Gokaraju Rangaraju
Institute of Engineering and Technology

Bachelor of Technology

(Electronics and Communication Engineering)

Industry Oriented Mini Project

Title: Density Based Traffic Light System

Name: P.Priyanka

Roll No.:15241A04M0

Contents

Design Document

- The aim of the project is to solve traffic congestion which is a severe problem in many modern cities all over the world.
- To solve the problem, we have designed a framework for a dynamic and automatic traffic control system.
- Generally each traffic light on an intersection is assigned a constant green signal time.
- It is possible to propose dynamic time-based coordination schemes where the green signal time of the traffic lights is assigned based on the present conditions of traffic.
- The intelligent work which is done by traffic inspector will be perfectly done by the microcontroller in the circuit with the help of sensors and the program which is coded to the microcontroller.

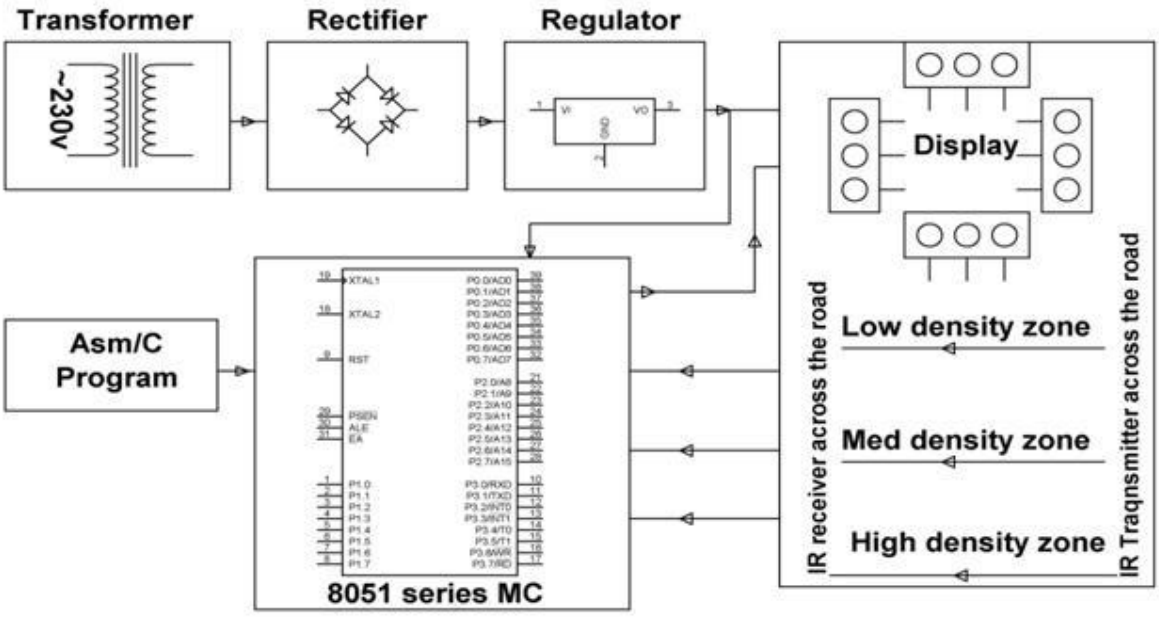
This project consists of

- ATMEGA 328 microcontroller
- IR TRANSMITTER
- IR RECEIVER
- BLUETOOTH
- Connecting Wires
- LED
- Power supply

Hardware The board consists of

The custom-built board contains the follows resources:

- Microcontroller: ATmega328 microcontroller in Arduino with clock frequency of 16 MHz
- Power supply: An adapter that supplies 9V supply is used in the project.
- The custom board has a voltage regulator that drops the 9v to a stable 5v.
- IR transmitter and receiver calculating the density on the road.
- On the basis of density the traffic is cleared.
- Bluetooth module is used to display the information.
- This information is seen in a custom built app.
- Female-Female pin headers for connecting the modules to custom board.



Software: Arduino IDE, the schematic and the board design will be done using the Eagle software .

Firmware:

The firmware will be written using the Arduino UNO. This is possible since the ATmega328 microcontroller is loaded with the Arduino bootloader. The firmware will do the following functions:

- Compile and upload the program with required headers of Gps module.
- A message is to sent to the system.
- A code is written to read values from the IR sensors and convert the values to density.
- Generally , with normal traffic a general delay is given to the system so that general traffic runs.
- If the density on one road is more compared to the other three then the traffic in that road is cleared and then general delay is followed.

Project Specifications

Functional Requirements:

- The system(IR TRANSMITTER) continuously emits ir radiations which are received by an IR receiver at other end.

- The system (IR RECEIVER) continuously receives the ir radiations emitted by the ir transmitter.

- The system (microcontroller) calculates the density based on the values form the ir receivers.

- The Bluetooth module is used to connect wirelessly and the information is printed in the custom app.

Non-functional Requirements:

- The Size of the systems is very small, we can fix it easily in any place.

- The power consumed by the system is very low, the system can be operated by a 9v battery.

- It is easy to operate.

- Power wastage is reduced.

- The system is very compact, so that it can be Mounted At all sides on the road .

- External wiring is required .

Deliverables:

Hardware:

- ATMEGA 328
- IR TRANSMITTER
- IR RECEIVER
- BLUETOOTH
- TRAFFIC JUNCTION CUSTOM BOARD
- LED`S
- Female-Female connecting wires
- Male header pins
- 9v DC adapter
- 9V battery
- DC jack
- Voltage regulator
- Resistors and capacitors
- Copper clad PCB board
- Sockets for atmega
- Fecl3 powder for etching

Software:

- Arduino IDE (For Arduino coding and code dumping)
- Eagle software (For Schematic& initial board design)

Theory

ATMEGA 328:

The Atmel AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

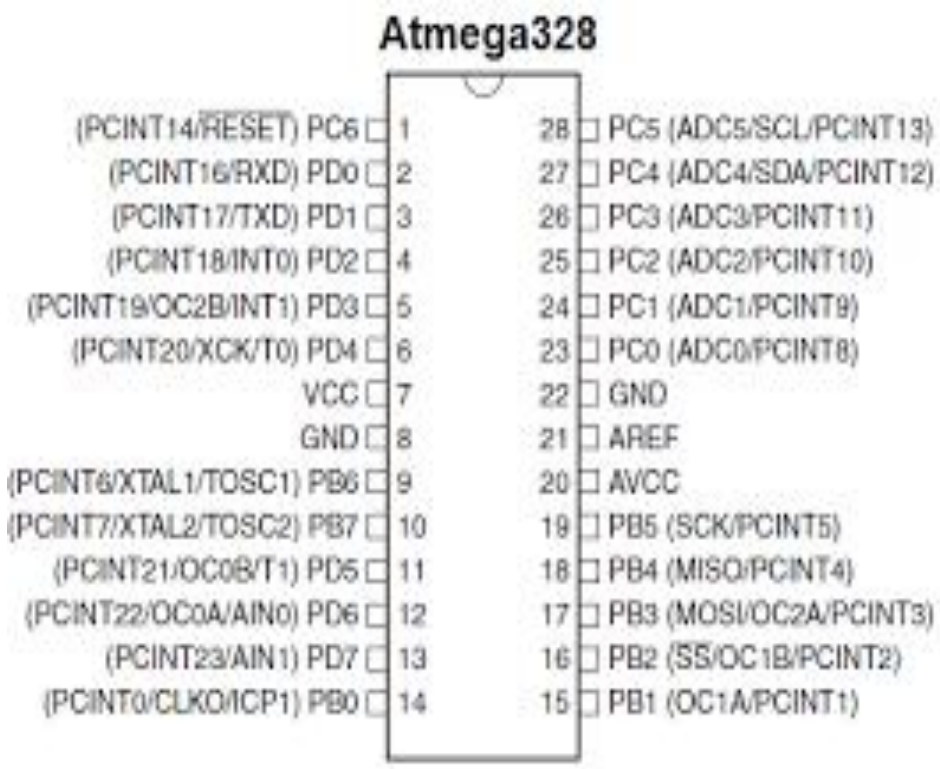


FIG : PIN DIAGRAM OF ATMEGA328 MICROCONTROLLER IC

The Atmega168 provides the following features: 16 Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1 Kbyte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two wire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM; Timer/Counters, SPI port, and interrupt system to continue function

