MAJOR PROJECT REPORT

ON

**ULTRASONIC DISTANCE METER**

Submitted in partial fulfillment of the requirements

For the award of the degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS & COMMUNICATION ENGINEERING**

Submitted By

**Aakash Singh Aditya Nahata Rohan Singh**

***(00196202817) (00496202817) (01796202817)***

**Under the guidance of**

**MS. PRIYANKA, Assistant Professor**, **ECE Department**

****

**Department of Electronics & Communication Engineering**

**DR. AKHILESH DAS GUPTA INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY)**

**NEW DELHI – 110053**

**March-July, 2021**

**CERTIFICATE**

I/We hereby certify that the work that is being presented in the project report entitled **Ultrasonic Distance Meter** to the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Electronics & Communication Engineering from Dr. Akhilesh Das Gupta Institute of Technology & Management**, New Delhi. This is an authentic record of our own work carried out during a period from March, 2021 to July, 2021 under the guidance of **Ms. Priyanka**, **Assistant Professor in ECE department.**

The matter presented in this project has not been submitted by us for the award of any other degree elsewhere.

**Aakash Singh Aditya Nahata**

***(00196202817) (00496202817)***

**Rohan Singh**

## *(01796202817)*

This is to certify that the above statement made by the candidates is correct to the best of our knowledge.

**Ms. Priyanka,**  **Prof. (Dr.) Rajiv Sharma** **Assistant Professor, ECE Deptt H.O.D. ECE Deptt.**

The B.Tech Major Project Viva-Voce Examination of **Aakash Singh (00196202817), Aditya Nahata (00496202817), Rohan Singh (01796202817),** has been held on

## Mr. Dev Raj Gautam (Signature of External Examiner)

## Project Coordinator,ECE Deptt.

**ACKNOWLEDGEMENT**

I/We would like to acknowledge the contributions of the following persons, without whose help and guidance this report would not have been completed.

I/We acknowledge the counsel and support of our project guide **Ms. Priyanka, Assistant Professor, ECE department,** with respect and gratitude, whose expertise, guidance, support, encouragement, and enthusiasm has made this report possible. Their feedback vastly improved the quality of this report and provided an enthralling experience. I/We are indeed proud and fortunate to be supervised by her.

We are thankful to **Prof.** (**Dr.) Rajiv Sharma**, **H.O.D. ECE department, Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi** for his constant encouragement, valuable suggestions and moral support and blessings.

I/We are immensely thankful to our esteemed, **Prof. (Dr.) Sanjay Kumar, Director, Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi** for his never ending motivation and support.

I/We shall ever remain indebted to **Mr. Devraj Gautam, Project Coordinator ECE department** and faculty and staff members of Dr. Akhilesh Das Gupta Institute of Technology & Management, New Delhi.

Finally, yet importantly, I/We would like to express our heartfelt thanks to God, our beloved parents for their blessings, our friends/classmates for their help and wishes for the successful completion of this project.

**Aakash Singh Aditya Nahata**

***(00196202817) (00496202817)***

**Rohan Singh**

***(01796202817)***

# ABSTRACT

# There is a rapid development in technology which is influencing the human life in several aspects due to rapid development in different fields but we still need to adopt that technology such that we can make human life easier to live. We know that in automobiles sector many countries are doing great and launching new products (vehicles).

# Instead of having many advantages and features that they may have some disadvantages too and one of them in vehicles is reverse car parking. To rectify this problem we have developed a system which will help us to park our vehicles in reverse direction without collision with any obstacle in its vicinity. The aim of our project is to present such a design which can identify the obstacles that comes in the range while parking the car in reverse direction.

# We have also used and expanded the applications of ultrasonic sensor by making a covid-19 protocol enforcer. Using a human to limit the number of people in a particular place would not be as effective as using an automatic system due to the lack of staff members in some areas. This prototype can be used in real-life situations by replacing the servo motor with a solid state relay module.

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| ACKNOWLEDGEMENT…………………………………………………………. | ii |
| CANDIDATE’S DECLARATION………………………………………………… | iii |
| ABSTRACT………………………………………………………………………… | iv |
| LIST OF FIGURES………………………………………………………………… | v |
| LIST OF TABLES…………..……………………………………………………… | vi |
| 1.INTRODUCTION | 1 |
| 1.1. GENERAL…………………………………………………………… | 1 |
| 1.2. HISTORY…………………………………………………………….. | 2 |
| 1.3. MECHANICAL CAUSE.……………………………………………. | 3 |
| 1.4 BENEFITS….………………………………………………………… | 4 |
| 2. PROJECT WORK | 5 |
| 2.1 HARDWARE…………………………………………………………. | 5 |
| 2.1.1 ULTRASONIC SENSOR (HC-SR04)..………………………… | 5 |
| 2.2.2 LCD……………………………………………………………... | 8 |
| 2.2.3 ARDUINO UNO..………………………………………………. | 12 |
| 2.2 OBSTACLE DETECTOR GLOVE……………………………………  2.3 COVID 19 PROTOCOL ENFORCER………………………………… | 20  31 |
| 3. RESULT | 33 |
| 4. CONCLUSION | 34 |
| 5. FUTURE SCOPE | 35 |
| REFERENCES………………………………………………………………………  APPENDICE………………………………………………………………………... | 36  40 |

# LIST OF FIGURES Page No.

Figure 1.1 Ultrasonic Sensor 2

Figure 1.2 Reverse car parking system 3 . Figure 2.1 ultrasonic sensor HC-SR04. 5

