



# Handleiding

IOT Smart Display



# 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	16
Het breadboard	18
Oefening 2: Een buzzer aansluiten	19
Oefening 3: functies	21
Oefening 4: de "loop" gebruiken	23
Oefening 5: variabelen	25
Oefening 6: Een knop toevoegen	28
Oefening 7: OLED display aansluiten	32
Oefening 8: Tekst weergeven	38
Oefening 9: Wifi	40
Oefening 10: Gegevens van het internet ophalen	43
Oefening 11: Alarm	46
Oefening 12: Bitcoin koers ophalen	49
Oefening 13: De temperatuur tonen via een API met authenticatie	53
Problemen oplossen	57
Gefeliciteerd!	58
Inspiratie	58
Meer bouwen?	59



# Welkom

Wat leuk dat je een Awesome Crate gaat bouwen!

Deze crate bevat de benodigdheden en uitleg om een display te maken waar je actuele informatie op weer kunt geven over bijvoorbeeld het weer, de bitcoin koers of jouw aantal volgers op twitter.

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:

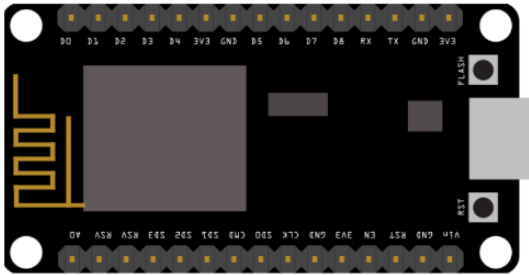
- Raak de microcontroller niet aan terwijl deze aan is.
- Zorg er altijd voor dat de microcontroller uit is (stroomkabeltje niet verbonden) voordat je onderdelen inpluigt.
- Sluit de USB kabel eerst aan op de microcontroller en daarna pas op de stroombron (computer, USB stekker, of power bank). Daarmee 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.



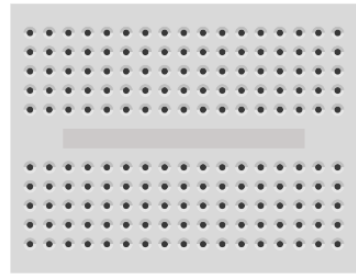
Dit is geen speelgoed. Bij het gebruik van dit artikel dienen kinderen onder de 14 jaar altijd begeleid te worden door een volwassene. Houd altijd toezicht op het product tijdens gebruik. Ontkoppel een eventuele stroombron na gebruik.



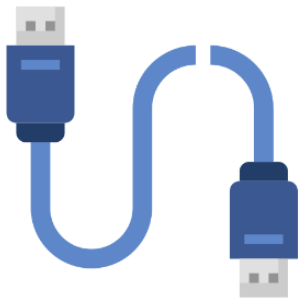
Microcontroller



Breadboard



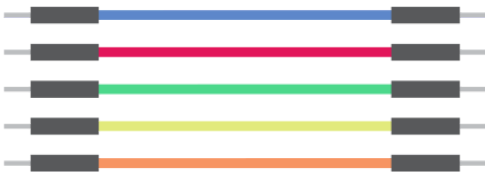
USB kabel



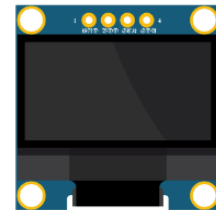
Weerstandjes  
100Ω



Jumper kabels



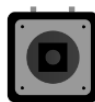
OLED display



Buzzer



Knop



Kapje

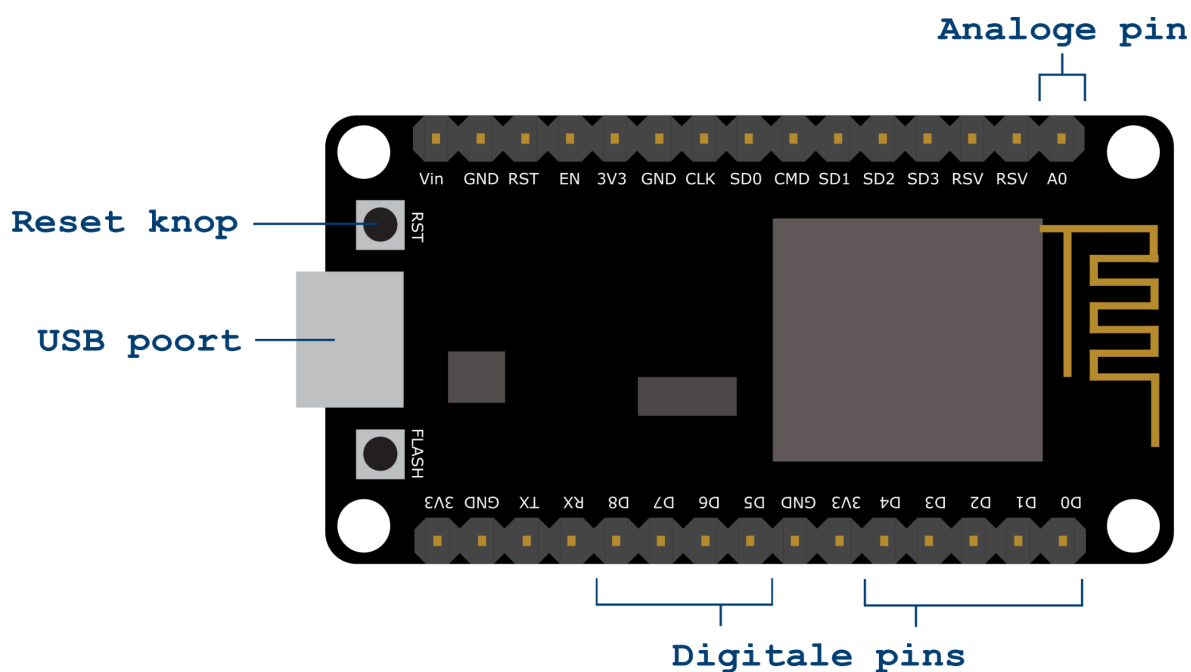


# De microcontroller

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 (Internet Of Things) projecten. Dat zijn projecten waarbij je een verbinding met het internet gebruikt.



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. 5)

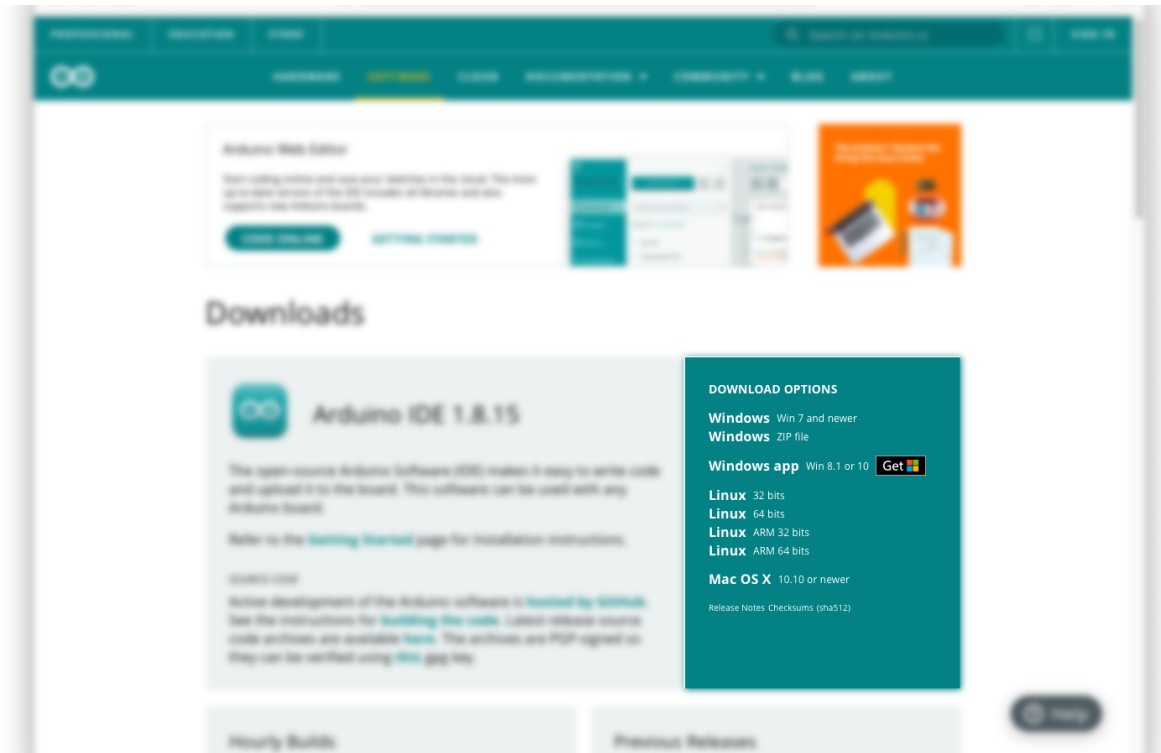
## **Gebruik je Windows?**

Ga dan verder bij [Arduino IDE installeren op Windows](#) (blz. 7)

# 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".

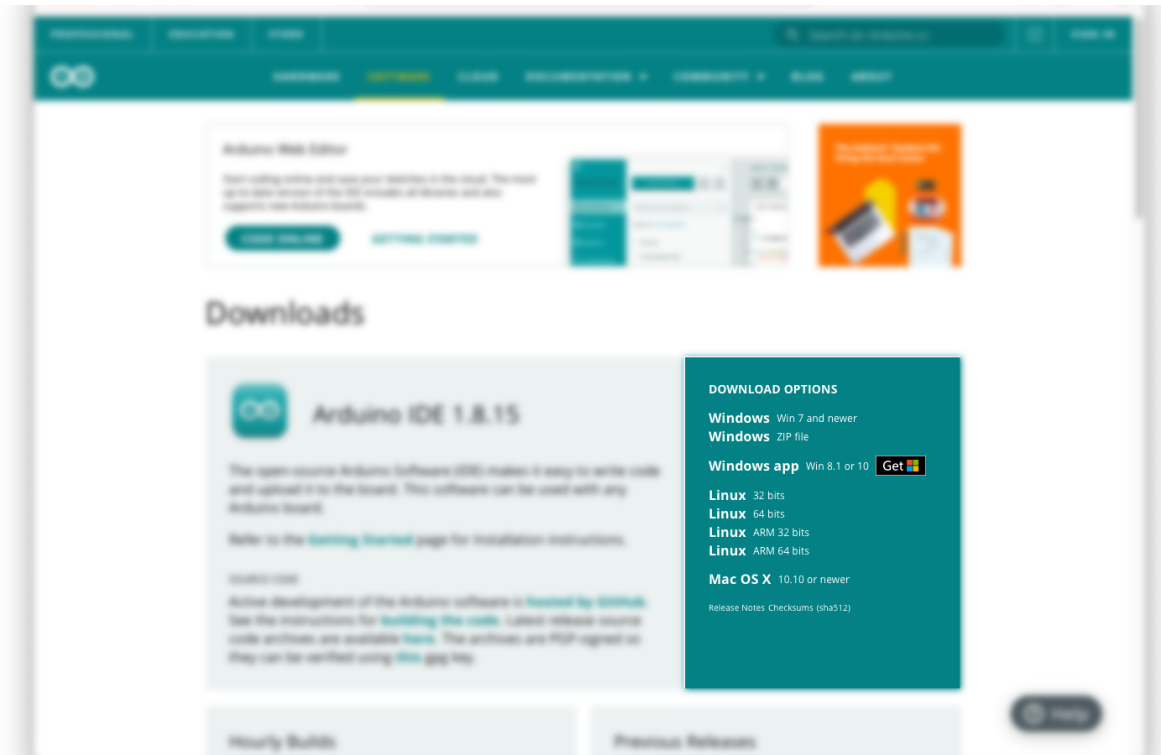


4. De Arduino IDE wordt nu gedownload. Verplaats het bestand daarna naar de applicatie map.
5. Open het programma. Het kan zijn dat jouw besturingssysteem nu om toestemming vraagt om de applicatie te openen. Geef in dit geval toestemming hiervoor.
6. De Arduino IDE wordt nu geopend.

