Mini Project Report

On

# "Adaptive Cruise Control using Arduino"

Submitted in the partial fulfilment for the award of the
Degree of Bachelor of Technology

In

Electronics and Communication Engineering

By


**Aditya Raj (Roll No- 22030102)**

**Divyansh Pandey (Roll No- 22030113)**

**Pritam Saha (Roll No- 22030145)**

**Shashanka Mandal (Roll No- 22030159)**

**Sumit Kumar (Roll No- 22030161)**


B. Tech, IV Semester

Under the guidance of

**Dr. Rajiv Dey**

**Associate Professor**


DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SCHOOL OF STUDIES IN ENGINEERING AND TECHNOLOGY
GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR (C.G.)


**SESSION: 2023-24**

## ACKNOWLEDGEMENT

We are highly indebted to **Dr. Rajiv Dey** for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his/her support in the project. We owe our special thanks to our Head of Department **Dr. Soma Das** Professor, for encouraging us to acquire courage and knowledge through this project. We would like to express our gratitude towards our parents and faculty members of the Department of Electronics and Communication Engineering, School of Studies in Engineering & Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur CG for their kind co-operation and encouragement which helped us in the completion of this project. We would like to express our special gratitude and thanks to all the staffs of Electronics and Communication Engineering department for giving us such attention and time. Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

**Aditya Raj (Roll No- 22030102)**
**Divyansh Pandey (Roll No- 22030113)**
**Pritam Saha (Roll No- 22030145)**
**Shashanka Mandal (Roll No- 22030159)**
**Sumit Kumar (Roll No- 22030161)**

# DECLARATION

We the undersigned solemnly declare that this report of the minor project work, entitled **"Adaptive Cruise Control using Arduino"** is carried out during the course of our study during IV semester under the guidance of **Dr. Rajiv Dey**, Associate Professor, Department of Electronics and Communication Engineering, School of Studies in Engineering & Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur (C.G.). We further declare that this mini project work is presented for the partial fulfilment of the requirement of degree of Bachelor of Technology in Electronics & Communication Engineering, School of Studies in Engineering & Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur (C.G.).

Date: 12/07/2024

Signature- Aditya Raj
Signature- Divyansh Pandey
Signature- Pritam Saha
Signature- Shashanka Mandal
Signature- Sumit Kumar

## APPROVAL SHEET

This mini project report entitled **"Adaptive Cruise Control using Arduino"** by **Aditya Raj**, **Divyansh Pandey**, **Pritam Saha, Shashanka Mandal** and **Sumit Kumar** is approved for the partial fulfilment of the requirement of the degree of Bachelor of Technology in Electronics and Communication Engineering.

Signature of Guide

**Dr. Rajiv Dey**
(Associate Prof)

Examiners:

Signature of Examiners

Date: 12/07/2024

Place: Bilaspur

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**SCHOOL OF STUDIES IN ENGINEERING AND TECHNOLOGY**

**GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR (C.G.)**

(A Central University established by the Central University Act 2009 No. 25 of 2009)

# CERTIFICATE

It is certified that the mini project entitled **“Adaptive Cruise Control using Arduino”** submitted by **Aditya Raj**, **Divyansh Pandey**, **Pritam Saha, Shashanka Mandal** and **Sumit Kumar** in partial fulfilment of the requirements of the award of the degree of Bachelor of Technology in Electronics and Communication Engineering, School of studies in Engineering and Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur, is carried out by them in the Department of Electronics and Communication Engineering during session 2023-24 under supervision and guidance of **Dr. Rajiv Dey**, Associate Professor, Department of Electronics & Communication Engineering, School of Studies in Engineering & Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur CG.

**Dr. Soma Das**
Head of Department
Department of Electronics & Communication Engineering
School of Studies in Engineering & Technology
Guru Ghasidas Vishwavidyalaya, Bilaspur CG

# PREFACE

This project report presents our work on **Adaptive Cruise Control**, a significant advancement in automotive technology that enhances driving safety and convenience. Conducted under the guidance of **Dr. Rajiv Dey**, and overseen by **Dr. Soma Das**, Head of the Department of Electronics and Communication Engineering, this project aims to explore the design, implementation, and performance evaluation of an adaptive cruise control system.

As a team of five members—**Aditya Raj, Divyansh Pandey, Pritam Saha, Shashanka Mandal, and Sumit Kumar**—we collaborated closely to research existing technologies, develop a comprehensive system architecture, and perform rigorous testing to assess the effectiveness of our implementation. This experience has not only deepened our understanding of embedded systems and control mechanisms but also honed our skills in teamwork and project management.

We express our gratitude to our guide and HOD for their invaluable support and guidance throughout this project. We hope that this report serves as a valuable resource for understanding the complexities and innovations associated with adaptive cruise control systems.

