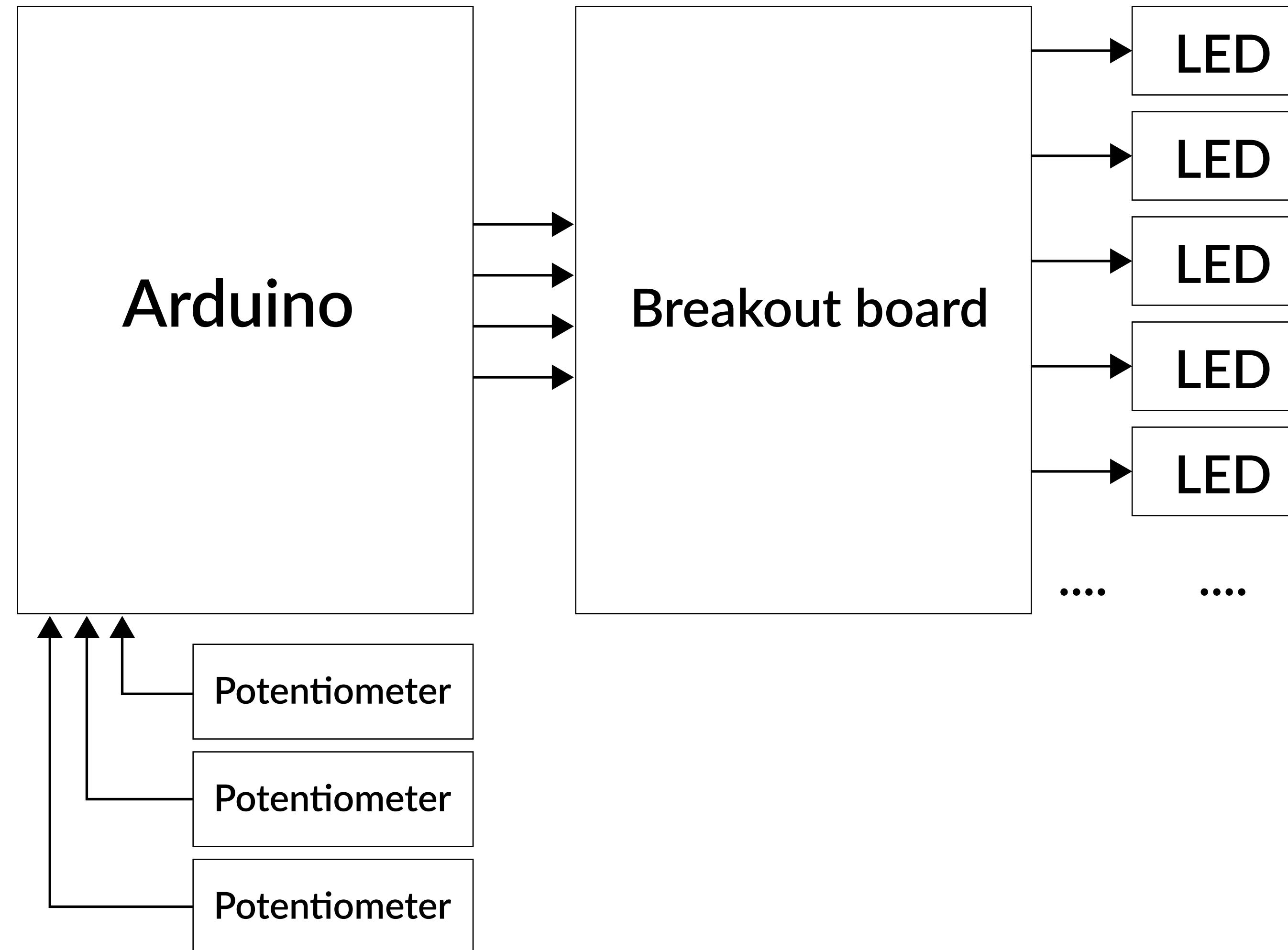


# **PWM LED Multiplexing**

Digital Media, HFK

Stuidio workshop, Lee, Sangbong

## Goal of the day



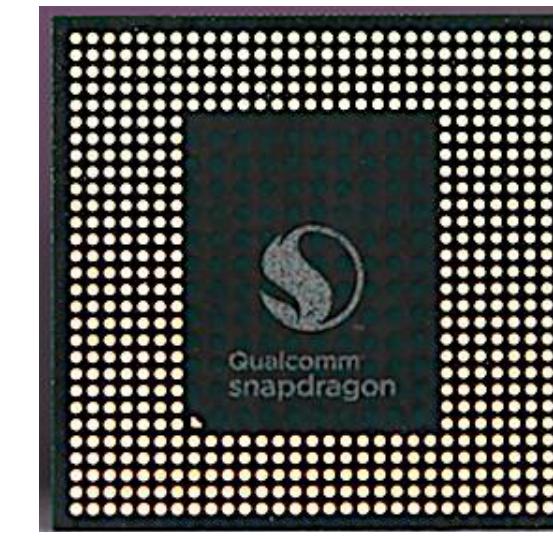
## Microprocessor



Intel i9



Apple M1



Snapdragon 845

Computation →

Expensive  
Fast  
High tech  
High power  
...

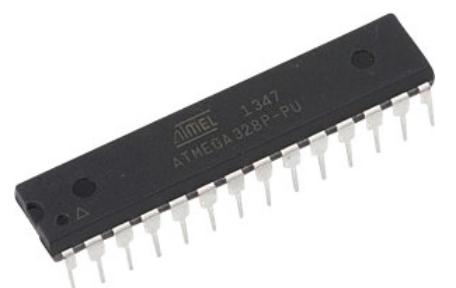
## Microcontroller



Attiny



STM32 F103



Atmega328

Hardware control →

Cheap  
Slow  
Reliability  
Low power  
...

## Differences amongst microcontroller models

### In Arduino family



Enhanced features

Shields

## ARM chipset boards



Teensy 4.0, USB,  
ohne Header -...  
**€22.70**  
reichelt.de  
+€5.95 shipping



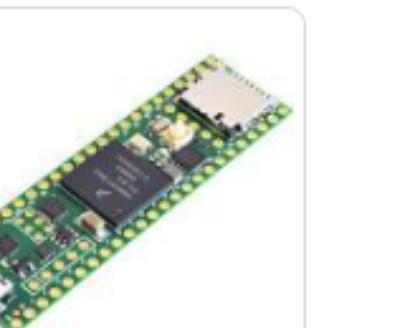
Teensy 4.1, USB,  
ohne Header -...  
**€29.70**  
reichelt.de  
+€5.95 shipping



ADAFRUIT  
INDUSTRIES...  
**€24.24**  
RS Components D...  
+€8.27 shipping



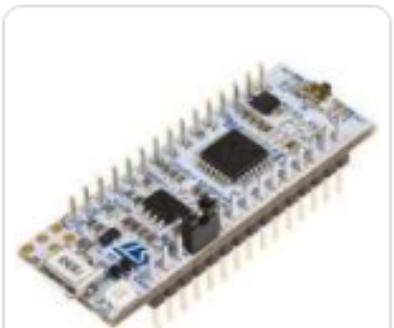
Teensy 4.0 with  
pins  
**€23.09**  
Anratek Deutschla...  
+€3.95 shipping



Teensy 4.1  
**€31.95**  
Anratek Deutschla...  
+€3.95 shipping



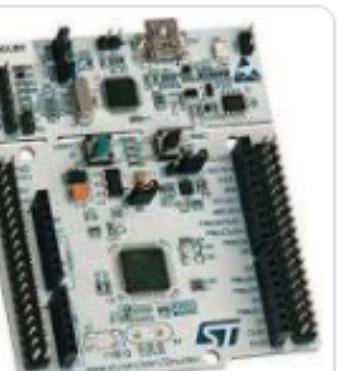
STMicroelectronics  
STM32 Nucleo-32...  
**€12.53**  
RS Components D...  
+€8.27 shipping



STMicroelectronics  
STM32 Nucleo-32...  
**€12.55**  
RS Components D...  
+€8.27 shipping



STMicroelectronics  
STM32 Nucleo-64...  
**€15.79**  
RS Components D...  
+€8.27 shipping



Nucleo-64, ARM  
Cortex M4, STM3...  
**€24.45**  
reichelt.de  
+€5.95 shipping



STMicroelectronics  
STM32 Nucleo-14...  
**€23.12**  
RS Components D...  
+€8.27 shipping

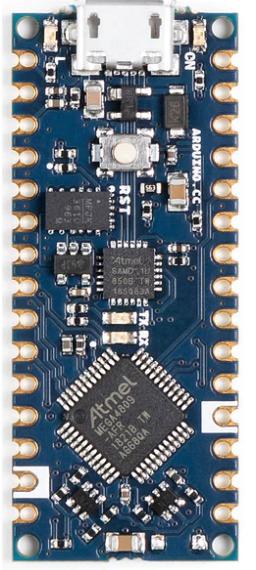


Nucleo-32, ARM  
Cortex M4, STM3...  
**€20.80**  
reichelt.de  
+€5.95 shipping

Teensy series

STM32 Nucleo series

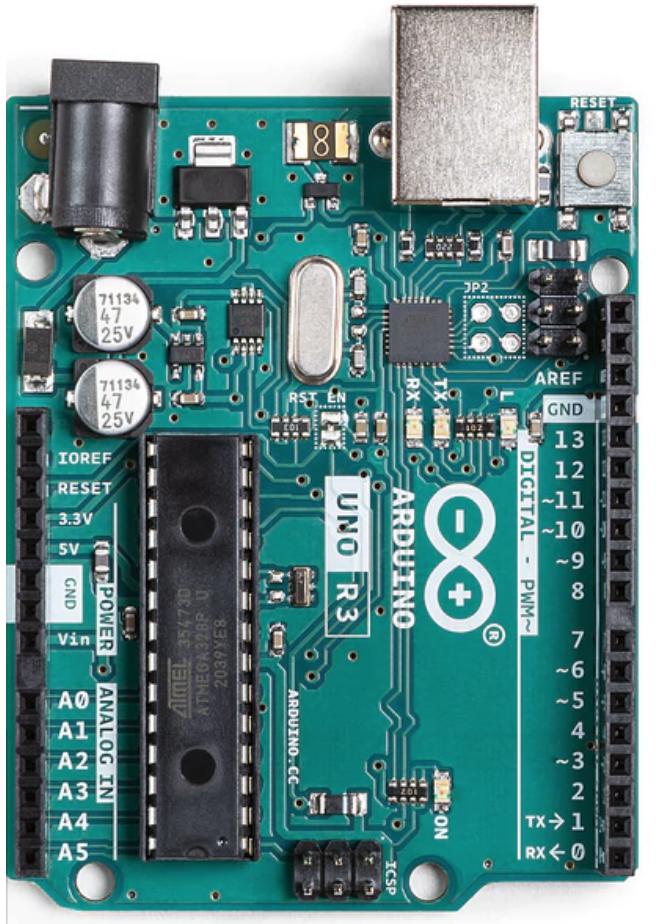
## Size



Arduino Nano

L 45mm X W 18mm

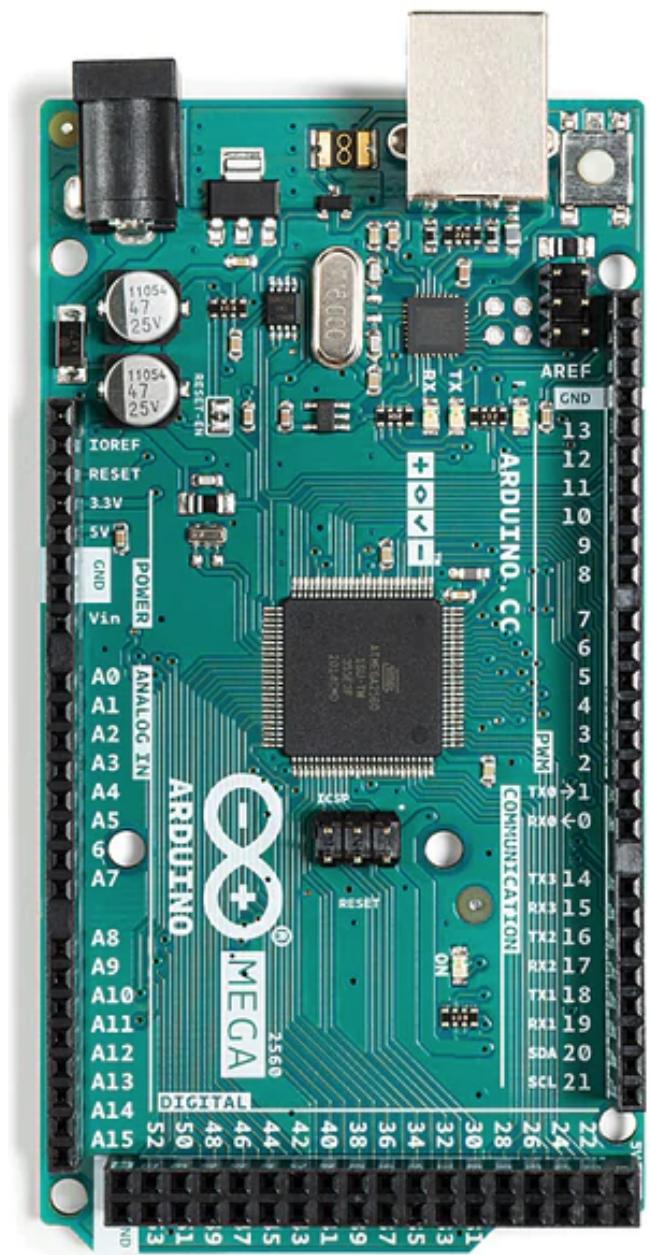
## Compatibility



Arduino UNO

Standard board

## Scalability



Arduino MEGA

54 digital pins,  
16 analog pins,  
4 UART (Serial),  
...

