

# Strojna in programska oprema jelke

predstavitev

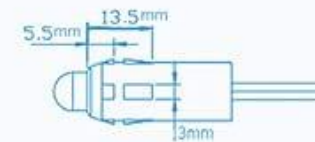
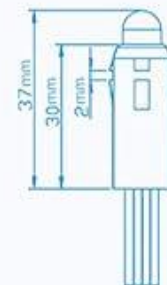
# Strojna oprema --- LED

- Tri verige sto LED verig z IC WS2811

## WS2811 Individuell adressierbares LED-Pixel

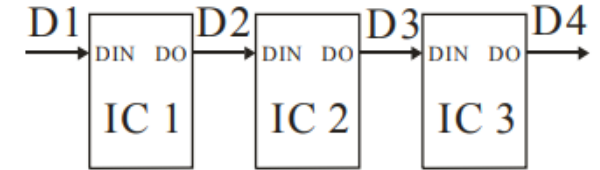
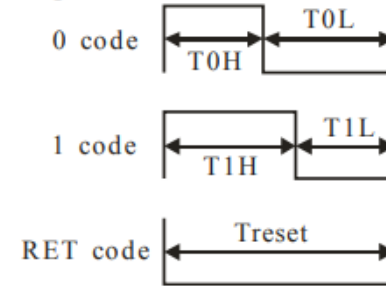
160 Fuß / 500 LEDs / 12 V / IP68 Wasserdicht

