AUTOMATIC TRIPOD

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

BACHELOR OF TECHNOLOGY

in

Electronics and Communication Engineering

by

Harsh Pandey (190013125021)

Abhishek Gupta (190013125003)

Under the Guidance of

Dr. Manoj Kumar Jain

FACULTY OF ENGINEERING AND TECHNOLOGY
UNIVERSITY OF LUCKNOW, LUCKNOW.

2022 – 23

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person or material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher education, except where due acknowledgement has been made in the text.

Harsh Pandey
(190013125021)
Date: 22/05/2023

Abhishek Gupta
(190013125003)
Date: 22/05/2023

CERTIFICATE

Certified that **Harsh Pandey** (190013125021) and **Abhishek Gupta** (190013125003) has carried out the project work presented in this project report entitled **“Automatic Tripod”** for the award of **Bachelor of Technology** (Electronics and Communication Engineering) from **Faculty of Engineering and Technology, University of Lucknow, Lucknow** under my/our guidance. The project report embodies results of original work, and studies are carried out by the student themselves and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Dr. Manoj Kumar Jain
(Department of Electronics and Communication Engineering)

ABSTRACT

Target/object detection, recognition, position, movement speed, etc. is easy when the object is near or easily visible. But the same doesn't stand true especially when the object is far or not visible due to so many factors like weather conditions, day/night cycle, etc. Therefore, Radio Detection and Ranging (RADAR) were invented, which uses radio waves to determine the range, angle, or velocity of objects. But it uses long time to detect, has short detection range, not target specific because of wide range, oversensitive, costly, etc. A cheaper, easy and effective alternate solution is to use ultrasonic sensor which use sound waves for detection and ranging. Therefore, this paper provides a method in which the Ultrasonic Sensor (HC-SR04) acts as RADAR. The HC-RS04 is connected to Tripod the rotation/movement purpose. These components are connected to Arduino Uno. Usually, the range of ultrasonic wave is 20 kHz but here the HC-SR04 range is 3cm to 150m as it is smaller in terms of project usage. Advantages are: it is not affected by colour or transparency of objects, can be used in dark environments, not highly affected by dust, dirt, or high-moisture environments, etc. The results show the object detected with its range/distance and angle in a java-based GUI, different ranges of object in cm at which it is detected and the detection message sent to the admin.

ACKNOWLEDGEMENT

I would like to express my sincere thanks to Dr. Manoj Kumar Jain Sir for his valuable guidance and support in completing my project.

I would also like to express my gratitude towards our Head of Department Dr. Siddharth Singh Sir for giving me this great opportunity to do a project on Automatic Tripod. Without their support and suggestions this project would not have been completed.

I am ensuring that this project is done by me and not copied from anywhere.

Harsh Pandey

Abhishek Gupta

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

[x]	Integer value of x.
≠	Not Equal
∈	Belongs to
€	Euro- A Currency
_	Optical distance
–o	Optical thickness or optical half thickness

LIST OF ABBREVIATIONS

AAM	Active Appearance Model
ICA	Independent Component Analysis
ISC	Increment Sign Correlation
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristics

CHAPTER 1
INTRODUCTION

1.1 Introduction

A Tripod is a device which is used to hold the camera or mobile phones in shooting of photos and videos for content creation or any other purposes. It is a portable device and has three legs or stand. The legs of tripod are spread in a way so that it can make equal distance to each other. Tripods are light in weight and are sturdy. It helps to shoot steady photos, videos or time-lapsed videos. An Automatic Tripod is a device which helps in shooting photos and videos with ease and without need of any other person. Which means only a single person can use this device to shoot photos and videos. Basically, it uses motorized units for its movement. The motorized units automate its movement into horizontal as well as vertical in direction. It is equipped with transmitters and sensors that make it more useful. It is a battery-operated device. It also uses ultrasonic sensors as input for detecting/capturing the movement of body which emits the signal.

1.2 Need of an Automatic Tripod

An automatic tripod using an ultrasonic sensor can provide several benefits and functionalities. Here are a few reasons why such a device could be useful:

(A) Stability: By using an ultrasonic sensor, the tripod can automatically adjust its legs to ensure a stable and level platform. The sensor can detect uneven surfaces and adjust the length of the legs accordingly, providing a stable base for the camera or other equipment mounted on the tripod.

(B) Motion Detection: Ultrasonic sensors can detect motion in their surroundings. This capability can be utilized to automatically activate the tripod when a moving subject is detected within its range. This feature can be particularly useful for wildlife photography or capturing fast-moving subjects.

(C) Remote Control: With an ultrasonic sensor and appropriate control mechanisms, an automatic tripod can be remotely operated. This allows you to control the movement, height adjustment, and other functionalities of the tripod from a distance, providing convenience and flexibility during photography or videography.

(D) Precise Positioning: Automatic tripods allow for precise positioning of the camera or equipment. With motorized movements, they can pan, tilt, and rotate with high accuracy, enabling photographers or videographers to achieve the desired framing and composition without physically adjusting the tripod.

1.3 Current Option Available

Motorized Pan-Tilt Heads: Motorized pan-tilt heads can be attached to a tripod to provide automated movement along the horizontal (pan) and vertical (tilt) axes. These heads can be controlled remotely, allowing for precise positioning and automated tracking.

Wireless Remote Control: Wireless remote-control systems can be integrated into a tripod to enable remote operation. This allows users to control the tripod's movements, adjust the camera angle, and trigger the shutter from a distance.

Electric Motors and Actuators: Electric motors and actuators can be used to automate the extension and retraction of the tripod legs. This eliminates the need for manual adjustment and provides stability and levelling even on uneven surfaces.

Sensors and Object Tracking: Sensors such as motion sensors or depth sensors can be used to detect the presence of objects or subjects and track their movement. This information can be used to automatically adjust the tripod's position and keep the camera focused on the subject.

Integrated Camera Control: Some tripods come with built-in camera control capabilities, allowing users to control the camera settings and capture images or videos directly from the tripod itself. This integration can provide a seamless and automated shooting experience.

Smartphone Apps and Connectivity: Many modern tripods have smartphone apps that can be used to control the tripod remotely. These apps often offer features like automatic panorama capture, time-lapse photography, and remote triggering of the camera.

1.4 Idea behind this project

Automatic tripods incorporate intelligent tracking systems that use sensors or computer vision technology. These systems can detect and track moving subjects, ensuring that the camera remains focused on the target. This feature is especially useful for capturing action shots or video recordings where the subject is constantly in motion. If the subject is moving in left direction, then the device would move in left direction. And if the subject is moving in right direction, then the device would move in right direction.

Brief about our upcoming chapters

In our upcoming chapters we will discuss about the history, literature survey, general survey, hardware specifications, software specification, and final result.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

Ultrasonic sensors and Arduino are widely used components in the field of electronics and robotics, offering a range of applications and benefits. Ultrasonic sensors utilize sound waves to measure distances, detect objects, and perform various proximity sensing tasks. Arduino, on the other hand, is an open-source microcontroller platform that provides an accessible and versatile platform for creating interactive projects.

The combination of ultrasonic sensors with Arduino brings about numerous advantages and opens up possibilities for a wide range of projects. By integrating ultrasonic sensors with Arduino, users can leverage the features and capabilities of both components to create smart, interactive, and responsive systems.

One significant advantage of using ultrasonic sensors with Arduino is their ability to accurately measure distances. Ultrasonic sensors emit ultrasonic waves and measure the time it takes for the waves to bounce back after hitting an object. This information can be easily processed and interpreted by Arduino, enabling precise distance calculations. This feature is particularly useful in robotics, automated systems, and projects that require obstacle detection, proximity sensing, or distance measurements.

2.2 Previous Works

1. A wireless sensor enabled by wireless power–Yu-Hong Liu, Chii-Ruey Lin

Through harvesting energy by wireless charging and delivering data by wireless communication, this study proposes the concept of a wireless sensor enabled by wireless power (WPWS) and reports the fabrication of a prototype for functional 8 tests. One WPWS node consists of wireless power module and sensor module with different chip-type sensors. Its main feature is the dual antenna structure. Following RFID system architecture, a power harvesting antenna was designed to gather power from a standard reader working in the 915 MHz band. Referring to the Modbus protocol, the other wireless communication antenna was integrated on a node to send sensor data in parallel. The dual antenna structure integrates both the advantages of an RFID system and a wireless sensor.

Using a standard UHF RFID reader, WPWS can be enabled in a distributed area with a diameter up to 4 m. Working status is similar to that of a passive tag, except that a tag can only be queried statically, while the WPWS can send dynamic data from the sensors. The function is the same as a wireless sensor node. Different WPWSs equipped with temperature and humidity, optical and airflow velocity sensors are tested in this study. All sensors can send back detection data within 8 s. The accuracy is within 8% deviation compared with laboratory equipment. A wireless sensor network enabled by wireless power should be a totally wireless sensor network using WPWS. However, distributed WPWSs only can form a star topology, the simplest topology for constructing a sensor network. Because of shielding effects, it is difficult to apply other complex topologies. Despite this limitation, WPWS still can be used to extend sensor network applications in hazardous environments. Further research is needed to improve WPWS to realize a totally wireless sensor network.

