**CHAPTER 1**

**INTRODUCTION**

**1.1 HISTROY OF ARDUINO**

The Arduino project was started at the [Interaction Design Institute Ivrea](https://en.wikipedia.org/wiki/Interaction_Design_Institute_Ivrea) (IDII) in [Ivrea](https://en.wikipedia.org/wiki/Ivrea), Italy. At that time, the students used a [BASIC Stamp](https://en.wikipedia.org/wiki/BASIC_Stamp) [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) at a cost of $50, a considerable expense for many students. In 2003 Hernando Barragán created the development platform [*Wiring*](https://en.wikipedia.org/wiki/Wiring_(development_platform)) as a Master's thesis project at IDII, under the supervision of Massimo Banzi and [Casey Reas](https://en.wikipedia.org/wiki/Casey_Reas). Casey Reas is known for co-creating, with Ben Fry, the [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) development platform. The project goal was to create simple, low cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a [printed circuit board](https://en.wikipedia.org/wiki/Printed_circuit_board) (PCB) with an [ATmega](https://en.wikipedia.org/wiki/ATmega)168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller. In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the project and renamed it *Arduino*.

The initial Arduino core team consisted of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis, but Barragán was not invited to participate.

Following the completion of the Wiring platform, lighter and less expensive versions were distributed in the open-source community.

It was estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands.

In October 2016, Federico Musto, Arduino's former CEO, secured a 50% ownership of the company. In April 2017, [Wired](https://en.wikipedia.org/wiki/Wired_(magazine)) reported that Musto had "fabricated his academic record.... On his company's website, personal LinkedIn accounts, and even on Italian business documents, Musto was until recently listed as holding a PhD from the Massachusetts Institute of Technology. In some cases, his biography also claimed an MBA from New York University." Wired reported that neither university had any record of Musto's attendance, and Musto later admitted in an interview with Wired that he had never earned those degrees.

Around that same time, Massimo Banzi announced that the Arduino Foundation would be "a new beginning for Arduino." But a year later, the Foundation still hasn't been established, and the state of the project remains unclear.

The controversy surrounding Musto continued when, in July 2017, he reportedly pulled many [Open source](https://en.wikipedia.org/wiki/Open_source_model) licenses, schematics, and code from the Arduino website, prompting scrutiny and outcry.

In October 2017, Arduino announced its partnership with [ARM Holdings](https://en.wikipedia.org/wiki/ARM_Holdings) (ARM). The announcement said, in part, "ARM recognized independence as a core value of Arduino ... without any lock-in with the [ARM architecture](https://en.wikipedia.org/wiki/ARM_architecture).” Arduino intends to continue to work with all technology vendors and architectures.

**1.2 Features of Arduino**

The **features of Arduino Uno ATmega328** includes the following.

* The operating voltage is 5V
* The recommended input voltage will range from 7v to 12V
* The input voltage ranges from 6v to 20V
* Digital input/output pins are 14
* Analog i/p pins are 6
* DC Current for each input/output pin is 40 mA
* DC Current for 3.3V Pin is 50 mA
* Flash Memory is 32 KB
* SRAM is 2 KB
* EEPROM is 1 KB
* CLK Speed is 16 MHz

**1.2.1** **POWER SUPPPLY**

The **Arduino Uno power supply** can be done with the help of a USB cable or an external power supply. The external power supplies mainly include AC to DC adapter otherwise a battery. The adapter can be connected to the Arduino Uno by plugging into the power jack of the Arduino board. Similarly, the battery leads can be connected to the Vin pin and the GND pin of the POWER connector. The suggested voltage range will be 7 volts to 12 volts.

**Input & Output**

The 14 digital pins on the Arduino Uno can be used as input & output with the help of the functions like pinMode(), digitalWrite(), & Digital Read().

**Pin1 (TX) & Pin0 (RX) (Serial):** This pin is used to transmit & receive TTL serial data, and these are connected to the ATmega8U2 USB to TTL Serial chip equivalent pins.

**Pin 2 & Pin 3 (External Interrupts):** External pins can be connected to activate an interrupt over a low value, change in value.

**Pins 3, 5, 6, 9, 10, & 11 (PWM):** This pin gives 8-bit PWM o/p by the function of analogWrite().

**SPI Pins (Pin-10 (SS), Pin-11 (MOSI), Pin-12 (MISO), Pin-13 (SCK):** These pins maintain SPI-communication, even though offered by the fundamental hardware, is not presently included within the Arduino language.

