

# **Bangladesh University of Business and Technology (BUBT)**



## **Arduino Based Radar System for Object Detection**

Course Code: CSE- 426

Course Title: Microcontroller and Embedded Systems

### **Submitted to**

Amir-UL Haque Bhuiyan

Assistant Professor, Department of CSE

Bangladesh University of Business and Technology

### **Submitted by**

Shakil Ahmed (17182103010)

Arifur Rahman Kawser (17182103012)

Md. Arifur Rahman (17182103038)

**Date of Submission: 31/7/2021**

# CONTENTS

LIST OF FIGURES	iv
LIST OF TABLES	v
LIST OF ABBREVIATIONS	v
ABSTRACT	vi

<b>CHAPTER 1</b>	<b>INTRODUCTION</b>	<b>7-8</b>
1.1	Introduction	7
1.2	Problem Statement	7
1.3	Objectives	8
1.4	Project Scope	8
1.5	Methodology	8
1.6	Project/Thesis Outline	8
<b>CHAPTER 2</b>	<b>OVERVIEW OF THE PROJECT</b>	<b>9-13</b>
2.1	The Idea	9
2.2	Historical Background Radar	9
2.3	Introduction	9
2.4	Technological Overview	9
2.4.1	Arduino Technology	10
2.4.2	Arduino Architecture	11
2.4.3	How to Program an Arduino	11
2.4.4	Basics Function of Arduino Technology	12
2.4.5	Advantages of Arduino Technology	13
<b>CHAPTER 3</b>	<b>HARDWARE COMPONENTS</b>	<b>14-29</b>
3.1	Introduction	14
3.1.1	Arduino Uno Board	14
3.1.2	Ultrasonic Sensor	16

3.1.3	Processing Software	18
3.1.4	Bread board	19
3.1.4.1	Introduction	22
3.1.4.2	What is inside of a Breadboard? How do they work?	23
3.1.4.3	Breadboard labels: rows, columns and buses	23
3.1.4.4	Using a breadboard	25
3.1.5	Wire	29
<b>CHAPTER 4</b>	<b>DESIGN AND IMPLEMENTATION</b>	<b>30</b>
4.1	Introduction	30
4.2	Schematic Diagram of The Project	30
<b>CHAPTER 5</b>	<b>RESULTS AND DISCUSSION</b>	<b>33-36</b>
5.1	Introduction	33
5.2	Result	35
5.2.1	Object Detection	35
5.2.2	Distance and Angle Calculation	36
<b>CHAPTER 6</b>	<b>CONCLUSIONSION</b>	<b>38- 39</b>
6.1	Introduction	38
6.2	Limitations of the Work	38
6.3	Future Scopes	39
6.4	Conclusions	39
6.5	Discussion.	39
<b>REFERENCES</b>		<b>40</b>
<b>APPENDIX A</b>		<b>41</b>

# LIST OF FIGURES

Fig.2.2.1	Arduino Board	4
Fig. 2.2.2	Arduino Architecture	5
Fig.2.2.3	Arduino Board Programming	6
Fig.3.1.1	Arduino Uno board	11
Fig.3.1.2	Ultrasonic Sensor	12
Fig.3.1.2. a	Working Principle of Ultrasonic Sensor	12
Fig.3.1.3	Processing Software	13
Fig.3.1.4.1. a	Breadboard Circuit	14
Fig.3.1.4.1. b	Better Options	15
Fig.3.1.4.1.c	Compatible with Breadboards	15
Fig.3.1.4.2	Inside a Breadboard	16
Fig.3.1.4.3	Paper Layer	17
Fig.3.1.4.4	Flexible Jumper Wires	19
Fig.3.1.5	Electronic Components	21
Fig.3.1.5. a	Printed Circuit Board	22
Fig4.2. a:	Schematic Diagram for Arduino Radar	22
Fig4.2. b	Complete connection for Arduino Radar	23
Fig.5.1	Processing Display Outlook	24
Fig.5.2.1. a	Normal Condition of Radar	25
Fig.5.2.1. b	When in object in the coverage area of Radar	26
Fig.5.2.2. a	Distance and Angle Calculation-1	26
Fig.5.2.2. b	Distance and Angle Calculation-2	27

## LIST OF TABLES

Table 3.1.1	Specification of Arduino Microcontroller	9
Table 3.1.4:	TWI pins on Arduino Boards	14
Table 3.1.4.2	Positive Vs Negative	23
Table 5.2.2	Distance and Angle Calculation	25

## LIST OF ABBREVIATIONS

AWG	American Wire Gauge
DC	Direct Current
DFU	Device Firmware Update
EPROM	Erasable Programmable Read-Only Memory
HWB	Heavy Water Board
IC	Integrated Circuit
IDE	Integrated Development Environment
IDE	Integrated Development Environment
KB	Kilo Byte
KHZ	Kilo Hertz
LED	Light Emitting Diode
MHZ	Mega Hertz
PCB	Printed Circuit Board
PPI	Points Per Inch
PWM	Pulse width Modulation
RADAR	Radio Detection and Ranging
SRAM	Static Random-Access Memory
TWI	The Welding Institute
USB	Universal Serial Bus

# ABSTRACT

RADAR is an object detection system which uses radio waves to determine the range, altitude, direction, or speed of objects. The radar dish or antenna transmits pulses of radio waves or microwaves which bounce off any object in their path. Arduino is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible. This project aims at making a RADAR that is efficient, cheaper and reflects all the possible techniques that a radar consists of.

The proposed system "ultrasonic radar for the object detection distance and the speed measurement" employs an ultrasonic module that includes an ultrasonic transmitter and receiver. It is operated by transmitting a 40 kHz frequency pulse which is not audible to the human ear.

Module rotate with step angle of the stepper motor with specific angle for a specific time interval and the distance between the object and system is measured by calculating time interval taken by the signal to transmit and the echo reception Whereas the detected signal is shifted toward the module or away from the module which give the information about the speed of that detected object which is shown on PPI DISPLAY

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Introduction**

RADAR is an object detection system which uses radio waves to determine the range, altitude, direction, or speed of objects. Radar systems come in a variety of sizes and have different performance specifications. Some radar systems are used for air-traffic control at airports and others are used for long range surveillance and early-warning systems. A radar system is the heart of a missile guidance system. Small portable radar systems that can be maintained and operated by one person are available as well as systems that occupy several large rooms.

Radar was secretly developed by several nations before and during World War II. The term RADAR itself, not the actual development, was coined in 1940 by the United States Navy as an acronym for radio Detection and Ranging. The term radar has since entered English and other languages as the common noun radar, losing all capitalization.

The modern uses of radar are highly diverse, including air traffic control, astronomy, air, antimissile systems; marine radars to locate landmarks and other ships; aircraft anti-collision systems ;ocean surveillance systems, outer space surveillance and rendezvous systems; meteorological precipitation monitoring; altimetry and flight control systems; guided missile target locating systems; and ground-penetrating radar for geological observations. High tech radar systems are associated with digital signal processing and are capable of extracting useful information from very high noise levels.

### **1.2 Problem Statement**

Since, electronic components when used to form any circuit require some amount of troubleshooting to make the circuit work according to our expectations. In our project, there were some problems that we had to deal with.

### **1.3 Objectives**

1. To develop an ultrasonic RADAR.
2. To detect the fixed or moving object.
3. To measure the distance of the object from the system.
4. To measure the angle of the moving object.
5. Reducing human time and effort

### **1.4 Scopes**

There are a lot of scopes to develop our project. If we can use wireless Ultrasonic sensors it is possible to cover the full surrounding area. We can also see the object that comes in the coverage area of radar by using a camera.

### **1.5 Methodology**

We take the help of the official website of Arduino. We also take the help of the internet and books for various sources of information. Required equipment is purchased from the local market

### **1.6 Project/Thesis Outline**

This Project/thesis is organized as follows:

**Chapter1** is entitled “Introduction”. It introduces problem statements, objectives and methodology used in this project.

**Chapter 2** is “Overview of The Project”.

**Chapter 3** is entitled “Hardware and components”.

**Chapter 4** Circuit diagram

**Chapter 5** is entitled “Results”.

**Chapter 6** Conclusions, Limitations of the work



# **CHAPTER 2**

## **OVERVIEW OF THE PROJECT**

### **2.1 The Idea**

The Army, Navy and the Air Force make use of this technology. The use of such technology has been seen recently in the self-parking car systems launched by AUDI, FORD etc. And even the upcoming driverless cars by Google like Prius and Lexus.

The project made by us can be used in any systems the customer may want to use like in a car, a bicycle or anything else. The use of Arduino in the project provides even more flexibility of usage of the above-said module according to the requirements

### **2.2 Historical Background of Radar**

The history of radar (where radar stands for Radio Detection and Ranging) started with experiments by Heinrich Hertz in the late 19th century that showed that radio waves are reflected by metallic objects. This possibility was suggested in James Clerk Maxwell's seminal work on electromagnetism. However, it was not until the early 20th century that systems able to use these principles were becoming widely available, and it was German inventor Christian Hülsmeyer who first used them to build a simple ship detection device intended to help avoid collisions in fog (Reichspatent Nr. 165546). Numerous similar systems, which provided directional information to objects over short ranges, were developed over the next two decades.

### **2.3 Introduction**

This project aims on the use of Ultrasonic Sensor by connected to the Arduino UNO board and the signal from the sensor further provided to the screen formed on the laptop to measure the presence of any obstacle in front of the sensor as well as determine the range and angle at which the obstacle is detected by the sensor.

### **2.4 Technological Overview**

An Arduino board is a one type of microcontroller-based kit. The first Arduino Technology was developed in the year 2005 by David Cuartielles and Massimo Banzi. The designers thought to provide an easy and low-cost board for students, hobbyists and Professionals to build devices. Arduino boards can be purchased from the seller or

Directly we can make at home using various basic components. The best examples of Arduino for beginners and hobbyists include motor detectors and thermostats, and Simple robots. In the year 2011, Ad fruit industries expected that over 3lakhs Arduino Boards had been produced. But, 7lakhs boards were in user's hands in the year 2013. Arduino technology is used in many operating devices like communication or Controlling.

### 2.4.1 Arduino Technology

A typical example of the Arduino board is Arduino Uno. It includes an ATmega328 Microcontroller and it has 28pins. The pin configuration of the Arduino Uno board is shown in the above. It consists of 14digital I/O pins. Where in 6 pins are used as pulse width modulation O/Ps and 6 Analog I/Ps, a USB connection, a power jack, a 16MHz crystal oscillator, a reset Button, and an ICSP header. Arduino board can be powered either from the personal Computer through a USB or external source like a battery or an adaptor. This board can operate with an external supply of 712V by giving voltage reference through the IOREf pin or through the pin Vin.

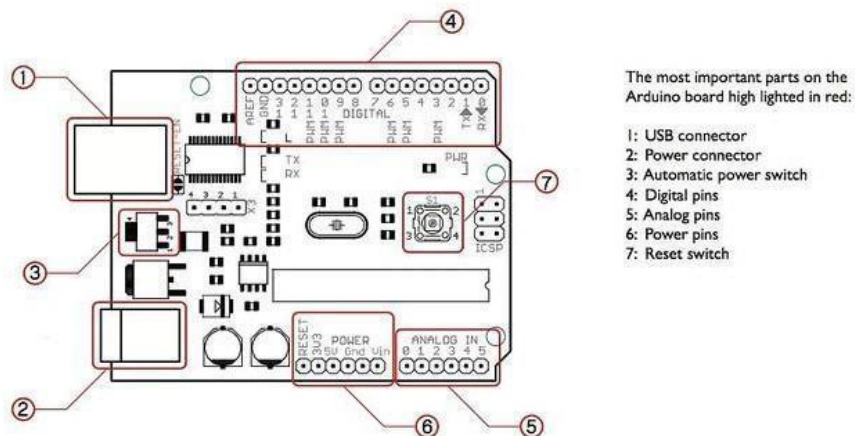


Fig. 2.2.1 Arduino board

This board can operate with an external supply of 712V by giving voltage reference Through the IOREf pin or through the pin Vin.

**Digital I/Ps:** It comprises of 14-digital I/O pins, each pin taking up and providing 40mA Current. Some of the pins have special functions like pins 0 & 1, which acts as a

Transmitter and receiver respectively. For serial communication, pins-2 & 3 are External interrupts, 3, 5, 6, 9, 11 pins deliver PWM O/P and pin-13 is used to connect LED.

**Analog I/Ps:** It has 6 analog I/O pins, each pin provides a 10 bits resolution.

**Aref:** This pin gives a reference to the analog i/ps.

**Reset:** When the pin is low, then it resets the microcontroller.

## 2.4.2 Arduino Architecture

Basically, the processor of the Arduino board uses the Harvard architecture where the Program code and program data have separate memory. It consists of two memories Such as program memory and data memory. Wherein the data is stored in data Memory and the code is stored in the flash program memory. The Atmega328 Microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and Operates with a 16MHz clock speed.

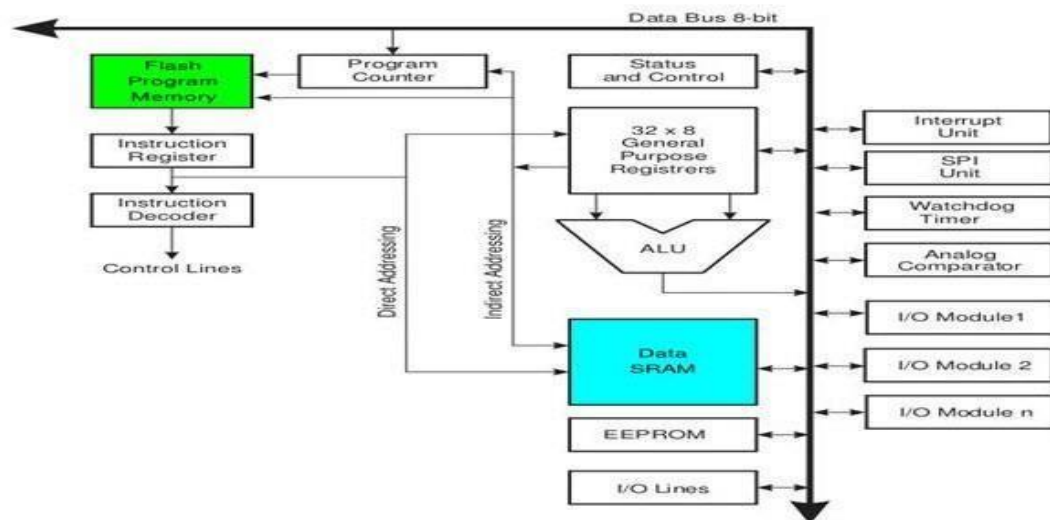


Fig. 2.2.2 Arduino Architecture

## 2.4.3 Program to Arduino

The main advantage of the Arduino technology is, we can directly load the programs into the device without the need of a hardware programmer to burn the program. This is done because of the presence of the 0.5KB of boot loader, that allows the program to be dumped into the circuit. The Arduino tool window contains a toolbar with various buttons like new, open, verify, upload and serial monitor. And additionally, it Comprises of a text editor (employed to write the code), a message space (displays the

Feedback) like showing the errors, the text console, that displays the o/p & a series of Menus just like the file, tool menu & edit.

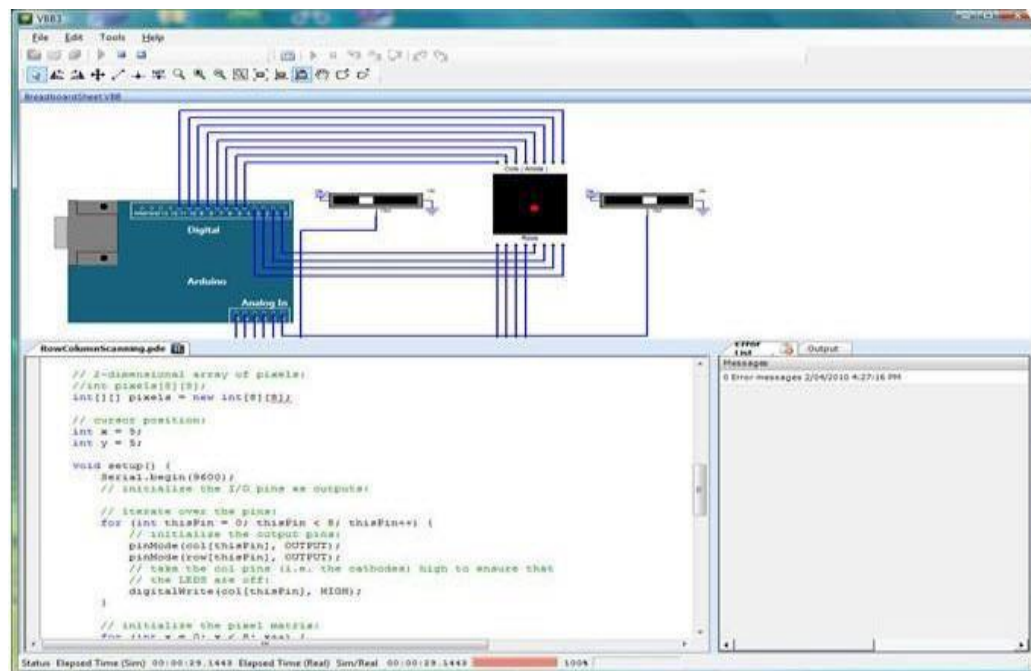


Fig. 2.2.3 Arduino board programming

1. Programming into the Arduino board is called as sketches. Each sketch contains of three parts such as Variables Declaration, Initialization and Control code. Where, Initialization is written in the setup function and Control code is written in the loop function.
2. The sketch is saved with .ino and any operation like opening a sketch, verifying and saving can be done using the tool menu.
3. The sketch must be stored in the sketchbook directory.
4. Select the suitable board from the serial port numbers and tools menu.
5. Select the tools menu and click on the upload button, then the bootloader uploads the code on the Arduino.

