Method Article

Arduino Mega Based Smart Traffic Control System

Abstract
AIM :
The main aim of this paper is design smart traffic system to develop a dynamic road signal based on density.
Study design: The conventional traffic signal system is based on constant traffic management time, so we proposed a smart system using Arduino Mega and ultrasonic sensors where green signal times change automatically according to traffic density.
Methodology: The performance of the proposed system is tested using Proteus simulator and a real model was designed.
Resolute and conclusion: the simulation and experimental test rig have shown that this system can be implemented, as it has been able to change the green signal time according to congestion density while developing this system.

Keywors- Traffic signal system, Arduino Mega ,Ultrasonic Sensor , Bluetooth module

1. Introduction

At present, with the increase in the population of large cities such as Al-bayda, Benghazi and Tripoli, the number of cars has increased significantly in Libya in recent years, especially after 2011. As the current traffic light system in Libya continues to adopt the traditional method, this has resulted in significant congestion at street intersections. With the significant development of microcontroller technology, there is an urgent need for a smart traffic system that will help reduce congestion, wait time and waste fuel.
In this context, several studies have been presented to develop traffic light and reduce congestion, many systems uses microcontroller [1,2], also RFID and GSM module[3,4] another based arduino platform [5,6,7],We propose a simple, low-cost system for a smart traffic signal.
our project is to develop a Density Based Traffic Signal System using An Arduino Mega, Ultrasonic Sensor and Bluetooth module are used to implement the traffic system. we will use Ultrasonic Sensor to measure the traffic density. We have to mount three Ultrasonic Sensor for each road; the distance between these sensors will depend on nature of traffic on a particular junction. These sensors determine the level of jam in the road , Different ranges of traffic light delay time intervals according to jam level are configured by microcontroller and updated regularly. Additionally, control the traffic signals by an android application throughout the Bluetooth and wireless connection. This will be controlled by the traffic warden or police officer. The prototype of the system has been built and tested. The model is better than the current system which is limited to the Fixed Mode Traffic Light Controller.

2- Components Used

A. Ultrasonic Sensor

The HC-SR04 ultrasonic sensor fig.1. uses sonar to determine the distance to an object. This sensor reads from 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inches), which is good for most hobbyist projects. In addition, this particular module comes with ultrasonic transmitter and receiver modules.
The ultrasonic sensor uses sonar to determine the distance to an object

The ultrasound transmitter (trig pin) emits a high-frequency sound (40 kHz).
The sound travels through the air. If it finds an object, it bounces back to the module.
The ultrasound receiver (echo pin) receives the reflected sound (echo).
The time between the transmission and reception of the signal allows us to calculate the distance to an object. This is possible because we know the sound's velocity in the air. Here's the formula:
distance to an object = ((speed of sound in the air)*time)/2
speed of sound in the air at 20ºC (68ºF) = 343m/s

VCC	*Power Sensor 5V*
Trig	*Trigger Input Pin*
Echo	*Echo Output Pin*
GND	*ground*

Fig.1 .Ultrasonic Sensor

B. Arduino Mega

The Arduino Mega (fig.2) is the muscle car of Arduino boards. It boasts a huge collection of input output ports, but cleverly adds these as extra connectors at one end of the board, It uses a processor with more input output pins, the ATmega1280, which is a surface mount chip that is fixed permanently to the board. So, unlike with the Uno and similar boards, you cannot replace the processor if you accidentally damage it.
The extra connectors are arranged at the end of the board. Extra features provided by the Mega include the following:
• 54 input/output pins
• 128KB of flash memory for storing sketches and fixed data
• 8KB of RAM and 4KB of EEPROM

