DEVELOPING A SELF-LEARNING BRAILLE KIT FOR VISUALLY IMPAIRED PEOPLE

By

RUBEL AHMED
MOHAMMAD IFTIKHAR RAHMAN NOMAN

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

MAY 2018

DEVELOPING A SELF-LEARNING BRAILLE KIT FOR VISUALLY IMPAIRED PEOPLE

By

RUBEL AHMED

MOHAMMAD IFTIKHAR RAHMAN NOMAN

A thesis/project
submitted as partial fulfilment of the requirement for the degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

MAY 2018

CERTIFICATE OF APPROVAL

The project entitled as **"Developing a Self-learning Braille Kit for Visually Impaired People"** submitted by **Rubel Ahmed**, bearing Metric No. **ET-141018**, **Mohammad Iftikhar Rahman Noman** bearing Metric No. **ET-141037** of session **Spring 2014** to the department of Electrical & Electronic Engineering, International Islamic University Chittagong, has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of Bachelor of Science in Engineering and approved for the examination held on **19 May, 2018**.

____

Supervisor
Mohammed Abdul Kader
Assistant Professor,
Department of Electrical and Electronic Engineering
International Islamic University Chittagong.

DECLARATION

It is hereby declared that this work has been done by us and no portion of the work contained in this thesis/project has been submitted elsewhere for the award of any degree or diploma.

Rubel Ahmed

Mohammad Iftikhar Rahman Noman

ACKNOWLEDGMENT

All praises and thanks to Allah, the Lord of the world, the most Beneficent, the most Merciful for helping us to accomplish this work.

We are beholden to a number of people, who supported us to carry out this work .We would also grateful to our project supervisor Mohammed Abdul Kader (Assistant Professor,Department of Electrical and Electronic Engineering, IIUC) for helping us to get started by giving the basic ideas and concepts we required for the project and for his continuous guidance and support. we would like to thank our two senior brother Rabbi Al Mamun and Zakaria Islam they help us to give solution in several problem. Finally we would like to thank our family and friends for putting up with our constant moaning about the project and supporting us through the difficult times.

Authors

ABSTRACT

Bengali Braille kit is a very essential device for learning of visually impaired people in Bangladesh. It is the method which helps them to learn bangle alphabet .The literacy rate in case of visually impaired people in our country in very low .Various factors lack government initiatives, limitations of tutors lack of personal attention etc are the possible reason of this state. Our project aims at developing a self learning Braille device which can help people with visually impairments .The solution aids at developing an easy to learn device that will behave as a teacher and teaches the visually impaired people for learning the Braille learning system .This device is operate by single AVR microcontroller. Braille device which consists of two (3*2) matrix cell one is (3*2) solenoid switch matrix cell and other is (3*2) IR sensor cell which is as like as the Braille cell. Also IR matrix cell developed one extra IR sensor which give instruction to the user to placed card on the sensor board properly .The device mainly operate two different modes i.e. learning mode and training mode .The user first listen the instruction and select the mode operation. In learning mode the Braille sensor cell sense the Braille character from the card and pronounce corresponding output to the speaker and show these Braille character in solenoid matrix cell .In training mode the solenoid matrix cell sequentially show the bangle alphabet character and pronounce corresponding Alphabet to the speaker. Our project is an attempt to utilize technology to educate the visually impaired students.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1. Introduction

Bengali is the fourth most popular language in the world more than 210 million people in the world are speaking in Bengali language[1-2].About 0.75 million people of Bangladesh are visually impaired. It is a big challenge to provide education for them due to unavailability of education materials and expensive Braille system [2]. The Braille is a universal code for making character set of narrow language. The code is nothing but a combination of six dots. There are five special school of visually impaired people are situated in five division of Bangladesh. Each school has a capacity of 50 student so in total 250 student can study in the schools .But there are only one government Braille publication centre at tongi [2].Book published from this press are insufficient .The Braille printer is expensive and not portable.

We come up with a solution to strengthen the education for the visually impaired students at their early stages. In the initial stage of the schooling, teaching the Braille script to the visually impaired student is elementary. So the self learning Braille kit would be fully fledged device which will facilitate the learning of Braille and meets the objective in a cost effective manner.

1.2. Motivation

Self-reliance is a word that gives a value to any person by generating self-esteem in her/him .If is imperative for physically challenged to be self-reliant .To enable these people, It is our responsibility to provide them special care without affecting their self-esteem.so the first step in this regard is to give them the quality education . In the era of technology ,One must utilize the technical knowledge to increase the quality of education to enable physically challenge people is the society in order to avoid inferiority complex which is major curse on just society .Being Engineering student we have thought it would be better to do the project which has humanitarian approach. Particularly which addresses the problem of physically challenged people.

1.3. Objectives

The objectives of the project is to construct a self-learning Braille device which will have the following properties-

i. To increase the literacy rate of visually impaired people by this device.
ii. To enable reading Braille pattern from card and play sound of that Alphabet in speaker.
iii. To generate Braille pattern by using six solenoid switches and play sound of corresponding pattern to train visually impaired people.
iv. To learn the Bengali alphabet through this device without the help of any other person.

1.4. Chapter outline

Six chapters are covered in the course of design and development of this project. The chapters and their contents are as follows:

1. Chapter one is the introductory chapter that gives the objective, motivation.
2. Chapter two is literature review. All previous work related of this project discussed in this chapter.
3. Chapter three is hardware description. In this chapter all the main components used in this project are described elaborately.
4. Chapter four is system design of the project. In this chapter block diagram, flow chart, circuit description is elaborately described.
5. Chapter five is system implementation and results. This chapter also describes about upgradability of system.
6. Chapter six is conclusion. Advantage and future work of this system are described in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the previous project done by other researchers will be discussed. Also it will give a basic idea about the Braille kit.

2.2 Previous Work

Previous works of this project are described as follows.

2.2.1 E-Braille a self-learning Braille device

E-Braille is a electronic Braille device which help visually challenged people in learning the Braille. Braille is well known is an essential means of communication for visually challenged people which enable to read and write with the help of this device the visually challenged people learn easily without help of other person.

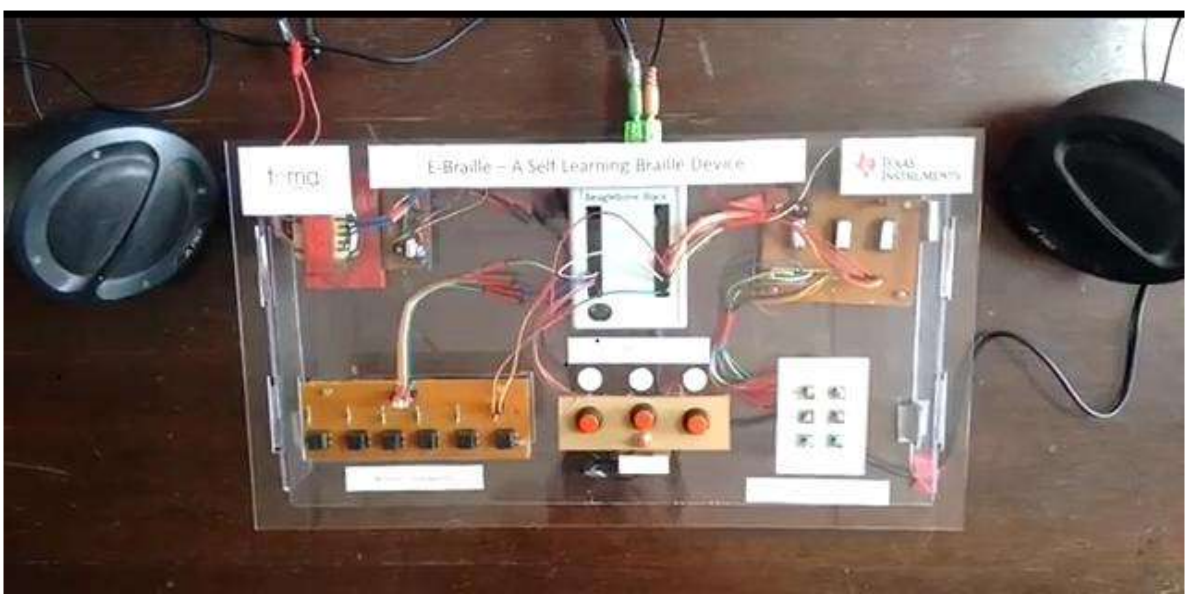


Fig 2.1: E-Braille a self-learning Braille device. [3]

E-Braille is a self-learning Braille device shown in **“Fig 2.1”.**The working of the device into three important modes-learning mode, search mode and quiz mode.

Mode-1: Learning mode

In this mode, system will generate Braille pattern on Braille cell sequentially for all alphanumeric characters. Audio samples for corresponding alphanumeric character will be played sequentially. For this purpose, Matlab code has been written to play single audio sample at a time and to access GPIO pins of Arduino to drive corresponding solenoids simultaneously.

Mode-2: Search mode

This mode is designed to know about Braille pattern of any alphanumeric character. In this mode, user can ask for Braille pattern of any alphanumeric character using microphone. The audio sample is captured and recorded via microphone in Matlab. The recorded audio signal is given to the speech processing algorithm for speech recognition. As per the recognized character, system will generate Braille pattern by sending the control signal to Arduino via serial port for corresponding pattern. It will access GPIO pins of Arduino to drive the solenoids of Braille cell.

Mode-3: Quiz mode

This mode is designed to test user's knowledge about Braille system. The system would ask random question to user related to Braille pattern. This mode is divided into two sub-modes. We have created database of some questions along with the respective answer to test user proficiency in Braille system. The database is divided into two parts for the sub-modes. The system can ask questions from both the sub-modes randomly.

2.2.2 Electronic Progressive Braille Learning Kit for Blind

Electronic Progressive Braille Learning Kit enables the Visually Impaired Students to expand their knowledge in a self-paced easy manner and it also makes their life easier to gain the expertise over Braille script. The kit enables the Visually Impaired Students to be independent without the need of constant guidance and monitoring from the teacher to recognize and practice the patterns of Braille script.

