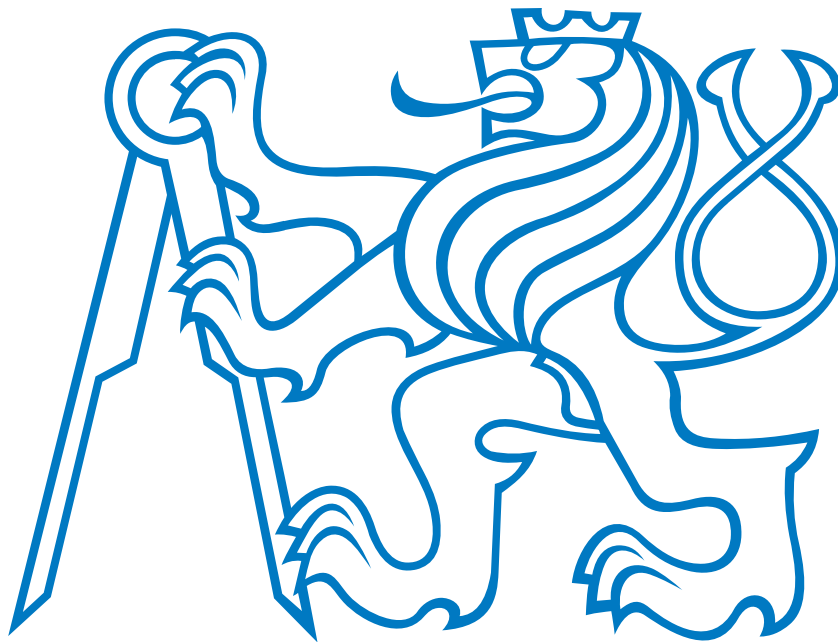


České vysoké učení technické v Praze  
Fakulta elektrotechnická

Katedra měření

Dokumentace  
**Adresovatelné RGB LED**

Zpracoval Ondřej Hruška  
10. dubna 2017



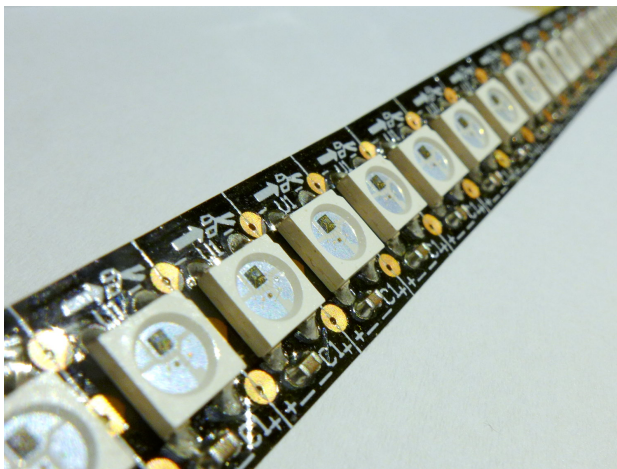
## 1 Základní informace

Adresovatelné (číslicově řízené) RGB LED se používají jako pixely ve světelných billboardech, jako náhrada neonových trubíc s možností různých barevných efektů, jako vánoční dekorace („světelní hadi“) a podobně. Jedná se o LED s vestavěným řadičem, který obstarává napájení jednotlivých segmentů a řízení jejich jasu pomocí PWM.

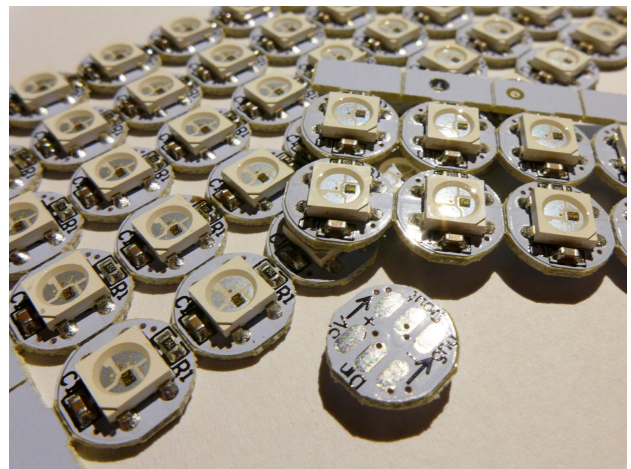


Obrázek 1: RGB pásek s WS2812B

Na trhu je několik typů adresovatelných RGB LED, z nichž nejrozšířenějším je WS2812 a nově se rozšiřující typ APA102C (a jeho klony jako např. SK9822). WS2812 jsou známé pod názvem „NeoPixel“, APA102C nesou obchodní název „DotStar“. DotStar mají rozhraní SPI a mnohem vyšší frekvenci PWM, což je činí vhodným pro tzv. light painting (fotografie s dlouhou expozicí, efekty ve výpletu jízdniho kola atd.) Starší LED pásky mívají řadič WS2811, který je v samostatném SMD pouzdře připájen vedle jednotlivých LED.



Obrázek 2: Detail pásku s WS2812; uvnitř pouzdra je vidět vestavěný čip řadiče



Obrázek 3: Miniaturní RGB moduly pro použití v tzv. wearables

## 2 Zapojení

Diody NeoPixel a DotStar se napájí 5 V, v nouzi lze použít i nižší napětí (většinou fungují bez problému i na 3.3 V). Spotřeba jednotlivých pixelů je v plném jasu až 60 mA, u delších LED pásků se proto doporučuje přivést napájení externím vodičem zhruba každé 1 až 2 metry, jinak dochází k úbytku napětí a tzv. brown-outu, který se projeví zkreslením barev kvůli nedostatečnému napětí pro některé segmenty.

Na přívodní vodiče je vhodné připojit větší elektrolytický kondenzátor, jinak dochází ke kolísání jasu. Při napájení většího množství LED je lepší místo USB použít externí síťový zdroj.

### 3 Komunikační protokol

Adresované LED jsou v páscích či maticích zpravidla zapojeny jako jeden dlouhý řetězec. Řídicí signál vstupuje do prvního řadiče, který si z dat vezme vlastní nastavení a zbytek propaguje do dalšího řadiče.

#### 3.1 NeoPixel

Diody NeoPixel (WS1812) se nastavují jediným vodičem, na kterém jsou nuly a jedničky signalizovány různě dlouhými pulzy s časováním dle obrázků 4 a 5. Konec dat (zobrazení barev a příprava na další rámeček) se signalizuje prodlevou asi 50  $\mu$ s. Přenos probíhá v pořadí [Green, Red, Blue] a bity se posílají v pořadí od nejvyššího (7) po nejnižší (0).

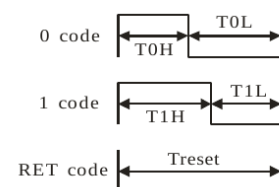
Komunikace s WS2812 vyžaduje přesné časování pulzů, a to především první části (vysoká úroveň). V případě problémů zkontrolujte časování osciloskopem. Jakákoliv fluktuace délky pulzů, např. kvůli načítání stránky do keše nebo obsluze přerušení, může způsobit chybný přenos dat. Mezi jednotlivými bajty je určitá vůle a můžeme si dovolit větší zdržení, ne však více než asi 10  $\mu$ s. To lze využít pro různé výpočty, jako např. úprava jasu následujícího pixelu.

##### RGB IC characteristic parameter

Emitting color	Model	Wavelength(nm)	Luminous intensity(mcd)	Voltage(V)
Red	13CBAUP	620-625	390-420	2.0-2.2
Green	13CGAUP	522-525	660-720	3.0-3.4
Blue	10R1MUX	465-467	180-200	3.0-3.4

##### Data transfer time( TH+TL=1.25 $\mu$ s $\pm$ 600ns)

TOH	0 code ,high voltage time	0.4 $\mu$ s	$\pm$ 150ns
T1H	1 code ,high voltage time	0.8 $\mu$ s	$\pm$ 150ns
T0L	0 code , low voltage time	0.85 $\mu$ s	$\pm$ 150ns
T1L	1 code ,low voltage time	0.45 $\mu$ s	$\pm$ 150ns
RES	low voltage time	Above 50 $\mu$ s	



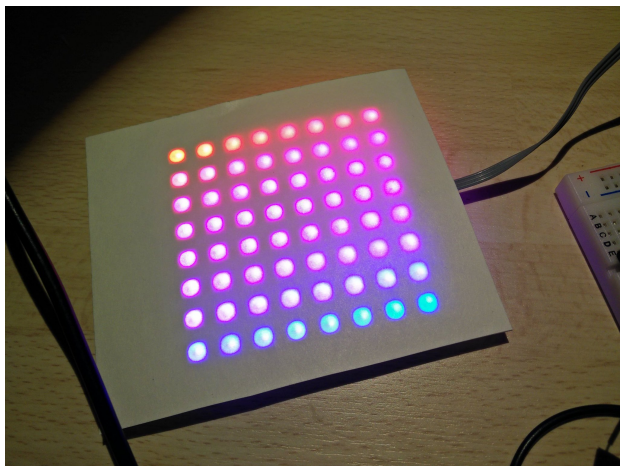
Obrázek 4: Parametry WS2812B dle datasheetu

Obrázek 5: Kódování bitů WS2812

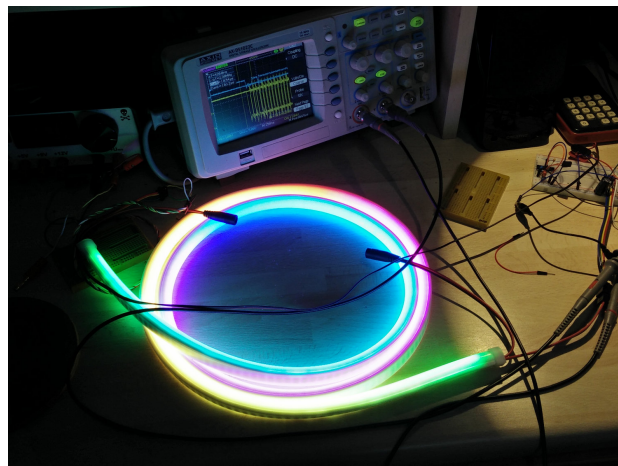
#### 3.2 DotStar

DotStar (APA102C) mají, na rozdíl od NeoPixelů, standardní rozhraní SPI a data do nich můžeme posílat rychlostí až kolem 20 MHz. LED jsou opět spojené do dlouhého řetězce, nyní ovšem navíc s hodinovým signálem.

Komunikace s APA102C je jednoduchá, stačí správně nastavit SPI periférii (8 bitů, CPOL=1, CPHA=1). Nejprve se pošle hlavička: 4 bajty 0x00. Následuje řada 4-bajtových slov [A,R,G,B] kde A je 0b111aaaaa (tři jedničky a 5 bitů jasu). V praxi můžeme místo prvního bajtu vždy posílat 0xFF a jas nastavovat pomocí intenzity jednotlivých složek. Po odeslání všech barev je nutné ještě poslat několik bajtů se samými jedničkami (0xFF), čímž indikujeme konec přenosu. Datasheet uvádí, že stačí 4 koncové bajty, v praxi jich bývá potřeba více—nejlépe ověřit experimentálně.



Obrázek 6: Efekt duhy na matici 8x8 NeoPixelů



Obrázek 7: „Neonová trubice“ s pixely DotStar

## 4 mbed knihovna

Pro použití v mbedu jsme připravili knihovnu „NeoPixel“ a sadu utilit pro práci s barvami (colorspace.cpp, .h). Ukázka použití knihovny je na začátku hlavičkového souboru.

Při ožiování knihovny je kritické dodržet správné časování řídicích impulzů, nejlépe pomocí osciloskopu. Kromě správné délky čekacích smyček je také nezbytné vypnout cache programu, což se v knihovně dělá voláním `__HAL_FLASH_INSTRUCTION_CACHE_DISABLE()` (pozn.: platí jen u některých procesorů). Bez zakázání cache se krátké pulzy mohou prodloužit do takové míry, že to NeoPixely interpretují jako jedničku.

Při použití knihovny na procesoru s pomalejšími hodinami může být nezbytné přepsat část knihovny do assembleru. Příklad jak to provést je uveden na následující stránce; pro úsporu místa lze bloky NOPů přepsat do smyček. Počtem NOPů se nastavují délky jednotlivých částí kódování bitů.

## Ukázka implementace poslání 1 bajtů do WS2812 v inline assembleru pro ARM CC

```
void NeoPixelOut::byte(register uint32_t byte)
{
    register uint32_t r0, r1, rset, rclr, mask;
    rset = (uint32_t)gpio.reg_set;
    rclr = (uint32_t)gpio.reg_clr;
    mask = (uint32_t)gpio.mask;

    __asm {
        MOV r1, #8
loop:
        MOV r0, byte
        STR mask, [rset]
        ANDS r0, #0x80
        BNE send1
        NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
        STR mask, [rclr]
        NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
        NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
        NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
        B    next1
send1:
        NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
        NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
        NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
        STR mask, [rclr]
        NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;NOP;
next1:
        LSL byte,byte,#1
        SUBS r1, #1
        BNE loop
    }
}
```