Handleiding

Kerstboom kit

Inhoudsopgave

Welkom

Wat leuk dat je een Awesome Crate gaat bouwen!

Deze crate bevat de benodigdheden en uitleg om een lichtgevende kerstboom te bouwen.

Deze kit is gericht op volwassenen. Uiteraard kan de kit ook door kinderen in elkaar gezet worden. Het kind dient dan wel altijd begeleid te worden door een volwassene.

In jouw Awesome Crate zit een microcontroller, een kleine computer waar we later meer over gaan uitleggen.

ⓘ Als je al eerder een van onze Awesome Crates hebt gebouwd zul je sommige uitleg al eens gelezen hebben. Voel je vrij om deze over te slaan, of juist nog een keer te lezen als opfrisser.

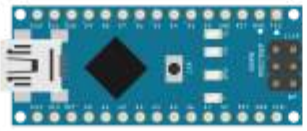
Veiligheidsregels

Om zo veel mogelijk plezier te beleven zijn hier een paar regels die ervoor zorgen dat de microcontroller niet beschadigt:

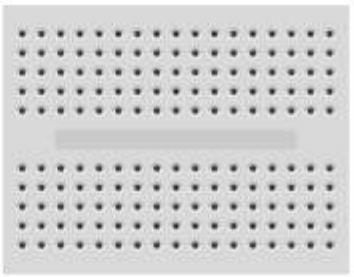
- Raak de microcontroller niet aan terwijl deze aan is.
- Zorg er altijd voor dat de microcontroller uit is (stroomkabeltje niet verbonden) voordat je onderdelen inplugt.
- Sluit de USB kabel eerst aan op de microcontroller en daarna pas op de stroombron (computer, USB stekker, of power bank). Daarmee voorkom je dat je de microcontroller aanraakt terwijl er stroom op staat.
- Plaats de microcontroller op een oppervlak dat geen stroom geleidt voordat je hem aan zet. Gebruik bijvoorbeeld een houten tafel, plastic mat of een tijdschrift.
- Schakel de microcontroller altijd uit nadat je er klaar mee bent.

ⓘ Dit is geen speelgoed. Bij het gebruik van dit artikel dienen kinderen onder de 14 jaar altijd begeleid te worden door een volwassene. Houd altijd toezicht op het product tijdens gebruik. Ontkoppel een eventuele stroombron na gebruik.

Microcontroller



Breadboard



USB kabel

Weerstandjes

LED lampjes

Ledring

Kerstboom

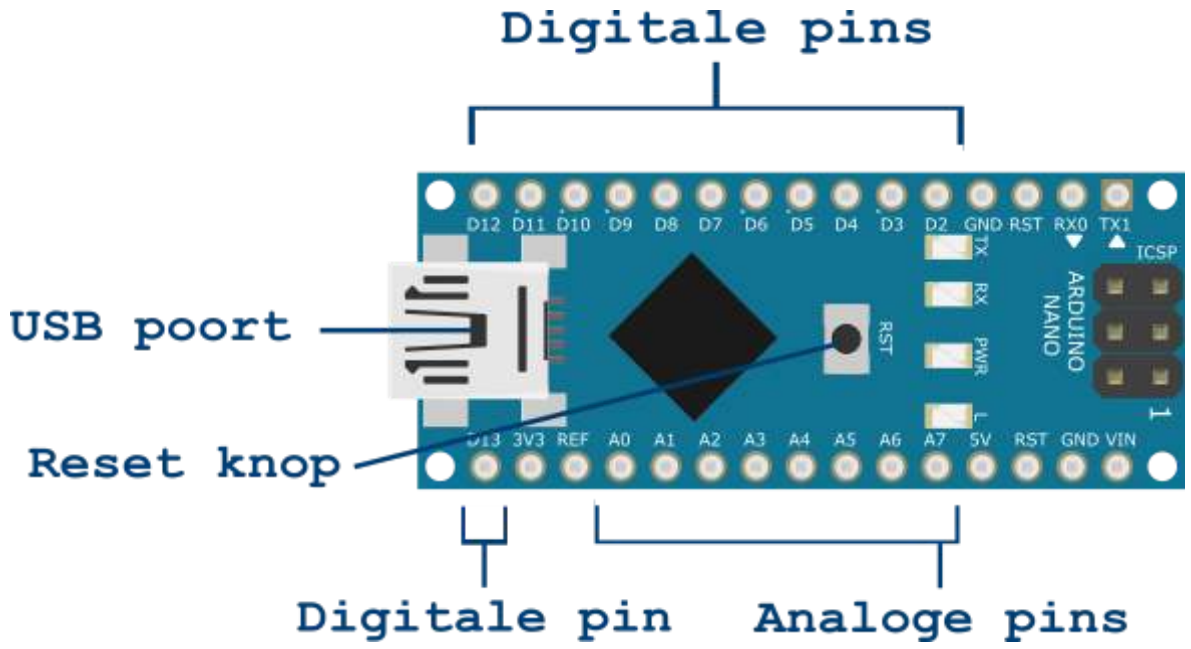
Ledring houder

De microcontroller

Het belangrijkste onderdeel in jouw Awesome Crate is de microcontroller. Dit is een kleine computer waarmee je andere elektronica kunt besturen. De microcontroller kan bijvoorbeeld een lampje aan en uit doen, of een motortje rond laten draaien.

De andere elektronica, bijvoorbeeld een lampje, kun je aansluiten op de microcontroller door deze met een pin te verbinden. (Zie Digitale pins en Analoge pin op het plaatje)

De microcontroller die wij gaan gebruiken is een Arduino Nano. Deze microcontroller lijkt veel op de Arduino UNO, welke in veel projecten gebruikt wordt. De Arduino Nano is echter veel kleiner en daardoor beter geschikt in projecten waarbij je de microcontroller wilt verstoppen.



Afbeelding 0.1

Arduino IDE installeren

We gaan straks de microcontroller programmeren. Hiervoor gebruiken we de Arduino IDE (Integrated Development Environment). De Arduino IDE is een applicatie waarmee je code voor een microcontroller kunt schrijven en waarmee je de code op de microcontroller kunt zetten.

Gebruik je Mac?

Ga dan verder bij Arduino IDE installeren op Mac

Gebruik je Windows?

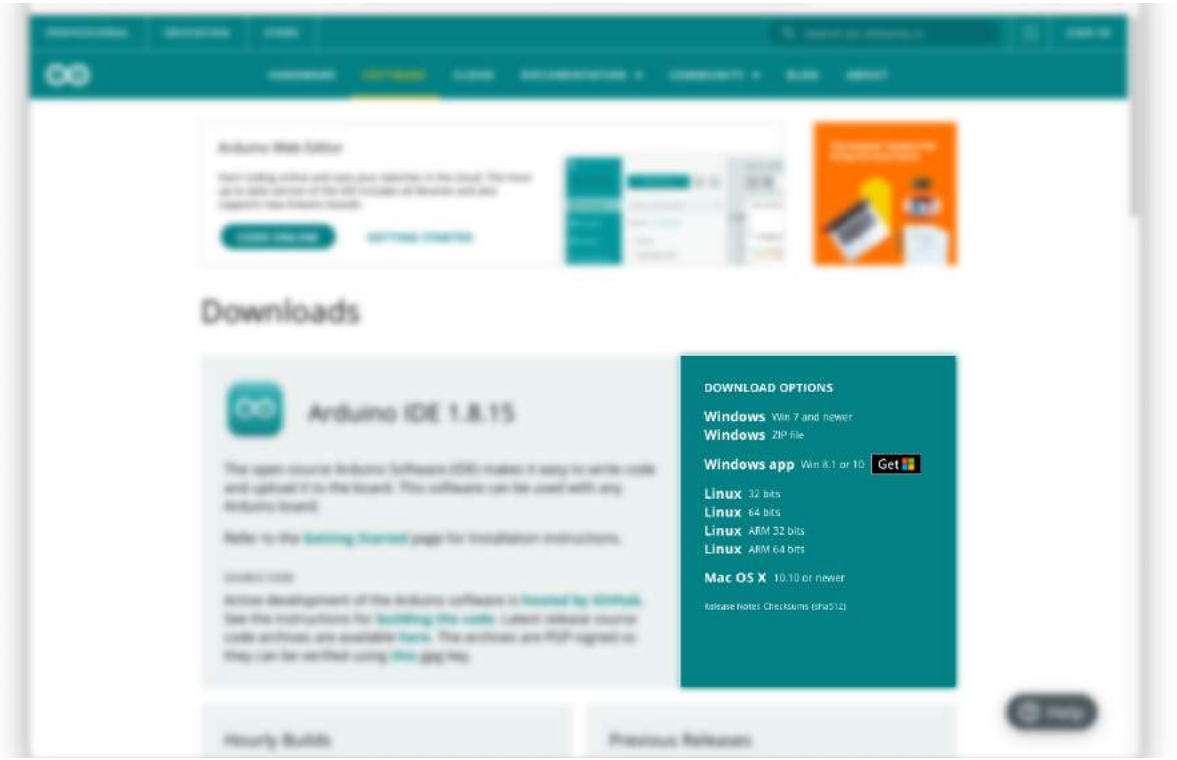
Ga dan verder bij Arduino IDE installeren op Windows

Arduino IDE installeren op Mac

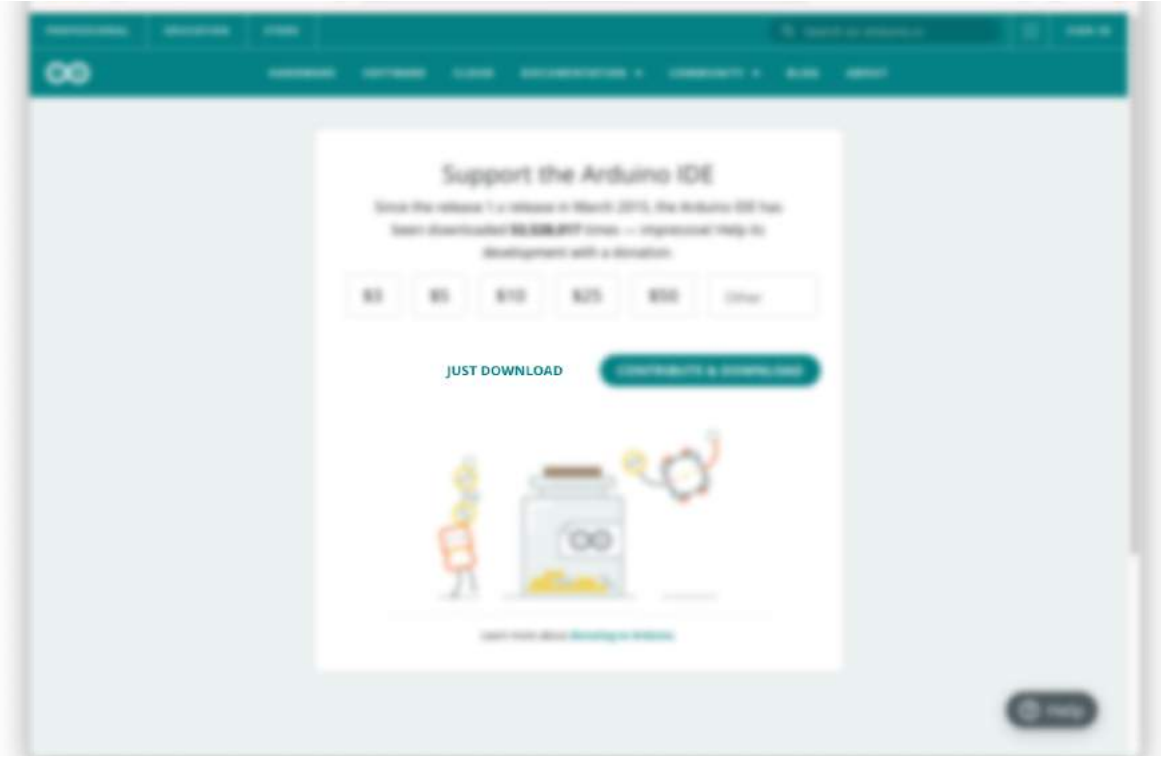
Stappen

1. Ga in de browser naar https://www.arduino.cc/en/software.

2. Klik in het vak "DOWNLOAD OPTIONS" (rechts op de pagina) op Mac OS X.



3. Je kan er voor kiezen om een bijdrage te leveren aan de ontwikkeling van de Arduino IDE maar dit is niet verplicht. Wanneer je geen bijdrage wenst te leveren klik je op de knop "JUST DOWNLOAD".



