

Guiding stick for visually challenged people based on TLC algorithm in IOT network

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

Vajja Pavankalyan(317126512173)

Yerra Bindu Srilekha(318126512L35)

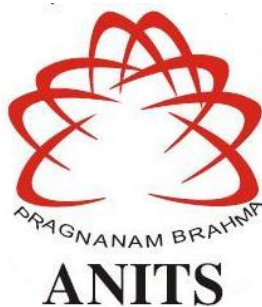
Bonthu Laxman Sai(318126512L25)

Piriya Vamsi(317126512161)

Under the guidance of

Mr.Nitin Kumar

Assistant Professor



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)**

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)

Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)

2020-2021

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)



ANITS

CERTIFICATE

This is to certify that the project report entitled "Guiding stick for visually challenged people based on TLC algorithm in iot network" submitted by Vajja Pavankalyan(317126512173),Yerra Bindu Srilekha(318126512L35), Bonthu Laxman Sai(318126512L25),Piriya Vamsi(317126512161) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics&Communication Engineering of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Nitin Kumar
Project Guide 19.07.21
Mr.Nitin Kumar
Assistant Professor
Department of E.C.E
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda

Institute of Technology & Sciences
Sangivalasa, Visakhapatnam - 531 102

[Signature]
Head of the Department
Dr. V.Rajyalakshmi
Professor & HOD
Department of E.C.E
ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 102

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mr.Nitin Kumar** Designation, Department of Electronics and Communication Engineering, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa**, for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

Vajja Pavan Kalyan(317126512173)

Yerra Bindu Srilekha(318126512L35)

Bonthu Lakshman Sai(318126512L25)

Piriya Vamsi(317126512161)

CONTENTS

ACKNOWLEDGEMENT	3
ABSTRACT	7
LIST OF FIGURES	8
LIST OF TABLES	9
CHAPTER 1 INTRODUCTION	10
1.1 Introduction	10
1.2 Internet of Things	11
1.3 Problems Faced by Visually Disabled People	12
1.4 Literature survey	13
CHAPTER 2 HARDWARE REQUIRED	15
2.1 Components Required	15
2.2 Arduino Uno	15
2.2.1 Power	16
2.2.2 Memory	16
2.2.3 Input Output	16
2.2.4 Communication	17
2.2.5 Programming	17
2.2.6 Automatic (Software) Reset	17
2.2.7 Physical Characteristics	18
2.3 UltraSonic Sensor	
2.3.1 What is ultrasonic wave?	18
2.3.2 How to use HC-SR04 Ultrasonic sensor?	18
2.3.3 HC-SR04 Ultrasonic Sensor – Working	19
2.3.4 Features of HC-SR04	20
2.3.5 Ultrasonic sensor Pin configuration	20

2.3.6 Applications	20
2.3.7 UltraSonic Sensor Connection with Arduino uno	21
2.4 Color Sensor	21
2.4.1 What is TCS3200?	22
2.4.2 Block Diagram of TCS3200	23
2.4.3 Circuit Diagram of TCS3200	23
2.4.4 Pin Diagram	25
2.4.5 Specifications of TCS3200	26
2.4.6 Applications of TCS3200	26
2.4.7 Color Sensor Connection with Arduino Uno	27
2.5 GSM Module	27
2.5.1 What is GSM Technology?	27
2.5.2 GSM Technology Architecture	28
2.5.3 GSM 900A Module	29
2.5.4 What are AT Commands?	29
2.5.5 Types of AT Commands	30
2.5.6 AT Commands List	30
2.5.7 Features of GSM Module	34
2.5.8 Applications	34
2.5.9 Steps to interface the GSM SIM900A with Arduino Uno	35
2.5.10 GSM Module Connection with Arduino Uno	36
2.6 Buzzer	37
2.6.1 How to use the Buzzer?	37
2.6.2 Buzzer Pin Configuration	38
2.6.3 Buzzer Features	38

2.6.4 Applications	38
2.6.5 Buzzer Connection with Arduino Uno	39
CHAPTER 3 SOFTWARE REQUIRED	40
3.1 Arduino IDE	40
3.1.1 Introduction to Arduino IDE	40
3.1.2 Arduino IDE Definition	40
3.1.3 How to get Arduino IDE	41
3.1.4 Libraries	42
3.1.5 Program structure of Arduino IDE	43
CHAPTER 4 METHODOLOGY	44
4.1 Working Principle	44
4.2 Obstacle Detection	45
4.3 Color Detection	45
4.4 Message Alert System	46
4.5 Practical Circuit	46
4.6 Algorithm	46
4.7 code	49
CHAPTER 5 RESULTS	55
CHAPTER 6 CONCLUSION	59
CHAPTER 7 REFERENCES	60

ABSTRACT

God gifted sense of vision to the human being is an important aspect of our life. But there are some unfortunate people who lack the ability of visualizing things. The visually impaired have to face many challenges in their daily life. The problem gets worse when there is an obstacle in front of them. Blind stick is an innovative stick designed for visually disabled people for improved navigation. We are proposing a low-cost walking stick based on internet of things for efficient interface to blind people.

Visually challenged people often face problems while navigating around the places and on roads. Moving along the unknown terrain is a great challenge to such people and therefore keeping these things in mind, the smart stick has been designed to minimize the difficulties of visually challenged people. A smart stick based on Traffic Light Crossing (TLC) Algorithm has been proposed for the navigation and guidance of visually challenged individuals while walking through roads and crossing traffic lights. While travelling on the road, if a traffic signal is sensed by the stick, the sensor identifies whether it is green or red for the person to cross the road. If the signal is red, the stick raises an alarm to go and again raise an alarm when the signal is green to stop.

LIST OF FIGURES

Fig 1.1	Internet of Things	11
Fig 1.2	Problems Faced by Blind People	13
Fig 2.1	Arduino Uno	15
Fig 2.2	How to use HC-SR04 Ultrasonic Sensor	18
Fig 2.3	Working of HC-SR04 Ultrasonic Sensor	19
Fig 2.4	Pin configuration of Ultrasonic Sensor	20
Fig 2.5	Applications of Ultrasonic Sensor	21
Fig 2.6	Ultrasonic Sensor Connection with Arduino Uno	21
Fig 2.7	Photodiode Spectral Responsivity	22
Fig 2.8	Color Sensor	23
Fig 2.9	Block Diagram of Color Sensor	23
Fig 2.10	Circuit Diagram of Color Sensor	24
Fig 2.11	Pin Diagram of Color sensor	25
Fig 2.12	Color Sensor Connection with Arduino uno	27
Fig 2.13	GSM 900A Module	28
Fig 2.14	GSM Technology Architecture	28
Fig 2.15	GSM module connection with Arduino Uno	36
Fig 2.16	Buzzer	37
Fig 2.17	Buzzer Pin Configuration	38
Fig 2.18	Buzzer Connection with Arduino Uno	39
Fig 3.1	Arduino IDE	40
Fig 3.2	How to Get Arduino IDE	41
Fig 4.1	Block Diagram	44
Fig 4.2	Block Diagram of Color Sensor	45
Fig 4.3	Practical Circuit	47
Fig 5.1	RED Color Detected and Object not Detected	55
Fig 5.2	RED Color Detected and Object is very close to the Person	55
Fig 5.3	GREEN Color Detected and Object not Detected	56

Fig 5.4	GREEN Color Detected and Object is very close to the Person	56
Fig 5.5	BLUE Color Detected and Object is very close to the Person	57
Fig 5.6	BLUE Color Detected and Object not Detected	57
Fig 5.7	Actual Output	58

LIST OF TABLES

Table 2.1	Ultrasonic Sensor Pin Description	19
Table 2.2	Selection Line Logic for Frequency Scaling in Color Sensor	24
Table 2.3	Buzzer Pin Description	37

CHAPTER 1

INTRODUCTION

1.1 Introduction:

The interconnection of the physical objects, buildings, vehicles and other items that are embedded with electronic sensors, software and network connectivity which thus allows the objects to collect and exchange data is referred to as the Internet of Things .The term Internet of Things was coined by Kevin Ashton. IoT has a large number of applications in the modern world such as healthcare, agriculture, smart cities and so on. IoT has played a vital role in the lifestyles of people. IoT has also contributed a lot to minimize the difficulties of people with There are many IoT devices such as smart canes, smart glass, speech synthesizer based IoT devices, smart shoes, and so on for physically disabled people to assist them in their livelihood. People with disabilities struggle a lot for their survival and in performing their day to day activities. For visually challenged people, travelling independently from one place to another and to cross traffic lights is one of the most challenging tasks. Considering the situation of visually challenged persons, a smart stick has been developed that can assist them to walk on roads and cross the traffic light safely and independently. This smart stick is capable of detecting obstacles with the help of an ultrasonic sensor,detecting traffic light colour detection Colour sensor module and a GSM based message alert system.A survey by WHO (World Health Organization) carried out in 2011 estimates that in the world, about 1% of the human population is visually impaired (about 70 million people) and amongst them, about 10% are fully blind (about 7 million people) and 90% (about 63 million people) with low vision. The main problem with blind people is how to navigate their way to wherever they want to go. Such people need assistance from others with good eyesight. As described by WHO, 10% of the visually impaired have no functional eyesight at all to help them move around without assistance and safely. This study proposes a new technique for designing a smart stick to help visually impaired people that will provide them navigation. The conventional and archaic navigation aids for persons with visual impairments are the walking cane (also called white cane or stick) and guide dogs which are characterized by a many imperfections. The most critical shortcomings of these aids include: essential 9 skills and training phase, range of motion, and very insignificant information communicated been communicated. Our approach modified this cane with some electronics components and sensors, the electronic aiding devices are designed to solve such issues. The ultrasonic sensor, colour sensor, buzzer and GSM are used to record information about the presence of obstacles on the road. Ultrasonic sensors have the capacity to detect any obstacle within the distance range of 2 cm-450 cm. Therefore whenever there is an obstacle in this range it will alert the user. Colour Sensor is used to detect which colour is shown by the traffic lights.Most blind guidance systems use ultrasound because of its immunity to the

environmental noise. Vision is the most important part of human physiology as 83% of information human being gets from the environment is via sight. The 2011 statistics by the World Health Organization (WHO) estimates that there are 70 million people in the world living with visual impairment, 7 million of which are blind and 63 million with low vision. The conventional and oldest mobility aids for persons with visual impairments are characterized with many limitations. Some inventions also require a separate power supply or navigator which makes the user carry it in a bag every time they travel outdoor. These bulky designs will definitely make the user to be exhausted.

1.2 Internet of Things (IOT):

The term IoT is mainly used for devices that wouldn't usually be generally expected to have an internet connection, and that can communicate with the network independently of human action. For this reason, a PC isn't generally considered an IoT device and neither is a smartphone -- even though the latter is crammed with sensors. A smartwatch or a fitness band or other wearable device might be counted as an IoT device.



Fig.1.1

Uses of IOT:

1.Efficient resource utilization:

If we know the functionality and the way that how each device work we definitely increase the efficient resource utilization as well as monitor natural resources.

2.Minimize human effort:

As the devices of IoT interact and communicate with each other and do lot of task for us, then they minimize the human effort.

3.Save time:

As it reduces the human effort then it definitely saves out time. Time is the primary factor which can save through IoT platform.

4.Enhance Data Collection.

5.Improve security:

Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.

1.3.Problems Faced by Visually Disabled People:

- Can't walk easily on their own
- Suffer from obstacles
- Difficult to reach a certain place:
- Difficult to recognize current location and remaining distance to the target
- Difficult to figure out directions and routes which direction to go.

Require permanent Human Help...

Dependence on a Person to lead...



But a person is not always there..