2. Arduino Uno, Ultrasonic Sensor HC-SR04 Motion Detector with Display of Distance in the LCD – Kamweru Paul Kuria Department of Physical Sciences, Chuka University, Mutinda Mutava Gabriel Department of Physical Sciences, Chuka University

In this study, a tool to detect motion of objects, display the recorded distances on the LCD screen, produce a recorded sound alarm by Piezo Buzzer and also light by LEDs was assembled. The circuit was successfully connected and the program was sent to the Arduino microcontroller chip to run the circuit. The ultrasonic sensor was able to send the ultrasonic sound waves to the approaching object and the alarm sound from the piezo buzzer was produced. Piezo Buzzer was set to produce sound at different levels. The limit set for the Piezo Buzzer was to produce the sound alarm at the distance greater than 0cm and less than 150cm. The results were correct since for the distance equal or greater than 150cm and the distance equal to 0cm produced no tone from the piezo buzzer. LEDs were also set to produce light signal in a particular set of distances starting with Green LED followed by Blue LED and lastly Red LED. On the outcome, LEDs produced light as they were expected. The LCD was also expected to display the variation of distances as the object approaches the ultrasonic sensor. When the power was connected to the set-up, there were values recorded in the LCD screen indicating that the connection was right. The intensity of the screen was expected to be controlled by the potentiometer of which the same was approved.

3. Distance Sensing with Ultrasonic Sensor and ArduinoN.- Anju Latha1, B.Rama Murthy2, K. Bharat Kumar3 Department of Instrumentation, Sri Krishnadevaraya University, Anantapur, A.P., India

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.

4. Ultrasonic Servo Motor With Arduino Microcontroller Based On Electronics and Automation Engineering – Daniel Asemota

Arduino micro-controller has already made a huge impact on learning. The widespread acceptance gave the open-source hardware a new life, potentially challenging many industrial products and new interests in hardware prototyping and electronics. Recent transition has been made from 8 bit to 32 bits and it is expected that in future Arduino may be seen in form of a cheap practical computer. You can also presently make custom cell phones using open-source boards like the Arduino. Looking at this actionable advancement towards Arduino, allured the ideology to make systematic and organized parking system for vehicles, using Arduino Uno. This project's main purpose is to produce a real-life solution to the car parking problem which the whole world is facing frequently. People usually roam around in the parking lots trying to find a suitable place to park in. To solve that problem, we have created the automatic car parking system, using an open-source hardware, programmable sensors and the use of computers to provide an interface to understand the digital output produced.

5. DESIGNING OF LOW-COST ARDUINO BASED ULTRASONIC SONAR SYSTEM KHUDAIJA-TUL-MURTAZA1, HAMZA IMTIAZ1

In the present research project, an Ultrasonic sensor-based SONAR system was prepared. It provides a cheaper, low cost, low power solution for security purposes. The solution was designed and developed in University of Management and Technology (UMT) Labs. The path and distance of the object 64 was accurately calculated and were shown on the screen. The range of the device was small due to unavailability of long-range sensors in local Hall Road market. Further research work is required to differentiate the objects and to increase the range of the device.

Options available in market:

There are several automatic tripods available in the market. All of them uses different technologies and having much higher pricing than our project. Some of the different Automated tripods are listed below-

1. MeFOTO RoadTrip Air: This is a lightweight and compact tripod that features an automatic leg deployment system. It allows you to quickly set up and adjust the tripod height using a twist-lock mechanism.

2. Manfrotto Befree Live: This is a versatile tripod designed for video shooting. It includes a fluid video head with a pan and tilt mechanism, making it suitable for smooth camera movements. The tripod legs can be easily adjusted using a release button.

3. Slik Pro 700 DX: The Slik Pro 700 DX tripod comes with an automatic leg release system, allowing you to set up and fold down the tripod quickly. It provides sturdy support and is suitable for various types of photography.

4. Vanguard Alta Pro 2+ 263AB: This tripod features a unique Multi-Angle Central Column (MACC) system that allows you to adjust the central column to various

angles, providing greater flexibility in positioning your camera. It also has an automatic leg locking mechanism for quick setup.

5. Gitzo Systematic Series: Gitzo offers a range of professional-grade tripods with an automatic leg deployment system. These tripods are known for their exceptional stability and durability, making them suitable for heavy cameras and professional photographers

CHAPTER 3
HARDWARE IMPLEMENTATION

3.1 Introduction

In this project, ultrasonic sensors and servo motor are the major components. These are connected with and controlled by an Arduino UNO. Ultrasonic sensors transmit ultrasonic sound and the servo motor rotates the ultrasonic sensor. The Arduino UNO consists of USB connector, microcontroller, analogue input pins, power port, digital pins, crystal, oscillator reset switch, USB interface chip, TX-RX LEDs. It functions as a controller and provides a coding environment so it requires direct connection to a computer. HC-SR04 ultrasonic sensor is used in this study as it transmits the sound wave of high frequency near about 40 kHz to 70 kHz. Humans cannot hear it. If the wave gets reflected by an obstacle, then the reflected sound will be picked up by the receiver. Sonar is used to identify the object. The four pins of the sensor are: VCC=5VDC, trig pin=trigger pin/input pin, echo pin=output pin and GND or ground pin. A 10us trigger pulse is generated with the help of trig pin and to produce sound wave of 8 cycles.

Normally echo pin stays in a low state (0V) but when it receives the sound wave it goes to a high state of 5V. The total high state (5V) of echo pulse is counted as the travelling time of the sound wave. Servo motor (SG90) is a rotatory motor which runs with the help of servo mechanism. Servo motor is used here to detect the angle of the moving object. 0 to 180 degree is the rotatory range of servo motor, but depending on the manufacturing it can go up to 360 degrees. Servo motor receives the control signal and tunes the shaft. Servo motor is used to set up the position and velocity of the ultrasonic sensor. Three wires from the servo motor have to be connected manually. For the implementation of the components, the trig pin and the echo pin of the ultrasonic sensor (HC-SR04) are connected to the PWM pins and the SPI pins of Arduino UNO.

Servo and ultrasonic sensor are connected in such a manner where ultrasonic sensor can be mounted on the servo motor. The Arduino UNO power is supplied by the computer. The code that has been uploaded onto the Arduino UNO software will run through the data generate results.

3.2 Block Diagram

The working principle of the Automatic Tripod has been shown in the following block diagram.

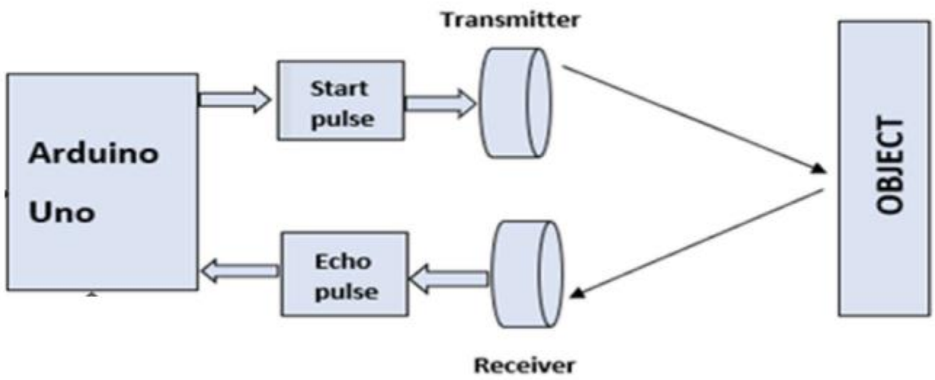


Fig.3.2 Block diagram of Automatic Tripod.

As shown in Fig.3.2, Arduino UNO receives power supply from the computer. Then microcontroller sends signals to the other two components: the ultrasonic sensor and servo motor. The feedback from these components is sent back to the microcontroller. The ultrasonic sensor has a transmitter to send out the ultrasound and a receiver to catch the reflected sound waves. The total travelling time of this sound wave is the distance.

3.3 Circuit Diagram

The automatic tripod circuit consists of an ultrasonic sensor, a servo motor, and an Arduino Uno R3 microcontroller. The ultrasonic sensor is used to detect the distance of an object in front of the tripod. The servo motor is connected to the camera mount of the tripod, allowing it to rotate horizontally. The Arduino Uno R3 acts as the brain of the system, receiving data from the ultrasonic sensor and controlling the servo motor.

When the circuit is powered on, the Arduino initializes the ultrasonic sensor and servo motor. The ultrasonic sensor continuously measures the distance to an object in front of the tripod. If the distance falls below a predefined threshold, indicating that an object is within range, the Arduino sends a signal to the servo motor to rotate the camera mount and point towards the detected object.