Jede LED kann einzeln angesteuert werden

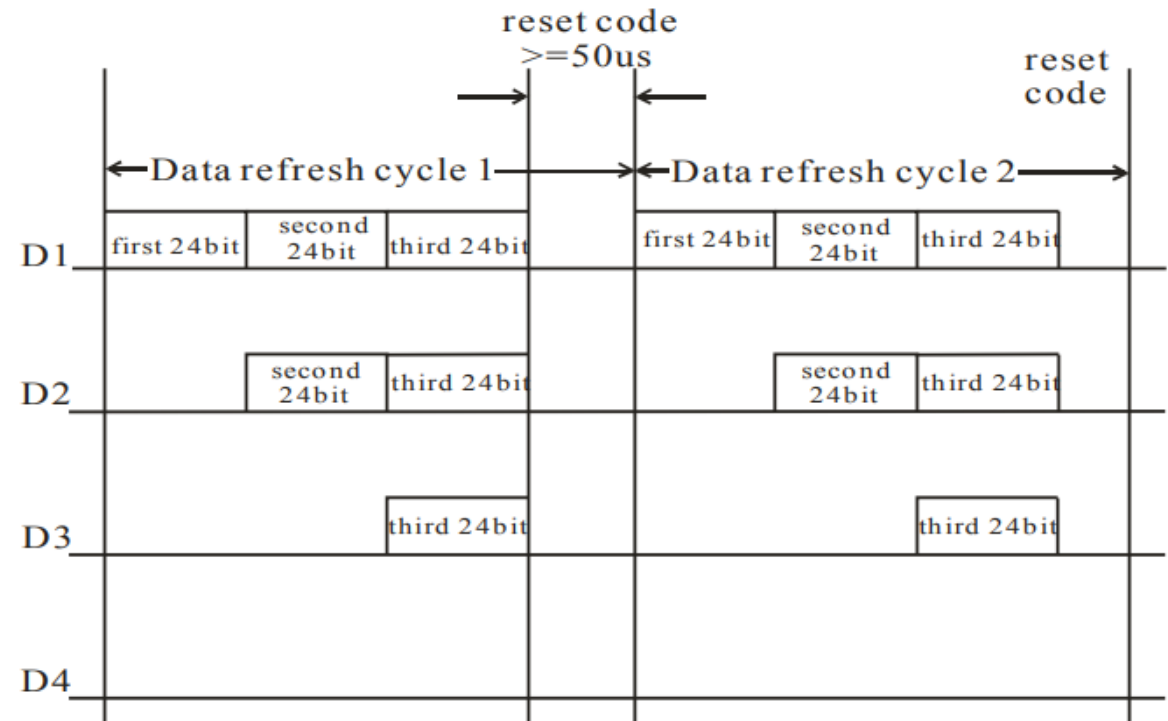


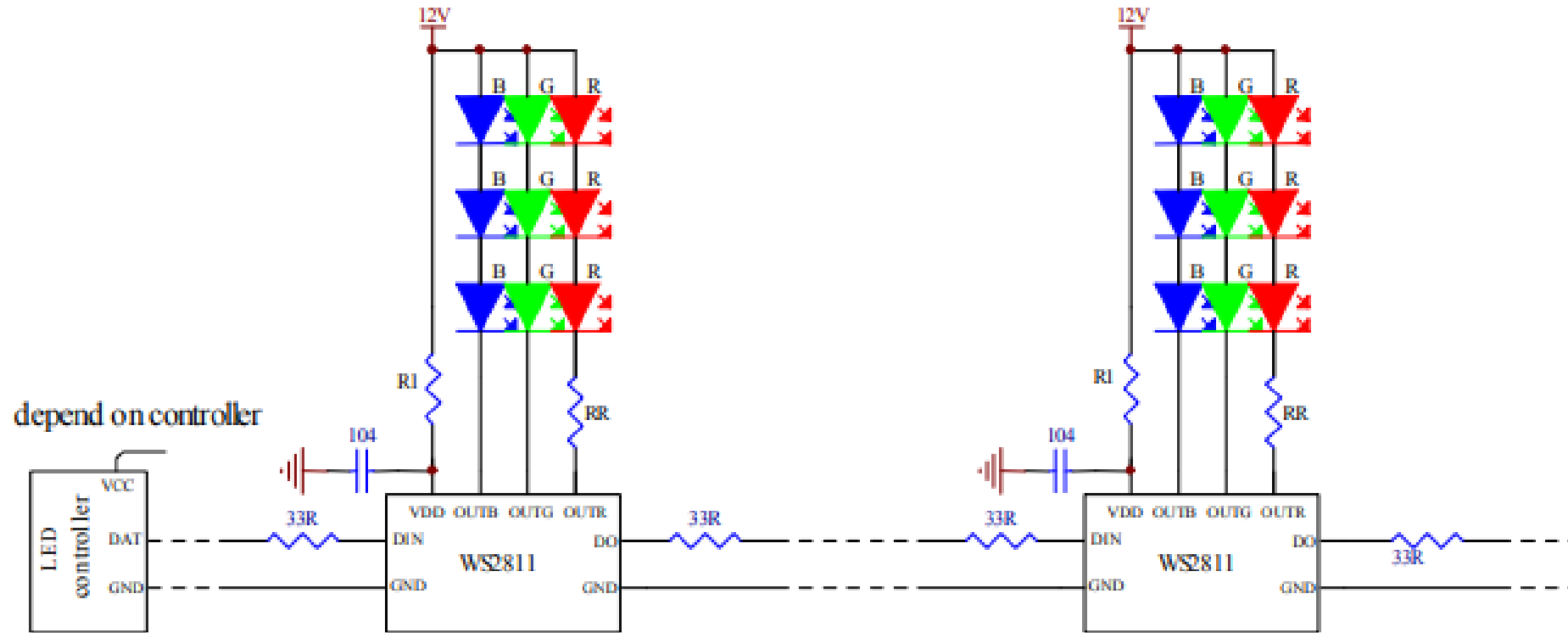
- Daisy-chaining

**Sequence chart:**



**Cascade method:**





**Fig 2**

The same as the front mode, it is also use constant current output. In this circuit, R1 is used as the IC internal LDO divider resistance and the value is 2.7K. The capacitance 104 as bypass capacitor. To prevent the reflection and hot-swap protection, we suggest to connect a 33ohm resistor at the data input or output port for impedance. At the OUTF port we should add a divider resistance RR. The value of RR can be derived by the following equation:

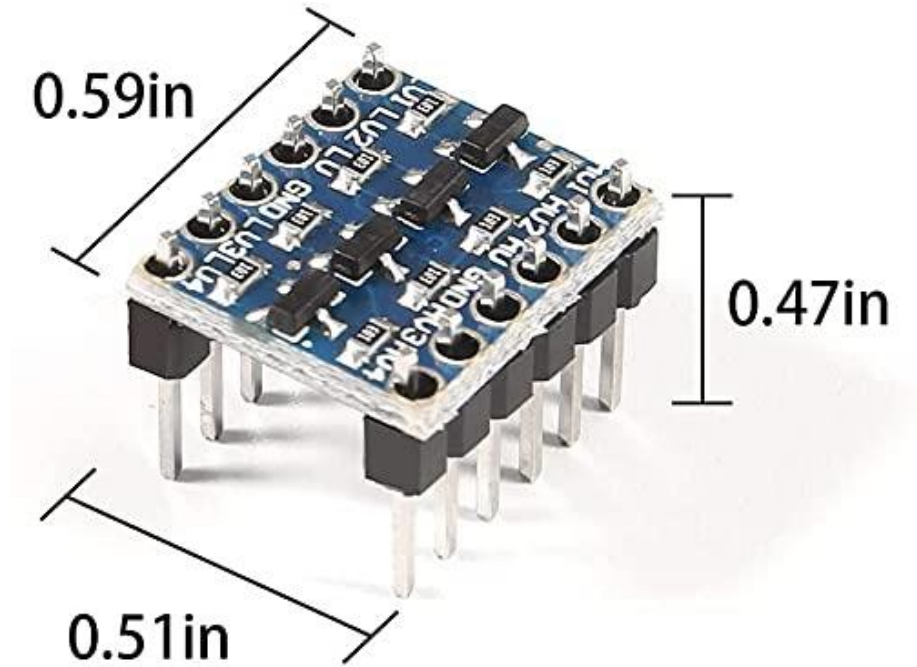
$$RR = \frac{12 - 3V_{LED R}}{18.5} \text{ K}\Omega$$



