Project Report

on

Design of Smart Dustbin System

Submitted in partial fulfilment of the requirements for the award of the degree

of

Bachelor of Technology

In

Electronics and Communication Engineering

by

Ms. Sani Kumar **(2001125)**

Mr. Sneh Ranjan **(2001047)**

Ms. Abhinav Kr. Thakur **(2001057)**

Under the esteemed Supervision

of

Dr. Suraj

Department of Electronics and Communication Engineering

Indian Institute of Information Technology Bhagalpur

May 2024

भारतीय सूचना प्रौद्योगिकी संस्थान भागलपुर
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHAGALPUR
An Institute of National Importance Under Act of Parliament

DECLARATION

We hereby declare that the work reported in this project on the topic ***"Design of Smart Dustbin System"*** is original and has been carried out by us independently in the **Department of Electronics and Communication Engineering, IIIT Bhagalpur** under the supervision of **Dr. Suraj**, Assistant Professor, IIIT Bhagalpur. We also declare that this work has not formed the basis for the award of any other Degree, Diploma, or similar title of any university or institution.

(Mr. Sani Kumar)	**(Mr. Sneh Ranjan)**	**(Mr. Abhinav Kr. Thakur)**
2001125	**2001047**	**2001057**

भारतीय सूचना प्रौद्योगिकी संस्थान भागलपुर

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHAGALPUR

An Institute of National Importance Under Act of Parliament

CERTIFICATE

This is to certify that the project entitled *"**Design of Smart Dustbin System**"* is carried out by

Mr. Sani Kumar	(2001125)
Mr. Sneh Ranjan	(2001047)
Mr. Abhinav Kr. Thakur	(2001057)

B. Tech. students of IIIT Bhagalpur, under my supervision and guidance. This project has been submitted in partial fulfilment for the award of "***Bachelor of Technology"*** degree in ***Electronics and Communication Engineering*** at ***Indian Institute of Information Technology Bhagalpur***.

No part of this project has been submitted for the award of any previous degree to the best of my knowledge.

(Supervisor)
Dr. Suraj
Assistant Professor
Electronics and Communication Engineering

(Head of Dept.)
Dr. Dhrubajyoti Bhattacharya
Assistant Professor
Electronics and Communication Engineering

Acknowledgement

With great pleasure we express our cordial thanks and indebtedness to our admirable Guide**,** ***Dr. Suraj****, Assistant Professor*, Department of Electronics and Communication Engineering**.** his vast knowledge, expert supervision and enthusiasm continuously challenged and motivated us to achieve our goal. We are eternally grateful to him for allowing us the opportunity to work on this project. His astonishing insight and ability to extract the key elements from a complex problem and incorporate them into a realistic and informative model is a talent we would love to acquire.

We have great pleasure in expressing our sincere gratitude and thanks to the ***Prof. Pradip Kumar Jain,*** *Director*, Indian Institute of Information Technology Bhagalpur for providing us all the relevant facilities that have made the work completed in time. We express our sincere gratitude to ***Dr. Dhrubajyoti Bhhattarcharya***, Assistant Professor and *Head of Department* and all the faculties of the Department for their valuable suggestions and constant encouragement. We would like to thank all the faculty members of the Department of Electronics and Communication Engineering, Indian Institute of Information Technology Bhagalpur, for their direct or indirect support for helping me in completion of this Thesis. I am grateful to all the office staff, Lab assistants, demonstrators and other non-teaching staff of the Electronics and Communication Engineering, Indian Institute of Information Technology Bhagalpur for their supporting attitude toward us.

During the course of this Project Report preparation, we have received lot of support, encouragement, advice and assistance from many people who spent their precious time listening us patiently in the subsequent progressive presentation helping, us to rectify our errors and to this end we are deeply grateful to all of them.

We are also indebted to all authors of research papers and books referred to, which helped us in carrying out this project work.

The present work certainly would not have been possible without the help of our friends, and seniors. With much love, we thank our parents and siblings for their loving care and unconditional support.

(Sani Kumar)	**(Sneh Ranjan)**	**(Abhinav Kr. Thakur)**
2001125	**2001047**	**2001057**

Abstract

The concept of a smart dustbin using Arduino Uno, an ultrasonic sensor, a servo motor, LCD 16x2, and a buzzer aims to provide an efficient and automated waste management solution. Traditional waste disposal methods often lack convenience and are prone to overflowing or littering. To address these challenges, the proposed smart dustbin utilizes advanced technologies to automate waste collection and improve overall cleanliness. The system employs an Arduino Uno microcontroller board as the central control unit, which processes information from an ultrasonic sensor to detect object from the dustbin.

The ultrasonic sensor accurately measures the distance between the object and the dustbin. This data is then analyzed by the Arduino Uno, which triggers the servo motor for lid movement. When the object is detected, the Arduino Uno commands the servo motor to open the lid, allowing users to dispose of their waste conveniently.

The implementation of a smart dustbin using Arduino Uno, an ultrasonic sensor, and a servo motor offers several benefits. It optimizes waste management processes by automating waste collection and improving efficiency. Users are provided with a hassle-free experience, reducing the chances of overflowing bins and littered surroundings. Moreover, the system can be expanded to incorporate more advanced features, such as waste segregation or data analytics for better waste management decision-making. Overall, the smart dustbin system demonstrates the potential of integrating technologies into everyday waste management, resulting in cleaner environments, reduced manual efforts, and improved efficiency in waste disposal processes.

Table of Contents

Table of Figures

List of Tables

CHAPTER 1

1. Introduction

1.1 Overview

In our busy lives, we often encounter the inconvenience of handling dustbins manually. But what if there was a way to make it efficient and more hygienic? This report introduces an exciting solution: the Automatic Dustbin Opener. By using Arduino Uno, an ultrasonic sensor, a servo motor, LCD 16x2, and a buzzer, this project aims to create a hands-free system for opening and closing dustbins with a display wastage level on LCD [1].

The Smart Dustbin System tackles common issues like touching dirty bins, difficulty in opening them, and the effort required. It works by using an ultrasonic sensor to detect when someone or something is nearby, and a servo motor to move the lid automatically. This means you can open the dustbin without touching it, making the process cleaner and more convenient.

This report will explain how to design, build, and program the Smart Dustbin System. We'll look at the materials needed, how to put them together, and how to write code for the Arduino Uno. Step-by-step instructions are explained and examined focuses on calibrating the sensor, controlling the motor, and creating a user-friendly interface.

Furthermore, we have explored future possibilities for the Smart Dustbin System, such as integrating it with smart home systems or adding features for sorting different types of waste. The aim is to inspire others to embrace automation and make everyday tasks simpler and more efficient [2].

We can say that the Smart Dustbin System is a practical and innovative solution that improves convenience and cleanliness in handling dustbins. By using Arduino Uno, an ultrasonic sensor, and a servo motor, this project demonstrates how automation can make our lives easier. Through this report, we hope to encourage individuals and industries to embrace automation and enjoy the benefits it brings to our daily routines [3].

1.2 Literature Survey

1.2.1 Introduction to Embedded System

An embedded system is a microprocessor-based system which is combination of computer hardware and software designed for a specific function. Embedded systems contain processing cores that are either microcontrollers or digital signal processors. Microcontrollers are generally known as "chip", which may itself be packaged with other microcontrollers in a hybrid system of Application Specific Integrated Circuit (ASIC). Embedded systems may also function within a larger system. The systems can be programmable or have a fixed functionality [4].

Here, we are using Arduino IDE for code execution, for sensing we have used ultrasonic sensor HC-SR04 which will sense the object and will send code to Arduino for generating command to open the lid, and used an LCD 16x2 to display the wastage level of the dustbin and wait for few moments. It will bring drastic changes in terms of cleanliness with the help of technology. Everything is getting smart with technology for betterment of human being. So, this helps in maintaining the environment clean with the help of technology. It is sensor best dustbin so it would be easy to access/use for any age/group person. Our aim is also to make it cost effective so that many members of people can get the benefit from this. And it would be usable to anyone and helpful for them [5].

1.2.2 Block Diagram

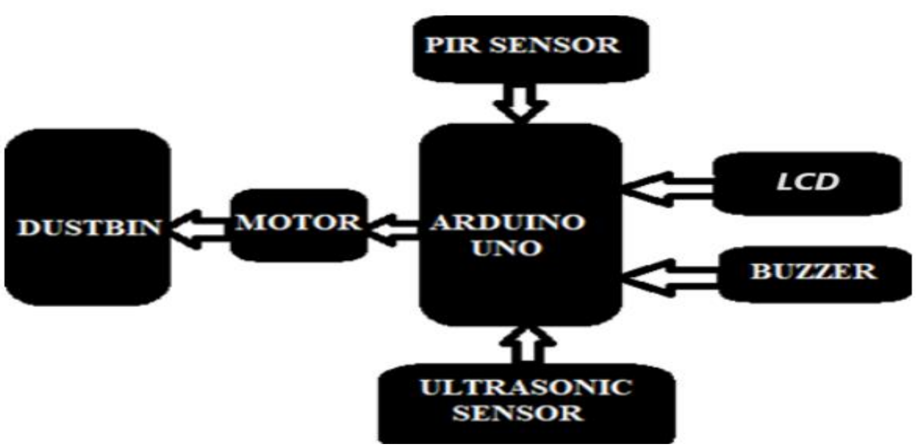

Figure 1:Block Diagram of System Architecture

Figure 1 depicts the block diagram of the system architecture of the concerned work. The brief details of the functional components of the system are mentioned below:

I. Power Supply: The primary function of a power supply is to convert one form of electrical energy into another and, as a result power supplies.
II. Microcontroller: The microcontroller is used to manipulate the serial operation base the program present in the output is taken from one of the four ports.
III. Ultrasonic Sensor: It uses SONAR to detect distance of object. Crystal oscillator is used in ultrasonic sensor to produce oscillated pulses which is given to the microcontroller.
IV. Servo Motor: It is used to opening and closing purpose of the dustbin's lid.
V. LCD: It is used to display the wastage level of dustbin material.

1.3 Motivation

- **Convenience:** Automatic dustbin opener adds convenience to waste disposal, allowing contactless operation.
- **Hygiene:** It promotes hygiene by eliminating the need to touch the dustbin lid, reducing the spread of germs and contamination.
- **Efficiency**: Users can quickly and efficiently dispose of waste without the hassle of manually opening and closing the lid.
- **Environmental Friendliness**: Encourages proper waste disposal, reducing littering and promoting environmentally friendly practices.
- **Safety:** Minimizes the risk of injury or exposure by coming in contact to hazardous waste materials by automating the process of lid opening and closing.
- **Accessibility:** It can be used by any kind of person even by a kid, an old person or even by a physically challenged person, making waste disposal easier and more inclusive.
- **Modern and Innovative:** Showcases a modern and innovative approach to waste management, adding sophistication to the living or working space.

1.4 Objective

The objective of the project is as follows:

1. Add the libraries of the hardware components and pick the devices in the software.

2. Make the circuit of Automatic Dustbin Opener using the picked hardware components in the software.
3. Add the program file to the components and run the simulation to obtain the result.
4. Take data signal from ultrasonic sensor and send.
5. Observing duty cycle of PWM signal waveform of servo motor for different angles.
6. We observed to found suitable measurement of distance between person and dustbin to opening and closing of lid.

1.5 Project Layout

The layout of project report is as follows:

Chapter 1: Introduction. This chapter offers a brief introduction of the Smart Dustbin System and outline the objective and workflow of the project work.

Chapter 2: Requirement Specification. This chapter deals with specifications of hardware components and software used in project.

Chapter 3: Design, Simulations and Implementation. This chapter deals with designing the project and simulating the Arduino Uno with LCD Display. This chapter deals with the connection of hardware components.

Chapter 4: Procedures and Result. This chapter focused on the Simulation process of the project. All the steps of simulation explained in this chapter. The simulation results are also mentioned in this chapter.

Chapter 5: Conclusion and Future Scope. This chapter concluded the project and further focuses on the future scope of the concerned project.

CHAPTER 2

2. Hardware and Software Specifications

2.1 System Specifications

As embedded system is combination of both hardware and software to perform specific function, we used hardware and software both. In this project, we have designed a simple system called Smart Dustbin and have used following components:

a) Arduino UNO Board
b) Ultrasonic Sensor
c) Servo Motor
d) LCD 16x2 Display
e) Buzzer
f) Battery
g) Proteus 8 Professional
h) Arduino IDE

To perform function on hardware we need a software platform to simulate and generate commands for the system. So, we used the following software for Arduino UNO.

2.2 Arduino UNO Board

Arduino Uno is an open-source microcontroller board based on the processor ATmega328P. There are 14 digital I/O pins, 6 analog inputs, a USB connection, a power jack, an ICSP header, and a reset button. It contains all the necessary modules needed to support the microcontroller [6]. Just plug it into a computer with a USB cable or power it with an adapter to get started. One can

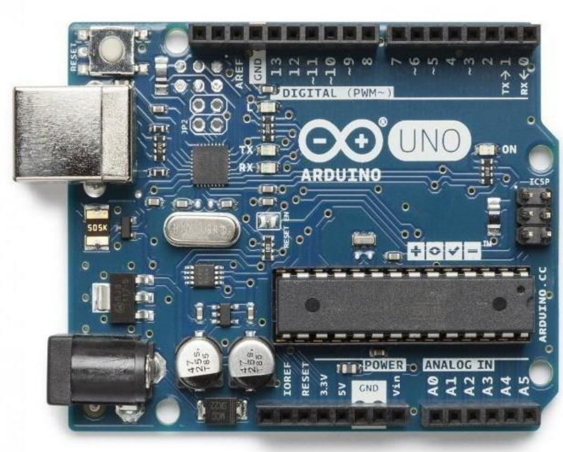

Figure 2:Arduino UNO

experiment with the Arduino without worrying too much about as it user-friendly, easily available and is very economical [7], [8]. Figure 2 shows the Arduino UNO board used in this concerned project work.

