# Manarat International University


# GSM and GPS Based Smart Stick for Helping Blind Person

A Project submitted in partial fulfillment of the requirements for the degree of "Bachelor of Science" in Electrical and Electronic Engineering (EEE)


Submitted By

| | |
|---|---|
| Md. Saklain Mostak | ID: 1917EEE00177 |
| Masud Rana | ID: 1916EEE00172 |
| Md. Kawsar Ali | ID: 2018EEE00187 |
| Md. Soumik Hasan | ID: 1710EEE00128 |

**Supervised By**

**Sayeed Islam**

Assistant Professor & Coordinator, EEE (Evening) Program
Department of Electrical & Electronic Engineering
Manarat International University (MIU)


Fall 2023

**Department of Electrical & Electronic Engineering**
**Manarat International University (MIU)**
**Ashulia, Savar, Dhaka Bangladesh**
**Fall 2023**

# GSM and GPS Based Smart Stick for Helping Blind Person

**Md. Saklain Mostak** Author
ID: 1917EEE00177
Department of Electrical & Electronic Engineering
Manarat International University (MIU)

---

**Masud Rana**
ID: 1916EEE00172
Department of Electrical & Electronic Engineering
Manarat International University (MIU)

---

**Md. Kawsar Ali**
ID: 2018EEE00187
Department of Electrical & Electronic Engineering
Manarat International University (MIU)

---

**Md. Soumik Hasan**
ID: 1710EEE00128
Department of Electrical & Electronic Engineering
Manarat International University (MIU)

---

**Sayeed Islam** Supervisor
Assistant Professor & Coordinator, EEE (Evening) Program
Department of Electrical & Electronic Engineering
Manarat International University (MIU)

# Declaration

We to hereby declare that the thesis titled "**GSM and GPS Based Smart Stick for Helping Blind Person**" Presented in this project is the outcome of the investigation performed by us under the supervision of **Mr. Sayeed Islam,** Assistant professor & Coordinator, EEE(Evening) Program, Department of Electrical & Electronic Engineering (EEE) of Manarat International University (MIU).

We also declare that no part of this project and therefor has been or in being submitted elsewhere for the award of any degree.

| Name | ID Number | Signature |
|---|---|---|
| **Md. Saklain Mostak** | ID:1917EEE00177 | |
| **Masud Rana** | ID:1916EEE00172 | |
| **Md. Kawsar Ali** | ID:2018EEE00187 | |
| **Md. Soumik Hasan** | ID:1710EEE00128 | |

# Certificate of Approval

This is to certify that the project entitled **"GSM and GPS Based Smart Stick for Helping Blind Person"** being submitted by Md. Saklain Mostak, (1917EEE00177), Md. Masud Rana, (1916EEE00172), Md. Kawsar Ali, (2018EEE00187 ), Md. Soumik Hasan, (1710EEE00128 ) has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Electrical and Electronic Engineering.

# Board of Examination

_____

(Supervisor)

**Sayeed Islam**

Assistant Professor & Coordinator, EEE (Evening) Program

Department of Electrical & Electronic Engineering (EEE)

Manarat International University (MIU)

_____

(Chairman)

**K.M. Aktheruzzaman**

Head

Department of Electrical & Electronic Engineering (EEE)

Manarat International University (MIU)

# Dedication

We dedicate this book to our parents for nursing us with affection and love and to our Honorable Teachers for inspiring us.

# Acknowledgement


## “All praises and thanks to ALLAH”

First, we want to express our gratefulness to almighty Allah (SWT) for His diving blessing makes us possible to complete this thesis successfully. We have benefited a lot from this thesis work. This project has been a rewarding knowledge. We have learnt about **GSM and GPS Based Smart Stick for Helping Blind Person** and also economic estimation by studying various journals, conference papers, textbooks, notes as well as internet. We would like to express our most gratitude and appreciation to our thesis supervisor, **Mr. Sayeed Islam**, Assistant professor & Coordinator, EEE(Evening) Program, Department of Electrical & Electronic Engineering (EEE) of Manarat International University (MIU). For his continuous support and encouragement throughout the work. It was quite impossible for us to complete this work without his valuable expertise, advice and motivation. Finally, we must acknowledge with due respect the constant support and patience of our parents. For any errors or inadequacies that may remain in this work, of course the responsibility is entirely our own. I would also like to thank all our faculty member, lab instructor of the Department of Electrical & Electronic Engineering (EEE) of Manarat International University (MIU). The Authors are also grateful to **Mr. K.M. Aktheruzzaman**, Assistant Professor & Head, Department of Electrical & Electronic Engineering (EEE). **Mr. Mir Muhammad Aminuzzaman**, Lecturer, Department of Electrical & Electronic Engineering (EEE) of Manarat International University (MIU) and to all respected teachers and significant help for completing the project work successfully. Thank you all.

# Abstract

This paper presents a theoretical model and a system concept to provide a smart stick for blind people. This system is intended to provide overall measures object detection and real time assistance via Global Positioning System (GPS). The system consists of ultrasonic sensor, GPS Module, GSM Module, and buzzer alarm circuit. This project aims at the help blind people to find obstacle free path. When the object is detected near to the blinds stick it alerts them. Arduino pro mini is the heart of the device. It stores the data of the current location which it receives from the GPS system. So that it can make use of the data stored to compare with the destination location of the user. When the stick is broken, it will send a message. The location of the blind is found using Global System for Mobile communications (GSM) and Global Position System (GPS). Sticks are the supporting devices to help visually impaired individuals in scanning their surroundings and identifying the obstacles without the need of assistance from external agents. The current mechanical form of the sticks needs up-gradation for better support to the subject. Therefore, this paper proposes a modification to these sticks with the accession of sensors, micro-controllers, and buzzers that aids the subject better navigate their environment/surroundings and thus provides enhanced mobility experience. The sensors detect the hurdles and obstacles from a safe distance and the micro-controller activates a buzzer to alert the user in case of threats. The model employs another unit to send location updates to kith and kins of the subject in emergency conditions with the help of GPS and GSM units. Any abnormal changes in these readings result in an automated text message sent to the mentioned contacts along with the GPS location. The model is aimed to be a cost-effective and user-friendly device for aiding the visually challenged people that guarantees highly reliable navigation performance and greater user experience by offering various additional features.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER ONE

# Introduction

# Introduction

## 1.1 Introduction

Blindness is a common disability among the peoples throughout the world. According to the Distance detector is any device capable of measuring the distance between two points. The origins of distance measurement by means of graduated lengths of material such as chain, tape measure or piece of knotted rope are lost to antiquity. Optical distance measurement also has a long history and is usually taken to stem from the work of James Watt in 1771. Electro-magnetic measurements make up a third method of instrument have been produced to make indirect distance measurement using light. All kinds of devices or equipment nowadays, begin with the basic design, basic theory and then all the weakness follows by improvement step by step.

This project will serve the same purpose. In this improvement can be applied to bring the advantages to the user when measuring the distance by troubleshooting system plays a major role. Ultrasonic sensors work on a principle similar to radar or sonar. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object. That signal is send to the embedded systems. This project is about an alternative to walking stick which is normally used by blinds. It is fully automated, easy to maintain, cheap and it is very comfortable to use.

The power consumption is low and can be operated easily. Above all the alternative is economic over the conventional options. It is an arm band that contains a circuit board. World Health Organization (WHO) 285 million people are visually impaired worldwide, 39million are blind and 246 million have low vision. Blindness is a curse as they need help to work outside and do all other daily essential works. This paper glows a system that tries to remove the curse of blindness and make them self-dependent to their daily chores. Moving through an unknown environment becomes a real challenge when we can’t rely on our own eyes.

Blindness is a common disability among the peoples throughout the world. According to the

Distance detector is any device capable of measuring the distance between two points. The origins of distance measurement by means of graduated lengths of material such as chain, tape measure or piece of knotted rope are lost to antiquity. Optical distance measurement also has a long history and is usually taken to stem from the work of James Watt in 1771. Electromagnetic measurements make up a third method, where the time of travel of radio or light waves is converted into a distance. Since James Watt, hundreds of different types of instruments have been produced to make indirect distance measurement using light.

All kinds of devices or equipment nowadays, begin with the basic design, basic theory and then all the weakness follows by improvement step by step. This project will serve the same purpose. In this improvement can be applied to bring the advantages to the user when measuring the distance by troubleshooting several problems that arises. In that process embedded system play to major role. This project is about an alternative to walking stick which is normally used by blinds. It is fully automated, easy to maintain, cheap and it is very comfortable to use. The power consumption is low and can be operated easily. Above all the alternative is economic over the conventional options. It is an arm band that contains a circuit board.