Fig 1.2

1.4 Literature survey:

Sachin Kumar et al. [3] focuses on the changed lifestyles due to several applications of IoT such as IoT in healthcare, smart city, smart homes, pollution control and so on. This paper also describes the IoT architecture, challenges of IoT, security and privacy issues, major IoT applications and also the importance of big data and its analysis concerning IoT. N. Sathya Mala et al. [4] proposed the theoretical approach to help visually impaired people by developing an electronic model. The model is based on a walking stick with Bluetooth headset and Global Positioning System (GPS). The author has used the ultrasonic module and navigation system module which is only able to help visually impaired people by detecting the obstacles but unable to identify the objects. Zeeshan Saquib et al. [5] proposed the Electronic Travelling Aid (ETA) named as BlinDar for helping blind people to make them walk without any constraints. The paper discussed about the BlinDar smart stick equipped with ultrasonic sensor, gas sensor, Wi-Fi module, RF module, GPS module, buzzer module, text-to-speech synthesis module interfaced with Arduino Mega 2560. Sankari Subbiah et al. [6] proposed a smart cane for obstacle detection, heat detection, water detection, light detection, pit and staircase detection, GPS (Global Positioning System) and GSM (Global System for Mobile). The paper discussed about the implementation of smart cane using a Raspberry Pi integrated with camera module, ultrasonic sensor, IR sensor, GPS, GSM for the navigation of visually impaired people. Karmel et al. [7] discussed about the device that can assist deaf, dumb and blind people for the convenient accomplishment of daily activities. The paper talks about the device that involves Google API (Application Programming Interfaces) and raspberry pi for vision, hearing and voice purpose. Image converted to text and speech synthesis helps the

person to read stuff in daily life. Mingyong Zhou et al. [8] proposed and discussed about an IoT system that includes wearing glass and walking assistance. The system has been designed to assist blind people that is based on ZigBee communication and cloud computing. The system is interfaced with voice speaker, microprocessor, ZigBee, GPRS module and cloud computing. Jee-Eun Kim et al. [9] proposed a three location-based service (LBS) which can be accessed from smartphones without the need of any other additional hardware. The author also described the infrastructure of IoT for positioning and web APIs for obtaining relevant information. This paper introduced three different smartphone based applications SaSYS, Talking Transit and Smart Buliding for assisting visually impaired persons. Vasanth K et al. [10] discussed about the Google speech API (Application Programming Interfaces) based help for deaf and blind people. The author explained the speech synthesis to text and text to audio using raspberry pi and Google API. The converted audio had been amplified for deaf people. Emil Dziadczyk et al. [11] discussed the GPS based satellite navigation system. It described the structure of GPS system, how GPS signal has been created, calculation of GPS position and GPS based applications have been achieved. Jianliang Meng et al. [12] discussed about the speech recognition technology, fields in speech technology and analysis of problems of speech recognition. Ganesh Kotalwar et al. [13] discussed about the deep learning based android application for identifying and detecting the object to help blind people in navigation and to perform their day-to-day activities conveniently. Sethuraman N Rao et al. [14] focused on the escort system for visually disable people. The proposed system takes help of the smart phones along with an RF module tied to the waist of a user. The module also uses the ultrasonic sensor for obstacle detection. The message is received by the RF module and it is then communicated to the smart phone over the bluetooth network. The received message is delivered to the visually disable person in a language chosen by him. Sapna Chaudhary et al. [15] discussed the various IoT applications that are new and are not covered in other works of literature and also they have covered various detailed literature of IoT in their research paper. M Saravanan et al. [16] discussed about the SWAPN tool for assessing all its features such as warning detection where the things like weather information, change in temperature information are all provided to the visually challenged people. Sapna Chaudhary et al. [17] focused on the routing in a wireless network where they have implemented machine learning in order to know the network type namely Bluetooth, MANET and DTN based on the characteristics features of a node that they have collected in real time and then if the network type is MANET or DTN then they consider the best node, a node that has good battery percentage at that time and good amount of internal storage as an intermediate node for delivering a message to the destination.

CHAPTER-2

HARDWARE REQUIRED

2.1 Components Required:

- Arduino uno
- UltraSonic Sensor
- Colour Sensor
- GSM Module
- Buzzer
- Connecting jumper wires

2.2 ARDUINO UNO:

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits. Arduino Uno is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller.



Fig 2.1

2.2.1 POWER

Power The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack.

Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- **VIN**:- The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**:-This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3**:- A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**:-Ground pins.

2.2.2 MEMORY

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

2.2.3 INPUT OUTPUT

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

- **Serial: 0 (RX) and 1 (TX)**:- Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **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 attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11**:- Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)**. These pins support SPI communication using the SPI library.
- **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 Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin:**-Support TWI communication using the `Wire` library. There are a couple of other pins on the board:
- **AREF:**- Reference voltage for the analog inputs, Used with `analogReference()`.
- **Reset:**-Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

2.2.4 COMMUNICATION:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). 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, a .inf 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 `SoftwareSerial` library allows for serial communication on any of the Uno'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; see the documentation for details. For SPI communication, use the `SPI` library.

2.2.5 PROGRAMMING

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, 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.

2.2.6 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines

(DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following halfsecond or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

2.2.7 Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

2.3 UltraSonic Sensor:

2.3.1 What is ultrasonic wave?

Sound wave is a vibration that is transmitted through a medium, such as air, water, and metals. Ultrasonic wave is defined as “inaudible sound with high frequency for human” the frequency of which generally exceeds 20 kHz. These days, sound wave which is not intended to be heard is also called ultrasonic wave.

2.3.2 How to use HC-SR04 Ultrasonic sensor:

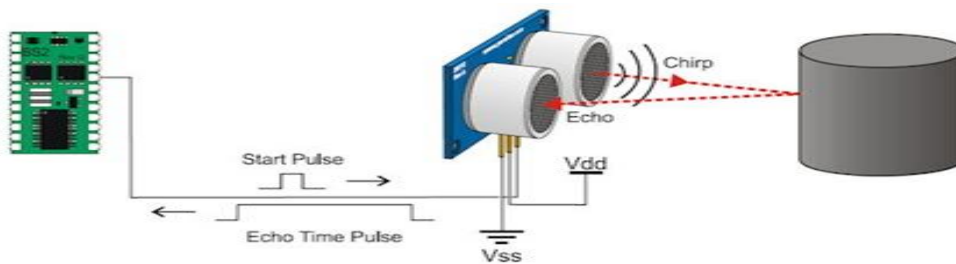


Fig 2.2

HC-SR04 distance sensor is commonly used with both microcontroller 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 and 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 microcontroller. 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.

2.3.3 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

$$\text{Distance} = \text{Speed} \times \text{Time}$$

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.



Fig 2.3

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 microcontroller or microprocessor.

2.3.4 Features of HC-SR04:

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <math><15^\circ</math>
- Operating Current: <math><15\text{mA}</math>
- Operating Frequency: 40Hz

2.3.5 Ultrasonic sensor Pin configuration:

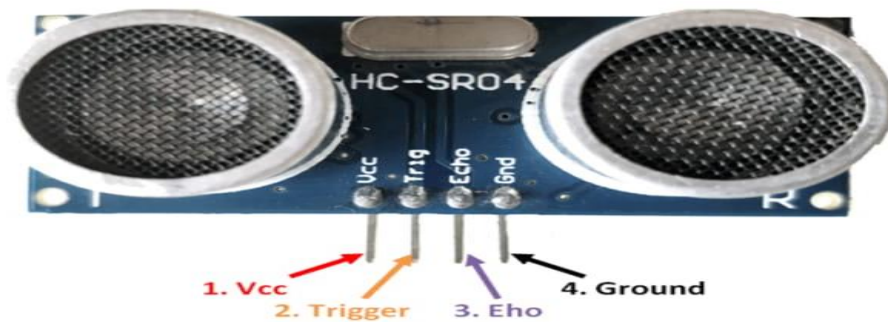


Fig 2.4

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.

Table 2.1

2.3.6 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.

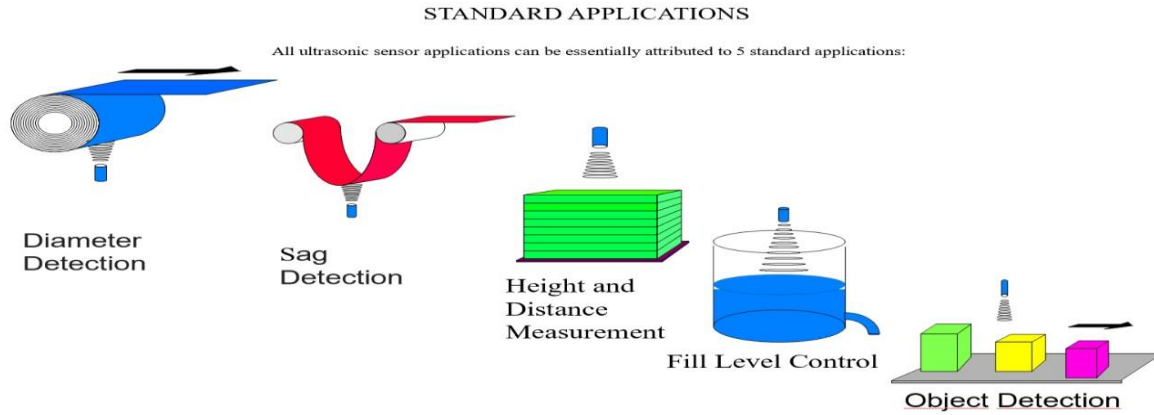


Fig 2.5

2.3.7 Ultrasonic Sensor connection with Arduino Uno:

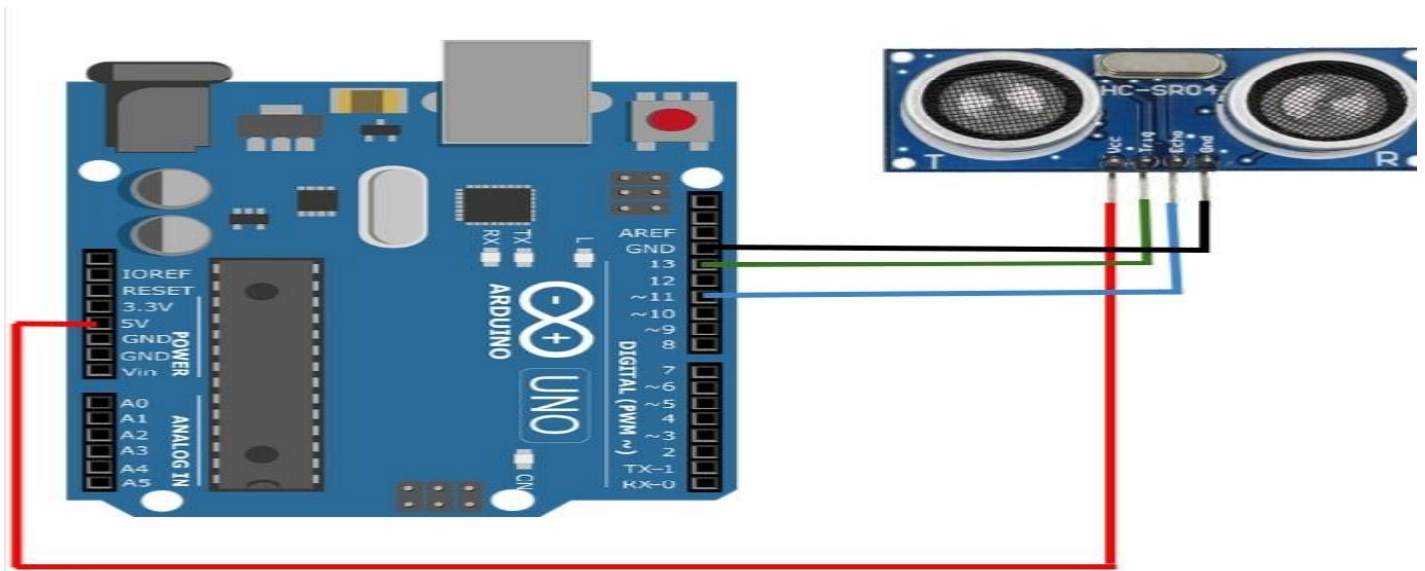


Fig 2.6

2.4 Colour sensor:

White light is composed of seven colors. This color spectrum of light is known as VIBGYOR, indicating Violet, Indigo, Blue, Green, Yellow, Orange, and Green. Different colors have different wavelengths. When light falls

on the objects, based on the properties of the object, some of the wavelengths get absorbed by the object while some are reflected back. These reflected wavelengths give the color of the object which is detected by the human eye. Combination of wavelengths in different proportions generated different colors. The color sensor is the device which is used to detect color. These sensors can classify the color of the object into either red or blue or green. One of such sensor is TCS3200.

