# A Mini-Project Report

# On

# SMART GLASSES FOR BLIND PERSONS

# AT LOW COST

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE AWARD OF DEGREE OF

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS & COMMUNICATION ENGINEERING**

BY

**JELLA AKSHAYA (1608-20-735-063)**

**GADDI VIVEK (1608-20-735-087)**

**SHREYASH (1608-20-735-324)**

Under the guidance of

**Mrs. A. Narmada**

**M. Tech**

**Assistant professor**

**

**Department of Electronics and Communication Engineering**

**(NBA Accredited)**

**MATRUSRI ENGINEERING COLLEGE**

**(Sponsored by Matrusri Education socitey, Estd1980)**

**(Approved by AICTE, Affiliated to Osmania University)**

**#16-1-486,Saidabad, Hyderbad, Telangana-500 059**

[**www.matrusri.edu.in**](http://www.matrusri.edu.in/)*(12”BTNR)*

**2022-2023**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

Date:

Certificate

This is to certify that the Mini-project report entitled **SMART GLASSES FOR BLIND PERSONS AT LOW COST** being submitted by Ms **JELLA AKSHAYA**

(1608-20-735-063), **Mr. GADDI VIVEK** (1608-20-735-087)**, Mr.SHREYASH** (1608-20-735-324)in partial fulfillment for the award of the Degree of Bachelor of Engineering in Electronics and Communication Engineering of the Osmania University, Hyderabad, during 2022-23, is a record of bonafide work carried out under our guidance and supervision.

The results presented in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.**JELLA AKSHAYA**

**(1608-20-735-063) GADDI VIVEK (1608-20-735-087) SHREYASH (1608-20-735-324)**

Mini-Project Coordinator Name of the Guide

Designation

Dr. N. Srinivasa rao

HOD-ECE

**Certificate**

This page may include the Certificate given by Organization or Company where you have done your project (on their letter head). College certificate should precede this certificate.

**DECLARATION**

This is to declare that the work submitted in the present Mini- project work report titled **“ SMART GLASSES FOR BLIND PERSONS AT LOW COST”** is a record of bonafide work done by me in the Department of Electronics & Communication Engineering, Matrusri Engineering College, Saidabad, Hyderabad.

No part of the report is copied from books, journals, internet and wherever the subject content is taken; the same has been duly referred in the text. The report generated is based on the project work carried out entirely by us and not copied from any other source.

JELLA AKSHAYA (1608-20-735-063)

GADDI VIVEK (1608-20-735-087)

SHREYASH (1608-20-735-324)

Place: Hyderabad,

Date: 24/01/2023.

**ACKNOWLEDGEMENT**

We would like to take this opportunity to place it on the record, that this Mini-project would never have taken shape but for the cooperation extended to us by certain individuals. Though this is not possible to name all of them, it would be a pardonable on us part if we don’t mention some of the very important persons Sincerely we acknowledge our deep sense of gratitude to the project guide, **Mrs.A.Narmada** Assistant Professor for his constant encouragement, help and valuable suggestions. we wish to thank him for his constant motivation and help throughout the Mini-project.

We would like to express our deep gratitude to the Head of Department **Dr.** **N SRINIVASA RAO** and Project Coordinators **Dr.PALLAVI KHARE, ALEDDULA ABHISHEK REDDY** for their timely cooperation while carrying the Mini-project. It is their friendliness that made me feel free and learn more from them.

Last but not the least we would thank to all those people associated directly or indirectly with the Mini-project.

Ms. JELLA AKSHAYA

Mr. GADDI VIVEK

Mr. SHREYASH

**ABSTRACT**

**Abstract**— This device includes a pair of glasses and an obstacle detection modules fitted in it on both sides , a processing unit, an output device i.e. a beeping component or we can also use a vibrator , and a power supply. The Obstacle detection module and the output device is connected to the processing unit. The power supply is used to supply power to the central processing unit. The obstacle detection module basically consists of a ultrasonic sensors, processing unit consist of a control module and the output unit consists of a buzzer or vibrator. The control unit controls the ultrasonic sensors and get the information of the obstacle present in front of the man and processes the information and sends the output through the buzzer accordingly. These Ultrasonic Smart Glasses for Blind people is a portable device, easy to use, light weight, user friendly and cheap in price. These glasses could easily guide the blind people and help them avoid obstacles.

Blind as a special group in society, the needs of society to give them more care and attention, so that they are better able to live independently. However, how safe walking blind life is the biggest problem. Traditional navigation device mostly blind cane, blind by tapping the ground or walking around the object to determine the direction, the structure is simple, single function, easy to use, but the secondary effect is not very obvious, in fact, will encounter many problems when using the blind such as poor road conditions, uneven, hanging in front of obstacles, ordinary cane can not be proven accurate, such a serious impact on the safety of blind travelers.

**Keywords**: obstacle detection , ultrasonic sensor, processing unit, blind peo

**CONTENTS**

**Page No.**

**Certificate i**

**Declaration ii**

**Acknowledgement iii**

**Abstract iv**

**List of figures vii**

**List of Tables vii**

**Chapter 1. INTRODUCTION** 1

1.1 Introduction 1

1.2 Problem specification 2

1.3 Objectives 3

1.4 Layout of the Project

**Chapter 2. LITERATURE SURVEY**

**2**.1 lightweightsmart glass for visually impaired people

2.2 an integrated assistive solution using Google Glass.

2.3 an indoor navigation wearable system based on visual markers recognition

**Chapter 3. HARD WARE/SOFT WARE IMPLEMENTATION**

3.1 Introduction

3.2 Major Components Hardware

3.2.1 Arduino Nano

3.2.2 Ultra sonic Sensor

3.2.3 Color sensor

3.2.4 DF Player Mini

3.3 Major Software Specification

3.3.1 Software IDE

3.4 BLOCK DIAGRAM

3.5 Process of the Project

**Chapter 4. RESULT**

**Chapter 5. CONCLUSION**

5.1 conclusion

5.2 future scope

**REFERENCES**

**APPENDIX**

**Appendix I**

Project code

**Appendix II**

**LIST OF FIGURES**

|  |
| --- |
| **Figure** |
| Figure 1.1 layout of the project |
| Figure 3.1 Arduino Nano |
| Fig .3.2 arduino nano schematic |
| Fig 3.3 ultra sonic sensor |
| Fig 3.4 ultra sonic sensor working |
| Fig. 3.5 color sensor |
| Fig . 3.6 color sensor pins |
| Fig. 3.7 df player mini |
| Fig 3.8 df player pins |
| Fig .3.9 COMMAND WINDOW |
|  |

**LIST OF TABLES**

|  |
| --- |
| **LIST OF TABLES** |
| **Table 3.1 color sensor pinout** |
| **Table 3.2 color sensor filter selection** |
| **Table 3.3 frequency scaling** |
| Table 3.4 DF Player Mini pin map |
| **Table 3.5 symbol and their indication in arduino software ide** |

**Chapter 1**

**1.1 INTRODUCTION**

* Visual impairment or low vision is a severe reduction in vision that cannot be corrected with standard glasses or contact lenses and reduces a person's ability to function at certain or all tasks
* Blind people are usually dependent on assistance from others. The assistance can be from human beings, dogs or some special electronic devices.
* With the recent advances in technology normal walking cane has been modified to a blind stick with an ultrasonic sensor attached to it. It has several limitations.
* Therefore, we have the solution that has been protrayed in this presentation which is cost effective, reliable, robust and portable device which would help a blind person to detect the obstacles  and walk on the streets almost like any other pedestrian.