# ABSTARCT

This project explores the design and implementation of an Adaptive Cruise Control (ACC) system using Arduino technology. The ACC system is an advanced driver-assistance feature that automatically adjusts a vehicle's speed to maintain a safe following distance from the vehicle ahead. Our project aims to replicate this functionality on a small scale by integrating Arduino with various sensors and actuators.

The system utilizes an ultrasonic sensor to measure the distance between the vehicle and obstacles in its path. Based on the sensor data, the Arduino microcontroller processes the information and dynamically adjusts the speed of a DC motor, simulating the vehicle's response. An H-bridge motor driver is employed to control the motor's direction and speed, while a Liquid Crystal Display (LCD) provides real-time feedback on the distance detected.

Our implementation highlights the potential for Arduino-based systems to enhance vehicular safety and autonomy. Through rigorous testing, we evaluated the performance and reliability of our ACC system, demonstrating its effectiveness in maintaining a safe distance from obstacles. This project not only deepened our understanding of embedded systems and control mechanisms but also emphasized the importance of sensor integration and real-time data processing in developing intelligent automotive solutions.

# LIST OF FIGURES

# INDEX

# CHAPTER 1

# 1.1 INTRODUCTION

Autonomous, self-driving, or driverless cars have the potential to reduce traffic fatalities and traffic congestion worldwide. A fully autonomous car would be capable of driving in all situations, without any input from a human driver. While cars have not yet achieved this level of autonomy, many new cars have a variety of driver-assist and safety features, such as automatic braking, lane departure warnings, and adaptive cruise control. Some have more autonomy and are able to self-drive in specific situations, such as on highways with well-marked lanes. Tesla is an example of a car company known for making cars with some autonomous driving features.

Just as humans use our senses (sight, sound, touch, etc.) to perceive the world around us when driving, autonomous cars use a variety of electronic sensors to monitor the roadway and surroundings. Cameras work with complex computer vision systems to process visual information, while radar or ultrasonic sensors measure the distance between the car and other objects.

Based on information from these sensors, the car decides how to react—without input from the driver. For example, if a sensor detects that the car in front of it is rapidly slowing down, the car may automatically apply the brakes in order to slow down and avoid a collision. Or if the sensors detect that the car is driving out of its lane and off the side of the road, the car may automatically correct its steering to stay in the lane.

# 1.2 COMPONENTS

a) **Arduino**: Acts as the central control unit, processing sensor data and controlling the motor speed.

b) **Ultrasonic Sensor**: Measures the distance to obstacles ahead of the car.

c) **LCD Display**: Displays the current distance detected by the ultrasonic sensor.

d) **H-Bridge**: Allows control of the motor's direction.

e) **PWM DC Motor**: Drives the car, with its speed controlled via PWM signals.

f) **L298N Motor Driver**: Interfaces the Arduino with the DC motor and H-bridge to manage power and control signals.

*The above-mentioned components are now explained in detail.

## Arduino:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It consists of a microcontroller, typically an Atmel AVR or ARM Cortex-M based, mounted on a board with various input and output pins that can be connected to other electronic components and devices. Arduino boards can be programmed using the Arduino Integrated Development Environment (IDE), which uses a simplified version of C++. This platform is widely used for creating interactive projects, from simple hobbyist gadgets to complex automation systems, due to its accessibility and versatility.

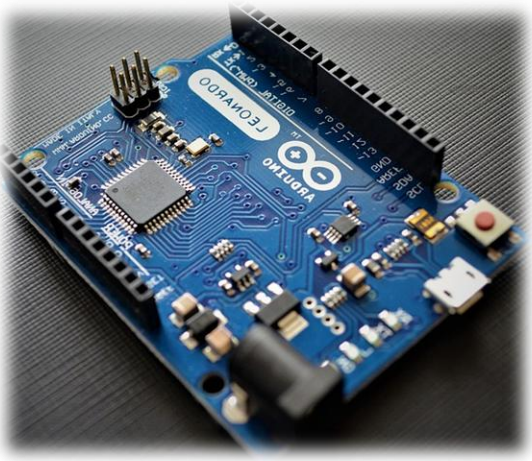

**Fig.-1**

# Ultrasonic sensor:

The HC-SR04 is an affordable and easy to use distance measuring sensor which has a range from 2cm to 400cm (about an inch to 13 feet).

The sensor is composed of two ultrasonic transducers. One is transmitter which outputs ultrasonic sound pulses and the other is receiver which listens for reflected waves. It’s basically a SONAR which is used in submarines for detecting underwater objects.

In order to generate the ultrasound, we need to set the Trig pin on a High State for 10 µs. That will send out an 8-cycle ultrasonic burst which will travel at the speed of sound. The Echo pins goes high right away after that 8-cycle ultrasonic burst is sent, and it starts listening or waiting for that wave to be reflected from an object.

