

**Sukkur Institute of Business Administration University**

Department of Computer Systems Engineering

ELE-111: Computer Engineering Workshop, fall 2021

**Lab # 12: Car Speed Detector**

Instructor: Engr. Irfan Ali Babar

**Submission Profile**

Name: Abdul Basit Memon Submission Date: 20/2/2022

Enrolment ID: Receiving Authority Name and Signature:

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Instructor Signature

# Lab Learning Objectives:

After completing this session, student should be able to:

* Interface or displayed speed on LCD with Arduino UNO

# Lab Hardware and Software Required:

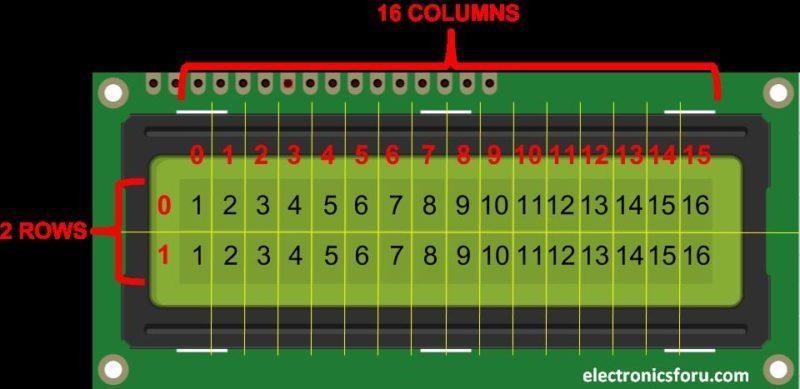
1. Arduino IDE
2. Desktop Computer/Laptop
3. Arduino UNO
4. Alphanumeric LCD,(16x2)
5. IR sensors
6. Buzzer
7. Resistors (330 Ω), LED, Breadboard, Connecting wires

# Theory:

## **ALPHANUMERIC LCD (16x2)**

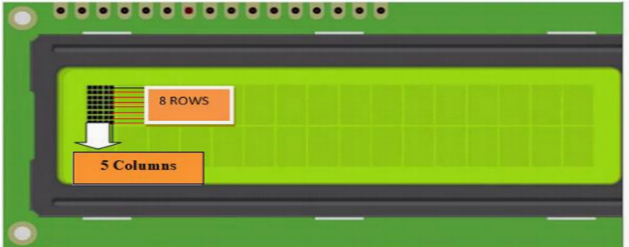
Liquid Crystal Display (LCD) is widely used these days due to declining prices, and ability to display numbers, characters, and graphics. LCD uses refreshing controller which relives the CPU job, in contrast LED must be refreshed by the CPU.

LCD stands for liquid crystal display. LCD's come in many sizes 8x1, 8x2, 10x2, 16x1, 16x2, 16x4, 20x2, 20x4, 24x2, 30x2, 32x2, 40x2 etc. In an MxN LCD, M denotes number of columns and n represents number of rows. Like if the LCD is denoted by 16x2 it means it has 16 columns and 2 rows as shown below.



*Figure 1: 16x2 Character LCD*

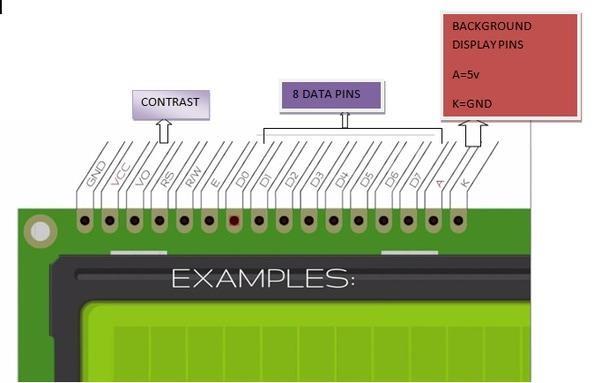
On a character LCD, a character is generated in a matrix of 5x8 or 5x7. Where 5 represents number of coulombs and 7/8 represent number of rows. Maximum size of the matrix is 5x8. You cannot display character greater then 5x8 dimension matrix. Normally we display a character in 5x7 matrix and left the 8th row for the cursor. If we use the 8th row of the matrix for the character display, then there will be no room for cursor. The picture below shows the 5x8 dot matrix pixels arrangement.



*Figure 2: Pixels arrangement in LCD character*

All character LCD's have

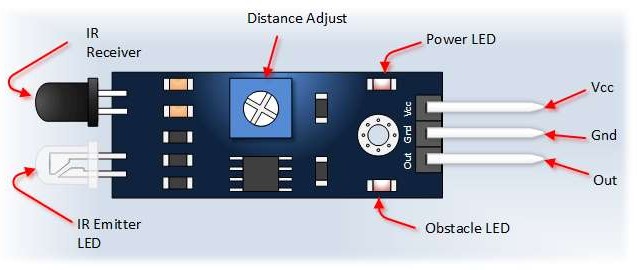
* Eight(8) data pins D0-D7
* Vcc (Apply +5 volt here)
* Gnd (Ground this pin)
* RS (Register select)
* Rw (read - write)
* En (Enable)
* V0 (Set LCD contrast)