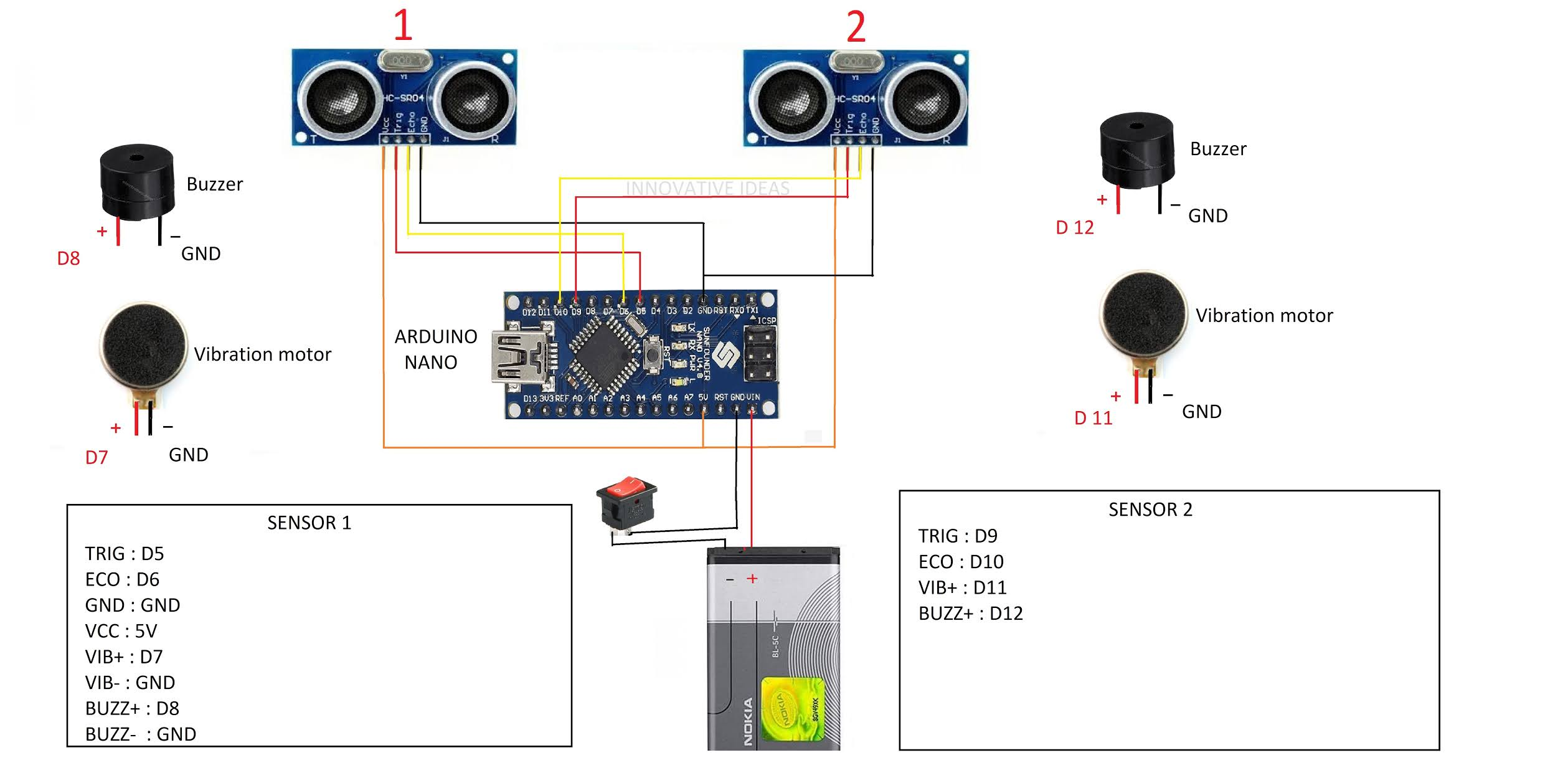
**1.2ProblemStatement**

* To help blind people to detect the obstacles and can walk like a normal people by using smart glasses and also implementation of color detection feature.

**1.3 Objective**

* To Design a low cost smart glasses for blind which can detect obstacles and intimate the user by using ultrasonic sensors.
* To implement the color detection feature which can detect the colors of the object and tell to user
* If the person is deaf then the output device mini vibrator can be used.

1.4 **layout of the project**

**fig. 1.1 layout of the project**

**Chapter 2**

**LITERATURE SURVEY**

2.1.Feng Lan, Guangtao Zhai and Wei L proposed a prototype system of lightweightsmart glass for visually impaired people. they demonstrated howthe smart glass was designed, including hardware design andsoftware design. And they have implemented many excellentimage processing, object recognition algorithms on the newlightweight smart glass system. This system can detect andrecognize the object in real time. The smart glass would beuseful for the visually impaired people in their city life. Andin the soon future, we will implement more useful applicationsin the smart glass system, such as talking to wikipedia, google,voice guidance and etc..