## 2.4.4 Basic Functions of Arduino Technology

1. Digital read pin reads the digital value of the given pin.
2. Digital write pin is used to write the digital value of the given pin.
3. Pin mode pin is used to set the pin to I/O mode.
4. Analog read pin reads and returns the value.
5. Analog write pin writes the value of the pin.

6. Serial Begins pin set the beginning of serial communication by setting threat of bit.

#### **2.4.5 Advantages of Arduino Technology**

1. It is cheap
2. It comes with an open supply hardware feature that permits users to develop their own kit
3. The software of the Arduino is wellsuited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
4. It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
5. For beginners, it is very simple to use.

# CHAPTER 3

## HARDWARE COMPONENTS

### 3.1 Introduction

In this chapter we are basically discussing about the hardware equipment and how we have connected them with our software. We are stating details about each and every component we have used to build this project. This project aims on the use of Ultrasonic Sensor by connected to the Arduino UNO board and the signal from the sensor further provided to the screen formed on the laptop to measure the presence of any obstacle in front of the sensor as well as determine the range and angle at which the obstacle is detected by the sensor. Therefore, when are making connections we are ought to be very careful. Therefore, every hardware setup is mentioned in this chapter

#### 3.1.1 Arduino Uno board

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

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

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5 V and with the Arduino Due that operates with 3.3V. The second one is a

not connected pin, that is reserved for future purposes.

Stronger RESET circuit.

Atmega 16U2 replace the 8U2.

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

Table 3.1.1 Specification of Arduino Microcontroller

<b>ARDUINO MICROCONTROLLER</b>	
Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	6
EEPROM	1 KB
DC Current per I/O Pins	40 mA on I/O Pins; 50 mA on 3,3 V Pin
<b>GENERAL</b>	
Input Voltage	7-12 V
Digital I/O Pins	20 (of wich 6 provide PWM output)
PWM Output	6
PCB Size	53.4 x 68.6 mm
Weight	25 g
Product Code	A000066 (TH); A000073 (SMD)

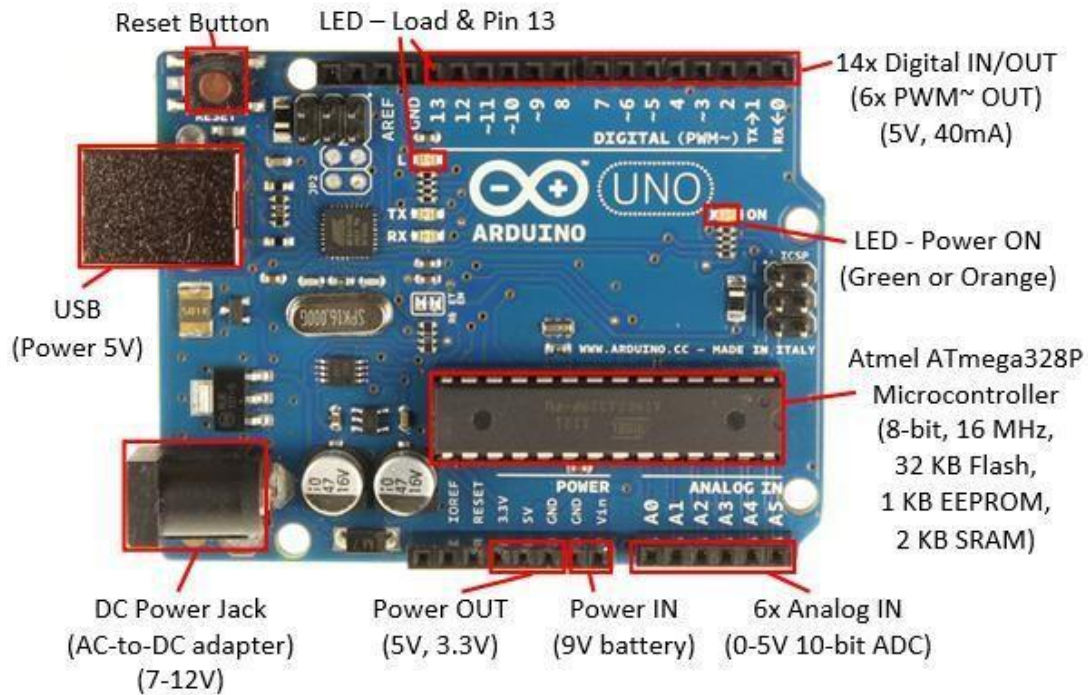


Fig 3.1.1: Arduino Uno board

### 3.1.2 Ultrasonic sensor

Ultrasonic Sensors(also known as transceivers when they both send and receive, but more generally called transducers) work on a principle similar to radar or sonar which evaluate attributes of a target by interpreting the echoes from radio or sound waves respectively. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object. This technology can be used for measuring wind speed and direction (anemometer), tank or channel level, and speed through air or water. For measuring speed or direction, a device uses multiple detectors and calculates the speed from the relative distances to particulates in the air or water. To measure tank or channel level, the sensor measures the distance to the surface of the fluid. Further applications include: humidifiers, sonar, medical ultra sonography, burglar alarms and nondestructive testing. Systems typically use a transducer which ge

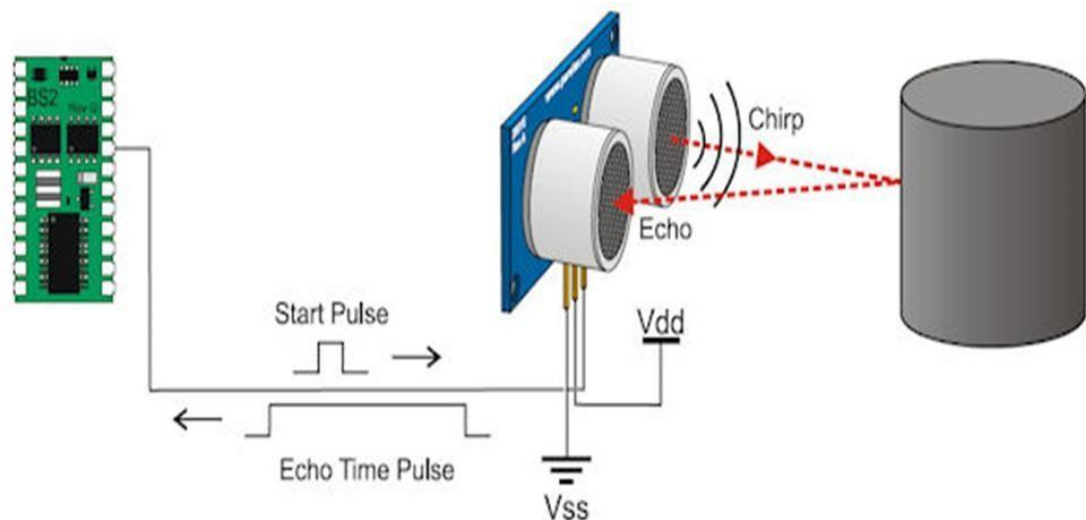


nerates sound waves in the ultrasonic range, above 18,000 hertz, by turning electrical energy into sound, then upon receiving the echo turn the sound waves into electrical energy which can be measured and displayed.

**Theory:** An Ultrasonic Sensor consists of four wires. One for Vcc, second for Ground, third for Echo pin and the last one for Trig pin. The ultrasonic sensor is mounted on the servo motor and both of them further connected to the Arduino board. The ultrasonic sensor uses the reflection principle for its working. When connected to the Arduino, the Arduino provides the pulse signal to the ultrasonic sensor which then sends the ultrasonic wave in forward direction. Hence, whenever there is any obstacle detected or present in front, it reflects the waves which are received by the ultrasonic sensor. If detected, the signal is sent to the Arduino and hence to the PC/laptop to the processing software that shows the presence of the obstacle on the rotating RADAR screen with distance and the angle at which it has been detected.