## 1.2 Aims and Objectives

We have some specific objectives for this project and they are pointed below:

- To design and construct an **GSM and GPS Based Smart Stick for Helping Blind Person.**
- To put the whole prototype under several tests by creating different scenarios to validate our work.
- To take notes from our real-time tests so that those can be used for future improvements.

## 1.3 Project Methodology

Our used methodology for the project:

- **GSM and GPS Based Smart Stick for Helping Blind Person** and designing a block diagram & circuit diagram to know which components need to construct it.
- Setting all components in a PCB board & soldering. Then assembling the whole block in a board and finally run the system & checking.
- To take necessary notes from the project for future improvements.

## 1.4 Thesis Outline

The report has been organized into five chapters.

**Chapter 1:** Discusses introduction, methodology and objective.
**Chapter 2:** Description of literature review.
**Chapter 3:** Description of circuit elements, such as power supply, micro-controller, and circuit board building.
**Chapter 4:** Description of result and discussion.
**Chapter 5:** Finally, Advantages, Limitations, Applications Discussion & Conclusion.

# CHAPTER TWO

# Background and Motivation

# Background and Motivation

## 2.1 Introduction

In this chapter we will discuss some literature review. Here are some of the ideas we got after this project related literature.

## 2.2 Literature Review

In previous decades, several numbers of work or project was done on solar system for different purpose. Some are described below:

Yusro et al. [10] proposed a system design of a white, GPS enabled smart environment explorer stick (SEES). This helped visually impaired subjects to roam 2 Sourodip et al.around freely, both indoors and outdoors. Further, James on et al. [5] design eda wearable system for warning visually impaired subjects to avoid any possible collisions. In comparison to other features, this device throws a notification alert when any hazard is detected. It uses two ultrasonic transducers for locating objects.

The authors also worked on optimizing the performance of the device in terms of obstacle range localization and reduction of false alarms maintaining low power consumption. Liarokapis et al. [7] uses a combination of computer vision technique in conjunction with GPS technology to create an AR/VR interface which offers a very efficient navigation tool for the concerned subjects. Further- more, Loomis et al. [8] conducted a test where visually impaired subjects were given a backpack containing computing facilities, each containing GPS modules, and headphones.

They also attached a motion-detector module specifically for assessing head movements. A GPS enabled audio interfaced system was also integrated with it to facilitate the subject to hear and know its current location. Laurent et al. [6] made use of a SONAR model that mimicked the echolocation property of bats as the blind mobility aid. It enhanced the reliability of navigation by using methods involving spatial hearing. Several such directional aids for these subjects can be found in the works proposed in [1, 2, 4, 9]. The above systems have proved to offer an effective aid to the visually impaired subjects. However, employing them is practically

infeasible due to the high computational resource requirements of the system and the financially exorbitant nature of the model. Hence, this paper proposes a smart stick model that is financially affordable for a larger section of such people with added features compared to the above-proposed models, thus giving them a better user experience, and making the lives of vision-impaired subjects easy and smooth.

# CHAPTER THREE
# The Design Methods and Procedures

# The Design Methods and Procedures

## 3.1 Introduction

In this section we will describe here about our project structure like block diagram, circuit diagram, instrument price with list, instrument details and its working principle.

## 3.2 System Design

In this system we try to make a **GSM and GPS Based Smart Stick**. So here we use Arduino pro mini, Buck Converter, Relay, Battery, Ultrasonic sensor, GSM, GPS, Vibrator Motor and Buzzer etc. All of this component's connection and design is given bellow-