2.2 ASM Iftekhar Anam, Sahinur Alam, Md Yeasina they proposed an integrated assistive solution using Google Glass. The key function of the system is to enable the user in perceiving social signals during a natural dyadic conversation. The design and implementation of the system addressed a number of technical and research challenges - video acquisition and communication over Wi-Fi, efficient detection and tracking of faces, overheating of Google Glass, robust detection of facial features and modeling behavioral expressions, and feedback system for perceiving social signals. Performance evaluation was conducted to ensure the completeness and generalizability of models

2.3 W.C.S.S. Simoes, V.F.de Lucena an indoor navigation wearable system based on visual markers recognition and ultrasonic obstacles perception used as an audio assistance for blind people. In this prototype, visual markers identify the points of interest in the environment; additionally this location status is enriched with information obtained in real time by other sensors. A map lists these points and indicates the distance and direction between closer points, building a virtual path. The blind users wear also glasses built with sensors like RGB camera, ultrasonic, magnetometer, gyroscope, and accelerometer enhancing the amount and quality of the available information. The user navigates freely in the prepared environment identifying the location markers. Based on the origin point information or the location point information and on the gyro sensor value the path to next marker (target) is calculated. To raise the perception of the environment, avoiding possible obstacles, it is used a couple of ultrasonic sensors. The audio assistance provided to the user makes use of an audio bank, with simple known instructions to indicate precisely the desired route and obstacles. Ten blind users tested and evaluated the system. The results showed rates of about 94.92% successful recognition of the markers using only 26 frames per second and 98.33% of ultrasonic obstacles perception disposed between 0.50 meters and 4.0 meters.

**Chapter 3**

**HARD WARE/ SOFTWARE IMPLEMENTAION**

**3.1 Introduction**

This chapter explains in details the various stages involved in this research, the stages are explaining in form of units such as input unit, receiver unit, processor unit and output unit. According to the proposed system, I designed a system structure shown in the block diagram in this chapter. The model was designed in such a way that it can be kept at a safe place within. This is done in the easiest and lowest cost possible. However, the system is flexible and can be customized for future enhancement. Changing one of the components setup has to be compatible with the right software available. Every component used in this system was Programmed and tested separately for safety measures and matching with the right driver. Each component was programmed separately with Arduino NANO.

**3.2 MAJOR HARDWARE COMPONENTS**

**3.2.1 ARDUINO NANO**

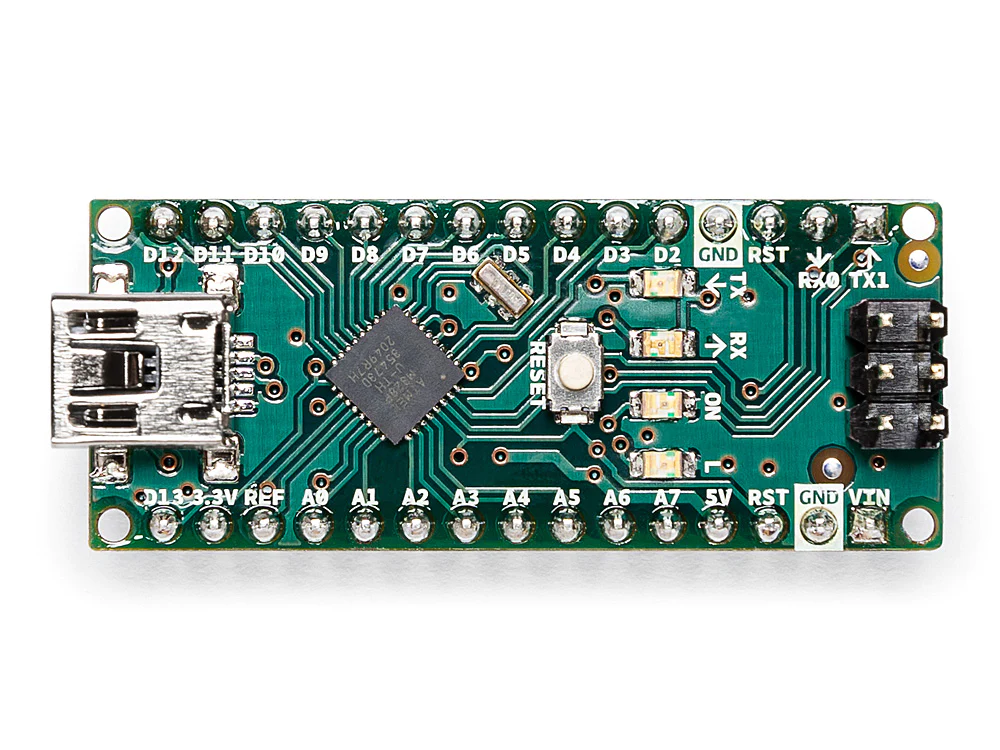
****

fig.3.2 Arduino Nano

Arduino Nano is one type of microcontroller board, and it is designed by Arduino.cc. It can be built with a microcontroller like Atmega328. This microcontroller is also used in Arduino UNO. It is a small size board and also flexible with a wide variety of applications. Other Arduino boards mainly include Arduino Mega, Arduino Pro Mini, Arduino UNO, Arduino YUN, Arduino Lilypad, Arduino Leonardo, and Arduino Due. And other development boards are AVR Development Board, PIC Development Board, Raspberry Pi, Intel Edison, MSP430 Launchpad, and ESP32 board.

This board has many functions and features like an Arduino Duemilanove board. However, this Nano board is different in packaging. It doesn’t have any DC jack so that the power supply can be given using a small USB port otherwise straightly connected to the pins like VCC & GND. This board can be supplied with 6 to 20volts using a mini USB port on the board.

**Arduino Nano Features**

The features of an Arduino nano mainly include the following.

ATmega328P Microcontroller is from 8-bit AVR family

Operating voltage is **5V**

Input voltage (Vin) is **7V to 12V**

Input/Output Pins are **22**

Analog i/p pins are 6 from **A0 to A5**

Digital pins are **14**

Power consumption is **19 mA**

I/O pins DC Current is **40 mA**

Flash memory is **32 KB**

SRAM is **2 KB**

EEPROM is **1 KB**

CLK speed is **16 MHz**

Weight-7g

Size of the printed circuit board is 18 X 45mm

Supports three communications like SPI, IIC, & USART

**Arduino Nano Pinout**

Arduino nano pin configuration is shown below and each pin functionality is discussed below.

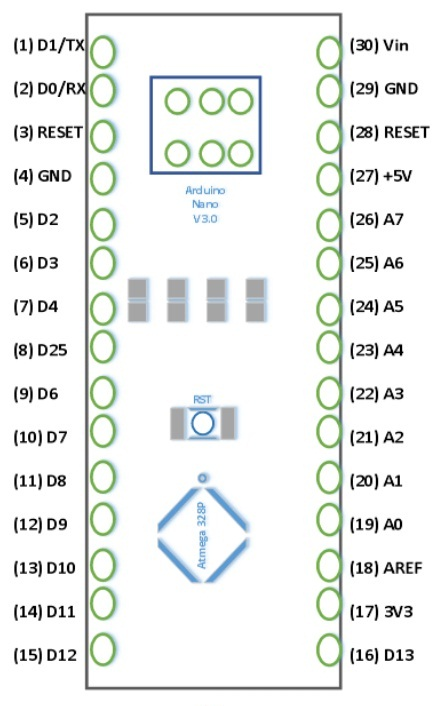


Fig .3.2 Arduino nano schematic

**Power Pin (Vin, 3.3V, 5V, GND)**: These pins are power pins

PCBWay

Vin is the input voltage of the board, and it is used when an external power source is used from 7V to 12V.