# Upravljalnik



- Raspberry Pi 4
- GPIO PWM napetost 3,3V
- DATA napetost na WS2811 je 5V
- Level shifter
- PWM0 na GPIO 18 DMA







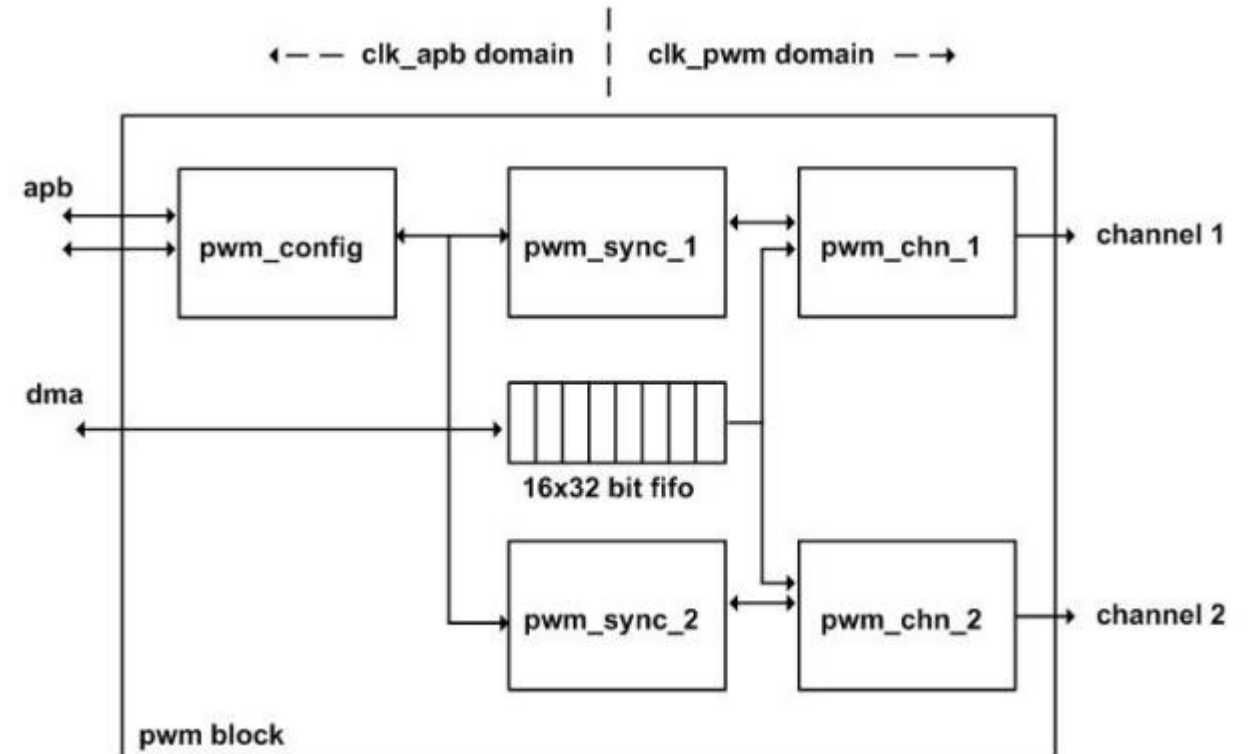
# BCM2836 (RPI SoC)

- DMA PWM kontroler
- Namesto big-banganja bitov na GPIO, kar bi bilo prepočasno in časovno nezanesljivo (linux ni real-time kernel)
- Omejitev max ~2700 WS2811 Icjev na PWM kanal

The **PWM** controller incorporates the following features:

- Two independent output bit-streams, clocked at a fixed frequency.
- Bit-streams configured individually to output either **PWM** or a serialised version of a 32-bit word.
- **PWM** outputs have variable input and output resolutions.
- Serialise mode configured to load data to and/or read data from a FIFO storage block, which can store up to eight 32-bit words.
- Both modes clocked by clk\_ **pwm** which is nominally 100MHz, but can be varied by the clock manager.

