



# Handleiding

IOT Mood Light



# Inhoudsopgave

Welkom	1
De microcontroller	3
Arduino IDE installeren	4
Arduino IDE installeren op Mac	5
Arduino IDE installeren op Windows	6
Arduino IDE configureren	9
Het juiste bord kiezen	11
De microcontroller op de computer aansluiten	12
Oefening 1: Laat een LED knipperen	17
Het breadboard	19
Oefening 2: Een LED aansluiten	20
Oefening 3: Functies	22
Oefening 4: de "loop" gebruiken	24
Oefening 5: variabelen	26
Oefening 6: LED dimmen	29
Oefening 7: Analog Write	32
Oefening 8: Meerdere LED's aansturen	34
Oefening 9: De lampion bevestigen	41
Oefening 10: Kleuren maken	44
Oefening 10: De seriële verbinding	46
Oefening 11: Wifi	49
Oefening 12: Aansturen via wifi	51
Oefening 13: Webpage met parameters	55
Oefening 14: IOT Mood Light bouwen	58
Problemen oplossen	61
Gefeliciteerd!	62
Meer bouwen?	62



# Welkom

Wat leuk dat je een Awesome Crate gaat bouwen!

Deze crate bevat de benodigdheden en uitleg om een lamp te bouwen waarvan je zelf de kleur kunt bepalen.

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

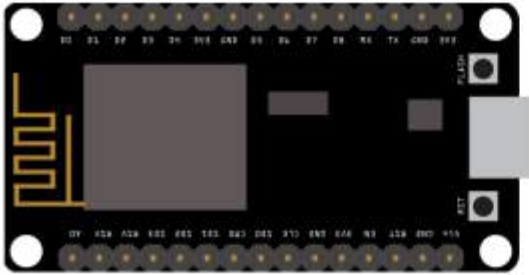
Om zo veel mogelijk plezier te beleven zijn hier een paar regels die ervoor zorgen dat de microcontroller niet beschadigt:

- Raak de microcontroller niet aan terwijl deze aan is.
- Zorg er altijd voor dat de microcontroller uit is (stroomkabeltje niet verbonden) voordat je onderdelen inpluigt.
- 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.

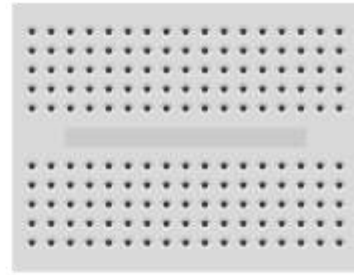


Dit is geen speelgoed. Bij het gebruik van dit artikel dienen kinderen onder de 14 jaar altijd begeleid te worden door een volwassene. Houd altijd toezicht op het product tijdens gebruik. Ontkoppel een eventuele stroombron na gebruik.

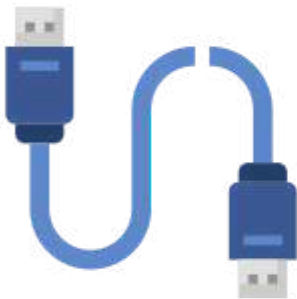
Microcontroller



Breadboard



USB kabel



Weerstandjes  
51  $\Omega$



LED lampjes



Lampion



Lampionbeneden



Lampionhouder

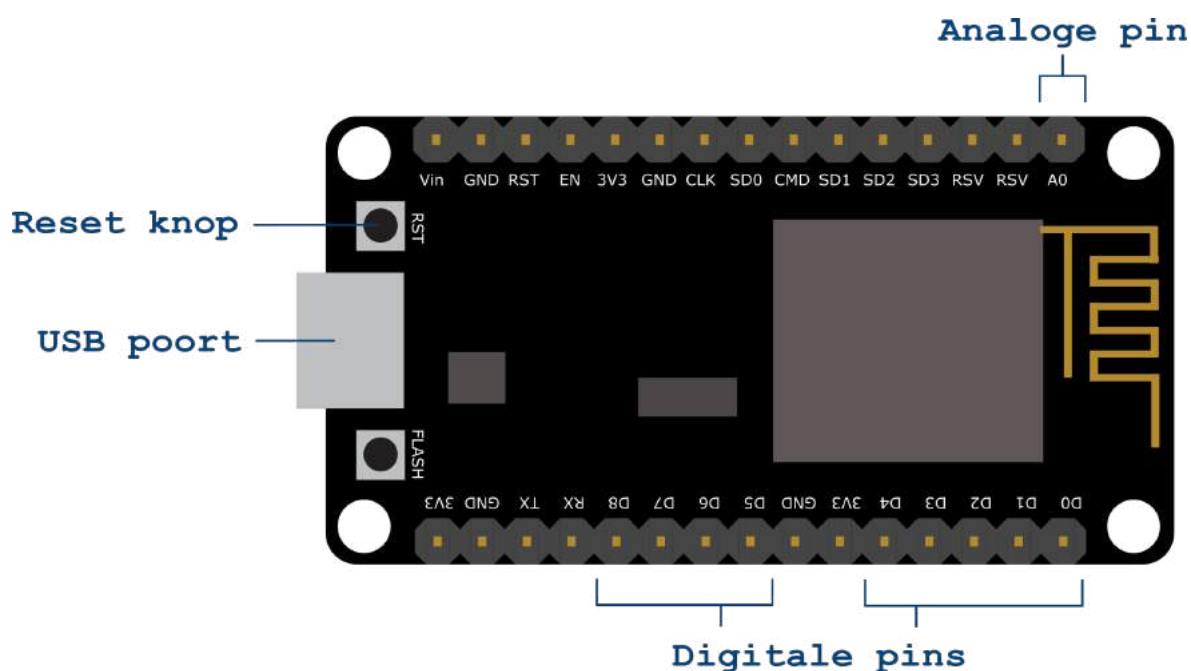


# De microcontroller

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 NodeMCU. Deze microcontroller heeft ingebouwde wifi waardoor hij heel geschikt is voor IOT projecten.



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](#) (blz x.)

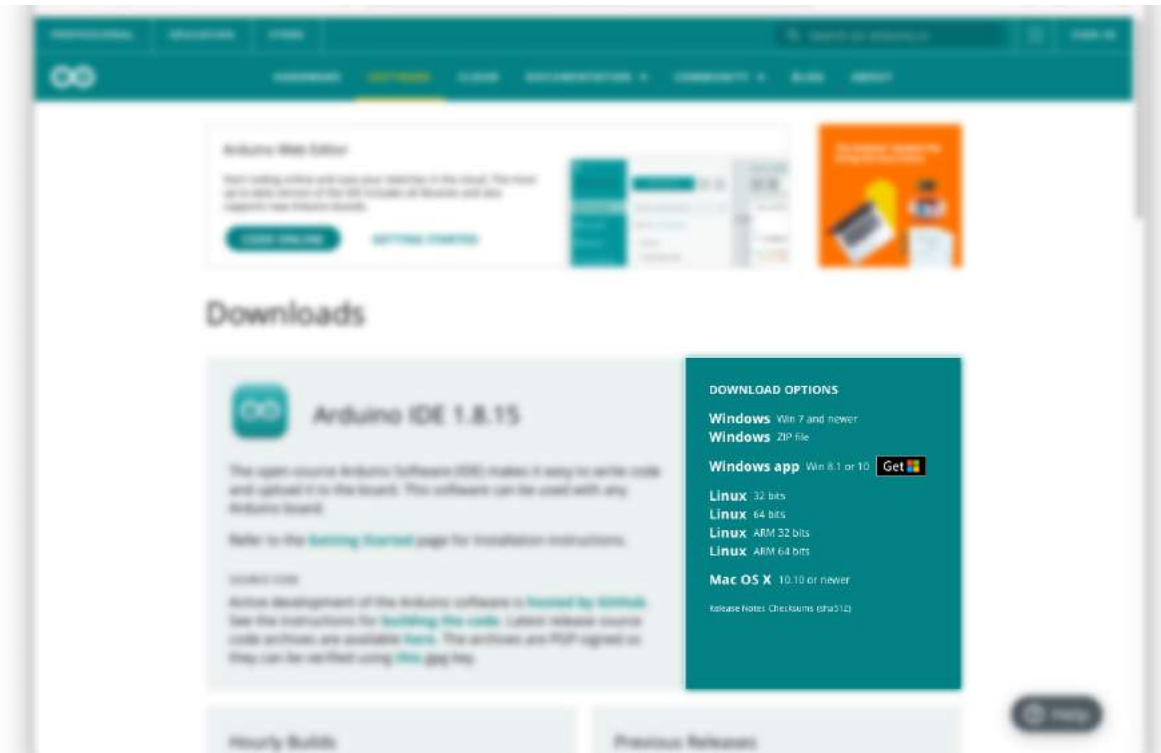
## **Gebruik je Windows?**

Ga dan verder bij [Arduino IDE installeren op Windows](#) (blz x.)

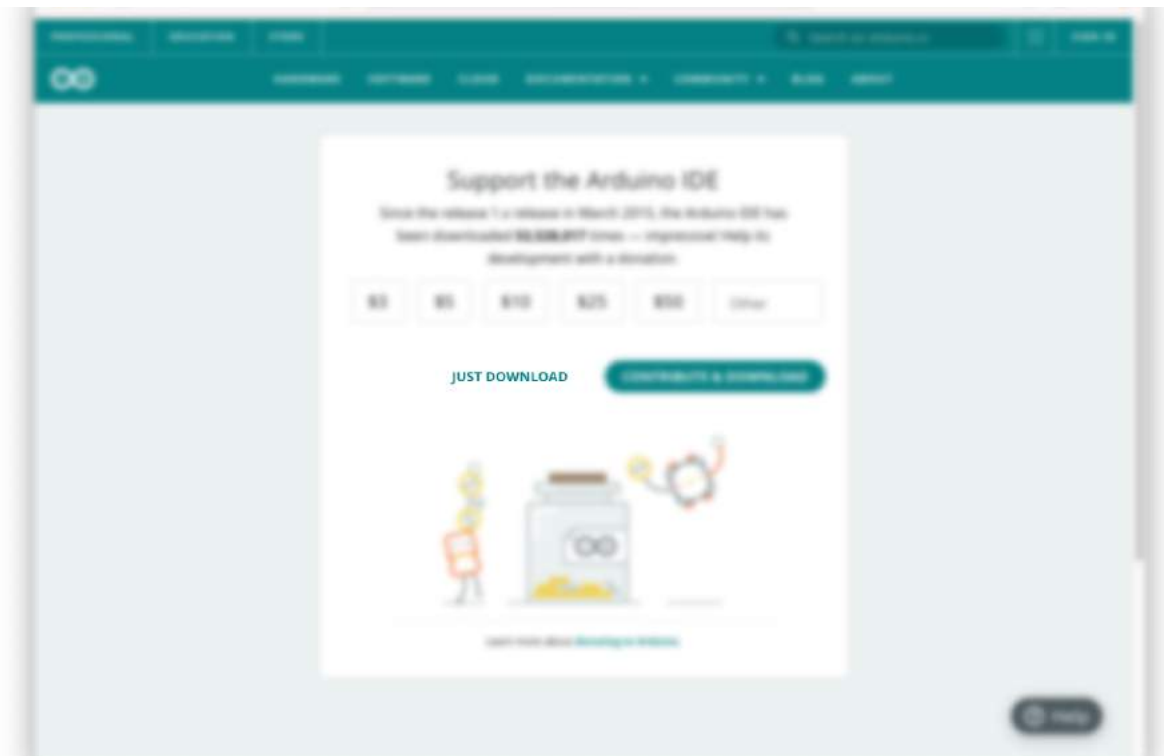
# Arduino IDE installeren op Mac

## 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".



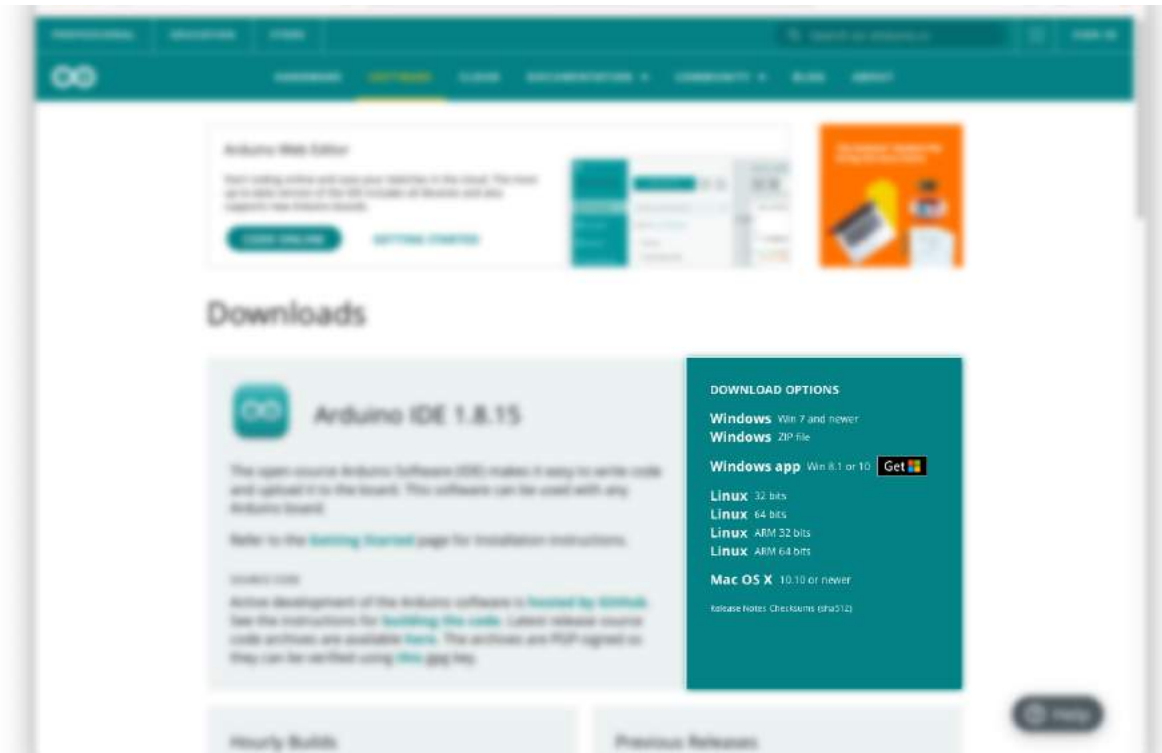
4. De Arduino IDE wordt nu gedownload. Verplaats het bestand daarna naar de applicatie map.
5. Open het programma. Het kan zijn dat jouw besturingssysteem nu om toestemming vraagt om de applicatie te openen. Geef in dit geval toestemming hiervoor.
6. De Arduino IDE wordt nu geopend.



