Mini Project Report on

Arduino Car Speed Detector

Submitted by

G.Venkata Sai (20711A0449)

G.Nitheesh (20711A0450)

G.Jagadeesh (21715A0452)

I.Chareesh (20711A0454)

K.V.Midhun (20711A0460)

Under the Guidance of

M.Sundeep

Assistant professor

Department of Electronics and Communication Engineering

NARAYANA ENGINEERING COLLEGE :: NELLORE

(Autonomous)

Approved by AICTE, NewDelhi & Permanently Affiliated to JNTUA, Anantapuramu, Accredited with" A" Grade by NAAC) Nellore, SPSRNellore(DT), A.P-524004.

June 2023

NARAYANA ENGINEERING COLLEGE :: NELLORE

(Autonomous)

Approved by AICTE, NewDelhi & Permanently Affiliated to JNTUA, Anantapuramu, Accredited with" A" Grade by NAAC)Nellore,SPSRNellore(DT), A.P-524004.

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the report entitled "**ARDUINO CAR SPEED DETECTOR**" being submitted byG.VenkataSai(20711A0449),G.Nitheesh(20711A0450),G.Jagadeesh(20711A0452),I.Chareesh(20711A0454),K.V.Midhun(20711A0460) in partial fulfillment for the award of the Degree of Bachelor of Technology in Electronics & Communication Engineering Department to the Jawaharlal Nehru Technological University Ananthapur,Ananthapuram is a record of bonafied work carried out by then under my guidance and supervision.

M.Sundeep

Assistant Professor

Project Supervisor

Dr.K.Murali M.Tech,Ph.D

Head of the Department

Department of ECE

ACKNOWLEDGEMENT

It is our privilege to express our gratitude and respect to all those who guided, inspired and helped us in the completion of this Project.

It is our humble responsibility to acknowledge our chairman Dr.P.Narayana, for permitting us to use facilities available in this college to accomplish the Project completely.

We express our great pleasure to record our deep sense of gratitude and profound thanks to our beloved Principal Dr. G. Srinivasulu Reddy, for providing the facilities to carry out our project work.

We express our deep sense of gratitude to Dr.K.Murali, Head of the department E.C.E for their help in constructive suggestions made at various levels of the project work.

It is our humble responsibility to acknowledge our guide **M.Sundeep**, for his kind help that he extended at every stage of our work for all provisions made us to complete this endeavor successfully and we are also thankful to all our staff members of Electronics and Communication Engineering for their co-operation. Last but not least, we wish to thank to all our friends, and who helped directly or indirectly in completion of our Socially relevant Project.

Project Associates

G.Venkata Sai (20711A0449)

G.Nitheesh (20711A0450)

G.Jagadeesh (21715A0452)

I.Chareesh (20711A0454)

K.V.Midhun (20711A0460)

ABSTRACT

This Arduino Car Speed Detector project can be used to detect speed of a moving car. There are definite rules laid out by authorities about driving cars on roads. The most common rule in any country is speed limit in certain roads i.e. you will be in violation of the law if your car speed exceeds this limit. In order to detect the speed of a moving car, the patrolling officers usually depend on a handheld gun that works on Radar Technology or Lidar Technology. This is a tedious process as the officer has to manually check for over speeding for each vehicle.A simple automatic detection of speed of a vehicle is designed in Arduino Car Speed Detector project, where you can place the system in one place and view the results instantly without any human intervention. IR Sensors are the main part of the project that detect the speed of a car. Practically, you can implement the setup of IR Sensors in many ways but in this project, I have used two reflective type IR Sensors and placed them 10cm apart.When a car travelling reaches the first sensor, the IR Sensor gets activated. From this moment onward, a timer is initiated and will continue to keep time until the car reaches the second IR Sensor.By simulating the distance between the two sensors to be 5 meters, you can calculate the speed at which the car travelled from IR Sensor 1 to IR Sensor 2 as you already know the time of travel.All the calculations and data gathering are done by Arduino and the final result is displayed on a 16X2 LCDModule.

CONTENTS

LIST OF FIGURES

CHAPTER -1

INTRODUCTION

The Arduino Car Speed Detector is an innovative project that utilizes Arduino microcontrollers and sensors to accurately measure and display the speed of moving vehicles. Monitoring and controlling vehicle speed is crucial for traffic management, road safety, and research purposes. This project provides an affordable and reliable solution for speed detection, making it accessible to a wide range of users.The system incorporates an ultrasonic sensor, which measures the distance between the sensor and the passing vehicle. By calculating the time it takes for the vehicle to travel a known distance, the Arduino board can determine the speed of the car. The obtained speed data is then displayed on an LCD screen in real-time, allowing for easy monitoring and analysis.

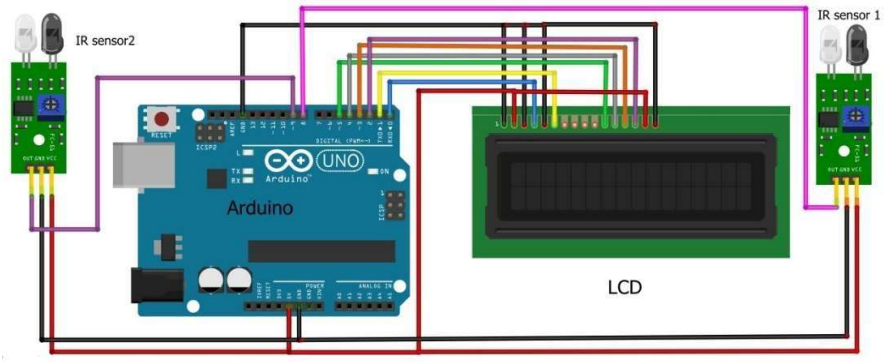

Figure1.1: Block diagram of car speed detector

One of the key advantages of this project is its versatility and scalability. The Arduino platform offers a wide range of compatible sensors and modules, enabling users to customize and expand the system according to their specific needs. For instance, the project can be enhanced to include data logging capabilities or wireless transmission of speed data to a remote monitoring station.By providing a cost-effective and accessible solution for car speed detection, this project contributes to the development of intelligent transportation systems. It can be utilized by traffic authorities, researchers, and even individual users to monitor and enforce speed limits, analyze traffic patterns, or conduct studies on road safety.

CHAPTER - 2

COMPONENTS

In this project we used different components that are required in Arduino Car Speed Detector.These components are easily available in market.

List of components:

- IR Sensor
- Buzzer
- Arduino Uno
- Preset
- LCD
- Bread Board
- Connecting wires

2.1 IR Sensor

An IR (Infrared) sensor is a type of electronic device that can detect infrared radiation. Infrared radiation lies in the electromagnetic spectrum between visible light and radio waves, with a wavelength longer than that of visible light.IR sensors work by utilizing a special material that reacts to infrared radiation. When exposed to IR radiation, the material generates a voltage or current proportional to the intensity of the detected infrared light. This change in voltage or current is then measured and interpreted by the sensor circuitry.

