VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JnanaSangama, Belagavi – 590018

A Project Report
On

"Braille Calculator"

Submitted in partial fulfillment of the requirements as a part of the curriculum,

Bachelors of Engineering in Mechanical Engineering

Submitted by

Pavan S B
(1CR16ME053)

Ravikumar K
(1CR16ME062)

Manojkumar R
(1CR16ME045)

Nithin M U
(1CR16ME052)

Under the Guidance of

Joseph Sajan
Associate Professor
Department of Mechanical Engineering

Department of Mechanical Engineering

CMR INSTITUTE OF TECHNOLOGY

132, AECS Layout, Kundalahalli, ITPL Main Rd, Bengaluru – 560037

CMR INSTITUTE OF TECHNOLOGY
132, AECS Layout, Kundalahalli colony, ITPL Main Rd, Bengaluru-560037
Department of Mechanical Engineering

CERTIFICATE

Certified that the project work entitled " **Braille Calculator"** is a bonafide work carried out by **Mr. Pavan S B, Mr. Ravikumar K, Mr. Manojkumar R, Mr. Nithin M U,** bonafide students of **CMR Institute of Technology** in partial fulfillment for of the requirements as a part of the curriculum,

Bachelors of Engineering in Mechanical Engineering, of **Visvesvaraya Technological University, Belagavi** during the year **2019-20**. It is certified that all correction/suggestion indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the bachelor of engineering degree.

(Joseph Sajan) **(Dr. Vijayanand Kaup)** **(Dr. Sanjay Jain)**
Signature of the Guide **Signature of the HOD** **Signature of the**
Principal

External Viva

Name of the examiners **Signature with date**

1.

2.

DECLARATION

We, students of Eighth Semester, B.E, Mechanical Engineering, CMR Institute of Technology, declare that the project work titled **"Braille Calculator"** has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree in **Bachelor of Engineering in Mechanical Engineering** of **Visvesvaraya Technological University, Belagavi,** during the academic year 2019-2020.

Mr. Pavan S B	**Mr. Ravikumar k**	**Mr. Manojkumar R**	**Mr. Nithin M U**
(1 CR16ME053)	**(1CR16ME062)**	**(1CR16ME045)**	**(1CR16ME052)**

Place: Bengaluru

Date:

ACKNOWLEDGMENTS

This project work marks the end of an unforgettable journey of obtaining my Bachelor’s degree. I have not travelled in vacuum on this memorable path, and am indebted to many people for making it a great and challenging experience.

I am extremely indebted to my Professor Joseph Sajan for accepting to be my internal guide and supporting my thesis. His understanding, encouragement and personal attention instilled in me the confidence to carry out my project work.

I would like to thank Dr. Vijayananda Kaup, Head of Department, Mechanical Engineering, CMR Institute of Technology, for his continuous support and guidance. I am also grateful to Dr. Sanjay Jain, Principal, CMR Institute of Technology for providing the necessary facilities to carry out the project.

It has been an honor to study for 4 years at CMR Institute of Technology, Bangalore, under the guidance of a constant encouraging faculty members. I would like to thank all the faculty and staff of CMR Institute of Technology, Bangalore who helped me throughout the course of my dissertation. Also, I would also like to recognize my fellow graduate students for their constant support and help.

This journey became much simpler and enriching experience when we traveled together. This thesis is the result of all the people who guided and provided constant support when I really needed. Thank you doesn’t seem sufficient but it is said with appreciation and respect to all for their support, encouragement, care and understanding.

PAVAN S B	**RAVI KUMAR K**	**MANOJ KUMAR R**	**NITHIN M U**
1CR16ME053	**1CR16ME062**	**1CR16ME045**	**1CR16ME052**

ABSTRACT

A thorough understanding in mathematics enhances educational and occupational opportunities for all people, whether sighted or visually impaired. In day-to-day routine, a practical understanding of mathematics allows a person to function more successfully and independently . Access to, and solving mathematical problems is one of the biggest obstacles for blind students in schools and universities as Calculators with smooth input keys and LCD outputs which are available in market are useless for the visually challenged persons. However, technology has removed barriers to educate and employ people with visual impairment. For them, an optimal solution is the Braille calculator with audio output that will present them better access to mathematics, and provide new tools for computation.

Braille is a tactile writing language of raised dots, which is widely used by the visually impaired persons. In this project, the smooth keypad has been replaced with Braille-embossed keypad and the output is obtained through audio via speakers and/or headphones. The input here is given through a 4 by 4 Membrane keypad which has been embossed with Braille lettering via indigenous method. The output is denoted by natural English language up to two places of decimal point. With the implementation of Braille keys and audio output via speaker/headphone in an affordable cost, it greatly enhances the computational skills of the visually impaired.

CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Our reasoning and thinking skills are enhanced by trained by mathematics. Though various . As a subject, Mathematics plays an important role in society and the school curriculum is formulated in such a way that Mathematics is given a central and significant place in it. Teaching and learning of Mathematics is compulsory right from the primary level to the secondary level of education. However reading and writing mathematics is fundamentally different than reading and writing text. Traditionally Mathematics has been inaccessible to visually impaired and blind students because its content is rich with visually presented concepts and information.

Calculators are one of the most commonly required tools at home, school and work, however, people with disabilities may have difficulty using standard calculators as calculators generally available in the market have smooth keys for input and LCD display for output. It offers few or no tactile clues to allow a person with a visual impairment to orient him or herself to the keypad. People with visual impairments may not be able to see the appropriate buttons too. Due to this, it is of little use to those who are visually impaired. This was the primary motivation of this project. The aim of this project is to explore alternatives which can improve the computational facilities of the visually impaired. This project aims at finding a viable optimal solution for this problem. This is done by the following ways: for input, we use tactile or haptic feedback in the form of Braille-embossed keypad in place of smooth keypads used in regular calculators; for output, there is provision of both headphones and speakers to spell out the inputted mathematical data and the result respectively.

From the literature survey, it was found that there are some ready-reckoners Talking Calculators available in the market made by Texas Instruments and other such technology vendors that can read the display and may announce each key as it is pressed. However none of the commercially available calculators have Braille keypads. Also, these products are of very high cost ranging from $200 to $500 [3]. Some of the existing systems lack portability with insufficient power management. Some system uses relays to generate 3 patterns and draws more current as relays are used.

Due to the use of embossing mechanism in the earlier Braille display Typewriter model, the weight of embossing tool will slow the rotational speed of stepper motor however light this may be, and also add to the cost of product [5]. The LEO Braille Scientific Calculator provides a refreshable 8 dot, 8 cell braille displays. The calculator can compute standard arithmetic functions, plus square root, trigonometry functions, logarithm, conversions, and financial functions. However, it is available from LS&S Group at a very high cost of $999 [4]. As such, the aim of our project is also to keep the cost of the handheld calculator in the range which is affordable by the visually impaired people from humble economic background and it should be convenient to use as well.

