

SMART BRAILLE BASED COMMUNICATION DEVICE FOR VISUALLY IMPAIRED PERSONS

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in
Electronics and Communication Engineering

By
HARINI DHEVI S(38130080)
LOGESHWARI M(38130115)



**Department of Electronics and Communication
Engineering
SCHOOL OF ELECTRICAL AND ELECTRONICS
ENGINEERING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY**
(Deemed University)

ACCREDITED WITH GRADE “A” BY NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600119

APRIL 2022



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDECERTIFICATE

This is to certify that this Project Report is the bonafide work of **HARINI DHEVI S**(Reg.No.38130080) and **LOGESHWARI M**(Reg.No.38130115) who carried out the project entitled **“SMART BRAILLE BASED COMMUNICATION DEVICE FOR VISUALLY IMPAIRED PERSONS”** under our supervision from December 2021 to March 2022.

Internal Guide

(Dr.S.Barani)

Dr.T.Ravi,M.E.,Ph.D.,

Head of the Department

Submitted for viva voce Examination held on:

Name:

Signature:

Internal Examiner

External Examine

DECLARATION

We **HARINI DHEVI S(38130080) and LOGESHWARI M(38130115)** hereby declare that the Project Report entitled **“SMART BRAILLE BASED COMMUNICATION DEVICE FOR VISUALLY IMPAIRED PERSONS”** done by us under the guidance of **DR.S.BARANI** at Sathyabama Institute of Science and Technology,**CHENNAI-600119** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1.S.HARINI DHEVI

2.M.LOGESHWARI

DATE:

PLACE:CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully, We are grateful to them.

We convey our thanks to **DR.N.M.Nandhita,Dean,School of Electrical and Electronics Engineering** and **DR.T.RAVI,Head of the Department,Dept.of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would also like to express our sincere and deep sense of gratitude to our internal guide **DR.S.BARANI** for her valuable guidance and suggestions during the progressive reviews for our project.

ABSTRACT

Communication plays a vital role for communicating one person to another person. Human beings who have problem in their vision and hearing, cannot able to communicate one person to another. Here is the some of the unique languages available for the deaf-blind persons which includes tactile signing, Braille, etc. Among these communication for visual and hearing persons one of the most important communication they are using this Braille patterns/Keypad. Braille contains arrangement of dots which form letters, numbers and punctuation marks. The Braille pattern convert this alphanumeric pattern to braille forms. Braille is nothing but it is a system of reading and writing by touch used by the blind. It consists of arrangement of dots which make up letters of the alphabet, numbers, and punctuation marks. In this project an abled person is sending some messages through the deaf blind person through the vibration motors in the palm of the deaf blind persons they can able to sense the message through these motors on the palm after that through matrix keypad they are pressing one button with that the braille keypad starts vibration with that the blind person can able to identify with that LCD board is attached deaf person can able to see the messages also in this proposed project we are implementing that deaf blind person can able to communicate with the abled person through the matrix keypad of predefined strings installed in arduino uno board

**KEYWORDS-BRAILLE DISPLAY, DEAF BLINDNESS, GSM, PIC
MICROCONTROLLER, VIBRATION BAND, VIBRATION MOTORS, MATRIX KEYPAD.**

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
ABSTRACT		v
LIST OF FIGURES		8
1. INTRODUCTION		9
1.1.	Microcontroller	10
1.2.	Embedded Systems	12
1.2.1.	Characteristics Of Embedded Systems	13
1.2.2.	Classification Of Embedded Systems	15
1.3.	Objective	16
2. RELATED LITERATURE		
2.1.	Literature Survey	17
2.2.	Related Papers And Discussions	19
3. PROBLEM DESCRIPTION		
3.1.	Existing System	20
3.1.1.	Problem Definition	20
3.1.2.	Solution adapted	20
3.2.	Proposed System	
3.2.1.	Advantages	21
4. METHODOLOGY		
4.1.	System Requirements	22
4.1.1.	Hardware Required	22
4.1.1.1.	Microcontroller	29
4.1.2.	Digital I/o Pins	29
4.1.3.	ADC Inputs	31
4.1.4.	UART Peripheral	32
4.1.5.	SPI Peripheral	32
4.1.6.	TWI	32
4.2.	PCB DESIGN	33
4.2.1	Keypad	34

4.2.2. Relay Driver	35
4.2.3 SIM800L	39
4.2.4. RF module	51
4.3. Power Supply	64
4.3.1. Software Requirements	75
4.3.2. Arduino IDE	75
4.3.3 GCC	75
5. EXPERIMENTATION	
5.1. Arduino	77
5.1.1. Arduino Configurations	77
5.1.2. Arduino Program	83
5.2. Electrical Block	83
5.2.1. Flow Diagram	84
5.2.2. Circuit Diagram	85
5.2.3. Description	86
6. RESULTS AND DISCUSSION	
6.1. Results	87
6.2. Future Scope	90
6.3. Advantages	90
6.4. Limitations	90
6.5. Photography	91
7.1 CONCLUSION	92
REFERENCE	93

LIST OF FIGURES

Fig No.	Description	Page No.
1	Basic Architecture Of Microcontroller	11
2	Overall Circuit For Arduino UNO Development Board	25
3	Atmega Pin Mapping	27
4	Arduino-UNO Pin Out	27
5	Atmega 328 Functional Block	28
6	Arduino Design(PCB Format not yet released)	30
7	SIM800L Functional Block Diagram	38
8	PCB Layout	48
9	Encoder with keyboard and RF Transmitter	56
10	Decoder With RF Receiver	58
11	Circuit Diagram(Power Supply)	71
12	Flow Diagram	81
13	Electrical Block Diagram	80
14	Circuit Diagram	82
15	The Assembled View of Proposed System	89

CHAPTER 1

INTRODUCTION

1. INTRODUCTION:

In today's world communication plays a vital role in every one's life especially for deaf-blind persons, whereas normal people communicate via mobiles etc. So let's see the deaf-blind people's communication through an abled person's.

Deaf-blind people usually communicate by sign language, tadooma, tactile finger spelling, braille etc. Usually for deaf-blind person, without any knowledge of the sign languages.. The most common languages used by deaf-blind person is of braille format. In this project deaf person can be able to see the screen that is nothing but LCD display but the blind person understands via touching the braille motor. The braille format contains six motors that are arranged one after the other, when the abled person is sending the messages deaf-blind persons understand via vibration. RF LINK in their palm through this vibration in braille motors started after that deaf-blind persons holding the D button for understanding the numbers one after the other and also incoming messages they can identify.

The main objective of this project is to give reply messages from the unabled persons through the abled person is by pressing matrix keypad which contains predefined strings, with that the deaf-blind persons are communication one after the other.

1.1 MICROCONTROLLER:

Even at a time when Intel presented the first microprocessor with the 4004 there was already a demand for microcontrollers: The contemporary TMS1802 from Texas Instruments, designed for usage in calculators, was by the end of 1971 advertised for applications in cash registers, watches and measuring instruments. The TMS 1000, which was introduced in 1974, already included RAM, ROM, and I/O on-chip and can be seen as one of the first microcontrollers, even though it was called a microcomputer. The first controllers to gain really widespread use were the Intel 8048, which was integrated into PC keyboards, and its successor, the Intel 8051, as well as the 68HCxx series of microcontrollers from Motorola. Today, microcontroller production counts are in the billions per year, and the controllers are integrated into many appliances we have grown used to, like

- Household appliances (microwave, washing machine, coffee machine, . . .)
- Telecommunication (mobile phones)
- Automotive industry (fuel injection, ABS, . . .)
- Aerospace industry
- Industrial automation.

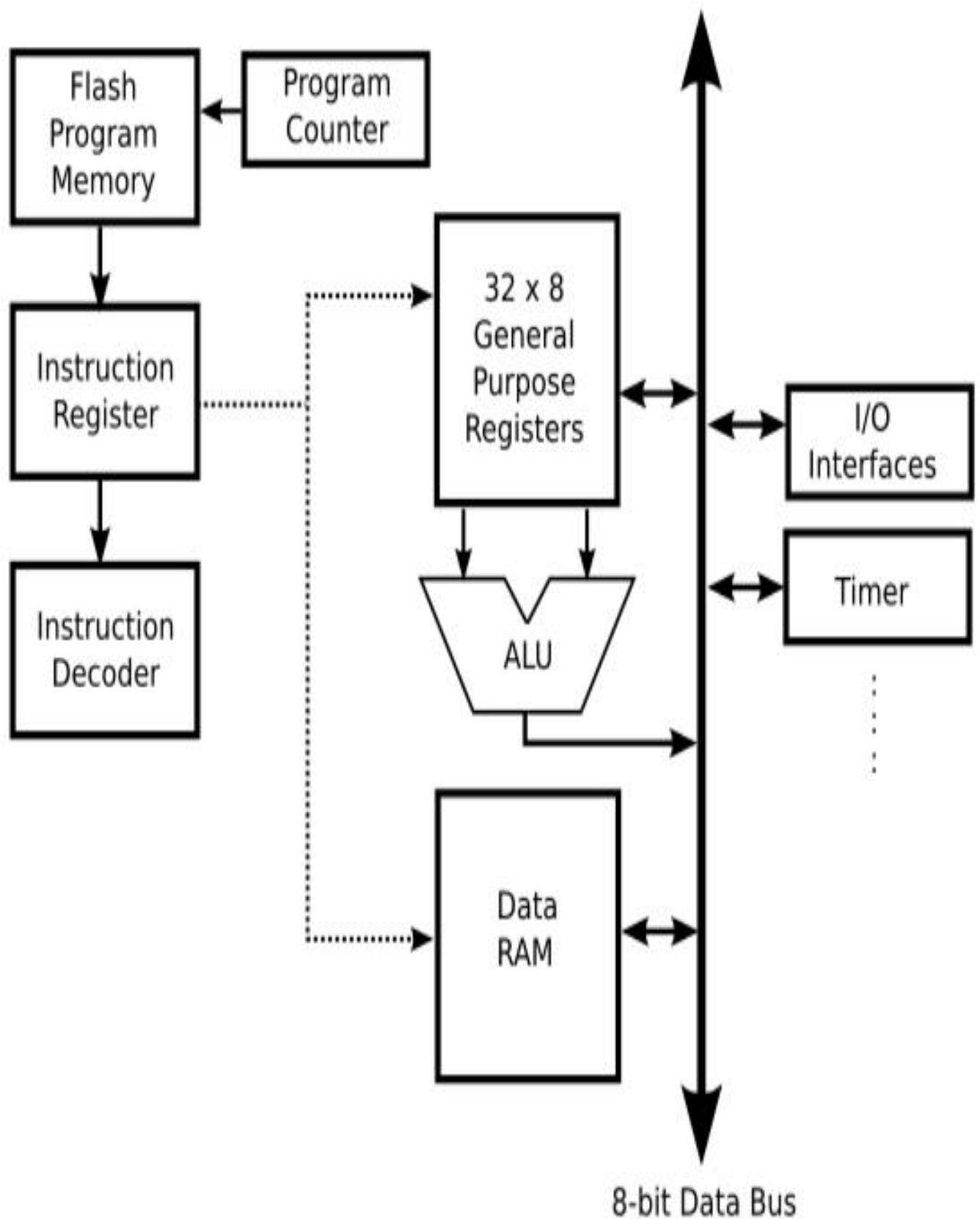


Fig 10: Basic Architecture of Microcontroller

1.2 EMBEDDED SYSTEM:

It can also be defined as a way of working, organizing or doing one or many tasks according to a fixed plan. For example, a watch is a time displaying system. Its components follow a set of rules to show time. If one of its parts fails, the watch will stop working. So we can say, in a system, all its subcomponents depend on each other. An embedded system is a system that has software embedded into computer-hardware, which makes a system dedicated for an application or specific part of an application or product or part of a larger system. An embedded system is one that has a dedicated purpose software embedded in a computer hardware. It is a dedicated computer based system for an application or product. It may be an independent system or a part of large system. Its software usually embeds into a ROM (Read Only Memory) or flash.

As its name suggests, Embedded means something that is attached to another thing. An embedded system can be thought of as a computer hardware system having software embedded in it. An embedded system can be an independent system or it can be a part of a large system. An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke. An embedded system has three components such as hardware, application software, Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small scale embedded system may not have RTOS. So we can define an embedded system as a Microcontroller based, software driven, reliable, real-time control system.

In an embedded system the following components such as sensor, analog to digital converter(ADC), processor and ASICs, digital to analog converter, actuator. The sensor is used to measure the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory. Analog-to-digital converter converts the analog signal sent by the sensor into a digital signal. Processors process the data to measure the output and store it to the memory. A digital-to-analog converter converts the digital data fed by the processor to analog data. The actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

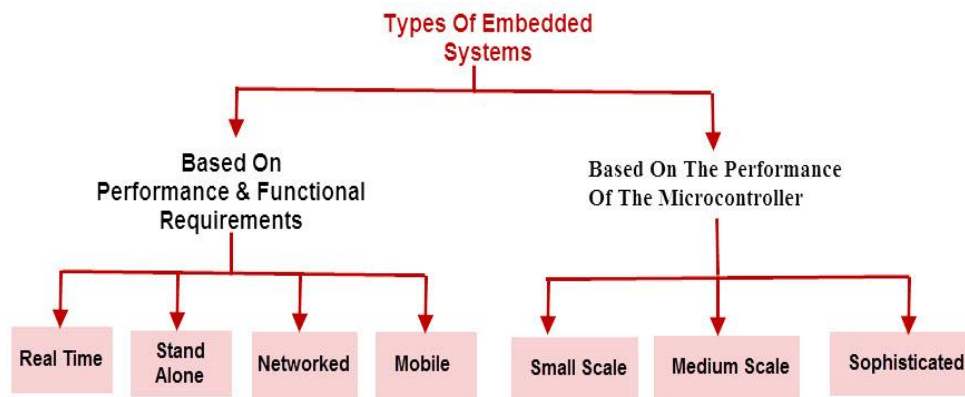
1.2.1 CHARACTERISTICS OF EMBEDDED SYSTEM

- Single-functioned– An embedded system usually performs a specialized operation and does the same repeatedly. For example: A pager always functions as a pager.
- Tightly constrained– All computing systems have constraints on design metrics, but those on an embedded system can be especially tight. Design metrics is a measure of an implementation's features such as its cost, size, power, and performance. It must be of a size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.
- Reactive and Real time– Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay. Consider an example of a car cruise controller; it continually monitors and reacts to speed and brake sensors. It must compute acceleration or de-accelerations repeatedly within a limited time; a delayed computation can result in failure to control of the car.
- Microprocessors based– It must be microprocessor or microcontroller based.
- Memory– It must have a memory, as its software usually embeds in ROM. It does not need any secondary memories in the computer.
- Connected– It must have connected peripherals to connect input and output devices.
- HW-SW systems – Software is used for more features and flexibility. Hardware is used for performance and security.
- Frequently connected to physical environment through sensors and actuators,
- Hybrid systems(analog + digital parts).
- A real-time system must react to stimuli from the controlled object (or the operator) within the time interval dictated by the environment.
- For real-time systems, right answers arriving too late (or even too early) are wrong.

- All other time-constraints are called soft.
- A guaranteed system response has to be explained without statistical arguments
- Reliability, maintainability, safety and security.
- Dedicated functions.
- Dedicated complex algorithms.
- Dedicated and other user interfaces for the application

1.2.2 CLASSIFICATION OF EMBEDDED SYSTEMS

Embedded systems can be classified into different types based on performance, functional requirements and performance of the microcontroller.



Embedded systems are classified into four categories based on their performance and functional requirements:

- **Stand alone embedded systems-** Does not require a host system like a computer, it works by itself. It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives and displays the connected devices. Examples for the stand alone embedded systems are mp3 players, digital cameras, video game consoles, microwave ovens and temperature measurement systems.
- **Real time embedded systems-** A real time embedded system is defined as, a system which gives a required o/p in a particular time. These types of embedded systems follow the time deadlines for completion of a task. Real time embedded systems are classified into two types such as soft and hard real time systems.

- **Networked embedded systems-** These types of embedded systems are related to a network to access the resources. The connected network can be LAN, WAN or the internet. The connection can be any wired or wireless. This type of embedded system is the fastest growing area in embedded system applications. The embedded web server is a type of system wherein all embedded devices are connected to a web server and accessed and controlled by a web browser. Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP.
 - **Mobile embedded systems-** Mobile embedded systems are used in portable embedded devices like cell phones, mobiles, digital cameras, mp3 players and personal digital assistants, etc. The basic limitation of these devices is the other resources and limitation of memory.
- Embedded Systems are classified into three types based on the performance of the microcontroller such as
- **Small scale embedded systems-** These types of embedded systems are designed with a single 8 or 16-bit microcontroller, that may even be activated by a battery. For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).
 - **Medium scale embedded systems-** These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs. These types of embedded systems have both hardware and software complexities. For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.
 - **Sophisticated embedded systems-** These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors. They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

1.3 OBJECTIVE:

1. To design a Braille pattern using vibration motors
2. Convert text message into Braille pattern
3. To design a Braille patterned keypad
4. System will reach the person through wireless
5. Deaf-blind person can able to communicate with each other

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY:

Louis Braille (4 January 1809 – 6 January 1852) was a French educator and an inventor of a system of reading and writing for use by the visually impaired, his system remains known worldwide simply as Braille. Braille consists of a system of six or eight possible dot combinations that are arranged in a fixed matrix, called a cell. Every dot can be set or cleared, giving 61 combinations in six-dot & 256 combinations in eight-dot Braille. All dots of a Braille page should fall on the intersections of an orthogonal grid. When texts are printed double-side (recto-verso), the grid of the verso text is shifted so that its dots fall in between the recto dots. Braille has a low information density. An average page of 25x 29 cm can have 32 characters on a line & 27 lines in a page. A typical dot has a diameter of 1.8 mm. This project presents a solution to such a problem, making learning process for an unsighted person easy. As all textbooks will not be available in Braille script as well as Audio recordings of all textbooks are not available. We will take the image of content in the textbook and it will be reproduced as Braille script for persons who are blind.

2.2 RELATED PAPERS & DESCRIPTIONS:

Researchers have proposed several models to ease the communication between an abled person and a deaf-blind person. F. Ramirez-Garibay, et al., (2014) proposed a model which uses USB keyboard to enter the message which is in turn processed by a Raspberry Pi controller. A 16 cell refreshable Braille display is used to display the text given by the user in the form of Braille language, so that the disabled person can sense the message. A speaker is also attached with this system, so that the person who is only blind can know the message by hearing it. A vibration motor is used as an alert system for the deaf blind people to indicate that a message has been received.

R.Shylaja, et al., (2018) in their model, has used a bluetooth module (HC-05) which is interfaced with an Arduino Mega microcontroller. Whenever a message is sent via bluetooth, present in the mobile phone to the communication device, the processor will convert the

message to braille and will send the braille pattern to the vibration motor which is arranged in braille cell format. Thus the corresponding motors will be raised up according to the received text. A 16x2 LCD display is also used to display the message and a piezo buzzer acts as an indicator whenever a message is received.

Sibila.R, et al., (2018) designed a software system for the conversion of English words, numbers and special characters into braille format. The inputs which are opened in a folder is read and it's corresponding braille format representation is assigned by MATLAB simulator by comparing the input characters with a look up table which contains the braille pattern for all 26 English alphabets, 10 digits and some special characters. The device stores the output text as a file, which can be printed as a Braille dots.

Y. Saraswathi, et al., (2017) in their paper designed a device which contains a GSM module which is interfaced with an Arduino UNO microcontroller. When the message is received in the GSM module from the mobile phone corresponding braille characters will be assigned by the controller and the vibration motors which is arranged in 3x2 matrix will raised up accordingly.

P. G. Anuradha, et al., (2016) proposed an efficient communication device for deaf-blind people which contains a QWERTY keyboard to enter the text. The keyboard is interfaced with a raspberry pi processor. Six servo motors are used which acts as braille display unit and performs raising of the braille dots. In this manner the disabled person can sense the message by touching the dots.