# 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.

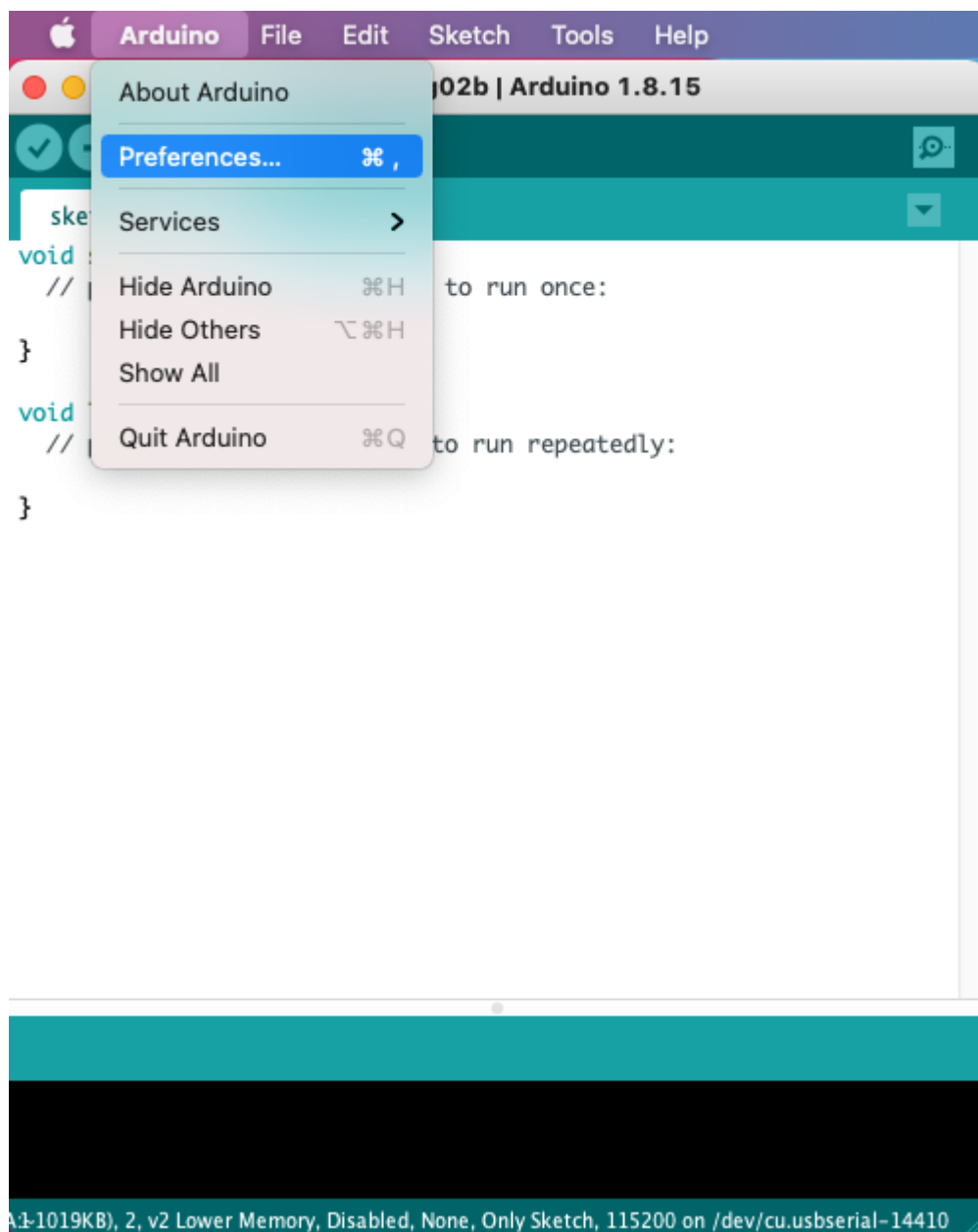
# Arduino IDE configureren

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

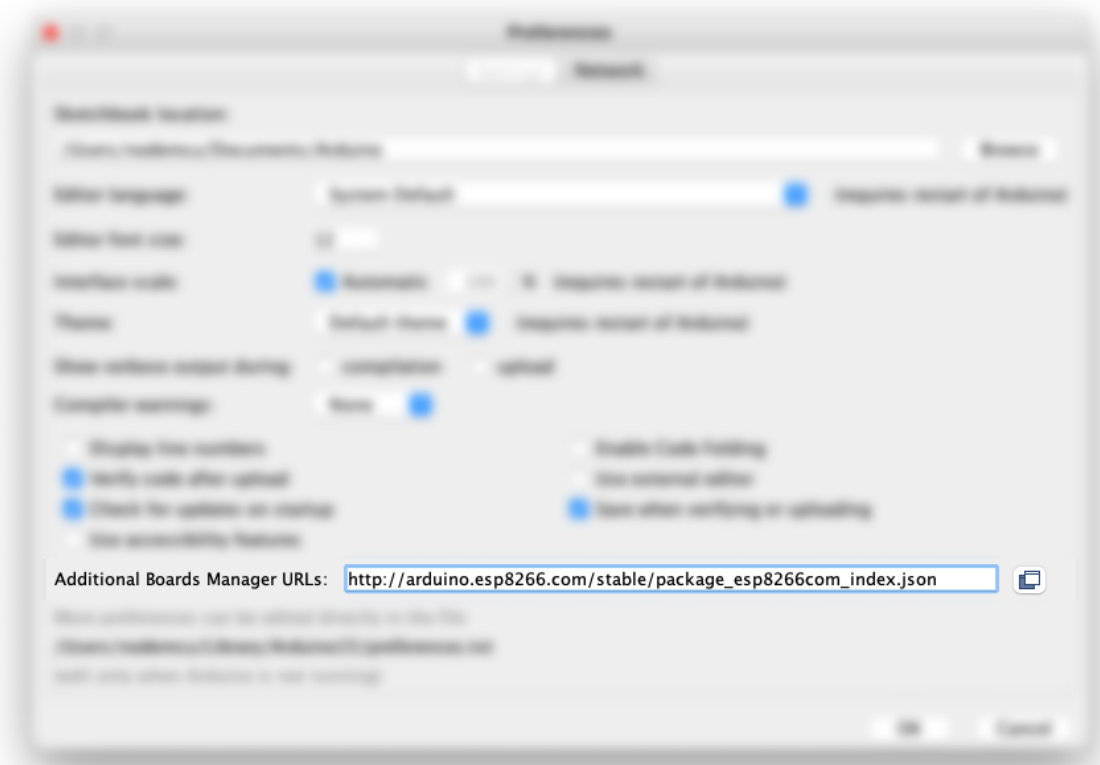
Eerst moeten we de NodeMCU installeren binnen de Arduino IDE.

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)



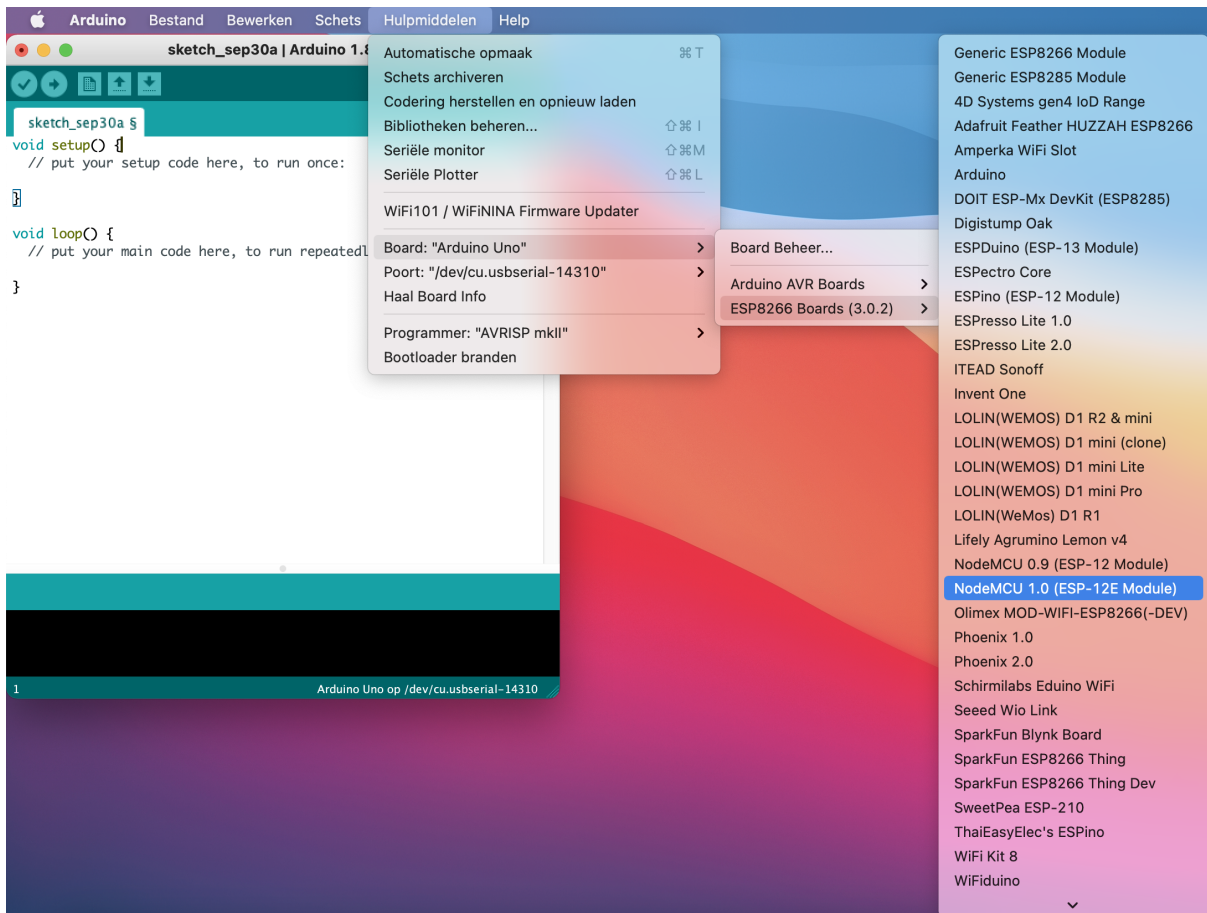
3. Klik op de knop **OK**.

# 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

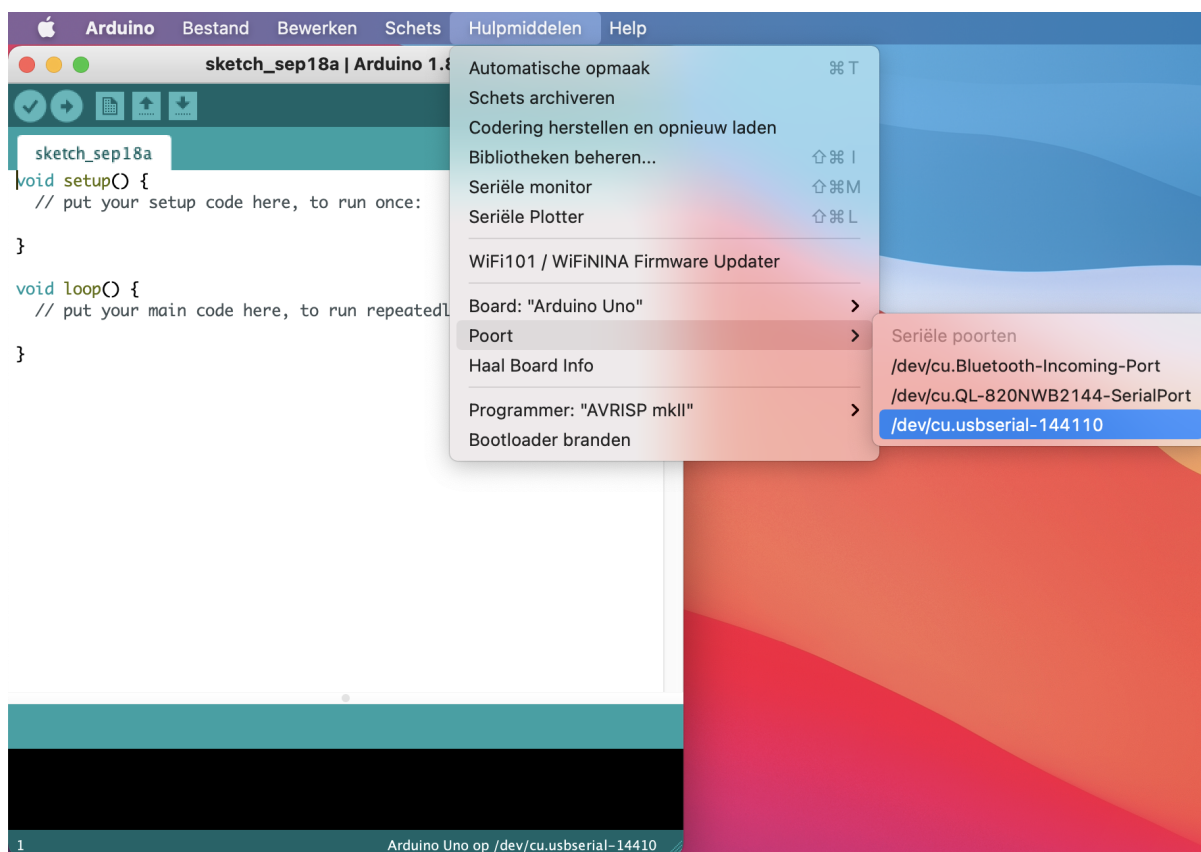
Nu gaan we de microcontroller via een USB kabel verbinden met de computer. Dit is de laatste stap voordat we code kunnen uploaden naar de microcontroller.

