Course Name: STEM 1 LAB

My section:?

Assignment Name: Assignment 1

Name: Abdallah Hamed Faleh Daradkeh

My ID: 21110446

My Tutor’s Name: Mohammad AlZubi



# Task 1

Coding:

The program is an infinite loop which take sensors reading from the user and determine the speed of fan based on sensors’ readings and store all information in an array structured

The code:

{{{{

while(1)%infinite loop

%Task 1

%Declare sensors tempreature readings

sensor1=0;

sensor2=0;

sensor3=0;

fan\_speed=0; % the speed of fan

data=0;

% Take temerature readings from the sensors

sensor1=input('sensor 1 Reading: ');

sensor2=input('sensor 2 Reading: ');

sensor3=input('sensor 3 Reading: ');

%calculate the avg of the sensors readings

avg=(sensor1+sensor2+sensor3)/3;

%Now The speed will determine based on the avg readins of sensors

if(avg>0 && avg<=40) %condition 1

fan\_speed=0.25;

elseif avg>40 && avg<=55 %condition 2

fan\_speed=0.35;

elseif avg>55 && avg<=70 %condition 3

fan\_speed=0.65;

else

fan\_speed=1.0; % if no condition fit

end

%store readings of the tempreature and fan speed in a matrix:

%All the stored data are in a matrix structured

data=['sensor1= ' num2str(sensor1),' sensor2= ' num2str(sensor2),' sensor3= ' num2str(sensor3),' avg= ' num2str(avg),' fan\_speed= ' num2str(fan\_speed)]

end

}}}}}}]]]]

Flowchart: A flowchart describe the program inputs/outputs/processes/start and end. Diagram

Description automatically generated

Infinite program

# Task 2

2-A)

Plot iL vs t per equation1.

This is the first equation representation in excel

|  |  |
| --- | --- |
| t | i(L)=(0.678\*sin(1.3227\*t\*(PI()/180))+\*cos(1.3227\*t\*(PI()/180)))\*e^-0.599\*t |
| 1 | 0.557812143 |
| 2 | 0.310919654 |
| 3 | 0.173176921 |
| 4 | 0.096387815 |
| 5 | 0.05361083 |
| 6 | 0.0297981 |
| 7 | 0.01655149 |
| 8 | 0.009187648 |
| 9 | 0.005096782 |
| 10 | 0.002825646 |
| 11 | 0.001565575 |
| 12 | 0.000866901 |
| 13 | 0.000479743 |
| This is the graph of the first equation.(drawing in excel)   |  |  | | --- | --- | | |  | | --- | |  | | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  |

|  |
| --- |
| 2-B)Steady:17.8,maximum:20.8, minimum:21 |
| time to reach within 1% of its final maximum value = 20.8 |
| time to reach within 1% of its final minimum value = 21 |
| time to reach within 1% of its final steady state value = 17.8 |
| How I found the previous times(values)>> I have multiple (maximum value by 1%) Then make the time  far than the range (0-13) because the 1% of the values is located far than this range for example:  maximum value 0.557512143\*1%=0.00557512143 so the time is 20.8.  How I found the time I make the time range each unit(0.2)  so it become(13,13.2,13.4.....22)so the number 21.8 is the most correct estimation and so on the rest times. |

