



Figure 1: Boek 1: LED

Contents

Voorwoord	1
Les 1a: Blink	2
Les 1b: Blink Blink Blink	10
Les 1c: Meer Kleur Fade Met For	16
Les 1d: For nog een keer	21

Voorwoord



Figure 1: Het logo van De Jonge Onderzoekers

Dit is het boek van de Arduino cursus. Een Arduino is een machine die je kunt programmeren. Dit boek leert je hoe je electronica op de Arduino aansluit, en hoe je deze programmeert.

Over dit boek

Dit boek heeft een CC-BY-NC-SA licentie.



Figure 2: De licentie van dit boek

(C) Arduino cursus Groningen 2017

Het is nog een beetje een slordig boek. Er zitten tiepvauten in en de opmaak is **niet altijd even mooi**.

Daarom staat dit boek op een GitHub. Om precies te zijn, op <https://github.com/richelbilderbeek/ArduinoCourse> . Hierdoor kan iedereen die dit boek te slordig vindt minder slordig maken.

Les 1a: Blink

In deze les gaan we de allergemakkelijkste schakeling maken die er is: Blink!

Een Arduino aansluiten

Sluit een Arduino zo aan:

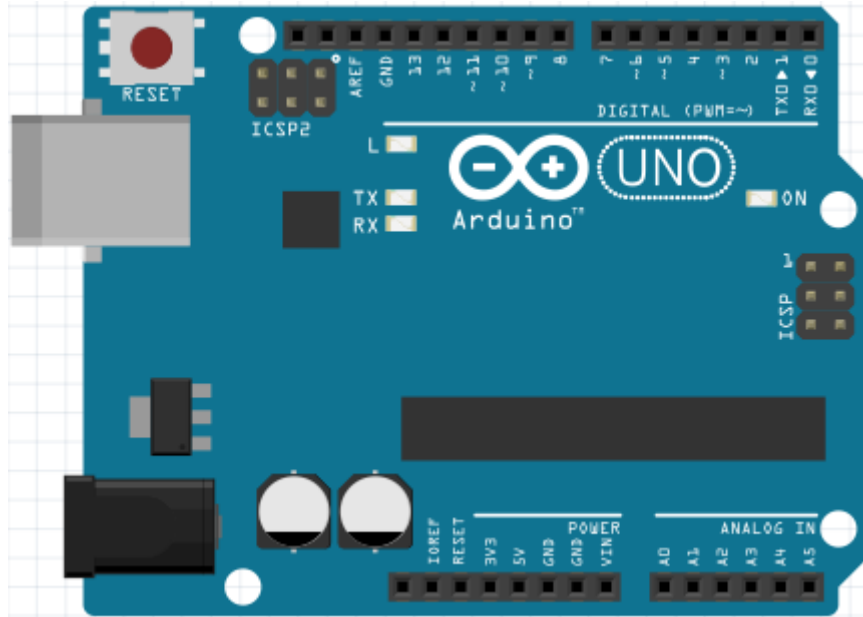


Figure 3: Blink zonder led

Je zult wel een USB kabel in de Arduino en in een laptop moeten doen

Opdracht

Sluit de Arduino aan.



De Arduino heeft zelf al een lampje dat je kunt programmeren.

De Arduino IDE opstarten

Om een Arduino te programmeren hebben we een programma nodig. Dit programma noemen we 'de Arduino IDE' (IDE spreek je uit als 'ie-dee-ee')

De Arduino IDE staat op alle laptops van de cursus. Hier zie je het logo van de Arduino IDE:

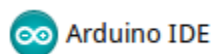


Figure 4: Logo van de Arduino IDE

Je kunt de Arduino IDE ook starten, door:

- Druk op de Windows toets (linksonder, tussen Ctrl en Alt)

- Type **arduino** (kleine letters) en dan Enter

Als het goed is, zie je de Arduino IDE:



IDE betekent 'Integrated Development Environment', het programma waarmee je programmeert.

