***Traffic lights execution by arduino on proteus version 8***

The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

# About my project:

Iam designing a single pole and four pole traffic lights in software. I design the whole project software, for writing the code i used arduino IDE and for prototype I am using proteus version 8 software

Diving into the details of the project let me explain about the components I am using in the whole software.

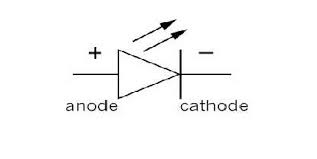
Here in my project i will be only using the digital pins to configure the entire led’s.

#### Electronic components:

1. Arduino board
2. LED’S- red , orange, green
3. Traffic lights - 4

#### Description of the components;

LED’S: The LED’s are abbreviated as light emitting diodes. A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it. When current passes through an LED, the electrons recombine with holes emitting light in the process. LEDs allow the current to flow in the forward direction and blocks the current in the reverse direction.

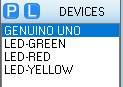
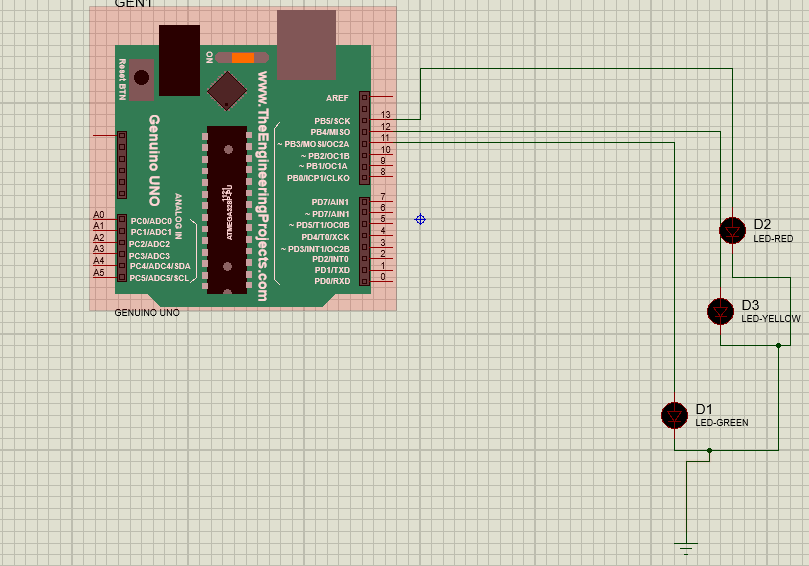


