QUICK REFERENCE · TK33

WS2812 RGB LED Bar

Five addressable RGB LEDs on one data wire: the top one keeps 24 bits and passes the rest down.

VCC takes 3.5 to 5.5 V: the 5V pin, not 3V3

A 3V3 pin is under the LEDs' 3.5 V minimum, and above 5.5 V is past their rating. **VCC goes to your board's 5V pin.** There is no resistor on DATA and no capacitor on the board, so make the wires with the power off.

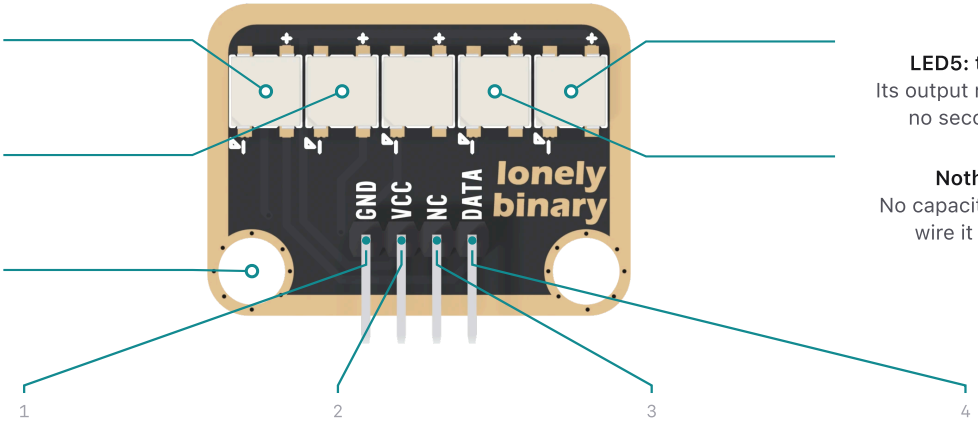
Pins, and which to wire

Drawn turned, the LEDs along the top with LED1 at the left, so the header lies along the bottom: **GND, VCC, NC, DATA**, GND on the square pad. DATA feeds LED1; NC is connected to nothing.

LED1: pixel 0
The only LED that reads your pin. Green first.

12 mA a colour
36 mA per LED at white, about 182 mA for five.

LEGO holes, 4.8 mm
24 mm apart, both on the header's side.



LED5: the chain ends
Its output reaches no pin: no second bar after it.

Nothing else fitted
No capacitor, no resistor: wire it with power off.

Front	GND	VCC	NC	DATA
Wire it	GND	VCC	●	DATA

Nothing leaves the fifth LED

LED5's data output is soldered to a pad that no trace reaches, so a second bar cannot follow this one, whatever the back of the board prints. Give a second bar its own GPIO, or the same GPIO to show the same colours on both.

What it draws, from 5V

One LED, one colour

12 mA at 255

About 2.2 mA at the sketch's brightness of 40.

One LED, white

36 mA at 255

Three colours at 12 mA each. About 5.9 mA at 40.

All five, white

182 mA at 255

The most one line of code can ask for. About 30 mA at 40.

Brightness is a share of time, so the current falls in proportion. All five at full white fit a USB port's 500 mA alone; with an ESP32-S3 on Wi-Fi, which peaks near 340 mA, they do not, so keep setBrightness low or give VCC its own 5 V.

Arithmetic on XINGLIGHT's XL-5050RGBC-WS2812B data sheet: 12 mA per colour, about 0.3 mA per LED dark; not measurements. The ESP32-S3 figure is its data sheet's Wi-Fi transmit peak. The sketch compiles for an ESP32-S3; the bar has not been bench-tested for this sheet.

The first colours, on an ESP32-S3

Arduino IDE · Board: ESP32S3 Dev Module · USB CDC On Boot: Enabled · Library: Adafruit NeoPixel

```
#include <Adafruit_NeoPixel.h>

const int DATA_PIN = 4;
const int LEDS = 5;   // five, and nothing after them

// GRB: this LED reads green first.
Adafruit_NeoPixel bar(LEDS, DATA_PIN, NEO_GRB + NEO_KHZ800);

// Top to bottom: red, green, blue, white, dark.
const uint32_t COLOUR[LEDS] = {
  0xFF0000, 0x00FF00, 0x0000FF, 0xFFFFFF, 0x000000
};
const char *NAME[LEDS] = {
  "red", "green", "blue", "white", "dark"
};

int step = 0;

void setup() {
  Serial.begin(115200);
  delay(500);
  bar.begin();
  bar.setBrightness(40);  // of 255: 30 mA at most
  bar.show();              // dark: clears power-up noise
}

void loop() {
  // Each colour moves one LED down a second.
  for (int i = 0; i < LEDS; i++) {
    bar.setPixelColor(i, COLOUR[(i + LEDS - step) % LEDS]);
  }
  bar.show();              // 120 bits, then a rest
  Serial.printf("LED1 (top) should be %s\n",
                NAME[(LEDS - step) % LEDS]);
  step = (step + 1) % LEDS;
  delay(1000);
}
```

```
LED1 (top) should be red
```

Once a second at 115200. If the top LED shows green when the line says red, the colour order is wrong: it must be NEO_GRB.

Wiring

	FRONT	ESP32-S3	ARDUINO UNO
1	**GND**	GND	GND
2	**VCC**	5V	5V
3	**NC**	—	—
4	**DATA**	GPIO 4	D6

Make the wires with the USB cable out. On an ESP32 the 5V pin is USB through a diode, about 4.6 V, which the LEDs are happy with. NC stays empty: a jumper there instead of in DATA leaves every LED dark.

Is a 3.3 V HIGH enough?

The LED reads DATA against its own supply: a HIGH must reach 0.7 × VCC. On an ESP32's 5V pin, about 4.6 V, that is 3.22 V, and a 3.3 V pin clears it by less than a tenth of a volt. On a full 5 V supply it is 3.5 V, and a 3.3 V pin is 0.2 V short on paper: it often works, and a fast level shifter such as a TXB0108 makes it certain. Only LED1 reads your pin, so a level problem shows as a wrong first LED.

When it does not work

Nothing lights

Check GND, then VCC on 5V, then DATA in the bottom hole and not in NC. Match DATA_PIN.

Only the first LED is wrong

The data level or the wire: keep DATA short, add the ground, VCC from the board's 5V.

A second bar stays dark

Nothing leaves LED5. Give the second bar its own GPIO and its own strip object.

Red and green swapped

The strip was made with NEO_RGB. These LEDs read green first: use NEO_GRB.

Resets when bright

Five at white is about 182 mA, plus the board's Wi-Fi. Lower setBrightness, or a 5 V supply.

Random colours at plug-in

DATA floats before the sketch runs. The dark frame in setup() clears it; wire it unpowered.

Specifications

LEDs	5 × XL-5050RGBC-WS2812B, 5 × 5 mm	**Supply**	3.5 to 5.5 V on VCC, the 5V pin
Current	12 mA a colour; about 182 mA, all white	**Data**	800 kbit/s, 24 bits per LED, green first
Data HIGH	At least 0.7 × VCC	**Frame end**	DATA low for 80 µs or more
Header	GND, VCC, NC, DATA; right-angle	**Board**	22.4 × 30.4 mm, header fitted