# 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.

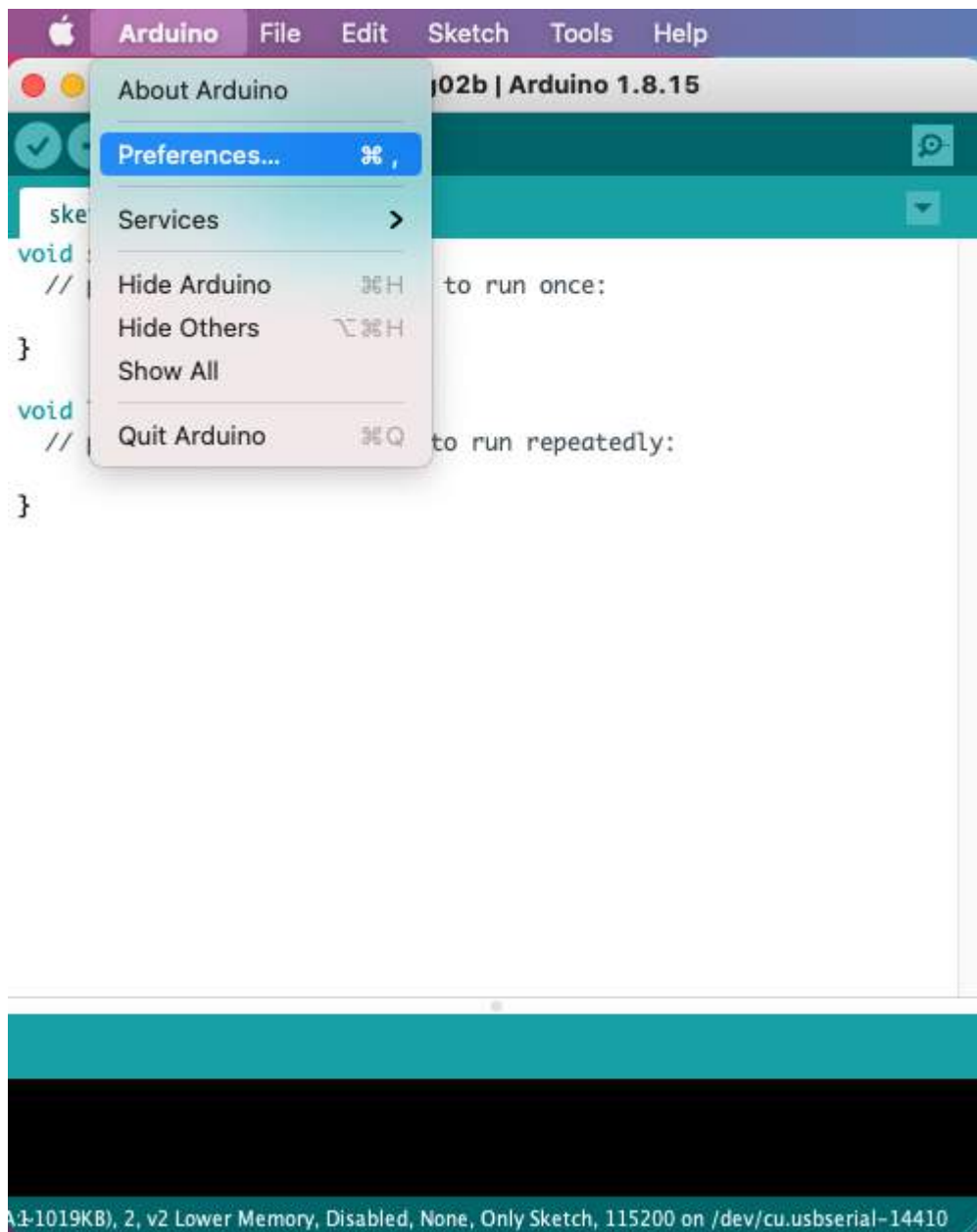
# Arduino IDE configureren

De Arduino IDE is standaard ingesteld voor het gebruik van Arduino Microcontrollers. Wij gaan een NodeMCU gebruiken, hiervoor moet de Arduino IDE eerst geconfigureerd worden. Maak je geen zorgen, dit zijn maar een paar simpele stappen!

## Stappen

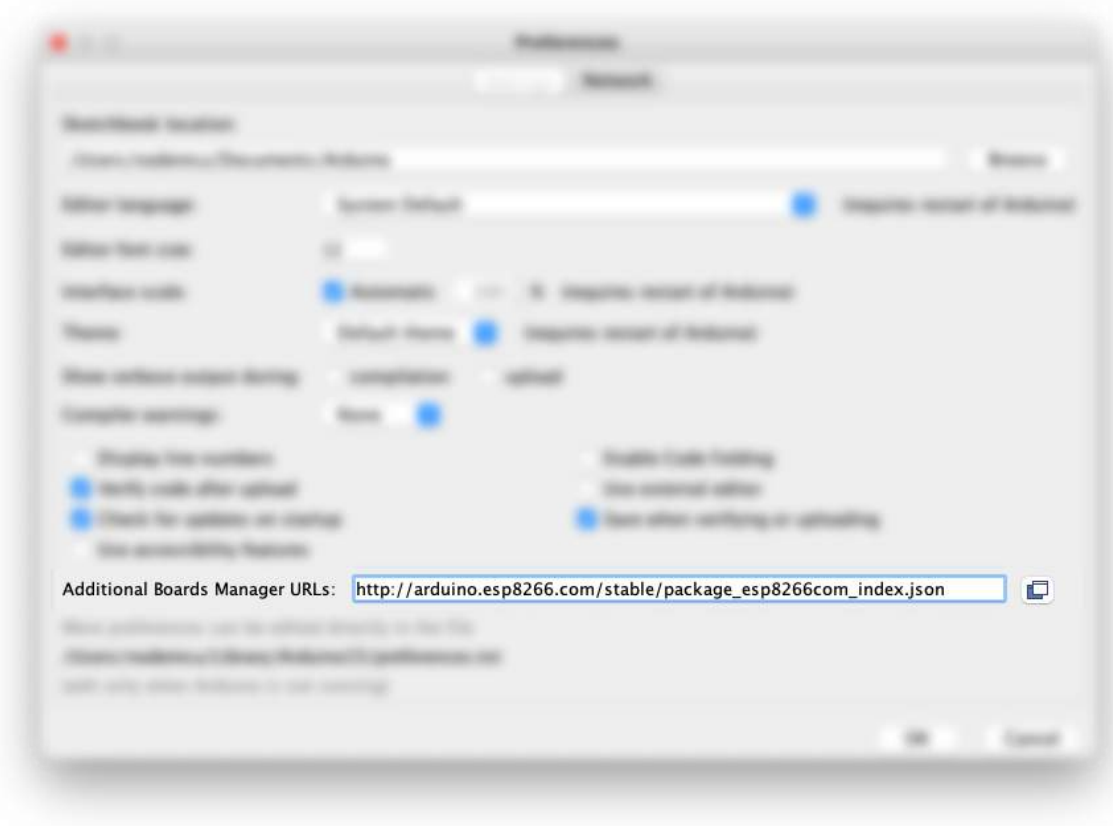
Eerst moeten we de NodeMCU installeren binnen de Arduino IDE.

1. Klik in de menubalk op **Arduino** en vervolgens op **Preferences...**



2. Plak in het veld **Additional Boards Manager URLs** de volgende url:

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



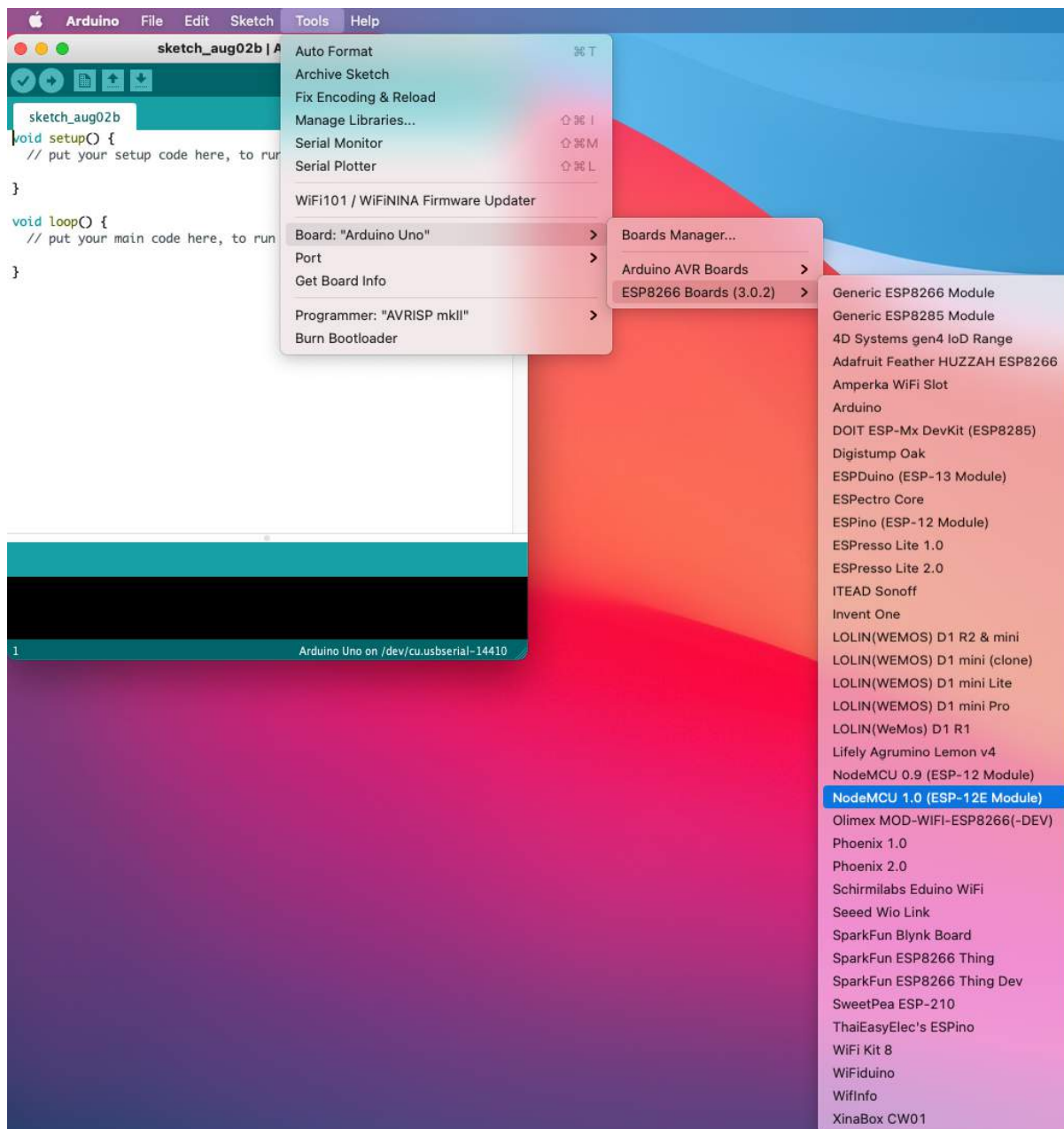
3. Klik op de knop **OK**.

# Het juiste bord kiezen

Nu de NodeMCU geïnstalleerd is kunnen we deze selecteren als de microcontroller waar we onze code naar toe willen sturen.

## Stappen

1. Klik in de menubalk op **Tools** en vervolgens op **Board**, dan op **ESP8266 Boards** en selecteer **NodeMCU 1.0 (ESP-12E Module)**.



