TRAFFIC LIGHT USING SEVEN SEGMENT

by

ARYAN PANDEY
ANANYA B
SAKSHI AGNIHOTRI

TABLE OF CONTENTS

ABSTRACT

Traffic problems on roads in the modern cities are increasing day by day as the number of vehicles on the road is increasing. Traffic lights control systems are used as the signaling devices for managing challenging traffic in cities. Over the years there is a drastic change in technology which is implemented to improve work-efficiency, to reduce time losses.

Traffic junctions are generally controlled by traffic lights controller. The function of traffic lights requires sophisticated control and accurate coordination to ensure that traffic moves as safely and smoothly as possible.

But irritating, poor traffic systems are available which in turn causes accidents. To reduce inconvenience, the system which we will propose in our course will have continuous down count display of waiting time.

This system is also capable to modify the timings of traffic signals according to a requirement of the signals as the density of the vehicles vary from place to place.

INTRODUCTION

The main cause of the huge accidents is irresponsible driving and poor traffic control systems. There are many traffic control systems available but due to limited resources provided by available traffic systems with microcontroller are unknowingly leading to ever increase in travelling time and waiting time of people.

The project provides the basic model of traffic light control system which has an additional feature of displaying a countdown timer on a 7-segment display.

The main objective of this traffic light controller is to provide sophisticated control and coordination to confirm that traffic moves as smoothly and safely as possible.

This project makes use of LED lights for indication purpose and a microcontroller is used for auto changing of signal at specified range of time interval. LED lights gets automatically turns on and off by making corresponding port pin of the microcontroller “HIGH”.

WHY THIS MICROCONTROLLER??

Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.

FEATURES OF THIS MICROCONTROLLER

Arduino UNO Features and Technical Specs		
No.	**Parameter Name**	**Parameter Value**
1	Microcontroller	Atmega328
2	Crystal Oscillator	16MHz
3	Operating Voltage	5V
4	Input Voltage	5-12V
5	Digital I/O Pins	14 (D0 to D13)
6	Analog I/O Pins	6 (A0 to A5)
7	PWM Pins	6 (Pin # 3, 5, 6, 9, 10 and 11)
8	Power Pins	5V, 3.3V, Vin, GND
9	Communication	UART(1), SPI(1), I2C(1)
10	Flash Memory	32 KB (0.5KB is used by bootloader)
11	SRAM	2 KB
12	EEPROM	1 KB
13	ICSP Header	Yes
14	Power sources	DC Power Jack & USB Port

SYSTEM IMPLEMENTATION

Generally, every traffic signal system has three signal lights. The green light which is on the bottom of the signal indicates road users to proceed, a yellow light which is located in the middle alerts to slow and to be prepared to stop, and red light which is on the top indicates to stop. Fig.1 shows structure of a chowk consisting of four main roads and each road is divided into two main ways (north-south and east-west). We are using two traffic signals A,B. The signals on East-West road is A where B is a traffic signal on North-South road. Both traffic signalshave three lights. It is the basic road structure which includes only two ways to travel north-south and east-west so has two signal lights which are considered with signals of array of 3 digits for representation of red, yellow and green respectively. The road structure has also shown ‘zebra crossings’ for pedestrians so system has provided 2 seconds time period in which both the signals will be off to allow pedestrians to use roads.