4. De Arduino IDE wordt nu gedownload. Verplaats het bestand daarna naar de applicatie map.

5. Open het programma. Het kan zijn dat jouw besturingssysteem nu om toestemming vraagt om de applicatie te openen. Geef in dit geval toestemming hiervoor.

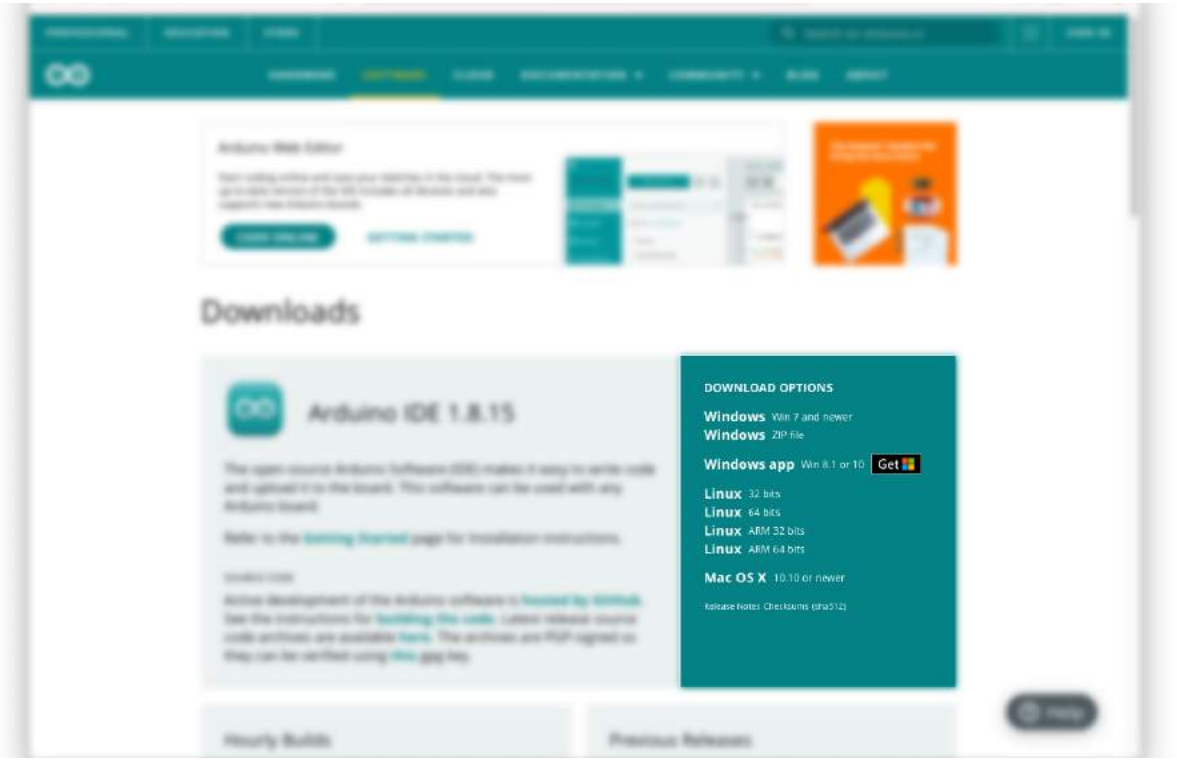
6. De Arduino IDE wordt nu geopend.

Arduino IDE installeren op Windows

Stappen

1. Ga in de browser naar https://www.arduino.cc/en/software.

2. Klik in het vak "DOWNLOAD OPTIONS" (rechts op de pagina) op Windows (Win 7 and newer).



3. Je kan er voor kiezen om een bijdrage te leveren aan de ontwikkeling van de Arduino IDE maar dit is niet verplicht. Wanneer je geen bijdrage wenst te leveren klik je op de knop "JUST DOWNLOAD".

4. De Arduino IDE wordt nu gedownload.

5. Open het bestand (deze staat in je Downloads map). Het kan zijn dat jouw besturingssysteem nu om toestemming vraagt om de applicatie te openen. Geef in dit geval toestemming hiervoor.

6. Volg de stappen in de installatiewizard, je kunt alle standaardwaarden gebruiken.

7. Je besturingssysteem vraagt mogelijk meerdere keren om verschillende onderdelen te installeren. Geef hiervoor toestemming.

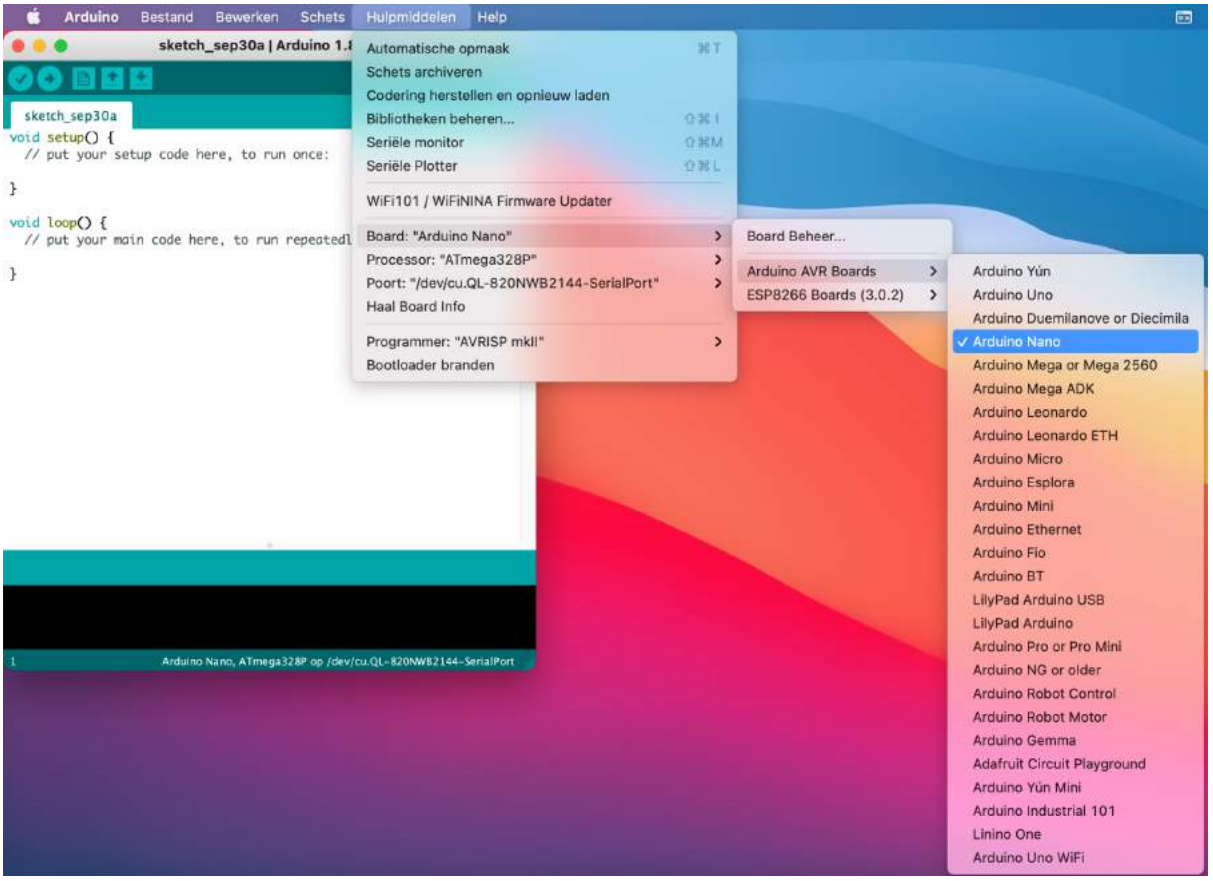
8. Open nu de Arduino IDE. Mogelijk vraagt je firewall om toestemming, geef ook hier toestemming.

Het juiste bord kiezen

Voor dit project gebruiken we als microcontroller een Arduino Nano. Er zijn een heleboel andere soorten microcontrollers die je ook kunt programmeren via de Arduino IDE. Daarom moeten we in de IDE het juiste bord selecteren.

Stappen

1. Klik in de menubalk op ***Hulpmiddelen*** en vervolgens op ***Board***, dan op ***Arduino AVR Boards*** en selecteer ***Arduino Nano***.



Mooi, de Arduino IDE is nu geconfigureerd. We kunnen de microcontroller nu op de computer gaan aansluiten.

De microcontroller op de computer aansluiten

Nu gaan we de microcontroller via een USB kabel verbinden met de computer. Dit is de laatste stap voordat we code kunnen uploaden naar de microcontroller.

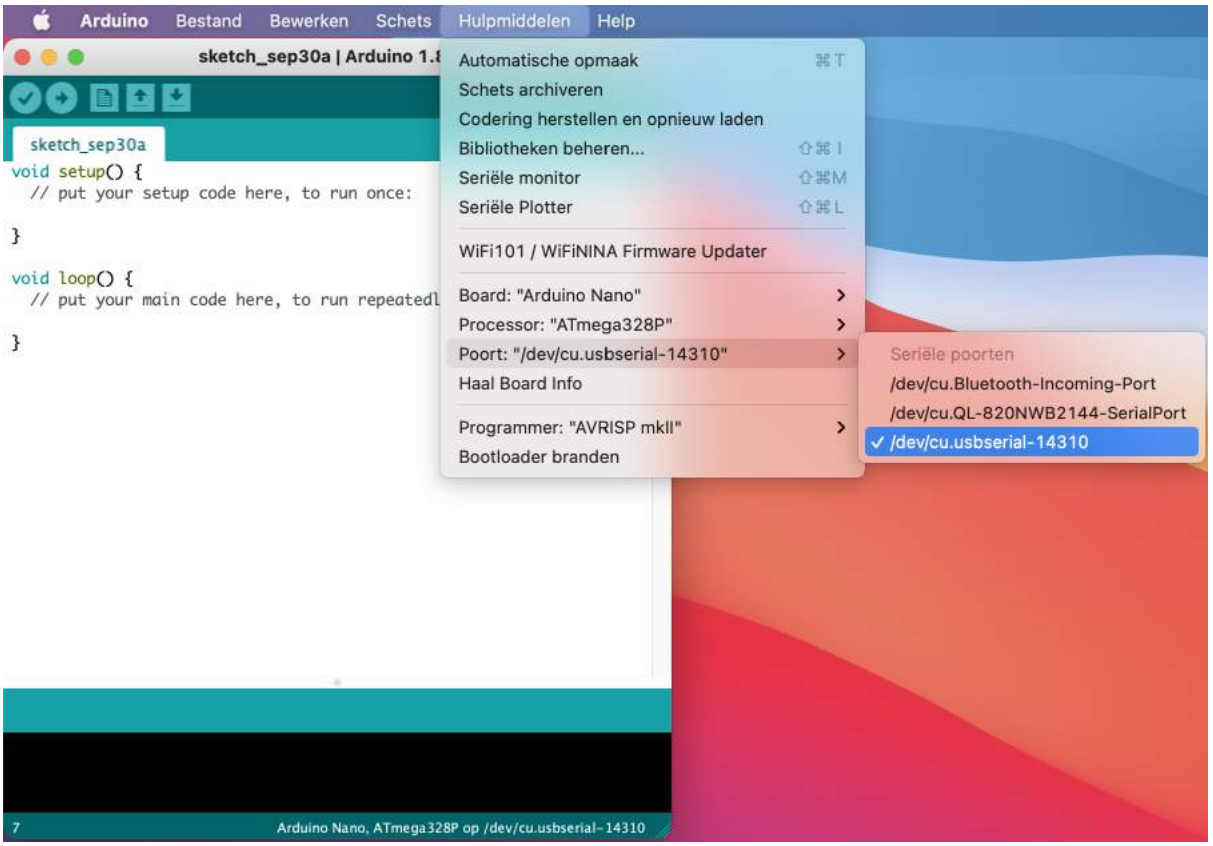
Stappen

Het is belangrijk om bij het aansluiten van de microcontroller de veiligheidsregels (blz. 1) na te leven.

