THIRD EYE FOR BLIND

PROJECT REPORT

Submitted by

ABISHEK.S (U17EE031)
BALAGANESAN.M (U17EE706)
ABISHEK .C (U17EE050)

In Partial fulfilment for the award of degree

Of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

BHARATH INSTITUTE OF HIGHER EDUCATION AND RESEARCH

BHARATH INSTITUTE OF SCIENCE AND TECHNOLOGY

CHENNAI – 600073, INDIA

APRIL – 2021

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BHARATH INSTITUTE OF HIGHER EDUCATION AND RESEARCH

BHARATH INSTITUTE OF SCIENCE AND TECHNOLOGY

CHENNAI – 600073, INDIA

BONAFIDE CERTIFICATE

Certified that this project report titled **“THIRD EYE FOR BLIND”** is the bonafide work of ABISHEK.S (U17EE706) and BALAGANESAN.M (U17EE706) and ABISHEK.C (U17EE050)who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. S.PRAKASH	**Dr. S.PRAKASH**
HEAD OF THE DEPARTMENT	HEAD OF THE DEPARTMENT
Department of EEE	Department of EEE
Bharath Institute of higher education and research	Bharath Institute of higher education and research
Chennai 600 073	Chennai 600 073

Submitted for the project viva-voce examination held on………………..

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We express our heartiest thanks to **Dr. J. SANDEEP AANAND, President, BIHER**, for this kind encouragement and support to carry out this project.

We express our heartiest thanks to, **Dr. V. KANAGASABAI, Vice-chancellor, Dr.M.PONNAVAIKKO, provost, BIHER, Dr.R.M.SURESH, Pro Vice-chancellor academics and Dr. R. HARIPRAKASH, Additional Registrar, BIHER**, for their constant support, great enthusiasm and perpetual motivation.

We convey our sincere thanks to **Dr. J. HAMEED HUSSIAN, Dean Engineering** and **Dr. R. VENKATESHBABU, Dean Academics, BIHER**, for their constant encouragement.

We extend our hearty thanks to **Dr. S.PRAKASH, HOD, Department of Electrical & Electronics Engineering, BIHER,** for his inspiring guidance, meticulous supervision and sincere advice through the project.

We are thankful to our guide**, Dr. S.PRAKASH, HOD , EEE Department, BIHER** for his motivation guidance through the courses of this training work.

We are thankful to our guide**, Dr. S.PRAKASH**,**HOD**, **EEE Department, BIHER** for his motivation guidance through the courses of this training work.

We sincerely wish to express our thanks to all our STAFF MEMBER‘S of the Department of **Electrical & Electronics Engineering, BIHER** and friends for their valuable support. We would like to express our whole hearted thanks to the almighty and our parents first The blessing of whom enable us to finish this project successful.

ABSTRACT

According to WHO 39 million people are estimated as blinds worldwide. They are suffering a lot of hardships in their daily life. For every one minute a death swoops in because of unpredictable and unexpected accidents. To save a life is auspicious as well as precious. The idea of Third eye for the blind is an innovation with the help of the multidiscipline subjects like computer science, electronics engineering and health science which helps the blind people to navigate with speed and confidence by detecting the nearby obstacles using the help of ultrasonic waves and notify them with a buzzer sound or vibration. When the sensor detects any object it will notify the user by beep or vibration. This is an automated device. This device will be of a great use for the blinds and help them travel different places.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

VCC	**-**	**VOLTAGE COMMON COLLECTOR**
GND	**-**	**GROUND**
LED	**-**	**LIGHT EMITING DIODE**
UART	**-**	**UNIVERSAL ASYNCHRONOUS**
TXD	**-**	**TRANSMIT DATA**
RXD	**-**	**RECIVE DATA**

CHAPTER 1

INTRODUCTION

1.1 Objectives

The objectives of this paper is an innovation with the help of the multidiscipline subjects like computer science, electronics engineering and health science which helps the blind people to navigate with speed and confidence by detecting the nearby obstacles using the help of ultrasonic waves and notify them with a buzzer sound or vibration. According to WHO 39 million people are estimated as blinds worldwide. They are suffering a lot of hardships in their daily life. The affected ones have been using the tradition white cane for many years which although being effective, still has a lot of disadvantages. This will be a wearable technology for the blinds. One of the main peculiarities of this device is that it will be affordable. The Arduino Pro Mini 328- 15/16 MHz board is worn like a device. This will be equipped with ultrasonic sensors, consisting of module. Using the sensor, visually impaired can detect the objects around them and can travel easily. When the sensor detects any object, it will notify the user by beep or vibration. Thus, this is an automated device.

1.2 General Introduction

Since the running of daily life of blind people is very difficult. This project helps them to run their life as usual. They can make this project as a gadget or a device in their hands which detects the obstacle. This project is more efficient than the existing system with cheaper and accurate one. Here we are using Arduino UNO board to perform this operation. To make the life to be as a normal one or the blind peoples this may be very helpful project for them. By making this as a gadget or a device in their hand they can easily judge an object by their own by knowing the buzzer sound. The system uses ultrasonic sensor as a wide range of field to detect an object with its higher detection range.

The first wearable technology for people who are blind. Using ultrasonic waves to detect the obstacles. Notifying the user through vibrations/buzzer sound. Third eye for people who are blind is an innovation which helps the blind people to navigate with speed and confidence by detecting the nearby obstacles using the help of ultrasonic waves and notify them with buzzer sound or vibration. They only need to wear this device as a band or cloths. According to WHO 39 million peoples are estimated as blind worldwide. They are suffering a lot of hardship in their daily life. The affected ones have been using the traditional white cane for many years which although being effective, still has a lot of disadvantages. Another way is, having a pet animal such as a dog, but it is really expensive. So, the aim of the project is to develop a cheap and more efficient way to help visually impaired to navigate with greater comfort, speed and confidence.

Third eye for the blind using ultrasonic sensor is a special wearable device based on the Arduino board which can be worn like a cloth for blinds. This device is equipped with five ultrasonic sensors, consisting of five modules which are connected to the different parts of the body. Among them, two for both shoulders, another two for both knees and one for the hand. Using the five ultrasonic sensors, blind can detect the objects in a five-dimensional view around them and can easily travel anywhere. When the ultrasonic sensor detects obstacle, the device will notify the user through vibrations and sound beeps. The intensity of vibration and rate of beeping increases with decrease in distance and this is a fully automated device.

I have designed a special wearable device based on the Arduino board which can be worn like a cloth for blinds. This device is equipped with five ultrasonic sensors, consisting of five modules which are connected to the different parts of the body. Among them, two for both shoulders, another two for both knees and one for the hand. Using the five ultrasonic sensors, blind can detect the objects in a five-dimensional view around them and can easily travel anywhere. When the ultrasonic sensor detects obstacle, the device will notify the user through vibrations and sound beeps.

1.3 Previous Methodology

The existing system consists of the devices or the supports like white cane for helping them to detect the obstacles and travel to places, pet dogs, smart devices like vision a torchwork blinds. But there were many limitations and problems in these existing systems like in the white cane, it may easily break or crack. The white cane may get stuck at the pavement cracks of the different objects. Whereas the pet dogs' cost is huge and need a lot of training. They are suffering a lot of hardship in their daily life.

The physically disabled ones have been using the traditional way that is the white cane for many years which although being effective, still has a lot of disadvantages and limitations. Another way is, having a pet animal such as a dog, but it is really expensive. Thus, the aim of the project Third eye for the Blind is to develop a cheap, affordable and more efficient way to help the blind people to navigate with greater comfort, speed and confidence. This is the wearable technology for the blinds which helps resolve all the problems of the existing technologies. Now a days there are so many technologies, things and smart devices for the visually impaired people for the navigation, but most of them have certain problems for the blind people and the major drawbacks are that those things need a lot of training and efforts to use. One of the main peculiarities of this innovation is, it is affordable for everyone, the total cost being less than $25 or ~1500 INR. There are no such devices available in the market that can be worn like a cloth and having such a low cost and simplicity. With the use of this improvised device in a large scale, with improvements in the prototype, it will drastically benefit the community of the visually impaired or the blind people. The walking cane is a simple and purely mechanical device dedicated to detect the static or the constant obstacles on the ground, uneven surfaces, holes and steps via simple tactile-force feedback. This device is light, portable but limited to its size and it is not used for dynamic obstacle detection. These devices operate like the radar and the system of the device uses the ultrasonic waves fascicle to identify the height, direction and the speed of the objects. The distance between the person and the obstacle is measured by the time of the wave travel. However, all the existing systems inform the blind the presence of the object at a specific distance in front of or near to him. These details help the user or the blind people in detecting the obstacles and thus change the way and walk accordingly. Information about the objects and their place in the way of the walking like an obstacle and their characteristics can create additional knowledge to enhance the space manifestation and memory of the blind or the visually impaired people. To overcome, the above-mentioned limitations this work offers a simple, efficient, configurable virtual for the blind.

1.4 Limitations

- Not designed for underwater use: ultrasonic sensors get spoil when use underwater, this means that the user is not permitted to use this device when it is raining.

- Sensing accuracy is affected by changes in temperature of 5-10 degrees or more: under-temperature or higher temperature affect the system operation. Most ultrasonic sensors have a working range of -25° C to +70° C.

- Have a limited detection range: ultrasonic sensor has a maximum range of 10 meters.

1.5 Proposed Systems

The design is based on a special wearable device based on the Arduino board which can be worn like a cloth for blinds or a band. This device is equipped with five ultrasonic sensors, consisting five modules which are connected to the different parts of the body. Among them, two for both the shoulders, another two for both the knees, and one for the hand. It's the choice of the visually impaired people, they can either use one band or put it anywhere on their body wherever they are comfortable. With the use of these five ultrasonic sensors in the device and by wearing it on the body, the blind can detect the objects in a five-dimensional view around them and can easily travel anywhere by detecting the obstacles. When the ultrasonic sensor detects obstacle, the device will notify the user through vibrations and sound beeps. The intensity of vibrations and the rate of beeping increases with decrease in distance and this is a fully automated device. The features of the Third Eye for Blind will help the visually impaired people in many ways. By wearing this device, they can fully avoid the use of the white cane and such other devices. This device will help the blind to navigate without holding a stick which is a bit annoying for them. They can wear the device as a band or like a cloth and it can function very accurately and they only need a very little training to use it as it is quite simple, efficient and easy to operate and wear.

1.6 Working Operations

Proposed system consists the equipment like Arduino mini pro, ultrasonic sensor, prefboard, vibrating motor, buzzers for detecting the obstacles and letting the user know about the obstacle, Red LEDs, Switches, Jumper cable, power bank, Male and female header pins, 3.3-volt old mobile battery which is unused or discarded, some elastic and stickers to make the device wearable as a band for wearing for the users.