WORKING PRINCIPLE

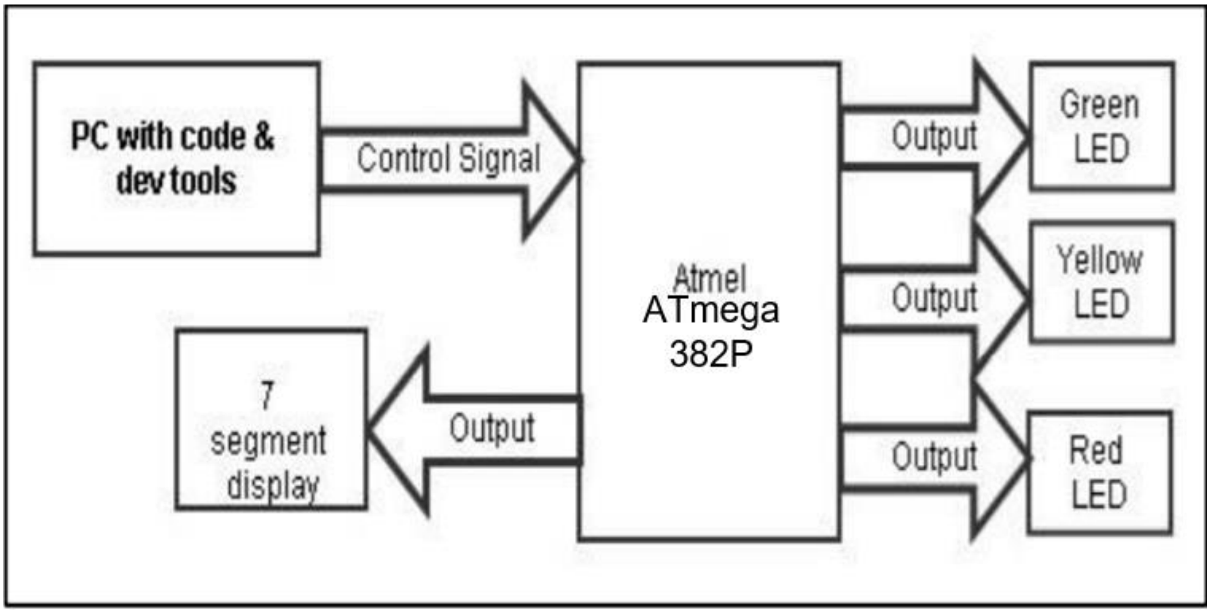
A seven-segment display is used as a counter display, and three LEDs are used for the purpose of traffic light control. A Microcontroller is the brain of this whole project and is used to initiate the traffic signal at the intersections on road. This circuit diagram makes use of a crystal oscillator for generating frequency clock pulses. The leds are interfaced to the Port zero of the microcontroller and are powered with 5v power supply. Seven-segment display is connected to the port1 pins of the microcontroller with a common anode configuration.

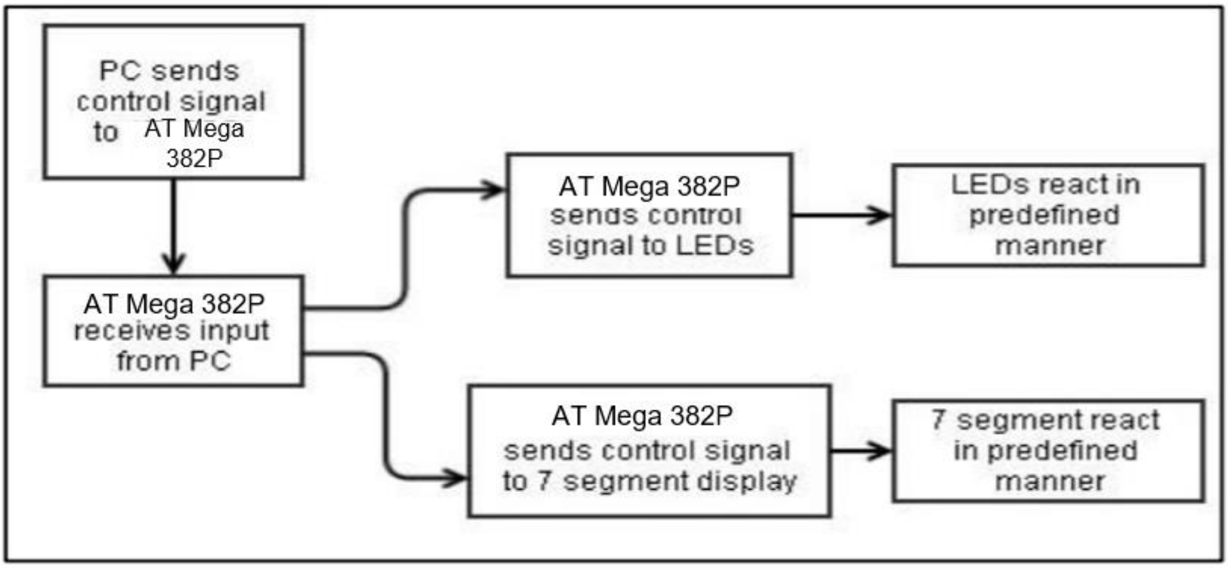
The leds get automatically switched on and off by making the corresponding port pins of the microcontroller high, based on the microcontroller and its programming done by using KEIL software.