1. Sluit de USB kabel eerst aan op de microcontroller en vervolgens op je computer.

2. Klik in de menubalk op ***Hulpmiddelen*** en vervolgens op ***Poort***. Selecteer nu de poort waarop de microcontroller is aangesloten. Deze poort zal een naam hebben zoals "*cu.usbserial-14410*"(op Mac) of *"COM1"* (op Windows).

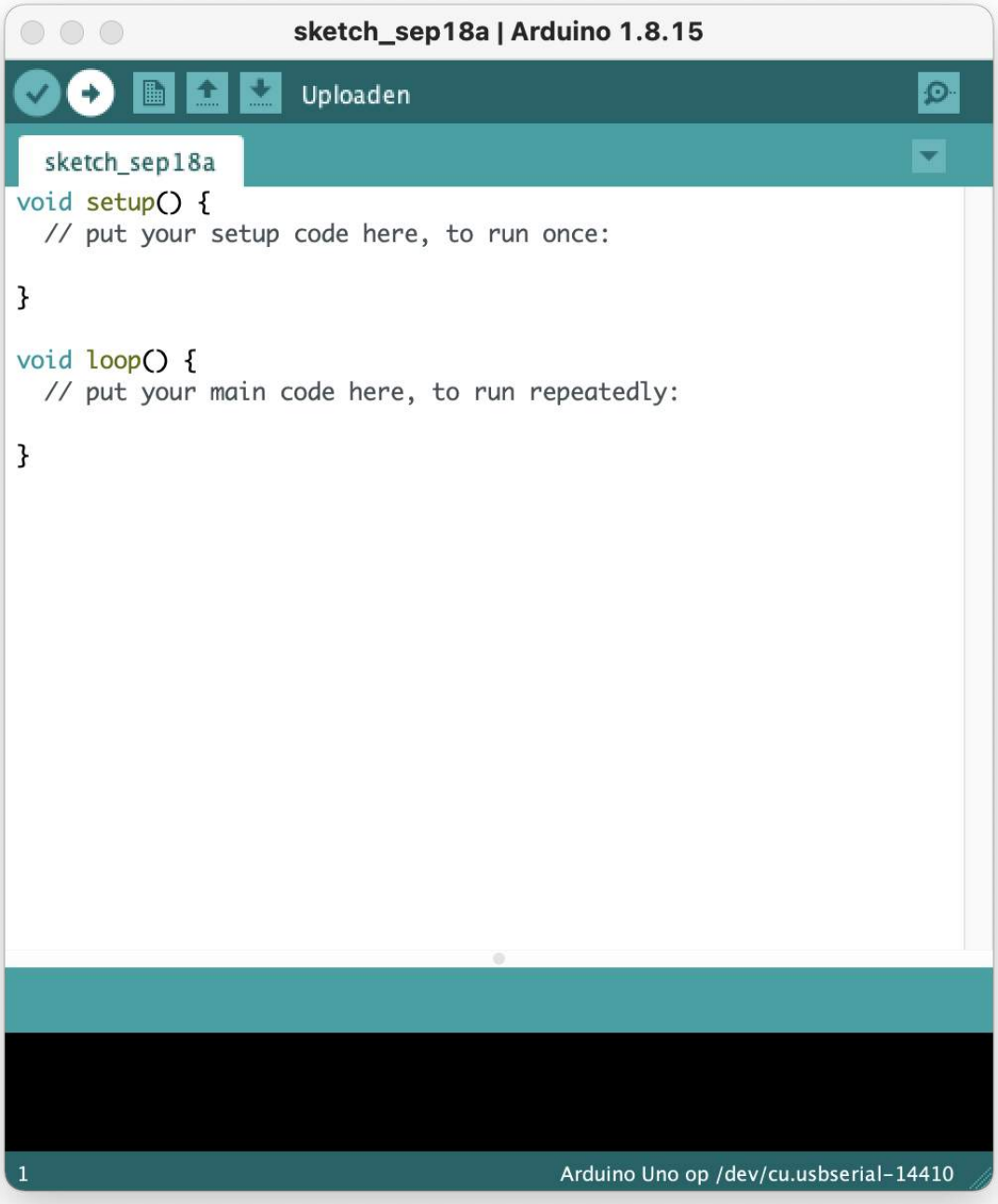
   Weet je niet welke poort je moet gebruiken? Kijk dan welke poorten er nu in de lijst staan. Sluit het menu, ontkoppel de microcontroller en kijk opnieuw welke poorten er in de lijst staan. De poort die niet langer in de lijst staat is de poort van de microcontroller.



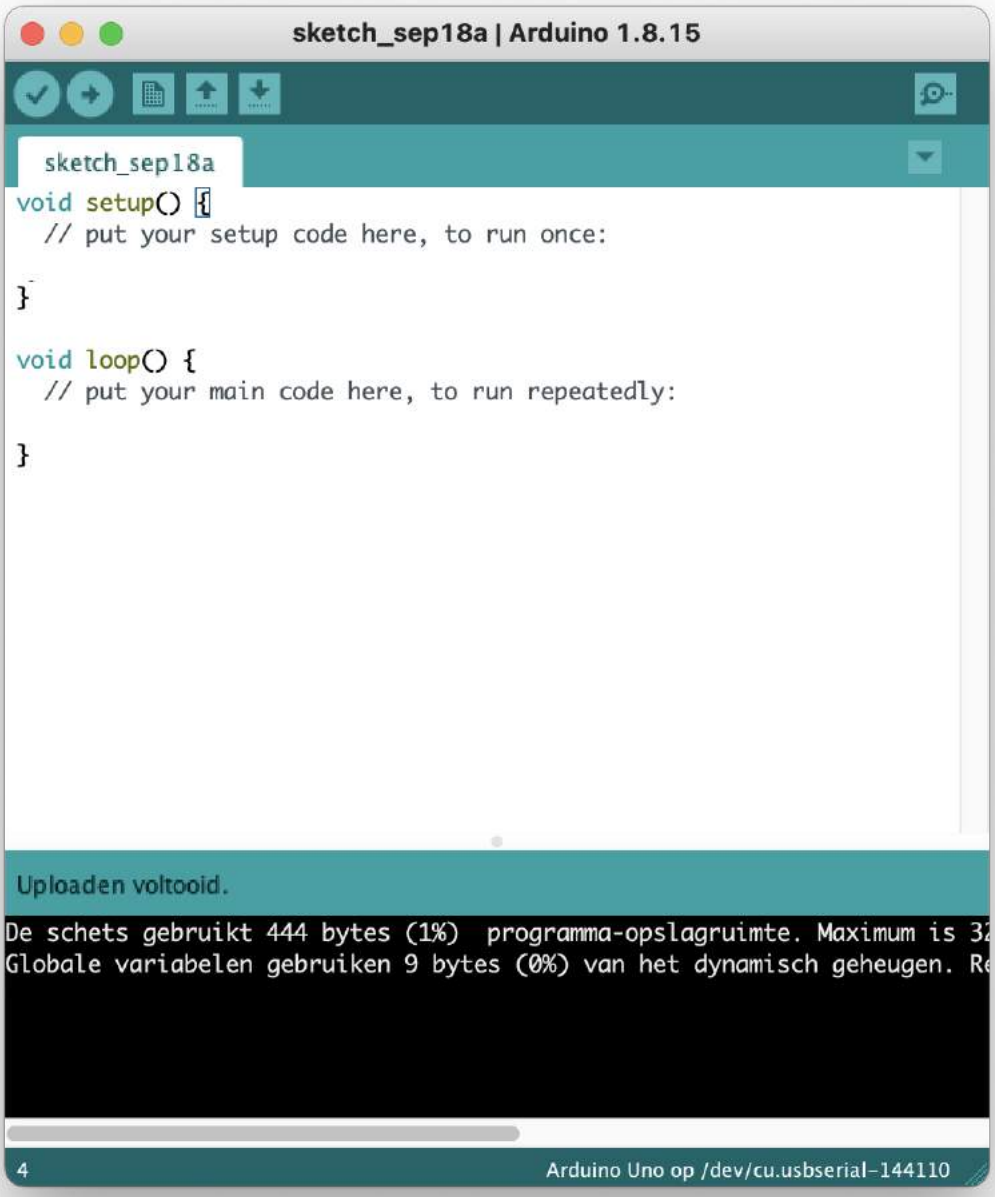
Lukt het niet om te verbinden met de microcontroller? Dan kan het zijn dat je eerst nog de CH340 driver moet installeren. Ga hiervoor naar https://www.awesomemakes.com/drivers/ en download en installeer de driver voor jouw besturingssysteem.

Nu de juiste poort geselecteerd is kunnen we de verbinding testen. Dit doen we door een vrijwel leeg programma naar de microcontroller te uploaden.

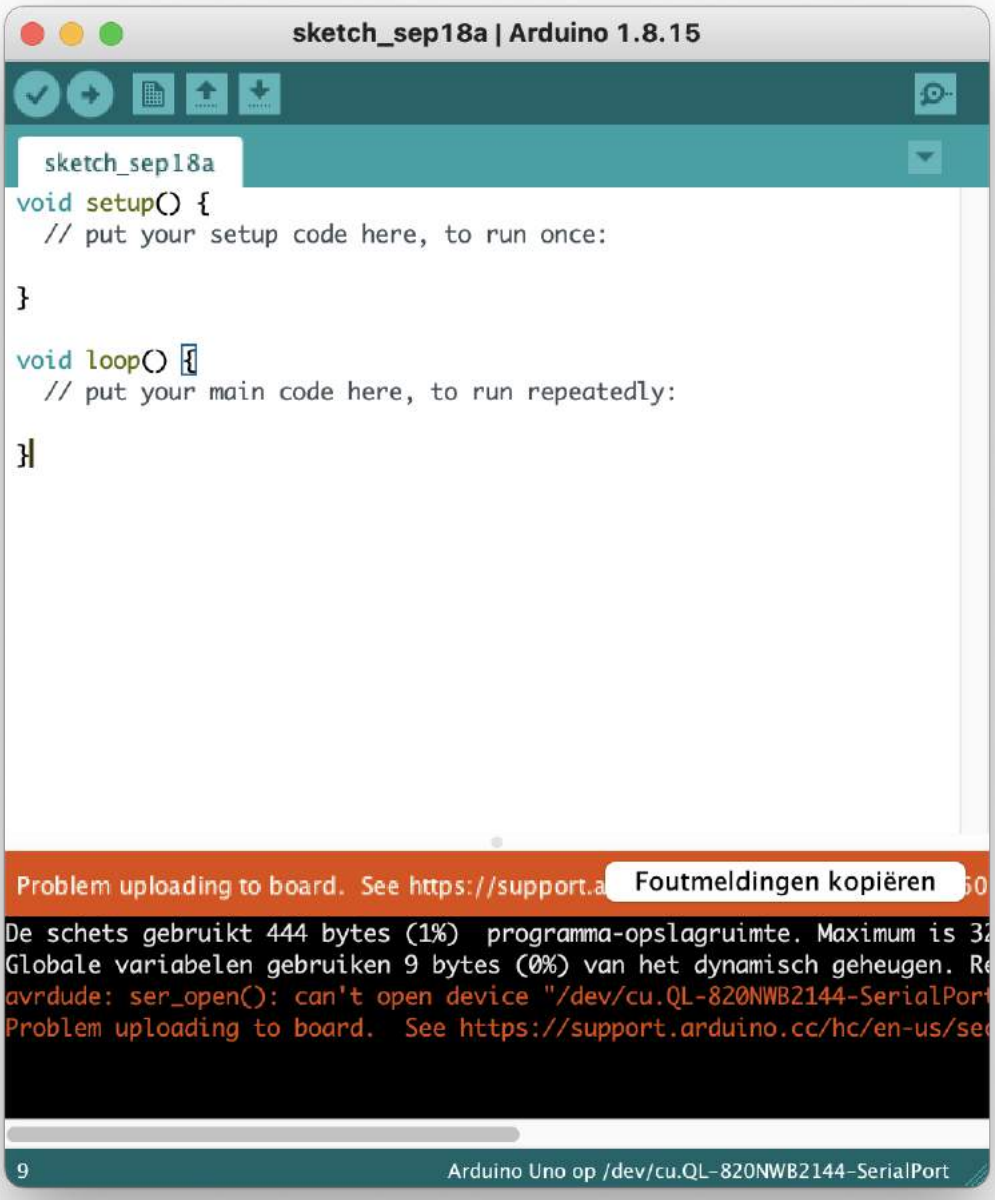
1. Klik op de ***Uploaden*** knop  .



