TITLE ARDUINO BASED HOME AUTOMATION USING BLUETOOTH

A PROJECT REPORT

Submitted by

NAME OF THE CANDIDATE: - Amritpal Kaur Dhillon

in partial fulfillment for the award of the degree of

NAME OF THE DEGREE: - Bachelor of Computer Science and Engineering Specialization in Information Security AIT-APEX

IN

BRANCH OF STUDY: - Bachelor of Computer Science and Engineering Specialization in Information Security AIT-APEX

Chandigarh University

MONTH: - APRIL & YEAR: - 2024

TITLE OF PROJECT REPORT

A PROJECT REPORT

Submitted by

Amritpal Kaur Dhillon (UID: - 22BIS80001)

in partial fulfillment for the award of the degree of

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING SPECIALIZATION IN INFORMATION SECURITY AIT-APEX

IN

COMPUTER SCIENCE AND ENGINEERING SPECIALIZATION IN INFORMATION SECURITY AIT-APEX

Chandigarh University

APRIL 2024

BONAFIDE CERTIFICATE

Certified that this project report **TITLE OF THE PROJECT: - ARDUINO BASED HOME AUTOMATION SYSTEM USING BLUETOOTH”** is the bonafide work of **NAME OF THE CANDIDATE: - Amritpal Kaur Dhillon”** who carried out the project work under my/our supervision.

<<Signature of the HoD>> **SIGNATURE**	<<Signature of the Supervisor>> **SIGNATURE**
<<Name: **-** Aman Kaushik Sir>	<<Name **SUPERVISOR: -**
RAVNEET KAUR MAM	
HEAD OF THE DEPARTMENT	<<Academic Designation>>
<<Department: **-** >>	<<Department>>

Submitted for the project viva voce examination held on

Project report

On

Home Appliances Controlling using Android Mobile via Bluetooth

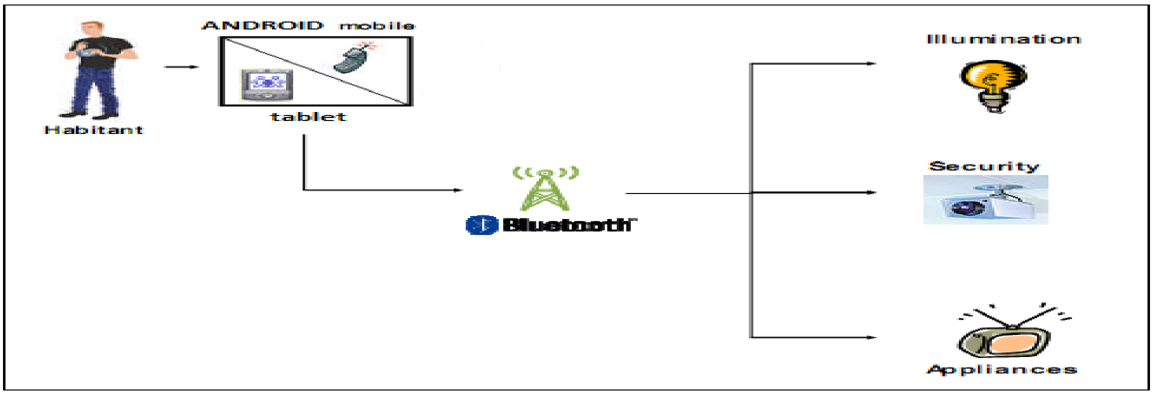
INDEX

1. INTRODUCTION

In the fast-moving world, time is the most important factor. You can turn on/off devices in the industry by giving commands through a cell phone. It will save time as well as manpower required to control industrial devices. By using key commands on android mobile, we can control six devices. We can increase the number of devices. Most important factor about this project is that it is controlled using an application on android mobile. The person who has installed this application on his/her android mobile can only interfere in the controlling devices. Also, it removes the need of carrying a remote control to turn on/off the devices.

This project has the integration of Android mobile technology and embedded systems. Android mobile user has to install an application on his/her mobile handset to control the devices. Then he/she can give commands using the buttons on that application. You have to turn on the Bluetooth on mobile, so the main wireless controlling technique used in this project is Bluetooth technology. Bluetooth receiver will be connected to the project. This Bluetooth device is connected to the circuit which has a decoder. It sends out a code for the respective command sent by user. Then the respective device connected to the circuit will be turned on or off depending on the command given. For example, turn on the Fan, Turn off the Fan. Turn on the buzzer etc. Such that by giving commands from mobile you can control industrial work

This is more advantages when we have to turn on the machinery at the time when we have another urgent task to do and we cannot get up from our place. In this case, we can turn on machinery by giving simply command through mobile phone. There is no need to go to the field.



1.1 LITERATURE SURVEY

We have undergone a detailed study of the Android mobile and various applications and implementation of those apps. As a part of our circular activity, we are making the project whose title is “Home Appliances Controlling using Android Mobile via Bluetooth”. We have used Arduino Uno microcontroller which is Arduino series Microcontroller as a main component of the project. Nowadays Microcontroller has become a main component of many of the electronic circuits.

USES: -

Health Care Industry-

1. Patient-centered outcome assessment/diagnostic instruments.
2. Functional status and satisfaction screening.
3. Clinical research surveys.
4. Appointment with doctors.

Call center -

1. Call center employee.
2. Customer satisfaction surveys.
3. Client survey/questionnaire administration.
4. Panel data collection.

HRD -

1. Personal screening and recruiting.
2. Employee satisfaction.
3. Benefits administration and information
4. Daily attendance.
5. Payroll-automated field time card.

2. BLOCK DIAGRAM

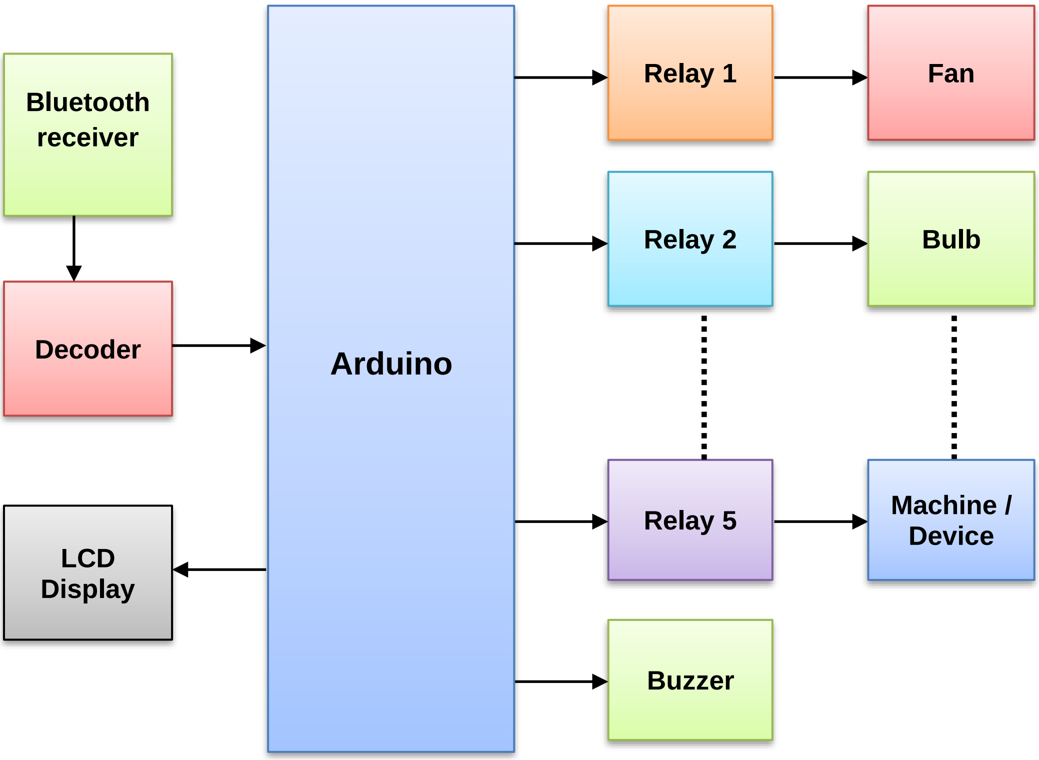


Fig 2.1 block diagram of a system

Transmitter / Controlling Unit

Bluetooth enabled android mobile

BLOCK DIAGRAM DESCRIPTION:-

1) Arduino

This is the most important segment of the project, i.e. the Arduino. The Arduino is responsible for the detection and polling of the peripheral's status. It is responsible for making decisions for the connected devices. It is responsible for prioritizing all the tasks.

We have used the Arduino Uno. It is a High Performance, Low Power AVR 8-Bit Arduino. It also has In-System Programming by an On-chip Boot Program. It has 23 Programmable I/O Lines.

It is the major part of the system that controls all the operation of the circuit such as LCD interfacing, WIFI module interaction. It also decides the messages to be displayed on the LCD along with the time duration for which they should be displayed on the LCD.

2) DISPLAY: -

It is used to display the current values of the measuring quantities. It can be used to display the various options and all the readings that have been stored in the EEPROM. LCD or 7-segment LED display can be used. Here the LCD used is the 16×2-line LCD. We can also use 16×4-line LCD. Liquid Crystal Display which is commonly known as LCD is an Alphanumeric Display it means that it can display Alphabets, Numbers as well as special symbols thus LCD is a user-friendly Display device which can be used for displaying various messages unlike seven segment display which can display only numbers and some of the alphabets. The only disadvantage of LCD over seven segment is that seven segment is robust display and be visualized from a longer distance as compared to LCD. Here I have used 16 x 2 Alphanumeric

Display which means on this display I can display two lines with maximum of 16 characters in one line.

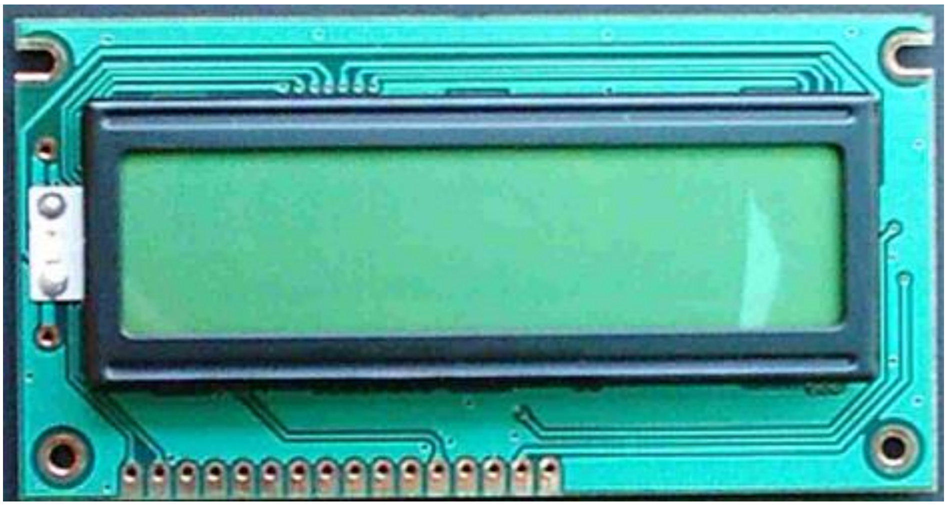


Figure: Photograph of A typical 16 by 2 alphanumeric LCD

3) Mobile: -It consists of an android mobile handset having Bluetooth. Also, user needs to install an application on this mobile. This android mobile will send commands using Bluetooth technology.

4) Bluetooth decoder: - It consists of a Bluetooth decoder. It gives ASCII code output. This receiver enables wireless transmission& reception of serial data. It has 10 meters range.

5) Relay: -Relay is used to control the on/off operation of the device. Relays are driven by the transistors. We are using a single pole double throw (SPDT) relay.

To perform switching of relay transistor BC 548 is used

1) Relay can switch AC & DC, whereas transistors can only switch DC.
2) Relays can switch high voltage, but transistors cannot.
3) Relays are a better choice for switching large currents (i.e.>5A).
4) Relays can switch many contacts at once.

The relay acts as a switch that is used to control the 230-volt AC supply. This relay will be turned off if there is no person inside the room. This relay can be used to turn off electrical appliances like fans, tubes, etc.

Figure: Photograph of a typical 12volt relay

Arduino Introduction

What is Arduino?

