

Project: kleurensensor

(C) Richèl Bilderbeek



August 25, 2014

1 Introductie

Een lichtsensor¹ kan geen kleur herkennen. Met behulp van een RGB LED kan dit wel.

Dit project bestaat uit meerdere onderdelen:

- Aansluiten RGB LED
- Aansluiten lichtsensor
- LED laten reageren op sensor
- Omprogrammeren naar kleurensensor

2 Aansluiten RGB LED

Als je naief bent, sluit je een RGB LED aan als figuur 1.

- Wat is onjuist aan figuur 1? Wat zou er gebeuren als je dit zo aansluit?

Als je slim bent, sluit je een RGB LED aan als figuur 2.

¹ook wel 'licht-afhankelijk weerstand' of 'LDR' ('Light Dependent Resistance') genoemd



Figure 1: Aansluiten RGB LED op een naieve manier

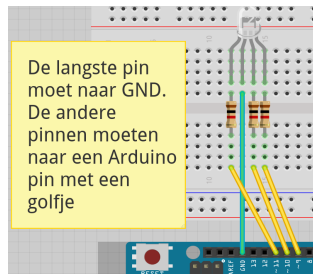


Figure 2: Aansluiten RGB LED op een slimme manier

Om een kleur van een RGB lampje aan te sturen kun je gebruik maken van het voorbeeld Fade².

Het laten schijnen van een LEDje gebeurt met de functie 'analogWrite', bijvoorbeeld als volgt:

```
analogWrite(9, 127);
```

Hierin wordt naar pin 9 de waarde 127 gestuurd, waardoor de LED op halve sterkte gaat branden.

- Sluit een RGB LED aan. Laat deze elke seconde een andere kleur geven, in de volgorde rood, geel (=rood + groen), groen, cyaan (=groen + blauw), blauw, magenta (=rood+blauw)

Algoritme 1 is een mogelijke oplossing.

3 Aansluiten lichtsensor

Als je naïef bent, sluit je een lichtsensor aan als in figuur 3. Om de waarde van een lichtsensor te lezen, kun je het voorbeeldprogramma AnalogReadSerial³ gebruiken.

Het lezen van een sensor gebeurt met 'analogRead', bijvoorbeeld als volgt:

```
int mijn_sensor_waarde = analogRead(A0);
```

Hierin wordt de waarde van pin A0 gelezen en opgeslagen in de variabele 'mijn_sensor_waarde'.

- Kun je niet voorspellen welke waarde je gaat meten? Bouw dit schema en noteer het resultaat.

Als je weet dat figuur 3 niet werkt, bouw je figuur 4.

- Schrijf op welke waarde je meet in het donker en in het volle licht

²File -> Examples -> 01. Basics -> Fade

³File -> Examples -> 01. Basics -> AnalogReadSerial

Algorithm 1 RGB LED voorbeeld code

```
const int pin_rood = 9;
const int pin_groen = 10;
const int pin_blaauw = 11;

void setup() {
  pinMode(pin_rood, OUTPUT);
  pinMode(pin_groen, OUTPUT);
  pinMode(pin_blaauw, OUTPUT);
}

void GeefKleur(
  const int roodheid,
  const int groenheid,
  const int blauwheid)
{
  analogWrite(pin_rood, r);
  analogWrite(pin_groen, g);
  analogWrite(pin_blaauw, b);
}

void loop() {
  GeefKleur(255, 0, 0); delay(500);
  GeefKleur(255, 255, 0); delay(500);
  GeefKleur(0, 255, 0); delay(500);
  GeefKleur(0, 255, 255); delay(500);
  GeefKleur(0, 0, 255); delay(500);
  GeefKleur(255, 0, 255); delay(500);
}
```



Figure 3: Lichtsensor naief aangesloten

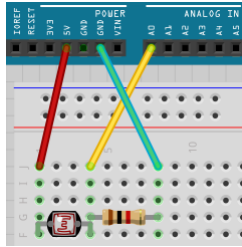


Figure 4: Lichtsensor juist aangesloten

4 LED laten reageren op sensor

Stel je voor, de sensor meet waarden van 400 voor donker en 600 voor licht. Je weet dat een LED als waarde maximaal 255 kan krijgen. Een naïeve manier om de LED aan te sturen is als volgt

```
const int gemeten_waarde = analogRead(sensor_pin);
analogWrite(led_pin, gemeten_waarde / 3);
```

- Wat is hiervan het nadeel? Tip: gaat het LEDje ooit volledig branden; gaat het LEDje ooit volledig uit?

Een slimme manier is de 'map' functie te gebruiken:

```
const int gemeten_waarde = analogRead(sensor_pin);
const int waarde_voor_led = map(gemeten_waarde, 400, 600, 0, 255);
analogWrite(led_pin, waarde_voor_led);
```

- Laat de RGB LED reageren op de gemeten sensorwaarde. Laat deze bijvoorbeeld wit schijnen als er veel licht is.

5 Omprogrammeren naar kleursensor

Laat de LED om de beurt rood, groen en blauw branden. Laat de lichtsensoren om de beurt de rood, groen en blauwwaarden meten van het ding waarop je schijnt. Laat dan de LED de kleur branden van het ding.