**Pin-13(LED):** The inbuilt LED can be connected to pin-13 (digital pin). As the HIGH-value pin, the light emitting diode is activated, whenever the pin is LOW.

**Pin-4 (SDA) & Pin-5 (SCL) (I2C):** It supports TWI-communication with the help of the Wire library.

**AREF (Reference Voltage):** The reference voltage is for the analog i/ps with analogReference ().

**Reset Pin:** This pin is used for resetting (RST) the microcontroller.

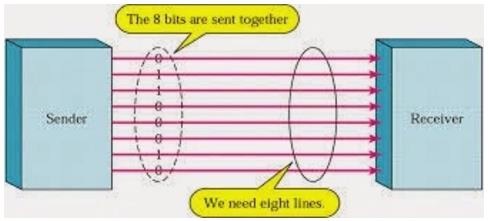
**Memory**

The memory of this Atmega328 Arduino microcontroller includes flash memory-32 KB for storing code, SRAM-2 KB EEPROM-1 KB.

## **1.3 COMMUNICATION**

## **1.3.1 Parallel Communication**

Parallel connection between the Arduino and peripherals via input/output ports is the ideal solution for shorter distances up to several meters. However, in other cases when it is necessary to establish communication between two devices for longer distances it is not possible to use parallel connection. Parallel interfaces transfer multiple bits at the same time. They usually require buses of data – transmitting **(FIG1.3)** across eight, sixteen, or more wires. Data is transferred in huge, crashing waves of 1’s and 0’s.



**FIG1.3(parallel connection)**

### Advantages and Drawbacks of Parallel Communication

Parallel communication certainly has its advantages. It is faster than serial, straightforward, and relatively easy to implement. However, it requires many input/output (I/O) ports and lines. If you have ever had to move a project from a basic Arduino Uno to a Mega, you know that the I/O lines on a microprocessor can be precious and few. Therefore, we prefer serial communication, sacrificing potential speed for pin real estate.

## **1.3.2 Serial Communication Modules**

Today, most Arduino boards are built with several different systems for serial communication as standard equipment.

Which of these systems are used depends on the following factors

* How many devices the microcontroller has to exchange data with?
* How fast the data exchange has to be?
* What is the distance between these devices?
* Is it necessary to send and receive data simultaneously?

One of the most important things concerning serial communication is the **Protocol**, which should be strictly observed. It is a set of rules, which must be applied such that the devices can correctly interpret data they mutually exchange. Fortunately, Arduino automatically takes care of this, so that the work of the programmer/user is reduced to simple write (data to be sent) and read (received data).

## **1.3.3 Types of Serial Communications**

Serial communication can be further classified as −

* **Synchronous** − Devices that are synchronized use the same clock and their timing is in synchronization with each other.
* **Asynchronous** − Devices that are asynchronous have their own clocks and are triggered by the output of the previous state.

It is easy to find out if a device is synchronous or not. If the same clock is given to all the connected devices, then they are synchronous. If there is no clock line, it is asynchronous.

For example, UART (Universal Asynchronous Receiver Transmitter) module is asynchronous.

The asynchronous serial protocol has a number of built-in rules. These rules are nothing but mechanisms that help ensure robust and error-free data transfers. These mechanisms, which we get for eschewing the external clock signal, are −

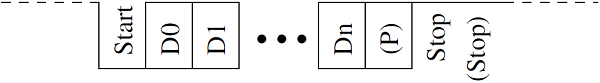
* Synchronization bits
* Data bits
* Parity bits
* Baud rate

### Synchronization Bits

The synchronization bits are two or three special bits transferred with each packet of data. They are the start bit and the stop bit(s). True to their name, these bits mark the beginning and the end of a packet respectively.

There is always only one start bit, but the number of stop bits is configurable to either one or two (though it is normally left at one).

The start bit is always indicated by an idle data line going from 1 to 0, while the stop bit(s) will transition back to the idle state by holding the line at 1.



### Data Bits

The amount of data in each packet can be set to any size from 5 to 9 bits. Certainly, the standard data size is your basic 8-bit byte, but other sizes have their uses. A 7-bit data packet can be more efficient than 8, especially if you are just transferring 7-bit ASCII characters.

### Parity Bits

The user can select whether there should be a parity bit or not, and if yes, whether the parity should be odd or even. The parity bit is 0 if the number of 1’s among the data bits is even. Odd parity is just the opposite.