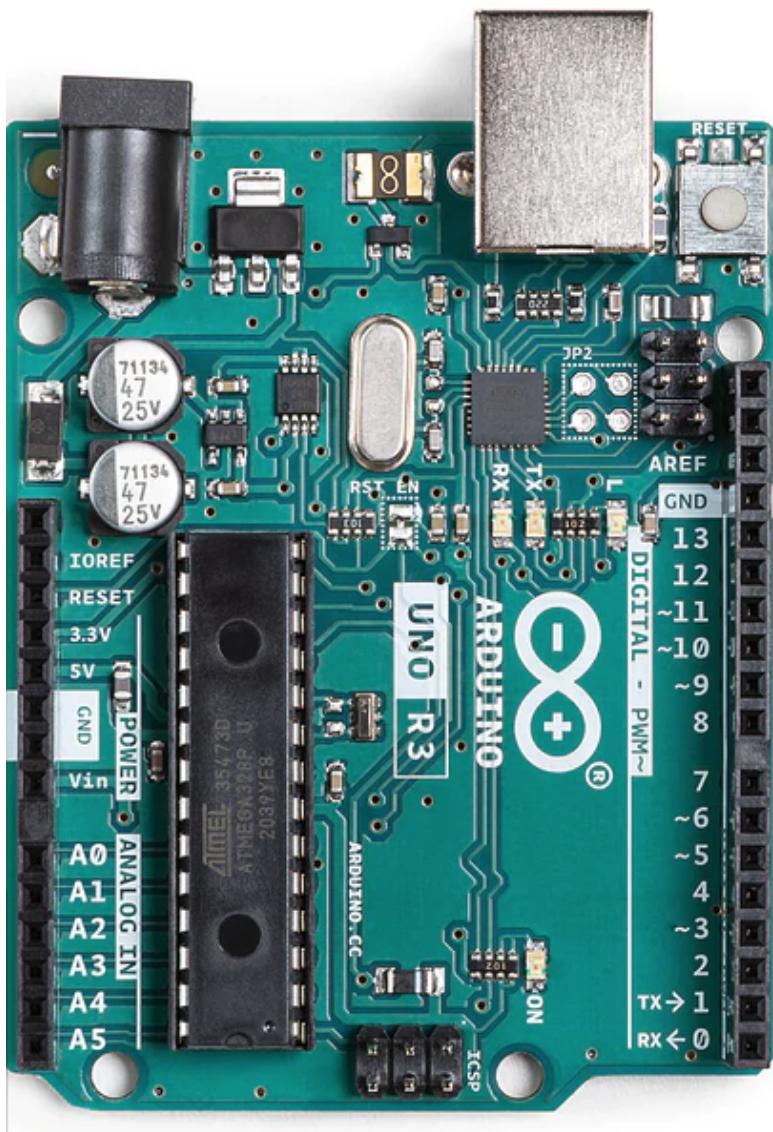
## Pro



Portenta H7

480 Mhz / 240 Mhz  
WiFi, BLE  
Flash memory,  
Computer Vision,  
...

# Introduction of Arduino Uno



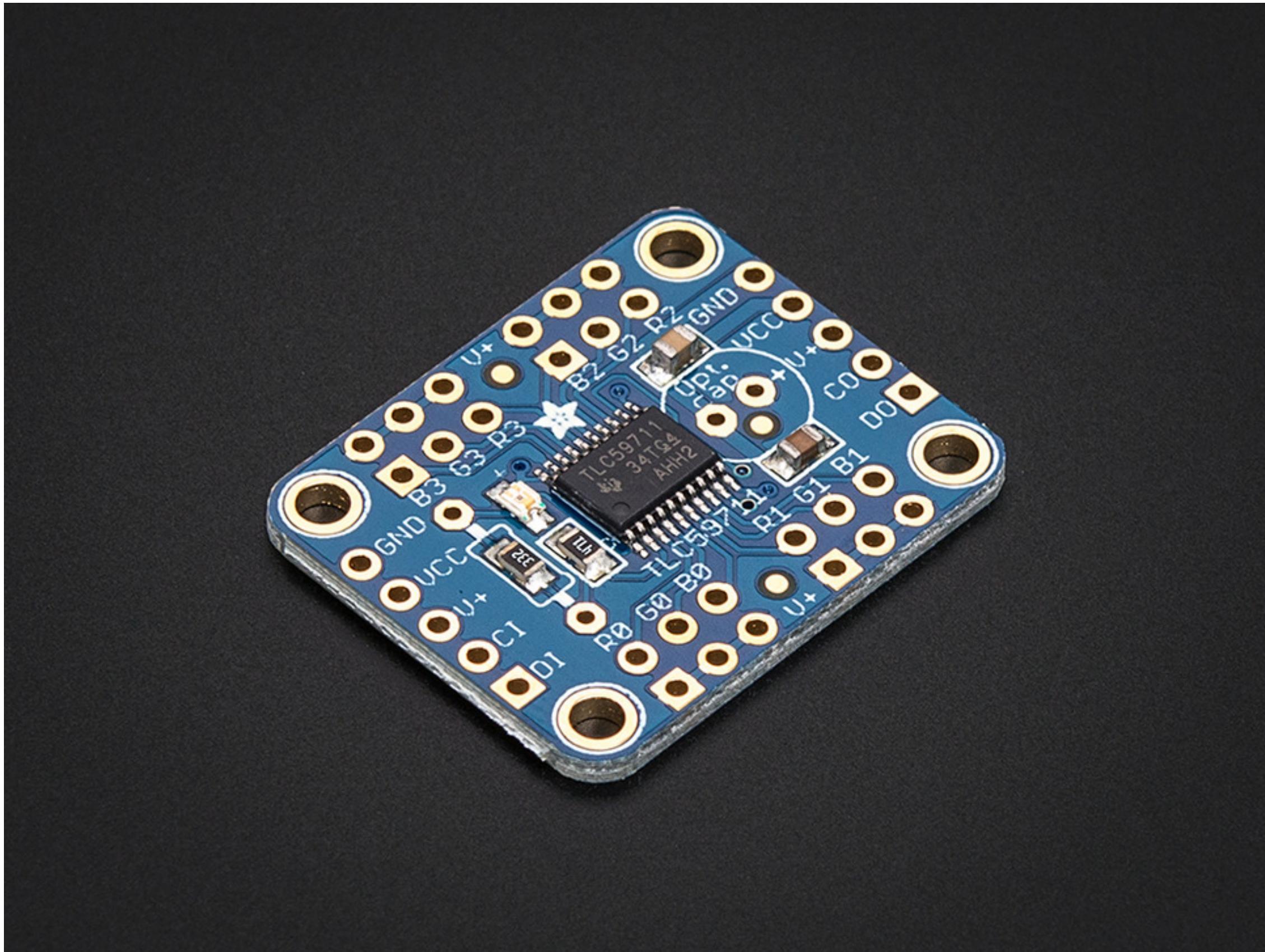
|                        |                          |                |
|------------------------|--------------------------|----------------|
| <b>Board</b>           | <b>Name</b>              | Arduino UNO R3 |
|                        | <b>SKU</b>               | A000066        |
| <b>Microcontroller</b> | ATmega328P               |                |
| <b>USB connector</b>   | USB-B                    |                |
| <b>Pins</b>            | <b>Built-in LED Pin</b>  | 13             |
|                        | <b>Digital I/O Pins</b>  | 14             |
|                        | <b>Analog input pins</b> | 6              |
|                        | <b>PWM pins</b>          | 6              |
| <b>Communication</b>   | <b>UART</b>              | Yes            |
|                        | <b>I2C</b>               | Yes            |
|                        | <b>SPI</b>               | Yes            |

|                    |                                |                                  |
|--------------------|--------------------------------|----------------------------------|
| <b>Power</b>       | <b>I/O Voltage</b>             | 5V                               |
|                    | <b>Input voltage (nominal)</b> | 7-12V                            |
|                    | <b>DC Current per I/O Pin</b>  | 20 mA                            |
|                    | <b>Power Supply Connector</b>  | Barrel Plug                      |
| <b>Clock speed</b> | <b>Main Processor</b>          | ATmega328P 16 MHz                |
|                    | <b>USB-Serial Processor</b>    | ATmega16U2 16 MHz                |
| <b>Memory</b>      | <b>ATmega328P</b>              | 2KB SRAM, 32KB FLASH, 1KB EEPROM |
| <b>Dimensions</b>  | <b>Weight</b>                  | 25 g                             |
|                    | <b>Width</b>                   | 53.4 mm                          |
|                    | <b>Length</b>                  | 68.6 mm                          |

Atmel

[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)

# TLC59711



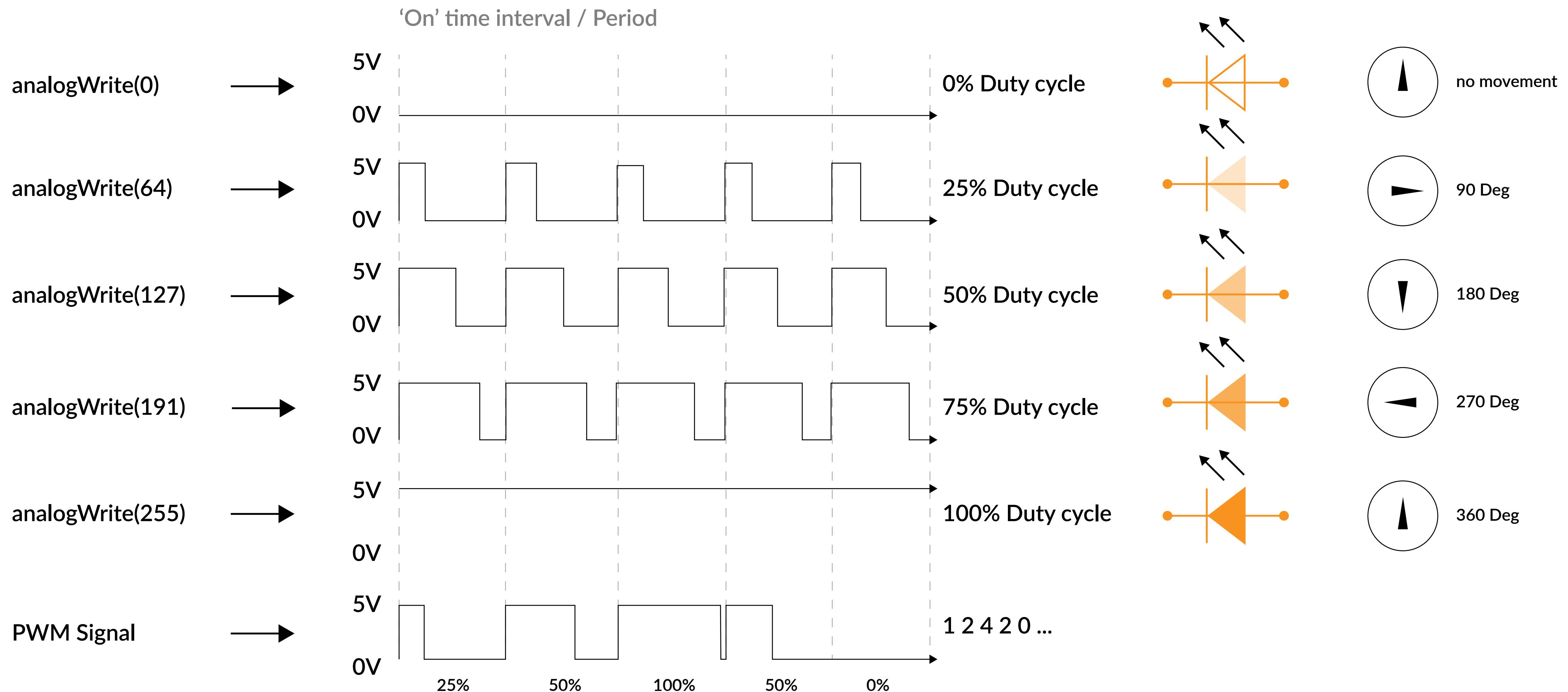
## Description

The TLC59711 is a 12-channel, constant-current sink driver. Each output channel has individually adjustable currents with 65536 PWM grayscale (GS) steps. Also, each color group can be controlled by 128 constant-current sink steps with the global brightness control (BC) function.

GS control and BC are accessible via a two-wire signal interface. The maximum current value for each channel is set by a single external resistor. All constant-current outputs are turned off when the IC is in an over-temperature condition.

Current sink? PWM? Grayscale?

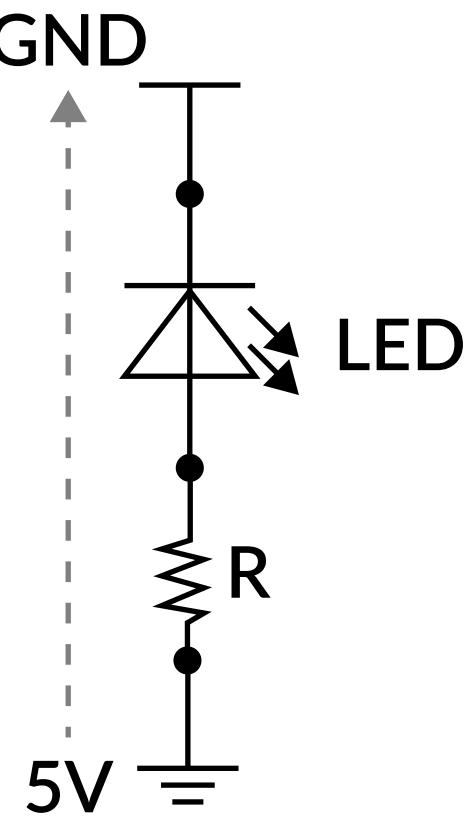
# Pulse Width Modulation



# Current source & sink

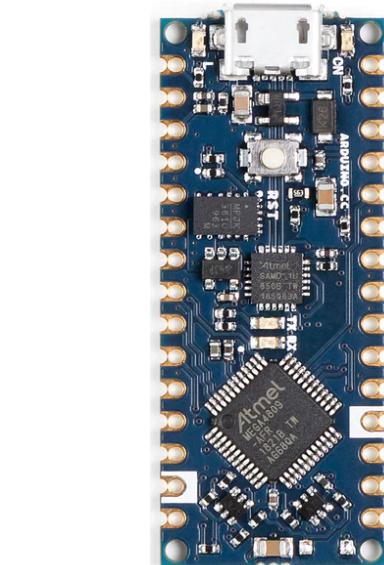
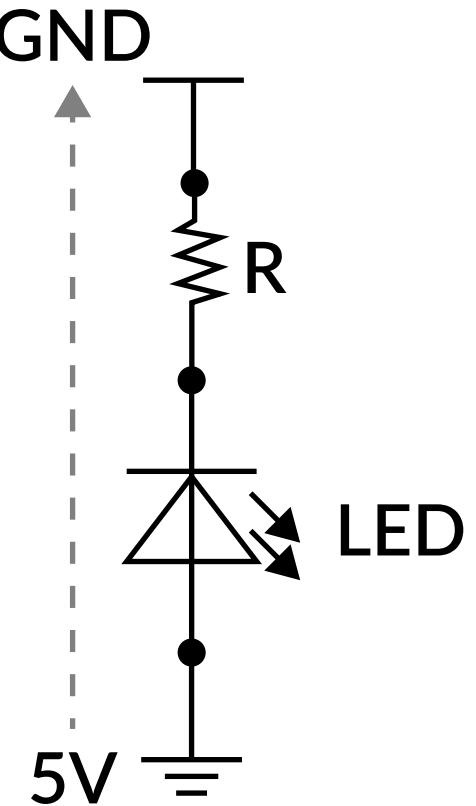
Current source

0V  
(Idle)

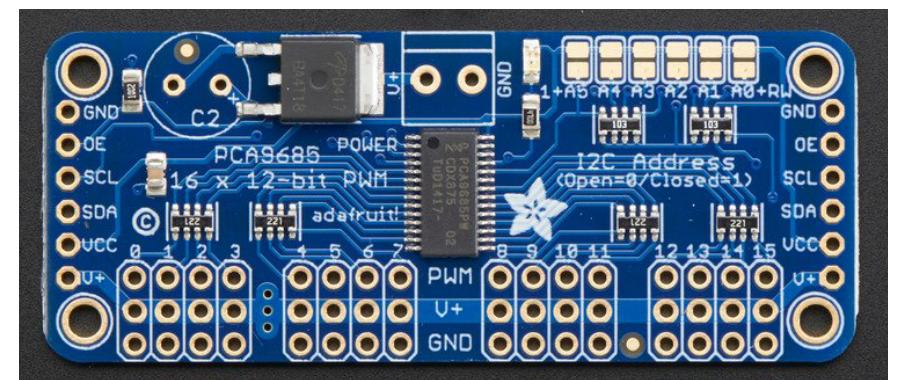


Current sink

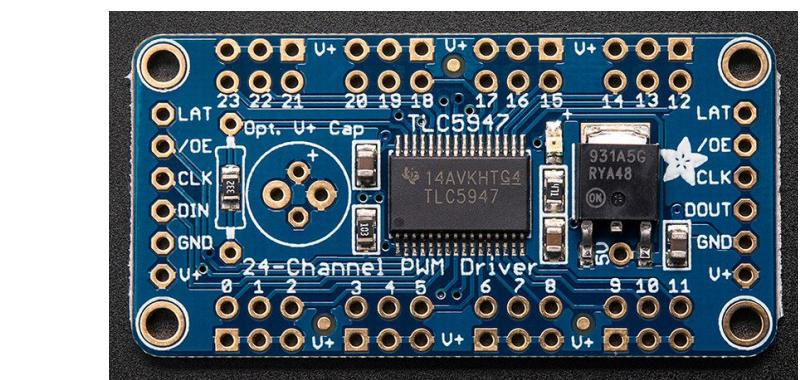
+5V  
(Idle)