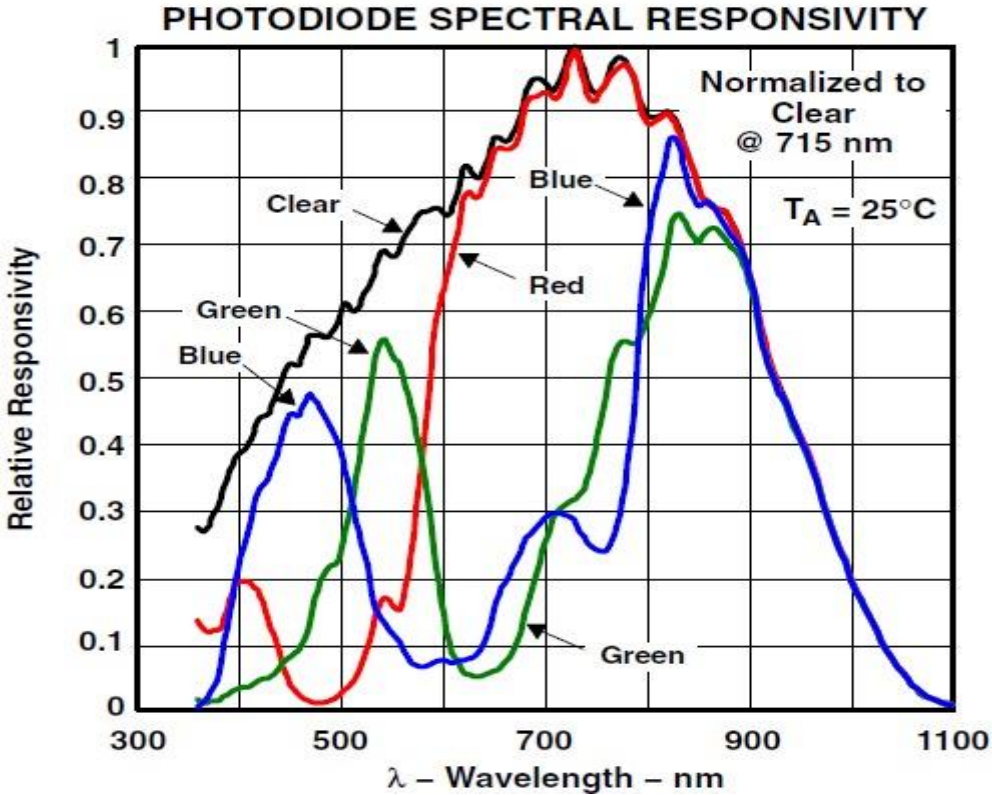


Fig 2.7

2.4.1 What is TCS3200?

TCS3200 is a programmable color light-to-frequency converter. This module can detect color with the help of microcontroller. It can detect a wide variety of colors like Red, Blue, and Green based on their wavelength. To optimize the sensor readings, frequency scaling is provided.

TCS3200 contains white LEDs to illuminate the object's surface whose color has to be detected. The intensity of the light reflected back by the object is calculated. A frequency proportional to the intensity is produced by the converter, using which the microcontroller predicts the color of the object.

For detecting different color, filters sensitive to different wavelengths are used. Usually, Red color is detected for wavelength 580nm, Green for wavelength 540nm and Blue for wavelength 450nm.



Fig 2.8

2.4.2 Block diagram of TCS3200:

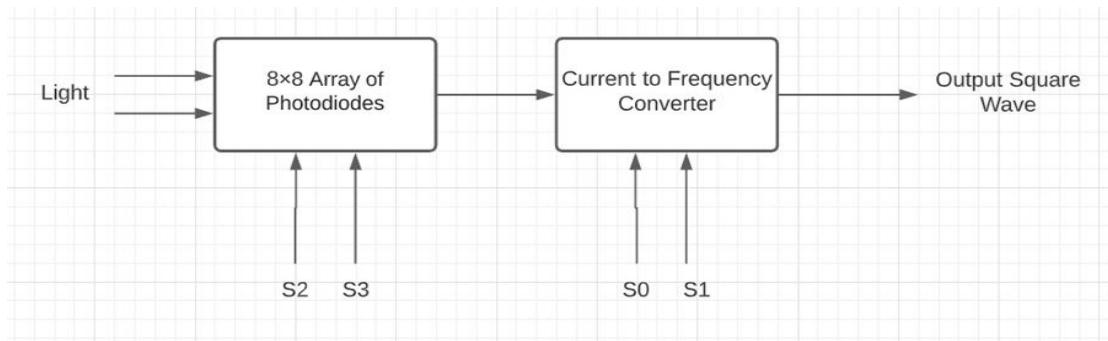


Fig 2.9

TCS3200 is a single monolithic CMOS IC. It consists of configurable silicon photodiodes and a Current-to-frequency converter. Photodiodes are arranged as an 8×8 array. From these photodiodes, 16 photodiodes have a blue filter, 16 photodiodes have a red filter, 16 photodiodes have a green filter and the remaining 16 photodiodes don't have any filter.

All photodiodes of the same color are connected in parallel. To minimize the effect of non-uniformity of incidence irradiance, four types of photodiodes are interdigitated. All the photodiodes are 110µm×110µm in size.

2.4.3 Circuit Diagram of TCS3200:

The output of TCS3200 is a square wave with 50% duty cycle. TCS3200 can be interfaced with any microcontroller directly using digital input and digital output pins. Using two control input pins, the full-scale output frequency can be scaled.

From the four different types of filter covered photodiodes, each diode can be activated using S2 and S3 selection inputs. When the diodes with red filter are chosen, only red incident light is measured and the remaining green and blue light are blocked. Then by measuring the frequency, the intensity of the red light can be detected. Using S0, S1 select inputs, frequency scaling factor can be set.

To measure the frequency, the 6th pin is used. This pin is connected to the microcontroller. To determine the color of the object, first set all the input pins as input and output pins as output. Here there is no use of analog pins. Next set the desired frequency scaling by setting the pins S0, S1 either high or low.

Now to detect the color of the object, activate each type of filter using S2 and S3 selection lines. After activation of filter measure the frequency generated on the 6th pin using a microcontroller. When both S2 and S3 pins are low, Red color filters are activated and the intensity of red components of the object is detected. Similarly, driving S2 low and S3 high detects Blue component and driving both S2, S3 to High detects Green components of the object. All these values are collected and compared to get the actual color of the object. Ambient light can cause variations in the measurements so sensor and object should be protected from the ambient light during measurement. 0.01 μ F to 0.1 μ F capacitors should be used for decoupling the power supply lines. For input noise immunity a low impedance electrical connection is required at the device output and device ground. A buffer is required if lines greater than 12 inches are used at the output.

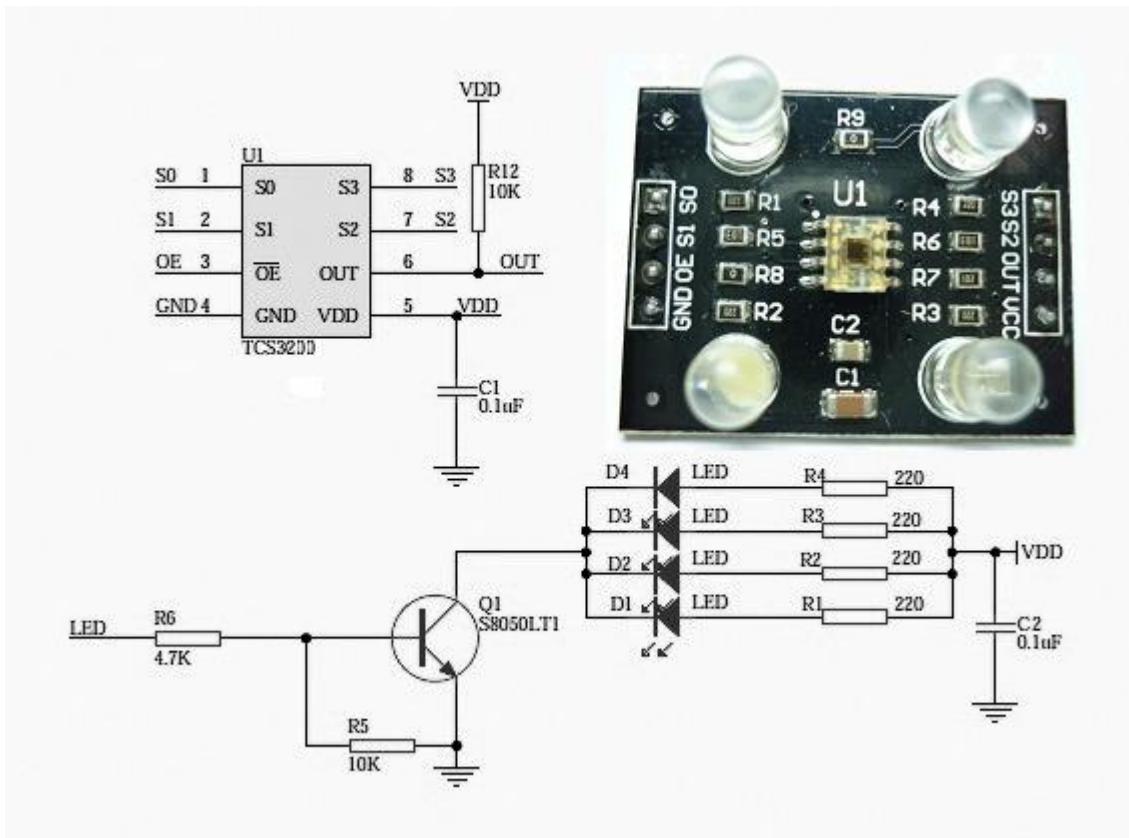


Fig 2.10

2.4.4 Pin Diagram

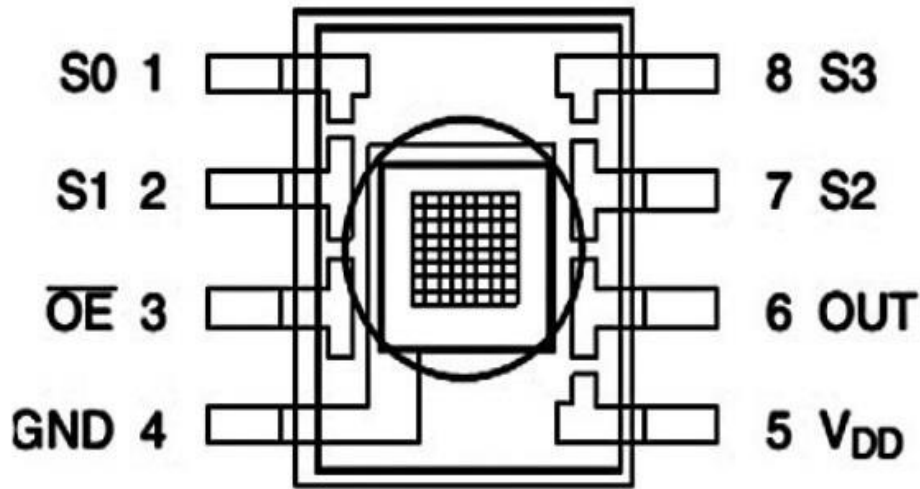


Fig 2.11

TCS3200 is available in the market as an 8-pin LEAD SOIC package D. Pin description of this module is given below-

- Pin-1 and Pin-2 are S0, S1 selection lines respectively. These pins are used for frequency scaling.
- Pin-3, OE, is the output enable pin. This pin is active low.
- Pin-4, GND, is the ground pin. This pin is the power supply ground.
- Pin-5, VDD, is the supply voltage pin.
- Pin-6, OUT, is the output frequency pin. This pin is connected to the microcontroller to read values.
- Pin-7 and Pin-8 are the S2, S3 selection lines respectively. These pins are used as Photodiode type selection inputs.