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards can read inputs - a light on a sensor, a finger on a button, or a Twitter message - and turn them into an output - activating a motor, turning on an LED, or publishing something online. You can tell your board what to do by sending instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearables, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, and musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just by following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module to understand how it works and save money.

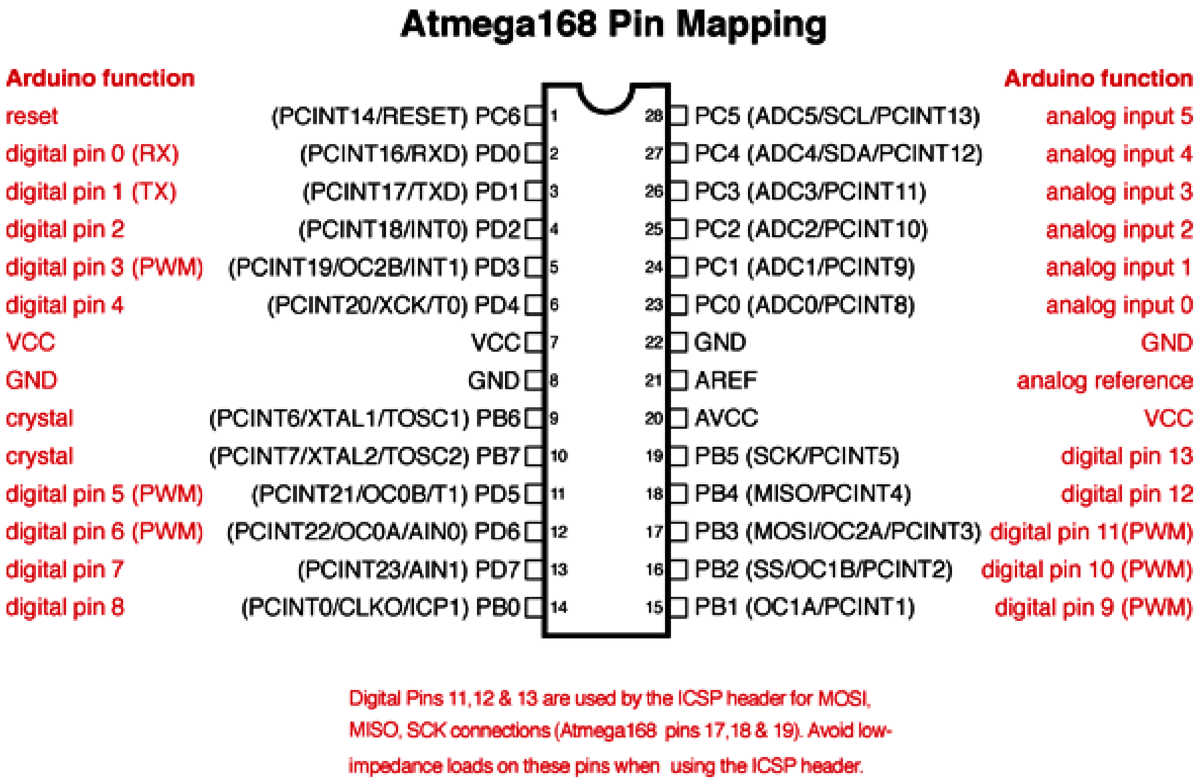
How to use Arduino?

See the getting started guide. If you are looking for inspiration you can find a great variety of Tutorials on Arduino Project Hub.

The text of the Arduino getting started guide is licensed under a Creative Commons Attribution-Share Alike 3.0 License. Code samples in the guide are released into the public domain.

Arduino8/328-Arduino Pin Mapping

Note that this chart is for the DIP-package chip. The Arduino Mini is based upon a smaller physical IC package that includes two extra ADC pins, which are not available in the DIP-package Arduino implementations.



Atmega168 Pin Mapping

Arduino function
reset (PCINT14/RESET) PC6 1
digital pin 0 (RX) (PCINT16/RXD) PD0 2
digital pin 1 (TX) (PCINT17/TXD) PD1 3
digital pin 2 (PCINT18/INT0) PD2 4
digital pin 3 (PWM) (PCINT19/OC2B/INT1) PD3 5
digital pin 4 (PCINT20/XCK/T0) PD4 6
VCC VCC 7
GND GND 8
crystal (PCINT6/XTAL1/TOSC1) PB6 9
crystal (PCINT7/XTAL2/TOSC2) PB7 10
digital pin 5 (PWM) (PCINT21/OC0B/T1) PD5 11
digital pin 6 (PWM) (PCINT22/OC0A/AIN0) PD6 12
digital pin 7 (PCINT23/AIN1) PD7 13
digital pin 8 (PCINT0/CLKO/ICP1) PB0 14
Arduino function
28 PC5 (ADC5/SCL/PCINT13) analog input 5
27 PC4 (ADC4/SDA/PCINT12) analog input 4
26 PC3 (ADC3/PCINT11) analog input 3
25 PC2 (ADC2/PCINT10) analog input 2
24 PC1 (ADC1/PCINT9) analog input 1
23 PC0 (ADC0/PCINT8) analog input 0
22 GND GND
21 AREF analog reference
20 AVCC VCC
19 PB5 (SCK/PCINT5) digital pin 13
18 PB4 (MISO/PCINT4) digital pin 12
17 PB3 (MOSI/OC2A/PCINT3) digital pin 11(PWM)
16 PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
15 PB1 (OC1A/PCINT1) digital pin 9 (PWM)

Digital Pins 11,12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

What is Bluetooth?

Bluetooth is a wireless technology used to transfer data between different electronic devices. The distance of data transmission is small in comparison to other modes of wireless communication. This technology eradicates the use of cords, cables, and adapters and permits electronic devices to communicate wirelessly with each other. Bluetooth technology permits hands-free headsets for incoming voice calls, the ability to print and fax, and automatic synchronization of PDA.

The key features of Bluetooth technology:

- Less complication
- Less power consumption
- Available at cheaper rates
- Robustness

History of Bluetooth

- 1998 - Birth of Bluetooth with the formation of Bluetooth Special Interest Group
  - Bluetooth is officially adopted as the name for the technology.
- 1999 - 1.0 Specification is officially released.
  - Bluetooth is declared the "Best of Show Technology Award" by COMDEX.
- 2000 - For the first time, Bluetooth-enabled mobile phone, PC card, mouse, headset and laptop are unveiled.
  - USB dongle prototype unveiled at COMDEX.
- 2001 - Bluetooth SIG Inc. is formed.
  - First hands-free car kit is launched.
- 2002 - First GPS receiver and Bluetooth enabled digital camera.
  - IEEE approves 802.15.1 specification for Bluetooth technology.
- 2003 - First Bluetooth-enabled MP3 player launched.
  - First FDA-approved Bluetooth-enabled medical system.
- 2004 - Bluetooth SIG adopts Enhanced Data Rate (EDR).
  - Bluetooth technology embedded in 250 million devices.
  - First Bluetooth-enabled stereo headphones launched.
- 2006 - Bluetooth is installed in 1 billion devices.

Bluetooth Decoder

This module enables you to wireless transmit & receive serial data. It is a drop in replacement for wired serial connections allowing transparent two way data communication. You can simply use it for serial port replacement to establish connection between MCU or embedded project and PC for data transfer. This board operates on 5V and has LED indication.

Features

- 5V power operation
- UART interface
- 10 meters range
- Status LED

Product Specifications

- Bluetooth protocol v2.0
- Baud Rate: 9600 bits per second
- Power Supply: +5 VDC 50mA
- Operating Temperate: -20C to +55 C
- Dimensions: 26.9 mm x 13 mm x 2.2 mm

4. INTRODUCTION TO 16X2 LCD DISPLAY

LCD stands for Liquid Crystal Display. The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters.

4.1 Pin Description

Most LCDs with two controllers have 16 Pins. Pin description is shown in the table below.

Pin No.	Name	Description
Pin no. 1	D7	**Data bus line 7 (MSB)**
Pin no. 2	D6	**Data bus line 6**
Pin no. 3	D5	**Data bus line 5**
Pin no. 4	D4	**Data bus line 4**
Pin no. 5	D3	**Data bus line 3**
Pin no. 6	D2	**Data bus line 2**
Pin no. 7	D1	**Data bus line 1**
Pin no. 8	D0	**Data bus line 0 (LSB)**
Pin no. 9	EN1	**Enable signal for row 0 and 1 (1st controller)**
Pin no. 10	R/W	**0 = Write to LCD module** **1 = Read from LCD module**
Pin no. 11	RS	**0 = Instruction input** **1 = Data input**
Pin no. 12	VEE	**Contrast adjusts**

Pin no. 13	VSS	**Power supply (GND)**
Pin no. 14	VCC	**Power supply (+5V)**
Pin no. 15	EN2	**Enable signal for row 2 and 3 (2ndcontroller)**
Pin no. 16	NC	**Not Connected**

Table No.4.1: Pin description of the LCD

4.2 DDRAM - Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So, whatever you send on the DDRAM is actually displayed on the LCD.

4.3 BF - Busy Flag

Busy Flag is a status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay. For the LCD processing. To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

4.4 Instruction Register (IR) and Data Register (DR)

There are two 8-bit registers controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

4.5 Commands and Instruction set

Only the instruction register (IR) and the data register (DR) of the LCD can be controlled by the MCU. Before starting the internal operation of the LCD, control information is temporarily stored into these registers to allow interfacing with various MCUs, which operate at different speeds, or various peripheral control devices. The internal operation of the LCD is determined by signals sent from the MCU.

4.6 Sending Commands to LCD

To send commands we simply need to select the command register. Everything is same as we have done in the initialization routine. But we will summarize the common steps and put them in a single subroutine.

Following are the steps:

- Move data to LCD port
- Select command register
- Select write operation
- Send enable signal
- Wait for LCD to process the command