FigT3.1.2:Ultrasonic sensor



FigT3.1.2.a:TWorKingTPrincipleToF Ultrasonic Sensor

### 3.1.3 Processing Software

Processing is an open source programming language and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching the fundamentals of computer programming in a visual context, and to serve as the foundation for electronic sketchbooks. The project was initiated in 2001 by Casey Reas and Benjamin Fry, both formerly of the Aesthetics and Computation Group at the MIT Media Lab. One of the stated aims of Processing is to act as a tool to get nonprogrammers started with programming, through the instant gratification of visual feedback. The language builds on the Java language, but uses a simplified syntax and graphics programming models

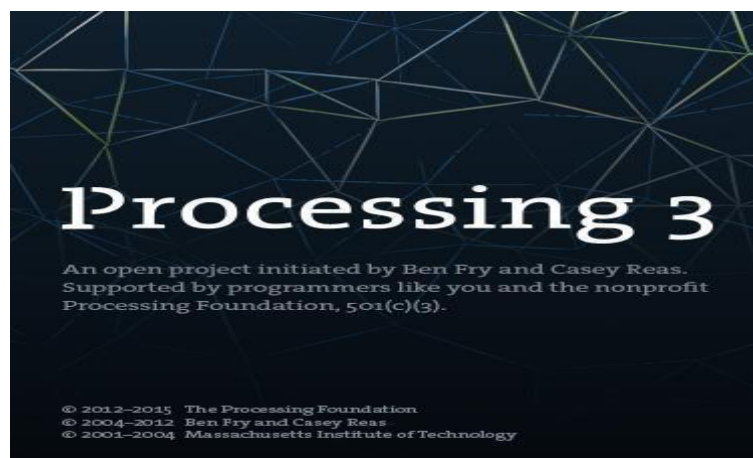


Fig:3.1.3 Processing Software 3

### 3.1.4 Bread Board 3.1.4.1 Introduction

A breadboard is used to build and test circuits quickly before finalizing any circuit design. The breadboard has many holes into which circuit components like ICs and resistors can be inserted. The breadboard has strips of metal which run underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally while the remaining holes are connected vertically. To use the breadboard, the legs of components are placed in the holes. Each set of holes connected by a metal strip underneath forms a node. A node is a point in a circuit where two components are connected. Connections between different components are formed by putting their legs in a common node. The long top and bottom row of holes are usually used for power supply connections. The rest of the circuit is built by placing components and connecting them together with jumper wires. ICs are placed in the middle of the board so that half of the legs are on one side of the middle line and half on the other. A completed circuit might look like the following.

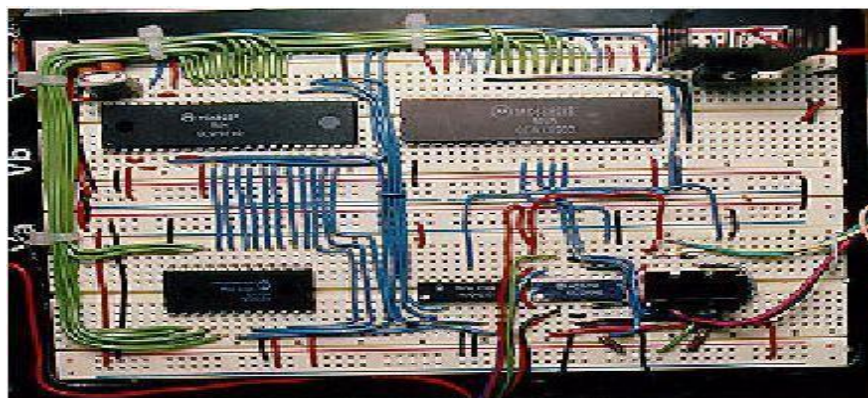


Fig 3.1.4.1.a: breadboard circuit

### Bread Boarding Tips:

It is important to breadboard a circuit neatly and systematically, so that one can debug it and get it running easily and quickly. It also helps when someone else needs to understand and inspect the circuit. Here are some tips:

- 1. Always use the sidelines for power supply connections. Power the chips from the side-lines and not directly from the power supply.
- 2. Use black wires for ground connections (0V), and red for other power connections.
- 3. Keep the jumper wires on the board flat, so that the board does not look cluttered.
- 4. Route jumper wires around the chips and not over the chips.
- This makes changing the chips when needed easier.
- 5. We could trim the legs of components like resistors, transistors and LEDs, so that they fit in snugly and do not get pulled out by accident.

We might be wondering what any of this has to do with bread. The term breadboard comes from the early days of electronics, when people would literally drive nails or screws into wooden boards on which they cut bread in order to connect their circuits. Luckily, since we probably do not want to ruin all our cutting boards for the sake of an electronics project, today there are better options.



Fig 3.1.4.1.b: better options

Modern breadboards are made from plastic, and come in all shapes, sizes, and even different colors. While larger and smaller sizes are available, the most common sizes we will probably see are "full-size," "halfsize," and "mini" breadboards. Most breadboards also come with tabs and notches on the sides that allow us to snap multiple boards together. A single half-sized breadboard is sufficient for many beginner-level projects.

So, how do electronic components fit into a breadboard? Many electronic components have long metal legs called leads (pronounced "Leeds"). Sometimes, shorter metal legs are referred to as pins instead. Almost all components with leads will work with a breadboard (to learn more about these components and which types work with a breadboard, see the advanced section).

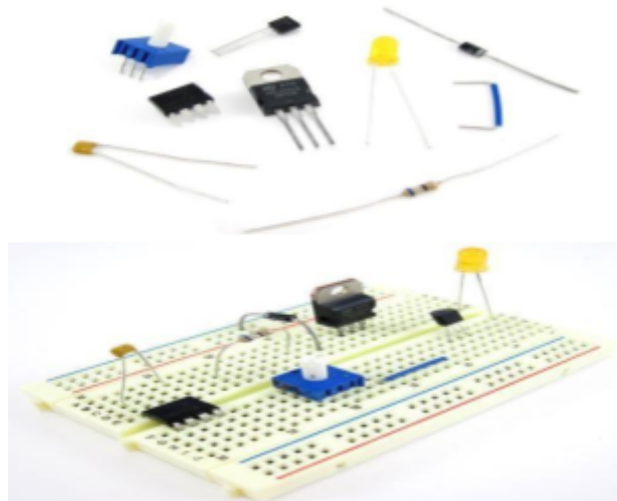


Fig 3.1.4.1.c:compatible with breadboards

Breadboards are designed so we can push these leads into the holes. They will be held in place snugly enough that they will not fall out (even if we turn the breadboard upside down), but lightly enough that we can easily pull on them to remove them.

We do not need any special tools to use a solderless breadboard. However, many electronic components are very tiny, and we may find them difficult to handle. A pair of miniature needle nose pliers or tweezers may make it easier to pick up small components.

#### 3.1.4.2 Inside a Breadboard:

The leads can fit into the breadboard because the inside of a breadboard is made up of rows of tiny metal clips. This is what the clips look like when they are removed from a breadboard.

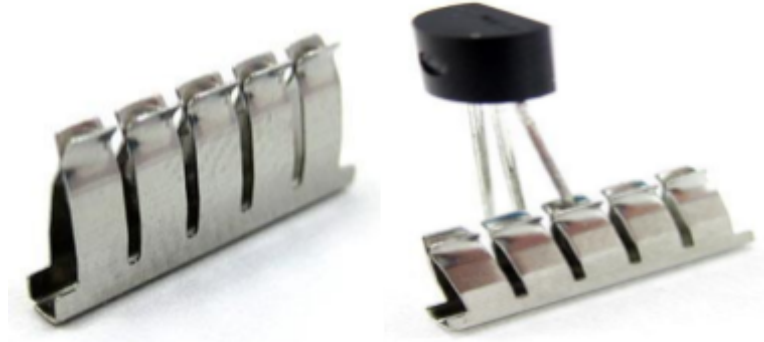


Fig 3.1.4.2: Inside a Breadboard

When we press a component's lead into a breadboard hole, one of these clips grabs onto it. Some breadboards are actually made of transparent plastic, so we can see the clips inside.

Most breadboards have a backing layer that prevents the metal clips from falling out. The backing is typically a layer of sticky, double-sided tape covered by a protective layer of paper. If we want to permanently "stick" the breadboard to something (for example, a robot), we just need to peel off the paper layer to expose the sticky tape underneath. In this picture, the breadboard on the right has had its backing removed completely (so we can see all the metal clips). The breadboard on the left still has its sticky backing, with one corner of the paper layer peeled up.



