ABSTRACT

ABSTRACT

It is difficult to arrange communication between blind and deaf person . The system Can be design & develop to help the communication between such physically impaired persons . And when creating a system, blind and deaf people should think about what language they know. The braille system has been used by the visually impaired for reading and writing. The Braille Lippi are pre -processed to enhance the dots and reduce the noise. The Braille cells are segmented and the dots from each cell is extracted and converted in to a number sequence. These are mapped to the appropriate alphabets of the language. There also provides a mechanism to type the Braille characters through the number pad of keyboard. The type Braille character is mapped to the alphabet and spoken out. Now a day's technology enhancement is reaching at personalized electronic assistance but we feel that there is a requirement for an electronic translator for physically challenge people. There is a specialized technique known as braille lipi by which they can read and communicate with the world as a realization for detecting or reading blind people depend on finger point by keeping hand on letter structure. In order to make them able to read the content of the message system has to make different pattern like Vibro sensor on the specific time interval based decoding and the deaf person can reading the content of which can presented on display .

Keywords : Blind , Deaf, Braille lipi, communication

Chapter 1

INTRODUCTION

Chapter 1

INTRODUCTION

1.1 Blind – Deaf People Communication System

There are difficult to do communication between blind and deaf person . There should be a system to communicate with them . And when creating a system, blind and deaf people should think about what language they know. So to create a System for their help.

Now a day's technology enhancement is reaching at personalize electronic assistance but we feel that there is a requirement for an electronic translator for physically challenge people. Among them, we are focusing on blind and deaf people.The Deaf person do communication with hand movement /with typing /with watching and .The Blind person do communication with hand movement /with voice / with Braille Lipi so There is a special technique known s braille lipi by using the this they can read and communicate with the world as a realization for detecting or reading blind people depend on finger point by keeping hand on letter structure.and the deaf person can read the data using to watching the display.

1.2 If Deaf Person do the Communication with Blind Person then how he can do?

Deaf person use typing or hand movement to communicate . He can type his words on typing board and send to display ; but blind person can not watch this display , the blind person do communication only with hand or voice so the word on display which can to present on Braille lipi format board of system which made by us and he can sense the using hand . The blind person sense the word using hand which presented on Braille lipi board.

1.3 If Blind Person Communicate with Deaf Person then how he can do ?

Blind people can communicate with voice or hand or their own language Of Braille lipi which used to writing ; but deaf people’s can do communication with watching word , picture or with typing so blind person can type word in their language of Braille Lipi and send to system

then it change into our alphabets this alphabets present on display..This display letters watch deaf person.the deaf person can read the word which can presented on display

1.4 What is Braille

The main part of system is braille lipi .This was derived in 1825 by Louis Braille, a blind Frenchman.each braille character or cell is made up of six dot position,arranged in a rectangle containing two column of three dots each a dot may be raised at any of six positions to form sixty four possible subset

.

1.5 What is braille cell

The Braille system comprises of a cell per character and these themselves consist of six embossed or raised dots arranged as three rows and two columns. These six cells allow 26 =64 characters to be represented, being split up into 26 letters. The Braille cell use world-wide. The different languages use different unique codes, mapping the alphabets, numbers and punctuation symbols to Braille cells according to NCD. Braille cell is represented as follows shown in figure

Braille cell International Journal of Research in Advent Technology,vol2,no4

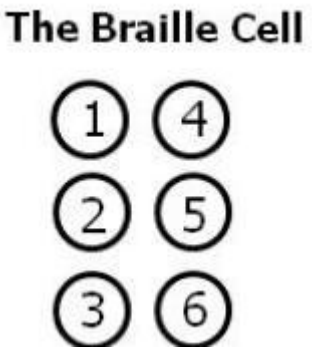


Fig. 1.1 : The braille cell

1.6 Braille lipi Alphabets :

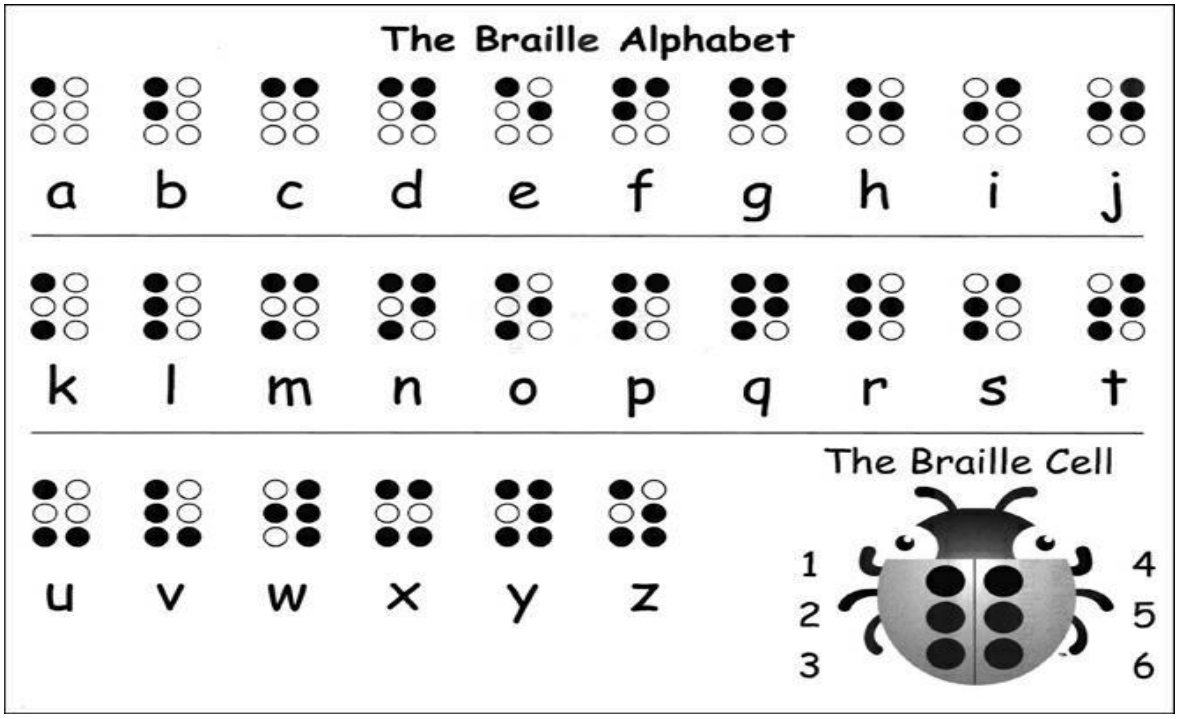


Fig. 1.2 : Braille lipi Alphabets

Need of project

The aim of this blind -deaf Communication system is that ,it can communicate with each other .the specialized technique for blind person is braille lipi and for deaf person is to watching . So to combine both things in one system to do communication between them . The system is very helpful to blind deaf person .

