

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in

Information Technology

By

UPENDHER RAO.P
(Reg.No. 37120054)



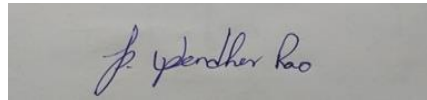
DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

JANUARY 2021

DECLARATION

I **P.UPENDHER RAO** (Reg.No.37120054)here by declare that the Project Report entitled **“THIRD EYE FOR BLIND PEOPLE”** done by me under the guidance of **Dr. A. SIVASANGARI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

A rectangular box containing a handwritten signature in blue ink. The signature appears to be 'P. Upendher Rao'.

DATE:

PLACE:

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, school of computing , **Dr.SUBHASHINI M.E., Ph.D. ,** Head of the department of information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sence of gratitude to my project Guide **Dr.A.SIVASANGARI** for his valuable guidance, completion of my project work.

I wish to express my thanks to all Teaching and Non-Teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

Training Certificate





VEGA PLACEMENT AND TRAINING

(Reg.No: TRH 22585)

Certificate of Internship

We present this Certificate to

Mr./Ms UPENDHER RAO POLISHETTY

of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, in appreciation for your successful work as an intern at VEGA PLACEMENT AND TRAINING. The internship on INTERNET OF THINGS(IOT). The Internship was conducted between 05-11-2020 to 05-12-2020.



Managing Director

ABSTRACT

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. 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 peculiarity 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. Thus this device will be of a great use for the blinds and help them travel different places.

TABLE OF CONTENTS

CHAPTER. NO	TITLE	PAGE.NO
1	INTRODUCTION	
	1.1 Objective of Project	1
	1.2 Literature review	2
2	IMPLEMENTATION OF PROJECT	
	2.1 Block Diagram	5
	2.2 Arduino	5
	2.3 Ultrasonic Sensor	6
	2.4 Hardware	7
	2.5 Software Development	9
	2.6 Applications	10
	2.7 Programming	11
	2.8 Power	14
	2.9 Input and Output	15
3	FUNCTIONAL MODULES AND SOFTWARE MODULES DESCRIPTION	
	3.1 Ultrasonic Sensor	17
	3.2 Buzzer	18
	3.3 Power Supply	19
	3.3.1 Transformer	20
	3.3.2 Rectifier	20
	3.3.3 Filter	20
	3.3.4 Voltage Regulator	20
	3.4 Arduino Software Installation and	21

	Description	
4	PROJECT IMPLEMENTATION	
	4.1 Design and Implementation	24
	4.2 Advantages	24
5	CONCLUSION AND FUTURE SCOPE	
	5.1 Conclusion	26
	5.2 Future Scope	26
6	RESULT AND SCREENSHOTS	
	6.1 Result	27
	6.2 Screenshots	27
	REFERENCES	28
	APPENDIX	29

LIST OF FIGURES

FIGURE.NO	TITLE OF THE FIGURE	PAGE.NO
2.1	Block Diagram	5
2.2	Arduino Circuit	6
2.4	Arduino Hardware	8
2.7.1	Technical Requirements	11
2.7.2	Arduino Layout	13
2.7.3	Pin Diagram	14
3.1	Ultrasonic Sensor	18
3.2	Buzzer	19
3.3	Components of Power Supply	20
3.4.1	Arduino Window	21
3.4.2	Launch Arduino IDE	22
3.4.3	Creating a File	23
6.2.1	Output	27

LIST OF ABBREVIATIONS

ACRONYM

EXPANSION

RFID

Radio Frequency Identification

IR

Infrared

USB

Universal Serial Bus

TTL

Transistor Transistor Logic

CHAPTER 1

INTRODUCTION

1.1 Objective of Project:

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 . 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. According to WHO or the World Health Organization, 39 million people are estimated as blinds worldwide. 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 peculiarity 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 helps 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.

1.2 Literature Review:

Over the last few years or we can say over the last decades, research has been conducted for new devices and technologies to design a good and reliable and efficient system for blind or visually impaired people to detect the obstacles and warn or alert them at danger places or the obstacles. There are some systems which has some limitations and clampdown. Shoval et al. in developed a Navbelt, an obstacle avoidance wearable portable computer which is only for indoor navigation. Navbelt was equipped with two modes, in the first one the system information was translated to audio in different sounds. One sound for free for travel direction and other for blocked, it was difficult for the person to differentiate the sounds. Other problem was the system would not know the user momentary position. D. Yuan et al. in have discussed about the