3. HARDWARE IMPLEMENTATION

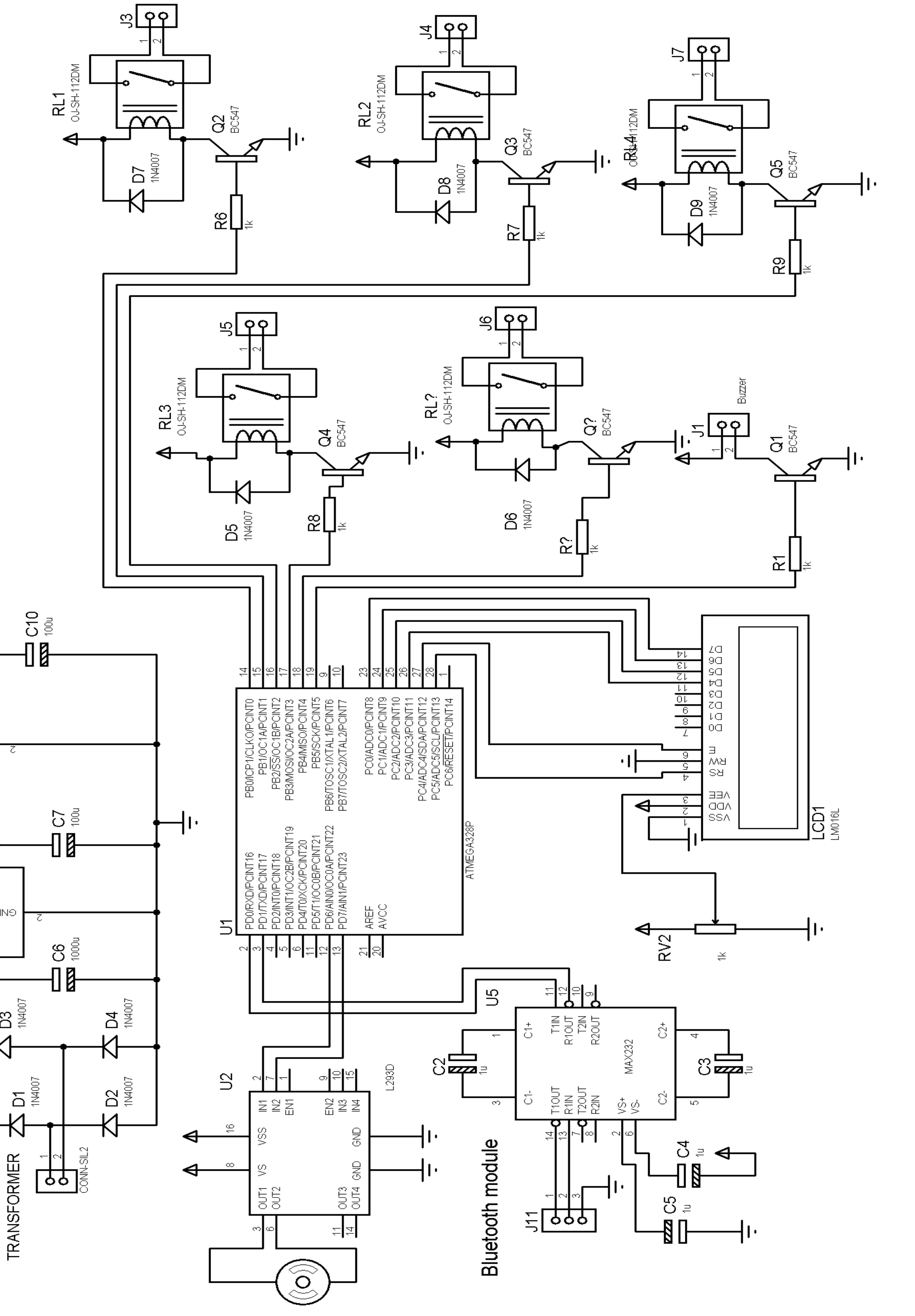


Fig 3.1 Circuit diagram

3.2 POWER SUPPLY

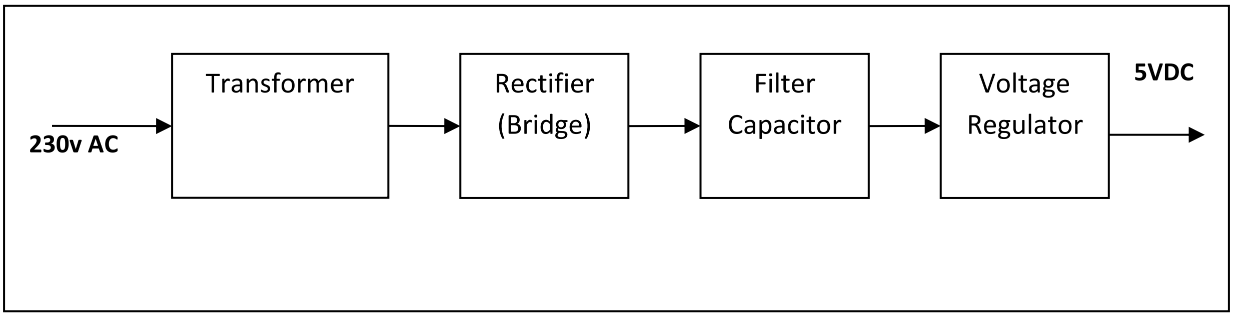


Fig. shows the block diagram of a typical power supply. The AC mains is given to the transformer primary to get the required voltage at the secondary. Then it is applied to the bridge rectifier, which converts the sinusoidal input into full wave rectified output. The output of the rectifier contains some ripple voltage. To remove this voltage filter circuit is used. A ripple voltage is nothing but a small value of AC over DC signal. Then a pure DC is given to the regulator. The function of the regulator is to give the constant or stable output DC in spite of changes in the load current.

The reasons for choosing IC regulator is that they are versatile in operation and relatively inexpensive with features like programmable output, current/voltage boosting, internal short circuit current limiting, thermal shutdown.

The 78XX are popularly known for regulation has been used. The 78XX series is a 3-terminal positive voltage regulator and 79XX series is a 3-terminal negative voltage regulator.

As name suggests it transforms the voltage level from one level to another. Transformer used is the step-down transformer to step 230 V to +9 V. It provides isolation too from the mains.

CIRCUIT DIAGRAM:-

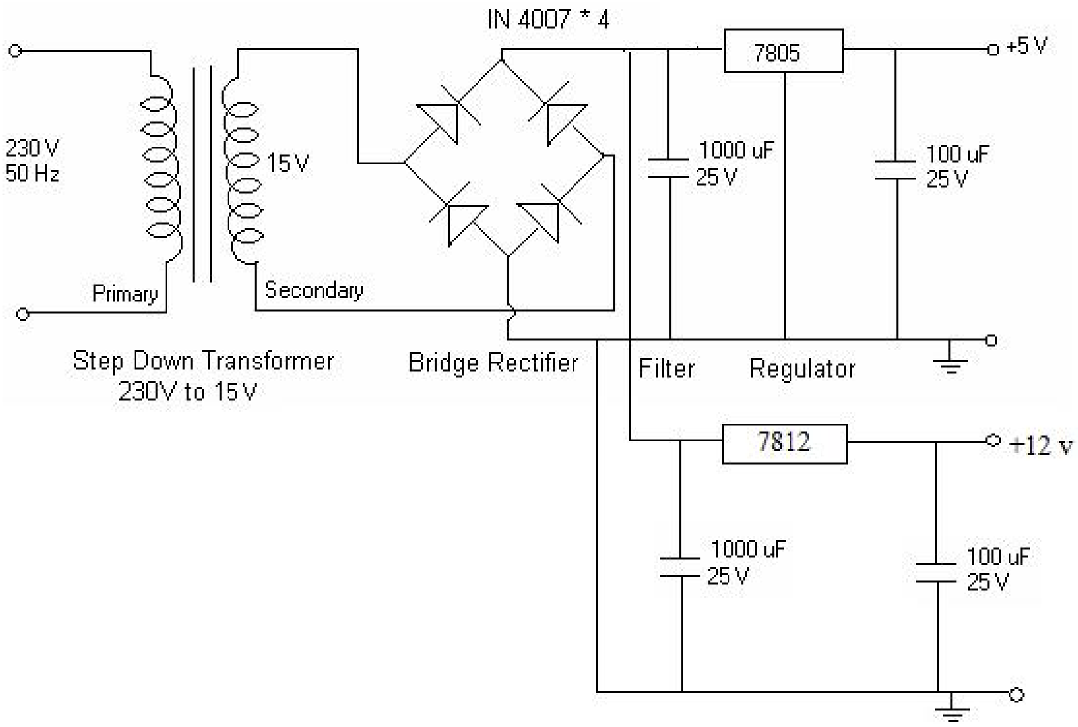


Fig 3.2 power supply design

Steps to design power supply: -

A] Determine the total current that the system sinks from supply.

B] Determine the voltage rating required for different components

Power supply is a vital part of all electronic systems. This circuit is required to drive the various components on the board. It is a normal Voltage Regulator built with ubiquitous Transformer-Bridge Rectifier-Filter-Regulator assembly. We require a 5 V supply for digital IC's.

A 230-15 V step down transformer is used, the output of which is 15V AC. This is applied to the bridge rectifier formed by diodes 1N 4007. The output of the bridge rectifier is DC, which contains a lot of ripples. To remove this ripple, a capacitor filter (C-filter) is used. The output of this filter is unregulated DC voltage. To regulate this, a 3-pin regulator IC-7805 is used.

IC 7805 and 7812 is used for the 12V and 5V supply. These are the standard ICs available for the required power supply. The supply to other 2 boards is given from this board through 2 and 3 pin connectors. The value of the capacitors is calculated according to the requirement. Here the requirement is of 12V hence we need a 1000µF capacitor. The other capacitor i.e. the 100µF is also selected according to it.

POWER SUPPLY DESIGN

The basic step in the designing of any system is to design the power supply required for that system. The steps involved in the designing of the power supply are as follows,

1) Determine the total current that the system sinks from the supply.

2) Determine the voltage rating required for the different components.

1) TRANSFORMER:

Transformer selection we required 12V for relay.

Min Input for 7805 is

= Drop across IC 7805 + Required Output voltage

= 3 V+ 5V

= 8 V

So, at Input of 7805 we required 8 V with margin

Consider drop across diode 0.7V so 2 diode conducts drop is 1.4 V

= 1.4 V +8 V

= 9.4 V

So, at secondary we required 10 V

FILTER:

For filter capacitor design

C= (I_l * t1)/Vr

Vr= ripple voltage

I_l = load current

t1= time during which the capacitor being discharge by load current

θ_1= sin-1[(E0 min)/ (E0 max)]

So unregulated power supply is design for 10 V

Vr = ripple voltage 10% of output voltage

Vr = 1.0 V

E0 min/E0 max = (10-0.7) / 10+0.7

= 9.3 / 10.7

θ_1 = sin-1 [9.3/10.7]

= 60°

Frequency 50 HZ

T1 = 1/50 = 20 ms

T for 360° = 20ms

For 180° = 10ms

For 60° = 20ms * (60°/360)

= 3.4m

2) RECTIFIER

For bridge

T1 = [time for 90° + time for θ1]

= 5ms + 3.4ms

= 8.4ms

I_l = load current supplied to various IC

I_l = (O/P current of IC Arduino Uno + O/P current of IC 232 + Current req. for display)

= 71mA + 30mA + 15.2 mA

=116.2 mA

C = (I_l * t1)/Vr

= (116.2 mA * 8.4 ms)/ 1 V

= 976.04 µf

So we select˜ 1000 µf capacitor

For diode design

PIV = Vm

Vm = E0 max + 2 Vf

= 10.7 + 1.4 V

= 12.1 V

$I_0 = I_l$ /2

= 116.2 mA/ 2

= 58.1 mA

Peak repetitive current

Ifm = [Il (t1+t2)]/t2

T2 = time for 90° - time for θ1

= 5ms - 3.4ms

=1.2ms

Ifm = 116.2mA(8.6ms+1.2ms) /1.2ms.

=833mA

From above specification diode 1N4007 is selected

PIV =100V

I = 1A

a) The TUF is increased to 0.812 as compared the full wave rectifier.
b) The PIV across each diode is the peak voltage across the load =V_m, not $2V_m$ as in the two-diode rectifier

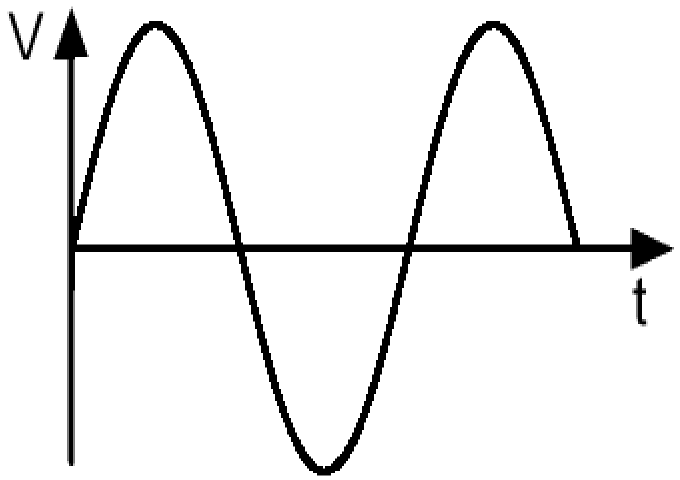
Output of the bridge rectifier is not pure DC and contains some AC some AC ripples in it. To remove these ripples, we have used capacitive filter, which smoothens the rippled out put that we apply to 7805 regulators IC that gives 5V DC. We preferred to choose capacitor filters since it is cost effective, readily available and not too bulky.