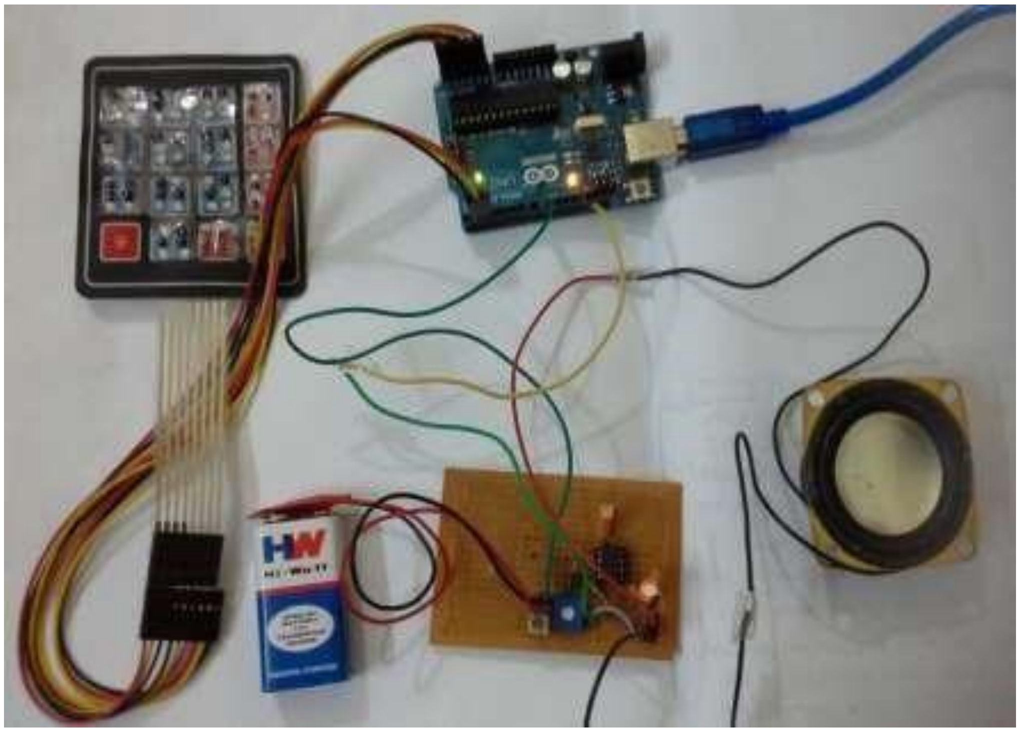


Figure 1: Set-up of project prototype

CHAPTER 2

Methodology

2.1 Problem Definition

285 million people are estimated to be visually impaired worldwide: 39 million people are blind and 246 million have low vision. About 90% of the world's visually impaired people live in low income settings [Source : World Health Organization].

Arithmetic calculators are crucial to computational purposes to students, teachers, academicians and people working in industry. Lack of suitable computational equipments pose a challenge to the visually impaired.

So far, it can be seen that the commercially available solutions of the given problem are either insufficient or are costly, and beyond the means of people from weaker economic background.

In this proposed portable electronic Braille calculator, the main objective is to eliminate the barriers that prevent visually challenged children and adults from using the embedded tech device due to their lack of knowledge to use computers.

2.2 Planning and Approach

This project utilizes Braille language as input to the calculator. Braille is a tactile writing system used by the blind and the visually impaired. Braille is named after its creator, Frenchman Louis Braille. It is developed for our haptic perception, a combination of the sense of touch, movement and finger pressure. Braille characters are small rectangular blocks called cells that contain tiny palpable bumps called raised dots. Each cell is made up of six raised dots arranged in a 3x2 membrane that relate to a letter, number or character. A fingertip can feel the whole cell at once [2]. The system gives 63 combinations and one blank step [3]. These dots are raised roughly 0.9 mm by pressure from a cylindrical pin, causing identification underneath the thick paper [3]. Numbers are written in Braille using the NUMERAL SIGN (number sign), dots 3, 4, 5 and 6, The following diagrams show sample Braille representations of numbers.

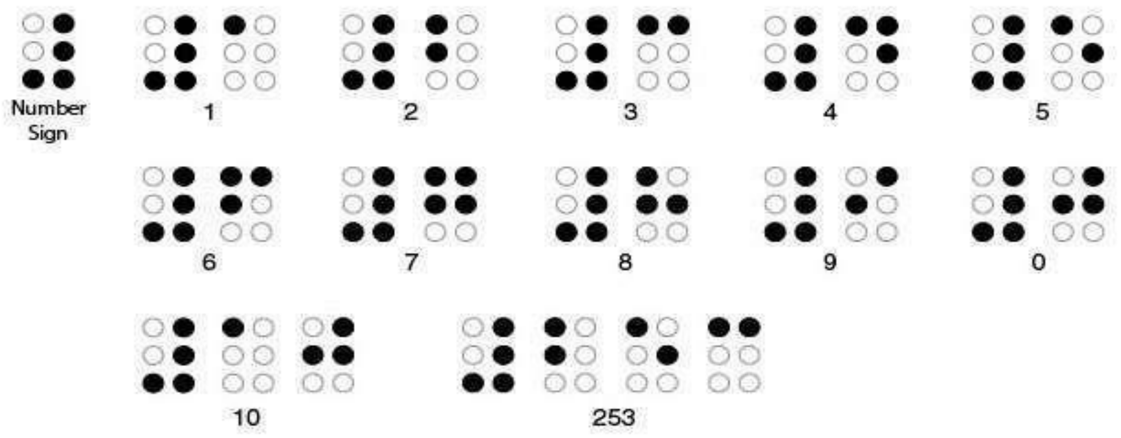


Figure 2.1: A sample example of Braille number system

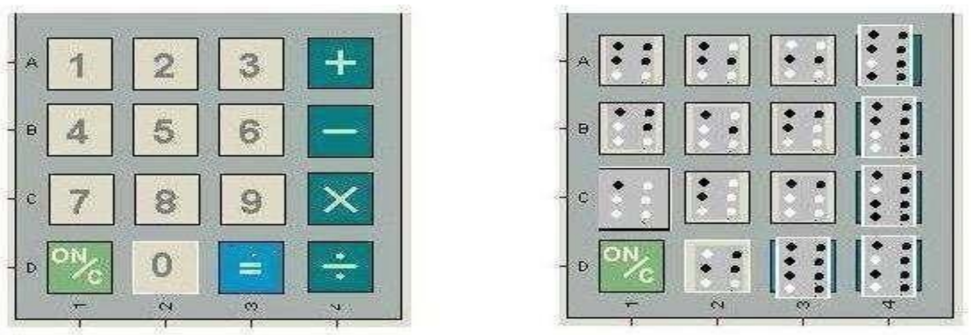


Figure 2.2: Push button keyboard used as Braille keyboard

In this project a regular 4 by 4 membrane keypad has been embossed with a plastic cover having raised dots, enabling the haptic feedback. It has 16 keys: 10 keys represent the ten digits of the number system, 4 keys denote four basic operations (addition, subtraction, multiplication and division), and 1 key for equal sign and the last key for Clear operation. The Clear operation deletes the last inputted number, ensuring better control of the user. To use the calculator, the user must press the first operand followed by an operator key, and then the second operand. The result is outputted on pressing the ‘equal to’ button.

The software code associated with the model ensures that the calculator does not store any input if an operator key is pressed before typing any operand key.

The heart of the project is the Arduino microcontroller. It serves has the basic calculating unit of the calculator. It has been programmed to perform the basic operations like addition, subtraction, multiplication and division.

It takes input from the keypad and stores it in its on-board memory. On getting all the required inputs and receiving the button-press of the equal to button it gives out the correct result.

The SD card module contains .wav file corresponding to each key (numbers, operators etc.) on the key-pad along with some extra values like ‘point’ (to denote correct representation following a division operation). Once the equal to button is pressed, Arduino breaks down the result into single digits and outputs the corresponding audio file for each number of the result, starting from the left-most digit.

The proposed system also has a SD card for storing the audio files corresponding to every numeric digits i.e. operands and the operators.

The output of the calculator is given out through audio. The model has provision for both headphone and speaker. The headphone is for personal use, and the speaker is used for demonstrative purpose.