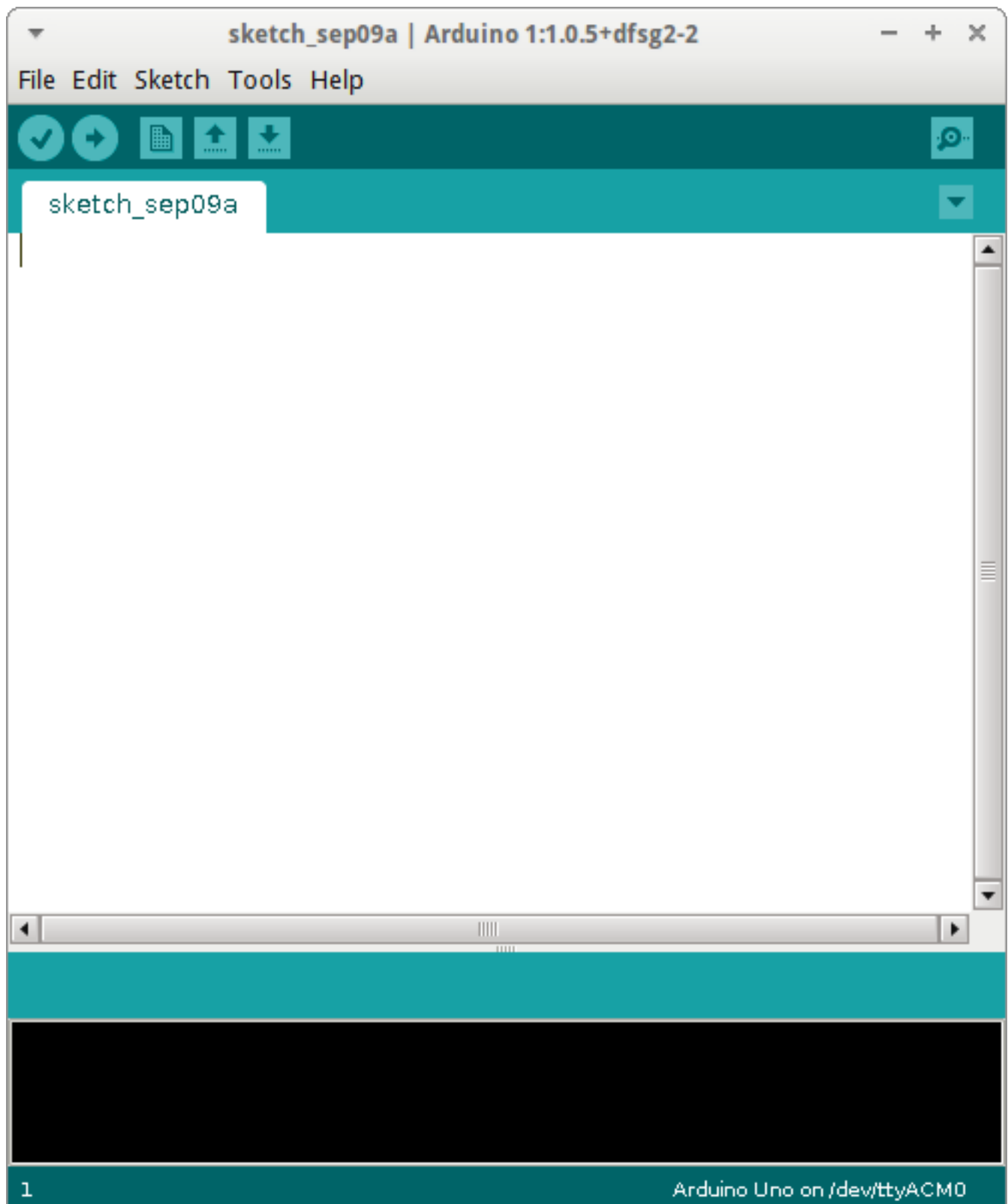


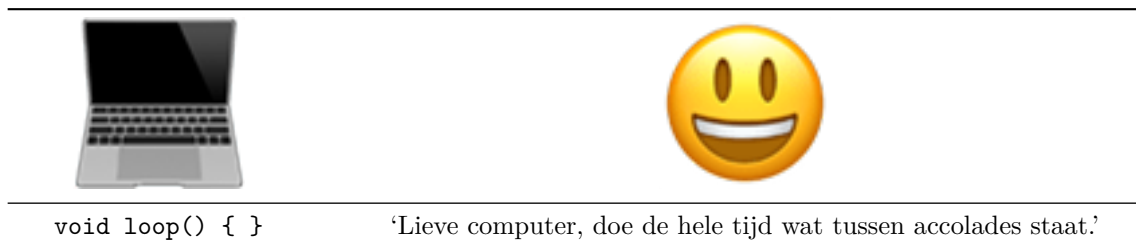
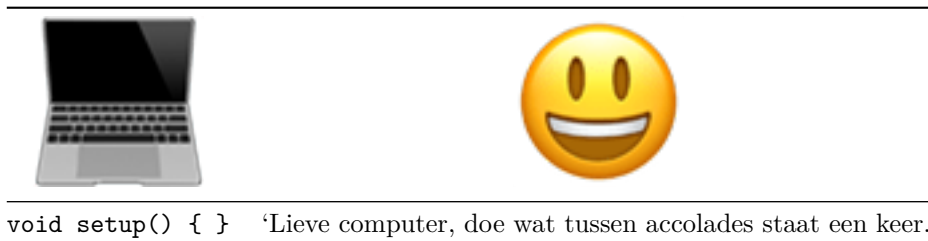
Figure 5: De Arduino IDE

Ons eerste programma

Dit is (een versie van) de code van **Blink**:

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```



Type de code over in de Arduino IDE en klik op **Upload**.

Opdracht

Laat het LEDje 10x zo snel knipperen. Hint: het LEDje is nu 1000 milliseconden aan en 1000 milliseconden uit.

Oplossing

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(100);
  digitalWrite(13, LOW);
  delay(100);
}
```



Figure 6: Hier kun je op Upload klikken



`delay(1000);` 'Lieve computer, doe nu 1000 milliseconden niks.'

Opdracht

Laat het LEDje zo snel als kan knipperen. Wat zie je?

Oplossing

Er zijn meer oplossingen, dit is er een:

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1);
  digitalWrite(13, LOW);
  delay(1);
}
```

Ook goed zijn `delay(0);` of de regels met `delay` helemaal weghalen.

Je ziet het lampje niet meer knipperen.



Het lampje knippert wel, maar onze ogen kunnen niet snel genoeg kijken om het te zien.

Blink

Nu is het tijd Blink aan te sluiten:

- Haal de USB snoer uit de computer, zodat de Arduino geen spanning meer heeft
- Sluit Blink aan zoals op de tekening
- Zet de spanning weer op de Arduino
- Als het goed is, knippert het rode LEDje nu mee met het LEDje op de Arduino.



Als het LEDje niet brandt, draai deze dan om.

Opdracht

Sluit je LEDje nu aan op pin 12.

Oplossing

Opdracht

Zorg nu dat het LEDje op pin 12 gaat knipperen op de seconde.