virtual white cane sensing device based on active triangulation that can measure distances at a rate of 15 measurements/second. A blind person can use this device for sensing the environment, pointing it as if it was a flash light. Beside measuring distances, this device can detect surface discontinuities, such as the foot of a wall, a step, or a drop-off. This is obtained by analysing the range data collected as the user swings the device around, tracking planar patches and finding discontinuities. Benjamin et al. in introduce a laser cane with three photo diodes and three laser diodes function as receiver making an optical triangulation. The laser cane generally detects the obstacle in three specified directions. One is 45° to the ground for overhanging obstacles, the second one is parallel to the ground and third one is for sharp deepness. The laser cane has no data or technology or we can say system for determining the location and the position of the obstacle, rather it is just like a hit and trial method. J. Na proposed an interactive guide system for indoor positioning of this, which can't detect the obstacles and hurdles. The system is not suitable for the outdoor activities. Sabarish. S in have described the development of a navigation aid in order to assist blind and visually impaired people to navigate easily, safely and to detect any obstacles. The system is based on a microcontroller with synthetic speech output. In addition to this, the device consists of two vibrators, two ultrasonic sensors which is mounted on the user's shoulders or any other body part and another one integrated into the cane. M.A Ungar S. has proposed methods for the visually impaired people for the urban cities. But they didn't considered about the people who cannot afford costly equipment and devices. This limitation is overcome by the device third eye for the blind. Ms. Pooja Sharma has discussed that the obstacles can be detected, but it has many limitations on the angles and the distance. On contrary, this project will have a wide angle for the detection where the sensors range will be wide. In today's world of innovations, there are many innovations for the visually impaired people like the white cane with the cane with a red tip for helping the movements of the blind people. There are many different types of canes used in today's world with growing technologies such as as the white cane, the smart cane , and the laser cane . The cost of the trained dogs is also very high and is not affordable option . A survey found that the remote guidance system is very difficult to carry and thus the wearable band will be more optimized version.

Likewise Guide Cane, this invention also has a control button on the handle, and the button has four different directions. This invention of the Third eye for blind has the same shortcomings or limitations as the Guide Cane where there will be a difficulty to save space for the cane or to place the smart cane like the other. Other than that, cost is also a problem in this project as it uses ultrasonic sensors and many servo motors. If the cost is high, users may not be able to afford for it because the average income of the visually impaired people is usually less. Smart Cane has been designed by students from Central Michigan University where this invention uses Radio Frequency Identification (RFID). RFID is used to detect objects or obstacles in front of the user and detects the RFID tag that has been placed in several locations to navigate the users. This invention is just like a normal stick but is provided with a bag, which the user to wear. The bag provides electrical power to the invention and informs the user through speakers inside the bag. For users who cannot hear, there are special gloves that will vibrate at every finger, in which different vibrations in each finger have different meanings. However, this invention has several shortcomings and is only suitable for small areas. This is because it only detects the area with RFID tag otherwise this invention only works as a regular blind cane.

CHAPTER 2

IMPLEMENTATION OF PROJECT

2.1 Block Diagram:

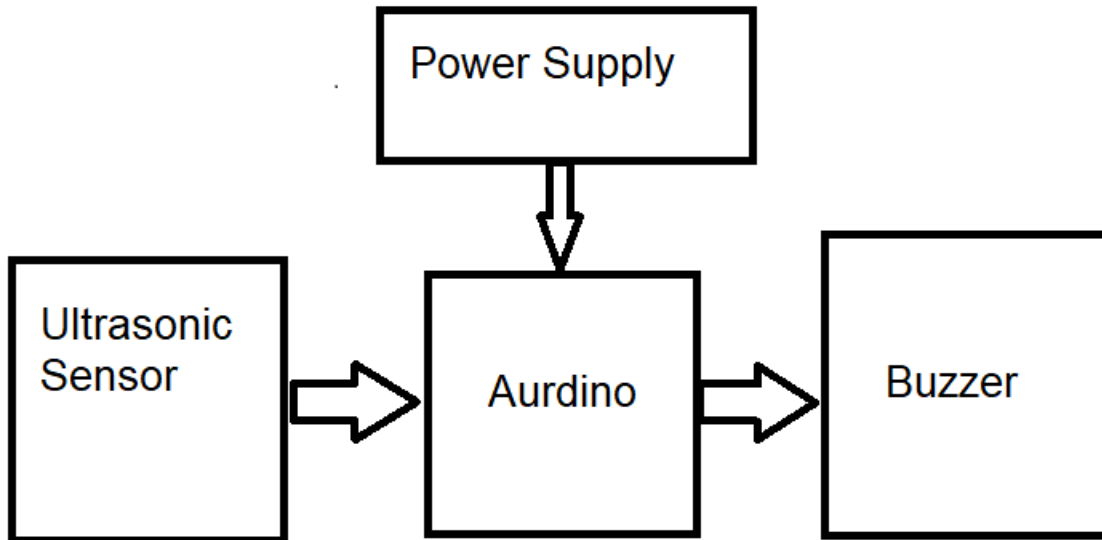


Fig-2.1: Block diagram

The ARDUNIO ships as a bare circuit board with standard connections Ultrasonic sensor and buzzer with power supply.