Block diagram of proposed circuit:

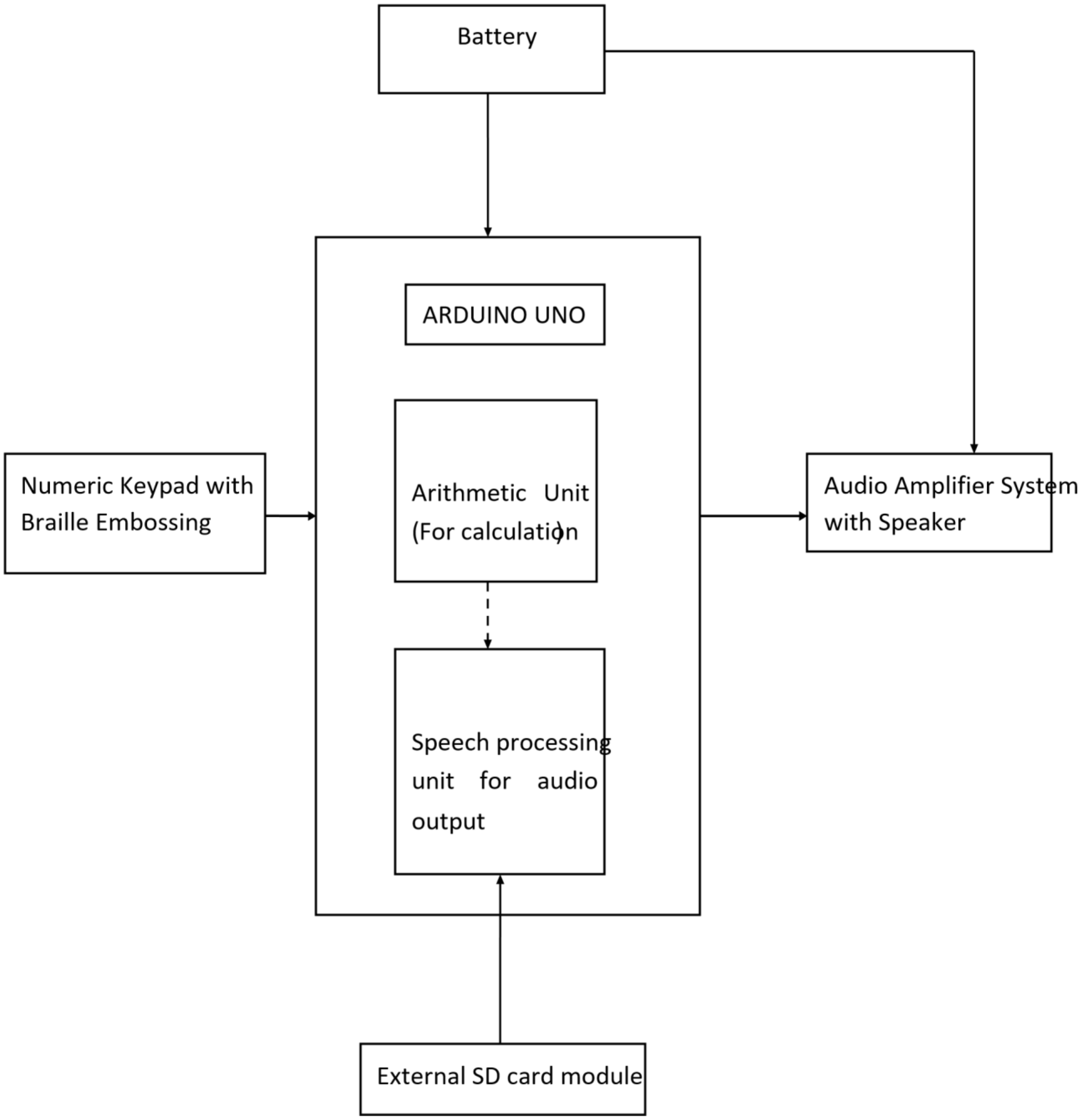

Battery
ARDUINO UNO
Arithmetic Unit (For calculation)
Speech processing unit for audio output
Numeric Keypad with Braille Embossing
Audio Amplifier System with Speaker
External SD card module

Figure 2.3: Functional Block diagram of the model

Circuit diagram:

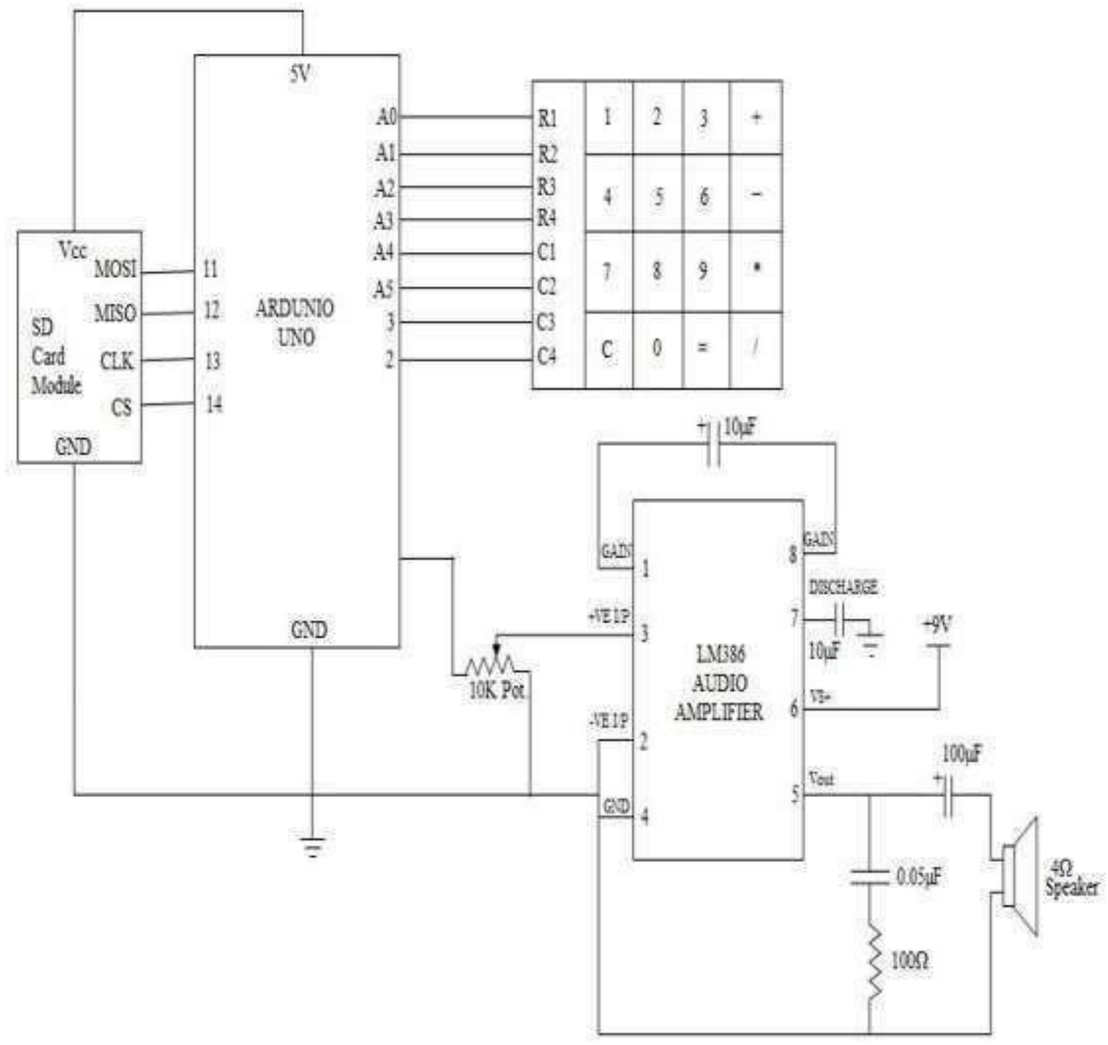
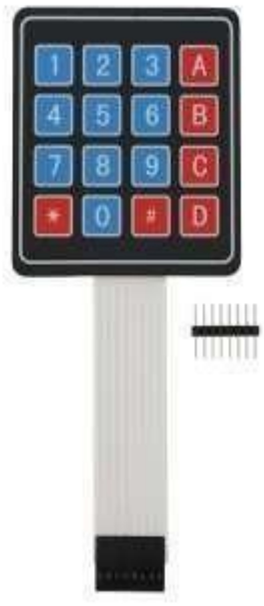


Figure 2.4: Circuit diagram of proposed model

2.2.1 Architecture: The design architecture can be analysed based on two platforms: hardware and software.

2.2.1.1 Hardware Architecture:

Numeric Keypad with Braille embossing: it serves as the input unit of the project. It provides the required tactile feedback to the visually impaired, so that they can easily understand which numbers the keys represent.