### Baud Rate

The term baud rate is used to denote the number of bits transferred per second [bps]. Note that it refers to bits, not bytes. It is usually required by the protocol that each byte is transferred along with several control bits. It means that one byte in serial data stream may consist of 11 bits. For example, if the baud rate is 300 bps then maximum 37 and minimum 27 bytes may be transferred per second.

**1.4 OUTLINE OF PROJECT**

**INTRODUCTION**

Bats are wonderful creatures. Blind from the eyes but the vision is sharper than humans, Ultrasonic ranging is the technique used by bats. Ultrasonic sensor provides an easy way in distance measurement. The sensor is perfect for distance measurements between moving or stationary objects. Ultrasonic Sensor measure the distance of the objects in air through non-contact technique. They measure distance without damage and are easy to use and reliable. These distance measurement sensors connect with all common types of automation and telemetry equipment. Machinery and processes in a wide range of industries use distance measurement sensors where size or position feedback is required. Distance measurement sensors are used to control or indicate the position of objects and materials. Distance measurement sensors can determine the dimensions of objects such as height, width and diameter, using one or more sensors. The echo time response of ultrasonic sensor detector is based on time of travel after trigger pulse to the surrounding objects is non-linear and depends on the reflectance characteristics of the object surface. UltraSonic sensors are widely used for distance measurement purposes. They offer low cost and a precision of less than 1 cm in distance measurements of up to 6m [1, 4]. However, the most popular method used in these measurements is based on the time of flight (ToF) measurement. This ToF is the time elapsed between the emission and subsequent arrival after reflection of an Ultrasonic pulse train travelling at the speed of sound. This causes large response times for a single measurement.

**1.4.1 THEORY OF OPERATION**

This application is based upon the reflection of sound waves. Sound waves are defined as longitudinal pressure waves in the medium in which they are travelling. Subjects whose dimensions are larger than the wavelength of the impinging sound waves reflects them, the reflected waves are called the echo. If the speed of sound in the medium is known and the time taken for the sound waves to travel the distance from the source to the subject and back to the source is measured, the distance from the source to the subject can be computed accurately. This is the measurement principle of this application. Here the medium for the sound waves is air, and the sound waves used are ultrasonic, since it is inaudible to humans. Assuming that the speed of sound in air is 1100 feet/second at room temperature and that the measured time taken for the sound waves to travel the distance from the source to the subject and back to the source is t seconds, the distance d is computed by the formula d=1100 X 12 X t inches. Since the sound waves travel twice the distance between the source and the subject, the actual distance between the source and the subject will be d/2. A single I/O pin is used to trigger an ultrasonic burst (well above human hearing) and then "listen" for the echo return pulse. The sensor measures the time required for the echo return and returns this value to the microcontroller as a variable-width pulse via the same I/O pin. Ultrasonic sensors have definitely diversified functions including "detection" of what you cannot see, "measurement" of length, thickness and amount, and "destruction" of objects. Ultrasonic sensors are generally used for anti-collision and rangefinder purposes by measuring the distance to an obstacle [1] some application ideas where Ultra Sonic sensors can be used are security systems, parking assistant systems, interactive animated exhibits and robotic navigation.

**1.5** **LITERATURE REVIEW**

**1.5.1 FEASIBILITY STUDY**

The obstacle detector works on the principle of SONAR. An ultrasonic distance sensors and ultrasonic motion sensors. Ultrasonic distance sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object while, ultrasonic motion detector is a device that can detect movement of people or objects within a limited area.An ultrasonic transducer emits pulses at regular intervals, the interval being a function of the speed of travel. A receiver block listens to the echo. The echo round-trip time is a function of height. The receiver detects abrupt changes in terrain by comparing the round-trip time. This application measures range in terms of physical distance and controls the speed of the motors. As the transit time of the echo drops below each threshold the duty cycle of signal varies and thus speed is controlled. On the other hand, motion sensors sense motion by analysing sound waves in its environment. Some just listen for sounds while others send out ultrasonic signals and analyse how they are reflected back. This type of motion sensor is often used in home security system but can also be used in automobiles to detect the sudden appearance of any object on the path and hence based on the threshold of duty cycle again the speed is controlled.

