

Name: Gagan Jain

Branch: IT

Year: TY

Roll No: 11

Sub: IOT LAB - ITL504

Experiment 3

Aim of the Experiment :

Designing of Traffic lights with Arduino.

Objective :

Sequence of the Traffic Lights

1. Red On 3sec

Yellow on 1sec

Off yellow and red

On green 3 sec

Off green 0.5 sec

Then blink green 0.5 sec

Again off the green 0.5 sec

On blink green 0.5 sec

Off green 0.5 sec

On blink green 0.5 sec

Off all the lights

Software/Hardware Required: Breadboard , Led (Red , Green ,Yellow) , jumperwire ,

1. Arduino Uno R3

2. Arduino Compiler

Theory:

Building a traffic light using an Arduino is a fun and educational project that demonstrates the basics of programming and hardware interfacing. In this theory

overview, I'll outline the steps to create a simple traffic light system using an Arduino board.

Designing traffic lights with Arduino involves creating a traffic control system that emulates the functionality of real-world traffic lights. Traffic lights play a crucial role in regulating the flow of traffic at intersections and pedestrian crossings. With Arduino, you can create a cost-effective and customizable traffic light system for various applications. The system's goal was to emulate the behavior of real-world traffic lights, regulating the flow of traffic at an intersection and possibly accommodating pedestrian crossings.

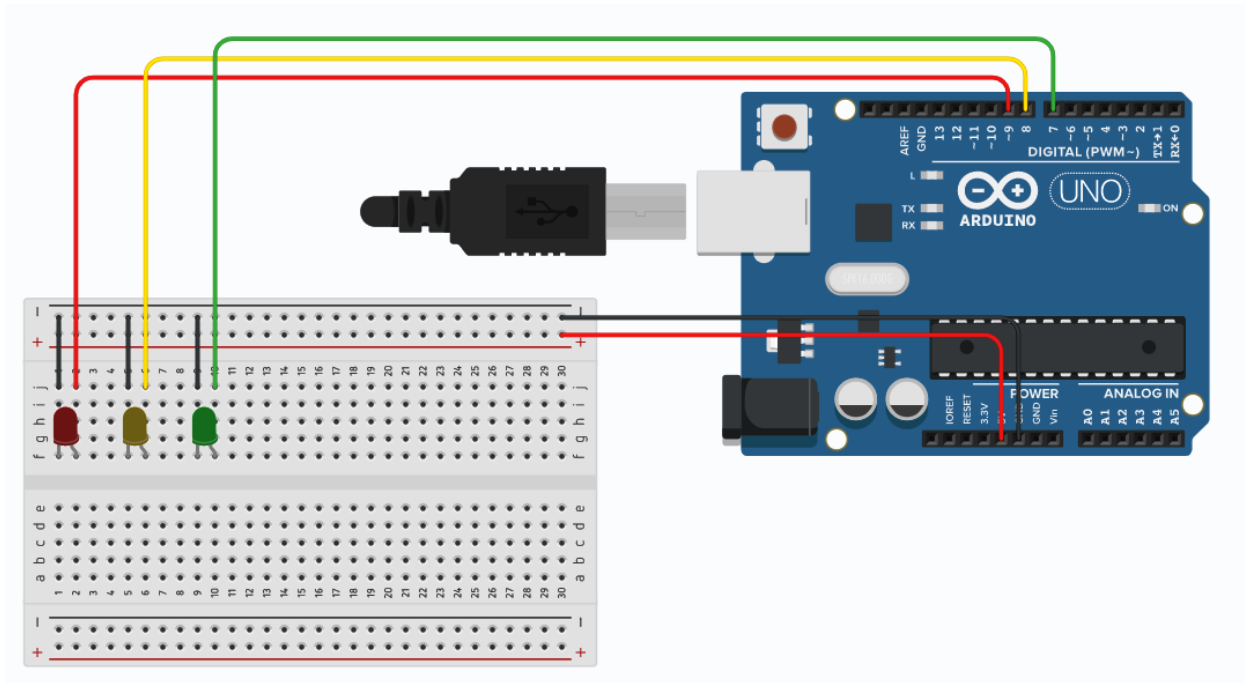
Theory of operation:

Connect the LEDs to the Arduino board using the breadboard and jumper wires. Place the LEDs on the breadboard, and connect the anode (longer leg) of each LED to a current-limiting resistor. Connect the other end of each resistor to different digital pins on the Arduino. Connect the cathode (shorter leg) of each LED to the ground (GND) pin on the Arduino.

In the Arduino IDE (Integrated Development Environment), write a simple code to control the traffic light. The code will define the pins used to control each LED and set the appropriate pin modes (input or output).

The traffic light should follow a sequence of states: red, red-yellow, green, yellow, and then back to red. You can define a delay for each state to simulate the timing of a real traffic light. The Arduino's loop function will repeatedly execute the traffic light sequence, causing the LEDs to turn on and off accordingly.