In het zwarte vlak onder in het scherm zie je nu een aantal berichten verschijnen. In de groene balk daarboven (statusbalk) zie je de status van het uploaden. Staat er in de groene balk "*Uploaden voltooid*"? Dan is het uploaden geslaagd.



Wordt de statusbalk oranje? Dan is er een fout opgetreden tijdens het uploaden. Doorloop in dit geval de stappen voor het kiezen van het bord en het aansluiten van de microcontroller nogmaals om te zien of je daar iets vergeten bent. Upload het programma daarna opnieuw.



Oefening 1: Laat een LED knipperen

De Arduino Nano heeft een ingebouwd LED lampje. Om gewend te raken aan het programmeren van de microcontroller gaan we deze LED eerst eens 2 keer aan en uit zetten.

De ingebouwde LED is te bedienen door een hoog of laag signaal op pin 13 te zetten. De interne LED is namelijk direct verbonden aan pin 13.

De ingebouwde LED is zo aangesloten dat deze aan gaat wanneer er een hoog signaal op pin 13 gezet wordt. De LED gaat uit wanneer er een laag signaal op de pin gezet wordt. We noemen dit active high aangezien de LED actief wordt bij een hoog signaal.

Benodigdheden

- Arduino Nano
- USB kabel
- Computer

Stappen

1. Neem de onderstaande code over in de Arduino IDE. (Vervang hierbij altijd alle bestaande code, tenzij anders vermeld.)

```
// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  pinMode(13, OUTPUT);      // We gaan pin 13 gebruiken als output

  digitalWrite(13, HIGH);   // Zet de LED aan door een hoog signaal op 13 zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(13, LOW);    // Zet de LED uit door een laag signaal op 13 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)

  digitalWrite(13, HIGH);   // Zet de LED aan door een hoog signaal op 13 zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(13, LOW);    // Zet de LED uit door een laag signaal op 13 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
}

void loop() {
}
```

Arduino code oefening 1

ⓘ In de code staan regels achter twee schuine strepen ("//"), dit zijn aantekeningen. Deze regels zijn puur ter verduidelijking en hebben geen invloed op de uitvoer van het programma.

2. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.
3. Klik op de ***Verifiëren*** knop  . De Arduino IDE controleert nu of de code die je hebt ingevoerd juist is.

   Waarschijnlijk vraagt Arduino IDE je nu om een locatie en een naam te kiezen om de sketch onder op te slaan. Kies hier een naam en locatie waarmee je jouw code weer terug weet te vinden. Iedere keer dat je een ***Verifiëren*** uitvoert wordt de nieuwe versie van je code op deze plek opgeslagen.

   Wanneer alles goed gaat verschijnt de tekst "*Compileren voltooid.*" in de statusbalk.

   Wordt de statusbalk oranje? Dan staat er een fout in de code of je hebt niet het juiste bord geselecteerd. Kijk of je de code juist gekopieerd hebt en controleer of de port en het bord dat je geselecteerd hebt juist zijn.

De code is nu gecontroleerd door de Arduino IDE. Deze stap wordt ook standaard uitgevoerd wanneer je op de ***Uploaden*** knop  klikt. Het is dus niet altijd nodig het programma te verifiëren voordat je het uploadt.

4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan gaat de LED op de microcontroller nu twee keer aan en uit. Goed gedaan!

Druk op de reset knop (zie afbeelding 0.1 van de microcontroller. Dit zorgt ervoor dat de microcontroller opnieuw opstart en dus de code nogmaals uitvoert.

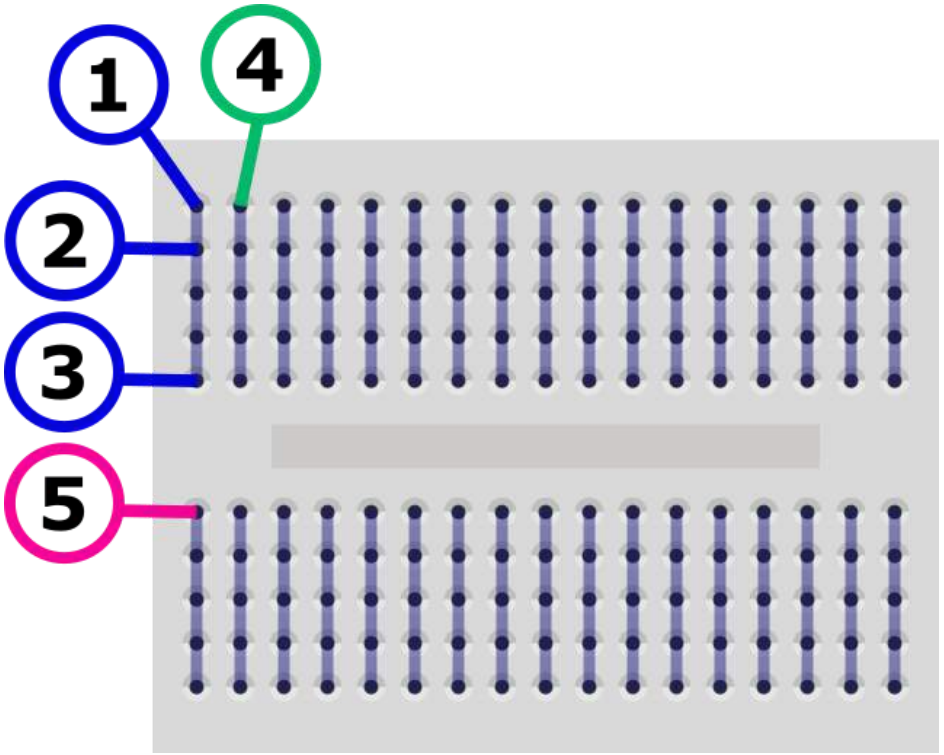
Knippert het lampje niet? Bekijk dan de tips onder Problemen oplossen.

Het breadboard

In oefening 1 hebben we een ingebouwde led op de microcontroller laten knipperen. Het mooie aan microcontrollers is dat je ook externe onderdelen aan kunt sluiten. Om gemakkelijk meerdere onderdelen aan te sluiten op de microcontroller gebruiken we een breadboard.

Om de onderdelen met elkaar te verbinden kun je de draadjes en pootjes van de onderdelen in de gaatjes op het breadboard steken. Binnenin het breadboard zijn de gaatjes op een speciale manier met elkaar verbonden, dit is te zien in het plaatje.

Dit betekent dat gaatje 1 verbonden is met gaatje 2 en 3, maar niét met gaatje 4 en ook niet met gaatje 5.



Afbeelding 0.2

Oefening 2: Een LED aansluiten

In deze oefening gaan we het breadboard gebruiken om een externe LED aan te sluiten en te laten knipperen.

Benodigdheden

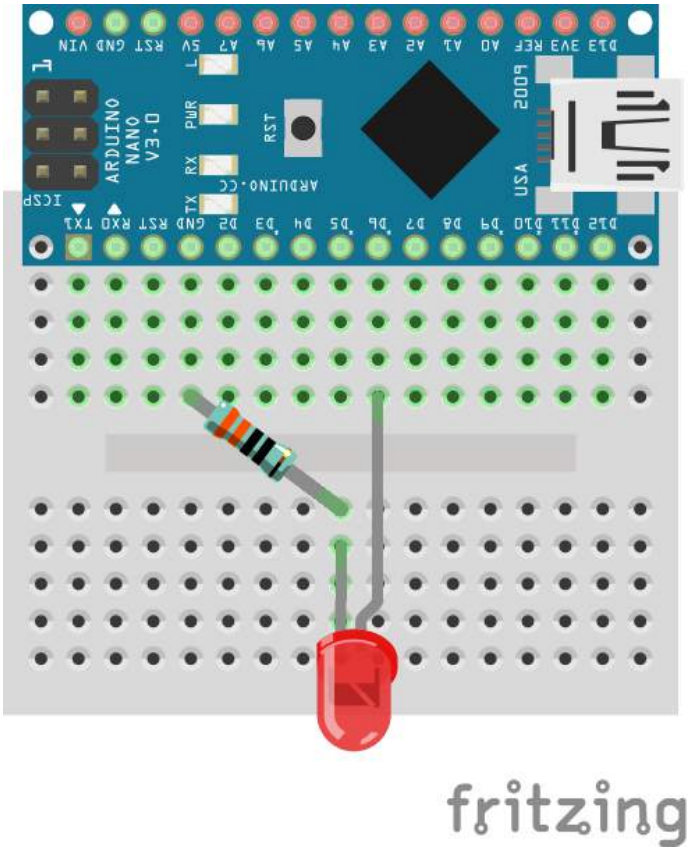
- Arduino Nano
- USB kabel
- Computer
- Weerstand
- Breadboard
- LED lampje

Verbindingen

Microcontroller	*Onderdeel*	*Via*
GND	LED korte pootje	weerstand
D6	LED lange pootje	-

Stappen

1. Maak de schakeling uit afbeelding 2.1.



Afbeelding 2.1

ⓘ Het lampje heeft een lang pootje en een korter pootje. Het lange pootje is de plus (+) het korte pootje is de min (-).

2. Neem de onderstaande code over in de Arduino IDE.

```
// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  pinMode(6, OUTPUT);       // We gaan pin 6 gebruiken als output

  digitalWrite(6, HIGH);    // Zet de LED aan door een hoog signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(6, LOW);     // Zet de LED uit door een laag signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)

  digitalWrite(6, HIGH);    // Zet de LED aan door een hoog signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(6, LOW);     // Zet de LED uit door een laag signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
}
```

Arduino code oefening 2

We gebruiken hier bijna dezelfde code als in oefening 1. We hebben de pin die we willen aansturen veranderd van pin 13 naar pin 6. We willen namelijk niet meer de ingebouwde LED op pin 13 gebruiken maar de LED die wij hebben aangesloten op pin 6.

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.

4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan knippert de LED die je op het breadboard hebt aangesloten.

Druk op de reset knop (zie afbeelding 0.1 van de microcontroller om de code nogmaals uit te voeren.

Knippert de LED niet? Bekijk dan de tips onder Problemen oplossen.

Oefening 3: Functies

In oefening 2 hebben we de LED aan en weer uitgezet met het volgende stukje code:

```
  digitalWrite(6, HIGH);    // Zet de LED aan door een hoog signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(6, LOW);     // Zet de LED uit door een laag signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
```

Dit stuk code herhalen we twee keer. Dat kan eenvoudiger. Hiervoor gaan we een functie gebruiken. Een functie is een verzameling van instructies welke samen een taak uitvoeren.

Wanneer we de code om de led aan en dan weer uit te zetten omzetten naar een functie dan ziet dat er als volgt uit:

```
void zetLedAanEnUit() {
  digitalWrite(6, HIGH);    // Zet de LED aan door een hoog signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(6, LOW);     // Zet de LED uit door een laag signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
}
```

Bovenstaande code noemen we de *definitie* van de functie. In ons programma kunnen we nu de functie `zetLedAanEnUit` aanroepen. Bij de *aanroep* van een functie wordt de inhoud van de functie op die plek uitgevoerd. Het programma ziet er dan als volgt uit:

```
// Deze functie zet de LED aan en na 1 seconde weer uit
void zetLedAanEnUit() {
  digitalWrite(6, HIGH);    // Zet de LED aan door een hoog signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(6, LOW);     // Zet de LED uit door een laag signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  pinMode(6, OUTPUT);       // We gaan pin 6 gebruiken als output
  zetLedAanEnUit();         // Voer de inhoud van de functie zetLEDAanEnUit uit
  zetLedAanEnUit();         // Voer de inhoud van de functie zetLEDAanEnUit uit
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
}
```

Arduino code oefening 3

Benodigdheden

- Schakeling uit oefening 2
- USB kabel
- Computer

Stappen

1. Maak de schakeling die we in oefening 2 gebruikt hebben.

2. Neem de code uit het blok Arduino code oefening 3 over in de Arduino IDE.

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.

4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan knippert de LED die je op het breadboard hebt aangesloten net als in oefening 2.

Knippert de LED niet? Bekijk dan de tips onder Problemen oplossen.

Oefening 4: de "loop" gebruiken

In oefening 3 hebben we de functie **zetLedAanEnUit()** twee keer aangeroepen om de LED te laten knipperen. In deze opdracht gaan we de LED laten knipperen zolang de microcontroller aan staat. We doen dit door de loop (Engels voor "lus") functie te gebruiken.