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

# De microcontroller op de computer aansluiten

De Arduino IDE weet nu hoe hij overweg moet met het NodeMCU bord. 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

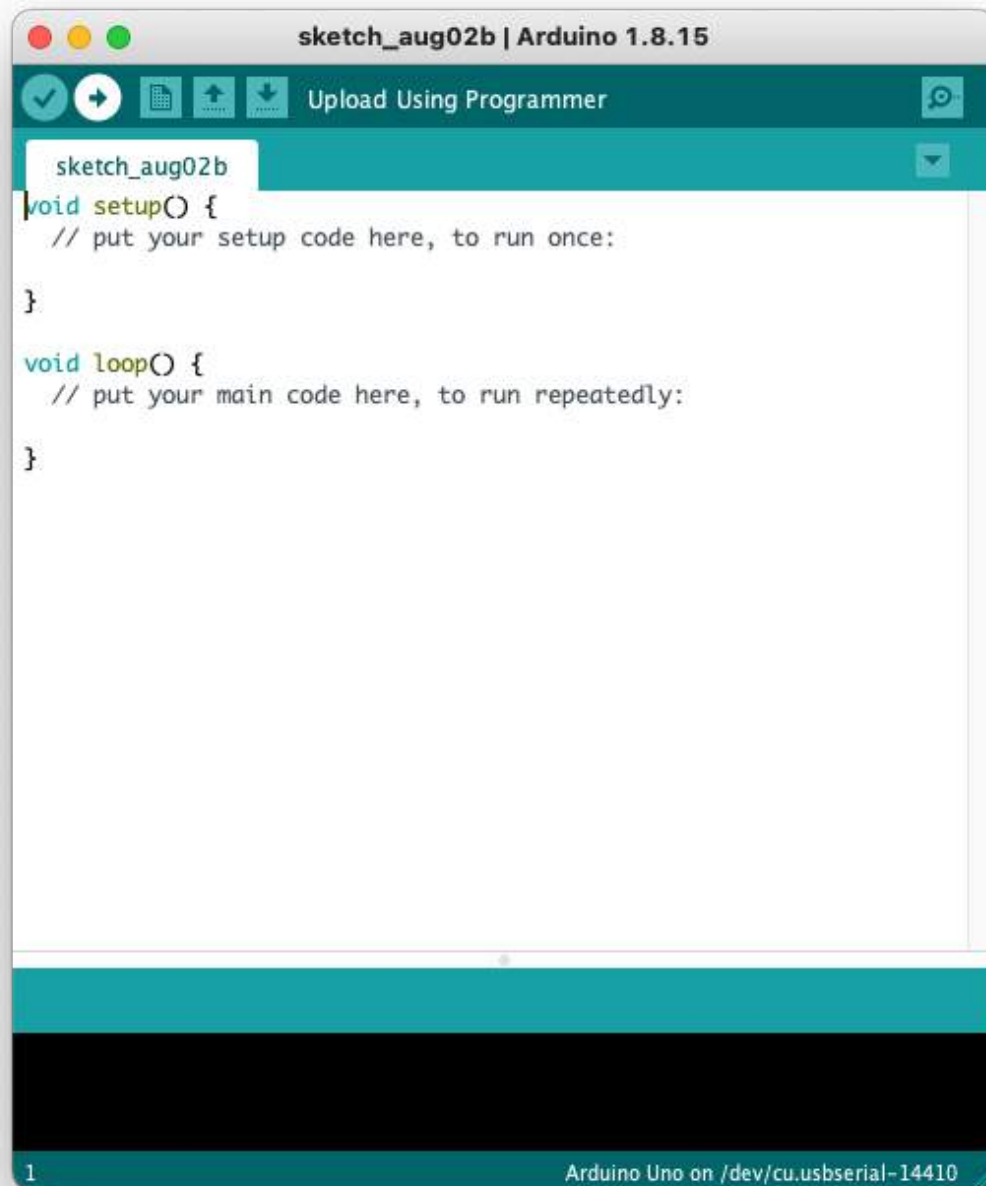
Het is belangrijk om bij het aansluiten van de microcontroller de [veiligheidsregels](#) (blz. x) na te leven.

1. Sluit de USB kabel eerst aan op de microcontroller en vervolgens op je computer.
2. Klik in de menubalk op **Tools** en vervolgens op **Port**. 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).

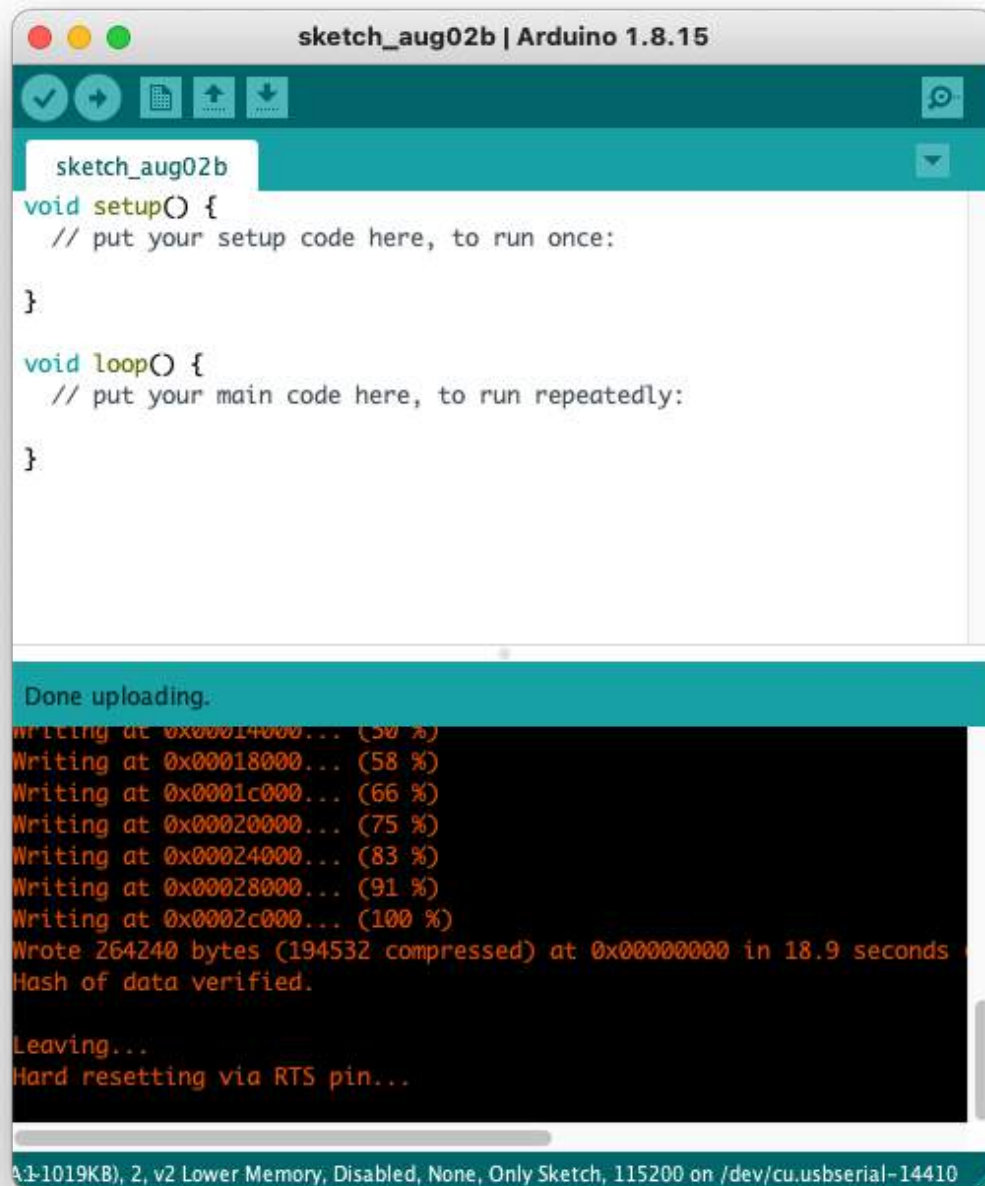
Weet je niet welke poort je moet gebruiken? Kijk dan welke poorten er nu in de lijst staan. Sluit het menu, ontkoppel de microcontroller en kijk opnieuw welke poorten er in de lijst staan. De poort die niet langer in de lijst staat is de poort van de microcontroller.

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

3. Klik op de *Upload* knop .

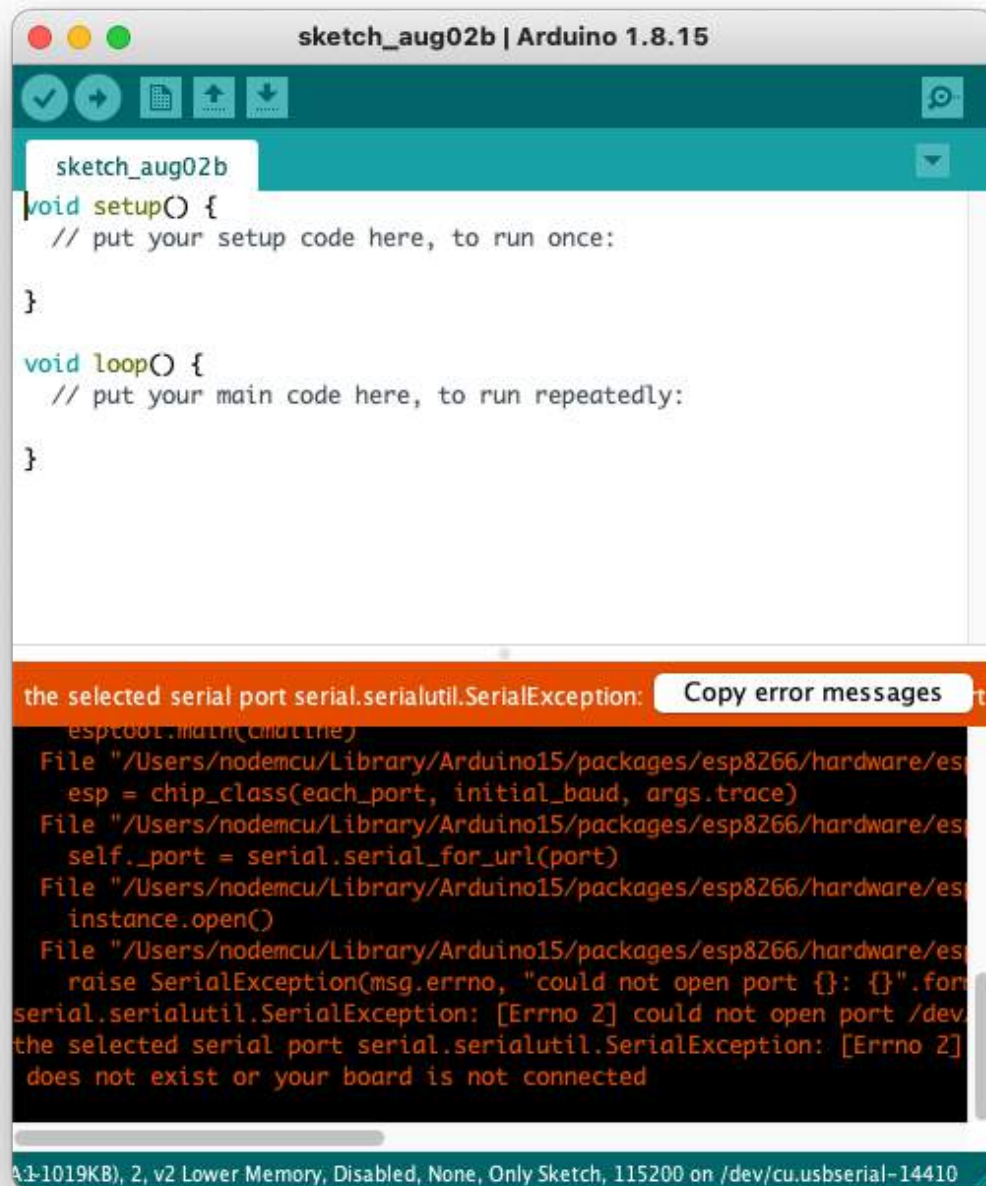


4. In het zwarte vlak onder in het scherm zie je nu een aantal berichten verschijnen. In de groene balk daarboven (statusbalk) zie je de status van het uploaden. Staat er in de groene balk "*Done uploading*"? Dan is het uploaden geslaagd.





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 \(blz. x\)](#) en [het aansluiten van de microcontroller \(blz. x\)](#) nogmaals om te zien of je daar iets vergeten bent. Upload het programma daarna opnieuw.



# Oefening 1: Laat een LED knipperen

De NodeMCU 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 D4 te zetten. De interne LED is namelijk direct verbonden aan pin D4.

De ingebouwde LED is zo aangesloten dat deze aan gaat wanneer er een laag signaal op pin D4 gezet wordt. De LED gaat uit wanneer er een hoog signaal op de pin gezet wordt. We noemen dit active low aangezien de LED actief wordt bij een laag signaal.

## Benodigheden

- NodeMCU
- 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(D4, OUTPUT);    // We gaan pin D4 gebruiken als output

  digitalWrite(D4, LOW);  // Zet de LED aan door een laag signaal op D4 zetten
  delay(1000);            // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D4, HIGH); // Zet de LED uit door een hoog signaal op D4 te zetten
  delay(1000);            // Wacht 1 seconde (1000 milliseconden)

  digitalWrite(D4, LOW);  // Zet de LED aan door een laag signaal op D4 zetten
  delay(1000);            // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D4, HIGH); // Zet de LED uit door een hoog signaal op D4 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](#) (blz. x) met de computer, dat je [het juiste board](#) (blz. x) en [de juiste poort](#) (blz. x) geselecteerd hebt.

3. Klik op de **Verify** knop . De Arduino IDE controleert nu of de code die je hebt ingevoerd juist is. Wanneer alles goed gaat verschijnt de tekst "*Done compiling.*" 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 **Upload** knop  klikt. Het is dus niet altijd nodig het programma te verifiëren voordat je het upload.

4. Klik op de **Upload** 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](#) (blz. x) van de microcontroller. Dit zorgt ervoor dat de microcontroller opnieuw opstart en dus de code nogmaals uitvoert.