5V is the regulated power supply voltage of the nano board and it is used to give the supply to the board as well as components.

3.3V is the minimum voltage which is generated from the voltage regulator on the board.

GND is the ground pin of the board

**RST Pin( Reset)**: This pin is used to reset the microcontroller

**Analog Pins (A0-A7)**: These pins are used to calculate the analog voltage of the board within the range of 0V to 5V

**I/O Pins (Digital Pins from D0 – D13)**: These pins are used as an i/p otherwise o/p pins. 0V & 5V

**Serial Pins (Tx, Rx)**: These pins are used to transmit & receive TTL serial data.

**External Interrupts (2, 3)**: These pins are used to activate an interrupt.

**PWM (3, 5, 6, 9, 11)**: These pins are used to provide 8-bit of PWM output.

**SPI (10, 11, 12, & 13)**: These pins are used for supporting SPI communication.

**Inbuilt LED (13)**: This pin is used to activate the LED.

**IIC (A4, A5)**: These pins are used for supporting TWI communication.

**AREF**: This pin is used to give reference voltage to the input voltage

**DIGITAL**

An electronic signal transmitted as binary code that can be either the presence or absence of current, high and low voltages or short pulses at a particular frequency.Humans perceive the world in Analog, but robots, computers and circuits use Digital. A digital signal is a signal that has only two states. These states can vary depending on the signal, but simply defined the states are ON or OFF.

In the world of Arduino, Digital signals are used for everything with the exception of Analog Input. Depending on the voltage of the Arduino the ON or HIGH of the Digital signal will be equal to the system voltage, while the OFF or LOW signal will always equal OV. To receive or send Digital signals the Arduino uses Digital pins #0 - # 13. You may also setup your Analog In pins to act as Digital pins. To set up Analog In pins as Digital pins use the command:

**Pin Mode** (pin number, value);

The pin number is an Analog pin (AO-AS) and value is either INPUT or OUTPUT. To setup Digital pins use the same command but reference a Digital pin for pin number instead of an Analog In pin. Digital pins default as input, so real. really you only need to set them to OUTPUT in pin Mode. To read these pins use the command:

> **Digital Read** (pin number, value);

The pin number is the Digital pin to which the Digital component is connected. The digital Read command will return either a HIGH or a LOW signal. To send a Digital signal to a pin uses the command:

> **Digital Write** (pin number, value);

The pin Number is the number of the pin sending the signal and value is either HIGH or LOW.The Arduino also has the capability to output a Digital signal that acts as an Analog signal; this signal is called Pulse Width Modulation (PWM). Digital Pins # 3, #5, # 6, # 9, # 10 and #11 have PWM capabilities.

**ANALOG**

Humans perceive the world in Analog. Everything we see and hear is a continuous transmission of information to our senses. The temperatures we perceive are never 100% hot or 100% cold, they are constantly changing between our ranges of acceptable temperatures. (And if they are out of our range of acceptable temperatures then what are we doing there?) This continuous stream is what defines Analog data. Digital information, the complementary concept to Analog, estimates Analog data using only ones and zeros. In the world of Arduino an Analog signal is simply a signal that can be HIGH (on), LOW (off) or anything in between these two states. The Arduino does this by sampling the voltage signal sent to these pins and comparing it to a voltage reference signal (5V). Depending on the voltage of the Analog signal when compared to the Analog Reference signal the Arduino then assigns a numerical value to the signal somewhere between 0 (0%) and 1023 (100%). The digital system of the Arduino can then use this number in calculations and sketches. To receive Analog Input the Arduino uses Analog pins #0 - # 5. These pins are designed for use with components that output Analog information and can be used for Analog Input. There is no setup necessary, and to read them use the command: Analog Read (pin Number); The pin number is the Analog In pin to which the Analog component is connected. The Analog Read command will return a number including or between 0 and 1023. The Arduino also has the capability to output a digital signal that acts as an Analog signal; this signal is called Pulse Width Modulation (PWM). Digital Pins #3, #5, #6, # 9, # 10 and #11 have PWM capabilities. To output a PWM signal use the command: > Analog Write (pin Number, value); The pin Number is a Digital Pin with PWM capabilities and value is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are signified by a ~ sign.

**Arduino Nano Communication**

The communication of an Arduino Nano board can be done using different sources like using an additional Arduino board, a computer, otherwise using microcontrollers. The microcontroller using in Nano board (ATmega328) offers serial communication (UART TTL). This can be accessible at digital pins like TX, and RX. The Arduino software comprises of a serial monitor to allow easy textual information to transmit and receive from the board.

The TX & RX LEDs on the Nano board will blink whenever information is being sent out through the FTDI & USB link in the direction of the computer. The library-like SoftwareSerial allows serial communication on any of the digital pins on the board. The microcontroller also supports SPI & I2C (TWI) communication.

**Arduino Nano Programming**

The programming of an Arduino nano can be done using the Arduino software. Click the Tools option and select the nano board. Microcontroller ATmega328 over the Nano board comes with preprogrammed with a boot loader. This boot loader lets to upload new code without using an exterior hardware programmer. The communication of this can be done with the STK500 protocol. Here the boot loader can also be avoided & the microcontroller program can be done using the header of in-circuit serial programming or ICSP with an Arduino ISP.

**3.2.2 ULTRA SONIC SENSOR**

****

Fig . 3.3 ultra sonic sensor

Ultrasonic sensors are used as proximity sensors. They can be found in parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems and manufacturing engineering. Compared to infrared (IR) sensors in proximity sensing applications, ultrasonic sensors are less susceptible to interference from smoke, gases, and other airborne particles (although the physical component is subject to variables such as heat).

Ultrasonic sensors are also used as level sensors to detect, monitor, and control liquid levels in closed vessels (such as chemical plant drums). Most notably, ultrasound technology has enabled the medical industry to image internal organs, identify tumours, and ensure the health of babies in the womb.

**PRINCIPLE OF ULTRASONIC SENSOR**

The principle of ultrasonic rangefinders is to measure the time it takes the signal sent by a transmitter and propagated back to the receiver. As the name implies ultrasonic sensor operates on ultrasonic frequencies. Frequencies beyond our hearing range are known as ultrasonic frequencies. Those frequencies are above 20k Hertz.

