# Opdracht 4.1

Led There Be Light

Deze opdracht bestaat uit drie kleinere opdrachten:

1. **Creëer een ‘Color changing light’ met het RGB-led op basis van PWM. De PWM moet in een Interrupt Service Routine (ISR) worden afgehandeld.**
2. Maak een programma waarin je via Serial communicatie een led aan of uit kan zetten. Dit mag door bijvoorbeeld het karakter ‘a’ of ‘u’ te sturen. Het programma geeft feedback over de seriële poort of het gelukt is ja of nee. Gebruik niet de gegeven usart.c en .h in je uitwerking, maar maak een oplossing in 1 source bestand.
3. Verander deelopdracht 2 zodat binnenkomende data afgehandeld wordt in een interrupt (USART\_RX\_vect) zodat je programma (zoals in het voorbeeld van hoofdstuk 5) niet ‘eeuwig’ op input wacht. Toon aan dat het werkt door via seriële communicatie een ‘r’, ‘g’ of ‘b’ door te geven aan de AVR en dat dan de corresponderende kleur gaat pulseren (PWM op 1 kleur).

## Aanpak en Uitvoering

Tijdens de les heb ik geleerd hoe je timers maakt een PWM waarde kan sturen naar een bepaalde pin. Met deze code werd er een led steeds gedimd. Deze code kon ik mooi gebruiken voor deze opdracht.

### Common Cathode of Common Anode

Eerst moest er uitgezocht worden wat voor soort RGB led het is. Het kan namelijk een **Common Cathode** of een **Common Anode** led zijn. Ik sloot pin 2 aan op 5v5 op de Arduino en pin 1 via een weerstand op de *ground* van de Arduino. De led ging meteen rood branden. Hierdoor wist ik dat het een **Common Anode** led is. De pinout is te vinden in het hoofdstuk over de datasheets.

### Code uit boek

In de les hadden we de PWM OC2A gebruikt om de led te laten dimmen. Om een RGB led van kleur te laten veranderen zijn er 3 PWM pinnen nodig om de rode, groene en blauwe waardes aan te passen. Ik wist niet precies de instellingen die nodig waren om dit te regelen voor PWM OC1A en OC1B. Gelukkig was er een hoofdstuk in het boek hieraan besteed. De code uit het boek heb ik gebruikt om groen en blauw van de RGB led aan te passen.

### Kleuren patroon

Het patroon dat ik gemaakt heb is het volgende. De led begint vol rood (255). Daarna gaat rood aftellen van 255 naar 0. Tegelijkertijd telt groen op naar 255. Dit zorgt ervoor dat de led overvloeid van rood naar groen. Als uiteindelijk groen op 255 is gaat groen weer aftellen naar 0. Als groen begint met aftellen begint de waarde van blauw op te tellen van 0 naar 255. Dit zorgt ervoor dat de led overvloeid van groen naar blauw. Als blauw eenmaal op 255 staat gaat deze waarde weer aftellen naar 0. Tergelijkertijd begint rood op te tellen van 0 naar 255. Dit zorgt ervoor dat de led overvloeid van blauw naar rood. Als dit gebeurd is zijn we weer aangekomen bij het begin. Deze code staat in de Interrupt Service Routine.

## Afbeelding



Opdracht 4.1 - Led There Be Light - Afbeelding

De afbeelding van de setup kan ook gedownload worden via de volgende link:

<https://raw.githubusercontent.com/maartenpaauw/IMTHE1/master/O4_1/assets/setup.jpg>

