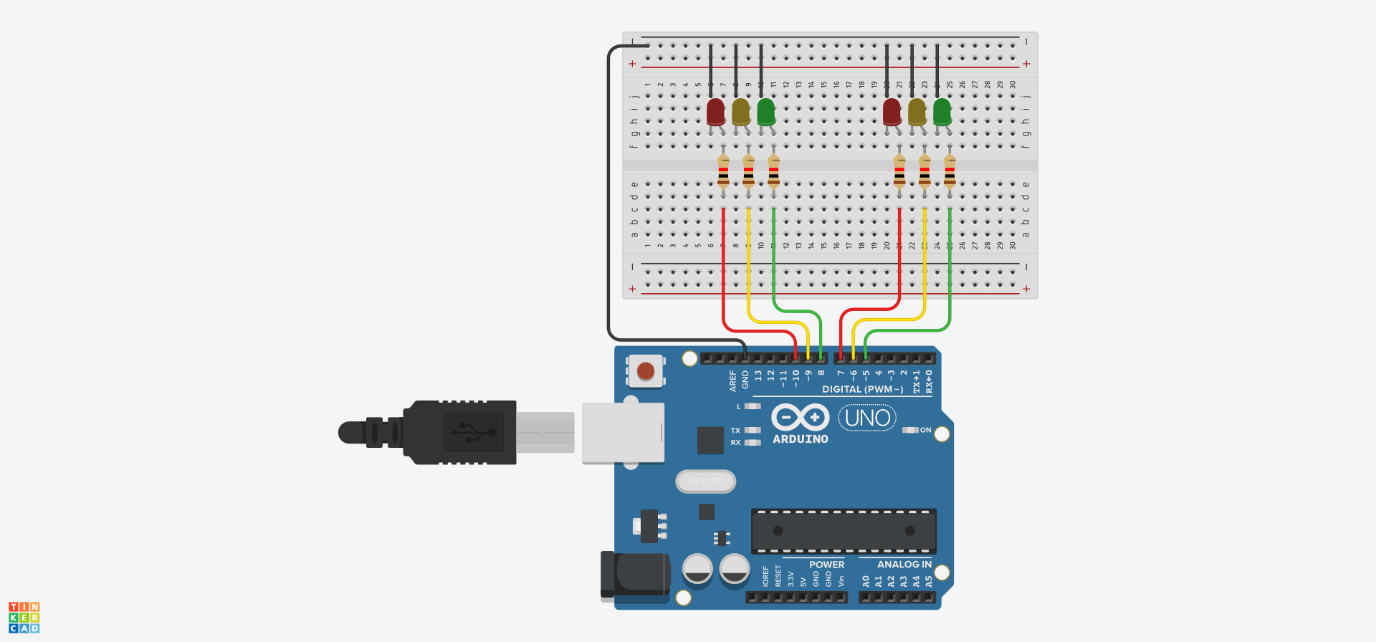
**Schematic diagram of the assembled circuit:**



**Program code in C++ using the Arduino IDE:**

// Constants

const short trafficLightSize **=** 3**;**

const int redAndGreenLightDuration **=** 10000**;**

const int yellowLightDuration **=** 3000**;**

// State variables

bool isGoingUp **=** **false;**

int currentLight **=** 2**;**

// Pin configuration for the two traffic lights (red, yellow, green)

const short firstTrafficLights**[**trafficLightSize**]** **=** **{**10**,** 9**,** 8**};**

const short secondTrafficLights**[**trafficLightSize**]** **=** **{**7**,** 6**,** 5**};**

void setup**()** **{**

initializeTrafficLights**();**

changeLightState**(**currentLight**,** HIGH**);**

**}**

void loop**()** **{**

waitBeforeTurningOff**(**currentLight**);**

changeLightState**(**currentLight**,** LOW**);**

updateLightDirection**();**

changeLightState**(**currentLight**,** HIGH**);**

**}**

void initializeTrafficLights**()** **{**

**for** **(**int i **=** 0**;** i **<** trafficLightSize**;** i**++)** **{**

pinMode**(**firstTrafficLights**[**i**],** OUTPUT**);**

pinMode**(**secondTrafficLights**[**i**],** OUTPUT**);**

**}**

**}**

void changeLightState**(**int trafficLightPosition**,** int state**)** **{**

digitalWrite**(**firstTrafficLights**[**trafficLightPosition**],** state**);**

digitalWrite**(**secondTrafficLights**[**getOppositeLightPosition**(**trafficLightPosition**)],** state**);**