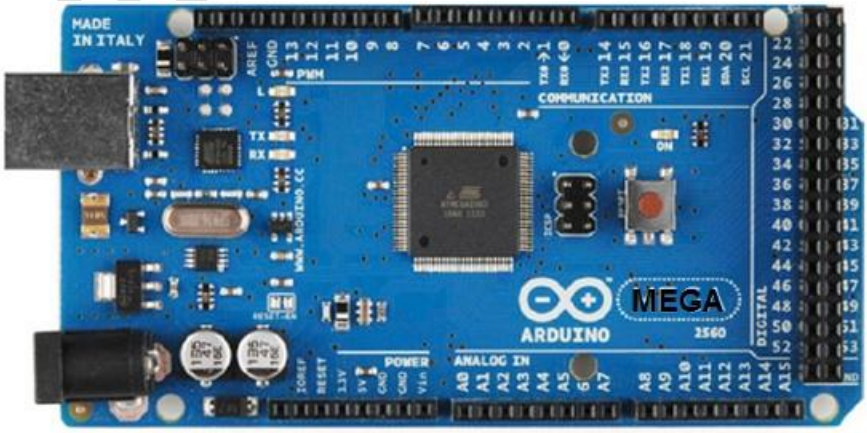


Fig.2 . Arduino Mega

C. Bluetooth Module

HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.

Key/EN:	HC-05 module has two modes, Data mode and Command mode
VCC:	Connect 5 V or 3.3 V to this Pin.
GND:	Ground Pin of module.
TXD:	Transmit Serial data
RXD:	Receive data serially
State:	It tells whether module is connected or not.

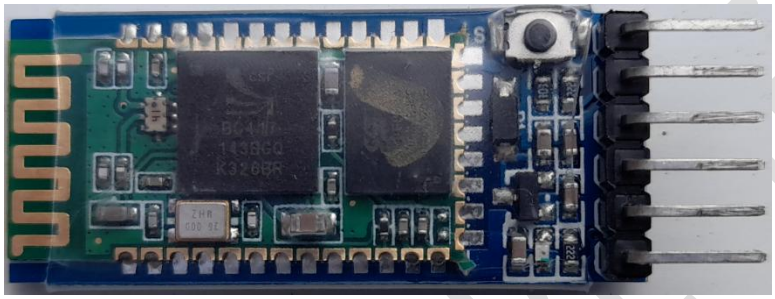


Fig .3 . HC-05 is a Bluetooth module

D. LCD display

Liquid Crystal Display (LCD):16x2 LCD used in the implemented to display data over 2 lines, each of 16 characters. Actually, two types of registers are used to configure the LCD; the command registers and control registers. Other aspects are LCD initialization, clearing the screen, setting the cursor position, and controlling display. While the data register holds the ASCII code of the characters that are appeared on the display.

fig.4. 16x2 LCD

3. Proposed System design and Methodology

In this paper, we utilized (Arduino Mega), that has a many pins, causing it exceptionally valuable in projects that to have a convenient number of sensors and gadgets to control.

As can be seen in figure (5), there is a intersection of two roads that normally has four sides, the first side provide with two sensors as well as the third side, first sensor in both side detects the traffic of cars , and the second sensor find out if there is a congestion , so we can determine whether there's a traffic congestion or not. We also added a Bluetooth unit which a police officer can control the signal by his mobile phone, the principle of operation of this Chamber is based on three cases.

First case: There's no congestion. Traffic signal works normally.

Second case: There is a congestion, a delay time should added to the green signal.

Third case: if there is no jam near the traffic-light , the system will skip it and move to the next side , so with that, we will diminish congestion.

This system also enables a traffic cop to control the traffic-light using an application that is loaded onto his mobile phone , simply by type the number of side to be played, so it works as follow :

- Typing number one leads to turn on green light in (Side1)
- Typing number two leads to turn on green light in (Side2)
- Typing number Three leads to turn on green light in (Side3)
- Typing number four leads to turn on green light in (Side1)