Selection line logic for frequency scaling is as below-

- When both S0, S1 are LOW, the device is in power-down mode.
- When S0 is LOW and S1 is HIGH, output frequency scaling will be 2%.
- When S0 is HIGH and S1 is LOW, output frequency scaling will be 20%.
- When both S0 and S1 are HIGH, output frequency scaling will be 100%.

S0	S1	OUTPUT FREQUENCY SCALING (f_o)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

S2	S3	PHOTODIODE TYPE
L	L	RED
L	H	BLUE
H	L	Clear (no filter)
H	H	GREEN

Table 2.2

2.4.5 Specifications of TCS3200:

Some of the specifications of TCS3200 are as follows-

- The main blocks of this module are TCS3200 RGB sensor chip and 4 white LEDs.
- The four LED's are given to provide sufficient lighting conditions for the sensor during color detection.
- TCS3200 chip consists of 8×8 array of photodiodes which can detect red, blue, green colors.
- This module works on the input supply voltage of 2.7V to 5.5V.
- Digital TTL interface is present in this module.
- TCS3200 chip converts light intensity into the frequency with high resolution.
- This module doesn't require an ADC to get digital values and can be connected to digital pins of microcontrollers directly.
- TCS3200 has a programmable color and full-scale output frequency.
- Power down feature is also given to this module.
- The operating temperature range of this module is from -40°C to 85°C.
- TCS3200 contains filters for red, blue, green colors.
- This module contains a non-linearity error of 0.2% at 50kHz.
- TCS3200 module has a stable 200 ppm/°C temperature co-efficient.

2.4.6 Applications of TCS3200

Some of the applications of TCS3200 are given below-

- TCS3200 is used to detect the color of the surfaces.
- This module finds application in Industries, health care, and manufacturing plants.
- TCS3200 is applied in medical diagnosis systems.
- For RGB LED consistency control TCS3200 is used.
- TCS3200 is applied for industrial process control.
- In laser edge banding machines to detect color TCS3200 is used.
- To detect chronic kidney diseases TCS3200 is used for urine analysis.
- TCS3200 is used in fruit sorting systems.
- The intensity of blue, red, green radiations can be measured using this sensor module.
- For classifying different types of metals this module is used.
- The food industry uses this sensor for sorting of fruits and vegetables.
- TCS3200 is also used in dental diagnosis and for ammonia detection.
- For designing of Light-to-Frequency receiver in Multi-color visible light communication, TCS3200 is used.

2.4.7 Colour sensor connection with Arduino Uno:

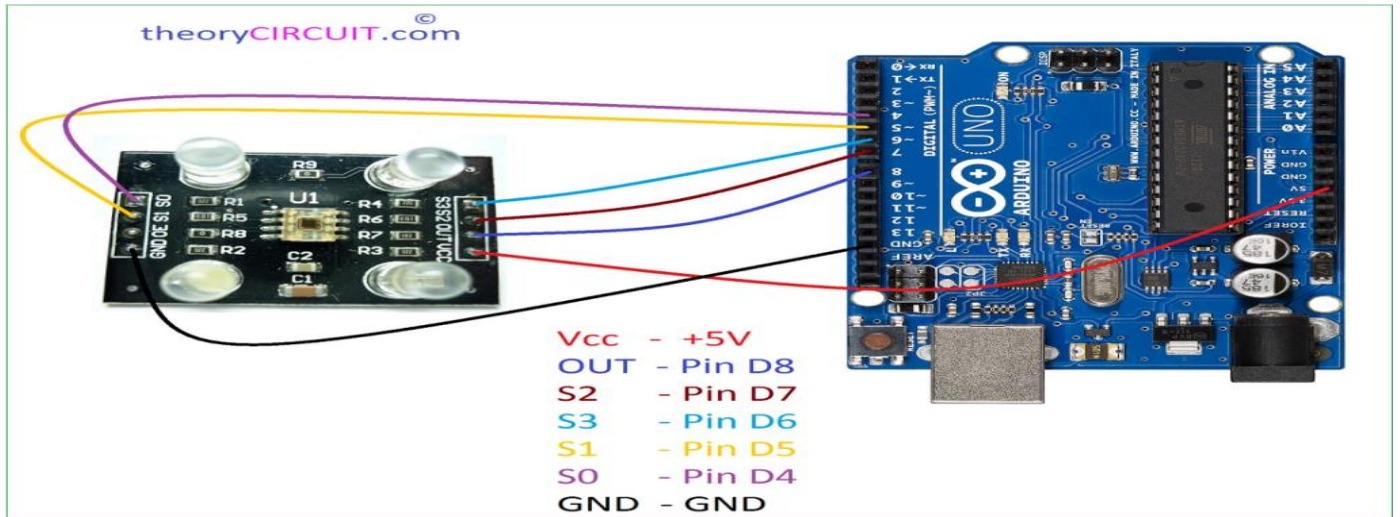


Fig 2.12

2.5 GSM Module:

Digital cellular technology like GSM (Global System for Mobile Communication) is used to transmit mobile data as well as voice services. This concept was implemented at Bell Laboratories using a mobile radio system in 1970. As the name suggests, it is the standardization group name that was established in the year 1982 to make a general European mobile telephone standard. This technology owns above 70% of the market share of the digital cellular subscriber around the world. This technology was developed by using digital technology. At present, GSM technology supports above 1 billion mobile subscribers around the world in the above 210 countries. This technology provides voice and data services from fundamental to complex.

2.5.1 What is GSM Technology?

GSM is a mobile communication modem; it stands for global system for mobile communication (GSM). The idea of GSM was developed at Bell Laboratories in 1970. It is a widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data services operate at the 850MHz, 900MHz, 1800MHz, and 1900MHz frequency bands.

GSM technology was developed as a digital system using the time division multiple access (TDMA) technique for communication purposes. A GSM digitizes and reduces the data, then sends it down through a channel with two different streams of client data, each in its own particular time slot. The digital system has the ability to carry 64 kbps to 120 Mbps of data rates.



Fig 2.13

There are various cell sizes in a GSM system such as macro, micro, pico, and umbrella cells. Each cell varies as per the implementation domain. There are five different cell sizes in a GSM network macro, micro, pico, and umbrella cells. The coverage area of each cell varies according to the implementation environment. The time division multiple access (TDMA) technique relies on assigning different time slots to each user on the same frequency. It can easily adapt to data transmission and voice communication and can carry 64kbps to 120Mbps of data rate.

2.5.2 GSM Technology Architecture

The main elements in the GSM architecture include the following.

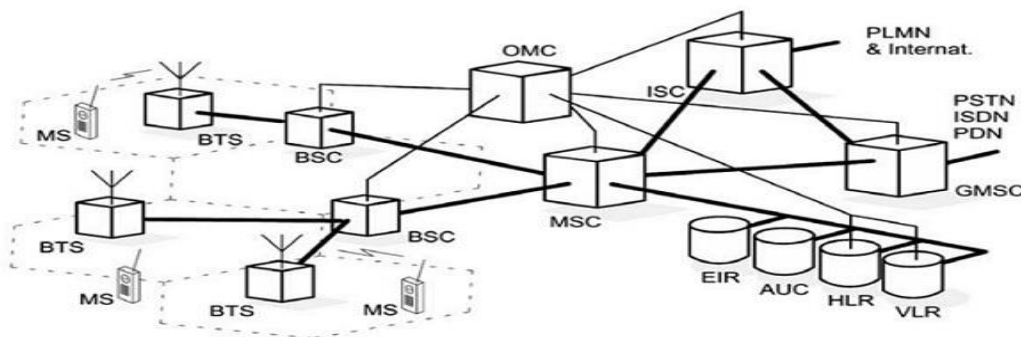


Fig 2.14

- Network and Switching Subsystem (NSS)
- Base-Station Subsystem (BSS)
- The mobile station (MS)
- Operation and Support Subsystem (OSS)

2.5.3 GSM 900A Module:

A GSM modem or GSM module is a hardware device that uses GSM mobile telephone technology to provide a data link to a remote network. From the view of the mobile phone network, they are essentially identical to an ordinary mobile phone, including the need for a SIM to identify themselves to the network. Here we are using GSM 900A Module. AT are used to control MODEMs. AT Commands we are used here are

1. To send SMS: AT+CMGS="mobile number"
2. Set the GSM modem to Text Mode AT+CMGF=1

The controlling of a MODEM can be done by using instructions or AT commands. Here term AT in the command is the short form of an Attention. The command line used in every modem starts with 'AT' otherwise 'at' so these commands are named as AT commands. There are many commands which are used for controlling modems (wired dial-up) like ATD – Dial, ATA – Answer, ATH – Hook control & ATO – Come back to the online data state. These are supported by modems like mobile phones, GSM or GPRS. There are some commands which support GSM. These commands which are used for GSM mainly include SMS based commands like AT+CMGS, AT+CMSS, AT+CMGL & AT+CMGR. Here the prefix AT in these commands informs the modem regarding the begin of a command line.

2.5.4 What are AT Commands?

In AT commands, AT stands for Attention and these commands are used for controlling MODEMs. These types of commands are taken from the commands like Hayes. The Hayes-commands mainly used in the Hayes smart modems. These commands are indicated with the term AT to specify the attention from MODEM. These commands are mainly used in the devices which use machine-to-machine communication to communicate with a PC. These devices consist of a subset of the Hayes command set with other extensive AT commands.

These commands are used in GSM, GPRS, or mobile phone MODEMs can be used to access the information as well as services which include the following.

- The information & configuration related to phone otherwise SIM card & MODEM.
- The services like SMS, MMS and Fax services.
- Voice and data link on a mobile network.

The basic commands are called as the Hayes subset commands and the commands which are exact to a GSM network are known as extended AT commands.

2.5.5 Types of AT Commands

These commands are classified into four types namely Test, Read, Set and Execution.

Test Command

The test AT command is mainly used for checking the command's compatibility using a modem. The SYNTAX for this command is AT < name of the command>. The best example of this command is AD =?

Read Command

The Read command is mainly used for changing the settings of mobile phone otherwise modem required for operations. The SYNTAX for this command is AT < name of the command>. The best example of this command is AT+CBC =?

Set Command

The Set command is mainly used for making modifications in the settings of mobile phone otherwise modem required for operations. The SYNTAX for this command is AT < name of the command> = Value 1, Value 2....Value N. The best example of this is AT+CBC ="+923140", 110

Execution Command

The Execution command is mainly used for executing the said operation. The SYNTAX for this command is AT < name of the command> = parameter-1, parameter-2... parameter-N. The best example of this is AT+CBC = 2,"+4867512120", 210.

2.5.6 AT Commands List

The list of AT commands is discussed below.

For Testing

AT command is used to check the communication among the computer as well as module

For Controlling Call

The commands used for controlling call mainly include the following.

- ATA command is an answer command
- ATH is used to hang up call
- ATD is a dial command
- ATM command is used to check speaker mode
- ATT command is used to fix tone dial like the default
- ATL command is used to check speaker loudness
- AT+CSTA command is used to select an address type
- ATP command is used to fix pulse dial like the default

- ATO is a Go on-line command
- AT+CRC command is used for cellular result codes

For Data Card Control

The commands used for data call control mainly include the following.

- ATI command is used for identification
- AT&F command is used to reinstate factory settings
- ATZ command is used to remind stored profile
- AT&W command is used to store parameters within the specified profile
- AT+CSTA command is used to select an address type
- AT+GMM command is used to identify request model
- AT&V command is used to examine the active configuration
- AT+CLCK command is used for facility lock
- AT+GCAP command is used to complete request of a capabilities list
- AT&Y command is used to choose power-up option
- AT+GMR command is used to identify request revision
- AT+COLP is a connected line recognition presentation
- AT+GMI is a request manufacturer identification command
- AT+GSN command is used to request product IMEI number.