Using this system the they communicate as read , write the content of message .

- The price of a braille display in now day's market is very high.
- For traditional braille system blind people required pad,stylus and the scale which is very difficult to handle it.
- In the Braille keypad solenoids are used for reading purpose which is very costly and hence not affordable by blind people
- The deaf not listening sound they only watch and blind peole do not write so to do interface between this category peoples we need of this project.

Chapter 2

LITERATURE REVIEW

Chapter 2

LITERATURE REVIEW

Table2.1:-List of Literature Survey

SR. NO	NAME OF INVENTER	YEAR	INVENTION
1	**DAVID FISHER**	1992	**Developed low cost Braille hand glove**
2	**MASAFUMI NISHID**	1997	**Analysis of finger Braille using electromyography**
3	**ANUPAM BASU**	1998	**A Pc based Braille ibrary system for the sightless**
4	**M.M USLAN**	1990	**Electronic Braille display for personal computers**
5	**PAUL BLENK HORN**	2001	**Automated Braille production fro word processes documents**
6	**RAJESINHATIPATHI ARTHNARI**	2010	**Convert English text toBraille code vibrationsignal**

S.Padmavath:

A Braille translator a software program that translates script into Braille cells, and sends it to a Brailie embosser, which produces a hard copy in Braille scripts of the original text. Basically only the script is transformed, not the language

Manojnak.s:-

As an embedded system, a central processing s a micro controler. For easy communication between a normal person and blind person this system creates a bridge as an input a normal alphabetic and numeric keyboard is use for easy typng of the message the controller will decode the alphabetic information and it is converted into equalivent six point Alphabetic braille Lippi code

Chapter 3

BLOCK DIAGRAM

& EXPLANATION

Chapter 3

BLOCK DIAGRAM & EXPLANATION

3.1 BLOCK DIAGRAM :

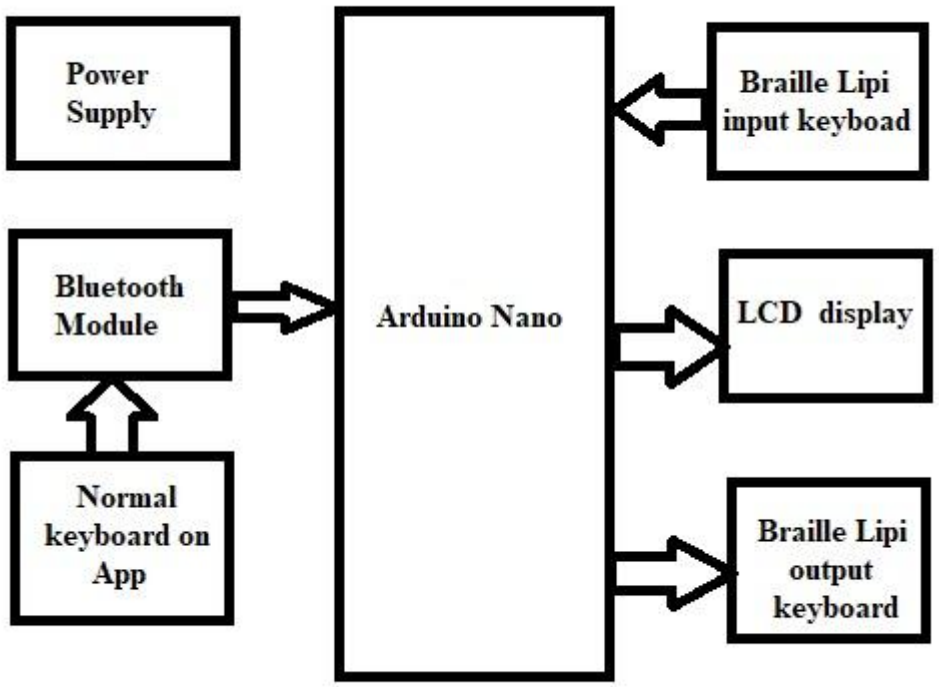


Fig. 3.1 : Block Diagram of Project

3.2 Explanation:

A. **Power supply :-** The system operate on 5 volt dc supply .so need a 5 volt dc power to operate the System . This 5 volt supply is give components of Arduino Nano , Bluetooth module , vibro sensor , switch .

B. **User Input Braille lipi Keyboard:- T**his keyboard is built according to the braille lipi .on this key board the blind person can type their word as they write using Braille lipi . This computer is give to arduino nano for con tolling action .

C. **User Normal Keyboard On Mobile :-** This keyboard present on mobile application the deaf person use this keyboard to do communication they can type that they want to say and this letter can send to Bluetooth module for to connect mobile application to Arduino Nano.

D. **Bluetooth module :- T**his is used to connect the mobile application to arduino .It used to do interface between mobile and arduino nano . The Bluetooth module is receive data from the mobile App and this data can present on display using the controlling action of Arduino Nano.

E. **Arduino Nano:-** It is a controller to control the action of system . It can receive signal from Braille lipi keyboard and from Bluetooth module is connected to mobile application .this data can control and the output can shown on display. And output is also present on vibro sensor as keyboard of braille lipi keyboard.

F. **Display :-** it is used to display the letters what can blind . it receive the signal from arduino nano It display letter which sensed by arduino nano . In Arduino where the display command used there display is used to present the data .

G. **User Output Braille Lipi Keyboard using Vibro Sensor :- T**his keyboard built as braille lipi . On this keyboard the the vibro motor are place like as braille lipi the arduino send signals to vibrator.As we change inputs that according the arduino do controlling action give to single to vibrator for present output .