If there is no object or reflected pulse, the Echo pin will time-out after 38ms and get back to low state.

If we receive a reflected pulse, the Echo pin will go down sooner than those 38ms. According to the amount of time the Echo pin was HIGH, we can determine the distance the sound wave travelled, thus the distance from the sensor to the object.

For that purpose we are using the following basic formula for calculating distance:

**Distance = Speed x Time**

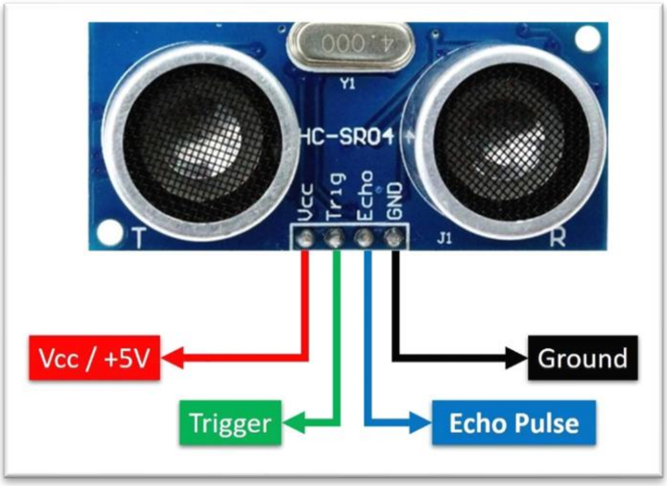


**Fig.-2**

# LCD Display:

An LCD character display is a unique type of display that can only output individual ASCII characters with fixed size. Using these individual characters then we can form a text.

If we take a closer look at the display, we can notice that there are small rectangular areas composed of 5×8 pixels grid. Each pixel can light up individually, and so we can generate characters within each grid.


**Fig.-3**

# H-Bridge:

For controlling the rotation direction, we just need to inverse the direction of the current flow through the motor, and the most common method of doing that is by using an H-Bridge. An H-Bridge circuit contains four switching elements, transistors or MOSFETs, with the motor at the centre forming an H-like configuration. By activating two particular switches at the same time, we can change the direction of the current flow, thus change the rotation direction of the motor.

So, if we combine these two methods, the PWM and the H-Bridge, we can have a complete control over the DC motor. There are many DC motor drivers that have these features and the L298N is one of them.

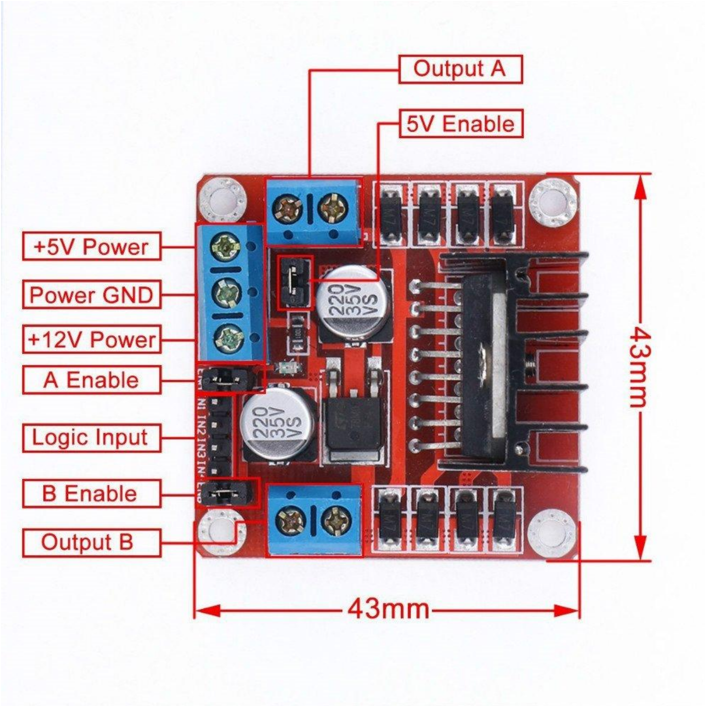


**Fig.-4**

# PWM DC Motor:

PWM, or pulse width modulation is a technique which allows us to adjust the average value of the voltage that's going to the electronic device by turning on and off the power at a fast rate. The average voltage depends on the duty cycle, or the amount of time the signal is ON versus the amount of time the signal is OFF in a single period of time.

So, depending on the size of the motor, we can simply connect an Arduino PWM output to the base of transistor or the gate of a MOSFET and control the speed of the motor by controlling the PWM output. The low power Arduino PWM signal switches on and off the gate at the MOSFET through which the high-power motor is driven.