R. Sarkar, et al., (2012) in his paper discusses about an approach of converting the text to the braille format through a computer application. Six stepper motors are arranged in braille cell format and it is connected to the computer system. An application program loaded in the computer, takes the English text as input and performs different language processing, parsing, interpreting and controlling operations to meet the desired output of the system.

The next approach by P. J. Swati Malik, et al., (2016) proposes a device of E-Braille reader where the WI-FI module (ESP8266) is used for receiving the e-books and to communicate with other android devices. The received material or text is sent to STM32f407 ARM cortex microcontroller for processing and conversion. Servo motors are used for displaying the Braille characters. An audio playback has also been used as output device to hear the text for the person who is visually impaired.

One other approach by Kaustubh Bawdekar, et al., (2016) discusses about a model that functions by collecting the raw text with the help of a camera. The camera sends the collected data to the microcontroller (Raspberry Pi), which in turn converts it into Braille text. This model uses a look up table for conversion of Standard English text to Braille text. Here the solenoids act as a Braille display from which a visually impaired person can sense the braille characters.

CHAPTER 3

PROBLEM DESCRIPTION

3.1 EXISING SYSTEM:

Existing system uses RF as medium to control the vehicle speed. That process has certain limitation including communication range. Zigbee and ESP8266 Networks may cover the maximum communication range at the same time communication speed is gradually decreased. That existing model based on HC05 BT Module which is limited in communication range as well as less security comparing with the proposed system. The proposed design consists of two parts namely Text to Braille converter unit and a Vibration Band. The Text to Braille converter unit converts the input alphanumeric text received through the GSM module to its corresponding Braille format and sends the output signals to the vibration motors to form the pattern. This pattern can be sensed by the Deaf-Blind person and can thus interpret the message received.

3.1.1 PROBLEM DEFINITION:

The existing system does not wait for user presence. The system indicates the incoming message and activates the braille pattern. There are more or less possibility that user can miss the pattern. The system does not implement the braille to text. So, user cannot send replay.

3.1.2 SOLUTION ADAPTED:

We cannot implement entire alphabet keypad to convert braille to text. But some predefined messages capable to replay the sender. Whenever incoming message received, the controller module will activate the vibration band. If user interrupt his presence by pressing any key from given keypad the vibrator stops and braille start to display. User need to press button a after every character he read. A simple matrix keypad resolves the all issues we faced in existing system.

3.2 PROPOSED SYSTEM:

In this project, a way to develop a low cost, portable communication device which converts English text and numbers to its corresponding Braille format is discussed. The proposed design consists of three parts namely Text to Braille converter unit, Braille to text converter and a Vibration Band. The Text to Braille converter unit converts the input alphanumeric text received through the GSM module to its corresponding Braille format and sends the output signals to the vibration motors to form the pattern. The Braille to text converter has limited key buttons with predefined commands. User can communicate with each other by pressing particular key which contain predefined messages. This pattern can be sensed by the Deaf-Blind person and can thus interpret the message received.

3.2.1 Advantages:

- Now user can send messages through braille to text converter
- User no need to hurry to read all characters received in GSM
- Text to braille converter generates received mobile number before message.
- Additional display will help deaf person to read received message

CHAPTER 4

METHODOLOGY

4.1. SYSTEM REQUIREMENTS

The hardware and software requirements for the proposed system are presented as follows

4.1.1. HARDWARE REQUIREMENTS

- Microcontroller
- Keypad
- Relay Driver
- SIM800L
- RF Modules
- Alarm
- LCD Display
- Power Supply

4.1.1.1.INTRODUCTION TO MICROCONTROLLER

Microcontrollers are destined to play an increasingly important role in revolutionizing various industries and influencing our day to day life more strongly than one can imagine. Since its emergence in the early 1980's the microcontroller has been recognized as a general purpose building block for intelligent digital systems. It is finding using diverse area, starting from simple children's toys to highly complex spacecraft. Because of its versatility and many advantages, the application domain has spread in all conceivable directions, making it ubiquitous. As a consequence, it has generate a great deal of interest and enthusiasm among students, teachers and practicing engineers, creating an acute education need for imparting the knowledge of microcontroller based system design and development. It identifies the vital features responsible for their tremendous impact, the acute educational need created by them and provides a glimpse of the major application area.

MICROCONTROLLER:

A microcontroller is a complete microprocessor system built on a single IC. Microcontrollers were developed to meet a need for microprocessors to be put into low cost products. Building a complete microprocessor system on a single chip substantially reduces the cost of building simple products, which use the microprocessor's power to implement their function, because the microprocessor is a natural way to implement many products. This means the idea of using a microprocessor for low cost products comes up often. But the typical 8-bit microprocessor based system, such as one using a Z80 and 8085 is expensive. Both 8085 and Z80 system need some additional circuits to make a microprocessor system. Each part carries costs of money. Even though a product design may requires only very simple system, the parts needed to make this system as a low cost product.

To solve this problem microprocessor system is implemented with a single chip microcontroller. This could be called microcomputer, as all the major parts are in the IC. Most frequently they are called microcontroller because they are used they are used to perform control functions.

The microcontroller contains full implementation of a standard MICROPROCESSOR, ROM, RAM, I/O, CLOCK, TIMERS, and also SERIAL PORTS. Microcontroller also called "system on a chip" or "single chip microprocessor system" or "computer on a chip".

A microcontroller is a Computer-On-A-Chip, or, if you prefer, a single-chip computer. Micro suggests that the device is small, and controller tells you that the device' might be used to control objects, processes, or events. Another term to describe a microcontroller is embedded controller, because the microcontroller and its support circuits are often built into, or embedded in, the devices they control.

Today microcontrollers are very commonly used in wide variety of intelligent products. For example most personal computers keyboards and implemented with a microcontroller. It replaces Scanning, Debounce, Matrix Decoding, and Serial transmission circuits. Many low cost products, such as Toys, Electric Drills, Microwave Ovens, VCR and a host of other consumer and industrial products are based on microcontrollers.

EVOLUTION OF MICROCONTROLLER

Markets for microcontrollers can run into millions of units per application. At these volumes of the microcontrollers is a commodity items and must be optimized so that cost is at a minimum. Semiconductor manufacturers have produced a mind-numbing array of designs that would seem to meet almost any need. Some of the chips listed in this section are no longer regular production, most are current, and a few are best termed as "smoke ware": the dreams of an aggressive marketing department.

Sl.	Manufacturer	Chip Designation	Year	No. of Pins	No of I/O	RAM	ROM	Other Features
	4 Bit MC							
1.	Texas Instruments	TMS 1000	Mid 1970	28	23	64	1K	LED Display
2.	Hitachi	HMCS 40	-	28	10	32	512	10 bit ROM
3.	Toshiba	TLCS 47	-	42	35	128	2K	Serial bit I/O
	8 bit MC							
1.	Intel	8048	1976	40	27	64	1K	External Memory 8K
2	Intel	8051	1980	40	32	128	4K	External Memory 128 K
3.	Motorola	6081	1977	-	31	128	2 K	
4.	Motorola	68HC11	1985	52	40	256	8K	Serial Port,

								ADC,
5.	Zilog	Z8	-	40	32	128	2K	External Memory 128K,
	16 Bit MC							
1.	Intel	80C196	-	68	40	232	8K	External Memory 64K, Serial Port, ADC, WDT, PWM
2.	Hitachi	H8/532	-	84	65	1K	32K	External Memory 1M, Serial Port, ADC, PWM
3.	National	HPC16164	-	68	52	512	16K	External Memory 64K, ADC, WDT, PWM
	32 Bit MC							
1.	Intel	80960	-	132	20 MHz clock, 32 bit bus, 512 byte instruction cache			

APPLICATION

Microcontrollers did you use today?

A microcontroller is a kind of miniature computer that you can find in all kinds of Gizmos. Some examples of common, every-day products that have microcontrollers are built-in. If it has buttons and a digital display, chances are it also has a programmable microcontroller brain.

Every-Day the devices used by ourselves that contain Microcontrollers. Try to make a list and counting how many devices and the events with microcontrollers you use in a typical day. Here are some examples: if your clock radio goes off, and you hit the snooze button a few times in the morning, the first thing you do in your day is interact with a microcontroller. Heating up some food in the microwave oven and making a call on a cell phone also involve operating microcontrollers. That's just the beginning. Here are a few more examples: Turning on the Television with a handheld remote, playing a hand held game, Using a calculator, and Checking your digital wrist watch. All those devices have microcontrollers inside them, that interact with you. Consumer appliances aren't the only things that contain microcontrollers. Robots, machinery, aerospace designs and other high-tech devices are also built with microcontrollers.

BLOCK DIAGRAM OF MICROCONTROLLER

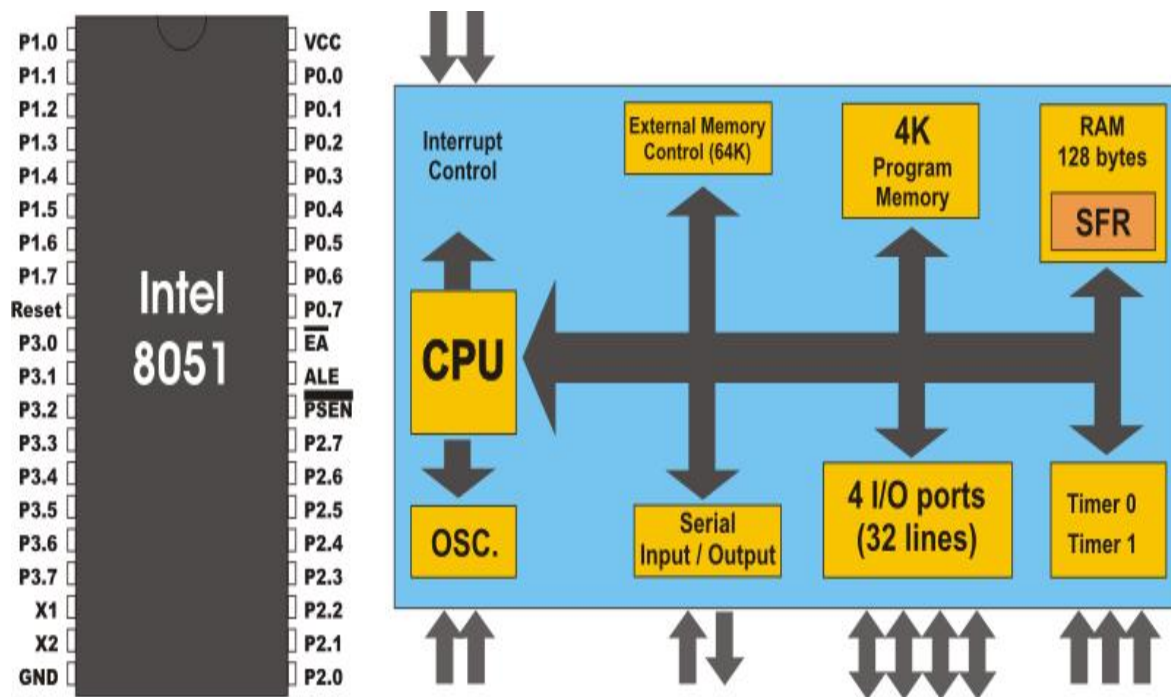


Fig 9: Basic block of Microcontroller

Arduino – uno features:

- Computer on a single integrated chip
 - ❖ Processer (CPU)
 - ❖ Memory (RAM/ROM/FLASH)
 - ❖ I/O Ports (USB,I²C,SPI,ADC,USART)
- Common microcontroller families
 - ❖ Intel: 4004,8008 etc.,
 - ❖ Atmel: AT and AVR
 - ❖ Microchip: PIC
 - ❖ ARM: (Multiple Manufactures)
- Controllers used in
 - ❖ Cell phones
 - ❖ Toys
 - ❖ Household applications
 - ❖ Cars and
 - ❖ Cameras

ATMEGA 328P (Arduino-uno)

- AVR 8-bit RISC architecture
- Available in DIP Package
- Up to 20Mhz clock source
- 32KB Flash memory
- 1KB Static ram
- 23 Programmable I/O Channels
- Six 10Bit ADC Inputs
- Three timers/counters
- Six PWM outputs

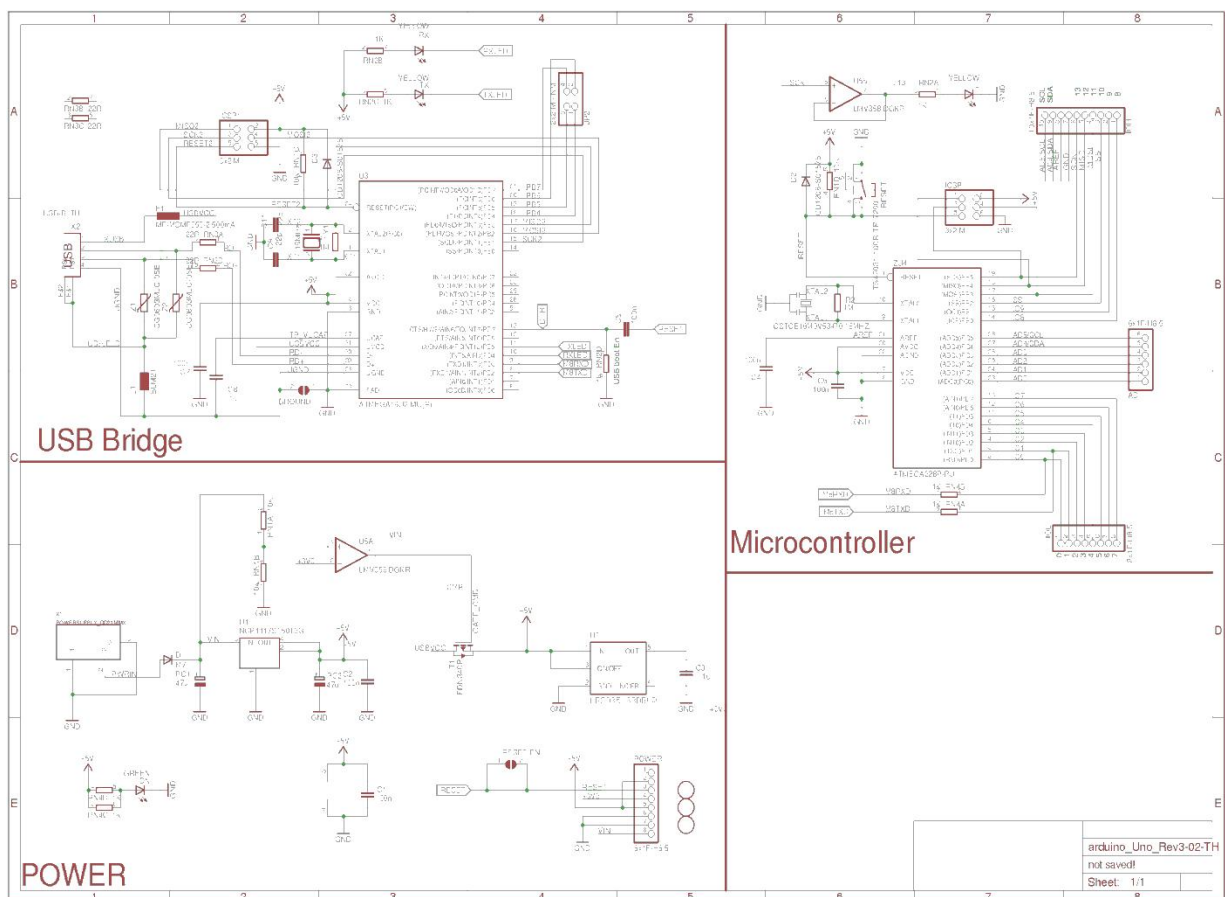


Fig 4.1: Overall circuit for arduino uno development board

ARDUINO UNO:

The ATmega328 microcontroller is the MCU used in Arduino UNO R3 as a main controller. ATmega328 is an MCU from the AVR family; it is an 8-bit device, which means that its data-bus architecture and internal registers are designed to handle 8 parallel data signals.

ATmega328 has three types of memory:

- **Flash memory:** 32KB nonvolatile memory. This is used for storing application, which explains why you don't need to upload your application every time you unplug arduino from its power source.
- **SRAM memory:** 2KB volatile memory. This is used for storing variables used by the application while it's running.
- **EEPROM memory:** 1KB nonvolatile memory. This can be used to store data that must be available even after the board is powered down and then powered up again.

4.5.1 POWER CONSUMPTION:

The MCU accepts supply voltages from 1.8 to 5.5 V. However, there are restrictions on the operating frequency; for example, if you want to use the maximum clock frequency (20 MHz), you need a supply voltage of at least 4.5 V.

4.1.2 DIGITAL I/O PINS:

This MCU has three ports: PORTC, PORTB, and PORTD. All pins of these ports can be used for general-purpose digital I/O or for the alternate functions indicated in the pinout below. For example, PORTC pin0 to pin5 can be ADC inputs instead of digital I/O. There are also some pins that can be configured as PWM output. These pins are marked with “~” on the Arduino board.

Atmega168 Pin Mapping

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

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

Fig 4.2: Atmega Pin mapping

Arduino Uno R3 Pinout

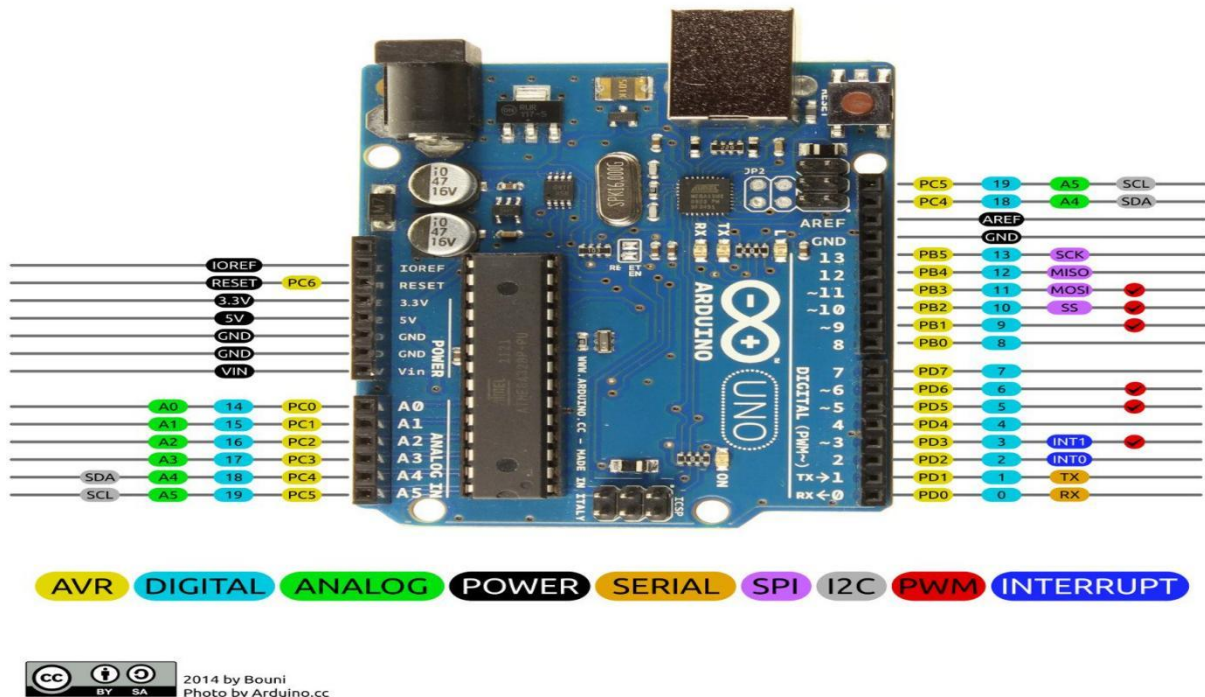


Fig 4.3: Arduino-uno pin out

4.1.3 ADC INPUTS:

This MCU has six channels—PORTC0 to PORTC5—with 10-bit resolution A/D converter. These pins are connected to the analog header on the Arduino board. One common mistake is to think of analog input as dedicated input for A/D function only, as the header in the board states "Analog". The reality is that you can use them as digital I/O or A/D.