As the object moves, the ultrasonic sensor provides updated distance measurements, and the Arduino adjusts the position of the servo motor accordingly. This allows the tripod to automatically track the object and keep it within the camera's field of view, providing smooth and stable footage. The circuit offers a convenient solution for capturing dynamic shots without the need for manual adjustment of the tripod position.

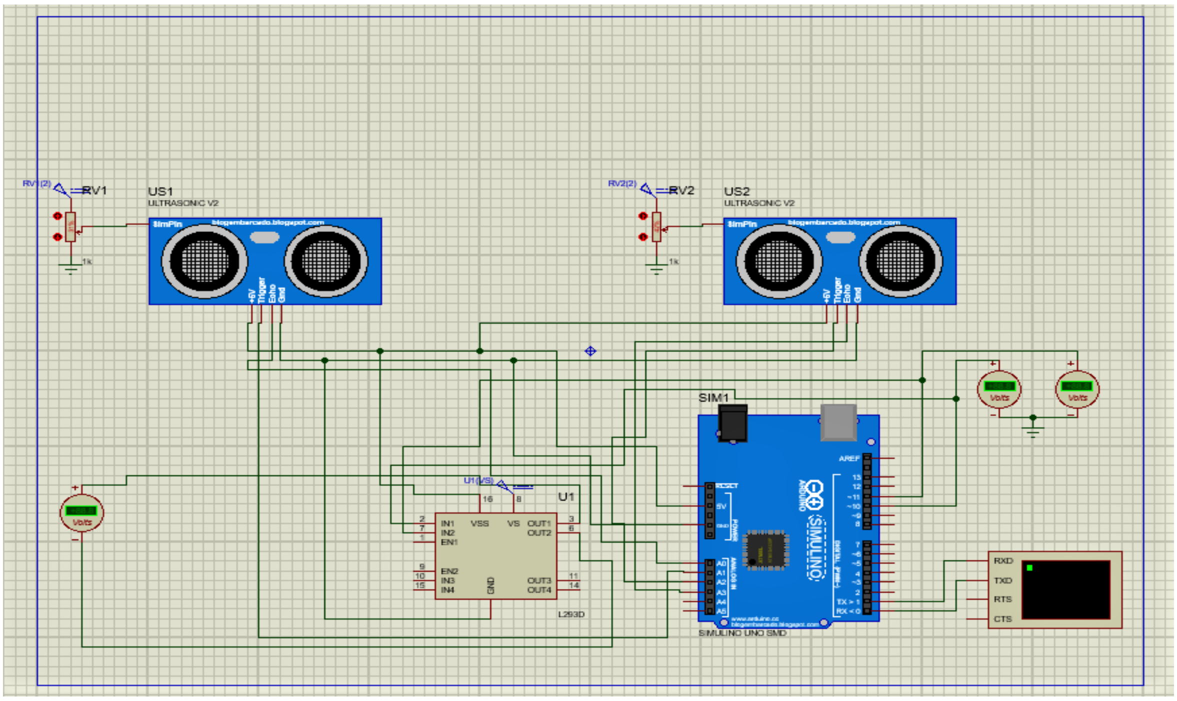


Fig.3.3 Circuit diagram of Automatic Tripod

3.4 List of Components

- Arduino Uno
- Ultrasonic
- DC Motor
- Spur Gear
- Wheel
- Motor Driver
- MDF Board
- Tripod

- Mobile Holder
- Wires
- Battery

Arduino Uno

Arduino ATMEGA-328 microcontroller has been programmed for various applications. By using the power jack cable, Arduino microcontroller has been programmed so that the execution of the program may takes place. Various kinds of Arduino board are present in the market. In this paper, Arduino UNO ATMEGA-328 microcontroller is described in a detailed manner.

Arduino software is installed in the computer and so that we can edit and upload the program according to the applications. Mainly these Arduino software supports C and C++ programming languages. Various inputs and outputs are present in the Arduino board and therefore simultaneously 8 input and output ports can be used for various applications. Some of the applications used by using Arduino boards are rotating general motor, stepper motor, control valve open, etc.

Nearly 7, 00,000 numbers of Arduinos are present in the market. Of these, Arduino ATMEGA-328 microcontroller consist of 14 input and output analog and digital pins (from these 6 pins are considered to be a PWM pins), 6 analog input pins and remaining are digital inputs. Power jack cable is used to connect Arduino board with the computer. External battery is connected with the Arduino microcontroller for the power supply. Arduino is an Open-Source microcontroller from which there is no feedback present in the microcontroller.

This Arduino board consist of I2C bus, that can be able to transfer the data from Arduino board to the output devices. These Arduino boards are programmed over RS232 serial interface connections with Atmega Arduino microcontrollers. The operating volt ranges from 5v. The input voltage recommended for Arduino microcontroller is from 7v and the maximum of 12v. The DC input current given to the Arduino board is in the range of 40mA.

It consists of different types of memories such as flash memory, EEPROM, SRAM. The length of the Arduino board is nearly about 68.64mm and the width of the microcontroller is about 53.4mm. The weight of the Arduino microcontroller is about 20g. We can use various

types of microcontrollers such as 8-bit AVL Atmel microcontroller and 32-bit Atmel arm microprocessor. From these different kinds of processors, we can use those processors for various engineering projects as well as industrial applications. Some of the examples of using the Arduino in the industrial applications are controlling the actuators and sensors. Some of the examples of Arduino microcontrollers are Arduino Duemilanove, Arduino UNO, Arduino Leonardo, Arduino Mega, Arduino MEGA 2560 R3, Arduino MEGA 2560 R3, Arduino Nano, Arduino Due, LilyPad Arduino, micro-Arduino. We have already mentioned, Arduino has been programmed by using C and C++ programming language. These C and C++ are high level languages. Normally it has 18 number of input and output pins.

Among those 6 pins are considered to be an analog input. From these analog inputs, we can be able to work the Arduino microcontroller using analog inputs supply. Normally analog inputs can be in the range of 0-5V. Similar to that digital inputs are present in the microcontroller which can act the use of microcontroller using digital inputs. Digital inputs can be in the range of 5V.

ATMEGA 328 microcontroller, which acts as a processor for the Arduino board. Nearly it consists of 28 pins. From these 28 pins, the inputs can be controlled by transmitting and receiving the inputs to the external device. It also consists of pulse width modulation (PWM). These PWM are used to transmit the entire signal in a pulse modulation. Input power supply such as Vcc and Gnd are used. These IC mainly consists of analog and digital inputs. These analog and digital inputs are used for the process of certain applications.

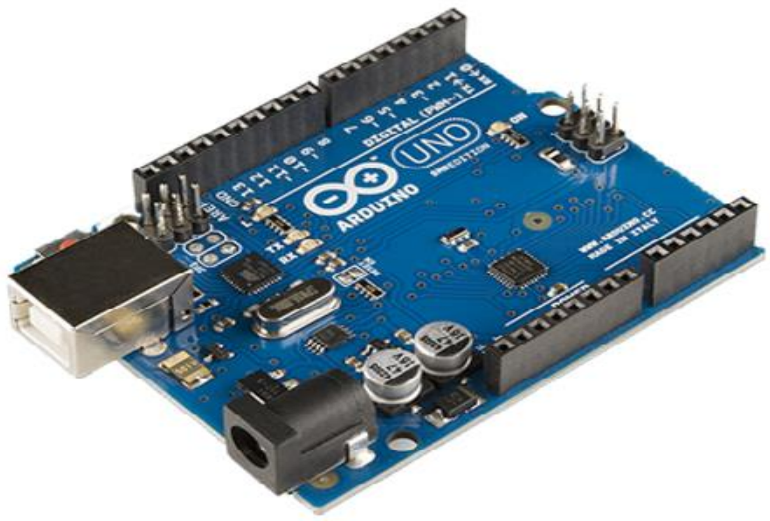


Figure 1.4(a) Arduino Atmega-328 Microcontroller

Arduino atmega-328 microcontroller board consist of 6 analog inputs pins. These analog inputs can be named from A0 to A5. From these 6 analog inputs pins, we can do the process by using analog inputs. Analog inputs can be used in the operating range of 0 to 5V.

Analog signal is considered as the continuous time signal, from which this analog signal can be used for certain applications. These are also called as non-discrete time signal.

Inputs such as voltage, current etc.., are considered to be either analog signal or digital signal only by analysing thentime signal properties. Various applications of Arduino microcontroller can use only an analog input instead of digital inputs. For these applications, analog input ports or pins can be used.

DIGITAL INPUT:

Digital inputs can be defined as the non-continuous time signal with discrete input pulses. It can be represented as 0’s and 1’s. These digital inputs can be either on state or in off state. Arduino atmega328 microcontroller also consists of 12 digital input pins. It can be stated as D0 to

D11. Nearly 12 inputs can be used for digital input/output applications. The working of the digital input ports is where the discrete input pulses can be triggered and supplied to the ports.

These ports receive the input and therefore the port can be used for both input and output process. These digital pins can access only the digital inputs.