2.2 ARDUINO:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast

7-12V Depending on current drawn

Cut to disable the auto-reset

This provides a logic reference voltage shields that use it. It is connected to the 5V bus.

R3 Only

The input voltage to the Arduino board when it is running from external power. Not USB bus power.

Absolute max per pin 40mA recommended 20mA

Absolute max 200mA for entire package

RESET Button

ARDUINO UNO

DIGITAL (PWM~)

TX+ RX=0

PCINT14

PC6

PCINT8

PCINT9

PCINT10

PCINT11

PCINT12

PCINT13

PCINT14

PCINT15

PCINT16

PCINT17

PCINT18

PCINT19

PCINT20

PCINT21

PCINT22

PCINT23

PCINT24

PCINT25

PCINT26

PCINT27

PCINT28

PCINT29

PCINT30

PCINT31

PCINT32

PCINT33

PCINT34

PCINT35

PCINT36

PCINT37

PCINT38

PCINT39

PCINT40

PCINT41

PCINT42

PCINT43

PCINT44

PCINT45

PCINT46

PCINT47

PCINT48

PCINT49

PCINT50

PCINT51

PCINT52

PCINT53

PCINT54

PCINT55

PCINT56

PCINT57

PCINT58

PCINT59

PCINT60

PCINT61

PCINT62

PCINT63

PCINT64

PCINT65

PCINT66

PCINT67

PCINT68

PCINT69

PCINT70

PCINT71

PCINT72

PCINT73

PCINT74

PCINT75

PCINT76

PCINT77

PCINT78

PCINT79

PCINT80

PCINT81

PCINT82

PCINT83

PCINT84

PCINT85

PCINT86

PCINT87

PCINT88

PCINT89

PCINT90

PCINT91

PCINT92

PCINT93

PCINT94

PCINT95

PCINT96

PCINT97

PCINT98

PCINT99

PCINT100

PCINT101

PCINT102

PCINT103

PCINT104

PCINT105

PCINT106

PCINT107

PCINT108

PCINT109

PCINT110

PCINT111

PCINT112

PCINT113

PCINT114

PCINT115

PCINT116

PCINT117

PCINT118

PCINT119

PCINT120

PCINT121

PCINT122

PCINT123

PCINT124

PCINT125

PCINT126

PCINT127

PCINT128

PCINT129

PCINT130

PCINT131

PCINT132

PCINT133

PCINT134

PCINT135

PCINT136

PCINT137

PCINT138

PCINT139

PCINT140

PCINT141

PCINT142

PCINT143

PCINT144

PCINT145

PCINT146

PCINT147

PCINT148

PCINT149

PCINT150

PCINT151

PCINT152

PCINT153

PCINT154

PCINT155

PCINT156

PCINT157

PCINT158

PCINT159

PCINT160

PCINT161

PCINT162

PCINT163

PCINT164

PCINT165

PCINT166

PCINT167

PCINT168

PCINT169

PCINT170

PCINT171

PCINT172

PCINT173

PCINT174

PCINT175

PCINT176

PCINT177

PCINT178

PCINT179

PCINT180

PCINT181

PCINT182

PCINT183

PCINT184

PCINT185

PCINT186

PCINT187

PCINT188

PCINT189

PCINT190

PCINT191

PCINT192

PCINT193

PCINT194

PCINT195

PCINT196

PCINT197

PCINT198

PCINT199

PCINT200

PCINT201

PCINT202

PCINT203

PCINT204

PCINT205

PCINT206

PCINT207

PCINT208

PCINT209

PCINT210

PCINT211

PCINT212

PCINT213

PCINT214

PCINT215

PCINT216

PCINT217

PCINT218

PCINT219

PCINT220

PCINT221

PCINT222

PCINT223

PCINT224

PCINT225

PCINT226

PCINT227

PCINT228

PCINT229

PCINT230

PCINT231

PCINT232

PCINT233

PCINT234

PCINT235

PCINT236

PCINT237

PCINT238

PCINT239

PCINT240

PCINT241

PCINT242

PCINT243

PCINT244

PCINT245

PCINT246

PCINT247

PCINT248

PCINT249

PCINT250

PCINT251

PCINT252

PCINT253

PCINT254

PCINT255

PCINT256

PCINT257

PCINT258

PCINT259

PCINT260

2.3 Ultrasonic Sensor:

xvi

electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is $D = \frac{1}{2} T \times C$ (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second).

Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology. In comparison to infrared (IR) sensors in proximity sensing applications, ultrasonic sensors are not as susceptible to interference of smoke, gas, and other airborne particles (though the physical components are still affected by variables such as heat).

Ultrasonic sensors are also used as level sensors to detect, monitor, and regulate liquid levels in closed containers (such as vats in chemical factories). Most notably, ultrasonic technology has enabled the medical industry to produce images of internal organs, identify tumors, and ensure the health of babies in the womb.

2.4 Hardware:

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. The source code for the IDE is released under the GNU General Public License, version 2. Nevertheless, an official Bill of Materials of Arduino boards has never been released by Arduino staff.

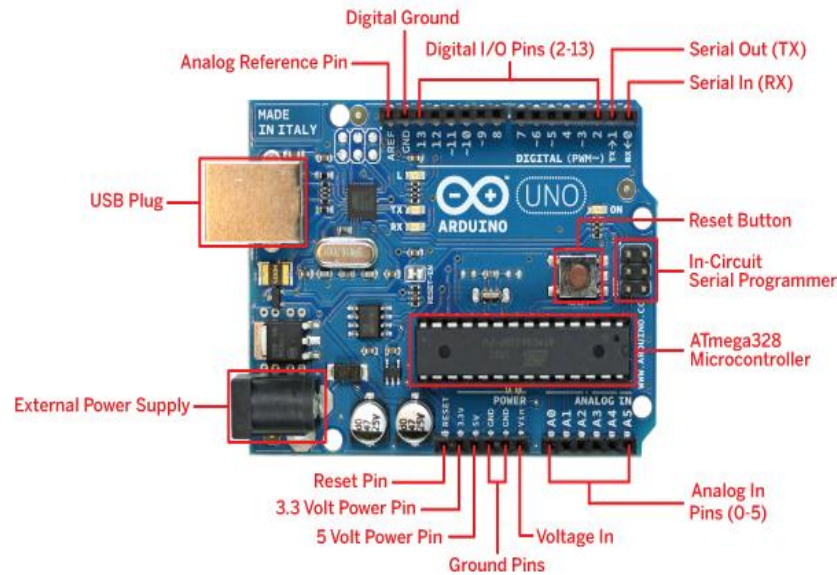


Fig-2.4 Arduino Hardware

Although the hardware and software designs are freely available under copy left licenses, the developers have requested that the name *Arduino* be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in *-duino*.

An Arduino board consists of an Atmel 8-, 16- or 32-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560), but other makers' microcontrollers have been used since 2015. The boards use single-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed *shields*. Multiple, and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the LilyPad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies

uploading of programs to the on-chip flash memory. The default bootloader of the Arduino UNO is the optibootbootloader.^[12] Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods, when used with traditional microcontroller tools instead of the ArduinoIDE, standard AVR in-system programming (ISP) programming is used.

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The *Diecimila*, *Duemilanove*, and current *Uno* provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The ArduinoNano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

2.5 Software Development:

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages *Processing* and *Wiring*. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a *sketch*. Sketches are saved on the development computer as text files with the file extension *.ino*. Arduino Software (IDE) pre-1.0 saved sketches with the extension *.pde*.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

2.6 Applications:

- Xoscillo, an open-source oscilloscope.
- Arduinome, a MIDI controller device that mimics the Monome.
- OBDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars.
- Ardupilot, drone software and hardware.
- Gameduino, an Arduino shield to create retro 2D video games
- ArduinoPhone, a do-it-yourself cellphone
- Water quality testing platform
- Automatic titration system based on Arduino and stepper motor

- Low cost data glove for virtual reality applications
- Impedance sensor system to detect bovine milk adulteration
- Homemade CNC using Arduino and DC motors with close loop control by Homofaciens
- DC motor control using Arduino and H-Bridge.

2.7 Programming:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Fig 2.7.1 Technical Requirements

The Arduino/Genuino Uno can be programmed with the (Arduino Software (IDE)). Select "Arduino/Genuino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino/Genuino Uno comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is

available in the Arduino repository. 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 reseing 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) or the 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.