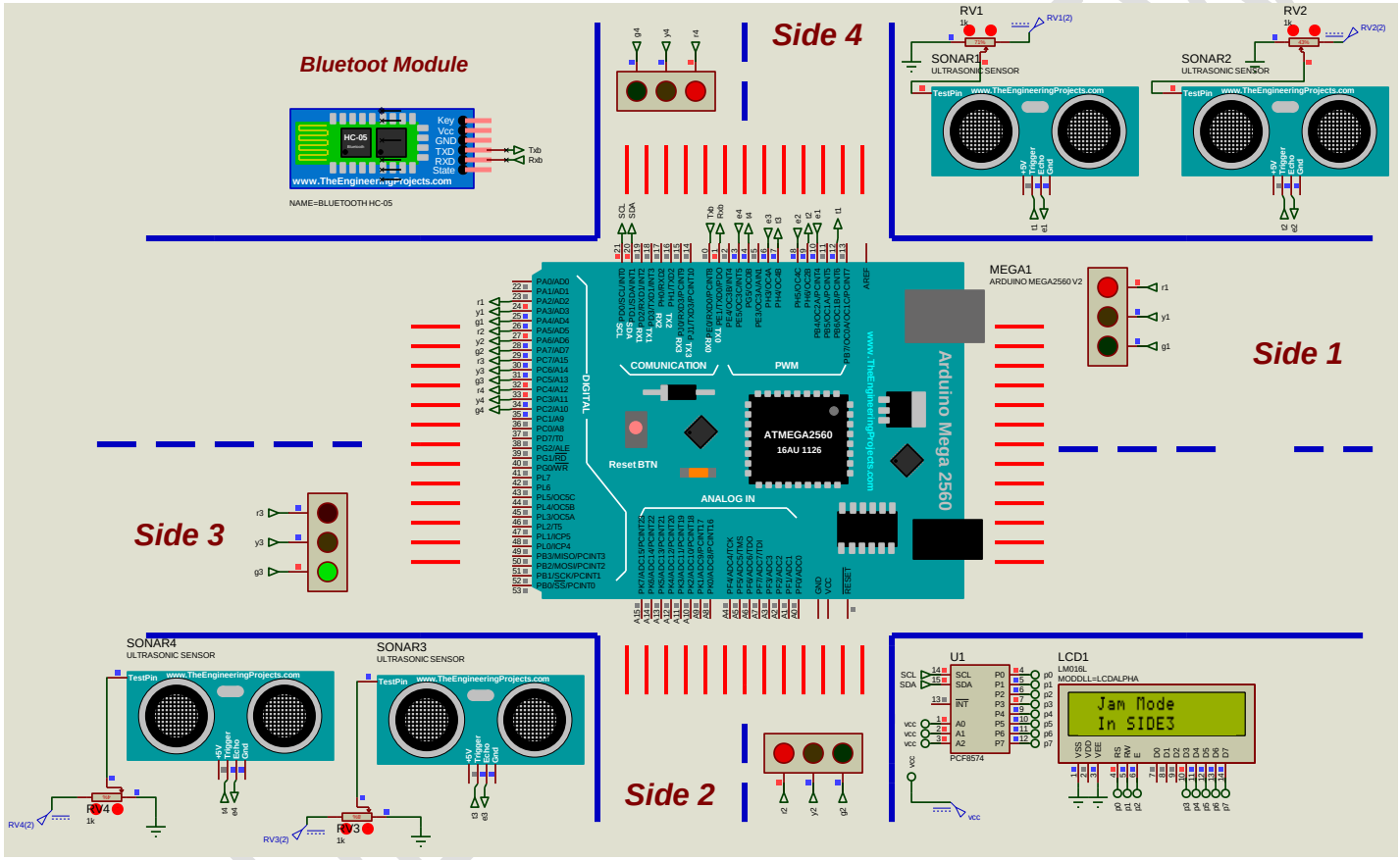


Fig.5 . The proposed System Simulation using Proteus