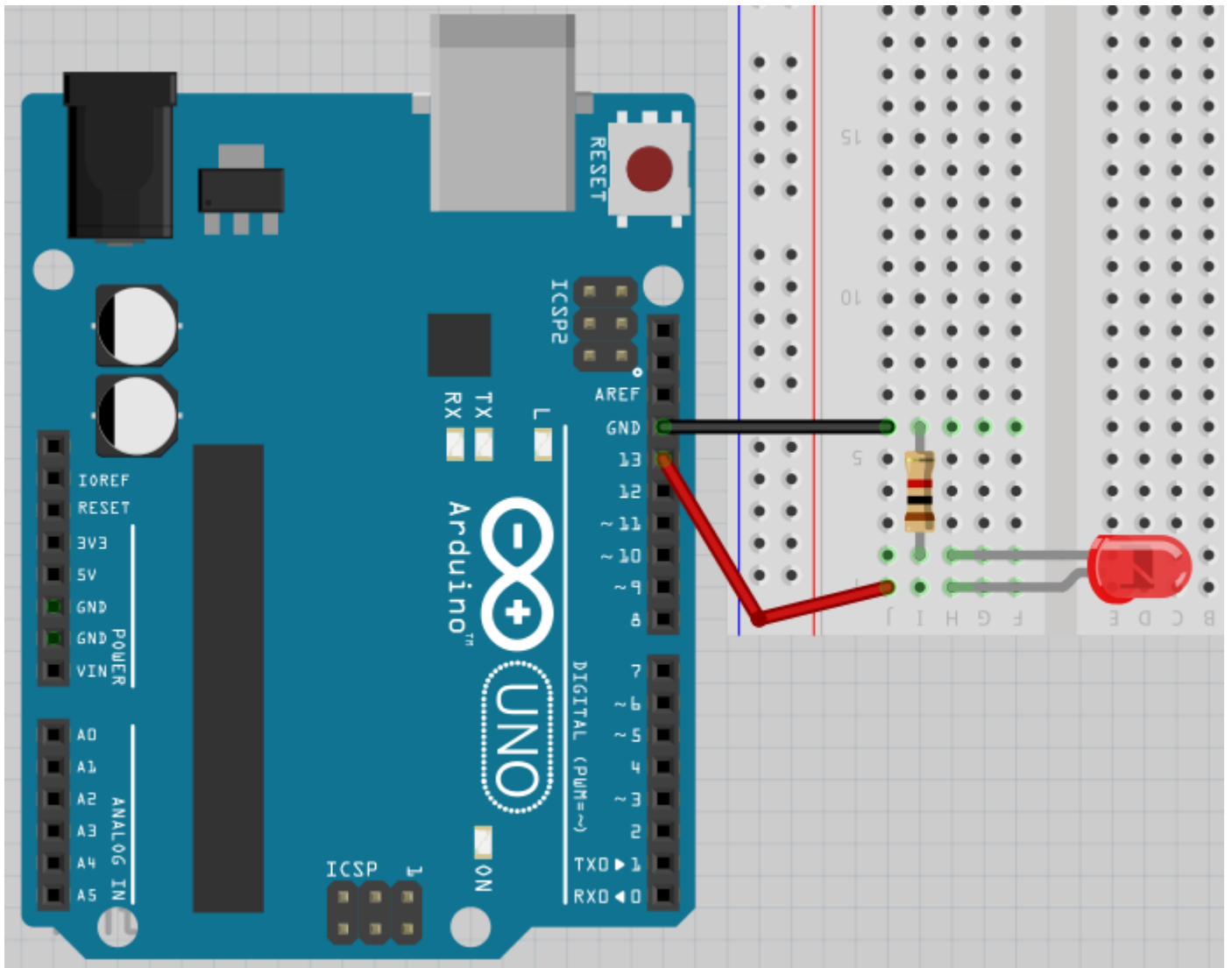


Figure 7: Blink

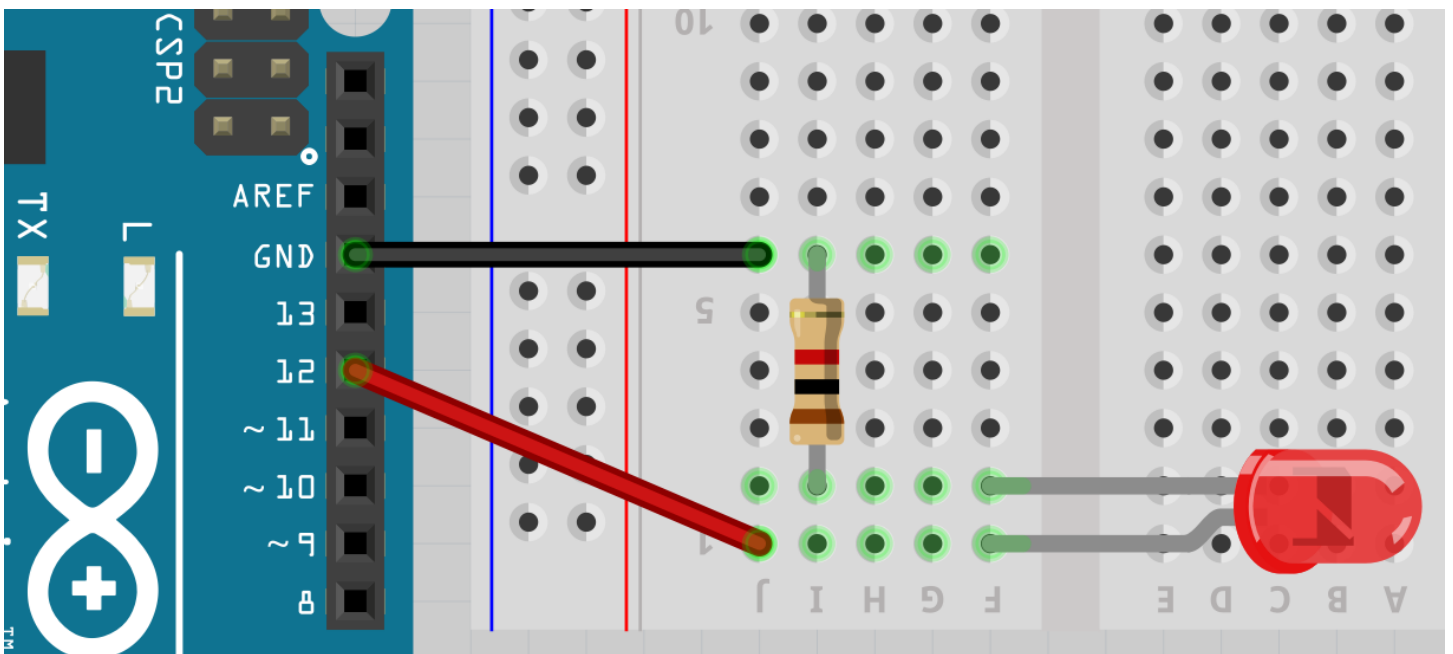


Figure 8: Blink met LEDje op pin 12

Oplossing

```
void setup()
{
  pinMode(12, OUTPUT);
}

void loop()
{
  digitalWrite(12, HIGH);
  delay(100);
  digitalWrite(12, LOW);
  delay(100);
}
```



```
pinMode(13, OUTPUT);
digitalWrite(13,
  HIGH);
```

‘Lieve computer, zorg dat er stroom uit pin 13 kan komen.’
‘Lieve computer, zet spanning op pin 13.’