## 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).

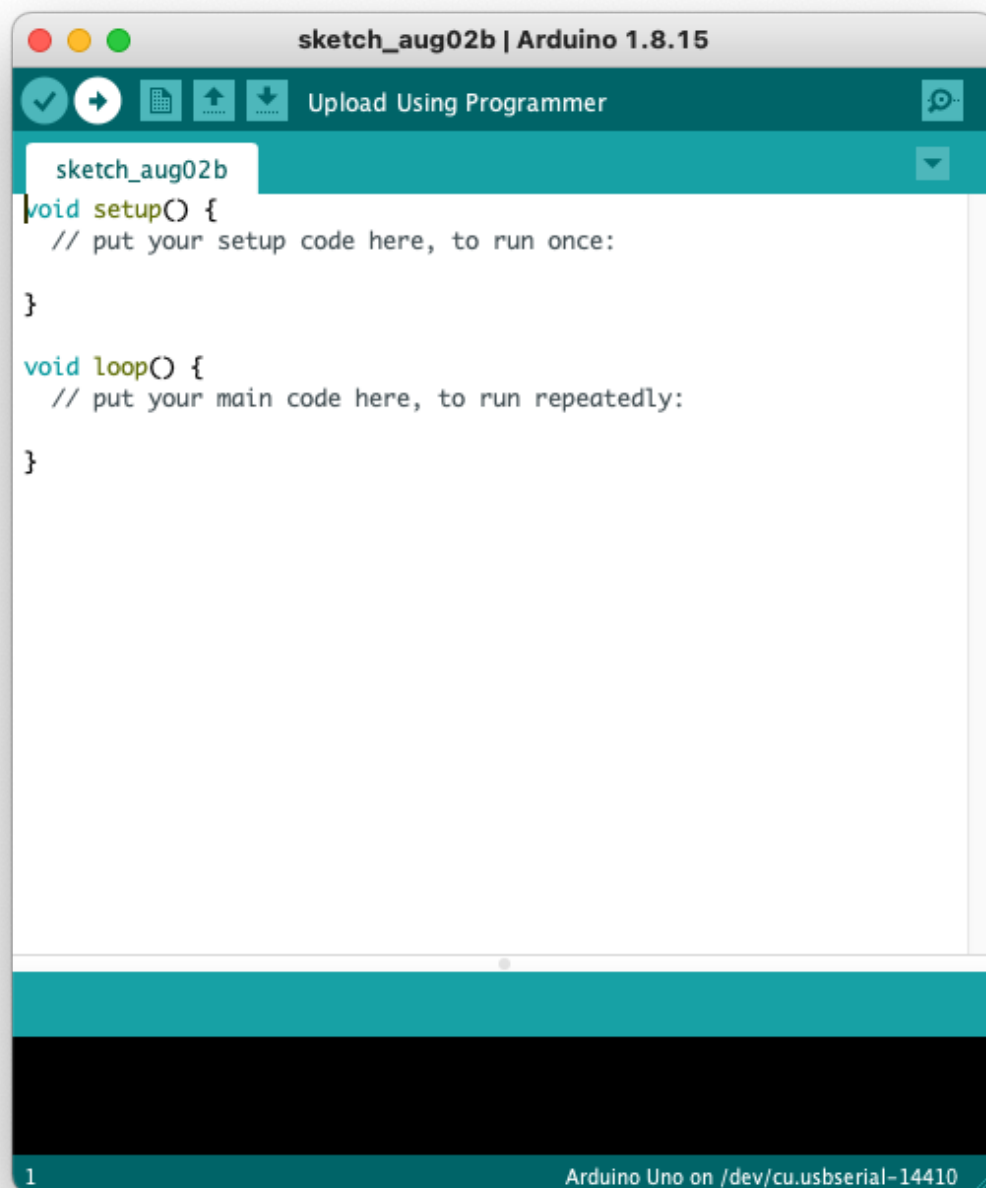
Weet je niet welke poort je moet gebruiken? Kijk dan welke poorten er nu in de lijst staan. Sluit het menu, ontkoppel de microcontroller en kijk opnieuw welke poorten er in de lijst staan. De poort die niet langer in de lijst staat is de poort van de microcontroller.



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.

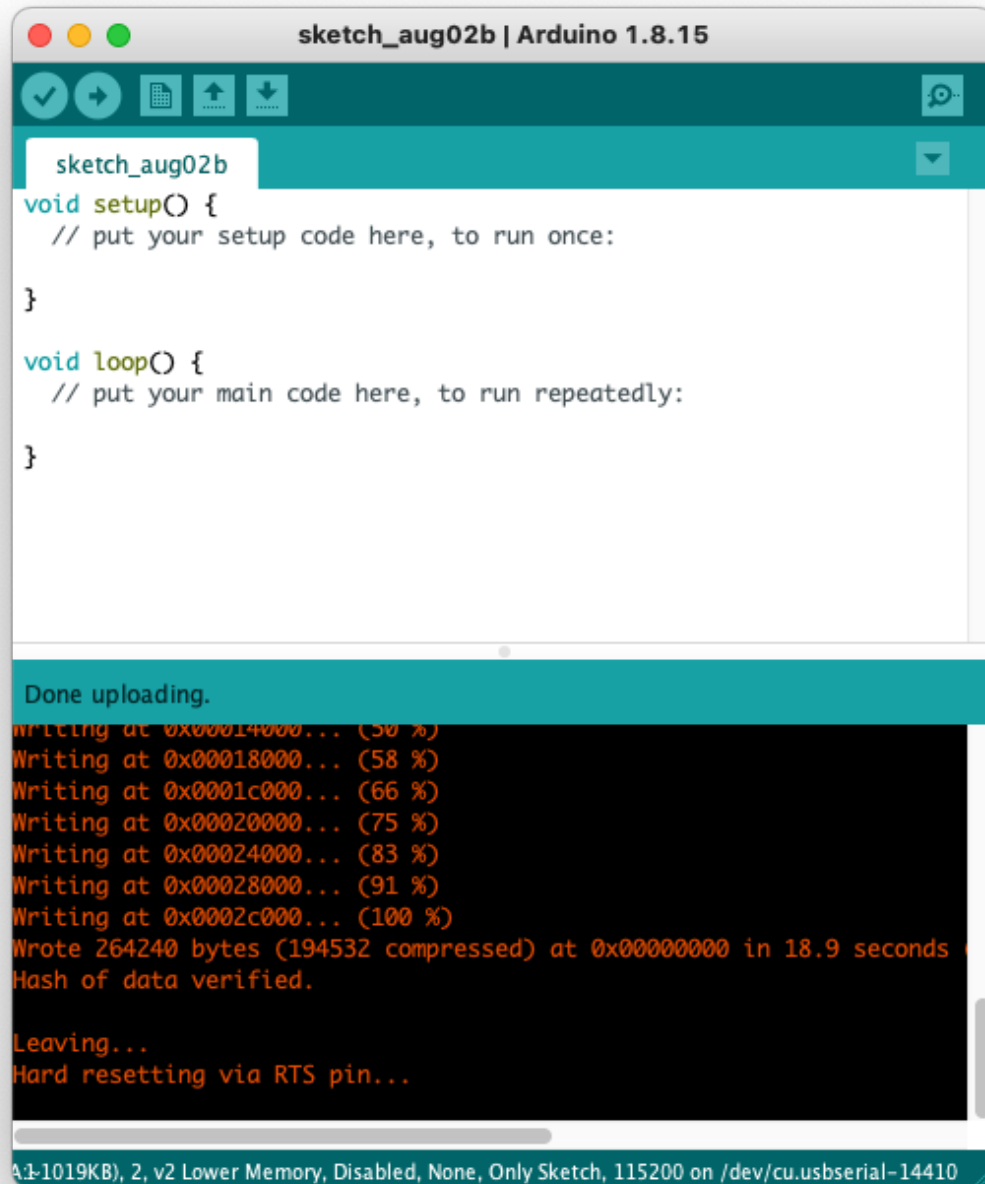
Nu de juiste poort geselecteerd is kunnen we de verbinding testen. Dit doen we door een vrijwel leeg programma naar de microcontroller te uploaden.

3. Klik op de **Upload** knop  .

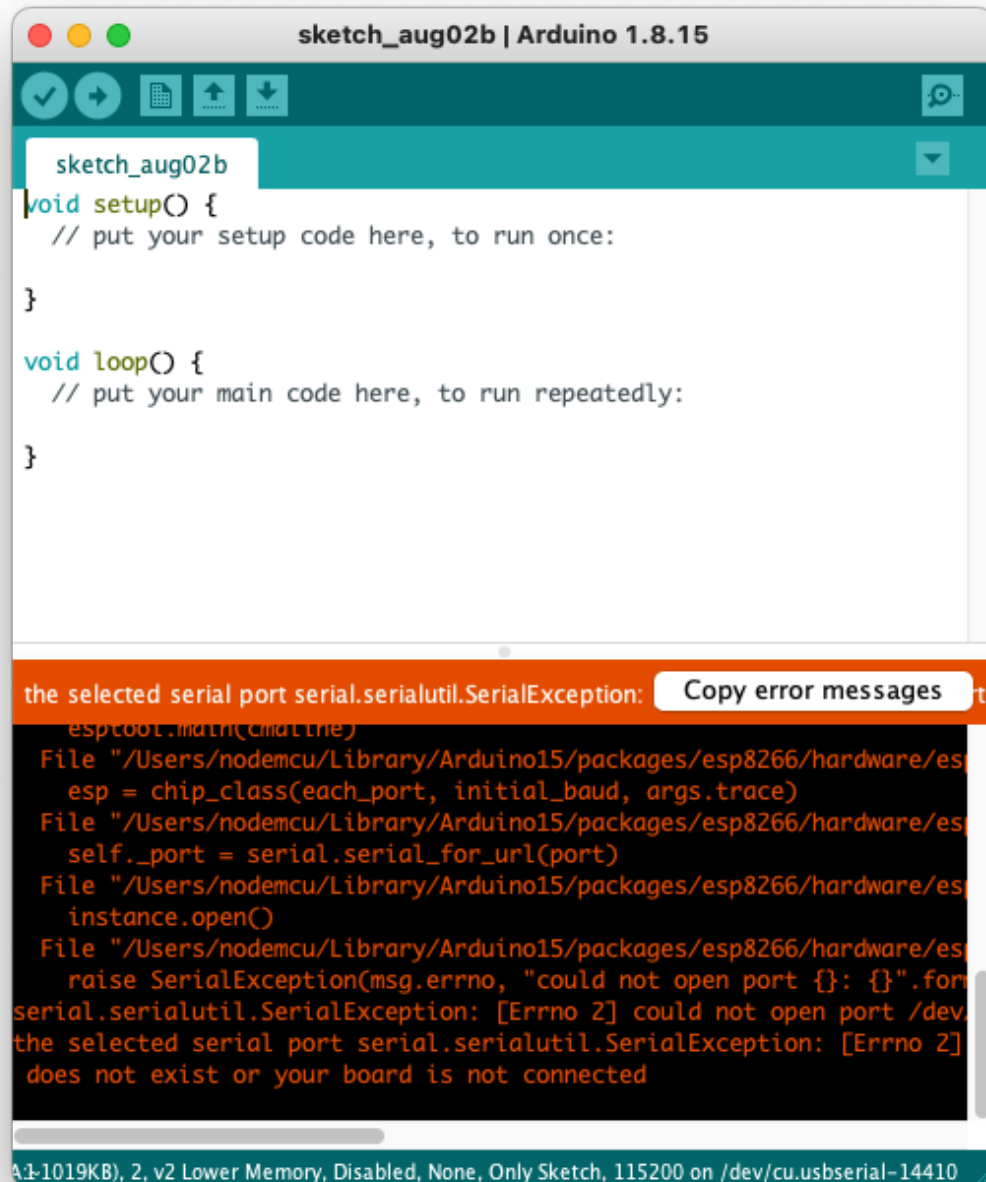




4. In het zwarte vlak onder in het scherm zie je nu een aantal berichten verschijnen. In de groene balk daarboven (statusbalk) zie je de status van het uploaden. Staat er in de groene balk "*Done uploading*"? Dan is het uploaden geslaagd.



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. 11) en [het aansluiten van de microcontroller](#) (blz. 12) 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.

## Benodigdheden

- 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. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) 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 **Uploaden** knop  klikt. Het is dus niet altijd nodig het programma te verifiëren voordat je het upload.