FIG : ATMEGA328

The Power down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

IR SENSOR :

The IR transmitter consists of the LED that emits the IR(Infra Red) radiation.This is received by the photo diode, which acts as IR receiver at the receiving end. Since the IR radiation is invisible to human eye it is perfect for using in wireless communication.

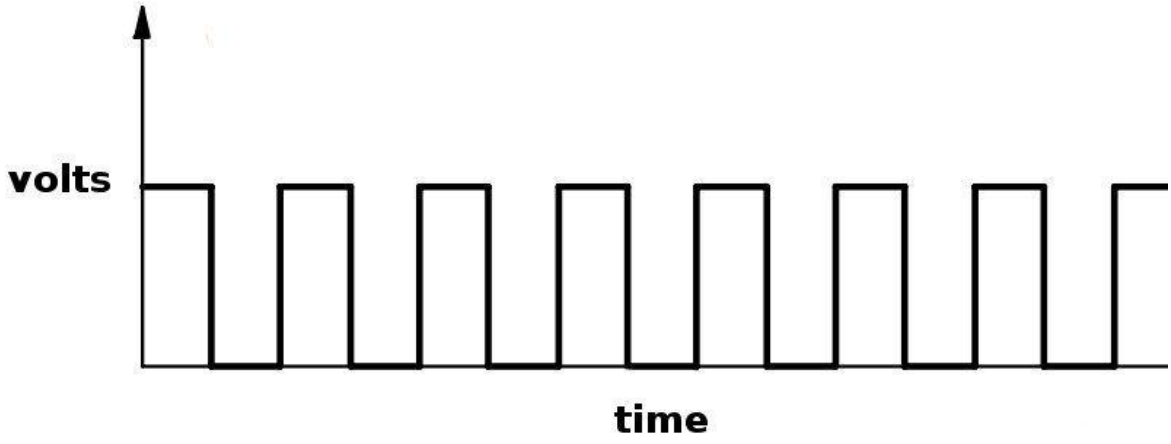
A electronic remote device mainly consists of this IR transmitter and receiver. A remote control patterns a flash of invisible light which is turned into an instruction and is received by the receiver module.

So how it works?

The IR signal is modulated during transmission. Modulation means assigning pattern to the data to be sent to the receiver.The most commonly used IR modulation is about 38khz.

When you hit a key on your remote, the transmitting IR LED will blink very quickly for a fraction of a second, transmitting encoded data to your appliance.

On the receiver side The modulated signal is demodulated and the pattern is obtained as



In this project, the transmitter section includes an IR sensor, which transmits continuous IR rays to be received by an IR receiver module. An IR output terminal of the receiver varies depending upon its receiving of IR rays. Since this variation cannot be analyzed as such, therefore this output can be fed to a comparator circuit. Here an operational amplifier (op-amp) of LM 339 is used as comparator circuit.

When the IR receiver does not receive a signal, the potential at the inverting input goes higher than that non-inverting input of the comparator IC (LM339). Thus the output of the comparator goes low, but the LED does not glow. When the IR receiver module receives signal to the potential at the inverting input goes low. Thus the output of the comparator (LM 339) goes high and the LED starts glowing. Resistor R1 (100), R2 (10k) and R3 (330) are used to ensure that minimum 10 mA current passes through the IR LED Devices like Photodiode and normal LEDs respectively. Resistor VR2 (preset=5k) is used to adjust the output terminals. Resistor VR1 (preset=10k) is used to set the sensitivity of the circuit Diagram. Read more about IR sensors.

BLUETOOTH :

HC-05 module is an easy to use **Bluetooth SPP (Serial Port Protocol) module**,designed for transparent wireless serial connection setup.The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication.This serial port bluetooth module is fully qualified **Bluetooth V2.0+EDR (Enhanced Data Rate)**3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses **CSR Bluecore 04**-External single chip Rluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

HC-05 Bluetooth Module

The Bluetooth module HC-05 is a MASTER/SLAVE module.By default the factory setting is SLAVE.The Role of the module (Master or Slave) can be configured only by AT COMMANDS.The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections.Master module can initiate a connection to other devices.The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project.

Hardware Features

- Typical -80dBm sensitivity.
- Up to +4dBm RF transmit power.
- 3.3 to 5 V I/O.
- PIO(Programmable Input/Output) control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

Software Features

- Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:”1234” as default.

POWER SUPPLIES:

In this project we use a 9v DC adapter that supplies a 9volts voltage to the board but the atmega328 is capable of and handles 5v so the incoming 9v is to be converted to 5v which is also useful for GSM module and GPS modules. This can be achieved by dropping down the voltage and by regulating it hence a voltage regulator is used.

LM 7805 :

7805 is a **voltagee regulator** integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output.

The **voltage regulator IC** maintains the output voltage at a constant value. The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

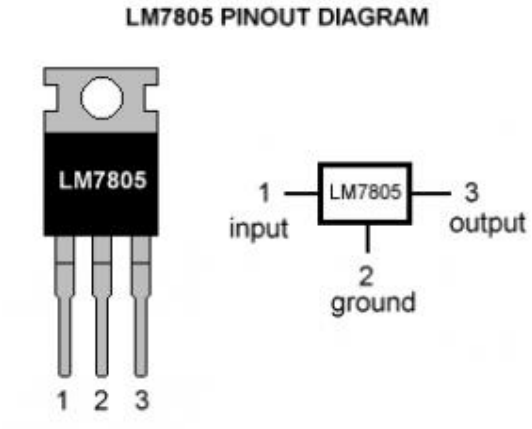
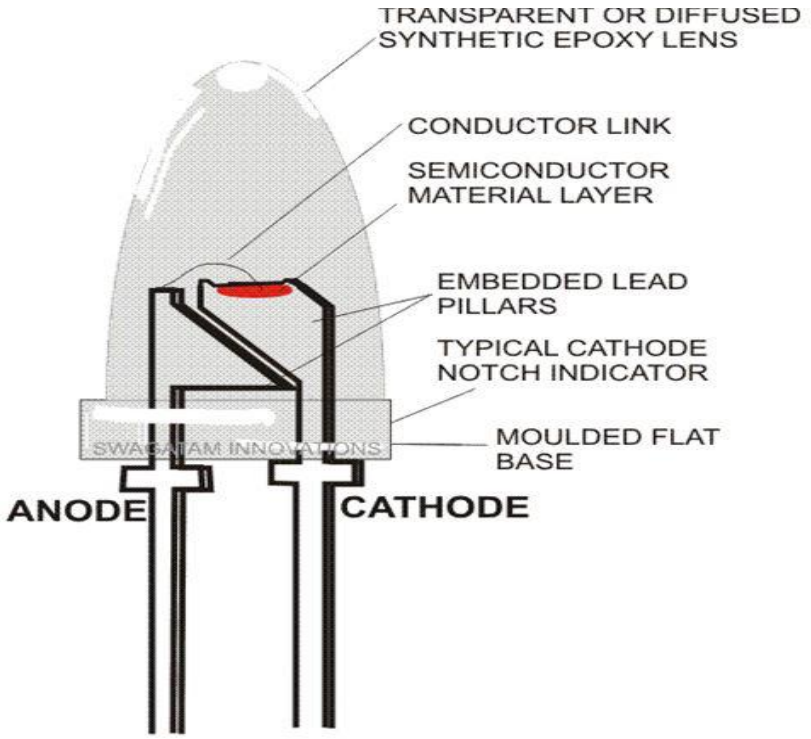