ATMEGA-328 IC:

This ATMEGA-328 integrated chip consists of 28 pins. It consists of 6 analog inputs that are shown in the pin diagram. Analog inputs can be represented as PC0 to PC5. These analog input pins possess the continuous time signal which acts as an analog input for the system. Further it also consists of 12 digital inputs. It can be represented as PD1 to PD11 which act as a digital input port based on pulse width modulation (PWM). These PWM, which transmits the signal in the form of discredited form. Both analog and digital input ports can be used for various applications for the input power supply, VCC and GND pins are used. Pins PB6 and PB7, which acts as a crystal to generate a clock signal. By using this crystal, we can generate the clock signals and by these clock signals, we can use this clock signals for input sources. PC6 pin are the one where it can be used for the reset option. Resetting the program can be done by using this PC6 pin. The pin diagram of atmega-328 microcontroller can be shown below.

POWER JACK CABLE / USB PORT:

This Arduino atmega-328 microcontroller can be interfaced with the other electronic devices such as computer by using USB port or power jack cable from these power jack cable, we can upload the program of Arduino for their applications. At first, the program can be initialised or can be edited by using Arduino software tools. Then these programs can be transferred through Arduino microcontroller board by using USB cable or power jack cable.

POWER SUPPLY:

There is an additional power supply source present in Arduino microcontroller. Power supply port is present at the corner of the Arduino microcontroller. Either we can use this power supply port by connecting with external power supply (ie, ac power supply), or by connecting an dc power supply through input pins. These power supplies produce an active form to the Arduino microcontroller. These Arduino microcontrollers can accept a range of power supply. When the power supply voltage range exceeds, the microcontroller gets damaged. Hence, only the particular range of power supply should be given to the Arduino microcontroller.

WORKING PRINCIPLE:

The working of Arduino microcontroller is where the proper connection is made. Checking all the input ports as well as the power supply connection. The output of the pins can be connected with the external devices according to their applications. The program to be executed for the applications can be done by using Arduino software. From this Arduino software, we can edit according to the applications. This software can work on c and c++ programming language. It is fully a high-level language. By using the conditions of working, we can create a program to proceed for the applications. Then after, these programs can be uploaded through the Arduino microcontroller by using the power jack cable. The program can be uploaded to the microcontroller and ready for further process. ATMEGA-328 microcontroller can save a program and these IC can acts as a processor to do the process without any error. After by giving an analog or digital input to the system, we can do the process according to the applications. We can control the process of the application by editing the program in the Arduino software and again can be uploaded to the Arduino microcontroller via power jack cable. There is an option of reset button. The purpose of reset button is to reset the program which means the previous programs are deleted and we can use the Arduino for the other application purposes. Likewise, these Arduino ATMEGA-328 microcontrollers can be used for n number of applications. These Arduino microcontrollers are widely used in automation industries for controlling the process and to

work the system in an automation mode. Here, I have provided a simple Arduino program to do the process of rotating a stepper motor for one revolution. There are many numbers of example programs that are present in the Arduino software. We can edit these programs for our applications purposes.

ULTRASONIC SENSOR

Introduction to Ultrasonic Sensor:

Ultrasonic sensors work by emitting sound waves at a frequency which is too high for humans to hear.

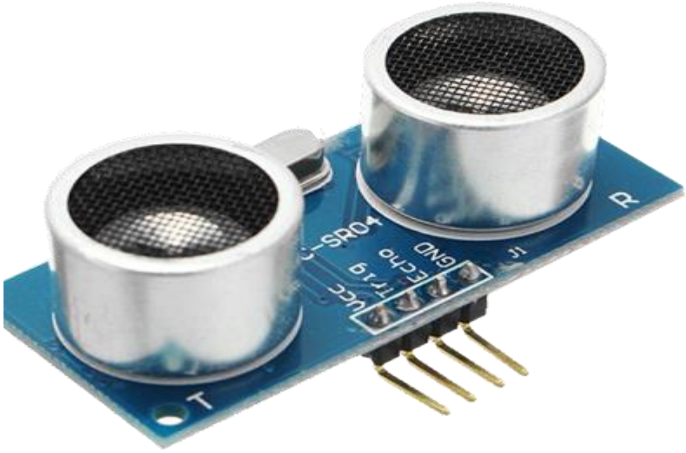


Fig.3.2 (b) Ultrasonic Sensor

An above image shows the HC-SR-04 Ultrasonic sensor which has a transmitter and a receiver. The pin configuration of Ultrasonic sensor is,

- **VCC** – +5 V supply
- **TRIGGER** – Trigger input of sensor. Microcontroller applies 10 us trigger pulse to the HC-SR04 ultrasonic module.
- **ECHO**–Echo output of sensor. Microcontroller reads/monitors this pin to detect the obstacle or to find the distance.
- **GND** – Ground

Sound is a mechanical wave traveling through the mediums, which may be a solid, or liquid or gas. Sound waves can travel through the mediums with specific velocity depends on the medium of propagation. The sound waves which are having high frequency reflect from boundaries and produce distinctive echo patterns.

Features of an Ultrasonic Sensor

1. Supply voltage: 5V (DC).
2. Supply current: 15mA.
3. Modulation frequency: 40Hz.
4. Output: 0 – 5V (Output high when obstacle detected in range).

5. Beam Angle: Max 15 degrees.
6. Distance: 2 cm – 400 cm.
7. Accuracy: 0.3cm.
8. Communication: Positive TTL pulse.

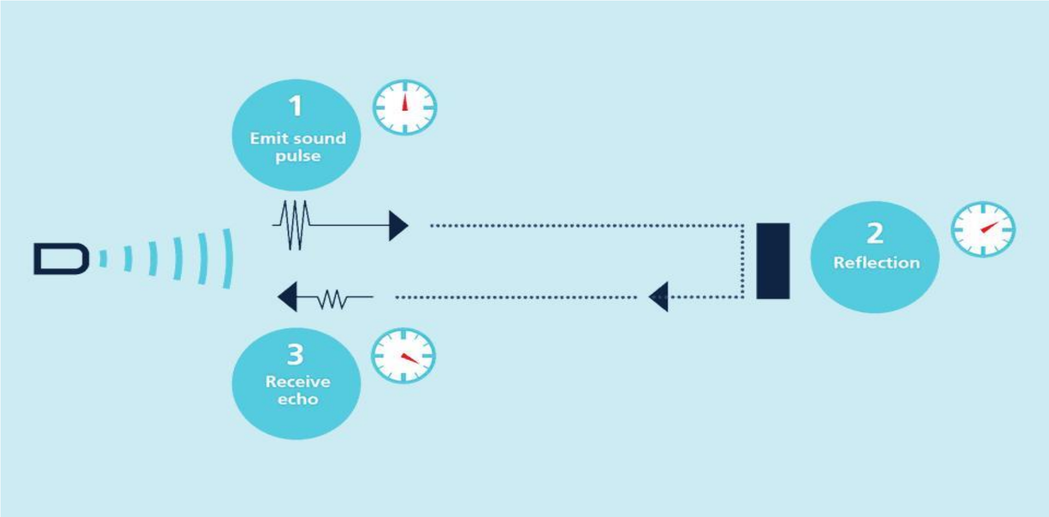


Fig.3.4(c) Working of Ultrasonic sensor

Ultrasonic Sensor Working Principle:

In industrial applications, an ultrasonic detection used to detect hidden tracks, discontinuities in metals, composites, plastics, ceramics, and for water level detection. For this purpose, the laws of physics which are indicating the propagation of sound waves through solid materials have been used since ultrasonic sensors using sound instead of light for detection. In this blog, we are going to learn about the ultrasonic sensor working principle and its applications. Ultrasonic sensors emit short, high-frequency sound pulses at regular intervals. These propagate in the air at the velocity of sound. If they strike an object, then they reflected back as an echo signal to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo.

Fig.3.4 (d) Ultrasonic sensor striking with obstacle

Ultrasonic sensors are excellent at suppressing background interference. Virtually all materials which reflect sound can be detected, regardless of their colour. Even transparent materials or thin foils represent no problem for an ultrasonic sensor.

Micro sonic ultrasonic sensors are suitable for target distances from 20 mm to 10 m and as they measure the time of flight, they can ascertain a measurement with pinpoint accuracy. Some of our sensors can even resolve the signal to an accuracy of 0.025 mm. Ultrasonic sensors can see through dust-laden air and ink mists. Even thin deposits on the sensor membrane do not impair its function.