2.3 Hardware Structure of Arduino UNO

ATMEGA328 Microcontroller:

The Atmel AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU),

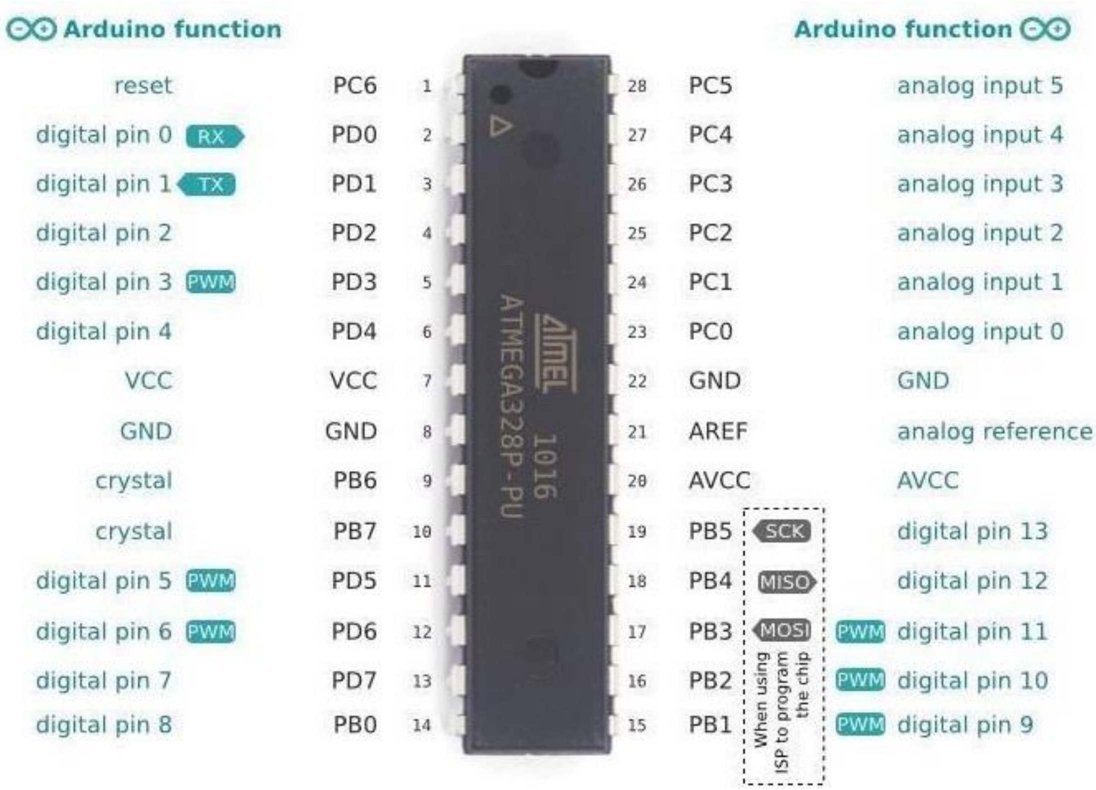

Figure 3:ATMEGA328P Micro-controller

allowing two independent registers to be accessed in a single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The ATmega328/P provides the following features: 32Kbytes of In-System Programmable Flash with Read-While Write capabilities, 1Kbytes EEPROM, 2Kbytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 1 serial programmable USARTs , 1 byte oriented 2-wire Serial Interface (I2C), a 6- channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages) , a programmable Watchdog Timer with internal Oscillator, an SPI serial port, and six software selectable power saving modes [3] [9]. Figure 3 shows the pin descriptions of the ATMEGA328P Micro-controller.

2.4 Features of ATMEGA328

The important features of the concerned microcontroller have been listed below:

1. 28-pin AVR Microcontroller
2. Flash Program Memory: 32 Kbytes
3. EEPROM Data Memory: 1 Kbytes
4. SRAM Data Memory: 2 Kbytes
5. I/O Pins: 23
6. Timers: Two 8-bit / One 16-bit
7. A/D Converter: 10-bit Six Channel
8. PWM: Six Channels
9. RTC: Yes, and with Separate Oscillator
10. MSSP: SPI and I2C Master and Slave Support
11. USART: Yes
12. External Oscillator: up to 20MHz

The pin categorization have been briefly detailed below and relevant details as per the categorization are depicted in the Table 1.

- **(i) Digital Pins:** There are 14 digital pins on Arduino Uno which can be connected to components like LED, LCD, etc.
- **(ii) Analog Pins:** There are 6 analog pins on the Uno. These pins are generally used to connect sensors because all the sensors generally have analog values. Most of the input components are connected here.
- **(iii)Power Supply:** The power supply pins are IOREF, GND, 3.3V, 5V, Vin are used to connecting sensors because all the sensors generally have analog values. Most of the input components are connected here.
- **(iv)Power Jack:** Uno board can be powered both by external supply and via USB cable.
- **(v) USB Port:** This port function is to program the board or to upload the program. The program can be uploaded to the board with the help of Arduino IDE and USB cable.

(vi) Reset Button: This is used to restart the uploaded program [7].

Table 1: Pin Description of ARDUINO UNO

Pin Category	Pin Name	Details
Power	Vin 5V 3.3V GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pin.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5 V
Input / Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial Communication	0(Rx), 1 (Tx)	Used to receive and transmit TTL serial data.
External Interrupts	10 (SS), 11 (MOSI),	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.

AREF	AREF	To provide reference voltage for input

2.5 Proteus 8 Professional

Proteus is a Virtual System Modelling and circuit simulation application. The suite combines mixed mode SPICE circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller-based designs. The schematic capture window of the Proteus 8 Professional is shown in the Figure 4.

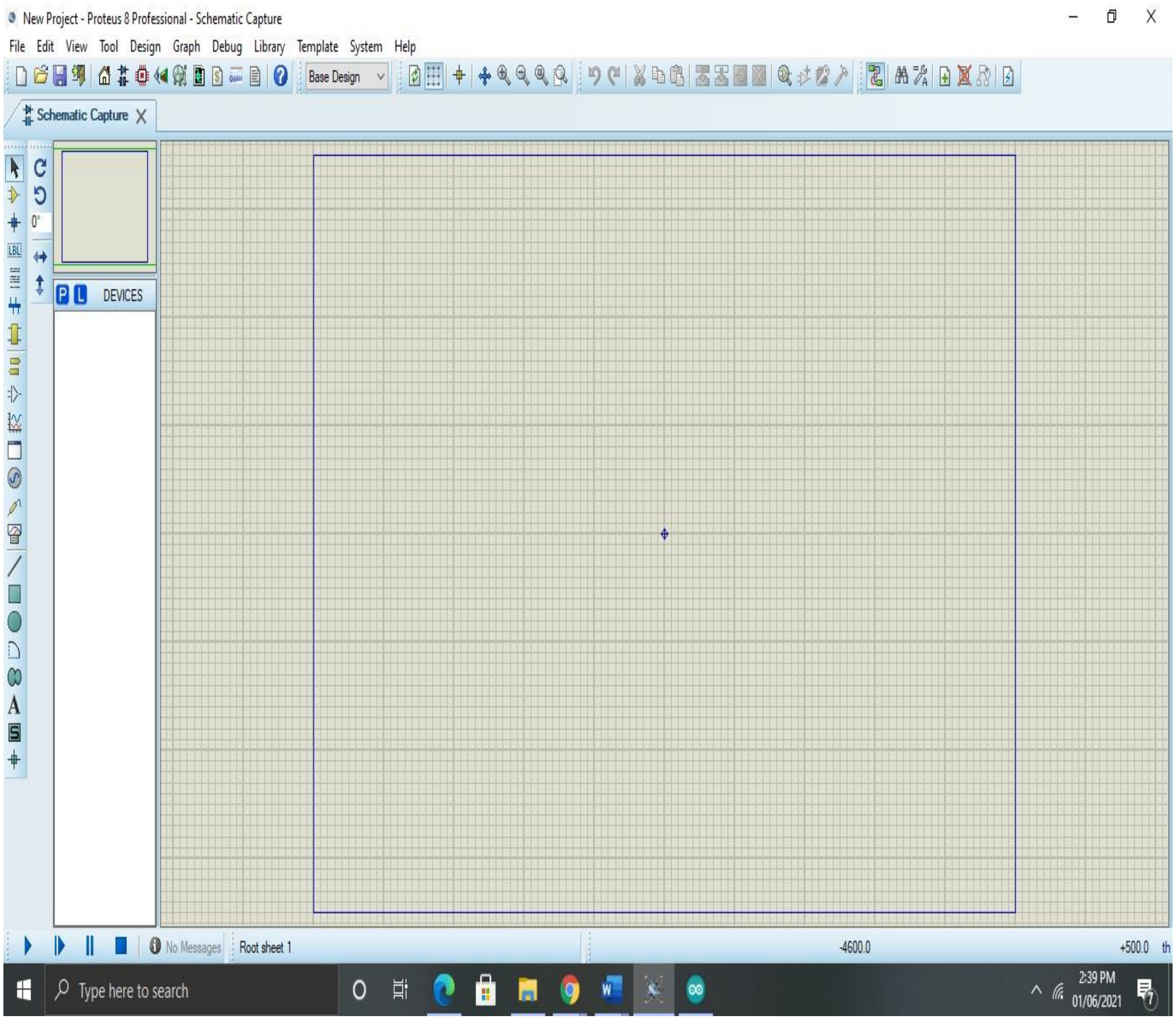

Figure 4:IDE of Proteus 8 Professional

2.6 Arduino IDE 2.1.0

The Arduino IDE (shown in Figure 5) is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment [9], [10].

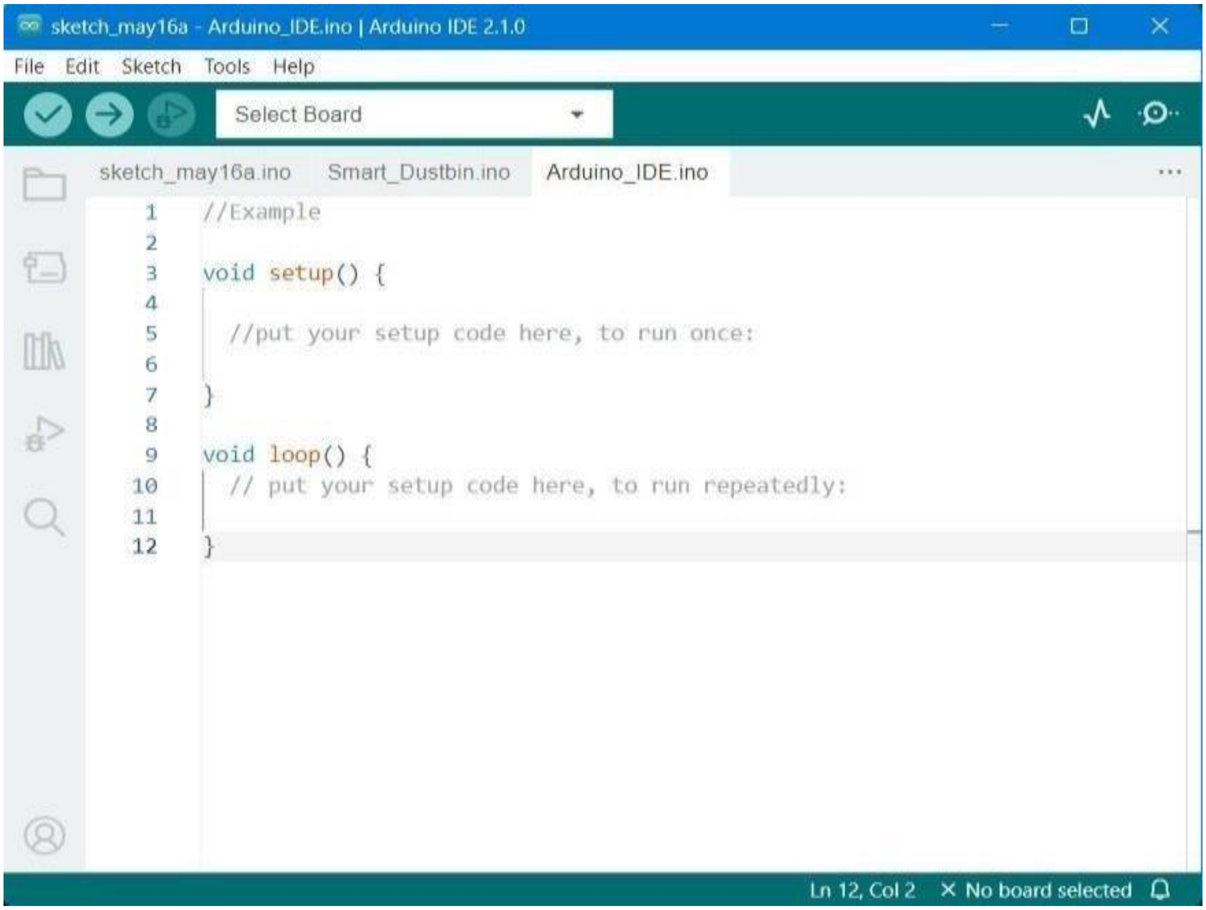

Figure 5:Arduino IDE 2.1.0

2.7 Ultrasonic Sensor

An ultrasonic sensor is a device that uses high-frequency sound waves to measure distance or detect objects. It consists of a transmitter that emits ultrasonic waves and a receiver that sense the reflected waves. The sensor measures the time it takes for the sound waves to bounce back from an object and calculates the distance based on the speed of sound [11]. In this project we have used **HC-SR04** to detect the object.

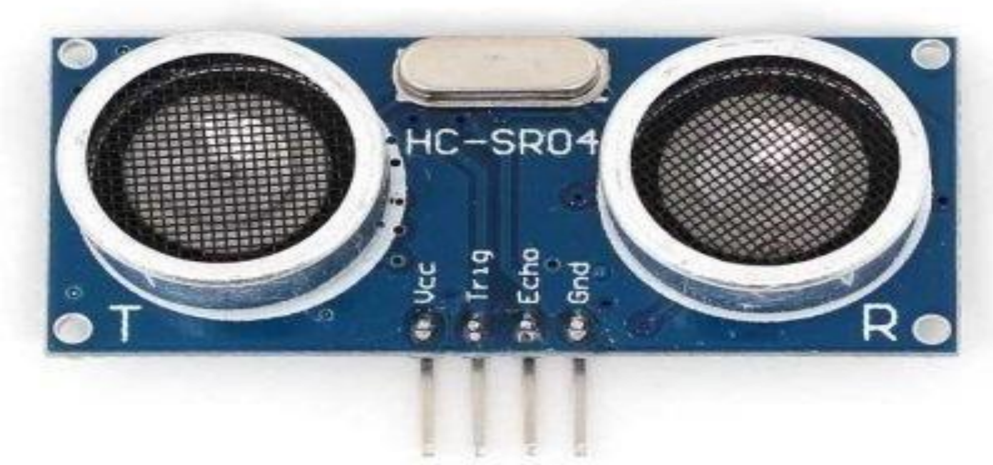

Figure 6:Ultrasonic Sensor HC-SR04

HC-SR04 is an ultrasonic distance sensor used for measuring the distance at which an object is located. The principle used by this sensor is called SONAR. It is perfect for small robotics projects such as obstacle avoiding robot, distance measuring device etc. It has two parts; one emits the ultrasound sonar to measure the distance to an object. The other part is the receiver which listens for the echo. As soon as the ultrasound hits the object it bounces back and is detected by the receiver. The time taken for the wave to come back decides the distance of the object being measured [12]. Figure 6 shows Ultrasonic sensor HC-SR04.