In al de vorige voorbeelden zie je onderaan de volgende code staan:

```
void loop() {
}
```

Wanneer de microcontroller wordt aangezet voert hij eerst de code uit binnen de setup functie. Daarna wordt de code die binnen de loop functie staat herhaald zolang de microcontroller aan staat.

> ⓘ Misschien is het je opgevallen dat **setup()** en **loop()** er hetzelfde uitzien als de functie **zetLedAanEnUit()** die we net hebben gemaakt. Goed gezien, **setup()** en **loop()** zijn functies waarvan jij de definitie mag schrijven, maar waarbij de aanroep automatisch wordt geregeld door de microcontroller.

Door de aanroep van onze functie **zetLedAanEnUit()** in de loop te zetten wordt deze herhaald tot de microcontroller uit gaat.

Benodigdheden

- Schakeling uit oefening 2
- USB kabel
- Computer

Verbindingen

Microcontroller	*Onderdeel*	*Via*
GND	LED korte pootje	Weerstand
6	LED lange pootje	Jumper kabeltje

Stappen

1. Maak de schakeling die we in oefening 2 gebruikt hebben.

2. Neem de onderstaande code over in de Arduino IDE.

```
// Deze functie zet de LED aan en na 1 seconde weer uit
void zetLedAanEnUit() {
  digitalWrite(6, HIGH);    // Zet de LED aan door een hoog signaal op pin 6 zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(6, LOW);     // Zet de LED uit door een laag signaal op pin 6 te zetten
  delay(1000);              // Wacht 1 seconde (1000 milliseconden)
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  pinMode(6, OUTPUT);       // We gaan pin 6 gebruiken als output
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  zetLedAanEnUit();         // Roep de functie zetLedAanEnUit aan
}
```

Arduino code oefening 4

De code waarmee we aangeven dat we pin 6 als output gaan gebruiken blijft in de setup functie staan. Dit hoeven we namelijk slechts één keer aan te geven.

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.
4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan knippert de LED die je op het breadboard hebt aangesloten. In tegenstelling tot de vorige oefeningen blijft de LED dit keer knipperen tot je de stroom van de microcontroller afhaalt.

Knippert de LED niet? Bekijk dan de tips onder Problemen oplossen.

Oefening 5: variabelen

In deze oefening gaan we onze LED sneller laten knipperen. Dit doen we door de instructie die de microcontroller laat wachten aan te passen. Deze code ziet er op dit moment als volgt uit:

```
delay(1000);              // Wacht 1 seconde (1000 milliseconden)
```

De door de functie delay aan te roepen met de waarde 1000 laten we de microcontroller 1000 milliseconden wachten. Door de waarde die we aan de functie delay meegeven aan te passen kunnen we de microcontroller korter of juist langer laten wachten.

ⓘ Net zoals de functie zetLedAanEnUit is delay ook een functie. De inhoud (we noemen dit de definitie van de functie) van de delay functie hoeven we alleen niet zelf te schrijven maar is ingebouwd in de Arduino programmeertaal. Dit zorgt ervoor dat we makkelijk en snel code kunnen schrijven zonder dat we veel gebruikte functionaliteit zelf opnieuw hoeven te schrijven.

We zijn inmiddels al 4 functies tegengekomen die op verschillende manieren worden gedefinieerd en aangeroepen. Mocht je even niet meer precies weten hoe het zat dan geeft onderstaande tabel misschien wat duidelijkheid.

functie	definitie	aanroep
setup en loop	Geschreven door jou	Automatisch aangeroepen door microcontroller
delay	Geschreven door Arduino ontwikkelaars	Aangeroepen door jou
zetLedAanEnUit	Geschreven door jou	Aangeroepen door jou

Op de plekken in onze functie **zetLedAanEnUit()** waar we **delay()** aanroepen kunnen we de waarde 1000 vervangen door een lager getal om de LED sneller te laten knipperen.

We gaan hier echter geen harde waardes invullen maar gebruik maken van een variabele. Een variabele is een plek waar je informatie kunt opslaan en die je later weer kunt uitlezen.

Je maakt een variabele op de volgende manier:

```
int aantalMillisecondenWachten = 1000;
```

Net als bij een functie, noemen we het maken van een variabele *declareren*. De declaratie van een variabele kent drie onderdelen:

Het datatype van de variabele
In dit voorbeeld is het datatype `int` wat staat voor integer oftewel "geheel getal". Aan de hand van het datatype bepaal je wat voor gegevens er in de variabele opgeslagen kunnen worden. In dit geval kunnen we hele getallen in onze variabele stoppen. Veel gebruikte datatypes zijn:

Datatype	*Voor het opslaan van*	*Voorbeeldwaarde*
`int`	Gehele getallen	12345
`boolean`	Iets dat slechts twee waardes kan hebben	TRUE of FALSE
`char`	Een teken (character)	'A' of '!'
`float`	Getallen met een komma *(Deze programmeertaal gebruikt de Engelse notatie voor getallen. We gebruiken daarom een punt in plaats van een komma)*	123.45

Naam van de variabele
In dit voorbeeld is de naam van de variabele `aantalMillisecondenWachten`. Je mag deze naam zelf verzinnen. Het is verstandig om een naam te kiezen die duidelijk omschrijft welke waarde er in de variabele zit.

De waarde van de variabele
In dit voorbeeld is de waarde van de variabele 1000. Het type waarde dat je kunt toewijzen is afhankelijk van het datatype van de variabele. In ons voorbeeld gebruiken we een `int` waardoor we alleen hele getallen kunnen toewijzen (zoals 1000). Hadden we bijvoorbeeld het datatype `char` gekozen dan hadden we een teken toe kunnen wijzen. De code had er dan zo uit kunnen zien:

```
char eenVoorbeeldTeken = 'c';
```

Benodigdheden

- Schakeling uit oefening 2
- USB kabel
- Computer

Stappen

1. Maak de schakeling die we in oefening 2 gebruikt hebben.

2. Neem de onderstaande code over in de Arduino IDE.

```
// Deze functie zet de LED aan en na 0,5 seconde weer uit
void zetLedAanEnUit() {
  int aantalMillisecondenWachten = 500;  // Variabele die bepaalt hoe lang we wachten
  digitalWrite(6, HIGH);                 // Zet de LED aan met een hoog signaal op pin 6
  delay(aantalMillisecondenWachten);     // Wacht een halve seconde (500 milliseconden)
  digitalWrite(6, LOW);                  // Zet de LED uit met een laag signaal op pin 6
  delay(aantalMillisecondenWachten);     // Wacht een halve seconde (500 milliseconden)
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  pinMode(6, OUTPUT);      // We gaan pin 6 gebruiken als output
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  zetLedAanEnUit();         // Roep de functie zetLedAanEnUit aan
}
```

Arduino code oefening 5

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.

4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan knippert de LED nu twee keer zo snel aangezien we de tijd die de microcontroller wacht gehalveerd hebben.

Je kunt nu de snelheid waarmee de LED knippert makkelijk aanpassen door de variabele **aantalMillisecondenWachten()** aan te passen en de code opnieuw te uploaden.

Knippert de LED niet? Bekijk dan de tips onder Problemen oplossen.

Oefening 6: LED dimmen

We kunnen nu de LED zo snel laten knipperen als dat wij willen. Daarmee hebben we eigenlijk al de basis gelegd voor een programma die de LED dimt. Een LED dimmen doe je namelijk door deze heel snel te laten knipperen.

Een LED die je steeds een halve seconde aan zet en dan weer een halve seconde uit geeft gemiddeld 50% minder licht dan een LED die constant brandt. Omdat de LED relatief lang uit is noemen we dit geen dimmen maar knipperen.

Wanneer we de LED snel genoeg laten knipperen zien we met onze ogen niet meer dat de LED knippert maar zien we alleen dat deze minder licht geeft.

In deze opdracht gaan we dit principe gebruiken om de LED feller en minder fel te laten branden.

Parameters

In de voorgaande opdrachten hebben we de ingebouwde functie `delay` gebruikt om de microcontroller een aantal milliseconden te laten wachten. Dit zag er als volgt uit:

```
delay(1000);              // Wacht 1 seconde (1000 milliseconden)
```

Door een waarde aan deze functie mee te geven (in het voorbeeld 1000) kunnen we aangeven hoe lang de microcontroller moet wachten. Wanneer we een waarde aan een functie meegeven noemen we deze waarde een parameter.

Ook bij onze eigen functie kunnen we parameters meegeven. In deze opdracht gaan we aan onze functie `zetLedAanEnUit()` twee parameters meegeven. Een parameter **aantalMillisecondenAan()** die bepaalt hoe lang de LED aan is en een parameter **aantalMillisecondenUit()** die bepaalt hoe lang de LED uit is.

Hiervoor gaan we eerst de functie `zetLedAanEnUit()` als volgt aanpassen:

```
1 void zetLedAanEnUit(int aantalMillisecondenAan, int aantalMillisecondenUit) {
2   digitalWrite(6, HIGH);                // Zet de LED aan met een hoog signaal op D6
3   delay(aantalMillisecondenAan);        // Wacht het aangegeven aantal milliseconden
4   digitalWrite(6, LOW);                 // Zet de LED uit met een laag signaal op D6
5   delay(aantalMillisecondenUit);        // Wacht het aangegeven aantal milliseconden
6 }
```

Op regel 1 zetten we tussen de ronde haakjes onze twee parameters genaamd **aantalMillisecondenAan** en **aantalMillisecondenUit**. Beiden hebben

het datatype `int`. Deze parameters kunnen we nu in de rest van deze functie gebruiken als variabelen.

De declaratie van de variabele `aantalMillisecondenWachten` hebben we verwijderd. Deze is niet meer nodig aangezien we nu gebruik maken van de parameters `aantalMillisecondenAan` en `aantalMillisecondenUit`.

Naast het aanpassen van de de functie `zetLedAanEnUit()` moeten we ook de plek waar we deze functie aanroepen wijzigen. Toen we nog geen parameters aan onze functie hadden toegevoegd riepen we deze als volgt aan:

```
zetLedAanEnUit();
```

Nu we parameters aan de functie hebben toegevoegd moeten we bij de aanroep van de functie waardes voor deze parameters meegeven. Dat ziet er bijvoorbeeld als volgt uit:

```
zetLedAanEnUit(1, 25);
```

In dit voorbeeld vullen we de parameter `aantalMillisecondenAan` met de waarde 1 en de parameter `aantalMillisecondenUit` met de waarde 25.

Benodigdheden

- Schakeling uit oefening 2
- USB kabel
- Computer

Stappen

1. Maak de schakeling die we in oefening 2 gebruikt hebben.

2. Neem de onderstaande code over in de Arduino IDE.

```
void zetLedAanEnUit(int aantalMillisecondenAan, int aantalMillisecondenUit) {
  digitalWrite(6, HIGH);                // Zet de LED aan met een hoog signaal op pin 6
  delay(aantalMillisecondenAan);        // Wacht het aangegeven aantal milliseconden
  digitalWrite(6, LOW);                 // Zet de LED uit met een laag signaal op pin 6
  delay(aantalMillisecondenUit);        // Wacht het aangegeven aantal milliseconden
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  pinMode(6, OUTPUT);                   // We gaan pin 6 gebruiken als output
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  zetLedAanEnUit(1, 25);                // Roep de functie zetLedAanEnUit aan
}
```

Arduino code oefening 6

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.
4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan brandt de LED nu minder fel dan voorheen.

Je kunt nu de helderheid van de LED aanpassen door de parameters die we meegeven aan de functie `zetLedAanEnUit()` aan te passen en de code opnieuw te uploaden.

Gaat de LED niet aan of brandt deze niet minder fel? Bekijk dan de tips onder Problemen oplossen.

Oefening 7: De NeoPixel ring

In deze oefening gaan we één LED op onze NeoPixel ring laten branden. Een NeoPixel is een module met daarop 3 hele kleine LED lampjes. Een rode, een groene en een blauwe. Door deze drie LED lampjes in verschillende sterktes te laten branden kan de NeoPixel heel veel verschillende kleuren genereren. Iedere NeoPixel heeft zijn eigen chip die de drie LED lampjes aanstuurt.

Op de ring zie je 12 witte blokjes. Ieder van deze blokjes is een NeoPixel. De NeoPixels zijn allemaal aan elkaar verbonden. Met onze microcontroller geven we een signaal door aan de eerste NeoPixel. De chips van de Neopixels geven het signaal vervolgens onderling aan elkaar door.