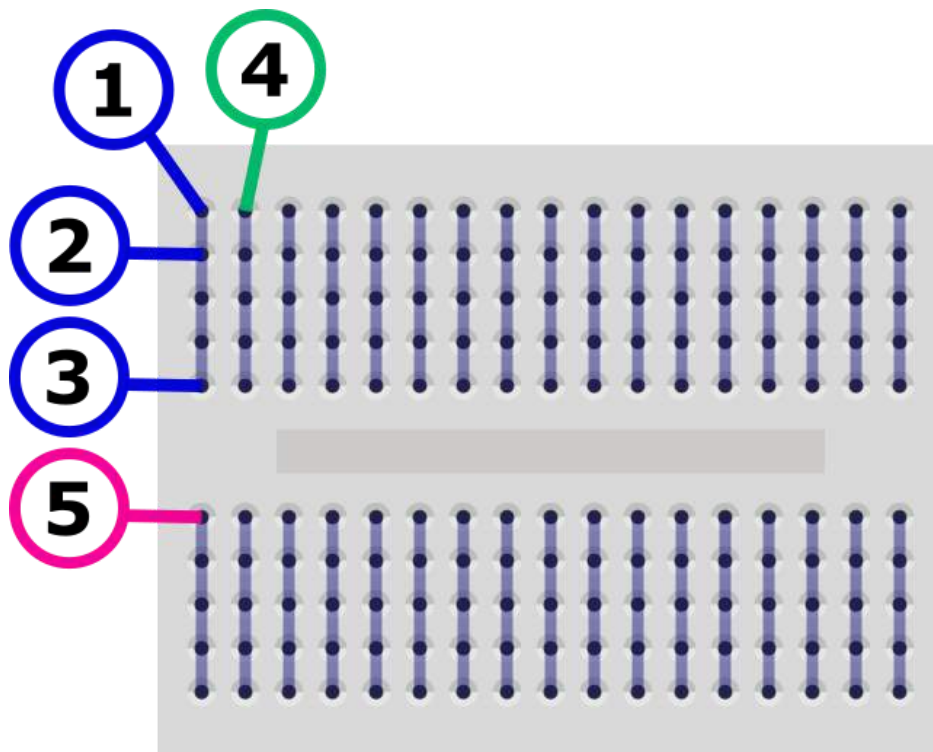
Knippert het lampje niet? Bekijk dan de tips onder [Problemen oplossen](#) (blz. x).

# Het breadboard

In [oefening 1](#) (blz. x) hebben we een ingebouwde led op de microcontroller laten knippen. 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.

## Benodigheden

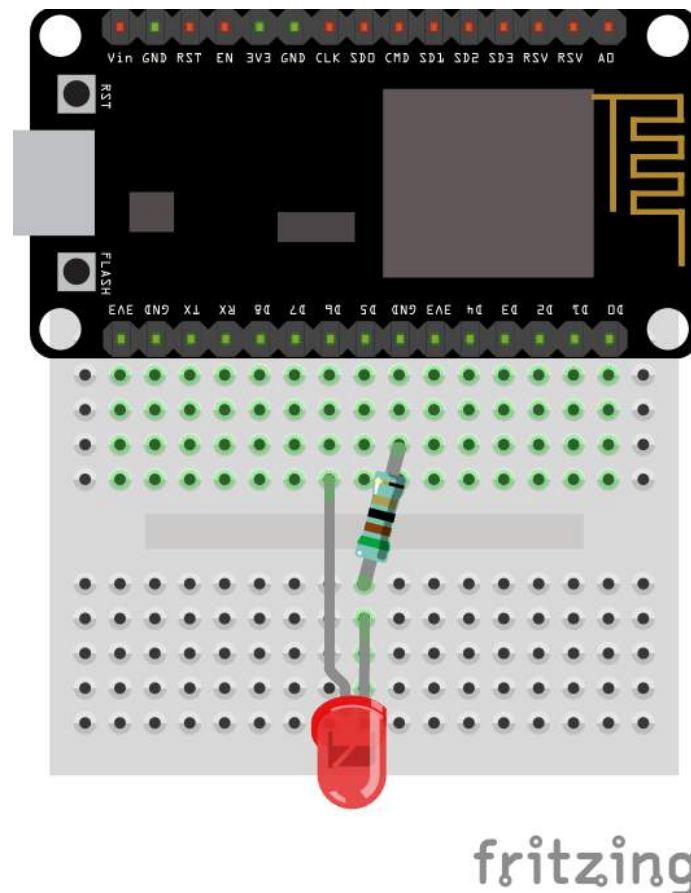
- NodeMCU
- USB kabel
- Computer
- Weerstand
- Breadboard
- LED lampje

## Verbindingen

Microcontroller	Onderdeel	Via
GND	LED korte pootje	51Ω 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(D6, OUTPUT);    // We gaan pin D6 gebruiken als output

  digitalWrite(D6, HIGH); // Zet de LED aan door een hoog signaal op D6 zetten
  delay(1000);            // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D6, LOW);  // Zet de LED uit door een laag signaal op D6 te zetten
  delay(1000);            // Wacht 1 seconde (1000 milliseconden)

  digitalWrite(D6, HIGH); // Zet de LED aan door een hoog signaal op D6 zetten
  delay(1000);            // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D6, LOW);  // Zet de LED uit door een laag signaal op D6 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](#) (blz. **x**). We hebben de pin die we willen aansturen veranderd van D4 naar D6. We willen namelijk niet meer de ingebouwde LED op pin D4 gebruiken maar de LED die wij hebben aangesloten op pin D6.

Daarnaast hebben we de waardes LOW en HIGH gewisseld. De externe LED gaat namelijk aan bij een hoog signaal en uit bij een laag signaal. Dit noemen we active high.

3. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. **x**) met de computer, dat je [het juiste board](#) (blz. **x**) en [de juiste poort](#) (blz. **x**) geselecteerd hebt.

4. Klik op de **Upload** 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](#) (blz. **x**) van de microcontroller om de code nogmaals uit te voeren.

Knippert de LED niet? Bekijk dan de tips onder [Problemen oplossen](#) (blz. **x**).

## Oefening 3: Functies

In [oefening 2](#) (blz. x) hebben we de LED aan en weer uitgezet met het volgende stukje code:

```
digitalWrite(D6, HIGH); // Zet de LED aan door een hoog signaal op D6 te zetten
delay(1000);           // Wacht 1 seconde (1000 milliseconden)
digitalWrite(D6, LOW); // Zet de LED uit door een laag signaal op D6 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(D6, HIGH); // Zet de LED aan door een hoog signaal op D6 te zetten
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D6, LOW); // Zet de LED uit door een laag signaal op D6 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(D6, HIGH); // Zet de LED aan door een hoog signaal op D6 te zetten
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D6, LOW); // Zet de LED uit door een laag signaal op D6 te zetten
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup() {
  pinMode(D6, OUTPUT); // We gaan pin D6 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

- NodeMCU
- USB kabel
- Computer
- Breadboard
- Jumper kabeltjes
- LED lampje
- 51Ω weerstand

## Stappen

1. Maak de schakeling die we in [oefening 2 \(blz. x\)](#) gebruikt hebben.
2. Neem de code uit het blok [Arduino code oefening 3 \(blz. x\)](#) over in de Arduino IDE.
3. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.
4. Klik op de **Upload** 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 \(blz. x\)](#).

Knippert de LED niet? Bekijk dan de tips onder [Problemen oplossen \(blz. x\)](#).



## Oefening 4: de "loop" gebruiken

In [oefening 3](#) (blz. x) 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

- NodeMCU
- USB kabel
- Computer
- 51Ω weerstand
- Breadboard
- Jumper kabeltjes
- LED lampje

### Verbindingen

<i>Microcontroller</i>	<i>Onderdeel</i>	<i>Via</i>
GND	LED korte pootje	51Ω weerstand
D6	LED lange pootje	Jumper kabeltje

### Stappen

5. Maak de schakeling die we in [oefening 2](#) (blz. x) gebruikt hebben.

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

```
// Deze functie zet de LED aan en na 1 seconde weer uit
void zetLedAanEnUit() {
  digitalWrite(D6, HIGH); // Zet de LED aan door een hoog signaal op D6 zetten
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D6, LOW); // Zet de LED uit door een laag signaal op D6 te zetten
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)
}


// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup() {
  pinMode(D6, OUTPUT); // We gaan pin D6 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 D6 als output gaan gebruiken blijft in de setup functie staan. Dit hoeven we namelijk slechts één keer aan te geven.

7. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.

8. Klik op de **Upload** 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 \(blz. x\)](#).

## 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	Aangeropen door jou
zetLedAanEnUit	Geschreven door jou	Aangeropen 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
<code>int</code>	Gehele getallen	12345
<code>boolean</code>	Iets dat slechts twee waardes kan hebben	TRUE of FALSE
<code>char</code>	Een teken (character)	'A' of '!'
<code>float</code>	Getallen met een komma ( <i>Deze programmeertaal gebruikt de Engelse notatie voor getallen. We gebruiken daarom een punt in plaats van een komma</i> )	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 eenVoorbeeldTekens = 'c';
```

### Benodigheden

- NodeMCU
- Breadboard
- USB kabel
- Jumper kabeltjes
- Computer
- LED lampje
- 51Ω weerstand

## Verbindingen

Microcontroller	Onderdeel	Via
GND	LED korte pootje	51Ω weerstand
D6	LED lange pootje	Jumper kabeltje

## Stappen

1. Maak de schakeling die we in [oefening 2 \(blz. x\)](#) 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(D6, HIGH); // Zet de LED aan met een hoog signaal op D6
  delay(aantalMillisecondenWachten); // Wacht een halve seconde (500 milliseconden)
  digitalWrite(D6, LOW); // Zet de LED uit met een laag signaal op D6
  delay(aantalMillisecondenWachten); // Wacht een halve seconde (500 milliseconden)
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup() {
  pinMode(D6, OUTPUT); // We gaan pin D6 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 \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.
4. Klik op de **Upload** 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 \(blz. x\)](#).

## 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(D6, HIGH);           // Zet de LED aan met een hoog signaal op D6
3   delay(aantalMillisecondenAan);     // Wacht het aangegeven aantal milliseconden
4   digitalWrite(D6, 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 waarden 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

- NodeMCU
- USB kabel
- Computer
- 51Ω weerstand
- Breadboard
- Jumper kabeltjes
- LED lampje

### Verbindingen

<i>Microcontroller</i>	<i>Onderdeel</i>	<i>Via</i>
GND	LED korte pootje	51Ω weerstand
D6	LED lange pootje	Jumper kabeltje

### Stappen


1. Maak de schakeling die we in [oefening 2 \(blz. x\)](#) gebruikt hebben.

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

```
1 void zetLedAanEnUit(int aantalMillisecondenAan, int aantalMillisecondenUit) {
2   digitalWrite(D6, HIGH);           // Zet de LED aan met een hoog signaal op D6
3   delay(aantalMillisecondenAan);     // Wacht het aangegeven aantal milliseconden
4   digitalWrite(D6, LOW);           // Zet de LED uit met een laag signaal op D6
5   delay(aantalMillisecondenUit);     // Wacht het aangegeven aantal milliseconden
6 }
7
8 // Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
9 void setup() {
10  pinMode(D6, OUTPUT);              // We gaan pin D6 gebruiken als output
11 }
12
13 // Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
14 void loop() {
15   zetLedAanEnUit(1, 25);           // Roep de functie zetLedAanEnUit aan
16 }
```

### Arduino code oefening 6

3. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.

4. Klik op de **Upload** 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 \(blz. x\)](#).



## Oefening 7: Analog Write

In [oefening 6 \(blz. x\)](#) hebben we gebruik gemaakt van de functie `digitalWrite` en `delay` om de LED snel achter elkaar aan en uit te schakelen. Dit is een prima manier om één LED te dimmen. Wanneer je meerdere LED's gaat dimmen wordt deze manier wat omslachtig. Gelukkig is er de ingebouwde functie `analogWrite` die het dimmen van een LED voor ons kan regelen.

De functie `analogWrite` kan worden aangeroepen met een waarde tussen de 0 en 255. Deze waarde bepaalt hoe hard de LED brandt. Bij de waarde 0 is de LED helemaal uit. Bij de waarde 255 brandt de LED op zijn felst.

De aanroep van de functie `analogWrite` ziet er als volgt uit:

```
analogWrite(127);
```

In dit voorbeeld zal de LED op halve kracht branden.

We kunnen de code uit [oefening 6 \(blz. x\)](#) nu als volgt herschrijven:

```
1 // Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
2 void setup() {
3   pinMode(D6, OUTPUT);           // We gaan pin D6 gebruiken als output
4   analogWrite(D6, 127);         // Roep de functie analogWrite aan
5 }
6
7 // Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
8 void loop() {
9 }
```

*Arduino code oefening 7*

Zoals je ziet hoeven we nu geen code meer aan te roepen in de `loop` functie. De `analogWrite` functie neemt dit voor ons uit handen en zorgt er achter de schermen voor dat er steeds een hoog en dan weer een laag signaal naar de LED gestuurd wordt.


### Benodigheden

- NodeMCU
- USB kabel
- Computer
- 51Ω weerstand
- Breadboard
- Jumper kabeltjes
- LED lampje

## Verbindingen

<i>Microcontroller</i>	<i>Onderdeel</i>	<i>Via</i>
GND	LED korte pootje	51Ω weerstand
D6	LED lange pootje	Jumper kabeltje

## Stappen

1. Maak de schakeling die we in [oefening 2 \(blz. x\)](#) gebruikt hebben.
2. Neem de code uit het blok [Arduino code oefening 7 \(blz. x\)](#) over in de Arduino IDE.
3. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.
4. Klik op de **Upload** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan brandt de LED nu op halve kracht.

Gaat de LED niet aan of brandt deze niet op halve kracht? Bekijk dan de tips onder [Problemen oplossen \(blz. x\)](#).

# Oefening 8: Meerdere LED's aansturen

In deze oefening gaan we drie LED's aansluiten en via onze microcontroller aansturen. Door drie LED's met de kleuren rood, groen en blauw (RGB) te combineren kunnen we alle kleuren van de regenboog maken.

We gaan de schakeling zo compact mogelijk maken zodat we straks de lampion over de LED's kunnen schuiven. We plaatsen de verschillende onderdelen van de schakeling heel dicht op elkaar. Hierdoor zijn de diagrammen soms moeilijk af te lezen. We maken in deze oefening daarom gebruik van meerdere diagrammen en foto's.

## Benodigdheden

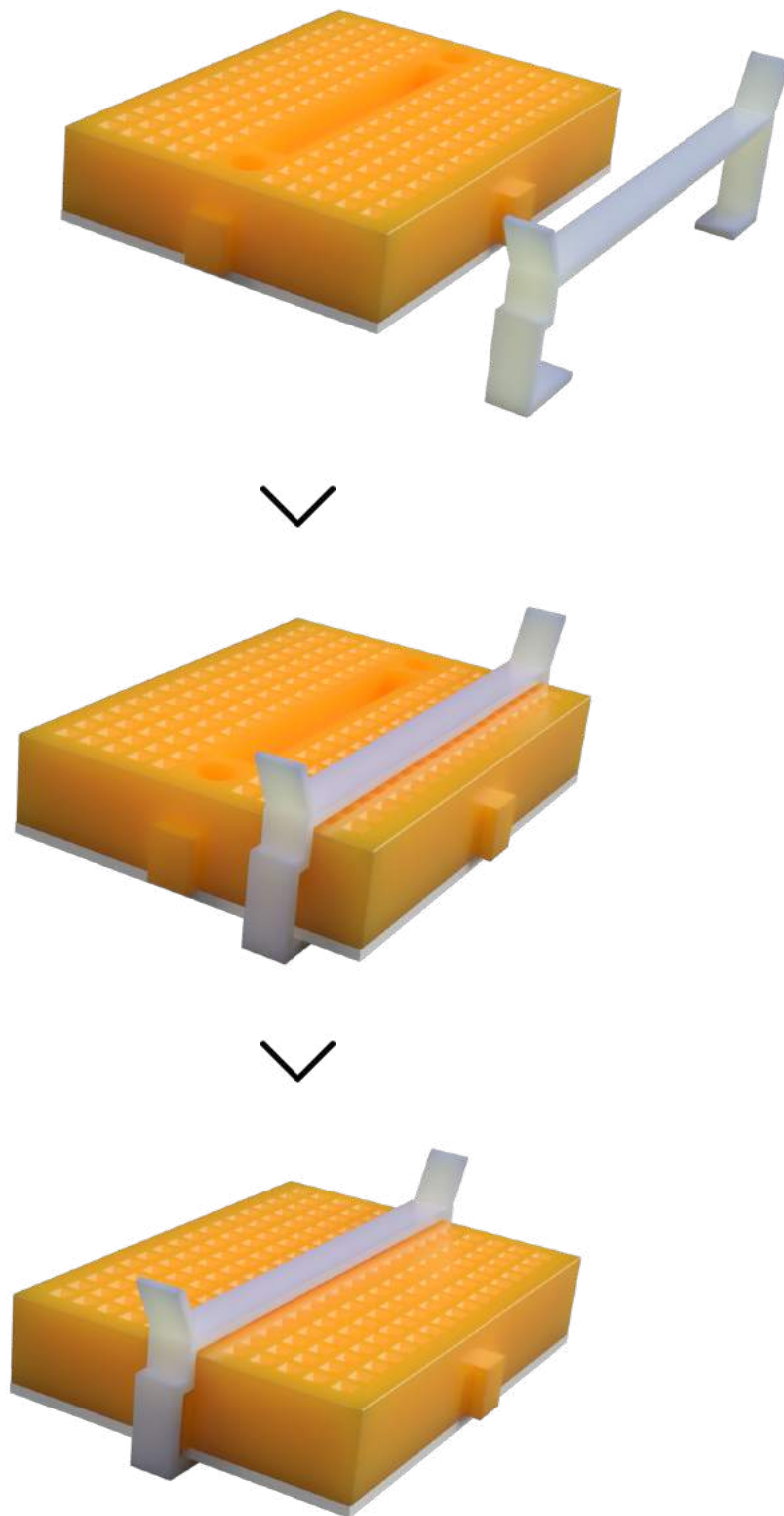
- NodeMCU
- USB kabel
- Computer
- 51 $\Omega$  weerstand x 3
- Lampionhouder
- Breadboard
- Rood LED lampje
- Groen LED lampje
- Blauw LED lampje

## Verbindingen

<i>Microcontroller</i>	<i>Onderdeel</i>	<i>Via</i>
GND	Rode LED korte pootje	51 $\Omega$ weerstand
GND	Groene LED korte pootje	51 $\Omega$ weerstand
GND	Blauwe LED korte pootje	51 $\Omega$ weerstand
D5	Rode LED lange pootje	-
D6	Groene LED lange pootje	-
D7	Blauwe LED lange pootje	-

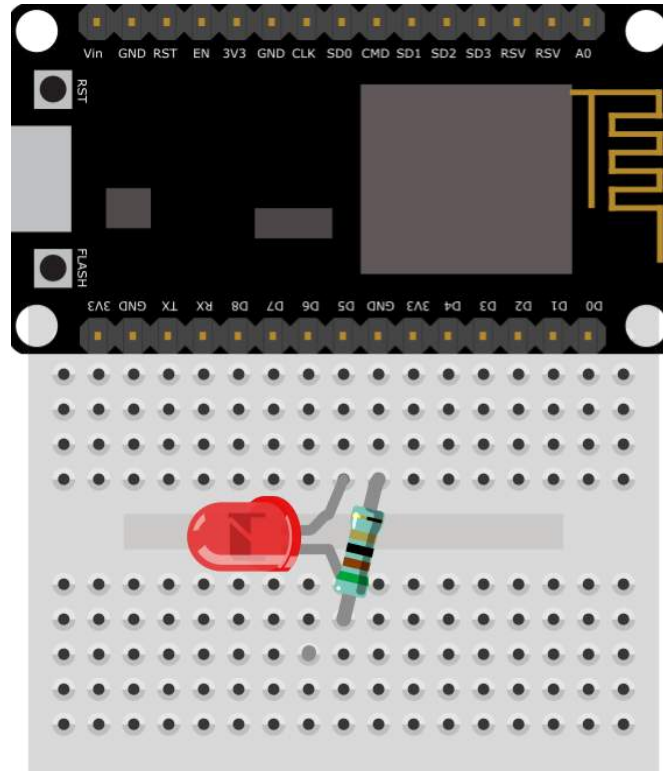
## Stappen

1. Schuif de lampionhouder over het breadboard zoals in afbeelding 8.1.



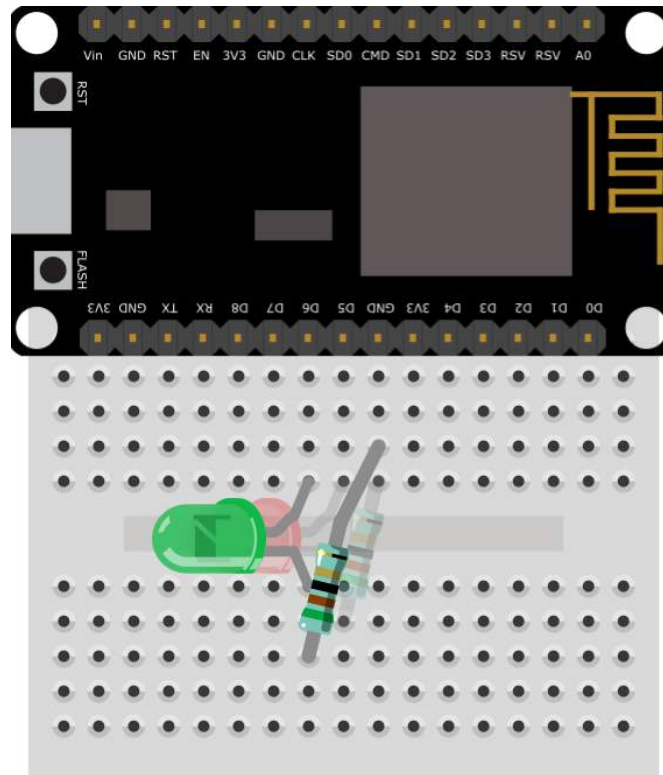
*Afbeelding 8.1*

2. Maak de schakeling uit afbeelding 8.2. De lampionhouder is hier niet in weergegeven. Steek de LED en weerstand over de lampionhouder heen.



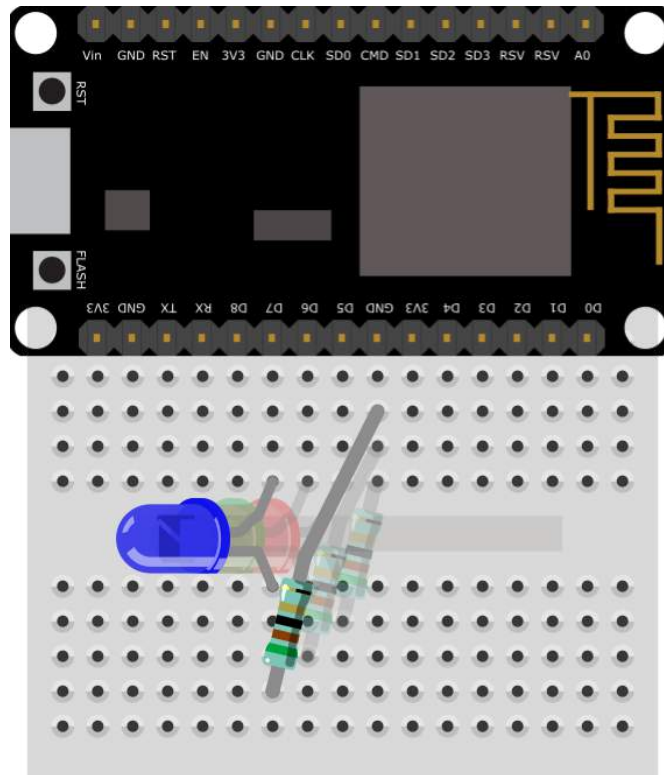
Afbeelding 8.2

3. Maak de schakeling uit afbeelding 8.3. We bouwen hierbij voort op de vorige schakeling.



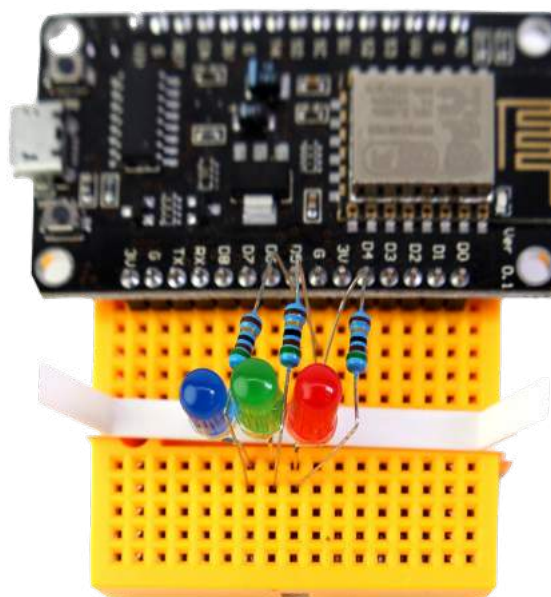
Afbeelding 8.3

4. Maak de schakeling uit afbeelding 8.4. We bouwen hierbij voort op de vorige twee schakelingen.



*Afbeelding 8.4*

5. Controleer of jouw schakeling overeenkomt met de foto in afbeelding 8.5



*Afbeelding 8.5*

6. Controleer de volgende punten:

- Eén weerstand is verbonden met het korte pootje van de rode LED (de min).
- Eén weerstand is verbonden met het korte pootje van de groene LED (de min).
- Eén weerstand is verbonden met het korte pootje van de blauwe LED (de min).
- De andere kant van al de weerstanden is verbonden met dezelfde GND pin op de microcontroller.
- De pootjes van de weerstanden raken elkaar en de pootjes van de LED's niet.
- Het lange pootje van de rode LED (de plus) is verbonden met pin D5.
- Het lange pootje van de rode LED (de plus) is verbonden met pin D6.
- Het lange pootje van de rode LED (de plus) is verbonden met pin D7.
- De pootjes van de weerstanden n LED's raken elkaar niet.

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