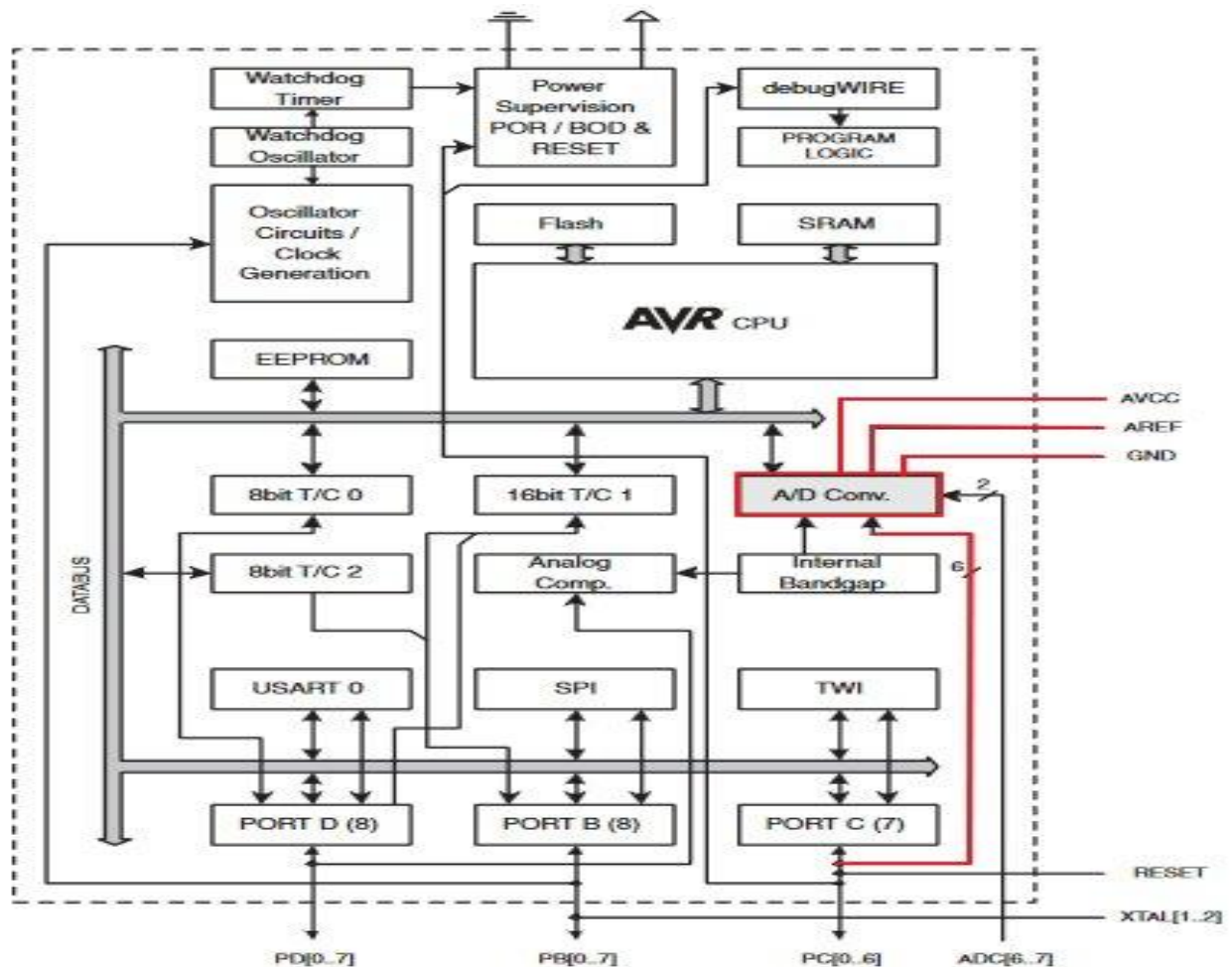


Fig 4.4: Atmega 328 functional block

As shown in the diagram above (via the red traces), the pins related to the A/D unit are:

- AVCC: The power pin for the A/D unit.
- AREF: The input pin used optionally if you want to use an external voltage reference for ADC rather than the internal Vref. You can configure that using an internal register.

Table 24-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

Fig 4.5: Internal Register select voltage reference for ADC**4.1.4 UART PERIPHERAL:**

A UART (Universal Asynchronous Receiver/Transmitter) is a serial interface. The ATmega328 has only one UART module. The pins (RX, TX) of the UART are connected to a USB-to-UART converter circuit and also connected to pin0 and pin1 in the digital header. You must avoid using the UART if you're already using it to send/receive data over USB.

4.1.5 SPI PERIPHERAL:

The SPI (Serial Peripheral Interface) is another serial interface. The ATmega328 has only one SPI module. Besides using it as a serial interface, it can also be used to program the MCU using a standalone programmer. You can reach the SPI's pins from the header next to the MCU in the Arduino UNO board or from the digital header as below:

11 - MOSI

12 - MISO

13 – SCK

4.1.6 TWI:

The I²C or Two Wire Interface is an interface consisting of only two wires, serial data, and a serial clock: SDA, SCL. User can reach these pins from the last two pins in the digital header or pin4 and pin5 in the analog header.

4.2 PCB DESIGN:

Arduino uno:

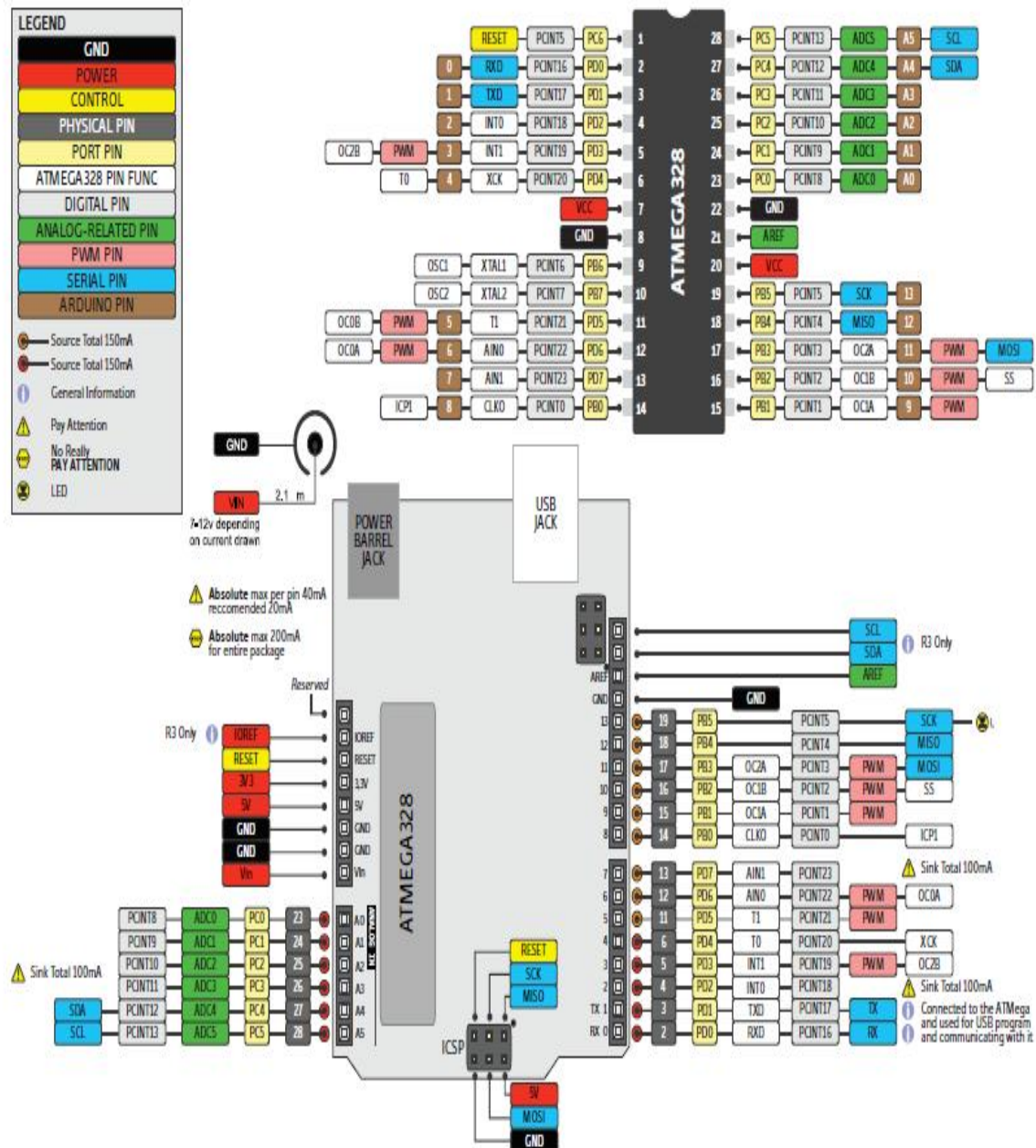


Fig x: Arduino design (PCB format not yet released)

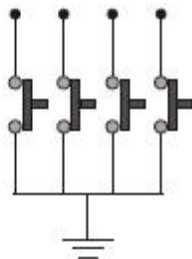
4.2.1 KEYPAD:

A numeric keypad, or numpad for short, is the small, palm-sized, seventeen key section of a computer keyboard, usually on the very far right. The numeric keypad features digits 0 to 9, addition (+), subtraction (-), multiplication (*) and division (/) symbols, a decimal point (.) and Num Lock and Enter keys. Laptop keyboards often do not have a numpad, but may provide numpad input by holding a modifier key (typically labelled "Fn") and operating keys on the standard keyboard.

Particularly large laptops (typically those with a 17 inch screen or larger) may have space for a real numpad, and many companies sell separate numpads which connect to the host laptop by a USB connection.

Numeric keypads usually operate in two modes: when Num Lock is off, keys 8, 6, 2, 4 act like an arrow keys and 7, 9, 3, 1 act like Home, PgUp, PgDn and End; when Num Lock is on, digits keys produce corresponding digits. These, however, differ from the numeric keys at the top of the keyboard in that, when combined with the Alt key on a PC, they are used to enter characters which may not be otherwise available: for example, Alt-0169 produces the copyright symbol. These are referred to as Alt codes.

On Apple Computer Macintosh computers, which lack a Num Lock key, the numeric keypad always produces only numbers. The num lock key is replaced by the clear key.

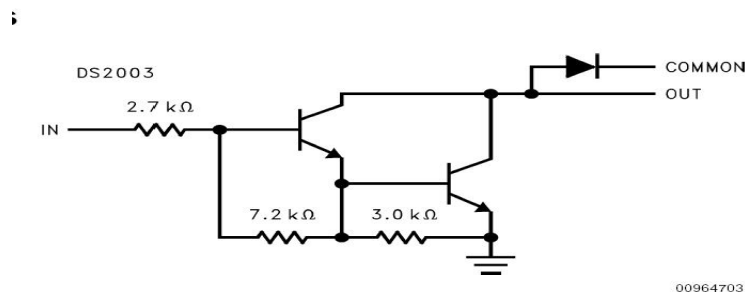


Numeric keypads usually operate in two modes: when Num Lock is off, keys 8, 6, 2, 4 act like an arrow keys and 7, 9, 3, 1 act like Home, PgUp, PgDn and End; when Num Lock is on, digits keys produce corresponding digits.

These, however, differ from the numeric keys at the top of the keyboard in that, when combined with the Alt key on a PC, they are used to enter characters which may not be otherwise available: for example, Alt-0169 produces the copyright symbol. These are referred to as Alt codes.

4.2.2 RELAY DRIVER:

RELAY DRIVER CIRCUIT



Absolute Maximum Ratings :

- Output voltage is 55V.
- Input voltage is -0.3V to 30V.
- Continuous Collector current 500mA.
- Continuous Base current 25mA
- Operating free air temperature range is 0°C to +70°C
- Storage temperature range is -65°C to +150°C.

Transistor as a switch :

When a transistor is used as driver circuit, it can withstand for high frequency signal. So a NPN transistor is used here as a driver circuit. When a transistor is used as a switch it is usually required to be brought alternatively in the saturation and in the cut-off conditions.

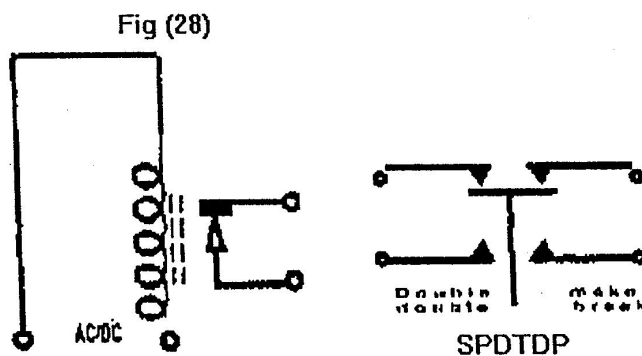
When in saturation condition, it should carry heavy current so that the voltage drop across it is as near to zero as possible and when in cutoff condition, it should carry almost no current so that it may be considered to be an open switch. It is found that the transistor does not respond instantaneously but takes a certain definite, though quite small, time in making a transition from one state to the other.

FUNCTIONS:

Relays are switching devices. Switching devices are the heart of industrial electronic systems. When a relay is energized or activated, contacts are made or broken. They are used to control ac or dc power. They are used to control the sequence of events in the operation of a system such as an electronic heater, counter, welding circuits, X-ray equipment, measuring systems, alarm systems and telephony. Electromagnetic relays are forms of electromagnets in which the coil current produces a magnetic effect. It pulls or pushes flat soft iron armatures or strips carrying relay contacts. Several relay contact can be operated to get several possible ON/OFF combinations.

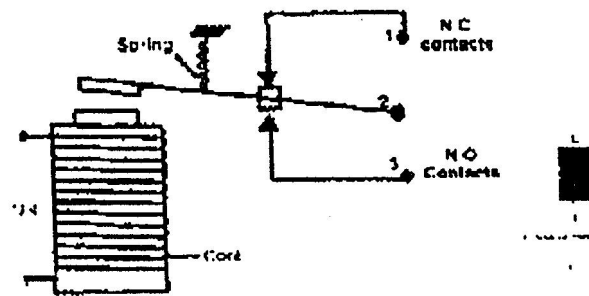
OPERATION OF ELECTROMAGNETIC RELAY :

Relays are usually dc operated. When dc is passed to the coil, the core gets magnetized. The iron armature towards the core contacts 1 and 2 open and contacts 2 and 3 close. When coil current is stopped, the attraction is not there and hence the spring tension brings 1 and 2 to closed position, opening the other set 2 and 3.



ELECTROMAGNETIC RELAY

Fig (29)



RELAY CONTACTS AND IDENTIFICATION :

The heart of the relay is the 'junction' of the contact points. The relay contact points may be flat, spherical, pointed and combination of all these. Flat contacts require more pressure for perfect contact closing. Half round contacts are better because the surface contamination will be minimum. The twin contacts give reliable operation.

Relay contacts are made of silver and silver alloys in small power applications. For large relays, contacts are made up of copper. Certain relays use silver – palladium or platinum – ruthenium alloys for contacts. The special types mentioned above give long life, carry moderate currents and keep shape for long time.

To identify relay contacts, some important contact arrangements must be remembered.

SPST	-	Single Pole Single Throw
SPDT	-	Single Pole Double Throw
NO	-	Normally Opened
NC	-	Normally Closed

Break	-	Relay action opens or breaks contacts
Make	-	Relay action makes or closes contacts

Relays are electromagnetic device by which operation of one or more circuits can be controlled by the operation of some other circuit. Relay is a type of switch where switching completely depends upon the electromagnetism. When winding of insulated wire is made on soft iron rod and apply is given across its end then magnetic field develops around the rod and due to this magnetic field, magnetism also becomes magnet. In this way, can be said that on giving supply to the coil winded over a core, it becomes magnet. This magnet is known as electromagnet.

Relay is a device which can turn ON/OFF any external circuit in some special circumstances. The principal relay is a one pole 2 way switch. The difference is that simple switch is manual switch where as relay is a automatic switch to some extent. It has a coil in it.

When this coil get enough supply then it becomes electromagnet and attracts the strip of pole towards itself and changes the position of switch. When supply cuts off then coil demagnetizes and thus switch comes in its normal position. In telephony, the relays are used widely. The relay that we used in this circuit has two states.

Normally closed state (NC)

Normally opened state (NO)

The control circuit of the relay transistor is shown in figure. When the input to transistor is logic 0, the transistor will be open. So the relay will be holding +12 and which will be in normally closed state.

4.2.3 SIM800L:

This chapter describes SIM800 hardware interface in great detail. This chapter can help user to quickly understand SIM800 interface specifications, electrical and mechanical details. With the help of this document and other SIM800 application notes, user guide, users can use SIM800 to design various applications quickly.

Overview:

Designed for global market, SIM800L is a quad-band GSM/GPRS module that works on frequencies GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM800 features GPRS multi-slot class 12/ class 10 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

With a tiny configuration of 24*24*3mm, SIM800 can meet almost all the space requirements in users' applications, such as M2M, smart phone, PDA and other mobile devices. SIM800 has 68 SMT pads, and provides all hardware interfaces between the module and customers' boards.

- Support up to 5*5*2 Keypads.
- One full function UART port, and can be configured to two independent serial ports.
- One USB port can be used as debugging and firmware upgrading.
- Audio channels which include a microphone input and a receiver output.
- Programmable general purpose input and output.
- One SIM card interface.
- Support Bluetooth function.
- Support one PWM.
- PCM

SIM800L is designed with power saving technique so that the current consumption is as low as 1.2mA in sleep mode.

SIM800 integrates TCP/IP protocol and extended TCP/IP AT commands which are very useful for data transfer applications.

Functional Diagram:

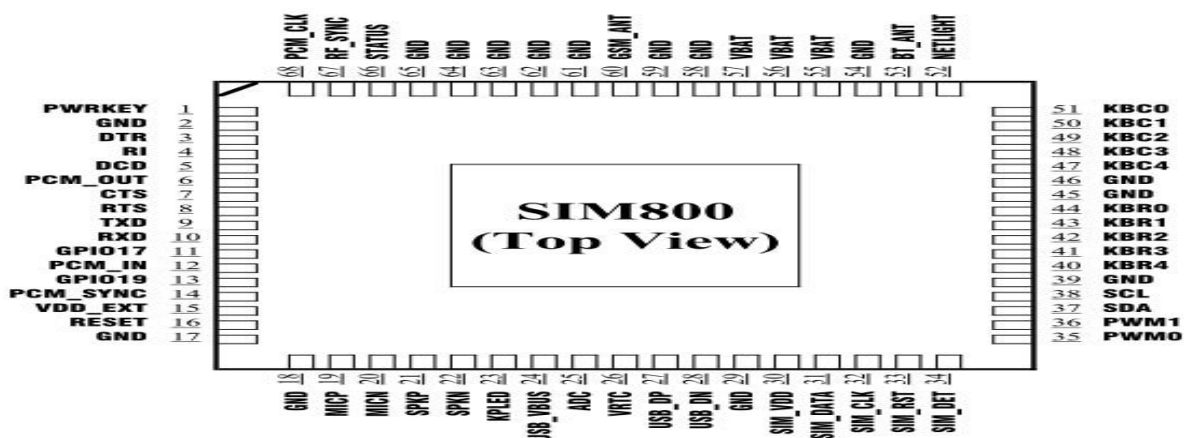
The following figure shows a functional diagram of sim800:

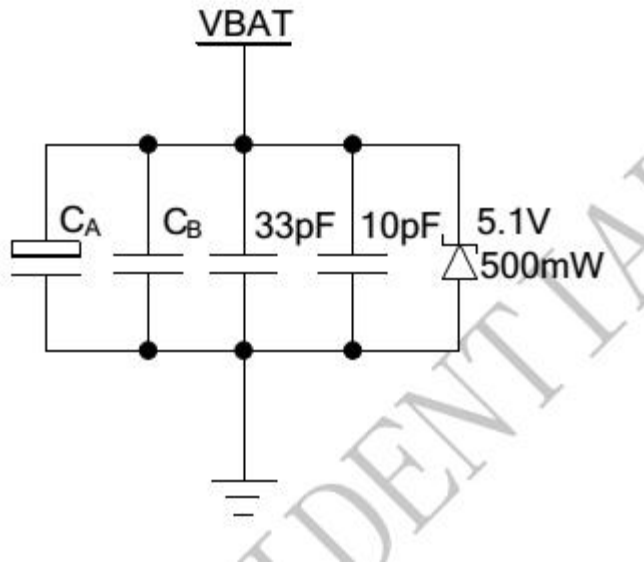
1. GSM baseband engine
2. PMU
3. RF part
4. Antenna interfaces
5. Other interfaces

Power supply:

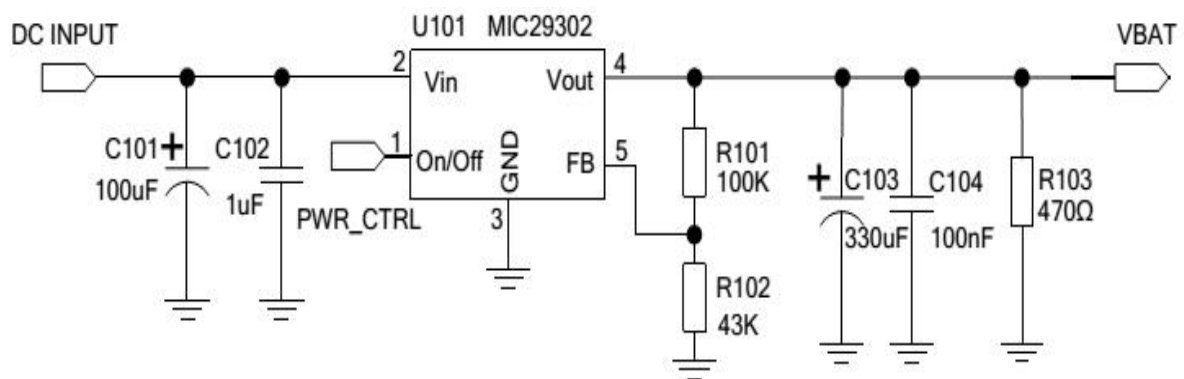
The power supply range of SIM800 is from 3.4V to 4.4V. Recommended voltage is 4.0V. The transmitting burst will cause voltage drop and the power supply must be able to provide sufficient current up to 2A. For the VBAT input, a bypass capacitor (low ESR) such as a 100μF is strongly recommended.