IR sensors are commonly used for various applications, including proximity sensing, object detection, motion detection, and distance measurement. They are particularly useful in scenarios where traditional sensors like ultrasonic sensors or optical sensors may not be suitable or reliable.

Figure 2.1

2.2 Buzzer

Buzzer is a common electronic component used in various projects and devices to generate audible signals or sounds. It is a type of transducer that converts electrical energy into sound energy. Buzzer can be found in alarm systems, timers, electronic games, musical instruments, and more.

Figure 2.2

2.3 Arduino Uno

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 supportthe microcontroller**.**

Figure 2.3

Arduino Uno features:

2.3.1 More Frequency and number of instructions per cycle.

2.3.2 Built in regulation.

2.3.3 Flexibility and Ease of use.

2.3.4 Configurable pins and Quick start.

2.3.5 Greater Flash memory.

2.3.6 Low voltage Requirement.

2.3.7 Plug and play.

2.3.8 USB interface.

2.3.9 Power alternatives.

Specifications:

2.3.10 Micro controller - ATmega38P- 8 bit microcontroller

2.3.11 Operating Voltage – 5V

2.3.12 Recommended Input Voltage – 7 to 12 V

2.3.13 Input Voltage limits – 6- 20 V

2.3.14 Analog Input pins - 6(A0 to A5)

2.3.15 Digital I/O pins – 14

2.3.16 DC current on I/O pins – 40mA

2.3.17 DC current on 3.3V pin – 50mA

2.3.18 Flash memory – 32KB

2.3.19 SRAM – 2KB

2.3.20 EEPROM – 1KB

2.3.21 Frequency – 16MHz

Pin Configuration:

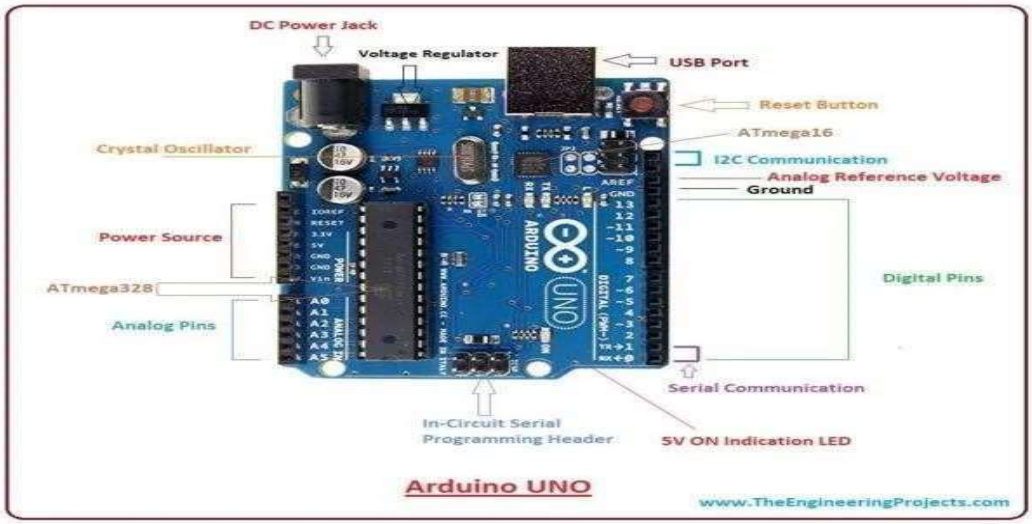

Figure 2.4

Pin Description:

1. LED - Arduino Uno comes with built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON. Providing LOW will turn it OFF.
2. Vin - It is the input voltage provided to the Arduino Board. This pin is used to supply voltage. It is different than 5 V supplied through a USB port. If a voltage is provided through power jack, it can be accessed through this pin.
3. 5V - This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack. USB supports voltage around 5V while Vin and Power Jack support a voltage ranges between 7V to 20V.
4. GND - These are ground pins. There are more than one ground pins are provided on the board. They can be used as per requirement.
5. Reset -This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.
6. IOREF -It is the abbreviation of Input Output Voltage Reference. This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then select the proper power source.
7. PWM – Pulse Width Modulation is provided by 3, 5, 6, 9, 10, 11 pins. These pins are configured to provide 8-bit output PWM.
8. SPI - It is abbreviation of Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of SPI library.
9. AREF - It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.
10. TWI - It is called Two-Wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.
11. Serial Communication - Serial communication is carried out through two pins called Pin 0 (Rx) and Pin 1 (Tx).
12. Rx. & Tx. - Rx (Receiver) pin is used to receive data while Tx (Transmitter) pin is used to transmit data.
13. External Interrupts - Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

2.4 Preset

Preset or trimmer potentiometers are small “set-and-forget” type potentiometers that allow for very fine or occasional adjustments to be easily made to a circuit, (e.g. for calibration).

Figure 2.5

2.5LCD

LCD (Liquid Crystal Display) is a type of flat panel display which uses liquid crystals in its primary form of operation. LEDs have a large and varying set of use cases for consumers and businesses, as they can be commonly found in smartphones, televisions, computer monitors and instrument panels.

LCDs were a big leap in terms of the technology they replaced, which include light-emitting diode (LED) and gas-plasma displays. LCDs allowed displays to be much thinner than cathode ray tube (CRT) technology. LCDs consume much less power than LED and gas-display displays because they work on the principle of blocking light rather than emitting it. Where an LED emits light, the liquid crystals in an LCD produces an image using a backlight.

A liquid crystal display or LCD draws its definition from its name itself. It is a combination of two states of matter, the solid and the liquid. LCD uses a liquid crystal to produce a visible image. Liquid crystal displays are super-thin technology display screens that are generally used in laptop computer screens, TVs, cell phones, and portable video games. LCD’s technologies allow displays to be much thinner when compared to a cathode ray tube (CRT) technology.

Figure 2.6

It consists of 14 pins. They are 8 data pins from D0 – D7 and three control pins such as RS,RW, and E. LED+ and LED- pins are used to control the backlight LED.

