Project work:

SMART BLIND STICK

Table of Content

ABSTRACT

The life of visually impaired people is very different. They face many problems in moving from one place to another. Since they cannot see, they often get hit by objects in roads like poles, walls, cars, people etc. and they get severely injured. They face humiliation and lose confidence in themselves. There are chances that they can get lost. In such cases, it is very difficult for their family members to find them. Even though they are provided with stick, in which they can identify the object by tapping it. It is not much helpful for them in order to avoid obstacles. We come with this Smart Stick for visually impaired people in which they can detect the object from a further distance and they could avoid it using ultrasonic sensors, and if they are lost, using GPRS and GSM modules their family members can track them easily.

1. INTRODUCTION

1.1 Objective

The main objective is to help the visually impaired people to detect obstacles from a farther distance and to track the location of the visually impaired person when an unexpected situation occurs. The technology used in this project is based on IoT integrated with GPS and GSM. It also aims at reducing the cost to provide optimal solution to the visually impaired.

1.2 Problem Overview

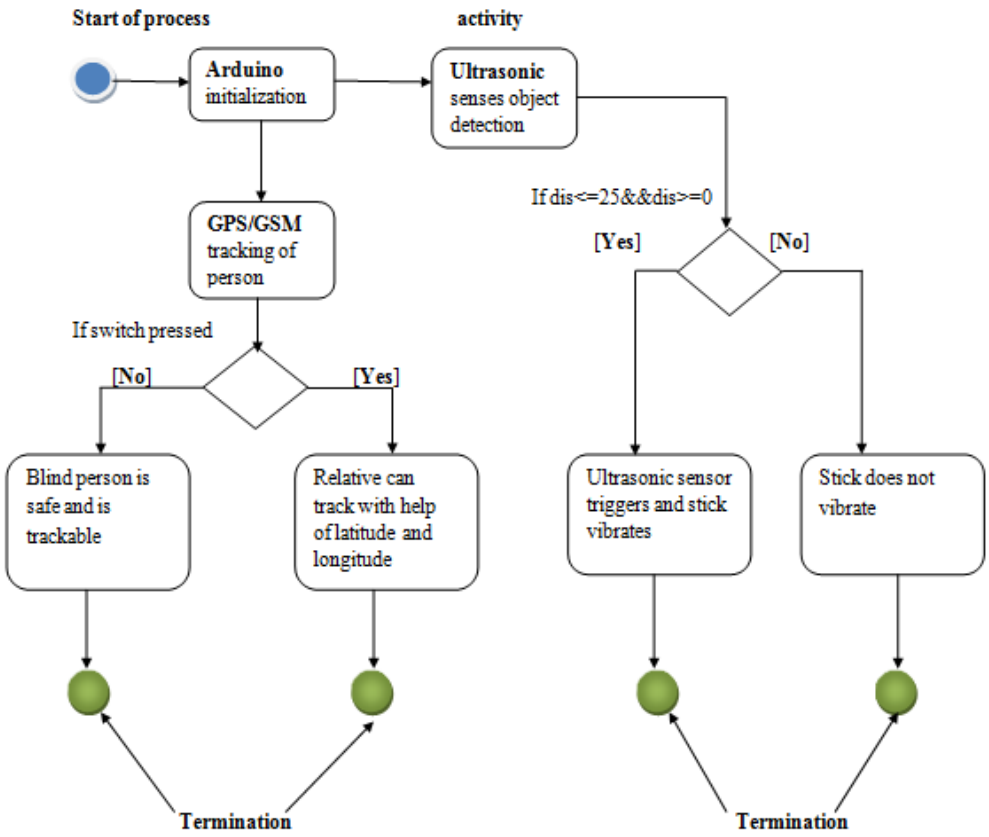
The world of visually impaired people is very different compared to us (who can see normally). They face many difficulties in their life. Out of many problems moving around in the streets, unknown places are the most difficult task for them.

According to WHO 285 million people are estimated to be visually impaired worldwide 39 million are blind and 246 have low vision. About 90% of the worlds visually impaired live in low-income settings. About 65 % of all people who are visually impaired are aged 50 and older, while this age group comprises about 20 % of the world's population.

The visually impaired people find it difficult to move in this world. They may even get lost. In such cases, it is very difficult for their family members to find them. Even though they are provided with stick, using which they can identify the object by tapping it. It is not much helpful for them in order to avoid obstacles.

We came up with this Smart Stick for Blind People in which visually impaired person can be able to detect the object and they could avoid it using ultra sonic sensors. Moreover, if they are lost using GPRS andGSM modules we can track them

1.3 Activity Diagram



The following **Figure 1.1** depicts the various activities of the scene. The process starts with the Initialization of Arduino Uno board. The activity of object detection takes place where if the obstacle is present within the distance mentioned the stick vibrates so that the visually impaired person can avoid the obstacle. If sensor does not sense any object the the stick does not vibrate. The other part of the diagram describes how the activity is performed if the visually impaired person is lost. He/She presses the switch mounted on the stick and the location of the person is sent with a link through a message to the relative's phone.

2. DESIGN MODEL

2.1 Internet of Things

The following **Figure 2.1.1** explains the Internet of Things.

Every object in our vicinity are all wirelessly connected to a network of communication and they interact using an embedded technology.

All these are continuously conveying data about them t the internet, where people can analyze the data and perform actions based on it. Using IoT we can provide a Digital Nervous System to the world.