Om al deze NeoPixels aan te sturen hebben we slechts 3 pins op onze microcontroller nodig: 5V, GND en een digitale pin. Via de digitale pin geven we door welke NeoPixels aan moeten en welke kleur ze moeten krijgen.

Om de juiste signalen door te geven aan de NeoPixels gaan we gebruik maken van een library (bibliotheek). Een library is een verzameling van code die al voor ons geschreven is. Dit maakt het een stuk makkelijker om verschillende componenten aan te sturen.

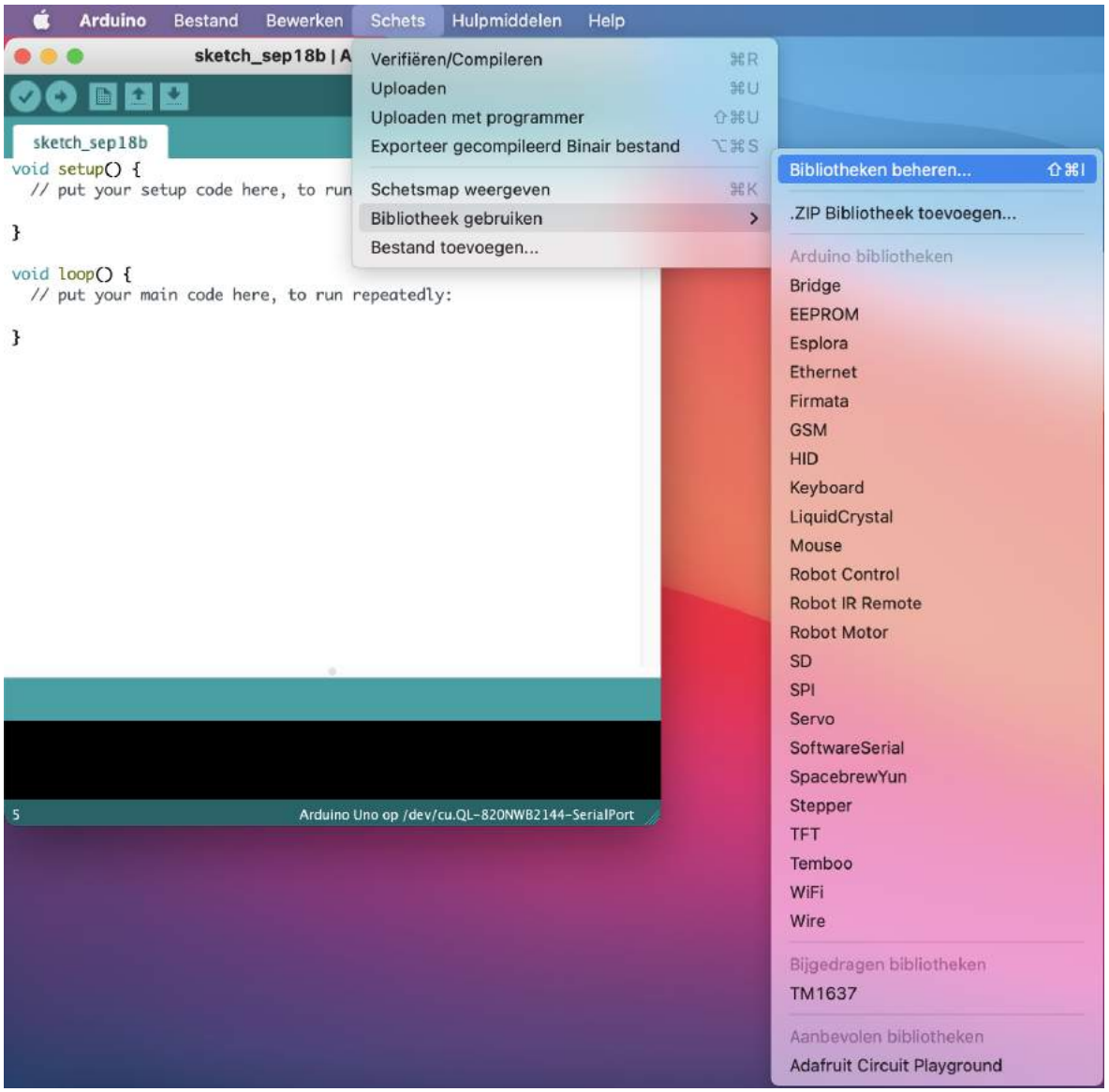
Voor onze NeoPixel ring gebruiken we de `Adafruit_NeoPixel` library.

Library installeren

Om de `Adafruit_NeoPixel` library te gebruiken moeten we deze eerst installeren in de Arduino IDE.

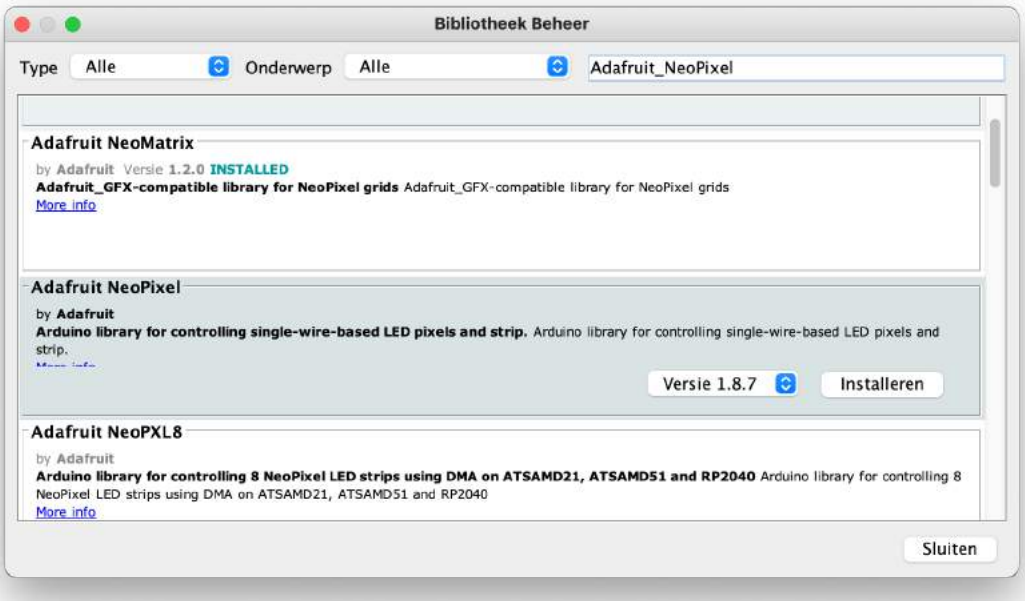
Stappen

1. Klik in de menubalk op ***Schets*** en vervolgens op ***Bibliotheek gebruiken*** en selecteer ***Bibliotheken beheren***.



Afbeelding 7.3

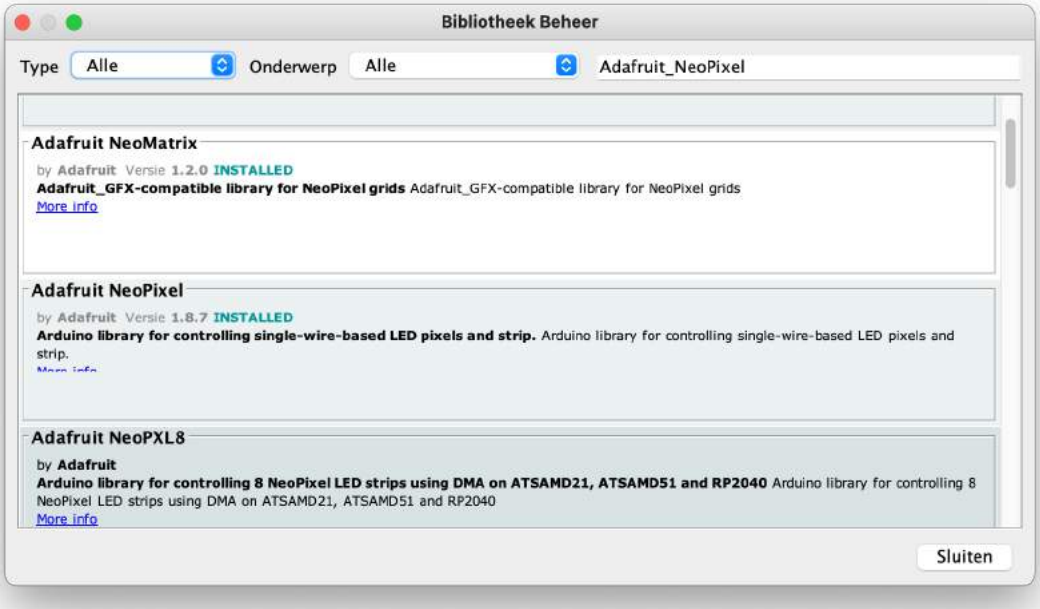
2. Er opent nu een nieuw scherm genaamd Bibliotheek Beheer. In het veld rechts boven in het scherm, waar nu "*Filter je zoekresultaten...*" staat. kan je een zoekterm invullen. Vul hier de naam van de library in die wij willen gebruiken (`Adafruit_NeoPixel`).



Afbeelding 7.4

3. Klik nu op de knop "**Installeren**" om de library te installeren.

4. Na een aantal seconden is de library geïnstalleerd. Een geïnstalleerde library herken je aan de groene tekst "INSTALLED" naast het versienummer van de library. Je kunt de Library Manager nu sluiten door op de knop "**Sluiten**" te klikken.



Afbeelding 7.5

Library gebruiken

Nu de Adafruit_NeoPixel library geïnstalleerd is kunnen we deze gebruiken in onze code. We beginnen met eenvoudige code welke de eerste drie LED's op de ring laat branden in drie verschillende kleuren.

Benodigdheden

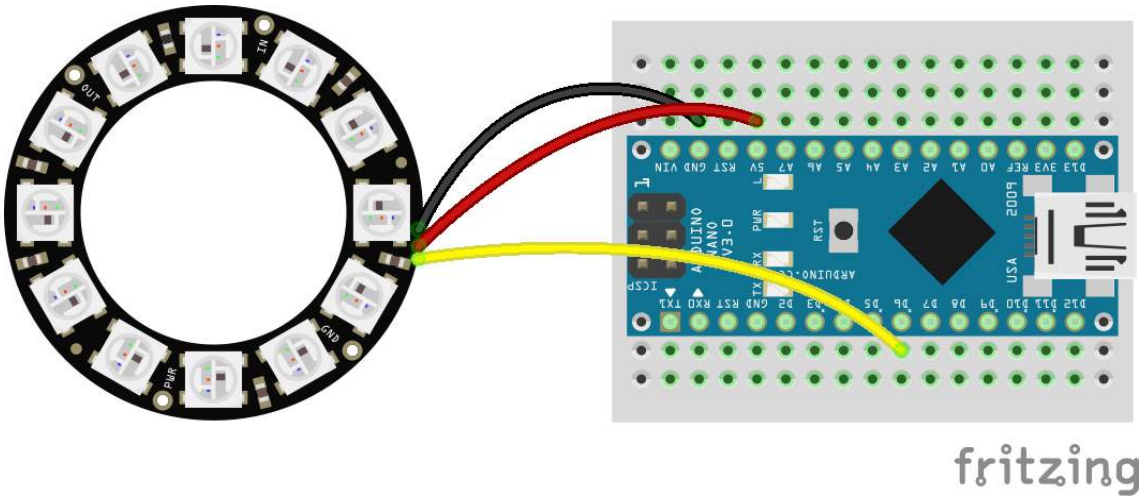
- Arduino Nano
- USB kabel
- Computer
- NeoPixel ring
- Breadboard

Verbindingen

Microcontroller	*Onderdeel*
GND	NeoPixel Ring GND (zwarte draadje)
5V	NeoPixel Ring 5V (rode draadje)
D6	NeoPixel Ring Din (gele draadje)

Stappen

1. Maak de schakeling uit afbeelding 7.1.



Afbeelding 7.1

2. Neem de onderstaande code over in de Arduino IDE.

```
#include <Adafruit_NeoPixel.h>

int aantalPixels = 12;            // Het aantal pixels op de NeoPixel ring
int pin = 6;                      // De GPIO pin waarop de NeoPixel ring aangesloten is

// Vertel de Adafruit_NeoPixel library hoeveel pixels we aan gaan sturen en met welke pin
// "NEO_GRB + NEO_KHZ800" geeft aan wat voor soort NeoPixels we gebruiken
```

```
Adafruit_NeoPixel neoPixelRing(aantalPixels, pin, NEO_GRB + NEO_KHZ800);

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  neoPixelRing.begin();           // Start de NeoPixel ring
  neoPixelRing.setBrightness(10); // Zet de helderheid van de NeoPixel ring op 10 (van de 255)
  neoPixelRing.clear();           // Zet alle pixels uit
  neoPixelRing.setPixelColor(0, 255,0,0); // Kleur pixel 0 (de eerste) rood
  neoPixelRing.setPixelColor(1, 0,255,0); // Kleur pixel 1 (de tweede) groen
  neoPixelRing.setPixelColor(2, 0,0,255); // Kleur pixel 2 (de derde) blauw
  neoPixelRing.show();            // Stuur de geupdate kleuren naar de NeoPixel ring
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
}
```

Arduino code oefening 7

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.
4. Klik op de ***Upload*** knop  om het programma te uploaden naar de microcontroller.