FIG : 7805 PIN DIAGRAM

LED :

A **light-emitting diode** (**LED**) is a two-lead semiconductor light source. It is a pn-junction diode, which emits light when activated.]When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons.

This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor.



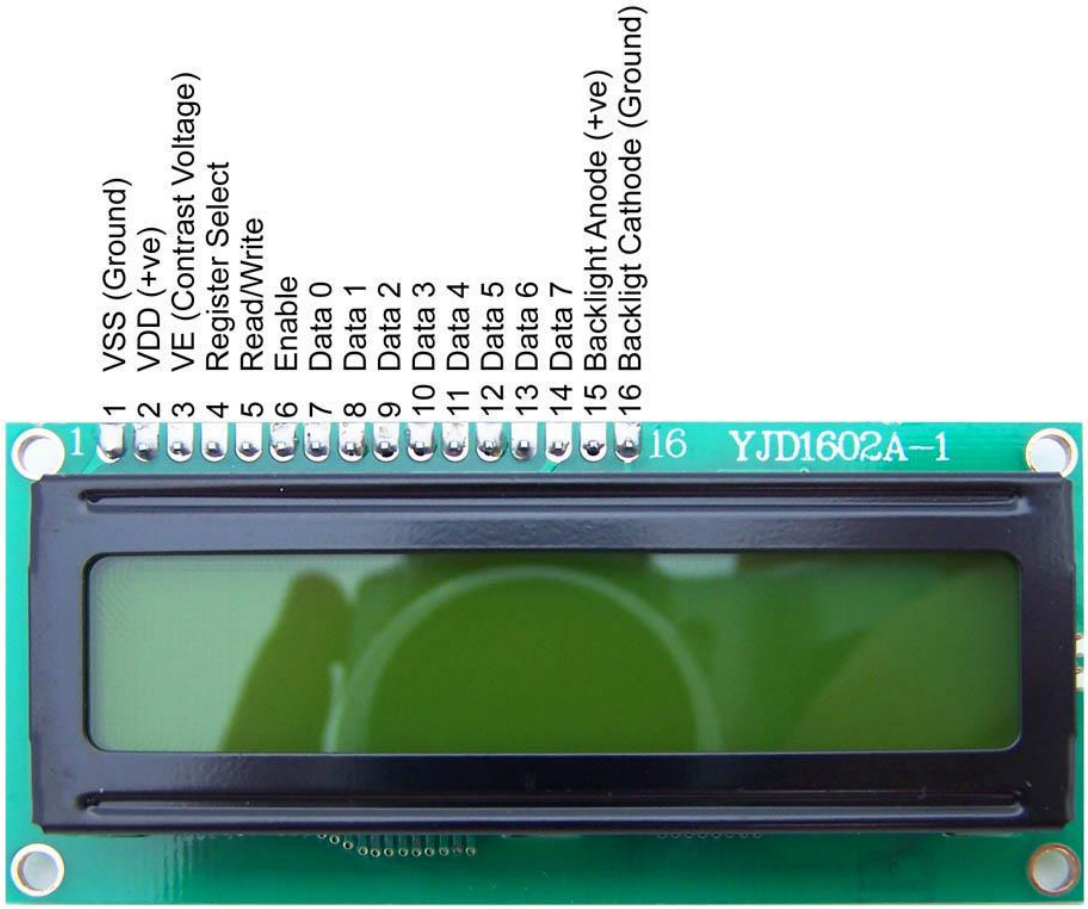
INTERNAL COMPOSITION OF A LED

LCD:

A liquid-crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly.

LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

LCDs are used in a wide range of applications including computer monitors, televisions, instrument panels, aircraft cockpit displays, and signage. They are common in consumer devices such as DVD players, gaming devices, clocks, watches, calculators, and telephones, and have replaced cathode ray tube (CRT) displays in most applications.The LCD screen is more energy efficient and can be disposed of more safely than a CRT. Its low electrical power consumption enables it to be used in battery-powered electronic equipment.



PCB CONSTRUCTION :

PCB Layout

No device can work if its connections are not according to specification, and if the proper resistance, capacitance, inductance etc. are not connected to the place where required. Thus a PCB designer has to first think of the very possible combination of voltages that are required by the circuit and make them available at points where they are needed with the minimum use of jumpers and keeping the circuit size compact and yet effective.

The layout of PCB has to incorporate all the information on the board before one can go on the artwork preparation. This means that a concept, which clearly defines all the details of the circuitry and partly of final equipment, is a prerequisite before the actual layout can start.

For PCB layout, the following points ought to be considered carefully

1. Record size of components used.
2. Overall area covered is normally kept rectangular or square.
3. Vcc and ground lines should be provided at the sides to facilitate external connection.
4. Input and output terminals may be placed giving through to external connection.
5. Make a rough sketch placing components and interconnect components with jumpers.
6. Do not place components pointing in differed direction unless needed. Make them parallel to the either side of the board.
7. Make the neat final scaled sketch on the inch graph sheet.
8. Lines mounted are of uniform width.
9. Invest the layout to confirm that all the components are connected properly and given sufficient place in the layout.

Etching

In all PCBs, etching is the most important step. The final copper pattern is formed by selective removal of all unwanted copper which is not protected by an etch resist. Amongst the Enchants, FeCl3 (Ferric Chloride) is commonly used for small PCBs where etching is only out carried out occasionally for a small number of boards.

For etching, the solution is made, wherein sample and standard solution are in 2*1 dilutions. In order to increases the copper dissolution capacity and to bring the etching time slightly down, HCL is added.

Etching temperature should be in the ranges of 20°C to 45°C. $FeCl_3$ is an enchant used in small-scale PCB production. In high volume production $FeCl_3$ is of not much importance because it cannot be regenerated and it attacks the common metal etch resist.

Drilling

Drilling of holes for mounting components is important mechanical operation in PCB production process. The importance of hole drilling intoPCB's has further gone up with electronic components miniaturization. After rinsing drilling is done using bit as per the circuit provided. The diameters of holes generally accepted are as follows.

1. D = 0.8 mm
2. D =1.1 mm
3. D = 1.5 mm
4. D = 3.2 mm

Where,
D = Hole diameter.

Component Mounting

1. Before mounting any components, examine the PCB carefully for any cracks, beaks or other defects in conduction paths.

2. The leads of components like resistors and capacitors should be fully inserted into the mounting holes taking care to mount the components so that any information written on the components is clearly visible.

