

RGB LED

Red, green and blue in one package, one pin each.



Keep the resistors

Each colour's 1 k Ω resistor is the only thing limiting its current. **Never bridge them** or drive the LED's pads directly from a pin: the pin and the LED both run far past their ratings.

Six pins, four wires

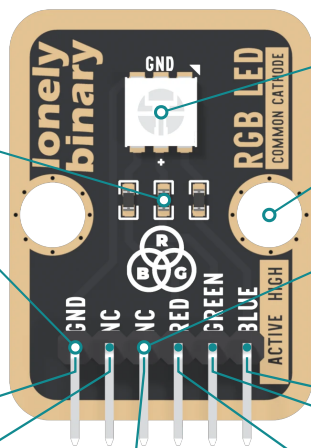
LED side up, header at the bottom. Wire GND, RED, GREEN and BLUE; the two NC holes are connected to nothing on the board.

3 $\times$ 1 k Ω , fitted

One per colour, between the pin and its anode.

GND is the square pad

The three cathodes join here. The back has no names.



5 $\times$ 5 mm RGB LED

Three chips behind one white diffuser.

2 mounting holes

$\varnothing$ 4.8 mm, 16 mm apart

Both NC pins

Connected to nothing on the board. Leave them.

	1	2	3	4	5	6
Front	GND	NC	NC	RED	GREEN	BLUE
On / off	GND	•	•	RED	GREEN	BLUE
Colour mixing	GND	•	•	RED	GREEN	BLUE

255, 255, 255 $\neq$ white

Green is far brighter per milliamp than blue. Equal numbers lean violet on 5 V and pink on 3.3 V.

Three colours, three currents

Red

3.1

mA from a 5 V pin

1.4 mA from 3.3 V. Forward voltage 1.95–2.35 V at 20 mA.

Green

2.2

mA from a 5 V pin

0.5 mA from 3.3 V. Forward voltage 2.75–3.35 V at 20 mA.

Blue

2.2

mA from a 5 V pin

0.5 mA from 3.3 V. Forward voltage 2.75–3.35 V at 20 mA.

Green gives several times the light of blue for the same current, so 255, 255, 255 is not white. Start near **167, 255, 122** on a 5 V board and trim by eye. On 3.3 V, red keeps far more headroom than green and blue, so the same numbers turn pink: start near 84, 255, 122.

Currents are worked out, not measured: the pin voltage less each LED's forward voltage at a few mA (red 1.9 V, green and blue 2.8 V, a little below the datasheet's 20 mA figures), over 1 k Ω . The white values come from the datasheet's light output and are a starting point only.

Four colours, no library

Arduino IDE · Board: your board · ESP32-S3: USB CDC On Boot: Enabled

```
// RED, GREEN, BLUE pins. Uno: 9, 10, 11. ESP32: 25, 26, 27.
// ESP32-S3: 4, 5, 6. Pico: 13, 14, 15.
const int RED_PIN = 4, GREEN_PIN = 5, BLUE_PIN = 6;

void colour(int r, int g, int b) {    // 0 is off, 255 is fully on
  analogWrite(RED_PIN, r);
  analogWrite(GREEN_PIN, g);
  analogWrite(BLUE_PIN, b);
}

void setup() {
  pinMode(RED_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(BLUE_PIN, OUTPUT);
}

void loop() {
  colour(255, 0, 0);   delay(1000); // red
  colour(0, 255, 0);   delay(1000); // green
  colour(0, 0, 255);   delay(1000); // blue
  colour(167, 255, 122); delay(1000); // near white on 5 V
}
```

Common cathode

The three LEDs share GND, so a colour lights when its pin is HIGH: 255 is fully on and 0 is off. A sketch written for a common-anode RGB LED counts the other way and shows every colour inverted. Change 255 - value back to value.

When it does not work

One colour never lights

That wire is on a different pin from the sketch, or one hole over: count from the square pad. Then check the pin is PWM-capable.

White looks pink or violet

Equal numbers are not white. Lower red and blue and trim by eye; on a 3.3 V board lower red further.

A colour flickers or steps

Nothing else in the sketch may use that pin's timer. On an Uno, Servo stops PWM on 9 and 10 and tone() disturbs 3 and 11.

Specifications

LED	5 × 5 mm PLCC-6, red, green, blue
At 5 V	Red 3.1, Green 2.2, Blue 2.2 mA
Logic	Common cathode, active high
Size	22.4 × 30.4 mm

Resistors	3 × 1 kΩ, one per colour, fitted
At 3.3 V	Red 1.4, Green 0.5, Blue 0.5 mA
Header	6 pins, right-angle, 2.54 mm
Mounting	2 × Ø 4.8 mm, 16 mm apart

Wiring

	FRONT	UNO	ESP32	S3	PICO
1	GND	GND	GND	GND	GND
2	NC				
3	NC				
4	RED	D9	GPIO 25	GPIO 4	GP13
5	GREEN	D10	GPIO 26	GPIO 5	GP14
6	BLUE	D11	GPIO 27	GPIO 6	GP15

Hole 1 is the square pad. Every pin in the table can do PWM. On an Uno only 3, 5, 6, 9, 10 and 11 can, and the Servo library takes PWM away from 9 and 10. On a classic ESP32 never use GPIO 6 to 11: they belong to the flash chip.

Every colour is inverted

The sketch was written for a common-anode LED. This board is common cathode: HIGH, or 255, is on.

Green and blue are faint on 3.3 V

Normal: they need about 2.8 V, so a 3.3 V pin leaves them about half a volt. A 5 V board gives them about four times the current.

Nothing lights at all

GND is not connected, or the cable is on the NC end. The three LEDs have no path home without GND.