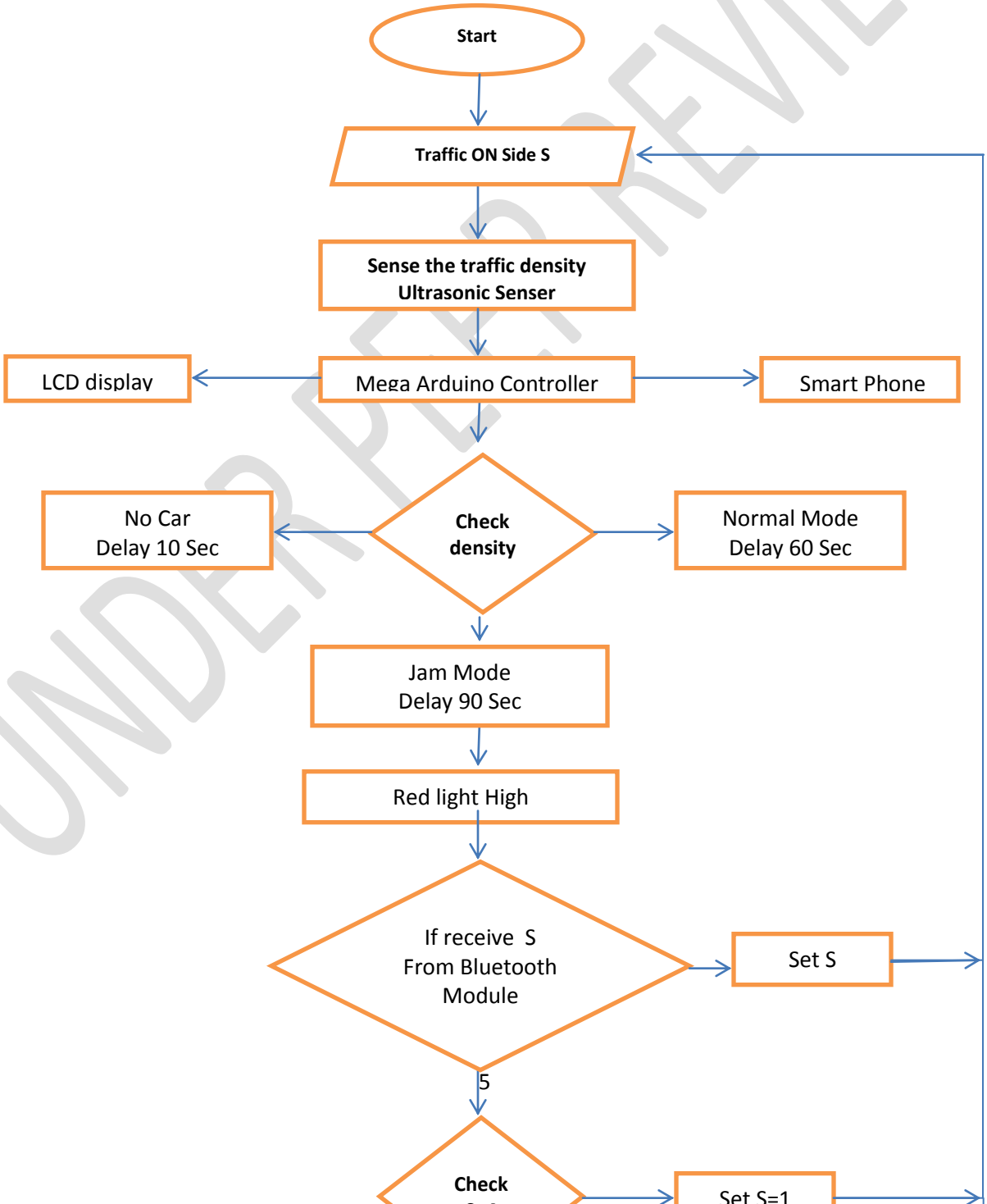
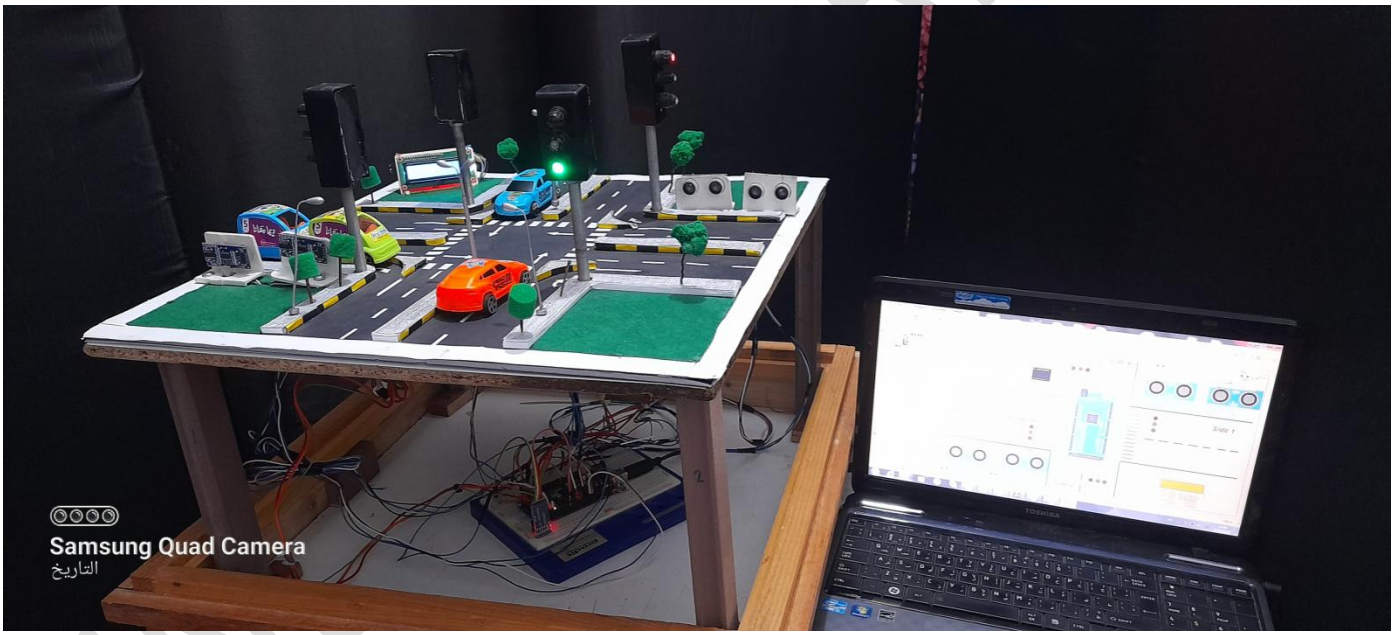