3. Carefully cut the leads of components so that about 3 mm of the end extended beyond the wiring side of the PCB. The ends of the leas are bent at right angles to make a firm contact with the surface where it is to be soldered.

4. In case of semiconductor devices like transistors and diodes, the length of the leads extending above the component side of the PCB should be about 1 cm. if transistor leads are too short we use a base. Metal cap should touch if they are not at ground potential. The right terminals should be at right places.

5. Certain components like transformers, potentiometers and variable capacitors, which are meant for use with PCB, are provided with pin type terminals that can be simply inserted into the hole in the PCB and soldered.

6. Use IC base for IC.

Soldering

PCB soldering required proper soldering technique, as explained below:

1. A light duty soldering iron of 25W or 30 W rating should be used to prevent damage to the printed circuit wiring due to excessive heating. The tip of soldering iron should not have an oxide coating. Clean it using sand paper.

2. Do not use excess solder to avoid solder flouring to adjacent conducing paths forming bridges, which cause short circuits.

3. Clean the surface of traces before you start soldering. It is advisable to use flux. Layout of desired circuit diagram and preparation is first and most important.

operation in any printed circuit board manufacturing process. First of all layout of component side is to be made in accordance with available components dimensions.

The following points are to be observed while forming the layout of P.C.B:

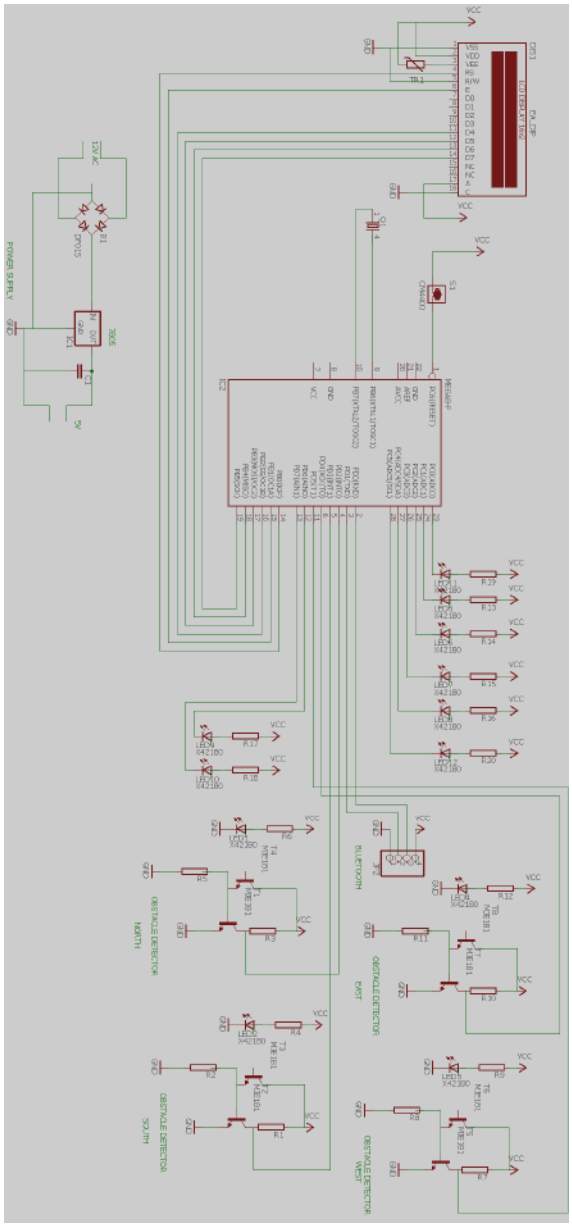
1. Between two components, sufficient space should be maintained.

2. High wattage/max, dissipated components should be mounted at a sufficient distance from semiconductors and electrolytic capacitors.

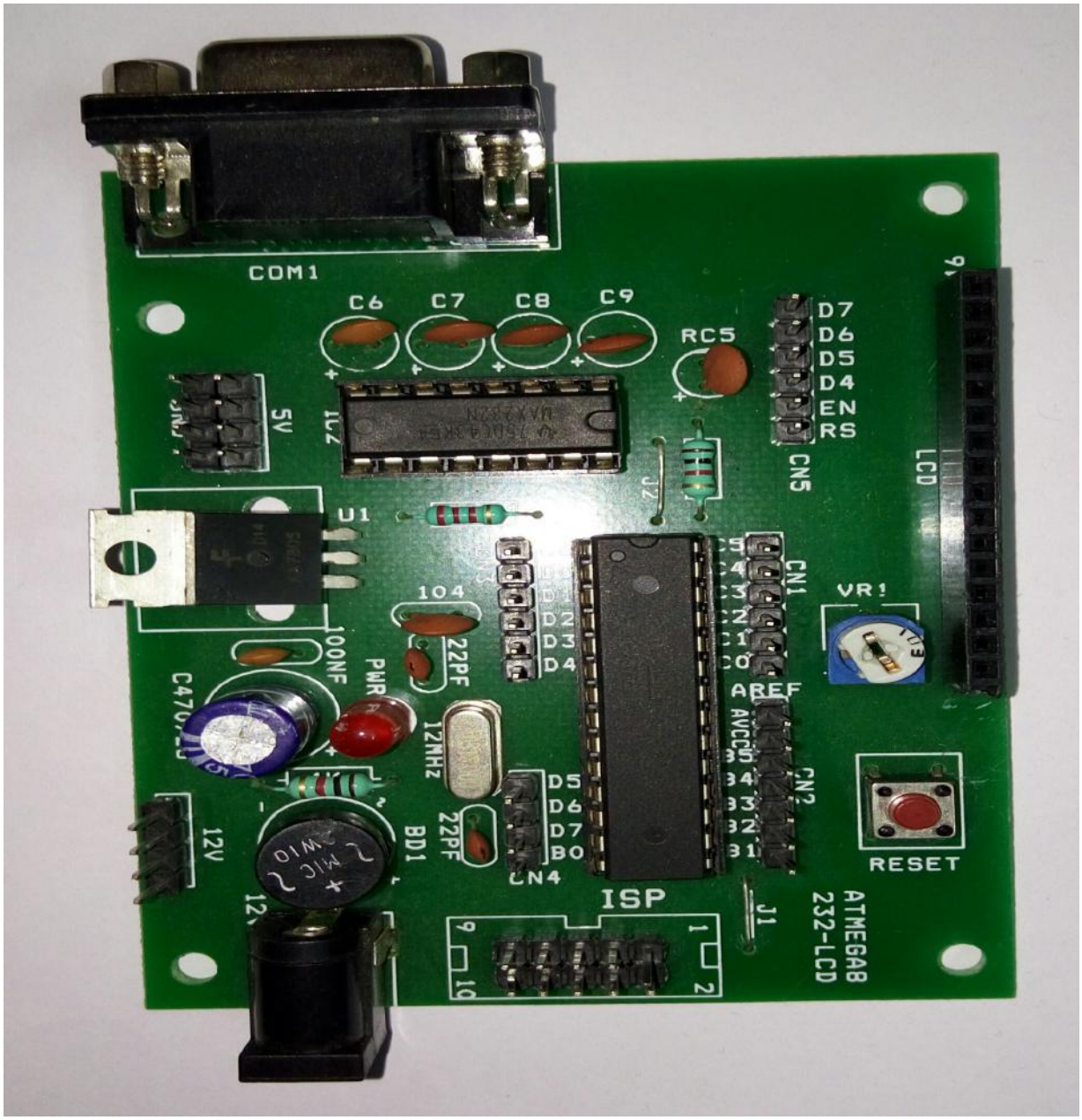
3. The most important point is that the components layout is making proper compression with copper side circuit layout.