Fig 3.1.4.2: Paper layer

### ***3.1.4.3 Breadboard labels: rows, columns, and buses***

Most breadboards have some numbers, letters, and plus and minus signs written on them. What does all that mean? While their exact appearance might vary from breadboard to breadboard, the general purpose is always the same. These labels help we locate certain holes on the breadboard so we can follow directions when building a circuit. If we have ever used a spreadsheet program like Microsoft Excel® or Google Sheets™, the concept is exactly the same. Row numbers and column letters help we identify individual holes in the breadboard, just like cells in a spreadsheet. For example, all of the highlighted holes are in "column C."

These strips are typically marked by red and blue (or red and black) lines, with plus (+) and minus (-)

) signs, respectively. They are called the buses, also referred to as rails, and are typically used to supply electrical power to our circuit when we connect them to a battery pack or other external power supply. We may hear the buses referred to by different names; for example, power bus, positive bus, and voltage bus all refer to the one next to the red line with the plus (+) sign. Similarly, negative bus and ground bus both refer to one next to the blue (or black) line with the minus (-)

) sign. Sound confusing? Use this table to help we remember—

there are different ways to refer to the buses, but they all mean the same thing. Do not worry if we see them referred to by different names in different places (for example, in different Science Buddies projects or other places on the internet). Sometimes we might hear "power buses" (or rails) used to refer to both of the buses (or rails) together, not just the positive one.



Positive	Negative
Power	Ground
Plus, sign (+)	Minus sign (-)
Red	Blue or black

Table 3.1.4.2: Positive vs Negative

Note that there is no physical difference between the positive and negative buses, and using them is not a requirement. The labels just make it easier to organize our circuit, similar to color-coding our wires.

Remember that the inside of the breadboard is made up of sets of five metal clips. This means that each set of five holes forming a halfrow (columns A–E or columns F–J) is electrically connected. For example, that means hole A1 is electrically connected to holes B1, C1, D1, and E1. It is not connected to hole A2, because that hole is in a different row, with a separate set of metal clips. It is also not connected to holes F1, G1, H1, I1, or J1, because they are on the other "half" of the breadboard—

the clips are not connected across the gap in the middle (to learn about the gap in the middle of the breadboard, see the Advanced section). Unlike all the main breadboard rows, which are connected in sets of five holes, the buses typically run the entire length of the breadboard (but there are some exceptions). This image shows which holes are electrically connected in a typical half-sized breadboard, highlighted in yellow lines.

Buses on opposite sides of the breadboard are not connected to each other. Typically, to make power and ground available on both sides of the breadboard, we would connect the buses with jumper wires, like this. Make sure to connect positive to positive and negative to negative.



#### 3.1.4.4 Using a breadboard

Jumper wires are wires that are used to make connections on a breadboard. They have stiff ends that are easy to push into the breadboard holes. There are several different options available when purchasing jumper wires.

Flexible jumper wires are made of a flexible wire with a rigid pin attached to both ends. These wires usually come in packs of varying colors.

This makes it easy to color-code our circuit (see the section on colorcoding).

While these wires are easy to use for beginner circuits, they can get very messy for more complicated circuits; because they are so long, we will wind up with a tangled nest of wires that are hard to trace (sometimes called a "rats nest" or "spaghetti").

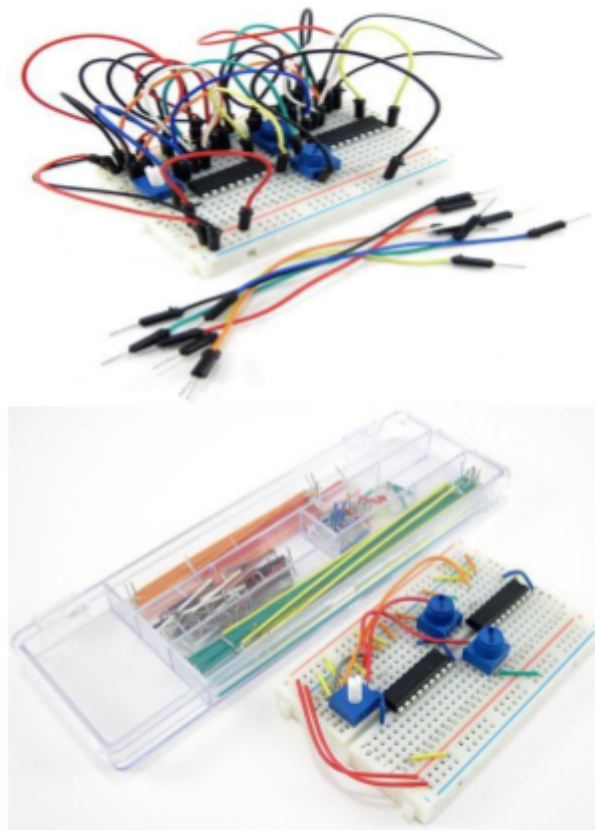


Fig 3.1.4.4 : Flexible jumper wires

Jumper wire kits are packs of precut lengths of wire that have their ends bent down 90 degrees, so they are ready to put into a breadboard. The kits are available in larger and smaller sizes. These kits are very convenient because they come with wires of many different precut lengths. The disadvantage is that there is typically only one length of each color. This can make it difficult to colorcode our circuit (for example, we might want a long

black wire, but our kit might only have short black wires). our circuit will work just fine, but colorcoding can help we stay more organized again, see the section on colorcoding for more information). Notice how this circuit looks much less messy than the previous one, since the wires are shorter.

Finally, we can also buy spools of solidcore hookup wire and a pair of wire strippers and cut our own jumper wires. This is the best longterm of if we plan on doing lots of electronics projects, because we can cut wires to the exact length we need, and pick which colors we want. It is also much more costeffective per length of wire. Buying a kit of six different colors is a good place to start. It is important to buy solid core wire (which is made from a single, solid piece of metal) and not stranded wire (which is made from multiple, smaller strands of wire, like a rope). Stranded wire is much more flexible, so it is very hard to push into a breadboard's holes. We also need to purchase the right wire gauge, which is a way of measuring wire diameter. 22 AWG (American Wire Gauge) is the most common gauge used for breadboards. To learn more about wire gauge and how to strip wire, see the Science Buddies Wire Stripping Tutorial. Notice how in this circuit, red and black are used for all the connections to the buses (see the section on color-coding to learn more).

To build a circuit:

1. Follow the breadboard diagram for the circuit, connecting one component at a time.
2. Always connect the batteries or power supply to our circuit last. This will give us a chance to doublecheck all our connections before we turn our circuit on for the first time.
3. Keep an eye out for common mistakes that many beginners make when using a breadboard.

How we test our circuit will depend on the specific circuit we are building. In general, we should follow this procedure:

1. Doublecheck our circuit and the breadboard diagram to make sure all our components are in the right place.
2. Check what our circuit is supposed to do according to the project of reactions. Is it supposed to flash lights, make noise, somehow respond to a sensor (like a motion or light sensor), or make a robot move?

Many Science Buddies projects will contain a written description and/or video of how our circuit should work.

3. Turn the power to our circuit on (for example, by sliding a battery pack switch from OFF to ON). If we see or smell smoke, turn off or disconnect the power supply immediately. This means we have a short circuit.
4. Follow the project directions to use our circuit (for example, shining a flashlight at a lighttracking robot, or waving our hand in front of a motion sensor).
5. If our circuit does not work, we need to troubleshoot (or **debug**, meaning to look for problems or "bugs" in our circuit). See the common mistakes section for things we should check.

### **Integrated circuits(ICs):**

Integrated circuits, or ICs for short (sometimes just referred to as "chips") are specialized circuits that serve a huge variety of purposes, such as controlling a robot's motors or making LEDs respond to music. Many ICs come in something called a dual inline package, or DIP, meaning they have two parallel rows of pins. The gap in the middle of a breadboard (between columns E and F) is just the right width for an IC to fit, straddling the gap, with one set of pins in column E, and one set of pins in column F. Projects that use ICs will always tell us to connect them to the breadboard in this manner.

### **Through-hole vs. surface mounts parts:**

Breadboards are designed to work with through-hole electronic components. These components have long metal leads that are designed to be inserted through holes in a printed circuit board (PCB) that are plated with a thin copper coating, which allows the components' leads to be soldered to the board.