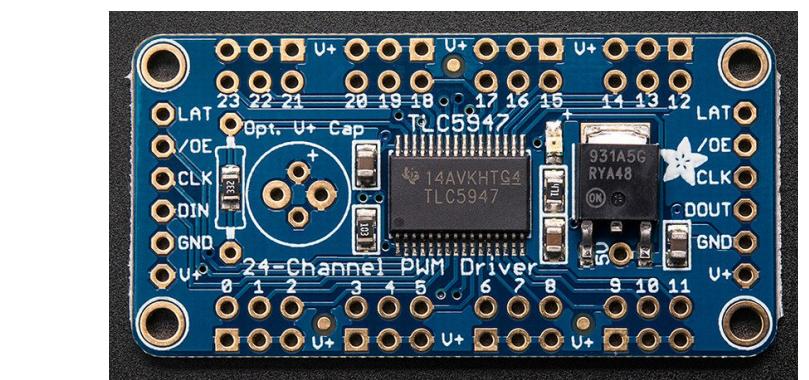
Arduino PWM



PCA9685 PWM

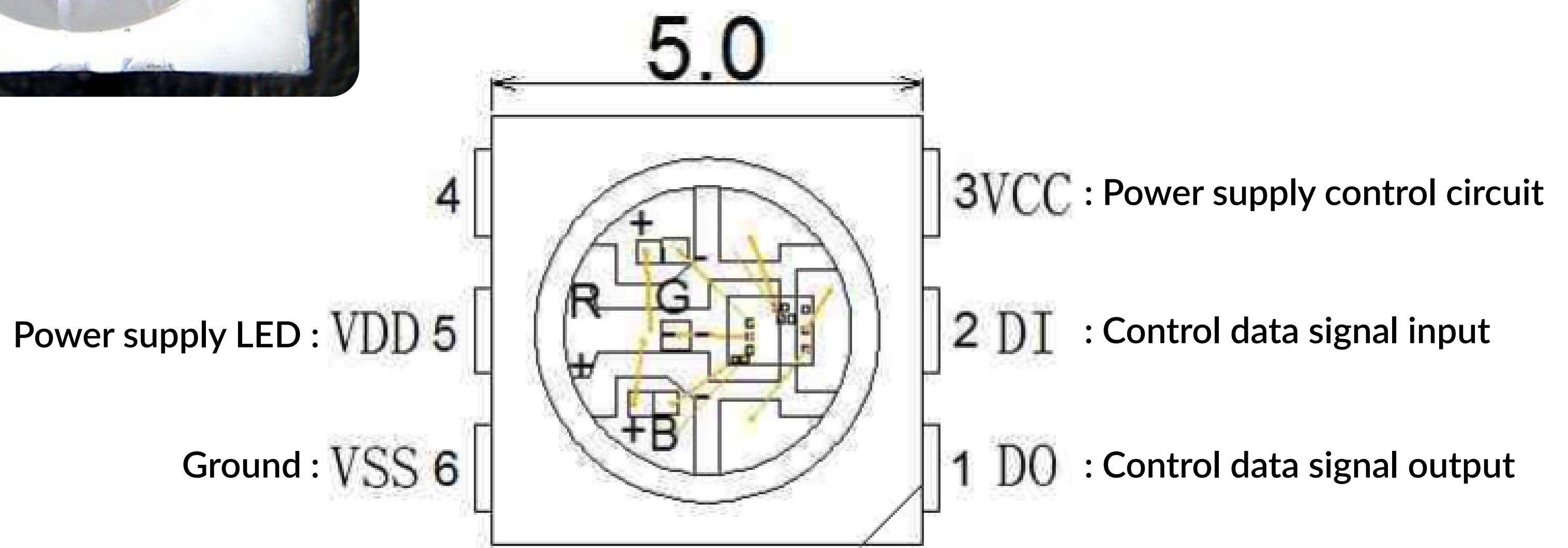
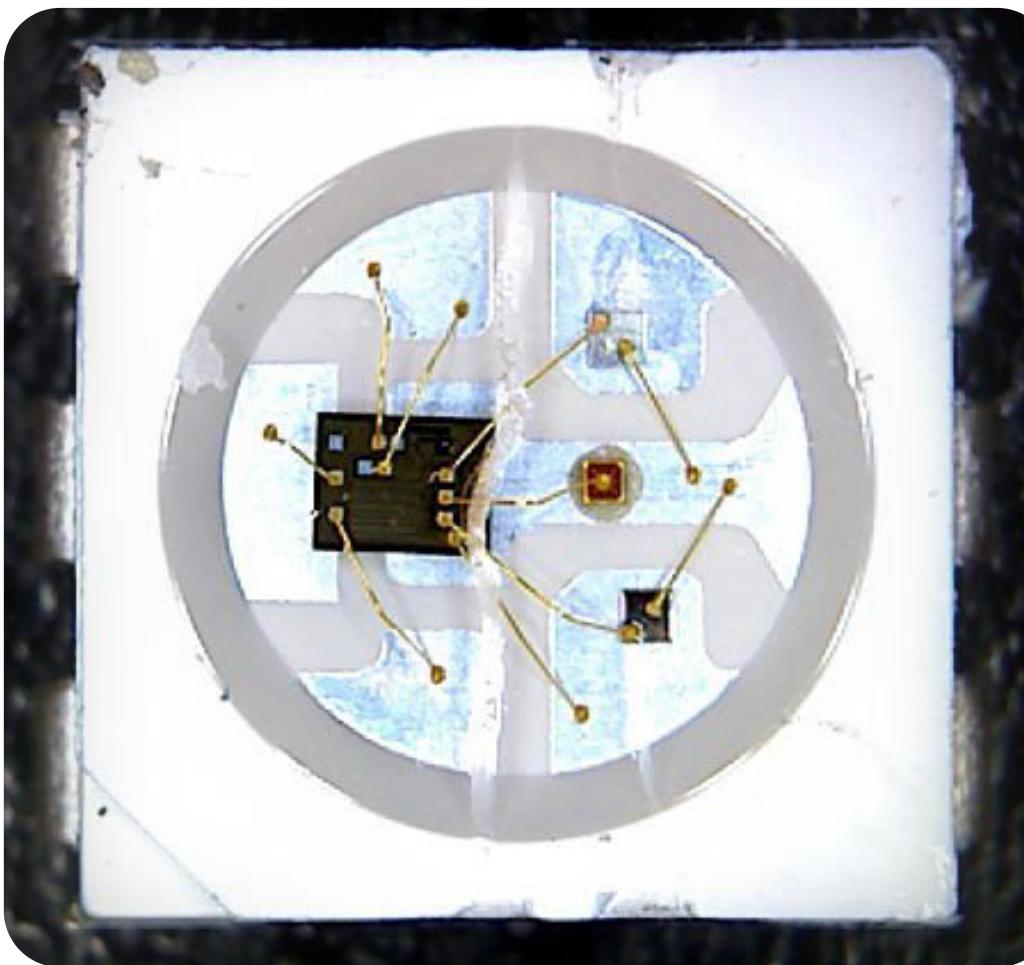
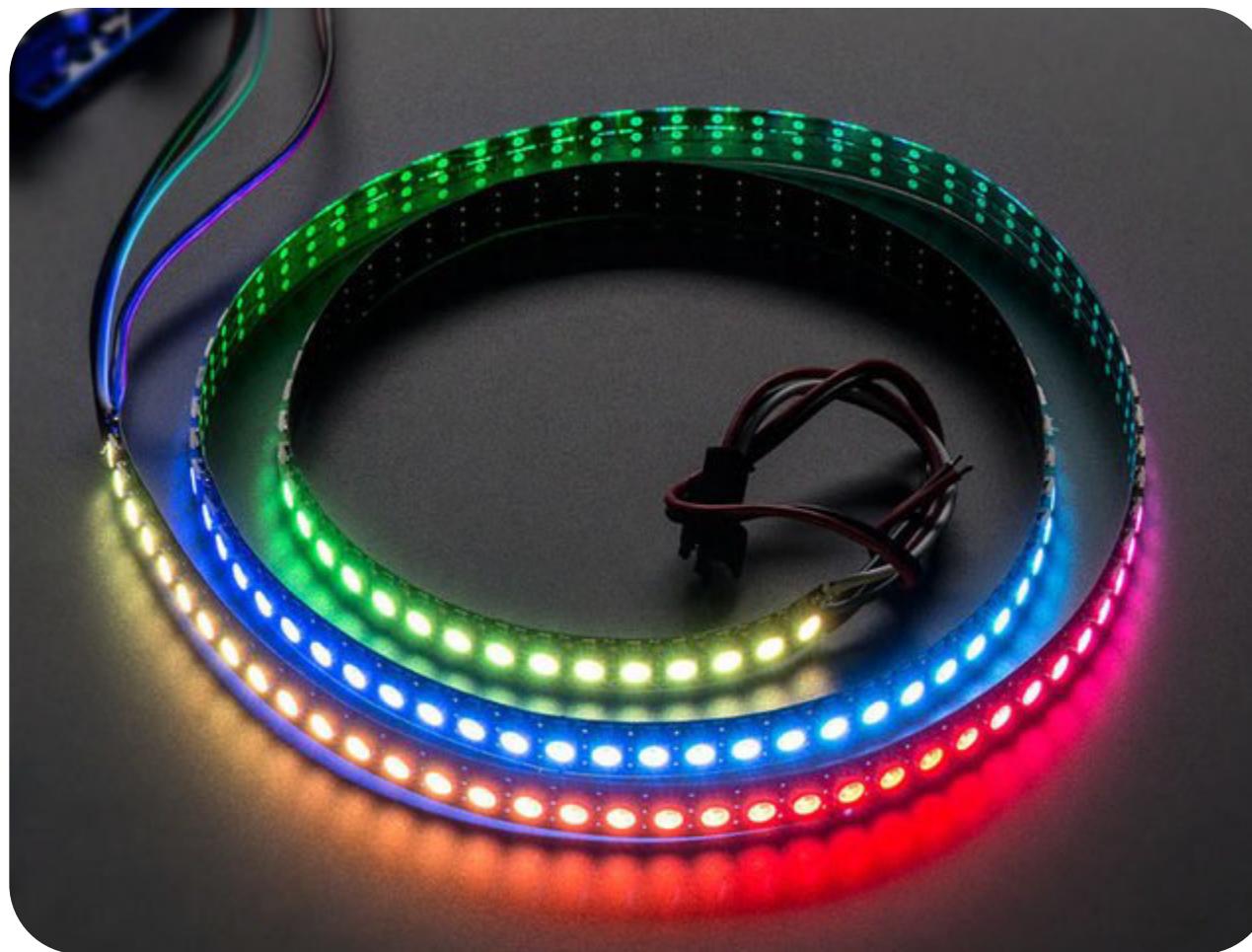


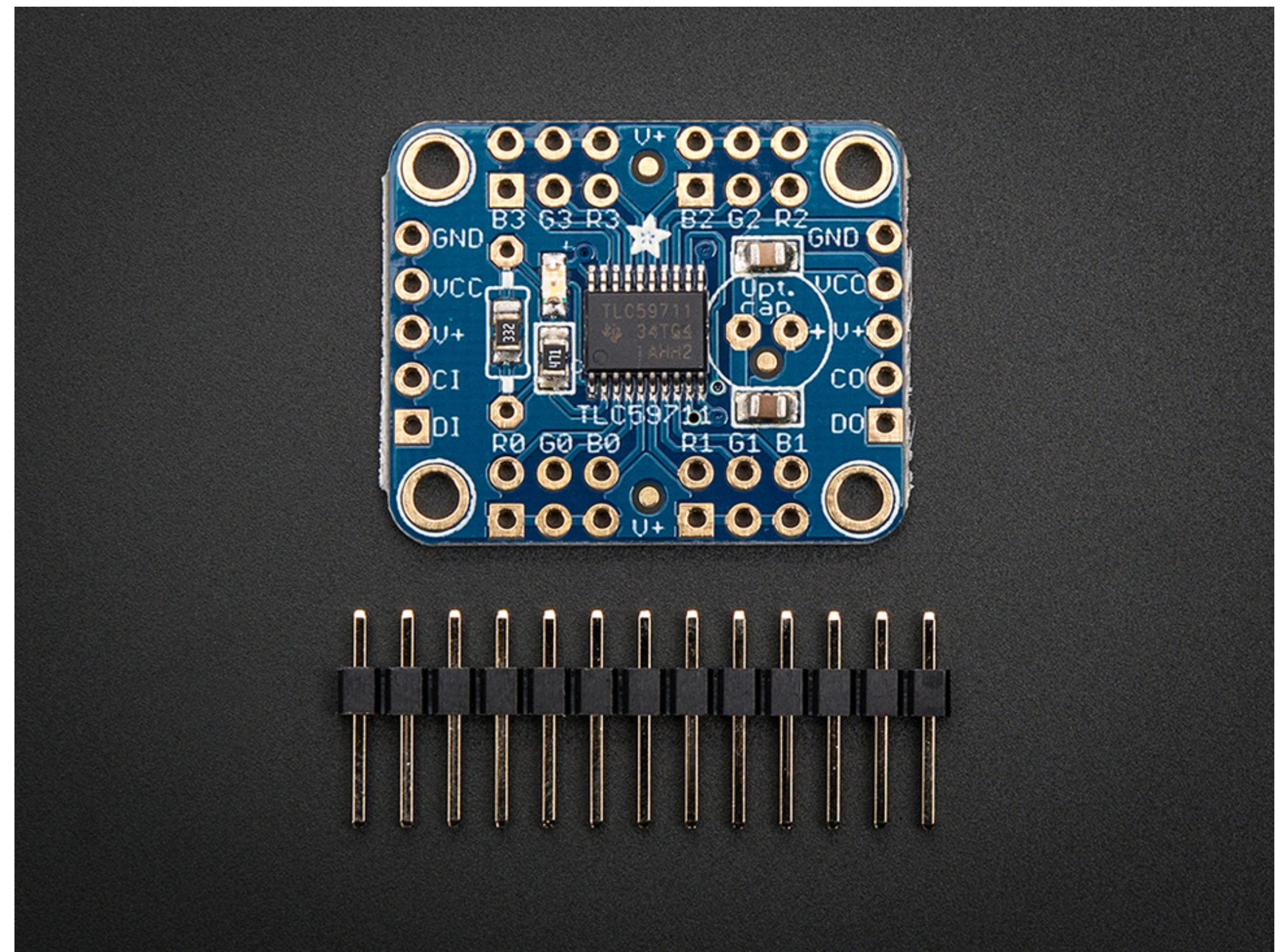
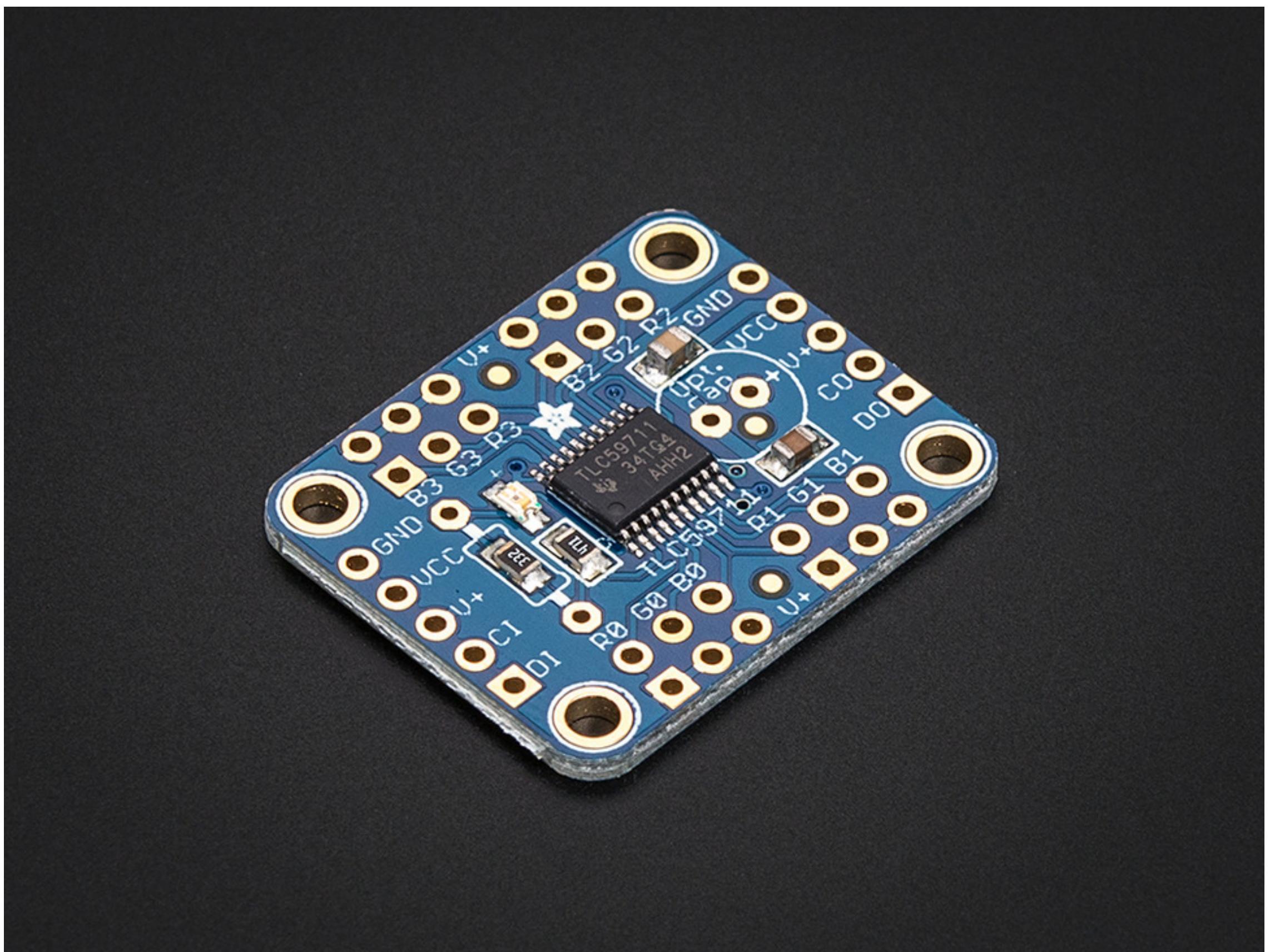
TLC 59711