Figure 2.2 Signal sending and receiving in ultrasonic sensor. 7

Figure 2.3 Timing Diagram. 7

Figure 2.4 PIN diagram of LCD 9

Figure 2.5 Actual model 19

Figure 2.6 Circuit Design 27

Figure 2.7 Hardware used in Glove 28

Figure 2.8 Obstacle Detector Glove 29

Figure 2.9 Future Scope 30

Figure 2.10 Covid 19 Protocol Enforcer 32

Figure 3.1 LCD Output 33

# LIST OF TABLES Page No.

Table 2.1 Electric Parameter 6

Table 2.2 Pin configuration of LCD 10

Table 2.3 Important command codes for LCD 11

Table 2.4 Technical Specifications 13

Table 2.5 Intensity of buzzer and LCD output with distance 33

# CHAPTER 1

# INTRODUCTION

# 1.1 General

A reverse car parking defect is a term used to describe several scenarios in which cars with automatic transmission can fail to properly engage the parking mechanism, causing the vehicle to unintentionally roll, sometimes resulting in injury or vehicular accidents. The term has significance in product liability law, and in a number of major cases in the United States have been brought in which car manufacturers were accused of negligence for not addressing an alleged dangerous flaw in the transmission.

A "park-to-reverse" situation involves a driver who believes that they have shifted into "park" and believing so, and the vehicle not moving when they pull their foot off the brake, proceeds to exit the vehicle. There can then be a delay in vehicle movement sufficient for the driver to either fully or partially exit the vehicle before the vehicle moves. Typically, the vehicle will move backwards. However, when on certain vehicles the shift selector can be placed between the detented park and reverse gear positions; i.e. in "false park" the transmission is in hydraulic neutral, without the parking pawl engaged. As such the vehicle can also roll either forward or back in neutral. While less common, transmissions with the defect, can also be shifted to between "neutral" and "drive", and then self shift into "drive" or roll (called a "neutral to drive" accident).

"Park-to-Reverse" issues are nearly always caused by several possible design flaws in a vehicle's transmission which makes it possible for a driver to unknowingly place the vehicle's shift selector into a position in between the "park" and "reverse" gear positions. Yet rather than being in "park', this area is a transitional zone between gears, which is sometimes called "false park".

Parking sensors use a type of sonar. The term sonar is an acronym for sound navigation and radar; it's used for calculating the distance and/or direction of an object from the time it takes for a sound wave to travel to the target and back.

An ultrasonic sensor is a speaker or microphone that emits or receives ultrasound. There is also a type that can handle both emission and reception. Vehicle parking sensors are equipped with this type of sensor.

Ultrasonic sensors initially found use in vehicles for detecting obstacles when parking but it is now evolving into an automatic parking system**.**

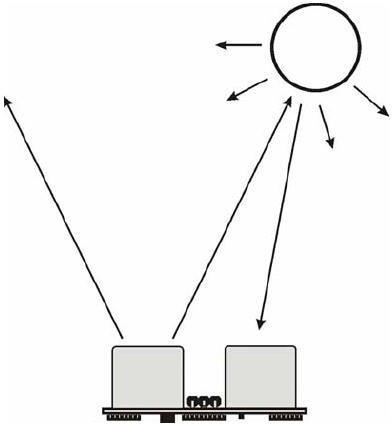


Figure 1.1: Ultrasonic sensor [1]

## History

Over the years, car parking systems and the accompanying technologies have increased and diversified. Reverse Car parking systems have been around almost since the time cars were invented. In any area where there is a significant amount of traffic, there are car parking systems.

Reverse Car Parking systems were developed in the early 20th century in response to the need for storage space for vehicles.

In the 1920s, forerunners of automated parking systems appeared in U.S. cities like Los Angeles, Chicago, New York City and Cincinnati. Some of these multi-storey structures are still standing, and have been adapted for new uses. One of the Kent Automatic Parking Garages in New York (now known as the Sofia Apartments) is an Art Deco landmark that was converted into luxury condominiums in 1983. A system that is now found all over Japan — the “ferris-wheel,” or paternoster system — was created by the Westinghouse Corporation in 1923 and subsequently built in 1932 on Chicago’s Monroe Street. The Nash Motor Company created the first glass- enclosed version of this system for the Chicago Century of Progress Exhibition in 1933, and it was the precursor to a more recent version, the Smart Car Towers in Europe.

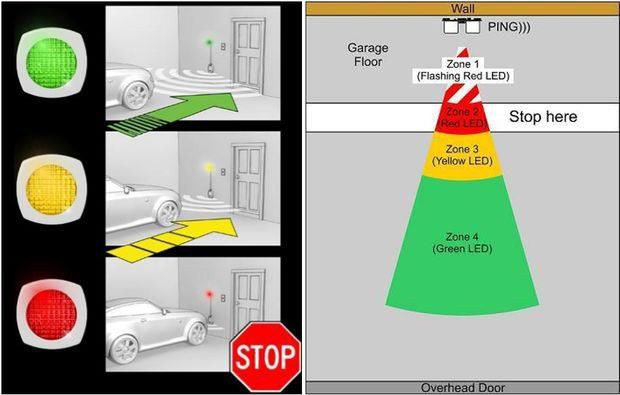


Figure 1.2: Reverse car parking system [2],[3]

## Mechanical cause

With some automatic transmissions, it was possible to place the shift selector at any point, either in an intended gear or between a gear. Because of the possible safety issue of this, and because driving a vehicle not fully in a gear over a long period of time could damage the transmission, automakers developed what is called the "detent system." The system of detents was often used in conjunction with the "push button" shifters used on many automatics in the 1950s. Typical detent systems use either a detent spring and ball or a cantilever spring and this spring moves up and down over a series of teethed gears (called a "rooster comb" for how it looks, or an "inner manuel lever") turning the rooster comb to fine center the transmission in the intended gear position at the bottom of each gear.