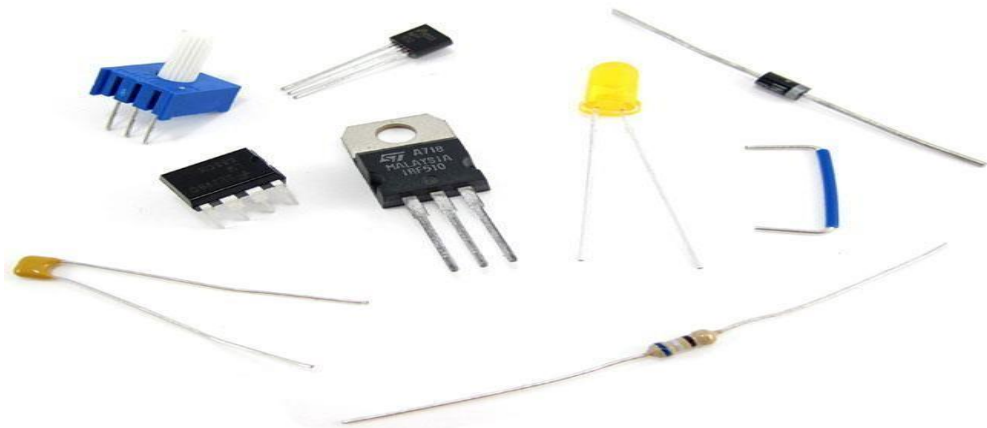


Fig 3.1.4.6 :electronic components

Breadboards do not work with surface mount components. These components have short, flat pins on their sides that are designed to be soldered to the surface of a printed circuit board, instead of through holes.

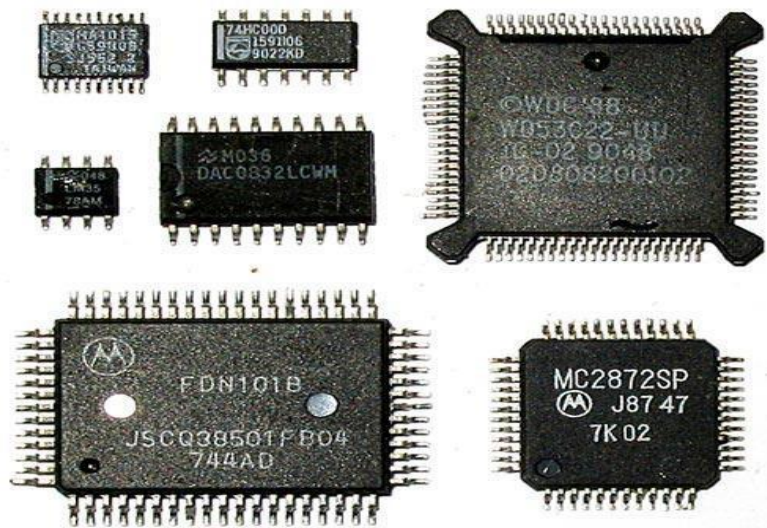


Fig 3.1.4.6.a : printed circuit board

Many electronic components are available in both through-hole and surface mount packages. For example, the LM3914 is an integrated circuit that is designed to drive 10 LEDs as a "bar graph" display. If we search Jameco Electronics for "LM3914", several different results come up. We can tell from looking at the thumbnails that this part is through hole and this part is surface mount. While most Science Buddies projects will link to exactly what parts we need to buy for a project, be careful if we are buying parts for our own project. If we are using a breadboard, make sure we buy through-hole parts and not surface mount.

### 3.1.5 Wire

This library allows us to communicate with I2C / TWI devices. On the Arduino boards with the R3 layout (1.0 pinout), the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin. The Arduino Due has two I2C / TWI interfaces SDA1 and SCL1 are near to the AREF pin and the additional one is on pins 20 and 21. As a reference the table below shows where TWI pins are located on various Arduino boards.

Table 3.1.4: TWI pins on Arduino Boards	
Board	I2C / TWI pins
Uno, EthernetA4	(SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

As of Arduino 1.0, the library inherits from the Stream functions, making it consistent with other read/write libraries. Because of this, `send()` and `receive()` have been replaced with `read()` and `write()`.

#### *Examples*

Digital Potentiometer: Control an Analog Devices AD5171 Digital Potentiometer. Master: Program two Arduino boards to communicate with one another in a Master Reader/Slave Sender configuration via the I2C. Master Writer/Slave receiver: Program two Arduino boards to communicate with one another in a Master Writer/Slave Receiver configuration via the I2C.

# CHAPTER 4

## DESIGN AND IMPLEMENTATION

### 4.1 Introduction

Radar still a trusted technology throughout the world. The design of the ultrasonic Radar is very useful for many applications like homes, shops, military and object detection. the aim of this work is to build an ultrasonic transceiver which is basically one kind of a radar system to get exact distance angle for fixed objects placed around the device based on the speed of ultrasonic waves in open air.

### 4.2 Connection Diagram of the Project

The main objective of our project is to determine the exact placement of a object. In the previous chapter we give all the information about hardware components. In this chapter we will discuss about the connection of all components.

For this project we need a 5v power supply. We use computer as a supply by a USB cable which will directly connected to the 5v power of arduino. After that we will take connection by a jumper wire from 5v pin of Arduino to positive pin of breadboard Then, we take another Wire that is connected to GND pin of Arduino to negative pin of breadboard.

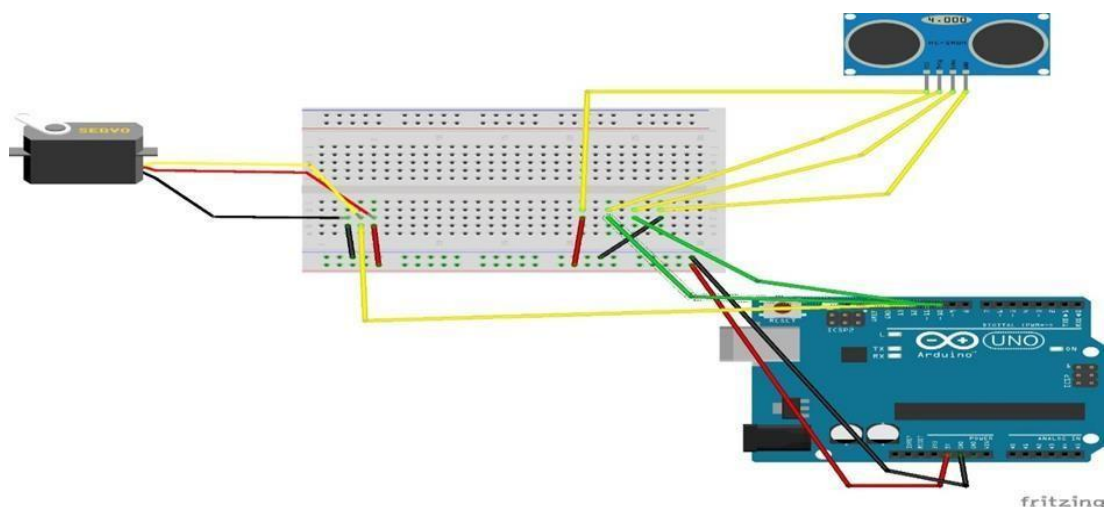


Fig4. 2.a: Connection Diagram for Arduino Radar

Now we will connect the servo motor to the Arduino. Servo motor has

3 wire. We have to give 5v supply to motor by Red wire. We used brown wire as a ground that is connected to the negative pin of breadboard. The last wire that is yellow we will connect it to the Arduino digital pin 12.

Ultrasonic sensor has 4 pins. Vcc pin connected to the 5v supply by jump wire. GND pin connected to the GND of breadboard. Trig and Echo pin connected to the 10 & 11 no pin of Arduino. Trig pin sends a sound wave to the air and Echo pin receive the return wave. So, we can say that Trig pin is the input pin and Echo pin is the output pin of sensor.