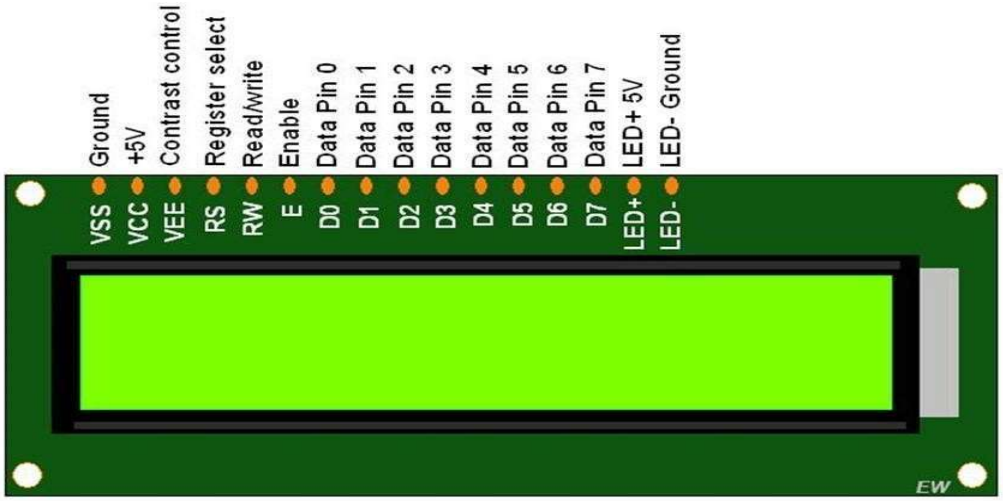

Figure 2.7

Pin configuration:

Pin1 : Ground – Connected to ground of power

Pin2 : VCC – Connected to supply pin of power source

Pin3:V0/VEE–Adjusts the contrast of LCD –Connected to variable POT that can source 0-5V

Pin 4: Register select – Connected to a MCU pin and gets either 0 or 1

0 -> Command mode

1 -> Data mode

Pin5 : Read/Write – Connected to a MCU and gets 0 or 1

0 -> Write operation

1 -> Read operation

Pin 6 : Enable – Connected to a MCU and always held high.

Pin 7 – 14 : Data bits – Pins used to send data to LCD

Pin 15 : LED positive – Connected to +5V

Pin 16 : LED negative – Connected to ground

Features :

- 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,

2.6Bread Board

A breadboard is used to make up temporary circuits for testing or to try out an idea. No soldering is required so it is easy to change connections and replace components. Parts are not damaged and can be re-used afterwards.

Breadboards have many tiny sockets (called 'holes') arranged on a 0.1" grid. The leads of most components can be pushed straight into the holes.

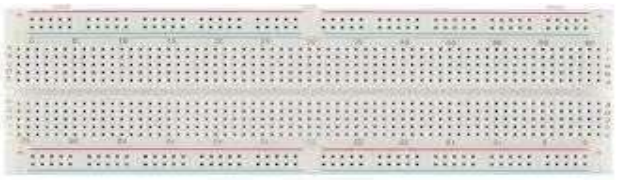

Figure 2.8

2.7 Connecting wires

Connecting wires are used to make connections between items on bread board and Arduino's header pins.Here mainly Jumper wires are used.Jumper wire are also connecting wires.

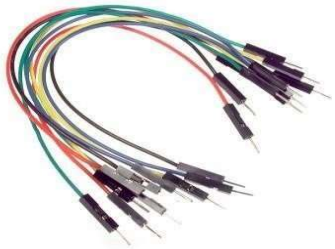

Figure 2.9

USB cable

Cable For Arduino UNO/MEGA is the most common A to B Male/Male type peripheral USB cable for Arduino. It is compatible with most Arduino boards such as Arduino Mega, Uno. Connect your Arduino, USB printer, scanner and more to your computer. Transmits data or program at high speeds with error-free, high-performance transmission.

Figure 2.10

Software Specifications:

Arduino programming can be used for the Arduino Uno . Arduini programs are written in the Arduino Integrated Development Environment (IDE). It is a special software running on your systems that allows you to write sketches for different Arduino boards.The Aduino programming language is based on a very simple hardware programming language like processing, which is similar to C language.

Steps for set up the Arduino IDE on the system and prepare the board to receive the program via USB cable.

Step 1: First we require the Arduino board and USB cable.

Step 2 : Download Arduino IDE software.

We can get different versions on Arduino official website . By selecting the software that is compatible to our operating system we can download the software. After file download unzip the file.

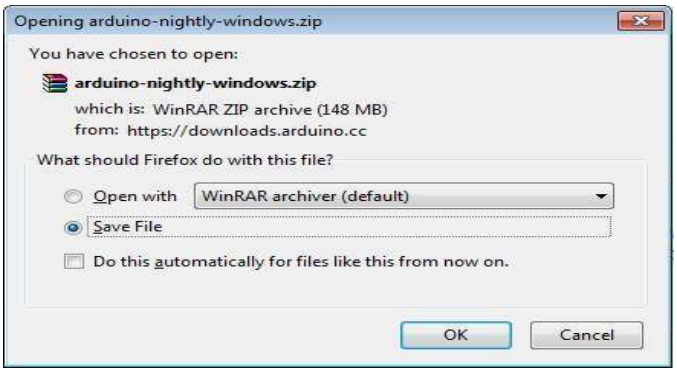

Figure 2.11

Step 3: power up your board

The Arduino Uno automatically draws power from either USB connection to the computer or an external power supply.Connect the Arduino board the system using the USB cable .the green LED should glow.

Step 4: Launch Arduino IDE

After Arduino IDE software is downloaded you need to Unzip the folder. Inside the folder you can find the application with infinity label.Double click to start IDE.

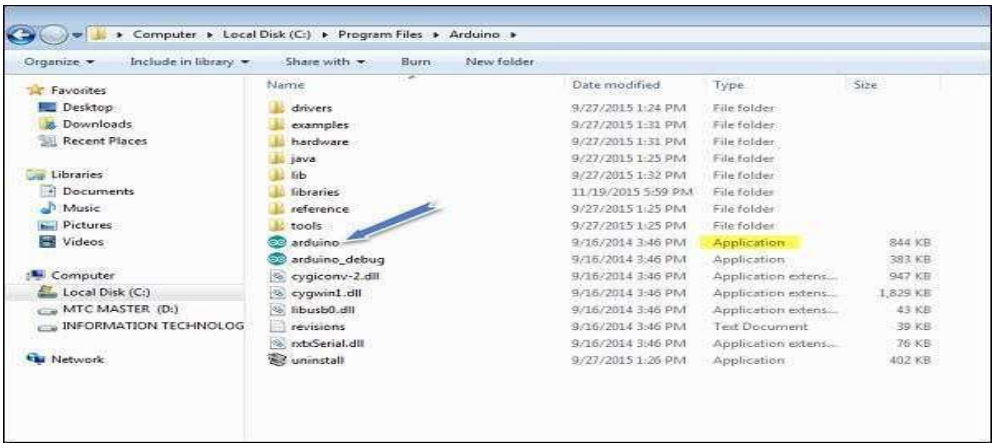

Figure 2.12

Step 5: Open your first project.