Chapter 4

CIRCUIT DIAGRAM &

EXPLAINATION

Chapter 4

4.1 CIRCUIT DIAGRAM

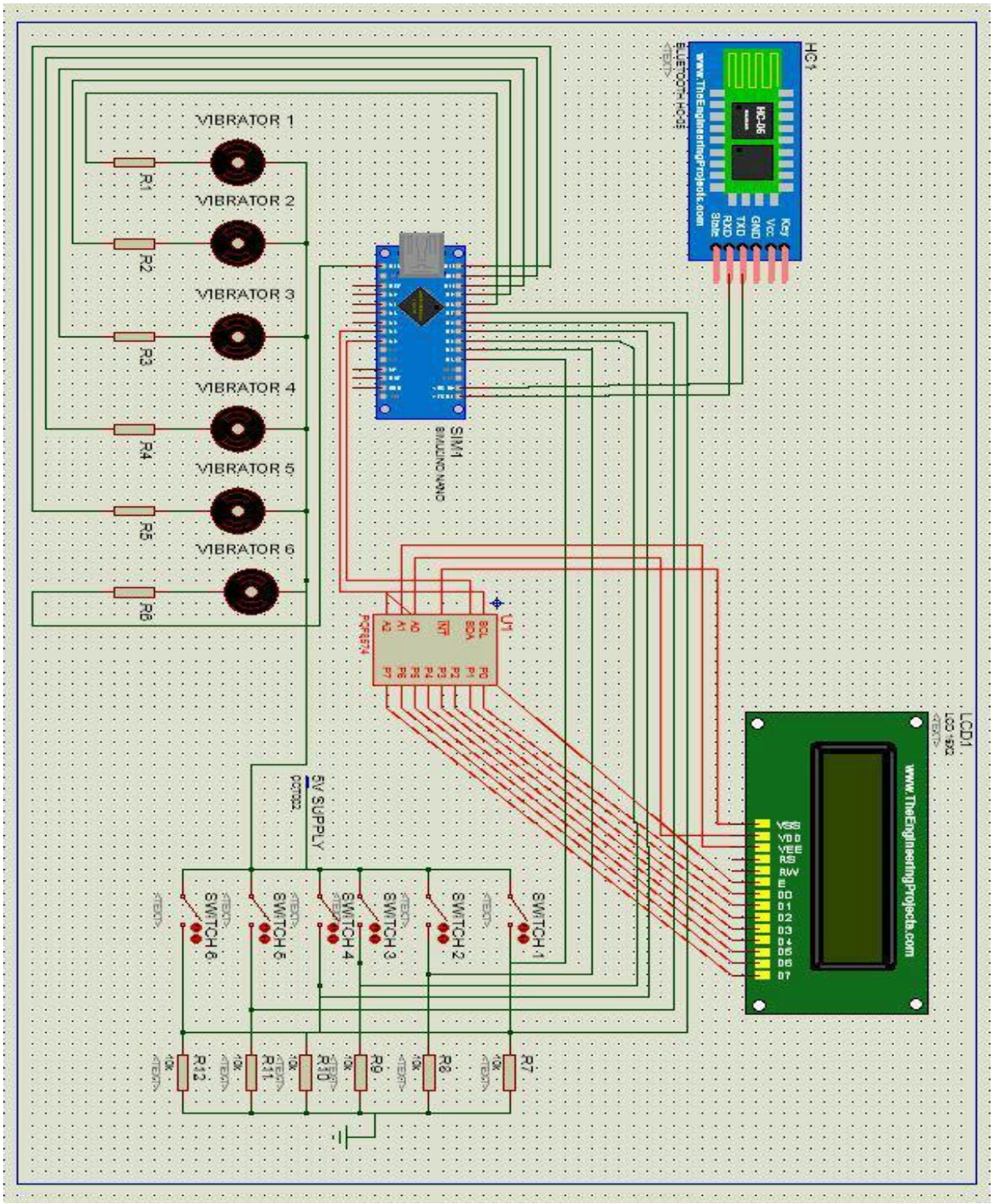


Fig. 4.1 : Circuit Diagram of Project

4.2 EXPLAINATION:-

The main part of system is controller so We used the arduino nano to build this system in arduino nano the ATmega328 IC used as controller . There are sufficient connection required to build the system so arduino nano is used.this arduino nano connected to 5 volt dc supply. The digital pin of arduino from D2 to D7 is connected to input of switch which used to create a braille lipi input keyboard. This individual pin of nano is connected to individual switch . Other terminal of switch is connect to Vcc with pull down resistor.

And there are six vibrator is used to build the Braille lipi output keyboard . 6 vibrator of six negative terminal are connected to digital pin of nano D8 to D13 .as input of D2 toD7 Changes that according the output of D8 to D13 are changed .

The pin of TXD of arduino nano is connected to RXD pin of Bluetooth module. And the pin of RXD of arduino nano is connected to TXD pin of Bluetooth module. And Bluetooth module also have connection of VCC and GND.

The display is connected to this system the 16×2LCD display connected to IIC module which module pin of SDA and SCL is connected to arduino pin of A4 and A5 . And IIC also have two connection VCC and GND.

Chapter 5

HARDWARE DESCRIPTION

Chapter 5

HARDWARE DESCRIPTION

Hardware Components:-

1. Arduino Nano
2. LCD Display 16×2
3. I2C Modules
4. Bluetooth Module
5. Push button switches
6. Vibro Motor
7. Resistor
8. Capacitor
9. 7805 Regulator voltage IC
10. Diode
11. Connecting wires
12. Flat wire
13. LED
14. Transformer

5.1.Arduino Nano At mega 328 :-

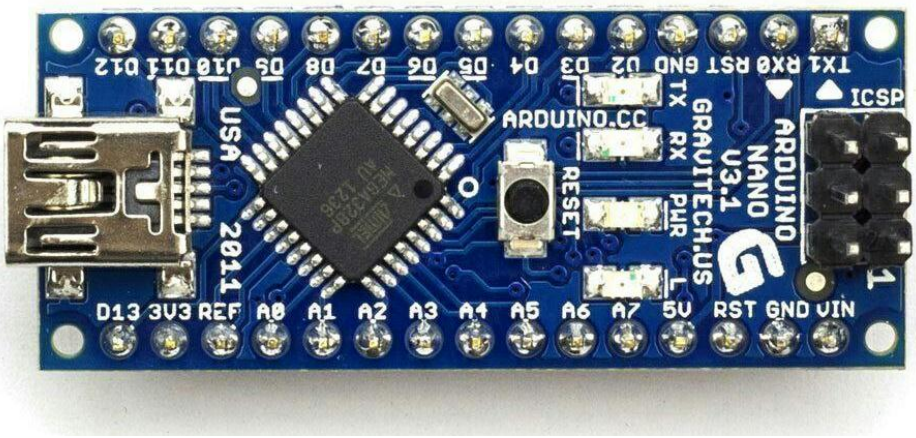


Fig. 5.1 : Arduino Nano At mega 328

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package.It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

Technical specifications :

Micro controller	-	Atmel ATmega168 or ATmega328
Operating Voltage(logic level)	-	5 V
Input Voltage(recommended)	-	7-12 V
Input Voltage(limits)	-	6-20 V
Digital I/O Pins	-	14 (of which 6 provide PWM output)
Analog Input Pins	-	8
DC Current per I/O Pin	-	40 mA

Flash Memory	-	16 KB (ATmega168) or 32 KB(ATmega328) of which2 KB used by boot loader
SRAM	-	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM (ATmega328)	-	512 bytes (ATmega168) or 1 KB
Clock Speed	-	16 MHz

Input and Output:-

Each of the **14 digital pins on the Nano** can be used as an input or output, using pinMode(), digital write(), and digital read() 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 k Ohms. In addition, some pins have specialized functions: Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach interrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog write() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

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.

The Nano has 8 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 analogReference() function. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

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

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

Programming:-

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the micro controller on your board). For details, see the reference and tutorials.

