

LED it be.

Van led tot intelligente led. Kleur en meer,
hoe werkt het en hoe aan te sturen.

LED:

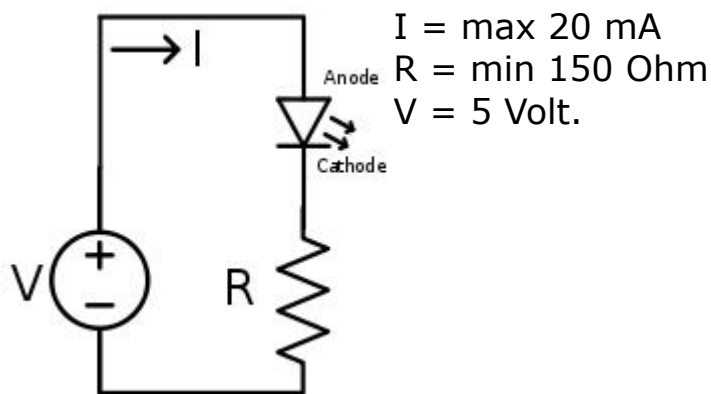
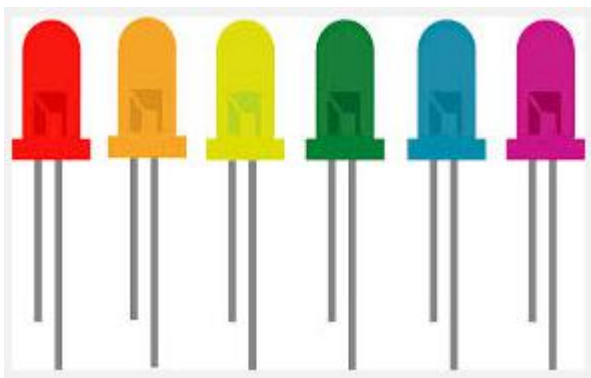


Fig.1 wet v Ohm

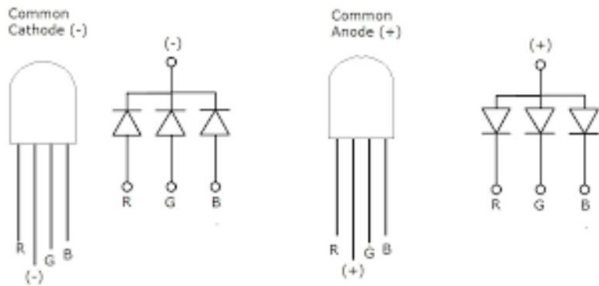
Een LED is in principe een diode die licht van een bepaalde golflengte uitzendt. Als we een LED aan willen sturen zetten we er een spanning op. Er valt een spanning van ongeveer 2 Volt over de led. Er mag maximaal 20 mA stroom door de LED stromen. Zetten we 5 Volt over de led, dan gebruikt de led er 2 en houden we er 3 over om de 20 mA te maken. Hier nemen we de Wet van Ohm $I = U / R$ en hieruit herleiden we dat we bij een spanning van 3 volt en een maximale stroom van 20 miliAmpere een weerstand van 150 Ohm mogen hebben. Elke kleur LED heeft een andere spanningsval dus moeten we in theorie voor elke kleur LED een andere weerstand kiezen.



Kort = - = Kathode
Lang = + = Anode

RGB-LED:

Willen we meerdere kleuren in een LED hebben dan kunnen we kiezen voor een RGB-LED. Bij een RGB-LED zitten er in feite 3 led's met een verschillende kleur in 1 behuizing. Er zitten dan 4 aansluitingen aan. 1 Algemene aansluiting en 3 aansluitingen voor de verschillende LED's. De algemene aansluiting kan de kathode (-) zijn of de anode (+) en op de 3 andere aansluitingen moeten we dan via een weerstand (die in theorie alle drie verschillend moeten zijn) een microcontroller of buffer-IC aansluiten.



Intelligente LED's

Eerder hebben we normale led en de RGB-led bekeken. Nu gaan we naar de intelligente LED's. Hiermee bedoelen we dat er IN de LED een controller in de behuizing aanwezig is. Er zijn 2 bekende controllers, te weten de WS2812 en de APA102. en nu behandelen we de WS2812.