Timing Diagram of Ultrasonic Sensor:

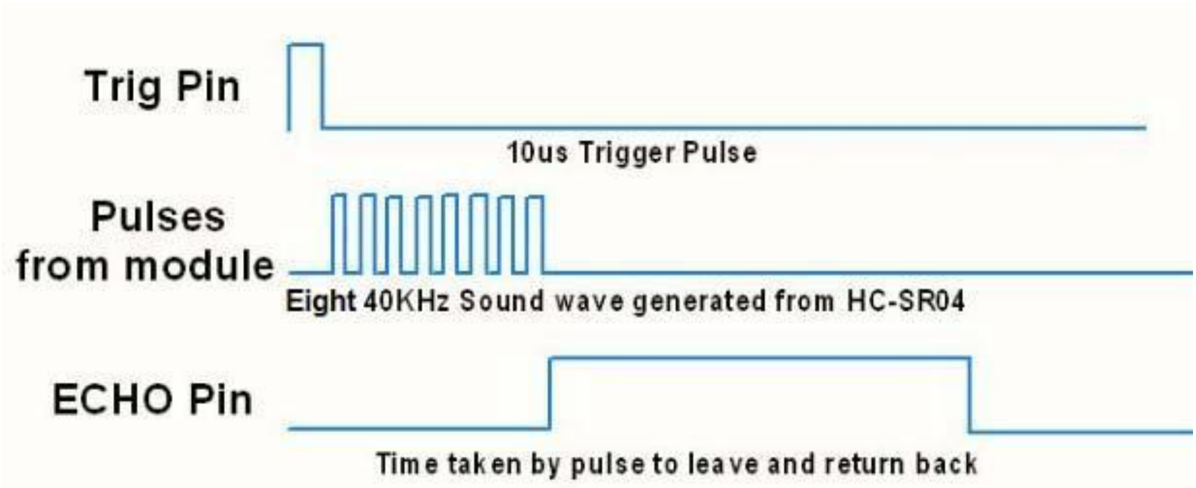


Fig.3.4 (e) Timing diagram of Ultrasonic sensor

1. First need to transmit trigger pulse of at least 10 us to the HC**-SR04** Trig Pin.
2. Then the HC-SR04 automatically sends Eight 40 kHz sound wave and wait for rising edge output at Echo pin.

3. When the rising edge capture occurs at Echo pin, start the Timer and wait for falling edge on Echo pin.
4. As soon as the falling edge captures at the Echo pin, read the count of the Timer. This time count is the time required by the sensor to detect an object and return back from an object.

How to calculate Distance?

If you need to measure the specific distance from your sensor, this can be calculated based on this formula:

We know that, **Distance= Speed* Time.** The speed of sound waves is 343 m/s. So,

Total Distance= (343 * Time of hight(Echo) pulse)/2

Total distance is divided by 2 because signal travels from HC-SR04 to object and returns to the module HC-SR-04.

Applications of an Ultrasonic Sensor:

- It Uses to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- It Used to measure the distance within a wide range of 2cm to 400cm.
- Used to map the objects surrounding the sensor by rotating it.
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water.

MOTOR

Motors play a critical role in various industries and applications by converting electrical energy into mechanical energy, enabling the movement of machines, vehicles, and devices. They offer different features, power outputs, and control mechanisms to meet the specific requirements of different applications.

A worm gear motor is a type of gear motor that utilizes a worm gear mechanism for power transmission. It consists of a worm (a type of gear with a screw-like thread) and a gear wheel, often referred to as a worm wheel or worm gear.

Fig.3.4 (f) Worm gear motor

Here are some key characteristics of worm gear motors:

1. Gear Ratio: Worm gear motors are known for their high gear reduction ratios. The worm screw's thread engages with the teeth of the worm wheel, resulting in a significant reduction in speed and an increase in torque output. This makes worm gear motors suitable for applications requiring high torque and low speed, such as conveyor systems, winches, and lifts.

2. Self-Locking: One unique characteristic of worm gear motors is their self-locking capability. Due to the friction between the worm and worm wheel, the worm gear motor can hold the load in place without the need for additional braking mechanisms. This self-locking feature makes them ideal for applications where the motor needs to maintain position and prevent backdriving, such as in hoists or positioning systems.

3. Efficiency and Noise: While worm gear motors offer high torque, they can have lower efficiency compared to other types of gear motors. The sliding action between the worm and worm wheel can result in energy loss and generate heat. Additionally, the meshing of the gears can produce noise and vibration. Therefore, it is important to consider the efficiency and noise requirements when selecting a worm gear motor for a specific application.

4. Compact Size: Worm gear motors are typically compact in size, making them suitable for applications where space is limited. The worm gear arrangement allows for a compact design by achieving high gear reduction in a single stage.

5. Lubrication: Proper lubrication is essential for the smooth operation and longevity of worm gear motors. Lubricants help reduce friction and wear between the gears, enhancing their efficiency and lifespan. Regular maintenance, including lubrication, is necessary to ensure optimal performance.

6. Temperature Considerations: The sliding action and higher gear reduction ratio in worm gear motors can generate more heat compared to other types of gear motors. This heat generation can affect the motor's performance and durability. It is important to consider the temperature limits and choose a motor that can operate within the required temperature range for the intended application.

3.5 How to select electronic components

Overview:

Surprisingly, many engineers do not pay enough attention to the task of selecting the right electronic component during the initial phase of their embedded product design.

Selecting the right electronic component is one of the most critical & challenging aspects of product design. In this article, I do not aim to explain how to select different components like resistor, capacitor, transistor, digital IC, microcontrollers, relay, connector, display, etc. as there is no *one* answer to this challenge. The correct answer is, you should know clearly what is your requirements w.r.t a particular component.

There are so many ways to solve the same problem, it will depend on various things like, if you need a compact design, easy to manufacture design, most affordable, most power efficient, least number of components, most reliable, etc. So, at the end it’s a trade-off, where, you are trying to decide which component matches most of your requirements. Every component is different and needs specific attention, but general rules of the game remain the same.

3.6 Which all parameters to consider?

To start with, I recommend making a list of components and their critical parameters which needs to be checked. There are a number of parameters which I always consider while choosing the right component. Hope this will help you next time when you are working on a project where you need new components to be selected. But before that one important point about parts from previous designs. Electronic component from a previous proven design

The easiest bet would be to use the part you have already used in your previous proven design, this helps reduce the risk of design issues, delays, also no additional part in inventory to manage.

But this approach has one drawback. If you stick on to an older proven component, you may miss the advantage of a new part, which may bring in benefits like more compact, more integration, more power efficient, better protection, better longevity. When starting a new design, it also brings an opportunity to try newer parts available (maybe you can have a fall-back option on board if you have doubts about the new part’s performance).

3.7 Conclusion

The combination of ultrasonic sensors and Arduino offers a cost-effective, versatile, and user-friendly solution for various projects involving distance measurement and object detection. Their compatibility, affordability, and availability make them a popular choice among DIY enthusiasts and project developers. However, it is crucial to assess specific project requirements and consider the limitations of both ultrasonic sensors and Arduino to ensure they align with the desired functionality and scope of the project. Worm gear motors are widely used in applications requiring high torque and low speed, along with self-locking capabilities. Their compact size, self-locking feature, and high gear reduction ratios make them suitable for a variety of applications, including conveyors, lifts, positioning systems, and more. However, it is important to consider factors such as efficiency, noise, lubrication, and temperature limits when selecting a worm gear motor for a specific application

CHAPTER 4
SOFTWARE REQUIREMENT

4.1 Introduction

In this project we have used Arduino IDE, short for Integrated Development Environment. Arduino IDE serves as the primary tool for creating and manipulating code that controls various electronic components connected to an Arduino board. It supports the Arduino programming language, a simplified variant of C and C++. One of the key advantages of Arduino IDE is its cross-platform compatibility, available for Windows, macOS, and Linux operating systems. This enables users to work seamlessly on different machines while maintaining a consistent programming environment.

4.2 Algorithm and flow chart

(A) Algorithm

1. Declare and assign the pin numbers for the ultrasonic sensors, motor control pins, and variables to store distance and duration measurements.

2. In the setup() function:

Initialize serial communication.

Set the pin modes for motor control and ultrasonic sensor pins.

3. In the loop() function:

Call the LDist() function to measure the left distance.

Call the RDist() function to measure the right distance.

Print the left and right distances to the serial monitor.

Compare the distances and perform the following actions:

If the left distance is greater than the right distance, call the TurnLeft() function.

If the right distance is greater than the left distance, call the TurnRight() function.