We get two options 1) create new project

2) Open an existing project.

- To create new project select File -> New
- To open an existing project

  Example,

  select File -> Examples -> Basics -> Blink

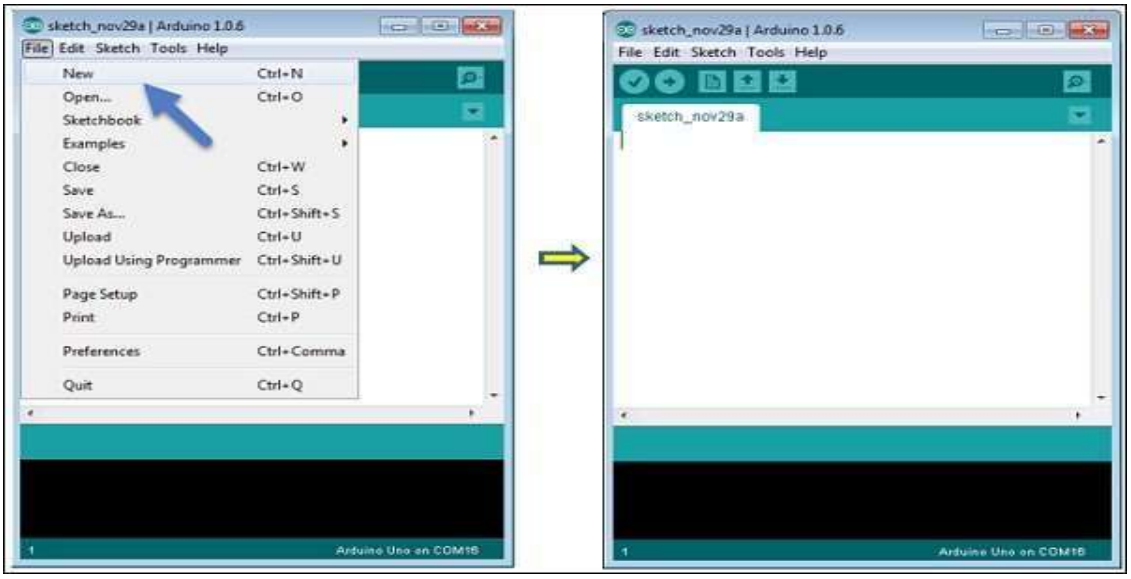

Figure2.13

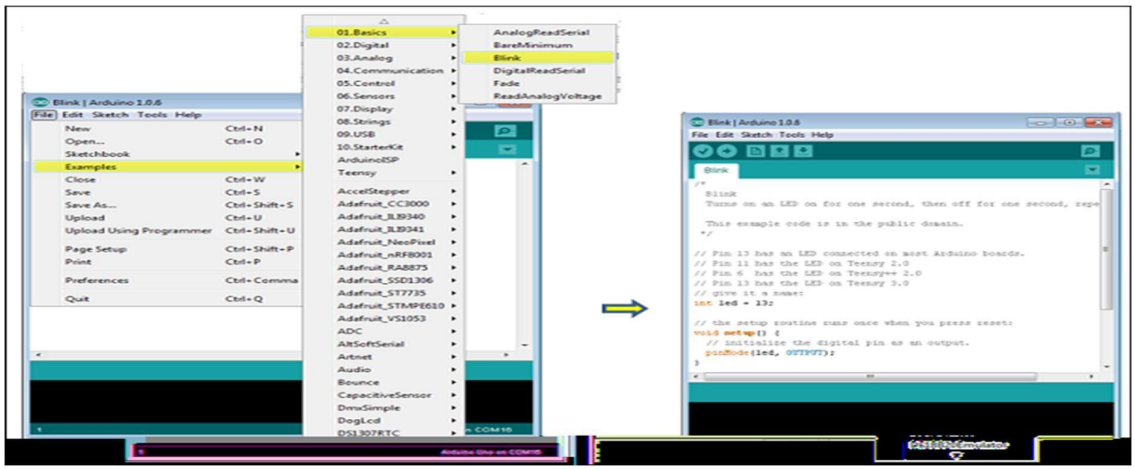

Figure 2.14

Step 6: Select your Arduino board

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer**.**

Go to Tools -> Board and select your Board

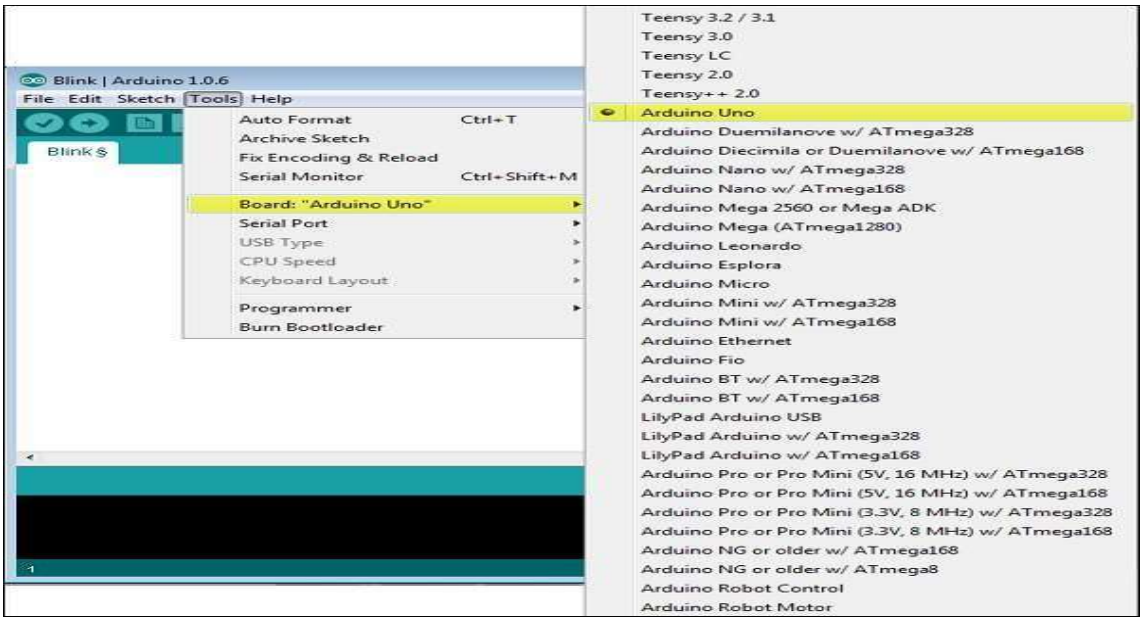

Figure 2.15

Step 7: Select your serial port

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

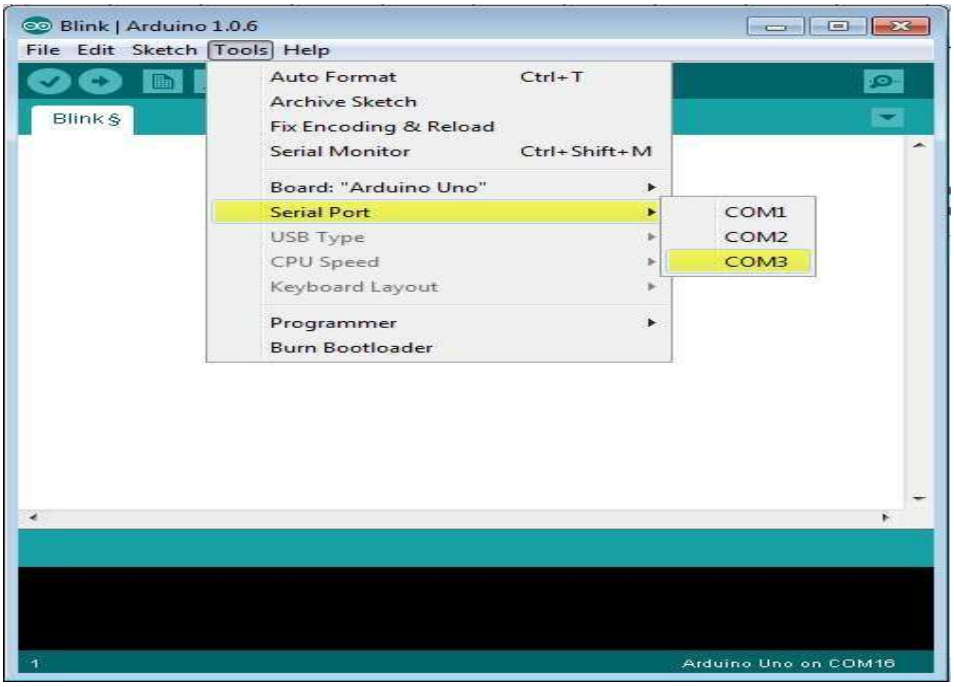

Figure 2.16

Step 8: Upload the program to your board

Figure 2.17

A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board**.**

CHAPTER -3

SCHEMATIC DIAGRAM

In this chapter we will see the circuit testing on the bread-board and working of the speed detector and Arduino in measuring its parameters. We have connected the circuit as per the circuit diagram using the components required. In this circuit the different pins of Arduino are connected to different elements. A0 and A1 pin in Arduino is connected to IRsensor, A1 pin is connected to resistors, 13 pin is connected to Buzzer. The pins of Arduino from 2 to 7 are connected to different pins on the LCD. IR, Buzzer, Resistors are connected on the breadboard.