However, if the spring is too weak to always move the rooster comb to the bottom of the trough between the teeth, the vehicle can be left between gears. On certain U.S. car manufacturers' vehicles, the problem is made worse as there is a flat spot between "Park" and "Reverse" detents where the ball can rest, also resulting in a "false park".

## Benefits

There are several advantages of employing a car park system for business owners and vehicle drivers. They offer convenience for vehicle users and efficient usage of space for urban-based companies. Automated car park systems save time, money, space and simplify the often tedious task of parking. Auto car lifts move vehicles into safe and secure storage areas until they are needed.

# CHAPTER 2

# PROJECT WORK

## Hardware

* + 1. **Ultrasonic sensor(HC-SR04)**

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

1. Using IO trigger for at least 10us high level signal,
2. The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
3. IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

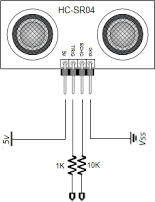


Figure 2.1: Ultrasonic sensor HC-SR04.

Wire connecting direct as following:

* + 5V Supply
  + Trigger Pulse Input
  + Echo Pulse Output
  + 0V Ground

Table 2.1: Electric Parameter

|  |  |
| --- | --- |
| Working Voltage | DC 5 V |
| Working Current | 15mA |
| Working Frequency | 40Hz |
| Max Range | 4m |
| Min Range | 2cm |
| MeasuringAngle | 15 degree |
| Trigger Input Signal | 10uS TTL pulse |
| Echo Output Signal | Input TTL lever signal and the range in proportion |
| Dimension | 45\*20\*15mm |

### Timing diagram:

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.

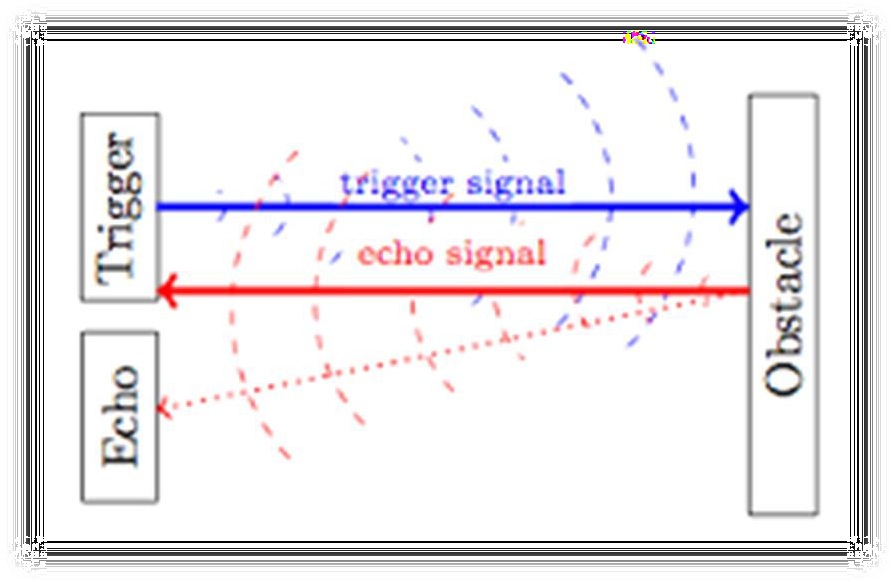


Figure 2.2: Signal sending and receiving in ultrasonic sensor.[4]

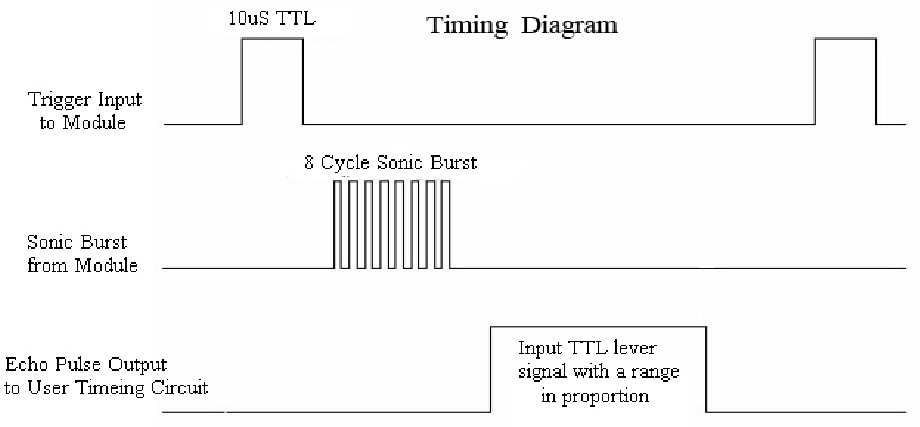


Figure 2.3: Timing Diagram. [5],[6]

### Precautions

* + The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
  + When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

## LCD

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. A full list of the characters and symbols is printed on pages 7/8 (note these symbols can vary between brand of LCD used). This booklet provides all the technical specifications for connecting the unit, which requires a single power supply (+5V).