**1.5.2 RESEARCH BACKGROUND**

1. “Ultrasonic anti crashing system for automobiles” IEEE paper published in 2013, attempted to develop an anti-crash warning system combined with ultrasonic ranging technology and sensor technology for automobiles. It mainly focusses on potholes in the road and its detection and hence automatic or manual reduction in the speed of the vehicle in order to avoid crashing.
2. In “Cooperative vehicle collision avoidance using inter-vehicle packet forwarding” IEEE paper publishes in 2005, proposes a broadcast oriented packet forwarding mechanism for intra-platoon cooperative collision avoidance (CCA) using dedicated short range communication (DSRC) based wireless networks. Using an implicit acknowledgement strategy it is shown that with inter-vehicle spacing of nearly one second, the proposed mechanism is capable of saving up to 90 percent of vehicles in a platoon from chain crashes following emergency events at the front of the platoon.
3. **ULTRASONIC DISTANCE SENSORS**

Ultrasonic distance sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object while, ultrasonic motion detector is a device that can detect movement of people or objects within a limited area.

1. **ULTRASONIC MOTION SENSORS**

This sensor senses motion by analysing sound waves in its environment. Some just listen for sounds while others send out ultrasonic signals and analyse how they are reflected back.

**CHAPTER 2**

**AIM AND SCOPE**

**2.1 AIM**

In this work, distance of the object is measured through ultrasonic distance sensor and the sensor output is connected to signal conditioning unit and after that it is processed through Arduino microcontroller. The measured results are displayed in liquid crystal display. The results are transferred to personal computer. The sensor is attached to servo motor to find the polar distance around the sensor upto 1800 rotations. This application is also used to find the obstacles detection and the exact distance can also be obtained. The measured distance is displayed on the LCD display

**2.2 SCOPE**

In the process of selecting the most suitable classification method for this scheme, several supervised algorithms were tested and compared in Arduino using c++, a powerful open-source such as Internet of things (IOT) is used for user access. Firebase Arduino was used to obtain more accurate performance results, measuring in each experiment:

1) Training time.

2) Classification time.

3) Accuracy on train samples.

4) Accuracy on test samples.

**CHAPTER 3**

**SYSTEM IMPLEMENTATION**

**3.1 SOFTWARE REQUIREMENTS**

**1) ARDUINO SOFTWARE**

The [**Arduino**](https://en.wikipedia.org/wiki/Arduino)**integrated development environment (**[**IDE**](https://en.wikipedia.org/wiki/Integrated_development_environment)**)** is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2. The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring.The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) 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](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/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**) PROCESSING SOFTWARE**

**Processing** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) graphical library and [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) built for the electronic arts, [new media art](https://en.wikipedia.org/wiki/New_media_art), and [visual design](https://en.wikipedia.org/wiki/Visual_design)communities with the purpose of teaching non-programmers the fundamentals of [computer programming](https://en.wikipedia.org/wiki/Computer_programming) in a visual context.

Processing uses the [Java language](https://en.wikipedia.org/wiki/Java_(programming_language)), with additional simplifications such as additional classes and aliased mathematical functions and operations. As well as this, it also has a graphical user interface for simplifying the compilation and execution stage.

The Processing language and IDE were the precursor to other projects including [Arduino](https://en.wikipedia.org/wiki/Arduino), [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) and p5.js

**3.2 HARDWARE REQUIREMENTS**

1. Arduino UNO board
2. USB cable connector for Arduino UNO
3. Ultra Sonic HC-SR04
4. Jumper wires male to female
5. Micro Server SG90

The Block diagram of Ultrasonic Distance Detection with Arduino.

In this work, distance of the object is measured through ultrasonic distance sensor and the sensor output is connected to signal conditioning unit and after that it is processed through Arduino microcontroller. The measured results are displayed in liquid crystal display. The results are transferred to personal computer. The sensor is attached to servo motor to find the polar distance around the sensor upto 1800 rotations. This application is also used to find the obstacles detection and the exact distance can also be obtained. The measured distance is displayed on the LCD display. The hardware components the system as explain below