The ATmega168 or ATmega328 on the Arduino Nano comes pre burned with a boot loader 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 boot loader and program the micro controller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

Communication :-

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other micro controllers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LED's on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Nano's digital pins.

The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega168 or ATmega328 data sheet.

Automatic(software) Reset:-

Rather then requiring a physical press of the reset button before an upload, the Arduino Nano 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 FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nano farad 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 boot loader 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 Nano 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 boot loader is running on the Nano. 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.

5.2.Bluetooth Module HC -05

HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with controller or PC. HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data.

Fig. 5.2 : Bluetooth Module HC -05

Specification:

- Model : HC-05
- Input Voltage : DC 5V
- Communication Method : Serial Communication
- Master and slave mode can be switched

Pin description :

- VCC - Connect to +5V
- GND- Connect to Ground
- TXD - Connect with the MCU's (Micro controller and etc) RXD PIN.
- RXD - Connect with the MCU's (Micro controller and etc) TXD PIN.
- KEY - If it is input low level or connect to the air, the module is at paired or communication mode If it's input high level, the module will enter to AT mode.

5.3 LCD Display 16×2

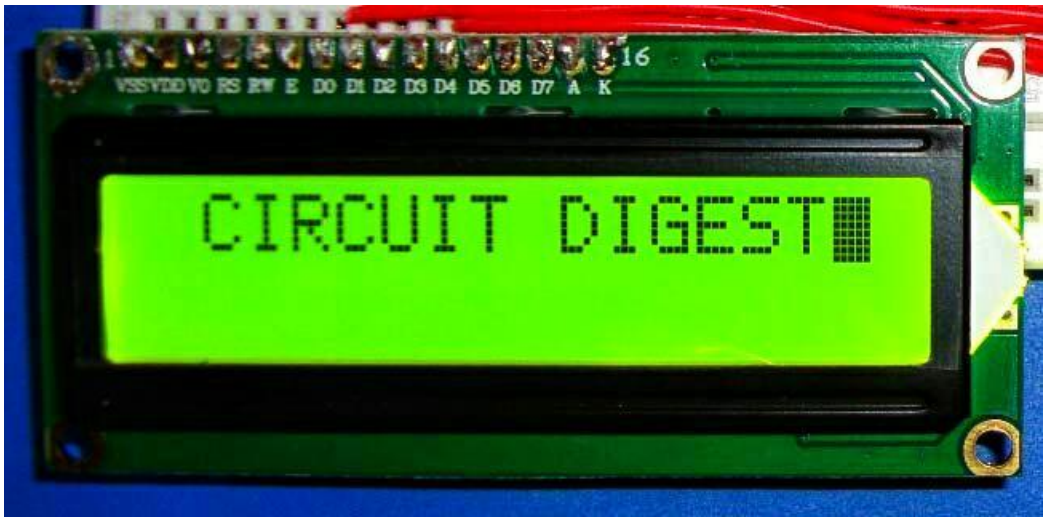
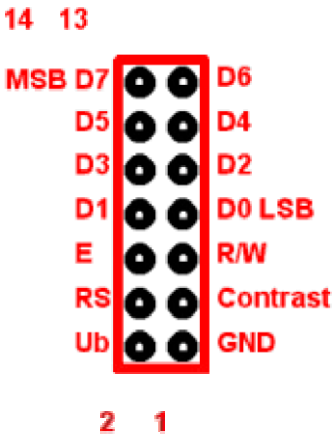


Fig .5.3 : LCD Display 16×2

The Dot matrix liquid crystal display controller and liver LSI displays alphanumeric , characters and symbol . it can be configured to drive a dot - matrix liquid crystal display under the control of bit 8 or 14 bit microprocessor , Since all the function such as display RAM . Character generator and liquid crystal driver , required for driving A dot - matrix liquid crystal display are internally provided one chip a minimal system can be Interfaced with us controller HD44780U can Display up two 8 – character lines (16 * 2)

16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots.



Pin#	Name		In/output
1	GND	Ground	Power
2	VCC	LCD Controller Power (+3 to +5V)	Power
3	VLCD	LCD Display Bias (+5 to -5V *see text)	Analog
4	RS	Register Select: H: Data L: Command	Input
5	R/W	H: Read L: Write	Input
6	E	Enable (Data strobe, active high)	Input
7	DB0	Data LSB	I/O
8	DB1	Data	I/O
9	DB2	Data	I/O
10	DB3	Data	I/O
11	DB4	Data	I/O
12	DB5	Data	I/O
13	DB6	Data	I/O
14	DB7	Data MSB	I/O
15	A	LED Backlight Anode (optional)	Power
16	K	LED Backlight Cathode (optional)	Power

Table5.1:-List of Pin Function

5.4.I2C Module

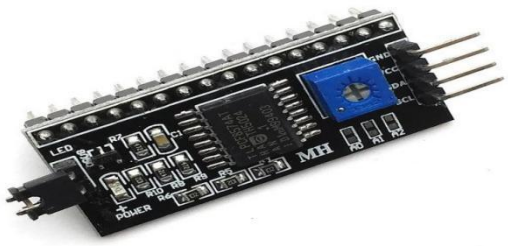


Fig. 5.4 :I I2C Module

I2C is a serial protocol for two-wire interface to connect low-speed devices like micro controllers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems. It was invented by Philips and now it is used by almost all major IC manufacturers. Each I2C slave device needs an address – they must still be obtained from p (formerly Philips semiconductors). I2C uses only two wires: SCL (serial clock) and SDA (serial data). Both need to be pulled up with a resistor to +VDD. There are also I2C level shifters which can be used to connect to two I2C buses with different voltages. Most I2C devices support repeated start condition. This means that before the communication ends with a stop condition, master device can repeat start condition with address byte and change the mode from writing to reading.

5.5.Vibro Motor :-

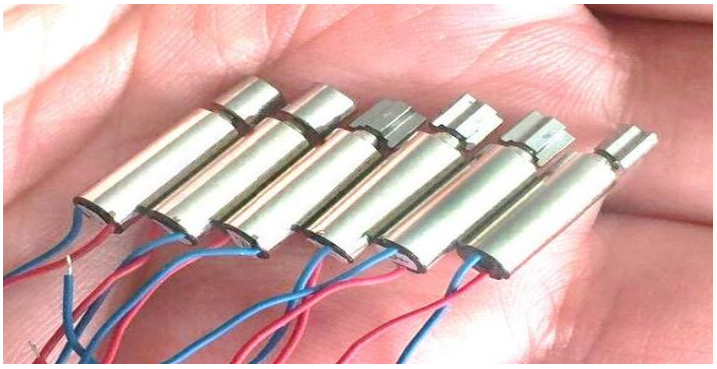


Fig 5.5.1: Vibro Motor

Vibrator

The model number 306-101 fully defines the model, variant and additional features of the Product Eccentric Rotating Mass vibration motors, also commonly known as ERMs or pager motors, make up our main product lines and have been our core business since Precision Micro drives was founded. Miniature DC vibration motors have the benefit of being easy to implement and are low cost, whilst dramatically augmenting how devices interact with users. All pager motors use miniature Eccentric Rotating Masses, and hence they tend to be abbreviated 'ERM motors'. ERM pager motors rely on the rotation of an unbalanced load to create vibration effects and are more recently being used for haptic feedback as well as vibration alerting.