Eindopdracht

Sluit de LED aan op pin 11 en laat deze 2 keer per seconde knipperen.



Gelukt? Laat dit zien aan een volwassene voor een sticker!

Les 1b: Blink Blink Blink

Deze les heet 'Blink Blink Blink', omdat het de les Blink is, maar dan met drie lampjes.

Blink

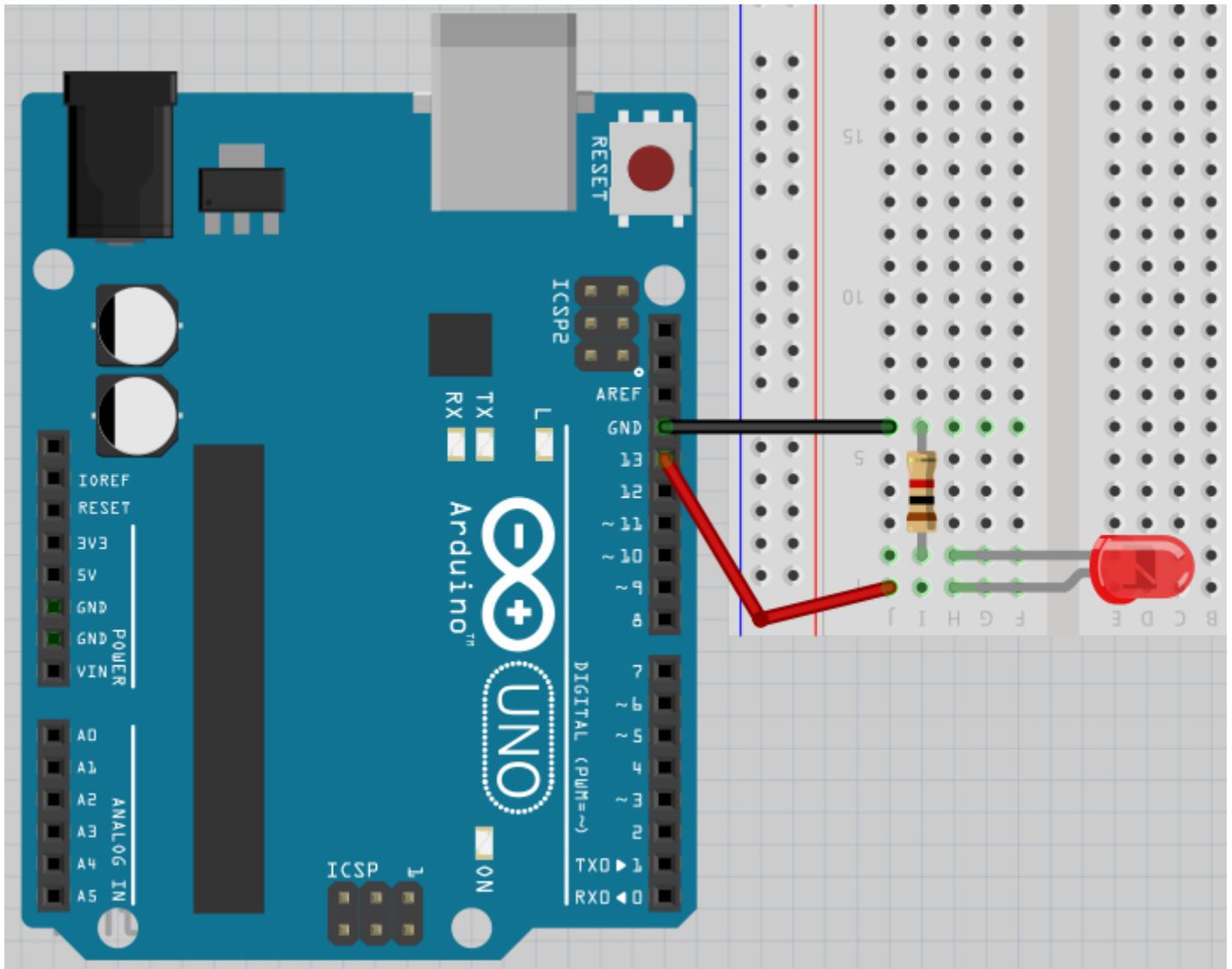


Figure 9: Blink

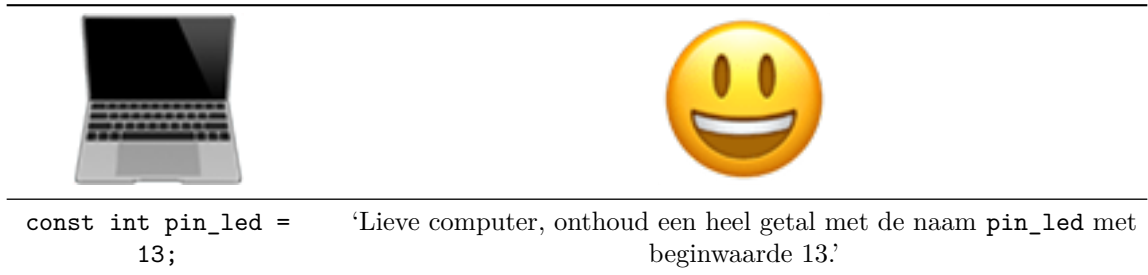
Dit is de code van Blink op een andere manier:

```
const int pin_led = 13;

void setup()
{
  pinMode(pin_led, OUTPUT);
}

void loop()
{
  digitalWrite(pin_led, HIGH);
  delay(1000);
  digitalWrite(pin_led, LOW);
  delay(1000);
}
```

pin_led wordt een variabele genoemd: een stukje computergeheugen met een naam.