At a particular period of time, only the green light holds ON and the other lights remains OFF, and after some time, the changeover traffic light control from green to red takes place by making the succeeding change for glowing of yellow LED. This process continues as a cycle and the timing for changing the leds can be displayed with the use of a seven-segment LED display in this project.

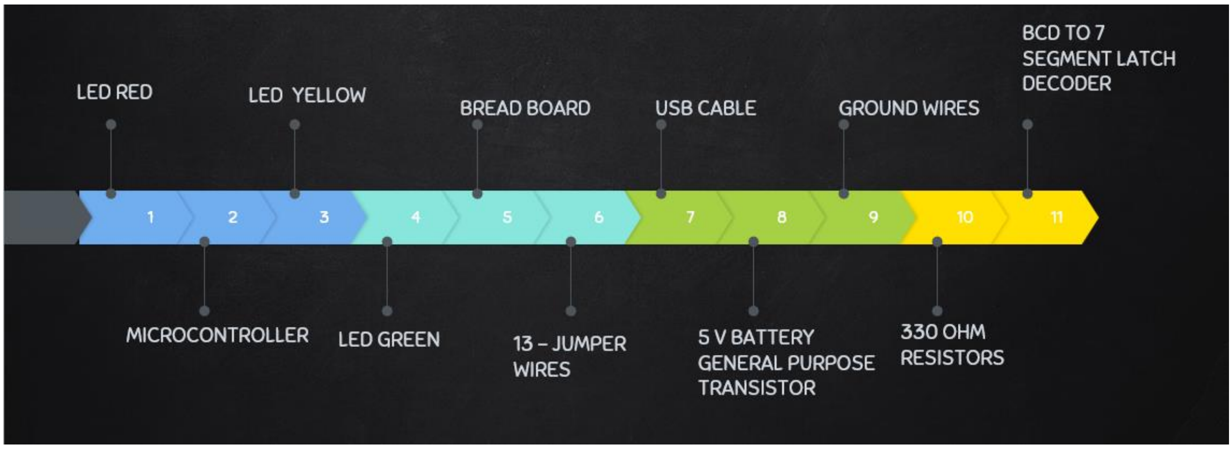
This traffic light control system can be further enhanced in such a way to control the traffic signals automatically based on the traffic density on roads with the help of IR sensor modules with automatic turnoff if there are no vehicles on either side of the road which leads to power consumption.

BLOCK DIAGRAM




PC sends control signal to AT Mega 382P
AT Mega 382P receives input from PC
AT Mega 382P sends control signal to LEDs
LEDs react in predefined manner
AT Mega 382P sends control signal to 7 segment display
7 segment react in predefined manner