The Braille keypad positioning of 6 keys is done in compliance with standard Braille matrix shown in **"Fig 2.2"**. The design of the system is done keeping the requirements of the end user in mind. We analyzed the problem and divided the working of the device into four important modes – Instruction Mode, Audio Mode, Tactile Mode and Advanced Mode.

Instruction Mode: Instructs the user on the application installed into the kit in the form of audio.

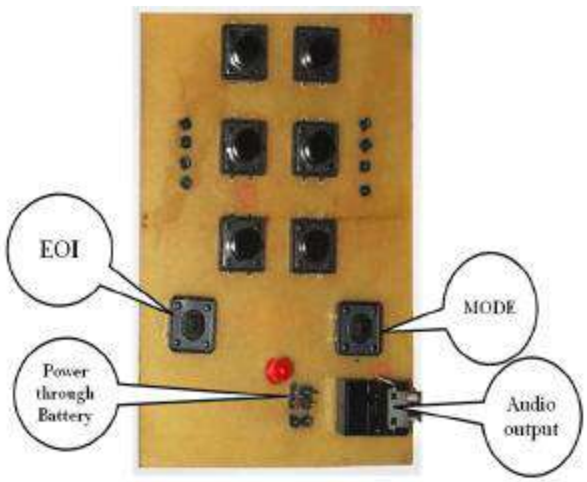


Fig 2.2: Braille Keypad [4]

Audio Mode: This mode enables the user to know various combinations of Braille Dots of letters/words through audio output.

Tactile Mode: This mode enables the user to enter the combination of various dots allowing user to practice the things that are learnt in Audio Mode.

Advanced Mode: Multi Line Braille Display to get tactile feedback.

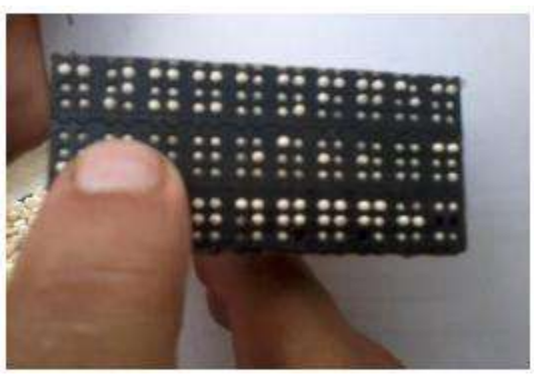


Fig 2.3: Typical Multi-Line Braille Display

2.2.3 Interactive Generalized Keyboard Driver for Bengali Braille Embosser

The system ready to work as shown connect with power supply. When any one press key of generalized keyboard conceded with the system, related sound file is played and store the character in SD card. Visual impaired person hear the sound of the character that pressed on the keyboard and correct it immediately if necessary. After completing the writing, user can playback entire document and use the pause button to stop the playing and able to editing the text at any poison of the document. In this case user uses arrow keys of the keyboard. When the cursor the passes by string any arrow key sound of the corresponding character is played. So visually person can edit the document independently. Overall the system is interactive and useful for any person visual or visually impaired.

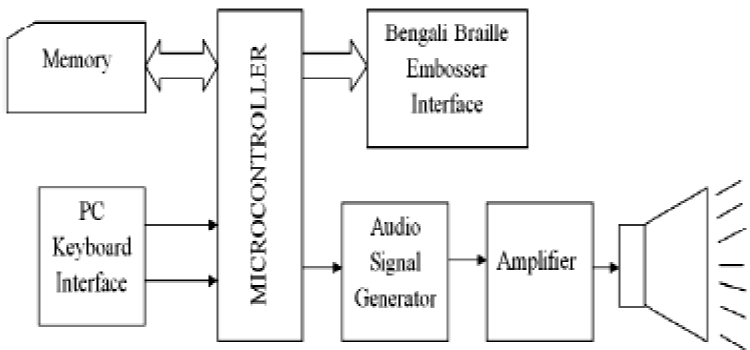


Fig 2.4: General block diagram of interactive generalizes keyboard driver for Bengali Braille embosser.[5]

General block diagram of interactive generalizes keyboard driver for Bengali Braille embosser shown in **"Fig 2.4".** This interactive driver uses microcontroller name Atmaga328P is a low cost edition of microcontroller. It is an 8bits microcontroller [6].

2.3 Comparison with Our Project

We developed a device for visually impaired people. Through this device we can teach visually impaired people the Bengali alphabet .With the help of this device the visually impaired people can take a complete idea of every Bengali alphabet in the Braille character. This device works in two modes one is learning mode and other is training mode. These modes operations have been created in different ways which do not match any previous project.

CHAPTER 3

COMPONENTS

3.1 Introduction

In this chapter main components of this project will be discussed. Pin description, pin diagram and feature of components will be discussed.

3.2 List of Components

- AVR Microcontroller (ATMEGA2560)
- Infrared (IR) sensor
- IR Transmitter
- IR Receiver
- LM358M
- IC LM386
- Power Transistor BD139
- 12V Solenoid Switch

3.3 Description of Components

Description of required components is given below

3.3.1 AVR Microcontroller

The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed.

3.3.1.1 Features of ATMEGA2560

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  – 135 Powerful Instructions
  – Most Single Clock Cycle Execution
  – Up to 16 MIPS Throughput at 16MHz

– On-Chip 2-cycle Multiplier

• High Endurance Non-volatile Memory Segments

– 64K/128K/256KBytes of In-System Self-Programmable Flash

– 4Kbytes EEPROM

– 8Kbytes Internal SRAM

– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

– Data retention: 20 years at 85□C/ 100 years at 25□C

– Optional Boot Code Section with Independent Lock Bits

• In-System Programming by On-chip Boot Program

• True Read-While-Write Operation

– Programming Lock for Software Security

• Endurance: Up to 64Kbytes Optional External Memory Space

• I/O and Packages

– 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)

– 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)

• Temperature Range: – -40 C Industrial°C to 85°

• Ultra-Low Power Consumption

– Active Mode: 1MHz, 1.8V: 500µA

– Power-down Mode: 0.1µA at 1.8V

• Speed Grade: – ATmega640V/ATmega1280V/ATmega1281V:

• 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V

– ATmega2560V/ATmega2561V:

• 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V

– ATmega640/ATmega1280/ATmega1281:

3.3.1.2 Pin function of Microcontroller

Table 3.1 describes the pin function of ATMEGA2560 .

Table 3.1: Pin description of ATMEGA2560

Pin Name	Pin Function
VCC	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).
Port C (PC7..PC0)	Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).
Port D (PD7..PD0)	Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. (selected for each bit).
Port E (PE7..PE0)	Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).
Port G (PG5..PG0)	Port G is a 6-bit I/O port with internal pull-up resistors (selected for each bit).The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running
RESET	A low level on this pin for longer than the minimum pulse length will generate a reset
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

3.3.2 Infrared (IR) sensor

An infrared sensor is an electronic device that emits in order to sense some aspects of the surroundings. These types of sensors measures only infrared radiation, rather than emitting it that is called as a passive IR sensor. Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These types of radiations are invisible to our eyes that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and these output voltages, change in proportion to the magnitude of the IR light received.

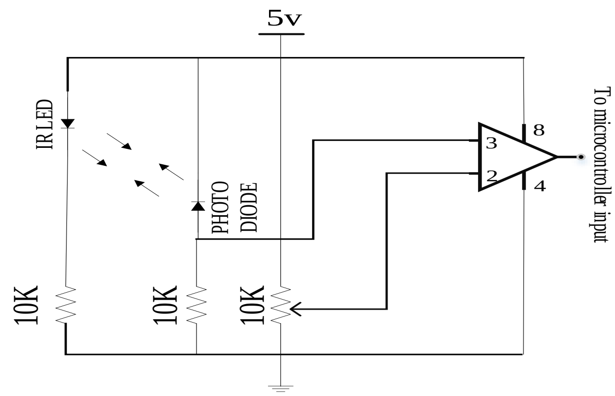


Fig 3.1 : Circuit diagram for Infrared ray sensor[7].

Describes the circuit diagram for Infrared Ray (IR) sensor shown in **"Fig 3.1".**

Fig 3.2: A typical Infrared LED .

Infrared Transmitter is a light emitting diode (LED) shown in "**Fig 3.2".**which emits infrared radiations. Hence, they are called IR LED's. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye.

Fig 3.3: A typical IR receiver or a photodiode.

Infrared receivers are also called as infrared sensors as they detect the radiation from an IR transmitter. IR receivers come in the form of photodiodes and phototransistors. Infrared

Photodiodes are different from normal photo diodes as they detect only infrared radiation. **"Fig 3.3".** A typical IR receiver or a photodiode is shown below.

IR Sensors work by using a specific light sensor to detect a select light wavelength in the Infra-Red (IR) spectrum. **"Fig 3.4"** shows the working principle of IR sensor.

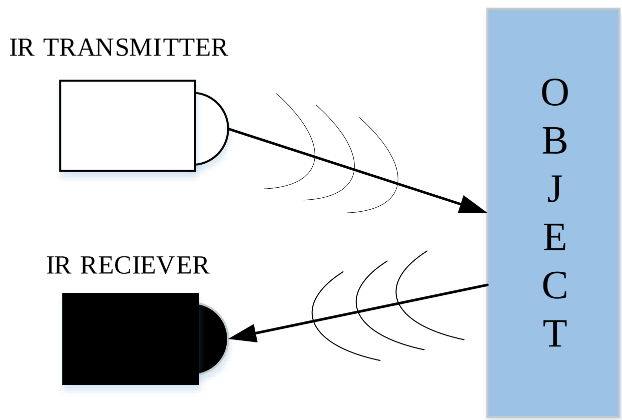


Fig 3.4: principle of IR Sensor [7].

The sensor emits IR light and gives a signal when it detects the reflected light. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, its resistance and correspondingly, its output voltage, change in proportion to the magnitude of the IR light received. This is the underlying principle of working of the IR sensor.

3.3.3 LM-358M

The LM158 series consists of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. Application areas include transducer amplifiers, dc gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems.

“Fig 3.5” shows the IC-LM358M pin diagram.