Opdracht

Sluit de LED aan op pin 12 en verander de code zodat deze gaat knipperen.

Oplossing

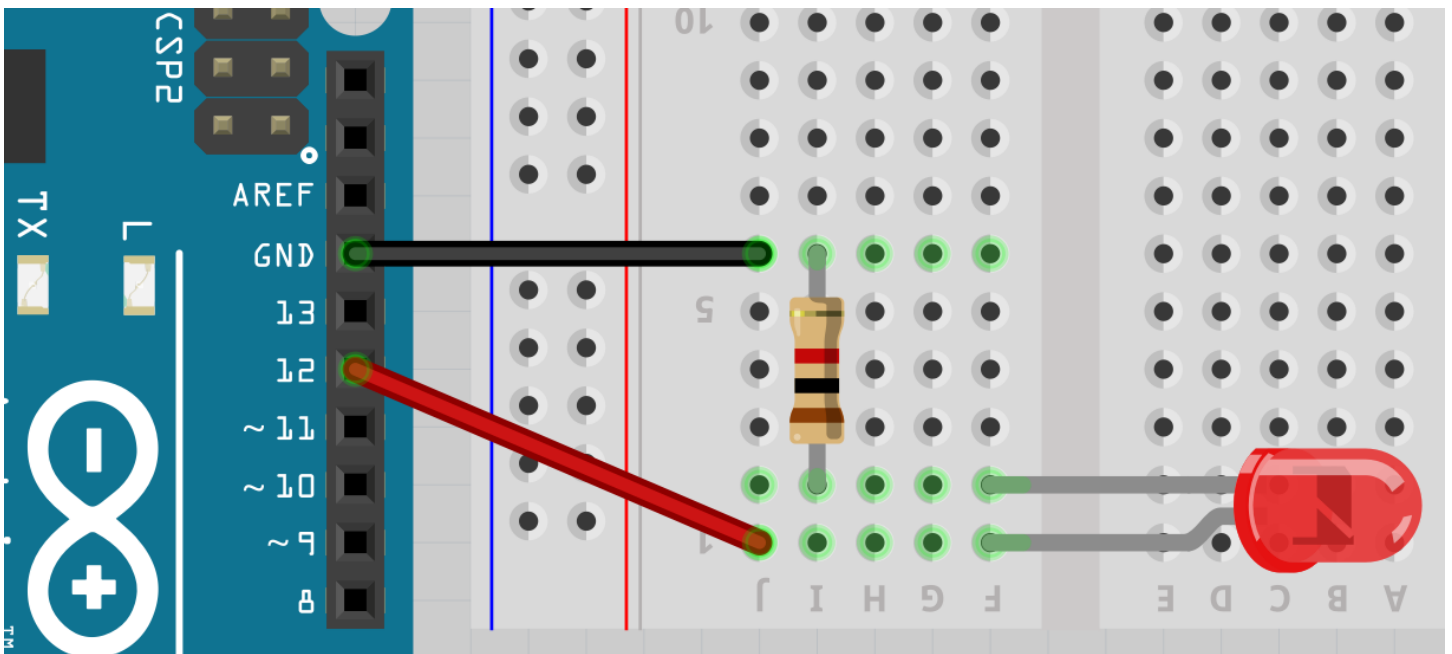


Figure 10: Blink met LEDje op 12

```
const int pin_led = 12;

void setup()
{
  pinMode(pin_led, OUTPUT);
}

void loop()
{
  digitalWrite(pin_led, HIGH);
  delay(1000);
  digitalWrite(pin_led, LOW);
  delay(1000);
}
```

Opdracht

Maak zelf een nieuwe variabele met de naam `wachttijd`. `wachttijd` is een heel getal met beginwaarde 1000. Gebruik `wachttijd` in de regels met `delay`.

Oplossing

```
const int pin_led = 12;
const int wachttijd = 1000;

void setup()
{
  pinMode(pin_led, OUTPUT);
}

void loop()
{
  digitalWrite(pin_led, HIGH);
  delay(wachttijd);
  digitalWrite(pin_led, LOW);
  delay(wachttijd);
}
```



Programmeurs gebruiken veel variabelen, omdat de code dan beter te snappen is.

Blink Blink Blink aansluiten

Nu is het tijd 'Blink Blink Blink' aan te sluiten:

- Haal de USB snoer uit de computer, zodat de Arduino geen spanning meer heeft
- Sluit de onderdelen aan zoals op de tekening

Opdracht

Sluit 'Blink Blink Blink' aan. Hernoem de variable `pin_led` naar `pin_led_1` en zorg dat 'ie de juiste beginwaarde heeft.