Printed circuit board (P.C.B's) is used to avoid most or all the disadvantages of conventional bread board. These also avoid the use to thin wired for connecting (the components they are small in size and efficient in performance) the two most popular boards are widely used for general purpose application where the cost is to be low and the layout is simple.

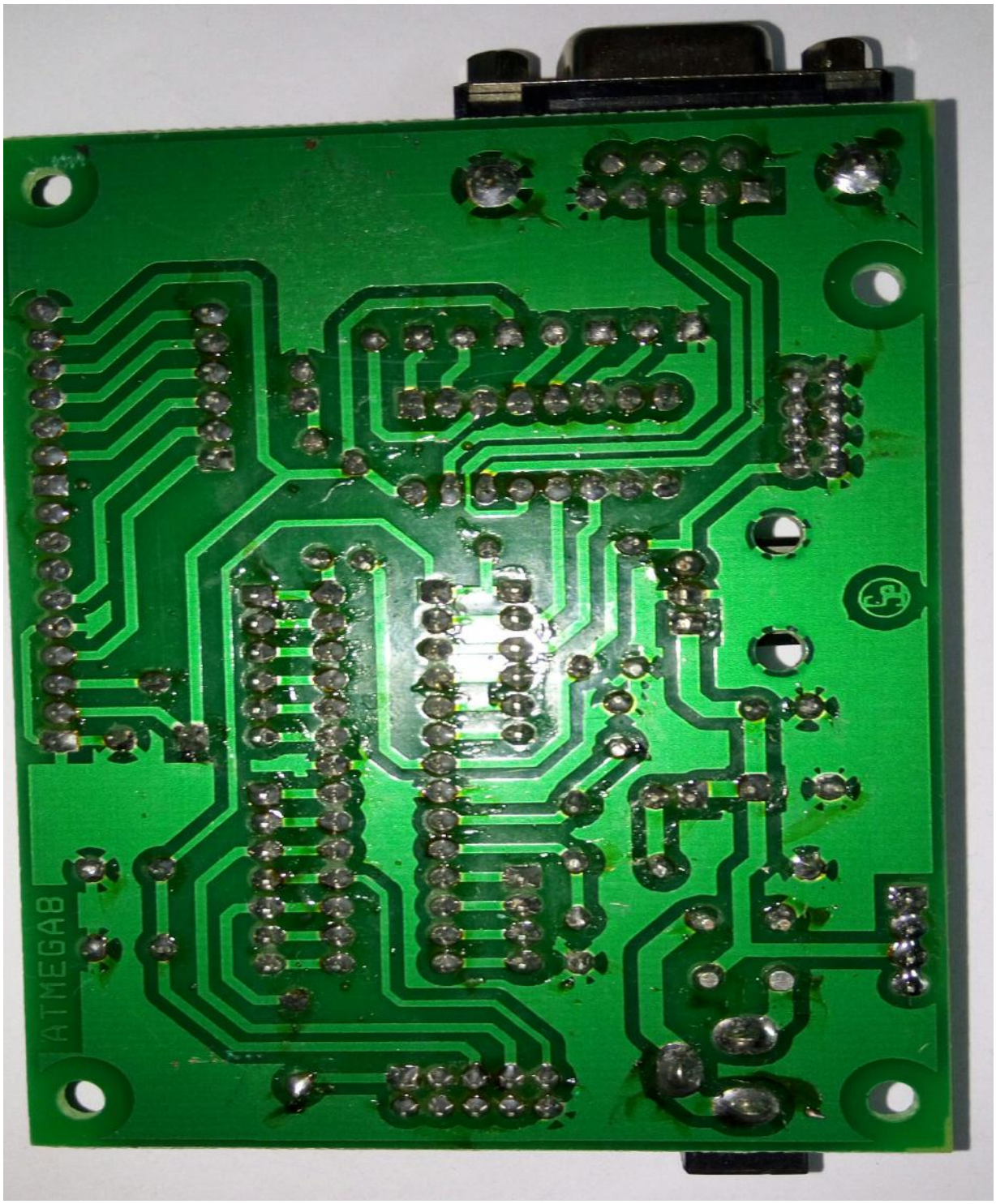
Schematic



PCB FRONT


COM1
C6
C7
C8
C9
RC5
D7
D6
D5
D4
EN
RS
CN5
LCD
5V
JZ
U1
104
CN1
VR1
D2
D3
D4
100NF
22PF
PWR
AREF
AVCC
12MHz
22PF
D5
D6
D7
B0
CN2
RESET
12V
BD1
CN4
ISP
J1
ATMEGA8
232-LCD

PCB BACK



Project Code:

Algorithm for code:

- When the system is turned ON then the system starts getting the values from the IR sensors.
- Generally, a normal delay is displayed to clear normal traffic.
- Based on the density on the roads which is calculated by the microcontroller.
- The traffic which is more is cleared by giving a green signal to that road and
- by stopping the remaining three.