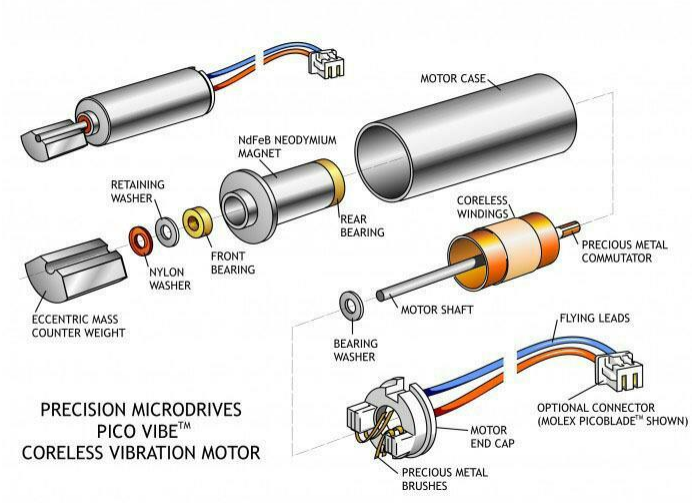


Fig. 5.5.2 :Internal Structure of Vibro

Features	
Body Diameter	6 mm
Body Length	13.3 mm
Typical Operating Current	140 mA
Typical Power Consumption	420 mW
Typical Normalised Amplitude	3.4 G
Rated Voltage	3 V
Rated Speed	11,500 rpm
Lead Length	100 mm
Lead Wire Gauge	32 AWG

Our Pico Vibe™ range is designed to be used for hand held equipment in the weight range of around 25 ~ 200 g (1 - 7 oz.). Typically based on DC motors up to 6mm (although some are slightly larger, especially when encapsulated in housing), they normally range from 1.5V to 3V Rated Voltage - which makes them ideal for battery power supplies.

5.6.Push button

Fig 5.6 : Push button

A push-button (also spelled push button) or simply button is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal.[1] The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, although many UN-biased buttons (due to their physical nature) still require a spring to return to their UN-pushed state. Terms for the "pushing" of a button include pressing, depressing, mashing, slapping, hitting, and punching.

Specification :

Current Rating	:	3A 125VAC; 1.5A 250VAC
Insulation Resistance	:	100 Megohms (min.) at 500VDC
Dielectric Strength	:	1000V RMS (min.), 1 minute
Temperature Rating	:	−13 to +185 F (−25 to +85 C)
Electrical Life	:	6000 cycles

Mechanical Life : 100,000 cycles

Terminal Type : Solder Lug

Mounting Hole : 500 (12.7)

5.7.Power supply

In most of electronic circuit required AC voltage regulated DC Supply .for this system required a 5 volt DC supply .for to implement this 5 volt power supply the 5 volt circuit required .

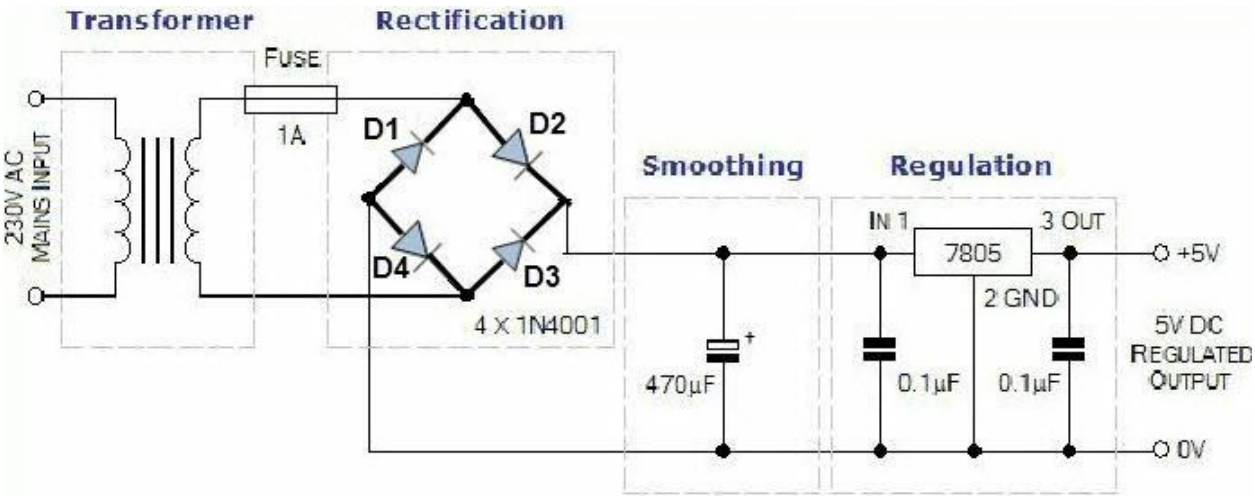


Fig 5.7 : Power supply ckt

Component list :

i. Step down transformer

ii. Rectifier

iii. Filter

iv. Regulator voltage

Transformer:

A step down transformer is used to step down the voltage from the input AC to the required voltage of the electronic device. This output voltage of the transformer is customized by changing

the turns ratio of the transformer according the electronic device specs. The input of the transformer being 230 Volts AC mains, the output is provided to a full bridge rectifier circuit.

Full Wave Rectifier Circuit :

The FWR consists of 4 diodes which rectifies the output AC voltage or current from the transistor to its equivalent DC quantity. As the name implies the FWR rectifies both half's of the AC input. The rectified DC output is given as input to the filter circuit. The diode 1N4001 is used for rectification .it acts on both cycle.

Filter Circuit:

The filter circuit is used to convert the high rippled DC output of the FWR to ripple free DC content. A π filter or RC filter used to make the wave forms ripple free The ac voltage, typically 230 Vrms is connected to a transformer which transforms that ac voltage to the level for the desired dc output. A bridge rectifier then provides a full-wave rectified voltage that is initially filtered by a $\prod$ (or C-L-C) filter to produce a dc voltage. The RC circuit is used for filtration operation. The resulting dc voltage usually has some ripple or ac voltage variation so regulator voltage circuit used.