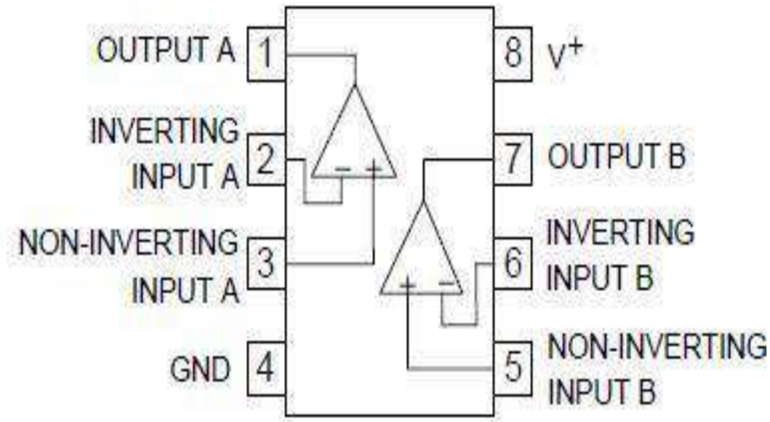


Fig 3.5: IC-LM358M pin diagram [8]

An operational amplifier (op-amp) of LM 358M is used as comparator circuit shown in **“Fig 3.5”**. When the IR receiver does not receive a signal, the potential at the inverting input goes higher than that non-inverting input of the comparator IC (LM358M). Thus the output of the comparator goes low. When the IR receiver module receives signal to the potential at the inverting input goes low. Thus the output of the comparator (LM 358M) goes high. A variable resistor is used to output terminals of the IC to set the sensitivity of the circuit Diagram.

Table 3.2 describes the pin function of IC-LM358M .

Table 3.2: Pin description of IC-LM358M

Pin No	**Function**	**Name**
1&8	Output of the comparator	OUTPUT
2&6	Inverting input	-INPUT
3&5	Non-inverting input	+INPUT
4	Ground terminal	GND
8	Supply	VCC+

3.3.3.1 Features

- Available in 8-Bump DSBGA Chip-Sized Package
- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain: 100 dB
- Wide Bandwidth (Unity Gain): 1 MHz(Temperature Compensated)
- Wide Power Supply Range:

– Single Supply: 3V to 32V

– Or Dual Supplies: ±1.5V to ±16V

- Very Low Supply Current Drain (500μA)—Essentially Independent of Supply Voltage
- Low Input Offset Voltage: 2 mV
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Large Output Voltage Swing

3.3.4 LM386 Low Voltage Audio Power Amplifier

The LM386M-1 is power amplifiers designed for use in low voltage 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-V supply, making the LM386M-1 deal for battery operation.

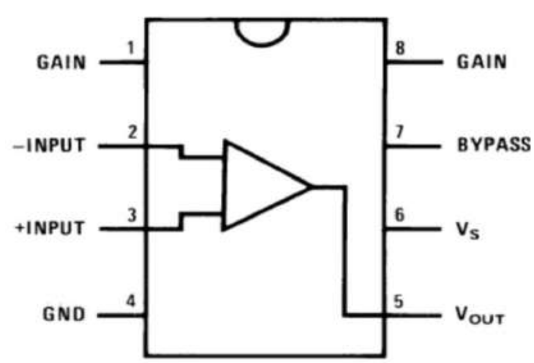


Fig 3.6: LM-386 pin diagram[9].

Pin diagram of IC-LM-386 shown in **“Fig 3.6”.**there are two pins (1 and 8) which provided for gain control. With pins 1 and 8 open the 1.35-kΩ resistor sets the gain at 20 (26 dB). If a capacitor is put from pin 1 to 8, bypassing the 1.35-kΩ resistor, the gain will go up to 200 (46 dB). If a resistor is placed in series with the capacitor, the gain can be set to any value from 20 to 200. Gain control can also be done by capacitive coupling a resistor (or FET) from pin 1 to ground. the IC LM386 Audio Amplifier can be separated into three functional blocks, namely Gain Control, Power & Output, Bypass. The circuit design of LM386 is very simple. First connect the power supply pins namely pi4 and pin6 to voltage supply and ground respectively.

Table 3.3 describes the Pin functions of IC-LM386.

Table 3.3: Pin description of IC-LM386

Pin No	**Function**	**Name**
1	Gain setting pin	GAIN
2	Inverting Input	-INPUT
3	Non-inverting input	+INPUT
4	Ground reference	GND
5	Output	V_{out}
6	Power supply voltage	V_s
7	Bypass decoupling path	BYPASS
8	Gain setting pin	GAIN

3.3.4.1 Features

- Battery Operation
- Minimum External Parts
- Wide Supply Voltage Range: 4 V–12 V or 5 V–18 V
- Low Quiescent Current Drain: 4 mA
- Voltage Gains from 20 to 200
- Ground-Referenced Input
- Self-Centering Output Quiescent Voltage
- Low Distortion: 0.2% (AV = 20, VS = 6 V, RL = 8 Ω, PO = 125 mW, f = 1kHz)

• Available in 8-Pin MSOP Package

3.3.5 Power Transistor BD139

NPN power transistor. These epitaxial planar transistors are mounted in the SOT-32 plastic package. They are designed for audio amplifiers and drivers utilizing complementary or quasi-complementary circuits.

"Fig 3.7" shows the pin diagram of NPN power transistor BD139

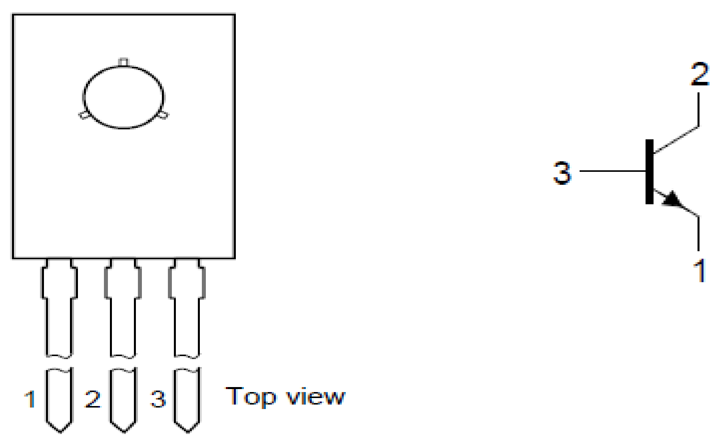


Fig 3.7:PIN diagram of BD139[10].

Table 3.4 describes the Pin functions of BD-139.

Table 3.4: Pin description of BD-139

Pin No	Function
1	Emitter
2	Collector, connected to metal part of mounting surface
3	Base

3.3.5.1 Features

· High current (max. 1.5 A)

· Low voltage (max. 80 V).

3.3.6 12V Push Solenoid

A solenoid is a very simple component that includes a coil of wire that is covered around a core made out of a metal. When a current is applied to the solenoid, it has the effect of assembling a consistent magnetic field. Electricity changes to magnetism then it changes to electricity and therefore these two forces are united into one. An attractive thing about the uniform field in a solenoid is that, if the solenoid has an immeasurable length, the magnetic field would be the similar everywhere along the element. In a solenoid, sometimes this translates to very small electrical components being able to do a marvelous amount of work.

Electric solenoids work on similar electromagnetic principles to those of DC motors, however, solenoids can use the magnetic energy to push or pull something rather than turn it. Solenoids are found in paintball guns, pinball machines, printers, valves and even automobiles. Solenoid is a coil that when energized, produces a controlled magnetic field down through its centre. By placing a magnetic armature inside that field, the armature can move in or out of the coil.

Fig 3.8: 12V Push Solenoid [11].

CHAPTER 4
SYSTEM DESIGN

4.1 Introduction

Design procedure is the most important part of a project. In this chapter interfacing different devices with microcontroller, designing the block diagram, circuit diagram and flow chart will be described.

4.2 Interfacing different devices with microcontroller

This project includes the interfacing of IR sensors, interfacing of push solenoid, interfacing of Audio amplifier circuit, interfacing of SD card with the microcontroller. All of the interfacings are shown in bellows.

4.2.1 Interfacing IR Sensors

In this project the IR sensor cell is developed by the combination of seven IR sensor. First six IR sensor are assign the Braille character which is given in card and the last IR sensor is define as a match bit which identify the card placed right or wrong .

Each of The IR sensor circuit has been created by IR led and photodiode. The IR led which transmit the IR rays and the photodiode which received the IR rays. The Braille character card are design by the black color. When this card are put on the IR sensor board, the transmitter of the IR sensor transmit IR rays on the card due to black color are assign Braille character on the card so that the black color absorbed all IR rays for this reason no IR ray can go to IR receiver. Depending on the receiver the comparator will give an output. Here an operational amplifier (op-amp) of LM 339 is used as comparator circuit.

Interfacing the IR sensor with microcontroller shown in **"Fig 4.1".** The sensor are connect with microcontroller pin no. 30,31,32,33,34,35 and pin no.36 that's are defined in program. So all of the sensors act as input of the microcontroller .