We come across LCD displays everywhere around us. Computers, calculators, television sets, mobile phones, digital watches use some kind of display to display the time. An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIYs and circuits.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

Further Information: Available as an optional extra is the Serial LCD Firmware, which allows serial control of the display. This option provides much easier connection and use of the LCD module. The firmware enables microcontrollers (and microcontroller based systems such as the PICAXE) to visually output user instructions or readings onto an LCD module. All LCD commands are transmitted serially via a single microcontroller pin. The firmware can also be connected to the serial port of a computer.

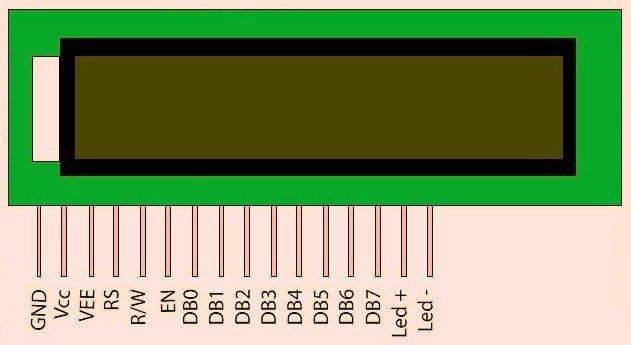


Figure2.4: PIN diagram of LCD

|  |  |  |
| --- | --- | --- |
| 13 |  | DB6 |
| 14 | DB7 |
| 15 | Backlight VCC (5V) | Led+ |
| 16 | Backlight Ground (0V) | Led- |

Table 2.2: Pin configuration of LCD

|  |  |  |
| --- | --- | --- |
| Pin No | Function | Name |
| 1 | Ground (0V) | Ground |
| 2 | Supply voltage; 5V (4.7V – 5.3V) | Vcc |
| 3 | Contrast adjustment; through a variable resistor | VEE |
| 4 | Selects command register when low; and data register when high | Register Select |
| 5 | Low to write to the register; High to read from the register | Read/write |
| 6 | Sends data to data pins when a high to low pulse is given | Enable |
| 7 | 8-bit data pins | DB0 |
| 8 | DB1 |
| 9 | DB2 |
| 10 | DB3 |
| 11 | DB4 |
| 12 | DB5 |

RS(Register select):A 16X2 LCD has two registers, namely, command and data. The register select is used to switch from one register to other. RS=0 for command register, whereas RS=1 for data register.

Command Register: The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. Processing for commands happen in the command register.

Data Register: The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. When we send data to LCD it goes to the data register and is processed there. When RS=1, data register is selected.

Table 2.3: Important command codes for LCD

|  |  |  |
| --- | --- | --- |
| Sr.No. | Hex Code | Command to LCD instruction Register |
| 1 | 01 | Clear display screen |
| 2 | 02 | Return home |
| 3 | 04 | Decrement cursor (shift cursor to left) |
| 4 | 06 | Increment cursor (shift cursor to right) |
| 5 | 05 | Shift display right |
| 6 | 07 | Shift display left |
| 7 | 08 | Display off, cursor off |
| 8 | 0A | Display off, cursor on |
| 9 | 0C | Display on, cursor off |

|  |  |  |
| --- | --- | --- |
| 10 | 0E | Display on, cursor blinking |
| 11 | 0F | Display on, cursor blinking |
| 12 | 10 | Shift cursor position to left |
| 13 | 14 | Shift cursor position to right |
| 14 | 18 | Shift the entire display to the left |
| 15 | 1C | Shift the entire display to the right |
| 16 | 80 | Force cursor to beginning ( 1st line) |
| 17 | C0 | Force cursor to beginning ( 2nd line) |
| 18 | 38 | 2 lines and 5×7 matrix |

### Displaying Custom Characters on 16X2 LCD:

Generating custom characters on LCD is not very hard. It requires the knowledge about custom generated random access memory (CG-RAM) of LCD and the LCD chip controller. Most LCDs contain Hitachi HD4478 controller. CG-RAM is the main component in making custom characters. It stores the custom characters once declared in the code. CG-RAM size is 64 byte providing the option of creating eight characters at a time. Each character is eight byte in size.

CG-RAM address starts from 0x40(Hexadecimal) or 64 in decimal. We can generate custom characters at these addresses. Once we generate our characters at these addresses, now we can print them on the LCD at any time by just sending simple commands to the LCD.

## Arduino UNO

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worring too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Table 2.4: Technical Specifications

|  |  |
| --- | --- |
| 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 |

### Programming in arduino

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).

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.

You can also bypass the bootloader 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 rese ing 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.

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.

### Differences with other boards

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

### Power supply

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

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply 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.

### System Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by 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 ports

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 is it 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.