The ultrasonic distance sensor module with signal conditioning

. Servo motor

. Arduino

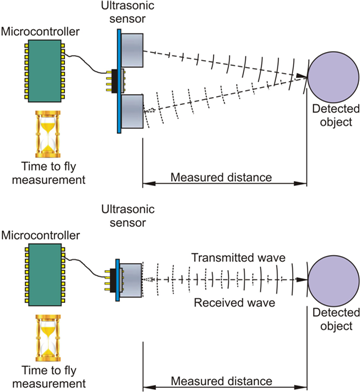
.ATmega 328 Microcontroller

.Personal computer

**3.3. MODULE**

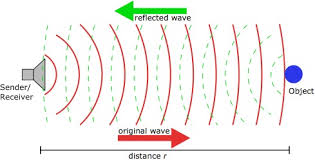
**3.3.1 METHOD 1:**

This device is used to measure the distance from an object. It can detect objects that are within a range of 2cm – 450cm (.78” – 14’ 9”). The device uses two digital pins to communicate the distance found. Ultrasonic Range Detection Sensor, works by sending an ultrasound pulse at around 40 KHz, It then waits and listens for the pulse to echo back, calculating the time taken in microseconds. We can trigger a pulse as fast as 20 times a second and it can determine objects up to 3 metres away and as near as 3cm. The snapshot of the sensor and working process of the sensor is shown in fig 2. The sensor needs a 5V power supply to run. The Timing diagram is shown in **(FIG 3.1).**10uS pulse is required to the trigger input and 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. Then calculate the range through the time interval between sending trigger signal and receiving echo signal. We suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal. There are only four connections, +5v, Gnd, trigger and Echo.



**FIG 3.1(OBJECT DETECTION)**

The Ultra Sonic sensor works as a burst signal is transmitted for short duration (is emitted) by the emitter. After that there will be a silent period. This period is actually called “response time” and is the time waiting for reflected waves(**FIG3.3).**The acoustic emitted signal may find an obstacle or not. If an obstacle is found, the acoustic signal will be bounced back from the obstacle. This backbounced signal is called “echo”. The echo is received by the receiving transducer and is converted into electrical signal. Usually this signal is amplified, filtered and can be converted into digital form . Using the elapsed time between transmission and reception, the distance between the Ultra Sonic system and obstacle/object can be calculated.

**FIG 3.2 SENSOR FIG 3.3 WAVEFORM**

The ultrasonic sensor is attached with the servo meter. The servo motor rotates rotate in clockwise and anticlockwise direction. The sensor **(FIG 3.2)** measure the distance around the sensor. The measured distance is calculated using Arduino controller within a predefined time interval. The analog output read from the sensor module is transferred to personal computer through serial port via Arduino.

**CHAPTER 4**

**RESULTS AND SCREENSHOTS**

**4.1 RESULTS**

1. To develop a safety system that includes ultrasonic sensors which detects obstacles like potholes and humps, and is accompanied by gradual decrease in speed of motors.
2. A prototype of the entire set up is developed that measures and monitors the road condition from a distance of about 3m from the obstacle.
3. To notify the driver if there is any sudden movement of obstacle under any threat which should have a result of decrease in the speed of the vehicle and hence avoid accidents or crashes.