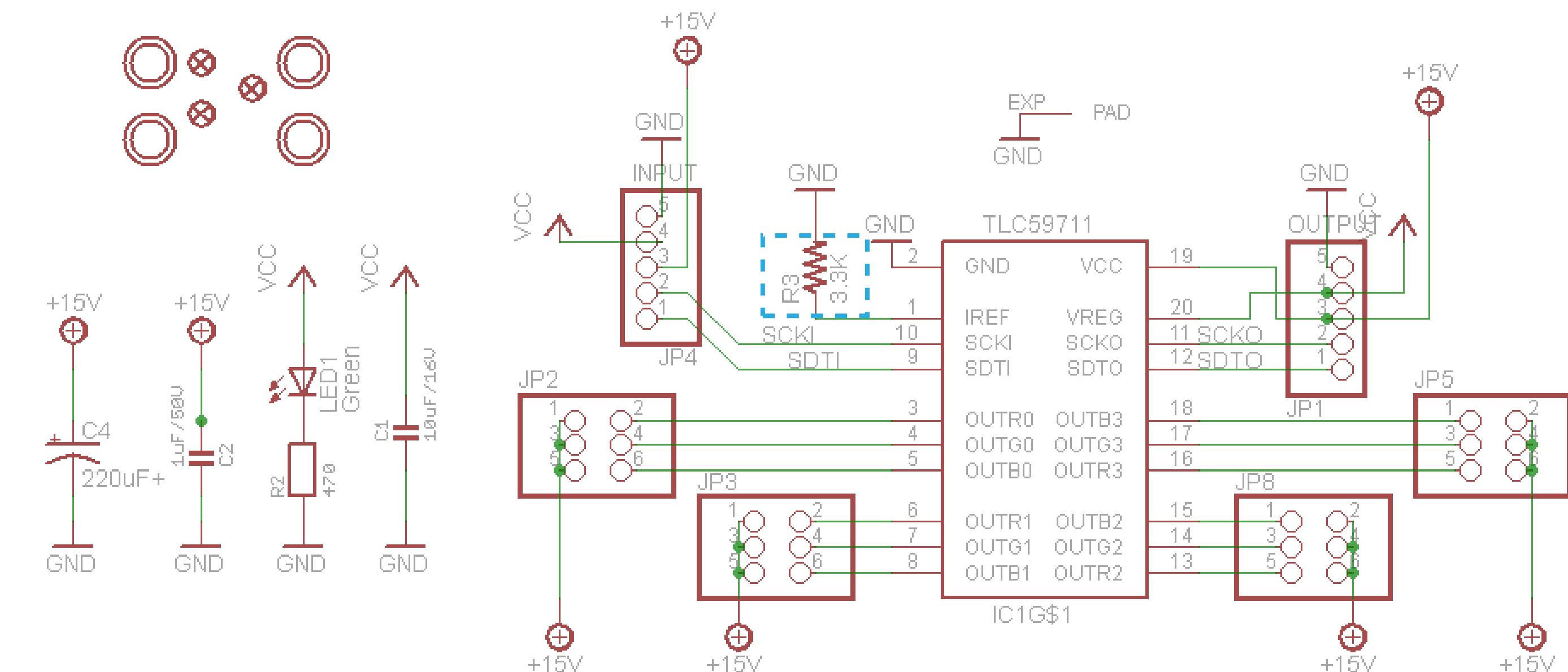
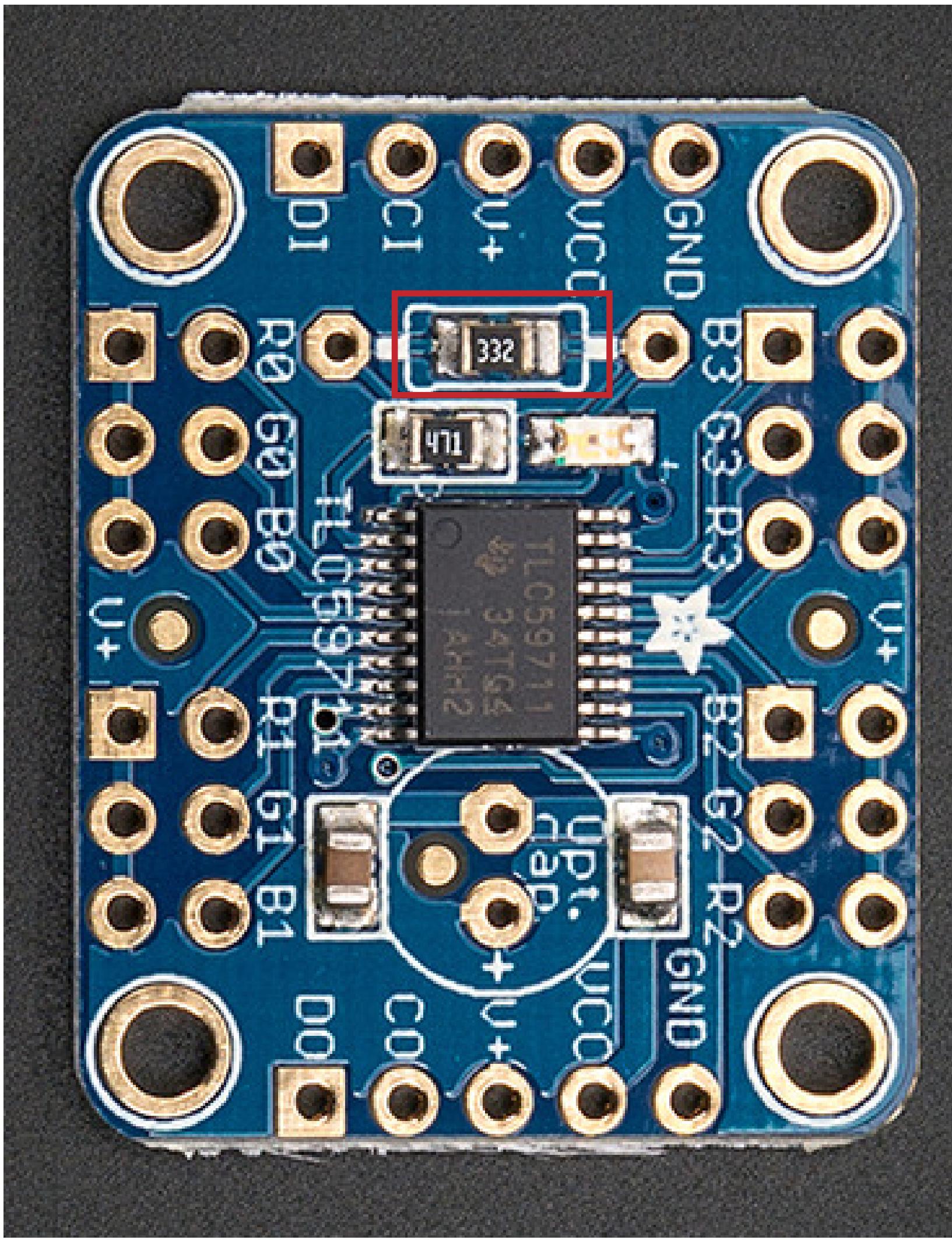