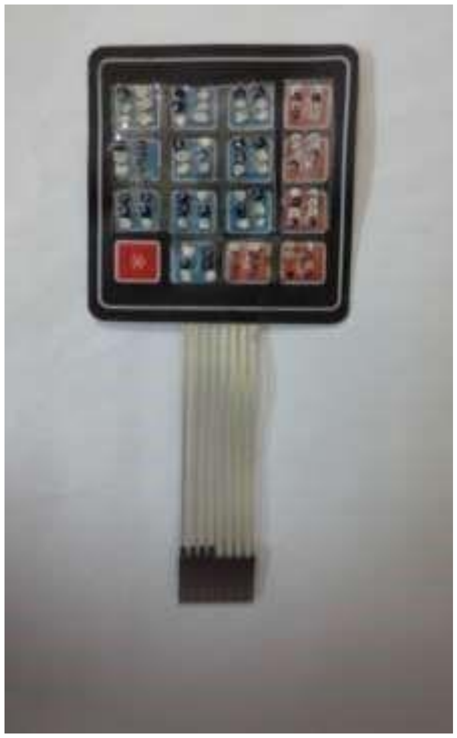


Figure 2.5: 4 by 4 Membrane Keypad

Figure 2.6: Braille-embossed keypad

Arduino Uno Microcontroller: The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. Pin 3, 5, 6, 9, 10, and 11 provide 8-bit PWM output with the analogWrite() function. Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library. LED 13 is a built-in LED connected to digital pin 13.

When the pin is HIGH 10 value, the LED is on, when the pin is LOW, it's off. The

Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality: They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions: It contains everything needed to support the microcontroller; it can simply be connected to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USBto-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX).

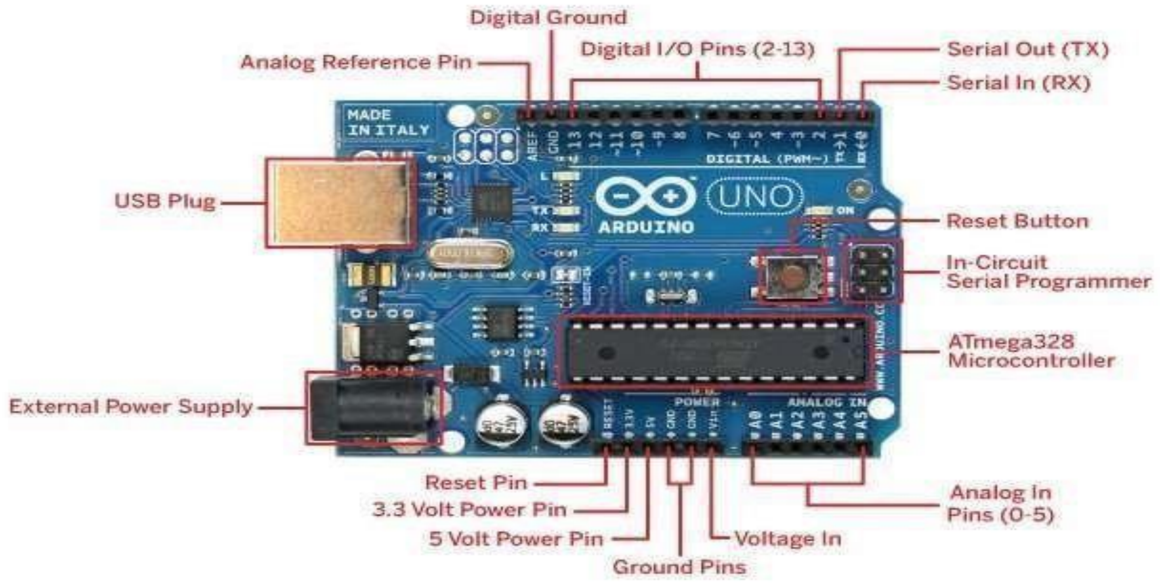


Figure 2.7: Description of pins of Arduino Uno board

Summary of Arduino Uno board:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

In this project the Arduino board serves two purposes: one, as the arithmetic unit of the calculator which performs the basic operations like addition, subtraction, multiplication and division; second, it acts as the speech processing unit which controls the audio output corresponding to the result. The result given is correct up to two places of decimal.

SD card module: The output audio samples are stored in the Adafruit SD card module as .wav files as the on-board memory of the Uno board (32 KB) is not sufficient to store all the audio files. The communication between the microcontroller and the SD card uses SPI, which takes place on digital pins 11, 12, and 13. Additionally, another pin must be used to select the SD card. This can be the hardware SS pin - pin 10 (on most Arduino boards).

Arduino Pin – SD Module Pin

Pin 10 (SS) to CS

Pin 11 (MOSI) to DI

Pin 12 (MISO) to DO

Pin 13 (SCK) to CLK

Also, G, + of SD Module are connected to GND and 5V of Arduino board respectively.

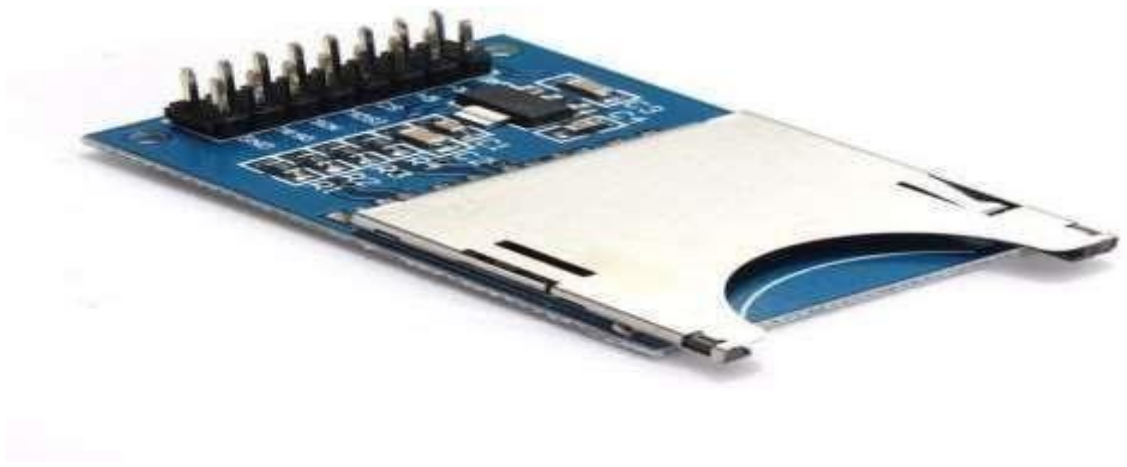
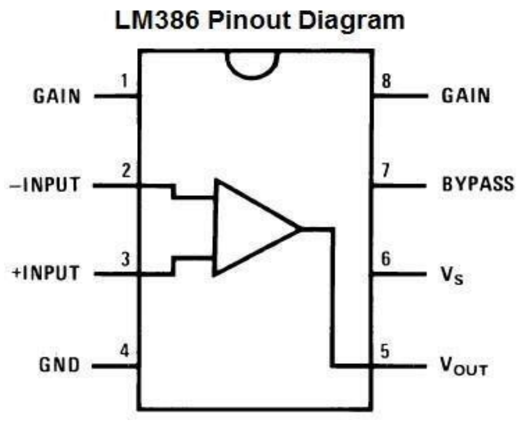


Figure 2.8: Adafruit SD Card Module

Audio Amplifier System with Speaker: The LM386 is a power amplifier designed for use in low volt-age consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value from 20 to 200.The inputs are ground referenced while the output automatically biases to one-half the supply voltage. The quiescent power drain is only 24 mW when operating from a 6 volt supply, making the LM386 ideal for battery operation If a speaker is not used then the output is directly obtained by connecting one pin of the headphone to the pin no. 9 of Arduino board and another pin to ground.