#### *Working of the project:*

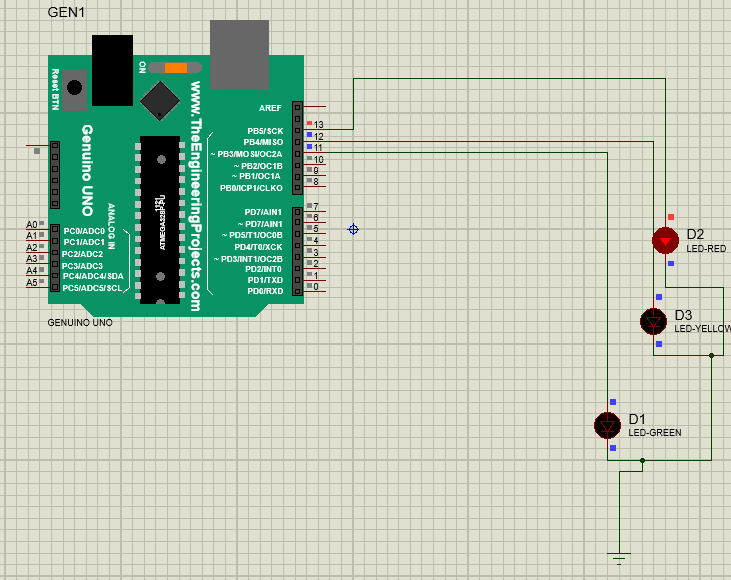
Single pole traffic light

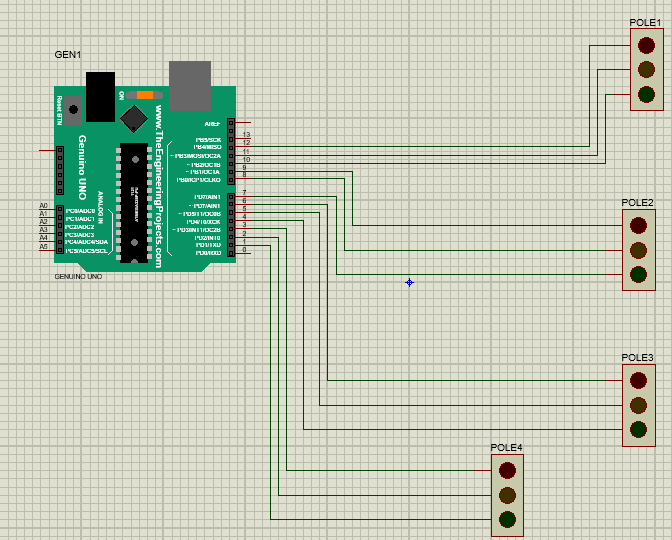
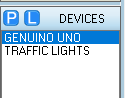
In this project i designed a single traffic light i.e only a single pole is present where i used arduino board and i wrote a coe to make the circuit work as the red should glow for 250 units of time when all the other leds are not glowing. Then switch the red led off for the rest of the operation. Then glows the orange led for 100 units of time. Then switches off and then green glows for 500 seconds goes back to orange and then red.

The circuit connection;

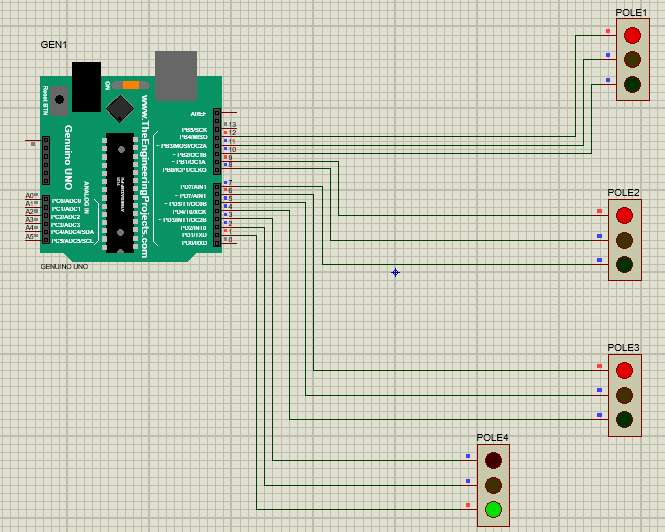


After execution the result is

For four pole traffic lights, i used a traffic light set as a pole instead of led’s. Here the condition for the traffic lights is that if one pole glows green the rest all 3 poles should be indicating red. The same timing formats are considered as the single pole .



Afterexecution the outcome will be



### *Code :*

Single pole

void setup() {

pinMode(13,OUTPUT);

pinMode(12,OUTPUT);

pinMode(11,OUTPUT);

}

void loop() {

digitalWrite(13,HIGH);

delay(250);

digitalWrite(13,LOW);

digitalWrite(12,HIGH);

delay(100);

digitalWrite(12,LOW);

digitalWrite(11,HIGH);

delay(500);

digitalWrite(11,LOW);

digitalWrite(12,HIGH);

delay(100);

digitalWrite(12,LOW);

}

4 pole traffic lights

void setup() {

pinMode(1,OUTPUT);//p4g

pinMode(2,OUTPUT);//p4y

pinMode(3,OUTPUT);//p4r

pinMode(4,OUTPUT);//p3g

pinMode(5,OUTPUT);//p3y

pinMode(6,OUTPUT);//p3r

pinMode(7,OUTPUT);//p2g

pinMode(8,OUTPUT);//p2y

pinMode(9,OUTPUT);//p2r

pinMode(10,OUTPUT);//p1g

pinMode(11,OUTPUT);//p2y

pinMode(12,OUTPUT);//p1r

}

void loop() {

//for pole1 green

digitalWrite(12,HIGH);

delay(100);

digitalWrite(9,HIGH);

digitalWrite(6,HIGH);

digitalWrite(3,HIGH);//all reds high

digitalWrite(12,LOW);//p1 turning to green

digitalWrite(11,HIGH);

delay(200);

digitalWrite(11,LOW);