2.8 Pin Configuration of HC-SR04 Ultrasonic Sensor

V_{CC}: Supplies power to the HC-SR04 ultrasonic sensor. We can connect it to the 5V output from Arduino.

Trig: This pin is used to trigger ultrasonic sound pulses. By setting this pin to HIGH for 10µs, the sensor initiates an ultrasonic burst. It emits ultrasonic sound wave.

Echo: This pin goes high when the ultrasonic burst is transmitted and remains high until the sensor receives an echo, after which it goes low. By measuring the time the Echo pin stays high, the distance can be calculated.

GND: This is the ground pin. We connect it to the ground of the Arduino.

2.9 Specifications of HC-SR04

- This is the HC-SR04 ultrasonic distance sensor. This economical sensor provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm.
- There are only four pins that you need to worry about on the HC-SR04: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground)
- Length: 4.5 cm (1 ¾ in)
- Width: 2.0 cm (¾ in)
- Height: 1.4 cm (½ in)
- Working Current: 15mA
- Working Frequency: 40Hz
- Power Supply: DC
- Resolution: 0.3 cm
- Dimension: 45mm x 20mm x 15mm
- Operating voltages: 3V or 5V (trigger), 5V all other I/O ports
- Operating range: 2 cm to 400 cm (1 in – 13 ft)
- Claimed precision: 0.3cm, more realistically: 1cm
- Measuring angle: 15 degrees
- Quiescent Current: <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance: 2cm – 400 cm/1″ 13ft
- Measuring Angle: 30 degrees
- Trigger Input Pulse width: 10uS

2.10 Working Principle of HC-SR04 Ultrasonic Sensor

Figure 7 explains the working of an ultrasonic sensor. An ultrasonic sensor operates by emitting high-frequency sound waves that are beyond human hearing range. These waves travel through the air and when they encounter an object, they bounce back to the sensor. By measuring the time, it takes for the sound waves to return, the sensor can determine the distance to the object. This principle is used in applications such as parking assist systems, object detection in robotics, and distance measurement. Ultrasonic sensors are effective because sound waves can travel through various environments and are not affected by color, transparency, or surface texture of objects [12], [13].

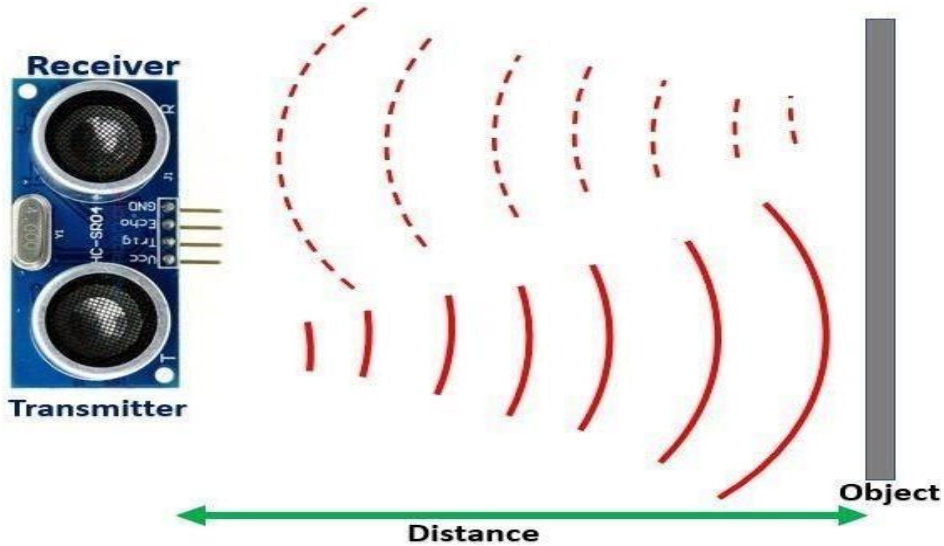

Figure 7:Working of Ultrasonic Sensor

2.11 Servo Motor

A servomotor (or servo motor) is a device that allows for precise control of angular or linear position, velocity, and acceleration. Servo motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU [14]. Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 180 degrees from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically, servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears [15]. There are lots of servo motors available in the market and each one has its own specialty and applications. Most of the Servo

Figure 8:Servo Motor

motors operates from 4.8V to 6.5V, the higher the voltage higher the torque we can achieve, but most commonly they are operated at +5V. Almost all servo motors can rotate only from 0° to 180° due to their gear arrangement [16]. Here, we preferred for a 0° to 180° SG90 servo motor (shown in the Figure 8) as per our requirement.

2.12 SG90 Servo Motor

The micro servo 9G is a light, good quality and very fast servo motor. This servo is designed to work with most radio control systems. It is perfect for small robotics projects. The SG90 mini servo with accessories is perfect for remote-controlled helicopters, planes, cars, boats and trucks [17].

Specifications

- Product: SG90 Servo
- Torque: 2.0kg/cm (4.8V), 2.2kg/cm(6V)
- Speed: 0.09s/60° (4.8V), 0.08s/60°(6V)
- Rotate angle: 180°
- Operating voltage: 4.8 ~ 6V
- Gear: plastic
- Dead band: 7us
- Weight: 10.5g
- Dimension: 22.8mm × 12.2mm × 28.5mm

Basic Information Modulation

- Analog Torque: 4.8V: 25.0 oz-in (1.80 kg-cm)
- Speed: 4.8V: 0.10 sec/60°
- Weight: 0.32 oz (9.0 g)
- Dimensions: Length: 0.91 in (23.1 mm)
- Width: 0.48 in (12.2 mm)
- Height:1.14 in (29.0 mm)
- Motor Type: 3-pole Gear Type: Plastic
- Rotation/Support: Bushing

Additional Specifications

- Rotational Range: 180°
- Pulse Cycle: ca. 20 ms
- Pulse Width: 500-2400 µs

2.13 Working Principle of Servo Motor

Most of the servo motors operates from 4.8V to 6.5V, the higher the voltage higher the torque we can achieve, but most commonly they are operated at +5V. Almost all servo motors can rotate only from 0° to 180° due to their gear arrangement so make sure you project can live with the half circle. The gears in the motors are easily subjected to wear and tear, so if our application requires stronger and long running motors, we could go with metal gears or just stick with normal plastic gear.

The SG90 servo motor operates based on the principle of pulse width modulation (PWM). It receives electrical signals, typically in the form of a square wave, with a specific pulse duration known as the duty cycle. The duty cycle represents the percentage of time the signal is "on" compared to the total cycle time. The motor's position is determined by the duty cycle: a shorter pulse duration corresponds to one position, while a longer pulse duration corresponds to another. By varying the duty cycle, the servo motor can be controlled to rotate to different angles within its range [18], [19].

2.14 LCD

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

1. The declining prices of LCDs and ease of programming for characters and graphics.
2. The ability to display numbers, characters and graphics.

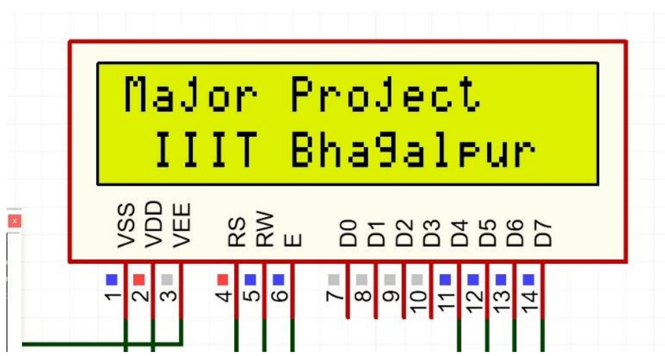

Figure 9:LCD 16x2

16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD which has been depicted in the Figure 9.

2.15 LCD 16×2 Pin Diagram

Figure 10 shows the pin diagram of the 16x2 LCD used in this project work. The figure also outlines the functional details of the concerned LCD. The 16×2 LCD pinout is detailed below.

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode.
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

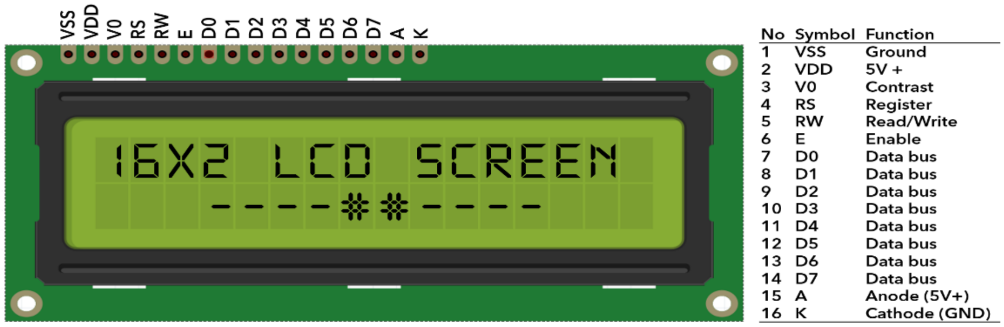

Figure 10:LCD 16×2 Pin Diagram

2.16 Features of LCD16x2

The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16-characters.
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8 pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit
- These are obtainable in Blue & Green Backlight
- It displays a few custom generated characters

2.17 Battery

We have used 9V power supply i.e., Hi- Waote 9V battery. Hi-Waote 9V Battery is the most commonly used and portable 9V battery. It is non-rechargeable and is a high capacity and low-cost solution for many electronic devices. It is based on Zinc Carbon Chemistry and can be used easily replaced if discharged just like any standard AA and AAA batteries. The battery can be used to power LEDs, Toys, Flashlight and Torch, electronic equipment like multi-meter, wall clocks, or other devices with a 9V system [20]. A battery snap connector is generally used to connect it with a breadboard. Figure 11 shows the Hi-Watt 9V battery used in this work.

Figure 11:Hi-Watt 9V Battery

Specifications

- Nominal Voltage(V): 9V
- Battery Type: Zinc Carbon battery
- Dimension: 26.5mm x 48.5mm x 17.5mm
- System: Zinc Carbon
- Discharge Resistance (Ohms): 620
- Cut-off Voltage(V): 5.4
- Discharge Tie: 270Hm, 9 Hrs
- Jacket: Metal
- Operating Temperature Range (deg. C): -20 to +85

CHAPTER 3

3. Design, Simulations and Implementation

3.1 Design and Simulation

We used Arduino IDE software to code the Arduino and Proteus software to simulate the whole project. The whole simulation process steps are given below:

3.2 Arduino IDE 1.8.13

- Step 1: Install Arduino IDE (LINK: https://www.arduino.cc/en/donate/)
- Step 2: Open Arduino IDE software and create a new file using File → New.
- Step 3: To avoid any error while uploading program to the board, we must select the correct Arduino board name using Go to Tools → Board and select board.
- Step 4: Now write your code (shown in Figure 12) to program the Arduino board.
- Step 5: After coding, compile your code using the  button
- Step 6: After compiling is done. Copy the hex file (as depicted in Figure 13) location and save it at suitable folder.