They are the all-rounders of sensor technology and can be used in any industrial application. There are several types of objects that can be detected, including solids, liquids, granules, and powders. They reliably detect transparent or glossy objects, as well as objects whose colors change.

**HOW ULTRASONIC SENSOR WORKS?**

An ultrasonic sensor is an electronic device that measures the distance to an object by emitting ultrasonic waves and converting the reflected sound into electrical signals. Ultrasound travels faster than audible sound (that is, sound that humans can hear). An ultrasonic sensor consists of two main components: a transmitter (which uses a piezoelectric crystal to emit sound) and a receiver. While some sensors use separate sound emitters and receivers, it is also feasible to merge both functions into a single device by using an ultrasonic element to switch between sending and receiving signals in a continuous cycle. The transmitter of the module transmits an ultrasonic sound. This sound will be reflected if an object is present in front of the ultrasonic sensor. The reflected sound is received by the receiver present in the same module. An ultrasonic signal is propagated by a wave at an angle of 30°. The above-depicted Figure illustrates how the ultrasonic signal propagates from the transmitter. Measuring angles should be at least 15° for maximum accuracy. In this case, external objects that fall under this measurement angle interfere with determining the distance to the desired object. The distance is determined by measuring the travel time of ultrasonic sound and its speed.

**Distance = Time x Speed of sound/2**

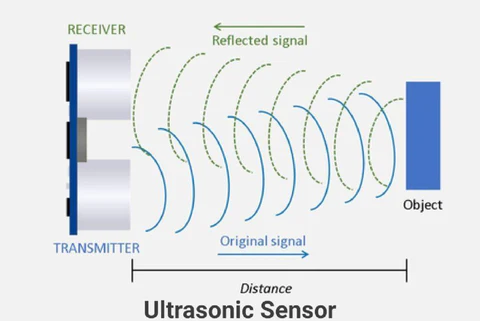


Fig 3.4 ultra sonic sensor working

**Applications of ultra sonic sensor**

* Ultronic anemometer
* Tide guage
* Tank level
* Web guiding systems
* UAV Navigations

**3.2.3 TCS230 Color Sensor**

The TCS3200 color sensor can detect a wide variety of colors based on their wavelength. This sensor is specially useful for color recognition projects such as color matching, color sorting, test strip reading and much more.

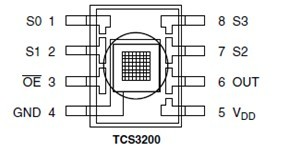
****

Fig. 3.6 color sensor pins

Fig. 3.5 color sensor

The TCS3200 has an array of photodiodes with 4 different filters. A photodiode is simply a semiconductor device that converts light into current. The sensor has:

16 photodiodes with red filter – sensitive to red wavelength

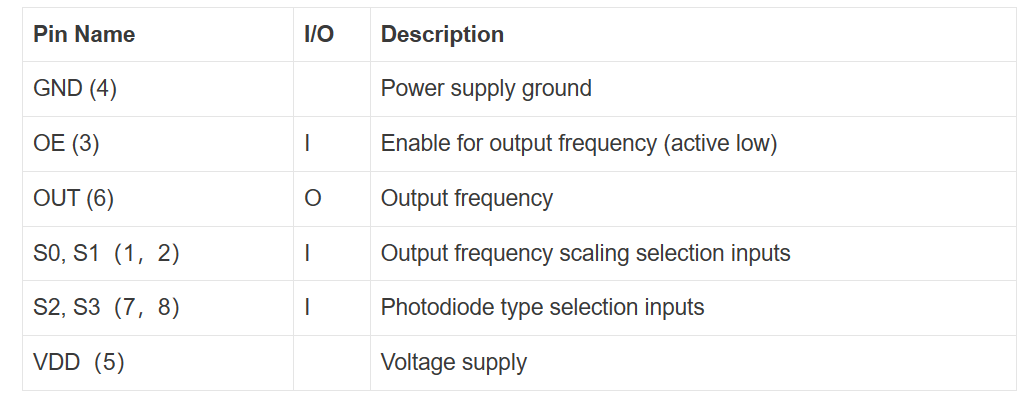
16 photodiodes with green filter – sensitive to green wavelength

16 photodiodes with blue filter – sensitive to blue wavelength

16 photodiodes without filter

If you take a closer look at the TCS3200 chip you can see the different filters.

**PINOUT**

**Table 3.1 color sensor pinout**

**Filter selection**

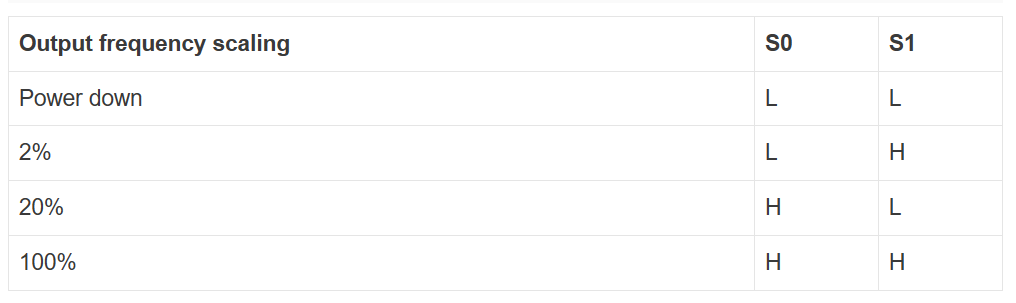
To select the color read by the photodiode, you use the control pins S2 and S3. As the photodiodes are connected in parallel, setting the S2 and S3 LOW and HIGH in different combinations allows you to select different photodidodes. Take a look at the table below:

**Table 3.2 color sensor filter selection**

**Frequency scaling**

Pins S0 and S1 are used for scaling the output frequency. It can be scaled to the following preset values: 100%, 20% or 2%. Scaling the output frequency is useful to optimize the sensor readings for various frequency counters or microcontrollers. Take a look at the table below:

**Table 3.3 frequency scaling**

**3.2.4 DF PLAYER MINI**

The DFPlayer Mini MP3 Player For Arduino is a small and low price MP3 module with an simplified output directly to the speaker. The module can be used as a stand alone module with attached battery, speaker and push buttons or used in combination with an Arduino UNO or any other with RX/TX capabilities