Fig.6 . Flow chart of the system

Chart 1 . The Code

```
#include <Wire.h> //include the library
#include <LiquidCrystal_PCF8574.h> //include the libraryLCD
LiquidCrystal_PCF8574 lcd(0x27);  // set the LCD address to 0x27 for a 16 chars and 2 line display
int side1[ ] = {24,25,26};//lights side1
int side2[ ] = {27, 28,29};//lights side2
int side3[ ] = {30,31, 32};//lights side3
int side4[ ] = {33,34,35};//lights side4
int green1Delay = 10000;//Time delay  NO cars
int green2Delay = 60000;//Time delay  Normalmode
int green3Delay = 90000;//Time delay  traffic Jam
int yellowDelay = 5000;//Time delay yallow light
volatile int trigpin[ ]={12,9,7,4 };//pins connected to Trig ultrasonic
volatile int echopin[ ]={10,8,6,3 };//pins connected to Echo ultrasonic
volatile int S1, S2,S3,S4, Sum1,Sum2, Sum3,Sum4,Avg1,Avg2,Avg3,Avg4;
//variable specify Distance
int volatile long dd= 200; int n=0;int x;
void setup(){
  Wire.begin();
  Wire.beginTransmission(0x27);
  Serial.begin (9600);
  lcd.begin(16, 2); //
  lcd.setBacklight(255);lcd.home();
  //set Pin as an Output
     for(int i=0; i<3; i++){
    pinMode(side1[i], OUTPUT);
    pinMode(side2[i], OUTPUT);
    pinMode(side3[i], OUTPUT);
    pinMode(side4[i], OUTPUT);
  }}
void loop(){
  ///////side1://///////
  side1:
   red();
   lcd.clear();
   lcd.print("   SIDE1  ");
   Serial.println("side1");
  digitalWrite(side1[0], LOW);
  digitalWrite(side1[2], HIGH);
  Sum1=0,Sum2=0,n=0,Avg1=0,Avg2=0;
     lcd.clear();
     Serial.println("  Normal Mode ");
     lcd.print("  Normal Mode ");
   lcd.setCursor(0,1);
     lcd.print("  In SIDE1  ");
     delay(green2Delay); }
 // No cars In Road
   else if(Avg1>=dd&&Avg2>=dd){
     lcd.clear();
     Serial.println(" NO Cars ");
     lcd.print("  NO Cars ");
     lcd.setCursor(0,1);
     lcd.print("  In  SIDE1  ");
     delay(green1Delay); }
     digitalWrite(side1[1], HIGH);
     digitalWrite(side2[1], HIGH);
     delay(yellowDelay);
     int x;  x=Serial.read();
   if (x=='2'){goto side2;}
   if (x=='3'){goto side3;}
   if (x=='4'){goto side4;}
     ///////side2:///////////
     side2:
      red();
      lcd.clear();
      lcd.print("   SIDE2   ");
      Serial.println("side2");
      digitalWrite(side2[0], LOW);
      digitalWrite(side2[2], HIGH);
      delay(green2Delay);
      digitalWrite(side3[1], HIGH);
      digitalWrite(side2[1], HIGH);
      delay(yellowDelay);
      x=Serial.read();
   if (x=='1'){goto side1;}
   if (x=='3'){goto side3;}
   if (x=='4'){goto side4;}
  ////////// sid3 ////////////
     side3:
     red();
     lcd.clear();
     lcd.print("   SIDE3   ");
     Serial.println("side3");
    digitalWrite(side3[0], LOW);
     lcd.clear();
     Serial.println("  Normal Mode ");
     lcd.print("   Normal Mode ");
     lcd.setCursor(0,1);
     lcd.print("  In SIDE3  ");
     delay(green2Delay);}