TLC 5947

## WS2812 LED ?

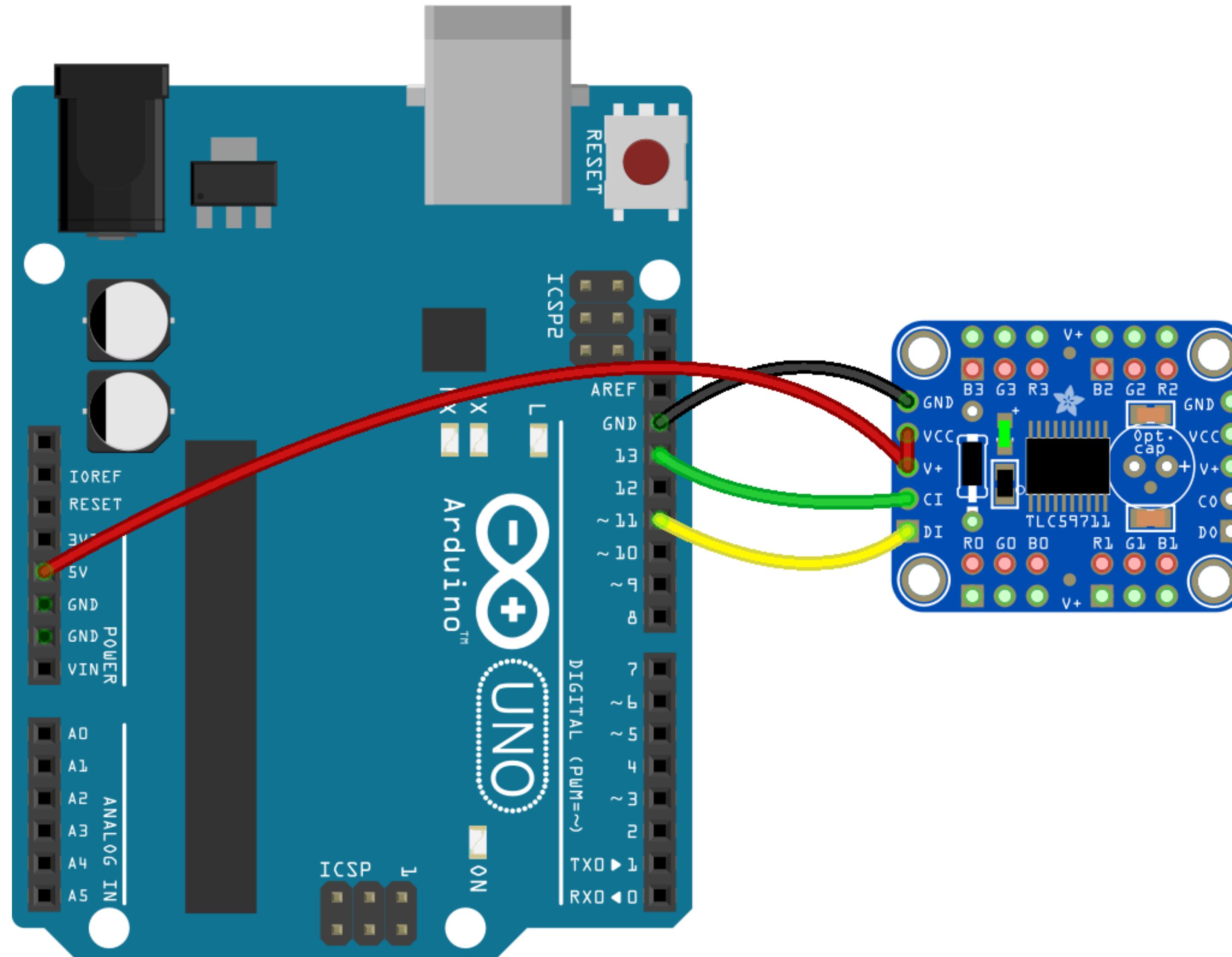




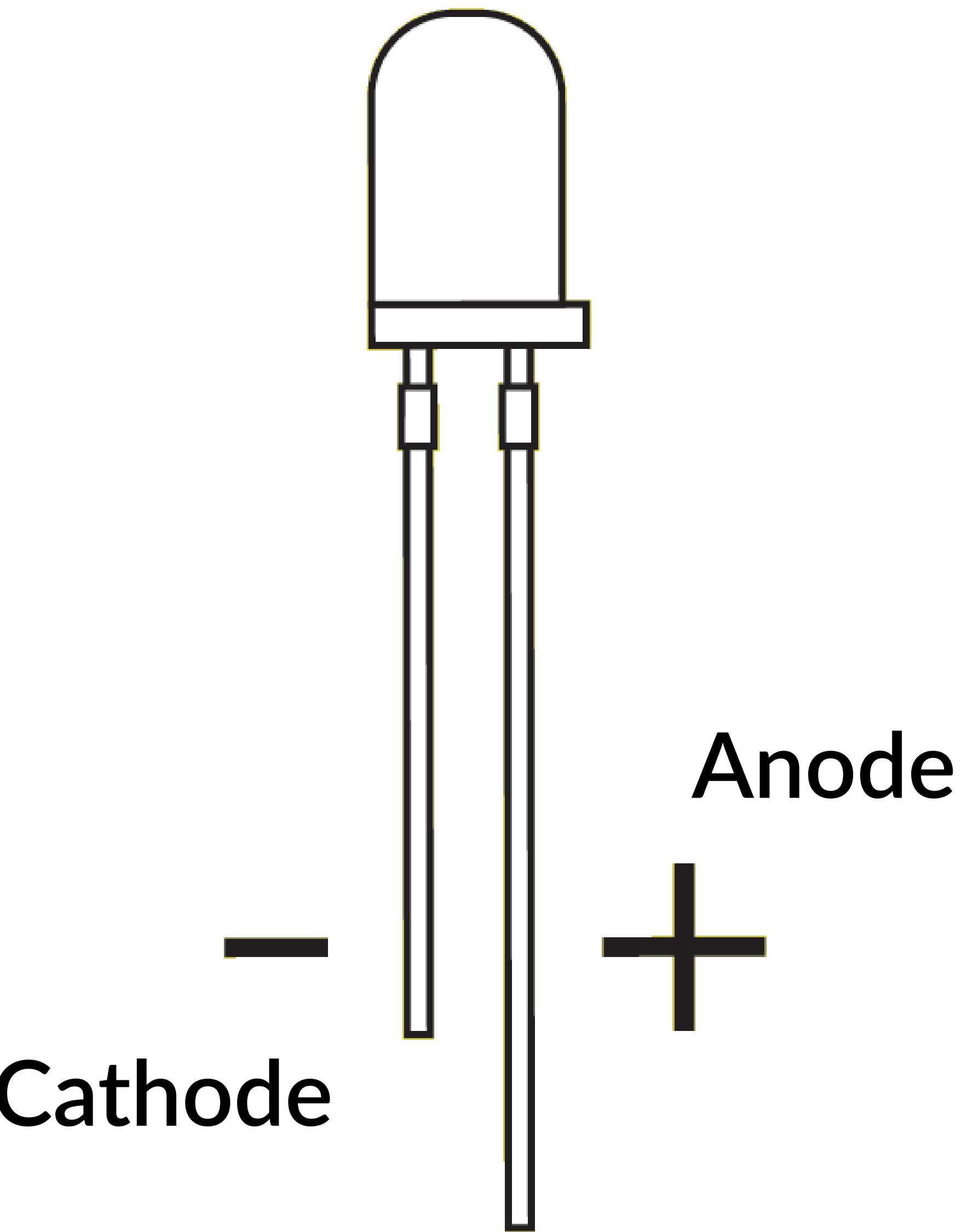
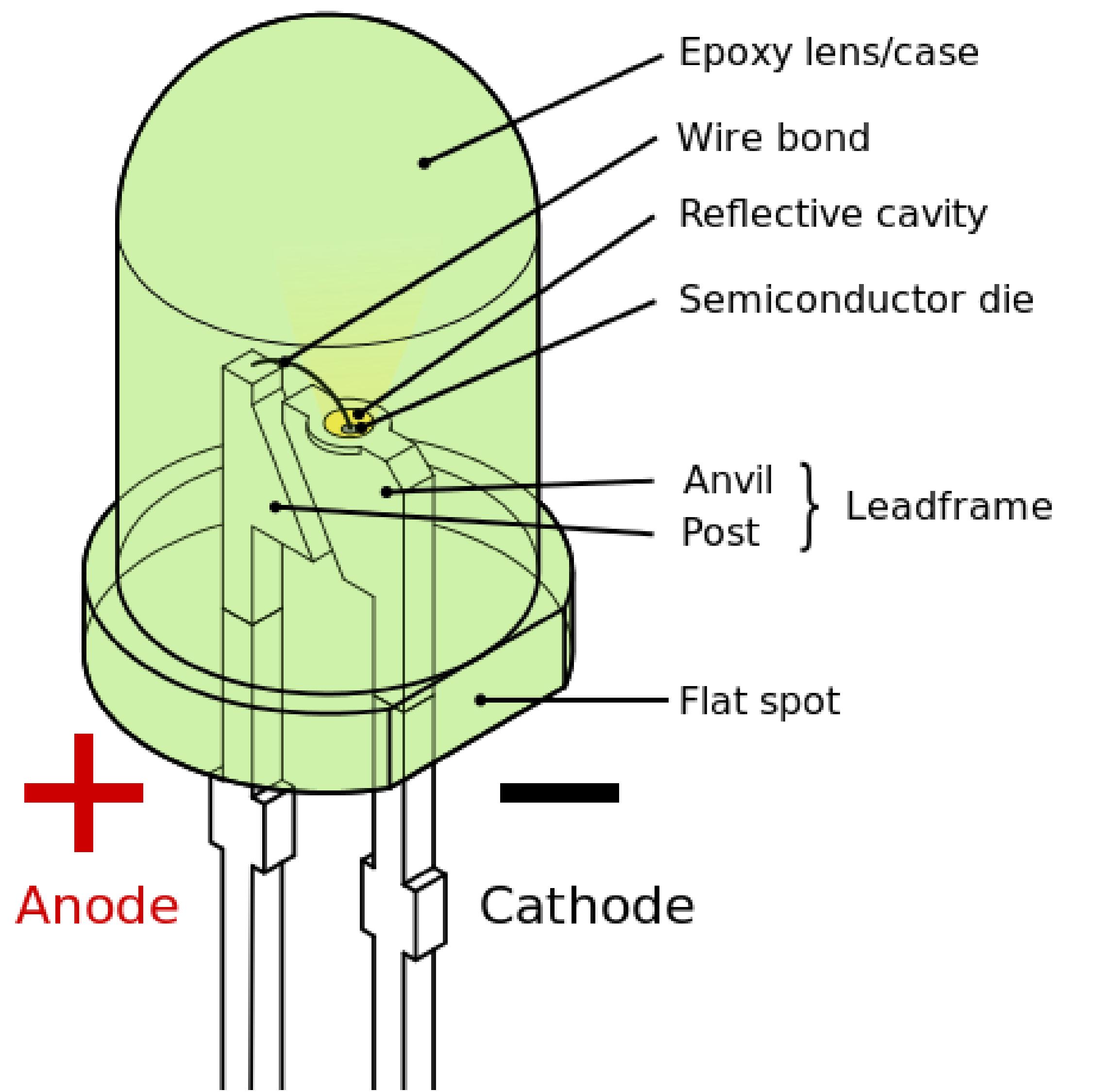


Datasheet page 8, 15

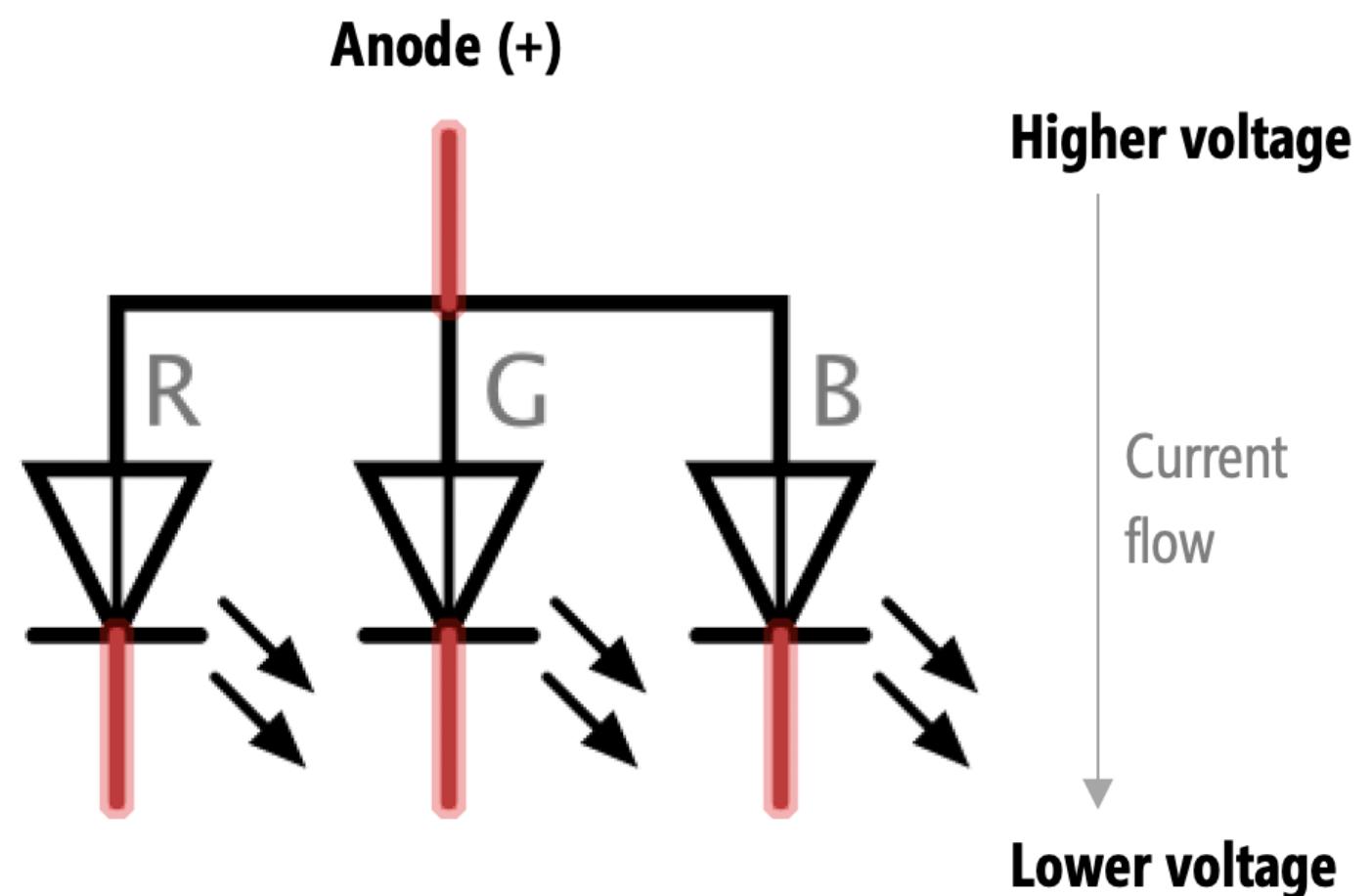
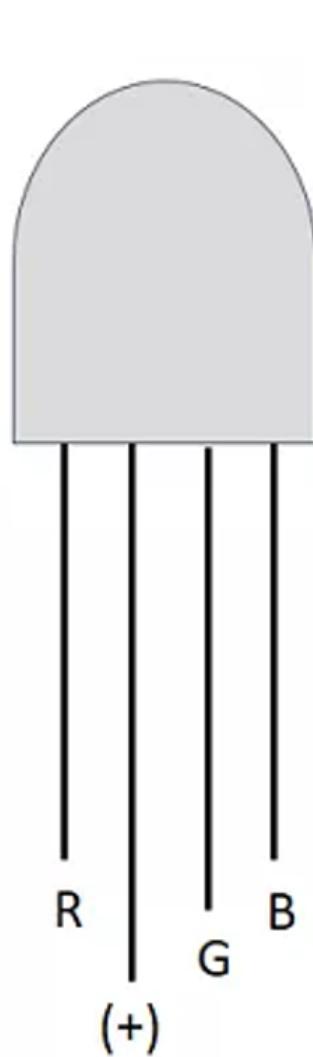
$$R_{IREF} (k\Omega) = \frac{V_{IREF} (V)}{I_{OLCMax} (mA)} \times 41$$



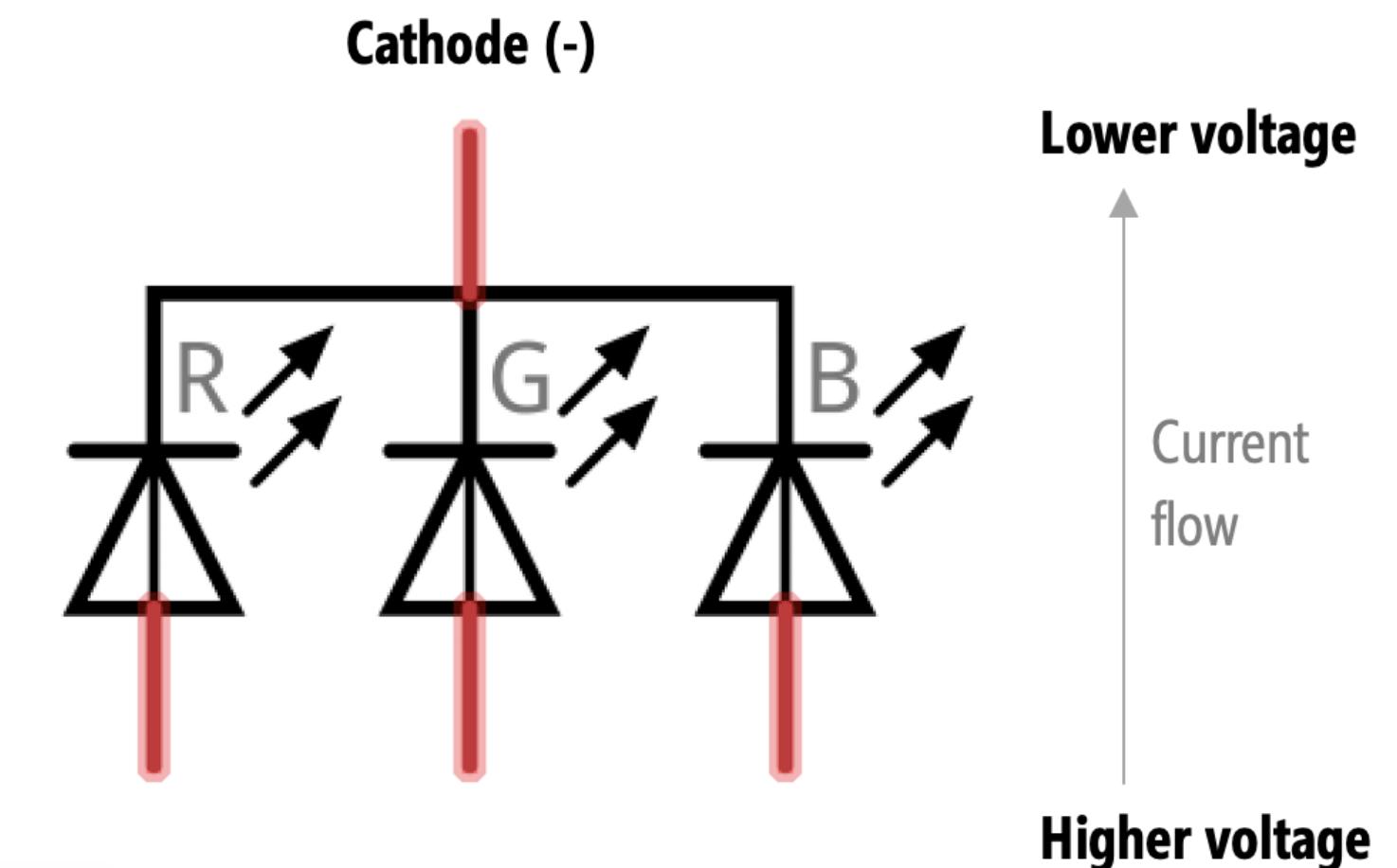
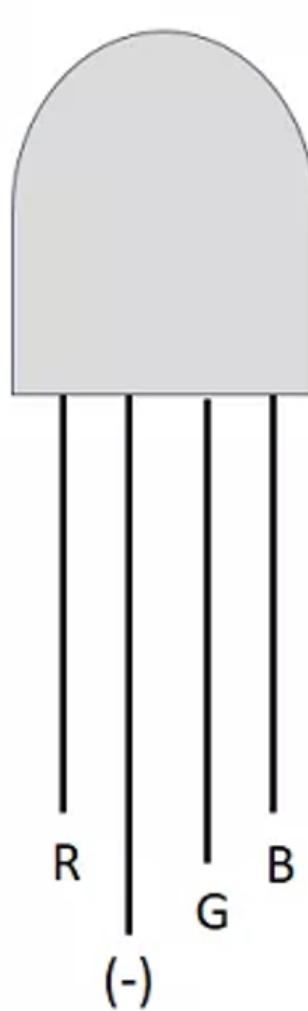
# fritzing