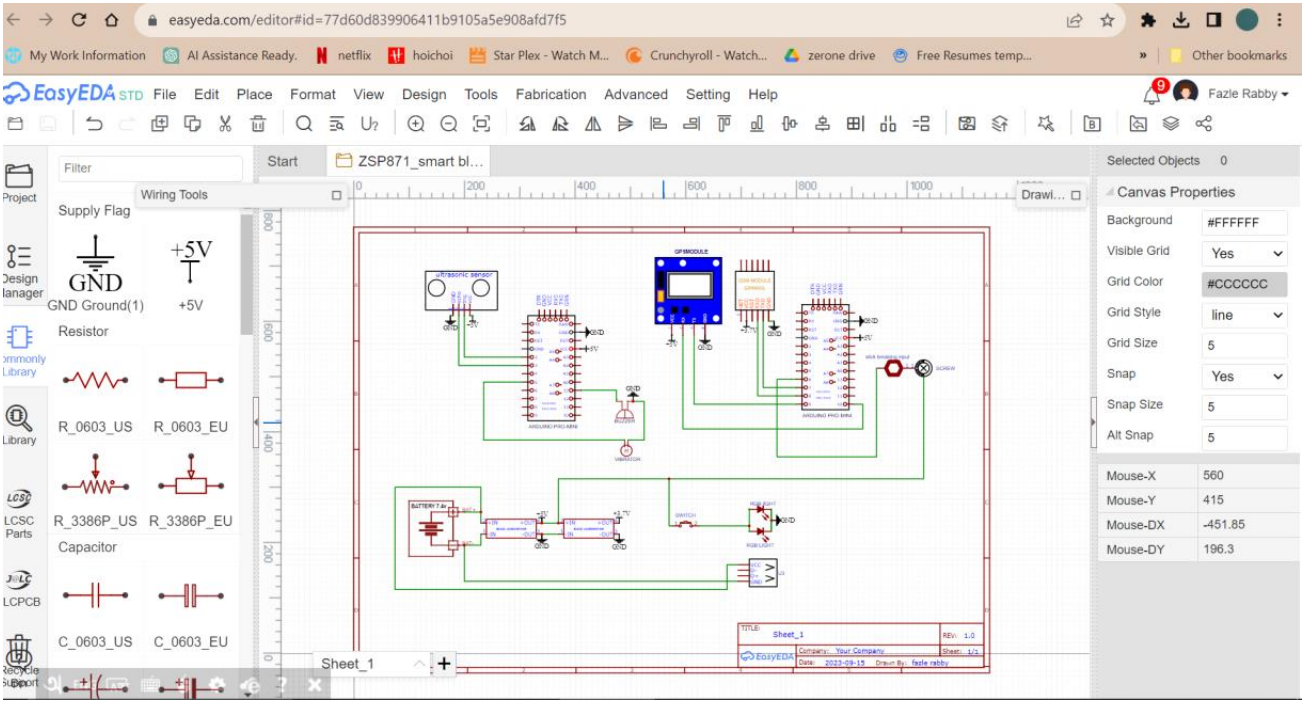


Figure 3.1: System Structure

## 3.3 Flow Chart

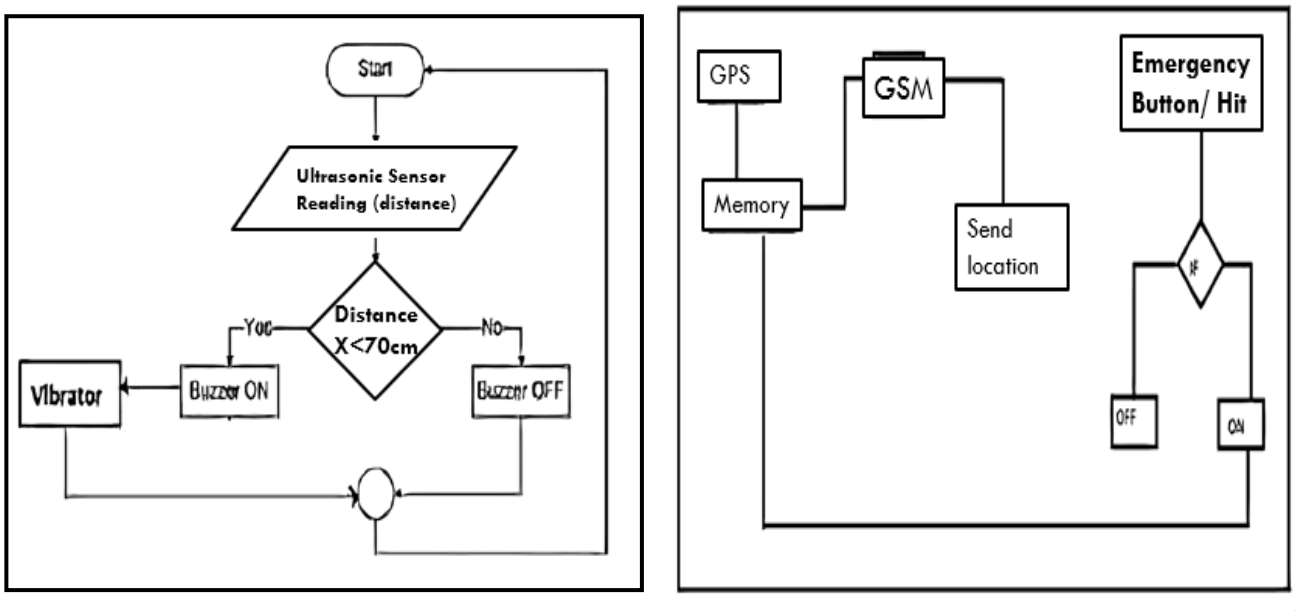


Figure 3.2: Block Diagram

## 3.4 Block Diagram

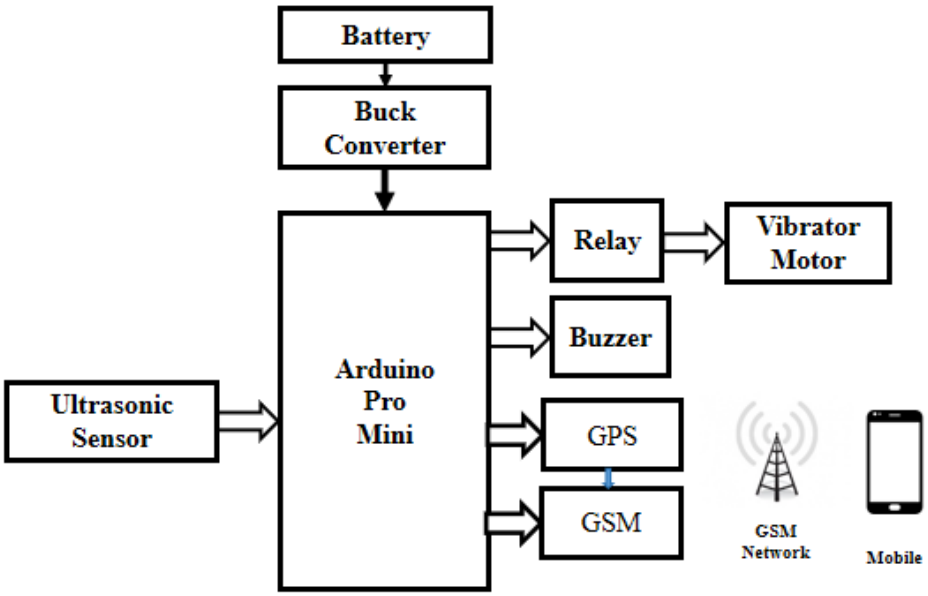


Figure 3.3: Block Diagram

## 3.5 Circuit Diagram

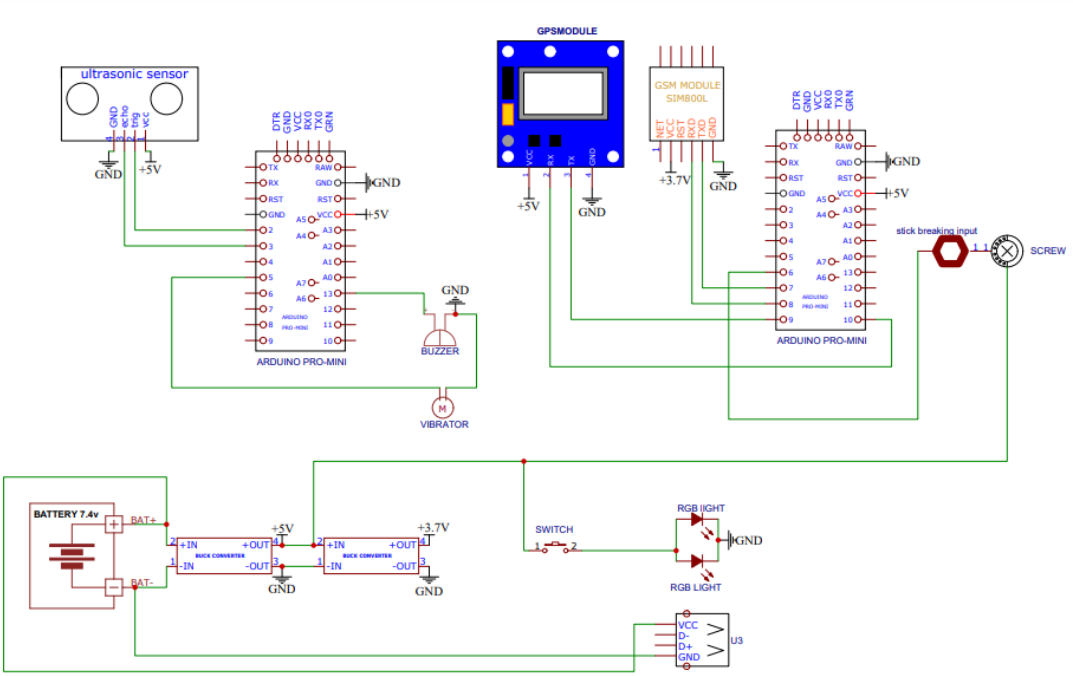


Figure 3.4: Project Circuit Diagram

## 3.6 Working Principle

The smart stick's ultrasonic sensor emits high-frequency sound waves in a specific direction (usually forward) and measures the time it takes for these waves to bounce back after hitting an obstacle. Based on the time taken for the sound waves to return, the smart stick calculates the distance to the nearest obstacle in the user's path. If an obstacle is detected, the sound or buzzer module generates audible alerts, such as beeping sounds or tones of varying intensity or frequency, to inform the user of the obstacle's proximity. Simultaneously, the vibrator module may provide tactile feedback, like vibrations of varying intensity or patterns, to convey the distance to the obstacle through touch. The smart stick continuously communicates with the GPS module to determine the user's real-time location. This information is essential for tracking the user's movements and providing location-based services. The GPS data, along with other relevant information, is transmitted via the GSM (mobile network) module. This module

allows the smart stick to connect to the cellular network and send messages or data to designate recipients. To keep the smart stick powered, it may have a built-in mobile charging system or a USB port. This is our main system of our project.

## 3.7 Project Prototype Image

Figure 3.5: Project Prototype Image

## 3.8 Components List with Price

Table 01: Components list with price

| Serial No | Product Name | Specification | Quantity | Unite Price | Total Price |
|---|---|---|---|---|---|
| 01 | Arduino | Pro Mini | 2 | 600 | 1200 |
| 02 | Buck Converter | LM2596 | 1 | 120 | 120 |
| 03 | Battery | 3.7V | 2 | 100 | 200 |
| 04 | Relay | | 1 | 90 | 90 |
| 05 | Ultrasonic Sensor | | 2 | 250 | 500 |
| 06 | GSM | SIM800L | 1 | 390 | 390 |
| 07 | GPS | | 1 | 390 | 390 |
| 08 | Vibrating Motor | 5V | 1 | 140 | 140 |
| 09 | Buzzer | 5V-12V | 1 | 30 | 30 |
| 10 | Other | | | | 1850/= |
| Total : Four thousand nine hundred ten | | | | | 4910/= |

## 3.9 Arduino Pro Mini

The Arduino Pro Mini is a micro-controller board based on the ATmega168. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. A six pin header can be connected to an FTDI cable to provide USB power and communication to the board.

**Specification**

- Micro-controller ATmega168
- Operating Voltage: 3.3V or 5V (depending on model)
- Input Voltage: 3.35 -12 V (3.3V model) or 5 - 12 V (5V model)
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- Flash Memory: 16 KB (of which 2 KB used by boot loader)
- SRAM: 1 KB
- EEPROM: 512 bytes
- Clock Speed: 8 MHz (3.3V model) or 16 MHz (5V model)

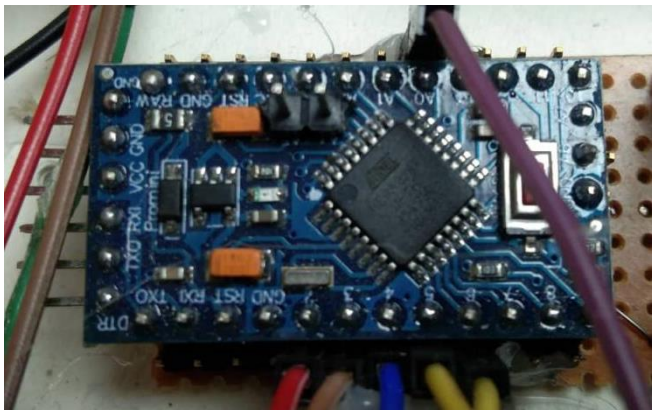

Figure 3.6: Arduino Pro Mini

**Pin Out**

Each of the 14 digital pins on the Pro Mini can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 3.3 or 5 volts (depending on the model). Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the TX-0 and RX-1 pins of the six pin header.

- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.

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

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

- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

  The Pro Mini has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). Four of them are on the headers on the edge of the board; two (inputs 4 and 5) on holes in the interior of the board. The analog inputs measure from ground to VCC. Additionally, some pins have specialized functionality:

  There is another pin on the board:

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

  See also the mapping between Arduino pins and ATmega168 ports.

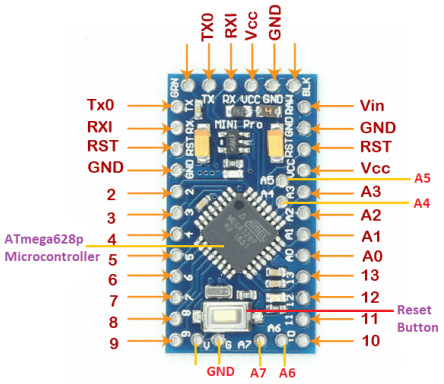


Figure 3.7: Arduino Pro Mini Pin Out

The high-performance Microchip Pico Power 8-bit AVR RISC-based micro controller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

**Micro controller IC ATmega328p**

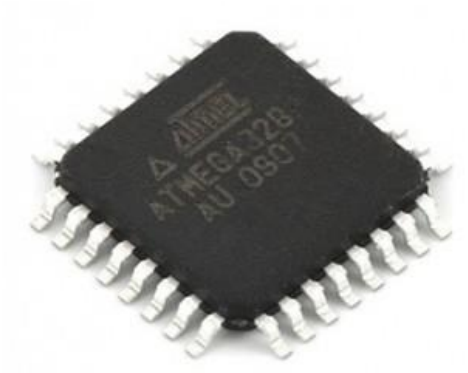


Figure 3.8: Micro controller IC AT Mega 328p

## 3.10 Relay

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

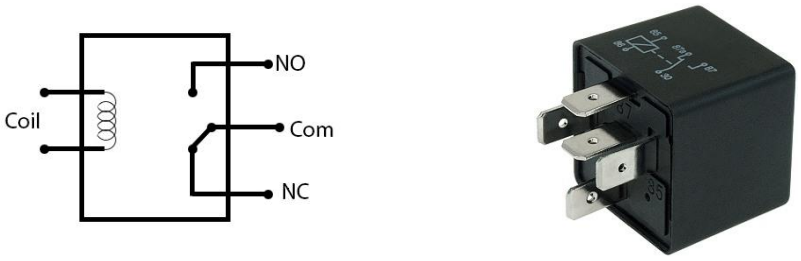


Figure 3.9: Relay

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts. Magnetic latching relays can have either single or dual coils. On a single coil device, the relay will operate in one direction when power is applied with one polarity, and will reset when the polarity is reversed. On a dual coil device, when polarized voltage is applied to the reset coil the contacts will transition. AC controlled magnetic latch relays have single coils that employ steering diodes to differentiate between operate and reset commands. The relay module is the one in the figure below.

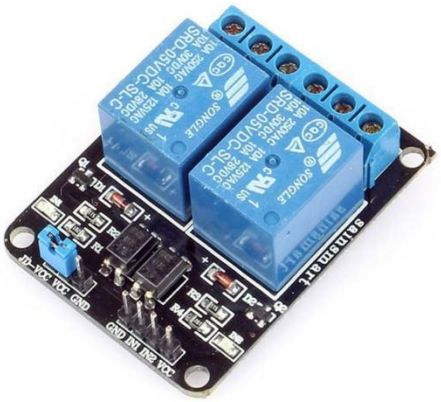

Figure 3.10: Relay Module.

This module has two channels (those blue cubes). There are other varieties with one, four and eight channels.Main's voltage connections.

In relation to mains voltage, relays have 3 possible connections:

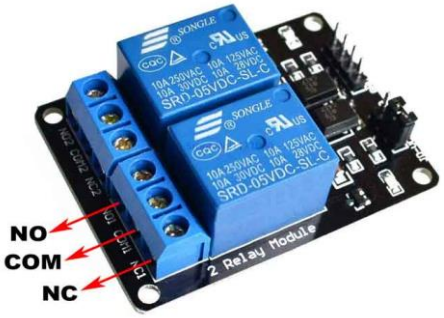


Figure 3.11: Main's voltage connections.

**COM**: Common pin

**NO (Normally Open):** There is no contact between the common pin and the normally open pin. So, when you trigger the relay, it connects to the COM pin and supply is provided to a load

**NC (Normally Closed):** There is contact between the common pin and the normally closed pin. There is always connection between the COM and NC pins, even when the relay is turned off. When you trigger the relay, the circuit is opened and there is no supply provided to a load.

# Pin wiring

The connections between the relay module and the Arduino are really simple:

**GND**: goes to ground
**IN**: Controls the first relay (it will be connected to an Arduino digital pin)
**IN2**: Controls the second relay (it should be connected to an Arduino digital pin if you are using this second relay. Otherwise, you don't need to connect it)
**VCC**: Goes to 5V

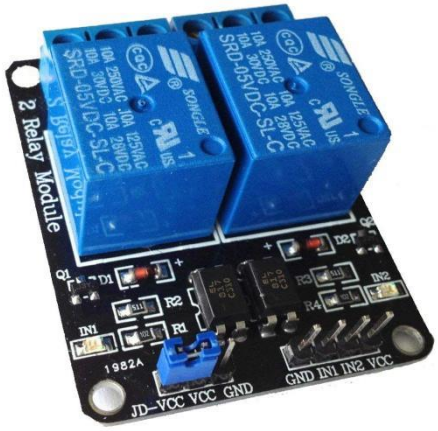


Figure 3.12: Voltage connections.

## 3.11 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

**HC-SR04 Ultrasonic Sensor - Working**

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4-pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in

many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

**Distance = Speed × Time ----------(1)**

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below-


Figure 3.13: Working of sensor

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a micro-controller or microprocessor.

**How to use the HC-SR04 Ultrasonic Sensor**

HC-SR04 distance sensor is commonly used with both micro-controller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used. Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the micro-controller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of

40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor. The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information, the distance is measured as explained in the above heading.

**Applications**

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc. can be measured since the waves can penetrate through water.

**Table 3.1: Ultrasonic Sensor Pin Configuration**

| Pin Number | Pin Name | Description |
|---|---|---|
| 1 | VCC | The VCC pin powers the sensor, typically with +5V |
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system. |

**HC-SR04 Sensor Features**

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

## 3.12 Battery

Lithium batteries are primary batteries that have metallic lithium as an anode. These types of batteries are also referred to as lithium-metal batteries. They stand apart from other batteries in their high charge density and high cost per unit. Depending on the design and chemical compounds used, lithium cells can produce voltages from 1.5 V (comparable to a zinc–carbon or alkaline battery) to about 3.7 V.

Disposable primary lithium batteries must be distinguished from secondary lithium-ion or a lithium-polymer, which are rechargeable batteries. Lithium is especially useful, because its ions can be arranged to move between the anode and the cathode, using an intercalated lithium compound as the cathode material but without using lithium metal as the anode material. Pure lithium will instantly react with water, or even moisture in the air; the lithium in lithium ion batteries is in a less reactive compound.

Lithium batteries are widely used in portable consumer electronic devices. The term "lithium battery" refers to a family of different lithium-metal chemistries, comprising many types of cathodes and electrolytes but all with metallic lithium as the anode. The battery requires from 0.15 to 0.3 kg of lithium per kWh. As designed these primary systems use a charged cathode, that being an electro-active material with crystallographic vacancies that are filled gradually during discharge.


Figure 3.14: 3.7V Battery

## Product Specification

| | |
|---|---|
| Voltage | 3.7 V |
| Product Type | Lithium-Ion |
| Battery Capacity | 2200mAh |
| Weight | 45 g |
| Model Number | ICR 18650 |

## 3.13 GSM

At the heart of the module is a SIM800L GSM cellular chip from Sim Com. The operating voltage of the chip is from **3.4V to 4.4V**, which makes it an ideal candidate for direct LiPo battery supply. This makes it a good choice for embedding into projects without a lot of space.