I make the following in excel

|  |  |
| --- | --- |
| t | i(L)=(0.678\*sin(1.3227\*t\*(PI()/180))+\*cos(1.3227\*t\*(PI()/180)))\*e^-0.599\*t |
| 1 | 0.557812143 |
| 1.2 | 0.496301875 |
| 1.4 | 0.441561131 |
| 1.6 | 0.392846424 |
| 1.8 | 0.349495745 |
| 2 | 0.310919654 |
| 2.2 | 0.276593345 |
| 2.4 | 0.246049573 |
| 2.6 | 0.218872356 |
| 2.8 | 0.194691365 |
| 3 | 0.173176921 |
| 3.2 | 0.154035545 |
| 3.4 | 0.137005993 |
| 3.6 | 0.121855722 |
| 3.8 | 0.108377741 |
| 4 | 0.096387815 |
| 4.2 | 0.085721963 |
| 4.4 | 0.07623424 |
| 4.6 | 0.067794755 |
| 4.8 | 0.06028791 |
| 5 | 0.05361083 |
| 5.2 | 0.047671966 |
| 5.4 | 0.04238985 |
| 5.6 | 0.037691987 |
| 5.8 | 0.03351387 |
| 6 | 0.0297981 |
| 6.2 | 0.026493606 |
| 6.4 | 0.023554947 |
| 6.6 | 0.020941692 |
| 6.8 | 0.018617874 |
| 7 | 0.01655149 |
| 7.2 | 0.014714071 |
| 7.4 | 0.013080291 |
| 7.6 | 0.011627619 |
| 7.8 | 0.010336014 |
| 8 | 0.009187648 |
| 8.2 | 0.008166661 |
| 8.4 | 0.00725895 |
| 8.6 | 0.006451966 |
| 8.8 | 0.005734552 |
| 9 | 0.005096782 |
| 9.2 | 0.00452983 |
| 9.4 | 0.004025843 |
| 9.6 | 0.003577842 |
| 9.8 | 0.003179616 |
| 10 | 0.002825646 |
| 10.2 | 0.002511019 |
| 10.4 | 0.002231371 |
| 10.6 | 0.001982819 |
| 10.8 | 0.00176191 |
| 11 | 0.001565575 |
| 11.2 | 0.001391086 |
| 11.4 | 0.001236014 |
| 11.6 | 0.001098202 |
| 11.8 | 0.000975733 |
| 12 | 0.000866901 |
| 12.2 | 0.00077019 |
| 12.4 | 0.000684251 |
| 12.6 | 0.000607888 |
| 12.8 | 0.000540034 |
| 13 | 0.000479743 |
| 13.2 | 0.000426173 |
| 13.4 | 0.000378576 |
| 13.6 | 0.000336287 |
| 13.8 | 0.000298715 |
| 14 | 0.000265335 |
| 14.2 | 0.000235679 |
| 14.4 | 0.000209333 |
| 14.6 | 0.000185928 |
| 14.8 | 0.000165137 |
| 15 | 0.000146666 |
| 15.2 | 0.000130259 |
| 15.4 | 0.000115685 |
| 15.6 | 0.000102739 |
| 15.8 | 9.12393E-05 |
| 16 | 8.10252E-05 |
| 16.2 | 7.1953E-05 |
| 16.4 | 6.38951E-05 |
| 16.6 | 5.67384E-05 |
| 16.8 | 5.03821E-05 |
| 17 | 4.4737E-05 |
| 17.2 | 3.97234E-05 |
| 17.4 | 3.52709E-05 |
| 17.6 | 3.13168E-05 |
| 17.8 | 2.78054E-05 |
| 18 | 2.46872E-05 |
| 18.2 | 2.19181E-05 |
| 18.4 | 1.94592E-05 |
| 18.6 | 1.72758E-05 |
| 18.8 | 1.53371E-05 |
| 19 | 1.36156E-05 |
| 19.2 | 1.20871E-05 |
| 19.4 | 1.07299E-05 |
| 19.6 | 9.52493E-06 |
| 19.8 | 8.45509E-06 |
| 20 | 7.50525E-06 |
| 20.2 | 6.66196E-06 |
| 20.4 | 5.9133E-06 |
| 20.6 | 5.24866E-06 |
| 20.8 | 4.65863E-06 |
| 21 | 4.13483E-06 |
| 21.2 | 3.66985E-06 |
| 21.4 | 3.25708E-06 |
| 21.6 | 2.89068E-06 |
| 21.8 | 2.56545E-06 |
| 22 | 2.27675E-06 |

To find that the correct estimate time is 20.8.