// digitalWrite(12,LOW);

digitalWrite(10,HIGH);

delay(500);

digitalWrite(10,LOW);

digitalWrite(11,HIGH);

delay(100);

digitalWrite(11,LOW);

//for pole2 green

digitalWrite(9,HIGH);

delay(100);

digitalWrite(12,HIGH);

digitalWrite(6,HIGH);

digitalWrite(3,HIGH);//all reds high

digitalWrite(9,LOW);//p2 turning to green

digitalWrite(8,HIGH);

delay(200);

digitalWrite(8,LOW);

digitalWrite(7,HIGH);

delay(500);

digitalWrite(7,LOW);

digitalWrite(8,HIGH);

delay(100);

digitalWrite(8,LOW);

//pole 3

digitalWrite(6,HIGH);

delay(100);

digitalWrite(12,HIGH);

digitalWrite(9,HIGH);

digitalWrite(3,HIGH);//all reds high

digitalWrite(6,LOW);//p2 turning to green

digitalWrite(5,HIGH);

delay(200);

digitalWrite(5,LOW);

digitalWrite(4,HIGH);

delay(500);

digitalWrite(4,LOW);

digitalWrite(5,HIGH);

delay(100);

digitalWrite(5,LOW);

// for pole 4

digitalWrite(3,HIGH);

delay(100);

digitalWrite(12,HIGH);

digitalWrite(6,HIGH);

digitalWrite(9,HIGH);//all reds high

digitalWrite(3,LOW);//p2 turning to green

digitalWrite(2,HIGH);

delay(200);

digitalWrite(2,LOW);

digitalWrite(1,HIGH);

delay(500);

digitalWrite(1,LOW);

digitalWrite(2,HIGH);

delay(100);

digitalWrite(2,LOW);

}

#### *Note:*

The connections are to be done shown in my above pictures if the code is copy pasted.

The linking of proteus with arduino board will be given down . without linking the proteus itself cannot execute and reduce output.

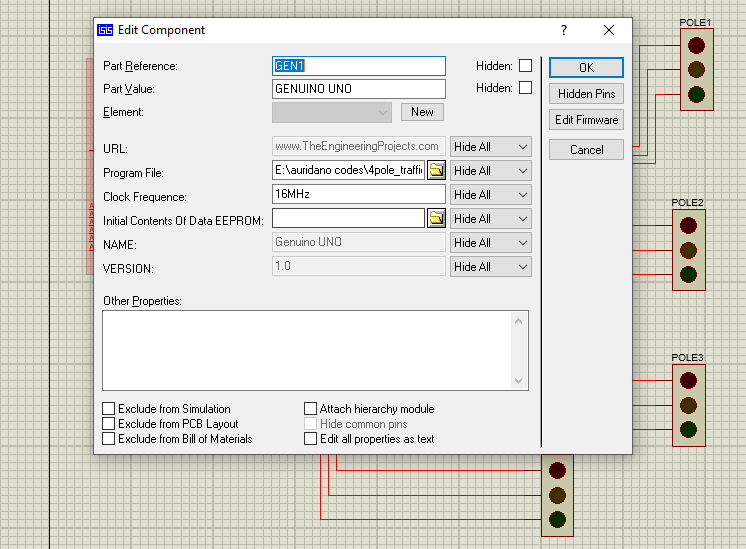
Steps for linking proteus and arduino

1. In arduino IDE for compiling the code

In taskbar > compile>export binary compile>select and code runs

There appears a command window below , search for .hex extension and copy paste the extension in the program file option in proteus

1. Double click on arduino board. A window pops up like this



Then click on the file option at the end of the program file and fetch the hex file of ur code

You have to compile and export the binary code in order to generate a hex file