The wiring of the device is done in a following manner. The Ground of LED, buzzer and vibration motor are connected to GND of the Arduino. The +ve of the LED and the middle leg of switch is connected to the Arduino pin 5. The +ve of the Buzzer is wired to the first leg of the switch and the +ve of the Vibration motor is wired to the third leg of the switch. The Ultrasonic sensor are wired accordingly. The Ultrasonic sensor pin VCC is connected to the Arduino pin VCC, Ultrasonic sensor pin GND is connected to the Arduino pin GND, Ultrasonic sensor pin Trig is attached to the Arduino pin 12, Ultrasonic sensor pin Echo is connected to the Arduino PIN 12. The switch used here is for selecting the mode. (Buzzer or vibration mode.) We first cut the prefboard in 5 X 3 cm dimension and solder the female headers for the Arduino to the board. Then soldering of the buzzer is carried out. Then using the glue connect the vibrating motor and solder the wires to it. Then connection of the LED is done. Then connect the switch. Connect the header pins for ultrasonic sensors and for the battery input. Then solder all the things and connect the Arduino and ultrasonic sensor to the board. Also connect the elastic band to all the modules. For making the module for the hand, connect the ultrasonic sensor to the board by using 4 jumper cables. Then connect a 3.7 velomobile battery to this module. Then connect the elastic band. In the end after all the connections are done to the Arduino board, upload the code to each Arduino board and power the 4 other modules using a power bank.

1.7 Introduction to Embedded System

Microcontrollers are widely used in Embedded System products. An Embedded product uses the microprocessor (or microcontroller) to do one task & one task only. A printer is an example of embedded system since the processor inside it performs one task only namely getting the data and printing it. Contrast this with Pentium based PC. A PC can be used for any no. of applications such as word processor, print server, bank teller terminal, video game player, network server or internet terminal. Software for variety of 14applications can be loaded and run. Of course, the reason a PC can perform multiple tasks is that it has RAM memory and an operating system that loads the application software into RAM & lets the CPU run it. In and Embedded system there is only one application software that is typically burn into ROM. An x86PC Contain or it's connected to various Embedded Products such as keyboard, printer, modem, Disc controller, Sound card, CD-ROM Driver, Mouse & so on. Each one of these peripherals as a microcontroller inside it that performs only one task.

For example, inside every mouse there is microcontroller to perform the task of finding the mouse position and sending it to PC.
Although microcontroller is preferred choice for many embedded systems, there are times that a microcontroller is inadequate for the task. For this reason, in recent years many manufactures of general-purpose microprocessors such as INTEL, Motorola, and AMD & Cyrix have targeted their microprocessors for the high end of embedded market. While INTEL, AMD, Cyrix push their x86 processors for both the embedded and desktop pc market, Motorola is determined to keep the 68000 families alive by targeting it mainly for high end of embedded system. One of the most critical needs of the embedded system is to decrease power consumptions and space. This can be achieved by integrating more functions into the CPU chips. All the embedded processors based on the x86 and 680x0 have low power consumptions in additions to some forms of I/O, Com port & ROM all on a single chip. In higher performance Embedded system, the trend is to integrate more & more function on the CPU chip & let the designer decide which feature he/she wants to us.

EMBEDDED SYSTEM

An Embedded System employs a combination of hardware & software (a ―computational engine‖) to perform a specific function; is part of a larger system that may not be a ―computer works in a reactive and time-constrained environment. Software is used for providing features and flexibility Hardware = {Processors, ASICs, Memory...} is used for performance (& sometimes security
An embedded system is a special purpose system in which the computer is completely encapsulated by the device it controls. Unlike a general-purpose computer, such as a PC, an embedded system performs predefined task ‘s usually with very specific tasks design engineers can optimize it reducing the size and cost of the product. Embedded systems are often mass produced, so the cost savings may be multiplied by millions of items.

The core of any embedded system is formed by one or several microprocessor or micro controller programmed to perform a small number of tasks. In contrast to a general-purpose computer, which can run any software application, the user chooses, the software on an embedded system is semi-permanent, so it is often called firmware.

1.8 Importance of the project

This is the first wearable technology for blinds which resolves all the problems of existing technologies. Now a days there are so many instruments and smart devices for visually impaired peoples for navigation but most of them have certain problems for carrying and the major drawbacks is those need a lot of training to use. The one of the main peculiarities of this innovation is, it is affordable for everyone. There are no such devices available in the market that can be worn like a cloth and having such a low cost and simplicity. When used on a large scale, with improvements in the prototype, it will drastically benefit the community. This device:

- Is the first wearable technology for blinds.
- Uses ultrasonic waves to detect the obstacles
- Notifying the user through vibrations/buzzer sound

1. By wearing this device, they can fully avoid the use of white cane and such other devices.
2. This device will help the blind to navigate without holding a stick which is a bit annoying for them.
3. They can simply wear it as a band or cloth and it can function very accurately and they only need a very little training to use it.

CHAPTER 2

LITERATURE SURVEY

2.1 Smart walking stick

Author: Mohammed H. Rana and Sayemil

This is based on ping sensor for detecting obstacle, wet electrode, vibration motor and the buzzer. The obstacle is detected by the ping sensor and the obstacle distance is communicates to the visually impaired by the vibration of the motor.

2.2 The electronic travelling aid for blind navigation and monitoring

Author: Mohan M.S Madulika

This is arm7 controller based that used ultrasonic technology for detecting the obstacle and inform the obstacle distance to the visually impaired, and also used the GPS and GSM technologies for localization of the visually impaired.

2.3 Haptic shoe for the blind

Author: Oladayo, Robert Dunne

A haptic device that can be installed in a shoe vibrating alert feature benefit for deafness. This device receives GPS information from a smartphone and provides vibration feedback at the right, left, front and back for the shoe in order to provide guidance to a destination. A proximity sensor in the front of the shoe can detect objects up to 3 metres and provide vibrational feedback.

2.4 Multi-dimensional walking aid

Author: Olakanmi. O. Oladayo

This system uses ultrasonic detection technology and the voice, the obstacle is detected by the ultrasonic sensor and the direction of the obstacle is communicates to the module user through voice output

2.5 3D ultrasonic stick for the blind

Author: Osama Bader Al-Barm

The system uses ultrasonic sensor for detecting the obstacle in three directions (i.e., front, left and the right sides of the visually impaired), and the vibration motor which vibrate with the intensity depending on the obstacle's distance. It also uses GPS and GSM for localization of the visually impaired [6]. In this paper, design and development of intelligent electronic travelling aid for visually impaired is presented. The device employs ultrasonic detection, GSM, GPS, voice recognition and voice synthesis technologies. The design process comprises of two parts; the first part is the obstacle detection and voice generation unit design using ultrasonic detection technology and voice synthesis technology respectively, and the second part is the localization and monitoring unit design using GPS technology GSM technology, as well as the voice recognition technology. The two units are then combining to form the complete device.

CHAPTER 3

PROJECT DESCRIPTION

3.1 INTRODUCTION

The objective of this project The Third Eye for the Blind is to design a product which is very much useful to those people who are visually impaired and those who often have to rely on others. Third eye for Blind project is an innovation which helps the visually impaired people to move around and go from one place to another with speed and confidence by knowing the nearby obstacles using the help of the wearable band which produces the ultrasonic waves which notify them with buzz sound or vibrations. It allows the user those who are visually impaired to walk freely by detecting the obstacles. They only need to wear this device as a band or cloth on their body. With the improvement of the living standards of the people, we have become so materialistic that we have forgotten how the physically disabled people live a tough life. They undergo rigorous, apathetic and indifferent behaviour towards them for being physically disabled. They become dependent on other people in a way for their day-to-day routine chores. Blind and impaired persons always depend on other people for their locomotion. Eye are prime sense of organ in perceiving the outside environment; dysfunction of such prime sense organ severely effects the knowledge perceiving capability of the outside environment. Therefore, going around to places in such environment is a very big challenge because the blind people cannot depend on their own eyes and thus face many difficulties.

3.2 SCOPE

1. The blind can detect the objects in a five-dimensional view around them and can easily travel anywhere by detecting the obstacles. When the ultrasonic sensor detects obstacle, the device will notify the user through vibrations and sound beeps.
2. The intensity of vibrations and the rate of beeping increases with decrease in distance and this is a fully automated device
3. The features of the Third Eye for Blind will help the visually impaired people in many ways.
4. By wearing this device, they can fully avoid the use of the white cane and such other devices. it is quite simple, efficient and easy to operate and wear.

3.3 Methodology

3.3.1 Arduino UNO

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. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analogue input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

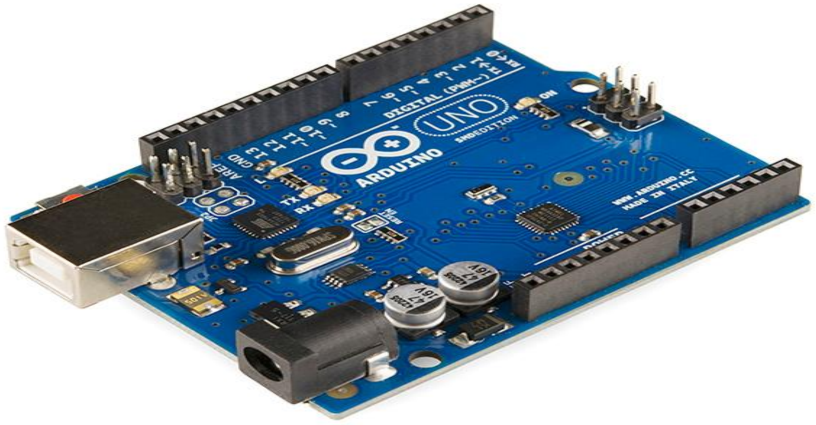


Fig 3.1: Arduino UNO

Arduino can be used to communicate with a computer, another Arduino board or other microcontrollers. The ATmega328P microcontroller provides UART TTL (5V) serial communication which can be done using digital pin 0 (Rx) and digital pin 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 ATmega16U2 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. There are two RX and TX LEDs on the Arduino board which will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328P also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

3.3.2 Arduino board use

GENERAL DESCRIPTION

The 14-digital input/output pins can be used as input or output pins by using pin Mode (), digital Read () and digital Write () functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 Ohms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

Serial Pins 0 (Rx) and 1 (Tx)

Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

External Interrupt Pins 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.

PWM Pins 3, 5, 6, 9 and 11

These pins provide an 8-bit PWM output by using analogWrite() function.

SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)

These pins are used for SPI communication.

In-built LED Pin 13

This pin is connected with a built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analogue input pins, each of which provide 10 bits of resolution, i.e., 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with analogue Reference () function.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below:

AREF

Used to provide reference voltage for analogue inputs with analogReference() function.

Reset Pin

Making this pin LOW, resets the microcontroller.

3.3.3 Upload Code to Arduino Pro mini using Arduino Uno

Connecting Pro mini we first need to remove the AT mega 328 chip from the UNO board. Take a flat head screw-driver and gently place it under the IC and pry it slowly now do the same from other side and the IC should pop out of the socket. Once the IC is out of the socket we can now move ahead and start making the connections. Now connect the pro mini to UNO.