## 9.2 Block Diagram





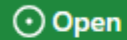
# Težave s strojno opremo

- Po podaljšanju podatkovnega kabla na okoli 1 m smo začeli dobivati motnje na barvah luči --- vsake toliko osvežitev so se vsi biti zamaknili za par bitov nazaj.
- Težava je povzročilo neskladje impedanc oddajnika (GPIO) in sprejemnika (prvega WS2811 IC)
- Težavo smo rešili s 33 Ohmskim upornikom tik po GPIO pinu 18.

# Programska oprema

- [GitHub - programerski-klub-fmf/jelka](#)
- [rpi-ws281x · PyPI](#) uporablja [GitHub - jgarff/rpi\\_ws281x: Userspace Raspberry Pi PWM library for WS281X LEDs](#)
- Koncept vzorcev, python skript, ki na LED luči rišejo neko animacijo.
- Ko se vzorec požene, sporoča, kaj želi risati, z uporabo lepega OOP APIja v pythonu, ki nato podatke napiše v deljen spomin, ki ga nato bere drug program, ki komunicira s knjižnico rpi\_ws281x

# RPi0W: SD card errors while running around 40 LEDs #224



Open

pinheadmz opened this issue on Sep 19, 2017 · 18 comments



pinheadmz commented on Sep 19, 2017



Running the test `sudo ./test -c` produces SD card errors in `dmesg`:

```
[33490.769029] mmcblk0: error -110 transferring data, sector 137216, nr 16, cmd response 0x900, card status 0xc00
```

More verbose log here: <https://pastebin.com/AZUijQmW>

This error will pop up every few minutes if I leave the test running just by itself. With my other processes running (full project: <https://github.com/pinheadmz/ClockJr>) these errors pop up as frequently as every few seconds.

My project displays the "rainbow wheel" effect for a few seconds then stops and goes blank. A few seconds later I will see this mmcblk0 error in dmesg.

I've been through FOUR SD CARDS, all different brands. It's not the card.

# Programska oprema --- struktura procesov

- `daemon.py`: s knjižnico `flask` ustvari HTTP frontend, na katerem lahko spremljamo aktiven vzorec in vzorec menjamo. Poleg tega poganja `wrapper.py`, ki komunicira z `rpi_ws281x` in vzorcem. Teče kot `root`.
- `wrapper.py`: požene vzorec, skrbi za to, da se knjižnica `rpi_ws281x` vedno lepo deinicializira, bere, kaj želi vzorec napisati na lučke in to pošilja `rpi_ws281x`. Ko dobi `SIGINT`, ubije vzorec in deinicializira `rpi_ws281x`. Teče kot `root`, da ima dostop do DMA.
- `vzorec.py`: teče v nepriviligiranem okolju (`chroot`) kot nepriviligiran uporabnik, na `/dev/shm` piše `RGBRGRGB...` vrednosti `luck` in na FD 7 sporoči, ko želi, da naj se stanje na lučkah osveži.



# Povezava v medmrežje

- Eduroam WiFi -> Wireshark -> nginx HTTP reverse proxy
- Težava: FMF ima na Eduroamu le /24

# Sprejemanje vzorcev študentov

- Študente smo prosili, da napišejo svoje vzorce in izdelajo PR na GitHub. Vsako minuto je Upravljalnik prenesel spremembe v repozitoriju in začel izvajati novonastale vzorce.

Q is:pr is:closed

✕ Clear current search query, filters, and sorts

🔗 0 Open ✓ 1 Closed

🔗 **Firework pattern** ✕  
#1 by Lavennie was merged on Dec 21, 2023

Name
..
adrian.py
barbers.py
debug_red.py
debug_scanx.py
fireworks.py
gradient.py
lines.py
one_color_change.py
pulse.py
rain.py
random_light.py
rotating_full.py
rotating_plane.py
scan_x.py
scan_y.py
scan_z.py
sphere.py
spiral.py

# Programska oprema --- pot podatkov do lučk

- vzorec.py kliče metode razreda Jelka knjižnice jelka.py. Slednja glede na stanje okoljske spremenljivke bodisi odpre simulacijo v pygame, bodisi uporabi knjižnico jelka\_hardware.py, ki komunicira z vmesnikom wrapper.py (/dev/shm za podatke, FD 7 za ukaz za osvežitev lučk).
- wrapper.py uporabi knjižnico ws\_281x, da podatke iz /dev/shm ob prispelem bajtu na FD 7 izriše na lučke.
- Knjižnica ws\_281x bajte po DMA pošlje na PWM kontroler, ki jih pošlje na daisychainane LED luči.

# Težava s programsko opremo

- Nekje v tem ciklu sta se zamenjala bajta za rdečo in zeleno vrednost luči, kar smo popravili z grdima hackom  $(r, g, b) \Rightarrow (g, r, b)$  *nekje v kodi*.



Hvala za pozornost