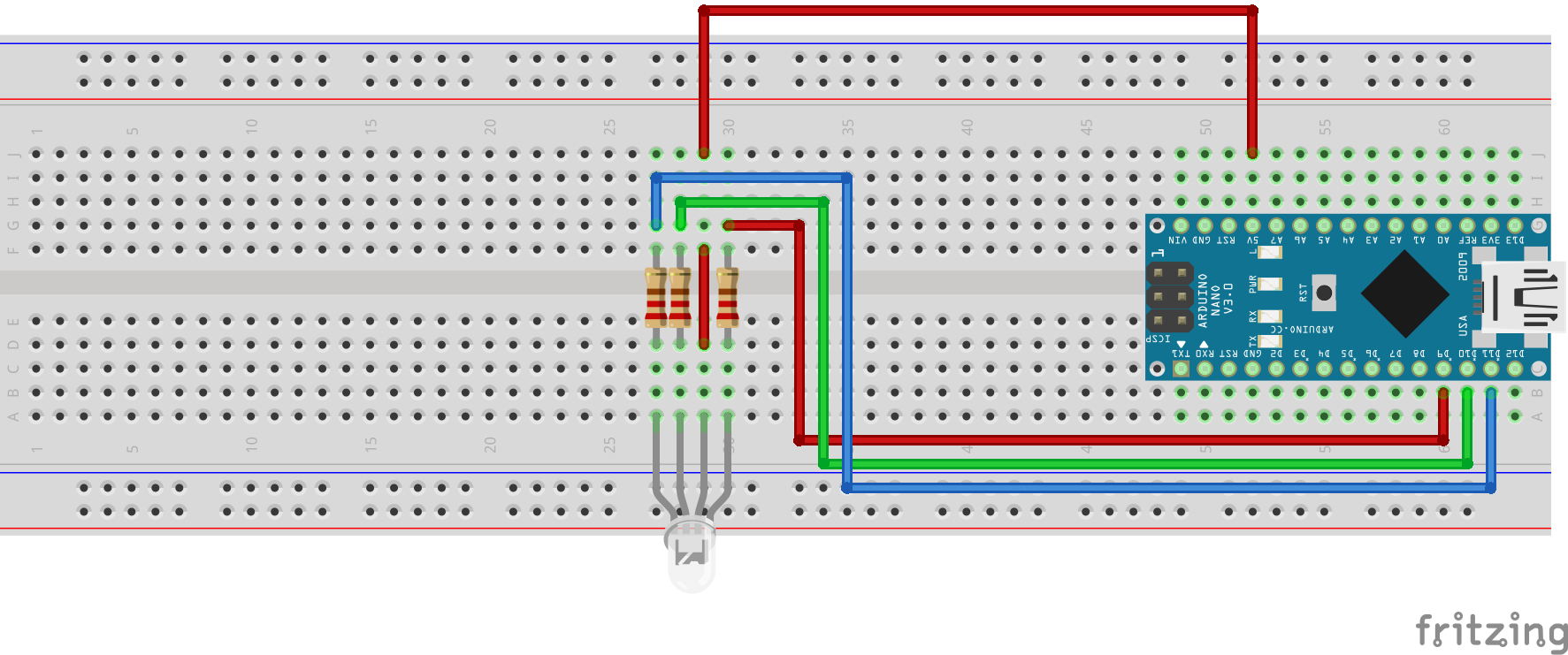
## Video

[](https://www.youtube.com/watch?v=dClmJYRgYDM)

Deze video is ook te vinden op **Youtube**:

<https://www.youtube.com/watch?v=dClmJYRgYDM>

## Breadboard Schema

[](https://raw.githubusercontent.com/maartenpaauw/IMTHE1/master/O4_1/assets/fritzing/schema.png)

Het **Fritzing** schema kan ook gedownload worden via de volgende link:

<https://github.com/maartenpaauw/IMTHE1/raw/master/O4_1/assets/fritzing/schema.fzz>

### Hardware

|  |
| --- |
| Onderdelen |
| Arduino Nano (v3.0) - 1× |
| RGB LED - 1× |
| 220Ω Resistor - 3× |
| Dupont kabel - 5× |