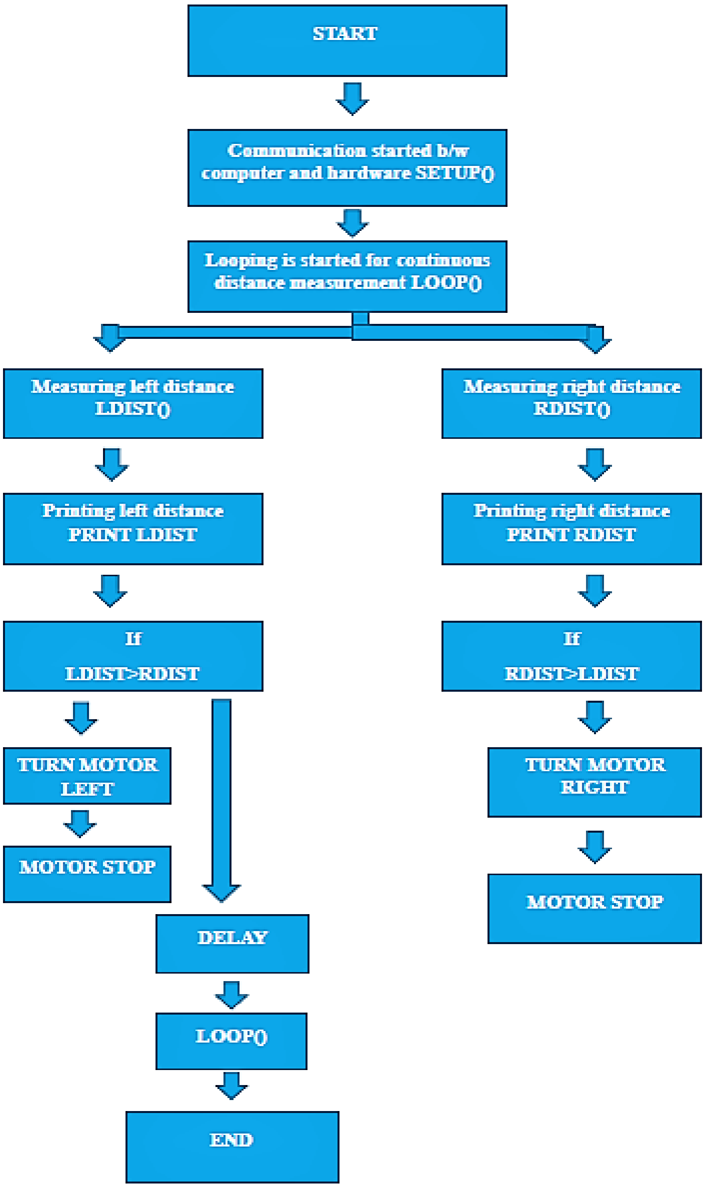
If both distances are equal, call the Stop() function and delay for 500 milliseconds.

If none of the conditions are met, call the Stop() function.

4. Implement the TurnLeft(), TurnRight(), and Stop() functions to control the motor movements.

5. Implement the RDist() and LDist() functions to calculate the distances using the ultrasonic sensors.

(B) Flow chart



4.3 Arduino Software (IDE)

Arduino IDE is a software development environment specifically designed for programming Arduino boards. It contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It provides a user-friendly interface that allows users to write, compile, and upload code to Arduino microcontrollers. The IDE supports the Arduino programming language, a simplified version of C and C++, making it accessible to beginners and experienced programmers alike. The Arduino IDE is cross-platform compatible to Windows, macOS, and Linux operating systems. This flexibility allows users to work on different machines without any compatibility issues. The IDE features a text editor with syntax highlighting, making it easier to write and debug code. One of the notable features of Arduino IDE is its extensive library collection. Libraries are pre-written code snippets that simplify complex tasks, such as controlling sensors, motors, and displays. These libraries can be easily imported into the IDE, saving time and effort in coding.

The compilation and uploading process in Arduino IDE is seamless. Once the code is written, it is compiled into machine-readable instructions. The compiled code is then uploaded to the Arduino board via a USB connection, enabling the microcontroller to execute the programmed tasks in real-time. Arduino IDE also has a thriving community of users who actively share projects, code examples, and provide support to fellow Arduino enthusiasts. This collaborative environment encourages learning and creativity, as users can explore various projects and gain inspiration from others.

Overall, Arduino IDE provides a powerful and intuitive platform for developing projects with Arduino boards. Its simplicity, versatility, and strong community support make it an excellent choice for hobbyists, students, and professionals interested in electronics and physical computing.

Fig.4.3(a) Arduino IDE installation interface.

(A) Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension '.ino'. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors.

The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

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

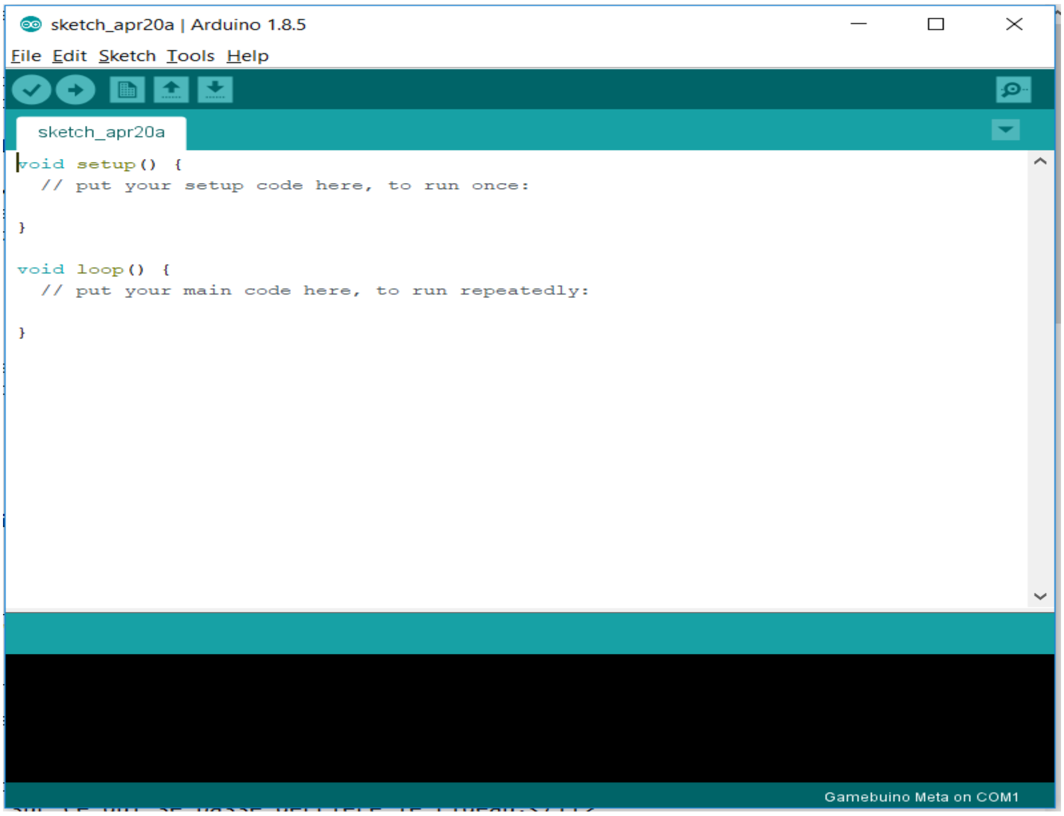


Fig.4.3(b) Arduino IDE home interface (where we write sketches).

(B) Tools used in Arduino Software (IDE)

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

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

Fix Encoding & Reload: Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

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

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

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

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

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

Types of boards of Arduino

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

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

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

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

Arduino Diecimila or Duemilanove w/ ATmega168: An ATmega168 running at 16 MHz with auto-reset.

Arduino Nano w/ ATmega328P: An ATmega328P running at 16 MHz with auto-reset. Has eight analog inputs.

Arduino/Genuino Mega 2560: An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

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

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

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

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

Arduino Esplora: An ATmega32u4 running at 16 MHz with auto-reset.

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

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

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

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

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

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

Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328P: An ATmega328P running at 16 MHz with auto-reset. Equivalent to Arduino Duemilanove or Nano w/ ATmega328P; 6 Analog In, 14 Digital I/O and 6 PWM.

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

Arduino Robot Control: An ATmega328P running at 16 MHz with auto-reset.

Arduino Robot Motor: An ATmega328P running at 16 MHz with auto-reset.

Arduino Gemma: An ATtiny85 running at 8 MHz with auto-reset, 1 Analog In, 3 Digital I/O and 2 PWM.