### Communication with computer

Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

### Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

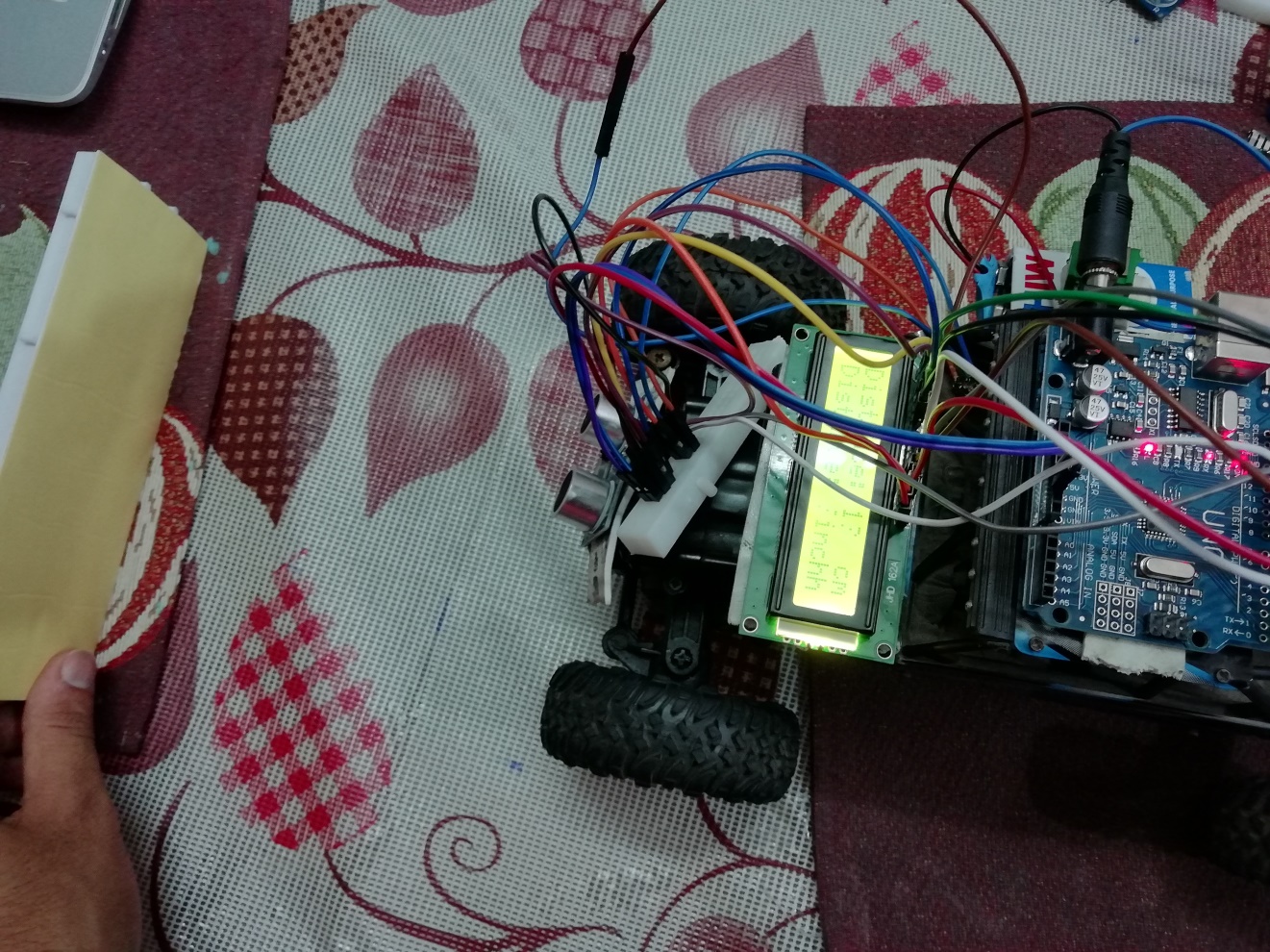


Figure 2.5: Actual model.

## 2.2 OBSTACLE DETECTOR GLOVES

**2.2.1 INTRODUCTION**

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 behavior 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. Eyes are prime sense of organ in perceiving the outside environment; dysfunction of such prime sense organ severely affects 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 Obstacle Detector Glove for 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. Obstacle Detector Glove 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 Obstacle Detector Glove for 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. Nowadays 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.

**2.2.2 Technical specifications**

* Microcontroller: Microchip ATmega328P
* Operating Voltage: 5 Volts
* Input Voltage: 7 to 20 Volts
* Digital I/O Pins: 14 (of which 6 can provide PWM output)
* UART: 1
* I2C: 1
* SPPI: 1
* Analog Input Pins: 6
* DC Current per I/O Pin: 20 mA
* DC Current for 3.3V Pin: 50 mA
* Flash Memory: 32 KB of which 0.5 KB used by bootloader
* SRAM: 2 KB
* EEPROM: 1 KB
* Clock Speed: 16 MHz
* Length: 68.6 mm
* Width: 53.4 mm
* Weight: 25 g

**Headers**

**General pin functions**

* **LED**: 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 is off.
* **VIN**: The input voltage to the Arduino/Genuino board when it is 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 - 20V), the USB connector (5V),

or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.

* **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.
* **Reset**: Typically used to add a reset button to shields that block the one on the board.

**Special pin functions**

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pinMode(), digitalWrite(), and digitalRead() functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 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 the range using the AREF pin and the analogReference() function.

In addition, some pins have specialized functions:

* **Serial** / UART: pins 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**: 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** (pulse-width modulation): pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.
* **SPI** (Serial Peripheral Interface): pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
* **TWI** (two-wire interface) / I²C: pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
* **AREF** (analog reference): Reference voltage for the analog inputs.

**2.2.3 Code used for project**

#define trigPin 13

#define echoPin 12

#define buzzer 11

#define motor 10

void setup()

{

Serial.begin (9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(buzzer, OUTPUT);

}

void loop()

{

long duration, distance;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration/2) / 29.1;

if (distance < 30)

{

digitalWrite(buzzer,HIGH);

digitalWrite(motor,HIGH);

}

else

{

digitalWrite(buzzer,LOW);

}

Serial.print(distance);

Serial.println(" cm");

delay(500);

}

## C:\Users\Suraj\Documents\mcode1.jpg

## 

We have tested this by limiting the range of 30 cm distances we can increase its distance.