CODE:

```
#include<string.h>

constintrf1 =A5;

constintrf2 =A4;

constintrf3 =A3;

constintrf4 =A2;

constinte_red =A0;

constinte_grn=A1;

constintw_red=12;

constintw_grn=13;

constintn_red=10;

constintn_grn=11;

constints_red=8;
```

```
constints_grn=9;

unsignedcharch,msgtype;

voidsetup()


pinMode(rf1,INPUT);

pinMode(rf2,INPUT);

pinMode(rf3,INPUT);

pinMode(rf4,INPUT);

pinMode(e_red,OUTPUT);

pinMode(w_red,OUTPUT);

pinMode(n_red,OUTPUT);

pinMode(s_red,OUTPUT);

pinMode(e_grn,OUTPUT);

pinMode(w_grn,OUTPUT);

pinMode(n_grn,OUTPUT);

pinMode(s_grn,OUTPUT);

Serial.begin(9600);

delay(200);

Serial.println("DENSITYBASEDTRAFFICLIGHTSCTRLSYS");

}
```

```
voidloop()

{

unsignedinti=0;


msgtype=0;

while(1)

{

msgtype=msgtype%4;

msgtype=msgtype+1;

start:

if(msgtype==1)

{

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(e_red,HIGH);

digitalWrite(e_grn,HIGH);

for(i=0;i<3000;i++)

{

if((digitalRead(rf1)==LOW)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==HIGH))

{
```

```
digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(e_red,HIGH);

digitalWrite(w_red,LOW);

digitalWrite(n_red,LOW);

digitalWrite(s_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("1   0           0           0");

while(digitalRead(rf1)==LOW)&&(digitalRead(rf1)==HIGH)&&
 (digitalRead(rf1)==HIGH)&&(digitalRead(rf1)==HIGH));
delay(1000);

msgtype=2;

gotostart;

}

if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==LOW)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==HIGH))

{

digitalWrite(e_grn,HIGH);
```

```
digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(w_red,HIGH);

digitalWrite(w_grn,LOW);

digitalWrite(e_red,LOW);

digitalWrite(n_red,LOW);

digitalWrite(s_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   0           1           0");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==LOW)&&
(digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=3;

gotostart;

}
if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)
==LOW)&&(digitalRead(rf4)==HIGH))
{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);
```

```
digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(n_red,HIGH);

digitalWrite(n_grn,LOW);

digitalWrite(e_red,LOW);

digitalWrite(w_red,LOW);

digitalWrite(s_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   1           0           0");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&

(digitalRead(rf3)==LOW)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=4;

gotostart;

}

if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==LOW))

{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);
```

```
digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(s_red,HIGH);

digitalWrite(s_grn,LOW);

digitalWrite(e_red,LOW);

digitalWrite(w_red,LOW);

digitalWrite(n_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   0           0           1");


while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&

(digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==LOW));

delay(1000);

msgtype=1;

gotostart;

}

delay(1);

}

digitalWrite(e_red,HIGH);

digitalWrite(e_grn,LOW);

digitalWrite(w_red,LOW);

digitalWrite(n_red,LOW);
```

```
digitalWrite(s_red,LOW);

if(msgtype==2)

{

digitalWrite(e_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(w_red,HIGH);

digitalWrite(w_grn,HIGH);

for(i=0;i<3000;i++)

{

if((digitalRead(rf1)==LOW)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==HIGH))

{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(e_red,HIGH);

digitalWrite(e_grn,LOW);

digitalWrite(n_red,LOW);

digitalWrite(s_red,LOW);

digitalWrite(w_red,LOW);
```

```
Serial.println("EASTWESTNORTHSOUTH");

Serial.println("1   0           0           0");

while(digitalRead(rf1)==LOW)&&(digitalRead(rf2)==HIGH)&&

(digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=2;

gotostart;

}

if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==LOW)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==HIGH))

{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(w_red,HIGH);

digitalWrite(w_grn,LOW);

digitalWrite(n_red,LOW);

digitalWrite(s_red,LOW);

digitalWrite(e_red,LOW);
```

```
Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   0           1           0");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==LOW)&&

(digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=3;

gotostart;

}

if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)

==LOW)&&(digitalRead(rf4)==HIGH))

{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(n_red,HIGH);

digitalWrite(n_grn,LOW);

digitalWrite(w_red,LOW);

digitalWrite(s_red,LOW);

digitalWrite(e_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");
```

```
Serial.println("0   1          0          0");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&

(digitalRead(rf3)==LOW)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=4;

gotostart;

}

if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==LOW))

{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(s_red,HIGH);

digitalWrite(s_grn,LOW);

digitalWrite(w_red,LOW);

digitalWrite(n_red,LOW);

digitalWrite(e_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   0          0          1");
```

```
while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&
 (digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==LOW));

delay(1000);

msgtype=1;

gotostart;

}

delay(1);

}

}

if(msgtype==3)

{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(n_red,HIGH);

digitalWrite(n_grn,HIGH);

for(i=0;i<3000;i++)

{

if((digitalRead(rf1)==LOW)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==HIGH))

{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);
```

```
digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(e_red,HIGH);


digitalWrite(e_grn,LOW);

digitalWrite(n_red,LOW);

digitalWrite(s_red,LOW);

digitalWrite(w_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("1   0           0           0");

while(digitalRead(rf1)==LOW)&&(digitalRead(rf2)==HIGH)&&
 (digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=2;

gotostart;

}
if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==LOW)&&(digitalRead(rf3)
==HIGH)&&(digitalRead(rf4)==HIGH))
{
digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);
```

```
digitalWrite(s_grn,HIGH);

digitalWrite(w_red,HIGH);

digitalWrite(w_grn,LOW);

digitalWrite(n_red,LOW);

digitalWrite(s_red,LOW);

digitalWrite(e_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   0           1           0");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==LOW)&&
(digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=3;

gotostart;

}
if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)
==LOW)&&(digitalRead(rf4)==HIGH))
{
digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(n_red,HIGH);
```

```
digitalWrite(n_grn,LOW);

digitalWrite(w_red,LOW);

digitalWrite(s_red,LOW);

digitalWrite(e_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   1           0           0");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&
(digitalRead(rf3)==LOW)&&(digitalRead(rf4)==HIGH));
delay(1000);

msgtype=4;

gotostart;

}
if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)
==HIGH)&&(digitalRead(rf4)==LOW))
{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(s_red,HIGH);

digitalWrite(s_grn,LOW);
```

```
digitalWrite(w_red,LOW);

digitalWrite(n_red,LOW);

digitalWrite(e_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   0           0           1");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&

(digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==LOW));

delay(1000);

msgtype=1;

gotostart;

}

delay(1);

}

}

if(msgtype==4)

{

digitalWrite(e_grn,HIGH);


digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_red,HIGH);

digitalWrite(s_grn,HIGH);
```

```
for(i=0;i<3000;i++)
{
if((digitalRead(rf1)==LOW)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)
==HIGH)&&(digitalRead(rf4)==HIGH))
{
digitalWrite(e_grn,HIGH);
digitalWrite(w_grn,HIGH);
digitalWrite(n_grn,HIGH);
digitalWrite(s_grn,HIGH);
digitalWrite(e_red,HIGH);
digitalWrite(e_grn,LOW);
digitalWrite(n_red,LOW);
digitalWrite(s_red,LOW);
digitalWrite(w_red,LOW);
Serial.println("EASTWESTNORTHSOUTH");
Serial.println("1   0          0          0");
while(digitalRead(rf1)==LOW)&&(digitalRead(rf2)==HIGH)&&
(digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==HIGH));
delay(1000);
msgtype=2;
gotostart;
}
```

```
if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==LOW)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==HIGH))

{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(w_red,HIGH);

digitalWrite(w_grn,LOW);

digitalWrite(n_red,LOW);

digitalWrite(s_red,LOW);

digitalWrite(e_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   0           1           0");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==LOW)&&

 (digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=3;

gotostart;

}

if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)

==LOW)&&(digitalRead(rf4)==HIGH))
```

```
{

digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(n_red,HIGH);

digitalWrite(n_grn,LOW);

digitalWrite(w_red,LOW);

digitalWrite(s_red,LOW);

digitalWrite(e_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   1           0           0");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&

(digitalRead(rf3)==LOW)&&(digitalRead(rf4)==HIGH));

delay(1000);

msgtype=4;

gotostart;

}

if((digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&(digitalRead(rf3)

==HIGH)&&(digitalRead(rf4)==LOW))

{
```

```
digitalWrite(e_grn,HIGH);

digitalWrite(w_grn,HIGH);

digitalWrite(n_grn,HIGH);

digitalWrite(s_grn,HIGH);

digitalWrite(s_red,HIGH);

digitalWrite(s_grn,LOW);

digitalWrite(w_red,LOW);

digitalWrite(n_red,LOW);

digitalWrite(e_red,LOW);

Serial.println("EASTWESTNORTHSOUTH");

Serial.println("0   0           0           1");

while(digitalRead(rf1)==HIGH)&&(digitalRead(rf2)==HIGH)&&

 (digitalRead(rf3)==HIGH)&&(digitalRead(rf4)==LOW));

delay(1000);

msgtype=1;

gotostart;

}

delay(1);

}

}

}
```