Fig 2.7.2 Arduino Layout

Arduino function			Arduino function
reset	(PCINT14/RESET) PC6	1	28 PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27 PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26 PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25 PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24 PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23 PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	22 GND GND
GND	GND	8	21 AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20 AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19 PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18 PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17 PB3 (MOSI/OC2A/PCINT3) digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16 PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15 PB1 (OC1A/PCINT1) digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

The Arduino/Genuino Uno 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.

The Arduino/Genuino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and

damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

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

2.9 Input and Output:

See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical.

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 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. In addition some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial

data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function.

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.

CHAPTER 3

FUNCTIONAL MODULES & SOFTWARE MODULES DESCRIPTION

3.1 Ultrasonic Sensor:

Ultrasonic transducers and ultrasonic sensors are devices that generate or sense ultrasound energy. They can be divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.

In a similar way to radar and sonar, ultrasonic transducers are used in systems which evaluate targets by interpreting the reflected signals. For example, by measuring the time between sending a signal and receiving an echo the distance of an object can be calculated. Passive ultrasonic sensors are basically microphones that detect ultrasonic noise that is present under certain conditions.

The design of transducer can vary greatly depending on its use: those used for medical diagnostic purposes, for example the range-finding applications listed above, are generally lower power than those used for the purpose of changing the properties of the liquid medium, or targets immersed in the liquid medium, through chemical, biological or physical (e.g. erosive) effects. The latter class include ultrasonic probes and ultrasonic baths, which apply ultrasonic energy to agitate particles, clean, erode, or disrupt biological cells, in a wide range of materials; See Sonication.

The ultrasonic sensor emits short bursts of sound and listens for this sound to echo off of nearby objects. The frequency of the sound is too high for humans to hear (it is ultrasonic). The ultrasonic sensor measures the time of flight of the sound burst. A user then computes the distance to an object using this time of flight and the speed of sound (1,126 ft/s).

This sensor uses ultrasonic sound to measure distance just like bats and dolphins do. Ultrasonic sound has such a high pitch that humans cannot hear it. This particular sensor sends out an ultrasonic sound that has a frequency of about 40kHz.

The sensor has two main parts:

A transducer that creates an ultrasonic sound and another listens to it's echo.

To use this sensor to measure distance, the robot's brain must measure the amount of time it takes for the ultrasonic sound to travel.

Sound travels at approximately 340 meters per second. This corresponds to about 29.412us (microseconds) per centimeter.

To measure the distance the sound has travelled we use the formula:

$$\text{Distance} = (\text{Time} \times \text{Speed of Sound}) / 2.$$

The "2" is in the formula because the sound has to travel back and fourth. First the sound travels away from the sensor, then it bounces off of a surface and returns back.

The easy way to read the distance as centimeters is use the formula:

$$\text{Centimeters} = ((\text{Microseconds} / 2) / 29).$$

For example, if it takes 100us (microseconds) for the ultrasonic sound to bounce back, then the distance is $((100 / 2) / 29)$ centimeters or about 1.7 centimeters.

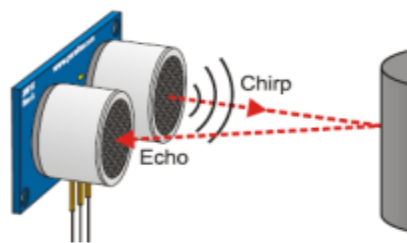


Fig 3.1:Ultrasonic Sensor

3.2 BUZZER:

We use a buzzer that is an important component of our surveillance system. In the event a trespasser bypasses the manual security, our surveillance system will not miss the trespasser. When detected near the restricted area, an alarm is raised with the help of the buzzer, which will alert the security people and force the trespasser to run away.

We make use of a piezo buzzer, which is readily available and cost-effective. A piezo buzzer is built of a piezo-electric material. The working of the piezo buzzer is based on the inverse of the principle of piezo electricity. The principle of piezo electricity states that whenever mechanical pressure is applied on piezo electric material, it generates electricity. And piezo buzzers follow the opposite principle, which states that when a voltage is applied on a piezo electric element, it deforms back and forth (a change in dimension takes place) and produces an audible sound.



Fig 3.2 :Buzzer

3.3 POWER SUPPLY :

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.