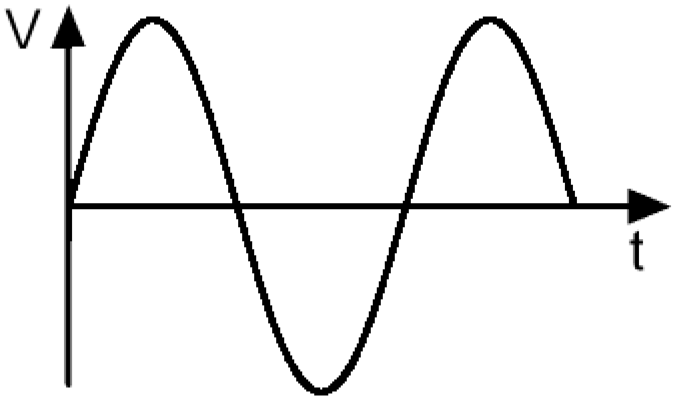
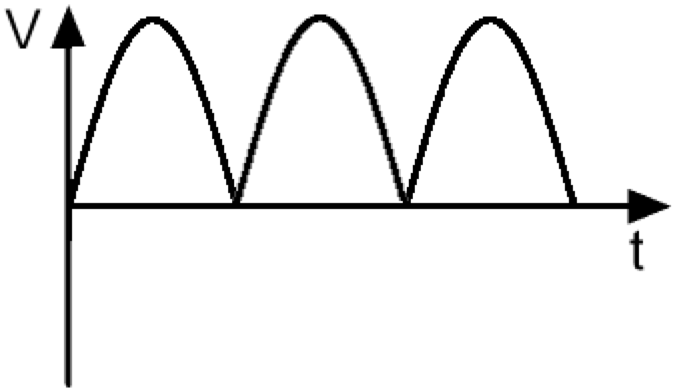
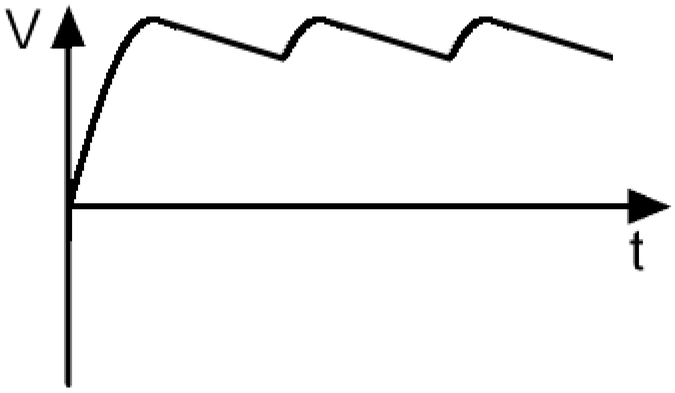
The value of the capacitor filter can be found by following formula,

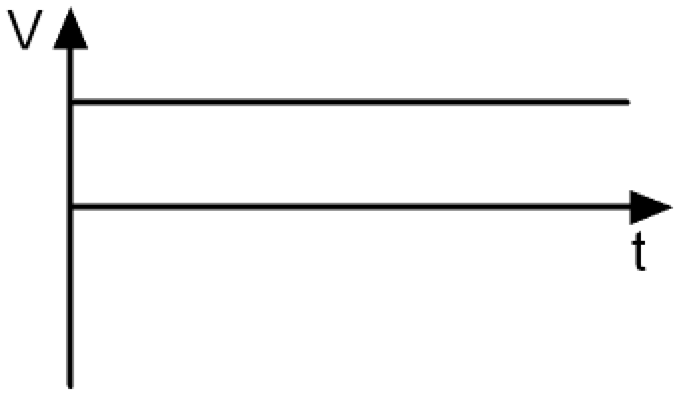
$$C = \frac{IL * t1}{Vr}$$

A regulator is a circuit that supplies a constant voltage regardless of changes in load current. The regulator used in our project is IC7805, which is a three terminal voltage regulator. A heat sink is used, so that the heat produced by the regulator dissipating power has a larger area from which to radiate the heat into the air by holding the case temperature to a much lower value than would result without the heat sink.

Waveforms for Power supply module

	We get 230 volts A.C. supply from the power grid (Electricity board). The voltage amplitude is of 230 volts and the frequency is 50 Hz.

	By using a Step down transformer, we are lowering the 230 volt AC supply to a lower value (e.g., 15 V) using a transformer. This lower voltage is still AC. The voltage amplitude is reduced but the frequency is the same, which is 50 Hz
	Then rectification is done by a set of 4 diodes (Bridge rectifier), this rectifier transforms this AC voltage into pulsating voltage. The negative half cycles of transformer output are converted to positive half cycles.
	The next step is filtering, which is done by an electrolytic capacitor of 100microF, this filter capacitor transforms this pulsating voltage into almost DC. This is having ripples.

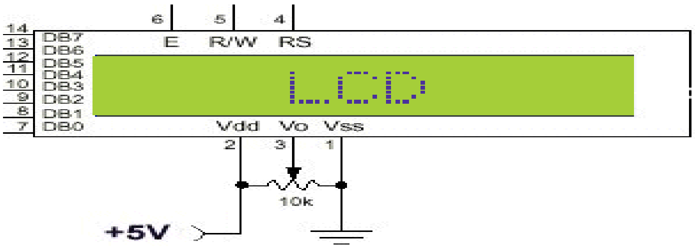
	The voltage obtained after the capacitor oscillates a little bit (this oscillation is called ripple), so a voltage regulating stage is necessary, done by a voltage regulator IC. After this stage, the output is true DC voltage

3.4 LCD DISPLAY INTERFACING

In market various displays are available like 7-segment, 5*7 matrix LED and LCD, bar graph, LCD, etc. It’s important for deciding the required display set for our project. Selection of display depends on various factors like power consumption, ambient light conditions, surrounding temperature, visibility from long distance, total information to be display, cost of display, circuit/lines required for display interfacing, etc.

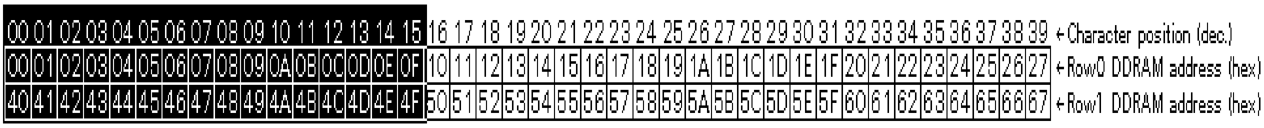
The most common way to accomplish this is with the LCD (Liquid Crystal Display). LCDs have become a cheap and easy way to get text display for an embedded system.

Common LCDs are set up as 16 to 20 characters by 1 to 4 lines and noted as 16*2, 20*2, 16*4, 20*4, etc. Following figure shows the basic pin diagram of 16*2 LCD display.



Display Data RAM (DDRAM):

Display data RAM (DDRAM) is where you send the characters (ASCII code) you want to see on the LCD screen. It stores display data represented in 8-bit character codes. Its capacity is 80 characters (bytes). Below you see DD RAM address layout of a 2*16 LCD.



00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	←Character position (dec.)
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	←Row0 DDRAM address (hex)
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	←Row1 DDRAM address (hex)

Figure 3.5 DDRAM Address Layou

In the above memory map, the area shaded in black is the visible display (For 16x2 display). For first line addresses for first 15 characters is from 00h to 0Fh. But for second line address of first character is 40h and so on up to 4Fh for the 16th character. So, if you want to display the text at specific positions of LCD, we require to manipulate address and then to set cursor position accordingly.

REGISTERS:

It has two 8-bit registers, an instruction registers (IR) and a data register (DR). The IR stores instruction codes. The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR is automatically written into DDRAM or CGRAM by an internal operation. These two registers can be selected by the register selector (RS) signal. See the table below:

Register Selection

RS	R/W	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Table 3.1: Register Selection

Busy Flag (BF):

When the busy flag is 1, the LCD is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/W = 1 (see the table above), the busy flag is output to DB7 (MSB of LCD data bus). The next instruction must be written after ensuring that the busy flag is 0.

LCD Commands: The LCD's internal controller accept several commands and modify the display accordingly. These commands would be things like:

– Clear screen

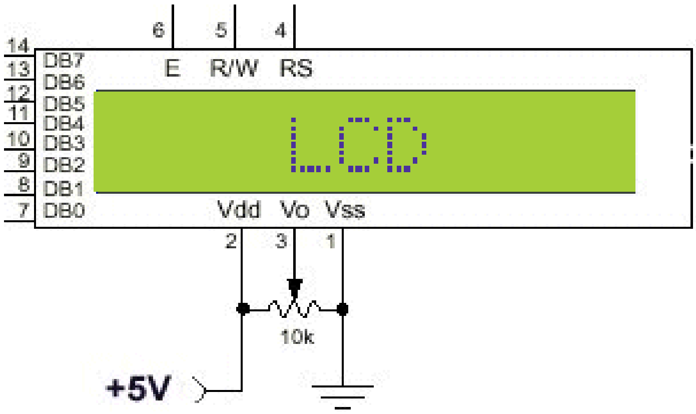
– Return home

– Shift display right/left

Instruction	Decimal	HEX
Function set (8-bit interface, 2 lines, 5*7 Pixels)	56	38
Function set (8-bit interface, 1 line, 5*7 Pixels)	48	30
Function set (4-bit interface, 2 lines, 5*7 Pixels)	40	28
Function set (4-bit interface, 1 line, 5*7 Pixels)	32	20
Entry mode set	See Below	See Below
Scroll display one character right (all lines)	28	1E
Scroll display one character left (all lines)	24	18
Home (move cursor to top/left character position)	2	2
Move cursor one character left	16	10
Move cursor one character right	20	14
Turn on visible underline cursor	14	0E
Turn on visible blinking-block cursor	15	0F
Make cursor invisible	12	0C
Blank the display (without clearing)	8	08
Restore the display (with cursor hidden)	12	0C
Clear Screen	1	01
Set cursor position (DDRAM address)	128 + addr	80+ addr
Set pointer in character-generator RAM (CG RAM address)	64 + addr	40+ addr

Table 3.2: LCD Commands

LCD Display PINOUT and Description:



Data pins D7:D0: Pins 7-14: Bi-directional data/command pins. Alphanumeric characters are sent in ASCII format. As shown in figure lcd1.0, there are 8 pins, Pin No.7-14 used for data lines.

RS: Register Select: Pin No.4:

RS = 0; Command Register is selected

RS = 1; Data Register is selected

R/W: Read or Write: Pin No.5:

R/W=0; Write.

R/W= 1; Read

E: Enable (Latch data): Pin No.6:

Used to latch the data present on the data pins.

A high-to-low edge is needed to latch the data.

Vo: contrast control: Pin No.2:

Used to control contrast of LCD.

NOTE:

When writing to the display, data is transferred only on the high to low transition of this signal. However, when reading from the display, data will become available shortly after the low to high transition and remain available until the signal falls low again.

The standard LCD Display requires 3 control lines as well as either 8 I/O lines for the data bus. The three control lines are EN, **RS**, and **RW**. Note that the EN line must be raised/lowered before/after each instruction sent to the LCD regardless of whether that instruction is read or write text or instruction. In short, you must always manipulate EN when communicating with the LCD. EN is the LCD's way of knowing that you are talking to it. If you don't raise/lower EN, the LCD doesn't know you're talking to it on the other lines.

Checking the Busy Flag:

We can use subroutine for checking busy flag or just a big (and safe) delay.

1. Set R/W Pin of the LCD HIGH (read from the LCD)

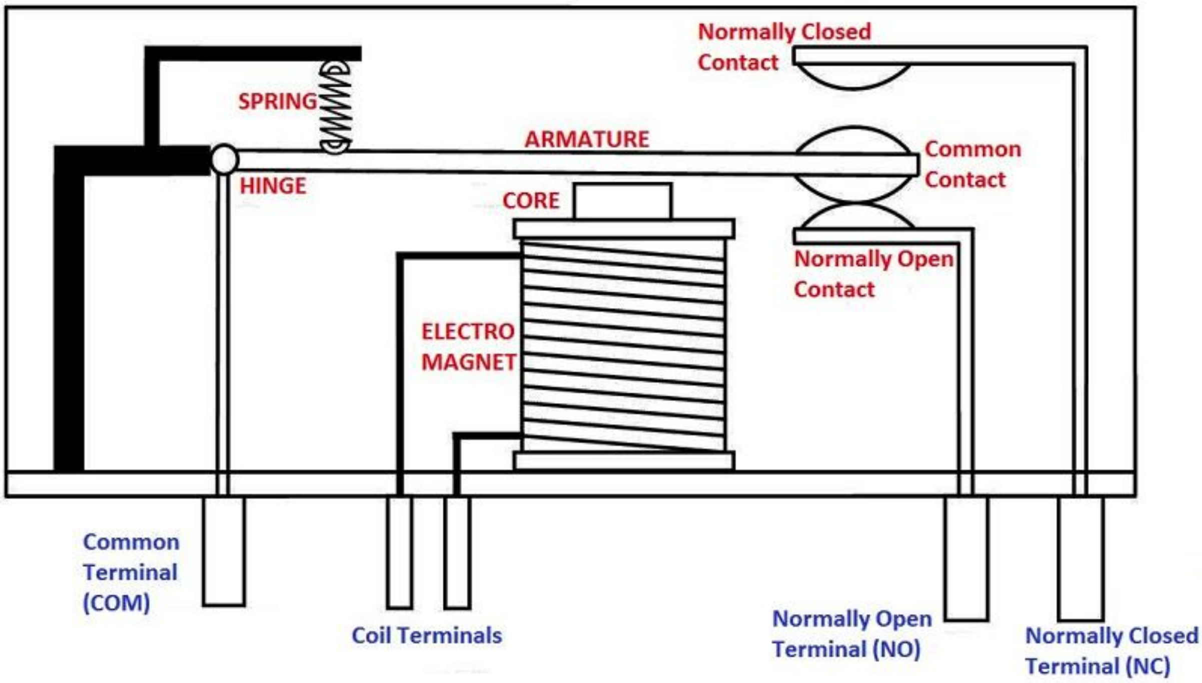
2. Select the instruction register by setting RS pin LOW
3. Enable the LCD by Setting the enable pin HIGH
4. The most significant bit of the LCD data bus is the state of the busy flag (1=Busy,0=ready to accept instructions/data). The other bits hold the current value of the address counter.