For Phone Control

The commands used for phone control mainly include the following.

- AT+CBC command is used to charge the battery
- AT+CGMR command is used to identify request revision
- AT+CGSN command is used to request product serial number
- AT+CMEE command is used to report the error of mobile equipment
- AT+CPBF command is used to find the entries of the phone book
- AT+CPBR command is used to read entries of the phone book
- AT+CPBS command is used to choose the storage of phone book memory
- AT+CSCS command is used to select TE character set
- AT+CPBW command is used to write phone book entry
- AT+CGMM command is used to identify request model
- AT+CGMI command is used to identify request manufacturer
- AT+CSQ is a signal quality command
- AT+CPAS command is used to check the status of phone activity

Computer Data Interface

The commands used for computer data interface mainly include the following.

ATE is an Echo command

- ATQ command results code suppression
- ATX command is used to select response rang
- AT+ICF command is used to framing the character of DTE-DCE
- AT&Q command is used to identify an option for communications mode

- AT&C command is used to identify the usage of DCD
- AT&D command is used to identify the usage of DTR
- ATV command is a define response format
- AT+IFC command is used to control local flow for DTE-DCE
- AT&K command is used to choose flow control
- AT&S command is used to identify an option for DSR
- AT+IPR command is used to set DTE rate

For Service

The commands used for service mainly include the following.

- AT+CLIP command is used for calling line recognition presentation
- AT+ILRR command is used for reporting DTE-DCE
- AT+CR command is used to control service reporting
- AT+DR command is used for reporting data compression

For Network Communication Parameter

The commands used for network communication parameter mainly include the following.

- ATB command is an option for a communications standard
- AT+DS command is used to compress the data
- AT+CEER command is used to extend error report
- AT+CBST command is used to choose the type of bearer service
- AT+CRLP command is a radio link protocol

For Miscellaneous

The commands used for miscellaneous mainly include the following.

- A/ command is used to the re-execute command line
- AT*C command is used to start SMS interpreter
- AT? is used to command help
- AT*V command is used to activate V.25bis mode
- AT*T command is used to enter SMS block mode protocol
- AT+CESP command is used to enter SMS block mode protocol
- AT*NOKIATEST command is used to test command

For SMS Text Mode

The commands used for SMS text mode mainly include the following.

- AT+CSMS command is used to select message service
- AT+CSMP command is used to fix text mode parameters
- AT+CMGF command is used to format message
- AT+CPMS command is used to choose message storage
- AT+CRES command is used to restore settings
- AT+CNMI command is used to indicate a new message to TE
- AT+CSCA command is used for service center address
- AT+CSCB command is used to choose types of cell broadcast message
- AT+CMGR command is used to read the message

- AT+CSDH command is used to illustrate text mode parameters
- AT+CSAS command is used to save settings
- AT+CMSS command is used to send a message from storage
- AT+CMGD command is used to delete the message
- AT+CMGL command is used to list messages
- AT+CMGS command is used to send message
- AT+CMGW command is used to write a message to memory

SMS PDU Mode

The commands used for SMS PDU mode mainly include the following.

- AT+CMGL command is used to list messages
- AT+CMGW command is used to write a message to memory
- AT+CMGS command is used to send message
- AT+CMGR command is used to read the message

ESP8266 AT Commands

The ESP8266 Commands mainly include the following.

- AT+RST command is used to restart the module
- AT+CWQAP command is used to quit the AP
- AT+ CIPSTATUS command is used to obtain the connection status
- AT+CWJAP command is used to join the AP
- AT+CWMODE command is used to wi-fi mode
- AT+CWLAP command is used to list the AP
- AT+CIPSTART command is used to set up the connection for TCP otherwise UDP
- AT+CIPCLOSE command is used to close the connection for TCP otherwise UDP
- AT+ CIPMUX command is used to fix multiple connections
- AT+ CWSAP command is used to fix the parameters of AP
- AT+ CIPSERVER command is used to set as serve
- AT+CIPSEND command is used to send data
- AT+CIFSR command is used to obtain an IP address
- IPD command is used to receive data

LINUX AT Commands

The LINUX AT-commands is discussed below.

ATA Command: It is used to respond an incoming call through MODEM

ATD – ATH Commands: An ATD command is used to dial a voice call through MODEM whereas ATH command is used to hang-up the call.

Extended AT-Commands: The extended AT-commands like AT+CGMI, AT+CGSN, and AT+CIMI are *used* to provide information regarding the mobile manufacturer, IMEI, and IMSI (International Mobile Subscriber Identity)

Extended AT-Commands: The extended AT-commands like AT+CSQ, AT+CBC are used to get the status of battery charge and quality of the signal.

AT+CMGF Command: This command is used to send an SMS to a precise number. These are used for controlling a MODEM.

2.5.7 Features of GSM Module

The features of the GSM module include the following:

- Improved spectrum efficiency
- International roaming
- Compatibility with integrated services digital network (ISDN)
- Support for new services.
- SIM phonebook management
- Fixed dialing number (FDN)
- Real-time clock with alarm management
- High-quality speech
- Uses encryption to make phone calls more secure
- Short message service (SMS)

The security strategies standardized for the GSM system make it the most secure telecommunications standard currently accessible. Although the confidentiality of a call and secrecy of the GSM subscriber is just ensured on the radio channel, this is a major step in achieving end-to-end security.

2.5.8 Applications:

- Domestic gas leakage detector
- Portable gas detector
- Factories
- Industrial Combustible gas detector
- Lpg storage
- Gas car

2.5.9 Steps to Interface the GSM SIM900A With Arduino Uno:

1. Insert your SIM card to GSM module and lock it.
2. power up your gsm by connecting it to Arduino's 5V and GND
3. Connect the Antenna
4. Now wait for some time (say 1 minute) and see the blinking rate of 'status LED' or 'network LED' (D6)
//GSM module will take some time to establish connection with mobile network//
5. Once the connection is established successfully, the status/network LED will blink continuously every 3 seconds. You may try making a call to the mobile number of the sim card inside GSM module. If you hear a ring back, the gsm module has successfully established network connection.

Basic AT Command:

1. To change sms sending mode : **AT+CMGF=1**

```
mySerial.println("AT+CMGF=1");
```

2. To read SMS in text mode : **AT+CNMI=2,2,0,0,0**

```
mySerial.println("AT+CNMI=2,2,0,0,0");
```

3. To make a call : **ATD+60XXXXXXXXXX;** //replace X with number you want to call, change +60 to your country code

```
mySerial.println("ATD+60XXXXXXXXXX;");
```

4. To disconnect / hangup call : **ATH**

```
mySerial.println("ATH");
```

5. To redial : **ATDL**

```
mySerial.println("ATDL");
```

6. To receive a phone call : **ATA**

```
mySerial.println("ATA");
```

2.5.10 GSM Module connection with Arduino Uno:

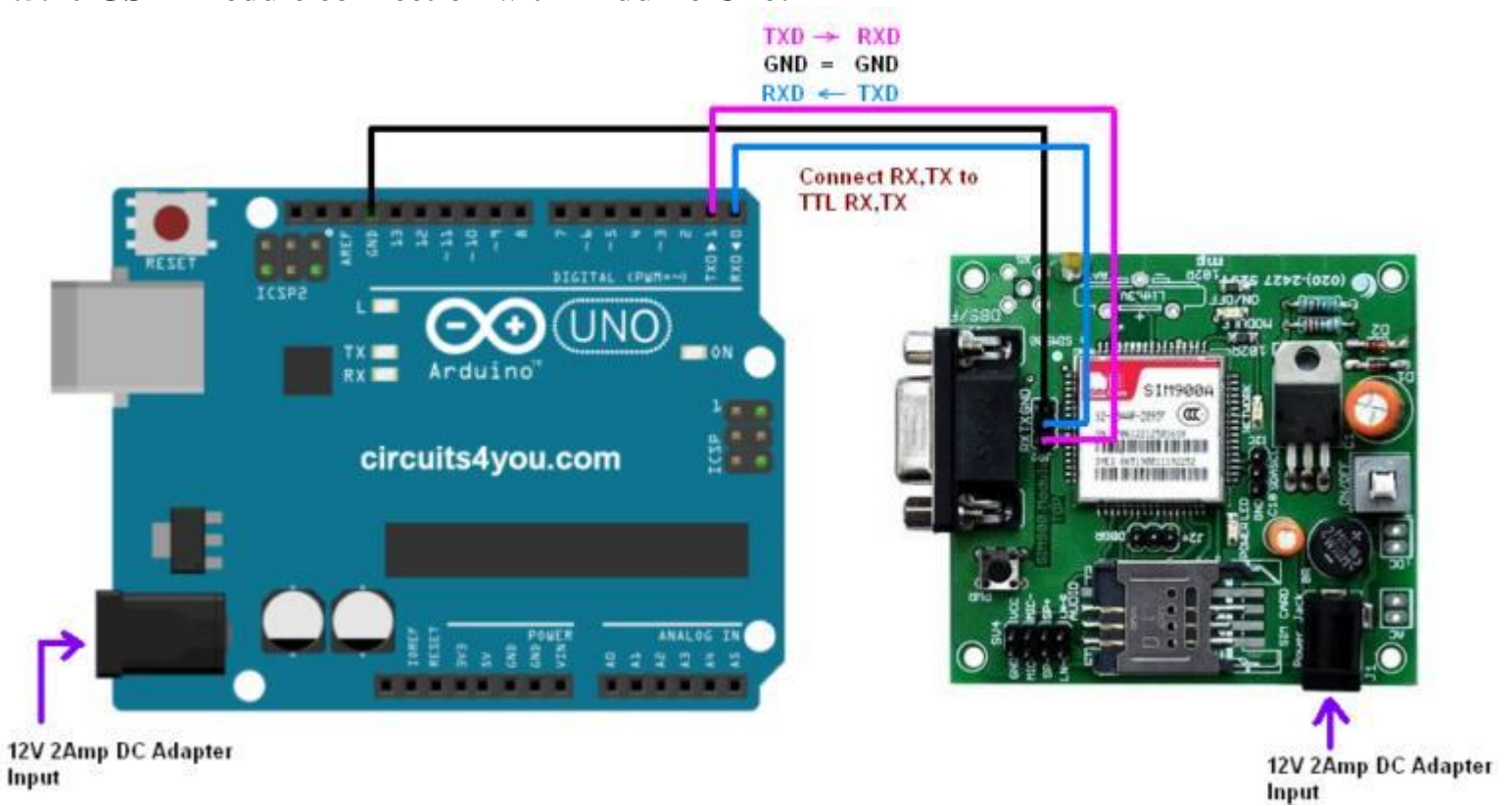


Fig 2.15

2.6 Buzzer:

2.6.1 How to use the Buzzer:

A **buzzer** is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on breadboard, Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

There are two types of buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beeeeeeppp.... sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep. Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customised with help of other circuits to fit easily in our application.



Fig 2.16

This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and require interval.