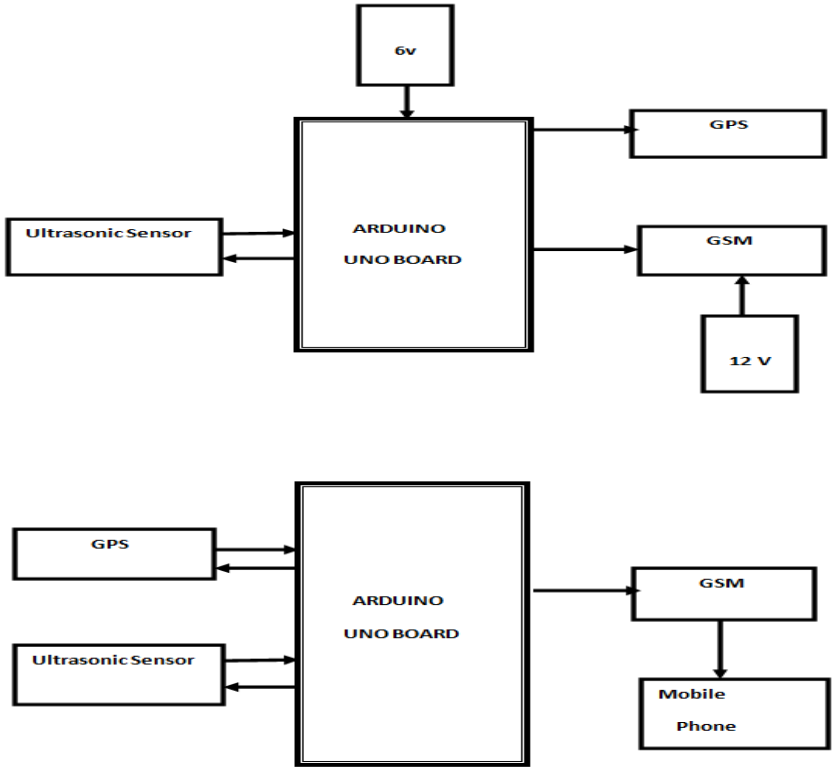
Connect with Things

1) Point a camera to a packet of food to learn more about it like its ingredients, manufacturing, Nutritional information etc.
2) Point a camera to a person to find out his/her interest or any information they want to share like name, internet, hobbies etc.

Applications of IoT can include:

1) Smart Home [Biggest application of IoT].
2) Smart Farming.

2.2 Hardware Diagram



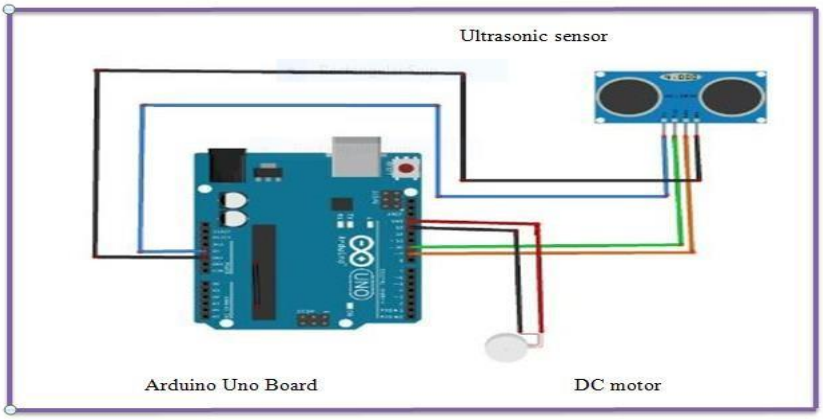
The above block diagram **2.2.1** and **2.2.2** gives a description about the working of the project where an external supply is given to Arduino Uno board and the code is uploaded to the board in order to perform the necessary functions.

The Ultrasonic Sensor acts as the input which is connected to the Arduino Uno board and the output is achieved through the DC motor when the stick vibrates.

The GPS unit acts as the input and the current location of the visually impaired person is shared to the relative so that they can track him/her.

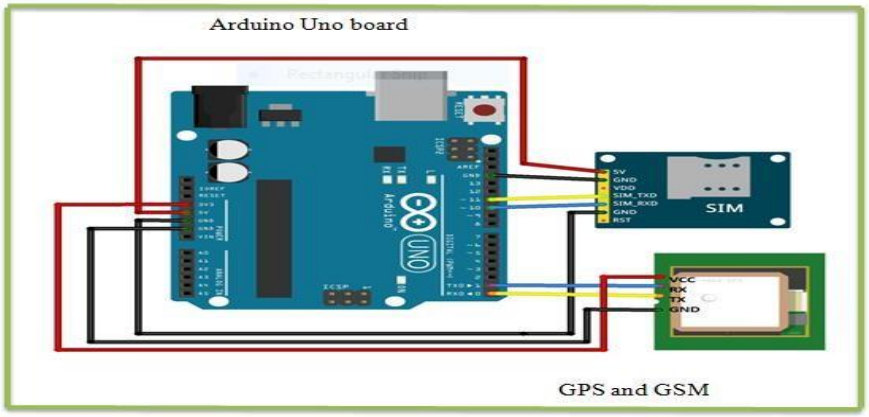
2.3 Circuit Diagram

❖ Ultrasonic Sensor



The above **figure 2.3.1** shows the circuit diagram of the implemented system. An Arduino UNO is the base of this circuit and all the other components are connected to this board. One ultrasonic Sensor is connected to pins of Arduino UNO. A DC motor is also connected at pin 3.When the first sensor detects the obstacle coming, it sends a high signal to pin 10. As soon as the Arduino UNO detects a high signal, it raises the signal at pin 9 and the components connected to this pin shows an output i.e. the DC motor vibrates.