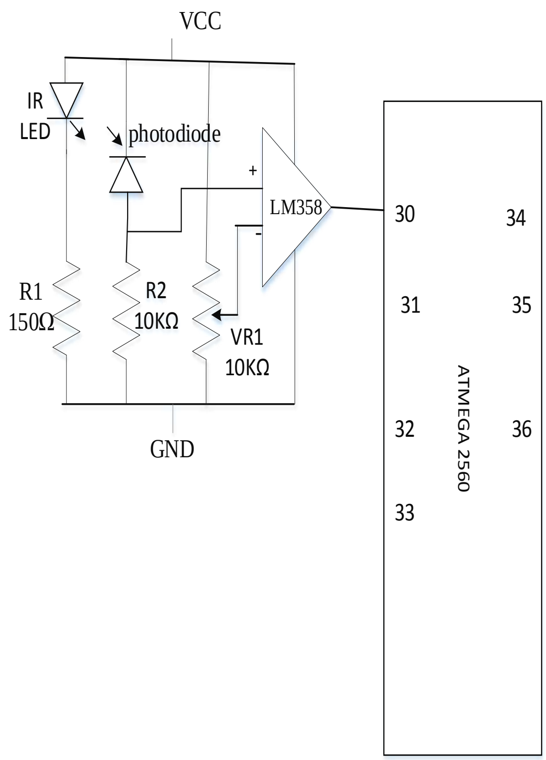


Fig 4.1: Interfacing IR sensors with microcontroller

The high or low level logic of a digital pin of ATMEGA 2560 is defined by following functions-

pinMode() function defines a digital pin as INPUT/OUTPUT. If we want to define pin number 30 as INPUT we have to write-

pinMode(30, INPUT);

and digitalWrite() function is used to make a digital pin HIGH or LOW. The format of the function is expressed as-

digitalWrite(30, HIGH/LOW);

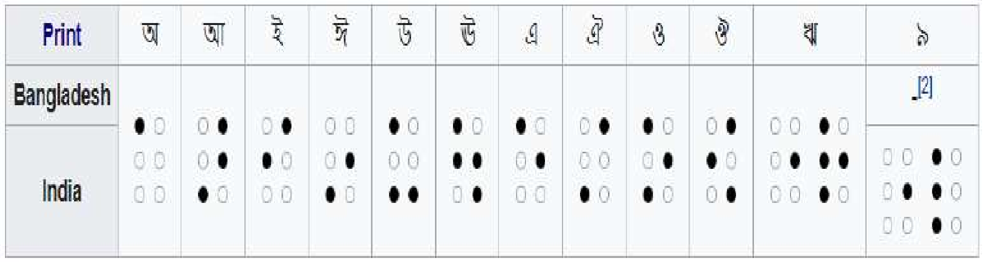
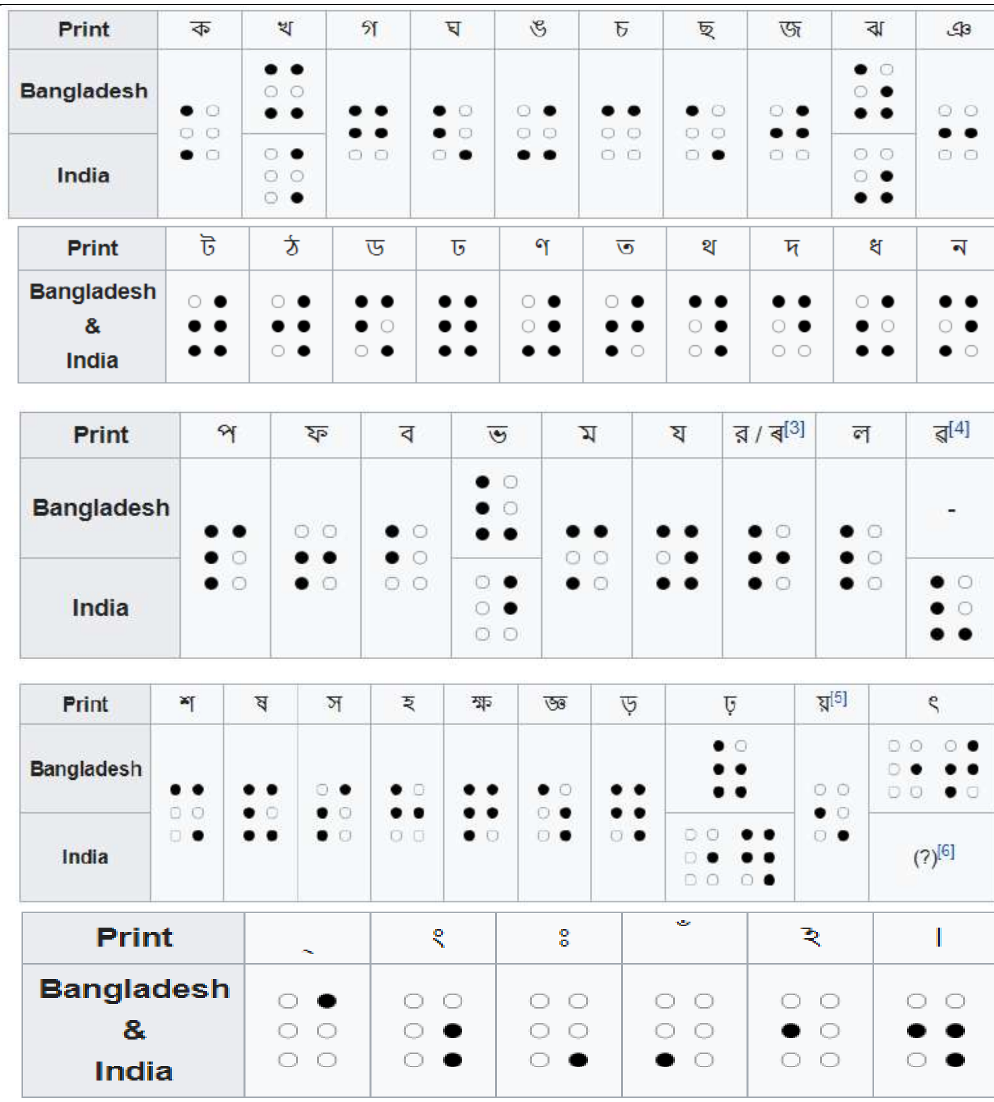
As example if we want to make the digital pin 30 as LOW we have to write-

digitalWrite(30,LOW);

Braille Character card have been made according to this format shown below:-

Table 4.1 describes the Bangle Alphabet Braille Character

Table 4.1: Bangle Alphabet Braille Character

Vowel Alphabets	
Consonant Alphabets	

4.2.2 Interfacing Audio Amplifier circuit.

The IC LM386 integrated is a low-power audio frequency amplifier, which uses batteries for the low-level power supply in electric circuit. The Audio amplifier is used to increase the amplitude of audio signal. This amplifier is intended as an 8-pin mini DIP package. This offers voltage amplification of 20. By using exterior parts, voltage gain can be higher up to 200. When it works from a 6V power supply, inactive power will be 24 mile-watts, which creates IC LM386 for perfect battery operation.

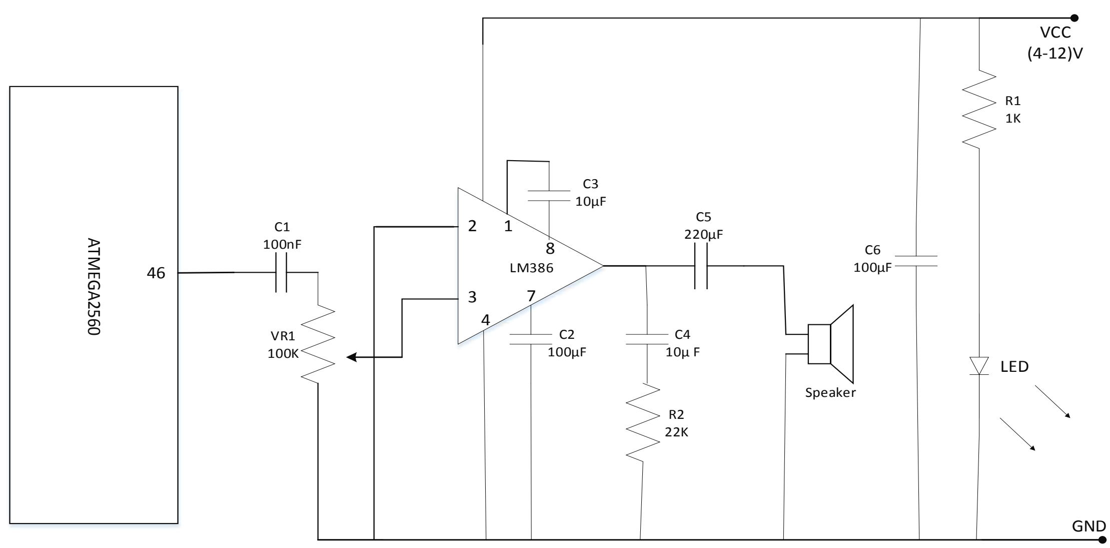


Fig 4.2: Interfacing the Audio Amplifier circuit with Microcontroller

An amplifier circuit connected between the 46 pin of the microcontroller and the speaker is connected with the amplification circuit shown in **"Fig 4.2".**The amplification circuit will increase the volume of the speaker. This circuit using the LM386 and a bunch of components. The supply is given in the LM386 connecting the external external 6V battery. The gain of the amplifier is given by the capacitor connected to pin 1 and 8 of the LM386. With the 10 µF capacitor the gain is set to 200, without the capacitor the gain is 50.the potentiometer connected series with capacitor to control the input and the capacitor is used to filter the input. In pin 7 of the LM386 connected with a bypass capacitor for decreasing the noise. The speaker connected through a capacitor for filtered out the excessive DC signals.

4.2.3 Interfacing SD card.

An SD card is a non-volatile memory card used extensively in portable devices. It works with standard Micro SD Cards which operating voltage is 3.3 V. Therefore, the module has a voltage regulator and a level shifter so that we can use it with the 5 V pins of the Arduino Board.

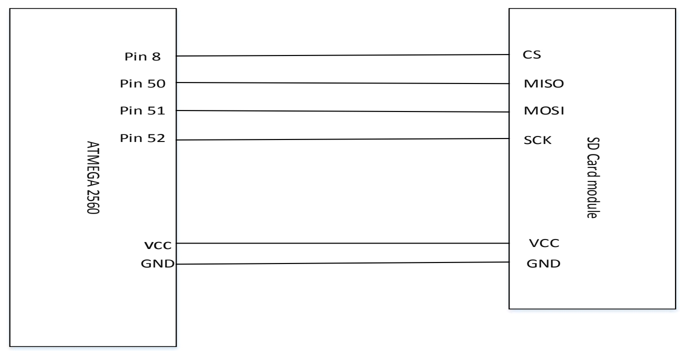


Fig 4.3: Interfacing the SD card with Microcontroller

Interfacing SD-card module with microcontroller shown in **"Fig 4.3".** So first we need to include the standard SD and SPI libraries, create a "File" object and define the Chip-Select pin of the SPI bus, the pin 53 in my case for the Arduino Mega Board. For this example we want our code to be executed only once, so all the code will be placed in the "setup" section, while the "loop" section will remain empty.

So first we need to start the serial communication and define the Chip-Select pin as output. We have to do this because the Chip-Select pin needs to be "Low" so that the SPI communication between the module and the Arduino works.