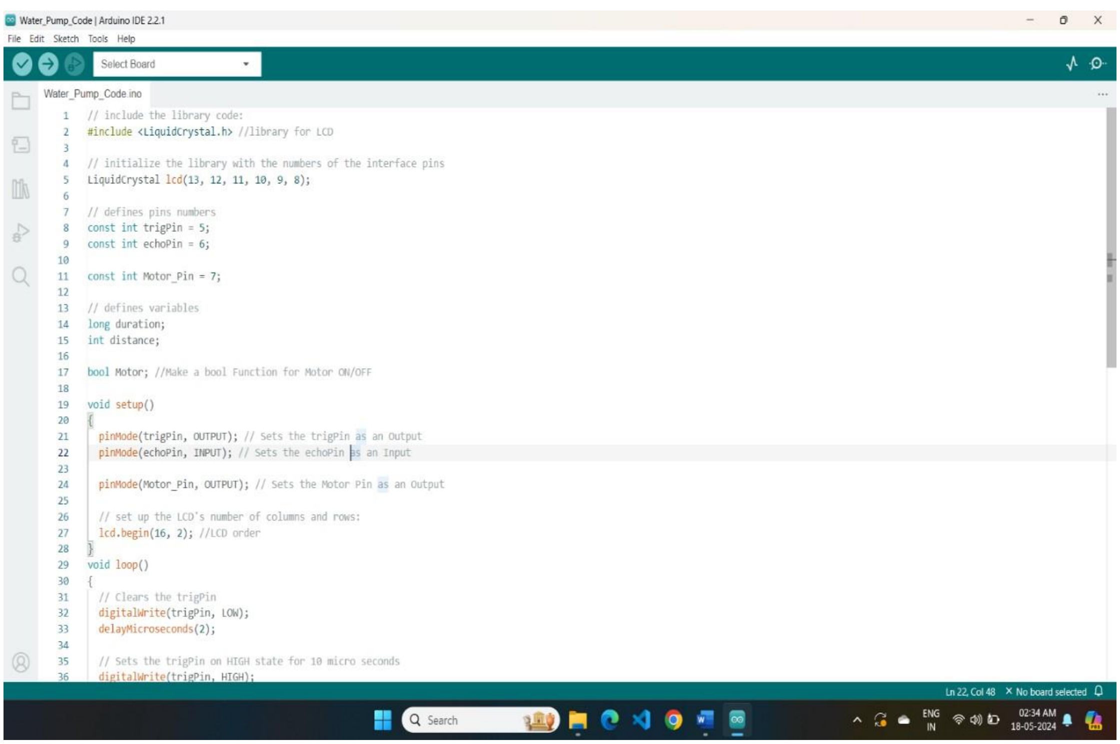

Figure 12:program code

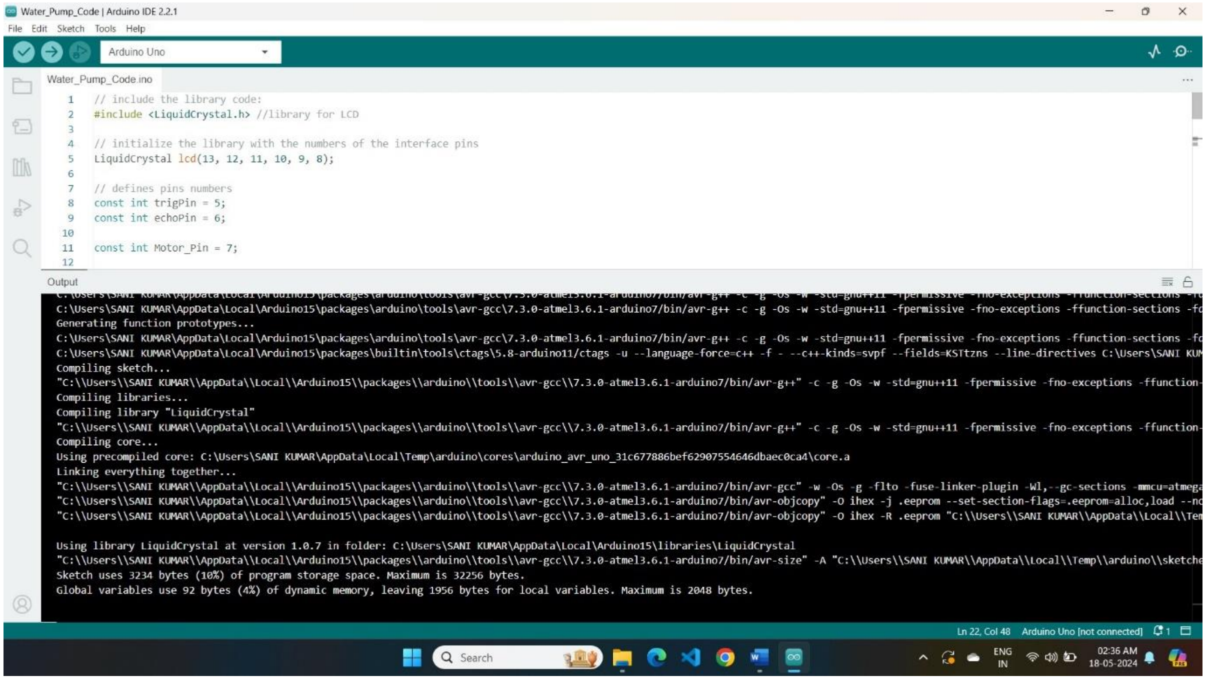

Figure 13:Copying the location of HEX file

3.3 Proteus 8 Professional

The Proteus simulation steps are given below:

Step 1: Install Proteus software (Follow this link to install and learn basics of Proteus https://www.youtube.com/playlist?list=PLNc6TkKhCYcyHrMFtsREjhq6ptbyuZQ8m)

Step 2: Open proteus software.

Step 3: Click on Schematic Capture (shown in Figure 14) to open a new project.

Figure 14:Click on the icon to open Schematic

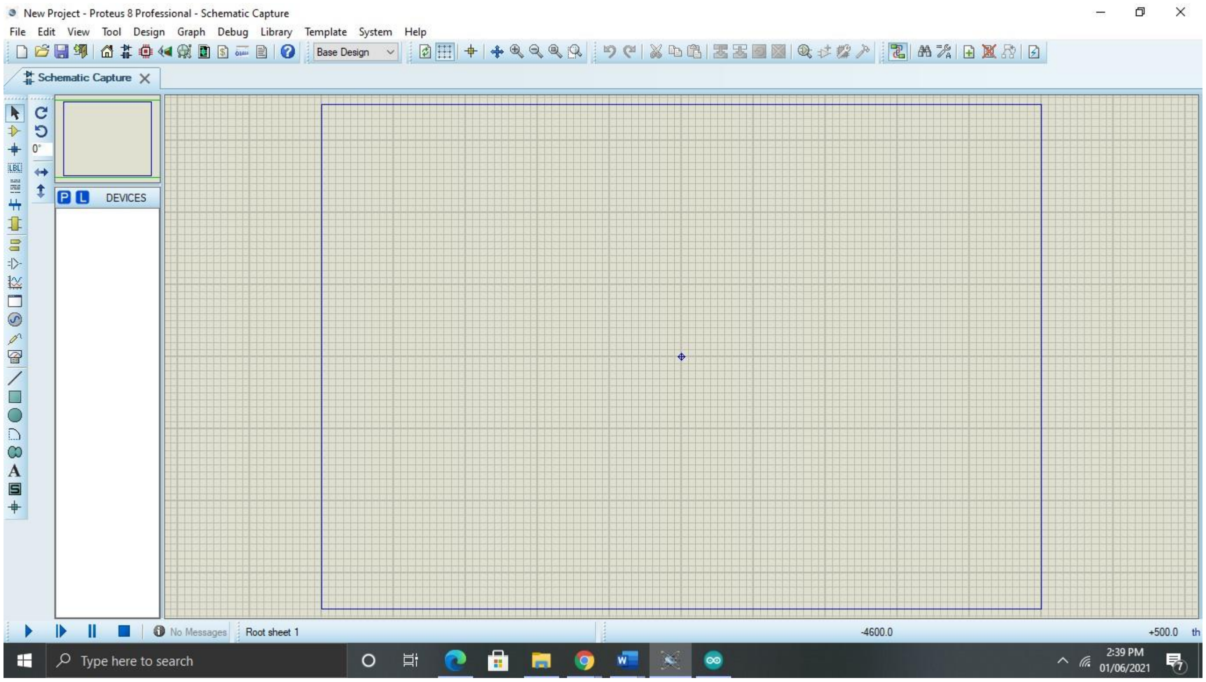

Figure 15:New Schematic Capture window

Step 4: After opening New Project, add devices by clicking Pick Devices icon in DEVICES (depicted in Figure 15 and Figure 16).

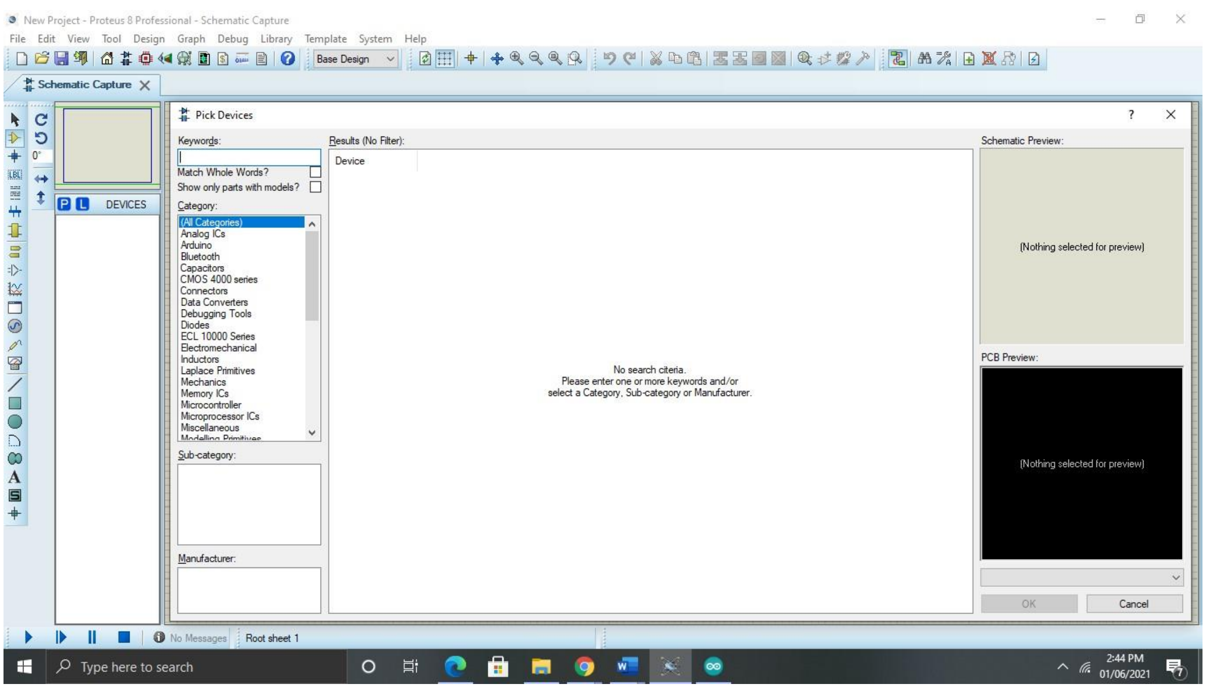

Figure 16:Pick devices using this icon

Step 5: Search for Arduino, LM016L, Ultra Sonic Sensor, HC-SR04 and double click to pick the device (shown in Figure 17 and Figure 18).

Arduino:https://www.theengineeringprojects.com/2015/12/arduino-library-proteus-simulation.html

Ultra-Sonic Sensor : https://www.theengineeringprojects.com/2015/12/ ultrasonic-sensor-library-proteus.html

Adding Libraries to proteus:

https://www.youtube.com/playlist?list=PLNc6TkKhCYcyHrMFtsREjhq6ptbyuZQ8m

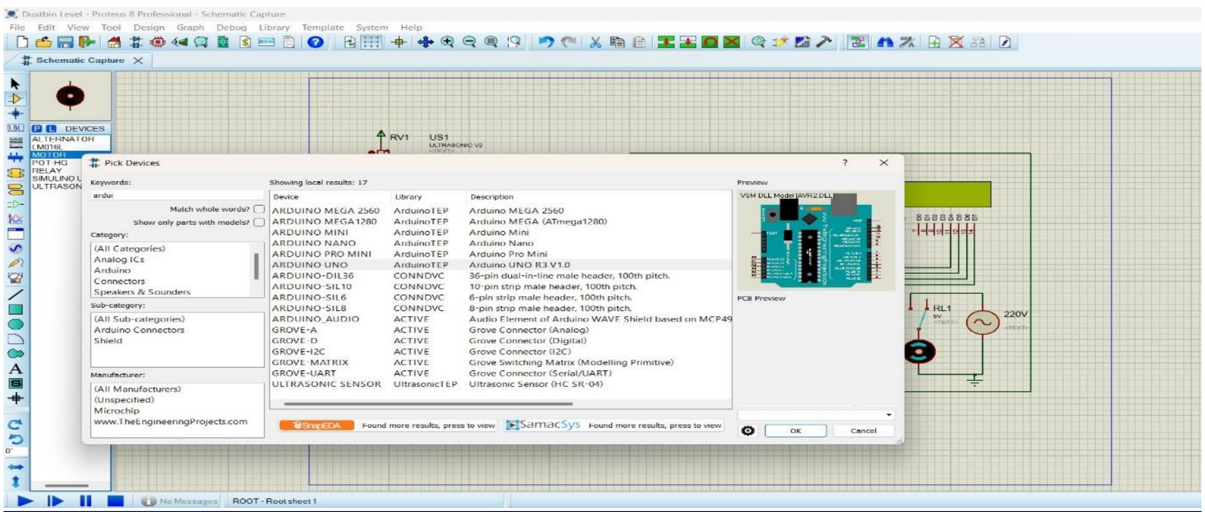

Figure 17:Picking the devices

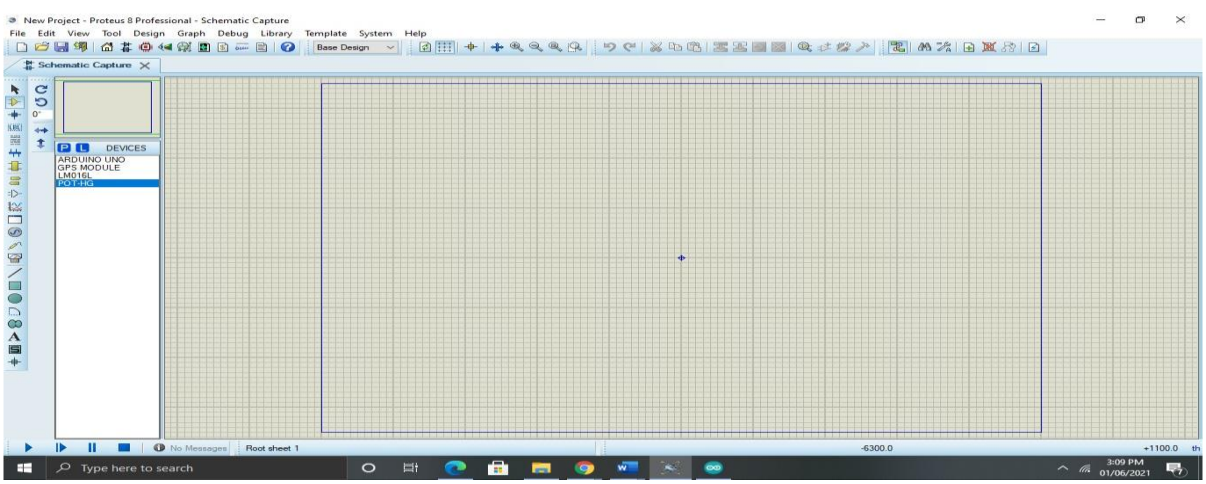

Figure 18:Picked devices are shown here

Step 6: After picking or adding devices, now select the devices and put them on the schematic layout (depicted in Figure 19).

Select the Device from the DEVICES and just click on the schematic layout and then click again to put it on the layout.