**Fig.-5**

# L298N Motor Driver:

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

# Breadboard:

A breadboard is a reusable platform for prototyping electronic circuits without the need for soldering, making it ideal for engineers, hobbyists, and students. Typically made of plastic with a grid of holes, it features metal strips underneath that create electrical connections. Breadboards usually have power rails along the sides for distributing power and terminal strips in the centre for inserting components. Their modular design allows for various sizes, and they can be easily modified, making them perfect for testing new circuit designs. Commonly used components include resistors, capacitors, integrated circuits, microcontrollers, and LEDs. Overall, breadboards are essential tools in electronics for experimenting, teaching, and prototyping circuits efficiently.

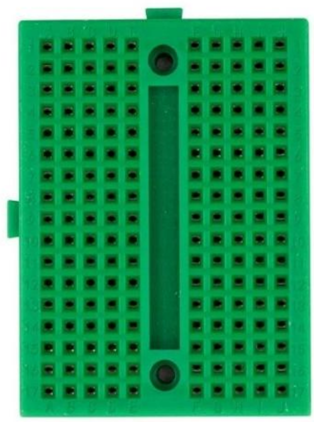

**Fig.-6**

# Caster Wheel:

A caster wheel is a wheeled device designed to facilitate easy movement and manoeuvrability of objects, commonly used in applications such as furniture, carts, and industrial equipment. Comprising a wheel mounted on a fork, caster wheels can be categorized into swivel casters, which allow for 360-degree rotation for enhanced manoeuvrability, and rigid casters, which provide straight-line movement and stability. Made from various materials, including rubber, polyurethane, nylon, and metal, caster wheels offer different benefits in terms of durability and load capacity. They are used in diverse settings, from office chairs and rolling desks to industrial carts and medical equipment, providing essential mobility and efficiency. Overall, caster wheels are integral components across multiple industries, ensuring stability and ease of transport for a wide range of goods and equipment.