❖ GPS and GSM



The above **figure 2.3.2** shows a proposed system where 5 volts is given as input to the arduino, 10 and 11 pin of Uno board is the input and RX is the input to GPRS module and TX is the output. The 13 pin of Uno is the ground pin connected to GSM and GPRS ground. A switch is connected to the bread board which when pressed sends the link and the latitude and longitude of the visually impaired person to his/her relative.

2.4 Hardware Requirements

The below components are been used in this project in order to fulfill the various needs based on these components a module is constructed and is tested for various strategies.

- Arduino Uno.
- GSM Module.
- GPS Module.
- 10 K POA Mini breadboard.
- 9 volt battery.
- 9 volt battery connector.
- DC male power jack.
- Some Jumper wires.
- Toggle switch.
- Stick.
- 10 K ohm resistor.
- Eccentric Rotating Mass (ERM) vibration motor.

2.4.1 Arduino Uno Board

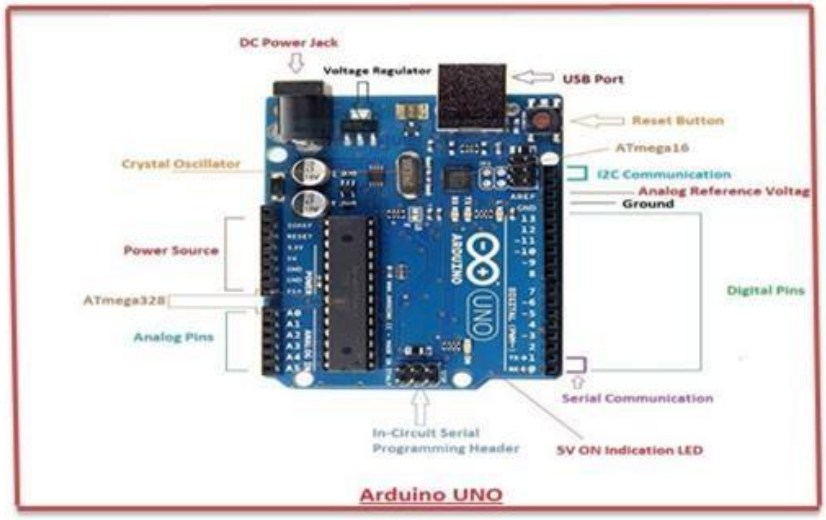


Fig 2.4.1 Arduino Uno Board

Arduino UNO is an open source tool which is readily available. The main advantage of using this controller is that it is very easy to implement as it follows the object oriented programming paradigm for the implementation of the code. Any special functionality which can be easily shared are interfaced by importing the required library files.

It has 14 digital pins and 6 analogue pins which can be used to input or output the data. This controller remains as the heart of our system as monitors the complete working of our model. The proposed system uses the analogue pins for input from the ultrasonic sensor and the digital pins to output the light signals, alarm sounds and the IDE for this controller is open source software.

2.4.2 GPS

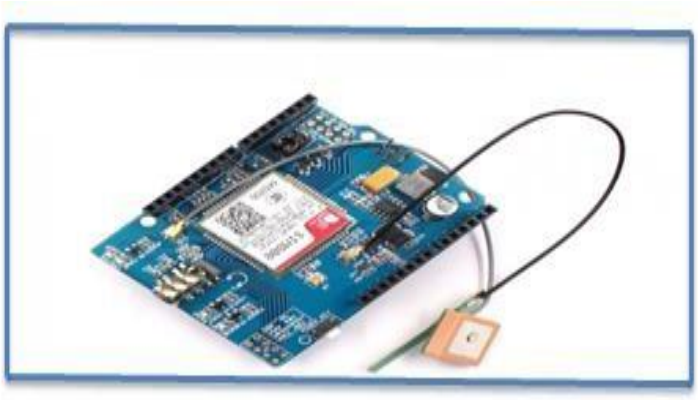


Fig 2.4.2 GPS

The Global Positioning System (GPS) is a global navigation satellite system that provides relocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The GPS system operates independently of any telephonic or internet reception, though these technologies can enhance the usefulness of the GPS positioning information. The GPS system provides critical positioning capabilities to military, civil, and commercial users around the world. The GPS concept is based on time and the known position of specialized satellites.

The satellites carry very stable atomic clocks that are synchronized with one another and to ground clocks. Any drift from true time maintained on the ground is corrected daily. Likewise, the satellite locations are known with great precision. GPS receivers have clocks as well; however, they are usually not synchronized with true time, and are less stable. GPS satellites continuously transmit their current time and position.

A GPS receiver monitors multiple satellites and solves equations to determine the precise position of the receiver and its deviation from true time. At a minimum, four satellites must be in view of the receiver fit to compute four unknown quantities (three position coordinates and clock deviation from satellite time).