```
// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup() {
  pinMode(D5, OUTPUT);           // We gaan pin D5 gebruiken als output voor de rode LED
  pinMode(D6, OUTPUT);           // We gaan pin D6 gebruiken als output voor de groene LED
  pinMode(D7, OUTPUT);           // We gaan pin D7 gebruiken als output voor de blauwe LED
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  analogWrite(D5, 200);           // Laat de rode LED fel branden (200 van de 255)
  analogWrite(D6, 50);            // Laat de blauwe LED zwak branden (50 van de 255)
  analogWrite(D7, 50);            // Laat de groene LED zwak branden (50 van de 255)
  delay(1000);                    // Wacht 1 seconde (1000 milliseconden)

  analogWrite(D5, 50);            // Laat de rode LED zwak branden (50 van de 255)
  analogWrite(D6, 200);           // Laat de blauwe LED fel branden (200 van de 255)
  analogWrite(D7, 50);            // Laat de groene LED zwak branden (50 van de 255)
  delay(1000);                    // Wacht 1 seconde (1000 milliseconden)

  analogWrite(D5, 50);            // Laat de rode LED zwak branden (50 van de 255)
  analogWrite(D6, 50);            // Laat de blauwe LED zwak branden (50 van de 255)
  analogWrite(D7, 200);           // Laat de groene LED fel branden (200 van de 255)
  delay(1000);                    // Wacht 1 seconde (1000 milliseconden)
}
```

*Arduino code oefening 8*

8. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.



9. Klik op de **Upload** knop  om het programma te uploaden naar de microcontroller.

De LED's zouden nu alle drie moeten branden en afwisselend feller en minder fel moeten worden.

Is dit niet het geval? Bekijk dan de tips onder [Problemen oplossen](#) (blz. x).

## Oefening 9: De lampion bevestigen

In [oefening 10](#) (blz. x) gaan we onze lamp verschillende kleuren laten maken. Dit doen we door de kleuren van de LED's (rood, groen en blauw) te mengen. Dit werkt het beste wanneer het licht van de leds diffuus maken. Hiervoor gaan we een lampion gebruiken.

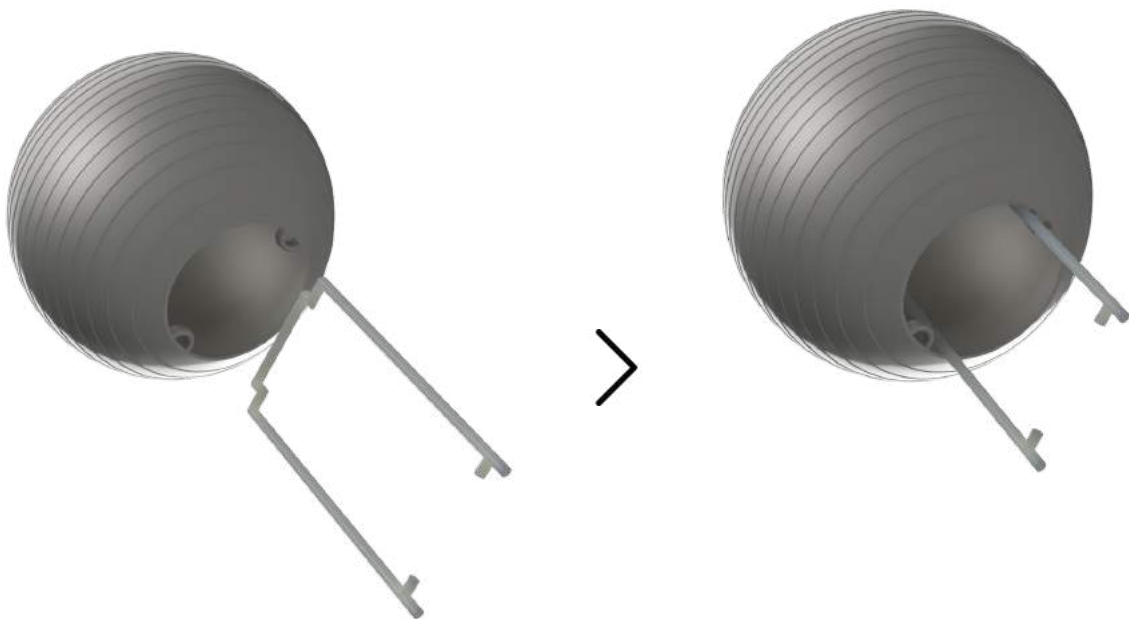
**Let op:** de lampion is gemaakt van papier en kan scheuren. Ga daarom voorzichtig met de lampion om.

### Benodigdheden

- [Schakeling uit oefening 8](#)
- Lampionbenen
- Lampion

### Stappen

1. Schuif de lampionbenen via het grote gat de lampion in. Let er op dat je het open stuk van de lampionbenen aan de onderkant houdt.



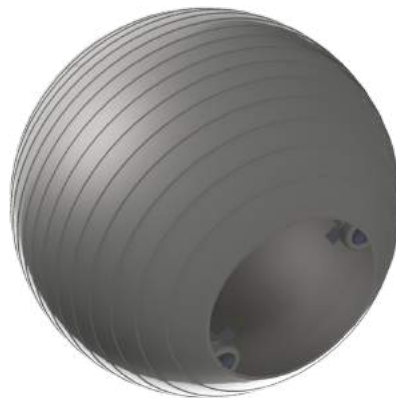
Afbeelding 9.1

2. Zorg ervoor dat de twee inkepingen boven aan de lampionbenen in het kleine gat vallen.



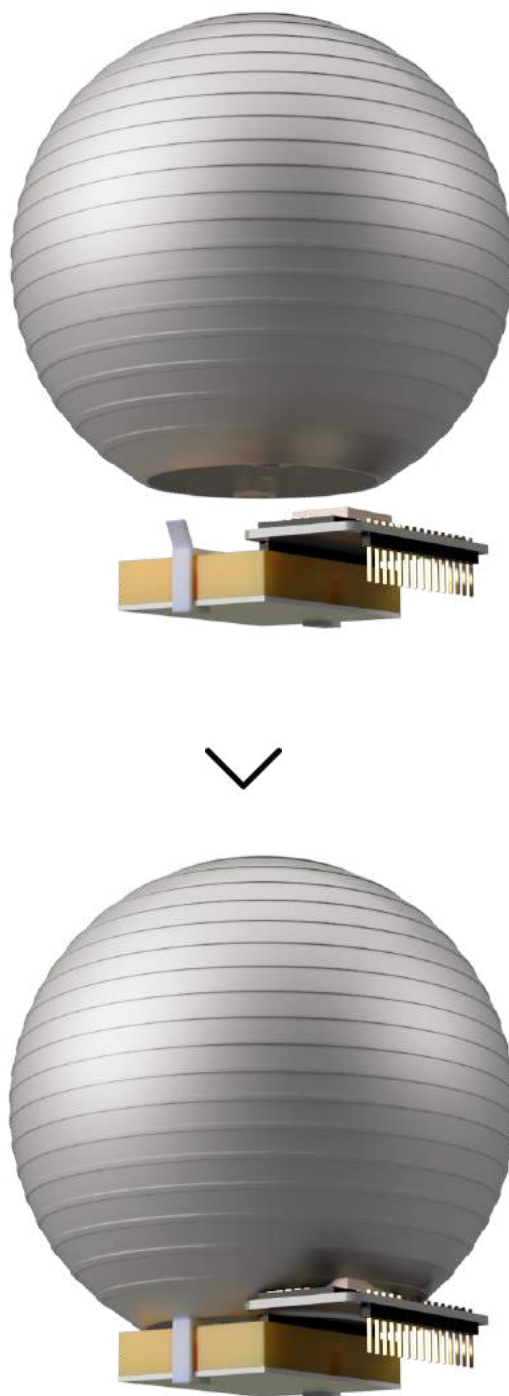
*Afbeelding 9.2*

3. Span de lampion door de onderkant van de lampionbenen in de cirkels onder aan de lampion te steken.



*Afbeelding 9.3*

- Schuif het grote gat van de lampion over de schuine benen van de lampionhouder. Je kan de benen iets naar binnen drukken om de lampion er makkelijk overheen te schuiven.



*Afbeelding 9.4*

De lampion maakt het licht van de leds diffuus waardoor de kleuren meer met elkaar versmelten. Dit werkt het beste in een donkere ruimte.

# Oefening 10: Kleuren maken

In onze schakeling hebben we een LED voor de primaire kleuren rood, groen en blauw. Door deze kleuren met elkaar te mengen kunnen we heel veel verschillende kleuren creëren. We noemen dit het RGB-kleurensysteem. Meer over dit kleurensysteem vind je op:

<https://nl.wikipedia.org/wiki/RGB-kleursysteem>

In onze code mengen we de kleuren door iedere LED met een bepaalde sterkte te laten branden. We gaan hiervoor een functie schrijven. Door onze functie aan te roepen met een waarde (van 0 tot en met 255) voor de kleuren rood, groen en blauw kunnen we verschillende kleuren creëren. Deze functie komt er als volgt uit te zien:

```
1 void maakRgbKleur(int rood, int groen, int blauw) {
2   analogWrite(D5, rood);      // Stel de waarde van de rode LED in
3   analogWrite(D6, groen);     // Stel de waarde van de groene LED in
4   analogWrite(D7, blauw);     // Stel de waarde van de blauwe LED in
5 }
```

In de rest van de code kunnen we onze LED's nu eenvoudig een kleur laten weergeven door de functie `maakRgbKleur()` aan te roepen. Om te bepalen welke waardes je mee moet geven aan deze functie om een bepaalde kleur te maken kun je gebruikmaken van een van de volgende websites:

<https://www.google.com/search?q=rgb+color+picker>

[https://www.rapidtables.com/web/color/RGB\\_Color.html](https://www.rapidtables.com/web/color/RGB_Color.html)

## Benodigheden

- NodeMCU
- USB kabel
- Computer
- 51Ω weerstand
- Breadboard
- Jumper kabeltjes
- LED lampje

## Verbindingen

<i>Microcontroller</i>	<i>Onderdeel</i>	<i>Via</i>
GND	LED korte pootje	51Ω weerstand
D6	LED lange pootje	Jumper kabeltje

## Stappen

1. Maak de schakeling die we in [oefening 8 \(blz. x\)](#) gebruikt hebben.
2. Neem de onderstaande code over in de Arduino IDE.

```
// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup() {
  pinMode(D5, OUTPUT);           // We gaan pin D5 gebruiken als output voor de rode LED
  pinMode(D6, OUTPUT);           // We gaan pin D6 gebruiken als output voor de groene LED
  pinMode(D7, OUTPUT);           // We gaan pin D7 gebruiken als output voor de blauwe LED
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  maakRgbKleur(255, 0, 0);        // Rood
  delay(1000);                   // Wacht 1 seconde (1000 milliseconden)
  maakRgbKleur(0, 255, 0);       // Groen
  delay(1000);                   // Wacht 1 seconde (1000 milliseconden)
  maakRgbKleur(0, 0, 255);       // Blauw
  delay(1000);                   // Wacht 1 seconde (1000 milliseconden)
  maakRgbKleur(255, 255, 125);   // Framboos
  delay(1000);                   // Wacht 1 seconde (1000 milliseconden)
  maakRgbKleur(0, 255, 255);     // Cyaan
  delay(1000);                   // Wacht 1 seconde (1000 milliseconden)
  maakRgbKleur(255, 0, 255);     // Magenta
  delay(1000);                   // Wacht 1 seconde (1000 milliseconden)
  maakRgbKleur(255, 255, 0);     // Geel
  delay(1000);                   // Wacht 1 seconde (1000 milliseconden)
  maakRgbKleur(255, 255, 255);   // Wit
  delay(1000);                   // Wacht 1 seconde (1000 milliseconden)
}

// Deze functie stelt aan de hand van de meegegeven parameters de helderheid van de LED's in
void maakRgbKleur(int rood, int groen, int blauw) {
  analogWrite(D5, rood);         // Stel de waarde van de rode LED in
  analogWrite(D6, groen);        // Stel de waarde van de groene LED in
  analogWrite(D7, blauw);        // Stel de waarde van de blauwe LED in
}
```

### Arduino code oefening 6

3. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.
4. Klik op de **Upload** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan verschijnen de volgende kleuren één voor één: rood, groen, blauw, framboos, cyaan, magenta, geel en wit.

Wil je een andere kleur maken? Gebruik dan één van de hierboven genoemde de websites om de waardes voor rood, groen en blauw te bepalen en roep de functie `maakRgbKleur()` aan met die waardes.

Worden de kleuren niet weergegeven? Bekijk dan de tips onder [Problemen oplossen \(blz. x\)](#).

## Oefening 10: De seriële verbinding

In de volgende oefeningen gaan we er voor zorgen dat we onze lamp via een website kunnen bedienen. Om dit te kunnen moeten we het IP-adres van onze microcontroller kunnen uitlezen (meer over IP-adressen lees je in de volgende oefeningen). We gaan dit doen door gebruik te maken van de seriële verbinding.

De seriële verbinding is een manier om data tussen de computer en de microcontroller uit te wisselen. Hiervoor wordt gebruikt gemaakt van de USB kabel die de computer en microcontroller met elkaar verbindt.

Wanneer we een programma via de Arduino IDE naar de microcontroller uploaden dan doen we dat via de seriële verbinding. Diezelfde verbinding kunnen we ook gebruiken om tekst van de microcontroller naar de Arduino IDE te sturen. In deze oefening laten we zien hoe dit werkt.

Meer details over de seriële verbinding lees je op:

[https://nl.wikipedia.org/wiki/Seri%C3%ABle\\_poort](https://nl.wikipedia.org/wiki/Seri%C3%ABle_poort)

Om de seriële verbinding met de Arduino IDE te starten maken we gebruik van de ingebouwde functie `Serial.begin()`. Als argument geven we de snelheid mee waarmee we met de Arduino IDE willen praten in bits per seconde (baud).

```
Serial.begin(9600);
```

In dit voorbeeld sturen we 9600 bits per seconden naar de Arduino IDE.

Nadat de seriële verbinding gestart is kunnen we informatie gaan versturen. Dit doen we met de ingebouwde functies `Serial.print()` en `Serial.println()`.

### **Serial.print**

Met `Serial.print()` sturen we characters naar naar de Arduino IDE. De characters die we willen versturen geven we mee als argument.