COMPONENTS REQUIRED



LED RED

MICROCONTROLLER

LED YELLOW

LED GREEN

BREAD BOARD

13 – JUMPER WIRES

USB CABLE

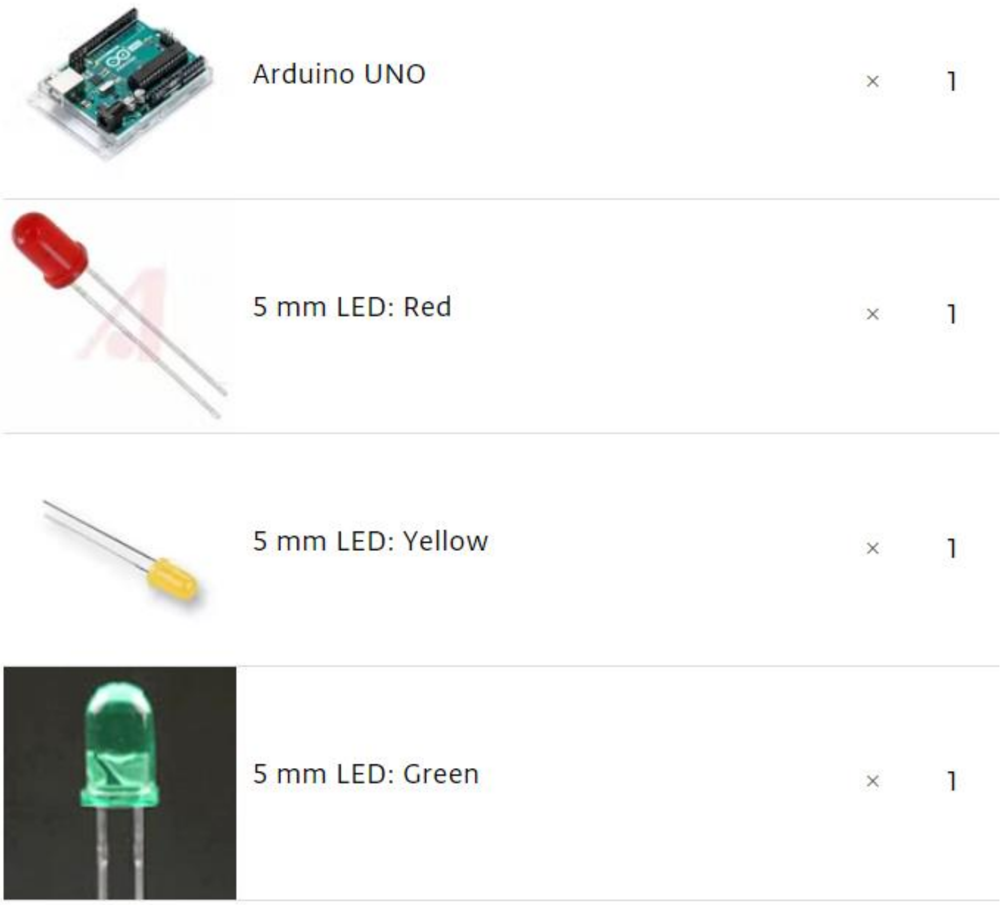
5 V BATTERY GENERAL PURPOSE TRANSISTOR

GROUND WIRES