**How it works?**

As the ultrasonic sensor senses, the distance as input is taken, the logic calculations are carried out, then the output will be shown in mechanical forms such as vibrations and buzzer effect i.e. as obstacles approach the sensor vibration and buzzer starts.

**2.2.4 CIRCUIT DIAGRAM**

## C:\Users\Suraj\AppData\Local\Microsoft\Windows\INetCache\Content.Word\4-Figure6-1.png

## Figure 2.6: Circuit design for proposed hand glove for the blind. [7]

**2.2.5 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.

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.

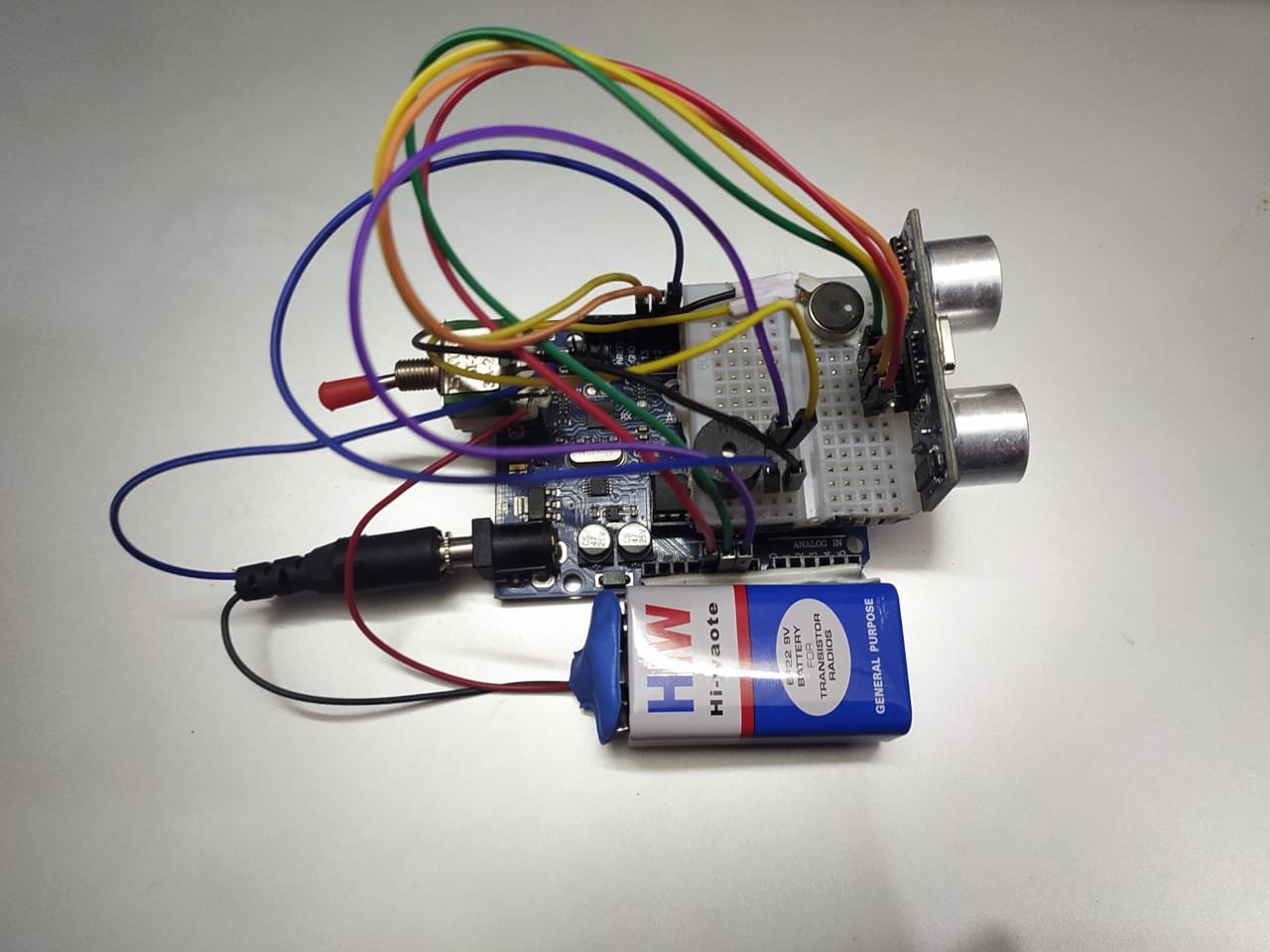


Figure 2.7 : Hardware of project

****

Figure 2.8: Obstacle Detector Glove. [8]

* 1. **2.2.6 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.