And the rest is the same

2-C)

This is the representation of the new equation on the excel

|  |  |  |
| --- | --- | --- |
| t | i(Lnew)(t)=-0.0774\*(e^(-5.277\*t))+1.0774\*(e^(-0.379\*t)) |  |
| 1 | 0.737134105 |  |
| 2 | 0.504870525 |  |
| 3 | 0.345608275 |  |
| 4 | 0.236584635 |  |
| 5 | 0.161952974 |  |
| 6 | 0.110864198 |  |
| 7 | 0.0758916 |  |
| 8 | 0.051951262 |  |
| 9 | 0.035563008 |  |
| 10 | 0.024344501 |  |
| 11 | 0.016664922 |  |
| 12 | 0.011407899 |  |
| 13 | 0.007809227 |  |

The graph

|  |  |
| --- | --- |
| |  | | --- | |  | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

2-D)

|  |
| --- |
| 2-D) iL needs approximately 6 seconds, while iLnew nees 9 seconds to reach to within 5% of its maximum value.  So iL equation is better because it take less time to reach |
| time to reach within 5% of its final maximum value(the first) = 6.4s |
| time to reach within 5% of its final maximum value(the second) = 9s |
| I find the times in the same way I find it to the previous part(3-c),  but this time was during the range (0-13, and this time I make the same thing the unit =0.2, so (1,1.2,1.4…).  The first equation is better in reach to the 5% of its maximum value because it is need a little time to reach,  the first equation need (6.4s), while the second needs(9s). So the first is better. |

I make the following on the excel.

|  |  |
| --- | --- |
| t | i(L)=(0.678\*sin(1.3227\*t\*(PI()/180))+\*cos(1.3227\*t\*(PI()/180)))\*e^-0.599\*t |
| 1 | 0.557812143 |
| 1.2 | 0.496301875 |
| 1.4 | 0.441561131 |
| 1.6 | 0.392846424 |
| 1.8 | 0.349495745 |
| 2 | 0.310919654 |
| 2.2 | 0.276593345 |
| 2.4 | 0.246049573 |
| 2.6 | 0.218872356 |
| 2.8 | 0.194691365 |
| 3 | 0.173176921 |
| 3.2 | 0.154035545 |
| 3.4 | 0.137005993 |
| 3.6 | 0.121855722 |
| 3.8 | 0.108377741 |
| 4 | 0.096387815 |
| 4.2 | 0.085721963 |
| 4.4 | 0.07623424 |
| 4.6 | 0.067794755 |
| 4.8 | 0.06028791 |
| 5 | 0.05361083 |
| 5.2 | 0.047671966 |
| 5.4 | 0.04238985 |
| 5.6 | 0.037691987 |
| 5.8 | 0.03351387 |
| 6 | 0.0297981 |
| 6.2 | 0.026493606 |
| 6.4 | 0.023554947 |
| 6.6 | 0.020941692 |
| 6.8 | 0.018617874 |
| 7 | 0.01655149 |
| 7.2 | 0.014714071 |
| 7.4 | 0.013080291 |
| 7.6 | 0.011627619 |
| 7.8 | 0.010336014 |
| 8 | 0.009187648 |
| 8.2 | 0.008166661 |
| 8.4 | 0.00725895 |
| 8.6 | 0.006451966 |
| 8.8 | 0.005734552 |
| 9 | 0.005096782 |
| 9.2 | 0.00452983 |
| 9.4 | 0.004025843 |
| 9.6 | 0.003577842 |
| 9.8 | 0.003179616 |
| 10 | 0.002825646 |
| 10.2 | 0.002511019 |
| 10.4 | 0.002231371 |
| 10.6 | 0.001982819 |
| 10.8 | 0.00176191 |
| 11 | 0.001565575 |
| 11.2 | 0.001391086 |
| 11.4 | 0.001236014 |
| 11.6 | 0.001098202 |
| 11.8 | 0.000975733 |
| 12 | 0.000866901 |
| 12.2 | 0.00077019 |
| 12.4 | 0.000684251 |
| 12.6 | 0.000607888 |
| 12.8 | 0.000540034 |
| 13 | 0.000479743 |

And I find the two times and compare between them to decide which equation is the best.