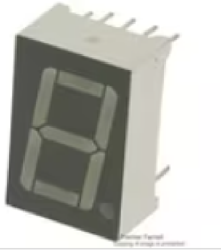
330 OHM RESISTORS

BCD TO 7 SEGMENT LATCH DECODER

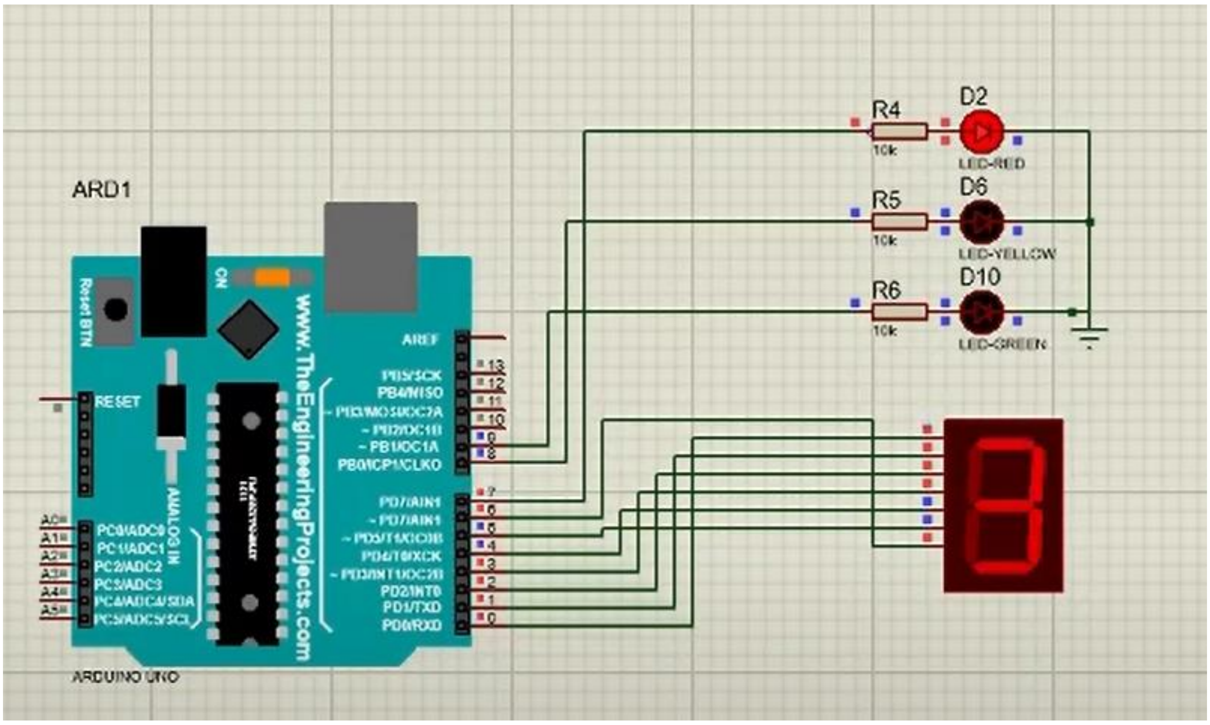
COMPONENTS AND SUPPLIES

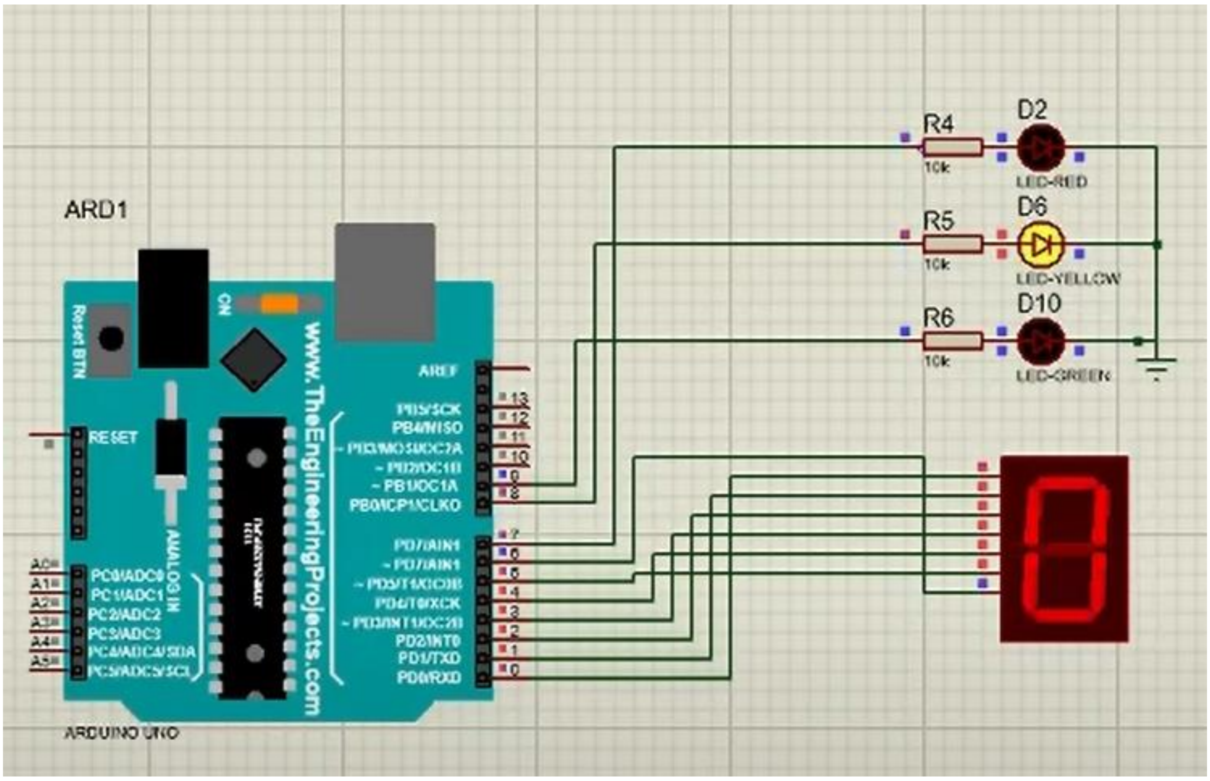