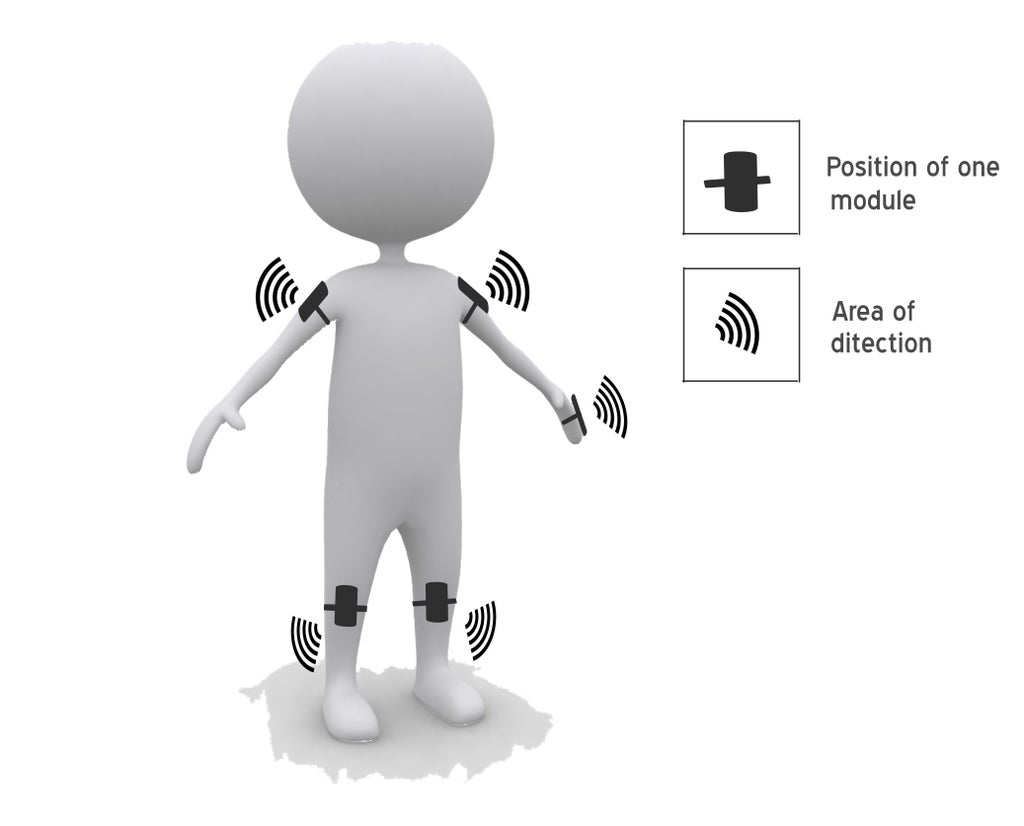


Figure 2.9: Future scope of this technology. [9]

## 2.3 COVID -19 PROTCOL ENFORCER

**2.3.1 About This Project**

The World Health Organization, WHO, has advised people to stay away from crowded places to protect themselves from the spread of Coronavirus disease. Even though people practice social distancing, it may not be effective when they are present in crowded places. I came up with this idea after I read the WHO guidelines about preventing the spread of COVID-19.

**2.3.2 Aim/ Objective**

This project is an automatic, cheap and effective way of limiting the crowd in public places - such as shopping malls, supermarkets, offices - and public transport vehicles, such as buses and trains.

Using a human to limit the number of people in a particular place would not be as effective as using an automatic system due to the lack of staff members in some areas. This prototype can be used in real-life situations by replacing the servo motor with a solid state relay module.

The solid state relay module will control the motor operating the automatic sliding doors in buildings and vehicles. There will be a slight change in code when replacing the servo motor.

**2.3.3 How It Works**



Figure 2.10: Covid 19 Protocol Enforcer.

If a person is going to enter the building or vehicle, he/ she can wave or hover his/ her hand over the ultrasonic/ IR tracking sensor module. If an IR tracking sensor module is used, it will send a LOW signal to the Arduino Uno microcontroller and according to my program, the door will be opened.

In our model, We have used a servo motor to open/ close the door. The door will be opened when the servo motor rotates 90 degrees.

If an ultrasonic sensor module is used, the door will be opened when the sensor detects an obstacle within 5 cm away from itself. The door will remain open for 5 seconds and the value stored in the count variable will increase by one after the door is closed. The count variable denotes the number of people inside a building or vehicle.

If the value stored in the count variable reaches the maximum occupancy value, the LCD display module will show that no one could enter and the door will remain closed until someone leaves the building. We have attached an IR tracking sensor module on the inside of the box (modelled as the building/ vehicle) as well. The process will be the same as that mentioned above, but the difference is that the value stored in the count variable will decrease by one as a person is leaving the building.

\*Note: The maximum occupancy in this case is the maximum number of people who can enter the building or vehicle without crowding it.

# CHAPTER 3

# RESULT

## Buzzer output

The output of buzzer depends on the minimum value of the distance calculated by thee sensors. Further the intensity of the buzzer depends on the value of minimum distance. This is divided in four categories-

Table 3.1: Intensity of buzzer and LCD output with distance.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.no | Value of distance(X) | Intensity of buzzer | LCD output |
| 1. | X<10 | Very high | In range |
| 2. | 10<X<20 | High | In range |
| 3. | 20<X<30 | Moderate | In range |
| 4. | X<30 | low | Out of range |



Figure 3.1: LCD output.

# CHAPTER 4

# CONCLUSION

Although science and technology has emerged a lot and is still progressing rather be in computer technology, automobiles etc. We know that in automobiles sector many countries are doing great and launching new products (vehicles). Instead of having many advantages and features they may have some disadvantages too and one of them in vehicles is reverse car parking. To rectify this problem we have developed a system which will help us to park our vehicles in reverse direction without collision with any obstacle in its vicinity.

By using following several components we have finally completed it.

* + - Arduino UNO
    - Ultrasonic sensor(HC-SR04)
    - LCD(16\*2)
    - PCB
    - Battery(9v)

In this we used three ultrasonic sensors, which will sense any obstacle in its desired range, with the help of arduino controller.