If the LCD never come out from “busy" status because of some problems, the program will “hang," waiting for DB7 to go low. So, in real applications it would be wise to put some kind of time limit on the delay--for example, a maximum of 100 attempts to wait for the busy signal to go low. This would guarantee that even if the LCD hardware fails, the program would not lock up.

Relay

Relays are components which allow a low-power circuit to switch a relatively high current on and off, or to control signals that must be electrically isolated from the controlling circuit itself. To make a relay operate, we have to pass a suitable pull-in and holding current (DC) through its energizing coil. And generally, relay coils are designed to operate from a particular supply voltage, often 12V, in case of many small relays used for electronics work.

Relay is an electromagnetic device which is used to isolate two circuits electrically and connect them magnetically. They are very useful devices and allow one circuit to switch another one while they are completely separate. They are often used to interface an electronic circuit (working at a low voltage) to an electrical circuit which works at very high voltage. For example, a relay can make a 5V DC battery circuit to switch a 230V AC mains circuit. Thus, a small sensor circuit can drive, say, a fan or an electric bulb.



A relay switch can be divided into two parts: input and output. The input section has a coil which generates magnetic field when a small voltage from an electronic circuit is applied to it. This voltage is called the operating voltage. Commonly used relays are available in different configuration of operating voltages like 6V, 9V, 12V, 24V etc. The output section consists of contactors which connect or disconnect mechanically. In a basic relay there are three contactors: normally open (NO), normally closed (NC) and common (COM). At no input state, the COM is connected to NC. When the operating voltage is applied the relay coil gets energized and the COM changes contact to NO. Different relay configurations are available like SPST, SPDT, and DPDTetc., which have different number of changeover contacts. By using proper combination of contactors, the electrical circuit can be switched on and off. Get inner details about structure of a relay switch.

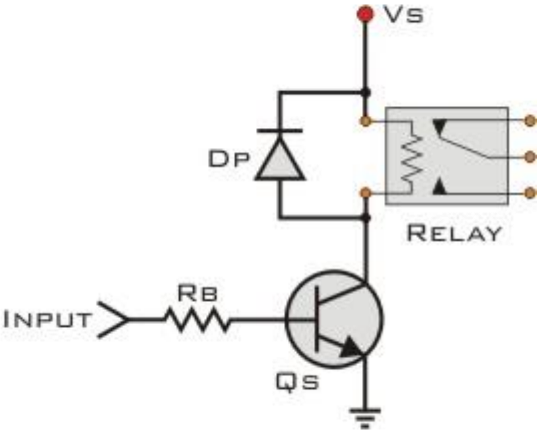
The construction and working of a relay can be understood from the above diagram. The main part of a relay is the electromagnet. The electromagnet is made by a coil of wire wrapped around a soft iron core which provides low reluctance path for magnetic flux. It also consists of a movable iron armature and one or more set of contacts. These are held in a position by a spring as shown in the above diagram.

Normally a SPDT relay has 5 terminals. Two them are used to energies the electromagnet and other three are COM, NO, NC. COM stands for Common, NO stands for Normally Open and NC stands for Normally Closed. When the electromagnet is not energized, the armature will be connected to NC contact. Thus, COM and NC will be connected. When the electromagnet is energized, the electromagnet attracts the iron armature and it will be connected to NO contact. Thus, COM and NO will be connected.

Relay Driver Circuit

We have to provide enough base current to turn the transistor on and off. NPN transistor BC547 is used to control a Relay with a 12V coil, operating from a +12V supply. Series base resistor R1 is used to set the base current for transistor, so that the transistor is driven into saturation (fully turned on) when the relay is to be energized. Thus, the transistor will have minimal voltage drop, and hence dissipate very little power as well as delivering most of the 12V to the relay coil.

Basic transistor relay driver, actuated on HIGH input (NPN)



This circuit will drive a relay coil from a low power output, usually from an IC like Arduino Uno. It is used to switch high loads or a load that needs AC current to operate. The relay will be actuated when the input of the circuit goes high. The protection diode Dp is used to protect the transistor from the reverse current generated from the coil of the relay during the switch off time. The values for Rb and Qs vary accordingly.

The way to calculate them is:

First, we calculate the load current:

IL = VS / RL

Then we calculate the transistor h_{FE}. It must be at least 5 times the load current IL divided by the maximum output current from the Input to the base of the transistor

$$\mathbf{h_{FE}(min) > 5\ X\ \frac{I_L}{I_{INPUT}}}$$

Now we can choose the transistor Qs. We must select it according to its IC that must be greater than I_L and its current gain h_{FE}.
So, we have chosen the BC547 transistor.

Then we calculate the base resistor R_B, If the input is taken from a component (possible an IC) that uses the same power supply as the transistor (that is Vs), then the form is:

$$\mathbf{R_B = 0.2 \text{ X } R_L \text{ X } h_{FE}}$$

Otherwise, if the component uses another power source (like V_{CC}) then the form is:

$$\mathbf{R_B = \frac{V_{CC} \text{ X } h_{FE}}{5 \text{ X } I_L}}$$

The protective diode could be the 1N4001 or any general-purpose diode.

The output from an Arduino Uno IC is required to operate a relay with a 160 Ohm coil. The supply voltage is 12V for the transistor and 5V for the IC. The IC can supply a max current of 2mA.

$$\mathbf{I_L = Vs / R_L \quad => \quad I_L = 12 / 160 = 75mA}$$

The transistor must have an h_{FE} greater than 5 X 75 / 2 => **h_{FE} > 187.5**. So, we have chosen a transistor with h_{FE} = 200 and IC = 100mA.

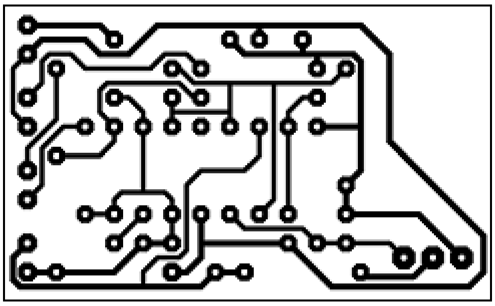
Now for the R_B resistor. Because the power supplies of transistor and I_C are different, we use the second formula:

$$\mathbf{R_B = \frac{V_{CC} \text{ X } h_{FE}}{5 \text{ X } I_L}}$$

This will produce R_B = 1333.3 Ohm. We have chosen the closest resistor possible to this value. RB = 1000 ohm = 1 K ohm

PCB LAYOUT

A printed circuit board, or PCB, is used to mechanically support and electrically connect electronic components using conductive pathways, tracks or traces etched from copper sheets laminated onto a non-conductive substrate. It is also referred to as printed wiring board (PWB) or etched wiring board. A PCB populated with electronic components is a printed circuit assembly (PCA), also known as a printed circuit board assembly (PCBA).

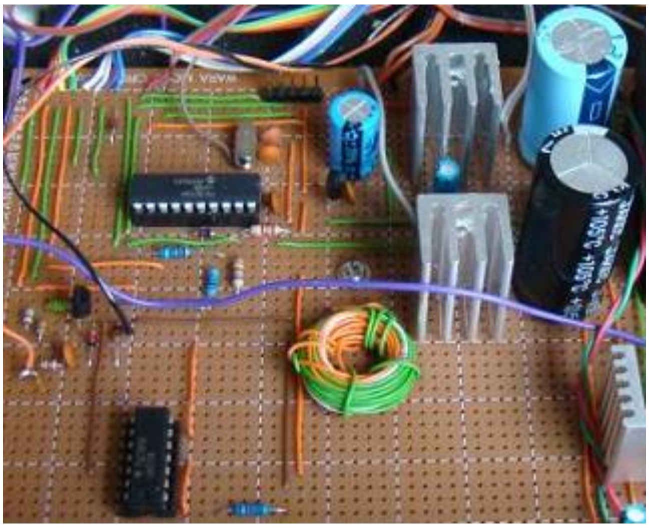


Printed Circuit Boards play a vital role here in determining the overall performance of the electronic equipment. A good PCB design ensures that the noise introduced as a result of component placement and track layout is held within limits while still providing components years of assembly maintenance and performance reliability.

Where and why are PCB's used?

Printed circuits boards are used to route electric signals through copper track which are firmly bonded to an insulating base.

Advantages of PCB over common wiring are:



1. PCBs are necessary for connecting a large number of electronic components in a very small area with minimum parasitic effects.

2. PCBs are simulated with mass production with less chance of writing error.

3. Small components are easily mounted.

4. Servicing in simplified.

Base Materials used for PCB

The base materials used for PCB's are glass epoxy, epoxy paper, polyester etc. Copper foil used for copper clad is manufactured by the process of electronic deposition.

Preparation of Single Sided PCB

In single-sided PCB conductor tracks are present on one side of copper-clad board. So, crossing of conductors is not allowed. It is mechanically & chemically cleansed. The photo resist is an organic solution which when exposed to light of proper wavelength, changes their solubility in developer but after exposure to light is not soluble.

Laminate coating of photo resist is done by: (i) Spray coating (ii) Dip coating (iii) Roller coating.

The coated copper clad and laminated film negative is kept in intimate contact with each other. The assembly is exposed to UV light and is rinsed in the developer tank. Proper developer has to be used for a particular photo resist and then the PCB is dyed in a tray. The dye reveals the flux to be used for a particular photo resist. Then the PCB is dyed in a tray.

FABRICATION

The required circuit is designed and the layout of the circuit is done on the component side as well as the copper clad side. Spaces are provided for holes to insert the respective components. Etch resistant ink coatings are given on the interconnecting marks.

ETCHING

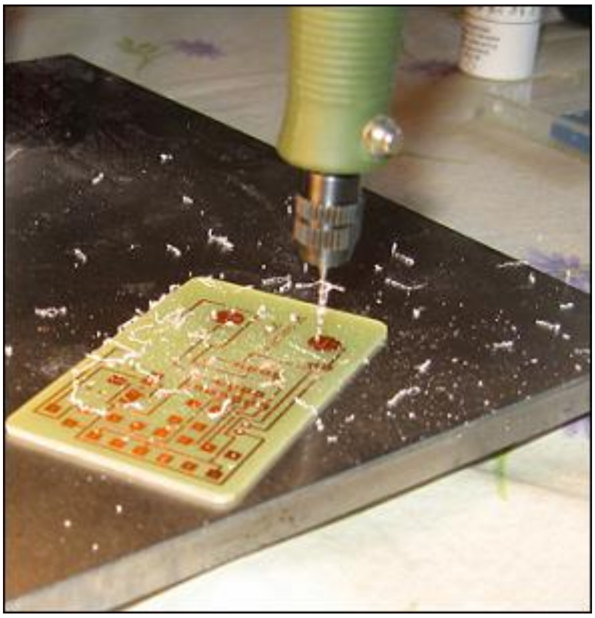
The copper clad PCB is etched with ferrous chloride solution containing a small amount of Hydro Chloric Acid for increasing activeness of Ferric Chloride in etching. Wherever the varnish coating is there the copper remains. Then it is washed with water and Oxalic Acid.