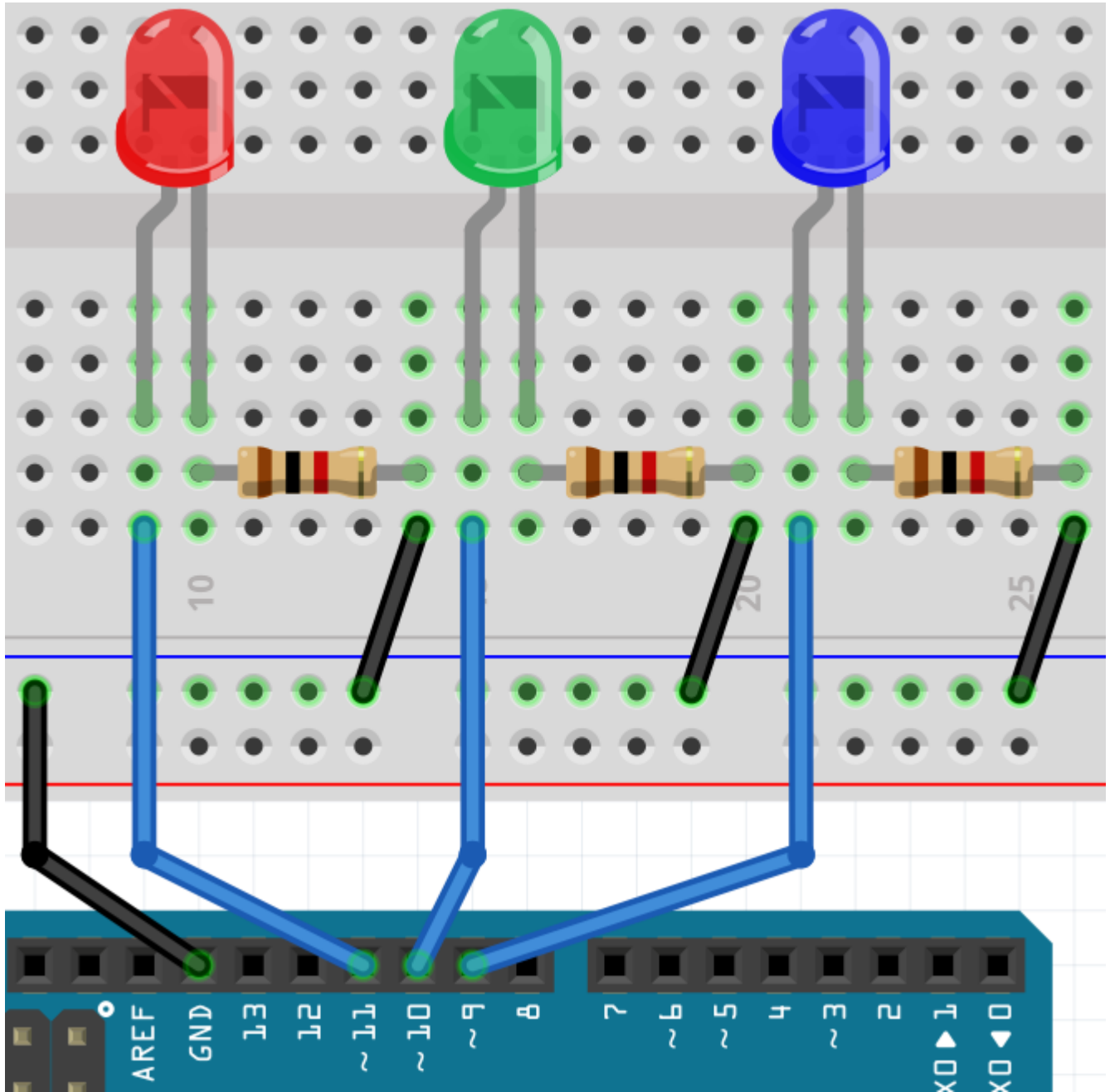


Figure 11: BlinkBlinkBlink

Oplossing

```
const int pin_led_1 = 11;
const int wachttijd = 1000;

void setup()
{
  pinMode(pin_led_1, OUTPUT);
}

void loop()
{
  digitalWrite(pin_led_1, HIGH);
  delay(wachttijd);
  digitalWrite(pin_led_1, LOW);
  delay(wachttijd);
}
```

Opdracht

Maak een nieuwe variable `pin_led_2`. Laat dan eerst het eerste LEDje aan en uit gaan, laat dan het tweede LEDje aan en uit gaan.

Oplossing

```
const int pin_led_1 = 11;
const int pin_led_2 = 10;
const int wachttijd = 1000;

void setup()
{
  pinMode(pin_led_1, OUTPUT);
  pinMode(pin_led_2, OUTPUT);
}

void loop()
{
  digitalWrite(pin_led_1, HIGH);
  delay(wachttijd);
  digitalWrite(pin_led_1, LOW);
  delay(wachttijd);
  digitalWrite(pin_led_2, HIGH);
  delay(wachttijd);
  digitalWrite(pin_led_2, LOW);
  delay(wachttijd);
}
```

Opdracht

Maak een derde variabele `pin_led_3`. Laat nu steeds alle lampjes tegelijk knipperen: allemaal aan, dan allemaal uit.

Oplossing

```
const int pin_led_1 = 11;
const int pin_led_2 = 10;
const int pin_led_3 = 9;
const int wachttijd = 1000;

void setup()
{
  pinMode(pin_led_1, OUTPUT);
  pinMode(pin_led_2, OUTPUT);
}

void loop()
{
  digitalWrite(pin_led_1, HIGH);
  digitalWrite(pin_led_2, HIGH);
  digitalWrite(pin_led_3, HIGH);
  delay(wachttijd);
  digitalWrite(pin_led_1, LOW);
  digitalWrite(pin_led_2, LOW);
  digitalWrite(pin_led_3, LOW);
  delay(wachttijd);
}
```

Eindopdracht

Laat de lampjes nu in een ‘Knightrider patroon’ gaan: 1-2-3-2. Er moet altijd precies een lampje branden.



Knightrider was een TV serie met een pratende auto.

Les 1c: Meer Kleur Fade Met For

Soms denk je ‘Kan dit slimmer?’. Een for-loop is een manier om iets slimmer te doen!

Regenboog

In deze les gaan we een regenboog programmeren!