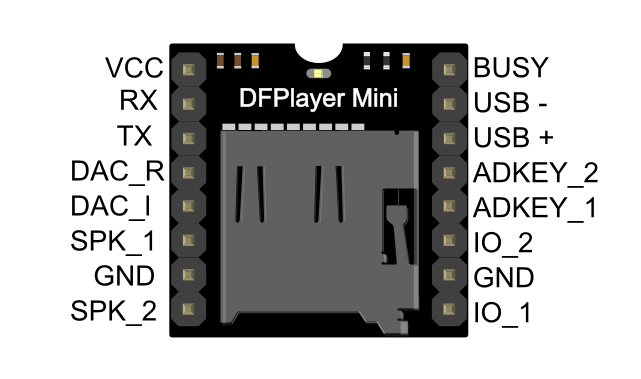
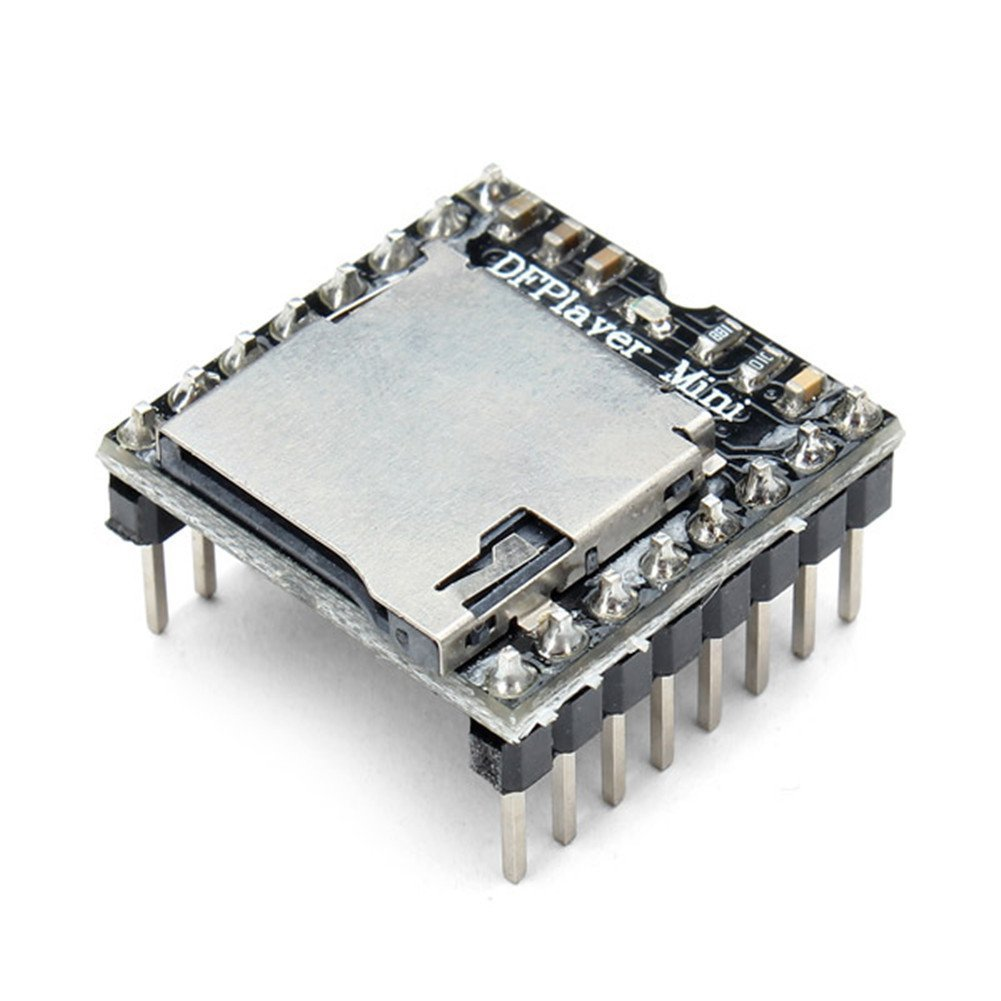
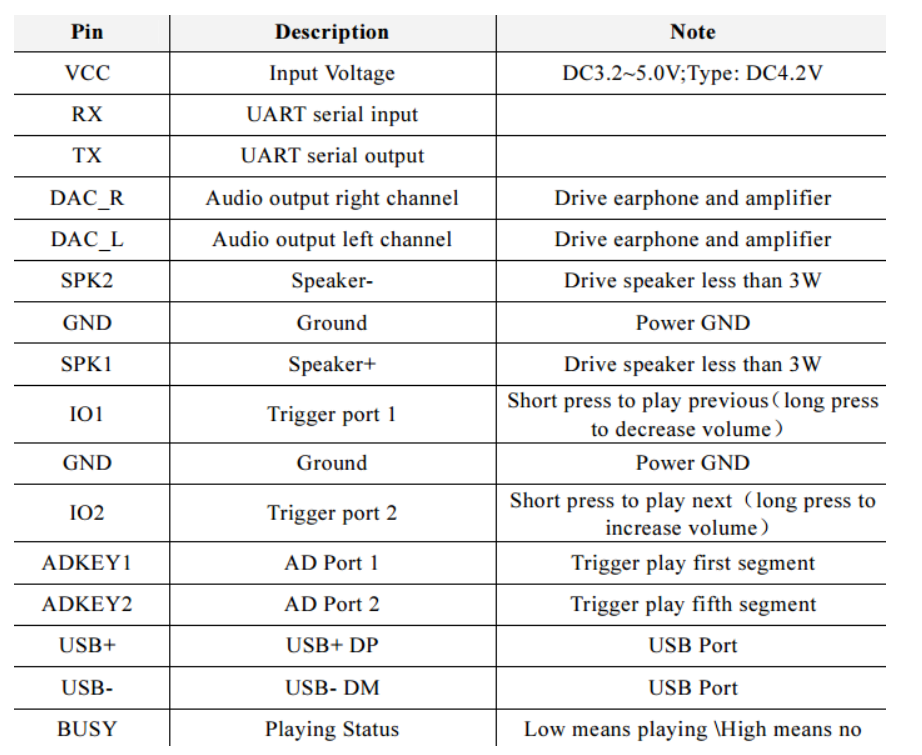
Fig. 3.7 df player mini