The vast majority of printed circuit boards are made by bonding a layer of copper over the entire substrate, sometimes on both sides, (creating a "blank PCB") then removing unwanted copper after applying a temporary mask (e.g. by etching), leaving only the desired copper traces. A few PCBs are made by adding traces to the bare substrate (or a substrate with a very thin layer of copper) usually by a complex process of multiple electroplating steps.

There are three common "subtractive" methods (methods that remove copper) used for the production of printed circuit boards:

- Silk screen printing uses etch-resistant inks to protect the copper foil. Subsequent etching removes the unwanted copper. Alternatively, the ink may be conductive, printed on a blank (non-conductive) board. The latter technique is also used in the manufacture of hybrid circuits.
- Photoengraving uses a photomask and chemical etching to remove the copper foil from the substrate. The photomask is usually prepared with a photoplotter from data produced by a technician using CAM, or computer-aided manufacturing software. Laser-printed transparencies are typically employed for photo tools; however, direct laser imaging techniques are being employed to replace photo tools for high-resolution requirements.
- PCB milling uses a two or three-axis mechanical milling system to mill away the copper foil from the substrate. A PCB milling machine (referred to as a 'PCB Prototyper') operates in a similar way to a plotter, receiving commands from the host software that control the position of the milling head in the x, y, and (if relevant) z axis. Data to drive the Prototyper is extracted from files generated in PCB design software and stored in HPGL or Gerber file format.

DRILLING



Holes through a PCB are typically drilled with tiny drill bits made of solid tungsten carbide. The drilling is performed by automated drilling machines with placement controlled by a drill tape or drill file. These computer-generated files are also called numerically controlled drill (NCD) files or "Excellon files".

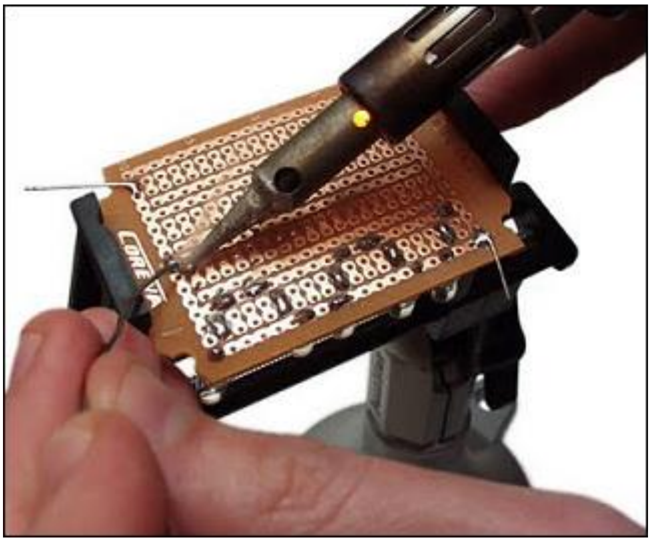
The drill file describes the location and size of each drilled hole. These holes are often filled with annular rings (hollow rivets) to create vias. Vias allow the electrical and thermal connection of conductors on opposite sides of the PCB.

Plating and Coating

PCBs are plated with Solder, Tin, or Gold over Nickel as a resist for etching (removal) away the (unneeded after plating) underlying copper. Matte solder is usually fused to provide a better bonding surface or stripped to bare copper.

Treatments, such as benzimidazolethiol, prevent surface oxidation of bare copper. The places to which components will be mounted are typically plated, because untreated bare copper oxidizes quickly, and therefore is not readily solderable. Traditionally, any exposed copper was coated with solder by hot air solder leveling (HASL). This solder was a tin-lead alloy, however, new solder compounds are now used to achieve compliance with the RoHS directive in the EU and US, which restricts the use of lead. One of these lead-free compounds is SN100CL, made up of 99.3% tin, 0.7% copper, 0.05% nickel, and a nominal of 60ppm germanium.

SOLDERING



Soldering is the process of joining two metals using an alloy solder consisting of Tin and Lead (Sn-Pb). Tin determines the melting whereas Lead is used to reduce the cost. After the PCB fabrication is done, the various components are arranged at proper locations on the PCB and then the soldering is done. All liquids consist of particles which attract each other. The surface is always trying to shrink and this is because of surface tension. The principle behind soldering is that when liquid particles are brought in contact with the walls of the solid surface, it may happen that the solid attracts the liquid surface. This property is called adhesive property. Care must be taken that the melting point of solder is below that of the metal so that its surface is melted without melting without the metal.

NEED FOR FLUX

During the soldering process the flux acts as a medium for improving the degree of melting. The basic functions of flux are mentioned below:

1. Removes oxide from the surface.
2. It transfers heat from source to the joining & provide liquid cover including air gap.
3. Removal of residue after the completion of the soldering operation.

PCB LAYOUT: The layout can be done either by hand or by using PCB designing software mentioned below:

- EAGLE by Cadsoft
- Dip Trace by Novarm
- FreePCB by Allan Wright (open-source Win2K/XP)
- Free Routing by AlfonsWirtz (Java, Autorouter with free angle support)
- Cadstar by Zuken
- gEDA, open-source PCB software project
- OrCAD by Cadence
- Kicad, open-source suite

- PADS by Mentor Graphics
- PCB123 Design by Sunstone Circuits
- Proteus

Here we have used eagle software to design PCB layout

3.6 PCB layout

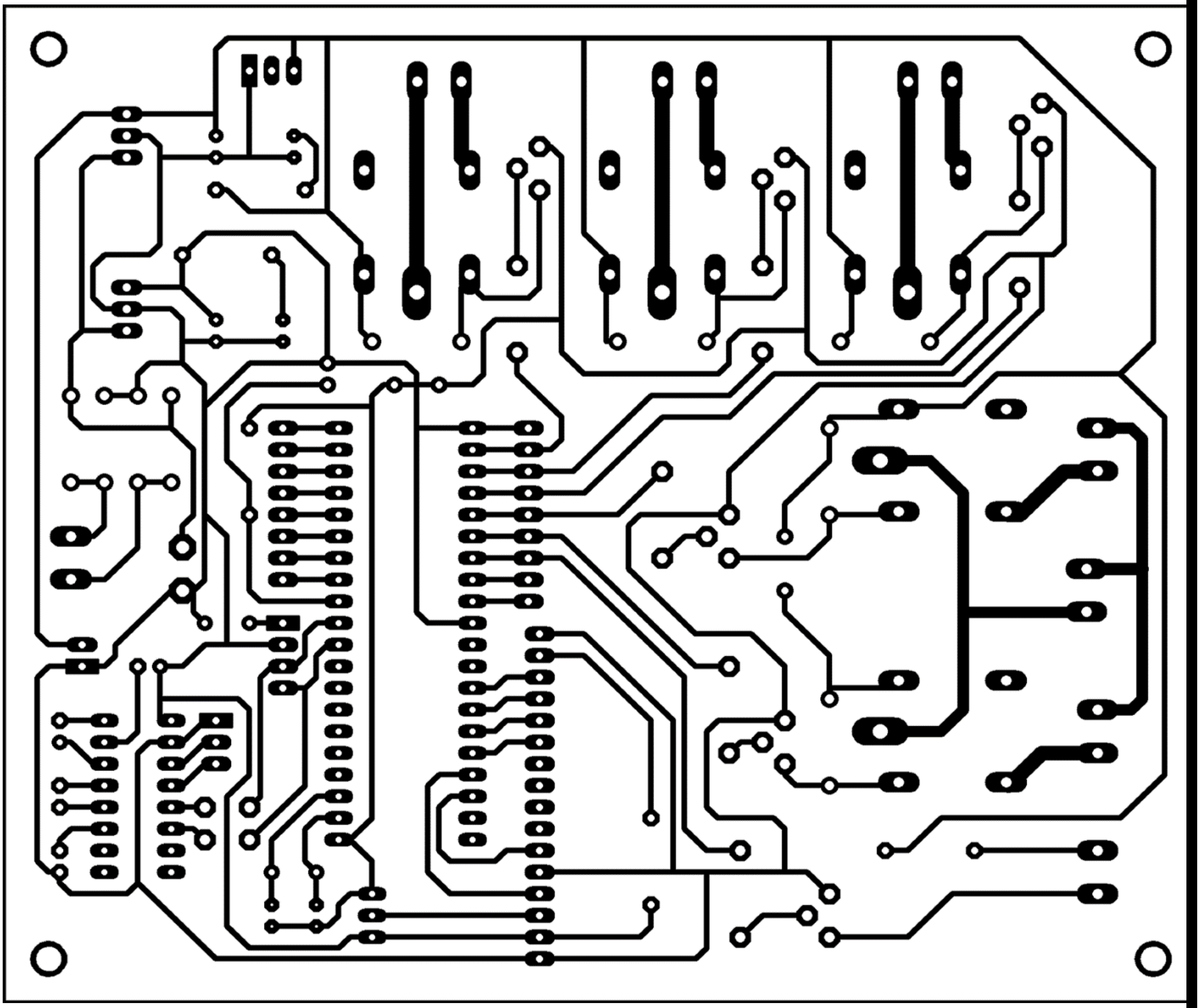


Fig 3.7 PCB layout

3.6PROJECT SPECIFICATION

LCD DISPLAY

Standard LCD module from LAMPEX is used for the DISPLAY.

1) Display format CHARA. X LINE 16 x 2 mm.
2) Font matrix of W x H 5 x 8 mm.
3) Viewing area of W x H 64.0 x 14.0 mm.
4) PCB size W x H 80.0 x 36.0 mm.
5) Supply voltage of 5 V and current of 1.2 mA.
6) Back Light voltage of 5 V and current 60 mA.

MICROCONTROLLER Arduino Uno

1) 4K Bytes of In-System Programmable (ISP) Flash Memory- Endurance: 10,000 Write/Erase Cycles
2) 4.0V to 5.5V Operating Range
3) Fully Static Operation: 0 Hz to 33 MHz
4) 128 x 8-bit Internal RAM
5) 32 Programmable I/O Lines
6) Two 16-bit Timer/Counters
7) Six Interrupt Sources
8) Full Duplex UART Serial Channel
9) Low-power Idle and Power-down Modes
10) Fast Programming Time
11) Flexible ISP Programming (Byte and Page Mode)

3.7 TESTING

- **PCB CHECKING:**

- All the tracks of PCB are checked.
- We checked continuity of all tracks.
- High voltage was applied between two independent tracks to check any hair size short or air gap.

- **VISUAL TESTING:**

- Polarities of all the components like capacitors, connectors etc. are checked.
- It is seen that all the IC sockets are soldered properly.

➢ **MULTIMETER TESTING:**

- All the IC sockets and power supply are soldered and continuity is checked
- Also, VCC and GND voltage are checked.
- Voltages at all the pins of the microcontroller are checked with respect to ground.
- Values of all possible components are checked on multimeter.

➢ **TESTING OF KIT:**

- We tested ANDROID decoder using LED connected at the output lines.
- Microcontroller's working was ensured by debugging the program.

4.SOFTWARE IMPLEMENTATION

4.1ALGORITHM

Step 1: Start

Step 2: Clear all the registers

Step 3: Turn off all the motors

Step 4: Initialize LCD

Step 5: Display 'WELCOME' message on LCD

0Step 10: Read the ANDROID input.

Step 11: check if input is = 1

Step 12: If yes, then turn on first relay.

Step 13: if no then check input is = 2

Step 14: If yes, then turn on second relay.

Step 15: if no then check input is =3

Step 16: If yes, then turn on third relay.

Step 17: if no then check input is =4

Step 18: If yes, then turn on forth relay.

Step 19: if no then check input is =5

Step 20: If yes, then turn on fifth relay.

Step 21: if no then check input is =6

Step 22: If yes, then turn on sixth relay.

Step 23: if no then check input is =7

Step 24: If yes, then turn off first relay.

Step 25: if no then check input is =8

Step 26: If yes, then turn off second relay.

Step 27: if no then check input is =9

Step 28: If yes, then turn off third relay.

Step 29: if no then check input is =*

Step 30: If yes, then turn off forth relay.