**4.1.1 METHODOLOGY**

The use of ultrasound can be classified into two main groups namely,

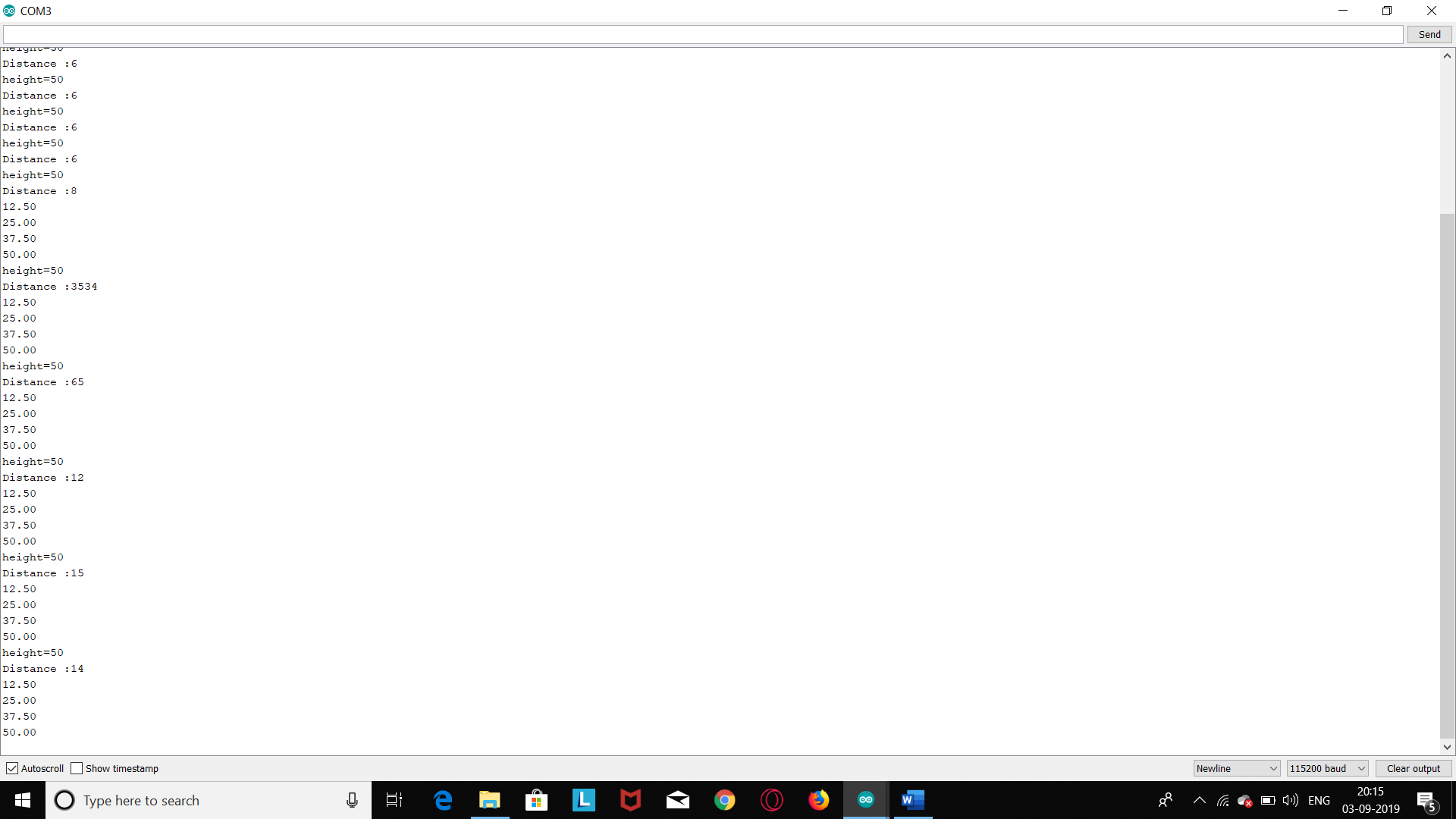
• Active ultrasound

• Passive ultrasound

Active ultrasound exhibits, physical or chemical effects when applied. Then reaches the highest values when generated. Speed of sound is dependent on the type of environment in which it moves, and the current temperature of the environment. The output of passive ultrasound output is generated only at lower values. In this paper to measure the spring heights during the car movement, ultrasonic sensor has been mounted onto the back of a car and equipped with four potentiometer sensors. To record the ultrasonic sensor and the potentiometer outputs a portable digital recorder is used. Sensor tests asphalt and rough ground and the four potentiometer outputs have been used to compute a distance reference value to be compared with the ultrasonic measured distance. By adding the tire deformations to the spring heights measured by the potentiometers, end spring heights, have been estimated which in turn used to identify the plane of the vehicle body. The distance reference value that corresponds to the distance the ultrasonic sensor should produce has been determined by putting the measuring head coordinates into the identified plane equation . For many applications ultrasonic plays a vital role not only for distance measurement but also around the world, indoors and outdoors in the harshest conditions, namely liquid level control, full detection, thread or wire break detection, robotic sensing, stacking height control and also for people detection for counting based upon proximity detection and range measurement.

**4.2 SCREENSHOTS**

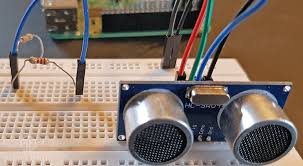
**4.2.1 SERIAL MONITOR**



**FIG 4.1 OUTPUT**

This window is called the Serial Monitor and it is part of the Arduino IDE software. Its job is to allow you to both send messages from your computer to an Arduino board (over USB) and also to receive messages from the Arduino.  
The message “Enter LED Number 0 to 9 or 'x' to clear” has been sent by the Arduino, and it is telling us what commands we can send to the Arduino which is either to send the 'x' (to turn all the LEDs off) or the number of the LED you want to turn on (where 0 is the bottom LED, 1 is the next one up right up to 7 for the top LED).  
  
Try typing the following commands, into the top area of the Serial Monitor that is level with the 'Send' button. Press 'Send', after typing each of these characters: x 0 3   
Typing x, will have no effect, if the LEDs are already all off, but as you enter each number, the corresponding LED should light and you will get a confirmation message from the Arduino board, so that the Serial Monitor will appear as shown above in **(FIG 4.1)**.

