AUTOMATION");
}

void loop()
{
```

```

while (Serial.available()){ //Check if there is an available byte to read
delay(10); //Delay added to make thing stable
char c = Serial.read(); //Conduct a serial read
if (c == '#') {break;} //Exit the loop when the # is detected after the word
voice += c; //Shorthand for voice = voice + c
}
if (voice.length() > 0){
if(voice=="*turn on light")
{
digitalWrite(light,HIGH);
Serial.println("light on");
lcd.clear();
lcd.print(" **LIGHT ON**");
}
else if(voice=="*turn off light")
{
digitalWrite(light,LOW);
Serial.println("light off");
lcd.clear();
lcd.print(" **LIGHT OFF**");
}
else if(voice=="*turn on fan")
{
digitalWrite(fan,HIGH);
Serial.println("fan on");
lcd.clear();
lcd.print(" **FAN ON**");
}
else if(voice=="*turn off fan")
{
digitalWrite(fan,LOW);
Serial.println("fan off");
lcd.clear();
lcd.print(" **FAN OFF**");
}
else if(voice=="*turn on TV")
{

```

```

digitalWrite(tv,HIGH);
Serial.println("tv on");
lcd.clear();
lcd.print("  **TV ON**");
    }
else if(voice=="*turn off TV")
    {
digitalWrite(tv,LOW);
Serial.println("tv off");
lcd.clear();
lcd.print("  **TV OFF**");
    }
else if(voice=="*turn on AC")
    {
digitalWrite(ac,HIGH);
Serial.println("ac on");
lcd.clear();
lcd.print("**AC RUNING**");
    }
else if(voice=="*turn off AC")
    {
digitalWrite(ac,LOW);
Serial.println("ac off");
lcd.clear();
lcd.print("**AC OFF**");
    }
else if(voice=="*turn on all")
    {
digitalWrite(ac,HIGH);
digitalWrite(light,HIGH);
digitalWrite(fan,HIGH);
digitalWrite(tv,HIGH);
Serial.println("all on");
lcd.clear();
lcd.print("  **ALL ON**");
    }
else if(voice=="*turn off all")

```

```
{  
digitalWrite(ac,LOW);  
digitalWrite(light,LOW);  
digitalWrite(fan,LOW);  
digitalWrite(tv,LOW);  
Serial.println("all off");  
lcd.clear();  
lcd.print(" **ALL OFF**");  
}  
digitalWrite(led,1);  
delay(100);  
digitalWrite(led,0);  
delay(100);  
}  
voice="";  
}
```