For the VBAT input, a 100uF Tantalum capacitor (CA low ESR) and a 1uF~10uF Ceramics capacitor CB are strongly recommended. The 33pF and 10pF capacitors can effectively eliminate the high frequency interference. A 5.1V/500mW Zener diode is strongly recommended, the diode can prevent chip from damaging by the voltage surge. These capacitors and Zener diode should be placed as close to SIM800 VBAT pins as possible.



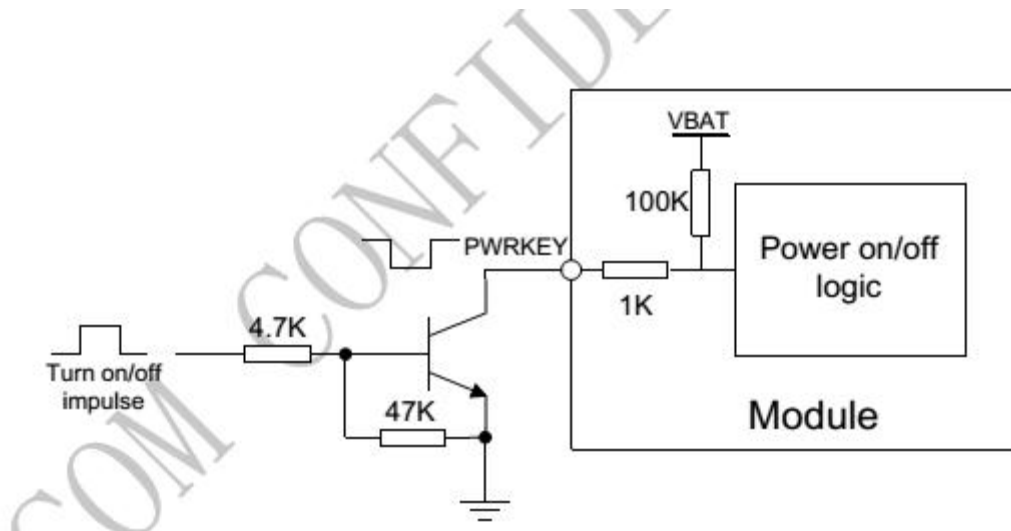


The following figure is the reference design of +5V input power supply. The designed output for the power supply is 4.1V, thus a linear regulator can be used.

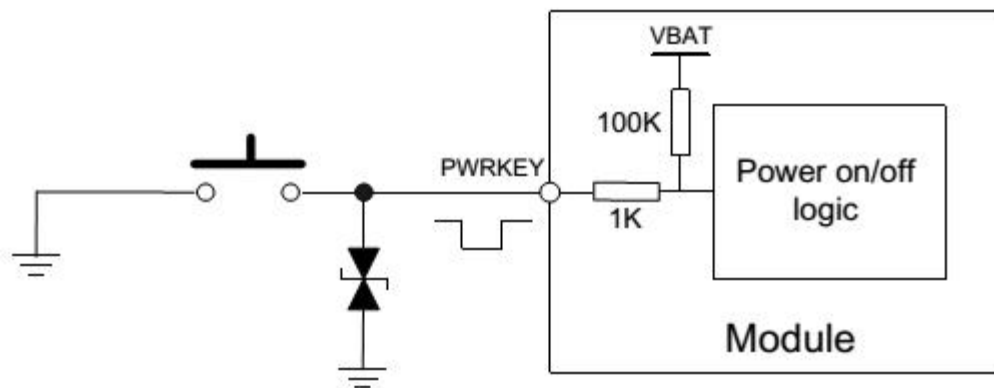


Power on/off module:

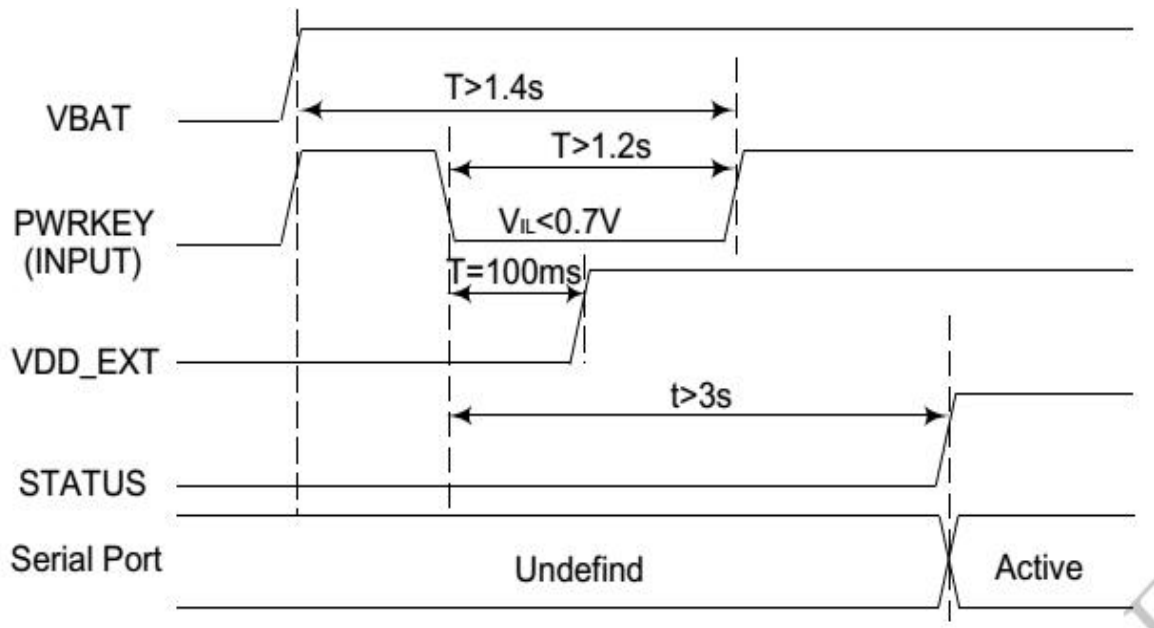
User can power on SIM800 by pulling down the PWRKEY pin at least 1.2 second and then release. This pin is already pulled up to VBAT in the module internal, so external pull up is not necessary. Reference circuit is shown as below



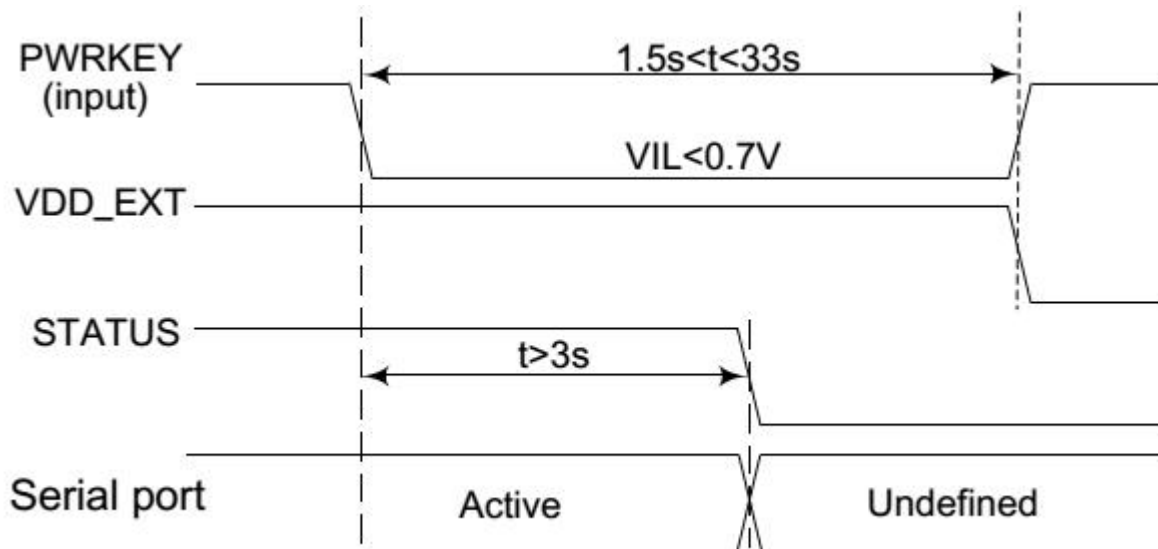
Power on/off module using transistor



Power on/off module using button



Power on timing diagram



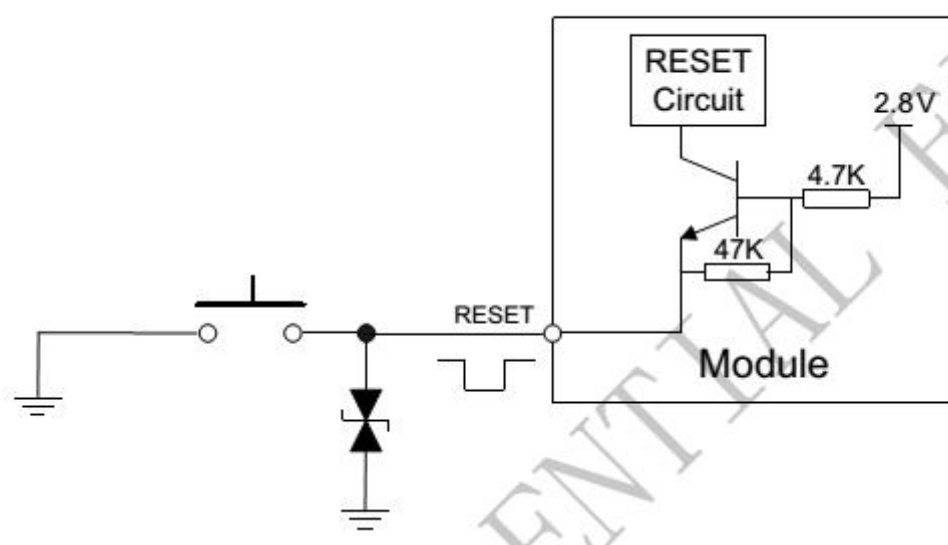
Power off timing diagram

SIM800 can be powered down by AT command "AT+CPOWD=1". This procedure makes the module log off from the network and allows the software to enter into a secure state to save data before completely shut down. Before the completion of the power off procedure, the module will send URC: ***NORMAL POWERDOWN***

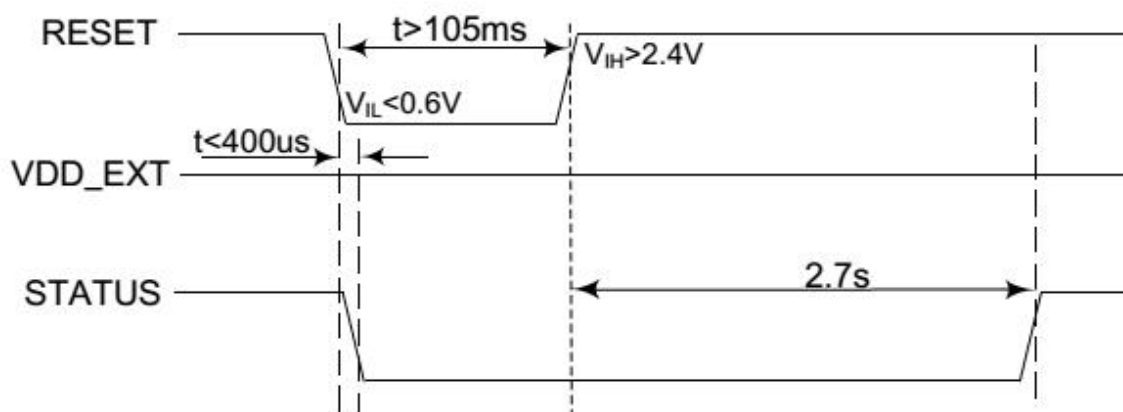
At this moment, AT commands cannot be executed any more, and only the RTC is still active. Power off mode can also be indicated by STATUS pin, which is at low level at this time.

Reset function:

SIM800L also have a RESET pin used to reset the module. This function is used as an emergency reset only when AT command “AT+CPOWD=1” and the PWRKEY pin have no effect. User can pull the RESET pin to ground, and then the module will restart. This pin is already isolated in the module, so the external isolation is not necessary. Following figure is internal circuit of the RESET pin.



Reset module using button



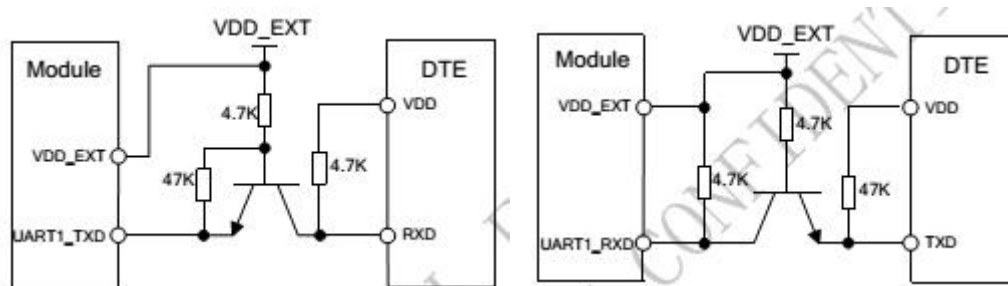
Reset timing diagram

TTL interface:

SIM800 provides one unbalanced asynchronous serial port. The module is designed as a DCE (Data Communication Equipment). The following figure shows the connection between module and client (DTE).

- Full modem device.
- Contains data lines TXD and RXD, hardware flow control lines RTS and CTS, status lines DTR, DCD and RI.
- Serial port can be used for CSDFAX, GPRS service and AT communication. It can also be used for multiplex function.
- Serial port supports the following baud rates:
 - 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps, 230400 and 460800 bps;
 - Auto bauding only supports the following baud rates:
 - 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bps
 - The default setting is auto bauding.

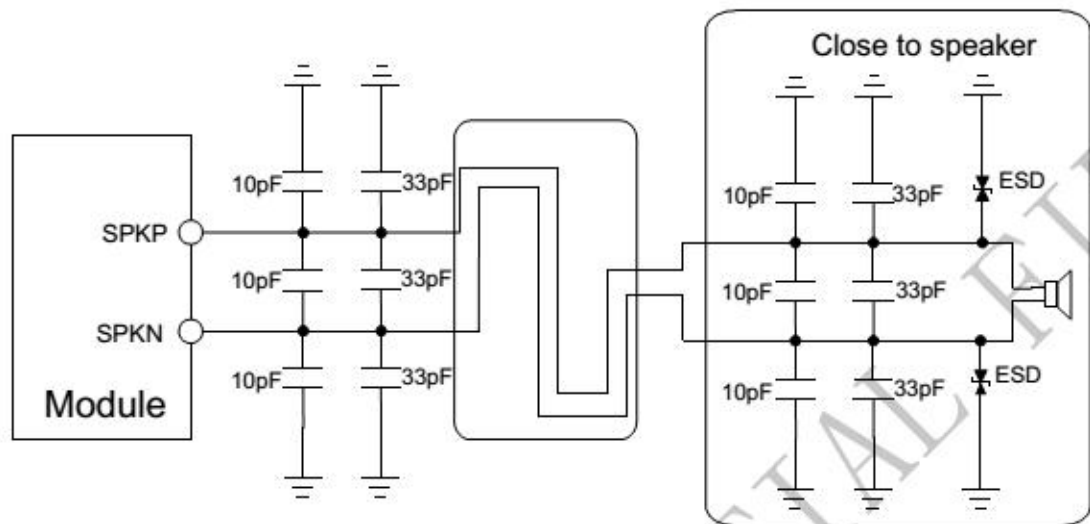
If the voltage of UART is 5V on customer side, customer can use the following reference circuits:



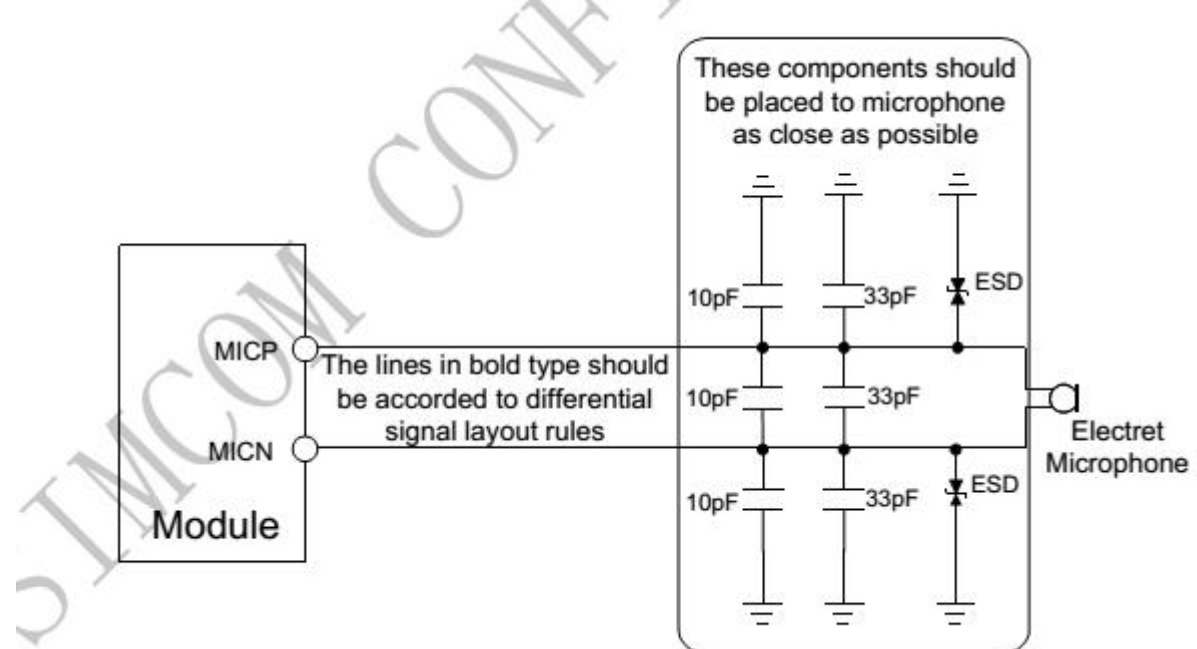
TX/RX level converting circuit

Audio interfacing:

SIM800 provides one analog input, which could be used for electret microphone. The module also provides one analog output. The output can directly drive 32 Ω receiver.