Regulations :

A regulating circuit use this dc input to provide a dc voltage that not only has much less ripple voltage but also remains constant even if the input dc voltage varies somewhat or the load connected to the output dc voltage changes. The regulated dc supply is available across a voltage divider. Often more than one dc voltage is required for the operation of electronic circuits. A single power supply can provide as many as voltages as are required by using a voltage (or potential) divider a potential divider is a single tapped resistor connected across the output terminals of the supply. The tapped resistor may consist of two or three resistors connected in series across the supply. In fact, a bleeder resistor may also be employed as a potential divider. The Ic 7805 Regulated Ic used for get a fixed 5 volt dc voltage .

8. Diode :

Fig. 5.8 :Diode

A diode is a two-terminal electronic component that conducts current primarily in one direction. is used to convert alternating current (ac) to direct current (dc). Forms of rectifiers, diodes can be used for such tasks as extracting modulation from radio signals in radio receivers.

9.Resister :

Fig. 5.9 : Resister

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can dissipate many wattsof electrical power as heat, may be used as part of motor controls, in power distribution systems, or as test loads for generators.

10. Capacitor

Fig. 5.10 : Capacitor

Capacitors contain at least two electrical conductors often in the form of metallic plates or surfaces separated by a dielectricmedium. When two conductors experience a potential difference, for example, when a capacitor is attached across a battery, an electric fielddevelops across the dielectric, causing a net positive charge to collect on one plate and net negative charge to collect on the other plate.

11.IC 7805 :

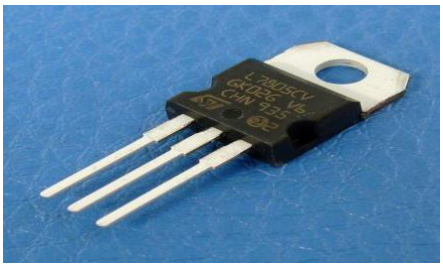


Fig. 5.11 :IC 7805

7805 is a family of self-contained fixed linear voltage regulator integrated circuits. The 7805 family is commonly used in electronic circuits requiring a regulated power supply due to their ease-of-use and low cost.

12. LED :

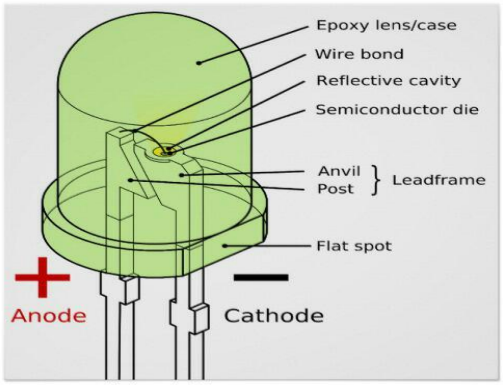


Fig. 5.12 : LED

A light-emitting diode (LED) is asemiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons

Chapter 6

SOFTWARE DESCRIPTION

Chapter 6

SOFTWARE DESCRIPTION

6.1 Software Description

The main part of system is programming so for to built the system we use the Arduino Nano . Arduino Nano is a micro controller board designed by Arduino.cc. The micro controller used in the Arduino Nano is Atmega328, the same one as used in Arduino UNO. It has a wide range of applications and is a major micro controller board because of its small size and flexibility. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The micro controller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP)

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software.

Developer	Arduino software
Stable release	1.8.8. 17 December 2018
Written in	Java ,c ,c++
Operating system	Windows , Linux, masos
Platform	A-32 , X86-64 ,ARM
License	LGPL ,GPL License