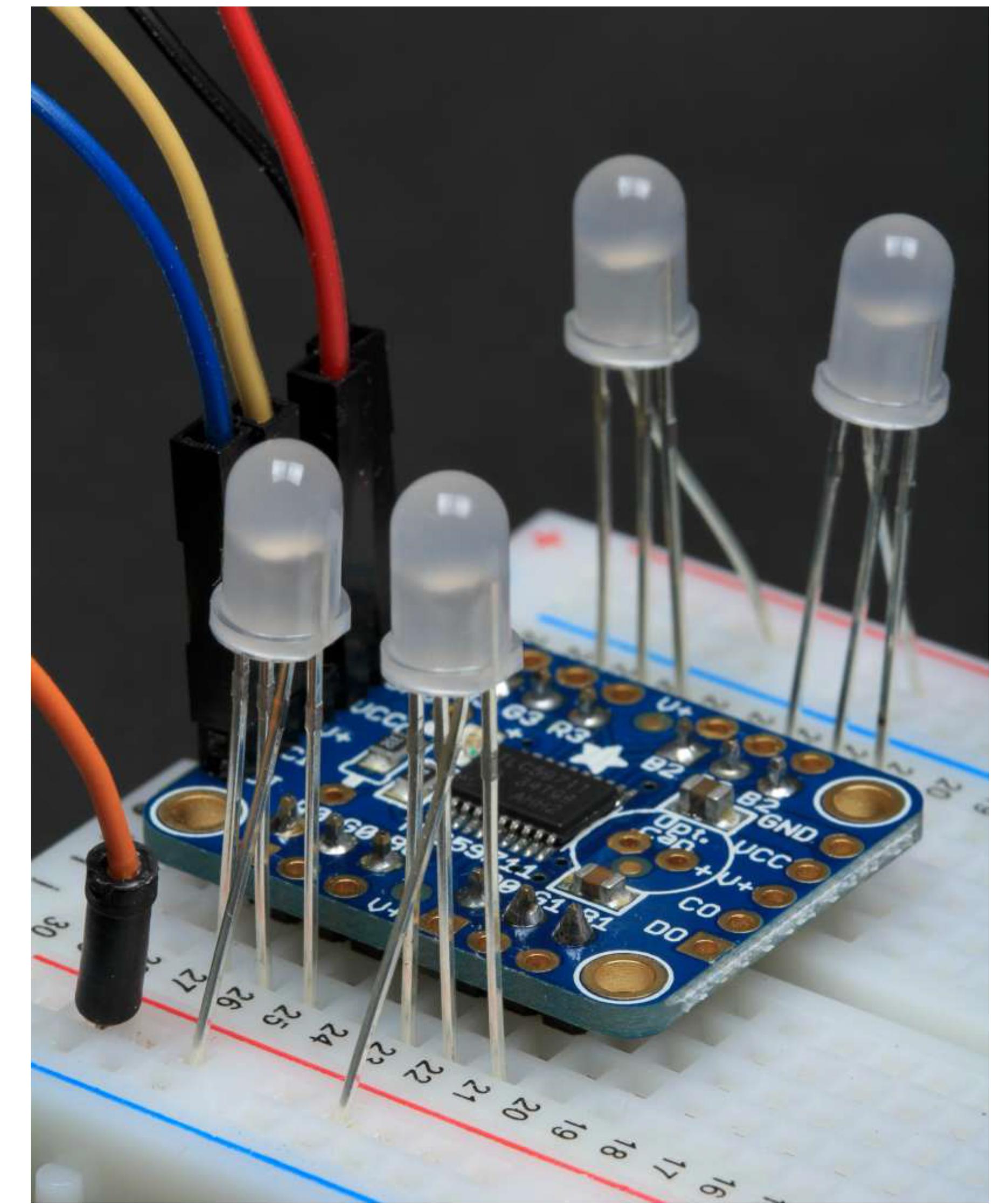
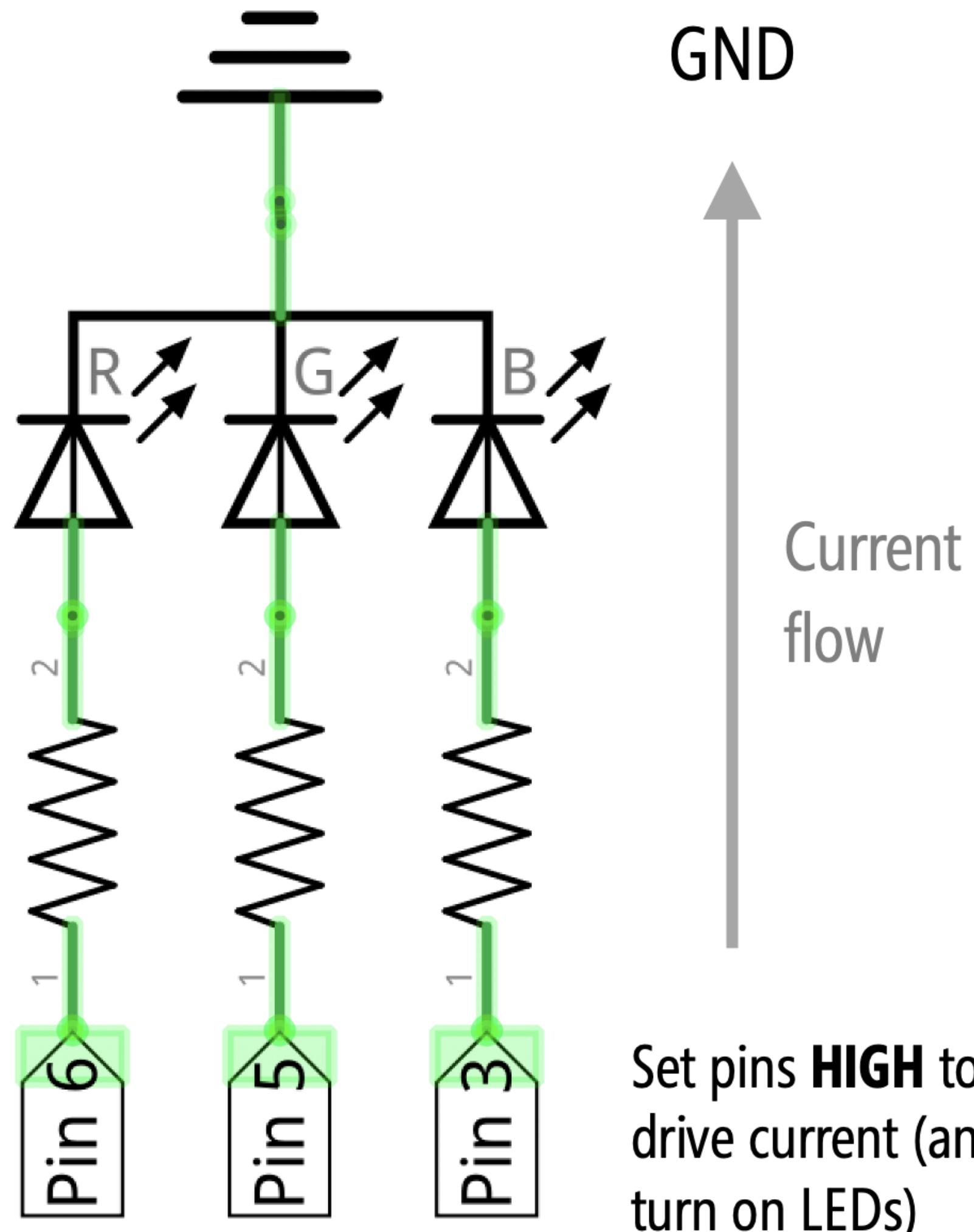


## COMMON ANODE (+)



## COMMON CATHODE (-)





```
Adafruit_TLC59711 tlc = Adafruit_TLC59711(NUM_TLC59711, clock, data);
```

```
Adafruit_TLC59711::Adafruit_TLC59711(uint8_t n, uint8_t c, uint8_t d) {  
    numdrivers = n;  
  
    BCr = BCg = BCb = 0x7F; // default 100% brigtness
```

```
pwmbuffer = (uint16_t *)calloc(2, 12 * n);
```

```
void setLED(uint8_t lednum, uint16_t r, uint16_t g, uint16_t b) {  
    setPWM(lednum * 3,      r); —  
    setPWM(lednum * 3 + 1,  g); —  
    setPWM(lednum * 3 + 2,  b); —
```

```
void setPWM(uint8_t chan, uint16_t pwm) {  
    if (chan > 12 * numdrivers)  
        return;  
    pwmbuffer[chan] = pwm;  
}
```

```
pwmbuffer = (uint16_t *)calloc(2, 12 * n);
```

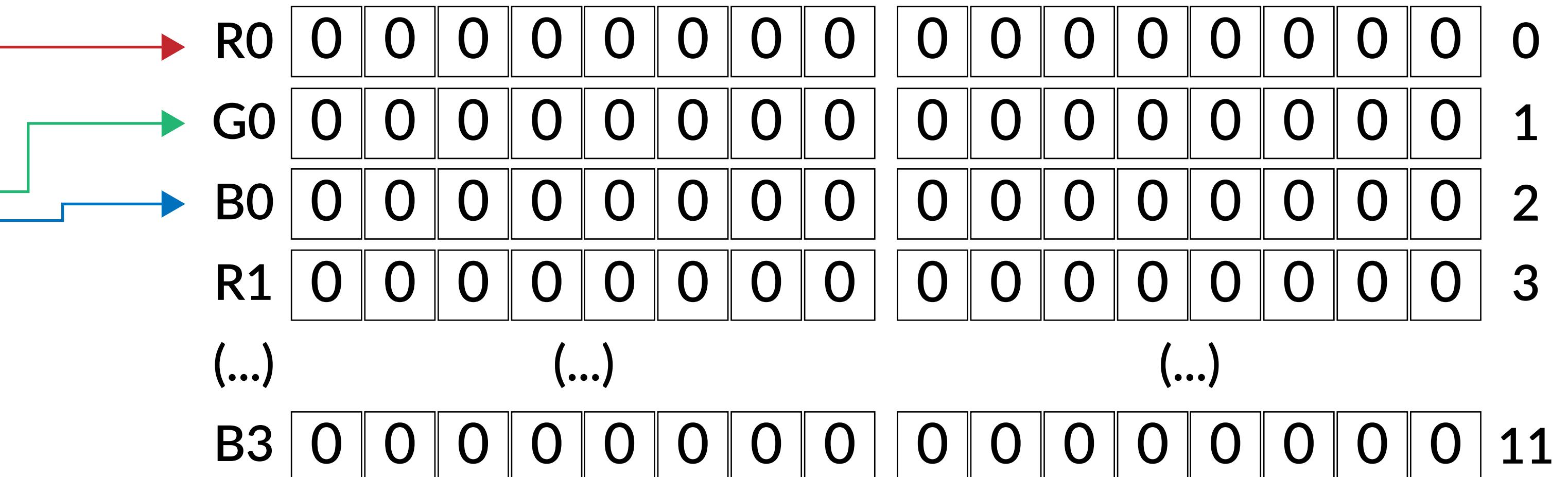
**1 block = unsigned 16bit**  
**2 Object, 12 element(byte), n = 1 (One driver chip)**