The ground pin in Arduino is connected to ground on the breadboard and Vin pin is connected to the common pin on bread board which requires power supply. Solar panel has two pins mainly Ground and output pin. The ground pin is connected to Ground and Output pin is connected to the resistors from which all the connections are made.

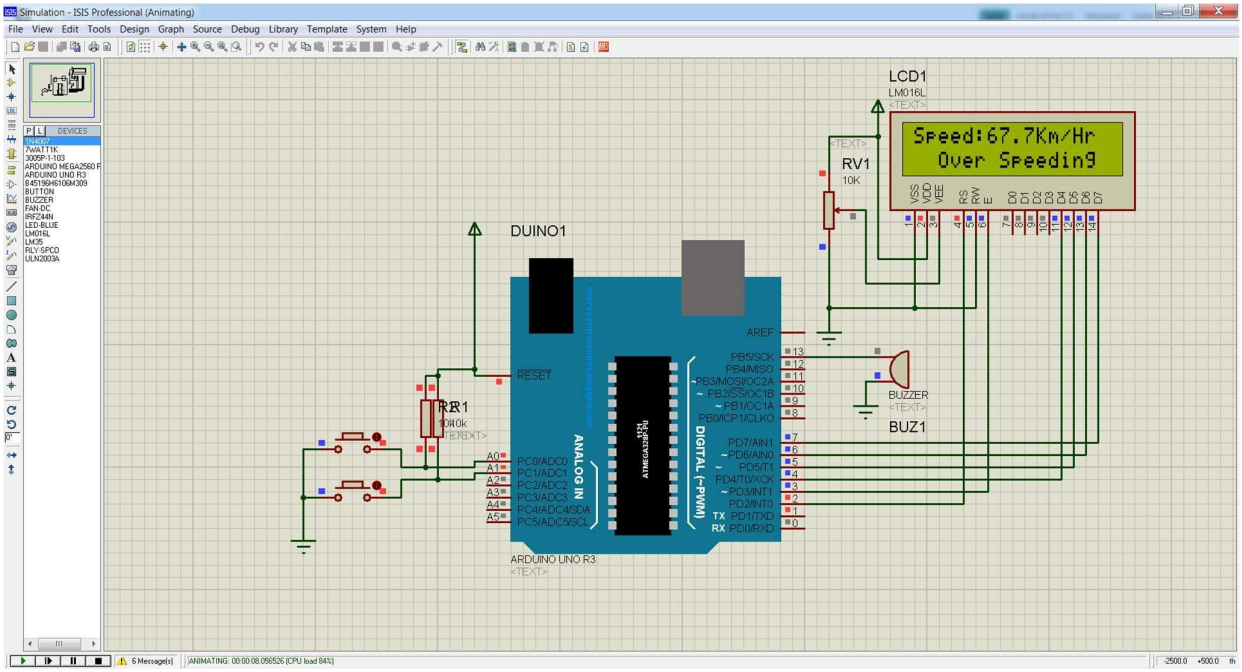

Figure 3.1

3.1 Working Principle

The circuit is used to display the parameters values on LCD screen. For this circuit to work, Arduino must be connected to the system or supply voltage with the required code using USB cable.

Here the two IR Sensors placed at a distance of 10cm (approximately).First we should move the vehicle from the IR Sensor1 to the IR sensor 2 then the Arduino continuously reads the inputs from the IR Sensors. When a car moving in front of the setup reaches the first sensor, Arduino becomes alert and capture a time stamp the moment the car leaves the first IR Sensor. Arduino then calculates the velocity by assuming the distance as 5 meters between the two IR Sensor and displays the result in kilometers per hour on the 16×2 LCD Display.