## Code

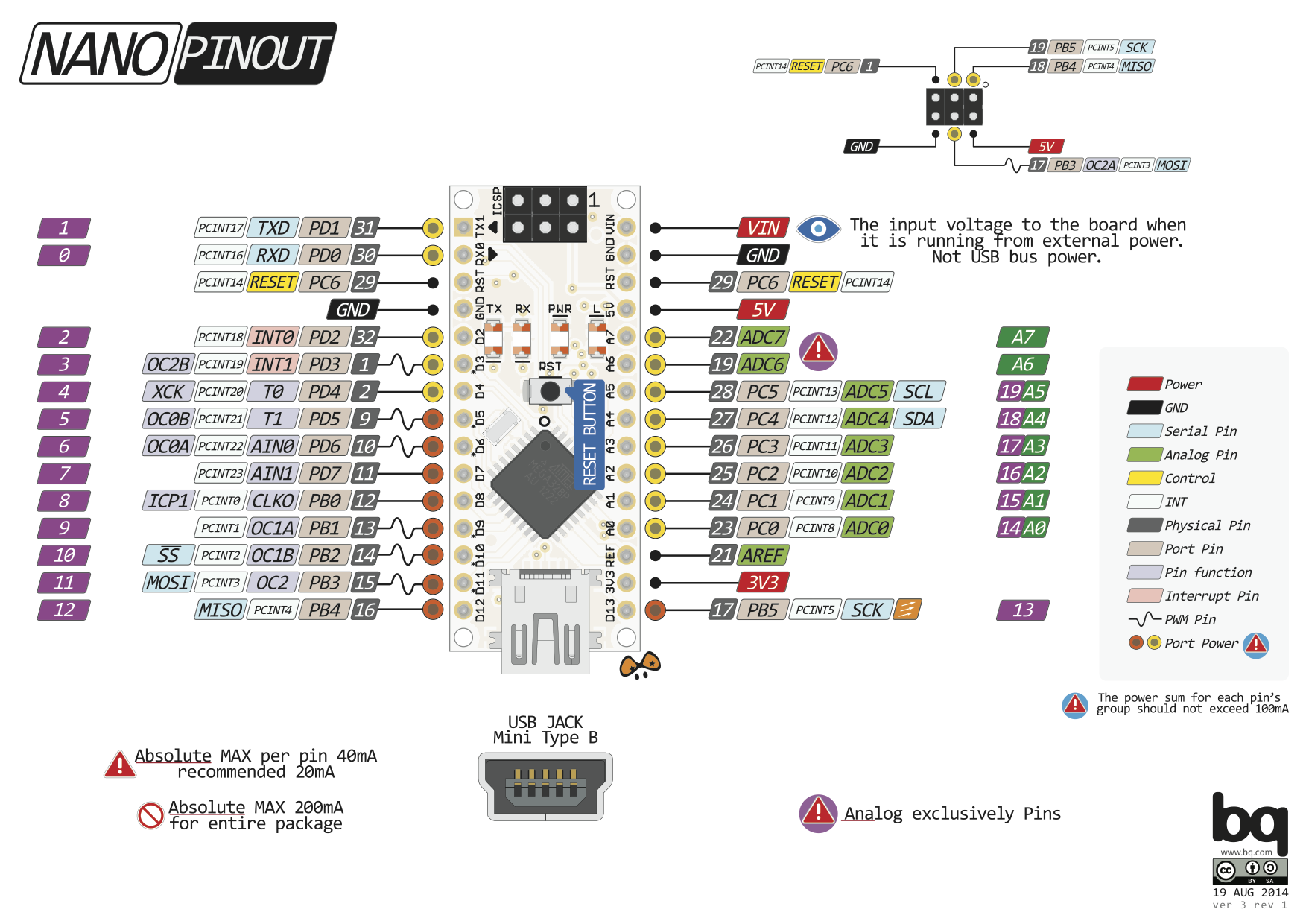
/\*  
 \* Opdracht 4.1 - Led There Be Light  
 \*   
 \* Eigenaar:  
 \* Maarten Paauw <s1094220@student.hsleiden.nl>  
 \* s1094220  
 \* INF3C  
 \*   
 \* Versie: 1  
 \* Aangemaakt: 12 maart 2018  
 \* Gewijzigd: 27 maart 2018  
 \*/  
  
#include <avr/io.h>  
#include <avr/interrupt.h>  
  
// LED pinnen.  
#define LED\_RED PB1  
#define LED\_GREEN PB2  
#define LED\_BLUE PB3  
  
// RGB waardes.  
int RGB[3] = {255, 0, 0};  
  
// Count.  
int count = 0;  
  
// Bereken degene die opgeteld moet worden.  
int increment(int count)  
{  
 // Bereken degene die opgeteld moet worden.  
 return (count + 1) % 3;  
}  
  
// Bereken degene die afgetrokken moet worden.  
int decrement(int count)  
{  
 // Bereken degene die afgetrokken moet worden.  
 return count % 3;  
}  
  
// Timer interrupt.  
ISR(TIMER0\_OVF\_vect)  
{  
 // Deze variabele hebben we twee keer nodig.  
 int dec = decrement(count);  
  
 // Trek R, G of B af.  
 RGB[dec] -= 1;  
  
 // Tel R, G of B af.  
 RGB[increment(count)] += 1;  
  
 // Als de waarde van R, G of B helemaal terug is op 0.  
 if (RGB[dec] <= 0)  
 {  
 // Tel op.  
 count++;  
 }  
  
 // Is de waarde niet gelijk of kleiner dan 0.  
 else  
 {  
 OCR1A = ~RGB[0]; // Rode led waarde.  
 OCR1B = ~RGB[1]; // Groene led waarde.  
 OCR2A = ~RGB[2]; // Blauwe led waarde.  
 }  
}  
  
// Overflow timers.  
void initTimerOverflow()  
{  
 // Timer mask.  
 TIMSK0 |= (1 << TOIE0);  
  
 // Timer instellingen.  
 TCCR0B |= (1 << CS02) | (1 << CS00);  
  
 // Start de interrupt.  
 sei();  
}  
  
// LED timers.  
void initPWMTimerLED()  
{  
 // Timer 1 (Voor rood en groen).  
 TCCR1A |= (1 << WGM10) | (1 << WGM12); // Fast PWM.  
 TCCR1B |= (1 << CS11); // PWM freq.  
  