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!



Tijdens het uploaden en opstarten knippert de ingebouwde LED ook, maar daarbij zal het lampje steeds maar kort aan gaan. Je weet dat de door jou geuploade code werkt als het lampje twee keer gedurende een seconde aan blijft.

Druk op de reset knop (zie [afbeelding 0.1](#) (blz. 3) van de microcontroller. Dit zorgt ervoor dat de microcontroller opnieuw opstart en dus de code nogmaals uitvoert.

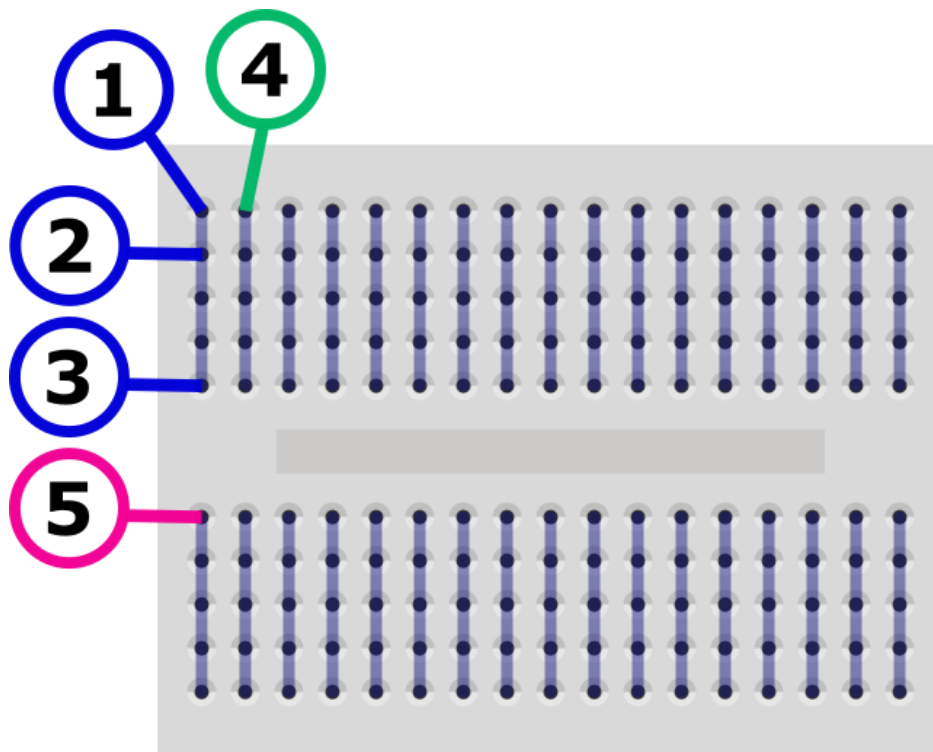
Knippert het lampje niet? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

# Het breadboard

In [Oefening 1](#) (blz. 16) 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 buzzer aansluiten

In deze oefening gaan we het breadboard gebruiken om een buzzer aan te sluiten en te laten piepen.

## Benodigheden

- Microcontroller
- USB kabel
- Computer
- Breadboard
- Buzzer
- Weerstand

## Verbindingen

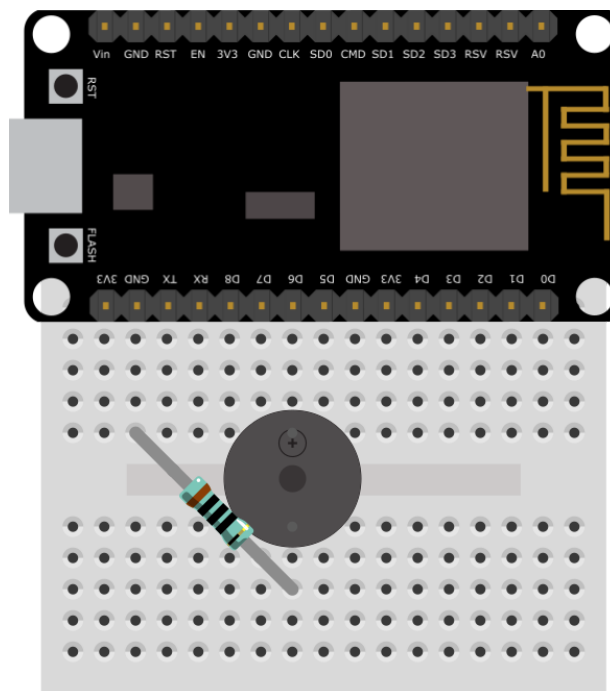
Microcontroller	Onderdeel	Via
GND	Buzzer korte pootje	Weerstand
D6	Buzzer lange pootje	-



In de diagrammen en tekst hebben we het vaak over de GND pin. Dit staat voor ground. Op sommige microcontrollers wordt deze pin aangegeven met de letters GND, op anderen staat simpelweg G.

## Stappen

1. Maak de schakeling uit afbeelding 2.1.



## Afbeelding 2.1



De buzzer heeft een lang pootje en een korter pootje. Het lange pootje is de plus (+) het korte pootje is de min (-). De plus (+) verbinden we in deze oefening met pin D6 van de microcontroller. De min verbinden we via de weerstand met de GND.

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

  digitalWrite(D6, HIGH);  // Zet de buzzer aan door een hoog signaal op D6 te zetten
  delay(1000);            // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D6, LOW);   // Zet de buzzer 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. 16). 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 buzzer die wij hebben aangesloten op pin D6.

3. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.

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

Wanneer alles goed is gegaan piept de buzzer die je op het breadboard hebt aangesloten twee keer.

Druk op de reset knop (zie [afbeelding 0.1](#) (blz. 3) van de microcontroller om de code nogmaals uit te voeren.

Piept de buzzer niet? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).





## Oefening 3: functies

In [Oefening 2](#) (blz. 19) hebben we de buzzer aan en weer uitgezet met het volgende stukje code:

```
digitalWrite(D6, HIGH); // Zet de buzzer aan door een hoog signaal op D6 te zetten
delay(1000);           // Wacht 1 seconde (1000 milliseconden)
digitalWrite(D6, LOW); // Zet de buzzer 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 buzzer aan en dan weer uit te zetten omzetten naar een functie dan ziet dat er als volgt uit:

```
void zetBuzzerAanEnUit() {
  digitalWrite(D6, LOW); // Zet de LED aan door een laag signaal op D4 zetten
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D6, HIGH); // Zet de LED uit door een hoog signaal op D4 te zetten
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)
}
```

In ons programma kunnen we nu de functie `zetBuzzerAanEnUit()` aanroepen. De inhoud van de functie wordt dan op die plek uitgevoerd. Het programma ziet er dan als volgt uit:

```
// Deze functie zet de buzzer aan en na 1 seconde weer uit
void zetBuzzerAanEnUit() {
  digitalWrite(D6, HIGH); // Zet de buzzer aan door een hoog signaal op D6 te zetten
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)
  digitalWrite(D6, LOW); // Zet de buzzer 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 Arduino wordt ingeschakeld
void setup() {
  pinMode(D6, OUTPUT); // We gaan pin D6 gebruiken als output
  zetBuzzerAanEnUit(); // Voer de inhoud van de functie zetBuzzerAanEnUit uit
  zetBuzzerAanEnUit(); // Voer de inhoud van de functie zetBuzzerAanEnUit uit
}

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

*Arduino code oefening 3*

## Benodigdheden

- Microcontroller
- USB kabel
- Computer
- Breadboard
- Buzzer
- Weerstand

## Stappen

1. Maak de schakeling die we in [Oefening 2](#) (blz. 19) gebruikt hebben.
2. Neem de code uit het blok [Arduino code oefening 3](#) (blz. 21) over in de Arduino IDE.
3. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
4. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan piept de buzzer die je op het breadboard hebt aangesloten net als in [Oefening 2](#) (blz. 19).

Piept de buzzer niet? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

## Oefening 4: de "loop" gebruiken

In [Oefening 3](#) (blz. 21) hebben we de functie `zetBuzzerAanEnUit()` twee keer aangeroepen om de buzzer te laten piepen. In deze oefening gaan we de buzzer laten piepen 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 `zetBuzzerAanEnUit()` in de `loop` te zetten wordt deze herhaald tot de microcontroller uit gaat.

### Benodigdheden

- Schakeling uit [oefening 2](#)
- USB kabel
- Computer

### Stappen

1. Maak de schakeling die we in [oefening 2](#) (blz. 19) 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 zetBuzzerAanEnUit() {  
  digitalWrite(D6, HIGH); // Zet de buzzer aan door een hoog signaal op D6 te zetten  
  delay(1000);           // Wacht 1 seconde (1000 milliseconden)  
  digitalWrite(D6, LOW); // Zet de buzzer 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 Arduino 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() {
  zetBuzzerAanEnUit();    // Voer de inhoud van de functie zetBuzzerAanEnUit uit
}

```

### 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.

3. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
4. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan piept de buzzer die je op het breadboard hebt aangesloten. In tegenstelling tot de vorige oefeningen blijft de buzzer dit keer aan en uit gaan tot je de stroom van de microcontroller afhaalt.

Piept de buzzer niet? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

## Oefening 5: variabelen

In deze oefening gaan we onze buzzer sneller laten piepen. 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);
```

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 `zetBuzzerAanEnUit()` is `delay()` ook een functie. De inhoud (definitie) 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	Aangeroeven door jou
zetBuzzerAanEnUit	Geschreven door jou	Aangeroeven door jou

Op de plekken in onze functie `zetBuzzerAanEnUit()` waar we `delay()` aanroepen kunnen we de waarde 1000 vervangen door een lager getal om de buzzer sneller te laten piepen.

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 waarden kan hebben	<code>true</code> of <code>false</code>
<code>char</code>	Een teken (character)	'A' of '!'
<code>float</code>	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 eenVoorbeeldTekens = 'c';
```

### Benodigdheden

- Schakeling uit [oefening 2](#)
- USB kabel
- Computer

## Stappen

1. Maak de schakeling die we in [oefening 2](#) (blz. 19) gebruikt hebben.
2. Neem de onderstaande code over in de Arduino IDE.

```
// Deze functie zet de buzzer aan en na 0,5 seconde weer uit
void zetBuzzerAanEnUit() {
  int aantalMillisecondenWachten = 500; // Variabele die bepaalt hoe lang we wachten
  digitalWrite(D6, HIGH); // Zet de buzzer 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 Arduino 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() {
  zetBuzzerAanEnUit(); // Roep de functie zetLedAanEnUit aan
}
```

### Arduino code oefening 5

3. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
4. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

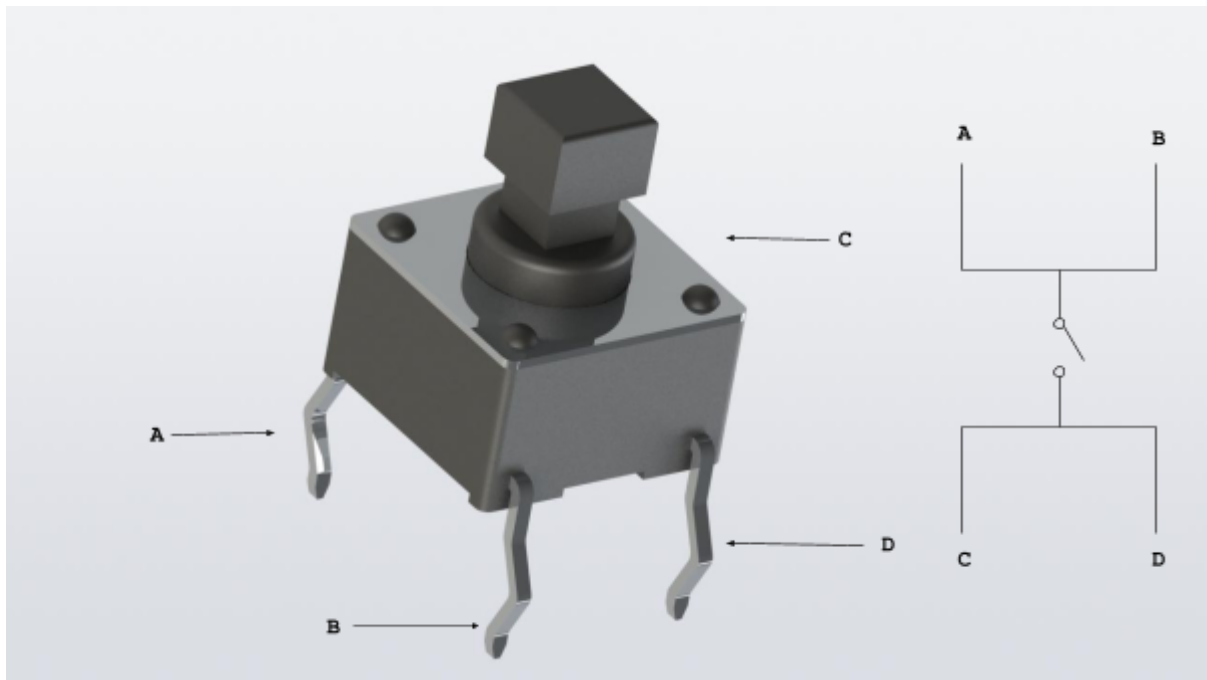
Wanneer alles goed is gegaan piept de buzzer nu twee keer zo snel aangezien we de tijd die de microcontroller wacht gehalveerd hebben.

Je kunt nu de snelheid waarmee de buzzer piept makkelijk aanpassen door de variabele `aantalMillisecondenWachten()` aan te passen en de code opnieuw te uploaden.

Piept de buzzer niet? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

## Oefening 6: Een knop toevoegen

We hebben nu geleerd hoe we de buzzer kunnen laten piepen op een manier die we van tevoren vastleggen in de code. Met een microcontroller kun je naast output genereren (zoals de buzzer laten piepen) ook input verwerken. In dit geval gaan we een knop gebruiken om de buzzer mee aan te sturen.



Afbeelding 6.1

Zoals te zien is in afbeelding 6.1, zijn de pootjes A en B intern met elkaar verbonden. Hetzelfde geldt voor de pootjes C en D. Tussen deze twee kanten van het knopje in, zit een schakelaar. Deze kun je beschouwen als een brug. Wanneer de schakelaar dicht is, kan de stroom er doorheen lopen. Wanneer de schakelaar open is, niet.

Waar we voor het genereren van output de `digitalWrite()` functie gebruikten, gebruiken we voor het uitlezen van input de `digitalRead()` functie.