Figure 3.15: GSM Module

All the necessary data pins of SIM800L GSM chip are broken out to a 0.1″ pitch headers. This includes pins required for communication with a micro-controller over **UART**. The module supports baud rate from **1200bps** to **115200bps** with Auto-Baud detection.

The module needs an external antenna to connect to a network. The module usually comes with a **Helical Antenna** and solders directly to NET pin on PCB. The board also has a U.FL connector facility in case you want to keep the antenna away from the board.

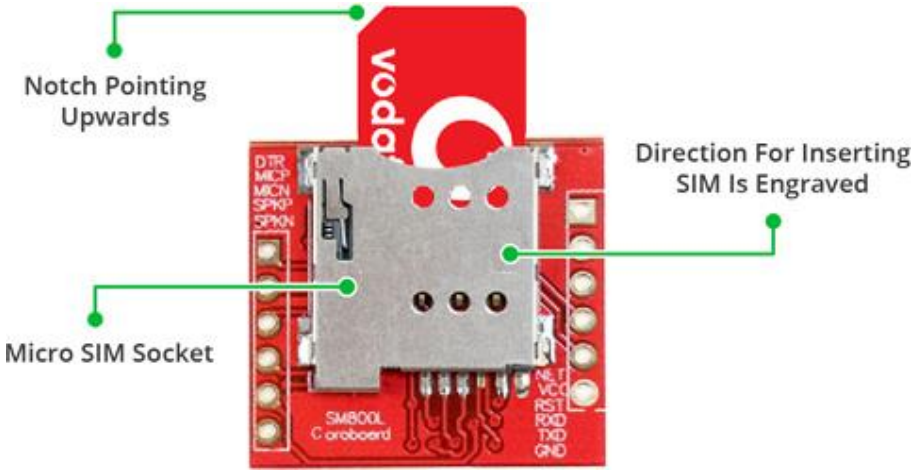


Figure 3.16: GSM Module Introducing

There's a SIM socket on the back! Any activated, **2G micro SIM card** would work perfectly. Correct direction for inserting SIM card is normally engraved on the surface of the SIM socket. This module measures only 1 inch² but packs a surprising amount of features into its little frame. Some of them are listed below:

- Supports Quad-band: GSM850, EGSM900, DCS1800 and PCS1900
- Connect onto any global GSM network with any 2G SIM
- Make and receive voice calls using an external 8Ω speaker & electret microphone
- Send and receive SMS messages
- Send and receive GPRS data (TCP/IP, HTTP, etc.)
- Scan and receive FM radio broadcasts
- Transmit Power:
    - Class 4 (2W) for GSM850
    - Class 1 (1W) for DCS1800
- Serial-based AT Command Set
- FL connectors for cell antennae
- Accepts Micro SIM Card

## SIM800L GSM Module Pinout

The SIM800L module has total 12 pins that interface it to the outside world. The connections are as follows:


Figure 3.17: GSM Module Pinou

NET is a pin where you can solder Helical Antenna provided along with the module. VCC supplies power for the module. This can be anywhere from 3.4V to 4.4 volts. Remember connecting it to 5V pin will likely destroy your module! It doesn't even run on 3.3 V! An external power source like Li-Po battery or DC-DC buck converters rated 3.7V 2A would work. RST (Reset) is a hard reset pin. If you absolutely got the module in a bad space, pull this pin low for 100ms to perform a hard reset. RxD (Receiver) pin is used for serial communication.

TxD (Transmitter) pin is used for serial communication.
GND is the Ground Pin and needs to be connected to GND pin on the Arduino.
RING pin acts as a Ring Indicator. It is basically the 'interrupt' out pin from the module. It is by default high and will pulse low for 120ms when a call is received. It can also be configured to pulse when an SMS is received.
DTR pin activates/deactivates sleep mode. Pulling it HIGH will put module in sleep mode, disabling serial communication. Pulling it LOW will wake the module up.
MIC± is a differential microphone input. The two microphone pins can be connected directly to these pins.
SPK± is a differential speaker interface. The two pins of a speaker can be tied directly to these two pins.

## 3.14 GPS

The GPS module is a well-performing complete GPS receiver with a built-in 25 x 25 x 4mm ceramic antenna, which provides a strong satellite search capability. With the power and signal indicators, you can monitor the status of the module. Thanks to the data backup battery, the module can save the data when the main power is shut down accidentally. Its 3mm mounting holes can ensure easy assembly on your aircraft, which thus can fly steadily at a fixed position, return to Home automatically, and automatic waypoint flying, etc. Or you can apply it on your smart robot car for automatic returning or heading to a certain destination, making it a real "smart" bot! The schematic diagram of the module is shown as below:[14]

This is a complete GPS module that is based on the GPS. This unit uses the latest technology from Ublox to give the best possible positioning information and includes a larger built-in 25 x 25mm active GPS antenna with a UART TTL socket. A battery is also included so that you

can obtain a GPS lock faster. This is an updated GPS module that can be used with ardupilot mega v2. This GPS module gives the best possible position information, allowing for better performance with your Ardupilot or other Multi rotor control platform.[15]

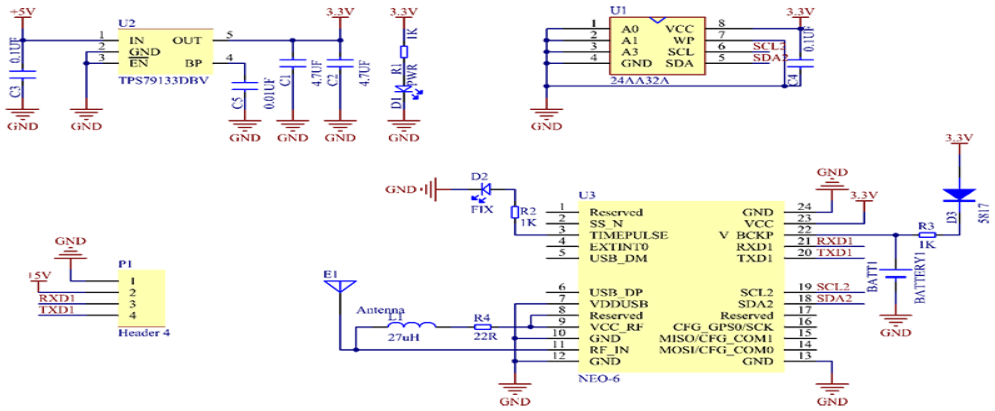


Figure 3.18: Schematic Diagram Of GPS

## Features:

1. 5Hz position update rate
2. Operating temperature range: -40 TO 85°CUART TTL socket
3. EEPROM to save configuration settings
4. Rechargeable battery for Backup
5. The cold start time of 38 s and Hot start time of 1 s
6. Supply voltage: 3.3 V
7. Configurable from 4800 Baud to 115200 Baud rates. (default 9600)
8. Super Sense ® Indoor GPS: -162 dBm tracking sensitivity
9. Support SBAS (WAAS, EGNOS, MSAS, GAGAN)
10. Separated 18X18mm GPS antenna


Figure 3.19: GPS

## 3.15 Buzzer

There are many ways to communicate between the user and a product. One of the best ways is audio communication using a buzzer IC. So during the design process, understanding some technologies with configurations is very helpful. So, this article discusses an overview of an audio signaling device like a beeper or a buzzer and its working with applications.An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.The **pin configuration of the buzzer** is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-'symbol or short terminal and it is connected to the GND terminal.


Figuire 3.20: Buzzer

**History**

The history of an electromechanical buzzer and piezoelectric is discussed below.**Electromechanical** This buzzer was launched in the year 1831 by an American Scientist namely Joseph Henry but, this was used in doorbells until they were eliminated in 1930 in support of musical bells, which had a smooth tone.

**Piezoelectric**

These buzzers were invented by manufacturers of Japanese & fixed into a broad range of devices during the period of 1970s – 1980s. So, this development primarily came due to cooperative efforts through the manufacturing companies of Japanese. In the year 1951, they recognized the Application Research Committee of Barium Titanate that allows the corporations to be cooperative competitively & bring about numerous piezoelectric creations.

**Specifications**

The **specifications of the buzzer** include the following.