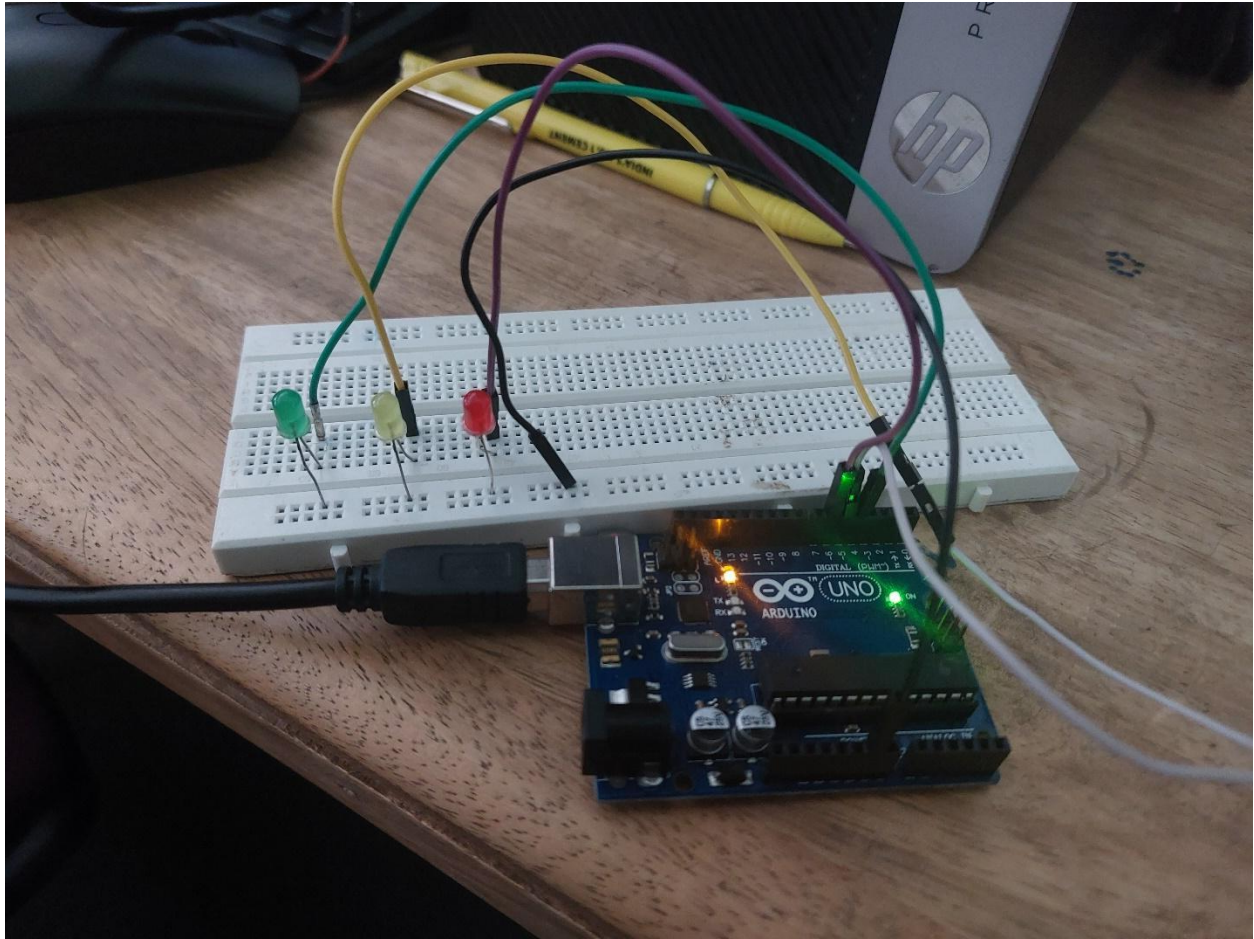
Circuit Diagram:



Code Screenshot:-

```
1  int rpin = 6;
2  int ypin = 5;
3  int gpin = 4;
4  int d1 = 3000;
5  int d2 = 2000;
6  int d3 = 1000;
7
8  void setup() {
9      // put your setup code here, to run once:
10     pinMode(rpin, OUTPUT);
11     pinMode(ypin, OUTPUT);
12     pinMode(gpin, OUTPUT);
13 }
14
15 void loop() {
16     // put your main code here, to run repeatedly
17     digitalWrite(rpin, HIGH);
18     delay(2000);
19     digitalWrite(ypin, HIGH);
20     delay(1000);
21     digitalWrite(rpin, LOW);
22     digitalWrite(ypin, LOW);
23
24     digitalWrite(gpin, HIGH);
25     delay(3000);
26     digitalWrite(gpin, LOW);
27     delay(500);
28
29     digitalWrite(gpin, HIGH);
30     delay(500);
31     digitalWrite(gpin, LOW);
32     delay(500);
33
34     digitalWrite(gpin, HIGH);
35     delay(500);
36     digitalWrite(gpin, LOW);
37     delay(500);
38
39     digitalWrite(gpin, HIGH);
40     delay(500);
41     digitalWrite(gpin, LOW);
42     delay(5000);
43 }
```

Output:



Conclusion:

In conclusion, building a traffic light using an Arduino is a rewarding and educational project that provides hands-on experience in programming and hardware interfacing. The project involves connecting LEDs to the Arduino board and controlling them through a simple code to simulate the behavior of a real traffic light. The theory of operation involves defining the pins for each LED, setting their modes, and creating a loop function that cycles through different states of the traffic light, such as red, red-yellow, green, and yellow, with appropriate delays to simulate the timing of a traffic light.