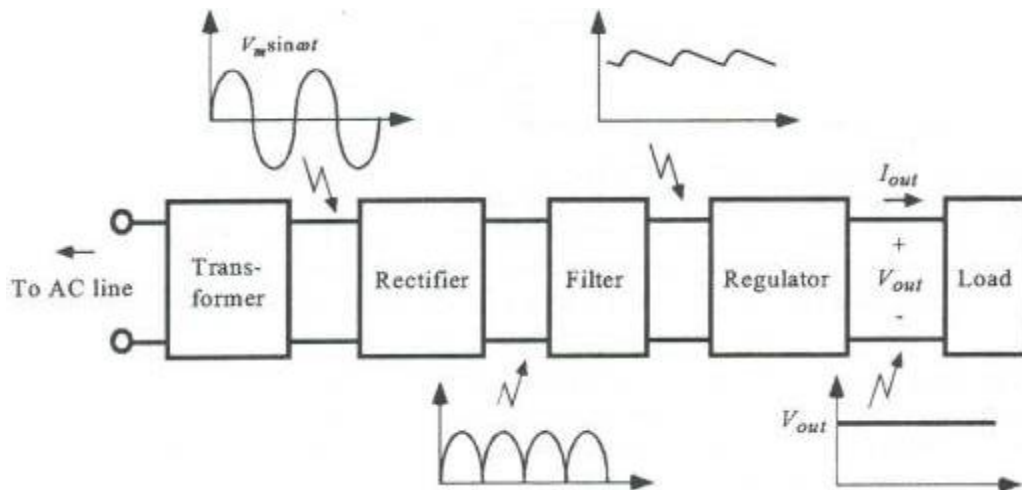


Fig 3.3: Components of Power Supply

3.3.1 Transformer :

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

3.3.2 Rectifier :

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

3.3.3 Filter :

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

3.3.4 Voltage Regulator:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this

project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.

3.4 ARDUINO SOFTWARE INSTALLATION AND DESCRIPTION:

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1: First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, ArduinoDuemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer.

Step 2: Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

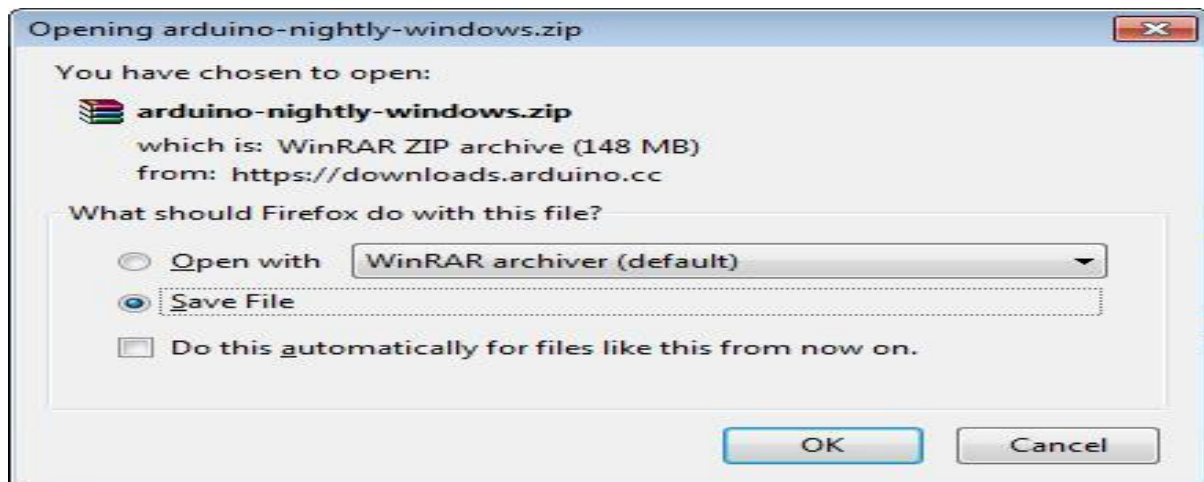


Fig 3.4.1 Arduino Window

Step 3: Power up your board: The Arduino Uno, Mega, Duemilanove and ArduinoNano automatically draw power from either, the USB connection to the

computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4: Launch Arduino IDE. After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