4.4 CODE WITH COMMENTS/BRIEF DESCRIPTION OF FRAGMENTS OF STATEMENTS

<u>ARDUINO CODE:</u>

```
int trigPin1    = A1;  //left

int echoPin1    = A0;

int trigPin2    = A2;  //right

int echoPin2    = A3;

int MotorLeft   = 10;  //digital pin no.10

int MotorRight  = 11; //digital pin no.11
```

// The above line of code is used to declare the variables. As we have two no. of sensors are present, that's why we have mentioned left and right. Pin no. 10 and 11 are used to control the motor.

```
long Ldistance;

long Lduration;

long Rdistance;

long Rduration;
```

// The above line of code is also used to declare the variables. These are used to store the left and right distances and durations from sensors.

```
void setup() {
  Serial.begin(9600);
  pinMode(MotorLeft, OUTPUT );
  pinMode(MotorRight, OUTPUT );


  pinMode(trigPin1, OUTPUT );
  pinMode(echoPin1, INPUT );
  pinMode(trigPin2, OUTPUT );
  pinMode(echoPin2, INPUT );
}
```

// In the setup() function, the code initializes the serial communication at a baud rate of 9600 using Serial.begin(9600). This allows data to be sent and received between the Arduino and the computer. It then sets the pin modes for MotorLeft and MotorRight as outputs and trigPin1, echoPin1, trigPin2, and echoPin2 as outputs and inputs, respectively.

```
void loop() {
  LDist();
  RDist();
  Serial.print("LDist: ");  Serial.print(Ldistance);  Serial.print(" || ");
  Serial.print("RDist: ");  Serial.println(Rdistance);
       if(Ldistance>Rdistance){TurnLeft();}
   else if(Rdistance>Ldistance){TurnRight();}
```

```
  else if(Rdistance==Ldistance){Stop();delay(500);}
  /*   if(Ldistance>50 && Ldistance <150){TurnLeft();  }
 else if(Rdistance>50 && Rdistance <150){TurnRight();}*/
 else Stop();
 }
```

// The loop() function is where the main code execution occurs repeatedly. It calls the functions LDist() and RDist() to measure the distances from the left and right sensors, respectively. It then prints the distances to the serial monitor. Depending on the distance measurements, it decides the movement of the motors. If the left distance is greater than the right distance, it calls the TurnLeft() function. If the right distance is greater than the left distance, it calls the TurnRight() function. If both distances are equal, it calls the Stop() function and delays for 500 milliseconds before proceeding. If none of these conditions are met, it simply calls the Stop() function.

```
void TurnLeft()
{
  digitalWrite(MotorLeft,HIGH);
  digitalWrite(MotorRight,LOW);
}
void TurnRight()
{
digitalWrite(MotorLeft,LOW);
digitalWrite(MotorRight,HIGH) ;
}
void Stop()
{
```

```
  digitalWrite(MotorLeft,LOW);
  digitalWrite(MotorRight,LOW);

}
```

//These functions TurnLeft(), TurnRight(), and Stop() control the movement of the motors. They set the appropriate pins to control the motor movements accordingly.

```
void RDist()
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  Lduration = pulseIn(echoPin1, HIGH);
  Ldistance = Lduration*0.034/2;
  }
void LDist()
{
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);
```

```
Rduration = pulseIn(echoPin2, HIGH);

Rdistance = Rduration*0.034/2;

}
```

//These functions RDist() and LDist() calculate the distances using the ultrasonic sensors. They trigger the sensors by sending pulses, measure the duration of the echo signal using pulseIn(), and calculate the distance based on the speed of sound.

4.5 CONCLUSION

In this chapter we have discussed about the Arduino IDE and its associated libraries that is used to develop the software for controlling motors based on ultrasonic sensor readings. The Arduino IDE provided a user-friendly environment for writing, compiling, and uploading code to the Arduino board.

The code implemented various functions and logical conditions to measure distances from left and right ultrasonic sensors and control the motors accordingly. The setup() function initialized the necessary pin modes and serial communication, while the loop() function continuously monitored the distances and made decisions based on the readings.

The project showcased the versatility of Arduino IDE and its capability to interface with hardware components. By integrating the software and hardware aspects, the project successfully achieved the goal of autonomously controlling motors based on real-time distance measurements

CHAPTER 5

RESULTS

5.1 SIMULATION RESULTS

The simulation has been done using Proteus Professional software. Proteus is a widely used software tool in the field of electronic design automation (EDA). It provides a comprehensive suite of tools for designing, simulating, and testing electronic circuits.

With its user-friendly interface and powerful features, Proteus has become a popular choice for engineers, students, and professionals. One of the standout features of Proteus is its ability to simulate circuit behaviour in real-time.

Users can design and assemble circuits using a vast library of components, including microcontrollers, sensors, and integrated circuits. The software enables users to verify the functionality of their designs before moving to the physical prototyping stage, saving both time and resources.

Proteus also offers a virtual environment for testing and debugging embedded software. Users can write and upload code to microcontrollers, simulate their execution, and observe the resulting behaviour. This feature is particularly useful for ensuring that the software and hardware components of a project work seamlessly together.

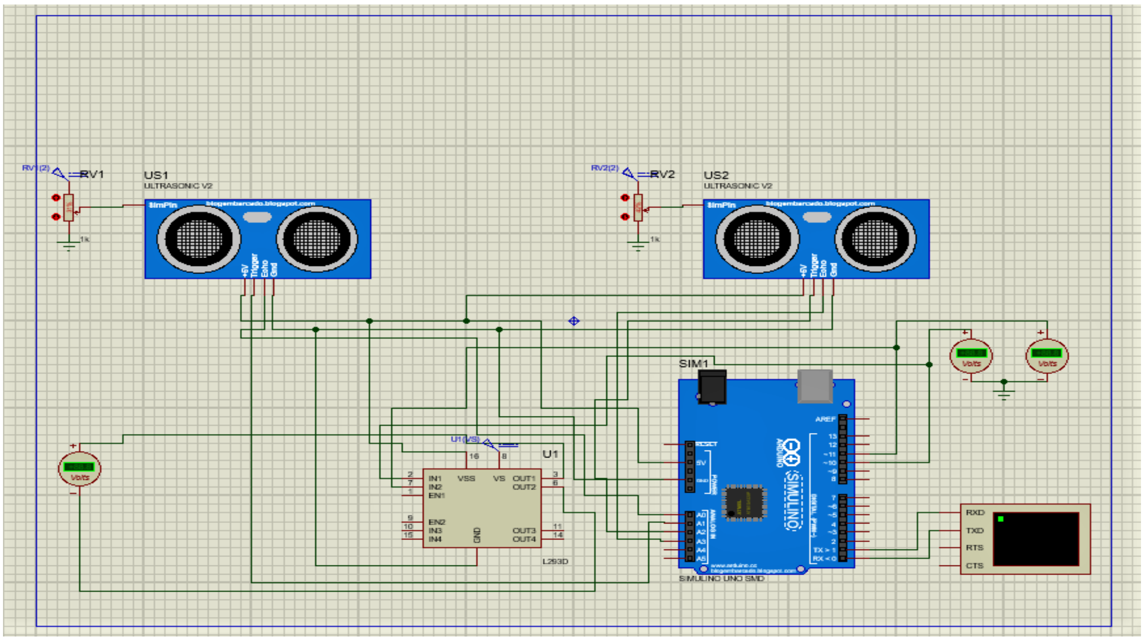


Fig.5.1(a) Simulation is carried out on Proteus.

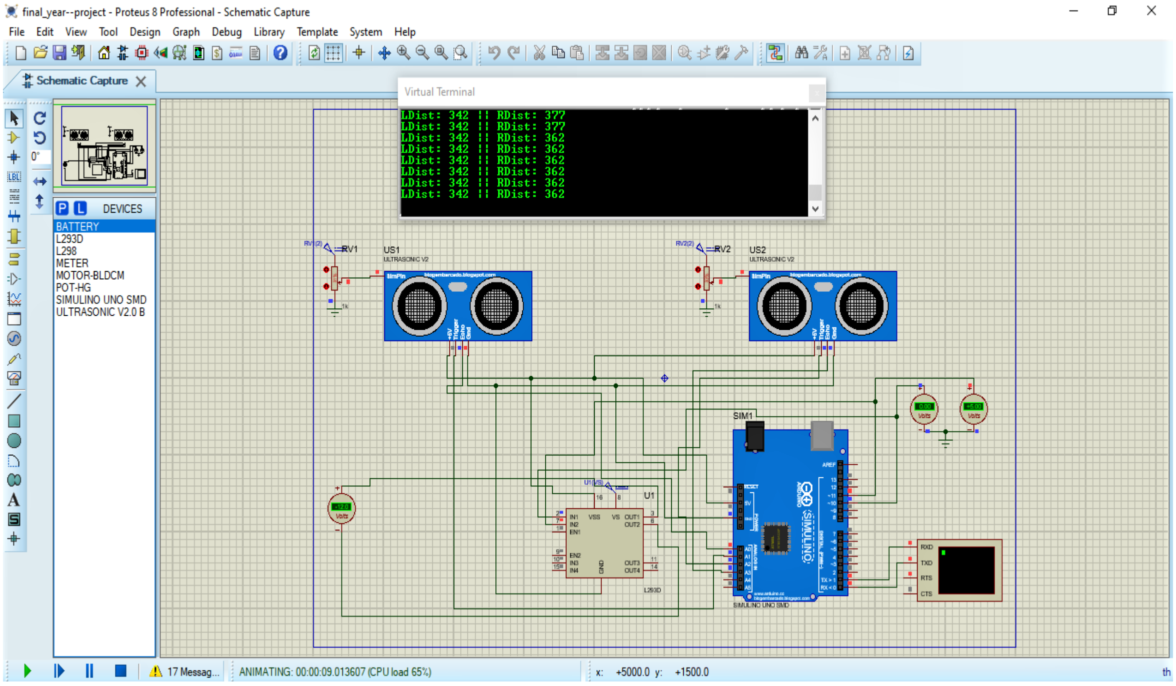


Fig.5.1(b) Result of the simulation is displayed at terminal.