Next, using the SD.begin() function we will initialize the SD card and if initialization is successful the "if" statement will become true and the String "SD card is ready to use." will be printed on the serial monitor, else the string "SD card initialization failed" will be printed and also the program will be terminated.

The .wav files are loaded in micro-SD card. Then it generates a signal and outputs it through the speaker connected to digital pin 46. This makes the speaker create sounds and play music. It can play many different songs saved on the micro-SD card.

The SD card module pin function describe below:-

1) Chip Select: Enables/Disables the slave device (SD Card)

2) SCLK: Preset scaled clock applied to SD Card (250KHz)

3) MOSI (Master Out Slave In): Carries data from ATMEGA2560 to SD CARD.

4) MISO (Master In Slave Out): Carries Data from SD Card to ATMEGA2560.

The .wav files used in this circuit have a slight limitation in playing audio. Since a transistor is used as an amplifier, it cannot read complex .wav files. Therefore, the .wav files should be converted to have these dimensions:

-Samples Per second (Hz): 11025Hz

-Channel: Mono

-Bits Per Sample: 8bit

-PCM format: PCM unsigned 8-bit

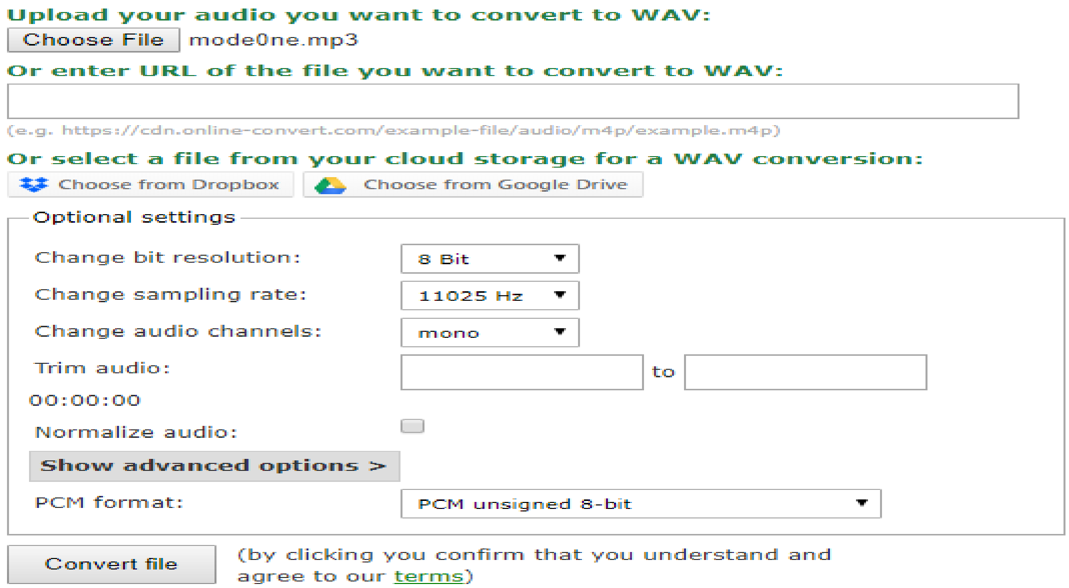


Fig 4.4: Wav file converting format

4.2.3.1 TMRpcm library.

Arduino library for asynchronous playback of PCM/WAV files direct from SD card.Utilizes standard Arduino SD library, SD card and output device (Speaker, Headphones, Amplifier, etc).This library should be added in the Arduino software and execute the Audio signal by following function-

TMRpcm tmrpcm :Create an object for use in this sketch for an example

tmrpcm.play("select.wav"): This function play this signal which are call in it.

tmrpcm.volume(2);This function will increase the volume two level of the audio signal tmrpcm.quality(1);This function will increase the quality of the audio signal.

4.2.4 Interfacing push solenoid.

In this project we build a solenoid cell by using six push solenoid, which are properly operate up to 12V. The solenoid switches are connected with microcontroller pin no.40,41,42,43,44 and 47. Since a solenoid is an inductive load we need to include a diode across the contacts. The diodes help eliminate transient voltages caused when a magnetic coil (such as those found in a motor, relay, or solenoid) suddenly loses power. Without this diode in place the transient voltage spikes can damage other elements of the circuit.

"Fig 4.6" shows Diode is placed from the negative side of the coil to the positive side. Since diodes only allow current to flow in one direction we need to make sure we get this right, otherwise it will be a dead short between power and ground.

The current draw of this solenoid is higher than a standard transistor can handle so we will be using a power Transistor (BD139). A power transistor is actually a pair of transistors that act as a single transistor with a high current gain (1.5A). The pin output is still the same as a standard transistor so it's like as a transistor with a higher current rating.

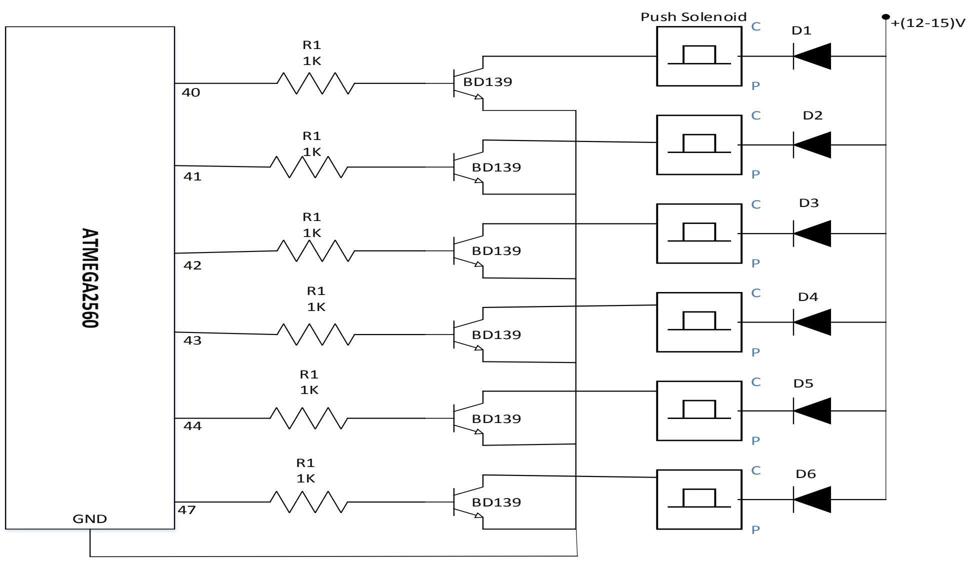


Fig 4.5: Interfacing push solenoid with microcontroller

A base resistor is placed on the base pin of the transistor. This resistor limits the current going to the base (control line) of the transistor. We will be using a 1K ohm resistor in this case; it can be placed from the base of the transistor as shown **"Fig 4.5".**

The high or low level logic of a digital pin of ATMEGA 2560 is defined by following functions for push solenoid-

pinMode() function defines a digital pin as INPUT/OUTPUT. If we want to define pin number 40 as OUTPUT we have to write-

pinMode(40, OUTPUT);

and digitalWrite() function is used to make a digital pin HIGH. The format of the function is expressed as-

digitalWrite(40, HIGH);

4.3 Circuit diagram

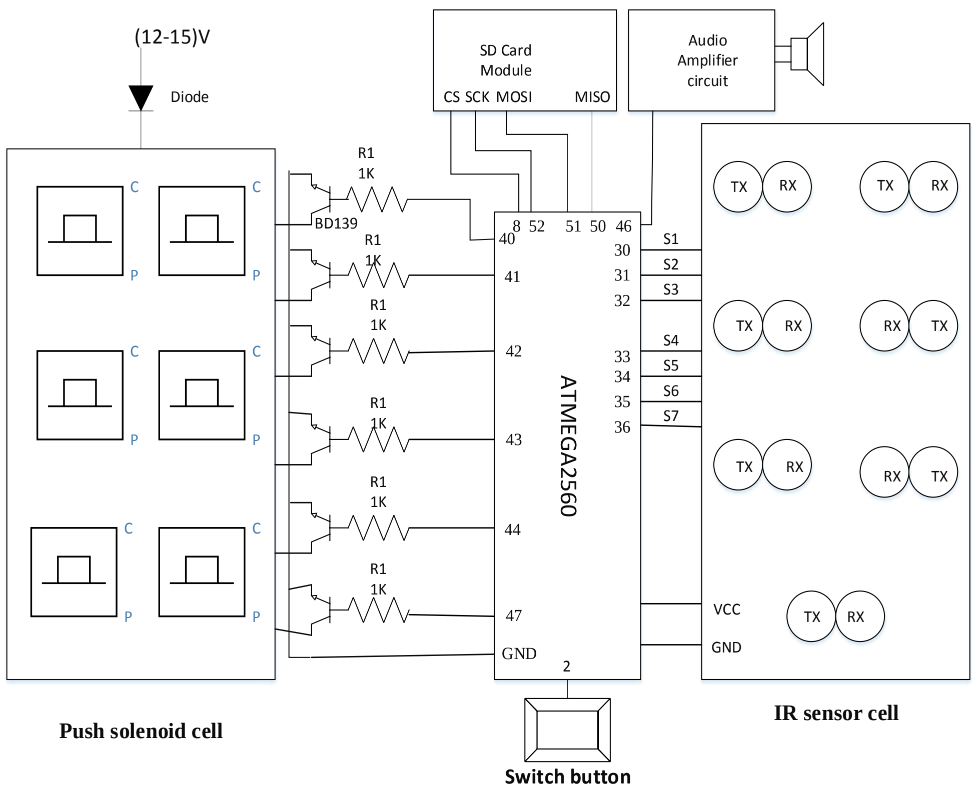


Fig 4.6: block diagram of the Braille circuit

The complete block diagram of the Braille Circuit shows in **"Fig 4.6".** In this circuit the IR sensor cell, push solenoid cell, Audio amplifier circuit and SD card are connected together. Here IR sensor cell has been created with every IR sensor and this cell is parallel connected with Arduino. Similarly the solenoid switch board and SD card are connected with Arduino.

