In this project, we will use an RGB LED and control its color by adjusting the brightness of its individual Red, Green, and Blue components using Pulse Width Modulation (PWM) with an Arduino. This will allow us to mix different intensities of the red, green, and blue channels to create a wide range of colors

Circuit Setup:

1. RGB LED Connections:
   - For Common Anode RGB LED:
     - The Anode pin (common pin) is connected to 5V.
     - The Red, Green, and Blue cathode pins are connected to PWM-capable pins on the Arduino (e.g., Pin 9, 10, and 11).
   - For Common Cathode RGB LED:
     - The Cathode pin (common pin) is connected to Ground (GND).
     - The Red, Green, and Blue anode pins are connected to PWM-capable pins on the Arduino (e.g., Pin 9, 10, and 11).
2. Resistors:
   - Place a 220Ω resistor in series with each RGB pin to limit the current flowing through the LED.

3. Power Supply:

    - The Arduino will power the RGB LED via its 5V pin, and all components share a common ground (GND).

Working Method:

1. PWM Signal:

    - The Arduino uses PWM to simulate the brightness of each color. For example, setting `analogWrite(redPin, 255)` turns the red component of the LED to full brightness, while `analogWrite(redPin, 0)` turns it off.

2. Color Creation:

    - By controlling the intensity of the Red, Green, and Blue channels, you can generate a variety of colors.

    - Color Mixing: Combining different intensities of the RGB components creates different colors. For example, if you have:

        - Red at full brightness (255), Green at medium brightness (128), and Blue at low brightness (64), it will create a unique shade of purple.

3. Cycle Through Colors:

    - The `loop()` function continuously changes the values sent to the RGB LED, cycling through various color combinations, creating a dynamic color-changing effect.