

DR. D.Y. PATIL INSTITUTE OF TECHNOLOGY, PUNE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA
SCIENCE

ASSIGNMENT NO: 7

Title:

Create a program so that when the user enters "B" the green light blinks, "g" the green light is illuminated "y" the yellow light is illuminated and "r" the red light is illuminated

Aim:

Connectivity, configuration and control of LED using Arduino circuit under different conditions.

Objectives

1. Hardware platforms and operating systems commonly used in IoT systems.
2. Help the students in providing a good learning environment and also work with real time problems faced in day to day life.

Hardware Requirement: Arduino, LED, 220-ohm resistor etc.

Software Requirement: Arduino IDE

Theory: Create a simple Arduino traffic light project, a fun endeavor that can be completed in under an hour. Learn how to construct it using an Arduino and explore the circuit for an advanced variation

To build an Arduino Traffic Light Controller, in addition to the basic Arduino, you'll require the following components:

- 1x 10k-ohm resistor
- 1x push-button switch
- 6x 220-ohm resistors
- A breadboard
- Connecting wires
- Red, yellow, and green LEDs

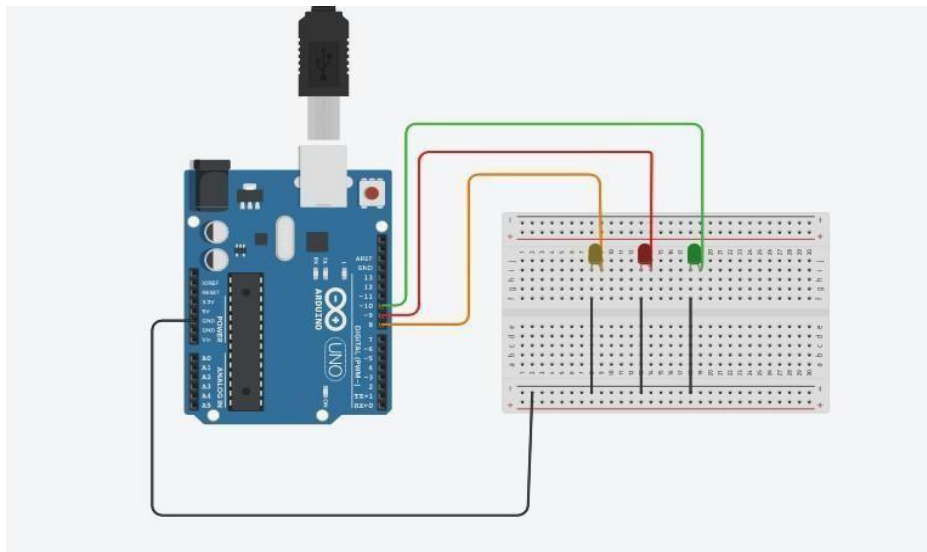
Arduino Traffic Light: Getting Started.

To begin, let's start with a simple single traffic light circuit.

Here's the basic circuit:

1. Connect the anode (long leg) of each LED to digital pins eight, nine, and ten (using a

2. Connect the cathodes (short leg) of the LEDs to the Arduino's ground



Syntax is as follow:

Begin by defining variables to address the lights by name rather than a number. Initiate a new Arduino project and start with these lines

```
int red = 10; int  
yellow = 9; int  
green = 8;
```

Next, let's incorporate the setup function, where you'll configure the red, yellow, and green LEDs as outputs. With the variables representing the pin numbers, you can now refer to the pins by name instead.

```
void setup(){ pinMode(red,  
    OUTPUT);  
pinMode(yellow, OUTPUT);  
    pinMode(green, OUTPUT);  
}
```

The pinMode function configures the Arduino to use a specified pin as an output. This is essential for the LEDs to function correctly

Conclusion : Thus we conclude program when the user enters "b" the green light blinks, "g" the green light is illuminated "y" the yellow light is illuminated and "r" the red light is illuminated

```
const int redPin = 9;

const int yellowPin = 10;

const int greenPin = 8;

char ch ;

void setup() {

  pinMode(redPin, OUTPUT);

  pinMode(yellowPin, OUTPUT);

  pinMode(greenPin, OUTPUT);

  Serial.begin(9600);

}

void loop()

{

  if(Serial.available()){

    ch=Serial.read();

  }

  if(ch=='R' || ch=='r'){

    digitalWrite(8, LOW);

    digitalWrite(9, HIGH);

    digitalWrite(10, HIGH);

  }

  if(ch=='G' || ch=='g'){

    digitalWrite(8, LOW);

    digitalWrite(9, HIGH);

    digitalWrite(10, HIGH);

  }

  if(ch=='B' || ch=='b'){

    digitalWrite(8, LOW);

    digitalWrite(9, HIGH);

    digitalWrite(10, HIGH);

  }

}
```