After the connection the circuit looks as follows:

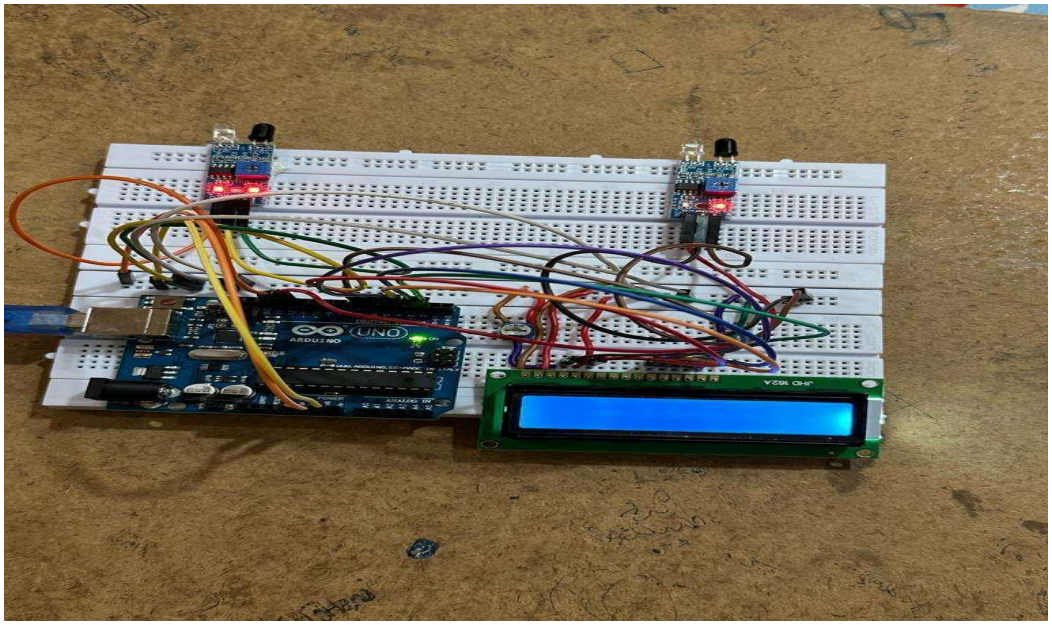

Figure 3.2

CHAPTER-4

SOURCE CODE

Arduino programs are written in the Arduino Integrated Development Environment (IDE). Arduino IDE is a special software running on your system that allows you to write sketches (synonym for program in Arduino language) for different Arduino boards. The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language. After the sketch is written in the Arduino IDE, it should be uploaded on the Arduino board for execution.

This programming contains mainly two functions setup() and loop() . setup() function is used to initialize the pin modes and starts serial communication. loop() function is used to execute a set of statements repeatedly.

Code:

```
#include<LiquidCrystal.h>

LiquidCrystal lcd(12, 11,5, 4, 3, 2);

int timer1;

int timer2;

float Time;

int flag1 = 0;

int flag2 = 0;

float distance = 5.0;

float speed;

int ir_s1 = 8;

int ir_s2 = 9;

int buzzer = 13;
```

```
void setup(){
  pinMode(ir_s1, INPUT);
  pinMode(ir_s2, INPUT);
  pinMode(buzzer, OUTPUT);
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
lcd.print(" SPEED DETECTOR");
  lcd.setCursor(0,1);
  lcd.print(" Using arduino");
  delay(2000);
  lcd.clear();
}
void loop() {
if(digitalRead (ir_s1) ==
LOW &&
flag1==0){timer1 =
millis(); flag1=1;}

if(digitalRead (ir_s2) ==
LOW &&
flag2==0){timer2 =
millis(); flag2=1;}
if (flag1==1 &&
flag2==1){
    if(timer1 >timer2)
```

```
{
Time = timer1 - timer2;
}
else if(timer2 > timer1)
{
Time = timer2 - timer1;
}
Time=Time/1000;//convert
millisecond to second
speed=(distance/Time);//v=d/t
speed=speed*3600;//multiply by seconds per hr
speed=speed/1000;//division by meters per Km
}
if(speed==0)
{
lcd.setCursor(0, 1);
if(flag1==0 && flag2==0)
{
lcd.print("No car  detected");
}
   else{
lcd.print("Searching...     ");
}
}
```

```
else{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Speed:");
    lcd.print(speed,1);
    lcd.print("Km/Hr ");
    lcd.setCursor(0, 1);
   if(speed < 60){
    lcd.print(" Normal Speed ");
  analogWrite(buzzer,300);
  delay(500);
  analog Write(buzzer, LOW);
    }
    else if(speed < 80){
    lcd.print(" Medium Speed ");
    digitalWrite(buzzer, HIGH);
    delay(1900);
    digitalWrite(buzzer, LOW);
    }
    else if(speed < 100){
    lcd.print(" Over Speed ");
    digitalWrite(buzzer, HIGH);
    delay(2500);
    digitalWrite(buzzer, LOW);
```

```
    }
    else{lcd.print(" Danger ");
    digitalWrite(buzzer, HIGH);
    delay(2993544);
    digitalWrite(buzzer, LOW);
    }
    delay(2000);
    digitalWrite(buzzer, LOW);
    speed = 0;
    flag1 = 0;
    flag2 = 0;
}
```

CHAPTER – 5

RESULTS AND DISCUSSIONS

The Arduino car speed detector using an IR sensor and a buzzer was implemented to detect and alert for excessive speeds of a moving vehicle. The IR sensor was used to measure the time taken for the vehicle to pass a certain distance, and the buzzer was activated if the measured speed exceeded a predetermined threshold.During testing, the Arduino speed detector effectively detected the presence of a vehicle using the IR sensor. The sensor was positioned across the road or a specific detection zone, and it emitted an infrared beam. When a vehicle passed through the beam, the sensor detected the interruption and triggered the Arduino to start the speed measurement.The reliability of the IR sensor was crucial for accurate speed measurements. The sensor had to be properly calibrated and positioned to ensure consistent and precise readings. Factors such as ambient light, sensor sensitivity, and distance between the sensor and the vehicle could impact the reliability of the measurements.

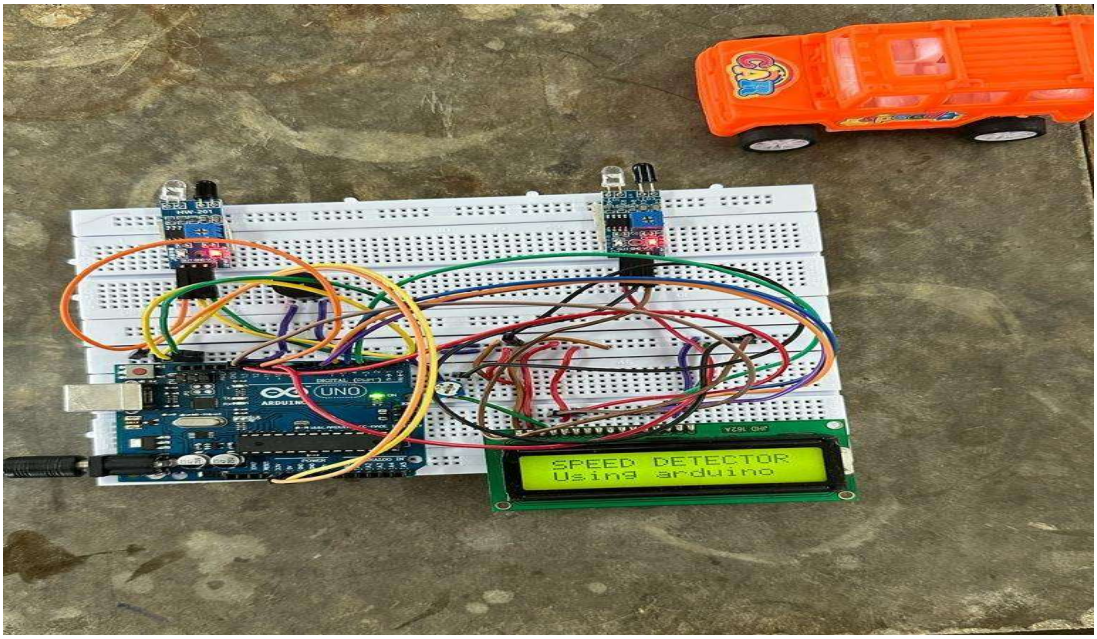

Figure 5.1

Careful consideration and adjustment of these factors were necessary to achieve reliable results.The buzzer played an important role in alerting for excessive speeds. Once the speed was calculated by the Arduino, it compared the measured speed with the predefined threshold. If the speed exceeded the threshold, the buzzer was activated to provide an audible warning.The effectiveness of the speed detection and alert system depended on the accuracy of the speed measurements and the responsiveness of the buzzer. The Arduino's programming played a crucial role in accurately calculating the speed based on the time measured by the IR sensor. The

threshold for excessive speed could be adjusted according to specific requirements or local speed limits.

Here the circuit has 4 different display mode as mentioned in code it splited into Normal Speed,Medium Speed,Over Speed,Danger.

The Normal Speed is displayed when the speed is between 0 to 60 Km/hr:

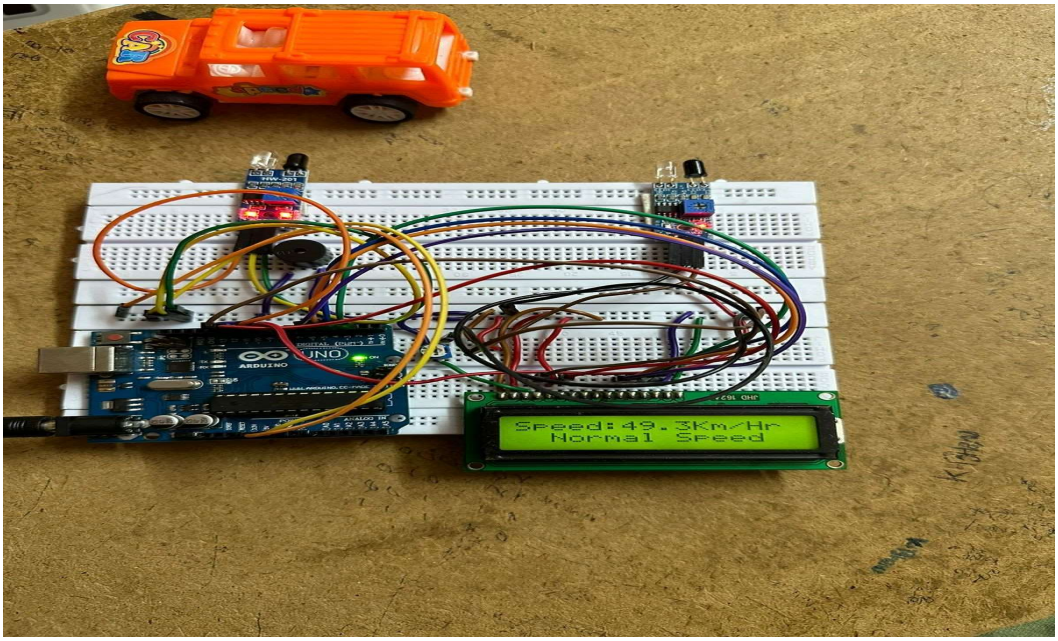

Figure 5.2

The Medium speed is displayed when the speed is between 60 to80 Km/hr:

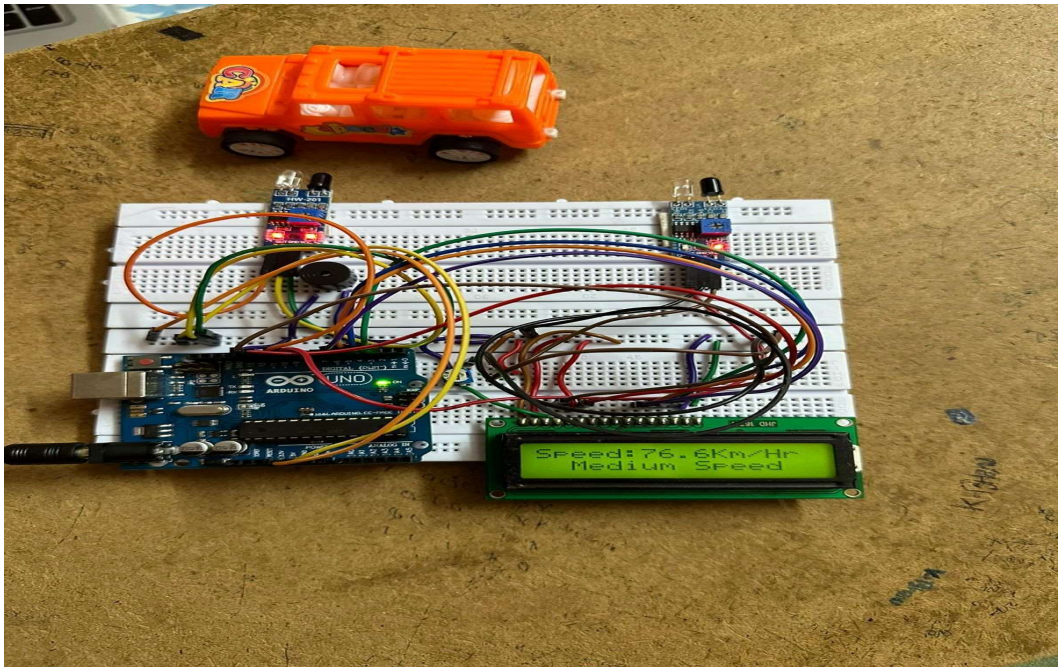

Figure 5.3

The Over speed is displayed when the speed is between 80 to100 Km/hr:

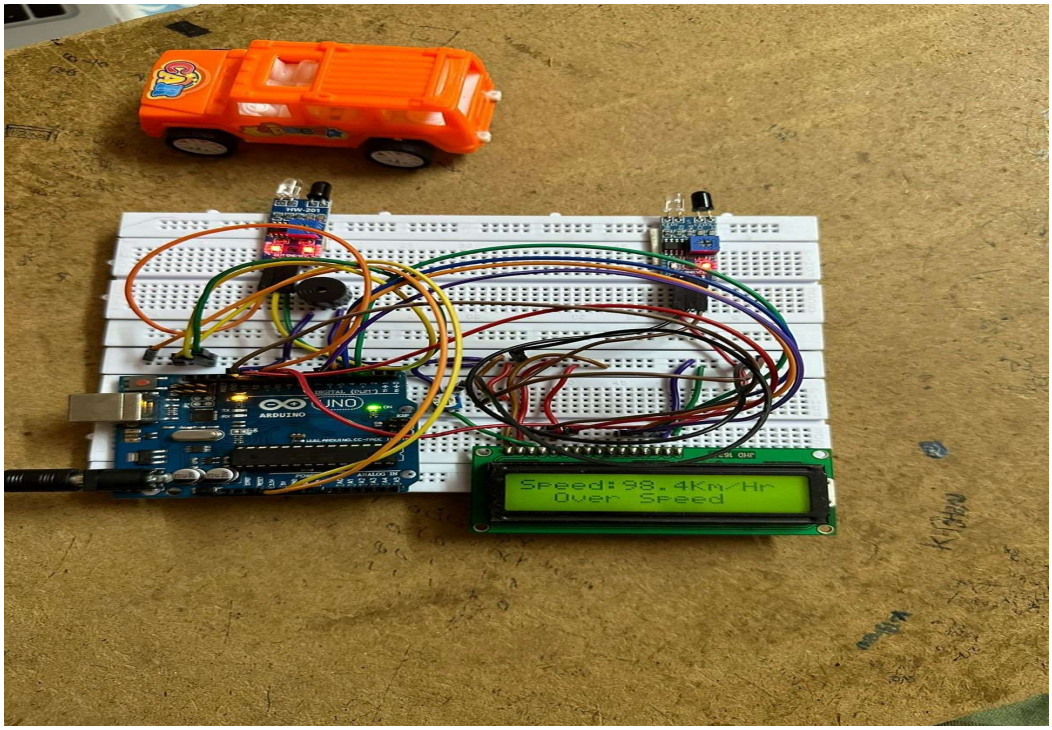

Figure 5.4

The Danger is displayed when the speed is above 100 Km/hr:

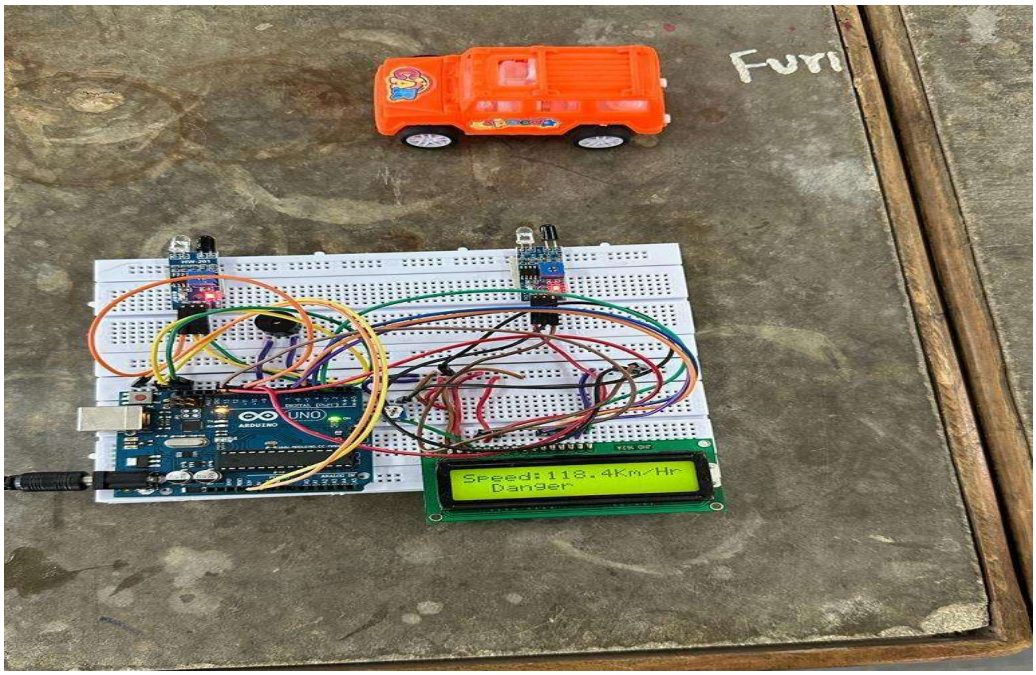

Figure 5.5

CHAPTER – 6

CONCLUSION AND FUTURE SCOPE

This project is about displaying speed of the vehicle in LCD with the help of Arduino. In conclusion, the Arduino car speed detector using IR sensors provides a reliable and cost-effective solution for measuring vehicle speed. By accurately detecting the interruption of an infrared beam caused by a passing vehicle, the system can calculate the speed based on the time taken for the interruption to occur.The IR sensor-based speed detector has proven to be effective in various applications, such as traffic monitoring, speed limit enforcement, and road safety initiatives. It offers a simple and accessible solution for measuring vehicle speed, requiring minimal hardware and programming complexity.

.FUTURE SCOPE:

The future scope for the Arduino car speed detector using IR sensors presents numerous opportunities for enhancements in accuracy, environmental adaptability, data analysis, integration with intelligent systems, and power efficiency. By exploring these possibilities, the system can continue to evolve and contribute to the improvement of road safety and traffic management efforts.

REFERENCES

1) https://www.electronicshub.org/arduino-car-speed-detector/#:~:text=%C3%972%20LCD.-,Working,leaves%20the%20first%20IR%20Sensor.
2) https://www.electroduino.com/car-speed-detection-car-speed-measurement-device/
3) https://www.electroniclinic.com/vehicle-speed-detection-or-vehicle-speed-measurement-using-ir-sensor-and-arduino/
4) https://www.circuitstoday.com/car-speed-detector-arduino
5) https://zadbuke-atharva18.medium.com/speed-detector-using-arduino-4812bb391bea
6) https://microdigisoft.com/how-to-make-a-car-speed-detector-using-arduino-and-ir-sensor-in-proteus/