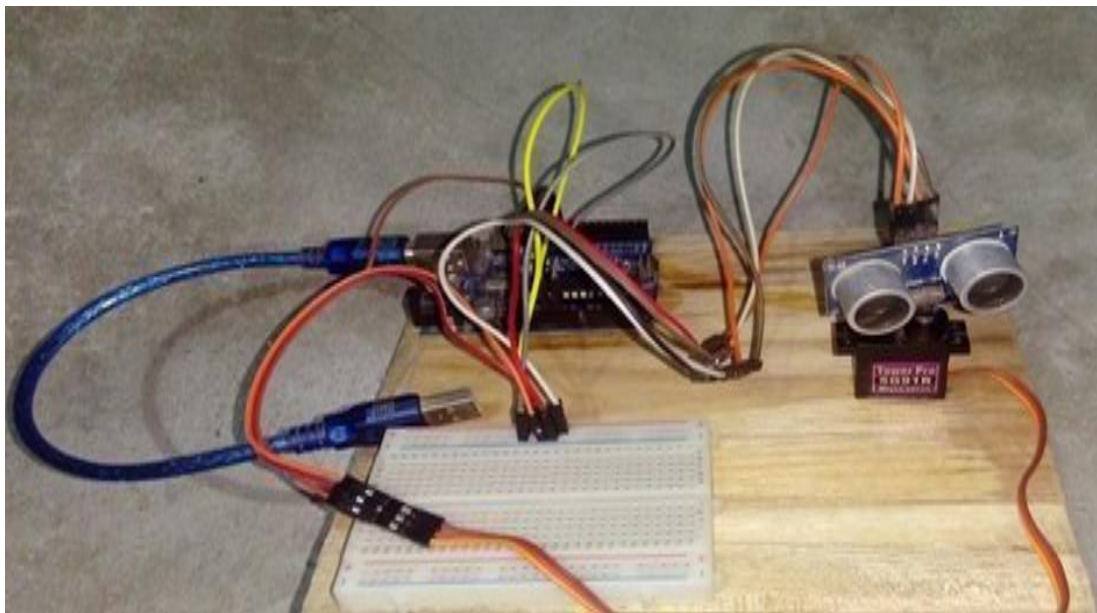


Fig 4.2.b: Complete connection for Arduino Radar

Ultrasonic is a noncontact level measurement method that uses sound waves to determine the process material being measured. Ultrasonic transmitters operate by sending a sound wave, generated from a piezo electric transducer, to the media being measured. The device measures the length of time it takes for the reflected sound wave to return to the transducer.

A successful measurement depends on reflection from the process material in a straight line back to the transducer.

However, there are various influences that affect the return signal. Factors such as dust, heavy vapors, tank obstructions, surface turbulence, foam and even surface angles can affect the returning signal. That is why the co

Conditions that determine the characteristics of sound must be considered when using Ultrasonic measurement.



# CHAPTER 5

## RESULTS AND DISCUSSION

### 5.1 Introduction

Initially, upload the code to Arduino after making the connections. We can observe the servo sweeping from  $0^{\circ}$  to  $180^{\circ}$  and again back to  $0^{\circ}$ . Since the Ultrasonic Sensor is mounted over the Servo, it will also participate in the sweeping action.

Now, open the processing application and paste the code that we already wrote. In the Processing Sketch, make necessary changes in the COM Port selection and replace it with the COM Port number to which our Arduino is connected to COM3.

After note the Processing Sketch, we used the output display size as  $1280 \times 720$ .

Now, run the sketch in the Processing and if everything goes well, a new Processing window opens up like the one shown below.

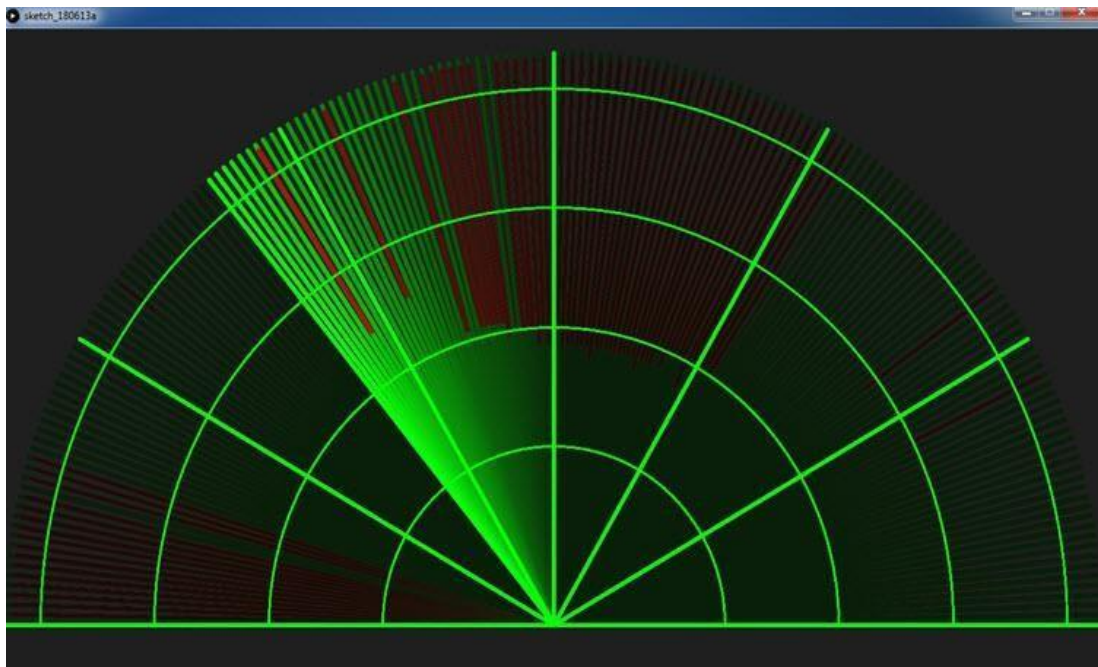


Fig 5.1: Processing Display Outlook

A Graphical representation of the data from the Ultrasonic Sensor is

represented

in a Radar type display. If the Ultrasonic Sensor detects any object within its range, the same will be displayed graphically on the screen.

## 5.2 Result

After complete all the hardware and software connections perfectly now we are ready to perform our test. Our ultrasonic radar is capable of detect any object both in normal and dark situation. It can also measure the distance and angle of the object. We will show all the test step by step. Our main objective was to build a project that can reduce the human effort. I think our project capable of more than as our expectations.

### 5.2.1 Object Detection

Under normal situation when the coverage area of ultrasonic sensor is clear that is there is no object or particle in the range of ultrasonic sensor the processing software whole display will show green light. When any object or particle comes in the range of ultrasonic sensor which is up to 40cm the display will show a red light and mark the position of particle or object. It will show red light until that object or particle don't leave the coverage area of sensor.

Two pictures are given below as an evidence of our statement,



Fig 5.2.1 a: Normal condition of radar

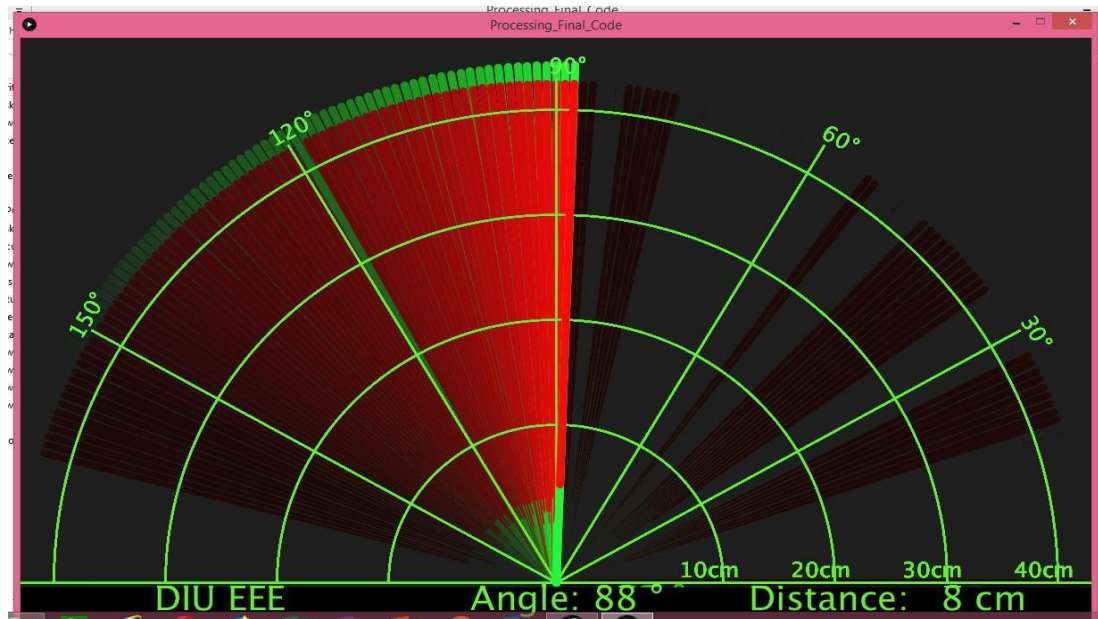


Fig 5.2.1.b: When any object in the coverage area of radar

### 5.2.2 Distance and Angle Calculation

Distance and angle calculation are the main part of our project. When this project idea came in our head, first our target was to make a radar that can detect an object only. Then we think that if we use some extra equipment it is possible to calculate the distance and angle. Finally, we make a radar that is capable of calculate the angle and distance with object detection. We provide two pictures as an evidence of our statement.