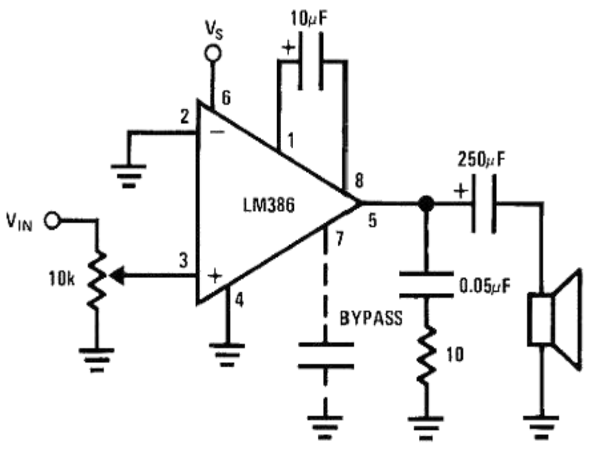


Figure 2.9: LM386 pin-out Figure 2.10: Circuit diagram of LM386

Pin details:

Terminals 1 and 8 represent the gain control of the amplifier. These are the terminals where you can adjust the gain by placing a resistor and capacitor or just capacitor between these terminals. In this circuit, we will place a 10μF capacitor between these terminals for the highest voltage gain.

Terminals 2 and 3 are the sound input signal terminals. These are the terminals where you place the sound which you want to amplify. Terminal 2 is the -input and Terminal 3 is the +input. In our circuit, the positive sound signal will be placed on terminal 3 and terminal 2 will be tied to ground.

Terminal 4 is GND (ground) and will be tied to ground in the circuit.

Terminal 5 is the output of the amplifier. This is the terminal in which the amplified sound signal comes out.

Terminal 6 is the terminal which receives the positive DC voltage so that the op amp can receive the power it needs to amplify signals.

Terminal 7 is the Bypass terminal. This can bypass 15KΩ resistors. This pin is usually left open or is wired to ground. However, for better stability, a capacitor is added in our circuit because this can prevent oscillations in the op amp chip

2.2.1.2 Software used: The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software that uses simplified library functions of C/C++. It makes input/output operations much easier. Arduino is an open-source single-board microcontroller, descendant of the opensource Wiring platform, designed to make the process of using electronics in multidisciplinary projects more accessible.

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

Features of the Arduino platform include:

i)Wide adoption and useful support from numerous websites, newsgroups, and user forums.

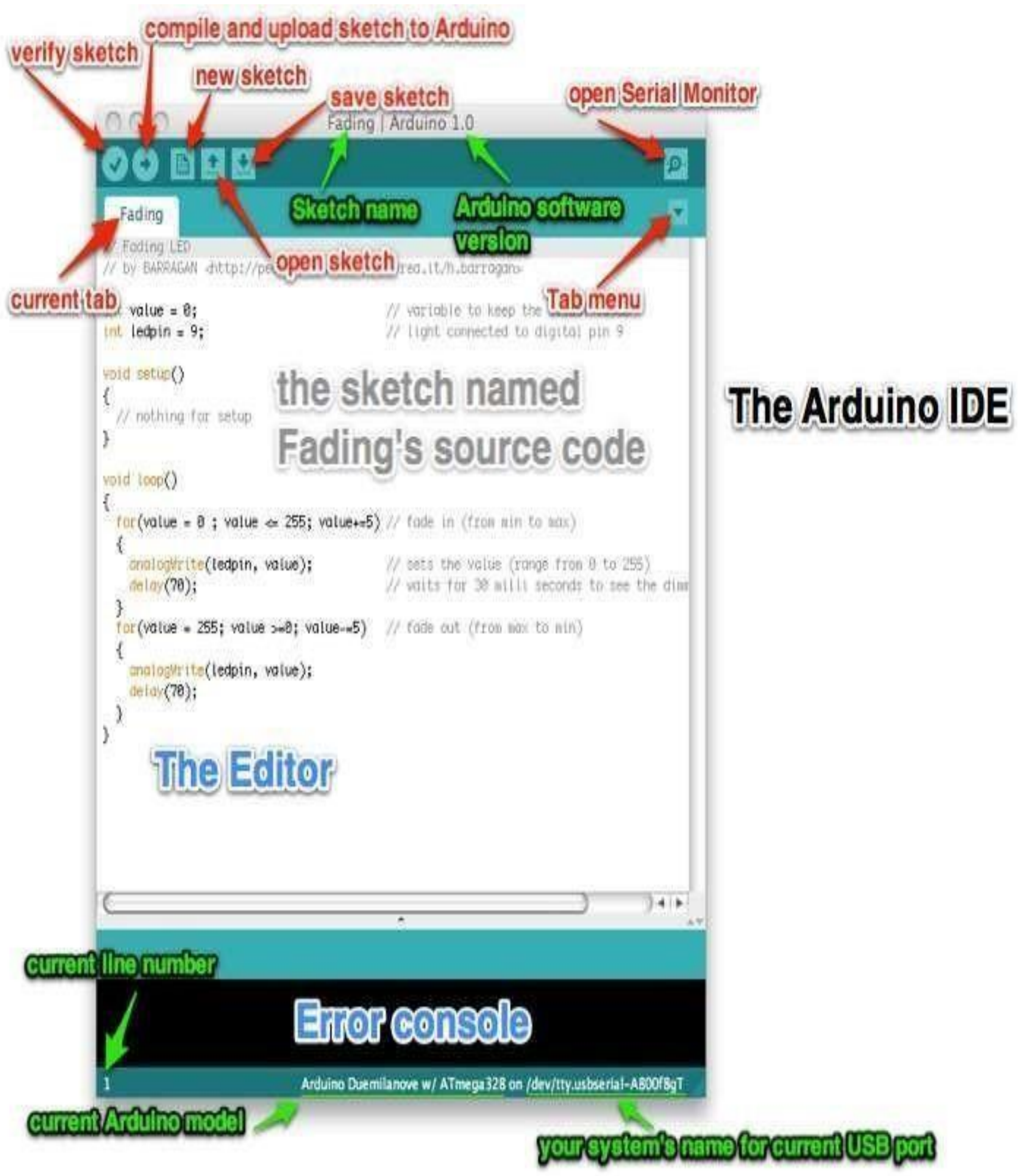
ii)Extensibility through plug-in boards for a broad range of applications, such as motor control, wireless communication, audio processing, and data logging.

iii) The Arduino IDE comes with a C/C++ library called "Wiring" (from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++, although users only need to two functions to make a run able program.

iv) Setup () – a function run once at the start of a program that can initialize settings.

v) Loop () – a function called repeatedly until the board powers off.

The Arduino IDE uses the GNU tool chain and AVR Library to compile programs, and uses AVR dude to upload programs to the board.



```
// Fading LED
// by BARRAGAN <http://pe                red.it/h.barragan>

   value = 0;                          // variable to keep the
int ledpin = 9;                        // light connected to digital pin 9

void setup()
{
  // nothing for setup
}

void loop()
{
  for(value = 0 ; value <= 255; value+=5) // fade in (from min to max)
  {
    analogWrite(ledpin, value);        // sets the value (range from 0 to 255)
    delay(70);                         // waits for 30 milli seconds to see the dim
  }
  for(value = 255; value >=0; value-=5)   // fade out (from max to min)
  {
    analogWrite(ledpin, value);
    delay(70);
  }
}
```

Figure 2.11: Snapshot of Arduino IDE

2.3 Design Issues

When using a speaker, the amplification factor provided by the LM386 amplifier is used. In this project, the gain is chosen to be a maximum, i.e. 200.

For such gain, the datasheet of LM386 dictates that a capacitor of 10 µF should be used between pin 1 and pin 8 of the LM386 IC.

At the input of LM386, a 10K trim pot is inserted in order to control the input voltage to the input pin of LM386 chip.

At the output of the LM386 amplifier (Pin no. 5), a high pass filter is added to bypass the high frequency ripples.

Here a filter is designed to bypass frequencies greater than 30 kHz.

$$f = 1/(2\pi RC) \qquad \text{eqn. (1)}$$

Chosen value of C = 0.05 μF

Putting this value in the equation (1), for f ≈ 32 kHz, we get R = 100 **Ω**.

For the gain control part of the amplifier:

For this part, we need to consider the internal structure of the LM386 operational amplifier.