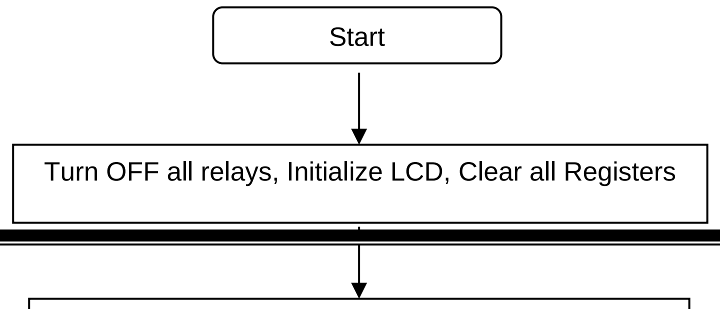
Step 31: if no then check input is =0

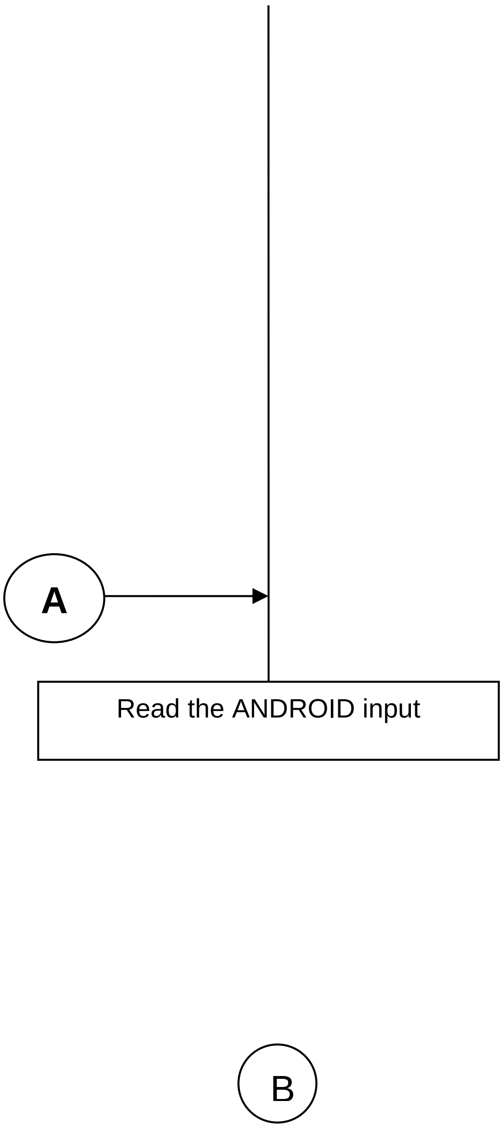
Step 32: If yes, then turn off fifth relay.

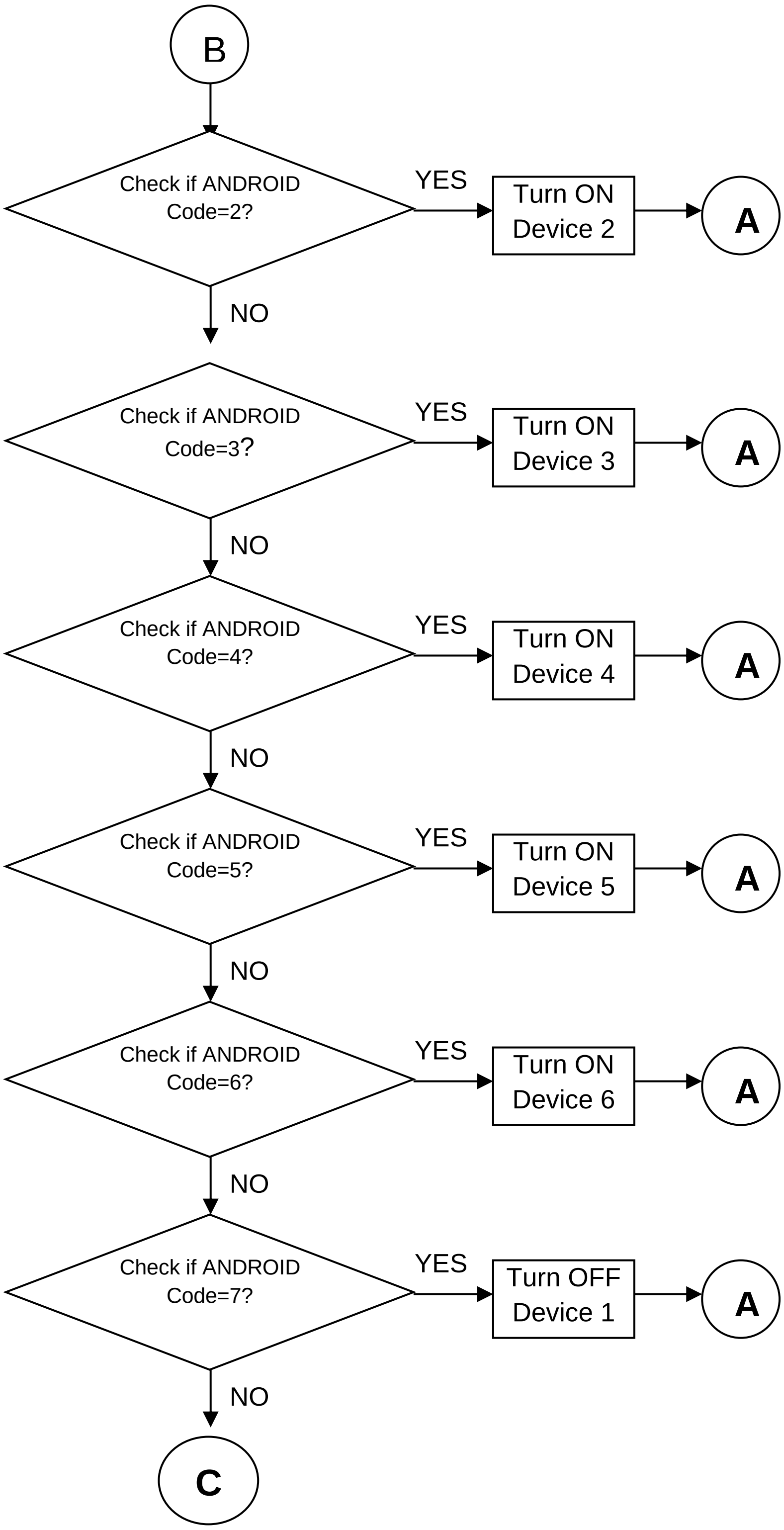
Step 33: if no then check input is = #

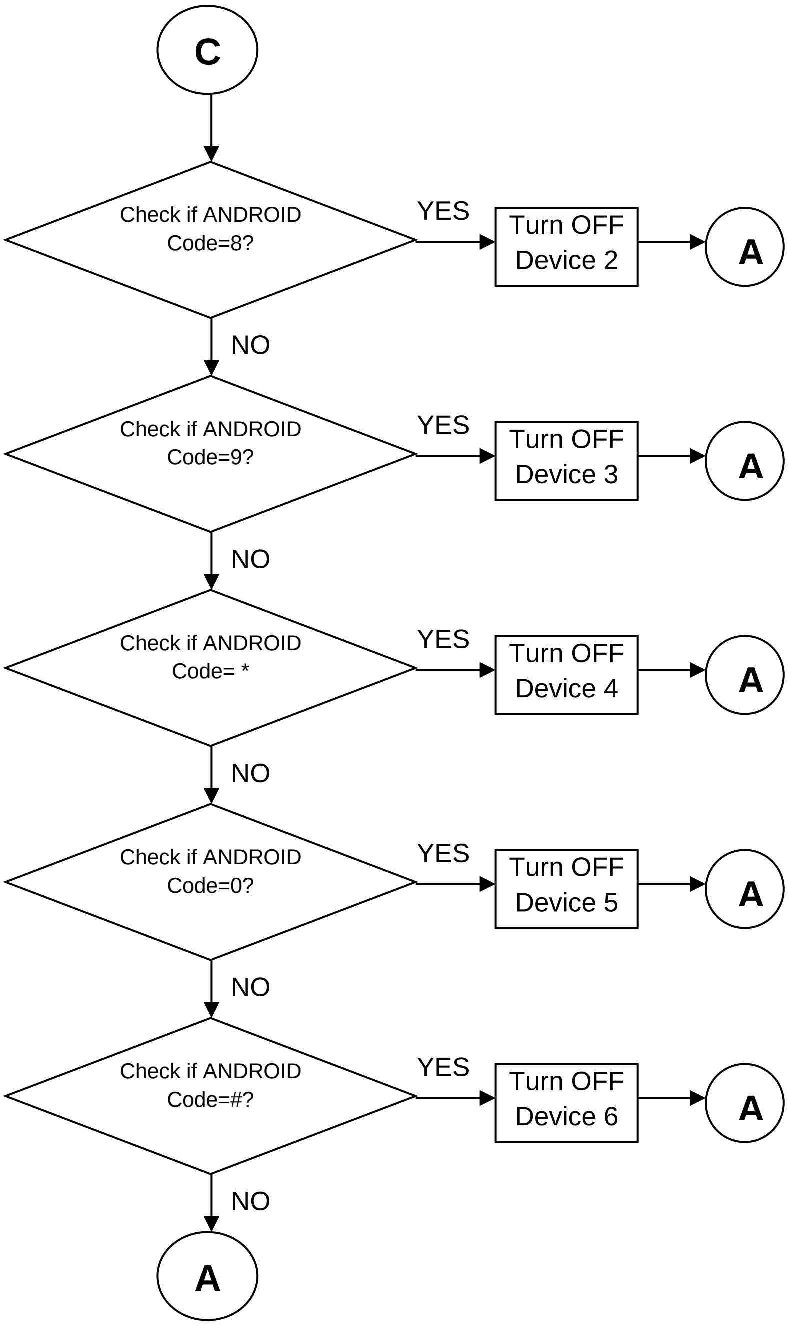
Step 34: If yes, then turn off Sixth relay.

4.2 FLOWCHART:-




A
Read the ANDROID input
B


B
Check if ANDROID Code=2?
YES
Turn ON Device 2
A
NO
Check if ANDROID Code=3?
YES
Turn ON Device 3
A
NO
Check if ANDROID Code=4?
YES
Turn ON Device 4
A
NO
Check if ANDROID Code=5?
YES
Turn ON Device 5
A
NO
Check if ANDROID Code=6?
YES
Turn ON Device 6
A
NO
Check if ANDROID Code=7?
YES
Turn OFF Device 1
A
NO
C



4.1FLOWCHART OF SYSTEM

EAGLE PCB Design Software

The name EAGLE is a shortform, which stands for Easily Applicable Graphical Layout Editor. This software offers user friendly and powerful solutions for PCB design, including Schematic Capture, Board Layout and Autorouter. This software is developed by Cadsoft.

EAGLE is popular among hobbyists because of its freeware license and rich availability of component libraries on the web. EAGLE has following 2 sections:

Schematic capture: EAGLE contains a schematic editor, for designing circuit diagrams. Parts can be placed on many sheets and connected together through ports.

PCB layout: The PCB layout editor allows back annotation to the schematic and auto-routing to automatically connect traces based on the connections defined in the schematic.

How to use EAGLE Software

1) Create new project

Start up the Eagle control panel, and right-click on "projects" to create a new project. You'll get to name it whatever you want

2) Create new schematic in the project

Once you have created the new project, it will be "opened" automatically (which doesn't do much other than tell EAGLE that "this is the current project".) Right click on the new project and follow the popup menus to create a new schematic.

3) Find and place ("add") components

Components are added to a schematic from the ADD dialog, which you get to by clicking the ADD button over on the GUI menu.

4) Neaten things up

We can make things a bit neater by moving some of the components, junctions, and wires around just a BIT using the Move button. When you select an object after clicking the move button, the object gets attached to your mouse pointer and behaves like when you were adding objects with respect to the other mouse buttons.

5) Re-zoom the drawing

Since we have all the components added, we can use the zoom-to-fit button to fit things better in our window.

6) Wiring components

Now it's time to make wire between components, find "Wire" to start

When you finish making wire between two components, hit "Escape" in the keyboard to move to the next one

7) Do Rule Check!

The button shown does an electrical rule check. It will check whether the pins designated outputs are connected to inputs, whether there are obvious missing junctions, and stuff like that.

8) Fix Errors and Warnings

When we run it on our schematics so far, we get a warning that the junction are missing. We could fix that with the name command, or just leave it as is.

9) Apply component values

When we plopped down the components during the "add" phase, we didn't assign specific values to any of them. Some of the components have inherent values that don't need to change. But the resistors, capacitors, and diodes should all have their values filled in appropriately.