2.4.3 GSM

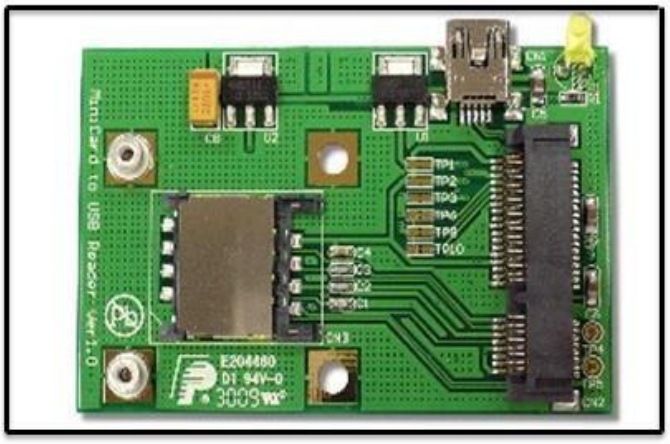


Fig 2.4.3 GSM

The Global Positioning System (GPS) is a global navigation satellite system that provides geo location and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The GPS system operates independently of any telephonic or internet reception, though these technologies can enhance the usefulness of the GPS positioning information.

The GPS system provides critical positioning capabilities to military, civil, and commercial users around the world. The GPS concept is based on time and the known position of specialized satellites. The satellites carry very stable atomic clocks that are synchronized with one another and to ground clocks. Any drift from true time maintained on the ground is corrected daily. Likewise, the satellite locations are known with great precision. GPS receivers have clocks as well; however, they are usually not synchronized with true time, and are less stable. GPS satellites continuously transmit their current time and position.

A GPS receiver monitors multiple satellites and solves equations to determine the precise position of the receiver and its deviation from true time. At a minimum, four satellites must be in view of the receiver for it to compute four unknown quantities (three position coordinates and clock deviation from satellite time).

2.4.4 Ultrasonic Sensors

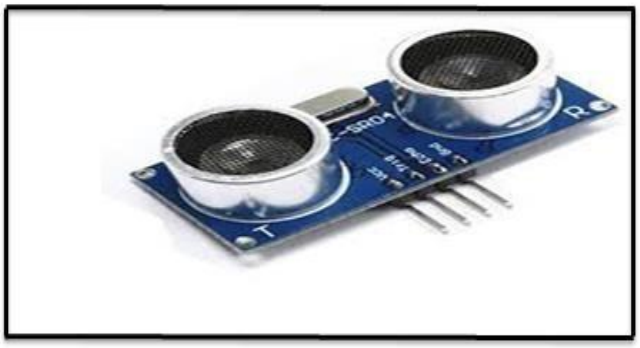


Fig 2.4.4 Ultrasonic Sensor

In industrial applications, inaudible sensors are characterized by their dependableness and outstanding skillfulness. Inaudible sensors is accustomed solve even the foremost complicated tasks involving object detection or level activity with millimeter exactness, as a result of their activity technique works depend on underneath conditions. No different activity technique is with success place to use on such a large scale and in such a big amount of completely different applications.

The devices are very strong, creating them appropriate for even the toughest conditions. The detector surface cleans itself through vibration, which is not the sole reason why the detector is insensitive to dirt. The physical Principle the propagation of sound works, with many exceptions, in much any atmosphere. The activity technique used by inaudible sensors has been viewed as Associate in Nursing to a fault complicated technology, and solely used as a "last resort" as an answer for significantly tough applications. Those times have long ago passed. Ultrasonic sensors have tried their dependableness and endurance in nearly all industrial sectors.

These sectors include:

- Mechanical engineering\machine tool.
- Food and drink.
- Woodworking and furnishings.
- Building materials.

2.4.5 10 K POA Mini breadboard

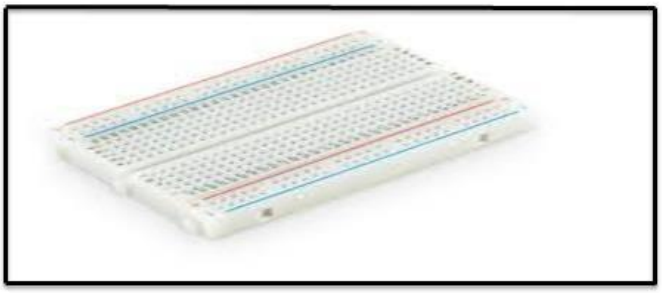


Fig 2.4.5 10K Bread Board

A bread board could be a construction base for prototyping of electronics. Originally, it absolutely was virtually a bread board, a sophisticated piece of wood used for slicing bread within the Nineteen Seventies the solder less bread board (AKA exchange, a terminal array board) became offered and today the term "breadboard" is often accustomed talk over with these.

2.4.6 Eccentric Rotating Mass (ERM) vibration motor

Fig 2.4.6 ERM motor

The Eccentric Rotating Mass vibration motor, or ERM, also known as a pager motor is a DC motor with an offset (non-symmetric) mass attached to the shaft. As the ERM rotates, the centripetal force of the offset mass is asymmetric, resulting in a net centrifugal force, and this causes a displacement of the motor. With a high number of revolutions per minute, the

motor is constantly being displaced and moved by these asymmetric forces. It is this repeated displacement that is perceived as a vibration.

Many mechanical engineering textbooks discuss the characteristics of ERMs, as a 'rotating unbalance', and do so in a negative context. Often engineers are trying to minimize the source of vibration from rotating machinery, because it generates noise and causes excessive machine wear and fatigue. As a result, there is little literature on the theory of maximizing the amplitude of vibration in applications.

2.4.7 Jumper Wires

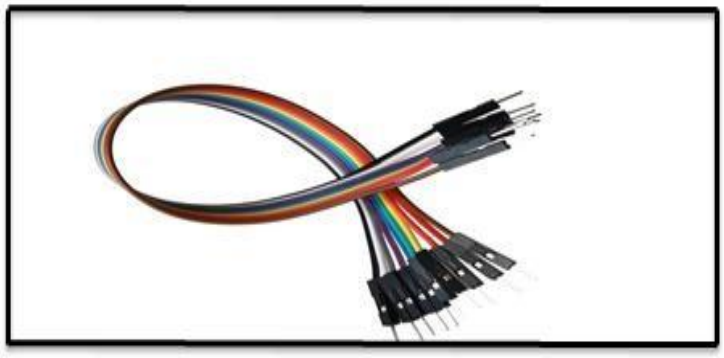


Fig 2.4.7 Jumper Wires

A **jumper wire** (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

2.5 Software Requirement

2.5.1 Arduino Software

Fig 2.5.1 Arduino Software

Arduino UNO is an ASCII text file component and software Package Company, project and user community that styles and manufactures microcontroller-based kits for building digital devices and interactive objects that may sense and management objects within the physical world. The project is predicated on microcontroller board styles, factory-made by many vendors, victimize numerous microcontrollers. These systems give sets of digital and analog I/O pins that may be interfaced to numerous enlargement boards ("shields") and different circuits. The boards feature serial communications interfaces, as well as USB on some models, for loading programs from personal computers.

For programming the microcontrollers, the Arduino project provides an Integrated Development Environment (IDE) supported the process project, which has support for the C++ programming languages. The first Arduino was introduced in 2005, reaching to give a reasonable and straight forward method for novices and professionals to make devices that move with their setting victimization sensors and actuators. Common samples of such devices supposed for beginner hobby is to embrace easy robots, thermostats, and motion detectors.

3. IMPLEMENTATION

The below coding for the detection of obstacle and the tracking of the location is been implemented in the project using coding in C language.

3.1Ultrasonic Sensor Code for Object detection

```
Int const trigPin=10;                          // trigger pin of sensor to 10 pin of arduino

Int const echoPin=9;                           //echo pin of sensor to 9 pin of arduino

Int const VibPin=13;                           //positive of dc motor to pin 13 of arduino

long duration;

int distance;

void setup()

{

pinMode(trigPin,OUTPUT);                       //trigger pin  senses gives the output to dc motor

pinMode(echoPin,INPUT);                        //echo pin is input to arduino

Serial.begin(9600);

pinMode(VibPin,OUTPUT);                        //the dc motor vibrates to give output

}

void loop()

{

digitalWrite(trigPin,LOW);

delayMicroseconds(2);

digitalWrite(trigPin,HIGH);
```

```
delayMicroseconds(10);
digitalWrite(trigPin,LOW);
duration=pulseIn(echoPin,HIGH);
distance=(duration/2)/29.1;
if(distance<=25&&distance>=0)                    //condition to check obstacle detection
{
digitalWrite(VibPin,HIGH);
}
else
{
digitalWrite(VibPin,LOW);
}
Serial.println(distance);                        //printing distance on serial monitor
}
```

3.2 GPS and GSM tracking

```
#include <TinyGPS.h>
#include <SoftwareSerial.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
SoftwareSerial gsmSerial(6,7);//rx,tx
//#include <LiquidCrystal.h>
//SoftwareSerial gsmSerial(3,4);//rx,tx
//LiquidCrystal lcd( 4, 5, 6, 7, 8, 9 );
// Create an instance of the TinyGPS object
int S1=8;
int S2=9;
int S3=10;
int S4=11;
char SMS[180];
 TinyGPS gps;
void getgps(TinyGPS &gps);
//#define GsmPutSSerial(string) gsmSerial.write( string )
//#define GsmGetCSerial( )  get_char()
//#define GsmPutCSerial(chr) gsmSerial.write(chr)
//
//#define SMS_DATA_LEN_MAX 180
//
//typedef struct {
//  unsigned char index[3];
//  unsigned char phone_num[15] ;
```

```
//  unsigned char dat[SMS_DATA_LEN_MAX + 1];
//
//} sms_data;
//
//sms_data sms_dat;
//char SMS_BUFF[130], PHONE_NUMBER[15];
char * dtostrf(
  double __val,
  signed char __width,
  unsigned char __prec,
  char * __s);
//void intGsm( void );
//char get_char(void);
//char deleteSms(char *index);
//void recivecontr(void);
//void recvResponse(unsigned char *response);
//char sendSMS(char *num, char * str );
const char lat_buff[100];
const char long_buff[100];
void setup()
{
```

```
//gsmSerial.begin(9600);
Serial.begin(9600);
gsmSerial.begin(9600);
pinMode(S1, INPUT_PULLUP);
pinMode(S2, INPUT_PULLUP);
pinMode(S3, INPUT_PULLUP);
pinMode(S4, INPUT_PULLUP);
digitalWrite(S1,HIGH);
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
digitalWrite(S4,HIGH);
//lcd.begin(16, 2);
 Serial.print("GPS TRACKING SYSTEM\r\n");


   intGsm("8861541701","PERSON GPS TRACKING SYSTEM..");
   intGsm("8861541701","GPS TRACKING SYSTEM...");
    delay(1000);
   Serial.print("sms sent");
}
void getgps(TinyGPS &gps)
// The getgps function will display the required data on the LCD
```

```
{
float latitude, longitude;
//decode and display position data
gps.f_get_position(&latitude, &longitude);
//lcd.setCursor(0,0);
Serial.print("Lat:");
//lcd.print("Lat:");
Serial.println(latitude,6);
//lat_buff=latitude;
sprintf(lat_buff, "LAT:%f", latitude,6);
//dtostrf(
//lcd.print(latitude,5);
//Serial.println(lat_buff);
//lcd.print(" ");
//lcd.setCursor(0,1);
Serial.print("Long:");
//lcd.print("Long:");
//Serial.print(floatToString());
Serial.println(longitude,6);
//long_buff=longitude;
sprintf(long_buff, "%f", longitude,6);
```

```
//dtostrf(long_buff,8,5,longitude);

//float latitude = -23.123456;

//float longitude = 135.123456;

String one = "PERSON LOCATION AT:https://www.google.com/maps/?q=";

String two = "," ;

String message = one +latitude +two + longitude;

// Convert String to char array

int str_len = message.length() + 1;

char textmessage[str_len];

message.toCharArray(textmessage,str_len);

Serial.println(textmessage);

delay(3000); // wait for 3 seconds

 if(digitalRead(S1)==LOW)

   {

Serial.println("PERSON PRESS THE SWITCH");

 intGsm("8861541701",textmessage);                                                    //
intGsm("9431309623","LA12.922826LG77.501867#S1@");

 delay(1000);

 }

}

void intGsm( char *num1, char * str1 )
```

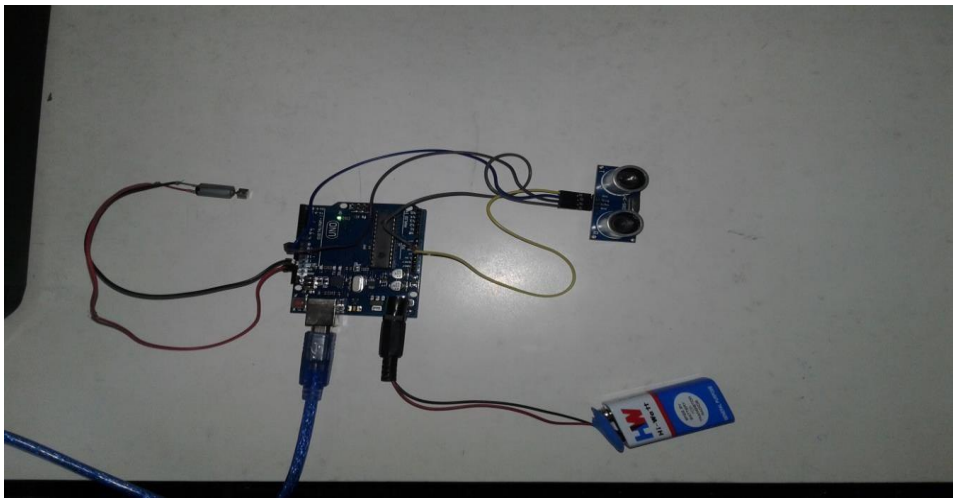
```
{
  char buff[10],i=0;
//while(i<3)
//{
  gsmSerial.write('A');
  delay(100);
  gsmSerial.write('T');
  delay(100);
  gsmSerial.write('E');
  delay(100);
  gsmSerial.write('0');
  delay(100);
  gsmSerial.write('\r');
//i++;
 // recvResponse(buff);
 //Serial.print("AT sent");
  //*****************************************
  gsmSerial.write("AT+CMGF=1\r"); //Initialize GSM For mobile
   delay(2000);
   //Serial.print("cmgf sent");
 //  sendSMS("9071295134","hiiiii");
```

```
//  recvResponse(buff);
delay(2000);
//Serial.print("ATcmgf sent");
gsmSerial.write("AT+CMGS=\"");
 delay(2000);
 gsmSerial.write(num1);
 delay(2000);
 gsmSerial.write("\"\r");
 // gsmSerial.write("AT+CNMI=0,0,0,0,0\r"); //Disabling unsolicited sms indication.
//  recvResponse(buff);
  //recvResponse(buff);
gsmSerial.write(str1);
 delay(2000);
gsmSerial.write(26);
 delay(2000);
 Serial.print("sms sent");
//}
}
void loop()
{
byte a;
```

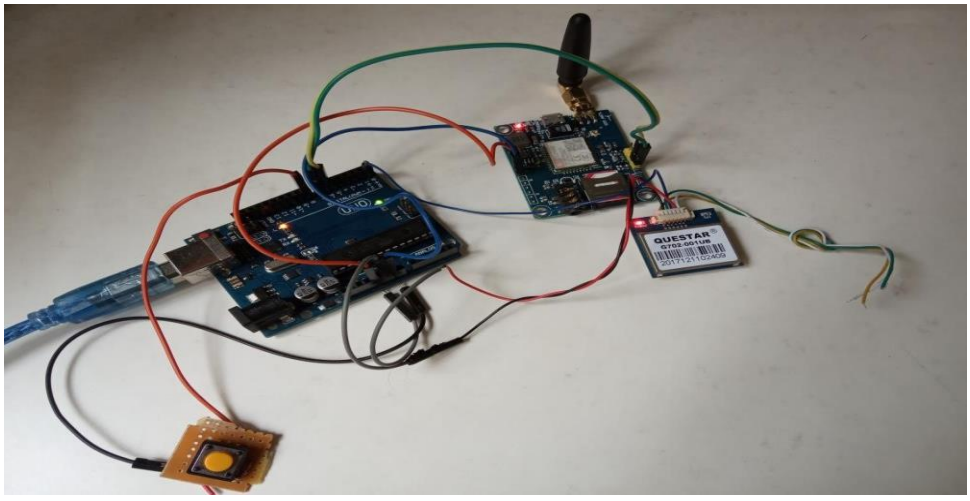
```
if ( Serial.available() > 0 ) // if there is data coming into the serial line

{

a = Serial.read(); // get the byte of data

if(gps.encode(a)) // if there is valid GPS data...

{

getgps(gps); // grab the data and display it on the LCD

}

}

}
```

4.OUTPUTS

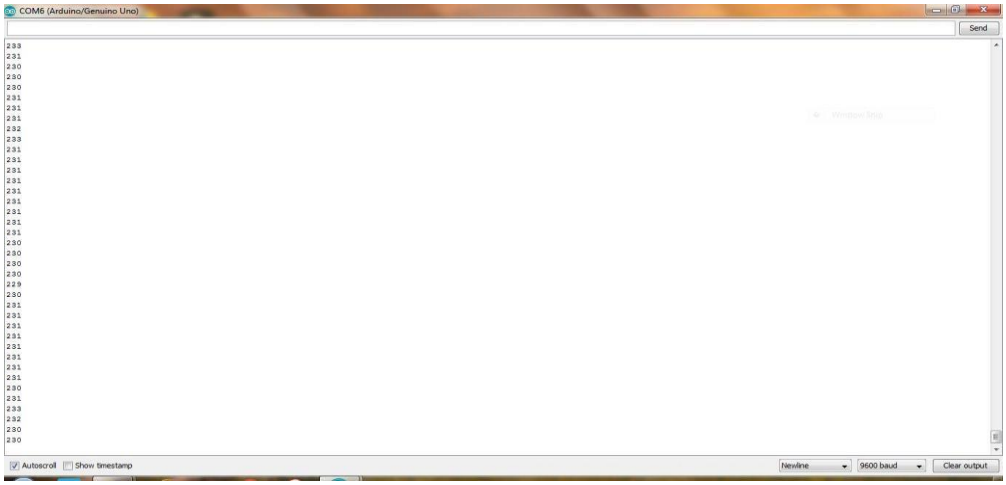
4.1Ultrasonic Sensor



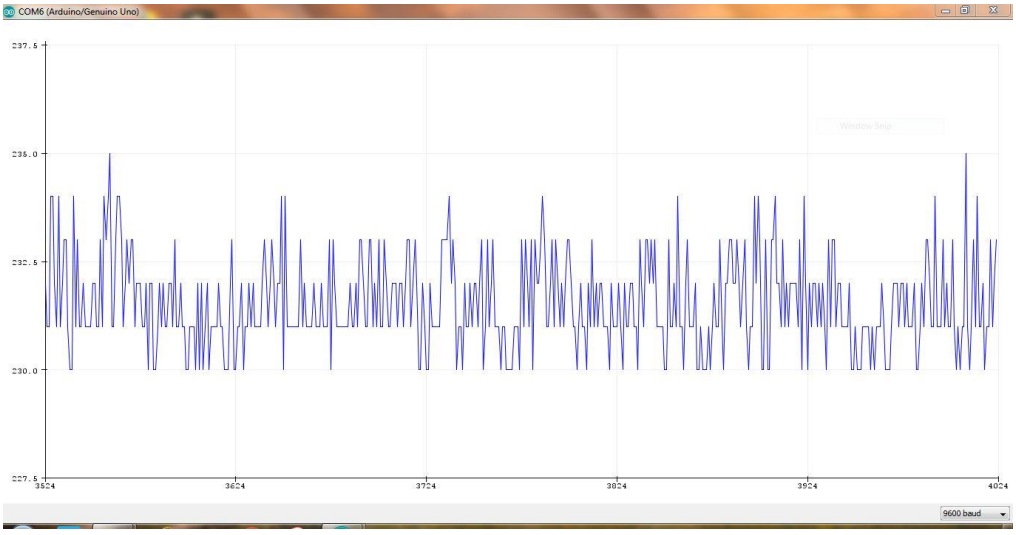
4.2 GPS and GSM



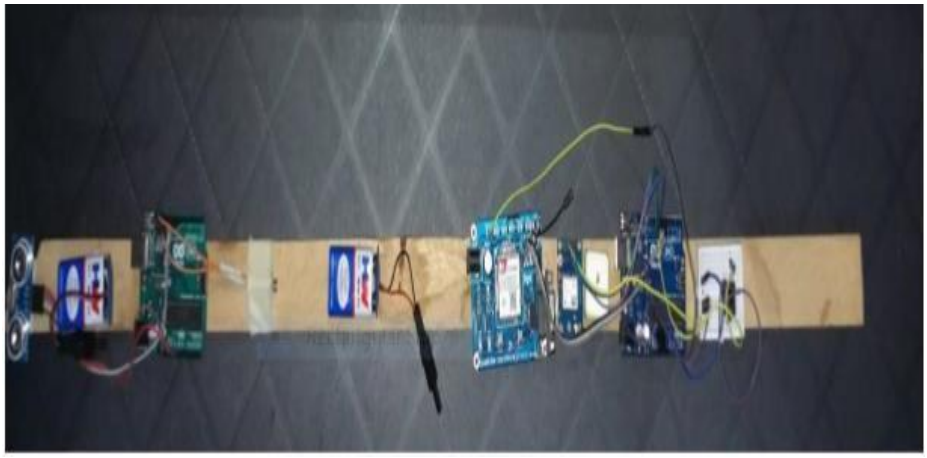
4.3 Ultrasonic Serial Monitor values



4.4 Ultrasonic Sensor Graph



4.5 Smart Blind Stick



5.S0FTWARE TESTING

5.1 Introduction

Software testing is the major process that provides the quality assurance of the software. It also ensures the reliability of the software. It is the process of executing test cases with intent of exposing errors. The increasing availability of the software as a system element and the increasing costs associated with software failure are the main motives behind the well planned thorough testing.