2.6.2 Buzzer Pin Configuration:

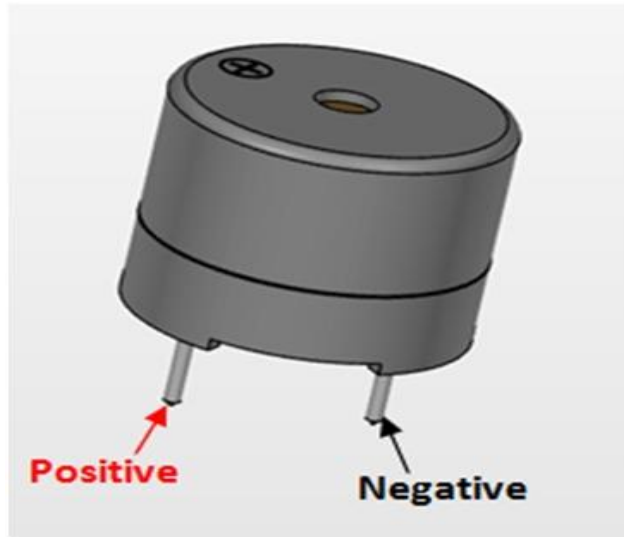


Fig 2.17

Pin Number	Pin Name	Description
1	Positive	Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC
2	Negative	Identified by short terminal lead. Typically connected to the ground of the circuit

Table 2.3

2.6.3 Buzzer Features:

- Rated Voltage: 6V DC
- Operating Voltage: 4-8V DC
- Rated current: <30mA
- Sound Type: Continuous Beep
- Resonant Frequency: ~2300 Hz
- Small and neat sealed package
- Breadboard and Perf board friendly

2.6.4 Applications:

- Alarming Circuits, where the user has to be alarmed about something
- Communication equipments
- Automobile electronics
- Portable equipments, due to its compact size

2.6.5 Buzzer connection with Arduino Uno:

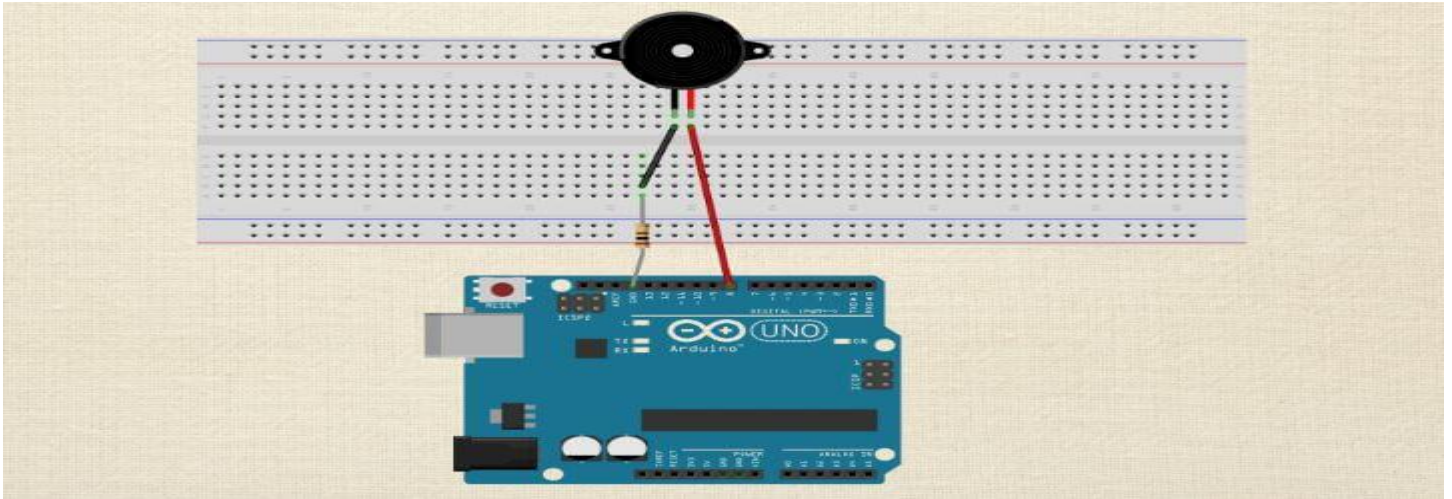


Fig 2.18

Chapter-3

Software Required

3.1 Arduino IDE

3.1.1 Introduction to Arduino IDE :

IDE stands for “Integrated Development Environment” :it is an official software introduced by Arduino.cc, that is mainly used for editing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go. In this article, we will introduce the Software, how we can install it, and make it ready for developing applications using Arduino modules.



Fig 3.1

3.1.2 Arduino IDE Definition :

1. Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
2. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
3. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
4. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.
5. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
6. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

7. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

8. This environment supports both C and C++ languages.

3.1.3 How to get Arduino IDE :

we can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MACOs,we select to download the correct software version that is easily compatible with our operating system.

Details on IDE:

The IDE environment is mainly distributed into three sections

- 1. Menu Bar
- 2. Text Editor
- 3. Output Pane

As we download and open the IDE software, it will appear like an image below

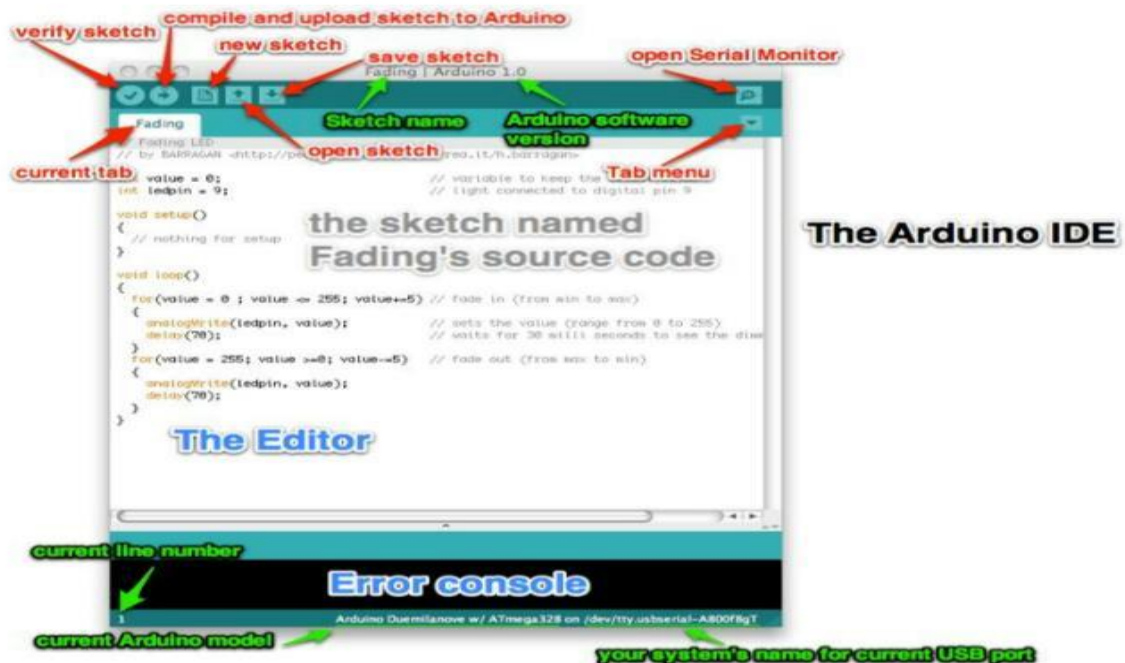


Fig 3.2

The bar appearing on the top is called Menu Bar that comes with five different options as follow

- File – You can open a new window for writing the code or open an existing one.
- Edit – Used for copying and pasting the code with further modification for font
- Sketch – For compiling and programming
- Tools – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
- Help – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

3.1.4 Libraries :

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library.

As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, Temperature sensors DHT11/22, LCD or I2C library it will appear on the text editor as

```
#include<EEPROM.h>
```

```
#include<dht.h>
```

```
#include<12Cdev.h>
```

```
#include <SoftwareSerial.h>
```

Most of the libraries are preinstalled and come with the Arduino software. However, we can also download them from the external sources.

Making Pins As Input or Output

The digitalRead and digitalWrite commands are used for addressing and making the Arduino pins as an input and output respectively.

These commands are text sensitive i.e. you need to write them down the exact way they are given like digitalWrite starting with small “d” and write with capital “W”. Writing it down with Digitalwrite or digitalwritewon’t be calling or addressing any function.

Example : if we want to use Pin D13 as output , the code will be; pinMode(13, OUTPUT); followed by digitalWrite(13,HIGH);

If we want to use Pin D13 as input, the code will be : `pinMode(13, INPUT);`
followed by `x=digitalRead(13);`

3.1.5 Program Structure of Arduino IDE :

Declarations

Variables:

Whenever you're using Arduino, you need to declare global variables and instances to be used later on. In a nutshell, a variable allows you to name and store a value to be used in the future. For example, you would store data acquired from a sensor in order to use it later. To declare a variable you simply define its type, name and initial value.

It's worth mentioning that declaring global variables isn't an absolute necessity. However, it's advisable that you declare your variables to make it easy to utilize your values further down the line.

Instances

In software programming, a class is a collection of functions and variables that are kept together in one place. Each class has a special function known as a constructor, which is used to create an instance of the class. In order to use the functions of the class, we need to declare an instance for it.

Setup():

Every Arduino sketch must have a setup function. This function defines the initial state of the Arduino upon boot and runs only once.

Here we'll define the following:

1. Pin functionality using the `pinMode` function
2. Initial state of pins
3. Initialize classes
4. Initialize variables
5. Code logic

Loop() :

The loop function is also a must for every Arduino sketch and executes once `setup()` is complete. It is the main function and as its name hints, it runs in a loop over and over again. The loop describes the main logic of your circuit.

Chapter-4 Methodology

Block Diagram:

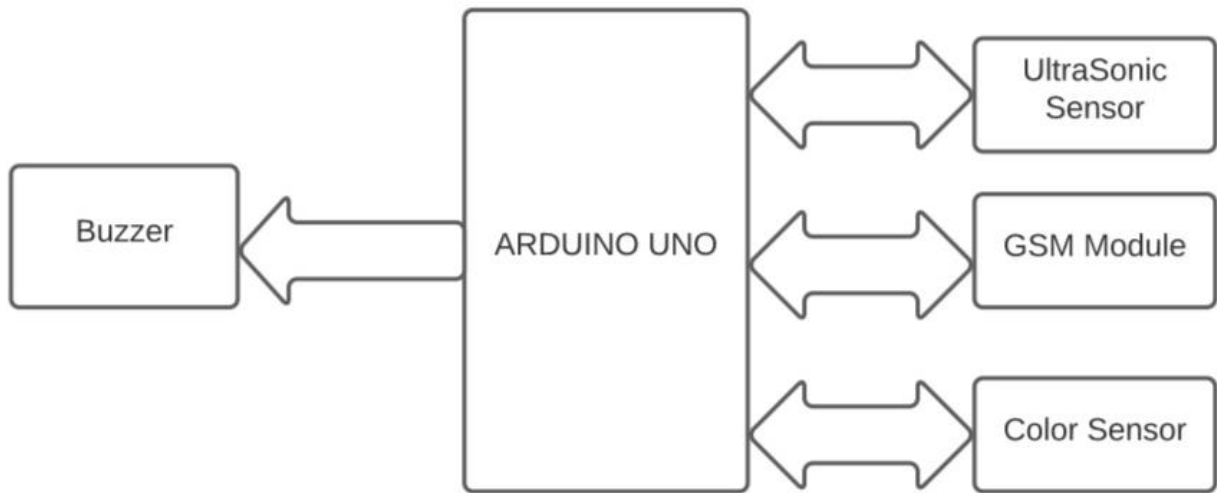


Fig 4.1

4.1 Working Principle:

The working of the obstacle detection module and the color detection module respectively, in the form of a flow chart. The flow begins with the compilation of code of each of the modules on the Arduino IDE to check for errors if any. Any prevailing error is debugged and the code is uploaded once it has been compiled successfully. The virtual terminal of Arduino shows the readings once the code is compiled. For the obstacle detection module, whenever an object is detected within the range of the detector, an LED blinks and buzzer gives an alert. Likewise, for the color detection module .whenever a color is detected, respective R G B values of the detected color is displayed on the virtual terminal. Similarly, represent the working of the message alert system and the audio alert system respectively, in the form of a flow chart. Following the same process as described above , once the code is compiled successfully, alert message is sent and an audio alert message is audible corresponding to the color of the traffic signal.

4.2 Obstacle detection:

- For obstacle detection, ultrasonic sensor (HC-SR04) is interfaced with Arduino Uno. The working of Ultrasonic sensor is based on SONAR (Sound Navigation and Ranging) and RADAR (Radio Detection and Ranging) principle. This sensor has been used to determine the proximity of an object by generating higher frequency sound waves (Ultrasound).
- The sensor's receiver senses the reflected echo after the ultrasound waves strike an obstacle. When an object comes in the range of sensor of obstacle detection module, the buzzer sound is heard. The buzzer sound alerts the visually challenged person about the obstacle in front of the stick.

