Voice Based Home Automation using Arduino

Submitted in partial fulfillment of the requirements

Of the degree of

Bachelor of Engineering in

ELECTRICAL

By

Mohd Hanif Tole	12EE61
Parvez Ahmad	13EE37
Ansari Mh Abuzar	15EE01
Siddiqui Arman Ahmed	15EE48

Supervisor

Prof. Ankur Upadhyay Sir

Department of Electrical Engineering

Anjuman-I-Islam's Kalsekar Technical Campus

2018-2019

CERTIFICATE

This is to certify that the project entitled "Voice Based Home Automation System" is the bonafide work carried out by **MR.M.HANIF TOLE, MR.PARVEZ AHMAD,MR.ANSARI MH ABUZAR,MR. SIDDIQUI ARMAN AHMED** student of ANJUMAN-I-ISLAM'S KALSEKAR TECHNICAL CAMPUS, during the Academic year 2018-2019, in partial fulfilment of the requirements for the award of the Degree in ELECTRICAL ENGINEERING and that the project has not formed the basis for the award previously of any degree, diploma, associate ship, fellowship or any other similar title.

---------- ----------

Supervisor **External**

---------- ----------

HOD **Director**

Project Report Approval for B. E

This project entitled "Voice Based Home Automation System" is done by **MR.M.HANIF TOLE, MR.PARVEZ AHMAD, MR.ANSARI MH ABUZAR, MR.SIDDIQUI ARMAN AHMED** approved for the Degree of Bachelor of Engineering in Electrical.

---------- ----------

Supervisor **External**

Director

Date Place

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature

Mohd Hanif Tole _____

Parvez Ahmad _____

Ansari Mh Abuzar _____

Siddiqui Arman Ahmed _____

Date:

ACKNOWLEDGEMENT

It is indeed a matter of great pleasure and proud privilege to be able to present this project on "**Voice Based Home Automation System using Arduino** ".

The completion of the project work is a millstone in student life and its execution is inevitable in the hands of guide. We are highly indebted the project guide **Prof. Ankur Upadhyay** for his invaluable guidance and appreciation for giving form and substance to this report. It is due to his enduring efforts; patience and enthusiasm, which has given a sense of direction and purposefulness to this project and ultimately made it a success.

We would like to tender our sincere thanks the staff members for their co-operation.

We would also like to express our deep regards and gratitude to the **Prof. Sayed Kaleem.**

We would wish to thank the non - teaching staff and our friends who have helped us all the time in one way or the other.

Really it is highly impossible to repay the debt of all the people who have directly or indirectly helped us for performing the project.

ABSTRACT

Now A Days home and building automation systems are used more and more. On the other hand, they provide increased comfort especially when employed in a private home .On the other hand, automation systems installed in commercial buildings do not only increase comfort ,but also allow centralized control of Security, Alarm Systems, Door lock, CCTV, Heating, Sensing, Air conditioning and lightning. Hence, they contribute to an overall cost reduction and also to energy saving which is certainly a main issue today.

Instead of PC based servers. Arduino based servers are becoming trend of today's market. Cost reduction is achieved using Arduino along with Ethernet module as Embedded Web Server. Idea is utilized for monitoring and controlling maximum no. of either home appliances or industry devices. Without using a computer, Ethernet module can communicate to the owner of the overall system, who is able to manage appliances from any location outside. This server provides a powerful networking solution and enables web access for automation and monitoring of different systems. For industry automation, instrumentation and household devices control, this is an optimized solution. System home page can be accessed using web browser. Operational status of the appliances can be observed and changed in case of necessity. This report proposes development of low cost system for above purpose. Different sensors installed at working place help in sensing real time environmental conditions like temperature, light, humidity etc.

LIST OF FIGURES

Figure no.	Figure Name	Page no
Fig. 1.1	Voice Based Home Automation System	1
Fig. 3.1.1	Android Based Home Automation	3
Fig. 3.1.2	Zigbee Based Voice Controlled Wireless Smart Home System	4
Fig. 4.1.1	Arduino Mega Board(328P)	9
Fig. 4.1.2	HC-05 Bluetooth Module	10
Fig. 4.1.3	Bluetooth Interfacing with Arduino	11
Fig. 4.1.5.1(a)	Relay Brick	16
Fig. 4.1.5.1(b)	Relay Contacts	17
Fig. 4.1.5.2	Relay Module Schematic	18
Fig. 4.1.5.3	Relay Module Layout	18
Fig. 5.3	Design and implementation of circuit	31

INDEX

Chapter No.	Title	Page No.
Chapter 1	**Introduction**	1
Chapter 2	**Objective of Project**	2
Chapter 3	**Review of Literature Survey**	
	3.1. Existing System.	
	3.1.1. Android Based Home Automation System……..	3
	3.1.2. Zigbee Based Voice Control Wireless Smart Home System	4
	3.2.Problem	5
	Description……..	5
	3.3.Proposed System……	5
	3.3.1.Voice Based Home Automation……...	6
	3.4.Potential Benefits……...	
Chapter 4	**Hardware And Software Specification**	
	4.1.Hardware Specification	
	4.1.1.Arduino Mega Board(328P)……	8
	4.1.2.HC-05 Bluetooth Module…….....	10
	4.1.3.Bluetooth interfacing with Arduino	11
	4.1.4.Power Supply……	12
	4.1.5.Relay Module……..	
	4.1.51.Relay Brick…….	16
	4.1.5.2.Relay Module Schematic……	18
	4.1.5.3.Relay Module Layout……	18
	4.1.5.4.PCB Etching Process……	19
	4.1.5.5.Soldering…….....	19

Chapter 5	**Software Specification** 5.2.2.Program …… 5.3.Working of Voice Home Automation…….	 20 31
Chapter 6	**Technical Specification**	32
Chapter 7	**Advantages,Drawbacks,Applications**	33
Chapter 8	**Future Scope**	34
Chapter 9	**Conclusion**	35
Chapter 8	**References**	36

Chapter 1

INTRODUCTION

Our project is based on Voice based Home Automation which uses Arduino as a processor to handle various web requests from web browser operated through a PC, mobile or tabs. A web page is created with basic security algorithm and it also contains user name password for authentication process. After getting access to the main page, we can see the various control switches, these control switches help us to switch on or of a particular appliance or device connected to the arduino control board. The web page also gives the status of the status of the appliance. After we send the command "ON" from the web page, the http request travel through the network or internet and reaches the arduino board. The arduino board has an Ethernet shield that has a designated IP and MAC address. The Ethernet shield helps the arduino to connect to a network or internet. The http request received by Ethernet shield and it is then forwarded to arduino mega board. The board process the request and perform the function accordingly. The function involves triggering the relay board to switch the desired appliances ON or OFF.