8bit = 1byte memory block

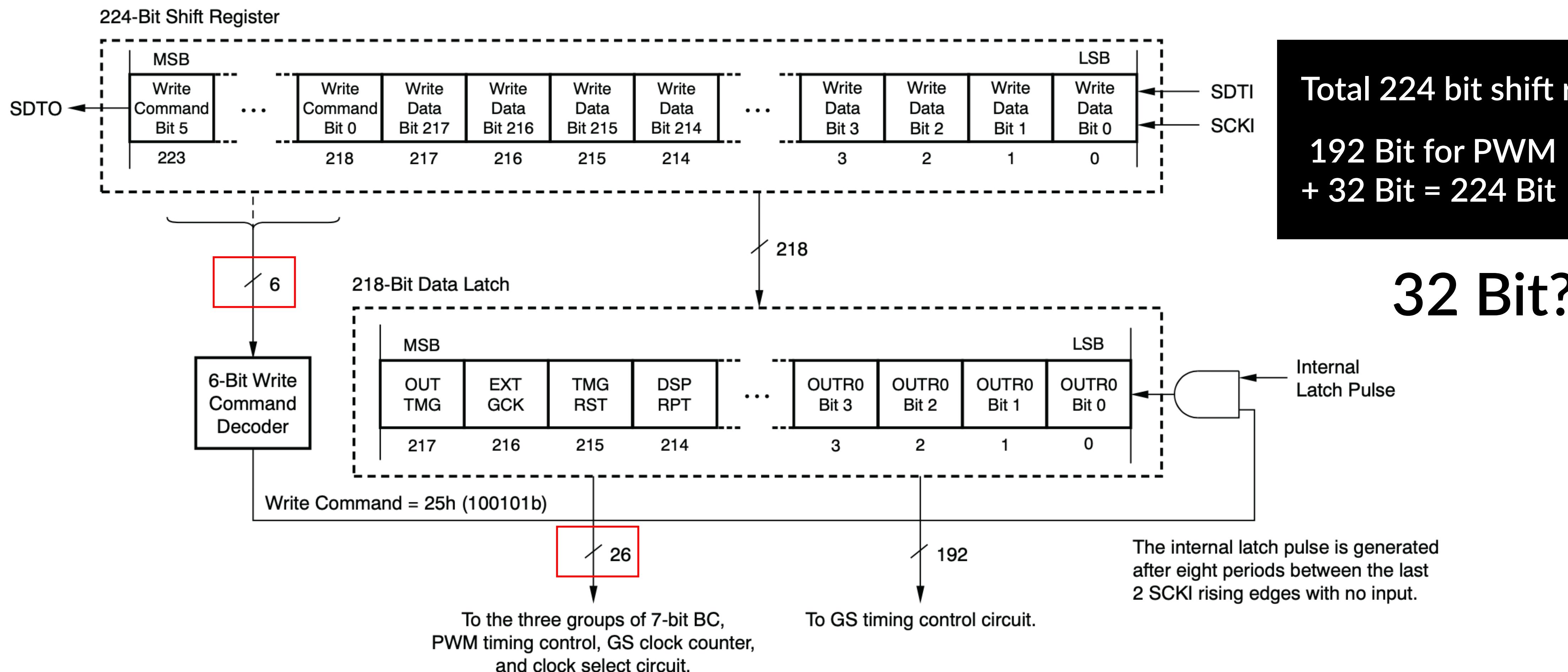
---

# Array object 1

8bit = 1byte memory block

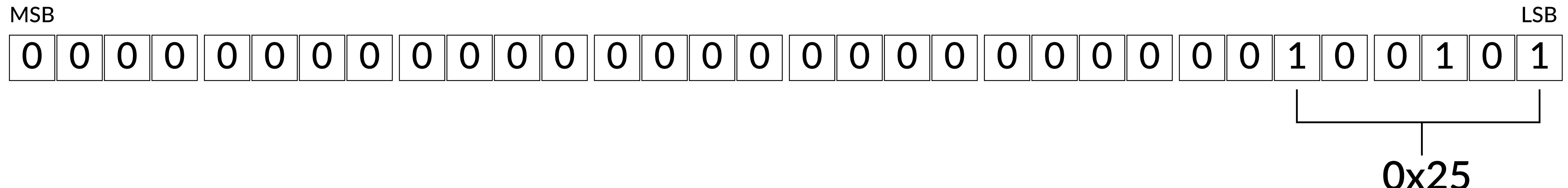


# 192 Bit

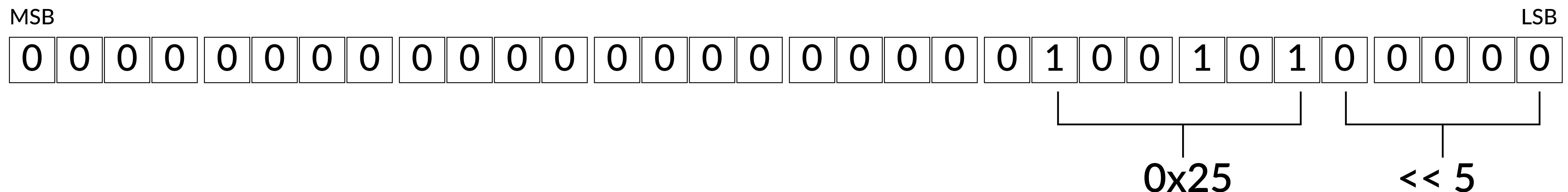


**Figure 23. Common Shift Register and Control Data Latch Configuration** Page 20 / 26

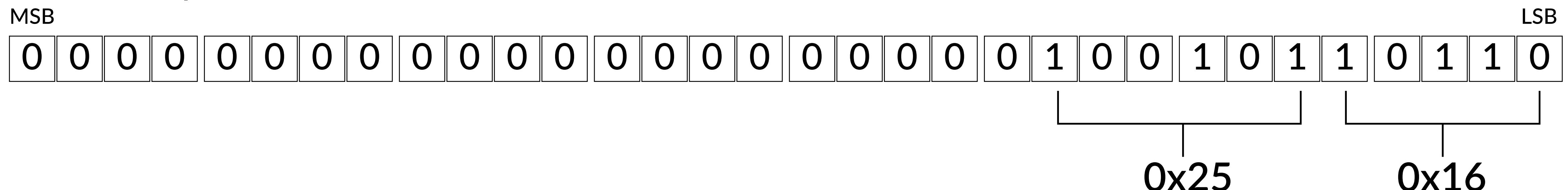
```
uint32_t Command = 0x25  (0010 0101)
```



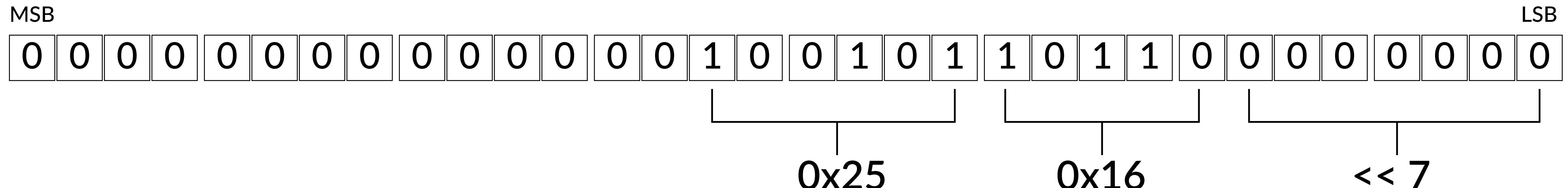
# Command <<= 5



Command |= 0x16 (0001 0110) (OUTTMG / EXTGCK / TMGRST / DSPRPT / BLANK) Page 21

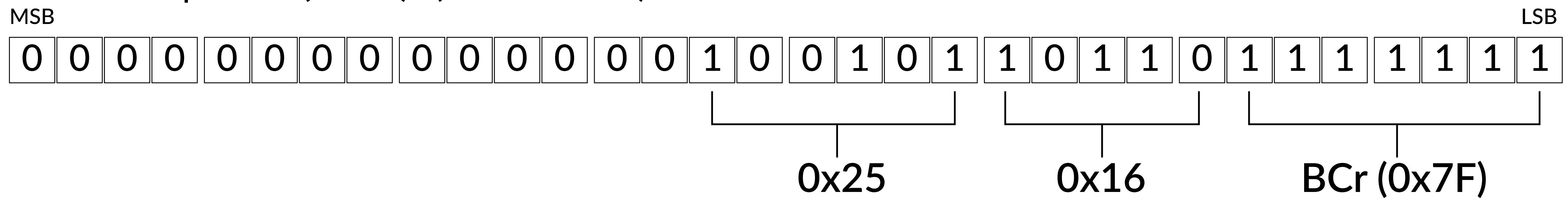


# Command <<= 7

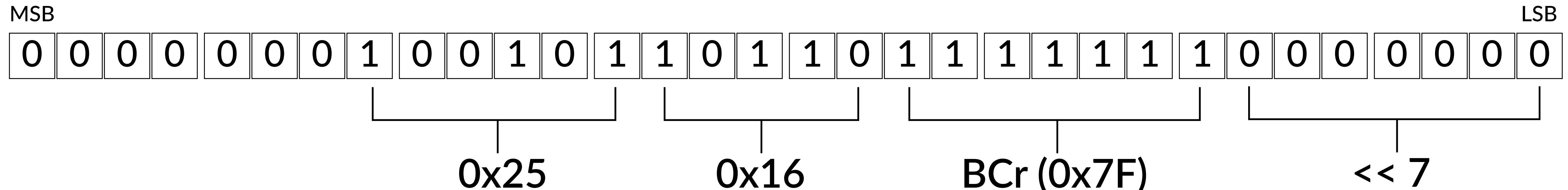


Command |= BCr (0x7F) (0111 1111) Page 16

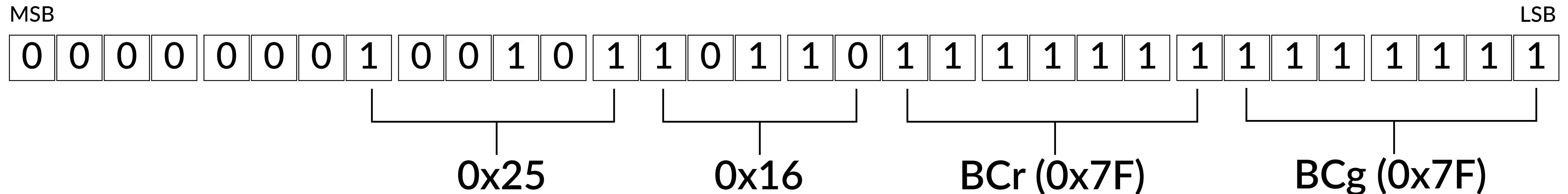
MSB



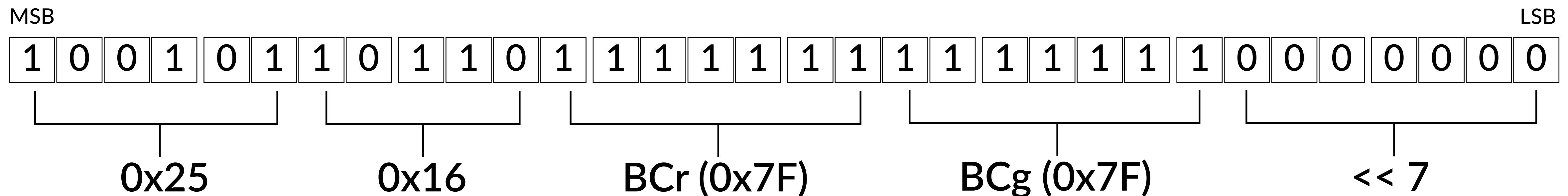
# Command <<= 7



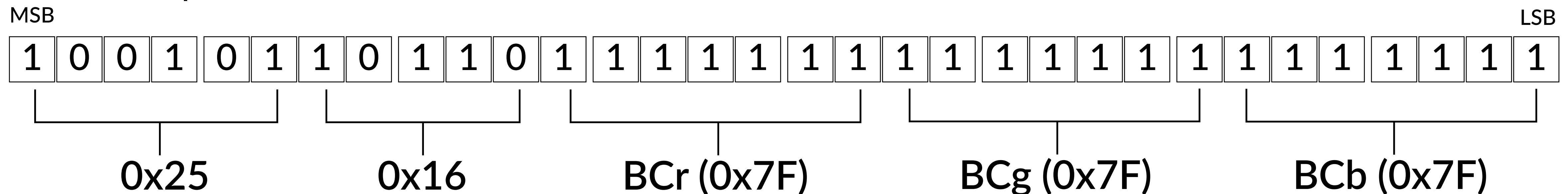
**Command |= BCg (0x7F)**



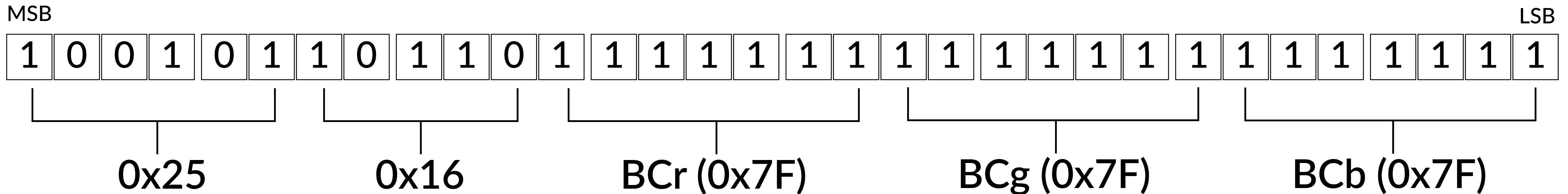
# Command <<= 7



# Command |= BCb (0x7F)



## Command (32 Bit)



(32b) command = **1001 0110 1101 1111 1111 1111 1111 1111**

**8 bit / MSB First (L->R)**

```
_spi_dev->transfer(command >> 24);
```

```
_spi_dev->transfer(command >> 16);
```

```
_spi_dev->transfer(command >> 8);
```

```
_spi_dev->transfer(command);
```



Ex) Number of driver = 1

C = 11    \_spi\_dev->transfer(pwmbuffer[11] >> 8);  
          \_spi\_dev->transfer(pwmbuffer[11]);

(16b)

(10000)

0000 0000 0010 0111  
0000 0000 0010 0111  
0010 0111 0001 0000

C = 10    \_spi\_dev->transfer(pwmbuffer[10] >> 8);  
          \_spi\_dev->transfer(pwmbuffer[10]);

(24300)

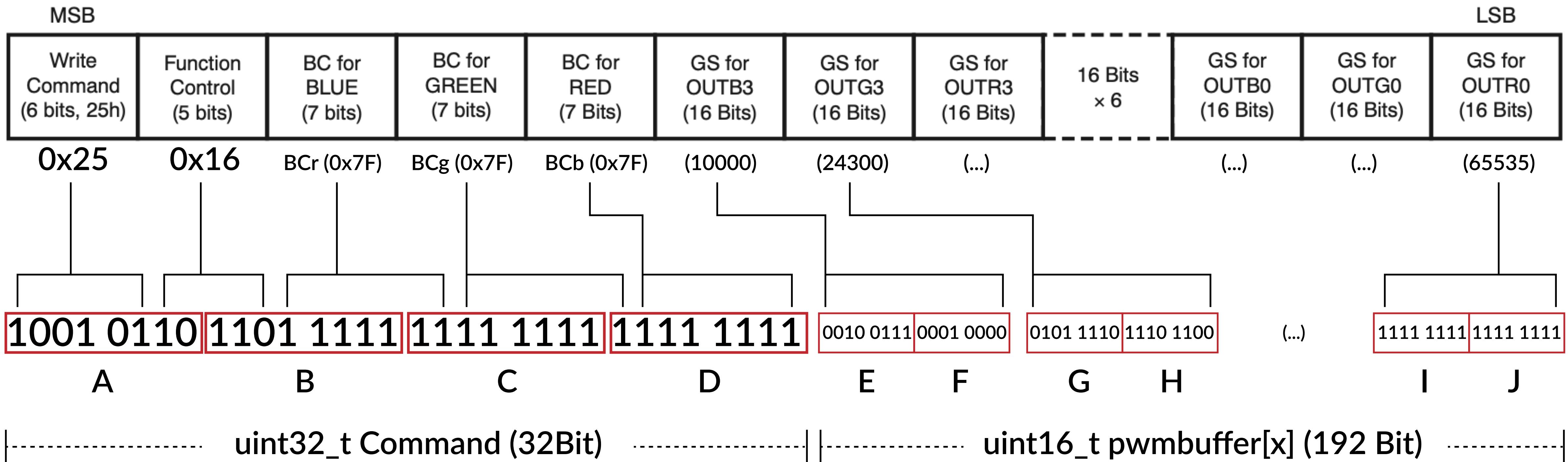
0101 1110 1110 1100  
0000 0000 0101 1110  
0101 1110 1110 1100

...                ...