4.3 Colour detection:

- The interfacing of colour detection module. For colour detection TCS3200 colour sensor has been used and the sensor consists of 8x8 array of photodiodes for sensing coloured lights. Square waves are obtained after the conversion of photodiode readings with the help of current-to-frequency converter.
- The converted frequency is directly proportional to the light intensity and therefore, when the square waves are analyzed with Arduino, colour of lights are detected. TCS3200 color sensor consists of 16 red filters, 16 green filters and 16 blue filters in photodiodes. It also consists of 16 clear with no filters in photodiodes. When the particular colour of traffic light is sensed by this sensor, it detects the threshold value of RGB color and recognizes the sensed color.

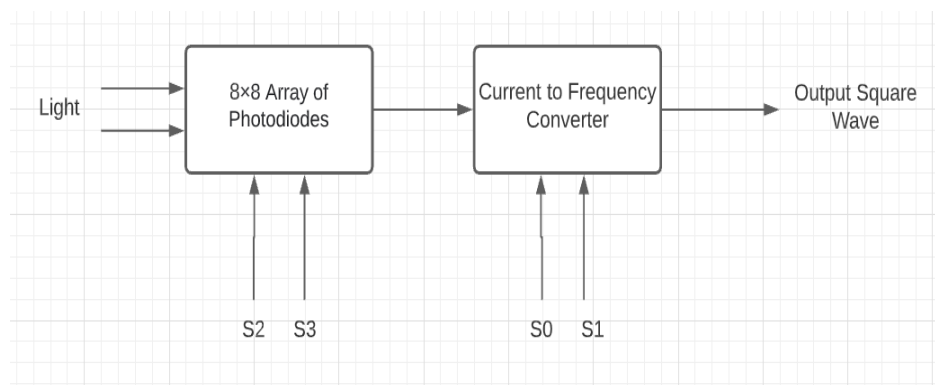


Fig 4.2

4.4 Message Alert system:

In case of emergency, the message alert is sent to the registered mobile no. to track the visually challenged person. When the attention command of GSM based system is run, it sends the message to the allocated number at the time of an emergency.

4.5 Practical circuit:

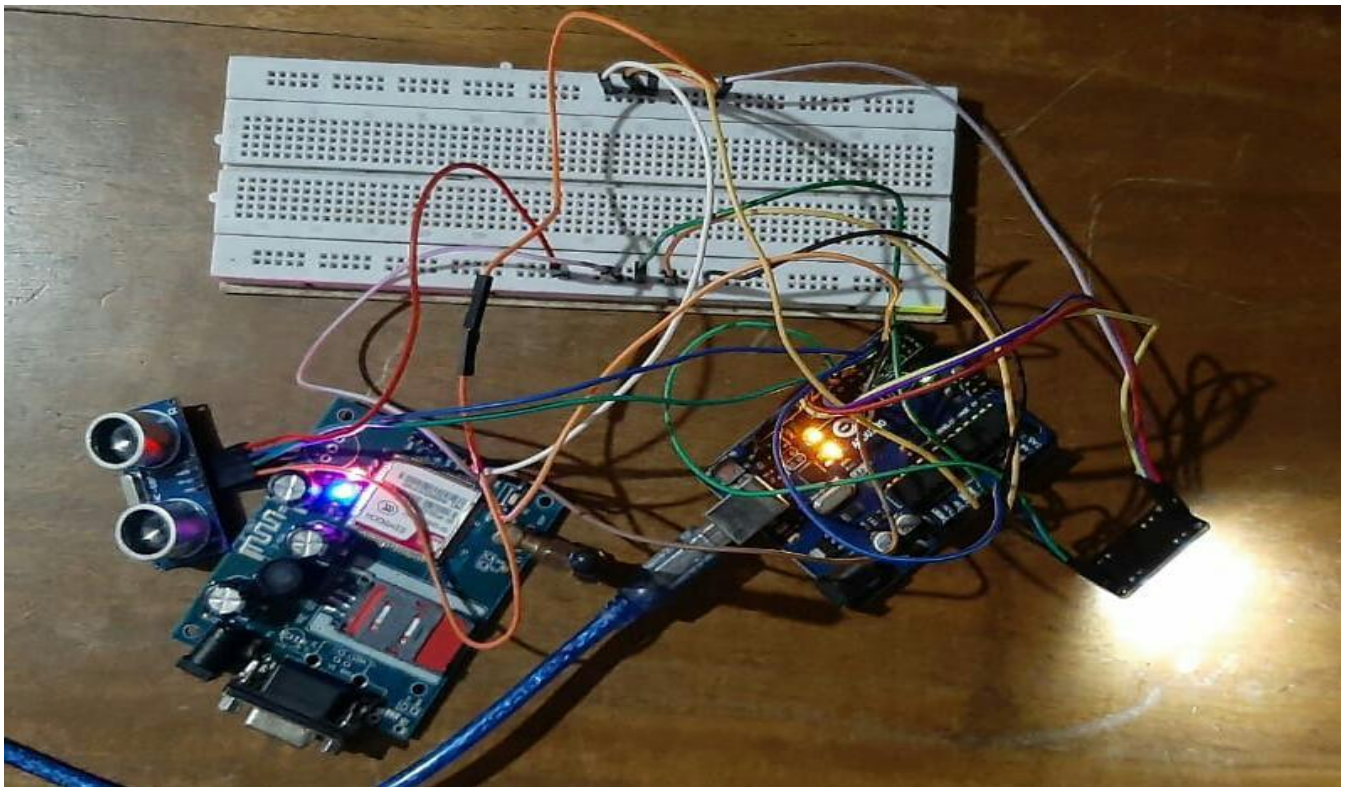


Fig 4.3

4.6 Algorithm:

In this Algorithm section the step by step functioning of Arduino code is shown. The Algorithm 1, Algorithm 2 and Algorithm 3 describes the obstacle detection, color detection and message alert system respectively. The trigger pin is used as an output and echo pin is used as an input of HC-SR04 (ultrasonic sensor), similarly, buzzer are used as an output in Algorithm 1 . Algorithm 2 is designed for color detection where threshold values of Red, Green and Blue

filters including clear filters of Photodiode are used to determine the detected color light. Algorithm 3 is for the working of GSM module in which message is sent using AT(Attention) commands.

Algorithm 1:Obstacle detection

Input:

trigger pin(ultrasonic sensor) set as OUTPUT

echo pin(ultrasonic sensor)set as INPUT

D = pulseIn(echoPin, HIGH)

D = D*0.034/2

Sd = d

if (Sd_i ≤ 20)

 Buzzer = HIGH

else

 Buzzer = LOW

Algorithm 2: Color detection

Input: s0, s1, s2, s3 are set as OUTPUT

s0, s1 digital pins are set as HIGH

if(R>B && R>G)

 Color = Red

else if(B>R && B>G)

 Color = Blue

else if(G>R && G>B)

 Color = Green

Algorithm 3: Message Alert System

```
if(G == "available")
```

```
  a = read
```

```
  if( a == "#")
```

```
    G = "0x1a"
```

```
else
```

```
  G = "a"
```


4.7 CODE:

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900(2, 3);
#define echoPin 5
#define trigPin 7
#define S0 8
#define S1 4
#define S2 10
#define S3 11
#define sensorOut 12
String textForSMS;
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
int redColor = 0;
int greenColor = 0;
int blueColor = 0;
long duration;
int distance;
String f1001 = "xxxxxxxxxxx";
String f1002 = "xxxxxxxxxxx";
String f1003 = "xxxxxxxxxxx";
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
```

```

pinMode(S0, OUTPUT);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(sensorOut, INPUT);
digitalWrite(S0,HIGH);
digitalWrite(S1,HIGH);
Serial.begin(9600);
SIM900.begin(9600);
}
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.print(distance);
Serial.println(" cm");
delay(1000);
digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
redFrequency = pulseIn(sensorOut, LOW);
redColor = map(redFrequency, 5, 15, 255,0);

```

```

delay(1000);
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);

greenFrequency = pulseIn(sensorOut, LOW);
greenColor = map(greenFrequency, 10, 26, 255, 0);
delay(1000);
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);

blueFrequency = pulseIn(sensorOut, LOW);
blueColor = map(blueFrequency, 13, 23, 255, 0);
delay(1000);
if(redColor > greenColor && redColor > blueColor)
{
  if(distance<=40 && distance>5)
  {
    Serial.println(" - RED detected! and Object detected Don't Cross");
  }
  else if(distance<=5)
  {
    Serial.println(" - RED detected! but Object is very close to the person");
  }
  else
  {
    Serial.println(" - RED detected! but Object not detected");
  }
}

```

```

}
else if(greenColor > redColor && greenColor > blueColor){
  if(distance <=40 && distance>5){
    Serial.println(" - GREEN detected! But Object detected Don't Cross");
  }
  else if(distance<=5)
  {
    Serial.println(" - GREEN detected! and Object is very close to the person");
    textForSMS = "The Person needs your Help";
    sendsms(textForSMS, f1001);
    Serial.println(textForSMS);
    Serial.println("message sent.");
    delay(1000);

    sendsms(textForSMS,f1002);
    Serial.println(textForSMS);
    Serial.println("message sent.");
    delay(1000);

    sendsms(textForSMS, f1003);
    Serial.println(textForSMS);
    Serial.println("message sent.");
    delay(1000);
  }
else
{
  Serial.println(" - Green detected! but Object not detected");

```

```

    }
}
else if(blueColor > redColor && blueColor > greenColor)
{
    if(distance<=40 && distance>5)
    {
        Serial.println(" - BLUE detected! but object detected now don't cross ");
    }
    else if(distance<=5)
    {
        Serial.println(" - BLUE detected! and Object is very close to the person");
    }
    else
    {
        Serial.println(" - Blue detected! but Object not detected");
    }
}
}

void sendsms(String message, String number)
{
String mnumber = "AT + CMGS = \""+number+"\"";
    SIM900.print("AT+CMGF=1\r");
    delay(1000);
    SIM900.println(mnumber);