- Color is black
- The frequency range is 3,300Hz
- Operating Temperature ranges from – 20° C to +60°C
- Operating voltage ranges from 3V to 24V DC
- The sound pressure level is 85dBA or 10cm
- The supply current is below 15mA

**Types of Buzzer**

A buzzer is available in different types which include the following.

- Piezoelectric
- Electromagnetic
- Mechanical
- Electromechanical
- Magnetic

**Piezoelectric**

As the name suggests, the piezoelectric type uses the piezoelectric ceramic's piezoelectric effect & pulse current to make the metal plate vibrate & generate sound. This kind of buzzer is

made with a resonance box, multi resonator, piezoelectric plate, housing, impedance matcher, etc. Some of the buzzers are also designed with LEDs.

**Mechanical**

These types of buzzers are subtypes of electromagnetic, so the components used in this type are also similar. But the main difference is that the vibrating buzzer is placed on the outside instead of the inside.

**Electromechanical**

The designing of these types of buzzers can be done with a bare metal disc & an electromagnet. The working principle of this is similar to magnetic and electromagnetic. It generates sound throughout the disc movement & magnetism.

**Magnetic**

Like a piezo type, magnetic is also used to generate a sound but they are different due to core functionality. The magnetic type is more fixed as compared to the piezo type because they work through a magnetic field.

Magnetic buzzers utilize an electric charge instead of depending on piezo materials to generate a magnetic field, after that it permits another element of the buzzer to vibrate & generate sound. The applications of magnetic buzzers are similar to the piezo type in household devices, alarms such as watches, clocks & keyboards.

**Working Principle**

The working principle of a buzzer depends on the theory that, once the voltage is given across a piezoelectric material, then a pressure difference is produced. A piezo type includes piezo crystals among two conductors.

Once a potential disparity is given across these crystals, then they thrust one conductor & drag the additional conductor through their internal property. So this continuous action will produce a sharp sound signal.

**Mounting Configurations**

The mounting configurations of buzzers include the following.

- Panel Mount

- Wire Leads
- Screw Terminals
- Through Hole
- Spring Contact
- Surface Mount

**How to use a Buzzer?**

A buzzer is an efficient component to include the features of sound in our system or project. It is an extremely small & solid two-pin device thus it can be simply utilized on breadboard or PCB. So in most applications, this component is widely used.

There are two kinds of buzzers commonly available like simple and readymade. Once a simple type is power-driven then it will generate a beep sound continuously. A readymade type looks heavier & generates a Beep. Beep. Beep. This sound is because of the internal oscillating circuit within it.

This buzzer uses a DC power supply that ranges from 4V – 9V. To operate this, a 9V battery is used but it is suggested to utilize a regulated +5V/+6V DC supply. Generally, it is connected through a switching circuit to switch ON/OFF the buzzer at the necessary time interval.

**Buzzer Circuit Diagram**

The **circuit diagram of the water level indicator using the buzzer** is shown below. This circuit is used to sense or detect the water level within the tank or washing machine or pool, etc. This circuit is very simple to design using few components such as a transistor, buzzer, 300K variable resistor, and power supply or 9V battery.

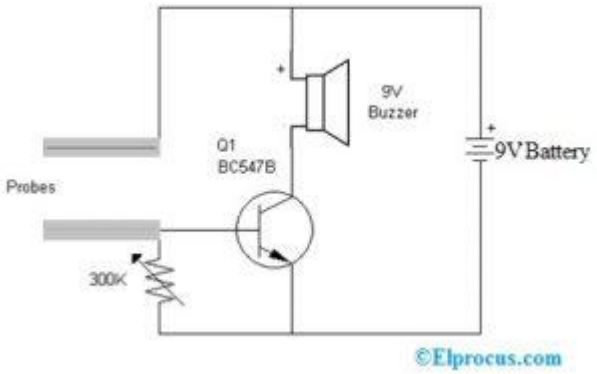


Figure 3.21: Water Level Circuit using Buzzer

Once the two probes of the circuit are placed in the tank, it detects the level of water. Once the water level exceeds the fixed level, then it generates a beep sound through a buzzer connected to the circuit. This circuit uses a BC547B NPN transistor however we can also use any general-purpose transistor instead of using 2N3904/2N2222.

This water level sensor circuit working is very simple and the transistor used within the circuit works as a switch. Once the two probes notice the water level within the tank, then the transistor turns ON & the voltage begins flowing throughout the transistor to trigger the buzzer.

**How to Choose a Buzzer?**

While choosing a buzzer or speaker, many principles need to consider like the following.

- Size of the product
- Consumption of Current
- Type of terminal
- Frequency Voltage
- Volume
- Type
- AC/DC Voltage
- The tone is Continuous/Pulsed
- Fixing – Pins, Leads/Surface Mount
- Output of Sound
- Feedback Option
- Piezo Elements

**Advantages**

The **advantages of a buzzer** include the following.

- Simply Compatible
- Frequency Response is Good
- Size is small
- Energy Consumption is less
- The Range of Voltage usage is Large
- Sound Pressure is high

**Disadvantages**

The **disadvantages of the buzzer** include the following.

- Controlling is a little hard
- Generates Annoying Sound
- Training is necessary to know how to repair the condition without just turning off.

**Applications**

The **applications of the buzzer** include the following.

- Communication Devices
- Electronics used in Automobiles
- Alarm Circuits
- Portable Devices
- Security Systems
- Timers
- Household Appliances
- Electronic Metronomes
- Sporting Events
- Annunciator Panels
- Game Shows

## 3.16 Buck Converter

A **buck converter** (**step-down converter**) is a DC-to-DC power converter which steps down voltage (while drawing less average current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor, although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter). It is called a buck converter because the voltage across the inductor “bucks” or opposes the supply voltage.

Figure 3.22: DC -DC Buck Converter

**DC-DC Buck Converter Step Down Module LM2596 Power Supply** is a step-down(buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version. The LM2596 series operates at a switching frequency of 150kHz, thus allowing smaller sized filter components than what would be required with lower frequency switching regulators.

**Specifications of DC-DC Buck Converter Step Down Module LM2596 Power Supply :**

- Conversion efficiency: 92% (highest)
- Switching frequency: 150KHz
- Output ripple: 30mA9maximum)
- Load Regulation: ± 0.5%
- Voltage Regulation: ± 0.5%
- Dynamic Response speed: 5% 200uS
- Input voltage:4.75-35V
- Output voltage:1.25-26V(Adjustable)
- Output current: Rated current is 2A, maximum 3A (Additional heat sink is required)
- Conversion Efficiency: Up to 92% (output voltage higher, the higher the efficiency)
- Switching Frequency: 150KHz
- Rectifier: Non-Synchronous Rectification
- Module Properties: Non-isolated step-down module (buck)
- Short Circuit Protection: Current limiting, since the recovery
- Operating Temperature: Industrial grade (-40 to +85) (output power 10W or less)

## 3.17 Arduino Software

The digital micro-controller unit named as Arduino Nano can be programmed with the Arduino software IDE. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino Nano from the Tools, Board menu (according to the micro-controller on our board). The IC used named as ATmega328 on the Arduino Nano comes pre burned with a boot loader that allows us to upload new code to it without the use of an external hardware programmer.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the micro-controller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

The Arduino Nano is one of the latest digital micro-controller units and has a number of facilities for communicating with a computer, another Arduino, or other micro-controllers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .in file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial Communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. Arduino programs are written in C or C++ and the program code written for Arduino is called sketch. The Arduino IDE uses the GNU tool chain and AVR Lab to compile programs, and for uploading the programs it uses

argued. As the Arduino platform uses Atmel micro-controllers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino. The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

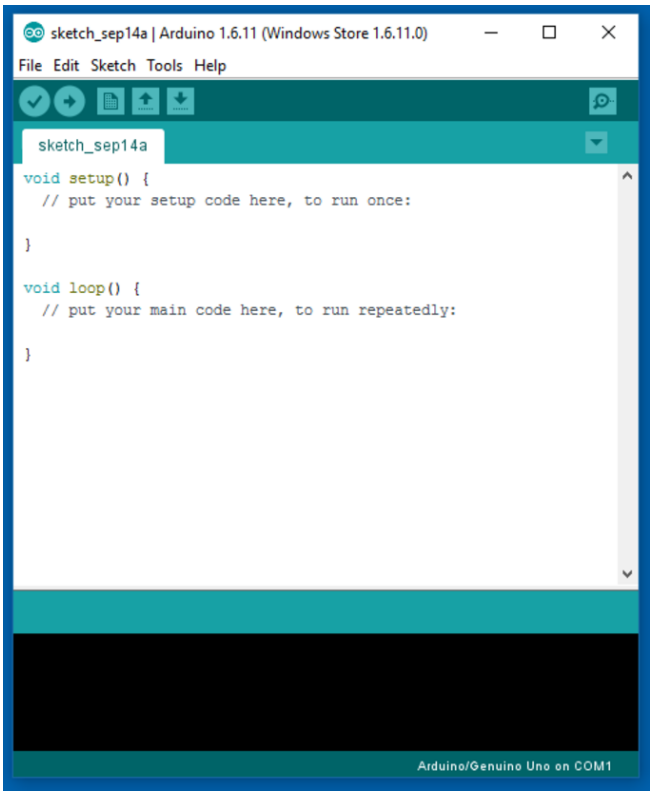


Figure 3.23: Arduino Software Interface IDE

## Writing Sketches

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

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension

.pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

**Verify**

Checks your code for errors compiling it.