2-E)

%plot the two equations

t=1:1:13; %t values from(1-13)

%iL equation

iL=exp(-0.599.\*t).\*(0.678.\*sin((pi()/180).\*1.3227.\*t)+cos((pi()/180).\*1.3227.\*t));

plot(t,iL) %plot the first equation

hold on % to plot both equations in the same graph

iLnew=-0.0774.\*exp(-5.227.\*t)+1.0774.\*exp(-0.379.\*t); % the second equations

plot(t,iLnew) %plot the second equation

xlabel('x Axis')% label x axis

ylabel('y Axis')%label y axis

grid on% add grid lines to the graph

title('t vs iL & iLnew')%add title to the graph

This is the graphChart, line chart

Description automatically generated

# Task 3:

%Task3 3-A)

%matrix represent 5 types of batteries over 7 years

%each row is for one year

%each column for one type of batteries

clc;clear all;close all;

A=[75 144 114 102 108;90 126 102 84 126;96 114 75 105 135;105 90 150 90 75;90 75 135 75 90;105 60 165 45 120;115 85 160 100 145]

%Dimentions of A is

Dimentions=size(A)

%Dimentions for A is [7,5]

%Task3 3-B)

%store the years 1994,1995,1996 of 32AH battery in a row vector(array have one row, it is not matter how columns it has)

B=[75 150 135]

%Task3 3-C)

%column vector(array has one column, it is not matter how row it has) total sales of each year(sum all sales in each year)

C=[543;528;525;510;465;495;605]

%Taske3 3-D)

%column vector(one column, it is not matter how row it has)average of sales of each year(number of sales on each year)/(number of batteries types)

D=[108.6;105.6;105;102;93;99;121]

%Task3 3-E)

%Add 10 to each element in the matrix which it less than 90 over the given years

for i=1:length(A) %loop to go through all rows loops to search every element on the array

for j=1:width(A) %loop to go through all columns

if A(i,j)<90 %condition

A(i,j)=A(i,j)+10; %add 10 to each sales less thatn 90(each round)

end

end

end

%Task3 3-F)

%substract 6 from number of sales of 35Ah battery over the given years

%each year you must substract 6 of this battery type sales

for i=1:length(A) %loop to go through all rows %loops to search go through each elements on the array

for j=4 %loop to go through all columns

A(i,j)=A(i,j)-3 %each element substract 3 from

end

end

%Task3 3-G)

%

for i=1:length(A) %loop to go through all rows

for j=3 %loop to go through all the columns in row three

A(i,j)=A(i,j).\*0

end

end

%Task3 3-H)

%5 curves on the same graph

%x- axis is the year

%y-axis is the sales

%each curve corresponds to one battery

x=1992:1:1998;

%The type of batteries

%but I put the number in the end(cannot put it in the beginning(error))

AH4=[75;90;96;105;90;105;115];

AH7=[144;126;114;90;75;60;85];

AH32=[114;102;75;150;135;165;160];

AH35=[102;84;105;90;75;45;100];

AH55=[108;126;135;75;90;120;145];

plot(x,AH4,'blue') %draw the graph

hold on %To connect the previous graph with the next graph

plot(x,AH7,'red') %draw the graph

hold on %To connect the previous graph with the next graph

plot(x,AH32,'green')%draw the graph

hold on %To connect the previous graph with the next graph

plot(x,AH35,'black')%draw the graph

hold on%To connect the previous graph with the next graph

plot(x,AH55,'magenta')%draw the graph

legend('4AH','7AH','32AH','35AH','55AH')%put the distributers(key of the graph)(name of graphs/lines)

xlabel('years')%label to x-axis

ylabel('Sales')%label to y-axis

grid on%add grid lines

title('Batteries sales Over the years')%add title to the graph

This is the graph

Chart, line chart

Description automatically generated

# Task 4

Two Aurdino projects:

1**) The first project in Aurdino** is to build a circuit using LDR sensor, that LDR sensor can read the intensity of light. And a buzzer will be added which will change his tone based on the intensity of light, so if there is no light and it is completely dark then: high tone level. If it is dark the tone level will be medium. If the light is normal, the tone will be low. If we turn the flashlight on it, the buzzer will turn off.