Hier zie je hoe een regenboog werkt:

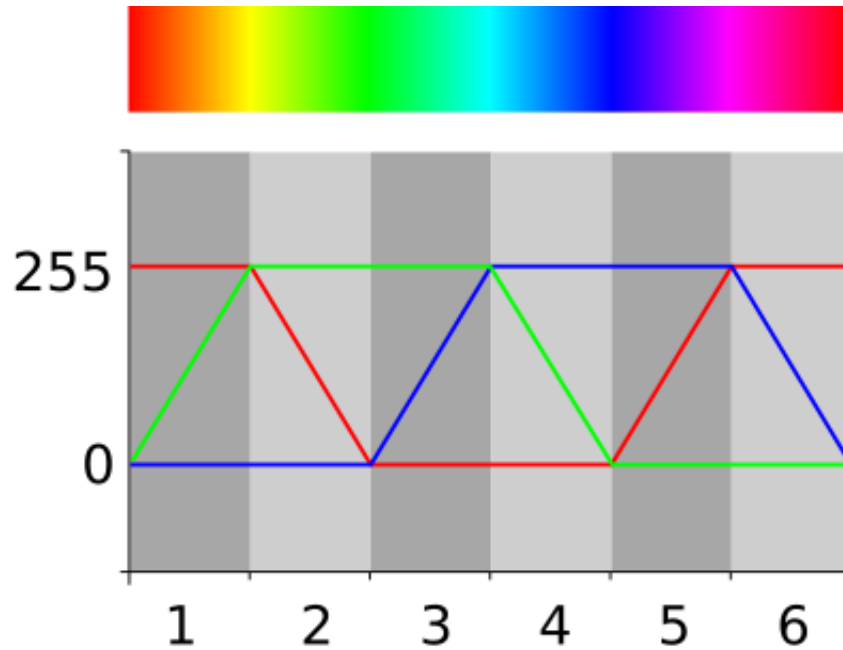


Figure 12: Regenboog



Geel licht is rood en groen licht samen.

Aansluiten

Sluit onderstaand schema aan:

Let op:

- De langste pin van een RGB LED moet naar GND
- Elke andere pin van een RGB LED moet een weerstandje krijgen van 1000 Ohm (bruin-zwart-rood-goud)
- De pinnen van de LED moeten naar een pin waar een golfje (~) voor staat, want alleen deze pinnen kunnen dimmen

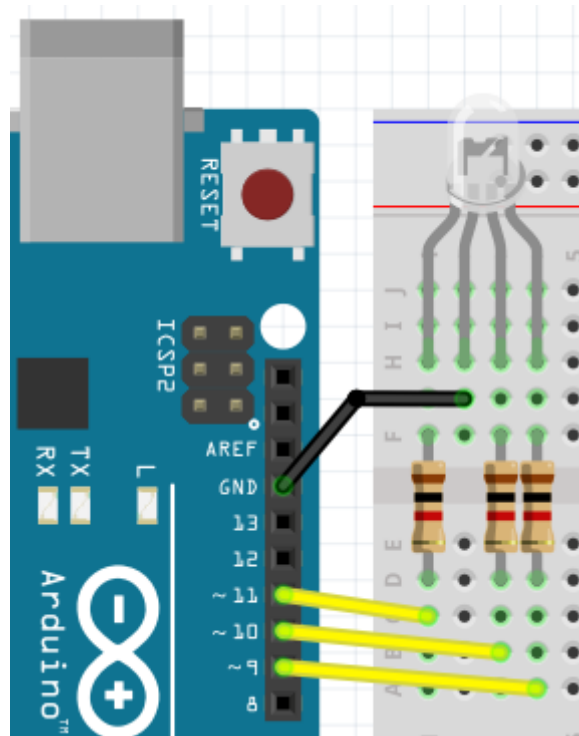


Figure 13: Schema

Opdracht

Gebruik deze 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 loop()
{
  digitalWrite(pin_rood, HIGH);
  digitalWrite(pin_groen, HIGH);
  digitalWrite(pin_blaauw, HIGH);
}
```

De getallen achter `pin_rood`, `pin_groen` en `pin_blaauw` zijn fout. Zorg dat dit de goede getallen worden. Hoe? Trek twee draden los en kijk welke kleur het LEDje krijgt.

Oplossing

Ha, dit mag je zelf uitvinden :-).

For loop

We beginnen met deze code:

```

const int pin_rood = 9; // Fout getal, zie vorige opdracht
const int pin_groen = 10; // Fout getal, zie vorige opdracht
const int pin_blaauw = 11; // Fout getal, zie vorige opdracht

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

void loop()
{
  for (int i=0; i<256; ++i)
  {
    analogWrite(pin_rood, i);
    analogWrite(pin_groen, 0);
    analogWrite(pin_blaauw, 255);
    delay(30);
  }
}

```



for (int i=0; i<256; ++i) { }	‘Lieve computer, laat een getal i lopen van 0 tot 256 in stapjes van 1.’
analogWrite(9, 255)	‘Lieve computer, zet pin 9 vol aan.’
analogWrite(9, 127)	‘Lieve computer, zet pin 9 half aan.’
analogWrite(9, 0)	‘Lieve computer, zet pin 9 uit.’



Tellen tot drie: ‘1, 2’. Tellen tot en met drie: ‘1, 2, 3’

Opdracht

Kijk naar het eerste gedeelte, links, in figuur **Regenboog**. De rode lijn staat daar op 255, de groene lijn gaat langzaam van 0 naar 255 en de blauwe lijn staat bij nul.

Pas de code zo aan dat het lampje van rood naar geel (rood en groen) gaat.

Oplossing

```
const int pin_rood   = 9; // Fout getal, zie vorige opdracht
const int pin_groen  = 10; // Fout getal, zie vorige opdracht
const int pin_blaauw = 11; // Fout getal, zie vorige opdracht

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

void loop()
{
  for (int i=0; i<256; ++i)
  {
    analogWrite(pin_rood, 255);
    analogWrite(pin_groen, i);
    analogWrite(pin_blaauw, 0);
    delay(30);
  }
}
```

Opdracht

Kijk naar het tweede gedeelte in figuur **Regenboog**. De rode lijn gaat van 255 naar 0, de groene lijn staat op 255 en de blauwe lijn staat bij nul.

Pas de code zo aan dat het lampje van rood naar geel (rood en groen) gaat.

Tip: voeg deze code toe:

```
for (int i=0; i<256; ++i)
{
  analogWrite(pin_rood, 255 - i);
  analogWrite(pin_groen, 255);
  analogWrite(pin_blaauw, 0);
  delay(30);
}
```

Oplossing

```
const int pin_rood   = 9; // Fout getal, zie vorige opdracht
const int pin_groen  = 10; // Fout getal, zie vorige opdracht
const int pin_blaauw = 11; // Fout getal, zie vorige opdracht