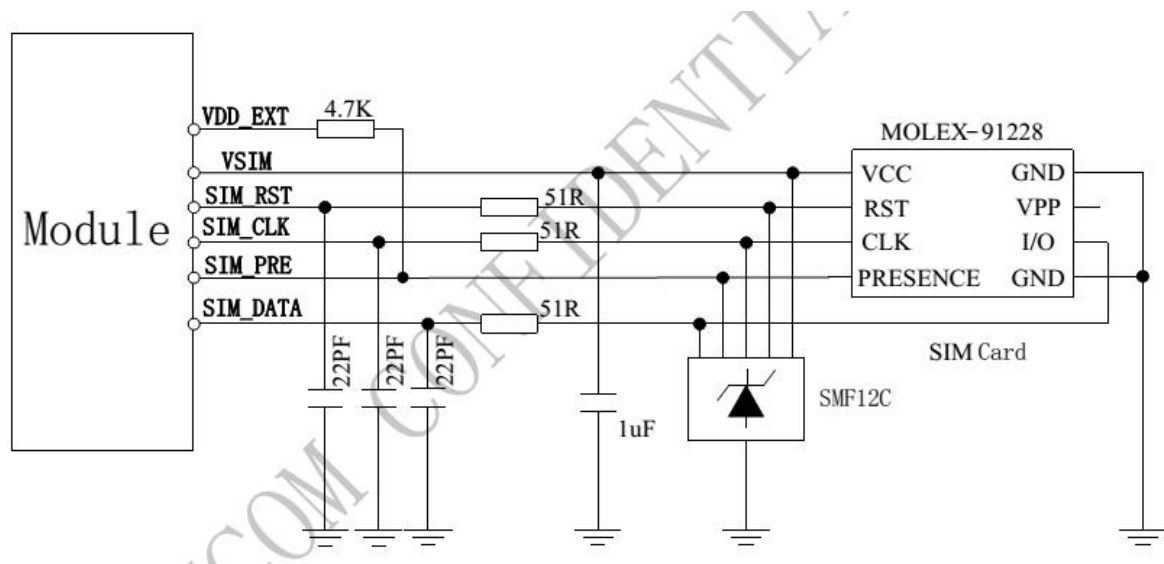
Speaker interfacing reference design



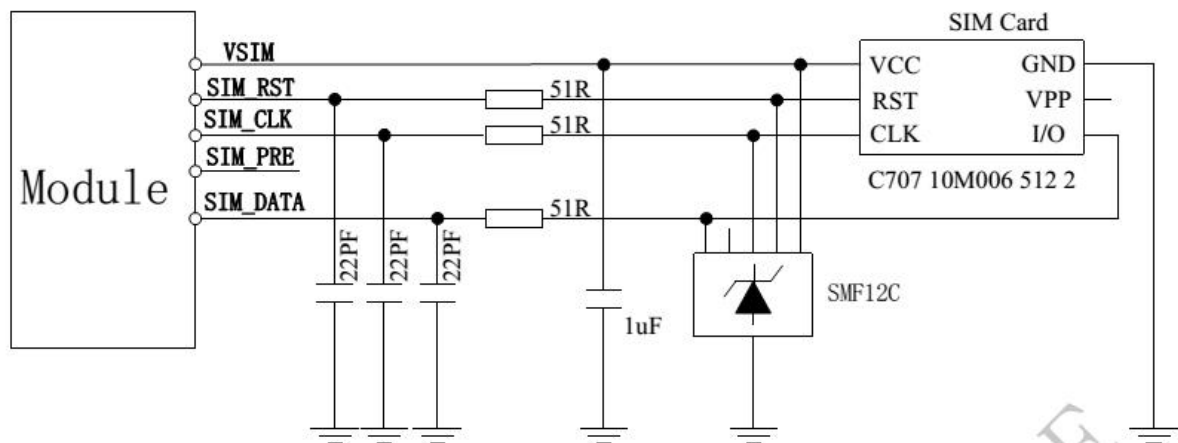
Microphone interfacing reference design

SIM card interfacing:

The SIM interface complies with the GSM Phase 1 specification and the new GSM Phase 2+ specification for FAST 64 kbps SIM card. Both 1.8V and 3.0V SIM card are supported. The SIM interface is powered from an internal regulator in the module.



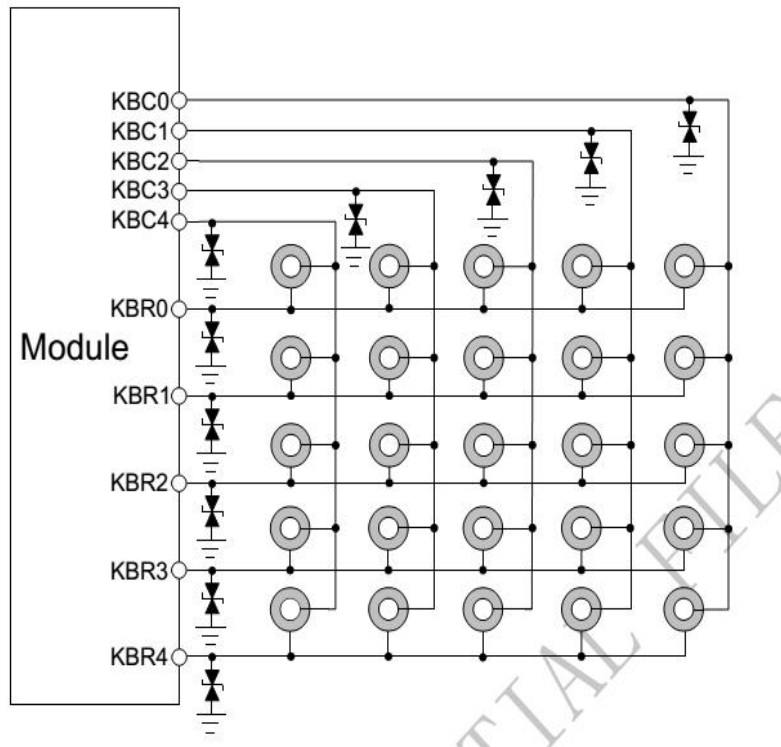
Reference circuit for 8-pin SIM card holder



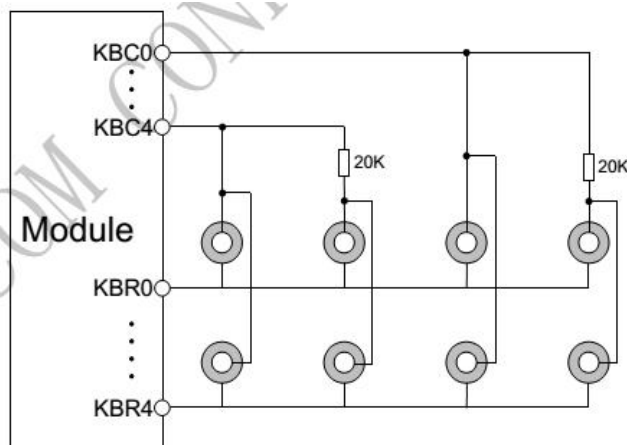
Reference circuit for 6-pin SIM card holder

Keypad interfacing:

SIM800 consists of 5 keypad column outputs and 5 keypad row inputs, and it can support two kinds of connections, the traditional 5*5 keypad matrix and the extended 5*5*2 keypad matrix.



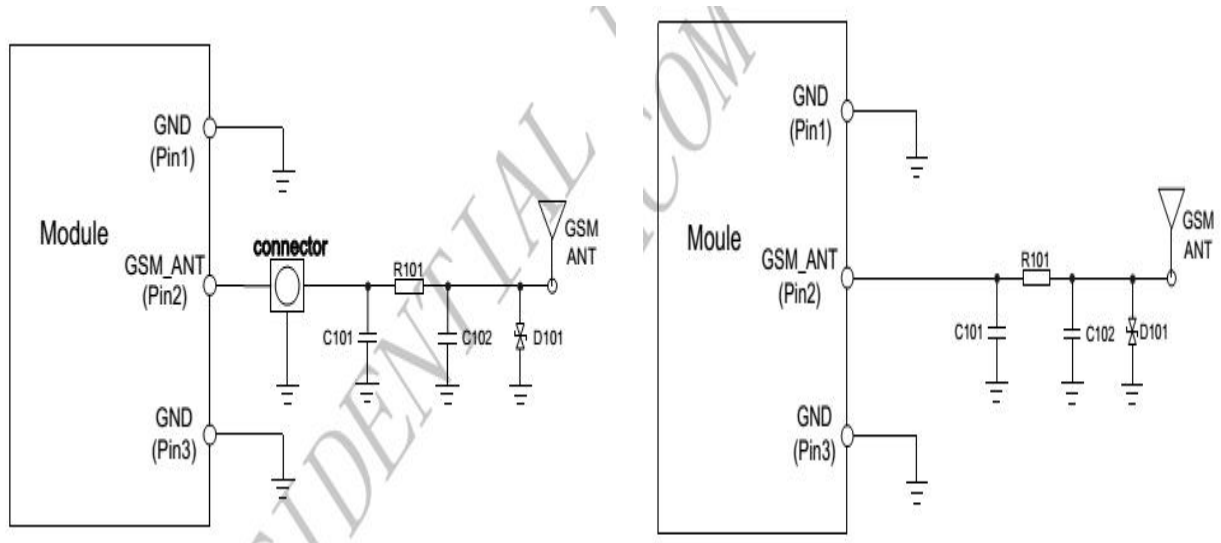
Traditional keypad interfacing



Additional keypad interfacing

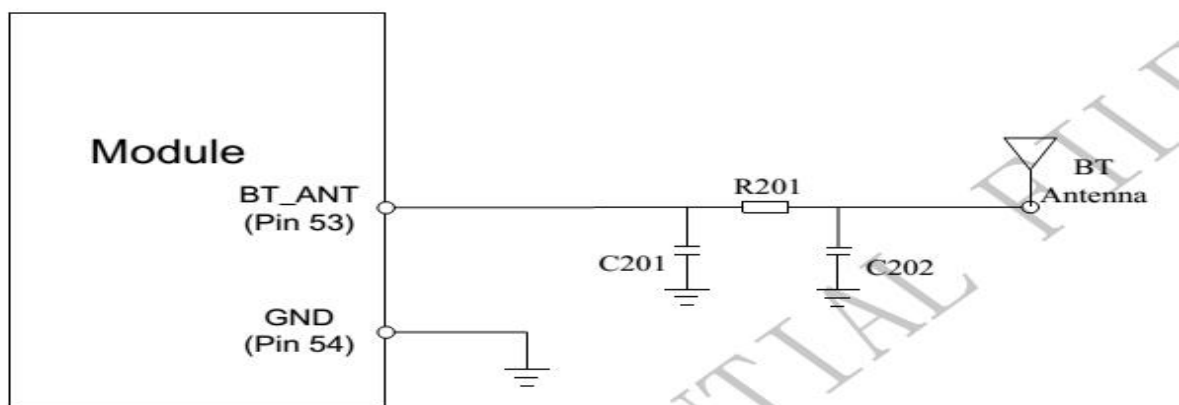
Antenna interfacing:

There is a GSM antenna pad named GSM_ANT to connect a GSM antenna, the connection of the antenna must be decoupled from DC voltage. This is necessary because the antenna connector is DC coupled to ground via an inductor for ESD protection. The GSM antenna must be matched properly to achieve the best performance, so the matching circuit is necessary. For the purpose of static electricity, we recommend to add D101, which is a TVS, the recommendation ESD component. The connection is recommended as following:

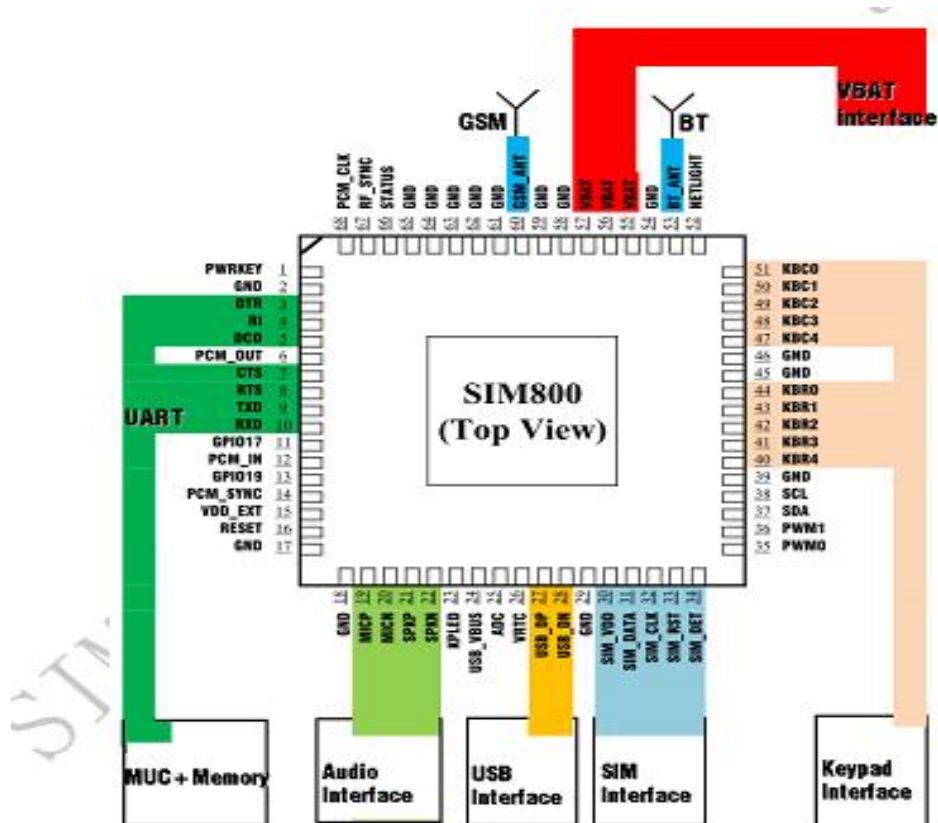


Antenna matching circuit

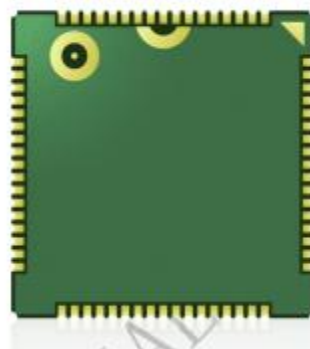
The module provides a Bluetooth antenna interface named BT_ANT to connect a Bluetooth antenna. The Bluetooth antenna must be matched properly to achieve best performance, so the matching circuit is necessary, the connection is recommended as the following figure:



PCB layout:



Recommendation layout



Top and bottom view of sim800x series

Note: This documentation contains some confidential details from SIM series. Documentation can be used for educational purpose only and user can recreate this documentation as per their usage no legal guarantee or warranty will provide regards this document.

4.2.4 RF MODULE:

General Description:

The 212 encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12_N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a TE trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 212 series of encoders. The HT12A additionally provides a 38 kHz carrier for infrared systems.

Features:

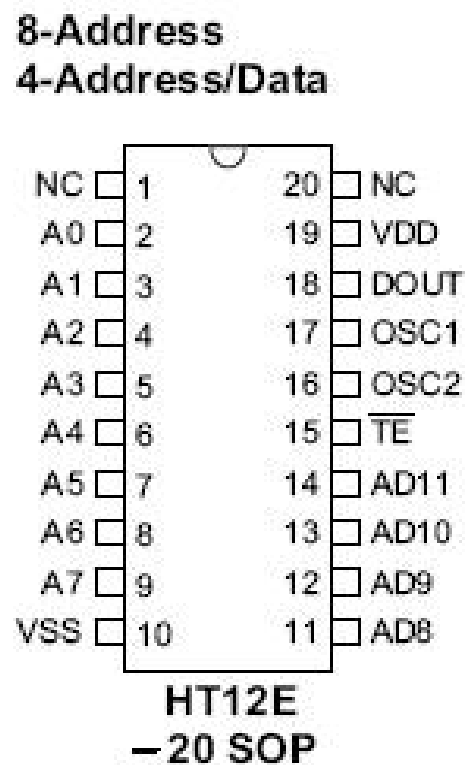
- _ Operating voltage
- _ 2.4V~5V for the HT12A
- _ 2.4V~12V for the HT12E
- _ Low power and high noise immunity CMOS technology
- _ Low standby current: 0.1_A (typ.) at VDD=5V
- _ HT12A with a 38kHz carrier for infrared transmission medium
- _ Minimum transmission word
- _ Four words for the HT12E
- _ One word for the HT12A
- _ Built-in oscillator needs only 5% resistor
- _ Data code has positive polarity
- _ Minimal external components

_ HT12A/E: 18-pin DIP/20-pin SOP package

Applications:

- _ Burglar alarm system
- _ Smoke and fire alarm system
- _ Garage door controllers
- _ Car door controllers
- _ Car alarm system
- _ Security system
- _ Cordless telephones
- _ Other remote control systems

Pin Configuration:

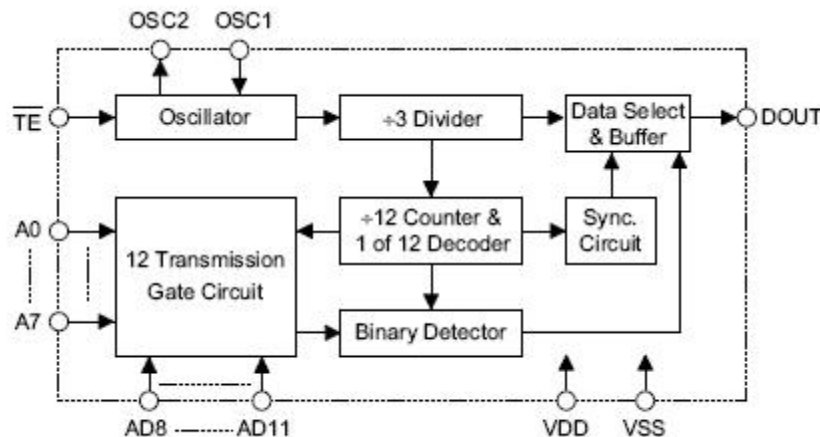


Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
		NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/MB	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS

Block Diagram

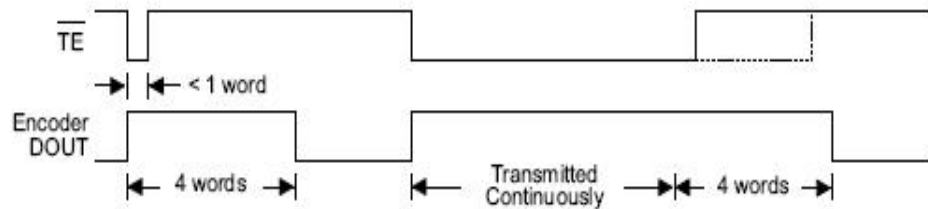
$\overline{\text{TE}}$ trigger
HT12E



Functional Description

Operation

The 2^{12} series of encoders begin a 4-word transmission cycle upon receipt of a transmission enable ($\overline{\text{TE}}$ for the HT12E or D8~D11 for the HT12A, active low). This cycle will repeat itself as long as the transmission enable ($\overline{\text{TE}}$ or D8~D11) is held low. Once the transmission enable returns high the encoder output completes its final cycle and then stops as shown below.