LM386 Input Stage:

The first block is a PNP Emitter Follower amplifier (Q1, Q3), it sets the input impedance and defines the DC operation points, raising the input voltages off ground so the circuit will accept negative input signal down to -0.4 V.

Both 50k input resistors (R1, R3) create the path to ground of the base current, the input needs to be coupled so not to disturb the internal biasing, hence the input impedance is dominated by these resistors and set to 50K.

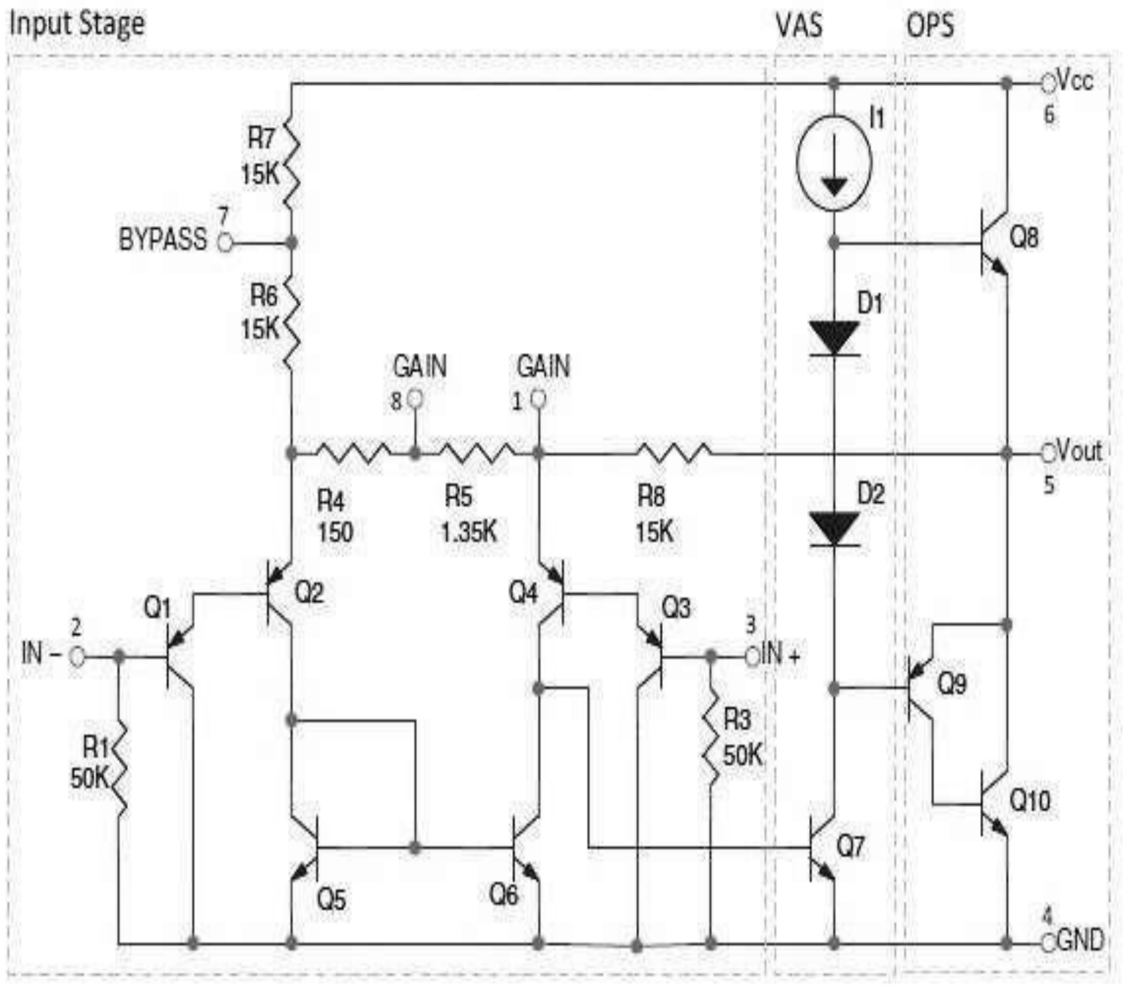


Figure 2.12: Internal circuit of LM386

Voltage Gain Analysis:

The differential amplifier Long Tailed Pair (Q2, Q4) gain is adjusted by two gainsetting resistors 1.35K + 150Ω (R5 + R5). External pins 1 and 8 provide access to adjust the gain from 20 to 200.

The voltage Gain can be calculated under quiescent conditions (no input signal applied) as follows:

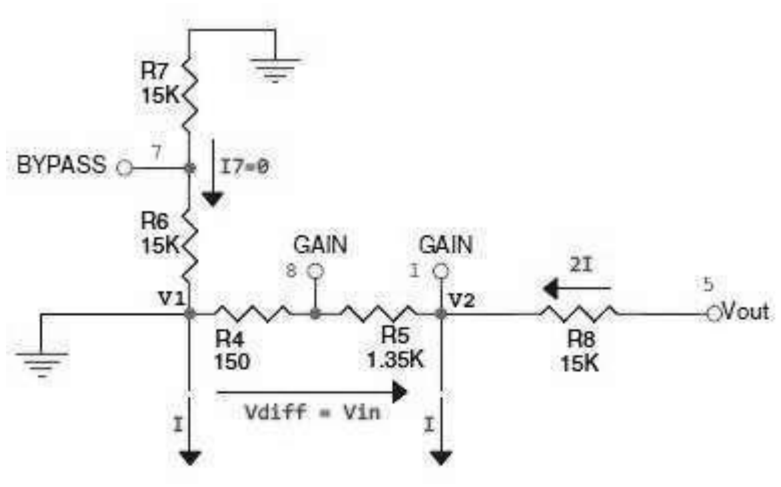


Figure 2.13: Equivalent circuit

Note:

1. The voltage across R4 and R5 (Vdiff) is simply the differential input voltage (Vin), because the base-emitter voltage drops in the PNP transistors (Q1, Q2, Q3 and Q4) are the same in each side of the LTP.
2. The current mirror formed by Q5 and Q6 generates equal currents in both sides of the LTP. This current is called "I".

Due to the current mirror, the current intensity through R8 is equal to 2i, neglecting the current (i7) through the two 15K resistors (R6, R7), which are large impedances compared to other ports of the circuit, thereby:

$$\frac{V_{out} - V_{in}}{R_8} = 2I$$

$$V_{out} \gg V_{in}$$

$$\frac{V_{out}}{R_8} = 2I$$

In the figure above, is easy to see that if i7=0 then:

$$I = \frac{V_{in}}{R_4 + R_5}$$

So we can write , $\frac{V_{out}}{R_8} = 2\frac{V_{in}}{R_4 + R_5}$

$$G_v = \frac{V_{out}}{V_{in}} = 2\frac{R_8}{R_4 + R_5}$$

This formula can also be rewritten in a more generic way as:

$$G_v = \frac{V_{out}}{V_{in}} = 2\frac{Z_{1-5}}{150 + Z_{1-8}}$$

Where Z1-5 and Z1-8 are the impedances between the respective pins. Without any external components, it has a gain of Gv = 2×15K/(150+1350) = 20 (26 dB). With a capacitor (or shortcutting) between pins 1 and 8, it has a gain of Gv = 2×15K/150 =200 (46dB). Using the derived gain formula,

Gain (A) = 2× Z1-5 /(150+ Z1-8)

= 2×15×103 /(150+1350)

= 20

To vary the gain we apply a capacitance between pins 1 to 8

Taking 10 µF capacitor

Xc = 0.5×3.14×3×103×10×10-6

= 102 /6.28×3

= 5 Ω

Now applying in the formulae:

Gain (A) = (2×15× 103)/ (150+5)

= 150

Serial No.	**Name of Component**	**Quantity**	**Approx. Cost (in ₹)**
1	Battery (9V)	1	10
2	Arduino Uno Microcontroller	1	560
3	SD Card Shield	1	200
4	LM 386 (OP AMP)	1	12
5	Speaker	1	200
6	4 by 4 Membrane Keypad	1	180
7	Point Vero Board	1	10
8	Capacitors	3	20
9	10K pot	1	10
10	Resistance	1	5
11	Jumper wires		10
12	Pin Headers		10
		Total	1227