The purpose of the project is: to make the buzzer notify us about the intensity level of the light. So, The buzzer notify us wherever the light is normal or high or low or when there is no light.

Components: Buzzer, LDR sensor, Wires, Arduino Uno Board, Breadboard, Resistor.

Procedural steps:

* Write the code on the Aurdino IDE, and this is contain the following:

1) Choose from the tools menu( The type of Aurdino Board>Uno)

2) Write the code that represents the process to upload it to the aurdino

.

* Connect the wires and LDR sensor and Buzzer on the Aurdino breadboard and the Aurdino Uno board. And this is as the following:
* Follow the picture below the instructions for more explain

1) Connect the Buzzer and the LDR sensor on the Aurdino Breadboard as you can see in the picture below.

2) Connect a wire between the Buzzer short leg and between the ground, it is an output from the Buzzer to port 9(PWM) and you can put it in other PWM ports, the port is PWM to enable us to control the analog signals, so change the voltage from (0v) to (5v). On the digital write you will got just only (0v) or (5v) but in the analog write you will got a values between them. And this is important because we need different value to control the buzzer tone( not always high or low).

3) Connect a resistor between the LDR and the ground to limit the protect the LDR from burning. Then we connect a wire from the LDR input to the analog ports for example (A0), so it can control value of voltage from (0-5v) based on the analog read of the LDR not always high or low.

4) Connect the other leg to the LDR to the 5v port, so the 5v port get the board 5v. The 5v port get the LDR 5v power.

5) The final step is to link the Aurdino Uno Board through USB to computer.