De eerste LED op de ring zou nu rood moeten worden, de tweede LED groen en de derde LED blauw.

Is dit niet het geval? Bekijk dan de tips onder Problemen oplossen.

Oefening 8: De for-loop

In oefening 4 hebben we geleerd wat een loop is. In deze oefening gaan we de for-loop gebruiken. In oefening 4 hebben we loops gebruikt die eeuwig duren. In de for-loop bepalen we vooraf hoe vaak de loop uitgevoerd moet worden. Vervolgens houden we met een teller bij hoe vaak de loop al is uitgevoerd. Wanneer de teller op het vooraf ingestelde aantal komt wordt de loop gestopt.

De for-loop ziet er als volgt uit:

```
for (initiele-waarde; conditie; teller-aanpassen) {
  code die uitgevoerd moet worden in de loop
}
```

Initiele waarde

Hier definiëren we de variabele die we als teller willen gebruiken. Wanneer we bijvoorbeeld de variabele "teller" willen gebruiken en bij 0 willen beginnen ziet dat er als volgt uit:

```
for (int teller=0; conditie; teller-aanpassen) {
  code die uitgevoerd moet worden in de loop
}
```

Conditie

Hier geven we aan tegen welke voorwaarde de loop door mag blijven gaan. Wanneer de uitkomst van deze vergelijking waar (TRUE) is dan wordt de loop nog een keer uitgevoerd. In onderstaand voorbeeld blijft de loop doorlopen zolang de variabele "teller" kleiner is dan 16. In het totaal wordt de code in de loop dus 16 keer uitgevoerd. We beginnen namelijk bij 0.

```
for (int teller=0; teller < 16; teller-aanpassen) {
  code die uitgevoerd moet worden in de loop
}
```

Teller-aanpassen

Iedere keer dat de loop wordt uitgevoerd passen we aan het einde van de loop de teller aan. In veel gevallen willen we de teller met 1 verhogen. Dit kan als volgt:

```
for (int teller=0; teller < 16; teller = teller + 1) {
  code die uitgevoerd moet worden in de loop
}
```

Er bestaat ook een kortere notering die precies hetzelfde doet:

```
for (int teller=0; teller < 16; teller++) {
  code die uitgevoerd moet worden in de loop
}
```

De teller hoeft niet op te tellen maar kan bijvoorbeeld ook verminderen:

```
for (int teller=0; teller < 16; teller--) {
  code die uitgevoerd moet worden in de loop
}
```

De teller kan ook in grotere stappen dan 1 optellen of verminderen:

```
for (int teller=0; teller < 16; teller = teller + 2) {
  code die uitgevoerd moet worden in de loop
}
```

De for-loop gebruiken

We gaan de for-loop nu gebruiken om al de LED's van de NeoPixel ring aan te zetten.

Benodigdheden

- Schakeling uit oefening 7
- USB kabel
- Computer

Stappen

1. Maak de schakeling die we in oefening 7 gebruikt hebben.

2. Neem de onderstaande code over in de Arduino IDE.

```
#include <Adafruit_NeoPixel.h>

int aantalPixels = 12;            // Het aantal pixels op de NeoPixel ring
int pin = 6;                      // De GPIO pin waarop de NeoPixel ring aangesloten is

// Vertel de Adafruit_NeoPixel library hoeveel pixels we aan gaan sturen en met welke pin
// "NEO_GRB + NEO_KHZ800" geeft aan wat voor soort NeoPixels we gebruiken
Adafruit_NeoPixel neoPixelRing(aantalPixels, pin, NEO_GRB + NEO_KHZ800);

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  neoPixelRing.begin();           // Start de NeoPixel ring
  neoPixelRing.setBrightness(10); // Zet de helderheid van de NeoPixel ring op 10 (van de 255)
  neoPixelRing.clear();           // Zet alle pixels uit
```

```
  // Voer onderstaande loop uit zolang teller kleiner is dan aantalPixels (16)
  for(int teller=0; teller < aantalPixels; teller++){
    neoPixelRing.setPixelColor(teller, 255,0,0); // Kleur pixel rood
    neoPixelRing.show();          // Stuur de geupdate kleuren naar de NeoPixel ring
    delay(600);                   // Wacht 600 milliseconden
  }

}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
}
```

Arduino code oefening 8

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.
4. Klik op de ***Upload*** knop  om het programma te uploaden naar de microcontroller.

De LED's zouden nu één voor één rood moeten worden.

Is dit niet het geval? Bekijk dan de tips onder Problemen oplossen.

Oefening 9: Gloeien

In deze oefening gaan we de for-loop die we in oefening 8 gebruikt hebben inzetten om de NeoPixel ring te laten gloeien. We doen dit door een extra for-loop de felheid van de LED's te laten bepalen.

Benodigdheden

- Schakeling uit oefening 7
- USB kabel
- Computer

Stappen

1. Maak de schakeling die we in oefening 7 gebruikt hebben.

2. Neem de onderstaande code over in de Arduino IDE.

```
#include <Adafruit_NeoPixel.h>

int aantalPixels = 12;            // Het aantal pixels op de NeoPixel ring
int pin = 6;                      // De GPIO pin waarop de NeoPixel ring aangesloten is

// Vertel de Adafruit_NeoPixel library hoeveel pixels we aan gaan sturen en met welke pin
// "NEO_GRB + NEO_KHZ800" geeft aan wat voor soort NeoPixels we gebruiken
Adafruit_NeoPixel neoPixelRing(aantalPixels, pin, NEO_GRB + NEO_KHZ800);

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  neoPixelRing.begin();           // Start de NeoPixel ring
  neoPixelRing.setBrightness(70); // Zet de helderheid van de NeoPixel ring op 10 (van de 255)
  neoPixelRing.clear();           // Zet alle pixels uit
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  // Voer onderstaande loop uit zolang de teller (felheid) kleiner of gelijk is aan 255
  // Aan het einde iedere "ronde" tellen steeds 1 bij de teller (felheid) op.
  for (int felheid = 0; felheid <= 255; felheid++){
    // Voer onderstaande loop uit zolang teller kleiner is dan aantalPixels (16)
    for (int teller = 0; teller < aantalPixels; teller++) {
      neoPixelRing.setPixelColor(teller, 0, felheid, 0); // Kleur pixel groen
    }
    delay(20);                    // Wacht 20 milliseconden
    neoPixelRing.show();          // Stuur de geupdate kleuren naar de NeoPixel ring
  }

  // Voer onderstaande loop uit zolang de teller (felheid) groter of gelijk is aan 0
  // Aan het einde van iedere "ronde" trekken we steeds 1 van de teller (felheid) af.
  for (int felheid = 255; felheid >= 0; felheid--){
    // Voer onderstaande loop uit zolang teller kleiner is dan aantalPixels (16)
    for (int teller = 0; teller < aantalPixels; teller++) {
      neoPixelRing.setPixelColor(teller, 0, felheid, 0); // Kleur pixel groen
    }
    delay(20);                    // Wacht 20 milliseconden
    neoPixelRing.show();          // Stuur de geupdate kleuren naar de NeoPixel ring
  }
}
```

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.
4. Klik op de ***Upload*** knop  om het programma te uploaden naar de microcontroller.

De LED's zouden nu allemaal tegelijk langzaam meer en minder fel moeten worden.

Is dit niet het geval? Bekijk dan de tips onder Problemen oplossen.

Oefening 10: Kerstboom in elkaar zetten

In deze oefening gaan we de NeoPixel ring in de kerstboom plaatsen. Daarvoor moeten we de NeoPixel ring eerst loskoppelen van het breadboard. Vergeet hierbij niet om eerst de stroombron los te koppelen van de microcontroller.

Wanneer de NeoPixel ring is losgekoppeld kan je onderstaande stappen volgen om deze in de kerstboom te installeren.

Benodigdheden:

- NeoPixel ring
- Plastic onderdelen

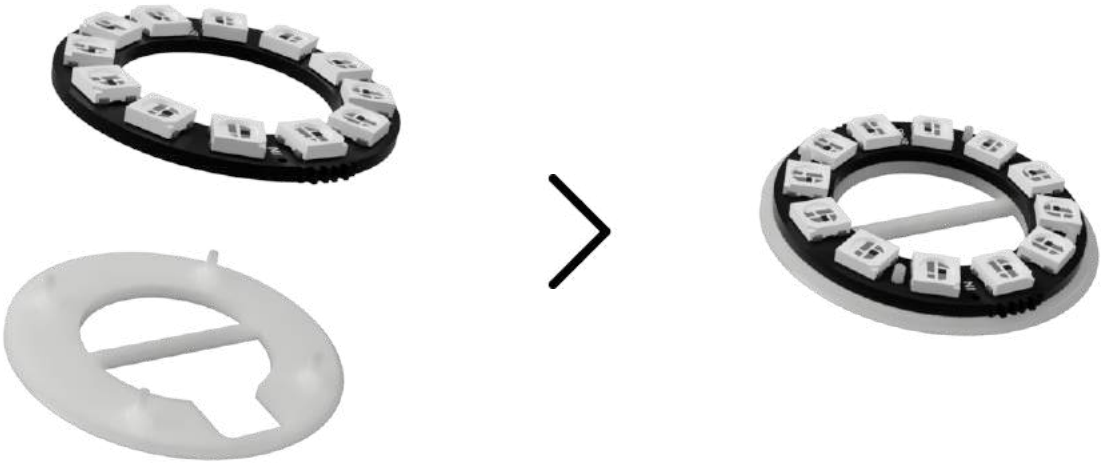
Stappen:

1. Koppel de NeoPixel ring los van de microcontroller.

2. Volg de instructies in de afbeeldingen hieronder om de kerstboom in elkaar te zetten.

   De draden die aan de NeoPixel ring vastzitten zijn in deze afbeeldingen niet weergegeven.

   Voor de plek waar de draden aan de NeoPixel ring gesoldeerd zijn is een uitsparing gemaakt in de plaat waar de NeoPixel ring op komt. De draden dienen door de uitsparing heen gestoken te worden.



3. Verbind de NeoPixel ring weer met de microcontroller zoals we dat in oefening 7 gedaan hebben.

Oefening 11: Verschillende effecten

Nu de kerstboom in elkaar staat is het tijd om een aantal effecten uit te proberen. We gaan daarvoor de technieken gebruiken die we tot nu toe geleerd hebben.

In deze oefening geven we drie codevoorbeelden om leuke effecten te maken. Uiteraard kan je de deze voorbeelden aanpassen of de technieken die je geleerd hebt ook gebruiken om je eigen effecten te maken.

Benodigdheden

- Schakeling uit oefening 7
- USB kabel
- Computer

Voorbeeld 1: Individueel gloeien

Met deze code gaan de LED's één voor één gloeien.