Fig 3.8 df player pins

**PIN MAP**



## Table 3.4 DF Player Mini pin map

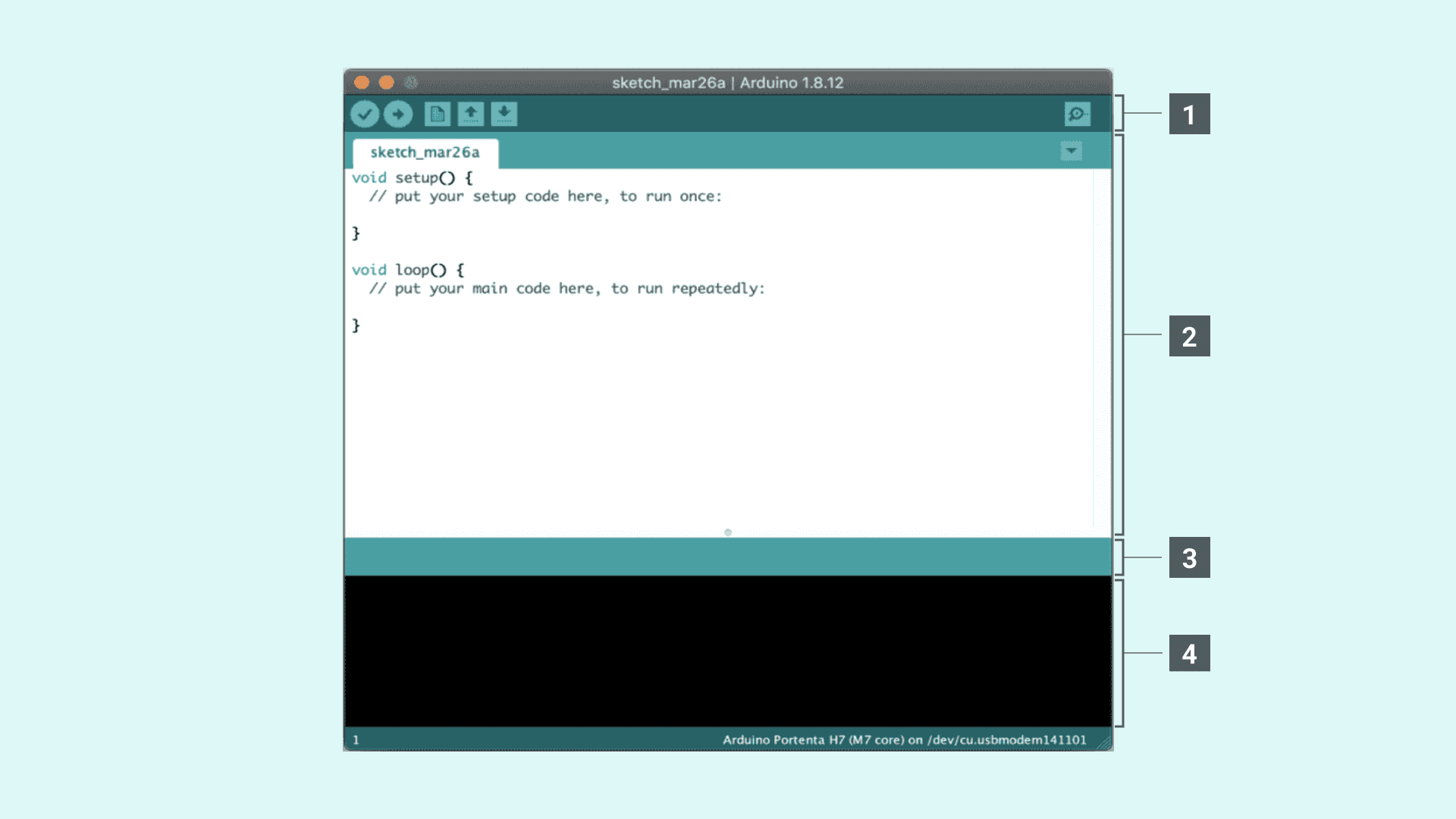
## Application

* Car navigation voice broadcast;
* Road transport inspectors, toll stations voice prompts;
* Railway station, bus safety inspection voice prompts;
* Electricity, communications, financial business hall voice prompts;
* Vehicle into and out of the channel verify that the voice prompts;
* The public security border control channel voice prompts;
* Multi-channel voice alarm or equipment operating guide voice;
* The electric tourist car safe driving voice notices;
* Electromechanical equipment failure alarm;
* Fire alarm voice prompts;
* The automatic broadcast equipment, regular broadcast

**3.3 MAJOR SOFTWARE SPECIFICATIONS**

Software used Aduino IDE

**3.3.1 ARDUINO IDE**



**Fig .3.9 COMMAND WINDOW**

Writing sketches

* File
* Edit
* Sketch
* Tools
* Help

The Arduino Integrated Development Environment or Arduino Software (IDE) contains a text - editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a IY D series of menus. It connects to the Arduino hardware to upload programs and communicate with them. Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with. ino extension on save.

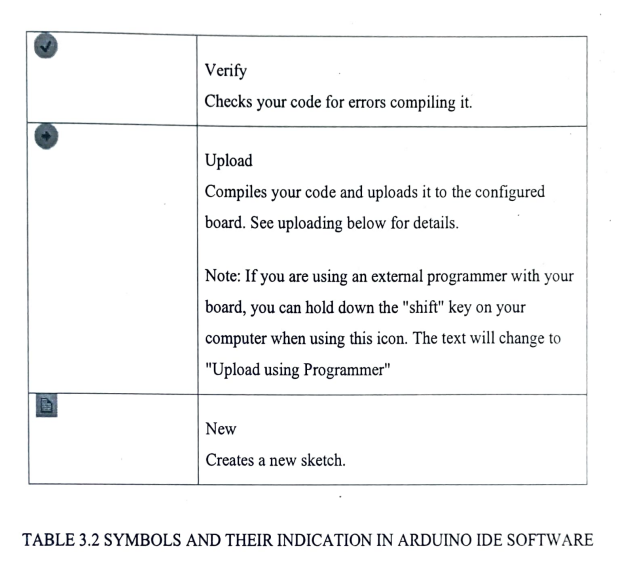


Table 3.5 symbol and their indication in arduino software ide

**File**

**New**

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

**Open**

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

**Recent**

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

**Sketch book**

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

**Tools**

**Auto Format**

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

**Archive**

Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch. Fixes possible discrepancies between the editor char map encoding and other operating systems char maps. Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

**Board**

Select the board that you're using. See below for descriptions if the board.

**Port**

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

**Programmer**

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

**Libraries**

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

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. To write your own library,watch the tutorial.

**Third-Party Hardware**

⚫ Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "Arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the Arduino third party specifications.

⚫ Arduino Software (IDE) includes the built-in support for the boards in the following list, all based on the AVR Core. The board manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

**SOFTWARE TIPS**

When boot loading an Atmega8 chip with Arduino 0010, there is a command (-1800) that makes bootloader delay 10 minutes. So, if you need to use bootloader, use command line instead of IDE, removing "-1800" command and adding "-F" command, or use Arduino 0007 IDE. To upload sketches Arduino 0010 works fine.

**3.4 BLOCK DIAGRAM**

Diagram

Description automatically generated

**3.5 Process of the project**

The ultrasonic sensor and the buzzer is connected to the processing unit.The control unit controls the ultrasonic sensors and get the information of the obstacle present in front of the man and processes the information and sends the output through the buzzer accordingly.