**}**

int getOppositeLightPosition**(**int position**)** **{**

**return** abs**(**trafficLightSize **-** position **-** 1**);**

**}**

void waitBeforeTurningOff**(**int trafficLightPosition**)** **{**

int waitTime **=** **(**trafficLightPosition **==** 1**)** **?** yellowLightDuration **:** redAndGreenLightDuration**;**

delay**(**waitTime**);**

**}**

void updateLightDirection**()** **{**

**if** **(**isGoingUp**)** **{**

currentLight**++;**

**if** **(**currentLight **>=** trafficLightSize **-** 1**)** **{**

isGoingUp **=** **false;**

**}**

**}** **else** **{**

currentLight**--;**

**if** **(**currentLight **<=** 0**)** **{**

isGoingUp **=** **true;**

**}**

**}**

**}**

**Description of the circuit operation, including a detailed explanation of each line of code and the behavior of the circuit during operation:**

This circuit controls two sets of traffic lights (each with red, yellow, and green lights) in an alternating sequence. It uses an Arduino microcontroller to toggle the lights on and off based on a preset timing for red/green and yellow lights. The behavior of both traffic lights mirrors each other, creating a synchronized system.

**Code Explanation**

1. **Constants**:
   * trafficLightSize = 3: This defines the number of lights per traffic light (red, yellow, green).
   * redAndGreenLightDuration = 10000: This sets the duration (in milliseconds) for red and green lights to stay on (10 seconds).
   * yellowLightDuration = 3000: This sets the duration for yellow lights to stay on (3 seconds).
2. **State Variables**:
   * isGoingUp = false: This flag tracks the direction of light changes (from red to green or green to red).
   * currentLight = 2: This tracks which light is currently on (starts with green).
3. **Pin Configuration**:
   * firstTrafficLights[3] = {10, 9, 8}: These are the pin numbers for the first set of traffic lights (red, yellow, green respectively).
   * secondTrafficLights[3] = {7, 6, 5}: These are the pin numbers for the second set of traffic lights (red, yellow, green respectively).

**Functions**

1. **setup()**:
   * It calls initializeTrafficLights() to set up the traffic light pins as outputs.
   * The function then turns on the starting light (currentLight which is green) using changeLightState().
2. **loop()**:
   * It waits for the current light's delay using waitBeforeTurningOff().
   * Then, it turns off the current light using changeLightState(currentLight, LOW).
   * It updates which light should be on next using updateLightDirection().
   * Finally, it turns on the next light.
3. **initializeTrafficLights()**:
   * This sets up the pins for both traffic lights as outputs so that they can be controlled by the Arduino.
4. **changeLightState(int trafficLightPosition, int state)**:
   * This turns a specific light on or off by writing HIGH (on) or LOW (off) to the corresponding pins.
   * It also ensures that both sets of traffic lights are synchronized by turning on the "opposite" light in the second traffic light.
5. **getOppositeLightPosition(int position)**:
   * This calculates the opposite light in the second set of traffic lights to ensure synchronization.
   * For example, if the first traffic light is showing red (position == 0), the second traffic light will show green.
6. **waitBeforeTurningOff(int trafficLightPosition)**:
   * This pauses the program for a set amount of time before changing the light.
   * The delay time depends on whether the current light is yellow (yellowLightDuration) or red/green (redAndGreenLightDuration).
7. **updateLightDirection()**:
   * This changes the direction of light cycling (from red to green or green to red).
   * If it's currently moving up (green to yellow to red), it increments currentLight.
   * When the last light (red) is reached, it reverses direction and starts moving down (back toward green).

**Circuit Behavior**

1. **Starting**: When powered on, the system turns on the green light for first traffic light and red light for second traffic.
2. **Cycle**:
   * After 10 seconds, the green light turns off, and the yellow light turns on for 3 seconds.
   * After the yellow light, the red light turns on for 10 seconds.
   * Then, the sequence reverses: from red back to yellow and then to green.
3. **Synchronization**: While one traffic light shows green, the other shows red, and vice versa. The lights remain in sync throughout the sequence.