    delay(1000);
    SIM900.println(message);

```

```
delay(1000);  
SIM900.println((char)26);  
delay(1000);  
SIM900.println();  
delay(100);  
}
```

Chapter-5

Results



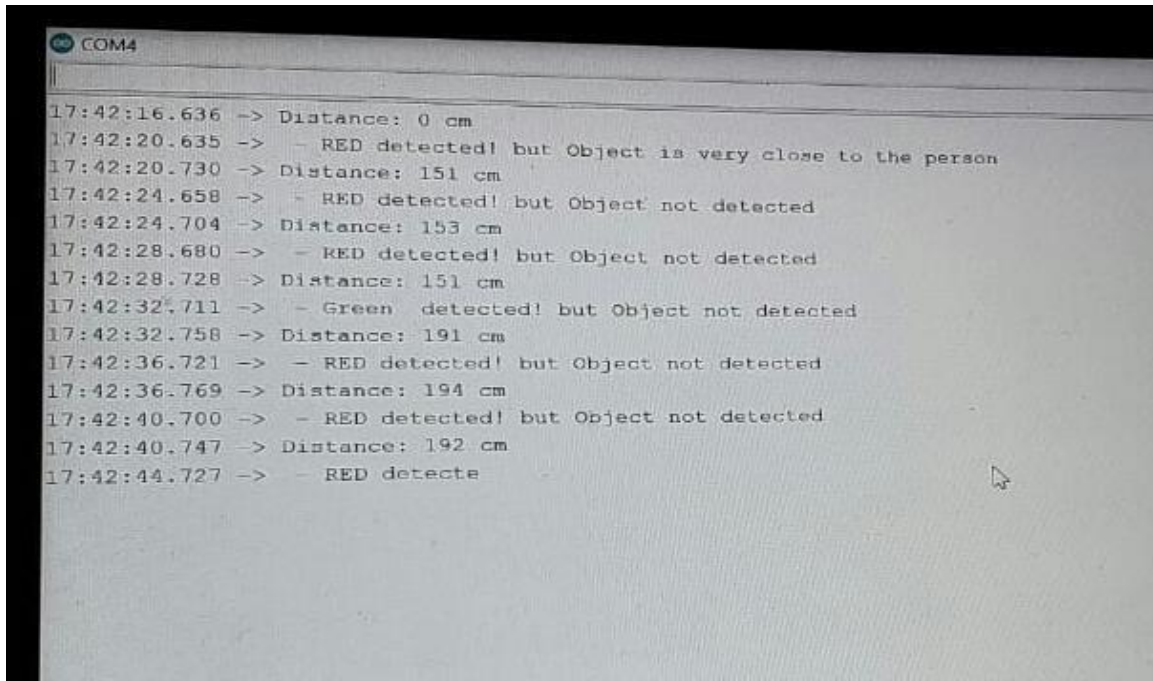


Fig 5.1

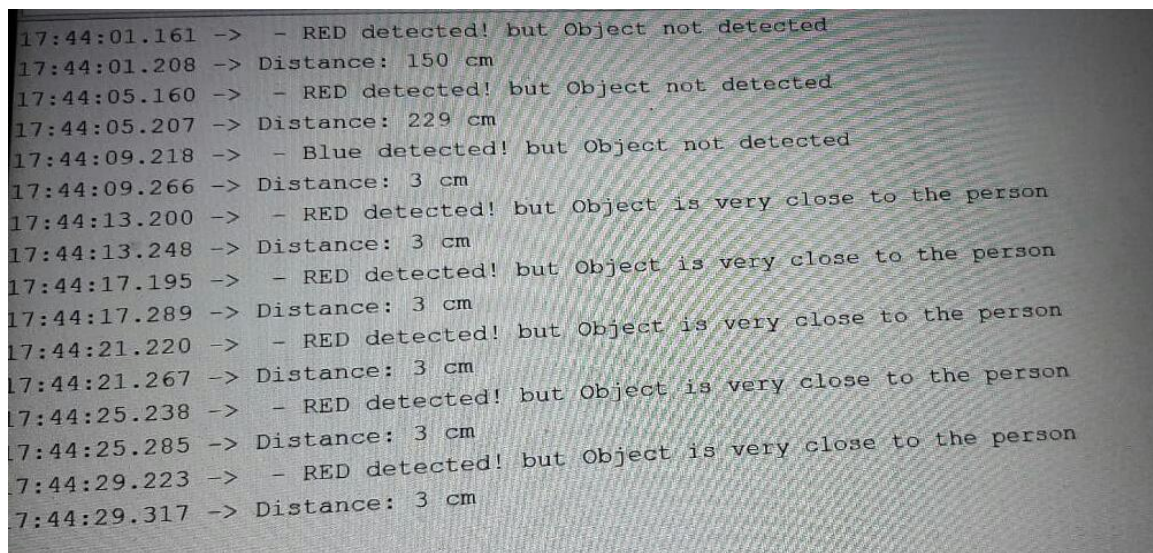


Fig 5.2


```
17:45:33.499 -> - Green detected! but Object not detected
17:45:33.546 -> Distance: 189 cm
17:45:37.535 -> - Green detected! but Object not detected
17:45:37.582 -> Distance: 156 cm
17:45:41.569 -> - Green detected! but Object not detected
17:45:41.615 -> Distance: 1187 cm
17:45:45.604 -> - Green detected! but Object not detected
17:45:45.651 -> Distance: 192 cm
17:45:49.658 -> - Green detected! but Object not detected
17:45:49.706 -> Distance: 152 cm
17:45:53.673 -> - Green detected! but Object not detected
17:45:53.721 -> Distance: 191 cm
17:45:57.666 -> - Green detected! but Object not detected
17:45:57.699 -> Distance: 154 cm
```

Fig 5.3

```
17:46:25.925 -> - Green detected! but Object not detected
17:46:25.972 -> Distance: 3 cm
17:46:29.929 -> - GREEN detected! and Object is very close to the person
17:46:30.023 -> Distance: 5 cm
17:46:33.951 -> - GREEN detected! and Object is very close to the person
17:46:33.998 -> Distance: 3 cm
17:46:37.969 -> - GREEN detected! and Object is very close to the person
17:46:38.016 -> Distance: 2 cm
17:46:41.988 -> - GREEN detected! and Object is very close to the person
17:46:42.036 -> Distance: 2 cm
17:46:46.017 -> - GREEN detected! and Object is very close to the person
17:46:46.064 -> Distance: 2 cm
17:46:49.990 -> - GREEN detected! and Object is very close to the person
17:46:50.085 -> Distance: 2 cm
```

Fig 5.4

```
:47:42.359 -> - Blue detected! but Object not detected
:47:42.406 -> Distance: 3 cm
:47:46.357 -> - BLUE detected! and Object is very close to the person
:47:46.451 -> Distance: 2 cm
:47:50.365 -> - BLUE detected! and Object is very close to the person
47:50.433 -> Distance: 2 cm
47:54.399 -> - BLUE detected! and Object is very close to the person
47:54.446 -> Distance: 2 cm
47:58.400 -> - BLUE detected! and Object is very close to the person
47:58.448 -> Distance: 2 cm
48:02.400 -> - BLUE detected! and Object is very close to the person
48:02.493 -> Distance: 2 cm
48:06.403 -> - BLUE detected! and Object is very close to the person
18:06.497 -> Distance: 2 cm
18:10.418 -> - BLUE detected! and Object is very close to the person
18:10.499 -> Distance: 3 cm
8:14.454 -> - BLUE detected! and Object is very close to the person
8:14.501 -> Distance: 3 cm
```

Fig 5.5

```
COM4
17:47:10.062 -> - Blue detected! but Object not detected
17:47:10.108 -> Distance: 189 cm
17:47:14.076 -> - Blue detected! but Object not detected
17:47:14.170 -> Distance: 1188 cm
17:47:18.173 -> - Blue detected! but Object not detected
17:47:18.221 -> Distance: 1188 cm
17:47:22.243 -> - Blue detected! but Object not detected
17:47:22.290 -> Distance: 154 cm
17:47:26.260 -> - Blue detected! but Object not detected
17:47:26.307 -> Distance: 192 cm
```

Fig 5.6

```
COM4
09:57:57.611 -> - Green detected! but Object not detected
09:57:57.644 -> Distance: 191 cm
09:58:01.595 -> - Green detected! but Object not detected
09:58:01.641 -> Distance: 189 cm
09:58:05.624 -> - Green detected! but Object not detected
09:58:05.667 -> Distance: 4 cm
09:58:09.628 -> - GREEN detected! and Object is very close to the person
09:58:13.822 -> The Person needs your Help
09:58:13.822 -> message sent.
09:58:18.989 -> The Person needs your Help
09:58:18.989 -> message sent.
09:58:24.145 -> The Person needs your Help
09:58:24.190 -> message sent.
09:58:25.162 -> Distance: 3 cm
09:58:29.145 -> - GREEN detected! and Object is very close to the person
09:58:33.345 -> The Person needs your Help
09:58:33.379 -> message sent.
09:58:38.525 -> The Person needs your Help
09:58:38.525 -> message sent.
09:58:43.679 -> The Person needs your Help
09:58:43.728 -> message sent.
09:58:44.708 -> Distance: 7 cm
09:58:48.717 -> - GREEN detected! But Object detected Don't Cross
09:58:48.761 -> Distance: 7 cm
09:58:52.683 -> - RED detected! and Object detected Don't Cross
09:58:52.773 -> Distance: 7 cm
09:58:56.712 -> - RED detected! and Object detected Don't Cross
09:58:56.758 -> Distance: 7 cm
09:59:00.725 -> - RED detected! and Object detected Don't Cross
09:59:00.768 -> Distance: 7 cm
```

Fig 5.7

Chapter-6

Conclusion

- In this the prototype of a smart stick based on TLC algorithm has been designed to assist visually challenged persons to walk on roads and to cross traffic lights independently.
- In the current proposed work the obstacle detection module with ultrasonic sensor, detection of traffic light with colour sensor, message alert system with GSM and audio alert system has been implemented and discussed.
- The smart stick consists of a buzzer to alert the person as well. In future, the smart stick can further be embedded with GPS navigation and guiding system, voice recognition system, security and privacy providing features.

Chapter-7

References

- [1] https://en.wikipedia.org/wiki/Internet_of_things
- [2] Gaurav, N.K., Johari, R., Seth, S., Chaudhary, S., Bhatia, R., Bansal, S. and Gupta, K., 2020. TLC Algorithm in IoT Network for Visually Challenged Persons. In *Ambient Communications and Computer Systems* (pp. 313-323). Springer, Singapore.
- [3] Kumar, S., Tiwari, P. and Zymbler, M., 2019. Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, 6(1), p.111.
- [4] Mala, N.S., Thushara, S.S. and Subbiah, S., 2017, February. Navigation gadget for visually impaired based on IoT. In *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)* (pp. 334-338). IEEE.
- [5] Saquib, Z., Murari, V. and Bhargav, S.N., 2017, May. BlinDar: An invisible eye for the blind people making life easy for the blind with Internet of Things (IoT). In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 71-75). IEEE.
- [6] Subbiah, S., Ramya, S., Krishna, G.P. and Nayagam, S., 2019, February. Smart Cane For Visually Impaired Based On IOT. In *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)* (pp. 50-53). IEEE.
- [7] Karmel, A., Sharma, A. and Garg, D., 2019. IoT based Assistive Device for Deaf, Dumb and Blind People. *Procedia Computer Science*, 165, pp.259-269.
- [8] Zhou, M., Li, W. and Zhou, B., 2017, November. An IOT system design for blind. In *2017 14th Web Information Systems and Applications Conference (WISA)* (pp. 90-92). IEEE.
- [9] Kim, J.E., Bessho, M., Koshizuka, N. and Sakamura, K., 2014, December. Mobile applications for assisting mobility for the visually impaired using IoT infrastructure. In *Proceedings of 2014 TRON Symposium (TRONSHOW)* (pp. 1-6). IEEE.

- [10] Vasanth, K., Macharla, M. and Varatharajan, R., 2019. A Self Assistive Device for Deaf & Blind People Using IOT. *Journal of medical systems*, 43(4), p.88.
- [11] Dziadczyk, E., Zabierowski, W. and Napieralski, A., 2007, February. Satellite navigation system GPS. In 2007 9th International Conference, The Experience of Designing and Applications of CAD Systems in Microelectronics (pp. 504-506). IEEE.
- [12] Meng, J., Zhang, J. and Zhao, H., 2012, August. Overview of the speech recognition technology. In 2012 fourth international conference on computational and information sciences (pp. 199-202). IEEE.
- [13] Kotalwar, G., Prajapati, J., Patil, S., Moralwar, D. and Jewani, K., 2018, December. Smart Navigation Application for Visually Challenged People in Indoor Premises. In International conference on Computer Networks, Big data and IoT (pp. 998-1004). Springer, Cham.
- [14] Rao, S.N. and Aswathy, V.R., 2016, December. An electronic escort for the visually challenged. In 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1-5). IEEE.
- [15] Chaudhary, S., Johari, R., Bhatia, R., Gupta, K. and Bhatnagar, A., 2019, April. Craiot: Concept, review and application (s) of iot. In 2019 4th international conference on internet of things: Smart innovation and usages (IoT-SIU) (pp. 1-4). IEEE.
- [16] Saravanan, M. and Vineet, G., 2014, September. SWAPN: A dream tool for the visually challenged. In 2014 IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS) (pp. 87-92). IEEE.
- [17] Chaudhary, S. and Johari, R., ORuML: Optimized Routing in wireless networks using Machine Learning. *International Journal of Communication Systems*, p.e4394. [18] <https://components101.com/tcs3200-color-sensor-module>.