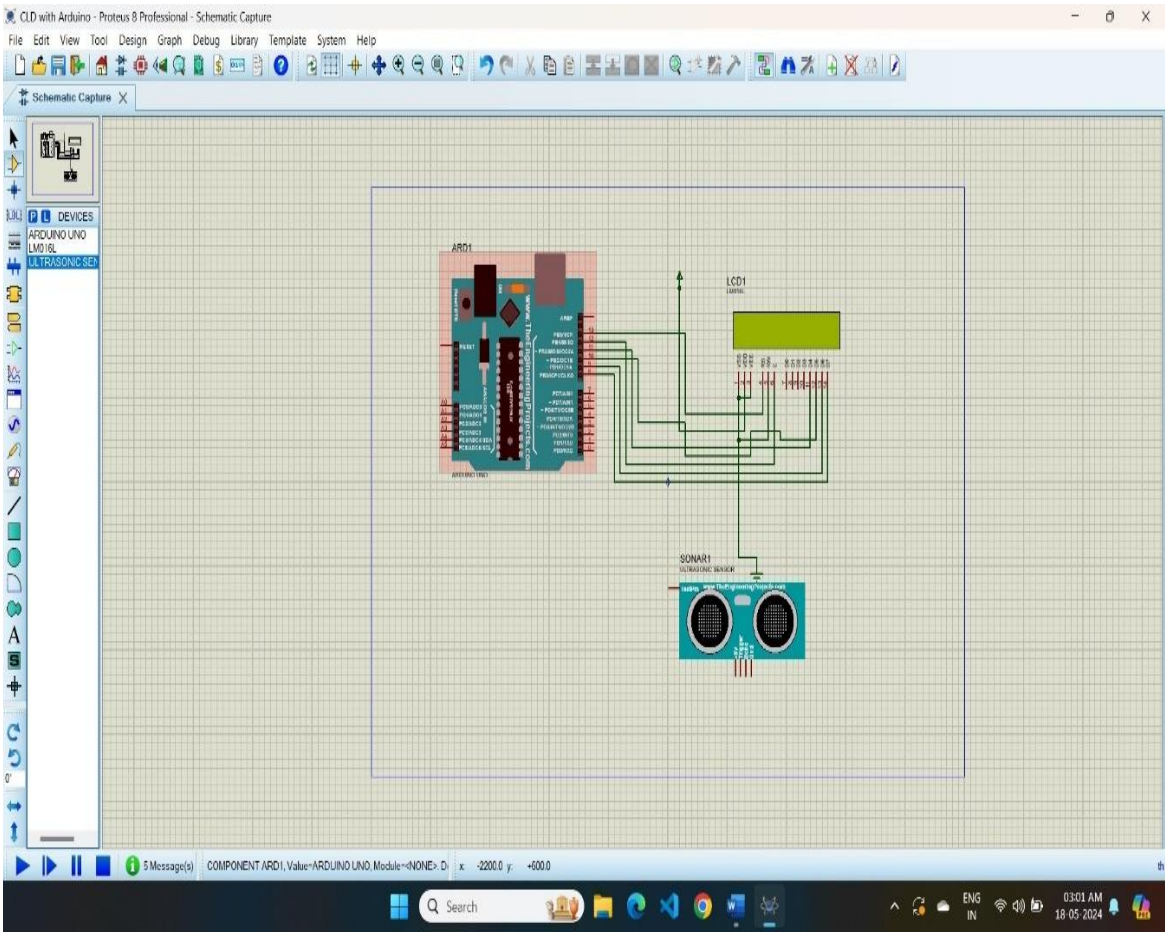

Figure 19:Adding devices to the Schematic Capture

Step 7: Add virtual terminal, Ground and Power using the following:

Right click on the layout → Place → Terminal → GROUND.

Right click on the layout → Place → Generator → DC.

Right click on the layout → Place → Virtual Instrument → VIRTUAL TERMINAL (shown in Figure 20).

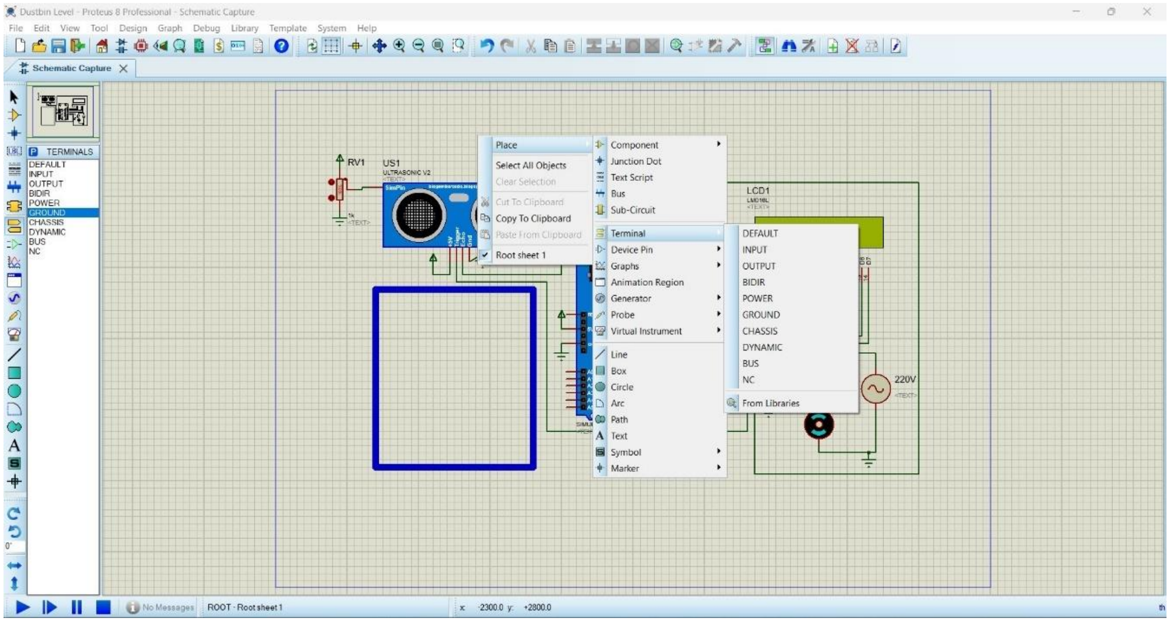

Figure 20:Adding the GROUND, DC and Virtual Terminal

Step 8: Complete the Connections (as depicted in Figure 21) and save the project using File → Save Project.

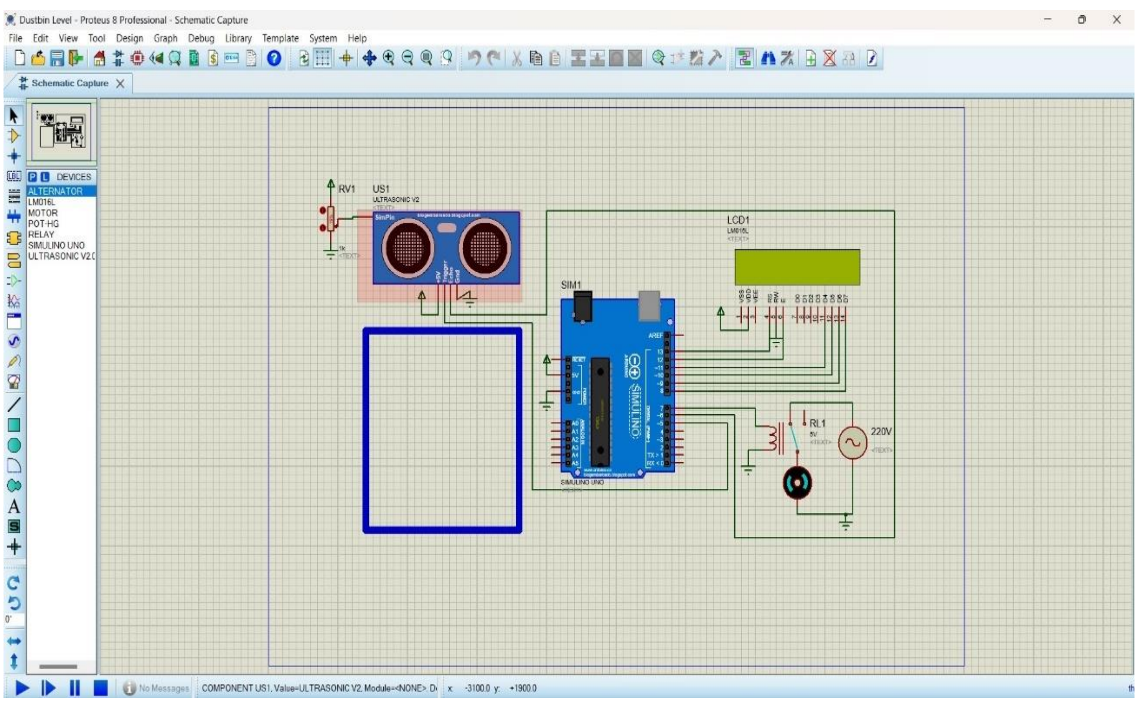

Figure 21:Complete connection of the project

Step 9: Now double click on the Arduino and paste the saved hex file location in the Program file option (as shown in Figure 22).

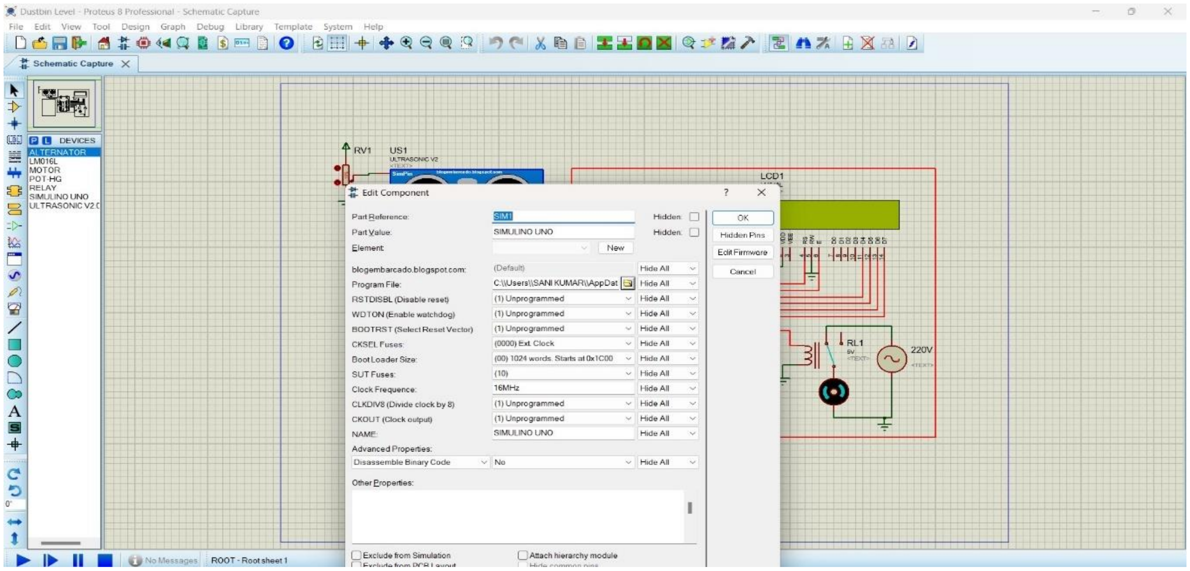

Figure 22:Adding program (HEX) file to Arduino

Step 10: Now click on Run the Simulation Button present on the bottom left side as marked and shown in Figure 23.

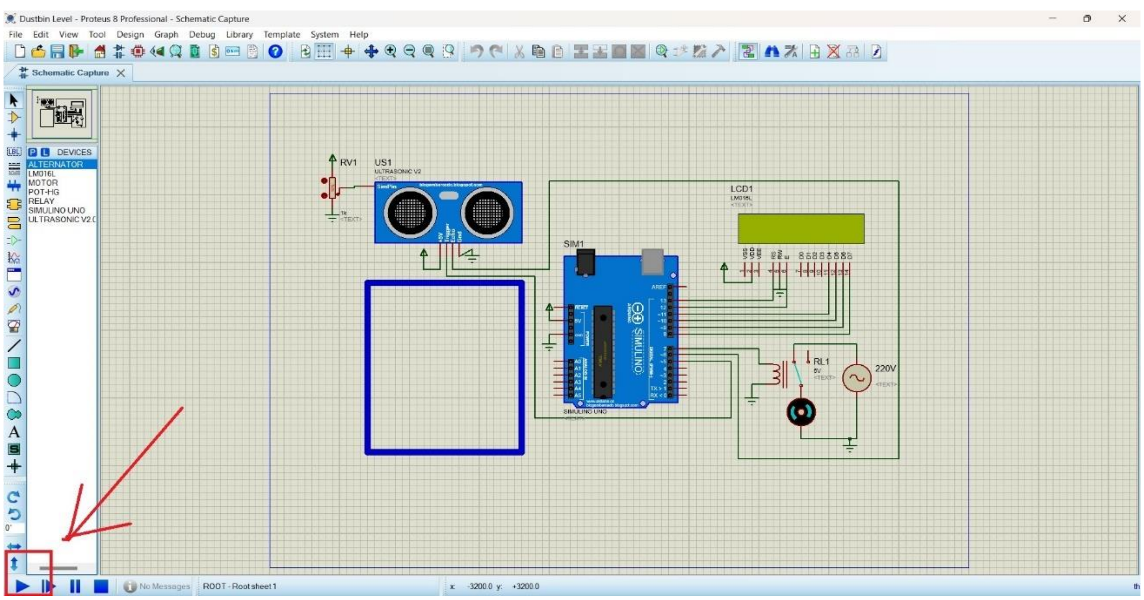

Figure 23:Run the project

3.4 Result of Simulation

The designed circuit is simulated for the different level of filled dustbin to test the functionality of the designed smart dustbin. The simulation results are shown below in the Figure 24, Figure 25, Figure 26 and Figure 27 for empty dustbin, 25% filled, half-filled and full filled dustbin respectively.