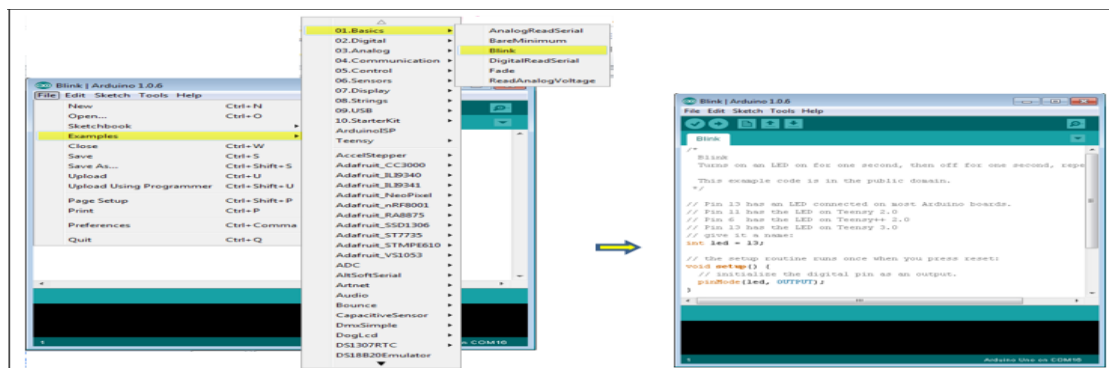


Fig 3.4.2 Launch Arduino IDE

Step 5: Open your first project.

- Once the software starts, you have two options:
- Create a new project.
- Open an existing project example.
- To create a new project, select File --> New.

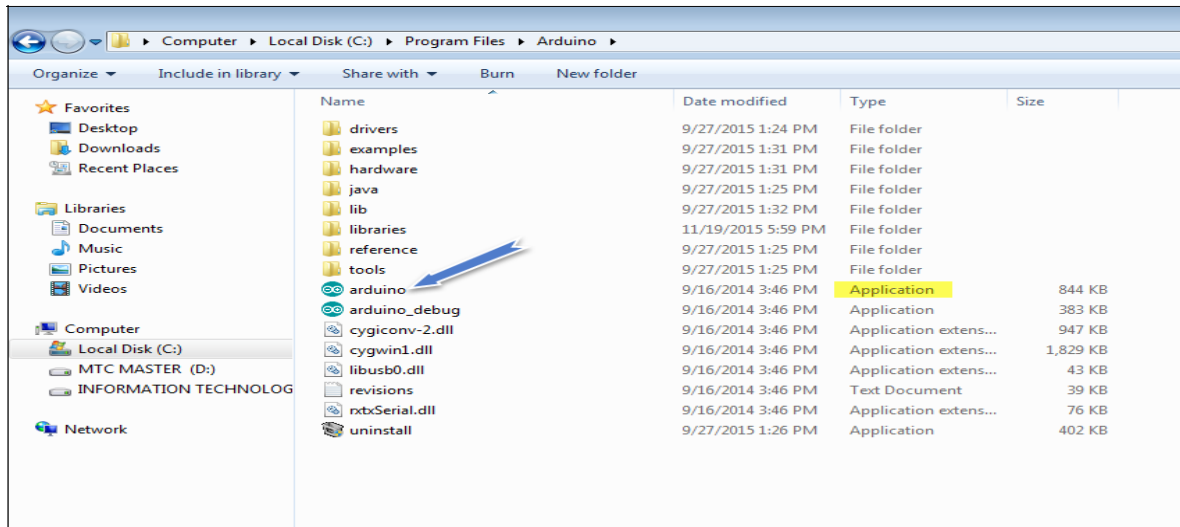


Fig 3.4.3 Creating a File

- To open an existing project example, select File -> Example -> Basics -> Blink.
- Here, we are selecting just one of the examples with the name **Blink**. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6: Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

- Go to Tools -> Board and select your board.

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 Design and Implementation :

This proposed system consists the equipment like Arduino mini pro, ultrasonic sensor, pref board, 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 pref board 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 volt mobile battery to this module. Then connect the elastic band. In the end after all the cnnctions are done to the Arduino board, upload the code to each arduino board and power the 4 other modules using a power bank.

4.2 Advantages:

- Low design time.

- Low production cost.
- This system is applicable for both the indoor and outdoor environment.
- Setting the destination is very easy.
- It is dynamic system.
- Less space.
- Low power consumption.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion:

Thus, this project proposed the design and architecture of a new concept of Arduino based Virtual Eye for the blind people. A simple, cheap, efficient, easy to carry, configurable, easy to handle electronic guidance system with many more amazing properties and advantages is proposed to provide constructive assistant and support for the blind and visually impaired persons. The system will be efficient and unique in its capability in specifying the source and distance of the objects that may encounter the blind. It is able to scan and detect the obstacles in the areas like left, right, and in front of the blind person regardless of its height or depth. With the proposed architecture, if constructed with at most accuracy, the blind will be able to move from one place to another without others help.