6.2 Program :-

```
PinMode(6,INPUT);
PinMode(7,INPUT);
PinMode(8,OUTPUT);
PinMode(9,OUTPUT);
PinMode(10,OUTPUT);
PinMode(11,OUTPUT);
PinMode(12,OUTPUT);
PinMode(13,OUTPUT);
Serial.begin(9600);
#include<software serial.h>
#include<liquid crystal.h>
PinMode(rx pin,input);
PinMode(txpin, output);
char=data
Void setup()
{
PinMode(2,INPUT);
PinMode(3,INPUT);
PinMode(3,INPUT);
PinMode(4,INPUT);
PinMode(5,INPUT);
lcd.begin(16,2);
}
Void loop()
{ data=serial.read();
lcd.write(data);
serial.write(data);
if(data=='a')
{digital write(2,high);
digital write(3,low);
digital write(4,low);
digital write(5,low);
digital write(6,low);
digital write(7,low);
digital write(8,low);
digital write(9,high);
digital write(10,high);
digital write(11,high);
digital write(12,high);
digital write(13,high);
delay (1000)
}
if(data=='b')
{digital write(2,high);
digital write(3,high);
digital write(4,low);
digital write(5,low);
digital write(6,low);
digital write(7,low);
digital write(8,low);
digital write(9,low);
digital write(10,high);
digital write(11,high);
digital write(12,high);
digital write(13,high);
delay (1000)
}
if(data=='c')
{digital write(2,high);
digital write(3,low);
digital write(4,low);
digital write(5,high);
digital write(6,low);
```

```
digital write(7,low);        if(data=='e')                    digital write(10,high);
 digital write(8,low);              {digital write(2,high);          digital write(11,low);
digital write(9,high);               digital write(3,low);            digital write(12,high);
digital write(10,high);              digital write(4,low);            digital write(13,high);
digital write(11,low);               digital write(5,low);            delay (1000)
digital write(12,high);              digital write(6,high);                                  }
digital write(13,high);              digital write(7,low);       if(data=='g')
delay (1000)                         digital write(8,low);            {digital write(2,high);
                              }      digital write(9,high);           digital write(3,high);
if(data=='d')                        digital write(10,high);          digital write(4,low);
      {digital write(2,high);        digital write(11,high);          digital write(5,high);
digital write(3,low);                digital write(12,low);           digital write(6,high);
digital write(4,low);                digital write(13,high);          digital write(7,low);
digital write(5,high);               delay (1000)                     digital write(8,low);
digital write(6,high);                                    }           digital write(9,low);
digital write(7,low);              if(data=='f')                      digital write(10,high);
digital write(8,low);                {digital write(2,high);         digital write(11,low);
digital write(9,high);               digital write(3,high);           digital write(12,low);
digital write(10,high);              digital write(4,low);            digital write(13,high);
digital write(11,low);               digital write(5,high);           delay (1000)
digital write(12,low);               digital write(6,low);                                   }
digital write(13,high);              digital write(7,low);       if(data=='h')
delay (1000)                         digital write(8,low);            {digital write(2,high);
                              }      digital write(9,low);            digital write(3,high);
```

```
        digital write(4,low);              digital write(13,high);          digital write(7,low);
        digital write(5,low);              delay (1000)                    digital write(8,low);
        digital write(6,high);                                    }        digital write(9,high);
        digital write(7,low);      if(data=='j')                           digital write(10,low);
        digital write(8,low);             {digital write(2,low);           digital write(11,high);
        digital write(9,low);              digital write(3,high);          digital write(12,high);
        digital write(10,high);            digital write(4,low);           digital write(13,high);
        digital write(11,high);            digital write(5,low);           delay (1000)
        digital write(12,low);             digital write(6,low);                                  }
        digital write(13,high);            digital write(7,low);      if(data=='a')
        delay (1000)                       digital write(8,high);          {digital write(2,high);
                                  }        digital write(9,low);           digital write(3,high);
if(data=='i')                              digital write(10,high);         digital write(4,high);
       {digital write(2,low);              digital write(11,low);          digital write(5,low);
        digital write(3,high);             digital write(12,low);          digital write(6,low);
        digital write(4,low);              digital write(13,high);         digital write(7,low);
        digital write(5,high);            delay (1000)                    digital write(8,low);
        digital write(6,low);                                     }        digital write(9,low);
        digital write(7,low);      if(data=='k')                           digital write(10,low);
        digital write(8,high);            {digital write(2,high);          digital write(11,high);
        digital write(9,low);              digital write(3,low);           digital write(12,high);
        digital write(10,high);            digital write(4,high);          digital write(13,high);
        digital write(11,low);             digital write(5,low);           delay (1000)
        digital write(12,high);            digital write(6,low);                                  }
```

```
if(data=='m')                    digital write(10,low);          digital write(4,high);
      {digital write(2,high);    digital write(11,low);          digital write(5,high);
       digital write(3,low);     digital write(12,low);          digital write(6,low);
       digital write(4,high);    digital write(13,high);         digital write(7,low);
       digital write(5,high);    delay (1000)                    digital write(8,low);
       digital write(6,low);                  }                 digital write(9,low);
       digital write(7,low);     if(data=='o')                   digital write(10,low);
       digital write(8,low);     {digital write(2,high);         digital write(11,low);
       digital write(9,high);    digital write(3,low);           digital write(12,high);
       digital write(10,low);    digital write(4,high);          digital write(13,high);
       digital write(11,low);    digital write(5,low);           delay (1000)
       digital write(12,high);   digital write(6,high);                    }
       digital write(13,high);   digital write(7,low);     if(data=='q')
      delay (1000)               digital write(8,low);           {digital write(2,high);
                  }              digital write(9,high);          digital write(3,high);
if(data=='n')                    digital write(10,low);          digital write(4,high);
      {digital write(2,high);    digital write(11,high);         digital write(5,high);
       digital write(3,low);     digital write(12,low);          digital write(6,high);
       digital write(4,high);    digital write(13,high);         digital write(7,low);
       digital write(5,high);    delay (1000)                    digital write(8,low);
       digital write(6,high);                 }                 digital write(9,low);
       digital write(7,low);     if(data=='p')                   digital write(10,low);
       digital write(8,low);     {digital write(2,high);         digital write(11,low);
       digital write(9,high);    digital write(3,high);          digital write(12,low);
```

```
        digital write(13,high);          digital write(7,low);      if(data=='u')
        delay (1000)                     digital write(8,high);          {digital write(2,high);
                                         digital write(9,low);            digital write(3,low);
if(data=='r')                            digital write(10,low);           digital write(4,high);
        {digital write(2,high);          digital write(11,low);           digital write(5,low);
        digital write(3,high);           digital write(12,high);          digital write(6,low);
        digital write(4,high);           digital write(13,high);          digital write(7,high);
        digital write(5,low);            delay (1000)                    digital write(8,low);
        digital write(6,high);                                           digital write(9,high);
        digital write(7,low);      if(data=='t')                         digital write(10,low);
        digital write(8,low);            {digital write(2,low);          digital write(11,high);
        digital write(9,low);            digital write(3,high);          digital write(12,high);
        digital write(10,low);           digital write(4,high);          digital write(13,low);
        digital write(11,high);          digital write(5,high);          delay (1000)
        digital write(12,low);           digital write(6,high);                          }
        digital write(13,high);          digital write(7,low);      if(data=='v')
        delay (1000)                     digital write(8,high);          {digital write(2,high);
                                         digital write(9,low);            digital write(3,high);
if(data=='s')                            digital write(10,low);           digital write(4,high);
        {digital write(2,low);           digital write(11,low);           digital write(5,low);
        digital write(3,high);           digital write(12,low);           digital write(6,low);
        digital write(4,high);           digital write(13,high);          digital write(7,high);
        digital write(5,high);           delay (1000)                    digital write(8,low);
        digital write(6,low);                                            digital write(9,low);
```

```
        digital write(10,low);
        digital write(11,high);
        digital write(12,high);
        digital write(13,low);
       delay (1000)
                        }
if(data=='w')
       {digital write(2,low);
        digital write(3,high);
        digital write(4,low);
        digital write(5,high);
        digital write(6,high);
        digital write(7,high);
        digital write(8,high);
        digital write(9,low);
        digital write(10,high);
        digital write(11,low);
        digital write(12,low);
        digital write(13,low);
        delay (1000)
                        }
if(data=='x')
       {digital write(2,high);
        digital write(3,low);
        digital write(4,high);
        digital write(5,high);
        digital write(6,low);
        digital write(7,high);
        digital write(8,low);
        digital write(9,high);
        digital write(10,low);
        digital write(11,low);
        digital write(12,high);
        digital write(13,low);
        delay (1000)
                        }
if(data=='y')
       {digital write(2,high);
        digital write(3,low);
        digital write(4,high);
        digital write(5,high);
        digital write(6,high);
        digital write(7,high);
        digital write(8,low);
        digital write(9,high);
        digital write(10,low);
        digital write(11,low);
        digital write(12,high);
        digital write(13,low);
        delay (1000)
                        }
if(data=='z')
       {digital write(2,high);
        digital write(3,low);
        digital write(4,high);
        digital write(5,low);
        digital write(6,high);
        digital write(7,high);
        digital write(8,low);
        digital write(9,high);
        digital write(10,low);
        digital write(11,high);
        digital write(12,low);
        digital write(13,low);
        delay (1000)
                        }
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==0;
        digital Read(5)==0;
        digital Read(6)==0;
        digital Read(7)==0;
```

```
        {lcd.println("a")}
If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==0;
        digital Read(5)==0;
        digital Read(6)==0;
        digital Read(7)==0;
        {lcd.println("b")}
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==0;
        digital Read(5)==1;
        digital Read(6)==0;
        digital Read(7)==0;
        {lcd.println("c")}
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==0;
        digital Read(5)==1;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("d")}
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==0;
        digital Read(5)==0;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("e")}
If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==0;
        digital Read(5)==1;
        digital Read(6)==0;
        digital Read(7)==0;
        {lcd.println("f")}
If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==0;
        digital Read(5)==1;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("g")}

If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==0;
        digital Read(5)==0;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("h")}

If(digitalRead(2)==0;
        digital Read(3)==1;
        digital Read(4)==0;
        digital Read(5)==1;
        digital Read(6)==0;
        digital Read(7)==0;
        {lcd.println("i")}
If(digitalRead(2)==0;
        digital Read(3)==1;
        digital Read(4)==0;
        digital Read(5)==1;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("j")}

If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==1;
        digital Read(5)==0;
        digital Read(6)==0;
```

```
        digital Read(7)==0;
        {lcd.println("k")}

If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==1;
        digital Read(5)==0;
        digital Read(6)==0;
        digital Read(7)==0;
        {lcd.println("l")}
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==1;
        digital Read(5)==1;
        digital Read(6)==0;
        digital Read(7)==0;
        {lcd.println("m")}

If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==1;
        digital Read(5)==1;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("n")}
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==1;
        digital Read(5)==0;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("o")}
If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==1;
        digital Read(5)==1;
        digital Read(6)==0;
        digital Read(7)==0;
        {lcd.println("p")}
If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==1;
        digital Read(5)==1;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("q")}
If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==1;
        digital Read(5)==0;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("r")}
If(digitalRead(2)==0;
        digital Read(3)==1;
        digital Read(4)==1;
        digital Read(5)==1;
        digital Read(6)==0;
        digital Read(7)==0;
        {lcd.println("s")}
If(digitalRead(2)==0;
        digital Read(3)==1;
        digital Read(4)==1;
        digital Read(5)==1;
        digital Read(6)==1;
        digital Read(7)==0;
        {lcd.println("t")}
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==1;
        digital Read(5)==0;
        digital Read(6)==0;
```

```
        digital Read(7)==1;
      {lcd.println("u")}
If(digitalRead(2)==1;
        digital Read(3)==1;
        digital Read(4)==1;
        digital Read(5)==0;
        digital Read(6)==0;
        digital Read(7)==1;
      {lcd.println("v")}

If(digitalRead(2)==0;
        digital Read(3)==1;
        digital Read(4)==0;
```

```
        digital Read(5)==1;
        digital Read(6)==1;
        digital Read(7)==1;
      {lcd.println("w")}
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==1;
        digital Read(5)==1;
        digital Read(6)==0;
        digital Read(7)==1;
      {lcd.println("x")}
If(digitalRead(2)==1;
        digital Read(3)==0;
      {lcd.println("z")}
```

```
        digital Read(4)==1;
        digital Read(5)==1;
        digital Read(6)==1;
        digital Read(7)==1;
      {lcd.println("y")}
If(digitalRead(2)==1;
        digital Read(3)==0;
        digital Read(4)==1;
        digital Read(5)==0;
        digital Read(6)==1;
        digital Read(7)==1;
```