**Fig.-7**

# SOME MORE COMPONENTS:


**Fig.-8**

HW-Battery

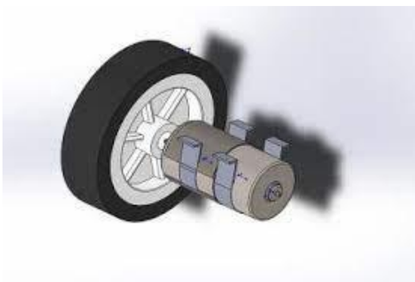

**Fig.-9**

Motor with wheel

**Fig.-10**

Switch

# CHAPTER 2

# 2.1 CIRCUIT DIAGRAM

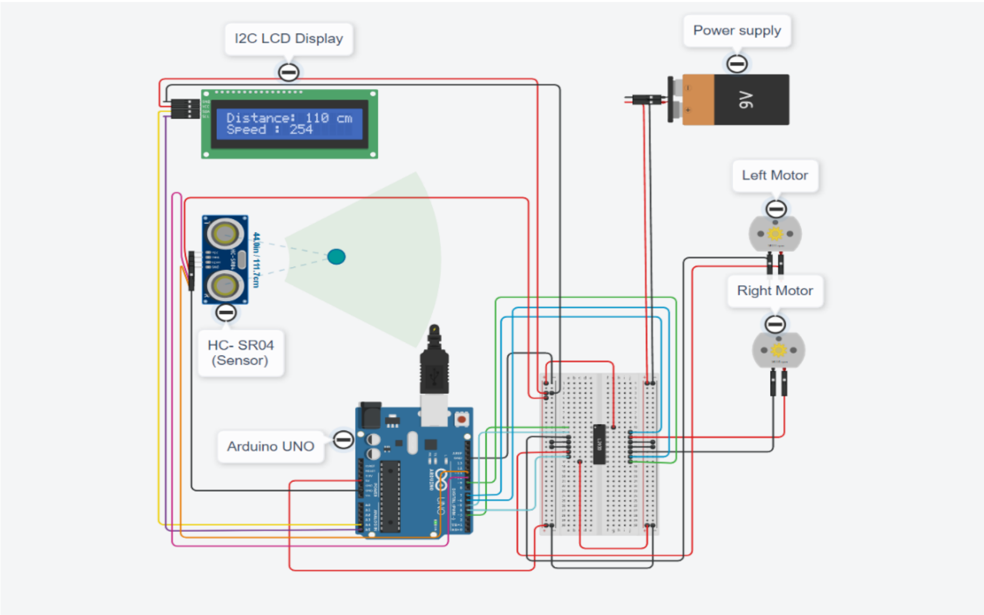


**Fig.-11**

When the motor is ON

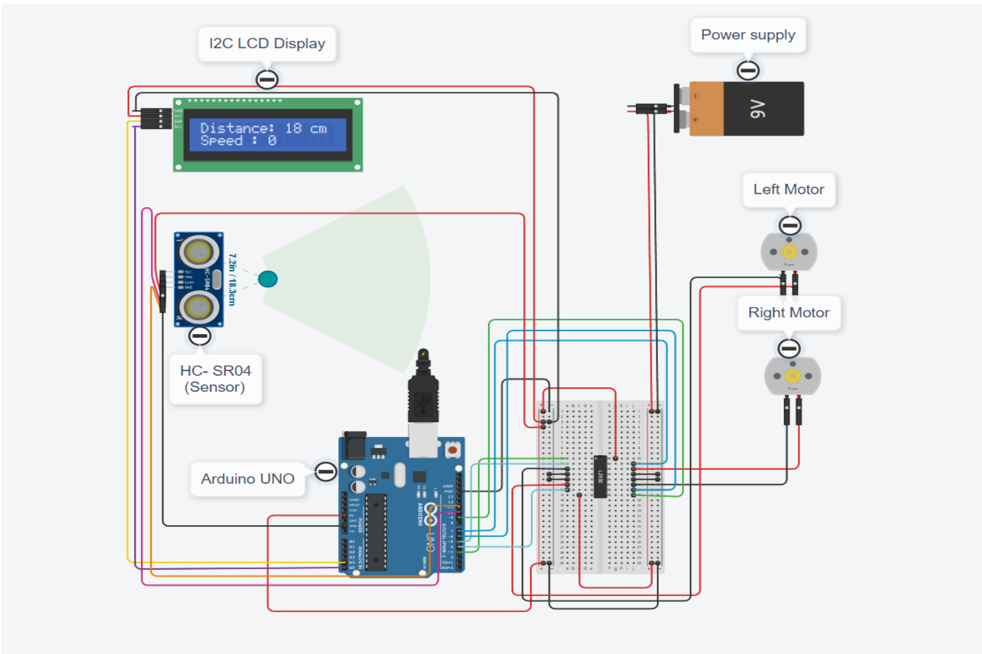


**Fig.-12**

When the motor is OFF

The above circuit diagram is made using Tinker CAD. We will demonstrate the live simulation during presentation.

## 2.2 SIMULATION

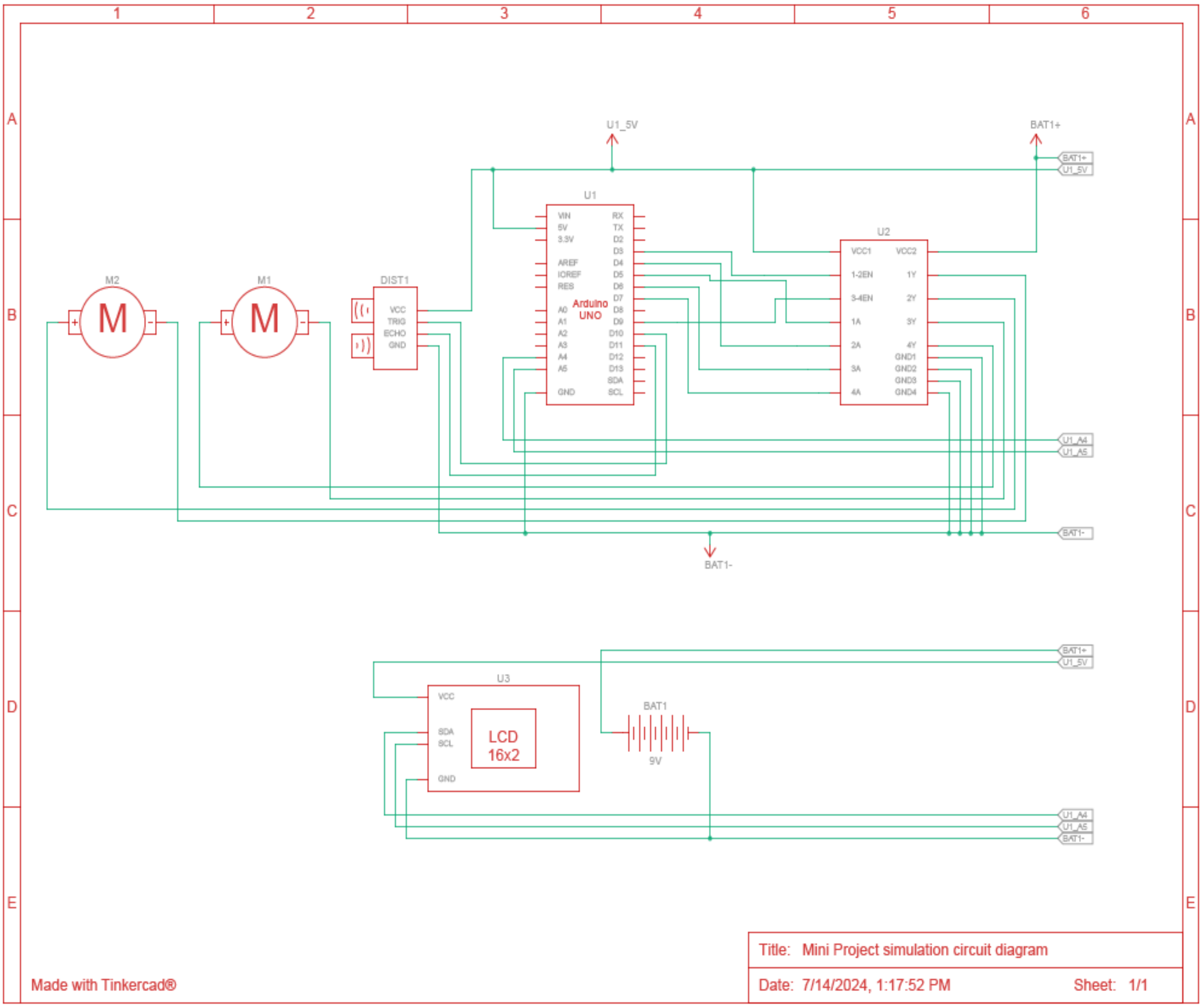


**Fig.-13**

Component List

| Name | Quantity | Component |
|---|---|---|
| U1 | 1 | Arduino Uno R3 |
| U2 | 1 | H-bridge Motor Driver |
| M1<br>M2 | 2 | DC Motor |
| BAT1 | 1 | 9V Battery |
| DIST1 | 1 | Ultrasonic Distance Sensor (4-pin) |
| U3 | 1 | PCF8574-based, 32 (0x20) LCD 16 x 2 (I2C) |

# CHAPTER 3

# 3.1 CODE

```
#include <LiquidCrystal.h>
#include <LCD_I2C.h>

LCD_I2C lcd(0x27, 16, 2);
const int trigPin = 10;
const int echoPin = 11;
long duration;
int distanceCm, distanceInch;

// Motor Pins
int ena = 3;
int in1 = 5;
int in2 = 4;

int in3 = 6;
int in4 = 7;
int enb = 9;

// Motor speed variables
int enaSpeed = 255; // Initial speed for motor A
int enbSpeed = 200; // Initial speed for motor B

void setup() {
  delay(2000);
  pinMode(ena, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(enb, OUTPUT);
```

```
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // LCD with Sensor
  lcd.begin();
  lcd.backlight();

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  // LCD with Sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distanceCm = duration * 0.034 / 2;
  distanceInch = duration * 0.0133 / 2;

  lcd.setCursor(1, 0);
  lcd.print("Distance: ");
  lcd.print(distanceCm);
  lcd.print(" cm");
  delay(1);

  // Motor control
  if (distanceCm <= 10) {
    // Stop motors
```

```
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    analogWrite(ena, 0);
    analogWrite(enb, 0);
  } else if (distanceCm > 10 && distanceCm <= 20) {
    // Gradually slow down motors
    int speed = map(distanceCm, 10, 20, 0, max(enaSpeed, enbSpeed));
    analogWrite(ena, map(speed, 0, max(enaSpeed, enbSpeed), 0, enaSpeed));
    analogWrite(enb, map(speed, 0, max(enaSpeed, enbSpeed), 0, enbSpeed));
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  } else {
    // Motor A and B Clockwise Max Speed
    analogWrite(ena, enaSpeed);
    analogWrite(enb, enbSpeed);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
  }
}
```

# 3.2 OPERATION

## 1. System Initialization

When the system powers on, the Arduino microcontroller initializes all connected components:

- **LCD Initialization**:
  - The `LiquidCrystal_I2C` library initializes the LCD screen to start displaying information.
  - The display is cleared, and the initial text ("Distance: ") is printed on the first line.
- **Pin Modes Setup**:
  - The pins connected to the ultrasonic sensor (trigPin and echoPin) are set as OUTPUT and INPUT, respectively.
  - The motor control pins (enA, in1, in2) are set as OUTPUT.

## 2. Distance Measurement with Ultrasonic Sensor

The Arduino continuously measures the distance to obstacles using the HC-SR04 ultrasonic sensor:

- **Triggering the Sensor**:
  - The Arduino sends a 10 microsecond HIGH pulse to the trigPin to start the measurement.
  - This action causes the ultrasonic sensor to emit an ultrasonic burst (8 cycles of 40kHz sound waves).
- **Echo Reception**:
  - The sensor waits for the echo of the sound wave to return after bouncing off an obstacle.
  - The echoPin goes HIGH for the duration of the time it takes for the echo to return.
- **Time Measurement**:

- The Arduino uses the `pulseIn` function to measure the duration for which the echoPin remains HIGH.
- The measured duration is in microseconds.

- **Distance Calculation**:
  - The distance to the obstacle is calculated using the formula:

$$\text{Distance (cm)} = \frac{\text{Duration } (\mu\text{s}) \times 0.034}{2}$$

  - This formula accounts for the speed of sound in air (approximately 343 meters per second), divided by 2 because the sound wave travels to the obstacle and back.

### *3. Displaying Distance on LCD*

- **Updating LCD**:
  - The Arduino updates the second line of the LCD with the current distance measurement.
  - Previous distance values are cleared to ensure a clean display.
  - Example display: "Distance: 15 cm".

### *4. Motor Speed Control Based on Distance*

The Arduino adjusts the speed of the DC motor based on the measured distance to ensure safe operation:

- **Distance > 20 cm**:
  - The car moves at full speed.
  - **Motor Control**:
    - `in1` is set to HIGH.
    - `in2` is set to LOW.
    - `enA` is set to 255 (maximum PWM value), driving the motor at full speed.
- **Distance between 10 cm and 20 cm**:
  - The car's speed decreases linearly as the distance decreases.

- **Linear Speed Control**:
  - The `map` function is used to convert the distance range (10 to 20 cm) to the PWM range (0 to 255).
  - For example, if the distance is 15 cm, the mapped PWM value would be approximately 127, resulting in half-speed.
  - `in1` is set to HIGH, `in2` is set to LOW, and `enA` is set to the mapped PWM value.
- **Distance ≤ 10 cm**:
  - The car stops to avoid collision.
  - **Motor Control**:
    - Both `in1` and `in2` are set to LOW, stopping the motor.
    - `enA` is set to 0, ensuring no PWM signal is sent to the motor.

### *5. Continuous Operation*

The above steps are executed in a loop to continuously monitor the distance and adjust the motor speed accordingly:

- **Real-time Adjustment**:
  - The Arduino continuously reads the distance from the ultrasonic sensor.
  - Updates the LCD display with the new distance.
  - Adjusts the motor speed based on the latest distance measurement.
- **Delay**:
  - A short delay (e.g., 100 milliseconds) is included in the loop to allow for stable operation and prevent excessive processing load on the Arduino.