5.2 Future Scope:

The entire project can be made in the form of jacket, so that the device doesn't need to be wear one by one. By specifically suing the specialized boards that are designed, using them instead of Arduino and also by using high quality ultrasonic sensors makes and gives faster response which make the device capable of working in crowded places and thus this will be implemented in the future enhancement of this device.

CHAPTER 6

RESULT AND SCREENSHOTS

6.1 Result:

The presented system is designed and configured for the use of the blind and visually disabled people. This device is able to handle several states that the visually impaired people face. This device responds to the user in all the circumstances which is faced by the blind people with the help of the use of the Ultrasonic sensors and the Arduino Board. Case 1: When the obstacle or the object is in the left it will tell the user that: The obstacle is in left Case 2: When the obstacle is in right it will say: The obstacle in right. Case 3: When the obstacle is in front, the device will say: the obstacle is in front. Similarly for all the directions like left, right, back etc it will notify the user wearing it.

6.2 Screenshots:

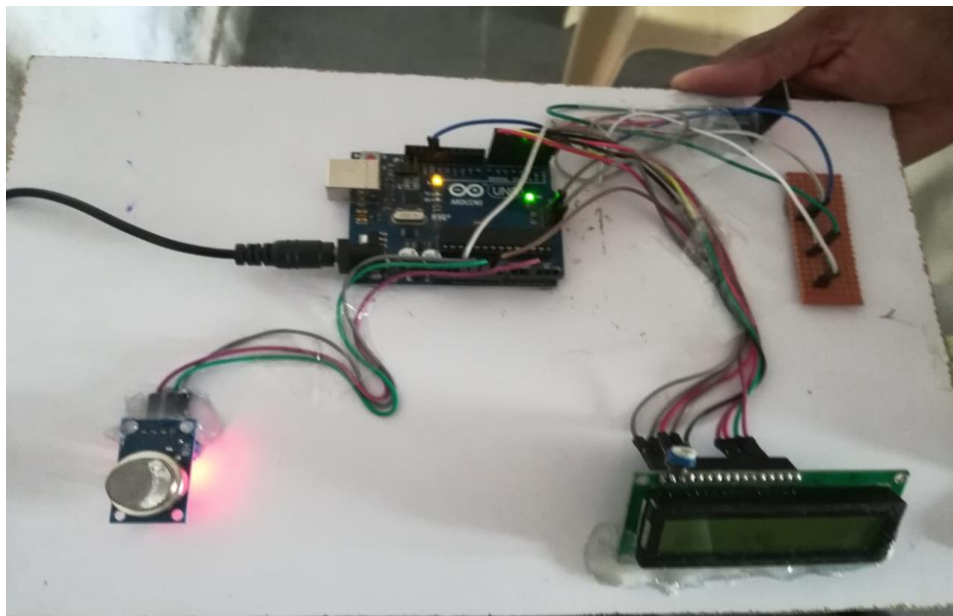


Fig 6.2.1 Output

REFERENCES

1. Shraga Shovel, Iwan Ulrich, Johann Borenstien. Nav Belt and the Guide Cane, IEEE Transactions on Robotics & Automation. 2003; 10(1):9-20.
2. Yuan D, Manduchi R. 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. Benjamin JM, Ali NA, Schepis AF. A laser cane for the blind”, Proceedings of San Diego Medical Symposium, 1973, 443-450.
4. Sabarish S. Navigation Tool for Visually Challenged using Microcontroller, International Journal of Engineering and Advanced Technology (IJEAT), 2013; 2(4):139-143.
5. Espinosa MA, Ungar S, Ochaíta E. 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, Shimi SL, Chatterji S. A Review on Obstacle Detection and Vision, International Journal of Science and Research Technology. 2015; 4(1):1-11.
7. Tahat AA. A wireless ranging system for the blind longcane 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. Bolgiano D, Meeks Jr E. A laser cane for the blind, IEEE Journal of Quantum Electronics. View at Google Scholar. 1967; 3(6):268.
9. Amjed Al-Fahoum S, Heba Al-Hmoud B, Ausaila AlFraihat A. A Smart Infrared Microcontroller-Based Blind 756 Guidance System”, Hindawi Transactions on Active and Passive Electronic Components. 2013; 3(2):1-7.
10. Przemyslaw Baranski, Maciej Polanczyk, Pawel Strumillo. A Remote Guidance System for the Blind, IEEE Transactions on Remote Guidance, 2010, pp. 386- 390.

APPENDIX

CODE:

```
#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
}

void loop() {
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
```

```
Serial.print(distance);  
Serial.println(" cm");  
}
```