



EXPERIMENT 1.4

Student Name: Rohan Jaiswal
Branch: CSE
Semester: 5th
Subject Name: Internet of Things Lab

UID: 21BCS2856
Section/Group: 626-B
Date of Performance: 29/08/23
Subject Code: 21CSP-344

Aim : Develop a smart traffic light management system With the help of IoT.

Objectives : Learn about interfacing and learn about IoT programming.

Hardware and Software : 1 × Breadboard, 1 × Arduino Uno R3, 3 × LEDs (Red, Yellow, Green),
3 × 220Ω Resistor, 3 × Jumper.

Description:

Nowadays, everyone prefers a personal vehicle. Hence, the number of vehicles on the road is increasing continuously, which results in traffic jams. Traffic light controller helps to manage the traffic and to maintain proper traffic management. These systems are placed at the intersections of the road or at the crossings to avoid congestions and accidents. The systems indicate to the driver by using different colors of light. Therefore it is simple to avoid congestion at the intersections. Roads without any supervision or guidance can lead to traffic conflicts and accidents. Traffic signals are required for an orderly flow of traffic. A traffic signal is used as an instructing device that indicates the road user to act as per the displayed sign. Traffic lights allow everyone to cross the intersection point one by one, reducing conflicts between vehicles entering intersection points from different directions. It provides road safety, also helps to solve traffic in simple manners.

Connections:

- This is the circuit diagram for the traffic light controller using Arduino.
- Connect LEDs on the breadboard as Red, Yellow, and Green, respectively.
- Connect the negative terminal of the LED and connect the 220 Ohm resistor in series.
- Connect these negative terminals to the ground.
- Connect the positive terminal of the LEDs to the pins 2 to 10, respectively.
- Power the breadboard by using 5V and GND on the Arduino.

The code for this traffic light controller is simple and easy to understand. We have exhibited traffic lights for a three-way road. The LEDs will be glowing in a particular sequence to form an actual traffic light controller system. At a time, two red LEDs will glow, and one Green LED will be ON. Also, two yellow LEDs will be ON for one second. Yellow LED will Glow on each transition between RED to GREEN. In short first RED LED will glow for 5 Seconds, then YELLOW for 1 second, and then Green LED will ON for 5 seconds. In the void setup of the code, we have defined the pins for the LEDs as outputs from 2 to 10. In the void loop section, we have defined the functions to turn LEDs ON and OFF into the sequence.

Code:

```
int red1 = 10;  
int yellow1 = 9;  
int green1 = 8;  
  
// Traffic light two  
int red2 = 13;  
int yellow2 = 12;  
int green2 = 11;
```

```
void setup () {  
  // Traffic light one  
  pinMode (red1, OUTPUT);  
  pinMode (yellow1, OUTPUT);  
  pinMode (green1, OUTPUT);  
  // Traffic light two  
  pinMode (red2, OUTPUT);  
  pinMode (yellow2, OUTPUT);  
  pinMode (green2, OUTPUT);  
}
```

```
void loop () {
  changeLights ();
  delay(10000);
}

void changeLights () {
  // Both yellow lights turns on
  digitalWrite(green1, LOW);
  digitalWrite(yellow1, HIGH);
  digitalWrite(yellow2, HIGH);
  delay (7000);
  // turns off both yellow and turns on red1 and green2
  digitalWrite(yellow1, LOW);
  digitalWrite(red1, HIGH);
  digitalWrite(yellow2, LOW);
  digitalWrite(red2, LOW);
  digitalWrite(green2, HIGH);
  delay(7000);

  // both of the yellow lights turns on
  digitalWrite(yellow1, HIGH);
  digitalWrite(yellow2, HIGH);
  digitalWrite(green2, LOW);
  delay(3000);

  // turns off both yellow light and turns on green1 and red2
  digitalWrite(green1, HIGH);
  digitalWrite(yellow1, LOW);
  digitalWrite(red1, LOW);
  digitalWrite(yellow2, LOW);
  digitalWrite(red2, HIGH);
  delay(7000);
}
```

Output:

