

## Programovatelný RGB LED čtverec



### POPIS

Modul je osazen 64 programovatelnými RGB LED diodami. LED diody jsou připojeny na DPS. Modul disponuje vstupním a výstupním datovým pinem. Vstupním pin slouží pro příjem dat z mikrokontroléru. Výstupní pin slouží pro připojení dalšího modulu či jiného uskupení LED diod WS2812B.

Základní charakteristika:

- čtvercové provedení – 64 LED diod
- napájení 5 VDC
- vstupní i výstupní datový pin
- regulace jasu a barev (RGB)



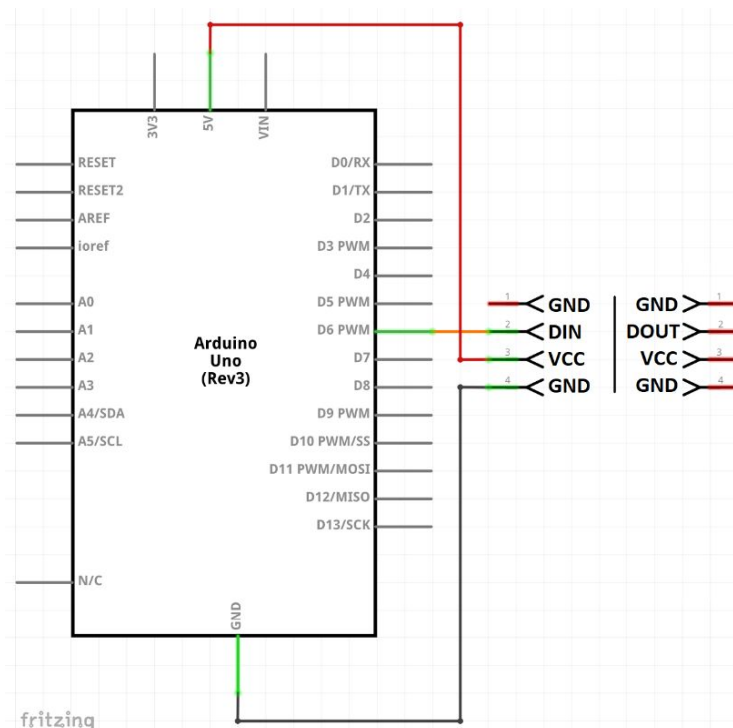
### SPECIFIKACE

<b>Typ LED diod</b>	WS2812B	<b>Rozměry</b>	65 x 65 mm
<b>Počet LED diod</b>	64	<b>Průměr mont. otvorů</b>	2,5 mm
<b>Barevné spektrum</b>	RGB	<b>Rozteč mont. otvorů</b>	61 x 8 mm
<b>Napájení</b>	5 VDC	<b>Typ pinů</b>	pájecí
<b>Max. proud</b>	2120 mA	<b>Hmotnost</b>	21 g



## ZAPOJENÍ

Z důvodu vysokého proudového odběru modulu (při plném rozsvícení LED diod) je vhodné využít pro napájení externí zdroj s příslušným výkonem (min. cca 2,2 A).



## UKÁZKA PROGRAMU

Aby bylo možné program zkompileovat, je nutné stáhnout knihovnu [NeoPixel](#). První ukázkový program rozsvítí na maximum všechny LED diody na modulu. Druhý program vytvoří tzv. duhu.

### Ukázka 1:

```
#include <Adafruit_NeoPixel.h>
#define INPUT_PIN 6
#define NUM_OF_LED 64
Adafruit_NeoPixel rgbWS = Adafruit_NeoPixel(NUM_OF_LED, INPUT_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  rgbWS.begin();
  rgbWS.setBrightness(255); //max jas
}

void loop() {
  for(int i = 1; i < NUM_OF_LED + 1; i++){
    rgbWS.setPixelColor(i, rgbWS.Color(255, 255, 255));
    rgbWS.show();
  }
}
```

## Ukázka 2:

```
#include <Adafruit_NeoPixel.h>
const int ledPin = 6;
const int numLeds = 64;
Adafruit_NeoPixel strip = Adafruit_NeoPixel(numLeds, ledPin, NEO_GRB + NEO_KHZ800);

void setup() {
  strip.begin();
  strip.setBrightness(80); // 1/3 brightness
}

void rainbow(uint8_t wait);
uint32_t wheel(byte WheelPos);

void loop() {
  rainbow(30);
  delay(10);
}

void rainbow(uint8_t wait) {
  for (uint16_t j = 0; j < 256; j++) {
    for (uint16_t i = 0; i < strip.numPixels(); i++) {
      strip.setPixelColor(i, wheel((i * 1 + j) & 255));
    }
    strip.show();
    delay(wait);
  }
}

uint32_t wheel(byte WheelPos) {
  if (WheelPos < 85) {
    return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
  }
  else if (WheelPos < 170) {
    WheelPos -= 85;
    return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
  }
  else {
    WheelPos -= 170;
    return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
  }
}
```