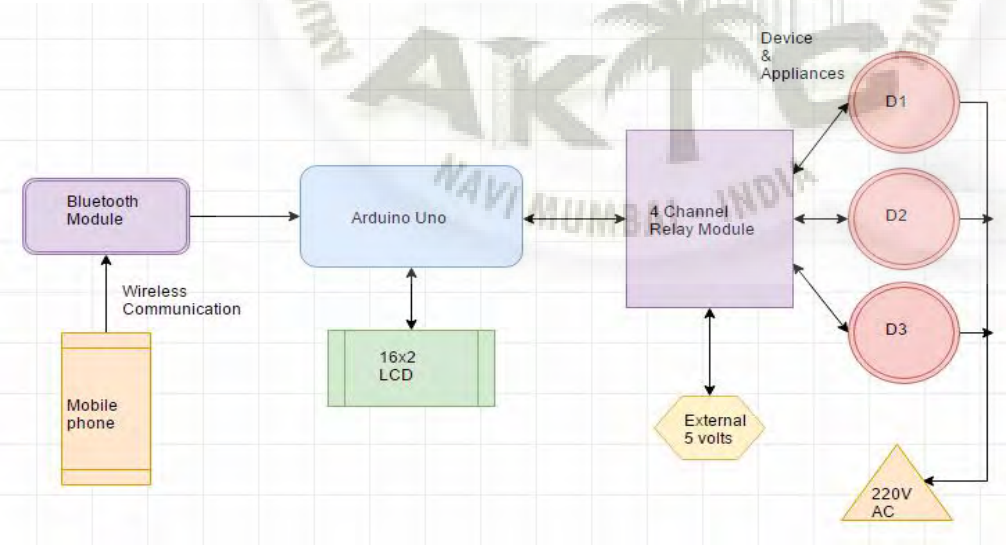

Fig 1.1 Voice based Home Automation

Chapter 2

OBJECTIVE OF THE PROJECT

1. The goal of this project is to develop a home automation system that gives the user complete control overall remotely controllable aspects of his or her home.
2. The automation system will have the ability to be controlled from a central host PC, the Internet, and also remotely accessed via a Pocket PC with a Windows Mobile based application.
3. The System will also sense the Accidental Gas leakage, water level and will notify the user by SMS.

Chapter 3

REVIEW OF LITERATURE SURVEY

3.1 Existing System

3.1.1 Android Based Home Automation

Generally in today's modern world human beings are addicted to using modern equipment. The intention of this project is to make an Android OS based smartphone or tablet workable for controlling each and every appliance of industries or household. There are several Android applications available in the market to turn our Android-based smart phone or tablet into a remote control for our home. If we want to control a system in home or want to get start, we need an INSTEON controller and also INSTEON controllable devices. In addition to this, we need to install different Android applications for home automation. Some of these Android applications are INSTEON Hub, MobiLinc, Conductor, wdISY, Touch Switch,etc.

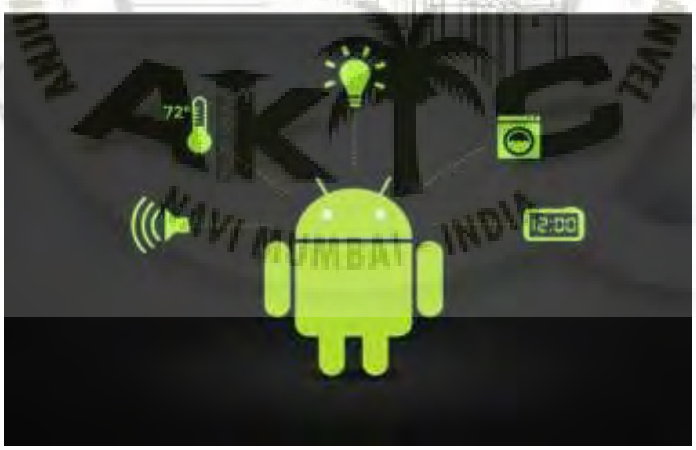

Fig 3.1.1 Android Based Home Automation

In recent years, technology is advancing and houses are becoming smarter by gradually shifting from conventional to centralized switches. Conventional switches are located in different places of the house. These switches are difficult to operate for the users and especially for physically handicapped people as they are difficult for them to approach and operate. Android technology

based remote controlled system provides a simple solution to home automation. In this project, a microcontroller is used from the 8051 family, and the loads are interfaced with the 8051 microcontroller by using TRIACs and Opto-Isolators. The remote operation is achieved through any Android smartphone with a Graphical User Interfaced based touch screen operation. GUI is nothing but a type of user interface that allows users to interact with the electronic devices through visual indicators and graphical icons. In order to achieve this, Android smartphone acts as a transmitter and sends on or off commands to the receiver where loads are connected. So, by using wireless technology, we can operate the remote switch on the transmitter and the loads can be turned on or off remotely.