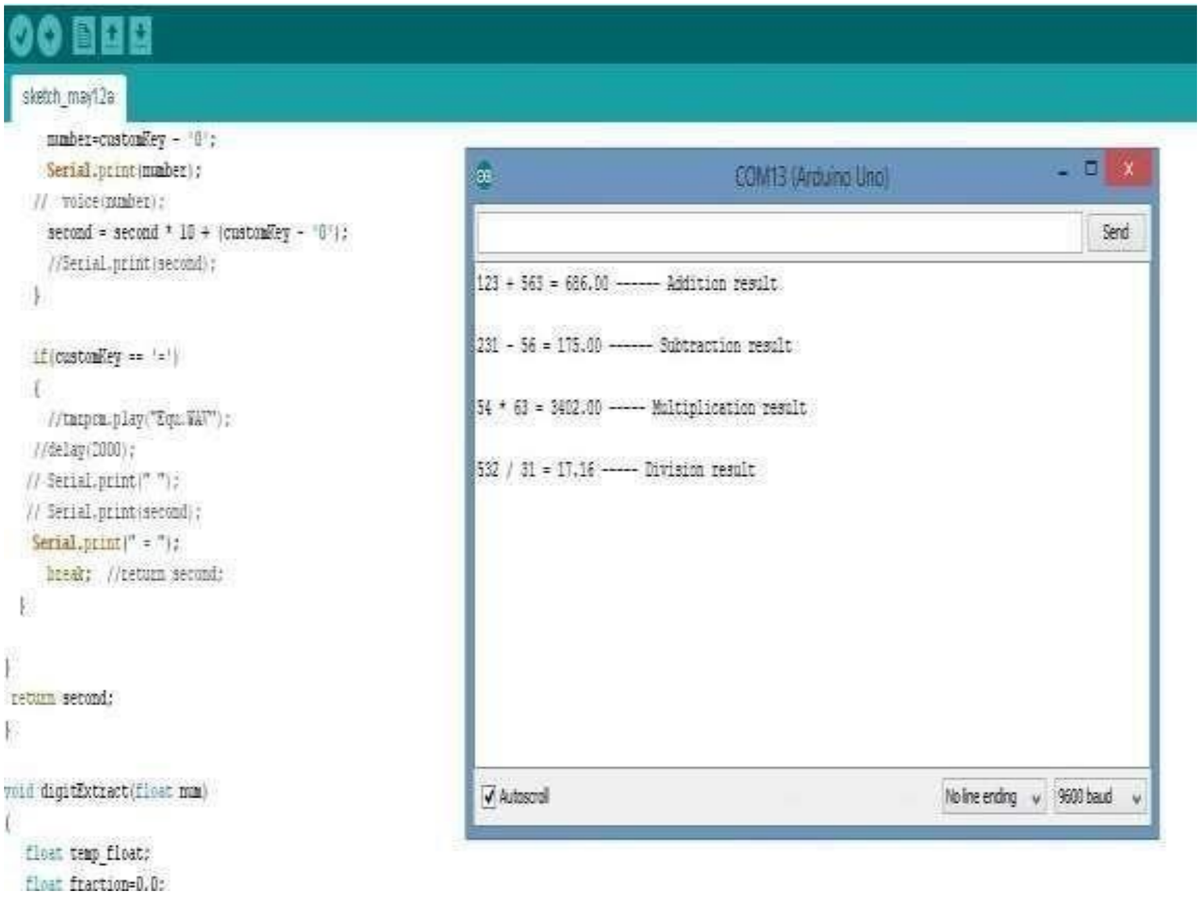
List of Components:

CHAPTER 3

Testing

When the audio feedback is not used, the results can also be verified by using the serial monitor of the Arduino IDE. The software code written for the calculator has provision for displaying the results on the computer as well, to which the Arduino board is connected. This method of testing was used a number of times to test the performance of the calculator before the audio output was introduced and it correctly showed the outputs.

The calculator has been demonstrated a number of times before students and teachers alike, and every time it performed to the satisfaction of the audience.



```
    number=customKey - '0';
    Serial.print(number);
  //  voice(number);
    second = second * 10 + (customKey - '0');
     //Serial.print(second);
  }

  if(customKey == '=')
  {
    //tmrpcm.play("Equ.WAV");
  //delay(2000);
 // Serial.print(" ");
 // Serial.print(second);
  Serial.print(" = ");
    break;  //return second;
 }


}
 return second;
}

void digitExtract(float num)
{
  float temp_float;
  float fraction=0.0;
```

```
123 + 563 = 686.00 ------ Addition result

231 - 56 = 175.00 ------ Subtraction result

54 * 63 = 3402.00 ----- Multiplication result

532 / 31 = 17.16 ----- Division result
```

Figure 3.1: Testing using Serial Monitor on Arduino IDE

CHAPTER 4
Results

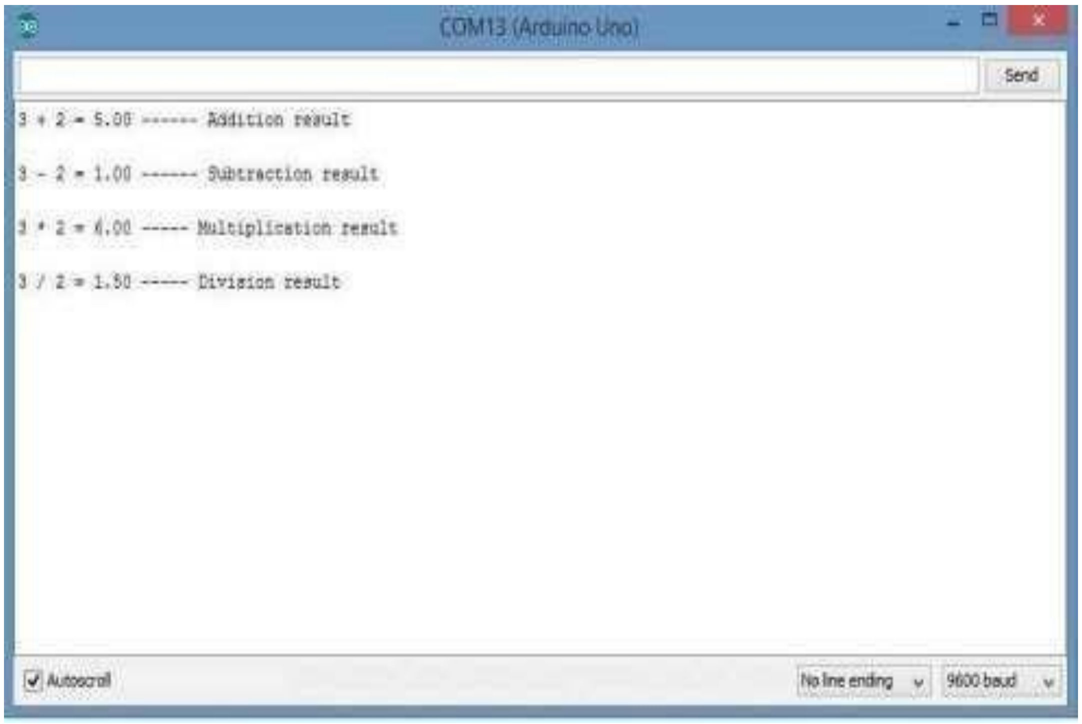
Addition: The result given by audio output for the operation 3 + 2 is given by “Three plus two equals to five point zero zero.”

Subtraction: The result given by audio output for the operation 3 - 2 is given by “Three minus two equals to one point zero zero.”

Multiplication: The result given by audio output for the operation 3 * 2 is given by “Three multiplied by two equals to six point zero zero.”

Division: The result given by audio output for the operation 3 / 2 is given by “Three divided by two equals to one point five zero.”