4.4 Flow chart

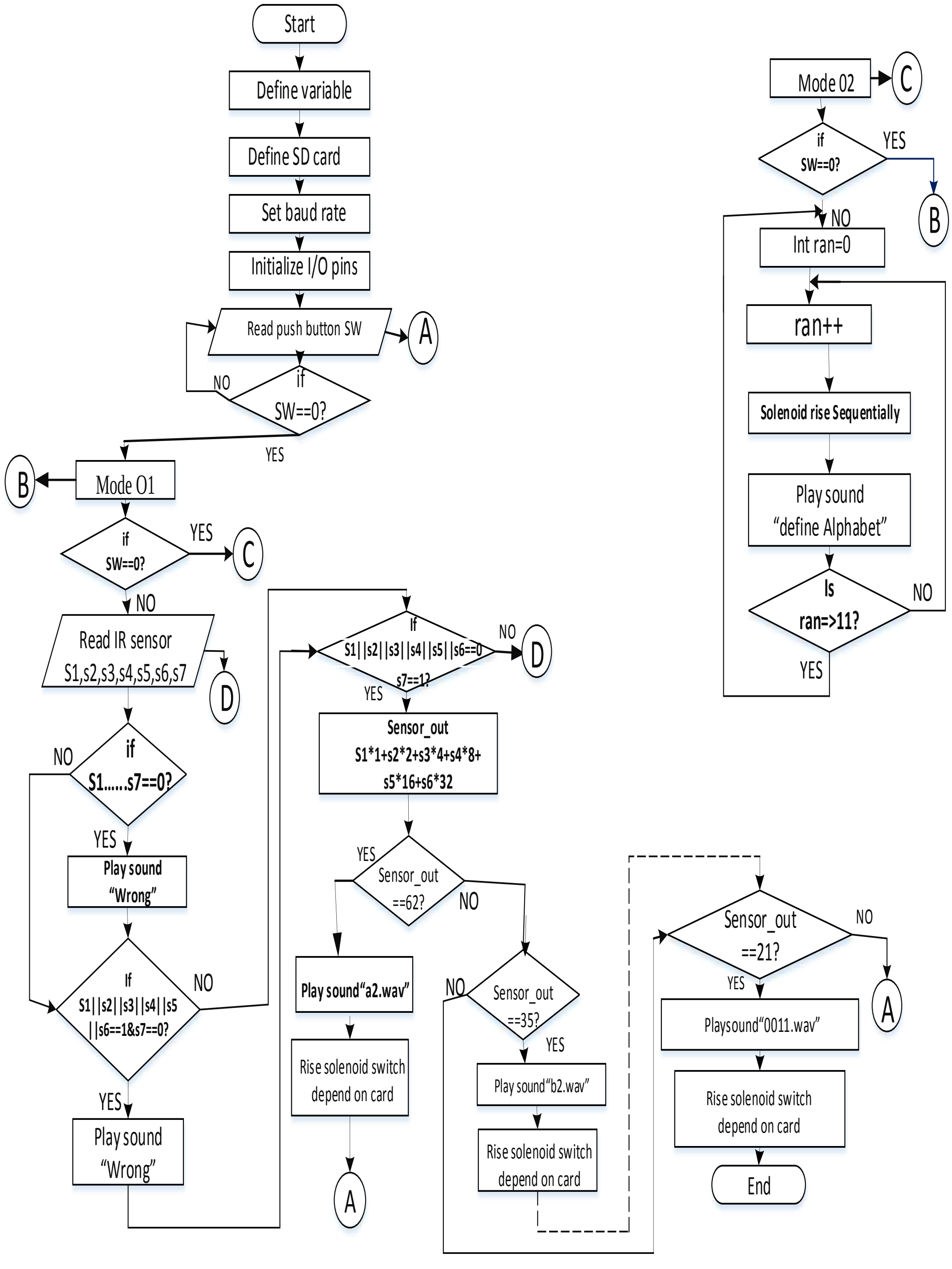


Fig 4.7: Flow chart for Braille Device.

“Fig 4.7” shows the flow chart of the Braille device. In flow chart firstly define the sensor and variable which given in program and then microcontroller read the push button if the button pressed then the program will be operate at mode one. In mode one first read the IR sensors and given feedback to the speaker by audio signal depend on the combination of the sensors. The IR sensor take data from the card which is designed by the depend on Braille character. When the card is placed on the sensor board then the sensor detect the card and then check the all condition in mode one then give feedback by audio signal. In mode one if the button pressed again the device operate in mode two . In mode two operation the solenoid shows the Braille character of the alphabet sequentially and speak the alphabet name to the speaker. But in the mode two operation if the button pressed again then the device operation return to mode one and try to operate in mode one.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 Introduction

This chapter will describe the system implementation, result, upgradability of the system. In this section we will discuss about overall performance of the system and analysis the performance of the system weather its working properly.

5.2 System Implementation

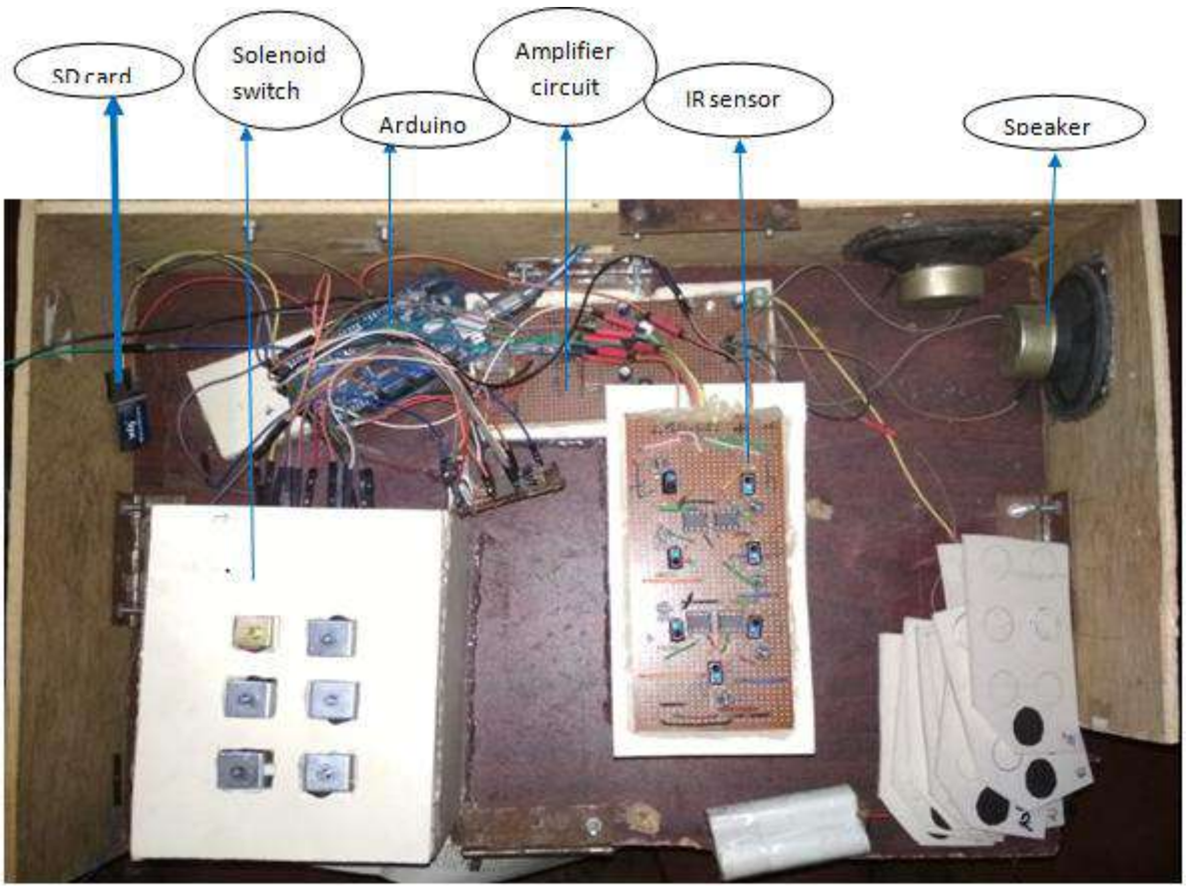


Fig 5.1: Overview of this Project.

The complete circuit connection of the system shows in **"Fig 5.1"**. Here we can see that the Braille cell and IR sensor cell have been implemented separately. Next we can see that some Braille characters have been designed on the card .In learning mode operation when we put card on the sensor cell then the sensor give an output through the sound. In addition the solenoid cell raises the solenoid and show the Braille character on the cell.

5.3 Performance of system

In this section we will discuss about the system regarding working in both modes and also analysis the modes performance. Like as in mode-01 the every function of the system weather it work properly or not and second mode shows the response of the solenoid for every operation.

5.3.1 Analysis

In this article we will analyses regarding special circumstances like when the card put on the sensor cell the sensor detect the Braille character from the card and give an output to speaker. The Braille character of corresponding alphabet shown in solenoid cell.

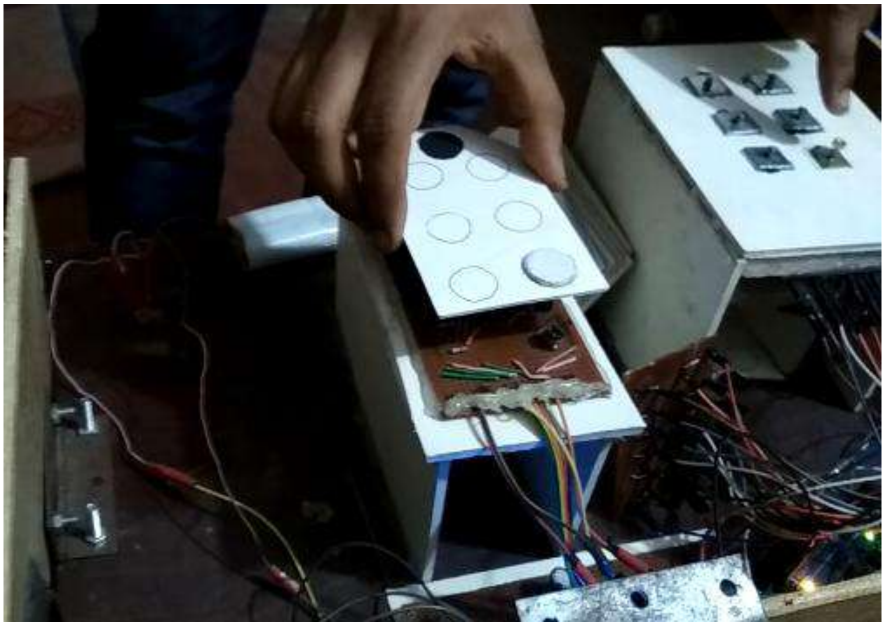


Fig 5.2: Solenoid rise depend on card character.