3.1.2 Zigbee Based Voice Controlled Wireless Smart Home System

Fig 3.1.2 Zigbee Based Voice Controlled Wireless Smart Home System

In this project,a voice controlled wireless smart home system has been presented for elderly and disabled people. The proposed system has two main components namely (a) voice recognition system, and (b) wireless system. LabView software has been used to implement the voice recognition system. On the other hand, ZigBee wireless modules have been used to implement

the wireless system. The main goal of this system is to control home appliances by using voice commands. The proposed system can recognize the voice commands, convert them into the required data format, and send the data through the wireless transmitter. Based on the received data at the wireless receiver associated with the appliances desired switching operations are performed. The proposed system is a low cost and low power system because ZigBee is used. Additionally the proposed system needs to be trained of voice command only once. Then the system can recognize the voice commands independent of vocabulary size, noise, and speaker characteristics.

In this method, the user with any mobile speaks voice commands, the mobile application convert the voice command in to text and payload the command on GSM network via SMS. Below Table shows basic command which is appended in SMS.

Voice Commands	**SMS Commands**
Main console on	'MCONE'
Main console off	'MCOFFE'
Zone 1 appliances on	'Z1ONE'
Zone 1 appliances off	'Z1OFFE'
Zone 2 appliances on	'Z2ONE'
Zone 2 appliances off	'Z2OFFE'
Light zone 1 on	'LZ1ONE'
Light zone 1 off	'LZ1OFFE'
Fan zone 2 on	'FZ2ONE'
Fan zone 2 off	'FZ2OFFE'

On the receiver side these commands are received and transfer to the controller using Bluetooth medium which is further preceded.

3.2 Problem Description

Referring to the title "Home Automation System", the system that will be developed is because of the several problems that arise in our today's modem society. They are:

1. A system of home automation products can cost more than RM 1000 for end users. Although this technology is available in the local and international market, not all people can afford it.

2. The increasing number of housebreaking caused the anxiety to people to leave their house. We will feel inferiors every time going out for a vacation because of the incomplete security system surrounding the house. We cannot always rely on our neighbour to look after our house while we are away from home unless we are rich enough to hire a house security guard.

3.3 Proposed System

3.3.1 Voice Based Home Automation System

Instead of PC based servers. Arduino based servers are becoming trend of today's market. Cost reduction is achieved using Arduino along with Ethernet module as Embedded Web Server. Idea is utilized for monitoring and controlling maximum no. of either home appliances or industry devices. Without using a computer, Ethernet module can communicate to the owner of the overall system, who is able to manage appliances from any location outside. This server provides a powerful networking solution and enables web access for automation and monitoring of different systems.For industry automation, instrumentation and household devices control, this is an optimized solution. System home page can be accessed using web browser. Operational status of the appliances can be observed and changed in case of necessity. This report proposes development of low cost system for above purpose. Different sensors installed at working place help in sensing real time environmental conditions like temperature, light, humidity etc.

Web server hosts a web site and provides reliable services for any requesting client. . Such web servers are developed using general purpose computers. They use different kind of operating systems such as NT, Unix, Linux Windows etc. Fig Different Devices Connected For Automation Such systems apply typical client-server architecture where, the client accesses the server through the LAN router and the Internet. Client sends the request to the server. This

request is processed by the router to connect to the Internet. The web processes the request made and finally connects to the desired web server. Requested data is sent to the client. An embedded web server is a microcontroller including software and application code to monitor and control the systems. Microcontroller or Arduino is an integral part of an embedded network and create a way for easy controlled activities of any device from any remote location. Such servers designed using very low resource usage, are highly reliable, portable and secure systems.

3.4 Potential Benefits

The potential benefits we can gain from home automation are almost only limited by imagination and as such it would be infeasible to create a comprehensive list of them. The short list below exemplifies potential benefits in four areas of home automation. The examples are meant to spark the imagination.

Energy Savings Through user tracking both in- and outdoors, a home automation system would potentially be able to make sure that, for example, no unnecessary light or heat is turned on in individual rooms.

Convenience Through Web based access to the home automation system a forgetful user will potentially no longer have to worry about if the coffee machine was left on when he left for work. Simply go to a Web page, check it, and turn it off if necessary.

Security Tracking user locations can assist in automatic alarm system arming. Also, security cameras might be accessed from a vacation to check that the house is alright.

Home Entertainment When engaging in movie watching, the lights might be set to an appropriate dimming level. When listening to music, speakers might be changing from room to room for your listening pleasure throughout the house. Digital paintings on the wall might change according to persons currently occupying the room.

Chapter 4
HARDWARE AND SOFTWARE SPECIFICATION

4.1 Hardware Specification

The following are the hardware we are proposing to use in our project:

4.1.1 Arduino Uno with Atmega 328P microcontroller:

Overview

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 a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to- serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to- serial converter.

Some Technical Specification of Arduino Uno are:

Summary	
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Circuit Diagram:

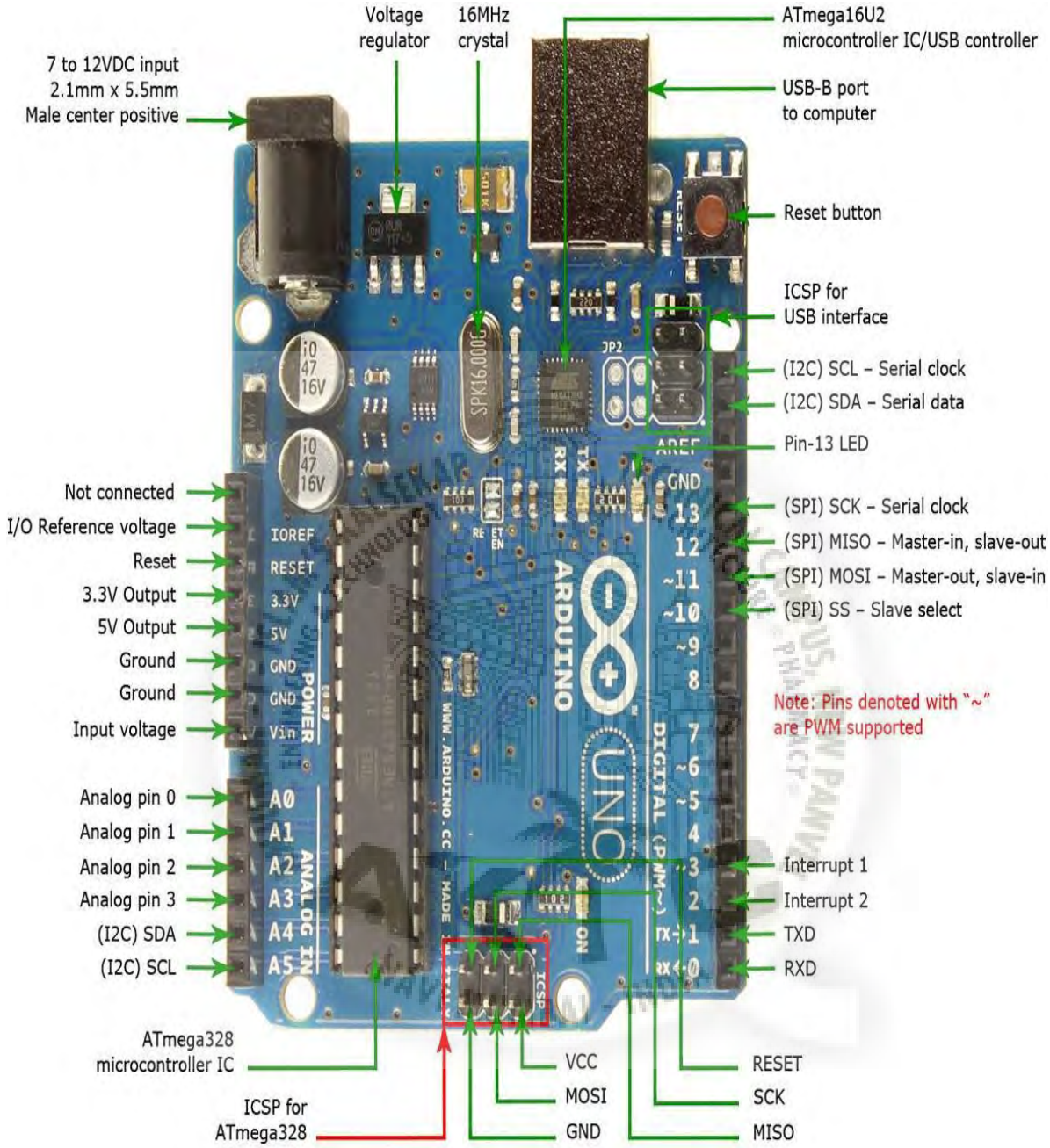

Fig.4.1.1 Arduino UNO with mega 328P

4.1.2 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 Bluecore 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 cycle.

Fig.4.1.2 Bluetooth Module

4.1.3 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.

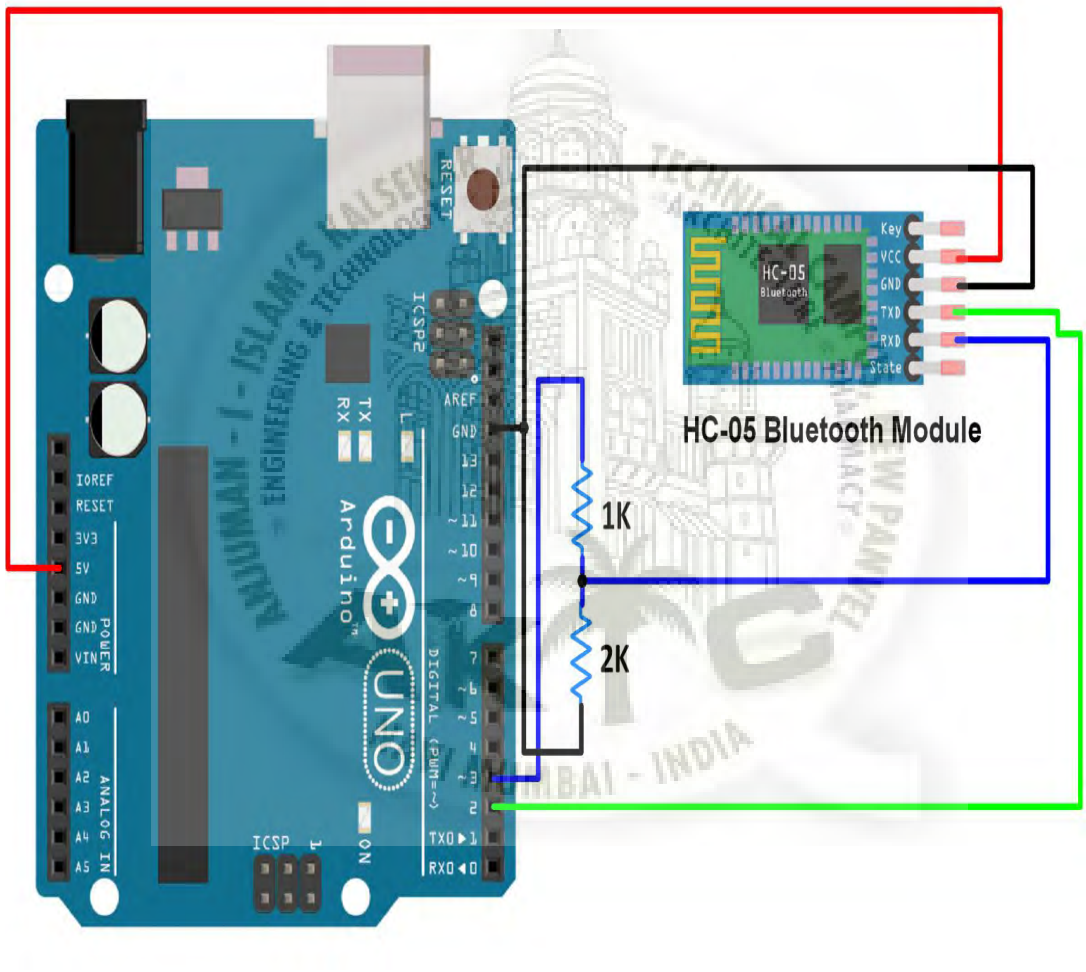

Fig.4.1.3 Bluetooth module interfacing with Arduino UNO

4.1.4 Power Supply:

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt()function for details.

PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analogWrite() function.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

LED: 13. 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.

TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega328 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2(ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .in f file, but OSX and Linux machines will

recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega328's digital pins. The ATmega328 also supports TWI and SPI communication. The Arduino software includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Mega can be programmed with the Arduino software (download). For details, see the reference andtutorials.

The ATmega328 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. TheATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Mega328 is designed in a way that allows it to be reset by software running on a connected

computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well- coordinated with the start of the upload.

This setup has other implications. When the Mega328 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega328. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega328 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Mega328 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega328 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations.

4.1.5 Relay Module

4.1.5.1 Relay Brick

Fig 4.1.5.1(a) Relay Brick

The Relay Brick uses some simple electronics principles that you can apply to other Output Devices for Arduino. Here's the fundamental principle:

"Electronic Devices using small amount of power, can Control a much larger amount of power"

The Relay Brick does this in two steps. Let's look at a diagram of what's on that little Brick:

First, notice the cable connector has a standard pattern of wires: Ground-Voltage-Signal (see the GVS labels on the Sensor Shield above). So the brick has +5 volt power available. Here's what happens: When the Blink Software Sketch does " digitalWrite(13, HIGH); " the Arduino connects the Signal wire to HIGH, which is +5 Volts.

+5 volts flow through 10,000 ohm resistor R1 to the Base of transistor Q1, and a current of about .0005 amps (500 microamps) flows and turns the transistor ON.

The transistor connects one end of the electromagnet inside the relay to Ground (the other end is already connected to 5 volts), and a current of about .07 amps flows. We say the transistor has a Current Gain of more than 100.

The electromagnet pulls the switch contact inside and moves it to connect the COM terminal to the NO (Normally Open) terminal.

A Lamp, or other load, is connected by the relay contacts. Let's say it might be a 100 watt lamp.

Details: the diode across the electromagnet conducts in the reverse direction when the transistor is turned off to protect against a voltage spike.

Details: An LED and it's current limiting resistor are also connected across the electromagnet and it lights up when the relay is turned on. Just for you...

This is the way high power can be controlled by a very small power from Arduino. Let's quickly think about the numbers:

Arduino outputs 5 Volts at .0005 amps. Multiply and that's .0025 watts (2.5 milliwatts) Not Much!

The transistor controls 5 Volts at .07 amps. That's, um, .35 watts.

The relay controls, say, 220 Volts at .5 amps. That's 110 watts.

So with this little brick, Arduino controls a power 4400 times it's own power. That's what this power control stuff is all about.

How Relay Contacts Work

Look at the photo of a relay down below on the right. Notice the 3 screw-type terminals. They are labelled "NO", "COM", "NC". Those labels mean:

Fig 4.1.3.1(B) Relay Contacts

NO: Normally Open

COM: Common Connection

NC: Normally Closed

Look at the diagram on the right. This shows the switch that is inside the relay. This switch is "thrown" by the electromagnet inside. The diagram shows that COM is connected to the Normally Closed contact. That's the case when the relay is off. When the relay is turned on the electromagnet flips the switch up and COM is then connected to Normally Open. So, if we want a lamp to be on when the relay is on, we connect our circuit from COM to NO.