```
Serial.print();
```

## Stappen

1. Voor deze oefening hebben we geen schakeling nodig. Je kan de schakeling uit de vorige opdracht gewoon laten zoals die is.
2. Neem de onderstaande code over in de Arduino IDE.

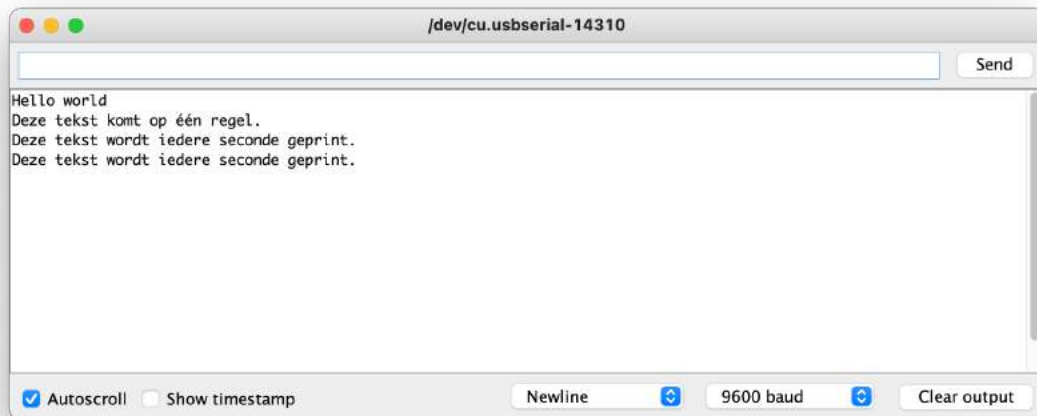
```
// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup() {
  Serial.begin(9600);           // Start de seriele verbinding
  Serial.println("Hello world"); // Print de tekst "Hello world" en start een nieuwe regel
  Serial.print("Deze tekst");   // Print de tekst "Deze tekst"
  Serial.print(" komt op");     // Print de tekst " komt op"
  Serial.println(" één regel."); // Print de tekst " één regel." en start een nieuwe regel
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
  // Print de tekst "Deze tekst wordt iedere seconde geprint." en start een nieuwe regel
  Serial.println("Deze tekst wordt iedere seconde geprint.");
  delay(1000);                 // Wacht 1 seconde (1000 milliseconden)
}
```

### Arduino code oefening 10

3. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.
4. Klik op de **Upload** knop  om het programma te uploaden naar de microcontroller.
5. Wacht tot de code geüpload is naar de microcontroller. Klik daarna op de Serial Monitor knop . Er opent nu een nieuw scherm. Dit scherm noemen we de Serial Monitor. Hier zie je tekst welke vanaf de microcontroller naar de computer verstuurd worden.





*Afbeelding 10.1*

In onze code hebben we aangegeven dat we met 9600 baud met de computer willen praten. Zorg ervoor dat ook in de Serial Monitor deze waarde is ingevuld.

Wanneer alles goed is gegaan verschijnen nu de teksten die je in de Arduino IDE getypt hebt in de Serial Monitor.

Verschijnt niet de juiste tekst? Bekijk dan de tips onder [Problemen oplossen](#) (blz. x).

# Oefening 11: Wifi

In deze oefening laten we de microcontroller verbinding maken met jouw wifi-netwerk en tonen we de gegevens van de verbinding in de Serial Monitor.

In de volgende oefeningen gaan we de wifi-verbinding gebruiken om de microcontroller dingen te laten doen via het netwerk.

## Stappen

1. Voor deze oefening hebben we geen schakeling nodig. Je kan de schakeling uit de vorige opdracht gewoon laten zoals die is.
2. Neem de onderstaande code over in de Arduino IDE.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>

const char* ssid = "JOUW-NETWERKNAAM"; // Vul hier jouw eigen netwerknnaam (ssid) in
const char* password = "JOUW-WACHTWOORD"; // Vul hier jouw eigen netwerkwachtwoord in

// Deze functie maakt verbinding met het netwerk
void maakVerbindingMetWifi() {
  WiFi.begin(ssid, password); // Gebruik WiFi.begin om verbinding te maken
  while (WiFi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
    delay(500); // Wacht een halve seconde
    Serial.print("."); // Print een punt (.) in de Serial Monitor
  }
}

// Deze functie print de gegevens van de wifiverbinding in de Serial Monitor
void printWifiGegevens() {
  Serial.println(""); // Print een lege regel
  Serial.print("Verbonden met ");
  Serial.println(ssid); // Print de naam (ssid) van het netwerk
  Serial.print("IP-adres van deze microcontroller: ");
  Serial.println(WiFi.localIP()); // Print het ip-adres van de microcontroller
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup(void) {
  Serial.begin(9600); // Start de seriele verbinding
  WiFi.mode(WIFI_STA); // Type wifi verbinding instellen


  maakVerbindingMetWifi(); // De functie maakVerbindingMetWifi aanroepen
  printWifiGegevens(); // De functie printWifiGegevens aanroepen
}


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

## Arduino code oefening 11

3. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste](#)

[poort](#) (blz. x) geselecteerd hebt.

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

5. Wacht tot de code geüpload is naar de microcontroller. Klik daarna op de Serial Monitor knop .

Wanneer alles goed is gegaan verschijnen er punten in de Serial Monitor totdat de microcontroller een verbinding met het wifi netwerk heeft opgebouwd. Daarna wordt de naam van het wifi netwerk en het ip-adres van de microcontroller weergegeven.

Het ip-adres dat wordt weergegeven gaan we later in een webbrowser invoeren om de microcontroller opdrachten te geven.

Verschijnen de gegevens niet in de Serial Monitor? Reset dan de microcontroller door op de reset knop te drukken. Zie hiervoor [afbeelding 0.1](#) (blz. x). Werkt het programma daarna nog niet? Bekijk dan de tips onder [Problemen oplossen](#) (blz. x).

## Oefening 12: Aansturen via wifi

Nu de microcontroller een verbinding heeft met het wifi-netwerk kunnen we deze opdrachten gaan geven via de webbrowser. We gaan hiervoor code voor een webserver op de microcontroller zetten.

Een webserver is een stuk software dat ervoor zorgt dat je informatie via een netwerk kunt uitwisselen. Wanneer je naar een website gaat zoals [www.awesomemakes.com](http://www.awesomemakes.com) dan is er een webserver die een signaal (we noemen dit een request) ontvangt van jouw webbrowser en de gevraagde informatie terug naar jouw webbrowser stuurt (we noemen dit een response).

Net als de webserver waar de website [www.awesomemakes.com](http://www.awesomemakes.com) op staat kan je ook de microcontroller straks om een website vragen. De microcontroller stuurt dan niet alleen de gevraagde website terug naar jouw webbrowser maar kan ook nog andere dingen doen zoals een LED aanzetten.

In deze oefening gaan we een LED aansturen via de webbrowser.

### Stappen

1. Maak de schakeling die we in [oefening 8](#) (blz. x) gebruikt hebben.
2. Neem de onderstaande code over in de Arduino IDE (let op: de code loopt door op de volgende bladzijde).

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char* ssid = "JOUW-NETWERKNAAM"; // Vul hier jouw eigen netwerk naam (ssid) in
const char* password = "JOUW-WACHTWOORD"; // Vul hier jouw eigen netwerk wachtwoord in

ESP8266WebServer server(80); // Start de webserver

// Deze functie wordt aangeroepen als we naar "/aan" gaan
void ledAan() {
  digitalWrite(D5, HIGH); // Zet de LED aan met een hoog signaal op D5
  server.send(200, "text/plain", "De led is aan!\r\n"); // Stuur tekst naar de webbrowser
}

// Deze functie wordt aangeroepen als we naar "/uit" gaan
void ledUit() {
  digitalWrite(D5, LOW); // Zet de LED uit met een laag signaal op D5
  server.send(200, "text/plain", "De led is uit!\r\n"); // Stuur tekst naar de webbrowser
}

// Deze functie maakt verbinding met het netwerk
void maakVerbindingMetWifi() {
  WiFi.begin(ssid, password); // Gebruik WiFi.begin om verbinding te maken
  while (WiFi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
    delay(500); // Wacht een halve seconde
  }
}
```

```

    Serial.print(".");                // Print een punt (.) in de Serial Monitor
  }
}

// Deze functie print de gegevens van de wifiverbinding in de Serial Monitor
void printWifiGegevens() {
  Serial.println("");                // Print een lege regel
  Serial.print("Verbonden met ");
  Serial.println(ssid);              // Print de naam (ssid) van het netwerk
  Serial.print("IP-adres van deze microcontroller: ");
  Serial.println(WiFi.localIP());    // Print het ip-adres van de microcontroller
  Serial.print("Ga naar http://");
  Serial.print(WiFi.localIP());
  Serial.println("/aan om de LED aan te zetten");
  Serial.print("Ga naar http://");
  Serial.print(WiFi.localIP());
  Serial.println("/uit om de LED uit te zetten");
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup(void) {
  pinMode(D5, OUTPUT);              // We gaan pin D5 gebruiken als output
  Serial.begin(9600);               // Start de seriële verbinding
  WiFi.mode(WIFI_STA);             // Type wifi verbinding instellen

  maakVerbindingMetWifi();          // De functie maakVerbindingMetWifi aanroepen
  printWifiGegevens();              // De functie printWifiGegevens aanroepen

  server.on("/aan", ledAan);        // Roep ledAan aan wanneer we naar "/aan" gaan
  server.on("/uit", ledUit);       // Roep ledUit aan wanneer we naar "/uit" gaan

  server.begin();                   // Start de webserver
  Serial.println("Server is gestart!");
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop(void) {
  // Kijk of er informatie door een webbrowser wordt gevraagd
  server.handleClient();
}

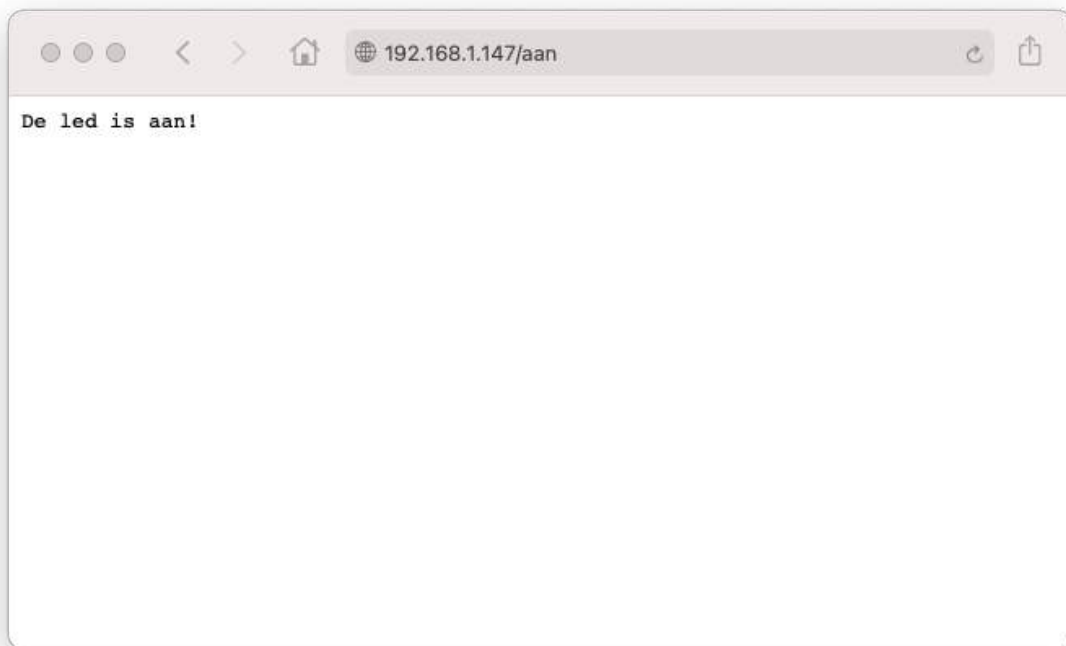
```

### Arduino code oefening 12

3. Zorg ervoor dat de [microcontroller verbonden is \(blz. x\)](#) met de computer, dat je [het juiste board \(blz. x\)](#) en [de juiste poort \(blz. x\)](#) geselecteerd hebt.
4. Klik op de **Upload** knop  om het programma te uploaden naar de microcontroller.
5. Wacht tot de code geüpload is naar de microcontroller. Klik daarna op de Serial Monitor knop .

6. Zoek in de Serial Monitor het ip-adres van jouw microcontroller op. Er zou een regel moeten zijn in de vorm van "IP-adres van deze microcontroller: <ip-adres>" Waarbij <ip-adres> een aantal getallen zijn die gescheiden worden door punten zoals "192.168.1.2". Verschijnen er geen puntjes en tekst in de Serial Monitor? Reset dan de microcontroller door op de reset knop te drukken. Zie hiervoor [afbeelding 0.1](#) (blz. x).
7. Open een webbrowser zoals Google Chrome, Firefox, Safari of Edge en typ in de adresbalk "http://<ip-adres>/aan" en vervang daarbij <ip-adres> door het ip-adres dat je in stap 6 gevonden hebt. Druk vervolgens op de enter knop om de pagina te laden.

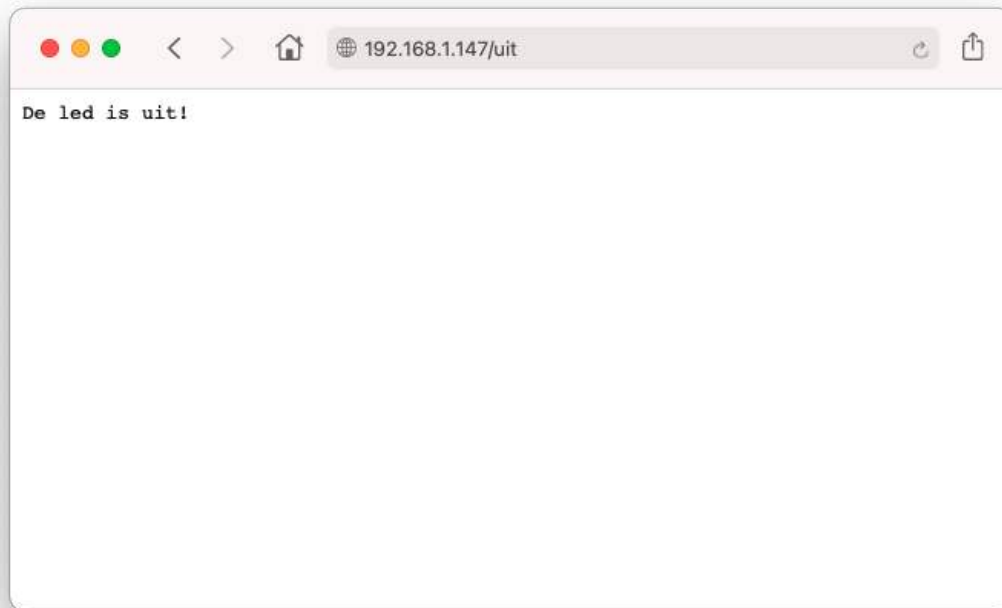
Als het goed is gaat de rode LED nu branden en krijg je een scherm zoals in afbeelding 12.1.