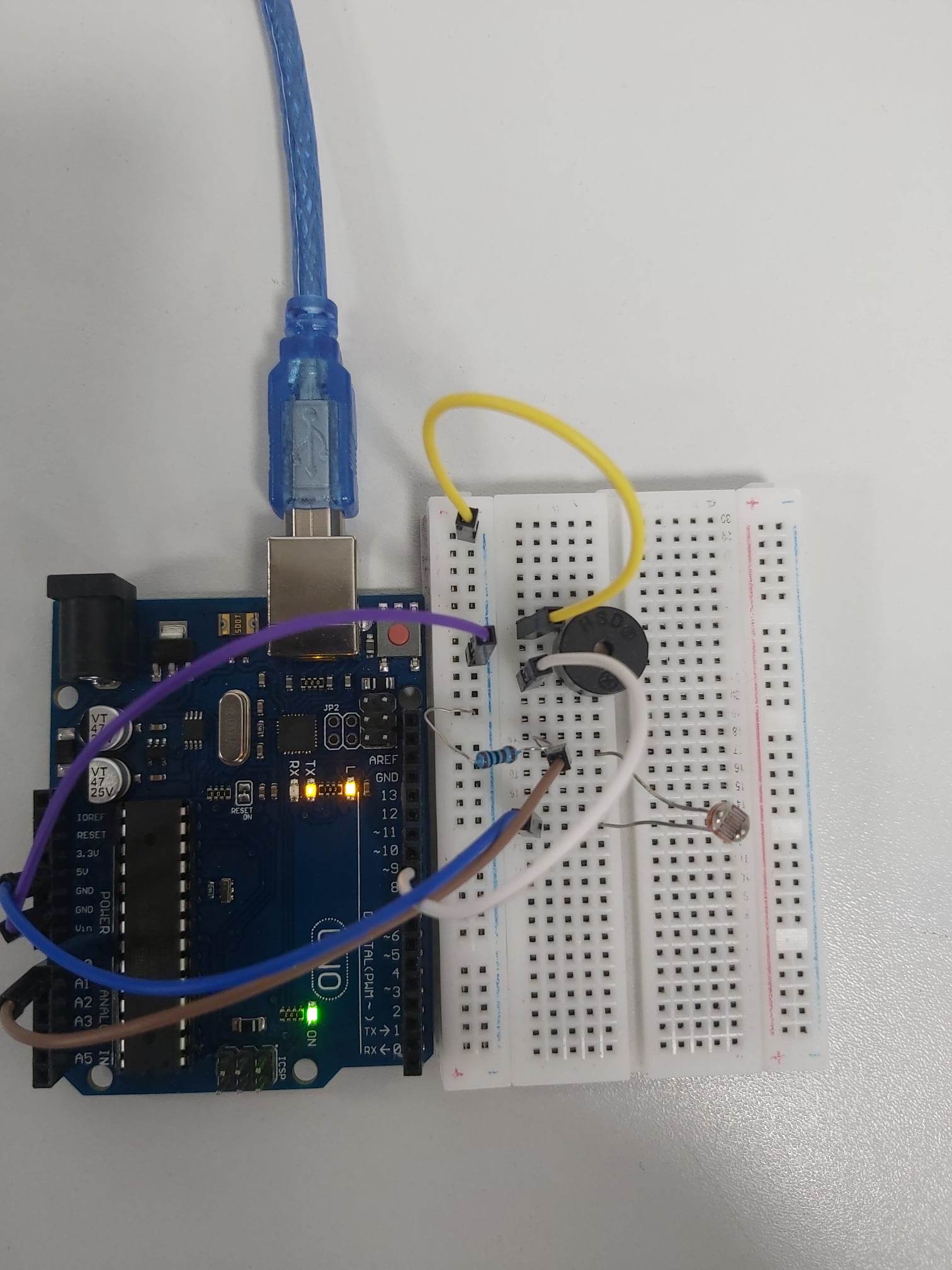
6) After connecting the Aurdino to the computer, define the port from the IDE, then upload the code to be implemented by aurdino.

7) Test different cases on the LDR to check its functionality.

* Important note: to connect wires to GND I used two methods:

1) connect wires to the GND into the aurdino uno board.

2) Connect the GND from the aurdino uno Board to one of the pins on the (-) field, and I link other wires to it.



The code part:

int LDR = A0; %the LDR in the analog pin A0

int BUZZER=9; %the Buzzer in pin 9

void setup()

{

  pinMode(LDR, INPUT); %The LDR is input

  Serial.begin(9600);

  pinMode(BUZZER, OUTPUT); % the buzzer is output

}

void loop()

{

  int LDR\_Status = analogRead(LDR); %analog read to the LDR

  Serial.print("The Reading of LDR is : "); %print

  Serial.println(LDR\_Status); %print ligt intensity

  delay(1000);

   if (LDR\_Status<300)

  {

    digitalWrite(BUZZER,5); %buzzer high tone

  }

  else if(LDR\_Status<550 & LDR\_Status>300)

  {

        digitalWrite(BUZZER,2.5); %buzzer medium tone

  }

  else if(LDR\_Status<750 & LDR\_Status>700)

  {

        digitalWrite(BUZZER,0); %buzzer off

  }

  else if(LDR\_Status>755)

  {

        digitalWrite(BUZZER,1.5); %buzzer low tone

  }

}

**The LDR is very useful in realistic project, since, it can be used on the Automatic Street Light. So it can turn of and on lights automatically based on the intensity of light. And the Buzzer is very useful to get a notification about a condition to take some procedure, in the cars alarms, when the car reverse back. The buzzer is very useful to know how far the car is about from the other obstacles behind it. And the resistor is very useful, because it prevent elements connected with from being burnt and damaged. In the cables which plugged in, the resistor is very useful, which protect the device of being damaged. The wires are very useful to build the circuit and connect elements to Aurdino Breadboard and Aurdino Uno Board. The wires are the base element to connect any elements together, so we can build the circuits. Every project contain wires. For example: Automatic street light, it contains wires which is to build the circuit.**