4.1.5.2 Relay ModuleSchematic

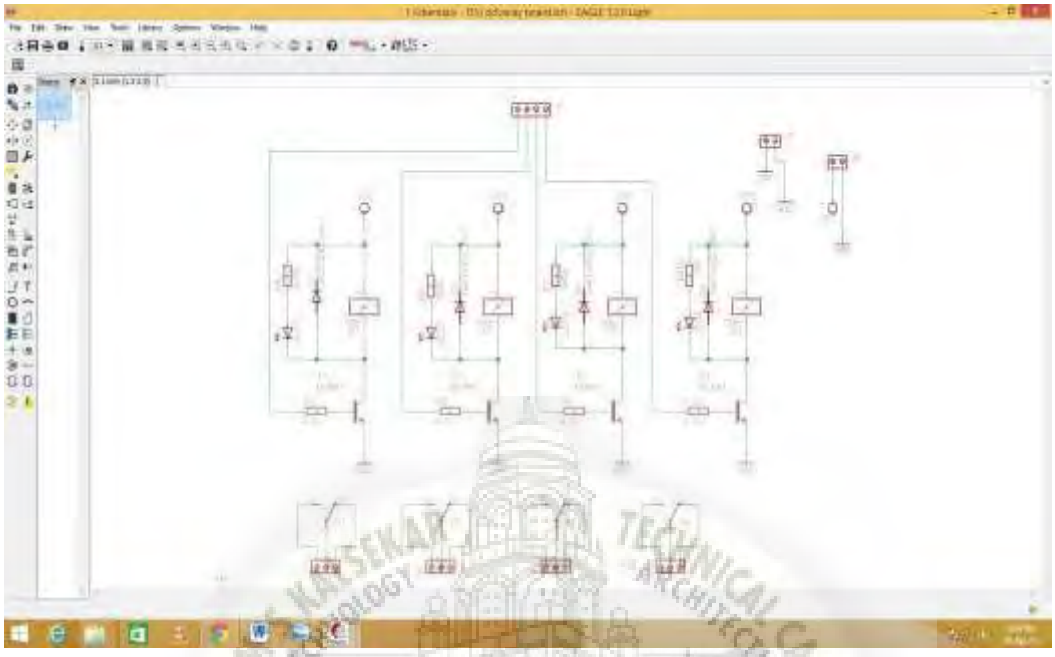

Fig 4.1.5.2 Relay Module Schematic

4.1.5.3 Relay ModuleLayout

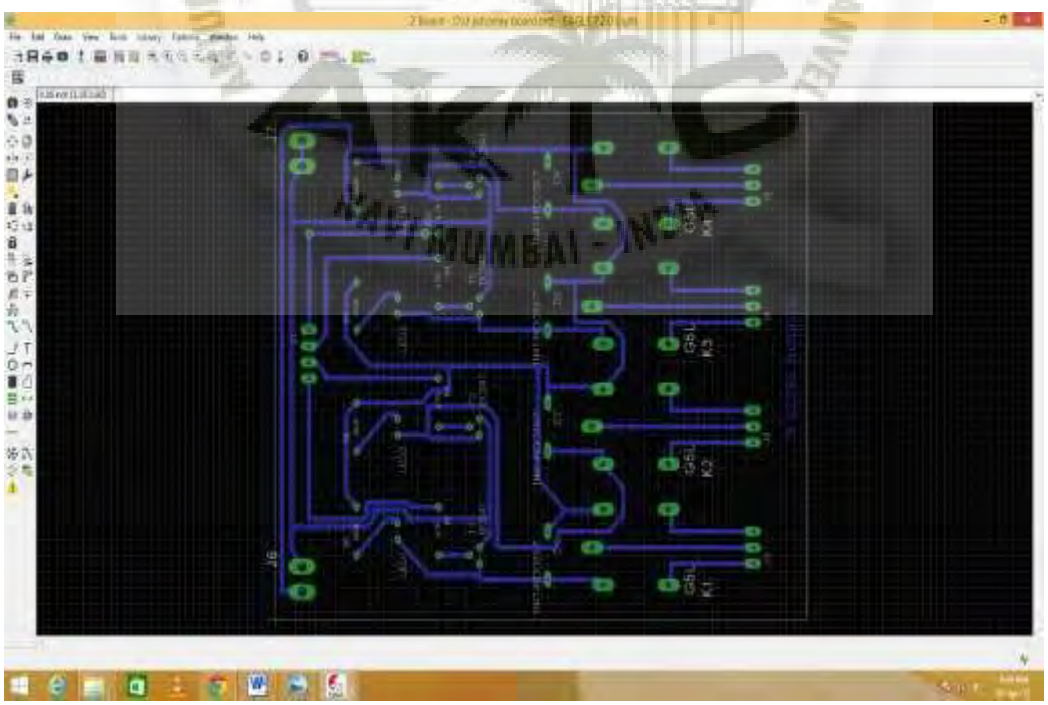

Fig 4.1.5.3 Relay Module Layout

Take a print of layout on photo paper or transparent sheet then put that sheet on copper clad and start ironig it.Ironing the printed layout transfers the ink from the paper going to the PCB board. You need to set your iron's temperature to the highest setting if your paper is thick but if not, set it to the medium setting.

4.1.5.4 PCB EtchingProcess:

All PCB's are made by bonding a layer of copper over the entire substrate, sometimes on both sides. Etching process has to be done to remove unnecessary copper after applying a temporary mask, leaving only the desired copper traces.

Though there are many methods available for etching, the most common method used by electronics hobbyists is etching using ferric chloride ir hydrochloric acid. Both are abundant and cheap. Dip the PCB inside the solution and keep it moving inside. Take it out at times and stop the process as soon as the copper layer has gone. After etching, rub the PCB with a little acetone to remove the black colour, thus giving the PCB a shining attractive look. The PCB layout is now complete.

3.1.5.5 Soldering:

- For soldering of any joints first the terminal to be soldered are cleaned to remove oxide film or dirt on it. If required flux is applied on the points to be soldered.
- Now the joint to be soldered is heated with the help of soldering iron. Heat applied should be such that when solder wire is touched to joint, it must melt quickly.
- The joint and the soldering iron are held such that molten solder should flow smoothly over the joint.
- When joint is completely covered with molten solder, the soldering iron is removed.
- The joint is allowed to cool, without any movement.
- The bright shining solder indicates good soldering.
- In case of dry solder joint, an air gap remains in between the solder maternal and the joint. It means that soldering is improper. This is removed and again soldering is done.
- Thus is this way all the components are soldered on P. C. B.

Chapter 5

SOFTWARE SPECIFICATION

5.2 Software Specification:

5.2.1 Sketch for Displaying Data Received via Bluetooth on Serial Monitor：

```
#include<SoftwareSerial.h>


/* Create object named bt of the class SoftwareSerial */
SoftwareSerial bt(2,3); /* (Rx,Tx) */


void setup() {
  bt.begin(9600);       /* Define baud rate for software serial communication */
  Serial.begin(9600);   /* Define baud rate for serial communication */
}


void loop() {


    if (bt.available()) /* If data is available on serial port */
    {
     Serial.write(bt.read());    /* Print character received on to the serial moni
tor */
    }
}
```

5.2.2 Arduino Development Environment

The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

5.2.2.1 Writing Sketches

Software written using Arduino are called sketches. These sketches are written in the text editor. Sketches are saved with the file extension .ino. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino environment including complete error messages and other information. The bottom righthand corner of the window displays the current board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the IDE prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Verify

Checks your code for errors.

Upload

Compiles your code and uploads it to the Arduino I/O board. See uploading below for details.

Note: If you are using an external programmer, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within

the current window.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.

Save

Saves your sketch.

Serial

Monitor

Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive which means only those items relevant to the work currently being carried out are available.

Edit

Copy for Forum

Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

Copy as HTML

Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

Sketch

Verify/Compile

Checks your sketch for errors.

Show Sketch Folder

Opens the current sketch folder.

Add File...

Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu.

Import Library

Adds a library to your sketch by inserting #include statements at the code of your code.

Tools

Auto Format

This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements instead curly braces are indented more.

Archive Sketch

Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

Board

Select the board that you're using.

Serial Port

This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

Programmer

For selecting a harware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.

Burn Bootloader

The items in this menu allow you to burn a boot loader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader.

Sketchbook

The Arduino environment uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino

software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

"'Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You 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 you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Serial Portmenus. The boards are described below. On the Mac, the serial port is probably something like/dev/tty.usbmodem241 (for an 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, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyUSB0, /dev/ttyUSB1 or similar.

Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the File menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino environment will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it

starts whichever sketch was most recently uploaded to the microcontroller. The bootloader 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 #includestatements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your 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 your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources. To install these third-party libraries, create a directory called libraries within your sketchbook directory. Then unzip the library there. For example, to install the DateTime library, its files should be in the /libraries/DateTime sub-folder of your sketchbook folder.

5.2.2.2 Program for Arduino:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(8,9,10,11,12,13);

#define white 3

#define blue 4

#define green 5

int tx=1;

int rx=0;

char inSerial[15];

void setup(){

Serial.begin(9600);

pinMode(white, OUTPUT);

pinMode(blue, OUTPUT);

pinMode(green, OUTPUT);

pinMode(tx, OUTPUT);

pinMode(rx, INPUT);

digitalWrite(white, HIGH);

digitalWrite(blue, HIGH);

digitalWrite(green, HIGH);

lcd.begin(16, 2);

lcd.clear();
```

```
lcd.print("MICROCONTROLLERS ");

lcd.setCursor(0,1);

lcd.print(" LAB ");

delay(2000);

lcd.clear();

lcd.print("VOICE CONTROL");

lcd.setCursor(0,1);

lcd.print("HOME AUTOMATION ");

delay(2000);

lcd.clear();

lcd.print("1. Bulb 1 WHITE");

lcd.setCursor(0,1);

lcd.print("2. Bulb 2 BLUE");

delay(2000);

lcd.clear();

lcd.print("3. Bulb 3 GREEN");

delay(2000);

lcd.clear();

lcd.print("Bulb 1 2 3 ");

lcd.setCursor(0,1);

lcd.print(" OFF OFF OFF");}

void loop(){
```

```
int i=0;

int m=0;

delay(500);

if (Serial.available() > 0) {

while (Serial.available() > 0) {

inSerial[i]=Serial.read();

i++;

}

inSerial[i]='\0';

Check_Protocol(inSerial);

}}

void Check_Protocol(char inStr[]){

int i=0;

int m=0;

Serial.println(inStr);

if(!strcmp(inStr,"*white on#")){

digitalWrite(white, LOW);

Serial.println("White ON");

lcd.setCursor(4,1);

lcd.print("ON ");

for(m=0;m<11;m++){

inStr[m]=0;}
```

```
i=0;}

if(!strcmp(inStr,"*white off#")){

digitalWrite(white, HIGH);

Serial.println("White OFF");

lcd.setCursor(4,1);

lcd.print("OFF ");

for(m=0;m<11;m++){

inStr[m]=0;}

i=0;}

if(!strcmp(inStr,"*blue on#")){

digitalWrite(blue, LOW);

Serial.println("Blue ON");

lcd.setCursor(8,1);

lcd.print("ON ");

for(m=0;m<11;m++){

inStr[m]=0;}

i=0;}

if(!strcmp(inStr,"*blue of#")){

digitalWrite(blue, HIGH);

Serial.println("Blue OFF");

lcd.setCursor(8,1);

lcd.print("OFF ");
```

```
for(m=0;m<11;m++){

inStr[m]=0;}

i=0;}

if(!strcmp(inStr,"*green on#")){

digitalWrite(green, LOW);

Serial.println("Green ON");

lcd.setCursor(12,1);

lcd.print("ON ");

for(m=0;m<11;m++){

inStr[m]=0;}

i=0;}

if(!strcmp(inStr,"*green of#")){

digitalWrite(green, HIGH);

Serial.println("Green OFF");

lcd.setCursor(12,1);

lcd.print("OFF ");