}

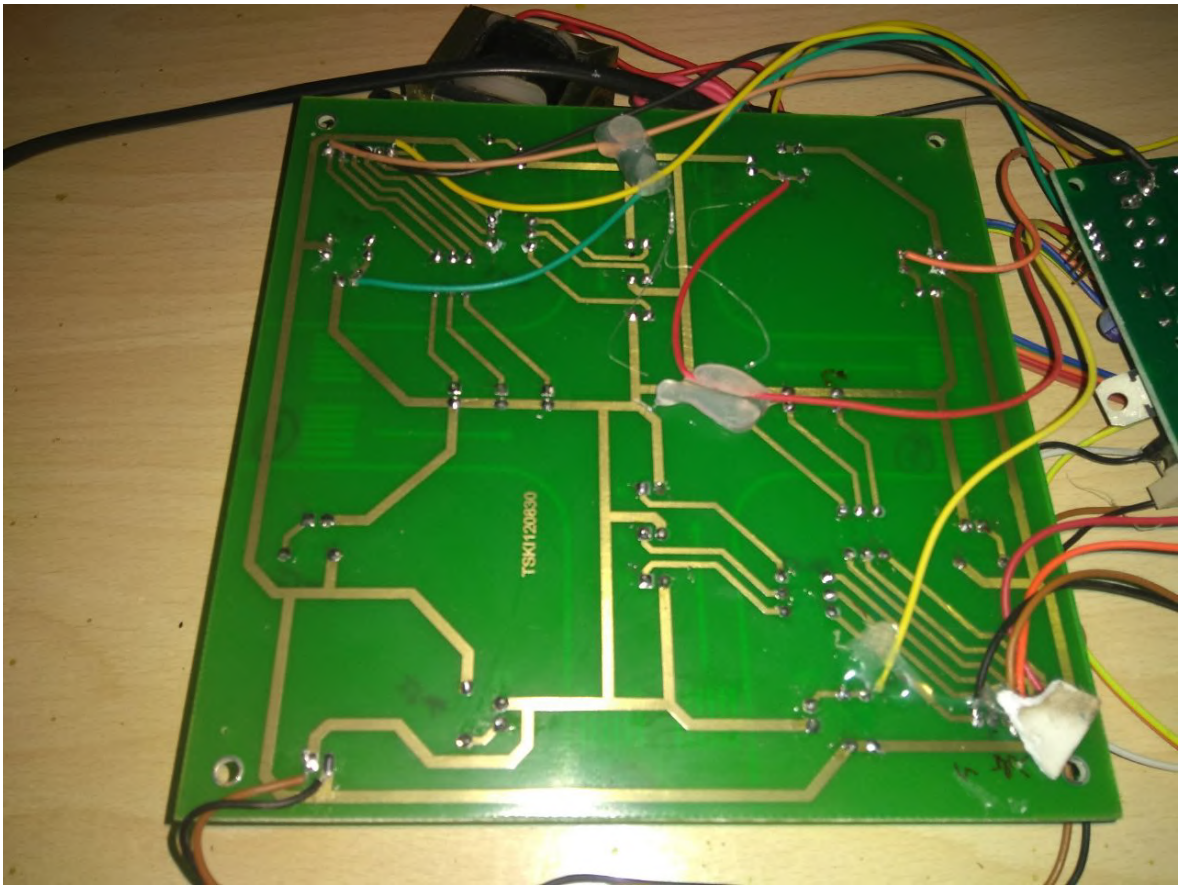
CONCLUSION

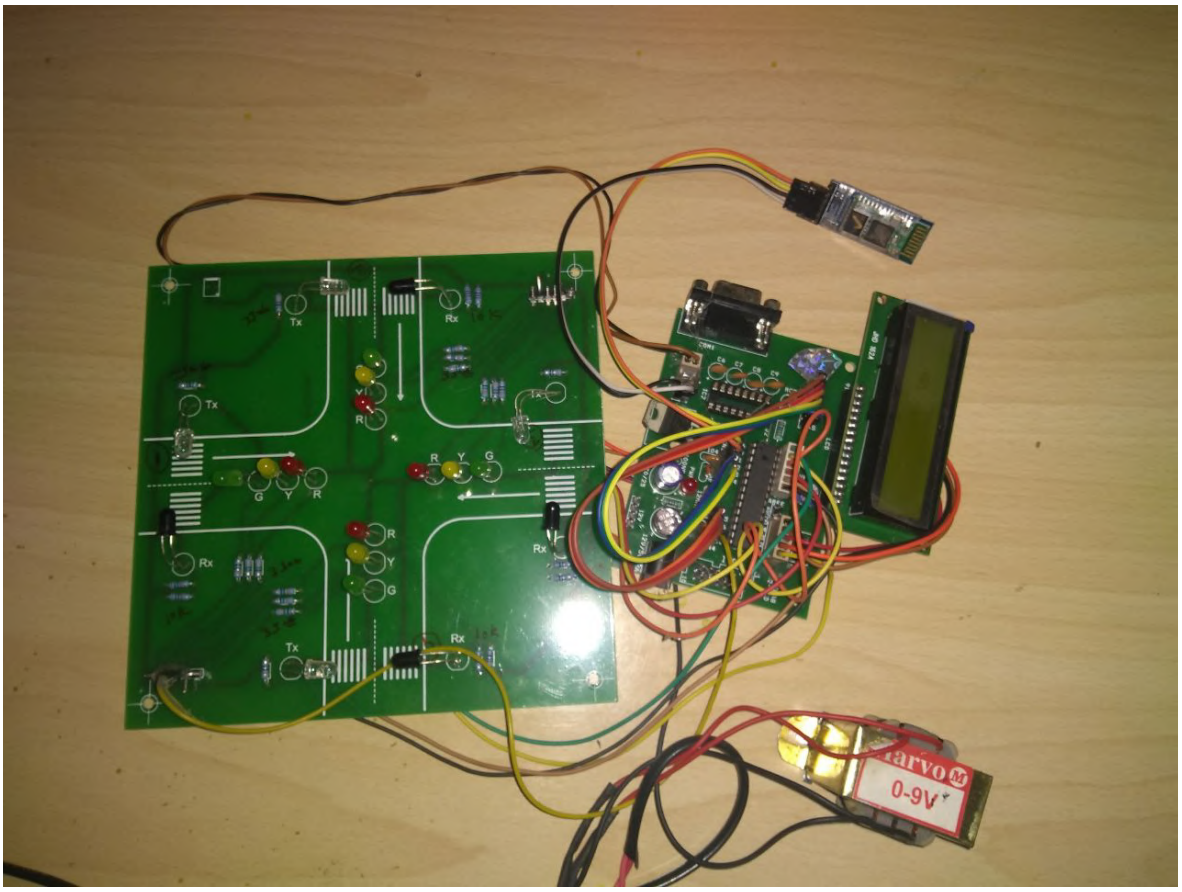
We conclude the Density measurement by using opencv tool as software for image processing by just displaying the various conversion of image in the screen and finally surrounding the box on the vehicle in the given image, the number of vehicle is calculated. We can calculate the density of the vehicle by using mat lab tool by comparing the four side of the image which is given as a input. we can simulate the result of the four given input image but this cannot be used in real time applications as it is very slow and the software is not free of cost like opencv to overcome this disadvantage of mat lab, opencv software is used which is very easy to install and is open source software and can be used in real time application in a quick manner. In this paper we have shown the density measurement in the signal by using opencv in the System