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

void loop()
{
  for (int i=0; i<256; ++i)
  {
    analogWrite(pin_rood, 255);
    analogWrite(pin_groen, i);
    analogWrite(pin_blaauw, 0);
    delay(30);
  }
  for (int i=0; i<256; ++i)
  {
    analogWrite(pin_rood, 255 - i);
    analogWrite(pin_groen, 255);
    analogWrite(pin_blaauw, 0);
    delay(30);
  }
}
```

Eindopdracht

Maak nu de regenboog af.

Les 1d: For nog een keer

In deze les gebruiken we meer `for` loops.

Intro

Bouw dit:

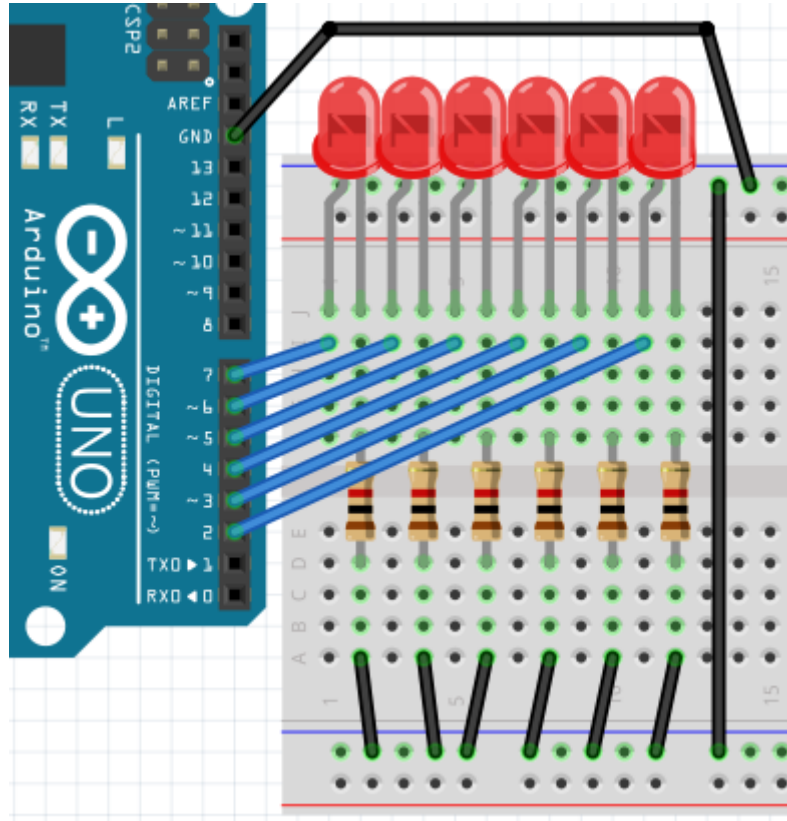


Figure 14: 1d_ForNogEenKeer

Programmeer dit:

```
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
}

void loop()
{
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  delay(1000);
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  delay(1000);
}
```

Opdracht

Verander het `setup` gedeelte naar een `for` loop.

Tip: gebruik:

```
for (int i=2; i<4; ++i)
{
    pinMode(i, OUTPUT);
}
```

Oplossing

```
void setup()
{
    for (int i=2; i<4; ++i)
    {
        pinMode(i, OUTPUT);
    }
}

void loop()
{
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    delay(1000);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    delay(1000);
}
```

Opdracht

Laat nu de LEDjes aan gaan met een for loop.

Oplossing

```
void setup()
{
    for (int i=2; i<4; ++i)
    {
        pinMode(i, OUTPUT);
    }
}

void loop()
{
    for (int i=2; i<4; ++i)
    {
        digitalWrite(i, HIGH);
    }
    delay(1000);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    delay(1000);
}
```

Opdracht

Laat nu de LEDjes uit gaan met een for loop

Oplossing

```
void setup()
{
  for (int i=2; i<4; ++i)
  {
    pinMode(i, OUTPUT);
  }
}

void loop()
{
  for (int i=2; i<4; ++i)
  {
    digitalWrite(i, HIGH);
  }
  delay(1000);
  for (int i=2; i<4; ++i)
  {
    digitalWrite(i, LOW);
  }
  delay(1000);
}
```

Opdracht

De zes lampjes zitten op pinnen 2 tot en met 7. Laat de for loops lopen van 2 tot 8. Als het goed is, gaan alle lampjes aan, dan alle lampjes uit.

Oplossing

```
void setup()
{
  for (int i=2; i<8; ++i)
  {
    pinMode(i, OUTPUT);
  }
}

void loop()
{
  for (int i=2; i<8; ++i)
  {
    digitalWrite(i, HIGH);
  }
  delay(1000);
  for (int i=2; i<8; ++i)
  {
    digitalWrite(i, LOW);
  }
  delay(1000);
}
```

Opdracht

Laat nu de lampjes omstebeurt aan en uit gaan, van links naar rechts. Je hebt hiervoor een for loop nodig.

Oplossing

```
void setup()
{
  for (int i=2; i<8; ++i)
  {
    pinMode(i, OUTPUT);
  }
}

void loop()
{
  for (int i=2; i<8; ++i)
  {
    digitalWrite(i, HIGH);
    delay(1000);
    digitalWrite(i, LOW);
    delay(1000);
  }
}
```

Opdracht

Laat nu de lampjes omstebeurt aan en uit gaan, van rechts naar links.

Tip: gebruik niet i, maar 7 - i.

Oplossing

```
void setup()
{
  for (int i=2; i<8; ++i)
  {
    pinMode(i, OUTPUT);
  }
}

void loop()
{
  for (int i=2; i<8; ++i)
  {
    digitalWrite(7 - i, HIGH);
    delay(1000);
    digitalWrite(7 - i, LOW);
    delay(1000);
  }
}
```

Eindopdracht

Maak nu een op-en-neer-gaand patroon in de LEDjes:

- *
- **
- ***
- ****
- *****
- ****
- ***
- **
- *