*Figure 3: 16x2 Character LCD Pins Description*

Now, we know that each character **has (5×8=40) 40 Pixels** and for **32 Characters** we will have **(32×40) 1280 Pixels**. Further, the LCD should also be instructed about the Position of the Pixels. Hence it will be a hectic task to handle everything with the help of MCU, hence an **Interface IC like HITACHI HD44780** is used, which is mounted on the backside of the LCD Module itself. The function of this IC is to get the **Commands and Data** from the MCU and process them to display meaningful information onto our LCD Screen. We actually talk with this controller in order to display character on the LCD screen. HD44780 must be properly handled and initialized before sending any data to it. HD44780 has some registers which are initialized and manipulated for character displaying on the LCD. These registers are selected by the pins of character LCD.

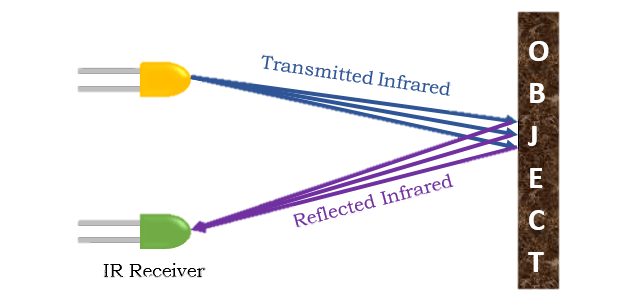
# Infrared Sensor (IR Sensor)

An [infrared sensor](https://www.elprocus.com/ir-remote-control-basics-operation-application/) is an electronic device that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it that is called as a sensor. The device consists of an Infrared Transmitter, an Infrared Detector (Receiver), and support circuitry. It only requires three connections. When it detects an obstacle within range it will send an output low. Infrared distance sensors are useful for measuring distances without touching a surface. The three wires protruding from a distance sensor represent +5V (in most cases), GND (Ground) and signal. These are almost always color coded with black as ground, red as +V and white or yellow as the signal.

*Figure 4: IR Sensor Module*

# Working Principle

Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These types of radiations are invisible to our eyes that can be detected by an infrared sensor. The emitter is simply an IR LED ([Light Emitting Diode](http://www.elprocus.com/explain-different-types-leds-working-applications-engineering-students/)) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and these output voltages, change in proportion to the magnitude of the IR light received.



*Figure 5: Working Principle of IR Sensor*

# BUZZER

An audio signaling device like a beeper or buzzer may be electromechanical or [piezoelectric](https://www.elprocus.com/what-is-a-piezoelectric-material-working/) or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.



*Figure 6: Buzzer Pin Configuration*

The **pin configuration of the buzzer** is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the ‘+’ symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the ‘-‘symbol or short terminal and it is connected to the GND terminal.

# Working principle of the project

When supply power to Arduino LCD displays **“TUSHAR KANJWANI”** on screen at **(0,0)** **“BASIT MEMON”** at **(3,1)** and after a second it indicates message “**No Car Detected**” on lcd screen. When a car/object is cross the first sensor then it save time1 of car/object and the LCD displays “**Searching**” on screen, and when out from the second sensor it also detect the time2 then higher time minus from lower it give total time then distance (5km) divide the total time which give speed (km/s) then it will convert into (km/h) by using formula with that speed the LCD indicate the speed which is display on lcd screen. The distance between sensors is approximately 10 meters. Now checked with fast moving car or object cross the sensors it indicates speed with (km\h) unit. When speed of car/object is less than 50 km\h the LCD screen displays speed as well as indicate message of “**Normal speed**”. When speed is high and up to 50 km\h the LCD indicate the message of “**over speeding**” with the buzzer alarm for 3 seconds. The speed limit is considered and initialize in code.

|  |
| --- |
| CODE #include<LiquidCrystal.h>  LiquidCrystal lcd(2, 3, 4, 5, 6, 7);  int timer1; // declare time1  int timer2; // declare time2  float Time; // declare time  int flag1 = 0; // declare flag1  int flag2 = 0; // declare flag2  float distance = 5.0; // declare distance  float speed; // declare speed  int ir\_s1 = A0; // declare IR1 pin  int ir\_s2 = A1; // declare IR2 pin  int buzzer = 13; // declare buzzur pin  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(" TUSHAR KANJWANI ");  lcd.setCursor(3,1);  lcd.print("BASIT MEMON");  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 > 100){  lcd.print(" Over Speeding "); digitalWrite(buzzer, HIGH);}  else{lcd.print(" Normal Speed "); }  delay(3000);  digitalWrite(buzzer, LOW);  speed = 0;  flag1 = 0;  flag2 = 0;  }  } |
|  |

# RESULT

**A picture containing text, electronics

Description automatically generated**

*FIGURE 7: PROJECT RESULT*

# REFERENCE

**Link:**

https://create.arduino.cc/projecthub/embeddedlab786/car-speed-detectord60ea0?ref=tag&ref\_id=arduino&offset=137