There is also provision for a clear button, denoted by the only smooth key of the keypad (denoted as ‘C’). It clears the last digit entered, which allows for rectification of any errors during input. Since it is the only smooth key, it can be identified easily, even by the visually impaired.



```
3 + 2 = 5.00 ------- Addition result

3 - 2 = 1.00 ------- Subtraction result

3 * 2 = 6.00 ------ Multiplication result

3 / 2 = 1.50 ------ Division result
```

Figure 4.1: Results as seen on Serial Monitor

CHAPTER 5

Conclusion

The calculator model is easy to use, and can be mounted easily in a small box, hence improving portability. It does not contain any bulky parts.

The results are spoken in clear, audible voice correct up to two places of decimal in every case.

The overall cost of the project is around Rs. 1200, which is cheaper than most of the commercially available products of this category. With batch processing, this cost will minimize even further.

Limitations

Due to the small pin count of the Arduino Uno board, it was not possible to incorporate more functions to the calculator. As of now, the calculator lacks a decimal point on the keypad, hence, presently the calculator does not have any provision for mathematical operations involving decimal numbers.

This problem can be readily solved by using any other board of the Arduino family which has a higher pin count (e.g. Arduino Due board). By introducing higher pin count, the number of input keypads can be increased, which will contribute to introduction of more mathematical functions. Also, the changes required in the software for this purpose is minimal and can be done easily

Future Scope

The easy-to-use Arduino platform allows for easier software up-gradation. Any other development board from the Arduino family can also be used with minimal changes to the software code. Hence, adding more functions to the present model of the only calls for a bigger development board and minimal software change. Also, precision of the results after the decimal point requires a software upgrade only. Here, it has been kept up to two places of decimal for demonstrative purposes only.

References:

1. Yoshit V. Gidh, Mahesh S. Latey, Arpita Roy, Kunal Shah, Savita , "Braille Calculator" , International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 2 Issue 2 Feb 2013 Page No. 481-382

2. Sunil Kumar M.E –Karnataka University Braille language learner for blinds International Journal Of infinite Innovations in technology 2012-2013Reg. No.:20120905|DOI:V1I2P05

3. M.A.Raja, A.Arunya, Dipti Yadav, G.Kanitha, R.Maheshwari , "Portable Refreshable E-Braille", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 1, January - February 2013, pp.1393-1397

4. Assistive Technology Quick Reference

5. Kumar J.A.V, Visu A , Raj M.S, Prabhu M.T, Kalaiselvi V.K.G , "A pragmatic approach to aid visually impaired people in reading, visualizing and understanding textual contents with an automatic electronic pen", Computer Science Automation Engineering (CSAE), 2011 IEEE International Conference.

6. http://www.datasheetlib.com/datasheet/52982/lm386_nationalsemiconductor.html

7. https://www.sparkfun.com/products/12761

8. http://arduino.cc/en/Main/arduinoBoardUno

9. http://www.analog.com/en/interfaceisolation/leveltranslators/adg3304/products/product.html

10. http://www.learningaboutelectronics.com/Articles/How-to-connect-a-LM386-audioamplifier-chip
11. http://en.wikipedia.org/wiki/Braille

Appendix

Software Code

```
#include <Keypad.h>
#include <SD.h>
#define SD_ChipSelectPin 4
#include <TMRpcm.h>          // also need to include this library

#include <SPI.h> long first = 0; int factor=1; int temp,number;

long clearFirst=0,clearSecond=0; long second = 0; double total

= 0; TMRpcm tmrpcm;     char customKey,clearkey; const byte

ROWS =

4; const byte COLS =

4;


char keys[ROWS][COLS] = {
  {'1','2','3','+'},
  {'4','5','6','-'},
  {'7','8','9','*'},

  {'C','0','=','/'}

};


byte rowPins[ROWS] = {A5,A4,A3,A2}; //connect to the row pinouts of the keypad byte

colPins[COLS] = {A1,A0,3,2}; //connect to the column pinouts of the keypad


//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  // put your setup code here, to run once:
tmrpcm.speakerPin = 9;
  Serial.begin(9600);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
customKey = customKeypad.getKey();   switch(customKey)

  {     case '0' ... '9': // This keeps collecting the first value until a operator is pressed
"+-*/"

number=  customKey - '0';

    Serial.print(number);     //voice(number);

    first = first * 10 + (customKey - '0');

    break;



  case '+':
   // first = (total != 0 ? total : first);
   // Serial.print(" ");
   // Serial.print(first);     Serial.print("

+ ");

tmrpcm.play("Plus.WAV"

);     number=0;

    second = SecondNumber(); // get the collected the second number



    total = first + second;
    Serial.print(total);
     Serial.print(" ------ Addition result");
    Serial.println();
    Serial.print('\n');
    //digitExtract(total);     first = 0, second = 0; // reset values back

to zero for next use     break;
```

```
case

'-':

  //  first = (total != 0 ? total : first);
   tmrpcm.play("Minus.WAV");
  // Serial.print(" ");
  // Serial.print(first);
    Serial.print(" - ");


     number=0;     second =

 SecondNumber();           //

 Serial.print(second);

 total = first - second;

    Serial.print(total);
    Serial.print(" ------ Subtraction result");
    //Serial.print(total);
    Serial.println();
    Serial.print('\n');

//digitExtract(total);     first

= 0, second =

0;     break;

case '*':

  //  first = (total != 0 ? total : first);
  //  Serial.print(" ");
  //  Serial.print(first);     Serial.print("

* ");

   tmrpcm.play("Mul.WAV");

    number=0;     second = SecondNumber();

//Serial.print(second);     total

= first * second;

    Serial.print(total);
     Serial.print(" ----- Multiplication result");
    //Serial.print(total);
    Serial.println();
    Serial.print('\n');

//digitExtract(total);

first = 0, second = 0;

break;

 case  '/':

    // first = (total != 0 ? total : first);       //Serial.print(" ");
   // Serial.print(first);

Serial.print(" / ");


tmrpcm.play("Div.WAV"

);    number=0;     second

= SecondNumber();


    //Serial.print(second);
    second == 0 ? Serial.print("Invalid") : total = (float)first / (float)second;
Serial.print(total);
Serial.print(" ----- Division result");
   //
Serial.print(total);      Serial.println();
    Serial.print('\n');

//digitExtract(total);     first

= 0, second =

0;     break;


  case 'C':

  if(second==0) {

clearFirst=first - (first%10);       first=clearFirst/10;}

else if(second!=0)

{
```

```
clearSecond=second - (second%10);       second=clearSecond/10;}

      else {         first = 0,

second = 0; }



    total =

0;
break;

}

}
long SecondNumber()
{

while(

1 )

  {
    customKey = customKeypad.getKey();     if(customKey

>= '0' && customKey <= '9')

    {
     number=customKey            -            '0';

Serial.print(number);     //  voice(number);
second = second * 10

+ (customKey - '0');

      //Serial.print(second);
    }


    if(customKey == '=')
    {
      tmrpcm.play("Equ.WAV");
        Serial.print(" = ");       break;

//return second;

  }
```

```
}  return second;

}  void

digitExtract(floa

t num)  {    float

temp_float;

float

fraction=0.0;

int

integer,int_tmp,

factor=1,fractio

n1=0,digit;

//Extraxting the

digits from the

results//

integer=

 (int)num;      //Serial.println(i nteger);   fraction =

 num - integer;      //Serial.println(fraction);

 int_tmp=integer;    //printf("The int_tmp value is

 %d\n\r",int_tmp/10);   while(int_tmp){

 int_tmp=int_tmp/10;       factor = factor*10;

      //printf("%d %d\n\r  ", factor, int_tmp);
     // i++;    }

while(factor>1){

factor = factor/10;

// printf("%d", factor);

digit=integer/factor;

     // Serial.println(digit);
//voice(digit);
```

```
      //tmrpcm.play("8.wav");

//  delay(2000);       //

Serial.println(digit);
delay(2000);       fraction1 = fraction1

% factor;

            //fraction1=fraction1 % 10;

            //printf(" %d ",fraction1);
  }
}
```