"Fig 5.2" shows the visually impaired people will be able to mark the Braille character From the Braille cell and besides the blind people can feel the Braille character for each Alphabet.

Fig 5.3: Solenoid rise continuously.

"Fig 5.3" shows the Braille cell continuously show the Braille character depend on following instruction and this functions are execute time delay like 3second.Here we implement first eleven alphabet that are continuously execute one by one in this mode operation.

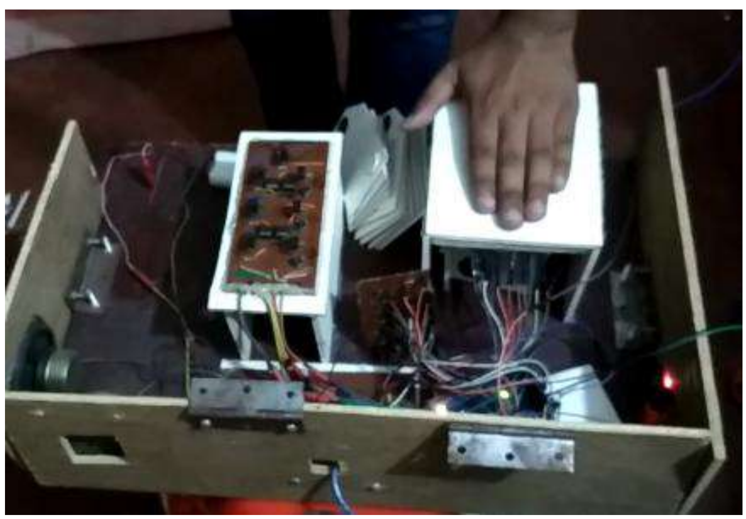


Fig 5.4: Blind people can sense character by hand.

"Fig 5.4" shows the blind people sense the Braille character by his hand. When the solenoids are executed every instruction then some solenoids push and pull depends on the instruction. When the solenoids are rise it hit on the hand of the people so that we can feel the character from the rising pin.

The first five push solenoid draws current of about 0.23A and voltage supply is 12v.So the total power consumption by the each solenoid switch is 2.76W. And last push solenoid draws current of about 0.55A and voltage supply is 12v.So the total power consumption by this solenoid switch is 6.6W.

Table 5.1 shows the approximate Cost Analysis for this project.

Table 5.1 approximate Cost Analysis for this project.

Components	Quantity	Unit Prize	Total prize
Atmega2560	1	1250	1250
Push solenoid	6	660	3960
Tx-Rx photodiode	14	10	140
Capacitor	6	3	18
Resistor	22	0.5	11
Diode	6	2	12
IC LM-358 & IC LM-386	7+1	15	120
Battery 12V(Rechargeable)	1	950	950
Battery 6V (Rechargeable)	1	250	250
Push Switch	1	2	2
Vero Board	3	20	60
Screw	15	5	75
Speaker	1	30	30
PVC sheet		150	150
Wires		150	150
Misc.		150	150
Transistors	6	7	42
Variable Resistor	9	5	45
SD card(2B)	1	200	200
SD module	1	60	60
TOTAL			**7675**

5.4 Prototype Vs Real System

This is made with primarily for research purpose as we design the prototype here in the project. Real system may need some modification to be done. The prototype and the real system are compared bellows:

I. In our prototype one solenoid draws current of about 0.55A and voltage drop is 12v.So the total power consumption by the solenoid is 6.6W.But for practical use all the solenoid in same current rating then the power consumption of solenoid will be less than the prototype.

II. Here in our prototype we have used push solenoid. But in practical purpose if we used another push switch alternative the solenoid then the total cost of the will be reduce.

III. In our prototype we have use one 12v dc battery for the solenoid cell and one 6v dc battery power sources for the Audio amplifier circuit. But in practical purpose by the one single power source can be operate whole operation of the system so that the total cost of the system will be reduce.

IV. In our prototype the size and weight of the system is much. But in practical purpose if it is made by an industry product then the full size of the device and the weight will be reduced.

CHAPTER 6

CONCLUTION AND FUTURE WORK

6.1 Conclusion

The problem of Braille literacy is creating a major problem in enabling the visually disabled people in achieving a rightful place in the society. Use of Braille system is an independent, user friendly, portable and cost effective manner. It can enhance the learning ability of visually challenged people in a comfortable and interactive way. The different modes of operation ensure user friendly approach for designed system. This device can be used effectively to make the learning of Braille easy. It can prove to be a major breakthrough in enhancing the literacy rate for visually challenged people.

Self-learning Braille device enables the Visually Impaired Students to expand their knowledge in a self-paced easy manner and it also makes their life easier to gain the expertise over Braille characters. The device enables the Visually Impaired Students to be independent without the need of constant guidance.

The device is useful for initial learners and primary school students. Visually impaired person can use this device to learn the basic letters. Very similar to how normal student studies the basic alphabets initially. Device can be considered as the friend for life.

6.2 Limitations

This system has some loopholes

i. In mode one when the card take from the sensor board then the sound play
ii. Required more than 13v when three or four solenoids run at a time
iii. In mode two when we try to back in mode one then the switch keep pressed on delay time

6.3 Future work

My project will pave the way for better research related this field.

i. Future developments in the device include reduction in size and making it more portable and battery operated. It can function independently and can prove a great aid in education and empowerment of blind people.

ii. Once our system is successfully implemented, we can make further modifications and improvement to make the propose system more robust. This improvement will be specifically with respect to making the system user independent. Further a more sophisticated micro controller, low power solenoid actuators will make this system more compact.

iii The product can be enhanced with addition of some extra mode operation like English Alphabet, Mathematical number and also can added a testing mode where they give a voice command by the microphone then the solenoid show the alphabet Braille character Depend on voice

REFERENCES

[1] Gahanggir Hossain, Asaduzzaman, M. Ahsan Ullah and A. Saif, "Bangla Braille Embosser: A Tool for Bengali Speaking Blinds", Bangladesh Education Journal (publications of UNESCO BAFED Researcher) Vol. 4, No. 1, June 2005, pp. 49-55.

[2] "Bengali language (2016)" . [Online].Available: [Accessed: 18-April- 2018]. https://en.wikipedia.org/wiki/Bengali_language

[3] Parag Wagh, Uday Prajapati, Mayuresh Shinde, Prafulla Salunke, Vinayak Chaskar, Sanchit Telavane, Vijaypal Yadav, "E-Braille-A Self-learning Braille Device".978-1-5090-2361-5/16/$31.00 ©2016 IEEE

[4] Navinkumar Jadamali, Pravinkumar Jadamali & Kartik Shirsat, Ashwini Basavaradder, "Electronic Progressive Braille Learning Kit for Blind (Low Cost, All Languages and Multiline Braille Screen)" International Journal of Computer Applications (0975 – 8887) International Conference in Distributed Computing & Internet Technology (ICDCIT-2014)

[5] Kishalay Dhar, M. Ashifur Rahman, M. Ahsan Ullah, "Interactive Generalized Keyboard Driver for Bengali Braille Embosser" International Conference on Electrical, Computer and Communication Engineering (ECCE), February 16-18, 2017, Cox's Bazar, Bangladesh

[6] "ATmega2560", Atmel.com, 2017". [Online]. Available: [Accessed: 20- April-2018] http://www.atmel.com/devices/ATMEGA2560.aspx.

[7] "IR Sensor Circuit and Working with Applications", ElProCus - Electronic Projects for Engineering Students, 2017. [Online]. Available: [Accessed: 20- April- 2018]. https://www.elprocus.com/infrared-ir-sensor-circuit-and-working/.

[8] "LM158/LM258/LM358/LM2904 Low Power Dual Operational Amplifiers".[Online].Available. [Accessed: 20- April- 2018]. http://pdf1.alldatasheet.com/datasheet-pdf/view/8674/NSC/LM358M.html

[9] "LM386 Low Voltage Audio Power Amplifier". [Online]. Available [Accessed: 20- April- 2018]. http://www.datasheetspdf.com/pdf/50292/NationalSemiconductor/LM386/1

[10] "BD135; BD137; BD139 NPN power transistors". [Online]. Available[Accessed: 20- April- 2018].http://pdf1.alldatasheet.com/datasheet-pdf/view/16171/PHILIPS/BD139.html

[11] "Different Types of Solenoid, Working Principle and Its Applications" [Online].Available.[Accessed 20-April -2018]..https://www.elprocus.com/different-types-of-solenoid-working-applications/

[12] Akanksha Avhad, Ruchali Jadhav, Abhishek Arbune, Kunal Khainar. "E-BRAILLE-A SELF LEARNING BRAILLE DEVICE" International Journal of Infinite Innovations in Technology|ISSN:2278-9057 IJIIT|Volume-IV|Issue-III|2015-2016 January|Paper-12 Reg. No.:20151212|DOI:V4I3P12.