**Upload**

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

**New**

Creates a new sketch.

**Open**

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.

**Save**

Saves your sketch.

**Serial Monitor**

Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

**File**

**New**

Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

**Open**

Allows to load a sketch file browsing through the computer drives and folders.

**Open Recent**

Provides a short list of the most recent sketches, ready to be opened.

**Sketchbook**

Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.

**Examples**

Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.

**Close**

Closes the instance of the Arduino Software from which it is clicked.

**Save**

Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.

**Save as...**

Allows to save the current sketch with a different name.

**Page Setup**

It shows the Page Setup window for printing.

**Print**

Sends the current sketch to the printer according to the settings defined in Page Setup.

**Preferences**

Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.

**Quit**

Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

**Edit**

Undo/Redo

Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

**Cut**

Removes the selected text from the editor and places it into the clipboard.

**Copy**

Duplicates the selected text in the editor and places it into the clipboard.

**Copy for Forum**

Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

**Copy as HTML**

Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

**Paste**

Puts the contents of the clipboard at the cursor position, in the editor.

**Select All**

Selects and highlights the whole content of the editor.

**Comment/Uncomment**

Puts or removes the // comment marker at the beginning of each selected line.

**Increase/Decrease Indent**

Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

**Find**

Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

**Find Next**

Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

**Find Previous**

Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

**Sketch**

**Verify/Compile**

Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

**Upload**

Compiles and loads the binary file onto the configured board through the configured Port.

**Upload Using Programmer**

This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.

**Export Compiled Binary**

Saves a .hex file that may be kept as archive or sent to the board using other tools.

**Show Sketch Folder**

Opens the current sketch folder.

**Include Library**

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

**Add File...**

Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

**Tools**

**Auto Format**

This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.

**Archive Sketch**

Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

**Fix Encoding & Reload**

Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

**Serial Monitor**

Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over

serial port opening.

**Board**

Select the board that you're using. See below for descriptions of the various boards.

**Port**

This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

**Programmer**

For selecting a harware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a boot loader to a new micro-controller, you will use this.

**Burn Boot loader**

The items in this menu allow you to burn a boot loader onto the micro-controller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega micro-controller (which normally come without a boot loader). Ensure that you've selected the correct board from the Boards menu before burning the boot loader on the target board. This command also set the right fuses.

**Sketchbook**

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

**Tabs, Multiple Files, and Compilation**

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

**Uploading**

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably

something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino boot loader, a small program that has been loaded on to the micro-controller on your board. It allows you to upload code without using any additional hardware. The boot-loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the micro-controller. The boot-loader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

**Libraries**

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

**Third-Party Hardware**

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, boot-loaders, and programmer definitions. To install, create the hardware

directory, then unzip the third-party platform into its own sub-directory. (Don't use "Arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

**Serial Monitor**

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board. You can also talk to the board from Processing, Flash, MaxMSP, etc. (see the interfacing page for details).

## 3.18 Easy EDA Software

Easy EDA is a web-based EDA tool suite that enables hardware engineers to design, simulate, share - publicly and privately - and discuss schematics, simulations and printed circuit boards. Other features include the creation of a bill of materials, Gerber files and pick and place files and documentary outputs in PDF, PNG and SVG formats. Easy EDA allows the creation and editing of schematic diagrams, SPICE simulation of mixed analogue and digital circuits and the creation and editing of printed circuit board layouts and, optionally, the manufacture of printed circuit boards.

Subscription-free membership is offered for public plus a limited number of private projects. The number of private projects can be increased by contributing high quality public projects, schematic symbols, and PCB footprints and/or by paying a monthly subscription.Registered users can download Gerber files from the tool free of charge; but for a fee, EasyEDA offers a PCB fabrication service. This service is also able to accept Gerber file inputs from third party tools

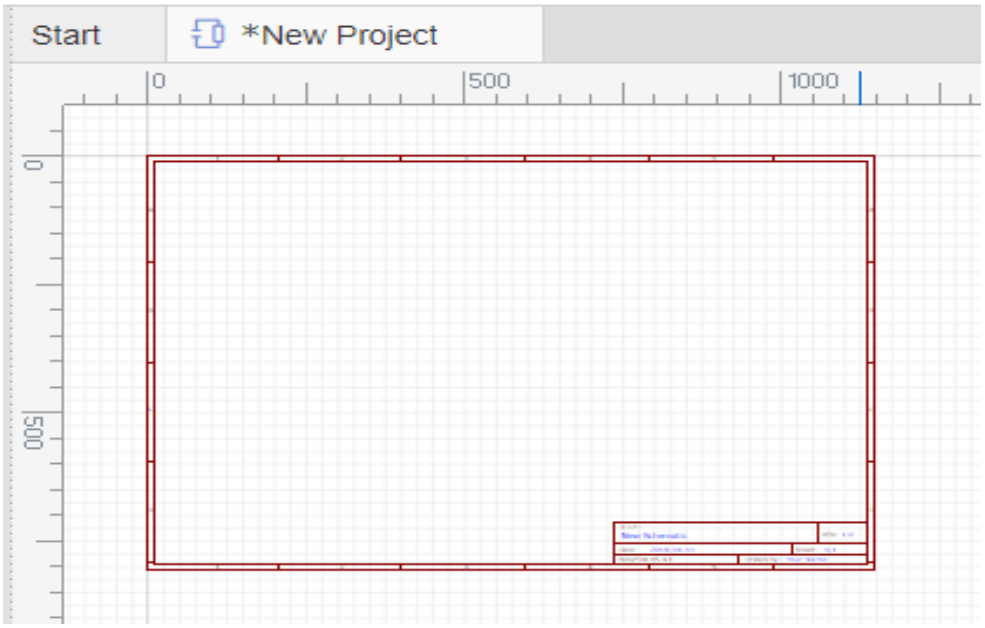


Figure 3.24: Easy EDA Software Interface

# CHAPTER FOUR

# Performance Evaluation

# Performance Evaluation

## 4.1 Introduction

This is the evaluation part. Here we mainly discuss about our system result and discussion.

## 4.2 Results

After making our project we observe it very careful. It works as we desire. Our project give output perfectly and all equipment are work perfectly. We check how much it work and we get perfect output from this project. Finally, we have completed our project successfully & check our project its run accurately according to our objective.

- At first, we start our system, **GSM and GPS Based Smart Stick.**
- Then if there is an obstacle in front, the ultrasonic sensor will sense, and the buzzer will sound and the stick will vibrate with the help of the vibrator motor.
- If the stick is broken, then SMS will go with the help of GSM. And will go to the location by GPS.
- Then the light on the stick will turn on when it gets dark at night.

## 4.3 Advantage

There are certainly many advantages of our project and some of the major ones have been given below:

- Main advantage of this project is safely will be walk street by stick without human help.
- When the blind person face she problem than push button click and her family member will get a SMS she facing problem and easily find out her position by GPS.
- We can easily make this project and cost is very low.
- This is one of the GPS projects for blind which is very easy to install and easy to use. Partially sighted person can be immediately tracked using the projects for blind people.

## 4.4 Application

Our project has many application areas and actually we need to use it in many places to verify the exact person which have the proper access. Some of the application areas of the project has been pointed out below: There are numerous Applications of GPS based tracker for Blind individual utilizing GSM innovation, not many of them are recorded underneath:

- 1) This undertaking can be utilized for Blind individual.
- 2) GPS for daze individual undertaking can likewise be utilized for following outwardly hindered individuals, senior residents, old individual in our home.
- 3) This project can also be used for tracking senior citizens, elderly person in our home.