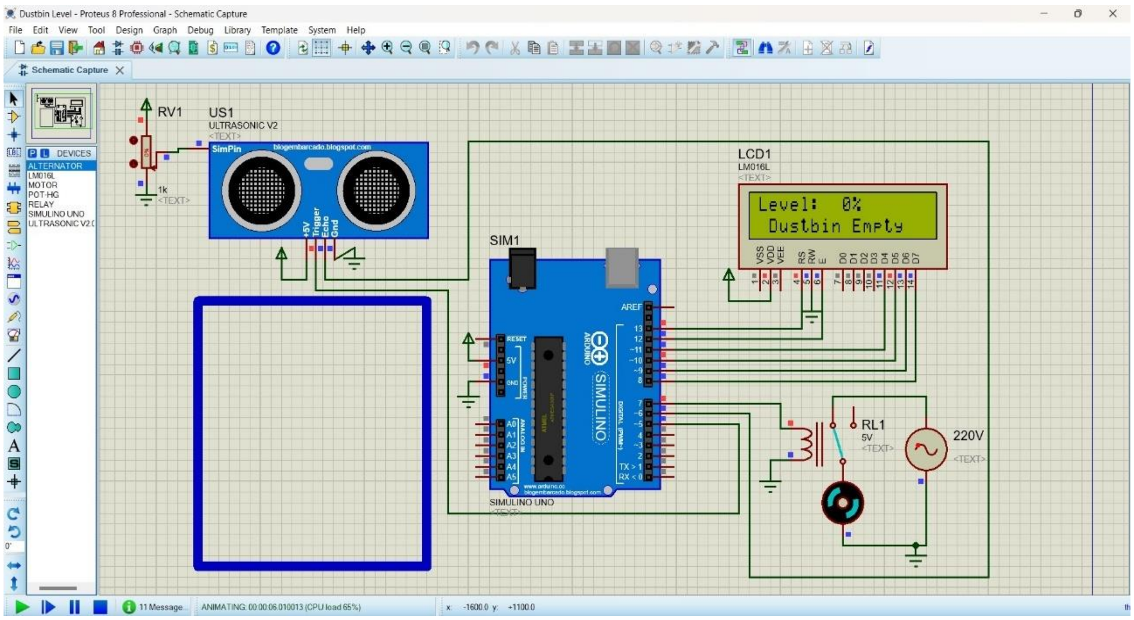

Figure 24: Simulation result with empty dustbin

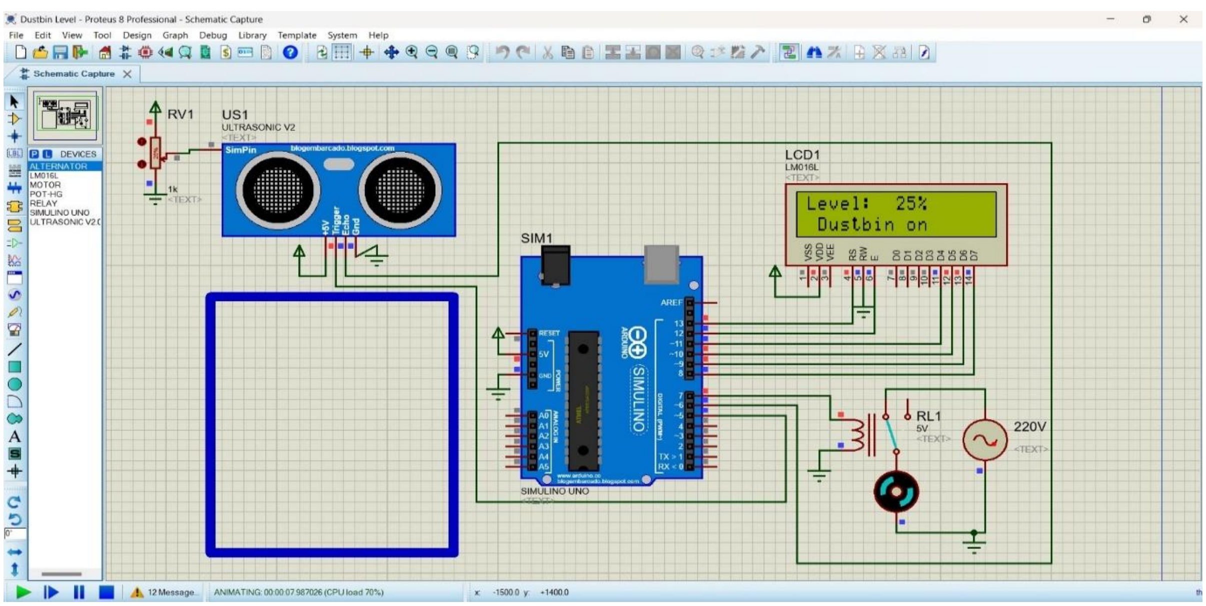

Figure 25: Simulation result with 25% garbage as per the capacity of the dustbin

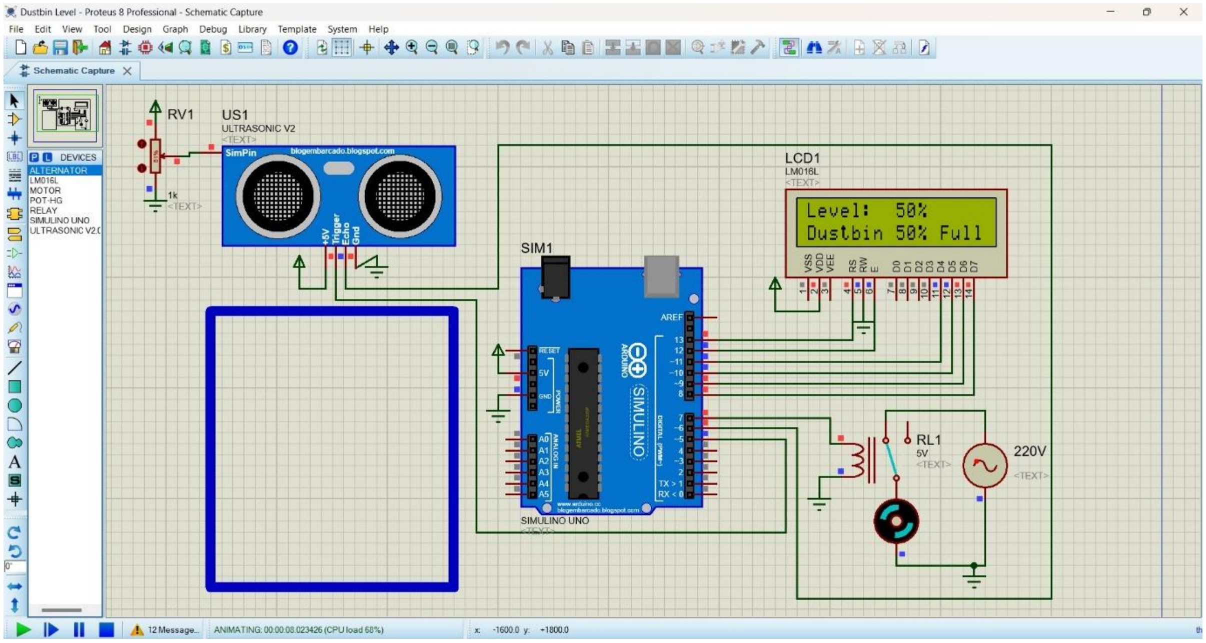

Figure 26: Simulation result with half-filled dustbin

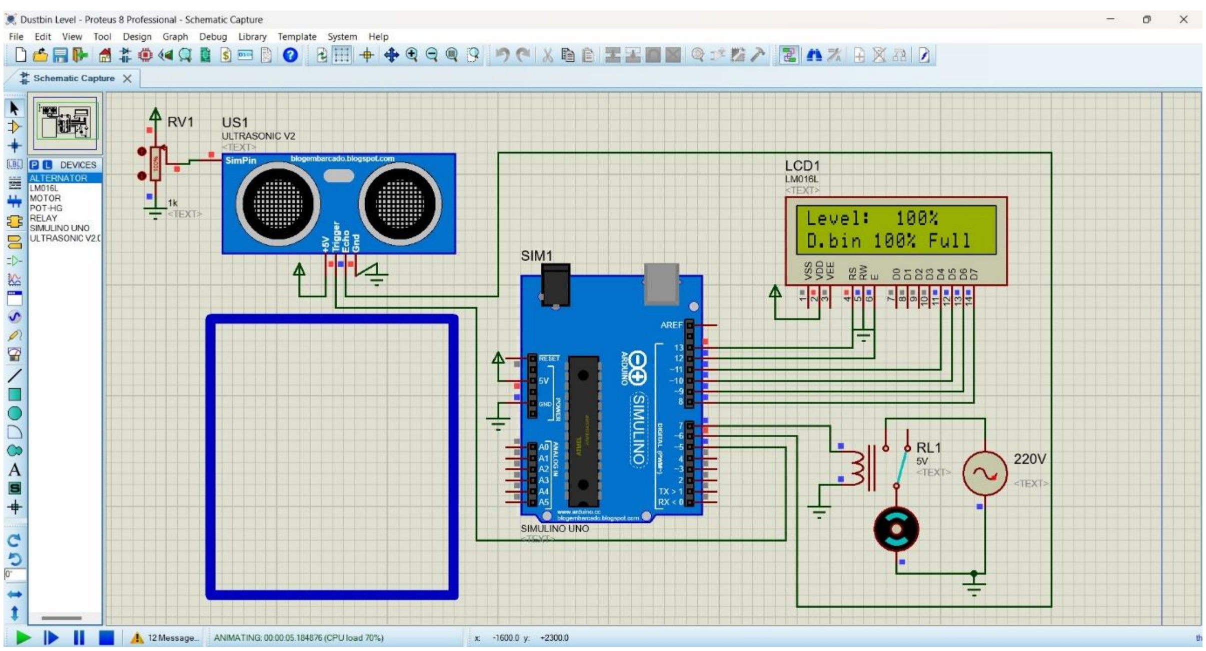

Figure 27: Simulation result with full dustbin

3.5 Circuit Diagram

The Main Part of the above Circuit diagram is the microcontroller ATmega328P. A servo motor SG90 is for output, data that is transmitted by ultrasonic sensor HC-SR04 and received at Arduino digital pin P6 (Digital PWM)). Figure 28 shows the designed circuit diagram of the smart dustbin in the Proteus 8 Professional IDE and Figure 29 shows the actual circuit which is implemented as per the designed circuit diagram after its successful simulation.