```
digitalRead(D3);
```

Omdat we de buzzer aan willen zetten wanneer de knop ingedrukt is, maar uit willen zetten wanneer de knop niet ingedrukt is, moeten we een manier hebben om hier in de code onderscheid tussen te maken. Dit gaan we doen met een `if...else statement`.

```
if(conditie){  
  // Deze code wordt uitgevoerd wanneer wordt voldaan aan de conditie  
} else {  
  // Deze code wordt uitgevoerd wanneer niet wordt voldaan aan de conditie
```



```
}
```

Een **if...else statement** is een manier om de microcontroller een stuk code uit te laten voeren wanneer voldaan wordt aan een conditie, en een ander stuk wanneer hier niet aan voldaan wordt.

De **conditie** moet een uitdrukking zijn waarvan de uitkomst van het datatype **boolean** is. (Kijk nog eens naar [Het datatype van een variabele](#) voor de uitleg over datatypes.)

In ons geval willen we de status van de knop in de conditie van het if-statement zetten. Als de knop ingedrukt is, is de waarde van de pin **LOW**. Dit is dus wat we willen controleren.

In de Arduino programmeertaal kunnen we hiervoor de **==** operator gebruiken. Deze operator vergelijkt de waarden van de uitdrukkingen aan beide kanten. Als deze hetzelfde zijn is de uitkomst van de operatie **true**, als ze niet gelijk zijn is de uitkomst van de operatie **false**. In de Arduino programmeertaal zijn een heleboel operatoren bekend, in de onderstaande tabel staan een aantal voorbeelden.

<i>Operator</i>	<i>Voorbeeld uitdrukking</i>	<i>Voorbeeld uitkomst</i>
+	1 + 1	2
-	2 - 1	1
/	10 / 2	5
% (de rest-operator)	10 % 2 11 % 2 13 % 2	0 1 3
==	1 == 2 2 == 2 false == false	false true true
>	1 > 2 2 > 1 1 > 1	false true false

Nu we begrijpen hoe de **==** operator werkt, kunnen we deze gaan toepassen. We gaan straks de knop aansluiten op pin D3, de conditie die we tussen de haakjes van het if-statement gaan zetten ziet er dan zo uit:

```
digitalRead(D3) == LOW
```

## Benodigdheden

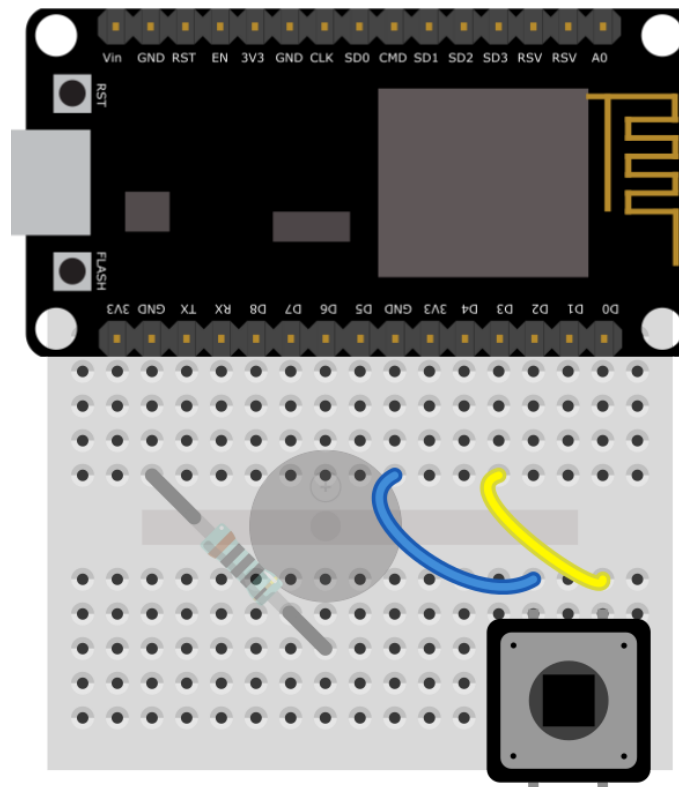
- Schakeling uit [Oefening 2](#)
- USB kabel
- Computer
- Jumper kabeltjes
- Drukknop met kapje

## Verbindingen

Microcontroller	Onderdeel	Via
GND	Button pin B	Blauwe jumper kabel
D3	Button pin D	Gele jumper kabel

## Stappen

1. Maak de schakeling uit afbeelding 6.2.



Afbeelding 6.2



De kleur van de kabeltjes maakt niet uit voor de werking van de schakeling. Deze kleuren gebruiken we puur ter verduidelijking van de schakeling. Zo wordt er voor 5V of 3V3 vaak rood gebruikt, en voor de GND vaak zwart, bruin of blauw. De andere kleuren worden gebruikt voor de signaal pins.

2. Klik het kapje op het knopje, zo kun je hem makkelijker indrukken.
3. Neem de onderstaande code over in de Arduino IDE.

```
// Deze functie zet de buzzer aan
void zetBuzzerAan() {
    digitalWrite(D6, HIGH);    // Zet de buzzer aan met een hoog signaal op D6
}

// Deze functie zet de buzzer uit
void zetBuzzerUit() {
    digitalWrite(D6, LOW);    // Zet de buzzer aan met een laag signaal op D6
}

// Dit gedeelte wordt slechts 1 keer uitgevoerd wanneer de Arduino wordt ingeschakeld
void setup() {
    pinMode(D6, OUTPUT);    // We gaan pin D6 gebruiken als output
    pinMode(D3, INPUT_PULLUP);    // We gaan pin D3 gebruiken als input met een pullupweerstand
}