NeoPixel WS2812:



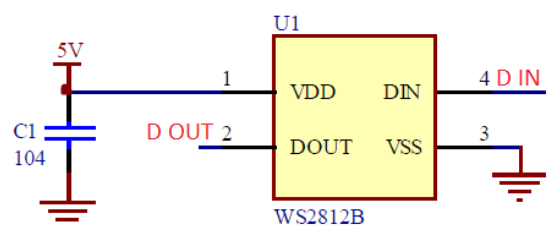
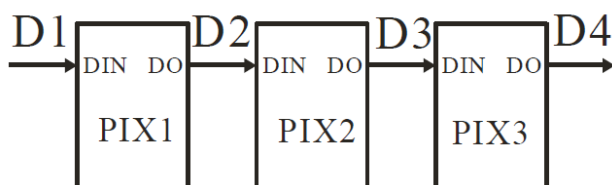
Deze afbeelding is een LED-array van 40 NeoPixels gemonteerd op een Arduino-shield van Adafruit.

Voor deze LED's zijn library's op internet te vinden voor verschillende processoren. Met behulp van deze library's zijn de NeoPixel's eenvoudig aan te sturen.

Werking van de NeoPixel:

Een NeoPixel is een RGB-LED met een ingang en een uitgang. In feite zetten we alle aan te sturen LED's in serie. Vanuit onze microcontroller sturen we naar de Data-IN van de 1e LED. Vanuit de Data-UIT van de 1e

Cascade method:



LED gaan we naar de Data-IN van de 2e LED. Enzovoort tot aan de laatste LED die we willen gaan besturen. Dit noemen we Daisy-Chain. Per NeoPixel sturen we 8 Bit's per kleur en dit geeft dus 24 bit's per LED.

Composition of 24bit data:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

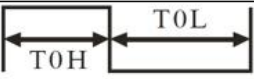
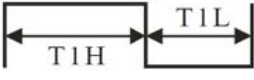
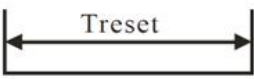
Note: Follow the order of GRB to sent data and the high bit sent at first.

Voor elke LED die op de rij aanwezig is (in ons voorbeeld 3) sturen we dus 24 bits en dat maakt dus $3 * 24$ bit's is 72 bit's.

Een bit is als volgt opgebouwd. Een Tijd_hoog gevolgd door een Tijd_laag , en dan is een 0 een korte Tijd_hoog gevolgd door een lange Tijd_laag en een 1 is een lange Tijd_hoog gevolgd door een korte Tijd_laag.

Sequence chart:

Data transfer time($T_H+T_L=1.25\mu s \pm 600ns$)

0 code		T0H	0 code ,high voltage time	0.35us	$\pm 150ns$
		T0L	0 code , low voltage time	0.8us	$\pm 150ns$
1 code		T1H	1 code ,high voltage time	0.7us	$\pm 150ns$
		T1L	1 code ,low voltage time	0.6us	$\pm 150ns$
RET code		RES	low voltage time	Above 50μs	