Fig5.2.2. a: Distance and Angle Calculation-1

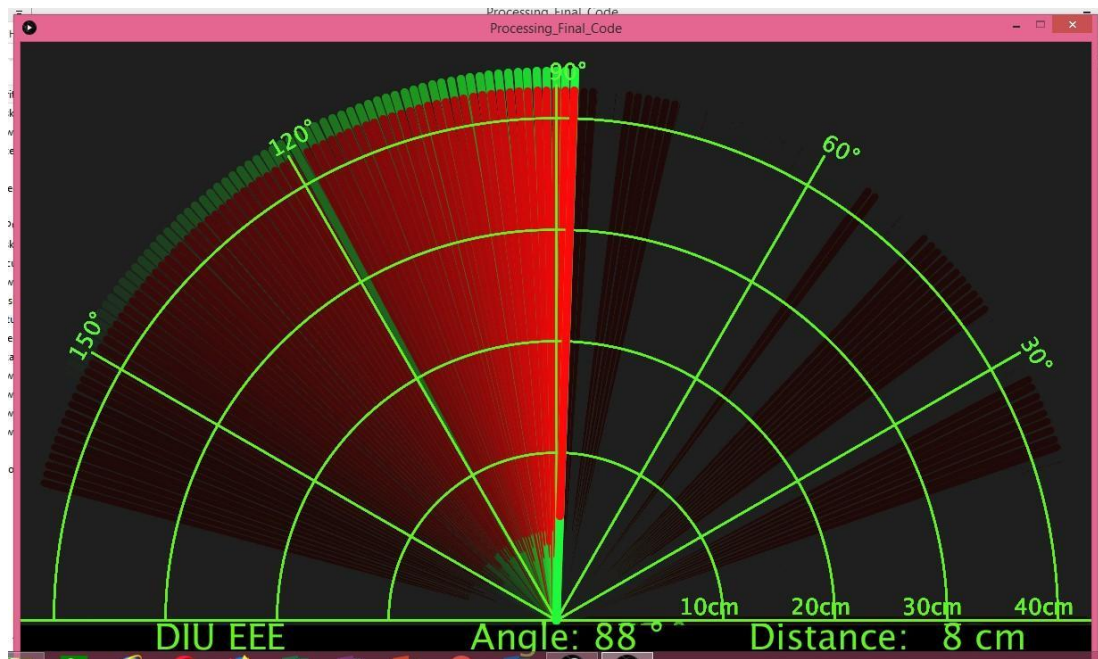


Fig 5.2.2.b: Distance and Angle Calculation-2

Serial & Object	Angle (Radar)	Angle (Manual)	Distance Measured by Radar	Distance Measured by Scale
1.Pen	88	90	9 cm	8cm
2.Human	139	142	24cm	24cm
3.Book	58	60	13cm	12cm
4.Cup	30	29	19cm	18cm
5				

Table 5.2.2: Distance and Angle Calculation

# CHAPTER 6

## CONCLUSIONS

### 6.1 Introduction

This project aims on the use of Ultrasonic Sensor by connected to the Arduino UNO board and the signal from the sensor further provided to the screen formed on the laptop to measure the presence of any obstacle in front of the sensor as well as determine the range and angle at which the obstacle is detected by the sensor. For this screen, we use Processing 2 software by Ben Fry and Casey Rease, Massachusetts Institute of Technology, Cambridge.

### 6.2 Limitations of the work

Although we fully believe in the capability of our sensors, we understand that ultrasonic are not suited for every application. Below we go into the *limitations of our project*.

1. Height of objects cannot be determined.
2. 3D mapping of object is not possible.
3. The range depends upon the characteristics of the sensor
4. Cannot work in a vacuum
5. Not designed for underwater use

### **6.3 Future Scopes**

We have represented a project on Ultrasonic RADAR for security system for human or object interference in a short range. The system has been successfully implemented and the aim is achieved without any deviation. There is a lot of future scope of this project because of its security capacity. It can be used in many applications. This project can also be developed or modified according to the rising needs and demand.

### **6.4 Conclusions**

In this paper, a lab scaled radar system was designed and implemented using an Arduino, a servomotor and an ultrasonic sensor. The developed system is able to read the distance of obstacles and the angle of incident and convert this data into visually represented information. The system performance measures up with other systems at its level as it adequately reports any obstacle it finds in its path and provides an estimated range of the object. A very handy application for this system would be in the area of object detection and avoidance systems for robotics or maybe in intrusion detection systems for location sizes where it may not be economical to use multiple units to provide adequate coverage. The system's range is dependent on the range of the ultrasonic sensor that is used. In this system, the HCSR04 ultrasonic sensor was used which has a range between 2cm and 40cm.

### **6.5 Discussion**

In this paper we have attempted to use ultrasonic sensor for implementation of RADAR and got results that exceeds our presumed expectations. The basic systems created for preventing human effort. With some enhancements the system can be used for real time purposes.



## REFERENCES

1. <https://en.wikipedia.org/wiki/Radar>
2. [https://en.wikipedia.org/wiki/Radar\\_\(disambiguation\)](https://en.wikipedia.org/wiki/Radar_(disambiguation))
3. <https://www.woodtv.com/weather/radar-and-satellite>
4. <https://howtomechatronics.com/projects/arduino-radar-project/>
5. <https://create.arduino.cc/projecthub/akshay6766/arduino-radar-withprocessing-097202>
6. <https://www.hackster.io/bharathrao64/ultrasonic-ranging-using-arduino-andprocessing-radar-4091b4>
7. <https://www.electronicshub.org/arduino-radar-project/>
8. <https://www.instructables.com/id/Arduino-Processing-HC-SR04-RADARUsing-Processing-/>
9. [https://www.instructables.com/id/How-to-make-a-simple-ultrasonic-radarsystem-/-/](https://www.instructables.com/id/How-to-make-a-simple-ultrasonic-radarsystem-/)
10. <https://hackaday.io/project/14871-make-a-radar-station-with-arduino-andprocessing>
11. <https://www.slideshare.net/akramalsafadi/arduino-radar-system>
12. [https://www.researchgate.net/publication/308415857\\_A\\_Short\\_Range\\_Radar\\_System](https://www.researchgate.net/publication/308415857_A_Short_Range_Radar_System)



# APPENDIX

## Arduino Program

A minimal Arduino C sketch, as seen by the Arduino IDE programmer, consists of only two functions:

`setup()`: This function is called once when a sketch starts after powerup or reset. It is

used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

`loop()`: After `setup()` has been called, function `loop()` is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

```
// Includes the Servo library
```

```
#include <Servo.h>.
```

```
// Defines Trig and Echo pins of the Ultrasonic Sensor
```

```
const int trigPin = 10; const int echoPin = 11;
```

```
// Variables for the duration and the distance
```

```
long duration; int distance;
```

```
Servo myServo; // Creates a servo object for controlling the servo motor
```

```
void setup() {
```

```
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
```

```
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
    Serial.begin(9600);
```

```
    myServo.attach(12); // Defines on which pin is the servo motor attached
```

```
}
```

```
}
```

```

// Function for calculating the distance measured by the Ultrasonic sensor

int calculateDistance(){

    digitalWrite(trigPin, LOW);

    delayMicroseconds(2);

    // Sets the trigPin on HIGH state for 10 micro seconds

    digitalWrite(trigPin, HIGH);

    delayMicroseconds(10);

    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound
    wave travel time in microseconds

    distance= duration*0.034/2;

    return distance;

}

```

## Processing Program

```

import processing.serial.*; // imports library for serial communication
import java.awt.event.KeyEvent; // imports library for reading the data from
    the serial port import java.io.IOException;
Serial myPort; // defines Object Serial
// defubes variables
String angle="";
String distance="";
String data="";
String noObject;
float pixsDistance;

```

```

int iAngle, iDistance;
int index1=0;
int index2=0;
PFont orcFont;
void setup() {

    size (1200, 700); // ***CHANGE THIS TO OUR SCREEN RESOLUTION
    ***

    smooth();
    myPort = new Serial(this,"COM5", 9600); // starts the serial communication
    myPort.bufferUntil('.'); // reads the data from the serial port up to the char
    acter '.'. So actually it reads this: angle,distance.
}
void draw() {

    fill(98,245,31);
    // simulating motion blur and slow fade of the moving line
    noStroke();
    fill(0,4);
    rect(0, 0, width, height-height*0.065);

    fill(98,245,31); // green color
    // calls the functions for drawing the radar
    drawRadar();
    drawLine();
    drawObject();
    drawText();
}

    translate((width-width*0.5104)+width/2*cos(radians(150)),(height-height*0.0574)wid
th/2*sin(radians(150)));
    rotate(radians(-60));
    text("150°",0,0);
    popMatrix();
}

```