for(m=0;m<11;m++){

inStr[m]=0;}

i=0;}

else{

for(m=0;m<11;m++){

inStr[m]=0;} i=0; }}
```

5.3 Working of Voice control Home Automation:

We speak the predefined commands to AMR_Voice application. The application sends the command to Bluetooth which is then received by Arduino and perform the described task. At the same time, Arduino displays the status on LCD and write on the serial monitor. Each command has its unique operations which are defined in code. You can change the commands according to your ease. Below is the list of commands.

Commands: Following commands should be spoken by the user to turn on and turn off devices.

Command send by mobile	Message display on monitor
white on	White ON
white shutdown	White OFF
blue on	Blue ON
blue shutdown	Blue OFF
green on	Green ON
green shutdown	Green OFF

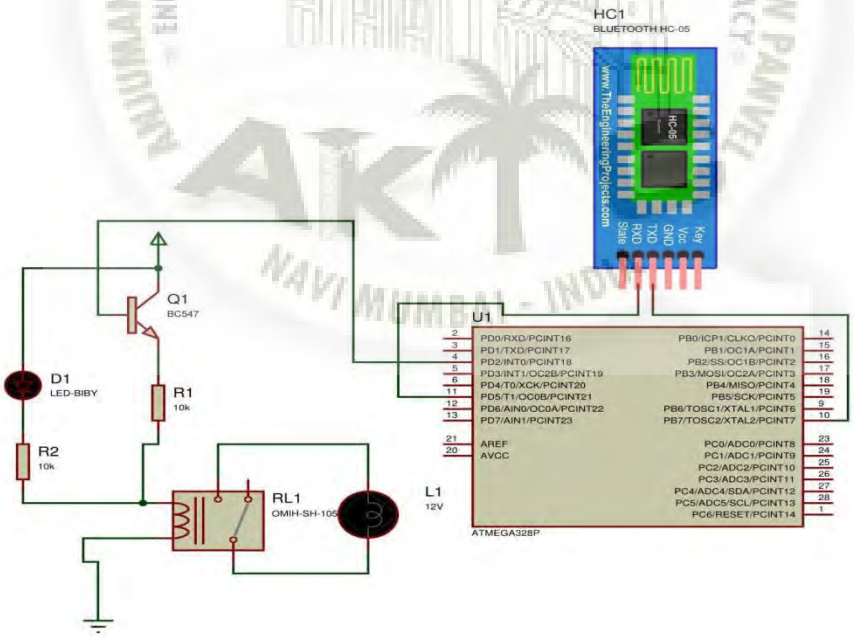

Fig.5.3 Design and implementation of the circuit

Chapter 6

TECHNICAL SPECIFICATION

1) A smartphone or an Android mobile which should have the android app installed in it.

2) Bluetooth receiver module – Our project will be connected to the smartphone using Bluetooth technology.

3) Controller or the main processing circuit- In this project, Arduino Uno is the main controlling / processing unit. Also, this project can be developed using PIC18F4550, AVR ATmega32 and 8051 series like: 89s51, 89c51, 89s52, 89v51RD2.

4) LCD Display – The Liquid Crystal Display is optional but shows important messages like device status once command is received from Bluetooth.

5) Relays to control devices – We have used 12volt Single push single throw relays.

6) Output devices – For the demo purpose, we connected a DC devices to a relay (12 volt DC bulb). You can connect any AC/DC devices to the remaining 3 relays. Also, Device 6 is a Buzzer.

Chapter 7

ADVANTAGES, DRAWBACKS AND APPLICATIONS

5.1 Advantages

- Energy saving.
- Security management.
- Home entertainment.
- Convenience Access through Internet.
- Adds Safety Through Appliance and Lighting Control.
- Secures Home Through Automated Door Locks.
- Increases Awareness Through Security Cameras.
- Increases Convenience Through Temperature Adjustment.

5.2 Drawbacks

- Microcontroller is necessary.
- User must be acquainted with internet services.

5.3 Applications

- Quickly and easily open and close curtains.
- Cameras to monitor your home to See who is at the front gate or front door.
- Heating and cooling.
- Adjust the lighting scenes yourself without calling the integrator for help.
- We Can turn ON & OFF any appliance from anywhere using android application.

Chapter 8

FUTURE SCOPES

- Doors & Gates can be controlled using proper mechanism.
- Using infrared light we can control Air Conditioner.
- Using CCTV camera we can monitor indoor & outdoor areas surround house.
- Using android app of arduino we can turn ON & OFF any appliance during any selected hours.
- Using image recognition,finger print,retina scan etc methods only authorized person is permitted to enter the house.
- Home automation and Device controlling can be done using Internet of Things – IOT technology.
- We can replace Bluetooth by GSM modem so that we can achieve device controlling by sending SMS using GSM modem.

Chapter 9

CONCLUSION

Home automation is not a new industry anymore, but it is still an emerging industry in developing countries. The huge potential market leads many electronics corporation into home automation. Sony, Siemens and even Apple company are trying to share a market of home automation. But the home automation industry scale hasn't formed yet. The good and bad mixed products are together in the market. There are also no unified industry standards. It is the challenge of home automation.

When compared with today's internet world, we can find that home automation has very big chance in the future. As, now a days every one uses internet it became very easy to work on product or device which can control by internet.

If the price of home automation is decreased or the practical applicability increased, there will b e more people willing to buy it. So in our project we tried to achieve this purpose. As we use open source software the cost of product is low. Using open source software also means you are not locked in to using a particular vendor's system that only work with their other systems. It continually evolving in real time as developers add to it and modify it, which means it can be better quality and more secure and less prone to bugs than proprietary systems.

Home Automation is undeniably a resource which can make a home environment automated. People can control their electrical devices via this Home Automation product and set up the controlling actions in the computer or smart phones with simple internet facility. We think this product have high potential for marketing in the future.

At last, we have to say home automation is like a rising sun. One day it will be like a burning sun bringing more decent, enjoyable and efficient life to people.

Chapter 10

REFERENCES

(1) Jyotsna A Nanajkar, Vismita D Nagrale "**Voice Based Automation**" International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 9, March 2013

.

(2) S.A.N.Sandeep, P.Malyadri "**Embedded Web Server Based on DAC System Using ARM**" International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 , Vol. 2, Issue 4, July-August 2012.

(3) Rajeev Piyare "**Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone**" Department of Information Electronics Engineering, Mokpo National University, Mokpo, 534-729, Korea South. International Journal of Internet of Things 2013, 2(1): 5-11 DOI: 10.5923/j.ijit.20130201.02

(4) Ahmed ElShafee, Karim Alaa Hamed "**Design and Implementation of a WiFi Based Home Automation System**" World Academy of Science, Engineering and Technology 68 2012.

(5) Malik Sikandar Hayat Khiyal, Aihab Khan, and Erum Shehzadi, "**SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security**", Issues in Informing Science and Information Technology Volume 6, 2009

(6) Wikipedia. (2012, 12th December). **Home automation. Available**: http://en.wikipedia.org/wiki/Home_automation

(7) Faisal Baig, Saira Beg and Muhammad Fahad Khan "**Zigbee Based Home Appliances Controlling Through Spoken Commands Using Handheld Devices** " International Journal of Smart Home Vol. 7, No. 1, January, 2013

(8) Thoraya Obaid, Haliemah Rashed, Ali Abu El Nour, Muhammad Rehan, Mussab Muhammad Saleh, and Mohammed Tarique "**Zigbee Based Voice Controlled Wireless Smart Home System**" International Journal of Wireless & Mobile Networks (IJWMN) Vol. 6, No. 1, February 2014.