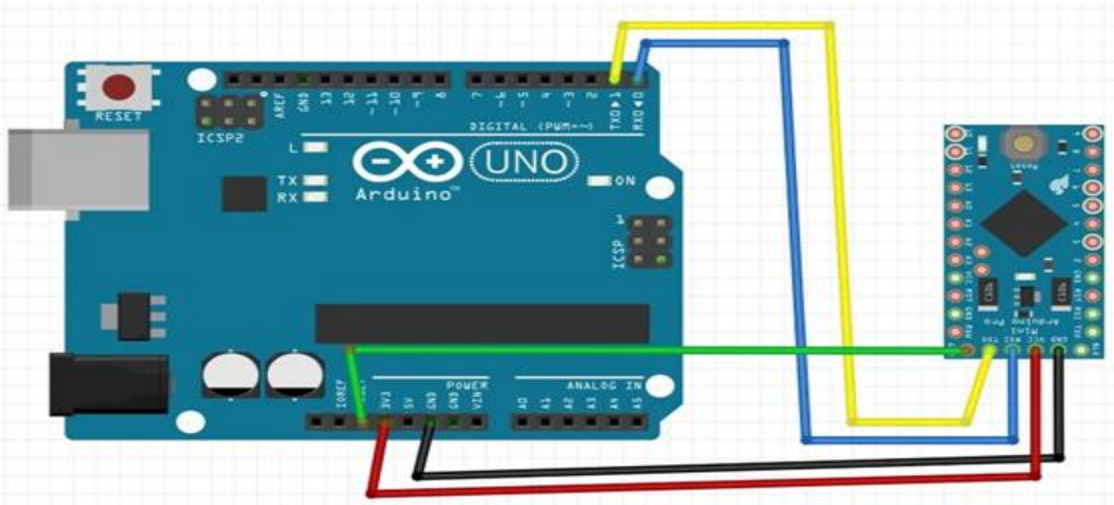


Fig 3.2: Circuit diagram to connect Arduino UNO to Arduino Pro mini

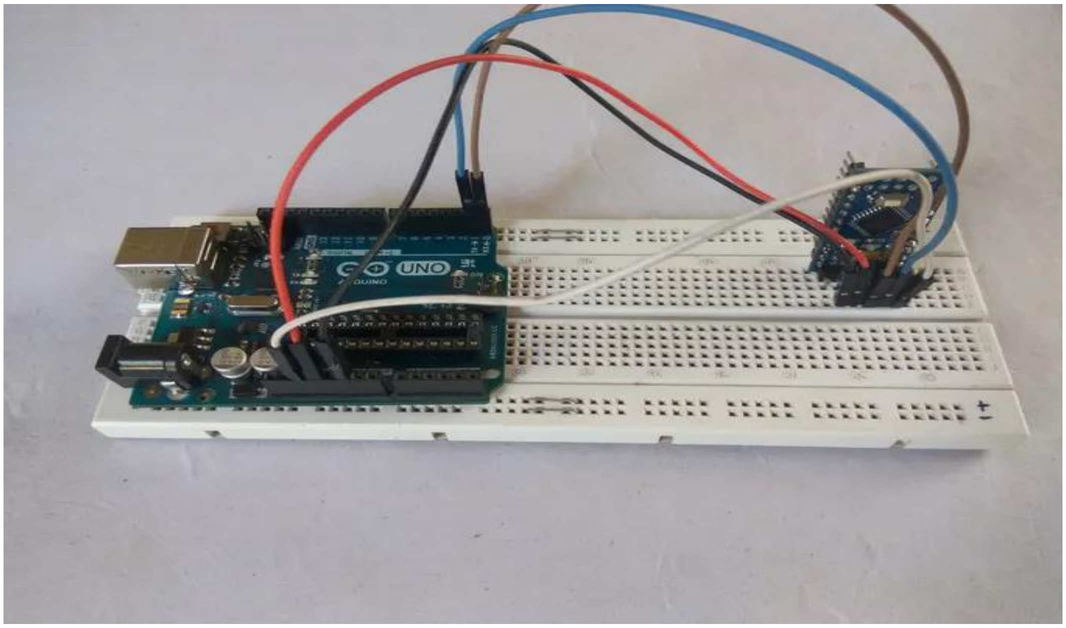


Fig 3.3: Arduino UNO to Arduino Pro mini connection in the bread board

In some cases, the pro mini may not be programmed in that case just swap the TX & RX pins. Connect UNO to PC then Open the Arduino IDE and Upload the code to the Arduino pro mini using Arduino UNO.

3.4 Ultrasonic Sensors

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound.

The Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse. It sends an ultrasonic pulse out at 40kHz which travels through the air and if there is an obstacle or object, it will bounce back to the sensor. By calculating the travel time and the speed of sound, the distance can be calculated. Ultrasonic sensors are a great solution for the detection of clear objects. For liquid level measurement, applications that use infrared sensors, for instance,

struggle with this particular use case because of target translucence. For presence detection, ultrasonic sensors detect objects regardless of the colour, surface, or material. To detect transparent and other items where optical technologies may fail, ultrasonic sensors are a reliable choice
Distance measurement is based on the measurement of time-of-flight. The time between sending and receiving the reflected sound signal is calculated by the sensor. Ultrasonic distance sensors, like the MB7360 HRXL-MaxSonar-WR, are used as height monitors, in bin level measurement and proximity zone detection applications.

Popular applications that we have worked on have been using our sensors in kiosks to detect the presence of a person approaching and detecting the presence of people in an environment, like an art installation.

3.5 EXISTING SYSTEM:

The existing system consists of the devices or the supports like white cane for helping them to detect the obstacles and travel to places, pet dogs, smart devices like vision a torchwork blinds. But there were many limitations and problems in these existing systems like in the white cane, it may easily break or crack. The white cane may get stuck at the pavement cracks of the different objects. Whereas the pet dogs' cost is huge and need a lot of training.

DISADVANTAGES:

1. White cane - May easily crack/break, the stick may get stuck at pavement cracks of different objects.
2. Pet dog - Huge cost.
3. Time Common Disadvantages (Including the smart devices) Cannot be carried easily, needs a lot of training to use

3.6 PROPOSED SYSTEM:

The proposed system is based on a special wearable device based on the Arduino board which can be worn like a cloth for blinds or a band. This device is equipped with five ultrasonic sensors, consisting five modules which are connected to the different parts of the body. Among them, two for both the shoulders, another two for both the knees, and one for the hand.

It’s the choice of the visually impaired people, they can either use one band or put it anywhere on their body wherever they are comfortable.

With the use of these five ultrasonic sensors in the device and by wearing it on the body, the blind can detect the objects in a five-dimensional view around them and can easily travel anywhere by detecting the obstacles. When the ultrasonic sensor detects obstacle, the device will notify the user through vibrations and sound beeps. The intensity of vibrations and the rate of beeping increases with decrease in distance and this is a fully automated device. The features of the Third Eye for Blind will help the visually impaired people in many ways. By wearing this device, they can fully avoid the use of the white cane and such other devices. This device will help the blind to navigate without holding a stick which is a bit annoying for them. They can wear the device as a band or like a cloth and it can function very accurately and they only need a very little training to use it as it is quite simple, efficient and easy to operate and wear.

ADVANTAGES:

1. Blind people live can be saved.
2. By wearing this device, they can fully avoid the use of white cane and such other devices.
3. This device will help the blind to navigate without holding a stick which is a bit annoying for them.
4. They can simply wear it as a band or cloth and it can function very accurately and they only need a very little training to use it.

3.7 BLOCK DIAGRAM:

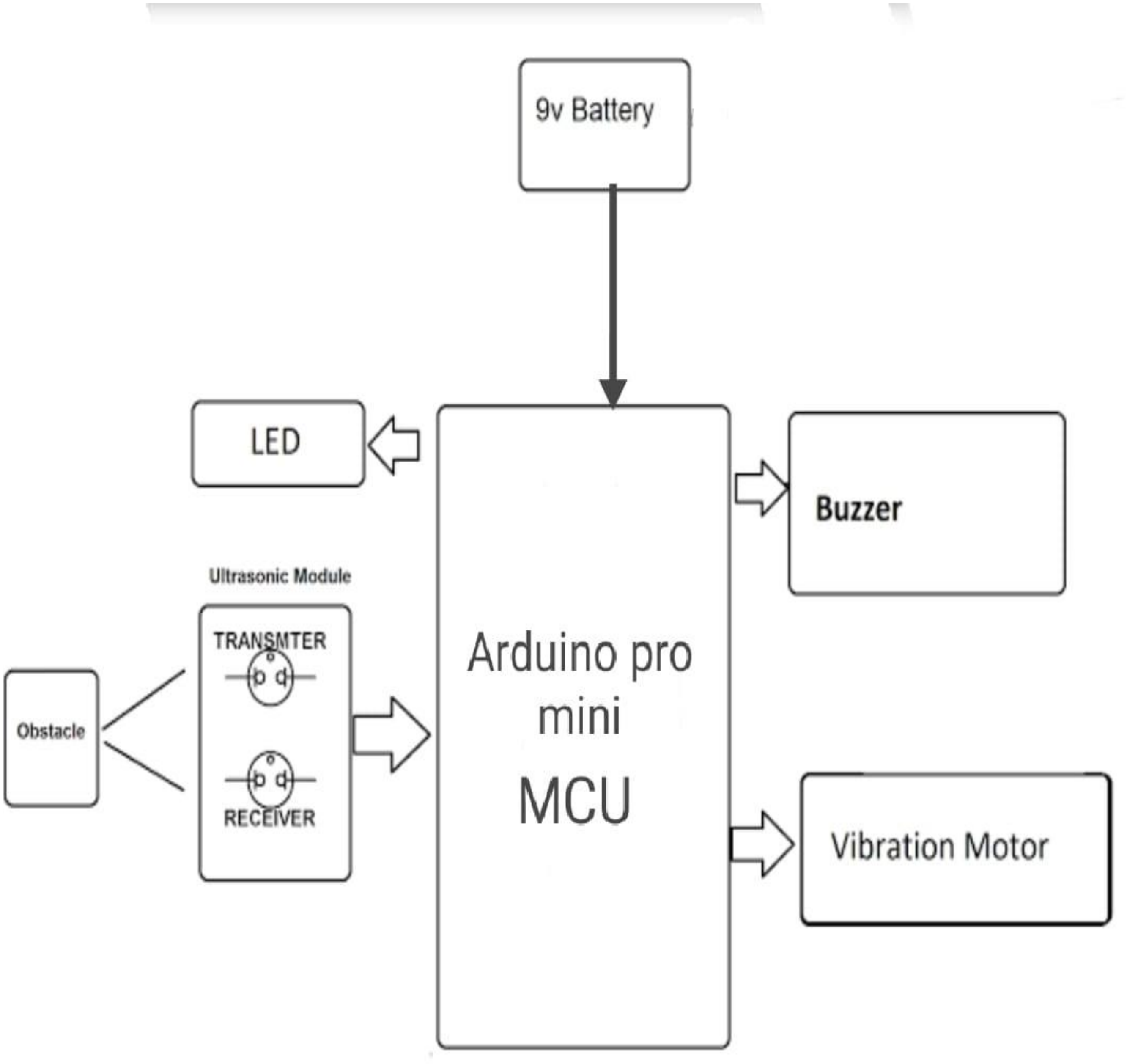


Fig 3.4: Block diagram for Third eye for blind people

3.8 HARDWARE REQUIREMENTS:

1. ULTRASONIC SENSOR
2. ARDUINO PRO MINI
3. PREF BOARD
4. VIBRATING MOTOR
5. BUZZER
6. RED LEDs
7. SWITCHES
8. MALE AND FEMALE HEADER PINS
9. JUMPER CABLE
10. ONE POWER CABLE
11. SOME ELASTICS AND STICKERS (TO MAKE IT AS A BAND FOR WEARING)
12. SWITCH

3.9 SOFTWARE REQUIREMENTS:

1. EMBEDDED C
2. ARDUINO IDE

3.10 EMBEDDED C

Embedded C is a set of language extensions for the C Programming the C standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks and basic input output operations. In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. Embedded C use most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch. case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, unions, etc.