C = 0    \_spi\_dev->transfer(pwmbuffer[0] >> 8);  
          \_spi\_dev->transfer(pwmbuffer[0]);

(65535)

1111 1111 1111 1111  
0000 0000 1111 1111  
1111 1111 1111 1111



```
_spi_dev->transfer(command >> 24); 0000 0000 0000 0000 0000 0000 1001 0110 A  
_spi_dev->transfer(command >> 16); 0000 0000 0000 0000 1001 0110 1101 1111 B  
_spi_dev->transfer(command >> 8); 0000 0000 1001 0110 1101 1111 1111 1111 C  
spi dev->transfer(command); 1001 0110 1101 1111 1111 1111 1111 1111 D
```

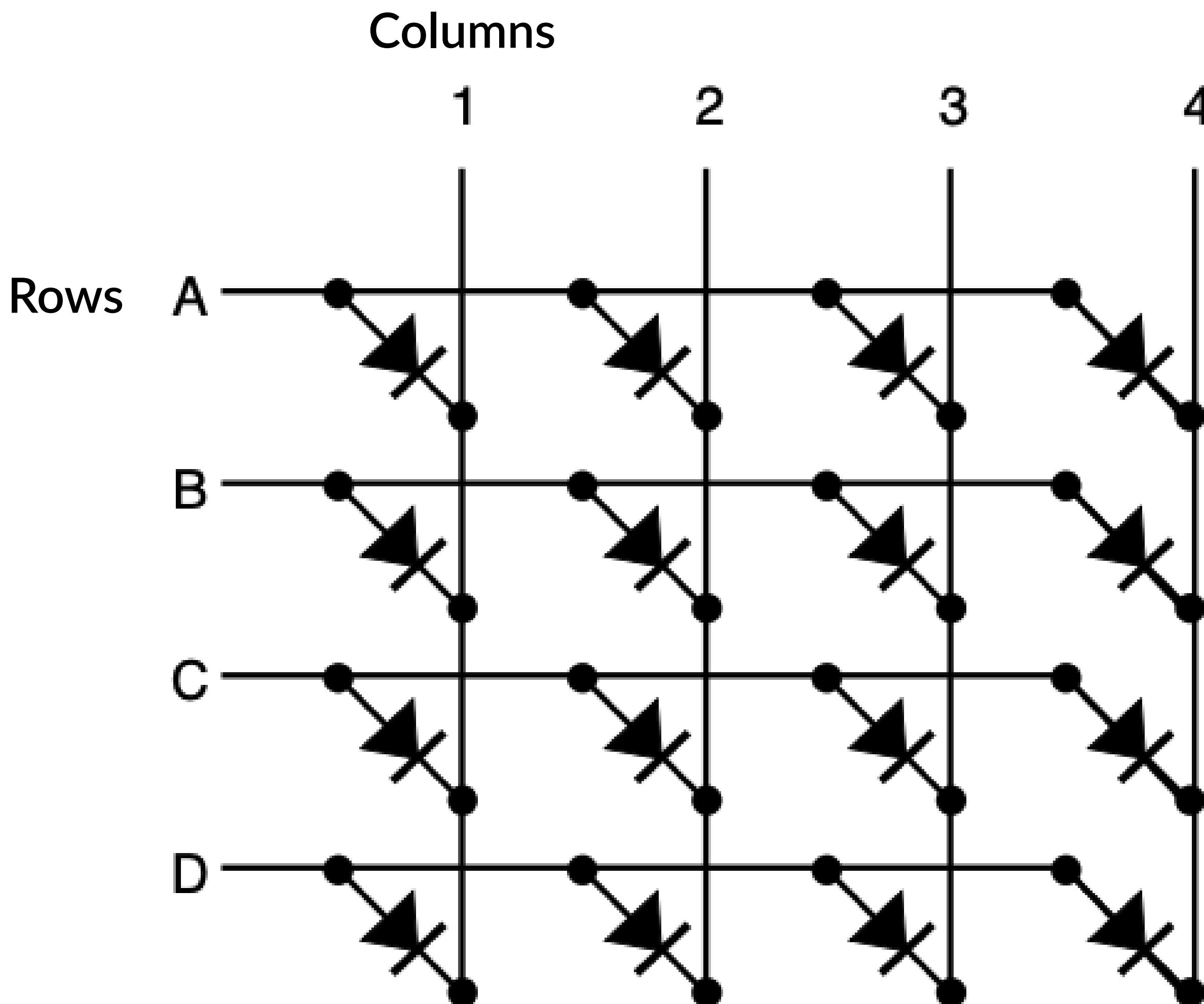
|   |           |           |     |
|---|-----------|-----------|-----|
| <code>_spi_dev-&gt;transfer(pwmbuffer[11] &gt;&gt; 8);</code> | 0000 0000 | 0010 0111 | E   |
| <code>_spi_dev-&gt;transfer(pwmbuffer[11]);</code>            | 0010 0111 | 0001 0000 | F   |
| <code>_spi_dev-&gt;transfer(pwmbuffer[10] &gt;&gt; 8);</code> | 0000 0000 | 0101 1110 | G   |
| <code>_spi_dev-&gt;transfer(pwmbuffer[10]);</code>            | 0101 1110 | 1110 1100 | H   |
| ...   | ...       | ...       | ... |
| <code>_spi_dev-&gt;transfer(pwmbuffer[0] &gt;&gt; 8);</code>  | 0000 0000 | 1111 1111 | I   |
| <code>spi dev-&gt;transfer(pwmbuffer[0]);</code>              | 1111 1111 | 1111 1111 | J   |

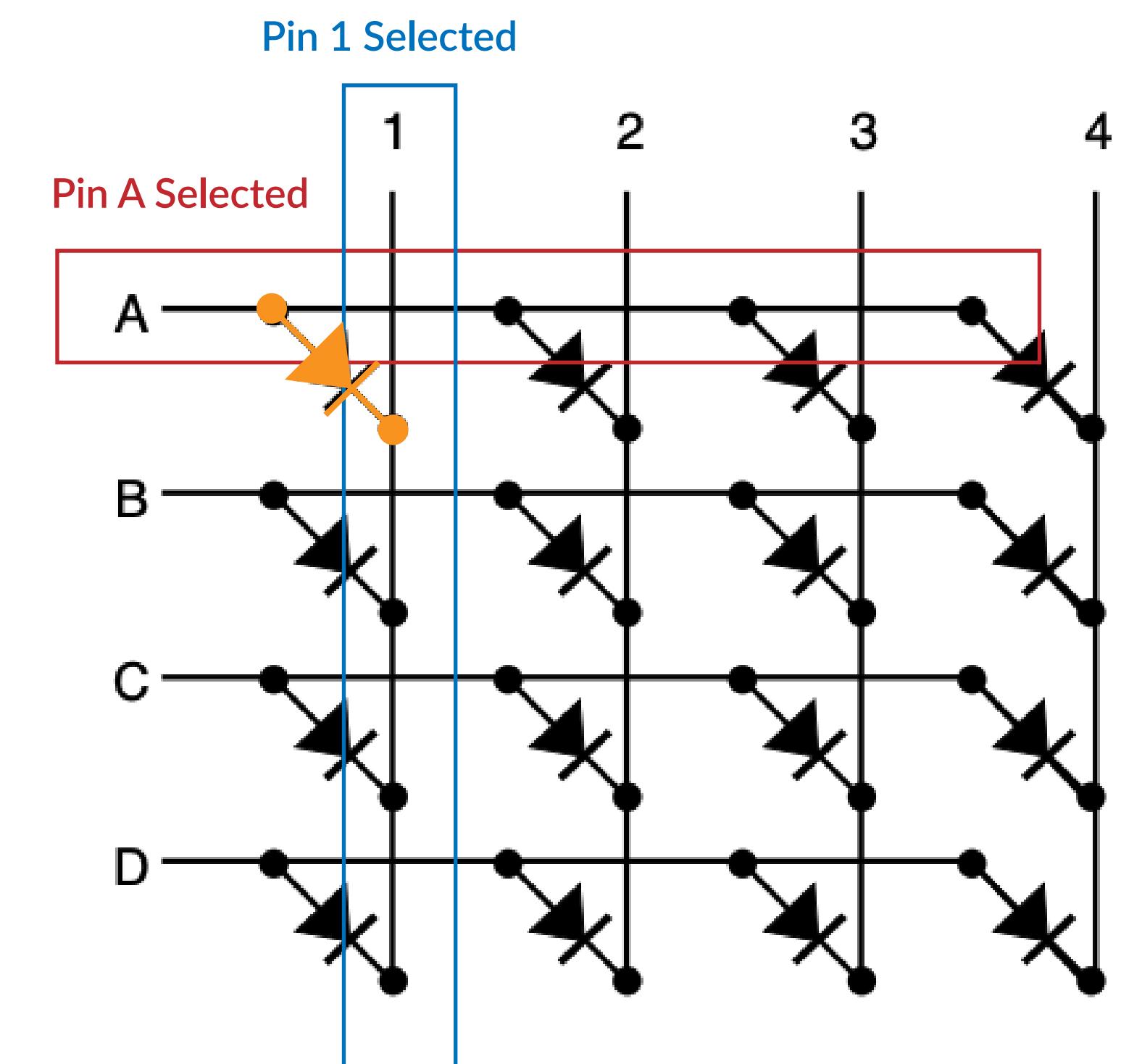
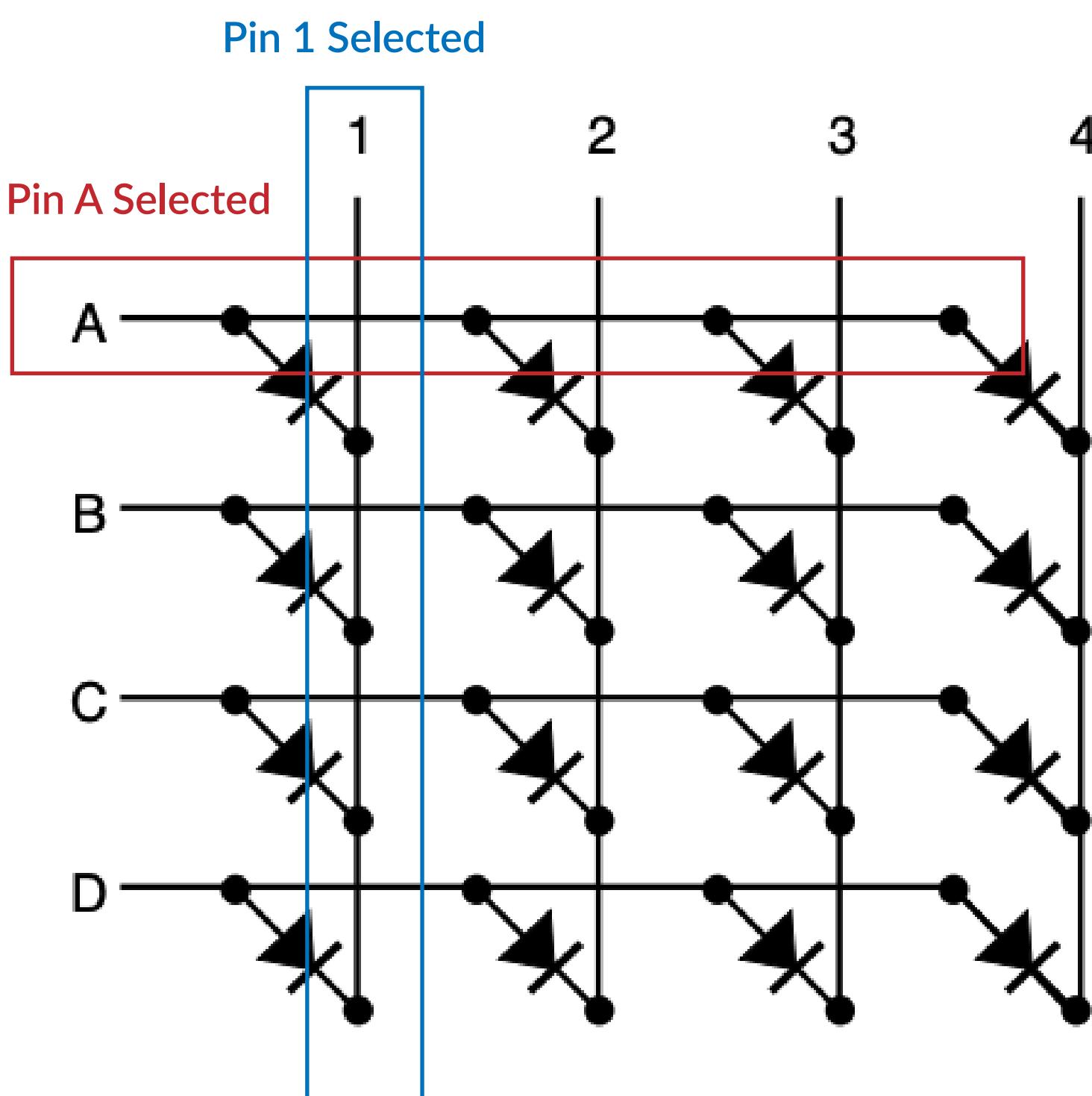
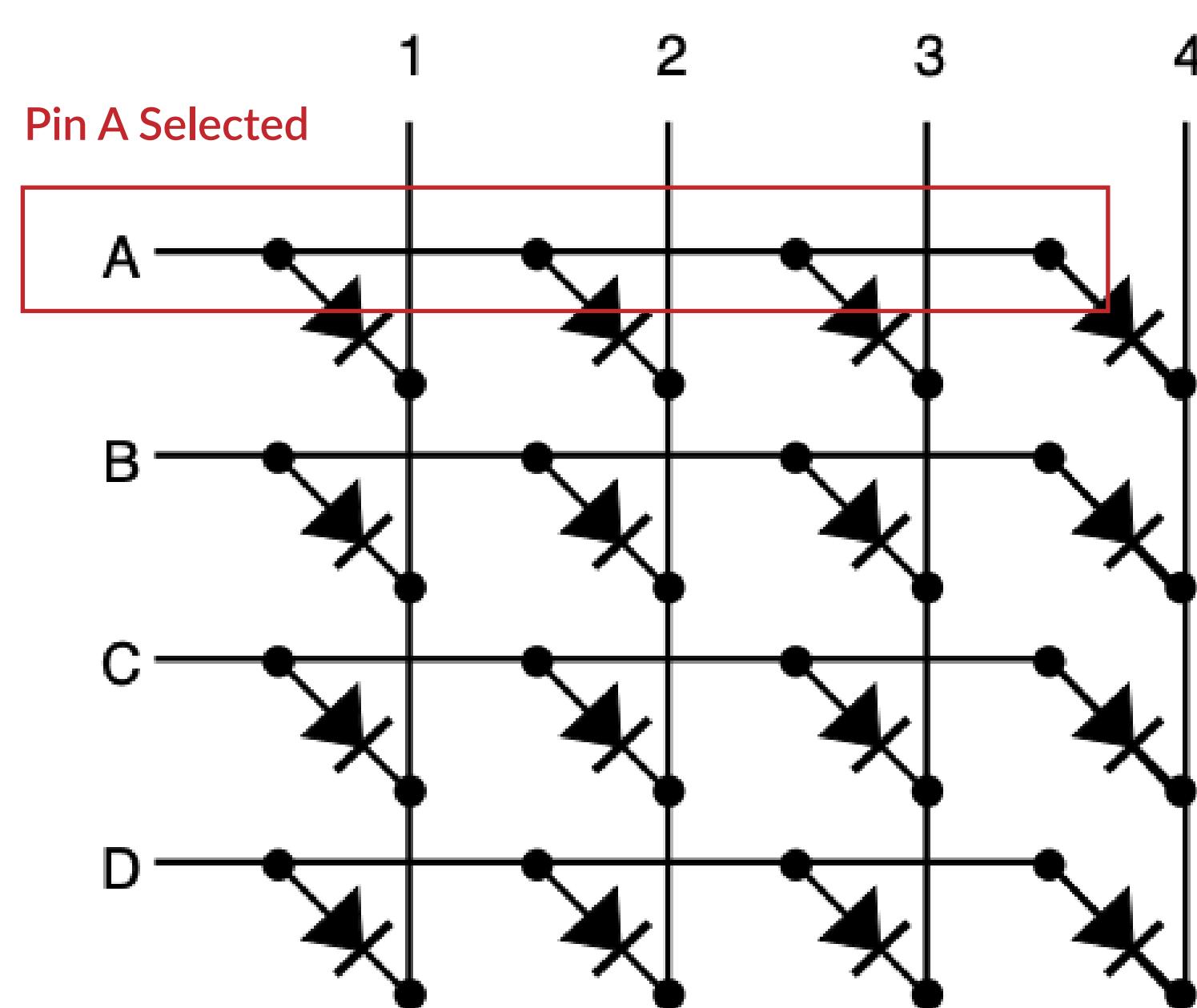
# Introduction of Multiplexing

Fewer component

Simplifying the pcb

Consuming less power





# Multiplexing PWM