This ultrasonic sensor can detect or sense the obstacles which are less than 300 cm and create sound through output device buzzer.

If the distance is greater than 3 meters then it will not sense.

**Chapter 4**

**RESULTS**

The program for the Arduino NANO microcontroller was written in C language and was then compiled into an executable file using the Arduino IDE. The executable file was then imported into the Proteus Design Suite, where the hardware circuit shown was designed and simulated.the Proteus simulation of the smart glasses results for detecting the objects by using the ultra sonic sensor which are below 400meters respectively. Upon successful completion of the software simulation, the project hardware was constructed on a bread board and programming of the Arduino microcontroller was carried out using Arduino IDE. The hardware construction with connections and various operations of the system are shown in the figure below. The response of the hardware of the smart glasses when communicating with the Arduino board. The system is designed to detect the number of objects wich comes in the of ultrasonic sensor.

**CHAPTER 5**

**CONCLUSION**

**5.1 conclusion**

The system has the ability to detect objects which comes in the way to the blind person, The system is developed using Arduino NANO development board and it's IDE. The project uses two Ultrasonic sensors.buzzer and mini vibration motor are used to intimate the user about the object. Arduino helps achieve the coordination among the sensors. This project is based on Arduino, and the coding is done on Arduino ide platform using the Arduino application. At the end of this research the aim and objectives of the project was achieved.This smart glass implemented for blind person who are unable to see any object so this person can aware about accident.It is safe to say that the main objectives and the aim of the project were achieved at the end of the project.

**5.2 Future scope**

* Face recognition can added to know the face gestures of the third party person
* Hand gestures can be added so that the user can know the hand gestures
* Color blindness features
* In future it can be implemented as a image recognization where sensor give information user about the object
* A personal Artificial Intelligence assistant can be added so that it can help or guide the blind person in every situation

**REFERENCES**

[1]Feng Lan, Guangtao Zhai, Wei Lin “Lightweight smart glass system with audio aid for visually impaired people” ,TENCON,IEEE,Region 10 Conference, 2015.

[2]ASM Iftekhar Anam, Sahinur Alam, Md Yeasin “A dyadic conversation aid using google glass for people who are blind or visually impaired”, Mobile Computing Applications and Services(MobiCASE),6th International Conference, 2015.

[3]Tailong Shi, Bruce, Ting-Chia Huang “Design, Demonstration and characterization of Ultra-Thin Low-Warpage Glass BGA Packages for smart mobile Application processor”, Electronics Components and Technology Conference(ECTC),2016 IEEE 66th ,2016.

[4]S. Pinto, T. Castro, N. Brito “ClimaWin: An intelligent window for optimal ventilation and minimum thermal loss”, Industrial Electronics(ISIE),2013 IEEE International Symposium,2013.

[5]W.C.S.S. Simoes, V.F.de Lucena “Blind user wearable audio assistance for indoor navigation based on visual markers and ultrasonic obstacle detection”, Consumer Electronics(ICCE),2016 IEEE International Conference,2016.

[6]Md Sheikh Sadi, Saifudin Mahmud, Md Mostafa Kamal, Abu Ibne Bayazid “Automated walk-in assistant for blinds”, Electrical Engineering and Information and Communication Technology(ICEEICT),2014 International Conference,2014.

[7]Dariush Forouher, Marvin Grobe Besselmann, Erik Maehle,”Sensor Fusion Of Depth camera and ultrasound data for obstacle detection and robot navigation”, Control,Automation,Robotics and vision(ICARCV),2016 14th Internatonal Conference,2016.

[8]T.O.Hoydal,J.A.Zelano,”An alternative mobility aid for the blind :the ultrasonic cane”, Bioengineering Conference Proceedings of the 1991 IEEE Seventeenth Annual NorthEast,1991.

[9]Chinese Author “An intelligent auxiliary system blind glasses”, CN106937909A, 11th July,2017.

[10]Humberto Orozco Cervantes “Intelligent glasses for he visually impaired”, US20150227778A1,13th Aug,2015. [11]Hsieh Chishenng “Electronic talking stick for blind”, US5097856A,24th March,1992

**APPENDIX 1**

**PROJECT CODE**

**CODE 1**

#define trigPin1 5

#define echoPin1 6

#define motor1 7

#define buzzer1 8

#define trigPin2 9

#define echoPin2 10

#define motor2 11

#define buzzer2 12

long duration, distance, UltraSensor1, UltraSensor2;

char data;

String SerialData="";

void setup()

{

Serial.begin (9600);

pinMode(trigPin1, OUTPUT);

pinMode(echoPin1, INPUT);

pinMode(motor1, OUTPUT);

pinMode(buzzer1, OUTPUT);

pinMode(trigPin2, OUTPUT);

pinMode(echoPin2, INPUT);

pinMode(motor2, OUTPUT);

pinMode(buzzer2, OUTPUT);

digitalWrite(motor1,LOW);

digitalWrite(motor2,LOW);

digitalWrite(buzzer1,LOW);

digitalWrite(buzzer2,LOW);

}

void loop()

{

SonarSensor(trigPin1, echoPin1);

UltraSensor1 = distance;

SonarSensor(trigPin2,echoPin2);

UltraSensor2 = distance;

while(Serial.available())

{

delay(10);

data=Serial.read();

SerialData+=data;

}

if(SerialData=="display distance")

{

Serial.print("distance measured by the first sensor: ");

Serial.print(UltraSensor1);

Serial.println(" cm");

Serial.print("distance measured by the second sensor: ");

Serial.print(UltraSensor2);

Serial.println(" cm");

Serial.println("---------------------------------------------------------------------------------------------------------");

}

SerialData="";

if(UltraSensor1 <=70)

{

digitalWrite(buzzer1,HIGH);

digitalWrite(motor1,HIGH);

}

else

{

digitalWrite(buzzer1,LOW);

digitalWrite(motor1,LOW);

}

if(UltraSensor2 <=70)

{

digitalWrite(buzzer2,HIGH);

digitalWrite(motor2,HIGH);

}

else

{

digitalWrite(buzzer2,LOW);

digitalWrite(motor2,LOW);

}

}

void SonarSensor(int trigPinSensor,int echoPinSensor)

{

digitalWrite(trigPinSensor, LOW);

delayMicroseconds(2);

digitalWrite(trigPinSensor, HIGH);

delayMicroseconds(10);

digitalWrite(trigPinSensor, LOW);

duration = pulseIn(echoPinSensor, HIGH);

distance= (duration/2) / 29.1;

}