INTRODUCTION TO EMBEDDED C

Looking around, we find ourselves to be surrounded by various types of embedded systems. Be it a digital camera or a mobile phone or a washing machine, all of them has some kind of processor functioning inside it. Associated with each processor is the embedded software. If hardware forms the body of an embedded system, embedded processor acts as the brain, and embedded software forms its soul. It is the embedded software which primarily governs the functioning of embedded systems. During infancy years of microprocessor-based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check correct execution of the program. Some _very fortunate 'developers had In-circuit Simulators (ICEs), but they were too costly and were not quite reliable as well.

As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the **embedded programming language** of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements.

Initially C was developed by Kernighan and Ritchie to fit into the space of 8K and to write (portable) operating systems. Originally it was implemented on UNIX operating systems. As it was intended for operating systems development, it can manipulate memory addresses. Also, it allowed programmers to write very compact codes. This has given it the reputation as the language of choice for hackers too.

EMBEDDED SYSTEMS PROGRAMMING

Embedded systems programming is different from developing applications on a desktop computer. Key characteristics of an embedded system, when compared to PCs, are as follows:

1. Embedded devices have resource constraints (limited ROM, limited RAM, limited stack space, less processing power)
2. Components used in embedded system and PCs are different; embedded systems typically use smaller, less power consuming components.
3. Embedded systems are more tied to the hardware.

Two salient **features of Embedded Programming** are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language. Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of languages:

1. Machine Code

2. Low level language, i.e., assembly

3. High level language like C, C++, Java, Ada, etc.

4. Application-level language like Visual Basic, scripts, Access, etc.

Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high-level languages are preferred for embedded systems programming.

USE OF C IN EMBEDDED SYSTEMS:

1. It is small and reasonably simpler to learn, understand, program and debug.

2. C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.

3. Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.

4. As C combines functionality of assembly language and features of high-level languages, C is treated as a ‗middle-level computer language 'or ‗high level assembly language ‘

5. It is fairly efficient

6. It supports access to I/O and provides ease of management of large embedded projects.

Many of these advantages are offered by other languages also, but what sets C apart from others like Pascal, FORTRAN, etc. is the fact that it is a middle level language; it provides direct hardware control without sacrificing benefits of high-level languages. Compared to other high-level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation. Compared to assembly language, C Code written is more reliable and scalable, more portable between different platforms (with some changes).

It is easier to write good code in C & convert it to an efficient assembly code rather than writing an efficient code in

assembly itself. Benefits of assembly language programming over C are negligible when we compare the ease with which C programs are developed by programmers.

3.11 Arduino Software (IDE):

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches:

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

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Verify

Checks your code for errors compiling it.

Upload

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

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

New

Creates a new sketch.

Open

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

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

Save

Saves your sketch.

Serial Monitor

Opens the serial monitor.

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

File

New

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

Open

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

Open Recent

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

Sketchbook

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

Examples

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

Close

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

Save

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

Save as

Allows to save the current sketch with a different name.

Page Setup

It shows the Page Setup window for printing.

Print

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

Preferences

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

Quit

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

Edit

Undo/Redo

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

Cut

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

Copy

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

Copy for Forum

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

Copy as HTML

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

Paste

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

Select All

Selects and highlights the whole content of the editor.

Comment/Uncomment

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

Increase/Decrease Indent

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

Find

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

Find Next

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

Find Previous

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

Sketch

Verify/Compile

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

Upload

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

Upload Using Programmer

This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch.

Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.

Export Compiled Binary

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

Show Sketch Folder

Opens the current sketch folder.

Include Library

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

Add File

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

Tools

Auto Format

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

Archive Sketch

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

Fix Encoding & Reload

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

Serial Monitor:

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

Board:

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

Port:

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

Programmer:

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

Burn Bootloader:

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

Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection.

The documents are a local copy of the online ones and may link back to our online website.

Find in Reference:

This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Sketchbook

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

Tabs, Multiple Files, and Compilation

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

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or / dev/tty. USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload.

With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

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

Libraries

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

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

Third-Party Hardware

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

Serial Monitor

This displays serial sent from the Arduino or Genuino board over USB or serial connector.

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

Boards

The board selection has two effects: it sets the parameters (e.g., CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection, you'll want to check it before burning the bootloader. You can find a comparison table between the various boards here.

Arduino Software (IDE) includes the built-in support for the boards in the following list, all based on the AVR Core. The Boards 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, and Galileo and so on.

1. Arduino Yun

An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

2. Arduino/Genuino Uno

An ATmega328P running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.

3. ArduinoDiecimila or Duemilanove w/ ATmega168

An ATmega168 running at 16 MHz with auto-reset.

4. Arduino Nano w/ ATmega328P

An ATmega328P running at 16 MHz with auto-reset. Has eight analog inputs.

5. Arduino/Genuino Mega 2560

An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

6. Arduino Mega

An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

7. Arduino Mega ADK

An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

8. Arduino Leonardo

An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

9. Arduino/Genuino Micro

An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

10. ArduinoEsplora

An ATmega32u4 running at 16 MHz with auto-reset.

11. Arduino Mini w/ ATmega328P

An ATmega328P running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 PWM.

12. Arduino Ethernet

Equivalent to Arduino UNO with an Ethernet shield: An ATmega328P running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.

13. ArduinoFio

An ATmega328P running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328P, 6 Analog In, 14 Digital I/O and 6 PWM.

14. Arduino BT w/ ATmega328P

ATmega328P running at 16 MHz The bootloader burned (4 KB) includes codes to initialize the on-board Bluetooth module, 6 Analog In, 14 Digital I/O and 6 PWM.

15. LilyPadArduino USB

An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM.

16. LilyPadArduino

An ATmega168 or ATmega132 running at 8 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.

17. Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328P

An ATmega328P running at 16 MHz with auto-reset. Equivalent to ArduinoDuemilanove or Nano w/ ATmega328P; 6 Analog In, 14 Digital I/O and 6 PWM.

18. Arduino NG or older w/ ATmega168

An ATmega168 running at 16 MHz without auto-reset. Compilation and upload is equivalent to ArduinoDiecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset); 6 Analog In, 14 Digital I/O and 6 PWM.

19. Arduino Robot Control

An ATmega328P running at 16 MHz with auto-reset.

20. Arduino Robot Motor

An ATmega328P running at 16 MHz with auto-reset.

CHAPTER 4

HARDWARE DETAILS

4.1 Ultrasonic Sensor:

Description:
The purpose of ultrasonic sensors is to measure the distance using ultrasonic waves. Ultrasonic sensors emit the ultrasonic waves and receive back the reflected. So, by this time the ultrasonic sensor will measure the distance to the object. It can sense from 2-400 cm.

Function:
In "Smart Glasses", the Ultrasonic sensor is used to measure the distance between the camera and an object to detect the text from the text image. The distance should be from 40 cm to 150 cm and that is because this is the required range to capture a clear image.

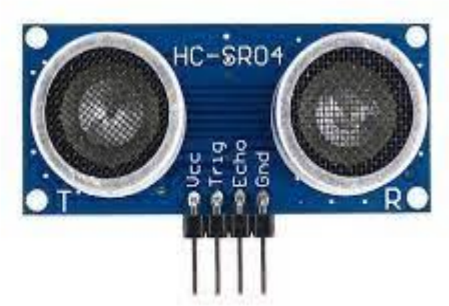


Fig 4.1: Ultrasonic sensor

Distance L = 1/2 × T × C
L: The distance
T: Time between the emission and reception
C: Sonic speed
*The value is multiplied by 1/2 because T is the time for the go-and-return distance.

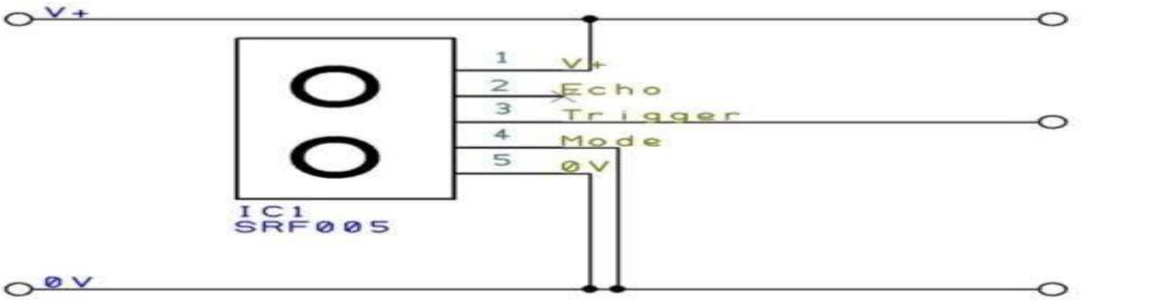


Fig 4.2: Schematic Ultrasonic sensor

4.1.1 Ways to Measure the distance using Ultrasonic sensor

Principle of Ultrasonic Sensor

The Ultrasonic sensor is based on the principle is measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object’s proximity. High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

Working:

The Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse. It sends an ultrasonic pulse out at 40kHz which travels through the air and if there is an obstacle or object, it will bounce back to the sensor. By calculating the travel time and the speed of sound, the distance can be calculated. Ultrasonic sensors are a great solution for the detection of clear objects. For liquid level measurement, applications that use infrared sensors, for instance, struggle with this particular use case because of target translucence. For presence detection, ultrasonic sensors detect objects regardless of the colour, surface, or material. To detect transparent and other items where optical technologies may fail, ultrasonic sensors are a reliable choice.

Ultrasonic Distance Measuring

Distance measurement is based on the measurement of time-of-flight. The time between sending and receiving the reflected sound signal is calculated by the sensor. Ultrasonic distance sensors, like the MB7360 HRXL-MaxSonar-WR, are used as height monitors, in bin level measurement and proximity zone detection applications.

People Detection with Ultrasonic Sensors

Popular applications that we have worked on have been using our sensors in kiosks to detect the presence of a person approaching and detecting the presence of people in an environment, like an art installation.

4.2 ARDUINO PRO MINI:

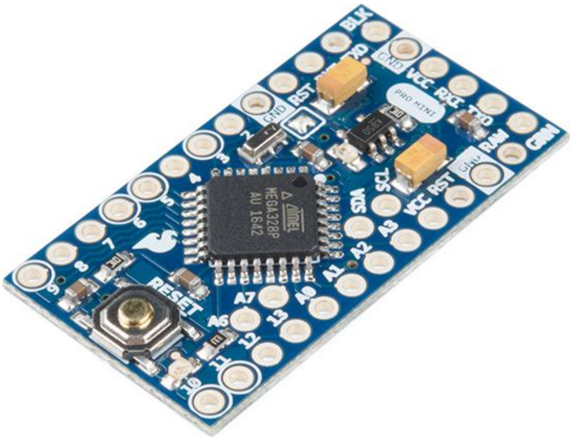


Fig 4.3: Arduino pro mini

General Description

The Arduino Pro Mini is a microcontroller board based on the ATmega168 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, an on-board resonator, are set button, and holes for mounting pin headers. A six-pin header can be connected to an FTDI cable or Spark fun breakout board to provide USB power and communication to the board. The Arduino Pro Mini is intended for semi-permanent installation in objects or exhibitions. The board comes without pre-mounted headers, allowing the use of various types of connectors or direct soldering of wires.

The pin layout is compatible with the Arduino Mini. There are two version of the Pro Mini.

One runs at 3.3V and 8 MHz, the other at 5V and 16 MHz The Arduino Pro Mini was designed and manufactured by Spark Fun Electronics.