Transmission timing for the HT12E

Decoders:

General Description:

The 212 decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 212 series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen. The decoders receive serial addresses and data from a Programmed 212 series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. The 212 series of decoders are capable of decoding information's that consist of N bits of address and 12_N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

Features:

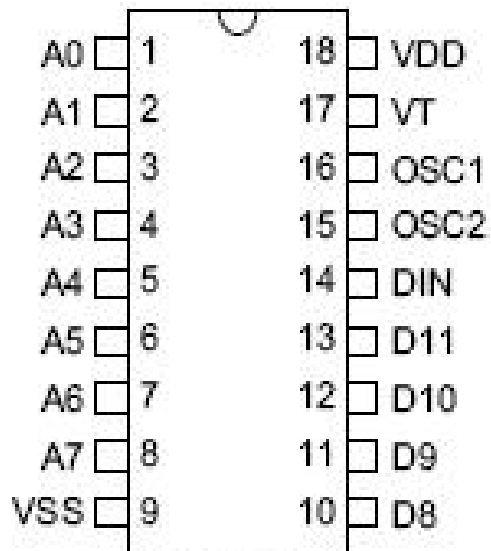
- _ Operating voltage: 2.4V~12V
- _ Low power and high noise immunity CMOS technology
- _ Low standby current
- _ Capable of decoding 12 bits of information
- _ Pair with Holtek's 212 series of encoders
- _ Binary address setting
- _ Received codes are checked 3 times
- _ Address/Data number combination
- _ HT12D: 8 address bits and 4 data bits
- _ HT12F: 12 address bits only
- _ Built-in oscillator needs only 5% resistor

- _ Valid transmission indicator
- _ Easy interface with an RF or an infrared transmission medium
- _ Minimal external components

Applications

- _ Burglar alarm system
- _ Smoke and fire alarm system
- _ Garage door controllers
- _ Car door controllers
- _ Car alarm system
- _ Security system
- _ Cordless telephones
- _ Other remote control systems

8-Address
4-Data

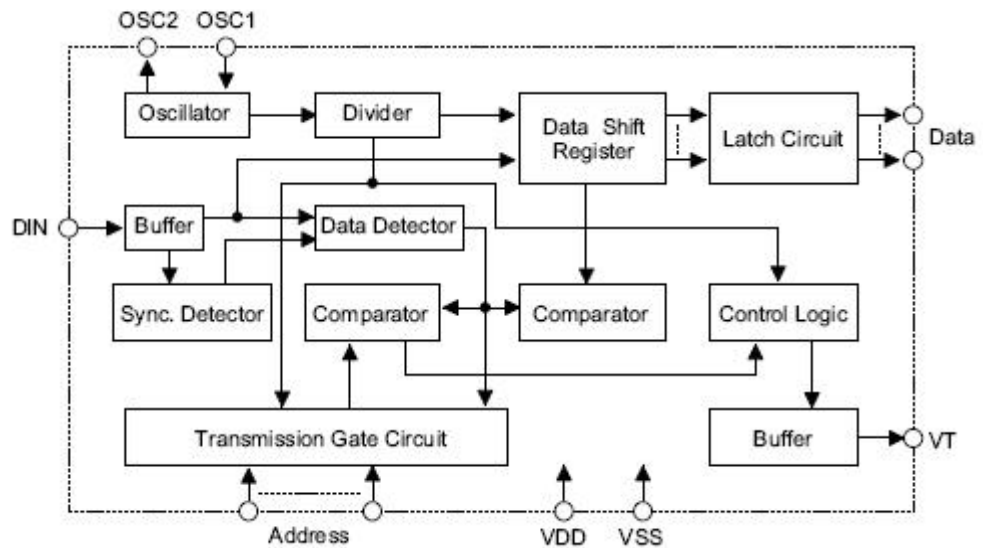


HT12D
-18 DIP

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A11	I	NMOS TRANSMISSION GATE	Input pins for address A0~A11 setting They can be externally set to VDD or VSS.
D8~D11	O	CMOS OUT	Output data pins
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	OSCILLATOR	Oscillator input pin
OSC2	O	OSCILLATOR	Oscillator output pin
VSS	I	—	Negative power supply (GND)
VDD	I	—	Positive power supply

Block Diagram



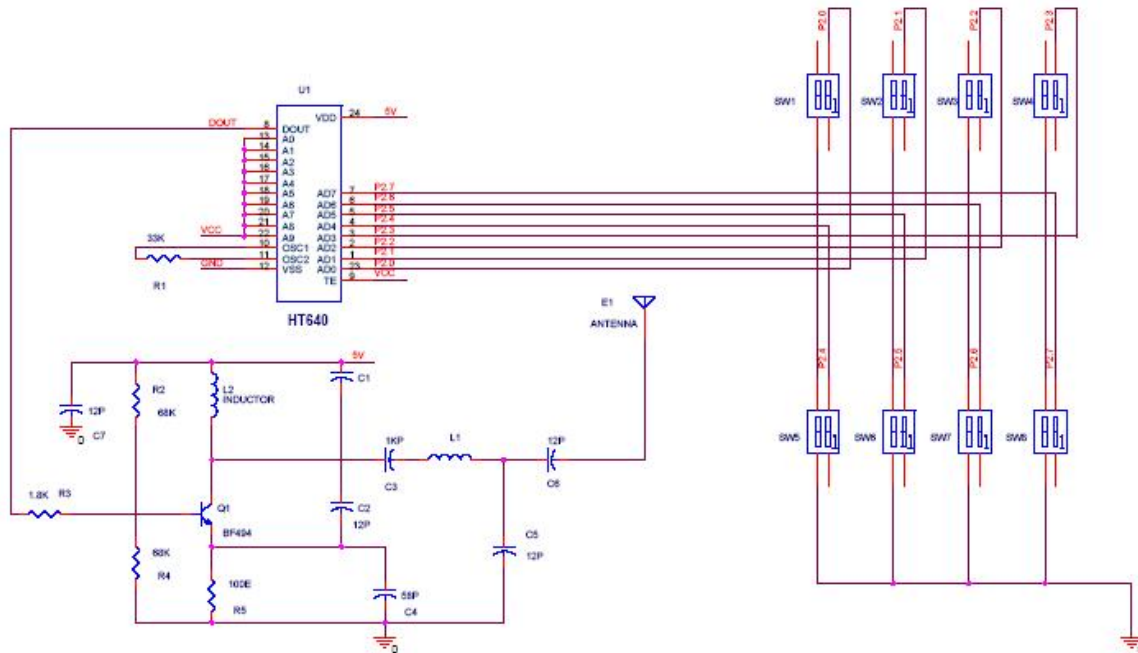
Functional Description:

Operation:

The 212 series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 212series of encoders. The decoders receive data that are transmitted

by an encoder and interpret the first N bits of code period as addresses and the last 12_N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoders local address, the 12_N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received. The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

ENCODER WITH KEYBOARD AND RF TRANSMITTER



Keypad:

The keypad consists of eight key in which each key represents the one operation. The key board is interfaced with encoder data lines. If any one key is pressed the corresponding signal is given to encoder.

Encoder:

In this circuit HT 640 is used as encoder. The 3^{18} encoders are a series of CMOS LSIs for remote control system application. They are capable of encoding 18 bits of information which consists of N address bit and 18-N data bits. Each address/data input is externally trinary programmable if bonded out. It is otherwise set floating internally. Various packages of the 3^{18} encoders offer flexible combination of programmable address/data is transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a

trigger signal. The capability to select a TE trigger type further enhances the application flexibility of the 3¹⁸ series of encoders.

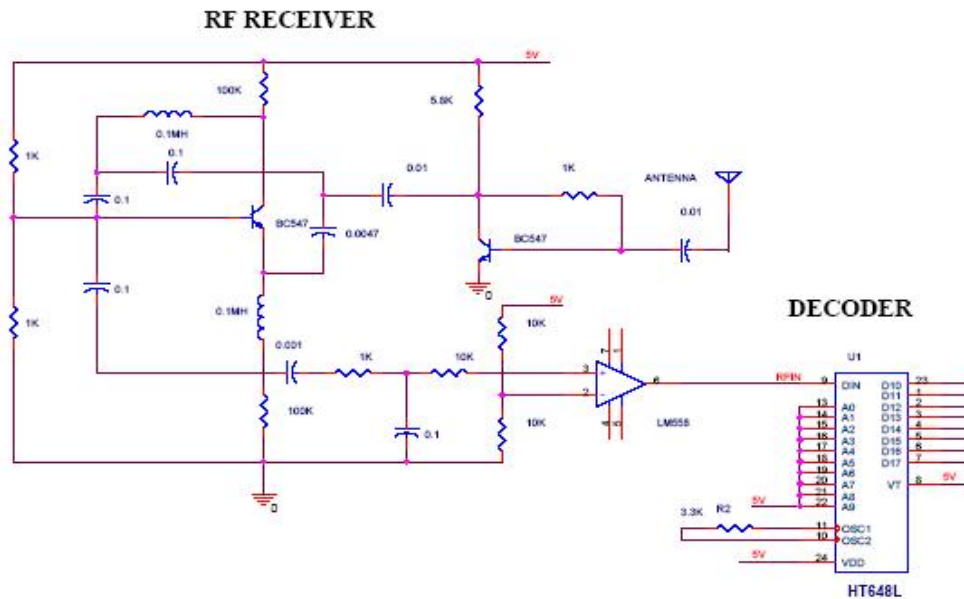
In this circuit the input signal to be encoded is given to AD7-AD0 input pins of encoder. The encoder output address pins are shorted so the output encoded signal is the combination of (A0-A9) address signal and (D0-D7) data signal. The output encoded signal is taken from 8th which is connected to RF transmitter section.

RF Transmitter:

Whenever the high output pulse is given to base of the transistor BF 494, the transistor is conducting so tank circuit is oscillated. The tank circuit consists of L2 and C4 generating 433 MHz carrier signal. Then the modulated signal is given LC filter section. After the filtration the RF modulated signal is transmitted through antenna.

RFReceiver:

DECODER WITH RF RECEIVER



RF Receiver:

The RF receiver is used to receive the encoded data which is transmitted by the RF transmitter. Then the received data is given to transistor which acts as amplifier. Then the amplified signal is given to carrier demodulator section in which transistor Q1 is turn on and turn off conducting depends on the signal. Due to this the capacitor C14 is charged and discharged so carrier signal is removed and saw tooth signal is appears across the capacitor. Then this saw tooth signal is given to comparator. The comparator circuit is constructed by LM558. The comparator is used to convert the saw tooth signal to exact square pulse. Then the encoded signal is given to decoder in order to get the decoded original signal.

Decoder:

In this circuit HT648 is used as decoder. The 3^{18} decoder are a series of CMOS LSIs for remote control system application. They are paired with 3^{18} series of encoders. For proper operation a pair of encoder/decoder pair with the same number of address and data format should be selected. The 3^{18} series of decoder receives serial address and data from that series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. It

then compares the serial input data twice continuously with its local address. If no errors or unmatched codes are encountered, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. The 3^{18} decoders are capable of decoding 18 bits of information that consists of N bits of address and 18-N bits of data. To meet various applications they are arranged to provide a number of data pins whose range is from 0 to 8 and an address pin whose range is from 8 to 18. In addition, the 3^{18} decoders provide various combinations of address/ data numbering different package.

In this circuit the received encoded signal is 9th pin of the decoder. Now the decoder separate the address (A0-A9) and data signal (D0-D7). Then the output data signal is given to microcontroller or any other interfacing device.

5.1.1.8 LCD Display



Fig 14: Liquid crystal display

Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal.

An LCD consists of two glass panels, with the liquid crystal material sandwiched between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed. Polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle.

On each polarizer is pasted outside the two glass panels. These polarizers would rotate the light rays passing through them to a definite angle, in a particular direction.

When the LCD is in the off state, light rays are rotated by the two polarizers and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent.

When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarizers, which would result in activating / highlighting the desired characters.

The LCD's are lightweight with only a few millimeters thickness. Since the LCD's consume less power, they are compatible with low power electronic circuits, and can be powered for long durations.

The LCD's do not generate light and so light is needed to read the display. By using backlighting, reading is possible in the dark. The LCD's have long life and a wide operating temperature range.

Changing the display size or the layout size is relatively simple which makes the LCD's more customer friendly.

The LCDs used exclusively in watches, calculators and measuring instruments are the simple seven-segment displays, having a limited amount of numeric data. The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range. These have resulted in the LCDs being extensively used in telecommunications and entertainment electronics. The LCDs have even started replacing the cathode ray tubes (CRTs) used for the display of text and graphics, and also in small TV applications.

4.3 POWER SUPPLY

The power supply should be of +5V, with maximum allowable transients of 10mv. To achieve a better / suitable contrast for the display, the voltage (VL) at pin 3 should be adjusted properly.

A module should not be inserted or removed from a live circuit. The ground terminal of the power supply must be isolated properly so that no voltage is induced in it. The module should be isolated from the other circuits, so that stray voltages are not induced, which could cause a flickering display.

HARDWARE

Develop a uniquely decoded 'E' strobe pulse, active high, to accompany each module transaction. Address or control lines can be assigned to drive the RS and R/W inputs.

Utilize the Host's extended timing mode, if available, when transacting with the module. Use instructions, which prolong the Read and Write or other appropriate data strobes, so as to realize the interface timing requirements.

If a parallel port is used to drive the RS, R/W and 'E' control lines, setting the 'E' bit simultaneously with RS and R/W would violate the module's set up time. A separate instruction should be used to achieve proper interfacing timing requirements.

MOUNTING

Cover the display surface with a transparent protective plate, to protect the polarizer.

Don't touch the display surface with bare hands or any hard materials. This will stain the display area and degrade the insulation between terminals.

Do not use organic solvents to clean the display panel as these may adversely affect tape or with absorbent cotton and petroleum benzene.

The processing or even a slight deformation of the claws of the metal frame will have effect on the connection of the output signal and cause an abnormal display.

Do not damage or modify the pattern wiring, or drill attachment holes in the PCB. When assembling the module into equipment, the space between the module and the fitting plate should have enough height, to avoid causing stress to the module surface.

Make sure that there is enough space behind the module, to dissipate the heat generated by the ICs while functioning for longer durations.

When an electrically powered screwdriver is used to install the module, ground it properly.

While cleaning by a vacuum cleaner, do not bring the sucking mouth near the module. Static electricity of the electrically powered driver or the vacuum cleaner may destroy the module.

ENVIRONMENTAL PRECAUTIONS

Operate the LCD module under the relative condition of 40°C and 50% relative humidity. Lower temperature can cause retardation of the blinking speed of the display, while higher temperature makes the overall display discolor.

When the temperature gets to be within the normal limits, the display will be normal. Polarization degradation, bubble generation or polarizer peel-off may occur with high temperature and humidity.

Contact with water or oil over a long period of time may cause deformation or colour fading of the display. Condensation on the terminals can cause electro-chemical reaction disrupting the terminal circuit

TROUBLE SHOOTING

When the power supply is given to the module, with the pin 3 (VL) connected to ground, all the pixels of a character gets activated in the following manner: All the characters

of a single line display, as in CDM 16108. The first eight characters of a single line display, operated in the two-line display mode, as in CDM 16116.

The first line of characters of a two-line display as in CDM 16216 and 40216. The first and third line of characters of a four-line display operated in the two-line display mode, as in CDM 20416.

If the above mentioned does not occur, the module should be initialized by software.

Make sure that the control signals 'E', R/W and RS are according to the interface timing requirements.

IMPROPER CHARACTER DISPLAY

When the characters to be displayed are missing between, the data read/write is too fast. A slower interfacing frequency would rectify the problem.

When uncertainty is there in the start of the first characters other than the specified ones are rewritten, check the initialization and the software routine.

In a multi-line display, if the display of characters in the subsequent lines doesn't take place properly, check the DD RAM addresses set for the corresponding display lines.

When it is unable to display data, even though it is present in the DD RAM, either the display on/off flag is in the off state or the display shift function is not set properly. When the display shift is done simultaneous with the data write operation, the data may not be visible on the display.

If a character not found in the font table is displayed, or a character is missing, the CG ROM is faulty and the controller IC has to be changed

If particular pixels of the characters are missing, or not getting activated properly, there could be an assembling problem in the module.

In case any other problems are encountered you could send the module to our factory for testing and evaluation.

CRYSTALONICS DISPLAY

Crystalloids dot –matrix (alphanumeric) liquid crystal displays are available in TN, STN types, with or without backlight. The use of C-MOS LCD controller and driver ICs result in low power consumption. These modules can be interfaced with a 4-bit or 8-bit micro processor /Micro controller.

The built-in controller IC has the following features:

- Correspond to high speed MPU interface (2MHz)
- 80 x 8 bit display RAM (80 Characters max)
- 9,920 bit character generator ROM for a total of 240 character fonts. 208 character fonts (5 x 8 dots) 32 character fonts (5 x 10 dots)
- 64 x 8 bit character generator RAM 8 character generator RAM 8 character fonts (5 x 8 dots) 4 characters fonts (5 x 10 dots)
- Programmable duty cycles
 - 1/8 – for one line of 5 x 8 dots with cursor
 - 1/11 – for one line of 5 x 10 dots with cursor
 - 1/16 – for one line of 5 x 8 dots with cursor
- Wide range of instruction functions display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift.
- Automatic reset circuit that initializes the controller / driver ICs after power on.

REGISTERS

The controller IC has two 8 bit registers, an instruction register (IR) and a data register (DR). The IR stores the instruction codes and address information for display data RAM (DD RAM) and character generator RAM (CG RAM). The IR can be written, but not read by the MPU. The DR temporally stores data to be written to /read from the DD RAM or CG RAM. The data written to DR by the MPU, is automatically written to the DD RAM or CG RAM as an internal operation. When an address code is written to IR, the data is automatically transferred from the DD RAM or CG RAM to the DR. data transfer between the MPU is then completed when the MPU reads the DR. likewise, for the next MPU read of the DR, data in DD RAM or CG RAM at the address is sent to the DR automatically.

Similarly, for the MPU write of the DR, the next DD RAM or CG RAM address is selected for the write operation.

The register selection table is as shown below:

RS	R/W	Operation
0	0	IR write as an internal operation
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DD, RAM or CG RAM)
1	1	DR read as an internal operation (DD RAM)

BUSY FLAG

When the busy flag is 1, the controller is in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/W = 1, the busy flag is output to DB7. The next instruction must be written after ensuring that the busy flag is 0.

ADDRESS COUNTER

The address counter allocates the address for the DD RAM and CG RAM read/write operation when the instruction code for DD RAM address or CG RAM address setting is input to IR, the address code is transferred from IR to the address counter. After writing/reading the display data to/from the DD RAM or CG RAM, the address counters increments/decrements by one the address, as an internal operation. The data of the address counter is output to DB0 to DB6 while R/W = 1 and RS = 0.

DISPLAY DATA RAM (DD RAM)

The characters to be displayed are written into the display data RAM (DD RAM), in the form of 8 bit character codes present in the character font table. The extended capacity of the DD RAM is 80 x 8 bits i.e. 80 characters.

CHARACTER GENERATOR ROM (CG ROM)

The character generator ROM generates 5 x 8 dot 5 x 10 dot character patterns from 8 bit character codes. It generates 208, 5 x 8 dot character patterns and 32, 5 x 10 dot character patterns.

CHARACTER GENERATOR RAM (CG RAM)

In the character generator RAM, the user can rewrite character patterns by program. For 5 x 8 dots, eight character patterns can be written, and for 5 x 10 dots, four character patterns can be written.

INTERFACING THE MICROPROCESSOR / CONTROLLER

The module, interfaced to the system, can be treated as RAM input/output, expanded or parallel I/O.

Since there is no conventional chip select signal, developing a strobe signal for the enable signal (E) and applying appropriate signals to the register select (RS) and read/write (R/W) signals are important.

The module is selected by gating a decoded module – address with the host – processor's read/write strobe. The resultant signal, applied to the LCDs enable (E) input, clocks in the data.

The 'E' signal must be a positive going digital strobe, which is active while data and control information are stable and true. The falling edge of the enable signal enables the data / instruction register of the controller. All module timings are referenced to specific edges of the 'E' signal. The 'E' signal is applied only when a specific module transaction is desired.

The read and write strobes of the host, which provides the 'E' signals, should not be linked to the module's R/W line. An address bit which sets up earlier in the host's machine cycle can be used as R/W.