*Afbeelding 12.1*

8. Typ in de adresbalk "http://<ip-adres>/uit" en vervang daarbij <ip-adres> door het ip-adres dat je in stap 6 gevonden hebt. Druk vervolgens op de enter knop om de pagina te laden.

Als het goed is gaat de rode LED nu uit en krijg je een scherm zoals in afbeelding 12.2.



*Afbeelding 12.2*

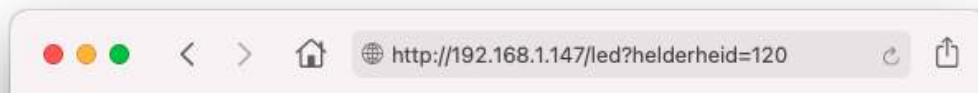
Lukt het niet om de LED aan en uit te doen via de browser? Bekijk dan de tips onder [Problemen oplossen](#) (blz. **x**).

## Oefening 13: Webpage met parameters

We kunnen nu de microcontroller iets laten doen door in onze webbrowser naar een bepaald adres te gaan. In de vorige opdracht hebben we daarmee een LED aan en uit gezet.

In veel gevallen wil je meer doen dan alleen een LED aan of uit zetten. In deze opdracht willen wij bijvoorbeeld de helderheid waarmee een LED brandt bepalen via de webbrowser.

Om dat te doen gaan we een parameter toevoegen aan het request dat wij naar de server (in dit geval de microcontroller) sturen. Een parameter is eigenlijk een variabele. In de browser ziet dit er als volgt uit:



Afbeelding 13.1

In dit voorbeeld is "helderheid" de naam van de parameter. "120" is de waarde die we aan deze parameter toewijzen.

### Stappen

1. Maak de schakeling die we in [oefening 8](#) (blz. x) gebruikt hebben.
2. Neem de onderstaande code over in de Arduino IDE (let op: de code loopt door op de volgende bladzijde).

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char* ssid = "JOUW-NETWERKNAAM"; // Vul hier jouw eigen netwerk naam (ssid) in
const char* password = "JOUW-WACHTWOORD"; // Vul hier jouw eigen netwerk wachtwoord in

ESP8266WebServer server(80); // Start de webserver

// Deze functie stelt de helderheid van de rode LED in
void helderheidInstellen() {
  // Lees de parameter "helderheid" uit en zet deze in de variabele "helderheid"
  int helderheid = server.arg("helderheid").toInt();
  analogWrite(D5, helderheid); // Stel de waarde van de rode LED in
  // Maak de variabele "bericht" en vul deze met een tekst en de waarde van "helderheid"
  String bericht = "De helderheid is ingesteld op: " + String(helderheid) + "\r\n";
```



```

Serial.println(bericht); // Stuur tekst naar de Serial Monitor
server.send(200, "text/plain", bericht); // Stuur tekst naar de webbrowser
}

// Deze functie maakt verbinding met het netwerk
void maakVerbindingMetWifi() {
  WiFi.begin(ssid, password); // Gebruik WiFi.begin om verbinding te maken
  while (WiFi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
    delay(500); // Wacht een halve seconde
    Serial.print("."); // Print een punt (.) in de Serial Monitor
  }
}

// Deze functie print de gegevens van de wifiverbinding in de Serial Monitor
void printWifiGegevens() {
  Serial.println(""); // Print een lege regel
  Serial.print("Verbonden met ");
  Serial.println(ssid); // Print de naam (ssid) van het netwerk
  Serial.print("IP-adres van deze microcontroller: ");
  Serial.println(WiFi.localIP()); // Print het ip-adres van de microcontroller
  Serial.print("Ga naar http://");
  Serial.print(WiFi.localIP());
  Serial.println("/led?helderheid=120 om de LED aan te zetten met de waarde 120");
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup(void) {
  pinMode(D5, OUTPUT); // We gaan pin D5 gebruiken als output
  Serial.begin(9600); // Start de seriële verbinding
  WiFi.mode(WIFI_STA); // Type wifi verbinding instellen

  maakVerbindingMetWifi(); // De functie maakVerbindingMetWifi aanroepen
  printWifiGegevens(); // De functie printWifiGegevens aanroepen
  // Roep helderheidInstellen aan wanneer we naar "/led" gaan
  server.on("/led", helderheidInstellen);

  server.begin(); // Start de webserver
  Serial.println("Server is gestart!");
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop(void) {
  // Kijk of er informatie door een webbrowser wordt gevraagd
  server.handleClient();
}

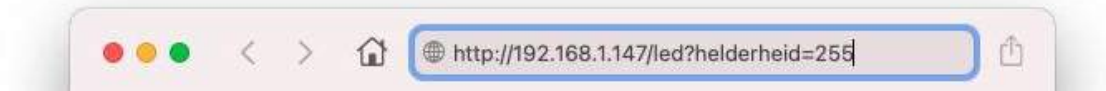
```

### Arduino code oefening 13

3. Zorg ervoor dat de microcontroller verbonden is (blz. x) met de computer, dat je het juiste board (blz. x) en de juiste poort (blz. x) geselecteerd hebt.
4. Klik op de **Upload** knop  om het programma te uploaden naar de microcontroller.
5. Wacht tot de code geüpload is naar de microcontroller. Lees daarna het ip-adres uit de Serial Monitor zoals je dat in oefening 12 gedaan hebt.

6. Typ in de adresbalk "http://<ip-adres>/led?helderheid=120" en vervang daarbij <ip-adres> door het ip-adres dat je in stap 6 gevonden hebt. Druk vervolgens op de enter knop om de pagina te laden.

Als het goed is gaat de rode LED nu op ongeveer halve kracht branden. Verander de waarde van "helderheid" in de adresbalk van je webbrowser om de LED harder of zachter te laten branden. In [afbeelding 13.2](#) laten we de LED bijvoorbeeld op volle kracht branden.



*Afbeelding 13.2*

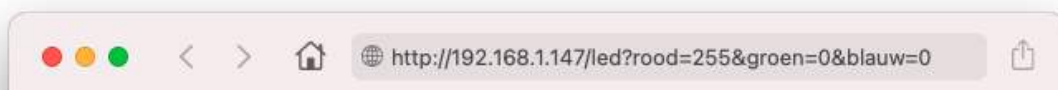
Lukt het niet om de LED te dimmen via de browser? Bekijk dan de tips onder [Problemen oplossen](#) (blz. x).

# Oefening 14: IOT Mood Light bouwen

Het is nu tijd om alles dat we in de voorgaande oefeningen geleerd hebben samen te voegen en de IOT Mood Light te bouwen.

De bedoeling is dat we een lamp maken die we van kleur kunnen veranderen via de webbrowser. We gaan dit doen door een request naar de microcontroller te sturen met drie parameters: rood, groen en blauw. Ieder van deze parameters bepaalt voor de betreffende kleur de sterkte waarmee de LED brandt.

In afbeelding 13.1 zie je een voorbeeld van hoe het adres er dan uit kan zien.



Afbeelding 14.1

In het voorbeeld laten we de rode LED op volle sterkte branden. De groene en blauwe led zetten we uit. Op die manier wordt onze lamp dus rood.

## Stappen

1. Maak de schakeling die we in [oefening 8](#) (blz. x) gebruikt hebben.
2. Neem de onderstaande code over in de Arduino IDE (let op: de code loopt door op de volgende bladzijde).

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char* ssid = "JOUW-NETWERKNAAM"; // Vul hier jouw eigen netwerk naam (ssid) in
const char* password = "JOUW-WACHTWOORD"; // Vul hier jouw eigen netwerk wachtwoord in

ESP8266WebServer server(80); // Start de webserver

// Deze functie stelt de helderheid van de LED's in
void maakRgbKleur() {
  // Lees de parameter "rood" uit en zet deze in de variabele "rood"
  int rood = server.arg("rood").toInt();
  // Lees de parameter "groen" uit en zet deze in de variabele "groen"
  int groen = server.arg("groen").toInt();
  // Lees de parameter "blauw" uit en zet deze in de variabele "blauw"
  int blauw = server.arg("blauw").toInt();
  analogWrite(D5, rood); // Stel de waarde van de rode LED in
  analogWrite(D6, groen); // Stel de waarde van de groene LED in
  analogWrite(D7, blauw); // Stel de waarde van de blauwe LED in
}
```

```

// Maak een variabele en vul deze met de waarde van de variabele "rood"
String berichtRood = "Rood: " + String(rood);
// Maak een variabele en vul deze met een tekst en de waarde van de variabele "groen"
String berichtGroen = "Groen: " + String(groen);
// Maak een variabele en vul deze met een tekst en de waarde van de variabele "blauw"
String berichtBlauw = "Blauw: " + String(blauw);

Serial.println(berichtRood);
Serial.println(berichtGroen);
Serial.println(berichtBlauw);

// Maak de variabele "htmlBericht" en voer hier de variabelen
// berichtRood, berichtGroen en berichtBlauw samen
String htmlBericht = "<h1>LED Mood Light</h1>" + berichtRood + "<br>" + berichtGroen +
"<br>" + berichtBlauw;
server.send(200, "text/html", htmlBericht); // Stuur tekst naar de webbrowser
}

// Deze functie maakt verbinding met het netwerk
void maakVerbindingMetWifi() {
  Wifi.begin(ssid, password); // Gebruik Wifi.begin om verbinding te maken
  while (Wifi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
    delay(500); // Wacht een halve seconde
    Serial.print("."); // Print een punt (.) in de Serial Monitor
  }
}

// Deze functie print de gegevens van de wifiverbinding in de Serial Monitor
void printWifiGegevens() {
  Serial.println(""); // Print een lege regel
  Serial.print("Verbonden met ");
  Serial.println(ssid); // Print de naam (ssid) van het netwerk
  Serial.print("IP-adres van deze microcontroller: ");
  Serial.println(Wifi.localIP()); // Print het ip-adres van de microcontroller
  Serial.print("Ga naar http://");
  Serial.print(Wifi.localIP());
  Serial.println("/led?rood=255&groen=0&blauw=0 om de LED aan te zetten met de waarde 100");
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de NodeMCU wordt ingeschakeld
void setup(void) {
  pinMode(D5, OUTPUT); // We gaan pin D5 gebruiken als output
  Serial.begin(9600); // Start de seriële verbinding
  Wifi.mode(WIFI_STA); // Type wifi verbinding instellen

  maakVerbindingMetWifi(); // De functie maakVerbindingMetWifi aanroepen
  printWifiGegevens(); // De functie printWifiGegevens aanroepen
  // Roep maakRgbKleur aan wanneer we naar "/led" gaan
  server.on("/led", maakRgbKleur);

  server.begin(); // Start de webserver
  Serial.println("Server is gestart!");
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop(void) {
  // Kijk of er informatie door een webbrowser wordt gevraagd
  server.handleClient();
}

```

### Arduino code oefening 14

3. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. x) met

de computer, dat je [het juiste board](#) (**blz. x**) en [de juiste poort](#) (**blz. x**) geselecteerd hebt.

4. Klik op de **Upload** knop  om het programma te uploaden naar de microcontroller.
5. Wacht tot de code geüpload is naar de microcontroller. Lees daarna het ip-adres uit de Serial Monitor zoals je dat in [oefening 12](#) gedaan hebt.
6. Typ in de adresbalk "http://<ip-adres>/led?rood=255&groen=0&blauw=0" en vervang daarbij <ip-adres> door het ip-adres dat je in stap 5 gevonden hebt. Druk vervolgens op de enter knop om de pagina te laden.

Als het goed is gaat de rode LED nu op ongeveer halve kracht branden. Verander de waarde van "helderheid" in de adresbalk van je webbrowser om de LED harder of zachter te laten branden. In [afbeelding 13.2](#) laten we de LED bijvoorbeeld op volle kracht branden.



*Afbeelding 14.2*

Lukt het niet om de LED aan en uit te doen via de browser? Bekijk dan de tips onder [Problemen oplossen](#) (**blz. x**).

# Problemen oplossen

## **Het verifiëren van mijn code is mislukt**

- Doorloop in dit geval de stappen voor het [kiezen van het bord \(blz. x\)](#) en [het aansluiten van de microcontroller \(blz. x\)](#) 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 \(blz. x\)](#)).

# Gefeliciteerd!

Nu heb je een zelfgemaakte IOT Mood Light.

Veel plezier!

## Meer bouwen?

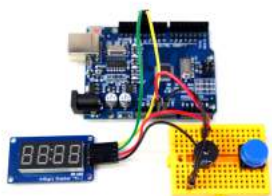
Heb je de smaak te pakken en wil je meer bouwen? Bekijk dan eens de andere Awesome Crates op [www.awesomemakes.com](http://www.awesomemakes.com)



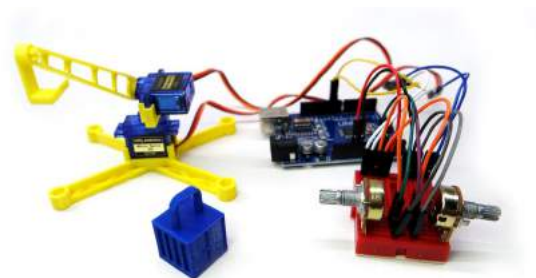
*IOT Smart Display*



*IOT Mood Light*



*Countdown Timer*



*Bestuurbare Hijskraan*