Arduino controller is used control these ultrasonic sensors through our preinstalled program, we also use LCD to display the distance for more accuracy in parking our vehicle, buzzers are used for indication of any obstacles which lie under its range.

# CHAPTER 5

# FUTURE SCOPE

* Using it in robots.
* Making it available for all types of vehicle.
* Using several sensor to get proper reading around the whole car not in just reverse direction.
* Automatic Car Parking.
* Effective implementation on Intelligent Parking Assist System (IPAS), also known as the Advanced Parking Guidance System (APGS).

# REFERENCES

1. D. Marioli, C. Narduzzi, C. Offelli, D. Petri, E. Sardini, and A. Taroni, “Digital time-of-flight measurement for ultrasonic sensors,” IEEE.Trans. Instrum. Meas., vol. 41, pp. 93–97, Feb. 1992.
2. C. Cai and P. L. Regtien, “Accurate digital time-of-flight measurement using self-interference,” IEEE Trans. Instrum. Meas., vol. 42, pp.990–994, Dec. 1993.
3. ENV 13 005, “Guide to the expression of uncertainty in measurement,”, May 1999
4. F. Gueuning, M. Varlan, C. Eugène, and P. Dupuis, “Accurate distance measurement by an autonomous ultrasonic system combining time-offlight and phase-shift methods,” in Proc. IMTC, vol. I, Brussels, Belgium, June 4–6, 1996, pp. 399–404.
5. G. Andria, F. Attivissimo, and A. Lanzolla, “Digital measuring techniques for high accuracy ultrasonic sensor application,” in Proc. IMTC, vol. II, St. Paul, MN, May 18–21, 1998, pp. 1056–1061.
6. A. Carullo, F. Ferraris, S. Graziani, U. Grimaldi, and M. Parvis, “Ultrasonic distance sensor improvement using a two-level neural network,”IEEE Trans. Instrum. Meas., vol. 45, pp. 677–682, April 1996.

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

[8] 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.

[9] Amjed Al-Fahoum S, Heba Al-Hmoud B, Ausaila Al- Fraihat A. A Smart Infrared

Microcontroller-Based Blind Guidance System”, Hindawi Transactions on Active and Passive Electronic Components. 2013; 3(2):1-7.

**Appendices**

#include <NewPing.h>

#include <LiquidCrystal.h>

#define TRIGGER\_PIN1 A0

#define ECHO\_PIN1 A1

#define MAX\_DISTANCE1 500

#define TRIGGER\_PIN2 A2

#define ECHO\_PIN2 A3

#define MAX\_DISTANCE2 500

#define TRIGGER\_PIN3 A4

#define ECHO\_PIN3 A5

#define MAX\_DISTANCE3 500

#define BUZZER 6

intpwm = 0; int a;

int a1;

LiquidCrystallcd(12, 11, 5, 4, 3, 2);

NewPingsonar1(TRIGGER\_PIN1, ECHO\_PIN1, MAX\_DISTANCE1);

NewPingsonar2(TRIGGER\_PIN2, ECHO\_PIN2, MAX\_DISTANCE2); NewPingsonar3(TRIGGER\_PIN3, ECHO\_PIN3, MAX\_DISTANCE3);

void setup() { Serial.begin(9600); lcd.begin(16, 2); pinMode(BUZZER, OUTPUT);

}

void loop() {

if (sonar1.ping\_cm() > 0)

{

lcd.setCursor(0, 0);

{

lcd.print("S1:"); lcd.print(sonar1.ping\_cm()); lcd.print(" ");

}

}

Serial.print("Ping2: "); Serial.print(sonar2.ping\_cm()); Serial.print("cm");

Serial.print("\t");

if (sonar2.ping\_cm() > 0)

{

lcd.setCursor(10, 0);

{

lcd.print("S2:"); lcd.print(sonar2.ping\_cm()); lcd.print(" ");

}

}

Serial.print("Ping3: "); Serial.print(sonar3.ping\_cm()); Serial.println("cm");

if (sonar3.ping\_cm() > 0)

{

lcd.setCursor(0, 1);

{

lcd.print("S3:"); lcd.print(sonar3.ping\_cm()); lcd.print(" ");

}

}

a1 = min(sonar1.ping\_cm(),sonar2.ping\_cm()); a = min(sonar3.ping\_cm(),a1); digitalWrite(BUZZER, LOW);

if (a > 10 && a < 20)

{

digitalWrite(BUZZER, HIGH); Serial.println("BUZZER BEEPS3"); lcd.setCursor(8, 1);

lcd.print("In Range"); delay(300); digitalWrite(BUZZER, LOW); delay(300); digitalWrite(BUZZER, HIGH); delay(300);

}

if (a > 20 && a < 30)

{

digitalWrite(BUZZER, HIGH); Serial.println("BUZZER BEEPS4"); lcd.setCursor(8, 1);

lcd.print("In Range"); delay(500); digitalWrite(BUZZER, LOW); delay(500); digitalWrite(BUZZER, HIGH); delay(500);

}

if (a > 0 && a < 10)

{

digitalWrite(BUZZER, HIGH); Serial.println("BUZZER BEEPS2"); lcd.setCursor(8, 1);

lcd.print("In Range");

}

if (a > 30)

{

digitalWrite(BUZZER, LOW); Serial.println(" No BUZZER"); lcd.setCursor(8, 1); lcd.print("Out Range");

}}