When the host processor is so fast that the strobes are too narrow to serve as the 'E' pulse

- a. Prolong these pulses by using the hosts 'Ready' input
- b. Prolong the host by adding wait states
- c. Decrease the Hosts Crystal frequency.

In spite of doing the above mentioned, if the problem continues, latch both the data and control information and then activate the 'E' signal

When the controller is performing an internal operation the busy flag (BF) will set and will not accept any instruction. The user should check the busy flag or should provide a delay of approximately 2ms after each instruction. The module presents no difficulties while interfacing slower MPUs. The liquid crystal display module can be interfaced, either to 4-bit or 8-bit MPUs. For 4-bit data interface, the bus lines DB4 to DB7 are used for data transfer, while DB0 to DB3 lines are disabled. The data transfer is complete when the 4-bit data has been transferred twice. The busy flag must be checked after the 4-bit data has been transferred twice. Two more 4-bit operations then transfer the busy flag and address counter data.

For 8-bit data interface, all eight-bus lines (DB0 to DB7) are used.

PIN NO	Symbol	Fuction
1	VSS	GND
2	VDD	+5V
3	V0	Contrast adjustment
4	RS	H/L Register select signal
5	RW	H/L Read/Write signal
6	E	H/L Enable signal
7	DB0	H/L Data bus line
8	DB1	H/L Data bus line
9	DB2	H/L Data bus line
10	DB3	H/L Data bus line
11	DB4	H/L Data bus line
12	DB5	H/L Data bus line
13	DB6	H/L Data bus line
14	DB7	H/L Data bus line
15	A	+4.2V for LED
16	K	Power supply for BKL(0V)

Fig 15: LCD Pin diagram

POWER SUPPLY

Block diagram

The ac voltage, typically 220V rms, is connected to a transformer, which steps that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation.

A regulator circuit removes the ripples and also remains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

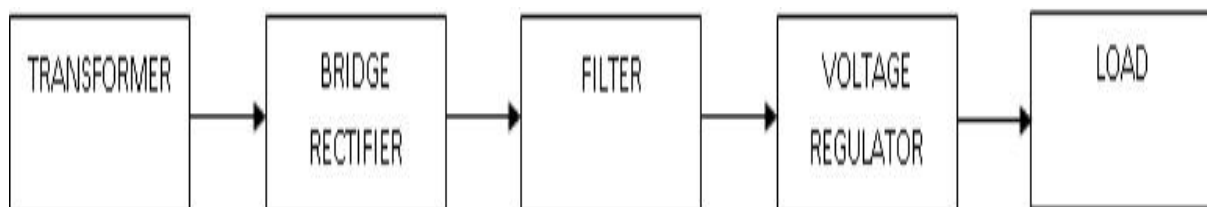


Fig18: Block diagram (Power supply)

Working principle

- **Transformer**

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op-amp. The advantages of using precision rectifier are it will give peak voltage output as DC, rest of the circuits will give only RMS output.

- **Bridge rectifier**

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.

Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

The path for current flow is from point B through D1, up through RL, through D3, through the secondary of the transformer back to point B. this path is indicated by the solid arrows. Waveforms (1) and (2) can be observed across D1 and D3.

One-half cycle later the polarity across the secondary of the transformer reverse, forward biasing D2 and D4 and reverse biasing D1 and D3. Current flow will now be from point A through D4, up through RL, through D2, through the secondary of T1, and back to point A. This path is indicated by the broken arrows. Waveforms (3) and (4) can be observed across D2 and D4. The current flow through RL is always in the same direction. In flowing through RL this current develops a voltage corresponding to that shown waveform (5). Since current flows through the load (RL) during both half cycles of the applied voltage, this bridge rectifier is a full-wave rectifier.

One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional full-wave circuit.

This may be shown by assigning values to some of the components shown in views A and B. assume that the same transformer is used in both circuits. The peak voltage developed between points X and y is 1000 volts in both circuits. In the conventional full-wave circuit shown—in view A, the peak voltage from the center tap to either X or Y is 500 volts. Since only one diode can conduct at any instant, the maximum voltage that can be rectified at any instant is 500 volts.

The maximum voltage that appears across the load resistor is nearly-but never exceeds-500 volts, as result of the small voltage drop across the diode. In the bridge rectifier shown in view B, the maximum voltage that can be rectified is the full secondary voltage, which is 1000 volts. Therefore, the peak output voltage across the load resistor is nearly 1000 volts. With both circuits using the same transformer, the bridge rectifier circuit produces a higher output voltage than the conventional full-wave rectifier circuit.

- **IC voltage regulators**

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to tens of watts.

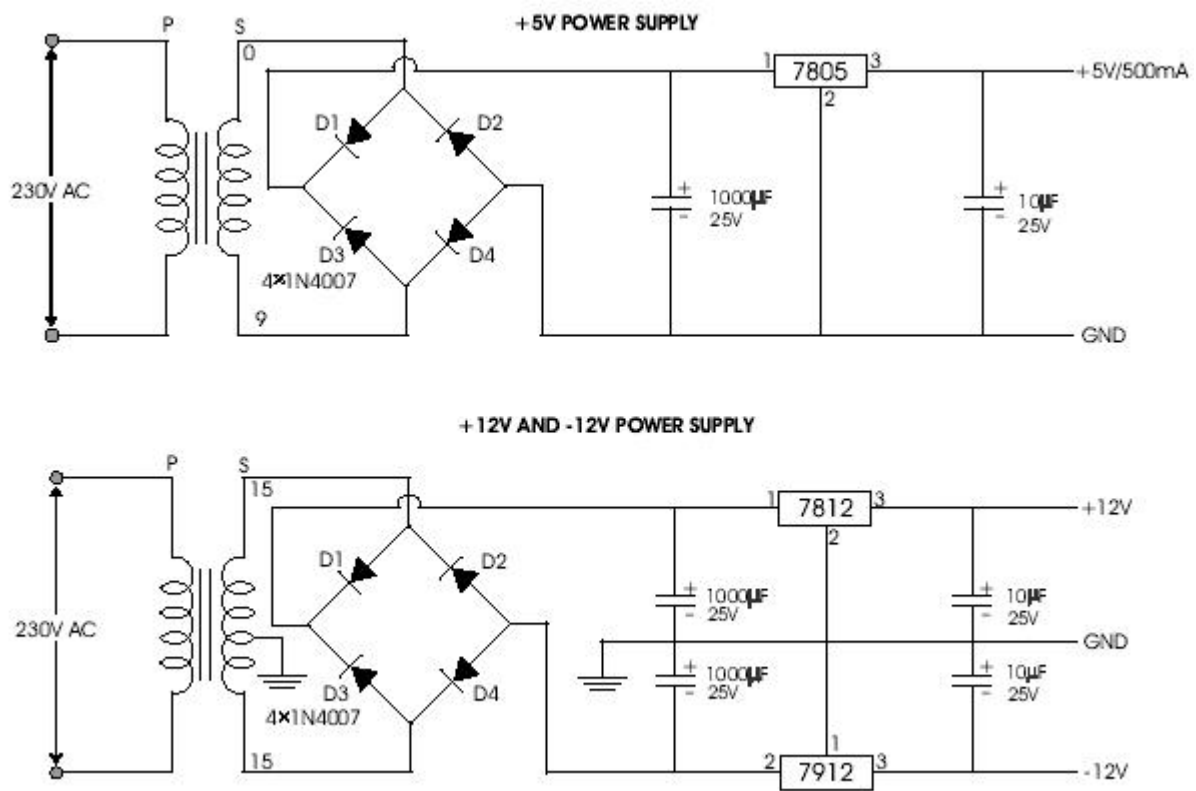


Fig19: Circuit diagram (Power supply)

A fixed three-terminal voltage regulator has an unregulated dc input voltage, V_i , applied to one input terminal, a regulated dc output voltage, V_o , from a second terminal, with the third terminal connected to ground.

The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts.

- For ICs, microcontroller, LCD ----- 5 volts
- For alarm circuit, op-amp, relay circuits ----- 12 volts

4.3.1 Software Requirements

- Arduino IDE
- GCC

4.3.2. ARDUINO IDE:

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

4.3.3 GCC:

GCC stands for “GNU Compiler Collection”. GCC is an integrated distribution of compilers for several major programming languages. These languages currently include C, C++, Objective-C, Objective-C++, Fortran, Ada, D, Go, and BRIG (HSAIL). The abbreviation GCC has multiple meanings in common use. The current official meaning is “GNU Compiler Collection”, which refers generically to the complete suite of tools. The name historically stood for “GNU C Compiler”, and this usage is still common when the emphasis is on compiling C programs. Finally, the name is also used when speaking of the language-independent component of GCC: code shared among the compilers for all supported languages. The language-independent component of GCC includes the majority of

the optimizers, as well as the “back ends” that generate machine code for various processors. The part of a compiler that is specific to a particular language is called the “front end”. In addition to the front ends that are integrated components of GCC, there are several other front ends that are maintained separately. These support languages such as Mercury, and COBOL. To use these, they must be built together with GCC proper. Most of the compilers for languages other than C have their own names. The C++ compiler is G++, the Ada compiler is GNAT, and so on. When we talk about compiling one of those languages, we might refer to that compiler by its own name, or as GCC. Either is correct. Historically, compilers for many languages, including C++ and Fortran, have been implemented as “preprocessors” which emit another high level language such as C. None of the compilers included in GCC are implemented this way; they all generate machine code directly. This sort of preprocessor should not be confused with the C preprocessor, which is an integral feature of the C, C++, Objective-C and Objective-C++ languages.

CHAPTER 5

EXPERIMENTATION

5.1 ARDUINO:

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

5.1.1 ARDUINO CONFIGURATION:

- Download Arduino IDE from official webpage.
- Right click to run on administrator mode.
- Click tools select communication port.
- If port not visible unplugs the usb cable and replug it.
- Click new file.
- Select and save destination folder.
- Type some code.
- Click right symbol button on top left corner to verify your program.
- Click the nearest one of right symbol to upload your code into board.
- If error occurs select board type and chip.
- Once the uploading complete it'll notify the status.

5.1.2 PROGRAMMING CODE:

Main.cpp

```
#include <Arduino.h>
#include <avr/io.h>
#include <stdlib.h>

#include "gpio.h"
#include "utils.h"
#include "i2c.h"
#include "lcd.h"
#include "keypad.h"
#include "uart.h"
#include "gsm.h"

#ifndef matrixconfig
#define matrixconfig
#define row1 7
#define row2 6
#define row3 5
#define row4 4
#define col1 3
#define col2 2
#define col3 AN1
#define col4 AN2
#endif

#ifndef rfoutputconfig
#define rfoutputconfig
#define rfoutput AN0
#endif

#ifndef braillepinconfig
#define braillepinconfig
#define braille1 13
#define braille2 12
#define braille3 11
#define braille4 10
#define braille5 9
#define braille6 8
#endif

#ifndef gsmpinconfig
#define gsmpinconfig
#define gsmtransmitter 1
#define gsmreceiver 0
#define gsmbaudrate 9600
#endif

const char *pattern =
"\x00\x01\x03\x09\x19\x11\x0B\x1B\x13\x0A\x1A\x05\x07\x0D\x1D\x15\x0F\x1F\x17\x0E\x1E\x2
5\x27\x3B\x2D\x3D\x35";
```

```

char ismessagereceived;
char isuserqueried;
char isreplydetected;

unsigned char lastdetectedkey;
unsigned char mobilenum[15];
unsigned int keydebouncetime;

char braille_indicate_new_message();
char braille_generate_pattern(unsigned char);
char braille_seperate_message(const char*);
char braille_seperate_number(const char*);
char braille_check_for_reply(const char*);

void setup()
{
    keypad_initialize(row1, row2, row3, row4, col1, col2, col3, col4);
    gpio_set_output(rfoutput);
    gpio_put_low(rfoutput);

    gpio_set_output(braille1);
    gpio_set_output(braille2);
    gpio_set_output(braille3);
    gpio_set_output(braille4);
    gpio_set_output(braille5);
    gpio_set_output(braille6);

    gpio_put_low(braille1);
    gpio_put_low(braille2);
    gpio_put_low(braille3);
    gpio_put_low(braille4);
    gpio_put_low(braille5);
    gpio_put_low(braille6);

    lcd_initialize(100);
    lcd_disp(0x80," BRAILLE 2 TEXT ");
    lcd_disp(0xC0," AND VICE-VERSA ");
    delay_ms(2500); lcd_function(clear_display);

    if(gsm_initialize(gsmbaudrate)) lcd_disp(0x80,"GSM FOUND");
    else lcd_disp(0x80,"GSM NOT FOUND"); delay_ms(500);
    lcd_function(clear_display);

    if(gsm_sim_status()) lcd_disp(0x80,"SIM FOUND");
    else lcd_disp(0x80,"SIM NOT FOUND"); delay_ms(500);
    lcd_function(clear_display);

    if(gsm_tower_wait()) lcd_disp(0x80,"SIM TOWERED");
    else lcd_disp(0x80,"N/W ERROR"); delay_ms(500);
    lcd_function(clear_display);

    if(gsm_msg_enable(true)) lcd_disp(0x80,"MSG READY");
    else lcd_disp(0x80,"MSG ERROR"); delay_ms(500);
    lcd_function(clear_display);

```

```

if(gsm_sim_terminal()) lcd_disp(0x80,"TERMINAL READY");
else lcd_disp(0x80, "TERMINAL ERROR"); delay_ms(500);
lcd_function(clear_display);

serial_flush();
}

void loop()
{
    lastdetectedkey = fetchkey(&keydebouncetime);

    if(gsm_msg_receive((char*)mobilenumber))
    {
        braille_generate_pattern(0x20);
        lastdetectedkey = false;

        if(braille_indicate_new_message())
        {
            braille_separate_number((char*)mobilenumber);
            braille_separate_message((char*)serialarray);
            braille_check_for_reply((char*)mobilenumber);
        }
        else gpio_put_low(rfoutput);

        braille_generate_pattern(0x20);
        memset(mobilenumber, '\0', sizeof(mobilenumber));
        serial_flush();
    }
    if(isdigit(lastdetectedkey) && keydebouncetime > 300)
    {
        braille_generate_pattern(lastdetectedkey + '1');
        delay_ms(500);
        braille_generate_pattern(0x20);
    }
    if(lastdetectedkey == 'A' && keydebouncetime > 300)
    {
        for(char k = 'a'; k <= 'z'; k++)
        {
            braille_generate_pattern(k);
            delay_ms(500);
        }
    }
}

char braille_indicate_new_message()
{
    unsigned long informationtimeout = millis() + 5000;
    unsigned long beepinterval = millis() + 500;

    lcd_disp(0x80,"NEW MSG ARRIVED ");
    lcd_disp(0xC0, mobilenumber);
    do
    {
        lastdetectedkey = fetchkey(&keydebouncetime);
        if(millis() > beepinterval)

```

```

    {
        gpio_put_toggle(rfoutput);
        beepinterval = millis() + 500;
    }
    if(isdigit(lastdetectedkey))
    {
        gpio_put_low(rfoutput);
        return true;
    }
}
while(millis() < informationtimeout);
lcd_disp(0x80,"NO RESPONCE DET ");
lcd_disp(0xC0,"BACK TO NORMAL ");
gpio_put_low(rfoutput);
return false;
}

char braille_generate_pattern(unsigned char c)
{
    if(c == 0x20) c = 0;
    else c = c - 'a' + 1;

    if(pattern[c] & (1 << 0)) gpio_put_high(braille1); else gpio_put_low(braille1);
    if(pattern[c] & (1 << 1)) gpio_put_high(braille2); else gpio_put_low(braille2);
    if(pattern[c] & (1 << 2)) gpio_put_high(braille3); else gpio_put_low(braille3);
    if(pattern[c] & (1 << 3)) gpio_put_high(braille4); else gpio_put_low(braille4);
    if(pattern[c] & (1 << 4)) gpio_put_high(braille5); else gpio_put_low(braille5);
    if(pattern[c] & (1 << 5)) gpio_put_high(braille6); else gpio_put_low(braille6);

    return true;
}

char braille_seperate_number(const char *number)
{
    unsigned char requireddigit = 0;
    lcd_function(clear_display);
    lcd_disp(0x80, number);
    do
    {
        lastdetectedkey = fetchkey(&keydebouncetime);
        if(lastdetectedkey == 'D' && keydebouncetime > 300) requireddigit++;
        lcd_disp(0xC0, "LETTER:"); lcd_write(0xC7, number[requireddigit]);
        if(isdigit(number[requireddigit]))
            braille_generate_pattern(number[requireddigit] + 0x31);
    }
    while(number[requireddigit]);
    braille_generate_pattern(0x20);
}

char braille_seperate_message(const char *holdable)
{
    char *p = strchr(holdable, '*');
    lcd_function(clear_display);
    lcd_disp(0x80, p);
    unsigned char requireddigit = 1;

```

```

do
{
    lastdetectedkey = fetchkey(&keydebouncetime);
    if(lastdetectedkey == 'D' && keydebouncetime > 300) requireddigit++;

    lcd_disp(0xC0, "LETTER:"); lcd_write(0xC7, p[requireddigit]);
    if(isdigit(p[requireddigit]) && keydebouncetime > 300) braille_generate_pattern(p[requireddigit] +
0x31);
    else if(isupper(p[requireddigit])) braille_generate_pattern(tolower(p[requireddigit]));
    else if(islower(p[requireddigit])) braille_generate_pattern(p[requireddigit]);
    else if(p[requireddigit] == ' ') braille_generate_pattern(0x20);
    else return true;
}
while(p[requireddigit] && p[requireddigit] != '\r' && p[requireddigit] != '\n');
return true;
braille_generate_pattern(0x20);
}

char braille_check_for_reply(const char *mobilenumber)
{
    lcd_function(clear_display);
    lcd_disp(0x80, "ENTER YOUR REPLY");
    unsigned long int delaytimeout = millis() + 10000;
    do
    {
        lastdetectedkey = fetchkey(&keydebouncetime);

        switch(lastdetectedkey)
        {
            case '1': gsm_msg_send(mobilenumber, "Hello..."); return true;
            case '2': gsm_msg_send(mobilenumber, "I'm Fine..."); return true;
            case '3': gsm_msg_send(mobilenumber, "Okey..."); return true;
            case 'A': return true;

            case '4': gsm_msg_send(mobilenumber, "Will Reach you..."); return true;
            case '5': gsm_msg_send(mobilenumber, "Text me later..."); return true;
            case '6': gsm_msg_send(mobilenumber, "I'm Busy Right Now..."); return true;
            case 'B': return true;

            case '7': gsm_msg_send(mobilenumber, "I'm in my home..."); return true;
            case '8': gsm_msg_send(mobilenumber, "I'm in Park..."); return true;
            case '9': gsm_msg_send(mobilenumber, "I'm in office..."); return true;
            case 'C': return true;

            case '*': gsm_msg_send(mobilenumber, "Good Morning..."); return true;
            case '0': gsm_msg_send(mobilenumber, "Good afternoon..."); return true;
            case '#': gsm_msg_send(mobilenumber, "Good Evening..."); return true;
            case 'D': return true;
        }
    }
    while(millis() < delaytimeout);
    lcd_disp(0xC0, "PRCOESS TERMINATED");
    return false;
}