Different types of tests are:

Unit testing.

Integration testing.

Validation testing.

System testing.

Acceptance testing.

5.2 Testing Strategies

The following are the main objectives of testing:

Testing is the process of executing the program with the intent of finding errors. A good testing is one that has a high probability of finding errors that are undiscovered yet. A successful test is one that uncovers an as-yet undiscovered error.

These objectives imply a dramatic change in viewpoint. They move counter to the commonly held view that a successful test is one in which no errors are found. The objective here is to design tests that systematically uncover different classes of errors and to do so with a minimum amount of time and effort.

5.2.1 Black Box Testing

Black box testing includes testing, that are conducted at the software interfaces. These are used to demonstrate the software functions operational whose input is properly accepted and the output are correctly produced.

Tools used for Black Box testing:

Black box testing tools are mainly record and playback tools. These tools are used for regression testing that to check whether new build has created any bug in previous working application functionality. These record and playback tools records test cases in the form of some scripts like TSL, VB script, Java script, Perl.

Advantages of Black Box Testing:

- Tester can be non-technical. – Used to verify contradictions in actual system and the specifications.
- Test cases can be designed as soon as the functional specifications are complete.

Methods of Black box Testing:

Graph Based Testing Methods: Each and every application is build up of some objects. All such objects are identified and graph is prepared. From this object graph each object relationship is identified and test cases written accordingly to discover the errors.

- **Error Guessing:**

  This is purely based on previous experience and judgment of tester. Error Guessing is the art of guessing where errors can be hidden. For this technique there are no specific tools, writing the test cases that cover all the application paths.

❖ **Boundary Value Analysis:**

Many systems have tendency to fail on boundary. So testing boundary values of application is important. Boundary Value Analysis (BVA) is a test Functional Testing technique where the extreme boundary values are chosen. Boundary values include maximum, minimum, just inside/outside boundaries, typical values, and error values.

5.2.2 White Box Testing

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as clear box testing, open box testing, logic driven testing or path driven testing or structural testing and glass box testing. It is one of two parts of the "box testing" approach of software testing. Its counter-part, black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing.

The term "white box" was used because of the see- through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

It is predicated on close examination of procedural detailed logical test cases that exercise specific set of conditions and/or loops test software. Basis path testing is a white box testing technique. The basis path method enables the test case designer to drive a logical complexity of a procedural design and use this measure as a guide for finding basis set of execution path.

White box testing involves the testing of the software code for the following

- Internal security holes.
- Broken or poorly structured paths in the coding processes.

The flow of specific inputs through the code

- Expected output
- The functionality of conditional loops
- Testing of each statement, object and function on an individual basis.
- Test cases derived to exercise the basis set are guaranteed to execute every statement in the program at least one time during testing.

5.2.3 Testing Strategies Applied

The various testing strategies applied in performing the testing operation are as follows:

- **Unit Testing**

Unit testing concentrates on each unit of the software as implemented in source code. In unit testing the entire system is divided into small parts or units according to their functionality and independence, then each of these units are tested individually. The main motive behind dividing the system into small units is that it becomes very easy to test them thoroughly. So many errors in a unit if any are debugged right there, thus saving the time.

- **Integration Testing**

In this testing focus is on design and the construction of the software architecture. Integration testing is the process of putting together all the units, which are tested independently, and then testing them collectively. Here we ensure that the different views work fine when they are put together and also the various links provided are pointing towards the right location. It is also ensured that the database is working properly through the views data.

❖ **Validation Testing**

Validation testing is designed to validate a fully developed system. In this testing, the requirements established as part of software requirements analysis through software requirement specification are validated against the software that has been constructed. It provides final assurance that software meets all functional, behavioral, and performance requirements.

❖ **System Testing**

In System testing, the software and other system elements are tested as a whole. It verifies that all elements mesh properly and that overall system function/performance is achieved.

❖ **Acceptance Testing**

Acceptance testing is performed when the system is fully ready to show the demonstration in front of the client in order to obtain their acceptance for the system behavior and other input or output standards. This testing is performed to demonstrate the client, on the real life data of the client, the operation of the system.External behavior of the system is also considered.

6.1 LIMITATIONS

- Recognition of object with the Blind stick can be only up to knee level.
- It is bulky to carry.
- Requirements of power source.

6.2 CONCLUSION

In this project a brief discussion is made on the working of smart stick for visually impaired people which not helps in detecting objects using ultrasonic sensors but also provides a feature of tracking them with the help of GSM and GSM modules where the relative can track the person with a link sent through a message.

It can be further improved to have more decision taking capabilities by employing varied types of sensors and thus could be used for different applications. It aims to solve the problems faced by visually impaired people in their daily life the system also takes measures to ensure their safety.

6.3 SCOPE AND ENHANCEMENT

- The future work will focus on enhancement of object recognition system so that it can detect and identify objects better in challenging environmental conditions.
- It will also include improving the charge capacity of the device.
- The object identification can be improved by enhancing the image sensor.

6.4 BIBLIOGRAPHY

1. https://www.ijltet.org/journal/151854302245.%202062.pdf.

2. https://www.irjet.net/archives/V5/i3/IRJET-V5I3586.pdf.

3.https://circuitdigest.com/microcontroller-projects/arduino-smart-blind-stick.

4. https://www.youtube.com/results?search_query=smart+blind+stick.