  // No cars In Road
   else if(Avg3>=dd&&Avg4>=dd){
      lcd.clear();
     Serial.println("  No Cars ");
     lcd.print("  No Cars ");
     lcd.setCursor(0,1);
     lcd.print("  In  SIDE3  ");
      delay(green1Delay);}
   digitalWrite(side3[1], HIGH);
     digitalWrite(side4[1], HIGH);
     delay(yellowDelay);
      x=Serial.read();
    if (x=='1'){goto side1;}
   if (x=='2'){goto side2;}
   if (x=='4'){goto side4;}
   //////////////sid4 //////////////
   side4:
    red();
   lcd.clear();
   lcd.print("    SIDE4    ");
   Serial.println("side4");
   digitalWrite(side4[0], LOW);
   digitalWrite(side4[2], HIGH);
   delay(green2Delay);
    digitalWrite(side1[1], HIGH);
    digitalWrite(side4[1], HIGH);
    delay(yellowDelay);
      x=Serial.read();
   if (x=='1'){goto side1;}
   if (x=='2'){goto side2;}
   if (x=='3'){goto side3;}
 }
 ////////function calculate distance
   int dist(int pingPin , int echoPin){
   long d,m;
   pinMode(pingPin, OUTPUT);
   digitalWrite(pingPin, LOW);
   delayMicroseconds(5);
```

```
  for(int i=0; i<=6; i++)  {
 S1=dist(trigpin[0] , echopin[0]);
 Sum1=Sum1+S1;
 S2=dist(trigpin[1] , echopin[1]);
 Sum2=Sum2+S2; n=n+1;
 delay(50);
  }
 Avg1=Sum1/n; Avg2=Sum2/n;
//condition Traffic Jam in Side 1
   if(Avg1<dd && Avg2<dd) {
 digitalWrite(side1[0], LOW);
digitalWrite(side1[2], HIGH);
  lcd.clear();
  lcd.print("   Jam Mode ");
  Serial.println("  Jam Mode ");
  lcd.setCursor(0,1);
  lcd.print(Avg1);
  delay(green3Delay);
 }
////condition Traffic Normal In Side1
   else if(Avg1<dd&&Avg2>=dd) {
   digitalWrite(side1[0], LOW);
   digitalWrite(side1[2], HIGH);
   digitalWrite(side3[2], HIGH);
Sum3=0,Sum4=0,n=0,Avg3=0,Avg4=0;
  for(int i=0; i<=6; i++)  {
 S3=dist(trigpin[2] , echopin[2]);
 Sum3=Sum3+S3;
 S4=dist(trigpin[3] , echopin[3]);
 Sum4=Sum4+S4; n=n+1;
 delay(50); }
 Avg3=Sum3/n; Avg4=Sum4/n;
//condition Traffic Jam in Side 3
   if(Avg3<dd && Avg4<dd){
  digitalWrite(side3[0], LOW);
  digitalWrite(side3[2], HIGH);
   lcd.clear();
    Serial.println("  Jam Mode ");
   lcd.print("  Jam Mode ");
   lcd.setCursor(0,1);
   lcd.print("  In SIDE3  ");
   delay(green3Delay);}
////condition Traffic Normal In Side3
   else if(Avg3<dd&&Avg4>=dd){
   digitalWrite(side3[0], LOW);
   digitalWrite(side3[2], HIGH);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(pingPin, LOW);
  pinMode(echoPin, INPUT);
  d = pulseIn(echoPin, HIGH);
  m= d*0.034/2;
  return m;
}
//All  Green light,and Yallow light  LOw ,All
Red light High
 void red(){
  for(int i=1; i<3; i++) {
  digitalWrite(side1[i], LOW);
  digitalWrite(side2[i], LOW);
  digitalWrite(side3[i], LOW);
   digitalWrite(side4[i],LOW);
  }
  for(int i=0; i<1; i++){
   digitalWrite(side1[i], HIGH);
   digitalWrite(side2[i], HIGH);
   digitalWrite(side3[i], HIGH);
   digitalWrite(side4[i], HIGH);
  }}
```