```

5.2 ELECTRICAL BLOCK:

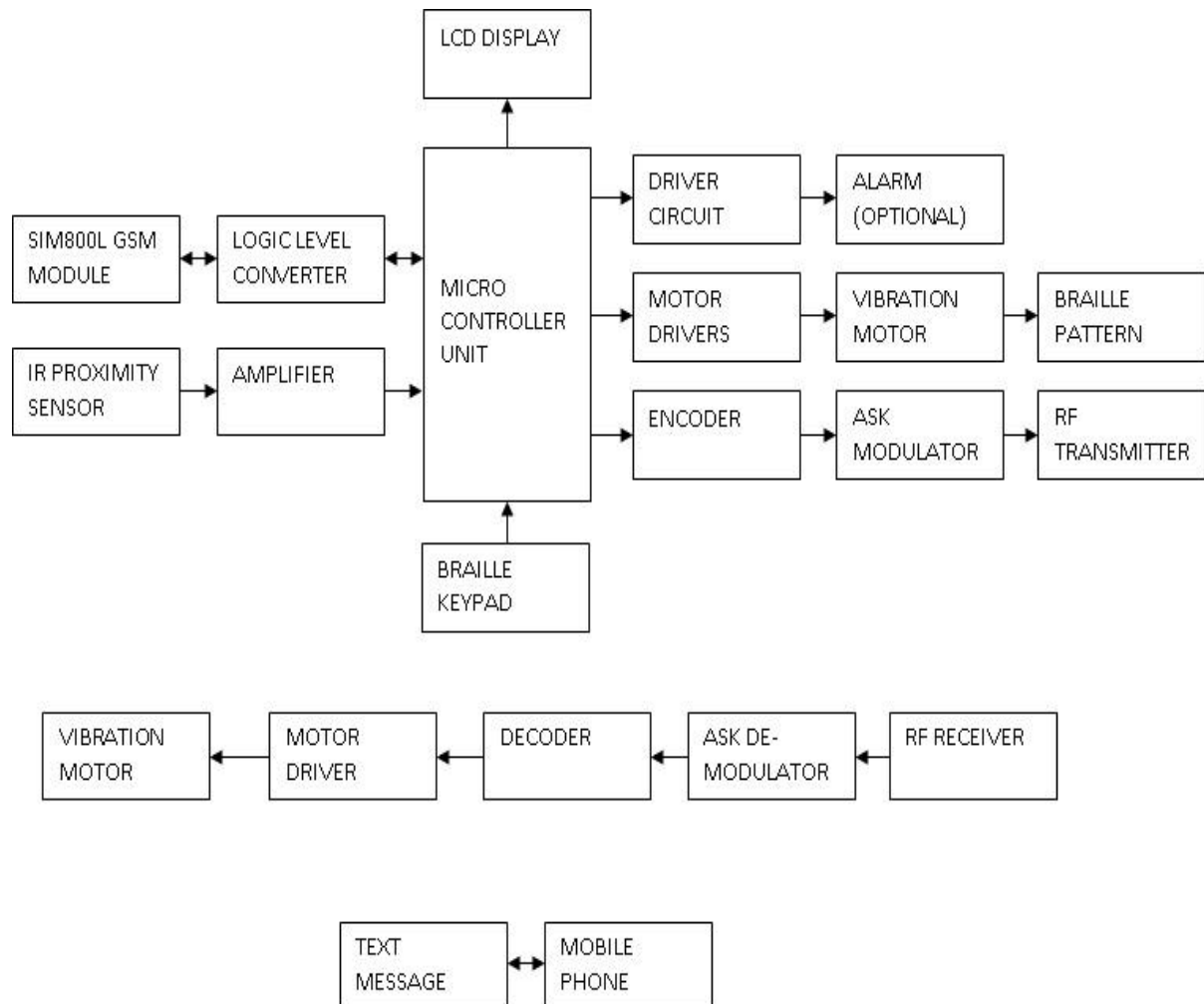


Fig 45. Block Diagram

5.2.1 FLOW DIAGRAM:

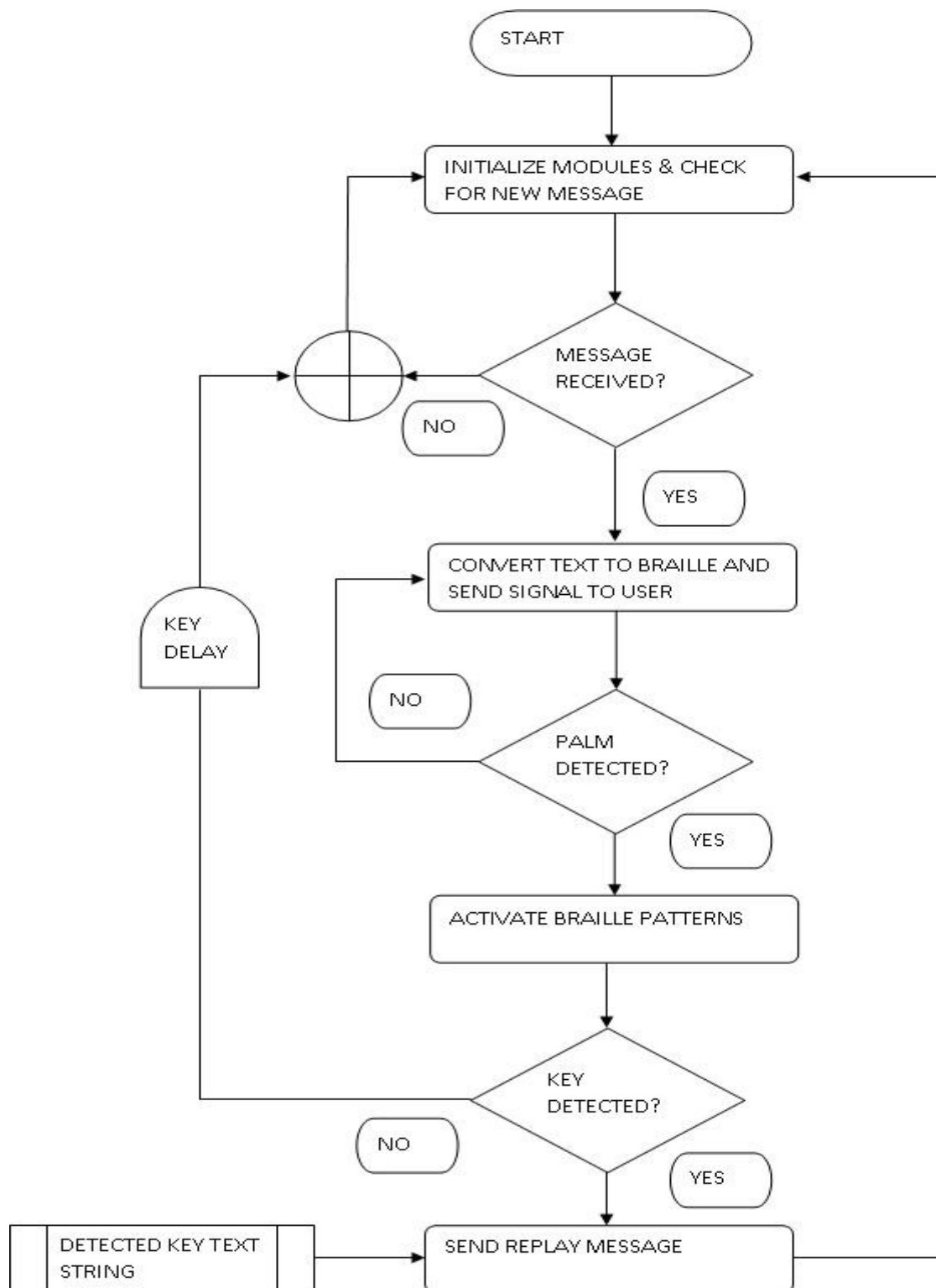


Fig 46: Flow Diagram

5.2.2 CIRCUIT DIAGRAM:

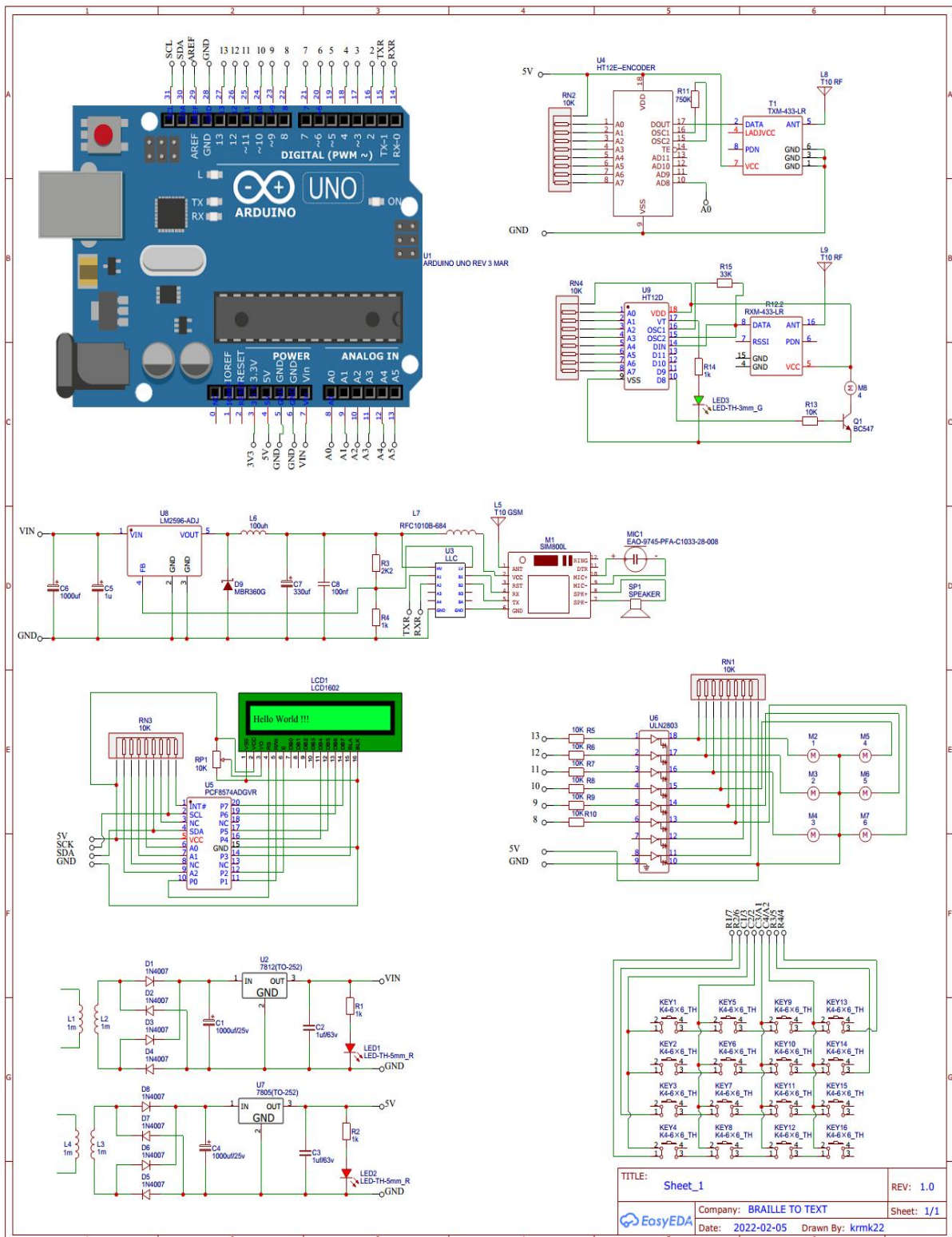


Fig 47: Circuit Diagram

5.2.3 DESCRIPTION:

The system described in this proposed project contains three section follows,

- **Text to Braille converter:** The section activated whenever new message received from GSM module and infrared proximity sensor detects palm in Braille patterned vibration motor. The system will separate text from message and start vibrating the motors according to the Braille key pattern.
- **Notifier:** The section activated whenever new message received from GSM module and the microcontroller will inform the new message arriving through radio frequency communication module. On receiver side, described as band in this project will start receive message and vibrate the motor till detection of key press.
- **Braille to text converter:** The section activated at end of the text to Braille section. This system will wait for the user replay for certain times, if user presses the button an acknowledge message will send to the last received number. Otherwise, it will keep checking the button for certain amount of delay or else new message arrival.

CHAPTER 6

RESULTS & DISCUSSIONS

6.1 RESULTS:



Power on

When power on it'll take few seconds to power up microcontroller unit and then controller initialize the liquid crystal display module and then print the project title on the display the data visible for few seconds after that controller initialize GSM Module.



GSM Initialization

After clearing the display data's controller initialize GSM module by using respective AT commands. The command consists with module found or not, echo cleared, GPRS

connection and server connection the result will print on the display. After particular time interval controller clear the display and execute next command.

Routine Execution

After initializing all sensor modules, controller will keep checks the gsm module for incoming message. If any message received and that message contains secret code which assures the message for deaf/blind people. Controller activates the vibrator module and waits for the user response.



Failed Response

Incase user not in home or near the proposed system. He/she not receive the radio frequency signal. The band will not vibrate so, we cannot process the message. But, that process has a limited time interval. If user fails to reach the transmitter kit within given timer controller will discard that message and wait for the new one.





Generating Braille pattern

After user informs his/her presence by pressing the keypad button, controller will activate the Braille pattern according received number and message. User need to press the button to move to next character. The button presses records until end of string. Once the string end detected, controller execute next routine.



Final Result

At the end of the message controller check the response from the user end. If user pressed the keypad button, predefined message strings are forward to the received number otherwise controller wait until the timer expired and restarts the loop again.

6.2.FUTURE SCOPE:

We look forward to add the facial recognition feature as part of the device. Likewise for additional improvement, a multi-language-Braille interpreter will be considered. Look-into tables for changed dialects could be put away in streak memory with the goal that when interpretation of content in a specific language is required, the microcontroller loads the relating look-into table. The vibration system may feel difficult to understand in future, we add braille punching system which will show exact braille pattern. Hereafter, we hope that the future hi-tech improvement will enlighten the life of the blind or deaf population.

6.3 ADVANTAGES:

- Less weight and flexibility
- User friendly
- Low power consumption
- Portable
- Low cost

6.4 LIMITATIONS:

- Chances of manual errors
- Lack of privacy

6.5 PHOTOGRAPHY:

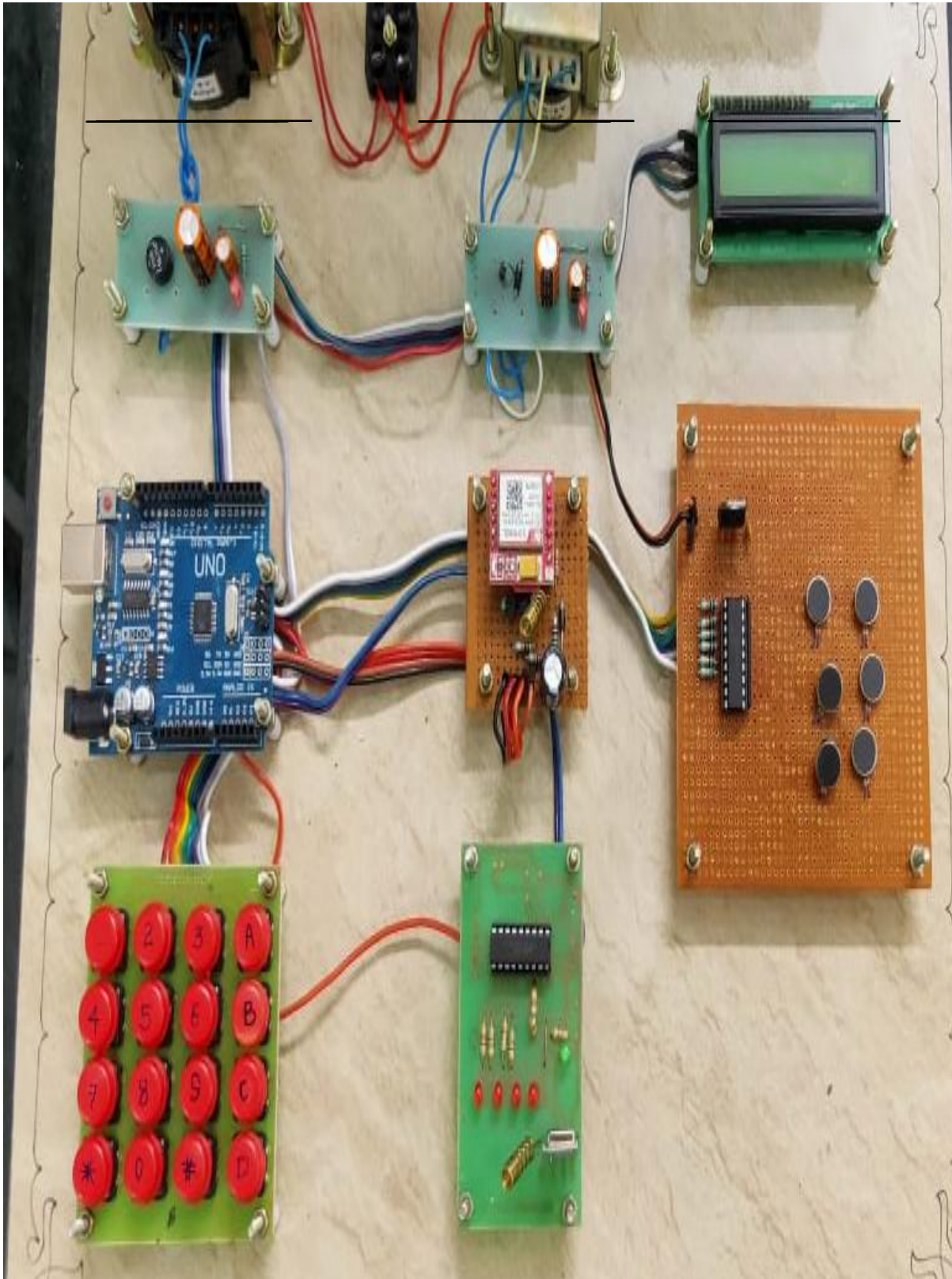


Fig 47: The assembled view of proposed system

CHAPTER 7

CONCLUSION

7.1 CONCLUSION:

The present project proposes a flexible and real-time text to Braille and display conversion system. In its present form, the framework can be utilized in a project, archive, book and so forth. For reading and easy understanding for the impaired audience. We've additionally added another usefulness of Text to Braille to help the outwardly hindered in understanding books, or tuning in to them straightforwardly as book recordings/content. In any case, there are a few upgrades which will be consolidated in future adaptations of the equipment interpreter. For instance, the present framework is an independent segment. Our device allows social, communicational and educational barriers to be removed for people with disabilities. Thus, with some modifications in previous conventional communicating devices, we can accommodate large number of visually impaired people in an updated communication system.

REFERENCE

- [1]. F. Ramirez-Garibay, C. M. Olivarria, A. F. E. Aguilera, and J. C. Huegel, “MyVox - Device for the communication between people: Blind, deaf, deaf-blind and unimpaired,” Proc. 4th IEEE Glob. Humanit. Technol. Conf. GHTC 2014, pp. 506–509, 2014, doi: 10.1109/GHTC.2014.6970330.
- [2]. R. Shylaja, R. L. Prasanna, M. Madhavi, and K. U. Rani, “Text to Braille Converter,” vol. 7, no. 5, pp. 6–9, 2018.
- [3]. R. Sibila, S. S. K, and B. Sowmya, “Text To Braille Conversion Using Matlab,” vol. 11, no. 1, pp. 1–8, 2018.
- [4]. Y. S. Saraswathi, S. Garg, S. Kulkarni, B. Kiran, and S. M. S. Read, “Advanced Braille System-Communication Device for Blind-Deaf People,” Int. Res. J. Eng. Technol., vol. 4, no. 6, pp. 319–322, 2017, [Online]. Available: <https://irjet.net/archives/V4/i6/IRJETV4I657.pdf>.
- [5]. P.G.Anuradha and K.Devibalan, “a Low Cost Portable Communication Device for the,” Int. J. Ind. Electron. Electr. Eng., pp. 22–25, 2016.
- [6]. R. Sarkar and S. Das, “Analysis of Different Braille Devices for Implementing a Cost-effective and Portable Braille System for the Visually Impaired People,” Int. J. Comput. Appl., vol. 60, no. 9, pp. 1–5, 2012, doi: 10.5120/9717-3073.
- [7]. P. J. Swati Malik, “Low Cost Portable E-Braille Device for Blind and Visually Impaired Persons,” Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ., vol. 3297, no. 6, pp. 11449–11455, 2016, doi: 10.15680/IJIRCCE.2016.
- [8]. K. Bawdekar, A. Kumar, and R. Das, “Text To Braille Converter,” Int. J. Electron. Commun. Eng. Technol., vol. 7, no. 4, pp. 54–61, 2016.