**2) The second project is to build a circuit that reads the distance of things that goes in front of the Ultrasonic sensor, and a tone is made by the buzzer is triggered based on the distance between the object and the Ultrasonic sensor.**

**Project purpose: The project purpose is to build a circuit to read the distance between the ultrasonic and the object and while the object is going a buzzer is made a note based on the distance between the object and the ultrasonic.**

**Components: Ultrasonic, wires, Buzzer, Aurdino Uno Board, Breadboard, USB.**

**Procedural steps:**

* Write the code on the Aurdino IDE, and this is contain the following:

1) Choose from the tools menu( The type of Aurdino Board>Uno)

2) Write the code that represents the process to upload it to the aurdino

* **Connect the wires and Ultrasonic and Buzzer to the AUrdino Uno Board and Breadboard as the following.**
* **Follow the picture below the instructions for more explain.**

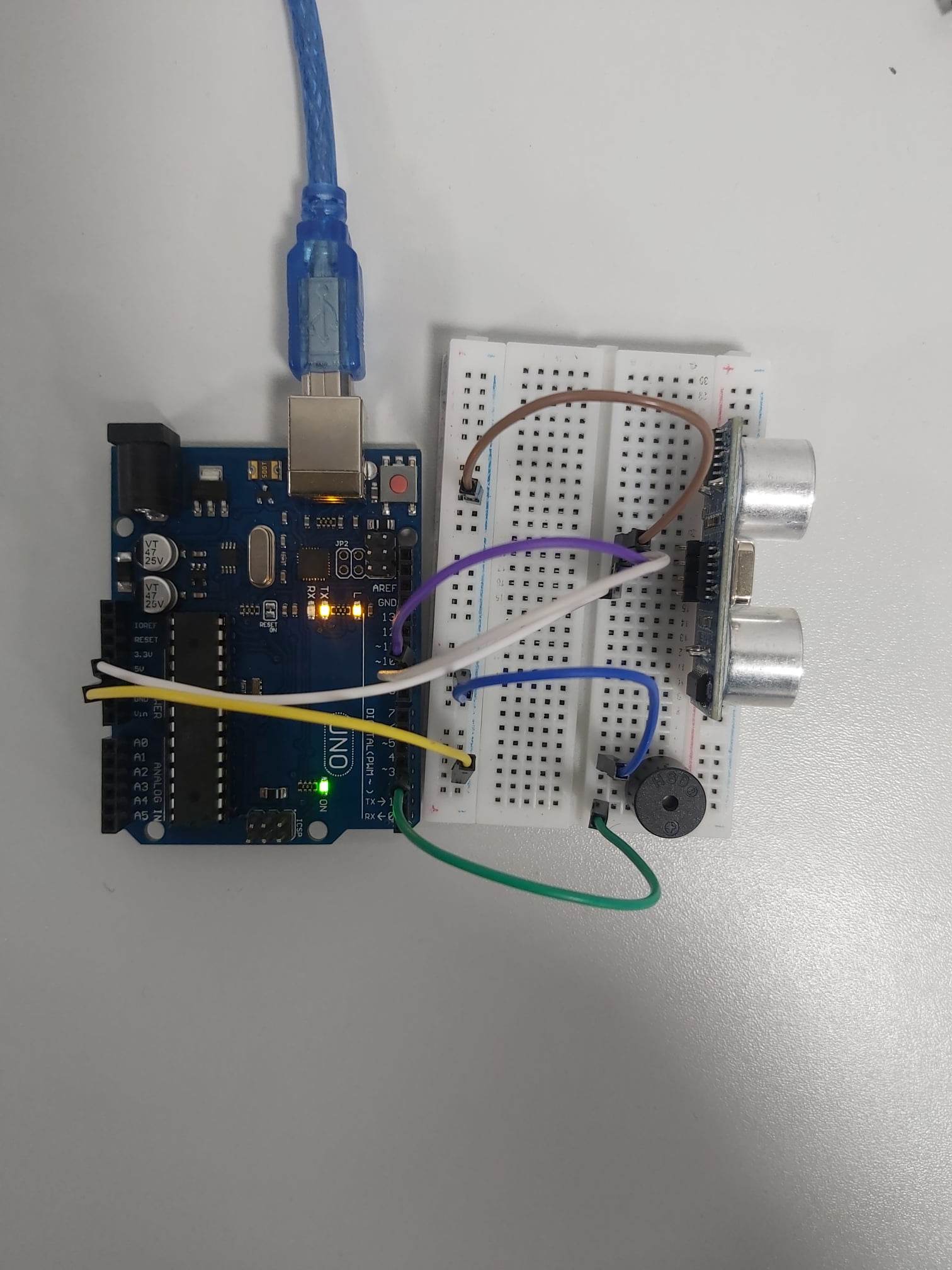
**1) Connect the Ultrasonic sensor and the buzzer to the breadboard as shown in the picture below.**