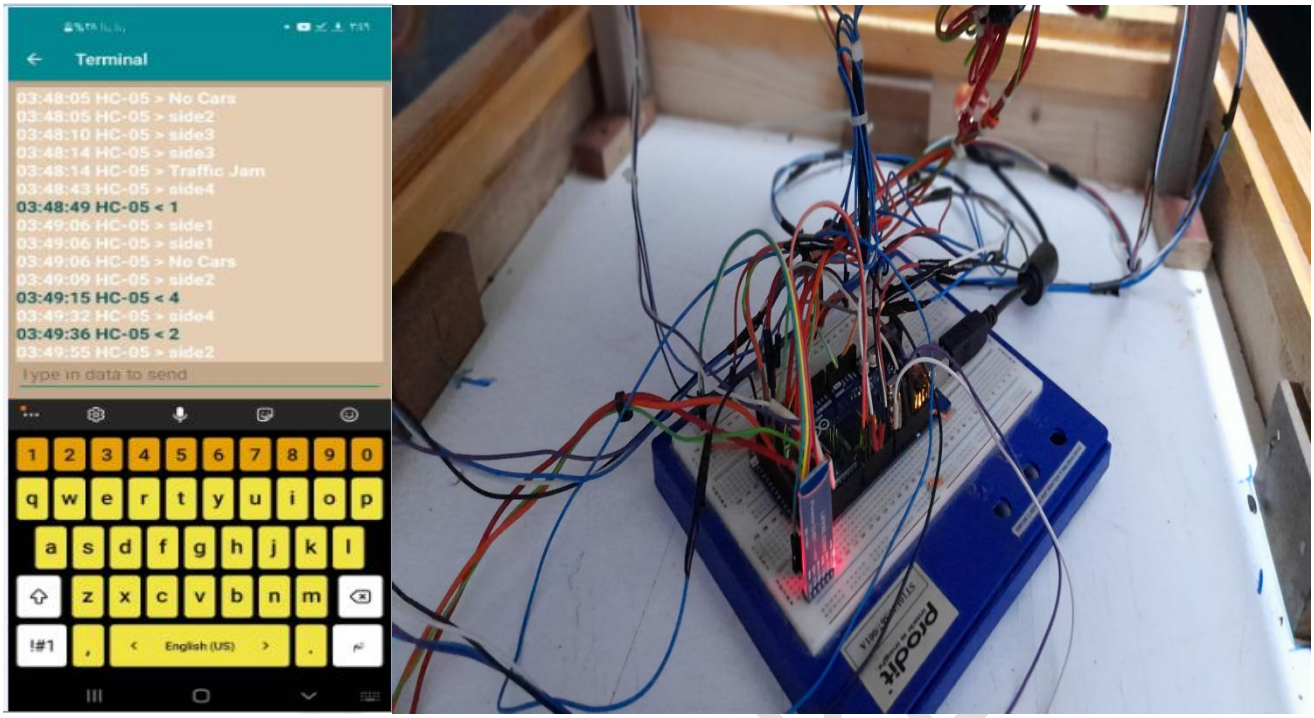


Fig.7 . Implementation of the System

4. Results And Discussion

We first implemented the project circuit using Proteus 8 as simulator and then a practical system was implemented using Arduino Mega 2560, with three LED's (red, yellow and green light) were connected in parallel, connected in series with resistors 220 Ohm to each LED where the cathodes connected together with the GND of Adduino , the positive end of each led is connected to the Urduino board and repeats this for each side as shown the order below:

Side1 signal(Red, Yallow, Green)	Pin 24, pin25 , pin26
Side2 signal(Red, Yallow, Green)	Pin 27, pin28 , pin29
Side3 signal(Red, Yallow, Green)	Pin 30, pin31 , pin32
Side4 signal(Red, Yallow, Green)	Pin 33, pin34 , pin35

Ultrasound sensor(US) pins was connected to to the Urduino board as described below:

Ultrasound sensors (US) are responsible for giving information's about the crowd to the controller for making a decision to operate the traffic light on side1 and Side3 according to the table .

Table 1 . Modes of traffic

Traffic Modes	US1	US2	Timing Green light
No Car	Low	Low	10 Sec
Normal Mode	High	Low	60 Sec
Jam Mode	High	High	90 Sec

US1(Trig , Echo) Pin 12,Pin 10
US2(Trig , Echo) Pin 9 ,pin8
US3(Trig , Echo) Pin 7,pin6
US4(Trig , Echo) Pin 4,pin3

The number and location of sensors depends on the length of the street , increasing the sensors gives more accuracy in determining the level of crowd , but the cost will increase.

The system provided with an LCD screen to display crowd level also waiting time.

Finally, Bluetooth Module (HC- 05) has been linked to the RX and Tx of arduino.

5. Conclusion

An intelligent traffic light control system was designed for a four-way intersection. The circuit was simulated and a practical model was made to test the circuit and ensure its correct operation. The system is effective and has low cost and helps reduce congestion and reduce waiting time at intersections, and with the use of Arduino controllers and the integrated development environment Arduino C gives this The system is flexible and highly adjustable

COMPETING INTERESTS DISCLAIMER:

Authors have declared that no competing interests exist. The products used for this research are commonly and predominantly use products in our area of research and country. There is absolutely no conflict of interest between the authors and producers of the products because we do not intend to use these products as an avenue for any litigation but for the advancement of knowledge. Also, the research was not funded by the producing company rather it was funded by personal efforts of the authors.

6. References

[1]. Saiba P A, Afeefa M U, Aruna T S, Clincy Jose, Radhika V M "Density Based Traffic Signal System using PIC Microcontroller" International Journal of Computer Trends and Technology (IJCTT) – Volume 47 Number 1 May 2017.

[2]. Md. Shamsul Arifin*, Sheikh Alimur Razi, Akramul Haque, Nur Mohammad "A Microcontroller Based Intelligent Traffic Control System". American Journal of Embedded Systems and Applications. 2019; 7(1): 21-25

[3]. R. Sundar, S. Hebbar, and V. Golla, "Implementing intelligent traffic control system for congestion control,

ambulance clearance, and stolen vehicle detection," IEEE Sens. J., vol. 15, no. 2, pp. 1109–1113, 2015.

[4]. A. Patnaik, V. Natarajan, B. Karthikeyan, and P. Surendran, "Automatic traffic control system for single lane tunnels,"
2014 Int. Conf. Adv. Electr. Eng. ICAEE 2014, pp. 1–4,2014.

[5]. Adithya Menon1, Ashutosh Dubey2, Farhan Ul Haque3, Lalit Kumar4, Mahesh Kamthe "Density Based Traffic Light Controller using Arduino – Nano". (IJESC) vol.10 Issue No.5

[6]. Pallavi H. B, Aishwarya Shrikanth, Sandeep S, Sushanth S. P, Hariprasad S. A, Sunil M.P " Density based Automatic Traffic Junction Synchronization". International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-2, July 2019

[7]. Saranya J, Jayashwanth J.S, Kiran J, Harish S, Linga Kumar T .Automated Density Based Traffic Light Control System Using Arduino Platform". International Journal of Engineering and Advanced Technology (IJEAT). ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019

[8]. Chinedu Okpara , Somtochukwu Oguchienti, Kingsley Nwosu, Ijeoma Ogidi "Design of a Smart Traffic Light Control System using Arduino Mega". International Journal of Scientific & Engineering Research Volume 11, Issue 5, May-2020

1350 ISSN 2229-5518