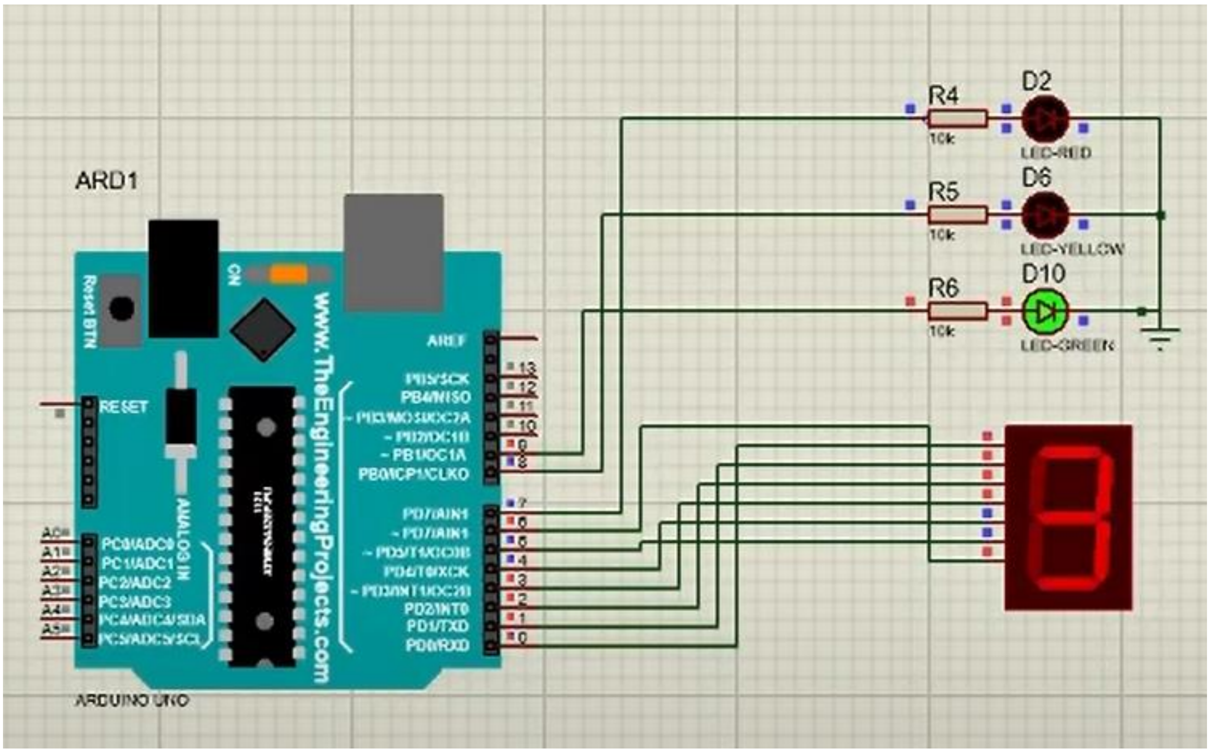
	Arduino UNO	×	1
	5 mm LED: Red	×	1
	5 mm LED: Yellow	×	1
	5 mm LED: Green	×	1

	Resistor 330 ohm	×	4
	7 Segment LED Display, InfoVue	×	1
	Jumper wires (generic)	×	1