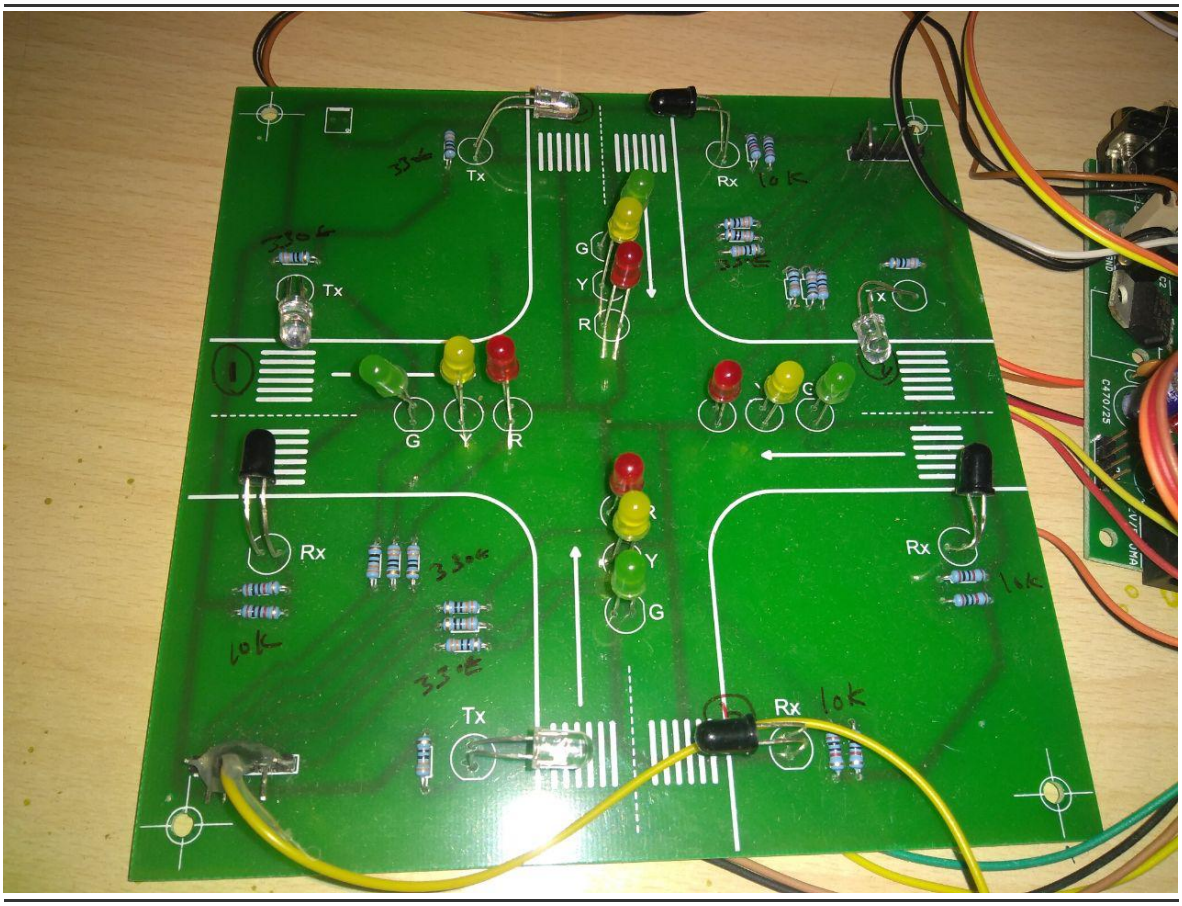
The sources that were reffered to complete this project are

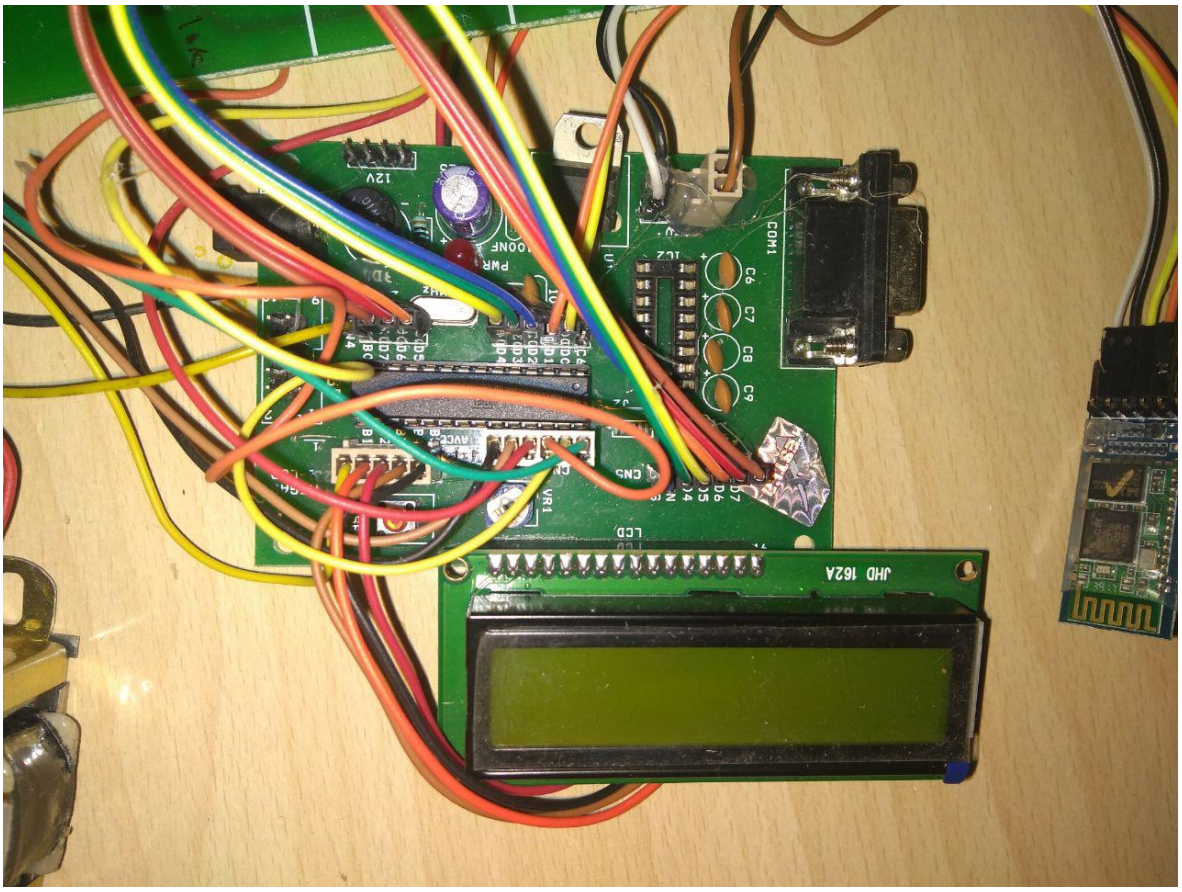
- Arduino official website for Arduino IDE
- Spark fun website for eagle libraries
- Eagle Official website for Eagle 7.0 software
- Youtube channel Jeremy Blum for creating schematic and board design using eagle software
- Circuit digest website for Circuit Diagram and Decoding the remote buttons

PHOTO GALLERY




Tx
Rx
R
Y
G
0-9V


Tx
Rx
G
Y
R
10K
330



Thanking You