5.2 HARDWARE

Project Pictures

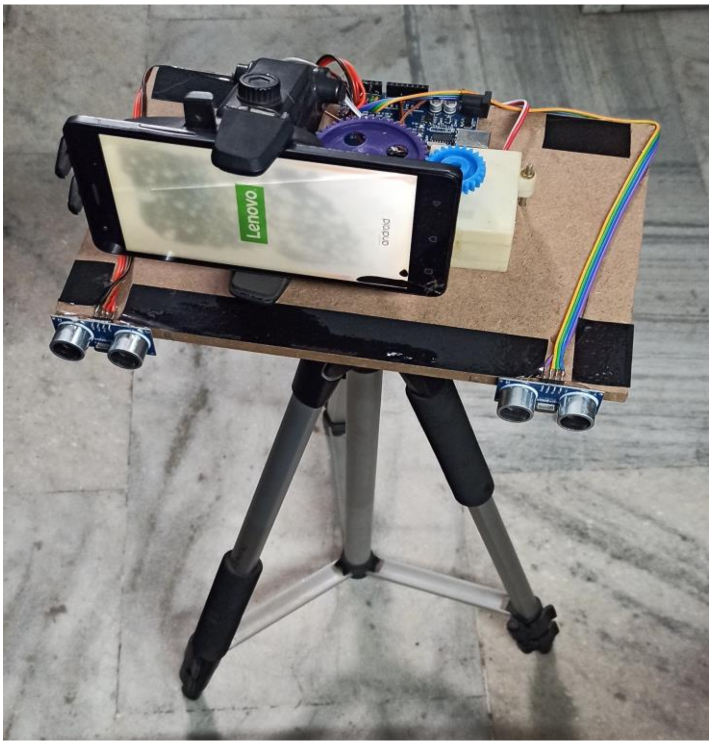


Fig.5.2 Final picture of Project

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The ultrasonic sensor-based automatic tripod project aims to develop a camera stabilization system that can automatically adjust the position of a camera mounted on a tripod using ultrasonic sensor technology. The project combines hardware components, such as an Arduino microcontroller, with the ultrasonic sensor and motorized tripod mechanisms to achieve the desired functionality.

The Arduino microcontroller serves as the brain of the system, responsible for processing data from the ultrasonic sensor and controlling the movement of the tripod. The ultrasonic sensor, typically mounted on the tripod, emits ultrasonic waves and measures the time it takes for the waves to bounce back after hitting an object. This information is used to calculate the distance between the sensor and the object in front of it. The Arduino code can be developed using the Arduino IDE (Integrated Development Environment) or other compatible programming environments.

Overall, the ultrasonic sensor-based automatic tripod offers a valuable tool for photographers and videographers seeking convenience, accuracy, and efficiency in their work. It simplifies the process of stabilizing cameras, reduces the need for manual adjustments, and enhances the overall quality of the captured content.

6.2 Future Scope

The future scope of the ultrasonic sensor-based automatic tripod project is quite promising. Here are some potential areas of development and expansion:

1. Advanced Sensor Technologies: While ultrasonic sensors are widely used in distance measurement, future iterations of the project could explore other sensor technologies, such as laser or infrared sensors, to further improve accuracy, range, and reliability. These sensors can provide more precise distance measurements and better adaptability to different environmental conditions.

2. Integration with Artificial Intelligence: Integrating AI algorithms into the automatic tripod system can enhance its capabilities. AI can enable the tripod to intelligently track and follow subjects, recognize specific objects or faces, and optimize camera settings based on the scene or shooting conditions. This integration would make the tripod more versatile and adaptable to various photography and videography scenarios.
3. Wireless Connectivity and Remote Control: Adding wireless connectivity capabilities to the automatic tripod would allow users to control and monitor the tripod remotely using smartphones or other devices. This feature would enable photographers and videographers to operate the tripod from a distance, expanding its usability in situations where physical proximity is challenging or inconvenient.
4. Multi-camera Support: Enhancing the tripod to support multiple cameras simultaneously would be beneficial for professionals and enthusiasts working with multiple camera setups. This feature would enable synchronized movements and allow for capturing different angles or perspectives simultaneously, enhancing the overall quality and versatility of the captured content.
5. Integration with Smart Home Systems: Integrating the automatic tripod with smart home systems and voice assistants could provide seamless control and automation. Users could issue voice commands to position the tripod, initiate specific shooting modes, or integrate it into broader smart home automation routines.
6. Gesture Control: Implementing gesture recognition technology would enable users to control the tripod's movements through hand gestures. This would provide a hands-free control option, particularly useful in situations where the user's hands are occupied or when working alone.
7. Expandability and Modular Design: Designing the tripod with a modular approach would allow users to customize and expand its functionality based on their specific needs. This could include adding accessories such as lighting attachments, microphones, or additional sensors for enhanced versatility and creative possibilities.

These future developments can take the ultrasonic sensor-based automatic tripod to new heights, transforming it into a sophisticated tool for photographers and videographers, offering advanced automation, intelligent tracking, and seamless integration with other technologies.

6.3 Cost Estimation:

This table represents the cost of various materials/components.

S No.	Materials	Qty.	Cost Per	Total
1	ULTRASONIC SENSORS	2	65	130
2	SERVO MOTOR	1	300	300
3	ARDUINO UNO	1	900	900
4	MOTOR DRIVER CIRCUIT	1	250	250
5	TRIPOD STAND	1	600	600
6	WORM GEAR	1	20	20
7	SPUR GEAR	2	10	20
8	TYRE	1	30	30
9	WIRES	10	10	10
TOTAL				2260/-

#All the prices written are in INR (₹)

REFERENCES

Examples of Journal/Article referencing:

[1]. Drucker, D. C., "Photoelastic Separation of Principal Stresses by Oblique Incidence”, *Journal of Applied Mechanics,* Vol. 65, pp. 156-160, 1943.

[2]. Maiers, J., and Sherif, Y. S., "Application of Fuzzy Set Theory," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-15, No.1, pp. 41-48, 1985.

Example of Book referencing:

[3]. Doe, N., *Control System Principles,* New York: John Wiley, 1999.

Example of Referencing of an Article in a Book:

[4]. Hwang, C. J., “Rule-based Process Control," in E. Kumar mangalam and L. A. Zadeh (Eds.), *Approximate Reasoning in Intelligent Systems, Decision and Control*, pp. 145-158, Oxford: Pergamon Press, 1987.

Example of referencing of a B. Tech. Report:

[5]. Nayak, T., "Application of Neural Networks to Nuclear Reactors,” M.Sc. Report, U.P. Technical University, 2005.

Example of referencing of a Ph. D. Dissertation:

[6]. Muskìn, H. L., "Development of A Knowledge-Based System for a Nuclear Power Plant," Ph.D. Dissertation, U. P. Technical University, 2003.

Example of referencing of a Conference Paper:

[7]. Lokhande, R., Arya, K. V., and Gupta, P., "Identification of Parameters and Restoration of Motion Blurred Images", *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC 2006),* pp. 89-95, Dijon, France, April 2- 7, 2006.

Example of referencing of a Paper presented at Conference but not Published:

[8]. Lokhande, R., and Gupta, P., "Identification of Parameters of Motion Images", presented at 5th International Conference on Cyber Systems, New Delhi, India, April 12-17, 2004

Example of referencing of a Report [Technical, Internal, or Memoranda]:

[9]. Das, A. R., Murthy D., and Badrinath J., A Comparison of Different Biometrics Traits, RSRE Memorandum No. 4157, RSRE

[10]. Jaishnu Moudgil, Shubhankar Mengi and Mudit Chopra, 360 Degree Rotating Vehicle to Overcome the Problem of Parking Space, *International Journal of Research in Mechanical Engineering and Technology*, *5(2)*, 2015, 22-25.

[11]. Sudip kachhia, Design of 360 Degree Rotating Car, *International Journal of Advance Research and Innovative Ideas In Education*, *2(5)*, 2016, 15–16.

[12]. K. Lohith, K. Lohith, Dr. S. R. Shankapal, M. H. Monish Gowda, Development of Four Wheel,*Scholars Journal of Engineering and Technology*,*12(1)*, 2013, 52-53.

[13]. Er. Amitesh Kumar, Dr.Dinesh.N.Kamble, Zero Turn Four Wheel Steering System, *International Journal of Scientific & Engineering Research*,*5(12)*, 2014, 22-24.

[14]. Mr. Sharad P. Mali, Mr. Sagar Jadhav, Prof. D.U.Patil, Zero Turn Four Wheel Mechanism, *International Engineering Research Journal*,*2(2)*, 2016, 484-486.

[15]. Website: https://www.elprocus.com/

[16]. Website: https://www.arduino.cc/

[17]. Website: https://www.tutorialspoint.com/index.htm

[18]. Website: www.javatpoint.com

[19]. Website: www.allaboutelectronics.org

[20]. Website: www.allaboutcircuits.com