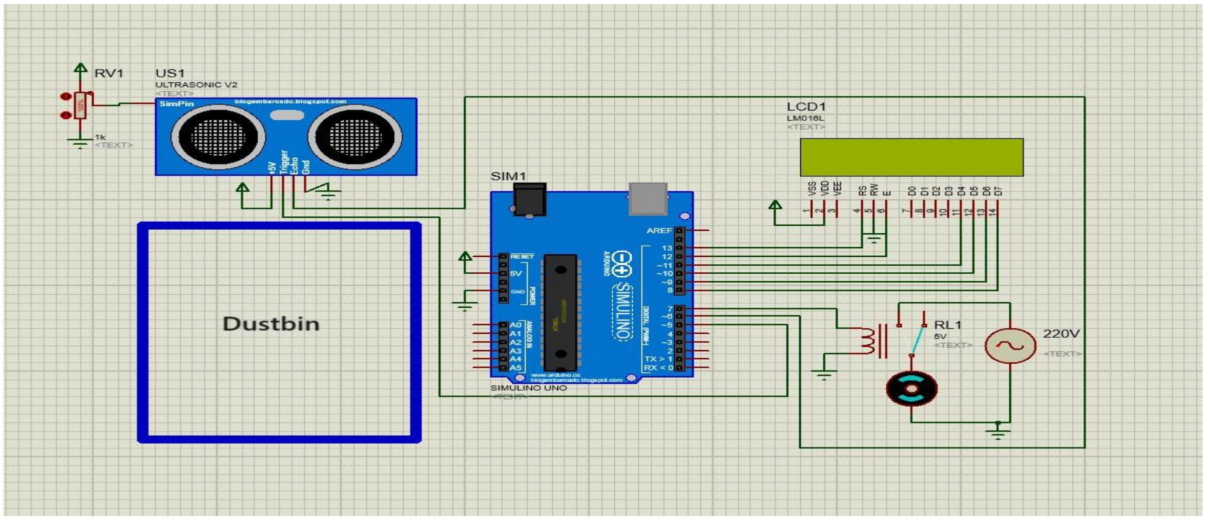

Figure 28:Circuit diagram of Smart dustbin system

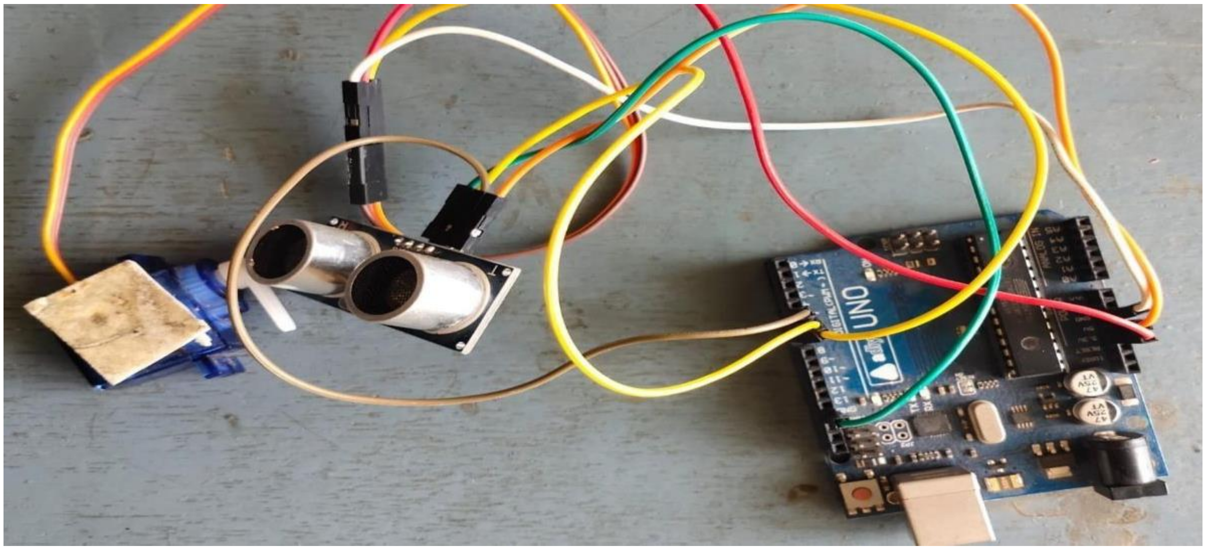

Figure 29:Circuit for Smart Dustbin system

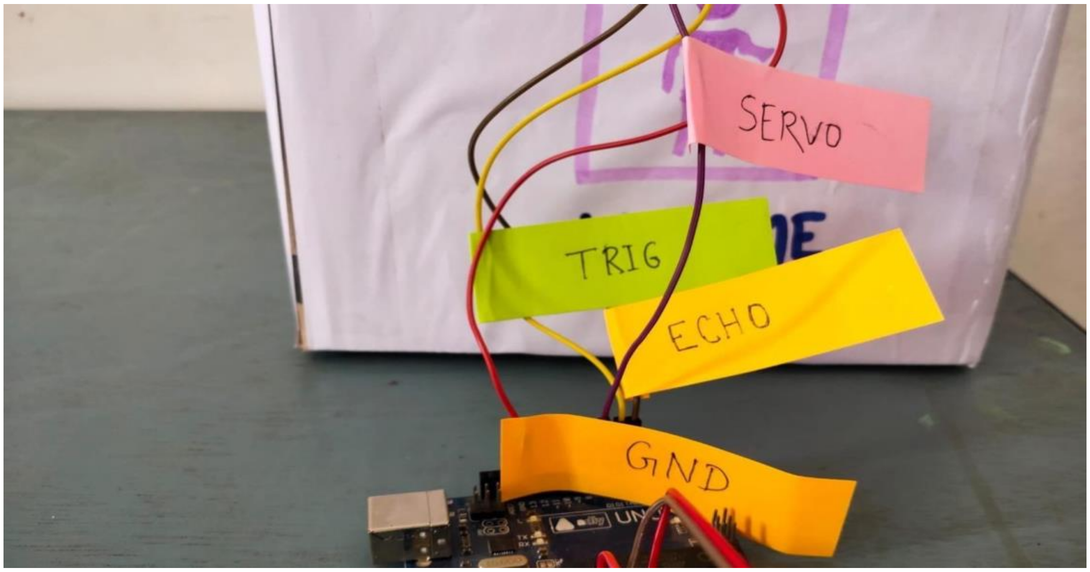

Figure 30:Wire labelled of the Circuit for Smart Dustbin System

3.6 Circuit Explanation

Figure 29 shows the circuit diagram depicting the connected essential hardware component as per the circuit diagram shown in Figure 28 and Figure 30 shows the wire labelled diagram of the circuit along with the dustbin used in this project work. Ultrasonic Sensor Trig pin connected to PIN 5 (Digital Pin (PWM)), Echo pin to PIN 6 (Digital Pin (PWM)), Vcc to 5V power supply, and Ground to GND of the Arduino UNO. Servo Motor ground wire (Brown) connected to GND, Power supply(red) pin connected to 3.3V pin and PWM signal (Orange) connected to PIN 7 (Digital Pin (PWM)) of the Arduino UNO. 9V Battery connected to Vin and GND pins of Arduino UNO. Ensure all connections are secure and well-insulated to prevent loose wires or short circuits and proceed with programming the Arduino Uno to control the ultrasonic sensor and servo motor.

3.7 Flow Chart

Figure 31shows the flow chart of the smart dustbin system used in the carried-out work. When the circuit is switched ON, Ultrasonic Sensor, Arduino UNO and servo motor. After initialization, Ultrasonic Sensor is looking for the obstacle/object. If Ultrasonic Sensor senses any object/obstacle by transmitting signal through trigger pin then it receives signal through echo pin then it sends signal to the Arduino and Arduino sends signal to servo motor and then it rotates.

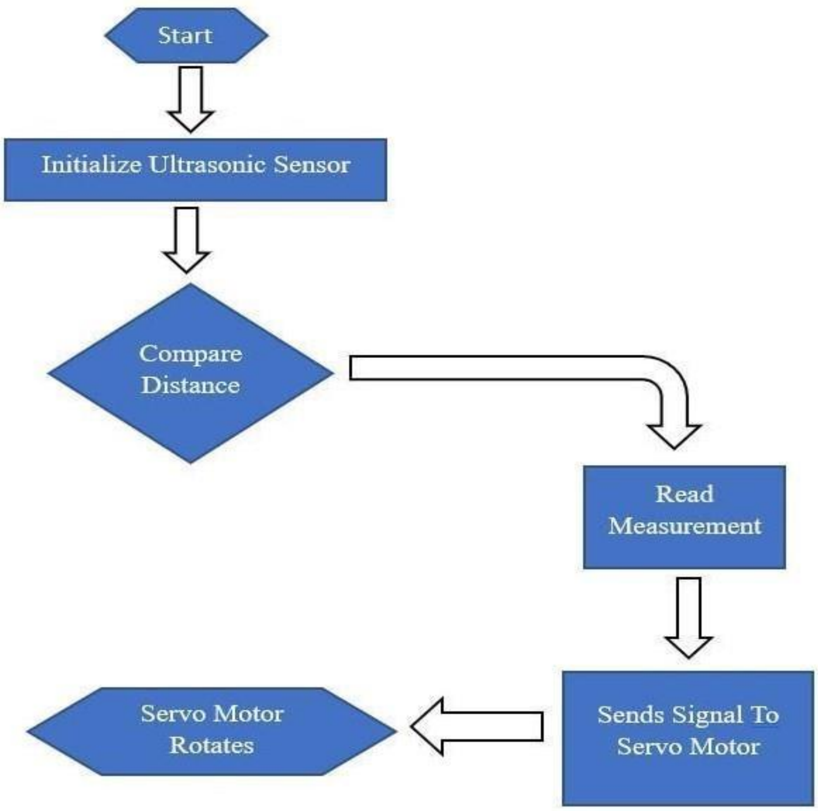

Figure 31:Flow chart for Smart Dustbin System

3.8 Working Concept

- The smart dustbin uses an Ultrasonic sensor HC-SR04 to detect objects in front. Here is a simple example of how an ultrasonic sensor works to measure distance:

I. Firstly, the transmitter (trig pin) sends a sound wave

I. The object picks the wave up, reflecting it back to the sensor.

II. The receiver (echo pin) picks it up

- It then sends the signals to Arduino UNO.
- The Arduino understands the signal and sends a signal to the Servo motor which opens the flap on top of the dustbin.

CHAPTER 4

4. Procedures and Result

4.1 Procedures

For the project we used hardware and software tools both. We used Arduino IDE software to code for Arduino UNO to generate required command for the function. The procedure's steps are mentioned in the subsequent topics in this chapter.

4.2 Arduino IDE Installation

Step 1: Install Arduino IDE (LINK: https://www.arduino.cc/en/donate/)

Step 2: Open Arduino IDE software and create a new file using File → New.

Step 3: To avoid any error while uploading program to the board, we must select the correct Arduino board name using Go to Tools → Board and select board.

Step 4: Now write your code to program the Arduino board. Step 5: After coding, compile your code using the button 

Step 6: After compiling is done. Copy the hex file location and save it anywhere.

After installation we wrote code in the IDE and run it for the command for testing and implementation.

4.3 Circuit

We constructed a circuit for the dustbin according to circuit design and perform the task accordingly.

4.4 Testing Angle of Rotation

We tested the angle of rotation for the shaft movement to function the servo motor. Figure 32, Figure 33 and Figure 34 show various angle of rotations for the shaft movement of the concerned servo motor used in this project work.

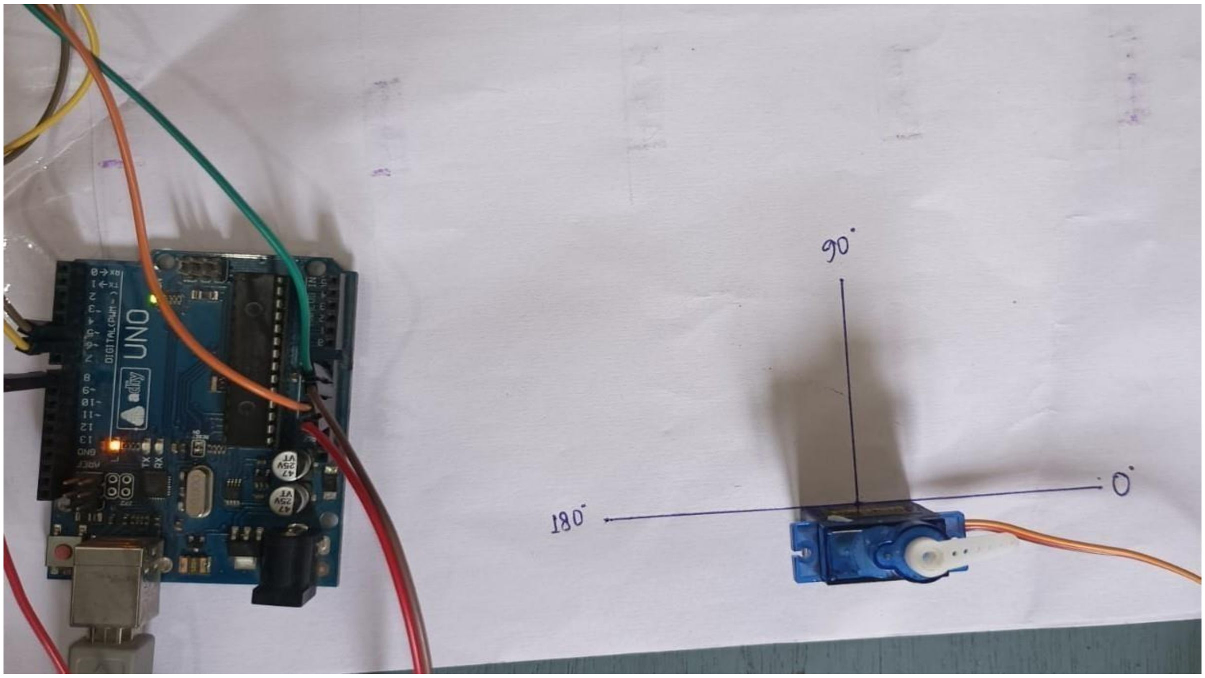

Figure 32:Angle of Rotation for 0 degree

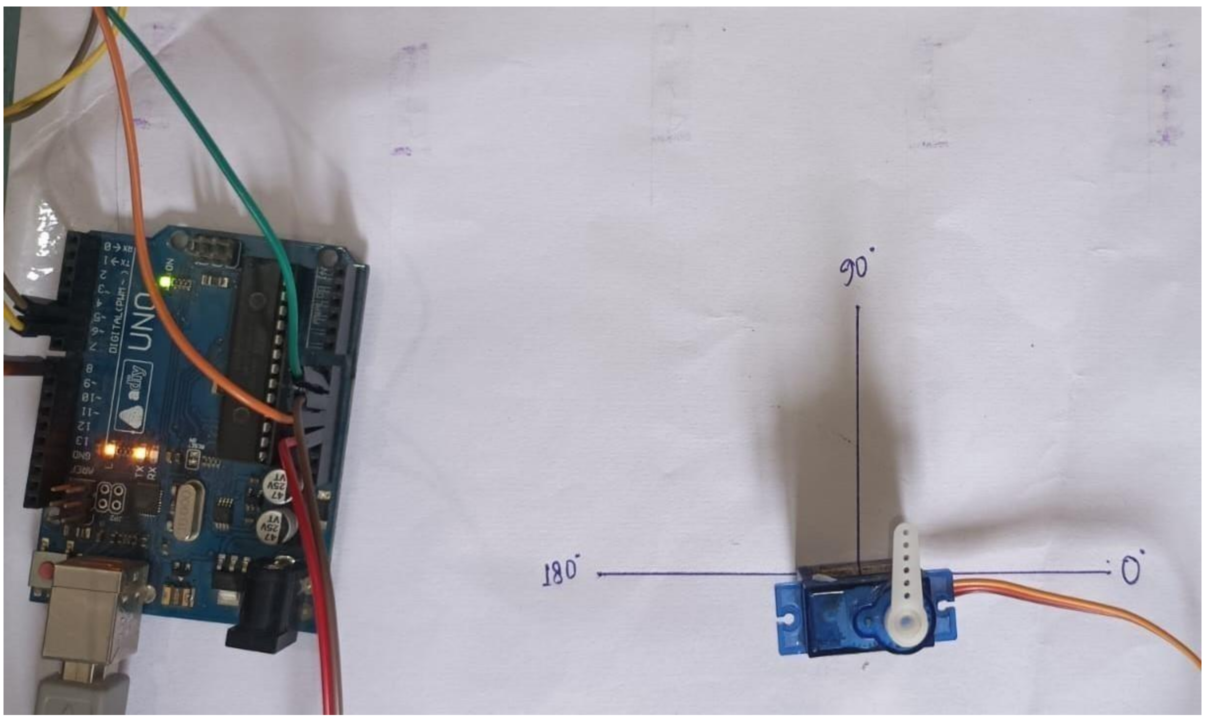

Figure 33:Angle of Rotation for 90 degree

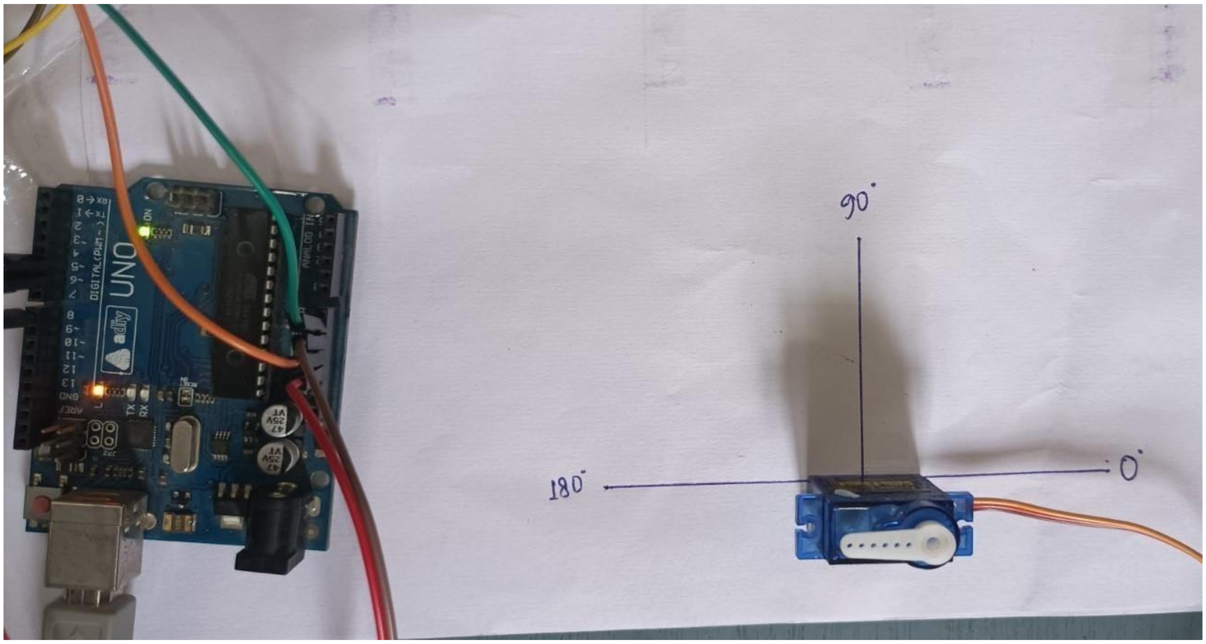

Figure 34:Angle of Rotation for 180 degree

4.5 Testing Suitable Distance Measurement

After testing angle rotation, we measured the suitable distance between object and dustbin. Figure 35 and Figure 36 shows measurement of object and dustbin to select the suitable distance up to which the opening action of the lid to be performed.