Chapter 7

RESULT

Chapter 7

RESULT

Deaf person use normal typing keyboard present on mobile application to communicate . He can type his words on typing board and send to display ; but blind people can not watch this display , blind person do communication only with hand or voice so the word on display which can to present on Braille lipi format board of system which make by we and he can sense the using hand .the vibrator is placed.hence this vibrate according structure of Braille lipi hence Blind person communicate with deaf person.

In other side Blind people do communication with voice or hand or their own language. Of Braille lipi which used to writing ; but deaf people's can do communication with watching word , picture or with typing .the blind person can type word in their language of Braille lipi and send to system then it change into our alphabets this alphabets present on display.this display letters watch deaf person hence blind to deaf communication take place.

Fig . 7.1 : Project Demo photo 1

Fig . 7.1 : Project Demo photo 2

We tested this project in National Association for Blind (NAB) School of ITI Nashik.as shown in picture. In this school we tested the project in actual .in the picture the boy sat on chair is Deaf boy and the boy is standing is blind boy.. we have given this system to that students as using this system they we and the teacher from bind school observed that communication is successfully tack place.

Chapter 8

ADAVAVTAGES, DISADVANTAGE,

& APPLICATION

chapter 8

ADAVAVTAGES, DISADVANTAGE,& APPLICATION

8.1 Advantages:-

1. As using this system the the difficulty of communication between blind &deaf person is solved out.
2. Portable personal assistance for physically challenged persons.
3. It is easy carrying system .

8.2 Disadvantage:-

1. It is slow system.
2. Blind person can know the Braille language .

8.3 Application:-

1. This system generally for deaf &blind person to do Communication between them..
2. The blind deaf school also used this for giving teaching to blind and deaf person..

8.4 Future Scope:-

1. In this system we can added the speaker system .
2. Also we can add a memory to stored this letter after specific key this letters to make a word. this word can shown on display and using dictionary word make a proper .

Chapter 9

CONCLUSION

Chapter 9

CONCLUSION

By considering above said that we are conclude that , this system will help to physically challenged people like blind and deaf persons face many difficaltys while communicaton .to overcome this diifuculty using that system devlodped to communicate with written way as use this system they type word by digitally one by one alphabet and the blind person sense the this wor1d by one by one alphabet sensing the vibration of motors and other side the blind person type the word by Braille lipi and deaf person can watch this word on display. This system digitally implemented for communication .

CHAPTER 10

LIST OF COMPONENTS

Sr.No	Component name	Quantity	Cost
1	Arduino Nano	1	390
2	Bluetooth Module	1	450
3	LCD display 16×2	1	290
4	I2C module	1	190
5	Vibrator	6	1500
6	Push Switch	6	240
7	Transformer	1	60
8	Diode	4	10
9	IC7805	1	10
10	Resistors	15	10
11	Capacitors	2	10
12	LED	2	5
13	Connecting wire	1	20
14	Band wire	1	50
15	2 PIN connector	1	10
16	Zero PCB	1	50
17	Switch board	1	75
Total			3,370

Chapter 11

REFERENCES

Chapter 11

REFERENCES

- http://www.atmel.com
- https://www.wikipedia.org
- http://w:protoelectronics.com
- htlp://www.ijareeie.com
- http://www.howstuffworks.com
- http://www.alldatasheet.com
- Braille Translation Program by Curtin University of Technology Department of Electrical and
- Computer Engineering.
- Intenational Journal of Computer Applications (0975-8887) volume 4-no.6,jully 2010.
- IEEE paper of Enabling the disabled with translation of source text to Braille"
- J. K. Depress and D. M. B. Baumann," Towards making Braille as accessible as print," Dept. of Mech. Engrg., Massachusetts Institute of Technology, Cambridge, Mass., Rept. DSR 702491 June 19, 1968.
- J. Y. Gill. "Electromechanical tape brailler," S.B. thesis, Dept. of Elec. Engrg., Massachusetts Institute of Technology, Cambridge, Mass., June 1965
- M. Lichtenstein, "The MEDCOMP Brail program," in Computer Work for the Blind. College of Medicine, University of Cincinnati, Cincinnati, pp. 23-27, September 1963.
- J. Boyer, "Brailletran: A comprehensive Braille transcription program," 1966 Proc. Braille ResDevelop. Conf. (sponsored by the Sensory Aids Evaluation)