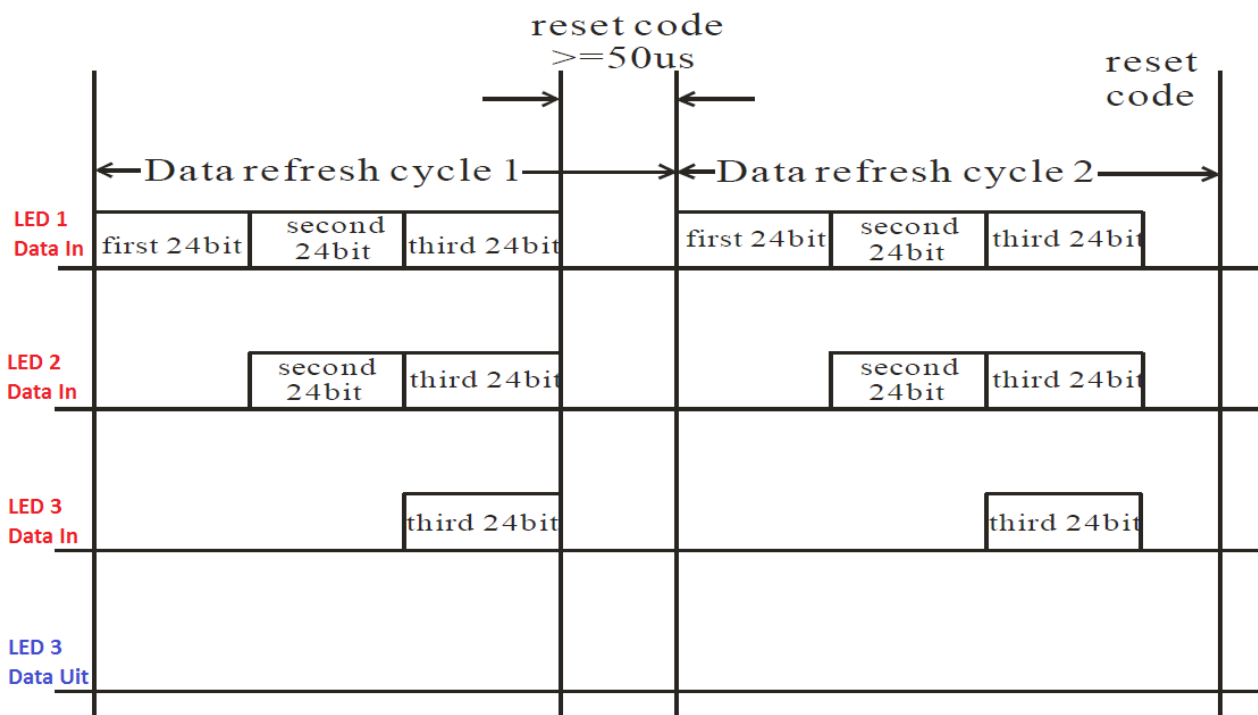
Elke LED gebruikt de 1e 24 bits die binnenkomen op Data-IN voor eigen gebruik (om de 3 LED's aan te sturen) en de overige bit's stuurt hij door naar de Data-UIT naar de volgende LED's. Krijgt een LED gedurende 50 microSeconde GEEN data op de Data-IN binnen dan ziet hij dit als een Reset en zal hij daarna de 1e 24 bit's weer voor eigen gebruik gebruiken en de rest doorsturen naar Data-UIT.

Dus de 1e NeoPixel krijgt als eerste datastroom binnen, gebruikt de 1e 24 bits voor eigen gebruik en stuurt de rest verder naar de uitgang. De 2e NeoPixel krijgt de volgende 24 bits binnen, houdt deze voor eigen gebruik en stuurt de rest weer verder naar de uitgang, enz. enz. enz. Aan de uitgang van de laatste NeoPixel verschijnt als de aansturing goed geprogrammeerd is geen databit's meer. Na een pauze van minimaal 50 uSec kan de reeks databit's weer opnieuw beginnen.