1.0 pinout:

added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

Stronger RESET circuit.

At mega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions

Power

The Arduino Pro Mini can be powered with an FTDI cable or breakout board connected to its six-pin header, or with a regulated 3.3V or 5V supply (depending on the model) on the Vcc pin or an unregulated supply on the RAW pin.

The power pins are as follows:

•**RAW**. For supplying a raw (unregulated) voltage to the board.

•**VCC**. The regulated 3.3- or 5-volt supply.

•**GND**. Ground pin.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader). It has 1 KB of SRAM and 512 bytes of EEPROM

Input and Output

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

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

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

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

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

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

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

I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library.

There is another pin on the board:

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.

Communication

The Arduino Pro Mini has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 provides UART TTL serial communication, which is available on digital pins 0 (RX) and 1 (TX). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board via a USB connection. A Software Serial library allows for serial communication on any of the Pro Mini's digital pins. The ATmega168 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the reference for details. To use the SPI communication, please see the ATmega168 datasheet.

Programming

The Arduino Pro Mini can be programmed with the Arduino software (download).
The ATmega168 on the Arduino Pro Mini 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 originalSTK500 protocol.

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 ATmega328on 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 STK500protocol (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 or 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. You can then use Atmel's FLIP software (Windows) orthe DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Pro Mini is designed in a way that allows it to be reset by software running on a connected computer. One of the pins on the six-pin header is connected to the reset line of the ATmega168 via a 100 nanofaradcapacitor. This pin connects to one of the hardware flow control lines of the USB-to-serial convertor connected to the header: RTS when using an FTDI cable, DTR when using the Spark Fun breakout board. 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 the reset line can be well-coordinated with the start of the upload. This setup has other implications. When the Pro Mini is connected to either a computer running MacOS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Pro. 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.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Memory

The ATmega328has 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)

Input and 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 ohms. 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 ATmega8U2USB-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 attach Interrupt () 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

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, labelled 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 is it 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. See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

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.

Pin diagram for Arduino pro mini

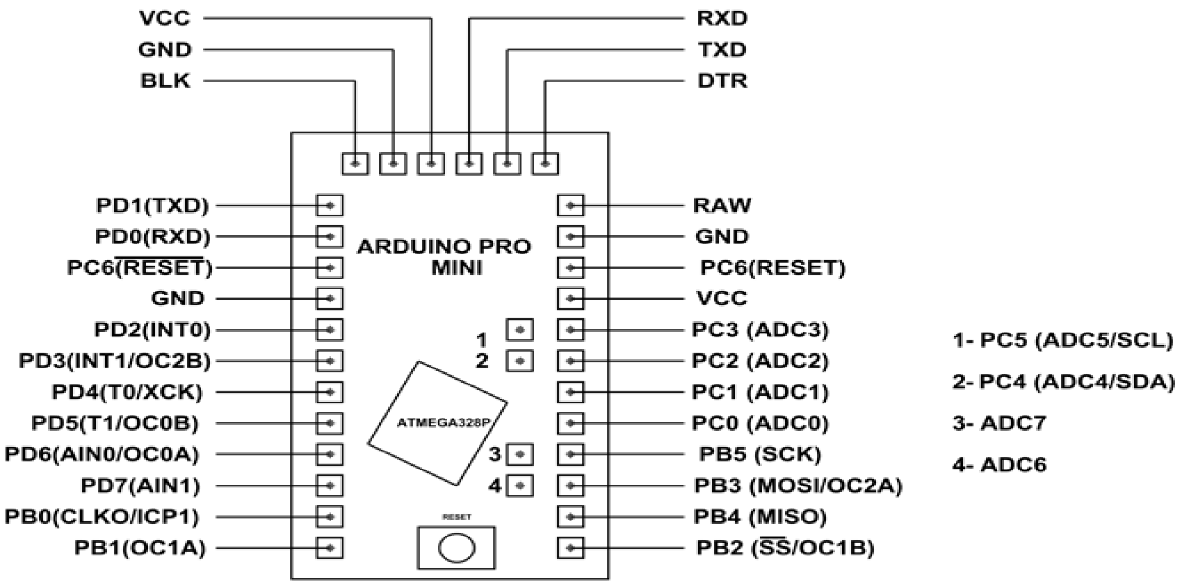


Fig 4.4: PIN diagram for Arduino Pro Mini

4.3 Prefboard

Prefboard is a material for prototyping electronic circuits (also called DOT PCB). It is a thin, rigid sheet with holes pre-drilled at standard intervals across a grid, usually a square grid of 0.1 inches (2.54 mm) spacing. These holes are ringed by round or square copper pads, though bare boards are also available. Inexpensive perfboard may have pads on only one side of the board, while better quality perfboard can have pads on both sides. Since each pad is electrically isolated, the builder makes all connections with either wire wrap or miniature point to point writing techniques. Discrete components are soldered to the prototype board such as resistors, capacitors and integrated circuits. The 0.1 inches (2.54 mm) grid system accommodates integrated circuits in DIP packages and many other types of through-hole components. Perfboard is not designed for prototyping surface mount devices. Before building a circuit on perfboard, the locations of the components and connections are typically planned in detail on paper or with software tools. Small scale prototypes, however, are often built ad hoc, using an oversized perfboard. Software for PCB layout can often be used to generate perfboard layouts as well. In this case, the designer positions the components so all leads fall on intersections of a 0.1 inches (2.54 mm) grid. When routing the connections more than 2 copper layers can be used, as multiple overlaps are not a problem for insulated wires. Once the layout is finalized, the components are soldered in their designated locations, paying attention to orientation of polarized parts such as electrolytic capacitors, diodes, and integrated circuits. Next, electrical connections are made as called for in the layout. For insulated wires thin solid core wire with temperature-resistant insulation such as kynar or Tefzel is preferred. The wire gauge is typically 24 – 30 AWG. A special stripping tool can be used, incorporating a thin steel blade with a slit that the wire is simply inserted into and then pulled loose, leaving a clean stripped end.

This wire was developed initially for circuit assembly by the wire wrap technique but also serves well for miniature point-to-point wiring on perfboard.

Bare copper wire is useful when merging a number of connections to form an electrical bus such as the circuit's ground, and when there is enough space to properly route connections, instead of wiring them rats-nest style. Intentional solder bridges can be used to connect adjacent pads when necessary. Careful hand–eye coordination is needed to avoid causing inadvertent short circuits. Circuits assembled on perfboard are not necessarily fragile but may be less impact-resistant than printed circuit boards. Perfboard differs from stripboard in that each pad on perfboard is isolated. Stripboard is made with rows of copper conductors that form default connections, which are broken into isolated segments as required by scraping through the copper. This is similar to the pattern of default connections on a solderless breadboard. However, the absence of default connectivity on perfboard gives the designer more freedom in positioning components and lends itself more readily to software-aided design than stripboard or breadboard.

4.4 VIBRATING MOTOR:

- Bar-type Vibration Motor
- Coin-type Vibration Motor

4.4.1 Bar-type Vibration Motor:

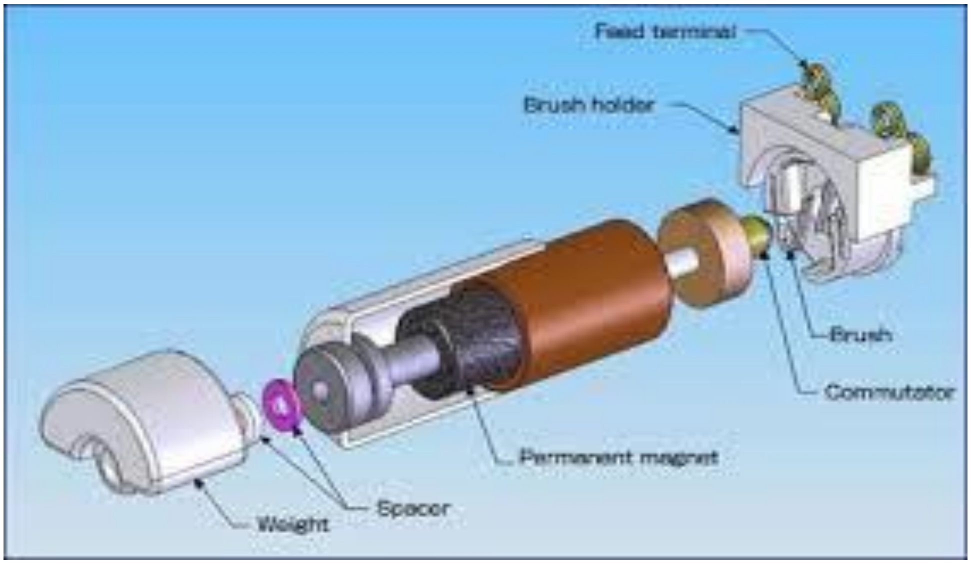


Fig 4.5: Bar-type vibration motor

The cylinder shape is also called bar-type vibration motor. This vibrating motor is essentially a motor that is improperly balanced. In other words, there is an off-cantered weight attached to the motor's rotational shaft that produces a centrifugal force while rotating. This unbalanced force displaces the motor. Its high-speed displacement makes the motor to wobble, which is known as the “vibrating”. The wobble can be changed by the weight mass you attach, the weight's distance to the shaft, and the speed at which the motor spins. What’s more, the centrifugal force, which is generated by the rotating an unbalanced weight, causes the motor vibrate in 2 axes (Z axis and X-axis). According to the relationship of each components in this equation, it is easy to tell that a larger weight Chen 8mass with a bigger offset from the shaft will produce more force and hence more vibration amplitude. Moreover, increasing the voltage supplied to the motor will increase its speed, and therefore the vibration frequency, as well as the vibration amplitude.

4.4.2Coin-type Vibration Motor

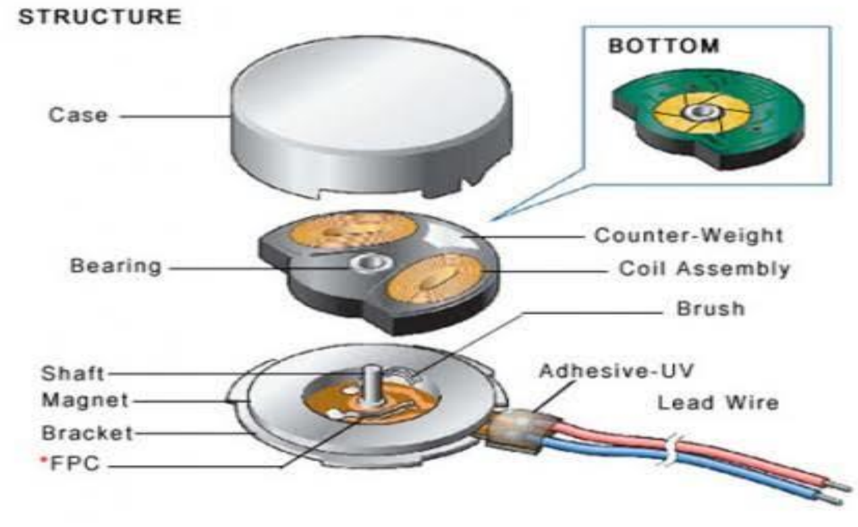


Fig 4.6: coin-type vibration motor

The need for smaller, thinner designs led to the adaptation of brush motor technology into the coin-type vibration motor. Similar to the bar-type vibration motor, coin-type vibration motorist comprised of a weight, a ring magnet, rotor with commutation points attached in the front and coils assembled on the back, and power supplied brushes attached to the ring magnet. The commutation points, which are the yellow part on the bottom pic, are in contact with the end of the brushes. It will energize the electrical coils in the rotor. Energizing the coils produce a magnetic field and it is strong enough to interact with the ring magnet integrated into the stator, causing rotation. A force is generated due to the magnetic field. This force causes the weight to displace. The repeated displacement of the weight produces a varying force which is felt as vibration. The commutation points are used in changing the polarity pairs, so that as the rotator moves, the coils are constantly reversing the polarity.