## 4.5 Limitations

Although our project has many applications and advantages but there are some limitations of the project as well and the good thing is that these limitations are minor and doesn't affect the efficiency of the system. Limitations are given below:

- The working procedure of this project is very easy, but we are facing some limitation for doing this project. Such as coding problem, program writing, connecting to PCB board, commend following etc.

## 4.6 Future Work

We are thinking about adding many features to our project in the future to get more desirable outcomes. Some of the steps that we are thinking about taking are given below:

- IOT and Wi-Fi Based stick.
- Ai Based Smart Stick.
- Ai Smart Object Based Stick.
- Every hour automatic notification system.
- Arduino nano replace to raspberry pi use.

## 4.7 Discussion

The project aims safely walk blind people on the street. When the blind people walk the street by help use stick. From some limit distance of object than automatic create buzzer alarm than he understands front side is wall, tree, Poll etc. Blind people move left or right side lessening

no buzzer than safely walk the road. When the blind people face any problem than emergency button push, and automatic send SMS get her family and position tracking by GPS module.

# CHAPTER FIVE

# Conclusion

# Conclusion

## 5.1 Conclusion

The project is successfully developed and met the stated objectives. If we implement very successfully reduce road accident for blind people, Life easier and comfortable. These systems provide the blind people safety and security. In this project, we develop a general purpose of safely walk of street by stick and without human help.

This paper presents the implementation of a smart stick that assists a visually impaired person to his destination safe and secure. We make use of various sensors to detect the obstacles ahead and warn the blind person about the obstacle through beep sound. The intensity of the beep sound increases as the person nears the obstacle which aid him to move aside of the obstacle. We take the benefits of GPS module and GSM/GPRS module, where GPS module helps to trace the blind person using the data collected by it. In case of dangerous circumstances, the person whose phone number has been saved is notified that the blind person is at risk, along with the current location of the blind person. The smart stick also facilitates the blind person to make calls at times of emergency. All these features are beneficial in lending a hand to make the visually impaired people become self-reliant while navigating

# References


[1] Amit Kumar, Rusha Patra, M. Manjunatha, J. Mukhopadhyay and A. K. Majumdar An electronic travel aid for navigation of visually impaired Communication Systems and Networks (COMSNETS), 2011 Third International conference on 4-8 jan 2011.

[2] Shamsi, M.A.; Al-Qutayri, M.; Jeedella, J.; Blind assistant navigation system Biomedical Engineering (MECBME), 2011 1st Middle East Conference on 2124 Feb. 2011

[3] Michel Mouly and Marie-Bernadette Pautet: GSM System for Mobile Communications, published by the authors 1992, ISBN 2-9507190-0-7

[3] Majid Al Shamsi, Mahmoud Al-Qutayri, and Jeedella, "Blind Assistant Navigation System" in IEEE Transactions, March 2011.

[4] Smart walking stick - an electronic approach to assist visually disabled persons Mohammad Hazzaz Mahmud, Rana Saha, Sayemul Islam

[5] Mohd Helmy Abd Wahab, Amirul A. Talib, Herdawatie A. Kadir, Ayob Johari, A. Noraziah, Roslina M. Sidek, Ariffin A. "Smart cane: assistive cane for visually impaired people", IJCSI, Vol.8 Issue 4, July 2011.

[6] M. Bousbia-Salah, A. Larbi, and M. Bedda, "An approach for the measurement of distance travelled by blind and visually impaired people," in Proc. 10th IEEE International Conference on Electronics, Circuits and Systems, Sharjah, United Arab Emirates, pp. 1312- 1315, 2003.

[7] Hashino, S.; Ghurchian, R.; A blind guidance system for street crossings based on ultrasonic sensors. Information and Automation (ICIA), 2010 IEEE International Conference on June 2010

[8] David Castells, Joao M.F. Rodrigues, J.M. Hans du Buf "Obstacle detection and avoidance on sidewalks" In Proc. Int. Conf. on Computer Vision-Theory and

Applications, Vol. 2, pp. 235-240, 201

[9] Shruti Dambhare M.E 3rd SEM (ESC) G.H.R.C.E. Nagpur, Prof. A. Sakhare M.Tech (ESC) G.H.R.C.E. Nagpur Smart stick for Blind: Obstacle Detection, Artificial vision and Real-time assistance via GPS.

# Appendix

**Programming-Code:**

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <TinyGPS++.h>

SoftwareSerial SIM900(7, 8);
SoftwareSerial gpsSerial(9, 10);
String textMessage;
char msg;
char phone_no[] = "+01745018953"; // Include the country code and remove
leading zero
String message;
char str[15];
int stick = 6;

float latitude = 0.0;
float longitude = 0.0;
bool getGPS(TinyGPSPlus &gps);

TinyGPSPlus gps;

void setup() {


  Serial.begin(9600);
    pinMode(stick, INPUT);
SIM900.begin(9600);
delay(200);
 SIM900.print("AT+CMGF=1\r");
  delay(100);
  SIM900.print("AT+CNMI=2,2,0,0,0\r");
  delay(100);
SIM900.println("AT+CNMI=2,2,0,0,0");// AT Command to receive a live SMS
Serial.println("AT+CNMI=2,2,0,0,0");
Serial.print("SIM900 ready...");
   gpsSerial.begin(9600);

}


void loop() {
  int broke = digitalRead(stick);
   if (SIM900.available() > 0) {
    textMessage = SIM900.readString();
    Serial.print(textMessage);
```

```
    delay(10);
  }

  if (textMessage.indexOf("Location") >= 0) {
    engine_off();
    textMessage = "";
  }
   if (broke == 0) {
    sendSMS("Stick has been broken!!!!! Time For Repair $$$$");
    delay(5000);
  }
 while(1)
  {
   while (gpsSerial.available() > 0)
   { gps.encode(gpsSerial.read()); }

      if (gps.location.isUpdated())
      {
       Serial.print("LAT=");  Serial.println(gps.location.lat(), 6);
       Serial.print("LONG="); Serial.println(gps.location.lng(), 6);
       latitude=gps.location.lat();
       longitude=gps.location.lng();
      break;
      }
}
delay(500);
}


void engine_off() {
  SIM900.print("AT+CMGF=1\r");
  delay(20);
  SIM900.print("AT+CMGS=\"");
  SIM900.print(phone_no);
  SIM900.println("\"");
  delay(20);
  String gpsLink = "https://maps.google.com/?q=" + String(latitude) + "," +
String(longitude);
  SIM900.println(gpsLink);
  delay(20);
  SIM900.write(0x1A);
  delay(20);

  SIM900.println();
}
void sendSMS(String message) {
  SIM900.println("AT+CMGF=1"); // Set SMS mode to text
  delay(100);
  SIM900.print("AT+CMGS=\"+01745018953\""); // Replace with the recipient's
phone number
```

```
  SIM900.write(0x0D); // Enter
  delay(100);
  SIM900.println(message);

  if (latitude != 0.0 && longitude != 0.0) {
    String gpsLink = "https://maps.google.com/?q=" + String(latitude) + "," +
String(longitude);
    SIM900.println(gpsLink);
  } else {
    SIM900.println("No GPS data available");
  }

  delay(100);
  SIM900.write(0x1A);
  delay(100);
}

//void address() {
//  while (gpsSerial.available() > 0) {
//    if (gps.encode(gpsSerial.read())) {
//      if (gps.location.isUpdated()) {
//        latitude = gps.location.lat();
//        longitude = gps.location.lng();
//        Serial.print("LAT=");
//        Serial.println(latitude, 6);
//        Serial.print("LONG=");
//        Serial.println(longitude, 6);
//      }
//    }
//  }
//}

#define trigPin 2
#define echoPin 3
long duration;
long distance;

int vibrator = 5;
int buzzer = 13;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(vibrator, OUTPUT);

  digitalWrite(buzzer, LOW);
  digitalWrite(vibrator, LOW);
}
```

```
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

   Serial.print("distance ");
  Serial.println(distance);
  delay(200);
  if (distance < 70) {
    digitalWrite(vibrator, HIGH);
    digitalWrite(buzzer, HIGH);
    delay(500);
    digitalWrite(vibrator, LOW);
    digitalWrite(buzzer, LOW);
    delay(500);
    digitalWrite(buzzer, HIGH);
  } else {
    digitalWrite(buzzer, LOW);
    digitalWrite(vibrator, LOW);
  }
}
```