Values are assigned using the "Value" button. After selecting the button, click on each component near its origin (little "+" sign), and you should be presented with an opportunity to change the value.

10) Transform the schematic into board

After finishing and saving your schematic, now we can move to the next step which is to move to board drawing by using "Board" as shown in the picture.

11) Rearrange your components

Use "move" and the right-button of the mouse to rearrange your circuit on the board

12) Autorouter

After rearranging your circuit, just go to "Tools / Auto" Then hit "OK"

The job is done

Your circuit is ready, save your work as usual and print it if you want

5. OPERATION OF THE SYSTEM

Each key is assigned for particular task. If you will press the password then, wait for few seconds. And give the command through the cell phone.

SR No.	Key	Device(demo)	Device(field)	Function
1	1	Bulb	Motor1	Turn on
2	2	Fan	Boiler	Turn on
3	3	Buzzer	Machinery	Turn on
4	4		Fan	Turn on
5	5		Motor2	Turn on
6	6		Lights	Turn on
7	7	Bulb	Motor1	Turn off
8	8	Fan	Boiler	Turn off
9	9	Buzzer	Machinery	Turn off
10	0		Fan	Turn off
11	*		Motor2	Turn off
12	#		Lights	Turn off

Table 5.1 function of keys

9. TESTING AND TROUBLE SHOOTING

DEBUGGING & TESTING

We have used simulator kit of Arduino Uno for hardware testing and debugging. For the software testing and debugging we have used KEIL software.

We divided our debugging and testing in different modules, which are as follows:

a) First of all, we developed assembly programming for KEIL software.
b) Then we developed programming of Arduino Uno to send the code of LCD, Serial communication.
c) Then we assign different ports for corresponding peripherals.
d) After connecting different devices to their corresponding ports, we burn the Arduino Uno with the help of Flash-magic software.
e) Implementation of hardware is carried out by affixing the components on the PCB and check the continuity of PCB with the help of multi meter.
f) Before implementing the IC on the respective place, we use IC tester to ensure that the IC is working properly or not.
g) The burnt Arduino Uno is placed in the circuit board.
h) Then we apply the power supply to our system to check whether the given code is working properly or not, message displayed on LCD.
i) We also used Rxd and Txd pin of Arduino Uno for serial communication. With the help of above given pin we connect max232 for making data in the format of pc compatibility.
j) After completion of all the connections and all the steps, we open HyperTerminal and check whether the message is displayed on HyperTerminal or not, if not, we check the code burned in Arduino Uno or check the connection between the systems.

1) Problem crops up

- Microcontroller at Arduino Uno was not providing the results or was giving errors
- The pot used in the LCD circuit for adjusting the brightness/contract of LCD display was not giving the appropriate results.

2) How problems are rectified

- The microcontroller was replaced by the Arduino Uno as the Arduino Uno does not have In System Programmer.

- Potentiometer was adjusted to a certain level to get the required contrast on LCD display.

3) Testing procedure

Once the hardware has been assembled, it is necessary to verify that, the design is correct & the prototype is built to the design drawing. This verification of the design is done by writing several small programs, beginning with the most basic program & building on the demonstrated success of each.

a) Crystal test:

The initial test is to ensure that, both, the crystal & the reset circuit are working. The micro-controller is inserted in the circuit, & the ALE pulse is checked, with an oscilloscope, to verify that the ALE frequency is 1/6th of the crystal frequency. Next, the reset button is pushed & all ports are checked to see that, they are in the high input state.

b) PCB testing:

The PCB was tested, by tracing the tracks from the net list & the artwork of the PCB. The errors in the artwork were eliminated while testing & after that it was given for PCB manufacturing.

The PCB was tested using the DMM & the continuity of the tracks was tested using the DMM, in the diode mode. The positive terminal was connected to the terminals of the other ICs to show the negligible resistance, if the track is continuous.

c) VISUAL TESTING:

- Polarities of all the components like capacitors, connectors etc. are checked.
- It is seen that all the IC sockets are soldered properly.

d) MULTIMETER TESTING:

- All the IC sockets and power supply are soldered and continuity is checked
- Also, VCC and GND voltage are checked.
- Voltages at all the pins of the microcontroller are checked with respect to ground.
- Values of all possible components are checked on multimeter.

e) Software testing:

While designing the software for our project, we considered the following points:

Firstly, in the software, all the ICs were initialized. After all the subroutines for each module were executed, they were displaying the proper results.

6. RESULT & CONCLUSION

6.1 RESULT: -

SR No.	Key	Device(demo)	Device(field)	Function
1	1	Bulb	Motor1	Turn on
2	2	Fan	Boiler	Turn on
3	3	Buzzer	Machinery	Turn on
4	4		Fan	Turn on
5	5		Motor2	Turn on
6	6		Lights	Turn on
7	7	Bulb	Motor1	Turn off
8	8	Fan	Boiler	Turn off
9	9	Buzzer	Machinery	Turn off
10	0		Fan	Turn off
11	*		Motor2	Turn off

12	#		Lights	Turn off

Table 8.1 function of keys

CONCLUSION

With the knowledge of new techniques in ‘Electronics’ we are able to make our life more comfortable. One such application of electronics is used in “Home Appliances Controlling using Android Mobile via Bluetooth” The approach we followed and which is explained in this project report is novel and has achieved the target of “Home Appliances Controlling using Android Mobile via Bluetooth” satisfying user needs and requirements.

Home Appliances Controlling using Android Mobile via Bluetooth is automatic versatile system. It can be implemented in industry, home, agricultural field, remote and hazardous applications. It provides the flexibility & system reliability with low cost as well as less maintenance. It provides remote access to the system to deliver service at any time of the day. With this system, we can control as well as monitor the devices at remote location.

The development of this project has shown how much hard work goes into the creation of a system. “Home Appliances Controlling using Android Mobile via Bluetooth” was a project based on microcontroller, due to which hardware requirement is reduced. Embarking of this project has helped us in developing a team spirit, patience and time management necessary for today's technical professionals.

Hence we can conclude that the required goals and objectives of our project have been achieved.

This project has built in us confidence that any problem can be solved with sheer determination, hard work and optimism. We feel that our product serves something good to this world and we like to present it before this prosperous world. By doing this project, we were better able to understand the various facets of doing an embedded system project which is emerging as one of the most 'in demand' technologies right now.

APPLICATIONS:

1. Industrial automation - This project can be used to control various devices in the Industry

2. Can also be used for security purpose after modification (we can control gate system or we can interface wireless camera and can control it using our mobile)

A) INDUSTRIAL CONTROL -

1. It is used to control or switch on any automatic process control machinery.
2. Electric grid could be controlled remotely.
3. Automatic production machinery could be controlled even during odd hours with your mobile phone.

B) HOME AUTOMATION –

1. To monitor status of home appliances.

2. To control (ON/OFF) the home appliances according to their status when we are going away from home.

C) FARM AUTOMATION -

1. To control & monitor the various irrigation equipment in the farm.

ADVANTAGES:

1. Can control device from a long distance, thus it gives ease of access.

2. Faster operation and efficient.

3. No need to carry separate remote or any other controlling unit.

- Eliminates the continuously monitoring, it facilitates 24 hours a day, 365 days in year communication between system and user.

- Commands can be given through remote place, directly to the machine.

- By further modification security system can be added.
- Easy to install & simple in operation.
- Low cost, high reliability & flexibility.

7.3 LIMITATIONS: -

- If the GSM network used in mobile does not have any coverage, then the operation cannot perform.
- It is not useful for large number of application but by using multiple keys for device operation we can increase number of applications.

7.4 FUTURE SCOPE

- Data logging facility can be included in case of recording historical data, special data, special events and system data.
- Computer can be interfaced for more complex & precise application.
- System reliability can be improved.
- Voice interactive services can be added to offer better interaction with user.

8. APPENDIX

8.1 COMPONENT LIST

Sr No.	Components	Specification	Quantity
1	Capacitor	22pf	2
		0.47uf	2
		33pf	2
2.	Resistors	Pull up resistor	2
		100Ω	6

		2.2k	6
		100k	2
		10k	6
3.	Diodes	1n4007	4
4.	Relays	12v dc	6
5.	IC sockets	40 pin	1
		16 pin	1
6.	ICs	Arduino Uno	1
		MAX232	1
7.	Transistors	BC547	6
8.	Crystal	11.0592MHz	1
		3.2MHz	1
9.	Connectors	2 pin connector	7
		3 pin connector	1
		LCD connector	1
10.	PCB	Glass epoxy	1
11.	Transformer	230-15v	1
12	Bluetooth receiver	Bluetooth receiver	1
13	LCD	16*2 char	

Table 13.1

component list

Websites:

- www.wikipedia.com

- www.google.com

- www.efy.com

- www.worldlogic .com

DATASHEETS

1. 89s51
2. LCD
3. 7805
4. BC547

Arduino Source code

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

int Buzzer      = 13;
int Relay1       = 8;
int Relay2       = 9;
int Relay3       = 10;
int Relay4       = 11;
int Relay5       = 12;
int LCDTempReg = 0;
char LCDmsgwelcome[]         = {"ARDUINO & BLUETOOTH"};
char LCDmsgwelcome_L2[]      = {"CONTROLLED DEVICES"};
char LCDmsgon[]              = {"DEVICE "};
char LCDmsgon_L2[]           = {"TURNED ON "};
char LCDmsgoff[]             = {"DEVICE "};
char LCDmsgoff_L2[]          = {"TURNED OFF "};
void disp_lcd_msg (char *, char *);
void setup ()
{
Pin Mode (Relay1, OUTPUT);
Pin Mode (Relay2, OUTPUT);
Pin Mode (Relay3, OUTPUT);
```

```
 pinMode (Relay4, OUTPUT);
 pinMode (Relay5, OUTPUT);
 pinMode (Buzzer, OUTPUT);
 lcd. begin (16, 2);
 disp_lcd_msg (LCDmsgwelcome, LCDmsgwelcome_L2);
 delay (2000);
 }
 void loop ()
 {
 unsigned char User Command, Device Number = 0;
 if (Serial. Available ())
 {
 User_Command = Serial. Read ();
 if (User_Command == 'A')
 {
  digital Write (Relay1, HIGH);
  Device Number = 1;
}
if(User_Command == 'a')
{
  digital Write (Relay1, LOW);
  Device Number = 1;
}
if(User_Command == 'B')
{
  digital Write (Relay2, HIGH);
  Device Number = 2;
}
if(User_Command == 'b')
{
  digital Write (Relay2, LOW);
  Device Number = 2;
}
if(User_Command == 'C')
{
  digital Write (Relay3, HIGH);
  Device Number = 3;
}
if(User_Command == 'c')
```

```
 {
   digital Write (Relay3, LOW);
   Device Number = 3;
 }
 if(User_Command == 'D')
 {
   digital Write (Relay4, HIGH);
   Device Number = 4;
 }
 if(User_Command == 'd')
 {
   digital Write (Relay4, LOW);
   device Number = 4;
 }
 if(User_Command == 'E')
 {
   digital Write (Relay5, HIGH);
   device Number = 5;
 }
 if(User_Command == 'e')
 {
   digital Write (Relay5, LOW);
   device Number = 5;
}
if(User_Command == 'F')
{
  digitalWrite(Buzzer, HIGH);
  Device_Number = 6;
}
if(User_Command == 'f')
{
  digital Write (Buzzer, LOW);
  device Number = 6;
}

if (device Number == 0)
{
        if (User_Command == 'A' && User_Command == 'B' && User_Command == 'C' &&
User_Command == 'D' && User_Command == 'E' && User_Command == 'F')
        {
          disp_lcd_msg(LCDmsgon,LCDmsgon_L2);
        }
        if(User_Command == 'a' && User_Command == 'b' && User_Command == 'c' &&
User_Command == 'd' && User_Command == 'e' && User_Command == 'f')
        {
          disp_lcd_msg(LCDmsgoff,LCDmsgoff_L2);
        }
        lcd.setCursor(8,0);
        lcd.print (device Number);
        device Number = 0;
        }
    }
}

 void disp_lcd_msg(char * LCD_Line_1, char * LCD_Line_2)
{
     lcd.print(LCD_Line_1);
     lcd.setCursor(1,0);
     lcd.print(LCD_Line_2);
}
```