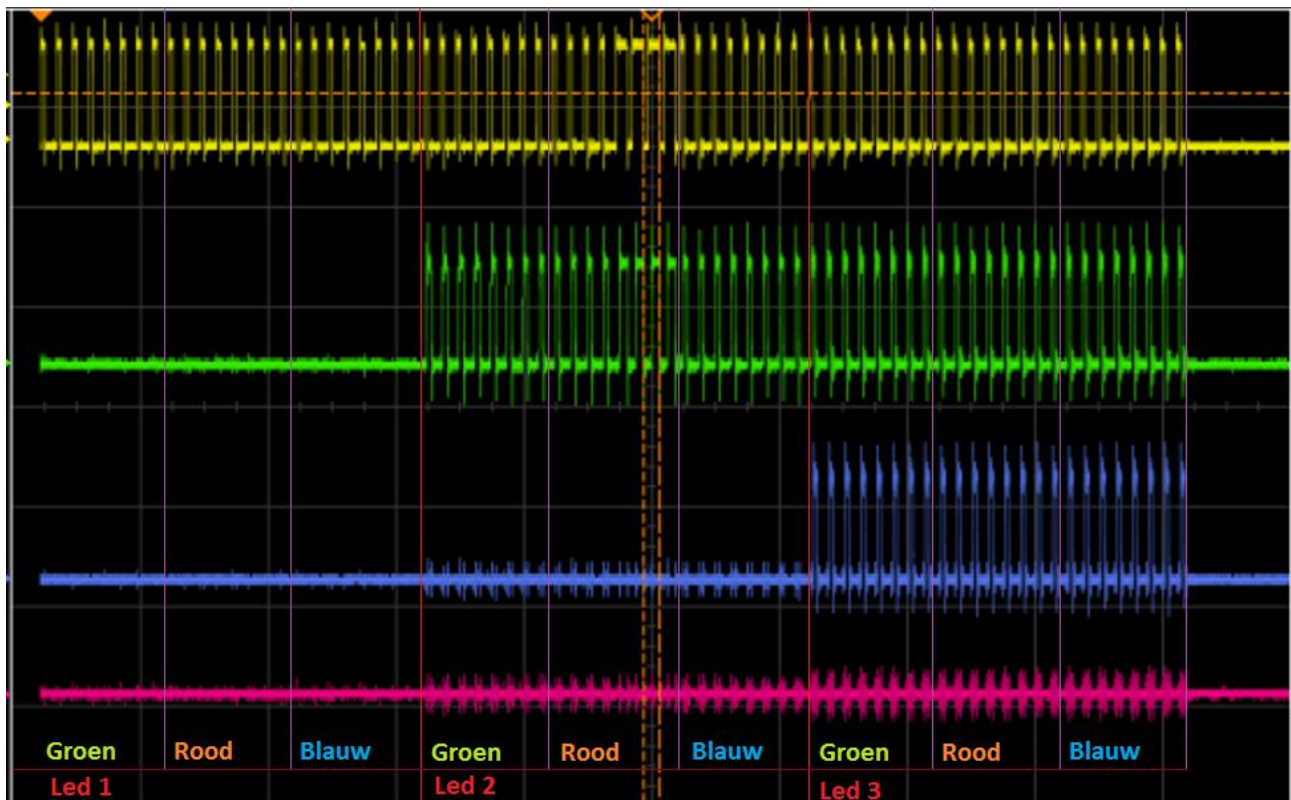
Van elke 24 databit's zijn de eerste bestemd voor de groene led, de 2e 8 bit's voor de rode led en de laatste 8 bit's voor de blauwe led in een NeoPixel.

In onderstande afbeelding en scoopbeeld is dit weergegeven.

Data transmission method:

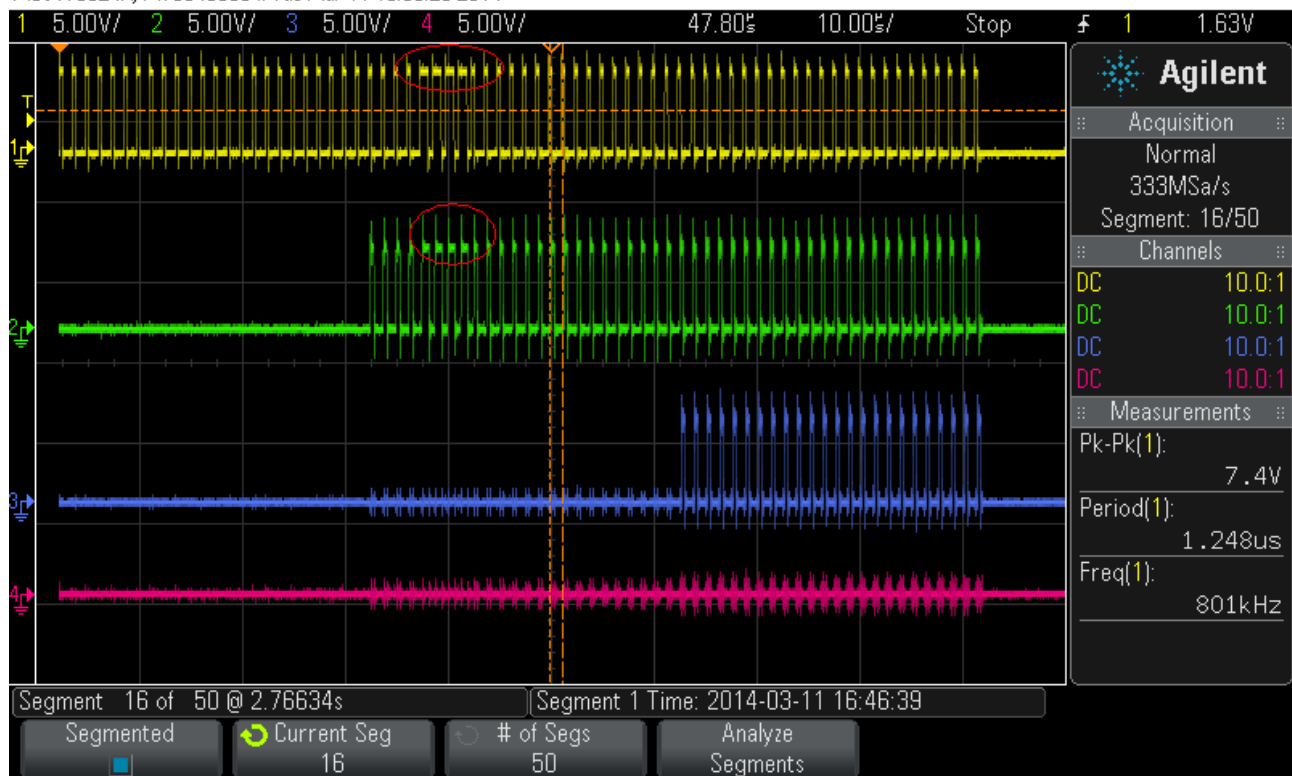


Bovenstaande afbeelding zien we als volgt op een scoop als we de LED's volgens het bijschrift aansturen. Let wel goed op dat de reeks Groen, Rood en Blauw is. Onderstaande scoopafdruk lat zien dat de 2e LED rood gaat branden en de overige led's niet gaan branden.

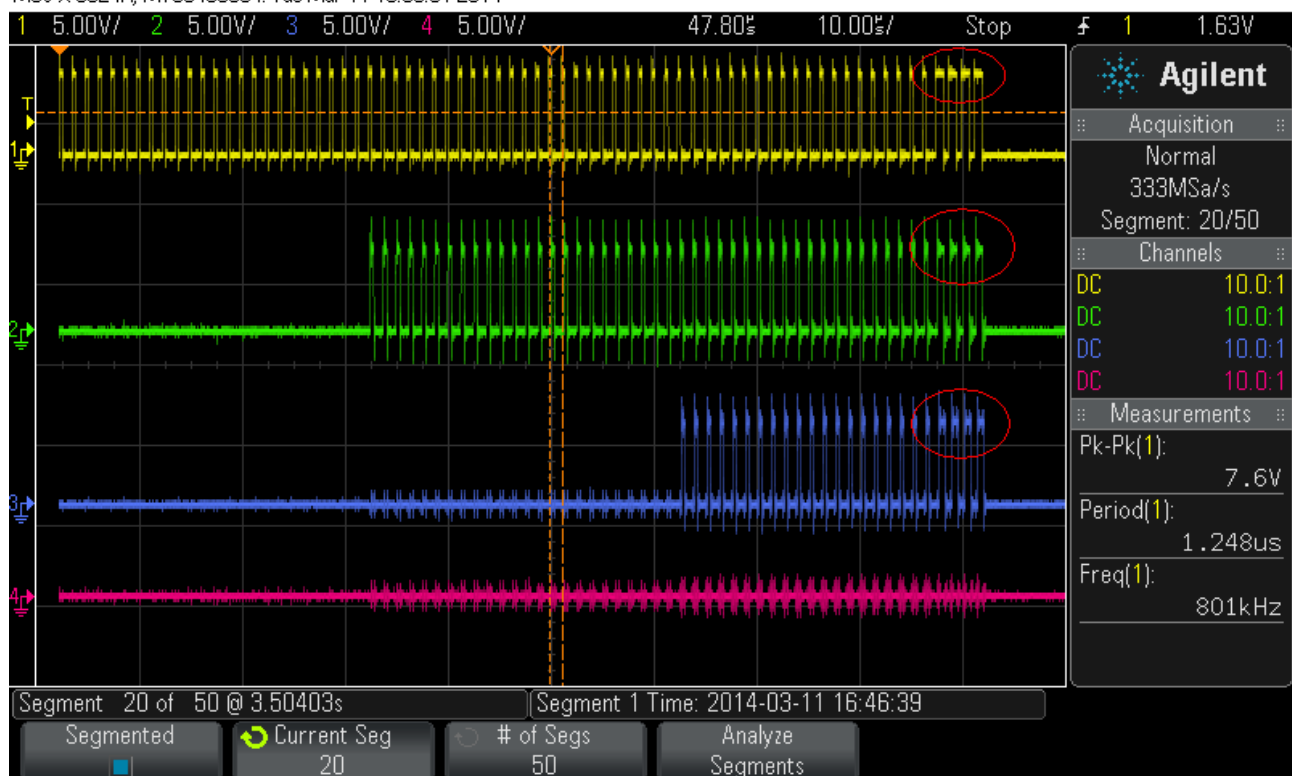


Op het volgende scoopbeeld zien we dat de 2e LED groen brand en op het 3e scoopbeeld dat de 3e led blauw brand.