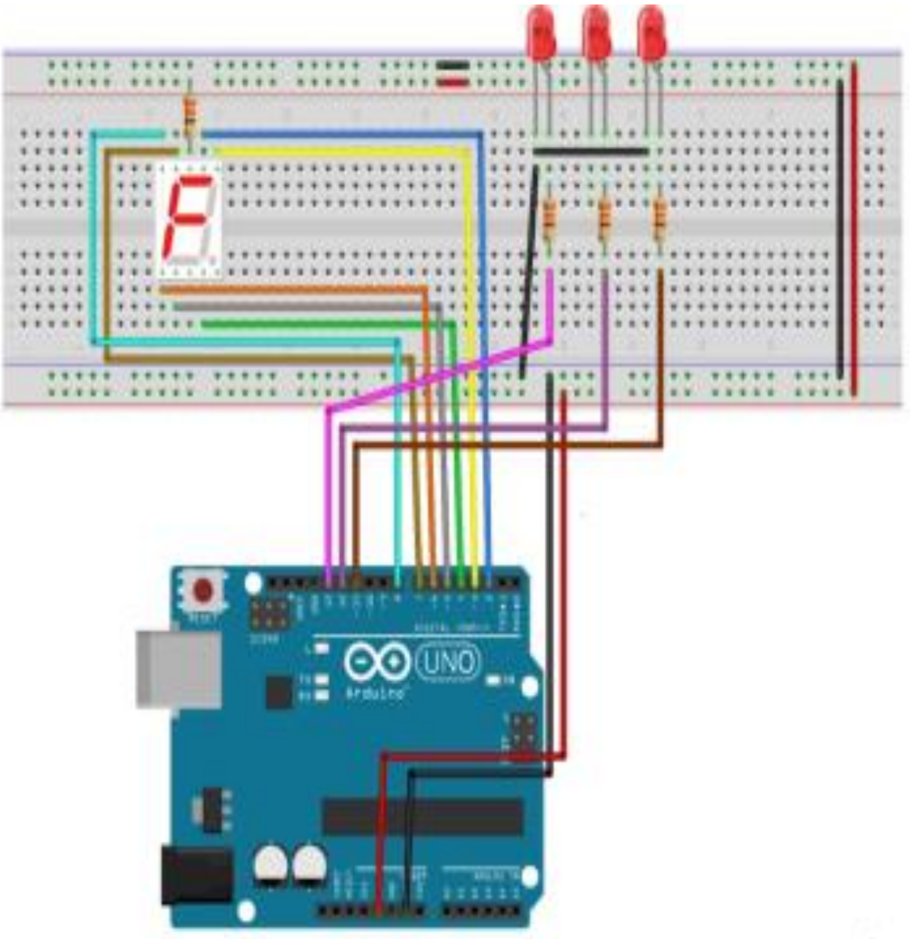
SOFTWARE ANALYSIS ON PROTEUS




ARD1
R4
10k
D2
LED-RED
R5
10k
D6
LED-YELLOW
R6
10k
D10
LED-GREEN
Reset BTN
RESET
ON
www.TheEngineeringProjects.com
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD6/AIN0
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
13
12
11
10
9
8
7
6
5
4
3
2
1
0
A0
A1
A2
A3
A4
A5
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
ANALOG IN
ARDUINO UNO


ARD1
R4
10k
D2
LED-RED
R5
10k
D6
LED-YELLOW
R6
10k
D10
LED-GREEN
Reset BTN
RESET
ON
www.TheEngineeringProjects.com
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD6/AIN0
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
13
12
11
10
9
8
7
6
5
4
3
2
1
0
A0
A1
A2
A3
A4
A5
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
ANALOG IN
ARDUINO UNO