# 3.3 PROJECT IMAGES

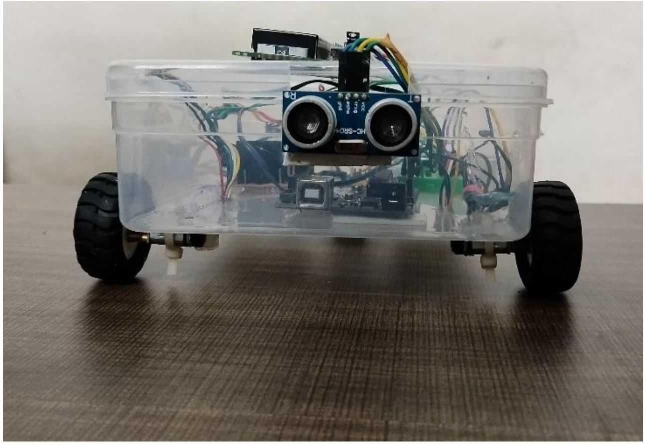

**Fig.14**

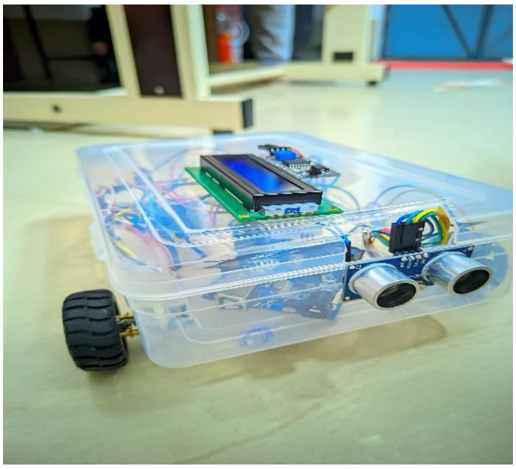

**Fig.-15**

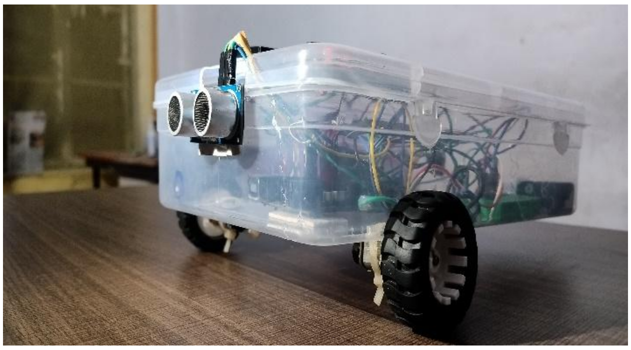

**Fig.16**

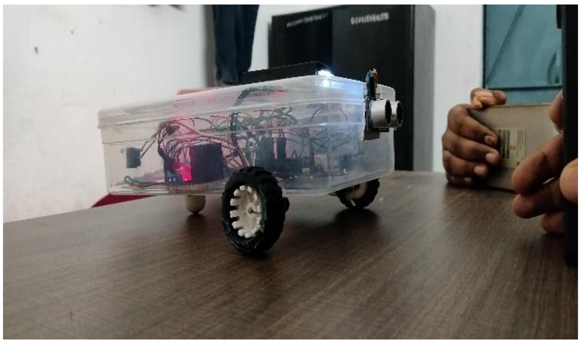

**Fig.17**

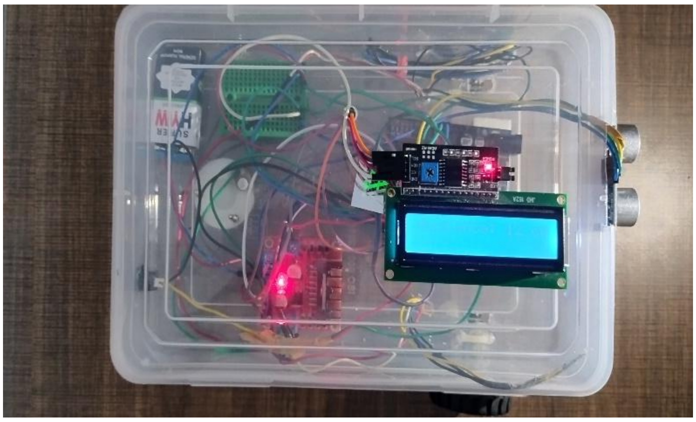

**Fig.-18**

# CHAPTER 4

# 4.1 CONCLUSION

The Adaptive Cruise Control (ACC) mini project successfully demonstrates the integration of basic electronic components and Arduino programming to create a functional safety system. By using an ultrasonic sensor for distance measurement, an LCD for real-time feedback, and a motor control setup (H-Bridge, PWM DC motor, and L298 driver), the system dynamically adjusts the car's speed to maintain a safe distance from obstacles.

## *Key Achievements*

- **Effective Distance Measurement**: Accurate real-time distance readings using the ultrasonic sensor.
- **Adaptive Speed Control**: Smooth adjustment of motor speed based on the distance, ensuring safe operation.
- **Obstacle Avoidance**: Automatic stopping of the car at 10 cm to prevent collisions.

## *Educational and Practical Value*

- **Learning Experience**: Offers hands-on experience in Arduino programming, sensor integration, and motor control.
- **Real-World Application**: Demonstrates principles applicable to advanced vehicle safety systems and robotics.

# 4.2 FUTURE SCOPE

The Adaptive Cruise Control (ACC) mini project has significant potential for future advancements and applications:

1. **Enhanced Sensor Integration**:
    - **Advanced Sensors**: Incorporate LIDAR, radar, or infrared sensors for better accuracy and reliability.
    - **Multi-Sensor Fusion**: Combine data from multiple sensors for comprehensive environmental perception.
2. **Extended Features**:
    - **Lane-Keeping Assistance**: Add vision-based systems to detect lane markings and maintain lane position.
    - **Adaptive Speed Control**: Implement algorithms for speed adjustment based on traffic conditions and road curves.
    - **Collision Avoidance**: Develop systems for evasive manoeuvres in addition to stopping.
3. **Connectivity**:
    - **V2X Communication**: Enable communication with other vehicles and infrastructure for enhanced safety.
    - **Remote Monitoring**: Integrate Bluetooth, Wi-Fi, or cellular modules for remote control and diagnostics.
4. **Autonomous Driving**:
    - **Higher Autonomy**: Expand capabilities for autonomous driving in diverse environments.
    - **AI and Machine Learning**: Use AI to improve decision-making and adaptive control.

5. **Robotics Applications**:
    - **Autonomous Robots**: Adapt the system for autonomous robots in logistics, healthcare, and agriculture.
    - **Swarm Robotics**: Develop systems for collaborative multi-robot operations.
6. **Educational and Research Opportunities**:
    - **Educational Kits**: Create kits for teaching electronics and robotics in schools and universities.
    - **Research Projects**: Provide a foundation for research in autonomous systems and intelligent transportation.
7. **Integration with Electric Vehicles (EVs):**
    - Investigate how ACC can be optimized for electric vehicles, including energy management and battery life considerations.
    - Develop algorithms that account for the unique performance characteristics of EVs

These enhancements can transform the ACC project into a sophisticated, versatile system with wide-ranging applications and educational value.

# 4.3 REFERENCES


a) https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p042/robotics/arduino-self-driving-car

b) https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

c) https://howtomechatronics.com/?s=Encoder

d) https://www.tinkercad.com/things/e864oSgATAa-tremendous-amur/editel?sharecode=mfvoHif2BcQNzXudfquf3G-OZ3o0_Pv-IikaBiywOyA