MSO-X 3024A, MY53480364: Tue Mar 11 16:58:20 2014



MSO-X 3024A, MY53480364: Tue Mar 11 16:58:51 2014

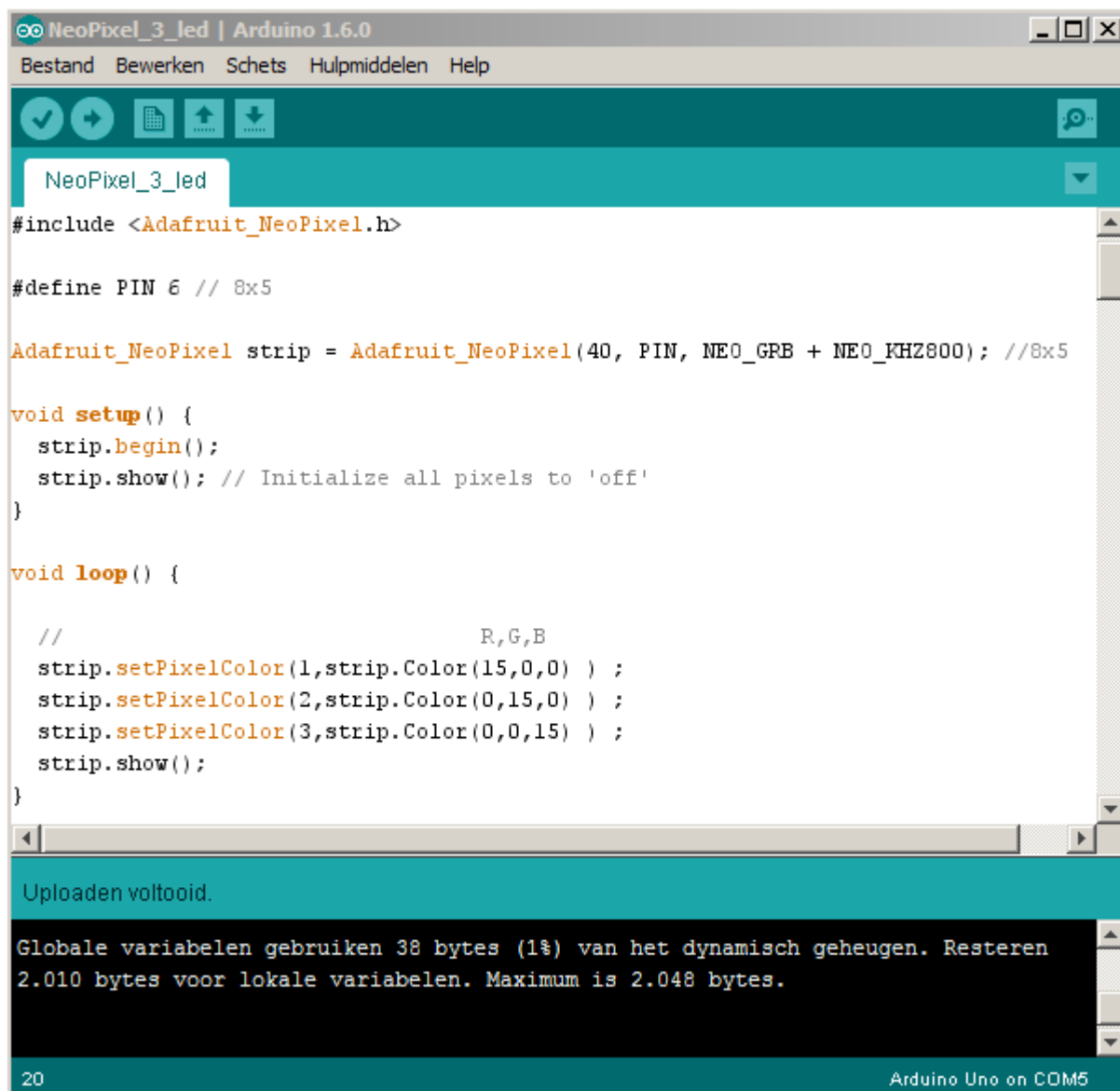


Bij binnenkomst van data voor eigen gebruik worden de 3 interne LED's direct geupdate. In elke NeoPixel wordt elke afzonderlijke LED door middel van een 8 bit's PWM aangestuurd.