// Dit gedeelte wordt herhaald tot de microcontroller wordt uitgezet
void loop() {
    if(digitalRead(D3) == LOW) { // Als het signaal op pin D3 LOW is:
        zetBuzzerAan();    // Roep de functie zetBuzzerAan() aan
    } else {                // Als het signaal op pin D3 niet LOW is:
        zetBuzzerUit();    // Roep de functie zetBuzzerUit() aan
    }
}
```

### Arduino code oefening 6

4. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
5. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan piept de buzzer nu alleen wanneer je op de knop drukt.

Gaat er nog iets mis? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

# Oefening 7: OLED display aansluiten

In deze oefening gaan we de OLED display aansluiten (het kleine beeldschermje). Dit scherm bestaat eigenlijk uit 8.192 hele kleine LED's. Door de LED's in een bepaalde volgorde aan te doen ontstaat er een beeld.

We zouden iedere LED individueel met de microcontroller kunnen aansturen. We hebben dan alleen wel heel veel pins nodig.

De microcontroller die wij gebruiken heeft te weinig pins om al deze LED's aan te sturen. Gelukkig is hier een oplossing voor.

De OLED display heeft een chip welke verbonden is met al de LED's. Deze chip zorgt ervoor dat we de LED's kunnen aansturen met slechts 2 pins (en een 5V en GND pin).

Om de chip van de OLED display aan te sturen gebruiken we in de Arduino IDE 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.

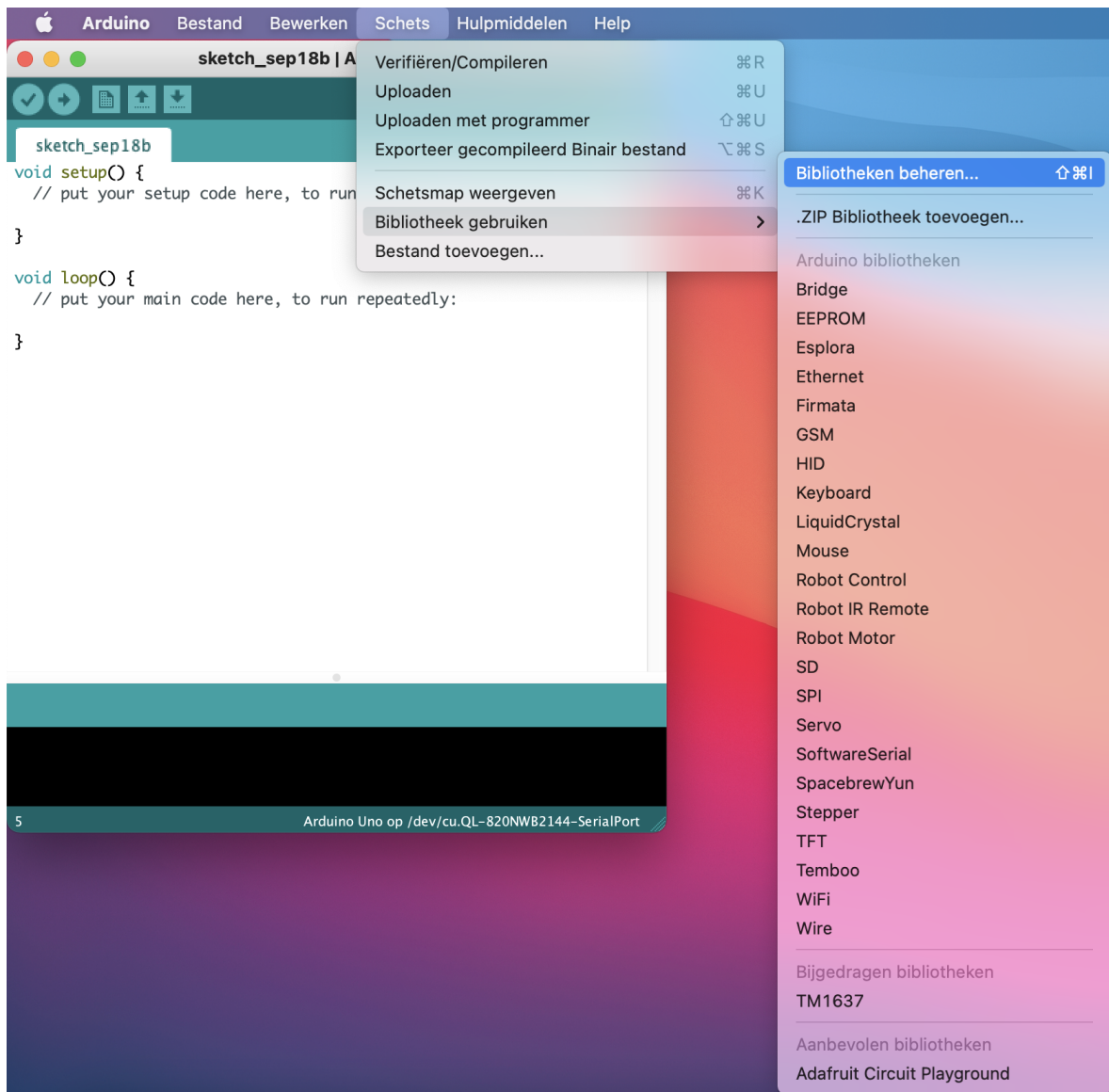
Voor onze OLED display gebruiken we de [Adafruit\\_SSD1306](#) library.

## Library installeren

Om de [Adafruit\\_SSD1306](#) library te gebruiken moeten we deze eerst installeren in de Arduino IDE.

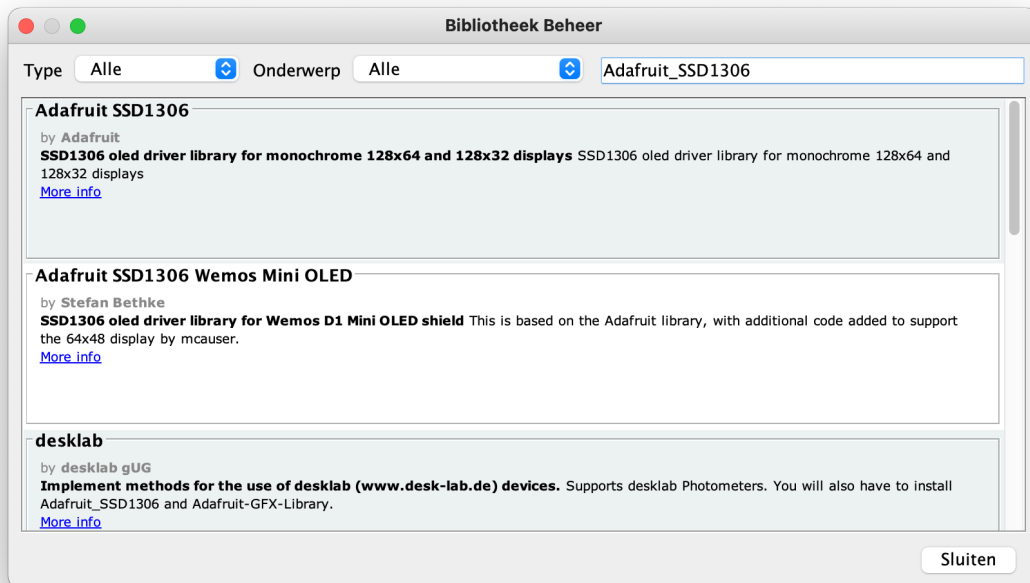
## Stappen

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



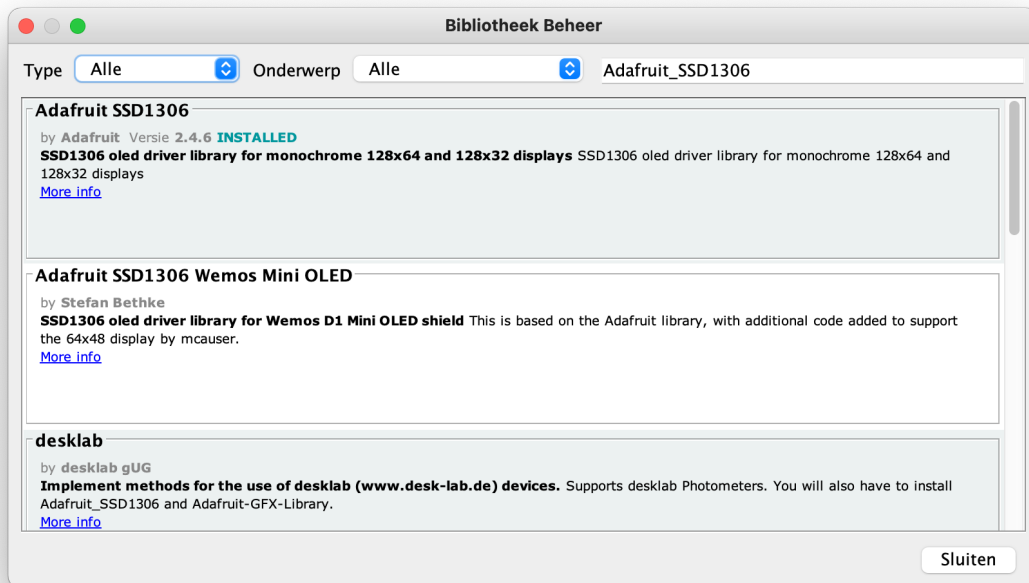
Afbeelding 7.1

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\_SSD1306*).



Afbeelding 7.2

3. Klik nu op de library met de naam **Adafruit\_SSD1306** en klik vervolgens op de knop "Installeren" om de library te installeren.
4. Er verschijnt nu een scherm dat vraagt of je ook de **Adafruit\_GFX** library wilt installeren. Klik hier op de knop "Install all".
5. 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 7.3

## Library gebruiken

Nu de `Adafruit_SSD1306` library geïnstalleerd is kunnen we deze gebruiken in onze code. We beginnen met eenvoudige code welke een punt op de display weergeeft.

## Benodigheden

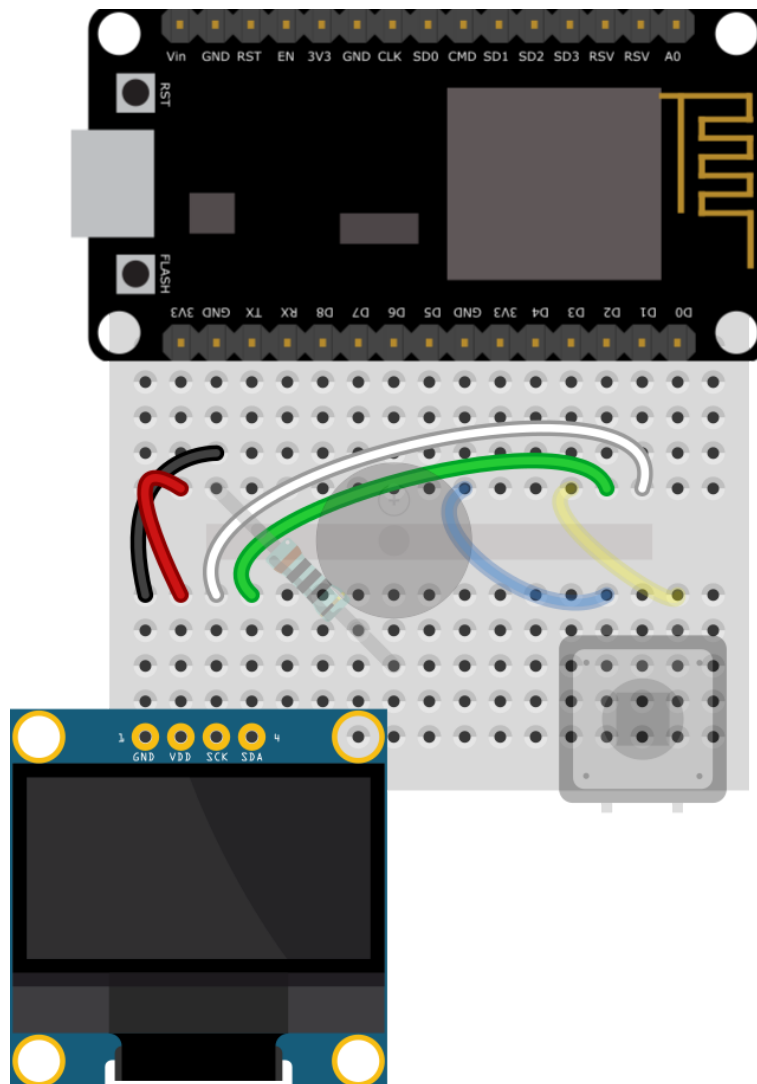
- Schakeling uit [oefening 6](#)
- USB kabel
- Jumper kabeltjes
- Computer
- OLED display

## Verbindingen

<i>Microcontroller</i>	<i>Onderdeel</i>	<i>Via</i>
D1	SCK	Witte jumper kabel
D2	SDA	Groene jumper kabel
3V3	VDD	Rode jumper kabel
GND	GND	Zwarte jumper kabel

## Stappen

1. Maak de schakeling uit afbeelding 7.4.



Afbeelding 7.4

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

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_SSD1306.h>

#define SCHERM_BREEDTE 128 // OLED display breedte, in pixels
#define SCHERM_HOOGTE 64 // OLED display hoogte, in pixels

#define OLED_RESET -1
#define SCHERM_ADRES 0x3C
Adafruit_SSD1306 display(SCHERM_BREEDTE, SCHERM_HOOGTE, &Wire, OLED_RESET);

void setup() {
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCHERM_ADRES)) {
    // Deze eeuwigdurende lus zorgt ervoor dat de applicatie niet verder gaat bij een fout
    for(;;);
  }
  // Maak het scherm leeg
  display.clearDisplay();

  // Teken een pixel met de coördinaten X = 10, Y = 10
```



```

display.drawPixel(10, 10, SSD1306_WHITE);

// Laat hetgeen we getekend hebben zien op het scherm
display.display();

// Wacht 2000 milliseconden (2 seconden)
delay(2000);

// Zet het scherm in negatief, de pixels die aan zijn gaan uit en andersom
display.invertDisplay(true);
// Wacht 1000 milliseconden (1 seconde)
delay(1000);
// Zet het scherm weer terug naar normaal.
display.invertDisplay(false);
// Wacht 1000 milliseconden (1 seconde)
delay(1000);
}

void loop() {
}

```

### Arduino code oefening 7

3. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
4. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan verschijnt er een puntje op het scherm. Daarna wordt het scherm even in negatief gezet (dan zijn alle pixels aan behalve degene die aan was) en als laatst verschijnt het puntje weer.

Gaat er nog iets mis? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

# Oefening 8: Tekst weergeven

De [Adafruit\\_SSD1306](#) library die we in [oefening 7](#) (blz. 32) gebruikt hebben zorgt ervoor dat we pixels op het scherm aan en uit kunnen zetten. Om meer complexe bewerkingen uit te voeren zoals het tonen van tekst hebben we nog een extra library nodig. We gaan daarvoor de [Adafruit\\_GFX](#) library gebruiken.

In [oefening 7](#) (blz. 32) hebben we deze library reeds geïnstalleerd. We hoeven dus geen extra library toe te voegen.

## Benodigdheden

- Schakeling uit [oefening 7](#)
- USB kabel
- Computer

## Stappen

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

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_SSD1306.h>

#define SCHERM_BREEDTE 128 // OLED display breedte, in pixels
#define SCHERM_HOOGTE 64 // OLED display hoogte, in pixels

#define OLED_RESET -1
#define SCHERM_ADRES 0x3C
Adafruit_SSD1306 display(SCHERM_BREEDTE, SCHERM_HOOGTE, &Wire, OLED_RESET);

void setup() {
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCHERM_ADRES)) {
    // Deze eeuwigdurende lus zorgt ervoor dat de applicatie niet verder gaat bij een fout
    for(;;);
  }
  // Maak het scherm leeg
  display.clearDisplay();

  display.setTextSize(1); // Zet het tekstformaat op 1
  display.setTextColor(WHITE); // Schrijf witte letters
  display.setCursor(0,0); // Begin met schrijven op positie 0,0 (links boven)
  display.println("Awesome Makes"); // Schrijf de tekst Awesome Makes

  display.setTextColor(BLACK, WHITE); // Schrijf zwarte letters met een witte achtergrond
  display.println("IOT Smart Display"); // Schrijf de tekst IOT Smart Display

  display.setTextSize(2); // Zet het tekstformaat op 2 (2x zo groot)
  display.setTextColor(WHITE); // Schrijf witte letters
  display.setCursor(10,40); // Begin met schrijven op positie 10,40
  display.print("Awesome!"); // Schrijf de tekst Awesome!

  display.display(); // Toon de teksten op het scherm
}

void loop() {
}
```

*Arduino code oefening 8*

2. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
3. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan verschijnen er nu drie teksten op het scherm.

Gaat er nog iets mis? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

# Oefening 9: Wifi

In deze oefening laten we de microcontroller verbinding maken met jouw wifi-netwerk en tonen we de gegevens van de verbinding op de display.

In de volgende oefeningen gaan we de wifi-verbinding gebruiken om de microcontroller informatie op te laten halen van het internet.

## Benodigdheden

- Schakeling uit [oefening 7](#)
- USB kabel
- Computer

## Stappen

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

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_SSD1306.h>

#define SCHERM_BREEDTE 128 // OLED display breedte, in pixels
#define SCHERM_HOOGTE 64 // OLED display hoogte, in pixels

#define OLED_RESET -1
#define SCHERM_ADRES 0x3C
Adafruit_SSD1306 display(SCHERM_BREEDTE, SCHERM_HOOGTE, &Wire, OLED_RESET);

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

void setup() {
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCHERM_ADRES)) {
    // Deze eeuwigdurende lus zorgt ervoor dat de applicatie niet verder gaat bij een fout
    for(;;);
  }
  maakVerbindingMetWifi();
}

void loop() {
}

// Deze functie maakt verbinding met het netwerk
void maakVerbindingMetWifi() {
  display.clearDisplay(); // Maak het scherm leeg
  display.setTextColor(WHITE); // Schrijf witte letters
  display.setTextSize(1); // Zet het tekstformaat op 1
  display.setCursor(0,0); // Begin met schrijven op positie 0,0
  display.println("Verbinden met wifi"); // Schrijf tekst naar het scherm
  display.display(); // Toon de teksten op het scherm

  WiFi.mode(WIFI_STA); // Type wifi verbinding instellen

  WiFi.begin(ssid, password); // Gebruik WiFi.begin om verbinding te maken
  while (WiFi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
```

```

display.print(".");           // Schrijf een punt op het scherm
display.display();           // Toon de teksten op het scherm
delay(500);                   // Wacht een halve seconde
}

display.clearDisplay();       // Maak het scherm leeg
display.setCursor(0,0);       // Begin met schrijven op positie 0,0
display.println("Verbonden met wifi"); // Schrijf tekst naar het scherm
display.print("SSID: ");      // Schrijf tekst naar het scherm
display.println(WiFi.SSID()); // Schrijf het SSID naar het scherm
display.print("IP: ");        // Schrijf tekst naar het scherm
display.println(WiFi.localIP()); // Schrijf het ip-adres naar het scherm
display.display();            // Toon de teksten op het scherm
}

```

### Arduino code oefening 9

- In de code moet je nog jouw netwerknaam en wachtwoord invullen. De microcontroller kan anders geen verbinding maken met de wifi. De plek waar dit moet gebeuren ziet er als volgt uit:

```

const char* ssid = "JOUW-NETWERKNAAM";
const char* password = "JOUW-WACHTWOORD";

```

Wijzig JOUW-NETWERKNAAM (de dubbele aanhalingstekens moeten blijven staan) naar de naam van jouw wifi netwerk (SSID) en verander JOUW-WACHTWOORD naar het wachtwoord van jouw wifi netwerk. Weet je niet wat je hier in moet vullen? Vraag dan om hulp aan de persoon die jouw wifi beheert.