 // Timer 2 (Voor blauw).  
 TCCR2A |= (1 << WGM20) | (1 << WGM21); // Fast PWM.  
 TCCR2B |= (1 << CS21); // PWM freq.  
  
 // Rode en groene led.  
 TCCR1A |= (1 << COM1A1) | (1 << COM1B1);  
  
 // Blauwe led.  
 TCCR2A |= (1 << COM2A1);  
}  
  
// Main functie.  
int main(void)  
{  
  
 // Initialiseer de RGB LED pinnen.  
 DDRB = (1 << LED\_RED) | (1 << LED\_GREEN) | (1 << LED\_BLUE);  
  
 // Initialiseer de overflow timer.  
 initTimerOverflow();  
  
 // Initialiseer de led timer.  
 initPWMTimerLED();  
  
 // Loop voor altijd.  
 while (1) { }  
  
 // Geef 0 terug.  
 return 0;  
}

De code kan ook gevonden worden in mijn **GitHub** *repository* via de volgende link:

<https://github.com/maartenpaauw/IMTHE1/blob/master/O4_1/src/main.c>

## Datasheets

### Arduino Nano

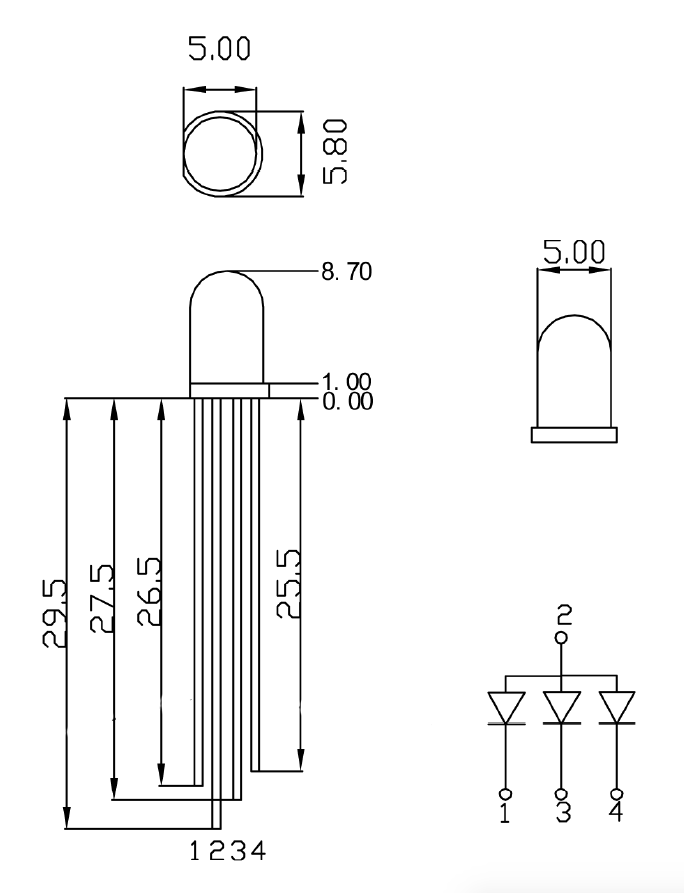


Arduino Nano Pinout

De [Arduino Nano Pinout](https://raw.githubusercontent.com/maartenpaauw/IMTHE1/master/O4_1/assets/data_sheets/nano.png) is uitgedeeld tijdens de eerste les en terug te vinden via het volgende forum post:

<https://forum.arduino.cc/index.php?topic=147582.0>

### RGB Led



RGB led Pinout

De [RGB Led Pinout](https://raw.githubusercontent.com/maartenpaauw/IMTHE1/master/O4_1/assets/data_sheets/rgb_led.png) heb ik gevonden in de datasheet van een common cathode led. De datasheet is terug te vinden via de volgende link:

<https://www.wiltronics.com.au/wp-content/uploads/datasheets/Datasheet.pdf>

## Bronvermelding

* <https://forum.arduino.cc/index.php?topic=147582.0> (Arduino Nano Pinout)
* <https://www.wiltronics.com.au/wp-content/uploads/datasheets/Datasheet.pdf> (RGB Led Datasheet)