Appendix

Code for Braille Device:

```
#include <pcmConfig.h>
#include <pcmRF.h>
#include <TMRpcm.h>
#include <pcmConfig.h>
#include <pcmRF.h>
#include <TMRpcm.h>
#include <SD.h>
#include <SPI.h>
#include <Wire.h>
#include <TMRpcm.h>          //  also need to include this library...
#define SD_ChipSelectPin 8  //using digital pin 4 on arduino nano 328, can use other pins
int ran=0;
TMRpcm tmrpcm;   // create an object for use in this sketch
unsigned long time = 0;
boolean SDfound;
#define sensor1 30
#define sensor2 31
#define sensor3 32
#define sensor4 33
#define sensor5 34
#define sensor6 35
#define match_Sensor 36
#define audio_output 46
//boolean match_bit=1, s1=0,s2=0,s3=0,s4=0,s5=0,s6=0;
//int sensor_out=0;
#define sensor
#define ss1 40
#define ss2 41
#define ss3 42
#define ss4 43
#define ss5 44
#define ss6 47
void setup() {
Serial.begin(9600);
tmrpcm.speakerPin = 46; //5,6,11 or 46 on Mega, 9 on Uno, Nano, etc
if (!SD.begin(8))
{
  tmrpcm.volume(2);
  tmrpcm.quality(1);
 Serial.println("SD fail");
  return;
}
pinMode(2,INPUT_PULLUP);
pinMode(sensor1,INPUT);digitalWrite(sensor1,LOW);
pinMode(sensor2,INPUT);digitalWrite(sensor2,LOW);
pinMode(sensor3,INPUT);digitalWrite(sensor3,LOW);
pinMode(sensor4,INPUT);digitalWrite(sensor4,LOW);
pinMode(sensor5,INPUT);digitalWrite(sensor5,LOW);
pinMode(sensor6,INPUT);digitalWrite(sensor6,LOW);
```

```
pinMode(match_Sensor,INPUT);
pinMode(audio_output,OUTPUT);
pinMode(ss1,OUTPUT);digitalWrite(ss1,LOW);
pinMode(ss2,OUTPUT);digitalWrite(ss2,LOW);
pinMode(ss3,OUTPUT);digitalWrite(ss3,LOW);
pinMode(ss4,OUTPUT);digitalWrite(ss4,LOW);
pinMode(ss5,OUTPUT);digitalWrite(ss5,LOW);
pinMode(ss6,OUTPUT);digitalWrite(ss6,LOW);
tmrpcm.play("select.wav");
  }
void loop()
Serial.println("Inside Loop")
int buttonstate=digitalRead(2);if(buttonstate==0){modeOne();
 /* Add speech "Now Mode One" */}
delay(30);
}
void modeOne(){
  delay(1500);
  int buttonstate;
  tmrpcm.play("mode0ne.wav");
 while(1){
  //User program Start
int match_bit=digitalRead(match_Sensor);//Serial.println(match_bit);
int s1=digitalRead(sensor1); int s2=digitalRead(sensor2);int s3=digitalRead(sensor3); int
      s4=digitalRead(sensor4); int s5=digitalRead(sensor5); int s6=digitalRead(sensor6);
if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0 && s6==0 && match_bit==0){Serial.println("put
      the Card Now");}
if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1 && s6==1 &&
      match_bit==0){Serial.println("Card Placed Wrong");tmrpcm.play("wrong.wav");}
if(match_bit==0 && (s1==1 || s2==1 || s3==1 || s4==1 || s5==1 || s6==1)){Serial.println("Card Placed
      Wrong");tmrpcm.play("wrong.wav");}
//If Card placed properly
if(match_bit==1 && (s1==0 || s2==0 || s3==0 || s4==0 || s5==0 || s6==0)){Serial.println("Card Placed
      Properly");
int sensor_out=(s1*1)+(s2*2)+(s3*4)+(s4*8)+(s5*16)+(s6*32);
Serial.print("Card Reading: "); Serial.print(sensor_out);
 switch (sensor_out) {
case 62:
      tmrpcm.play("a2.wav");//do something when var equals 1
       Serial.println("A");
        digitalWrite(ss1,HIGH);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
        digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
       break;
case 35:
       tmrpcm.play("b2.wav");
       Serial.println("002");
        digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);  digitalWrite(ss4,HIGH);
        digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
       break;
case 53:
       tmrpcm.play("c2.wav");
       //do something when var equals 2
```

```
        Serial.println("003");
        digitalWrite(ss1,LOW);  digitalWrite(ss2,HIGH);  digitalWrite(ss3,LOW);  digitalWrite(ss4,HIGH);
         digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
        break;
case 43:
        tmrpcm.play("music4.wav");
        Serial.println("004");
        digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);  digitalWrite(ss4,LOW);
         digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
        break;
case 26:
        tmrpcm.play("music5.wav");
        Serial.println("005");
        digitalWrite(ss1,HIGH);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);  digitalWrite(ss4,LOW);
         digitalWrite(ss5,LOW);  digitalWrite(ss6,HIGH);
        break;
case 12:
        tmrpcm.play("music6.wav");
       Serial.println("006");
        digitalWrite(ss1,HIGH);  digitalWrite(ss2,HIGH);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
         digitalWrite(ss5,HIGH);  digitalWrite(ss6,HIGH);
        break;
case 47:
        tmrpcm.play("music7.wav");
        Serial.println("007");
        digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
         digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
        break;
case 46:
        tmrpcm.play("music8.wav");
        Serial.println("008");
        digitalWrite(ss1,HIGH);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
         digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
        break;
 case 51:
        tmrpcm.play("music9.wav");
        Serial.println("009");
        digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);  digitalWrite(ss4,HIGH);
         digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
        break;
case 42:
        tmrpcm.play("music10.wav");
        Serial.println("0010");
        digitalWrite(ss1,HIGH);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);
       digitalWrite(ss4,LOW);  digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
        break;
case 21:
        tmrpcm.play("music11.wav");
        Serial.println("0011");
        digitalWrite(ss1,LOW);  digitalWrite(ss2,HIGH);  digitalWrite(ss3,LOW);  digitalWrite(ss4,HIGH);
         digitalWrite(ss5,LOW);  digitalWrite(ss6,HIGH);
         break;
```

```
default:
// if nothing else matches, do the default
// default is optional
break;
  }
}
```

// User program End

```
//Mandatory code bellow
buttonstate=digitalRead(2);Serial.print("ModeOne ");Serial.println(buttonstate);
if(buttonstate==0){tmrpcm.play("modetwo.wav");
 Serial.println("Breaking...");
 //break;
 modeTwo(); /* Add speech "Now Mode Two" */
  }
 }
 Serial.println("Breaked ");
 }
```

%% Add speech for “ Mode two”

```
void modeTwo(){
delay(1500);
int buttonstate;
while(1){
            //User program Start
  ran++;
Serial.println(ran);
delay(3000);
switch (ran) {
case 1:
    // statements
    digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
    digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
    delay(3000);
    Serial.println("I Am Kaw");
    digitalWrite(ss1,HIGH);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
    digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
    tmrpcm.play("music1.wav");
    break;

 case 2:
    // statements
    digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);      digitalWrite(ss4,LOW);
    digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
    delay(3000);
    Serial.println("I Am Khaw");
    digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);  digitalWrite(ss4,HIGH);
    digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
    tmrpcm.play("b2.wav");
    break;
 case 3:
```

```
    // statements
        digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);
       digitalWrite(ss4,LOW);  digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
        delay(3000);
        Serial.println("I Am Khaw");
        digitalWrite(ss1,LOW);  digitalWrite(ss2,HIGH);  digitalWrite(ss3,LOW);
        digitalWrite(ss4,HIGH);  digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
        tmrpcm.play("music3.wav");
        break;

case 4:
      digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
       digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
       delay(3000);
       Serial.println("I Am Khaw");
       digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);  digitalWrite(ss4,LOW);
       digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
        tmrpcm.play("music4.wav");
       break;

case 5:
    // statements
      digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
       digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
       delay(3000);
      Serial.println("I Am Khaw");
      digitalWrite(ss1,HIGH);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);  digitalWrite(ss4,LOW);
       digitalWrite(ss5,LOW);  digitalWrite(ss6,HIGH);
      tmrpcm.play("music5.wav");
      break;

case 6:
    // statements
   digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);    digitalWrite(ss4,LOW);
    digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
    delay(3000);
    Serial.println("I Am Khaw");
    digitalWrite(ss1,HIGH);  digitalWrite(ss2,HIGH);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
    digitalWrite(ss5,HIGH);  digitalWrite(ss6,HIGH);
    tmrpcm.play("music6.wav");
    break;
case 7:
     // statements
         digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);
        digitalWrite(ss4,LOW);  digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
         delay(3000);
         Serial.println("I Am Khaw");
         digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);
        digitalWrite(ss4,LOW);    digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
        tmrpcm.play("music7.wav");
        break;
case 8:
```

```
 // statements
 digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);
digitalWrite(ss4,LOW);  digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
 delay(3000);
 Serial.println("I Am Khaw");
digitalWrite(ss1,HIGH);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);
digitalWrite(ss4,LOW);  digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
tmrpcm.play("music8.wav");
break;

case 9:
 // statements
 digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);
digitalWrite(ss4,LOW);  digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
 delay(3000);
 Serial.println("I Am Khaw");
digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);
digitalWrite(ss4,HIGH);  digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
 tmrpcm.play("music9.wav");
 break;

case 10:
 // statements
 digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);  digitalWrite(ss4,LOW);
 digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
 delay(3000);
 Serial.println("I Am Khaw");
digitalWrite(ss1,HIGH);  digitalWrite(ss2,LOW);  digitalWrite(ss3,HIGH);
digitalWrite(ss4,LOW);    digitalWrite(ss5,HIGH);  digitalWrite(ss6,LOW);
tmrpcm.play("music10.wav");
 break;

case 11:
 // statements
 digitalWrite(ss1,LOW);  digitalWrite(ss2,LOW);  digitalWrite(ss3,LOW);
 digitalWrite(ss4,LOW);  digitalWrite(ss5,LOW);  digitalWrite(ss6,LOW);
 delay(3000);
 Serial.println("I Am Khaw");
 digitalWrite(ss1,LOW);  digitalWrite(ss2,HIGH);  digitalWrite(ss3,LOW);
  digitalWrite(ss4,HIGH);  digitalWrite(ss5,LOW);  digitalWrite(ss6,HIGH);
 tmrpcm.play("music11.wav");
 break
default:
  // statements
  break;

 if(ran>=11){ran=0;}
  //User program End
   //Mandatory Code bellow
   buttonstate=digitalRead(2);Serial.print("; Mode Two ");Serial.println(buttonstate);
   if(buttonstate==0){
     Serial.println("Breaking...");
```

```
    //break;
    modeOne();                              /* Add speech "Now Mode One" */
    }
  }
Serial.println("Breaked ");
}
```