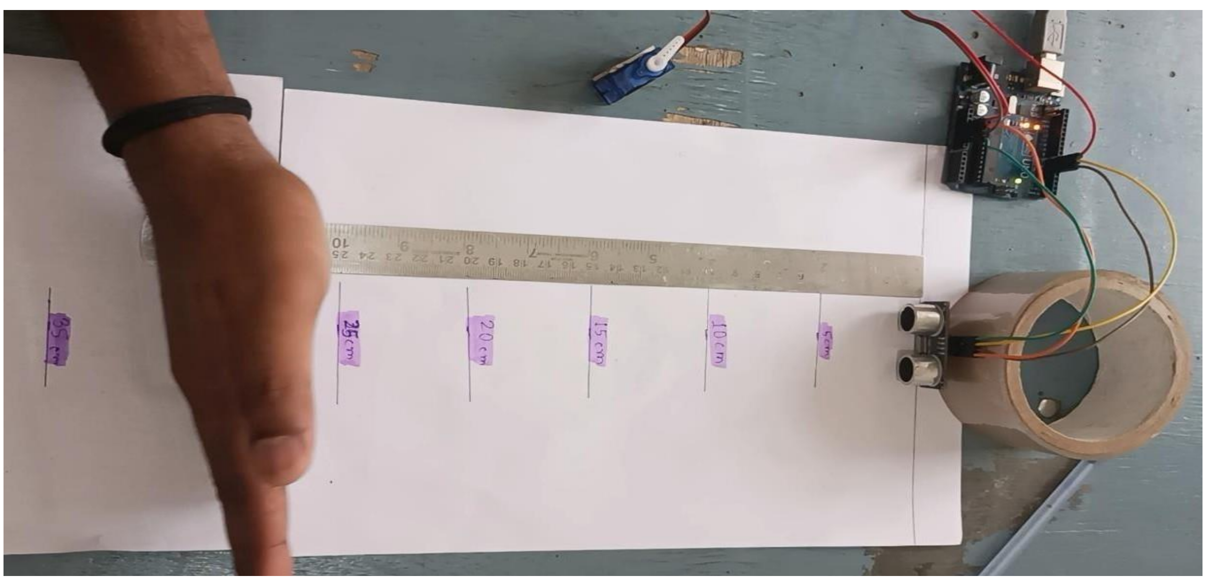

Figure 35:Testing the measurement of suitable Distance at 30 cm

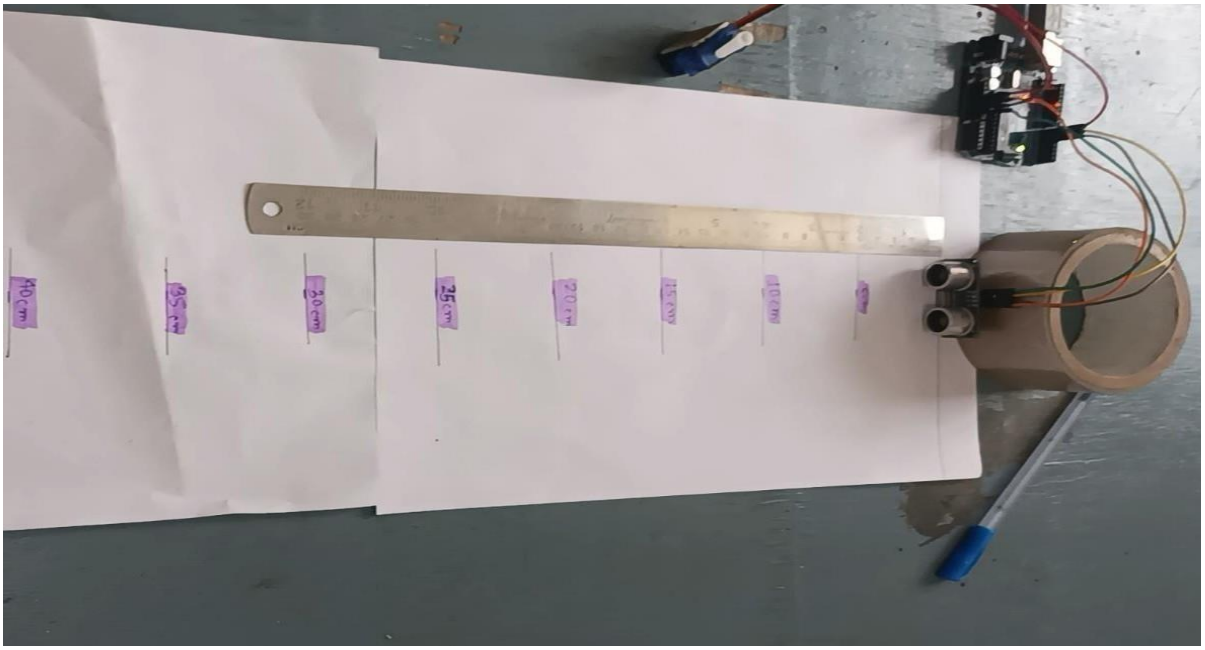

Figure 36: Testing the measurement of suitable Distance

4.6 Result

After testing angle & distance, we observed following things:

- For the opening of the lid of the dustbin 120 degree was suitable
- At 30 degree the lid was not opening properly and trouble to put garbage in the dustbin
- At 90 degree the lid gets toppled
- Initially, we tried at 50cm distance from object and distance, but later we did it for 30 cm.
- As when it was 50cm then whenever an object passes nearby the dustbin, the lid keeps opening unnecessarily.

Thus, after testing all these steps we implemented the designed dustbin to be used and named it as **Smart Dustbin System** which is shown in Figure 37.

Figure 37:Smart Dustbin System

4.7 Observations

For the working of servo motor at different parameters we also observed duty cycle by PWM waveform through DSO as shown in Figure 38 and obtained the duty cycles for the different angle of rotation. Duty cycle refers to the ratio of the time a signal is "on" or active, compared to the total time of one complete cycle. It is often expressed as a percentage or a fraction. Figure 39, Figure 40, Figure 41and Figure 42 shows PWM wave of different duty cycle for the various angle of rotations.

Figure 38:Set up to analyse the Duty Cycle

Figure 39:Duty Cycle for 0 degree of rotation

Figure 40:Duty Cycle for 40 degree of rotation

Figure 41:Duty Cycle for 90 degree of rotation

Figure 42:Duty Cycle for 120 degree of rotation

Here, this is how we calculated the duty cycle for various angle of rotation of the shaft connected to the servo motor. The duty cycle is expressed as a percentage and is calculated by dividing the duration of the "on" state by the total period of the signal and multiplying by 100. Mathematically, it is computed as:

$$Duty\ Cycle = \left(\frac{\Delta t_{on}}{\Delta t_{total}}\right)$$

Where:

- Duty Cycle is the percentage of time the signal is in the "on" state.
- Δt_{on} is the duration of the "on" state.
- Δt_{total} is the total period of the signal.

From the above equation we computed duty cycles for various angles for the selected PWM wave constant (20ms) and frequency (50Hz). The results so obtained are tabulated in Table 2.

Table 2: Angle of rotation and duty cycle relation

θ	Δt	$\Delta t/width$	Duty cycle
0°	20 ms	560 µs	2.8%
30°	20ms	840 µs	4.2%
40°	20ms	960 µs	4.8%
90°	20ms	1.48 ms	7.33%
180°	20ms	2.4 ms	12%

From above table we can conclude that at different duty cycle there will be specific different angles respectively.

CHAPTER 5

5. Conclusion and Future Scope

5.1 Conclusion

Handling dustbins can be a hassle, but we made it easier. This project introduced the Automatic Dustbin Opener, a simple solution using Arduino Uno, an ultrasonic sensor, and a servo motor. Its goal is to simplify the process of opening and closing dustbins effortlessly and contactless.

The Automatic Dustbin Opener eliminates the need for touching the bin and reduces the effort required. By using a sensor, it can detect when someone or something is in front of the dustbin. Then, a motor automatically opens the lid, making waste disposal hassle-free.

In conclusion, the Automatic Dustbin Opener with Arduino Uno, an ultrasonic sensor, and a servo motor offers a straightforward solution for effortless dustbin handling. This project aims to inspire others to embrace automation for a simpler and more convenient daily routine.

- Smart dustbins have potential benefits for waste management, including improved hygiene, convenience, accessibility, and energy efficiency.
- The potential drawbacks, such as higher cost and privacy concerns, should be considered before adoption.
- Modifications can enhance their functionality and suitability for specific needs.
- Overall, smart dustbins are a promising solution for modernizing waste management practices.

5.2 Future Scope

The Automatic Dustbin Opener project has promising potential for further enhancements and applications. Some future scope areas include:

- Smart Home Integration: Integrate the Automatic Dustbin Opener with smart home systems for seamless integration with other automated household processes.
- Waste Segregation: Expand the capabilities of the system to include waste segregation features, enabling the sorting of different types of waste into separate compartments.

- Remote Control: Develop a remote control or smartphone application to enable users to operate the dustbin opener from a distance, adding convenience and accessibility.
- Energy Efficiency: Implement energy-saving mechanisms, such as optimizing the power usage of the servo motor and ultrasonic sensor, to enhance the system's efficiency and longevity.
- Sensor Upgrades: Explore advanced sensor technologies, such as infrared or camera-based sensors, to improve accuracy and expand functionality.

Bibliography

[1] N. A. Latha, B. R. Murthy, and K. B. Kumar, "Distance sensing with ultrasonic sensor and Arduino," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 2, no. 5, pp. 1–5, 2016.

[2] Ramya, B. Palaniappan, and K. Karthick, "Embedded controller for vehicle InFront obstacle detection and cabin safety alert system," *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, vol. 4, 2012.

[3] W. Pereira, S. Parulekar, S. Phaltankar, and V. Kamble, "Smart bin (waste segregation and optimisation)," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 2019, pp. 274–279.

[4] R. Camposano and J. Wilberg, "Embedded system design," *Design Automation for Embedded Systems*, vol. 1, pp. 5–50, 1996.

[5] R. D. M. M. A. M. Janice Gillispie Mazidi, "The 8051 Microcontroller and Embedded Systems," *Using Assembly and C, Second Edition*, 2019.

[6] "ATMEL ATmega328P datasheet: ." https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA168A-PA-328-P-DS-DS40002061B.pdf (accessed May 26, 2023).

[7] Y. A. Badamasi, "The working principle of an Arduino," in *2014 11th international conference on electronics, computer and computation (ICECCO)*, 2014, pp. 1–4.

[8] S. C. Nilesh Bhaskarrao Bahadure, "Microcontrollers and Embedded System design," *Dreamtech Press Publication*, 2019.

[9] L. Louis, "working principle of Arduino and u sing it," *International Journal of Control, Automation, Communication and Systems (IJCACS)*, vol. 1, no. 2, pp. 21– 29, 2016.

[10] "Arduino IDE", Accessed: Apr. 26, 2023. [Online]. Available:

[11] https://www.arduino.cc/en/software

[12] M. H. Kamaludin, N. H. Mahmood, C. Omar, M. A. M. Yusof, and others, "Sonar assistive device for visually impaired people," *J Teknol*, vol. 73, no. 6, 2015.

[13] E. J. Morgan, "HC-SR04 ultrasonic sensor," *Nov*, 2014.

[14] A. Zhmud, N. O. Kondratiev, K. A. Kuznetsov, V. G. Trubin, and L. V Dimitrov, “Application of ultrasonic sensor for measuring distances in robotics,” in *Journal of Physics: Conference Series*, 2018, p. 32189.

[15] R. Firoozian, *Servo motors and industrial control theory*. Springer, 2014.

[16] X. Lin, “Linear servo motor operating mechanism and control technique for highvoltage circuit breaker,” *Proceedings of the CSEE, 2008*, vol. 28, no. 27, pp. 137–141, 2008.

[17] R. Krishnan, “Selection criteria for servo motor drives,” *IEEE Trans Ind Appl*, no. 2, pp. 270–275, 1987.

[18] N. Cameron and N. Cameron, “Servo and Stepper Motors,” *Arduino Applied: Comprehensive Projects for Everyday Electronics*, pp. 157–176, 2019.

[19] M. Haidar, C. Benachaiba, and M. Zahir, “Software interfacing of servo motor with microcontroller,” 2013.

[20] Servo Motor Controlwith Atmega328P:, “https://www.ee-diary.com/2023/04/servomotor-control-with-atmega328p.html”.

[21] C. Harshitha, “Characterization of In-house Developed Electroporator,” 2015.

Appendix I

%%%

CODE

%%%

```
// include the library code:
#include <LiquidCrystal.h> //library for LCD
// initialize the library with the numbers of the
interface pins
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
// defines pins numbers
const int trigPin = 5;
const int echoPin = 6;
const int Motor_Pin = 7;
// defines variables
long duration;
int distance;
bool Motor; //Make a bool Function for Motor ON/OFF
void setup()
{
pinMode(trigPin, OUTPUT); // Sets the trigPin as an
Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input

pinMode(Motor_Pin, OUTPUT); // Sets the Motor Pin as an
Output
// set up the LCD's number of columns and rows:
lcd.begin(16, 2); //LCD order
}
void loop()
```

```
{
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time
in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance in cm
distance = duration*0.034/2;
int Level = map(distance, 0,1106, 0,100);
// Prints the distance on the LCD
lcd.setCursor(0,0);
lcd.print("Level: ");
lcd.print(Level);
lcd.print("%     ");
if(Level < 100)  //if Water Level is Less than 30%
{
Motor = true;  //Make The Bool True
}
if(Level >= 100)  //if Water Level is Greater than or
Equal to 100%
{
Motor = false;  //Make The Bool False
}
if(Motor) //if Bool is True
{
```

```
digitalWrite(Motor_Pin, HIGH);
lcd.setCursor(0,1);
lcd.print("Dustbin On");
}
else //if Bool is False
{
digitalWrite(Motor_Pin, LOW);
lcd.setCursor(0,1);
lcd.print("Dustbin Full ");
}
}
```