4.5 BUZZER:

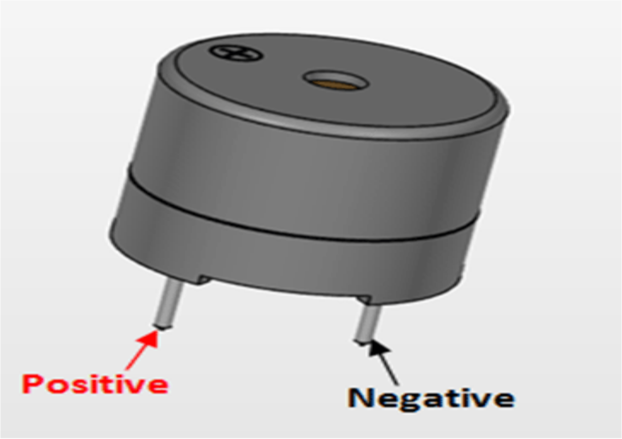


Fig 4.7: Buzzer

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

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 are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beep.... 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. 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.

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

Applications of buzzer

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

4.6 RED LED:

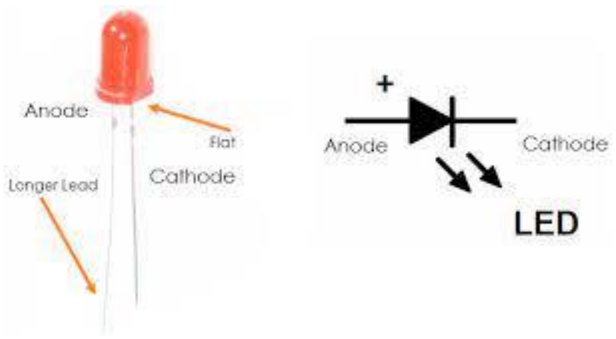


Fig 4.8: Red LED

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The colour of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device

Advantages:

- **Efficiency:** LEDs emit more lumens per watt than incandescent light bulbs. The efficiency of LED lighting fixtures is not affected by shape and size, unlike fluorescent light bulbs or tubes.
- **Colour:** LEDs can emit light of an intended colour without using any colour filters as traditional lighting methods need. This is more efficient and can lower initial costs.
- **Size:** LEDs can be very small and are easily attached to printed circuit boards.
- **Warmup time:** LEDs light up very quickly. A typical red indicator LED achieves full brightness in under a microsecond. LEDs used in communications devices can have even faster response times.
- **Cycling:** LEDs are ideal for uses subject to frequent on-off cycling, unlike incandescent and fluorescent lamps that fail faster when cycled often, or high-intensity discharge lamps (HID lamps) that require a long time before restarting.

- **Dimming:** LEDs can very easily be dimmed either by pulse-width-modulation or lowering the forward current. This pulse-width modulation is why LED lights, particularly headlights on cars, when viewed on camera or by some people, seem to flash or flicker. This is a type of stroboscopic effect.
- **Cool light:** In contrast to most light sources, LEDs radiate very little heat in the form of IR that can cause damage to sensitive objects or fabrics. Wasted energy is dispersed as heat through the base of the LED.
- **Slow failure:** LEDs mainly fail by dimming over time, rather than the abrupt failure of incandescent bulbs.
- **Lifetime:** LEDs can have a relatively long useful life. One report estimates 35,000 to 50,000 hours of useful life, though time to complete failure may be shorter or longer. Fluorescent tubes typically are rated at about 10,000 to 25,000 hours, depending partly on the conditions of use, and incandescent light bulbs at 1,000 to 2,000 hours. Several DOE demonstrations have shown that reduced maintenance costs from this extended lifetime, rather than energy savings, is the primary factor in determining the payback period for an LED product.

4.7 SLIDE SWITCH:

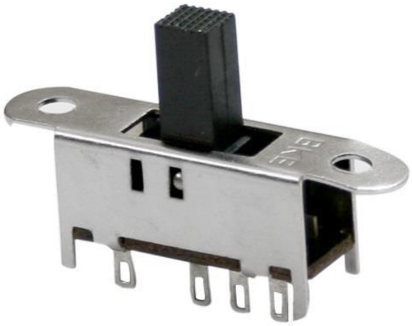


Fig 4.9: Slide switch

A slide switch is a bistable two-position switch and is used to control the current flow in a circuit. It can be closed or interrupted by moving the switch. A slide switch can also be used to connect two further circuits to an existing one and to connect them optionally.

Slide switches are used, for example, in smaller circuits, such as main switches in battery-powered electrical devices. Despite the functional parallels to pushbuttons, slide switches usually have a clearer haptic structure that accurately indicates the on and off states. The mounting depends on the design and can be done for example via contacts or direct mounting on a printed circuit board. Different industries rely on the use of slide switches. Especially small versions are referred to as sub miniature slide switches. There are also switch variants that use similar mechanical principles, such as DIP switches also known as dual-in-line switches, tact switches or coding switches.

4.8 MALE AND FEMALE HEADER PINS:

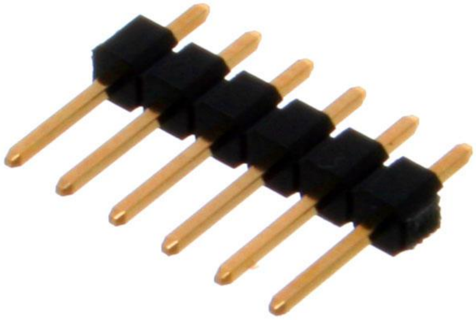


Fig 4.10: Male pin header

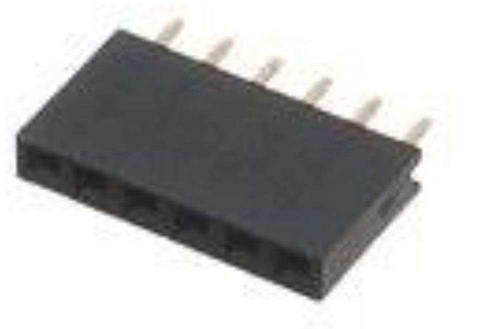


Fig 4.11: Female pin header

A pin header (or simply header) is a form of electrical connector. A male pin header consists of one or more rows of metal pins moulded into a plastic base, often 2.54mm (0.1in) apart, though available in many spacings. Male pin headers are cost-effective due to their simplicity. The female counterparts are sometimes known as a female socket header, though there are numerous naming variations of male and female connectors. Historically, headers are sometimes called “Betg connectors”, but headers are manufactured by many companies.

4.9 JUMPER CABLE:

Fig 4.12: Jumper cable

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires. Though jumper wires come in a variety of colours, the colours don't actually mean anything. This means that a red jumper wire is technically the same as a black one. But the colours can be used to your advantage in order to differentiate between types of connections, such as ground or power. While jumper wires are easy and inexpensive to purchase, it can also be a fun task to challenge students to make their own. Doing so requires insulated wire and wire strippers. However, beware that it is important not to nick the wire when stripping off the insulation.

Types of Jumper wire:

Fig 4.13: Jumper male-to-male cable

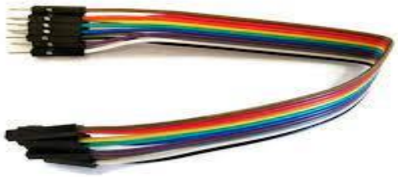


Fig 4.14: Jumper male-to-female cable

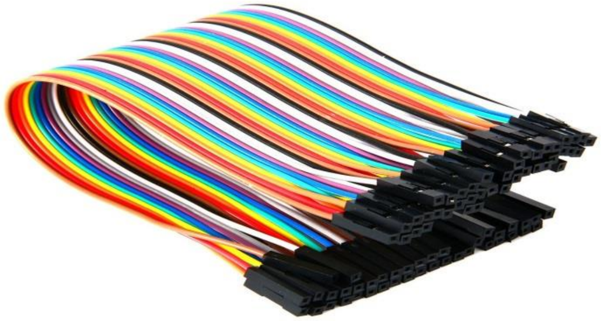


Fig 4.15: Jumper female-to-female cable

Jumper wires typically come in three versions:

1. male-to-male
2. male-to-female
3. female-to-female.

The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you’ll need.

4.10 LI-ION BATTERY:

NOKIA®
BLC-2
3.6 V
RECHARGEABLE Li-ion BATTERY.
USE SPECIFIED NOKIA CHARGER ONLY.
MAY EXPLODE IF DAMAGED OR DISPOSED
OF IN FIRE. DO NOT SHORT-CIRCUIT.
0670343802573244451
MADE IN JAPAN

Fig 4.16: Li-ion battery

Electrical energy powers our lives, whenever and wherever we need it, and can now be accessed with evermore ease and efficiency - even in the absence of nearby power outlets. We increasingly move in unbound and wireless ways, and enjoy high mobility in a potentially healthier local environment. This dramatic development has been made possible by efficient energy storage devices, where high-capacity batteries enable, for example, a variety of electrically-driven tools and vehicles. In principle, we all can enjoy the use of mobile phones, cameras, laptops, power tools, etc., relying on efficient batteries to power them. As a consequence of modern battery technology, electric vehicles are also becoming increasingly popular, and we are in the middle of a switch away from vehicles powered by fossil fuels. In addition, efficient energy storage is an important complement to fluctuating energy sources, such as wind and sunlight. With batteries, the supply-demand chain can thus be balanced over time, even in situations when no energy can be produced. To a large extent, these developments have been made possible by the lithium-ion battery. This type of battery has revolutionized the energy storage technology and enabled the mobile revolution.

Through its high potential, and high energy density and capacity, this battery type has already contributed to improving our lives, and arguably will continue to do so in the years to come.

However, battery development is very daunting and challenging in general, and perhaps particularly so when it comes to lithium-based cells. Ever since Alessandro Volta presented his famous "pile" around 1800,1 tremendous effort has been invested in the development of batteries. Many scientists and engineers, working in academia, industry, and even independently, have contributed to this development, realizing that the identification of solutions for efficient batteries is a highly difficult task. The development has thus been relatively sluggish and only very few efficient battery configurations have been successfully designed over the years. For example, we still rely on the lead–acid battery discovered in the mid-19th century.2,3 Nevertheless, due to several ground-breaking multidisciplinary scientific discoveries, encompassing electrochemistry, organic/inorganic chemistry, materials science, etc., these challenges could indeed be met, and the lithium-ion battery become a reality that essentially changed our world.

