



RGB LED Implementation

Application Status RGB LED

The background features a complex network of thin, light-colored lines and small circles, forming a web-like structure. Overlaid on this are several translucent triangles of various sizes and orientations. The overall aesthetic is technical and minimalist.

Our Implementation

RGB LED Status Colors

- About the Implementation

- Create a function for initializing the RGB LED settings per channel (R, G, and B), GPIO for each channel and timer configuration.
- Create function for setting the colors based on duty cycle for each channel.
- Create specific color functions to use as application status indicators.
- We will then test these functions in the main.c file, but we will use them later to indicate the WLAN application status (*WiFi application started, HTTP server started, WiFi connected*).



LED Control (LEDC) ESP-IDF, in Brief

LED Control (LEDC) Peripheral

LED Control (LEDC) ESP-IDF

- About & Suggested Reading

- About LED Control (LEDC) and API Reference → <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/ledc.html>
 - Primarily designed for LED intensity control and PWM signal generation.
 - 16 Channels – 2 Groups → 8 channels for high-speed mode and 8 low-speed.
-

LED Control (LEDC) ESP-IDF

- Configuration Steps & Other Details

- Timer Configuration → Each timer counts upwards, and the bits defined, determines the count number before it resets, and frequency determines the amount of time it takes to count to that number. We will set members of the [`ledc_timer_config_t`](#) structure and pass the configuration to, [`ledc_timer_config`](#).
 - Channel Configuration → When defined, the GPIO pin that the PWM output signal appears on, is specified along with the timer that is associated with that Channel. We will set members of the [`ledc_channel_config_t`](#) struct and pass the configuration to, [`ledc_channel_config`](#).
 - Set/Update Duty Cycle → The time duration within a period that the PWM output signal will be high before it goes low. We'll manipulate the duty cycle to create colors, by calling [`ledc_set_duty`](#), then [`ledc_update_duty`](#).
-

The background features a complex network of thin, light-colored lines and small circles, forming a web-like structure. Overlaid on this are several translucent triangles of various sizes and orientations. The overall aesthetic is technical and modern, with a focus on geometric shapes and connectivity.

Creating the Status LED Colors

Color Functions

- RGB Set Color Function

- We'll create a function that sets the color by updating the duty cycle per channel (R, G, B)

- The Status Functions Call the Set Color Function at Specified Duty Cycles

Note: ([Use This Link for Color Selection](#) input parameters shown below)

- WiFi Started → RGB input values to our rgb_set_color function (255, 102, 255).
 - HTTP Server Started → RGB input values to rgb_set_color (204, 255, 51).
 - WiFi Connected → RGB input values to rgb_set_color (0, 255, 153).

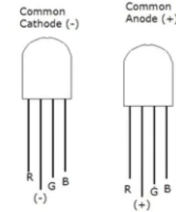
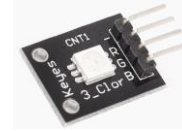
The background features a complex network of thin, light-colored lines and dots, forming various triangular shapes. Some triangles are filled with a very light yellow or green color, while others are just outlines. The overall effect is a subtle, technical, or network-like pattern.

Hardware Setup

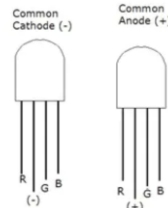
RGB LEDs

- Connecting RGB LEDs

- RGB LEDs come in different types e.g.



- Check the datasheet for your RGB LED to verify the required resistor values. Make connections with your jumper wires from the defined GPIOs (R, G, B) from the DevKit to the RGB LED.



- Common Anode/Cathode → Make the connections based on the diagram (note the long-leg difference – ground vs. positive voltage).

The background features a complex network of thin, light-colored lines and small circles, forming a web-like structure. Overlaid on this are several triangles of various sizes and orientations, some of which are filled with a light blue or yellow color. The overall aesthetic is clean and modern, with a focus on geometric shapes.

Let's get started!