**2) Connect the wires: Wires for ultrasonic sensor: a wire between VCC leg sensor and 5v, and a wire between TRIG leg sensor(output) and port 10, A wire between ECHO leg sensor(Input) and port 9. A wire between GND(ground) Ultrasonic sensor leg and GND(ground). Wires for Buzzer: Wire from buzzer short leg to GND, and a wire from buzzer long leg to port 3.**

3) The final step is to link the Aurdino Uno Board through USB to computer.

4) After connecting the Aurdino to the computer, define the port from the IDE, then upload the code to be implemented by aurdino.

5) Test different cases on the Ultrasonic to check its functionality and the buzzer tone.



The code part:

// defines pins numbers

int BUZZER=3;

int trigPin = 10;

int echoPin = 9;

// defines variables

long duration;

int distance;

void setup() {

   Serial.begin(9600);

  pinMode(BUZZER, OUTPUT);

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

  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

  Serial.begin(9600); // Starts the serial communication

}

void loop() {

  if(distance<33 & distance>27)

  {

      digitalWrite(BUZZER,HIGH);

      delay(250);

      digitalWrite(BUZZER,LOW);

      delay(250);

      digitalWrite(BUZZER,HIGH);

      delay(250);

      digitalWrite(BUZZER,LOW);

      delay(250);

  }

   if(distance<23 & distance>17)

  {

      digitalWrite(BUZZER,HIGH);

      delay(166.7);

      digitalWrite(BUZZER,LOW);

      delay(166.7);

      digitalWrite(BUZZER,HIGH);

      delay(166.7);

      digitalWrite(BUZZER,LOW);

      delay(166.7);

       digitalWrite(BUZZER,HIGH);

      delay(166.7);

      digitalWrite(BUZZER,LOW);

      delay(166.7);

  }

   if(distance<13 & distance>7)

  {

      digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

      digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

       digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

      digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

      digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

      digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

      digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

       digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

      digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

      digitalWrite(BUZZER,HIGH);

      delay(50);

      digitalWrite(BUZZER,LOW);

      delay(50);

  }

if(distance<8 & distance>2)

  {

      digitalWrite(BUZZER,HIGH);

      delay(250);

  }

  // 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

  distance = duration \* 0.034 / 2;

  // Prints the distance on the Serial Monitor

  Serial.print("Distance: ");

  Serial.println(distance);

}

**The Ultrasonic sensor is very useful on real projects. In Alerts on the cars when they get back reverse. The Ultrasonic sensor used to calculate the distance and an alarm is triggered based on distance between the cars and the object behind it.**