De aansturing van de NeoPixel is met behulp van de benodigde library's vrij eenvoudig. Een kort programma voor de Arduino om de NeoPixel shield aan te sturen verduidelijkt veel.

Eerst zeggen we dat we de Adafruit_NeoPixel.h library gaan gebruiken, dan zeggen we op welke pin we de NeoPixel-strip aan sluiten. We definiëren de strip. Zo zijn er grotere en kleinere NeoPixel-strip's en ook de snelheid kan verschillen. Er zijn er die werken op 400 kHz, maar de door ons gebruikte werkt op 800 kHz.

Als laatste in de void loop() hoofdlus zeggen we welke waarde er op welke NeoPixel gezet moet worden en laten we dit uiteindelijk zien door middel van het commande strip.show()



```
NeoPixel_3_led | Arduino 1.6.0
Bestand  Bewerken  Schets  Hulpmiddelen  Help

NeoPixel_3_led
#include <Adafruit_NeoPixel.h>

#define PIN 6 // 8x5

Adafruit_NeoPixel strip = Adafruit_NeoPixel(40, PIN, NEO_GRB + NEO_KHZ800); //8x5

void setup() {
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'
}

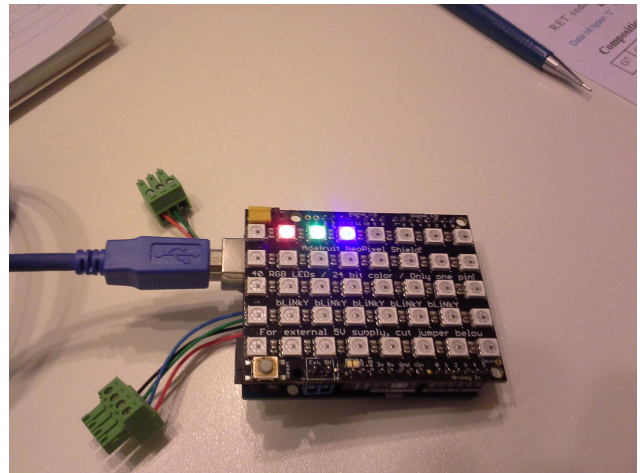
void loop() {

  //                                R,G,B
  strip.setPixelColor(1,strip.Color(15,0,0) );
  strip.setPixelColor(2,strip.Color(0,15,0) );
  strip.setPixelColor(3,strip.Color(0,0,15) );
  strip.show();
}

Uploaden voltooid.
Globale variabelen gebruiken 38 bytes (1%) van het dynamisch geheugen. Resteren
2.010 bytes voor lokale variabelen. Maximum is 2.048 bytes.

20 Arduino Uno on COM5
```

Het bovenstaande voorbeeld programma zien we hieronder in werking. Omdat we beginnen te tellen bij led-0 zien we de daaropvolgende drie LED's in de door ons gekozen kleur -Rood, Groen, Blauw-, met een lage intensiteit branden.



Het "nadeel" van de NeoPixel WS2812 is dat er bij een onderbreking van de datastroom er een reset van de NeoPixel's plaatsvindt waardoor de NeoPixel string weer vooraan begint. Dit kan een knipperend en storend effect zijn. Dus controllers zoals de Raspberry-Pi zijn minder geschikt voor de NeoPixel.

Adafruit_NeoPixel-master lib.:

<https://learn.adafruit.com/neopixels-and-servos/the-ticoservo-library>

https://github.com/adafruit/Adafruit_NeoPixel/archive/master.zip

Datasheet WS2812:

<https://www.adafruit.com/datasheets/WS2812.pdf>

****Let it be****

*"When the broken hearted people living in the world agree,
there will be an answer, let it be."*

Paul McCartney, 1969

Ronald Lokker
Abraham Vreugdenhil