Background

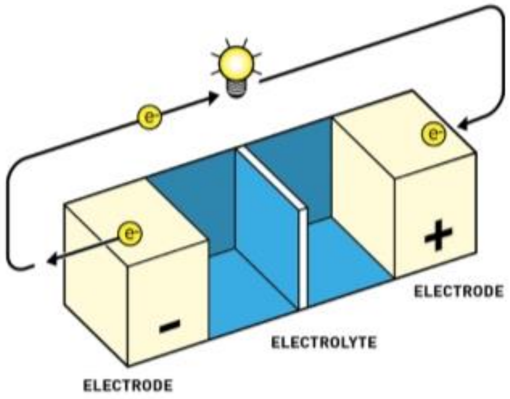


Fig 4.17: Working of battery

The working principle of a battery is relatively straightforward in its basic configuration. The cell is composed of two electrodes, each connected to an electric circuit, separated by an electrolyte that can accommodate charged species. Frequently, the electrodes are physically separated by a barrier material that prevents them from coming into physical contact with one another, which would cause the battery to short-circuit. In the discharge mode, when the battery serves to drive the electric current, an oxidation process takes place at the negative electrode (anode), resulting in electrons moving from the electrode through the circuit. A complementary reduction process takes place at the positive electrode (cathode), replenished by electrons from the circuit. The cell voltage largely depends on the potential difference of the electrodes, and the overall process is spontaneous. For rechargeable (secondary) batteries the process can be reversed and external electricity can be used to produce complementary redox reactions at the electrodes. This process is energy-dependent and non-spontaneous. Working principle of basic battery in the discharge mode (Galvanic element). Spontaneous redox processes at the electrodes result in electric current through the circuit. In the charge mode (electrolytic cell), electricity-driven redox processes take place at the electrodes resulting in reversal of the spontaneous process. The voltaic pile was made of alternating discs of two metals, one of which tin or zinc and the other copper or silver, separated by layers of cardboard or leather soaked in an aqueous electrolyte. Each pair of metal discs and an electrolyte layer made up a battery cell, and the pile was composed of about 20 stacked cells. During operation, in the case of the Zn/Cu cell, the zinc metal acted as an anode, releasing electrons to the circuit and producing metal ions (oxidation), whereas the opposite electrode reaction was dependent on the working conditions. In the presence of air, the copper metal became partially oxidized to CuO, and reduction of CuO to Cu took place at the electrode. In the absence of air, the protons in the electrolyte were instead reduced to hydrogen gas at the copper surface. The cell voltage was approximately 0.8–1.1 V, depending on air exposure. The voltaic pile was essentially a primary battery and not rechargeable. When connecting the poles of the whole device, Volta could demonstrate how the resulting current could generate a spark. After a demonstration of the discovery to Napoleon Bonaparte, the First Consul of the Cisalpine republic was so impressed that he immediately made Volta a count. battery has a working principle similar to the voltaic pile exposed to air, but was the first so called secondary battery that could be recharged. The term secondary was derived from early studies by Nicolas Gautherot, who in 1801 observed short secondary currents from disconnected wires used in electrochemical experiments. During discharge, oxidation takes place at the lead electrode (anode), producing electrons, protons, and lead sulphate (PbSO4), whereas the lead oxide is reduced to PbSO4 at the cathode. In this case, the cell potential is about 2 V, and a typical 12-V car battery is composed of six cells connected in series.

CHAPTER 5

WORKING AND MAKING MODULES

5.1 Steps involved in making modules

Circuit Diagram Description:

Step 1:

Circuit Diagram Descriptions

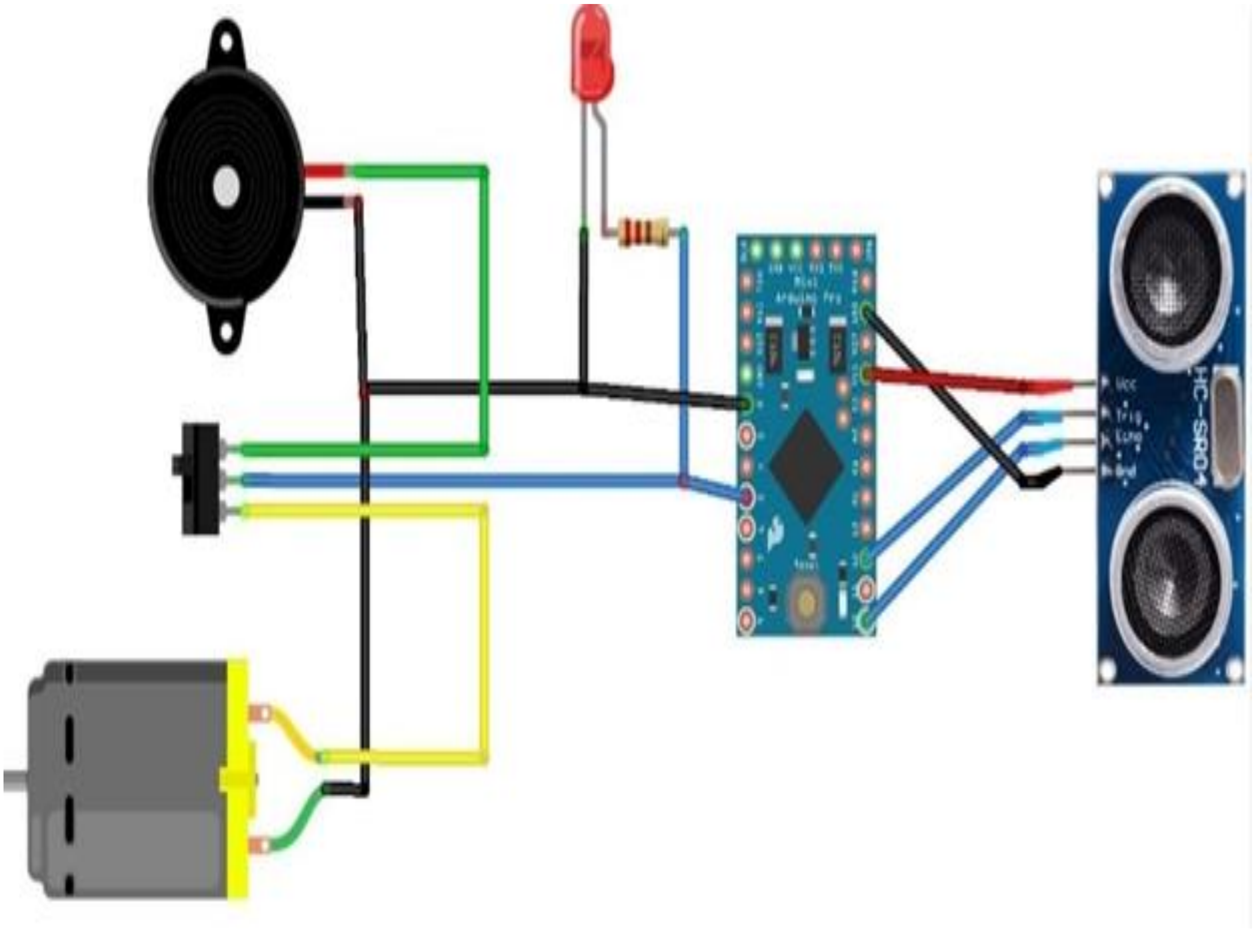


Fig 5.1: Circuit Diagram for third eye for blind

5.2 Module connection with power bank

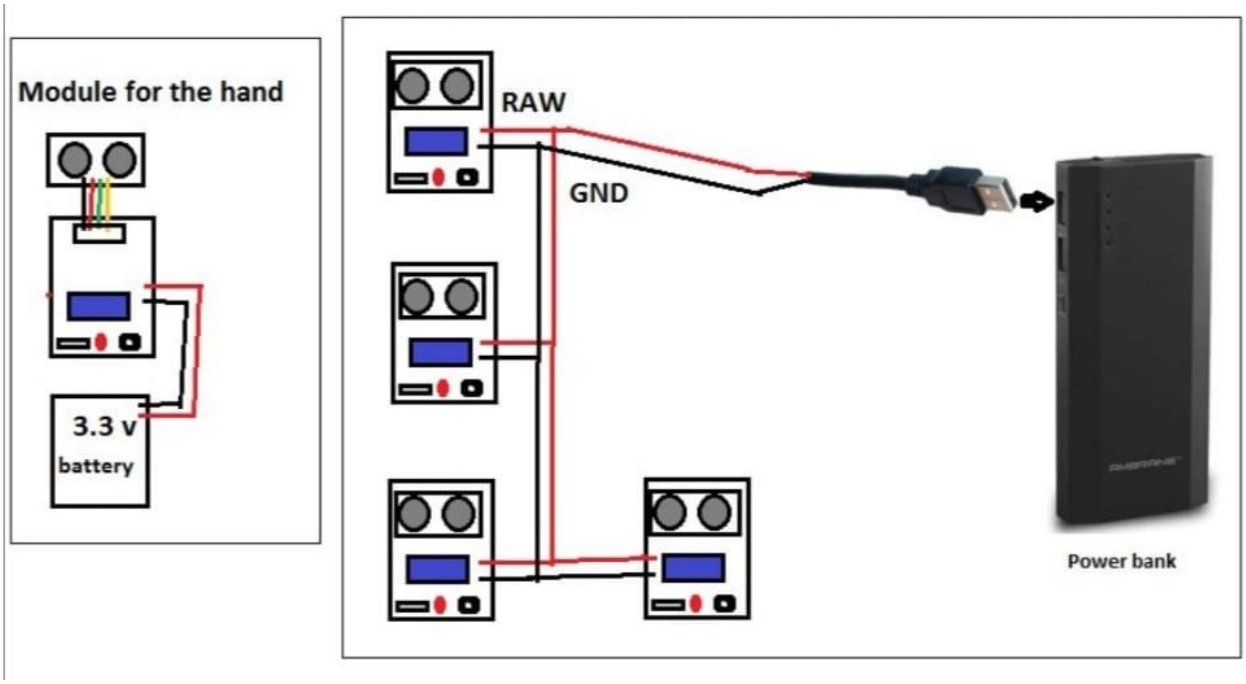


Fig 5.2: Module connection with Power bank

Wiring instruction.

Ground of LED, buzzer and vibration motor to GND of Arduino

+ve of LED and middle leg of switch to Arduino pin 5

+ve of Buzzer to first leg of switch

+ve of Vibration motor to third leg of switch

Ultrasonic sensor

Ultrasonic sensor pin VCC - Arduino pin VCC

Ultrasonic sensor pin GND - Arduino pin GND

Ultrasonic sensor pin Trig - Arduino pin 12

Ultrasonic sensor pin Echo - Arduino PIN 12

The switch used here is for selecting the mode. (buzzer or vibration mode.)