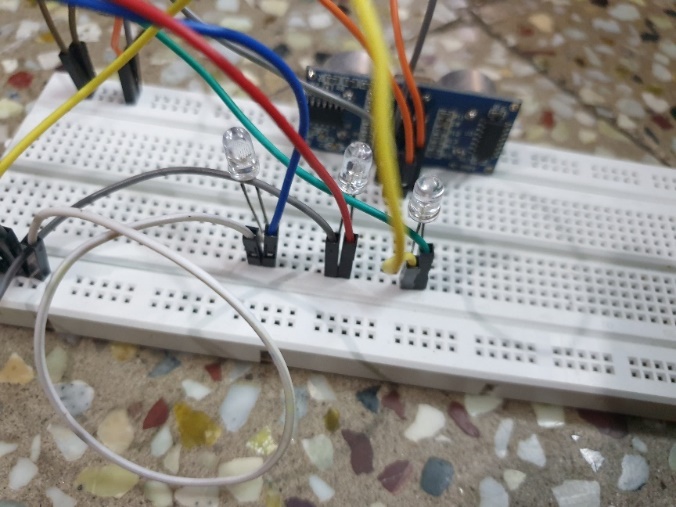
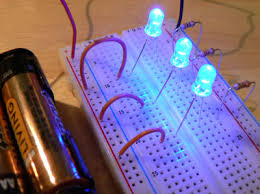
**4.3 SETUP IMAGES**



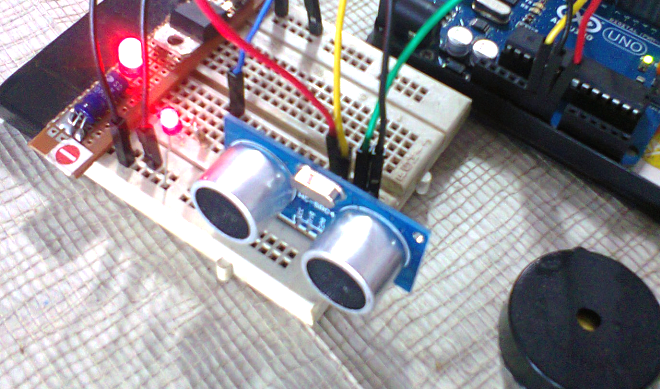
**FIG 4.2 SENSOR**

Sensors are used in everyday objects such as touch-sensitive elevator buttons ([tactile sensor](https://en.wikipedia.org/wiki/Tactile_sensor)) and lamps which dim or brighten by touching the base, besides innumerable applications of which most people are never aware. With advances in [micromachinery](https://en.wikipedia.org/wiki/Micromachinery) and easy-to-use [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) platforms, the uses of sensors have expanded beyond the traditional fields of temperature, pressure or flow measurement,[[1]](https://en.wikipedia.org/wiki/Sensor#cite_note-1) for example into [MARG sensors](https://en.wikipedia.org/wiki/Attitude_and_heading_reference_system) **(FIG 4.2)**. Moreover, analog sensors such as [potentiometers](https://en.wikipedia.org/wiki/Potentiometer) and [force-sensing resistors](https://en.wikipedia.org/wiki/Force-sensing_resistor) are still widely used. Applications include manufacturing and machinery, airplanes and aerospace, cars, medicine, robotics and many other aspects of our day-to-day life.

A sensor's sensitivity indicates how much the sensor's output changes when the input quantity being measured changes. For instance, if the mercury in a thermometer moves 1  cm when the temperature changes by 1 °C, the sensitivity is 1 cm/°C (it is basically the slope Dy/Dx assuming a linear characteristic). Some sensors can also affect what they measure; for instance, a room temperature thermometer inserted into a hot cup of liquid cools the liquid while the liquid heats the thermometer. Sensors are usually designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages.

 ****

**FIG 4.3 BREADBORD FIG 4.4 LED**



**FIG 4.5 INDICATION**

These Breadboard **(FIG4.3)** that are identical to the ones that run horizontally, except they are, typically[\*](https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all#power), all connected. When building a circuit, you tend to need power in lots of different places. The power rails give you lots of easy access to power wherever you need it in your circuit. Usually they will be labeled with a ‘+’ and a ‘-’ and have a red **(FIG 4.5)** and blue **(FIG 4.4)** or black stripe, to indicate the positive and negative side.

It is important to be aware that the power rails on either side are not connected, so if you want the same power source on both sides, you will need to connect the two sides with some jumper wires. Keep in mind that the markings are there just as a reference. There is no rule that says you have to plug power into the '+' rail and ground into the '-'rail, though it's good practice to keep everything in order.