CIRCUIT DIAGRAM



CODE:

```
// Traffic Lights with 7-Segment Timer
// Every 8 seconds, switch from Red Light to Green Light and vice-versa
// with 2 seconds delay in between for the Yellow Light
const int a=2; //a of 7-segment attach to digital pin 6
const int b=3; //b of 7-segment attach to digital pin 5
const int c=4; //c of 7-segment attach to digital pin 4
const int d=5; //d of 7-segment attach to digital pin 10
const int e=6; //e of 7-segment attach to digital pin 9
const int f=7; //f of 7-segment attach to digital pin 7
const int g=8; //g of 7-segment attach to digital pin 8

const int red=13; // Red LED attached to digital pin 11
const int yellow=12; // Yellow LED attached to digital pin 12
const int green=11; // Green LED attached to digital pin 13

void setup()
{
  // loop over thisPin from 4 to 13 and set them all to output
  for(int thisPin = 4; thisPin <= 13; thisPin++)
  {
    pinMode(thisPin,OUTPUT);
  }
}

void loop()
{
  // Initial State of LEDs
  digitalWrite(red,HIGH);
  digitalWrite(yellow,LOW);
  digitalWrite(green,LOW);
  countdown_from_8();

  // Countdown 2 seconds and turn on Yellow LED
  digitalWrite(red,LOW);
  digitalWrite(yellow,HIGH);
  digitalWrite(green,LOW);
  countdown_from_2();

  // Countdown 8 seconds and turn on Green LED
  digitalWrite(red,LOW);
  digitalWrite(yellow,LOW);
  digitalWrite(green,HIGH);
  countdown_from_8();

  // Countdown 2 seconds and turn on Yellow LED
  digitalWrite(red,LOW);
  digitalWrite(yellow,HIGH);
  digitalWrite(green,LOW);
  countdown_from_2();
}
```

```
void countdown_from_2()
{
  digital_2();// display 2 to the 7-segment
  delay(1000); //wait for a second
  digital_1();// display 1 to the 7-segment
  delay(1000); //wait for a second
  digital_0();// display 0 to the 7-segment
  delay(1000);//wait for a second
}

void countdown_from_8()
{
  digital_8();// display 8 to the 7-segment
  delay(1000);//wait for a second
  digital_7();// display 7 to the 7-segment
  delay(1000); //wait for a second
  digital_6();// display 6 to the 7-segment
  delay(1000); //wait for a second
  digital_5();// display 5 to the 7-segment
  delay(1000); //wait for a second
  digital_4();// display 4 to the 7-segment
  delay(1000); //wait for a second
  digital_3();// display 3 to the 7-segment
  delay(1000); //wait for a second
  digital_2();// display 2 to the 7-segment
  delay(1000); //wait for a second
  digital_1();// display 1 to the 7-segment
  delay(1000); //wait for a second
  digital_0();// display 0 to the 7-segment
  delay(1000);//wait for a second
}

void digital_0(void) // display 0 to the 7-segment
{
  for(int j = 4;j <= 10;j++)
  digitalWrite(j,HIGH);
  digitalWrite(g,LOW); // Turn off g of the 7-segment
}

void digital_1(void) // display 1 to the 7-segment
{
  for(int j = 4;j <= 10;j++)//turn off the others
  {
    digitalWrite(j,LOW);
  }
  digitalWrite(c,HIGH);//turn the c of the 7-segment on
  digitalWrite(b,HIGH);//turn the b of the 7-segment on
}
void digital_2(void) // display 2 to the 7-segment
{
  digitalWrite(b,HIGH);
  digitalWrite(a,HIGH);
  for(int j = 8;j <= 10;j++)
  digitalWrite(j,HIGH);
  digitalWrite(c,LOW);
  digitalWrite(f,LOW);
}
void digital_3(void) // display 3 to the 7-segment
{
  unsigned char j;
  digitalWrite(g,HIGH);
  digitalWrite(d,HIGH);
  for(j=5;j<=7;j++)
  digitalWrite(j,HIGH);
  digitalWrite(f,LOW);
  digitalWrite(e,LOW);
}
void digital_4(void) // display 4 to the 7-segment
{
  digitalWrite(c,HIGH);
  digitalWrite(b,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,HIGH);
  digitalWrite(a,LOW);
  digitalWrite(e,LOW);
  digitalWrite(d,LOW);
}
void digital_5(void) // display 5 to the 7-segment
{
  for(int j = 4;j <= 10;j++) {
    digitalWrite(j, HIGH);
  }
  digitalWrite(b,LOW);
  digitalWrite(e,LOW);
}
void digital_6(void) // display 6 to the 7-segment
{
  for(int j = 4;j <= 10;j++) {
    digitalWrite(j, HIGH);
  }
  digitalWrite(b,LOW);
}
void digital_7(void) // display 7 to the 7-segment
{
  for(int j = 4;j <= 10;j++) {
    digitalWrite(j, LOW);
  }
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
  digitalWrite(c,HIGH);
  digitalWrite(f,HIGH);
}
void digital_8(void) // display 8 to the 7-segment
{
  unsigned char j;
  for(j = 4;j <=10;j++)
  digitalWrite(j,HIGH);
}
```

ASM CODE:

```
ORG 0000H
MOV P1, #03H
LJMP start
MOV P1, #0CH
LJMP start
MOV P1, #30H
LJMP start
start: MOV R0, #0AH
MOV DPTR, #string
BACK:MOV A , #00H
LABEL: MOVC A, @A+DPTR
MOV P1,A
ACALL DELAY
INC DPTR
DJNZ R0, BACK
delay: MOV R2, #180
   again: MOV R3, #255
   here: DJNZ R3, here
DJNZ R2, again
string: db 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH,07H,7FH,6FH
RET
END
```

CONCLUSION

- Seven segments are widely used in digital clocks to display the time.
- These are used in electronic meters for displaying the numerical information.
- Used in Instrument panels and used in digital readout displays.
- The complexity is increased to display large information.
- It is not possible to display the symbols on seven segments.

In this Case Study we have studied the importance of embedded system in real life by implementing the optimization of Microcontroller-based Traffic Light Controller to reduce the possibility of traffic jams and improve the safety issues surrounding major traffics by applying accurate timing and delay over traffic light sequences. But this case study outcome can be enhanced in such way as to control automatically the signals depending on the traffic density on the roads.

REFERENCES: -

1. https://org0020.files.wordpress.com/2014/07/ct5003-traffic-light-casestudy.pdf

2. https://create.arduino.cc/projecthub/mohammadsohail0008/traffic-lights-using-7-segment-display-ace1c1

3. https://www.rs-online.com/designspark/what-is-arduino-uno-a-getting-started-guide#:~:text=Arduino%20UNO%20is%20a%20low,and%20motors%20as%20an%20output

4. https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html