1. Maak de schakeling die we in oefening 7 gebruikt hebben.

2. Neem de onderstaande code over in de Arduino IDE.

```
#include <Adafruit_NeoPixel.h>

int aantalPixels = 12;            // Het aantal pixels op de NeoPixel ring
int pin = 6;                      // De GPIO pin waarop de NeoPixel ring aangesloten is

// Vertel de Adafruit_NeoPixel library hoeveel pixels we aan gaan sturen en met welke pin
// "NEO_GRB + NEO_KHZ800" geeft aan wat voor soort NeoPixels we gebruiken
Adafruit_NeoPixel neoPixelRing(aantalPixels, pin, NEO_GRB + NEO_KHZ800);

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  neoPixelRing.begin();           // Start de NeoPixel ring
  neoPixelRing.setBrightness(150); // Zet de helderheid van de NeoPixel ring op 150 (van de 255)
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  neoPixelRing.clear();           // Zet alle pixels uit
  // Voer onderstaande loop uit zolang teller kleiner is dan aantalPixels (16)
  for (int teller = 0; teller < aantalPixels; teller++) {
    for (int helderheid = 0; helderheid <= 255; helderheid++) {
      neoPixelRing.setPixelColor(teller, 0, 0, helderheid); // Kleur pixel blauw
      delay(5);                   // Wacht 5 milliseconden
      neoPixelRing.show();        // Stuur de geupdate kleuren naar de NeoPixel ring
    }
  }
}
```

Arduino code oefening 11 - Voorbeeld 1

3. Zorg ervoor dat de microcontroller verbonden is met de

computer, dat je het juiste board en de juiste poort geselecteerd hebt.

4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Voorbeeld 2: Rondjes draaien

Met deze code zetten we steeds één LED aan waardoor het lijkt alsof een LED een rondje draait.

1. Maak de schakeling die we in oefening 7 gebruikt hebben.

2. Neem de onderstaande code over in de Arduino IDE.

```
#include <Adafruit_NeoPixel.h>

int aantalPixels = 12;            // Het aantal pixels op de NeoPixel ring
int pin = 6;                      // De GPIO pin waarop de NeoPixel ring aangesloten is

// Vertel de Adafruit_NeoPixel library hoeveel pixels we aan gaan sturen en met welke pin
// "NEO_GRB + NEO_KHZ800" geeft aan wat voor soort NeoPixels we gebruiken
Adafruit_NeoPixel neoPixelRing(aantalPixels, pin, NEO_GRB + NEO_KHZ800);

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  neoPixelRing.begin();           // Start de NeoPixel ring
  neoPixelRing.setBrightness(150); // Zet de helderheid van de NeoPixel ring op 150 (van de 255)
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  // Voer onderstaande loop uit zolang teller kleiner is dan aantalPixels (16)
  for (int teller = 0; teller < aantalPixels; teller++) {
    neoPixelRing.clear();         // Zet alle pixels uit
    neoPixelRing.setPixelColor(teller, 125, 50, 200); // Kleur pixel paars
    delay(100);                   // Wacht 100 milliseconden
    neoPixelRing.show();          // Stuur de geupdate kleuren naar de NeoPixel ring
  }
}
```

Arduino code oefening 11 - Voorbeeld 2

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.

4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Voorbeeld 3: Ping pong

Net als in voorbeeld 2 laten we de led een rondje draaien. Wanneert er een rondje gemaakt is laten we de LED's in de tegenovergestelde richting draaien.

1. Maak de schakeling die we in oefening 7 gebruikt hebben.

2. Neem de onderstaande code over in de Arduino IDE.

```
#include <Adafruit_NeoPixel.h>

int aantalPixels = 12;            // Het aantal pixels op de NeoPixel ring
int pin = 6;                      // De GPIO pin waarop de NeoPixel ring aangesloten is

// Vertel de Adafruit_NeoPixel library hoeveel pixels we aan gaan sturen en met welke pin
// "NEO_GRB + NEO_KHZ800" geeft aan wat voor soort NeoPixels we gebruiken
Adafruit_NeoPixel neoPixelRing(aantalPixels, pin, NEO_GRB + NEO_KHZ800);

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de microcontroller wordt ingeschakeld
void setup() {
  neoPixelRing.begin();           // Start de NeoPixel ring
  neoPixelRing.setBrightness(150); // Zet de helderheid van de NeoPixel ring op 150 (van de 255)
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  // Voer onderstaande loop uit zolang teller kleiner is dan aantalPixels (16)
  for (int teller = 0; teller < aantalPixels; teller++) {
    neoPixelRing.clear();         // Zet alle pixels uit
    neoPixelRing.setPixelColor(teller, 50, 168, 156); // Kleur pixel cyaan
    delay(50);                    // Wacht 50 milliseconden
    neoPixelRing.show();          // Stuur de geupdate kleuren naar de NeoPixel ring
  }
  // Voer onderstaande loop uit zolang teller groter is dan 0
  for (int teller = aantalPixels - 2; teller > 0; teller--) {
    neoPixelRing.clear();         // Zet alle pixels uit
    neoPixelRing.setPixelColor(teller, 50, 168, 137); // Kleur pixel cyaan
    delay(50);                    // Wacht 50 milliseconden
    neoPixelRing.show();          // Stuur de geupdate kleuren naar de NeoPixel ring
  }
}
```

Arduino code oefening 11 - Voorbeeld 2

3. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.

4. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Oefening 12: Voorgeprogrammeerd voorbeeld gebruiken

Tijdens de installatie van de Adafruit_NeoPixel library zijn er ook direct een aantal voorgeprogrammeerde voorbeelden geïnstalleerd. In deze oefening gaan we deze voorbeelden met onze kerstboom gebruiken.

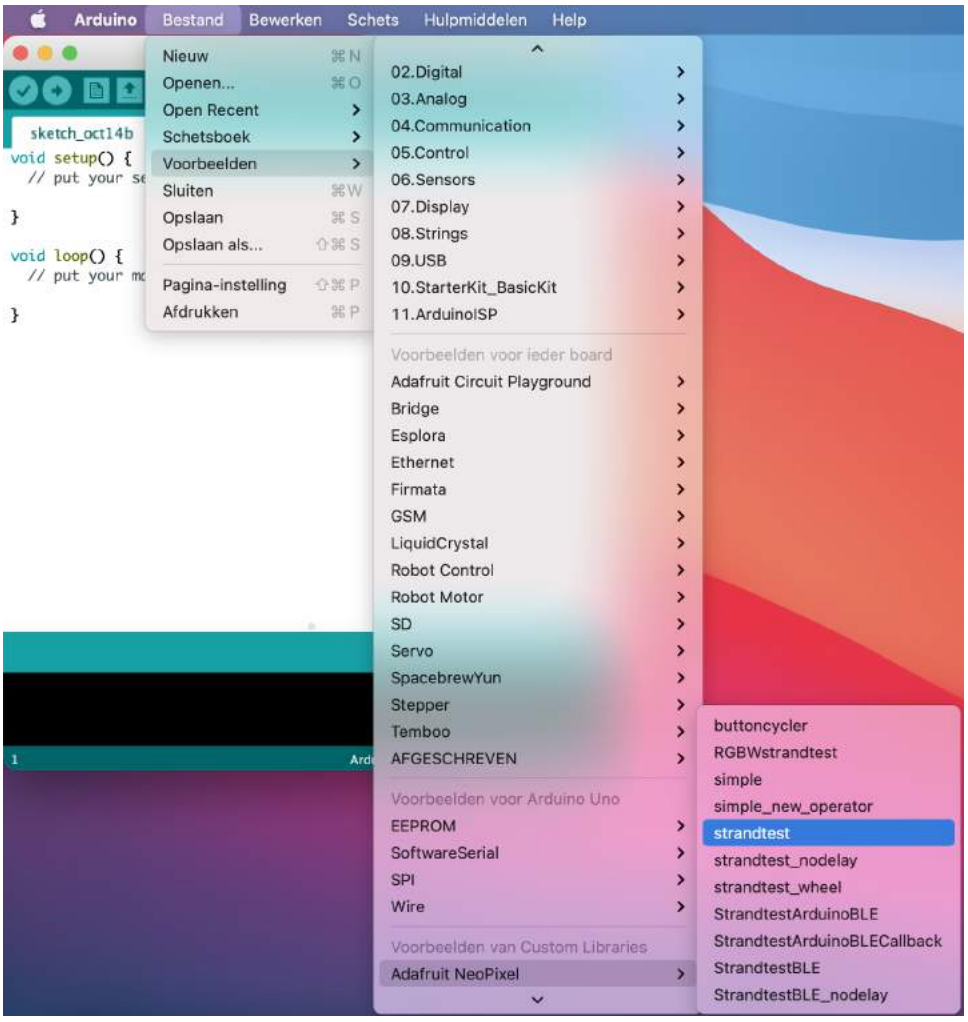
Benodigdheden

- Schakeling uit oefening 7
- USB kabel
- Computer

Stappen

1. Maak de schakeling die we in oefening 7 gebruikt hebben.

2. Klik in de menubalk op ***Bestand*** en vervolgens op ***Voorbeeld***, dan op ***Adafruit Neopixel*** en selecteer ***strandtest***.



Afbeelding 12.1

Er opent nu een nieuw scherm met daarin de code van het voorgeprogrammeerde voorbeeld.

3. We moeten nog wel een wijziging in de code doen voordat deze met onze kerstboom werkt. Zoek in de code de volgende regel:

```
#define LED_COUNT 60
```

hier wordt bepaald hoeveel LED's de code moet aansturen. In ons geval is dat 12. Verander de code naar het volgende:

```
#define LED_COUNT 12
```

4. Zorg ervoor dat de microcontroller verbonden is met de computer, dat je het juiste board en de juiste poort geselecteerd hebt.
5. Klik op de ***Uploaden*** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan krijg je nu diverse effecten te zien. Probeer de code eens wat aan te passen om te zien of je zelf nog nieuwe effecten kunt creëren.

Problemen oplossen

Mijn computer kan geen verbinding maken met de microcontroller

Lukt het niet om te verbinden met de microcontroller? Dan kan het zijn dat je eerst nog de CH340 driver moet installeren. Ga hiervoor naar https://www.awesomemakes.com/drivers/ en download en installeer de driver voor jouw besturingssysteem.

Het verifiëren van mijn code is mislukt

- Doorloop in dit geval de stappen voor het kiezen van het bord en het aansluiten van de microcontroller nogmaals om te zien of je daar iets vergeten bent. Upload het programma daarna opnieuw.

- Heb je de code goed gekopieerd? Als je wijzigingen hebt gedaan kan het zijn dat je bijvoorbeeld een ";" bent vergeten. Begin eerst met het exact kopiëren van onze code. Als je daarna aanpassingen wilt doen, doe dit dan één voor één en controleer na iedere aanpassing of de code nog werkt.

- Gaat het uploaden van de code naar de microcontroller niet goed? Probeer eens een andere poort te selecteren, nog een keer te uploaden, weer terug te switchen naar de juiste poort en nog een keer te uploaden.

- Gaat het uploaden van de code fout en zie je in het rood de melding: “programmer is not responding”. Druk dan op de reset knop op de microcontroller vlak voordat je op upload klikt. Helpt dit nog niet? Haal de microcontroller dan van de stroom, sluit hem weer aan en probeer opnieuw te uploaden.

Mijn code is geupload maar de microcontroller doet niet wat ik verwacht.

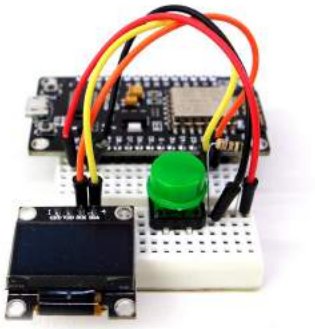
- Het kan helpen om de microcontroller te resetten. Dit doen je door op de reset knop te drukken (zie afbeelding 0.1).

Gefeliciteerd!

Nu heb je een zelfgemaakte lichtgevende kerstboom. Je kan de kerstboom weer uit elkaar halen en de NeoPixel ring verwerken in één van jouw eigen projecten. Ook is het mogelijk om allerlei andere NeoPixel producten aan te sturen zoals een NeoPixel LED Strip of een grotere ring.

Meer bouwen?

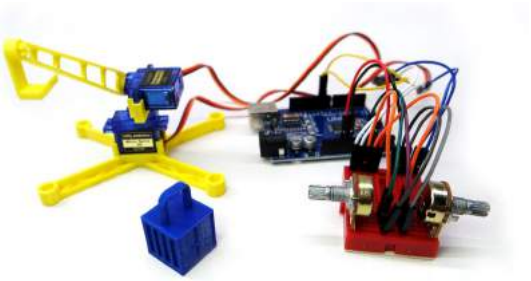
Heb je de smaak te pakken en wil je meer bouwen? Bekijk dan eens de andere Awesome Crates op www.awesomemakes.com



IOT Smart Display

IOT Mood Light

Countdown Timer



Bestuurbare Hijskraan