Step 2:

Making the Modules

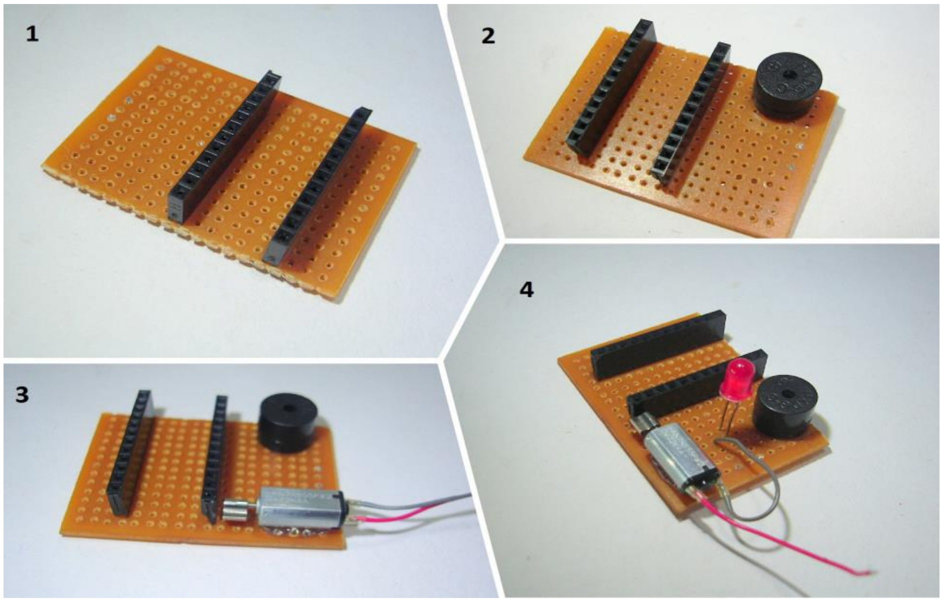


Fig 5.3: Connect Vibration motor and buzzer to the board

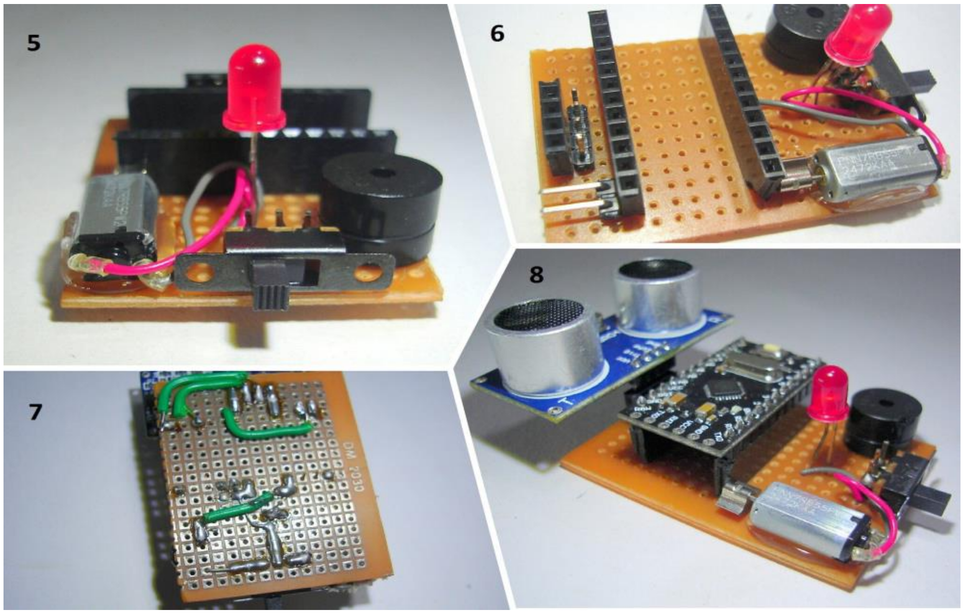


Fig 5.4: Connect sensor and Arduino to the board

First cut the prefboard in 5 X 3 cm dimension and solder the female headers for the Arduino to the board.

1. Then solder the buzzer.
2. Then connect the vibrating motor using the glue gun and solder wires to it.
3. Then connect the LED.
4. Then connect the switch.
5. Then connect header pins for ultrasonic sensors and for battery input.
6. Then solder everything as shown in the circuit diagram.
7. Now connect the Arduino and ultrasonic sensor to the board

Step 3:

Code and Making the Module for the hand

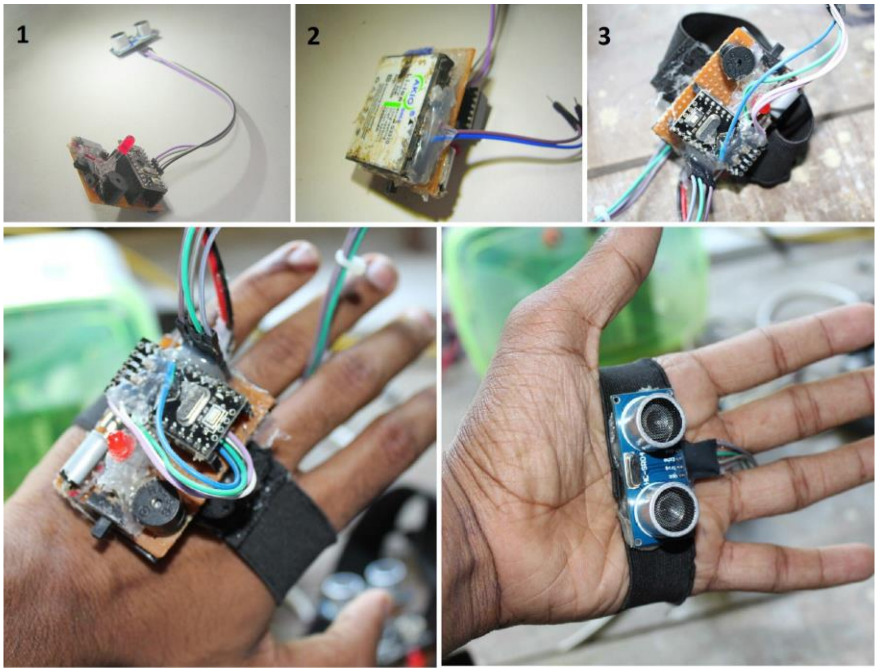


Fig 5.5: Connect the elastic band

Connect the ultrasonic sensor to the board by using 4 jumper cables.

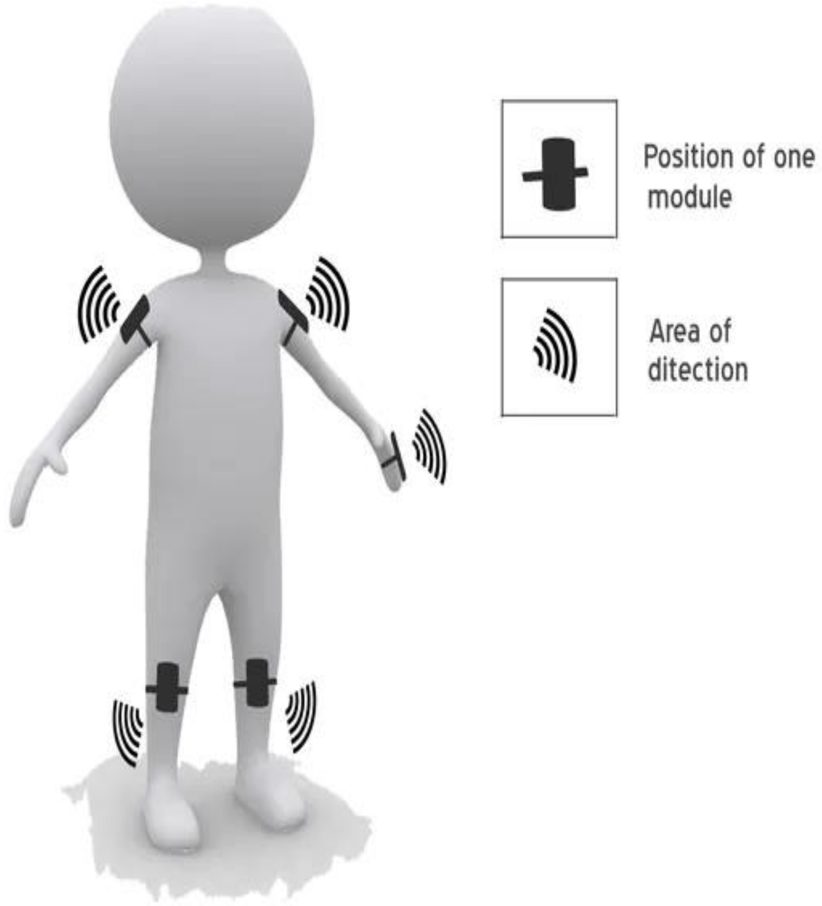
1. Then connect a 3.7-volt mobile battery to this module.
2. Then connect the elastic band.

Step 4:

At last upload the code to each Arduino board and power the 4 other modules using a power bank.

Step 5:

Fixing the sensor to the blind people.



5.2 Code used in the arduino:

```
const int pingTrigPin = 12; //Trigger connected to PIN 12
const int pingEchoPin = 10; //Echo connected yo PIN 10
int buz=5; //Buzzer to PIN 5
long microsecondsToCentimeters(long microseconds)
{
return microseconds / 29 / 2;
}
void setup() {
Serial.begin(9600);
pinMode(buz, OUTPUT);
}
void loop()
{
long duration, cm;
pinMode(pingTrigPin, OUTPUT);
digitalWrite(pingTrigPin, LOW);
delayMicroseconds(2);
digitalWrite(pingTrigPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingTrigPin, LOW);
pinMode(pingEchoPin, INPUT);
duration = pulseIn(pingEchoPin, HIGH);
cm = microsecondsToCentimeters(duration);
if(cm<=50 && cm>0)
{
int d= map(cm, 1, 100, 20, 2000);
digitalWrite(buz, HIGH);
delay(100);
digitalWrite(buz, LOW);
delay(d);
}
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
}
```

CHAPTER 6

CONCLUSION

The objective of this project is Third Eye for the Blind is to design a product which is very much useful to those people who are visually impaired and those who often have to rely on others. The third eye for Blind project is an innovation which helps the blind person to move around and go from one place to another with speed and confidence by knowing the nearby obstacles using the help of the wearable band which produces the ultrasonic waves which notify them with buzz sound or vibrations. It allows the user those who are visually impaired to walk freely by detecting the obstacles. They only need to wear this device as a band or cloth on their body. Thus, this project Arduino based obstacle detector for blind people is a new method to resolve their problems. A less complex portable, cost efficient, easy to manage an effective system with many more amazing properties and advantages are proposed to provide support for the blind. The system will be very easy to find the distance between the objects and the sensor. It can detect the objects in every direction the blind person. Without the help of others, the blind person can move from one place to other and lead their regular lives independently.

REFERENCES

[1] S. Shovel, I Ulrich, J. Borenstien.Nav Belt and the Guide Cane, IEEE “Transactions on Robotics & Automation”. 2003; 10(1):9-20.

[2] D. Yuan R. Manduchi. “Dynamic Environment Exploration Using a Virtual White Cane”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), University of California, Santa Cruz, 2005, 1-7.

[3] JM. Benjamin,NA. Ali, AF. Schepis. “A laser cane for the blind”, Proceedings of San Diego Medical Symposium, 1973, 443-450.

[4] S.Sabarish. “Navigation Tool for Visually Challenged using Microcontroller”, International Journal of Engineering and Advanced Technology (IJEAT), 2013; 2(4):139-143.

[5] MA. Espinosa,S. Ungar,E. Ochaíta. “Blades comparing methods for Introducing Blind and Visually Impaired People to unfamiliar urban environments.”, Journal of Environmental psychology. 1998; 18:277-287.

[6] Pooja Sharma,SL. Shimi,S. Chatterji. “A Review on Obstacle Detection and Vision”, International Journal of Science and Research Technology. 2015; 4(1):1-11.

[7] AA. Tahat.” A wireless ranging system for the blind long-cane utilizing a smart-phone”, in Proceedings of the 10th International Conference on Telecommunications. (ConTEL '09), IEEE, Zagreb, Croatia, June. View at Scopus. 2009, 111-117.

[8] D. Bolgiano, E. Meeks.” A laser cane for the blind”, IEEE Journal of Quantum Electronics. View at Google Scholar. 1967; 3(6):268.