**CHAPTER 5**

**CONCLUSION AND FUTURE SCOPE**

**5.1 CONCLUSION**

The objective of the project was to design and implement an ultrasonic distance meter. The device described here can detect the target and calculate the distance of the target. The ultrasonic distance meter is a low cost, low a simple device for distance measurement. The device calculates the distance with suitable accuracy and resolution. It is a handy system for non-contact measurement of distance. The device has its application in many fields. It can be used in car backing system, automation and robotics, detecting the depth of the snow, water level of the tank, production line. This device will also have its application in civil and mechanical field for precise and small measurements. For calculating the distance using this device, the target whose distance is to be measured should always be perpendicular to the plane of propagation of the ultrasonic waves. Hence the orientation of the target is a limitation of this system. The ultrasonic detection range also depends on the size and position of the target. The bigger is the target, stronger will be the reflected signal and more accurate will be the distance calculated. Hence the ultrasonic distance meter is an extremely useful device.

**5.2 FUTURE SCOPE**

1. Making the glove Wireless using X-Bee or Bluetooth.
2. Using an IMU to get directions easily and more accurately for all the motions of the hand.
3. Implementing more gestures using the current hardware.
4. Implementing plug and play, by making a driver for the hard.

**APPENDIX**

**SOURCE CODE**

#include <ESP8266WiFi.h>

#include <FirebaseArduino.h>

#define FIREBASE\_HOST "ultrasonic-4082d.firebaseio.com"

#define FIREBASE\_AUTH "i6PIeWKWLBmPuPT6VHftu8zoTIDeFNLdp83vKpgz"

#define WIFI\_SSID "sonish"

#define WIFI\_PASSWORD "123456789"

int trig=D1;

int Echo=D2;

int bulb[4]={D5,D6,D7,D8};

long duration;

int distance;

void setup() {

Serial.begin(115200);

pinMode(trig,OUTPUT);

pinMode(Echo,INPUT);

pinMode(D5,OUTPUT);

pinMode(D6,OUTPUT);

pinMode(D7,OUTPUT);

pinMode(D8,OUTPUT);

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

Serial.print("connecting");

while(WiFi.status()!=WL\_CONNECTED){

Serial.print(".");

delay(500);

}

Serial.println();

Serial.print("connected: ");

Serial.println(WiFi.localIP());

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);

Firebase.set("Height",50);

}

void firebasereconnect(){

Serial.println("Trying to reconnect");

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);

}

int i; int height=50;

int temp=-1;

void loop() {

if (Firebase.failed()) {

Serial.print("setting number failed:");

Serial.println(Firebase.error());

firebasereconnect();

return;

}

int height= Firebase.getInt("Height");

Serial.print("height=");

Serial.println(height);

float arr[4];

percentage(height,arr);

digitalWrite(trig, HIGH);

delayMicroseconds(10);

digitalWrite(trig, LOW);

duration=pulseIn(Echo, HIGH);

distance=duration\*0.034/2;

Serial.print("Distance :");

Serial.println(distance);

if(temp!=distance)

{

for(i=0;i<4;i++)

digitalWrite(bulb[i], LOW);

for(i=0;i<4;i++)

{

Serial.println(arr[i]);

if(distance>arr[i])

digitalWrite(bulb[i], HIGH);

temp=distance;

}

}

delay(1000);

}

void percentage(int a, float b[]){

float p[]={25.0,50.0,75.0,100.0};

for(int i=0;i<4;i++)

b[i]=(p[i]/100)\*a;

}

**REFERENCES**

<https://en.wikipedia.org/wiki/Arduino_Uno#Communication>

<https://www.elprocus.com/atmega328-arduino-uno-board-working-and-its-applications/>

[https://www.tutorialspoint.com/arduino/arduino\_communication.htm#](https://www.tutorialspoint.com/arduino/arduino_communication.htm)

<https://www.ijariit.com/manuscripts/v2i5/V2I5-1189.pdf>

[file:///C:/Users/91978/Downloads/1488-4099-1-PB.pdf](file:///C:\Users\91978\Downloads\1488-4099-1-PB.pdf)

<https://www.google.com/search?q=ULTRASOnic+sensor+WORKING&rlz=1C1CHBF_enIN857IN857&source=lnms&sa=X&ved=0ahUKEwinh_SRrrfkAhWXH7cAHc39AhQQ_AUIDCgA&biw=1536&bih=754&dpr=1.25>

<https://www.google.com/search?q=ultrasonic+with+breadboard+circuit&rlz=1C1CHBF_enIN857IN857&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi4ue6fubfkAhXKL48KHcUcBS8Q_AUIEigB&biw=1536&bih=754>