Na het invullen zou de code er bijvoorbeeld als volgt uit kunnen zien:

```

const char* ssid = "Ziggo87D3Z9";
const char* password = "Qr8c23zI4";

```

- Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
- Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan verschijnen er punten op de display 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.

Verschijnen de gegevens niet op de display? Reset dan de microcontroller door op de reset knop te drukken. Zie hiervoor

[afbeelding 0.1](#) (blz. 3). Werkt het programma daarna nog niet?  
Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

# Oefening 10: Gegevens van het internet ophalen

Nu we verbinding met het wifi netwerk kunnen maken, kunnen we ook gegevens van het internet gaan lezen. Wanneer er gegevens worden uitgewisseld via het internet, gebeurt dit vaak via een API. Dit staat voor Application Programming Interface. We beginnen door de inhoud van de website <http://tijd.awesomemakes.com> op te halen en op de display weer te geven.

De website <http://tijd.awesomemakes.com> bevat de huidige tijd. Je kan dit zelf bekijken door de website in je browser te openen.

## Benodigheden

- Schakeling uit [oefening 7](#)
- USB kabel
- Computer

## Stappen

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

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_SSD1306.h>

#define SCHERM_BREEDTE 128 // OLED display breedte, in pixels
#define SCHERM_HOOGTE 64 // OLED display hoogte, in pixels
#define OLED_RESET -1
#define SCHERM_ADRES 0x3C
Adafruit_SSD1306 display(SCHERM_BREEDTE, SCHERM_HOOGTE, &Wire, OLED_RESET);

WiFiClientSecure client;

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

const String server = "tijd.awesomemakes.com"; // Het adres van de server
const String url = "/"; // De url waar de informatie te vinden is

void setup() {
  if (!display.begin(SSD1306_SWITCHCAPVCC, SCHERM_ADRES)) {
    // Deze eeuwigdurende lus zorgt ervoor dat de applicatie niet verder gaat bij een fout
    for (;;)
  }
  maakVerbindingMetWifi();
  client.setInsecure(); // Gebruik geen beveiligde verbinding
}

void loop() {
  gegevensOphalenEnWeergeven(); // Haal de tijd op van de server en geef deze weer op de display
  delay(1000 * 60); // Wacht 1 minuut (60000 milliseconden)
}

// Deze functie maakt verbinding met het netwerk
void maakVerbindingMetWifi() {
```

```

display.clearDisplay();           // Maak het scherm leeg
display.setTextColor(WHITE);     // Schrijf witte letters
display.setTextSize(1);          // Zet het tekstformaat op 1
display.setCursor(0, 0);         // Begin met schrijven op positie 0,0
display.println("Verbinden met wifi"); // Schrijf tekst naar het scherm
display.display();               // Toon de teksten op het scherm

WiFi.mode(WIFI_STA);             // Type wifi verbinding instellen

WiFi.begin(ssid, password);       // Gebruik WiFi.begin om verbinding te maken
while (WiFi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
    display.print(".");           // Schrijf een punt op het scherm
    display.display();           // Toon de teksten op het scherm
    delay(500);                  // Wacht een halve seconde
}

display.clearDisplay();           // Maak het scherm leeg
display.setCursor(0, 0);         // Begin met schrijven op positie 0,0
display.println("Verbonden met wifi"); // Schrijf tekst naar het scherm
display.print("SSID: ");         // Schrijf tekst naar het scherm
display.println(WiFi.SSID());     // Schrijf het SSID naar het scherm
display.print("IP: ");           // Schrijf tekst naar het scherm
display.println(WiFi.localIP()); // Schrijf het ip-adres naar het scherm
display.display();               // Toon de teksten op het scherm
}

// Deze functie haalt gegevens op van de server en geeft deze weer op de display
void gegevensOphalenEnWeergeven() {
    display.clearDisplay();       // Maak het scherm leeg
    display.setTextSize(1);       // Zet het tekstformaat op 1
    display.setCursor(0, 0);      // Begin met schrijven op positie 0,0 (links boven)

    // Uitvoeren als er een verbinding gemaakt is met de server
    if (client.connect(server, 443)) {
        display.print("Verbonden met server"); // Schrijf tekst naar het scherm
    } // Uitvoeren als er een fout is opgetreden
    else {
        display.print("Kan geen verbinding maken met server"); // Schrijf tekst naar het scherm
    }
    display.display();            // Toon de tekst op het scherm

    // Een website opvragen bij de server
    client.println("GET https://" + server + url + " HTTP/1.0");
    client.println("Host: " + server);
    client.println("Connection: close");
    client.println();

    // Wachten tot de server klaar is om de gegevens te versturen
    while (client.connected()) {
        String line = client.readStringUntil('\n');
        if (line == "\r") {
            break;
        }
    }


    String response = client.readString(); // De gegevens van de server uitlezen
    display.clearDisplay();               // Maak het scherm leeg
    display.setCursor(6, 20);            // Begin met schrijven op positie 6,20
    display.setTextSize(4);              // Zet het tekstformaat op 4
    display.print(response);              // Schrijf de opgehaalde gegevens naar het scherm
    display.display();                    // Toon de tekst op het scherm

    client.stop();                        // Stop de verbinding met de server
}

```



## Arduino code oefening 10

2. Vul net als in [oefening 9](#) jouw netwerknaam en wachtwoord in.
3. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
4. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan verschijnen nu de huidige tijd op de display. Iedere 60 seconden wordt de tijd opnieuw opgehaald bij de server en op het scherm weergegeven.

Gaat er nog iets mis? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

# Oefening 11: Alarm

Met de code uit [oefening 10](#) halen we iedere 60 seconden de huidige tijd op bij de server. Met een eenvoudige aanpassing kunnen we nu een wekker af laten gaan op een vooraf opgegeven tijd.

## Benodigdheden

- Schakeling uit [oefening 7](#)
- USB kabel
- Computer

## Stappen

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

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_SSD1306.h>

#define SCHERM_BREEDTE 128 // OLED display breedte, in pixels
#define SCHERM_HOOGTE 64 // OLED display hoogte, in pixels

#define OLED_RESET -1
#define SCHERM_ADRES 0x3C
Adafruit_SSD1306 display(SCHERM_BREEDTE, SCHERM_HOOGTE, &Wire, OLED_RESET);

WiFiClientSecure client;

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

const String server = "tijd.awesomemakes.com"; // Het adres van de server
const String url = "/"; // De url waar de informatie te vinden is

void setup() {
  pinMode(D6, OUTPUT);
  pinMode(D3, INPUT_PULLUP);

  if (!display.begin(SSD1306_SWITCHCAPVCC, SCHERM_ADRES)) {
    // Deze eeuwigdurende lus zorgt ervoor dat de applicatie niet verder gaat bij een fout
    for (;;)
  }
  maakVerbindingMetWifi();
  client.setInsecure(); // Gebruik geen beveiligde verbinding
}

void loop() {
  gegevensOphalenEnWeergeven(); // Haal de tijd op van de server en geef deze weer op de display
  delay(1000 * 60); // Wacht 1 minuut (60000 milliseconden)
}

// Deze functie maakt verbinding met het netwerk
void maakVerbindingMetWifi() {
  display.clearDisplay(); // Maak het scherm leeg
  display.setTextColor(WHITE); // Schrijf witte letters
  display.setTextSize(1); // Zet het tekstformaat op 1
  display.setCursor(0, 0); // Begin met schrijven op positie 0,0
```

```

display.println("Verbinden met wifi"); // Schrijf tekst naar het scherm
display.display(); // Toon de teksten op het scherm

WiFi.mode(WIFI_STA); // Type wifi verbinding instellen

WiFi.begin(ssid, password); // Gebruik WiFi.begin om verbinding te maken
while (WiFi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
    display.print("."); // Schrijf een punt op het scherm
    display.display(); // Toon de teksten op het scherm
    delay(500); // Wacht een halve seconde
}

display.clearDisplay(); // Maak het scherm leeg
display.setCursor(0, 0); // Begin met schrijven op positie 0,0
display.println("Verbonden met wifi"); // Schrijf tekst naar het scherm
display.print("SSID: "); // Schrijf tekst naar het scherm
display.println(WiFi.SSID()); // Schrijf het SSID naar het scherm
display.print("IP: "); // Schrijf tekst naar het scherm
display.println(WiFi.localIP()); // Schrijf het ip-adres naar het scherm
display.display(); // Toon de teksten op het scherm
}

// Deze functie haalt gegevens op van de server en geeft deze weer op de display
void gegevensOphalenEnWeergeven() {
    // Uitvoeren als er een verbinding gemaakt is met de server
    bool verbindingGeslaagd = client.connect(server, 443);
    // Uitvoeren als er een fout is opgetreden
    if (!verbindingGeslaagd) {
        display.clearDisplay(); // Maak het scherm leeg
        display.setTextSize(1); // Zet het tekstformaat op 1
        display.setCursor(0, 0); // Begin met schrijven op positie 0,0 (links boven)
        display.print("Kan geen verbinding maken met server"); // Schrijf tekst naar het scherm
        display.display(); // Toon de tekst op het scherm
    }

    // Een website opvragen bij de server
    client.println("GET https://" + server + url + " HTTP/1.0");
    client.println("Host: " + server);
    client.println("Connection: close");
    client.println();

    // Wachten tot de server klaar is om de gegevens te versturen
    while (client.connected()) {
        String line = client.readStringUntil('\n');
        if (line == "\r") {
            break;
        }
    }

    String response = client.readString(); // De gegevens van de server uitlezen
    display.clearDisplay(); // Maak het scherm leeg
    display.setCursor(6, 20); // Begin met schrijven op positie 6,20
    display.setTextSize(4); // Zet het tekstformaat op 4
    display.print(response); // Schrijf de opgehaalde gegevens naar het scherm
    display.display(); // Toon de tekst op het scherm

    client.stop(); // Stop de verbinding met de server

    //Uitvoeren als de ingestelde tijd bereikt is
    if (response == "13:00") {
        alarmLatenAfgaan(); // Het alarm af laten gaan
    }
}

// Deze functie laat de buzzer aan en uit gaan tot de knop ingedrukt owrdt.
void alarmLatenAfgaan() {

```

```

int aantalMillisecondenWachten = 500; // Variabele die bepaalt hoe lang we wachten

// Onderstaande uitvoeren zolang het signaal op D3 HIGH is (knop is niet ingedrukt)
while (digitalRead(D3) == HIGH) {
    digitalWrite(D6, HIGH); // Zet de buzzer aan met een hoog signaal op D6
    delay(aantalMillisecondenWachten); // Wacht het opgegeven aantal milliseconden
    digitalWrite(D6, LOW); // Zet de LED uit met een laag signaal op D6
    delay(aantalMillisecondenWachten); // Wacht het opgegeven aantal milliseconden
}
}

```

### Arduino code oefening 11

2. Vul net als in de voorgaande oefeningen jouw netwerknnaam en wachtwoord in.
3. Met het volgende stuk code controleren we of het al tijd is om het alarm af te laten gaan:

```

if(response == "13:00"){
    alarmLatenAfgaan();
}

```

door de waarde 13:00 aan te passen kan je het alarm op een ander moment af laten gaan. Vul nu een tijd in die bijvoorbeeld 5 minuten in de toekomst ligt.

4. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
5. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan verschijnt nu de huidige tijd weer op de display. Op het opgegeven tijdstip zou de buzzer aan moeten gaan. Om de buzzer weer uit te zetten houdt je de knop ingedrukt tot je de buzzer niet meer hoort.

Gaat er nog iets mis? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

# Oefening 12: Bitcoin koers ophalen

De informatie die we in [oefening 10](#) en [oefening 11](#) ophaalden hebben wij voor je klaargezet maar er zijn veel meer gegevensbronnen waar je interessante informatie kunt ophalen.

In deze oefening gaan we de huidige koers van de Bitcoin ophalen en op de display tonen. Met de technieken die we hier gebruiken kan je ook andere gegevens ophalen. Hoe je dit doet zullen we in de volgende oefening toelichten.

## Benodigdheden

- Schakeling uit [oefening 7](#)
- USB kabel
- Computer

## Stappen

1. In deze oefening gaan we de URL <https://api.coindesk.com/v1/bpi/currentprice.json> gebruiken om data op te halen. Open deze URL in de browser. Je ziet hier welke data de microcontroller gaat ontvangen.



Het formaat waarin de data is opgeslagen noemen we JSON. Wanneer je meer wilt weten over dit dataformaat neem dan een kijkje op [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp).

De informatie is op deze manier echter niet erg overzichtelijk. Wanneer je Google Chrome gebruikt kan je een plugin zoals JSON Formatter (<https://chrome.google.com/webstore/detail/json-formatter/bcjindccaagfppajmmgkqhgoa>) gebruiken om de gegevens beter leesbaar te maken.

Zoals je ziet levert deze website veel meer informatie dan dat wij gebruiken. Wij zijn alleen geïnteresseerd in de huidige Bitcoin koers in euros. Deze informatie vinden we onder "bpi" > "EUR" > "rate". Op regel 97 van de code hieronder zie je dat we deze informatie uit het JSON-object halen en opslaan in de variabele "huidigePrijs":

```
float huidigePrijs = jsonObject["bpi"]["EUR"]["rate_float"].as<float>();
```

2. De URL die wij hier gebruiken bestaat uit een aantal verschillende onderdelen:

```
https://api.coindesk.com/v1/bpi/currentprice.json
```

**Protocol**    **Domeinnaam**                    **Padnaam**

Deze verschillende onderdelen gaan we in de code hieronder gebruiken.

Op regel 21 vullen we de domeinnaam van de server in. In dit geval is dat `api.coindesk.com`.

Op regel 22 vullen we de padnaam in. In dit geval is dat `/v1/bpi/currentprice.json` (vergeet de slash aan het begin niet).

Het protocol staat op regel 81. Deze code kan alleen overweg met het https protocol. Vrijwel alle moderne websites gebruiken dit protocol.

3. Nu je in hoofdlijnen begrijpt hoe de code werkt kunnen we deze bijna gaan uploaden naar de microcontroller. Voor we dat doen hebben we nog een library nodig zodat de microcontroller ook het JSON formaat kan lezen. Herhaal de stappen onder [library installeren](#) uit oefening 7. Vul hierbij voor de library naam in: **ArduinoJson**. Kies de library gemaakt door **Benoit Blanchon**.
4. Neem de onderstaande code over in de Arduino IDE.

```
1 #include <ArduinoJson.h>
2 #include <ESP8266WiFi.h>
3 #include <WiFiClientSecure.h>
4 #include <SPI.h>
5 #include <Wire.h>
6 #include <Adafruit_SSD1306.h>
7
8 #define SCHERM_BREEDTE 128 // OLED display breedte, in pixels
9 #define SCHERM_HOOGTE 64 // OLED display hoogte, in pixels
10
11 #define OLED_RESET -1
12 #define SCHERM_ADRES 0x3C
13 Adafruit_SSD1306 display(SCHERM_BREEDTE, SCHERM_HOOGTE, &Wire, OLED_RESET);
14
15 WiFiClientSecure client;
16 DynamicJsonDocument jsonObject(2048);
17
18 const char* ssid = "JOUW-NETWERKNAAM"; // Vul hier jouw eigen netwerknaam (ssid) in
19 const char* password = "JOUW-WACHTWOORD"; // Vul hier jouw eigen netwerkwachtwoord in
20
21 const String server = "api.coindesk.com"; // Het adres van de server
22 const String url = "/v1/bpi/currentprice.json"; // De url waar de informatie te vinden is
23
24 void setup() {
25     if (!display.begin(SSD1306_SWITCHCAPVCC, SCHERM_ADRES)) {
26         // Deze eeuwigdurende lus zorgt ervoor dat de applicatie niet verder gaat bij een fout
27         for (;;);
28     }
29     maakVerbindingMetWifi();
30     client.setInsecure(); // Gebruik geen beveiligde verbinding
31 }
32
```

```

33 void loop() {
34     gegevensOphalenEnWeergeven(); // Haal de tijd op van de server en geef deze weer op de
35     display
36     delay(1000 * 60);           // Wacht 1 minuut (60000 milliseconden)
37 }
38
39 // Deze functie maakt verbinding met het netwerk
40 void maakVerbindingMetWifi() {
41     display.clearDisplay();           // Maak het scherm leeg
42     display.setTextColor(WHITE);     // Schrijf witte letters
43     display.setTextSize(1);         // Zet het tekstformaat op 1
44     display.setCursor(0, 0);        // Begin met schrijven op positie 0,0
45     display.println("Verbinden met wifi"); // Schrijf tekst naar het scherm
46     display.display();              // Toon de teksten op het scherm
47
48     WiFi.mode(WIFI_STA);           // Type wifi verbinding instellen
49
50     WiFi.begin(ssid, password);     // Gebruik WiFi.begin om verbinding te maken
51     while (WiFi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
52         display.print(".");         // Schrijf een punt op het scherm
53         display.display();          // Toon de teksten op het scherm
54         delay(500);                 // Wacht een halve seconde
55     }
56
57     display.clearDisplay();           // Maak het scherm leeg
58     display.setCursor(0, 0);        // Begin met schrijven op positie 0,0
59     display.println("Verbonden met wifi"); // Schrijf tekst naar het scherm
60     display.print("SSID: ");        // Schrijf tekst naar het scherm
61     display.println(WiFi.SSID());   // Schrijf het SSID naar het scherm
62     display.print("IP: ");          // Schrijf tekst naar het scherm
63     display.println(WiFi.localIP()); // Schrijf het ip-adres naar het scherm
64     display.display();              // Toon de teksten op het scherm
65 }
66
67 // Deze functie haalt gegevens op van de server en geeft deze weer op de display
68 void gegevensOphalenEnWeergeven() {
69     // Uitvoeren als er een verbinding gemaakt is met de server
70     bool verbindingGeslaagd = client.connect(server, 443);
71     // Uitvoeren als er een fout is opgetreden
72     if (!verbindingGeslaagd) {
73         display.clearDisplay();     // Maak het scherm leeg
74         display.setTextSize(1);     // Zet het tekstformaat op 1
75         display.setCursor(0, 0);    // Begin met schrijven op positie 0,0 (links boven)
76         display.print("Kan geen verbinding maken met server"); // Schrijf tekst naar het scherm
77         display.display();          // Toon de tekst op het scherm
78     }
79
80     // Een website opvragen bij de server
81     client.println("GET https://" + server + url + " HTTP/1.0");
82     client.println("Host: " + server);
83     client.println("Connection: close");
84     client.println();
85
86     // Wachten tot de server klaar is om de gegevens te versturen
87     while (client.connected()) {
88         String line = client.readStringUntil('\n');
89         if (line == "\r") {
90             break;
91         }
92     }
93
94     String gegevens = client.readString(); // De gegevens van de server uitlezen
95     deserializeJson(jsonObject, gegevens); // De gegevens omzetten naar een JSON object
96     // De huidige prijs uitlezen
97     float huidigePrijs = jsonObject["bpi"]["EUR"]["rate_float"].as<float>();
98     display.clearDisplay();              // Maak het scherm leeg

```

```

99   display.setCursor(16, 27);           // Begin met schrijven op positie 16,27
100  display.setTextSize(2);             // Zet het tekstformaat op 2
101  display.print(huidigePrijs);         // Schrijf de huidige prijs gegevens naar het scherm
102  display.display();                   // Toon de tekst op het scherm
103
104  client.stop();                       // Stop de verbinding met de server
105  }

```

### Arduino code oefening 12

5. Vul net als in de voorgaande oefeningen jouw netwerknnaam en wachtwoord in.
6. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
7. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan verschijnt na enige tijd de huidige Bitcoin koers in euro's op de display. Iedere minuut wordt de waarde ververst.

Gaat er nog iets mis? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).



# Oefening 13: De temperatuur tonen via een API met authenticatie

Je hebt nu gezien hoe je informatie van het internet kunt ophalen en kunt tonen op de display. Op deze manier zul je de meeste API's gewoon kunnen aanroepen.

Sommige API's vereisen, net als sommige websites, authenticatie voordat ze bepaalde informatie teruggeven. Op een website moet je dan bijvoorbeeld inloggen met je e-mailadres en wachtwoord. Om je te authenticeren voor een API gebruik je vaak een zogenoemde API-key.

Een dergelijke API-key kan je vaak opvragen op de website waar je de informatie uiteindelijk op wilt gaan halen. Je krijgt dan vaak een lange combinatie van letters en cijfers welke werken als een soort wachtwoord. In jouw programmacode stuur je de API key dan mee tijdens de aanroep en zo weet de API server dat jij het bent.

Wij gaan nu de KNMI API gebruiken om de huidige temperatuur in jouw woonplaats te tonen op de display.

## Benodigheden

- Schakeling uit [oefening 7](#)
- USB kabel
- Computer

## Stappen

1. In deze oefening gaan we de weerinformatie ophalen bij het KNMI via <http://weerlive.nl/delen.php>.

Hiervoor is het eerst nodig om je te registreren. Dit kun je doen via <https://weerlive.nl/api/toegang/>. Wanneer je de stappen hebt gevolgd ontvang je een mail waarmee je je account definitief kunt activeren.

In deze mail staat ook de API-key. Kun je de mail niet vinden? Het kan zijn dat de mail in je spambox terecht is gekomen.

2. De URL die wij in de vorige oefeningen gebruikten bestaat uit 3 verschillende onderdelen. Dit waren het protocol, de domeinnaam en de padnaam. Voor de KNMI API wordt dit:

<https://weerlive.nl/api/json-data-10min.php>

Protocol    Domeinnaam                      Padnaam

Hier voegen we nu nog een onderdeel aan toe, namelijk: parameters. In deze parameters kunnen we straks de API key en jouw woonplaats kwijt. Een parameter wordt achter de URL geplakt in de volgende vorm:

**?key=JOUW API-KEY&locatie=JOUW LOCATIE**

We beginnen met een **?** om aan te geven dat hierna parameters volgen. Daarna komt de naam van de eerste parameter, in dit geval **key**. De naam van de parameter wordt gevolgd door een **=** waarna de waarde van de parameter komt. Hier vul jij straks jouw API-key in. Als er nog meer parameters volgen worden deze steeds gescheiden met een **&** zoals je hierboven kunt zien.

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

```
1 #include <ArduinoJson.h>
2 #include <ESP8266WiFi.h>
3 #include <WiFiClientSecure.h>
4 #include <SPI.h>
5 #include <Wire.h>
6 #include <Adafruit_SSD1306.h>
7
8 #define SCHERM_BREEDTE 128 // OLED display breedte, in pixels
9 #define SCHERM_HOOGTE 64 // OLED display hoogte, in pixels
10
11 #define OLED_RESET -1
12 #define SCHERM_ADRES 0x3C
13 Adafruit_SSD1306 display(SCHERM_BREEDTE, SCHERM_HOOGTE, &Wire, OLED_RESET);
14
15 WiFiClientSecure client;
16 DynamicJsonDocument jsonObject(2048);
17
18 const char* ssid = "JOUW-NETWERKNAAM"; // Vul hier jouw eigen netwerknaam (ssid) in
19 const char* password = "JOUW-WACHTWOORD"; // Vul hier jouw eigen netwerkwachtwoord in
20
21 const String server = "weerlive.nl"; // Het adres van de server
22 const String url = "/api/json-data-10min.php"; // De url waar de informatie te vinden is
23 const String parameters = "?key=JOUWKEY&locatie=JOUWLOCATIE"; // Jouw parameters
24
25 void setup() {
26     if (!display.begin(SSD1306_SWITCHCAPVCC, SCHERM_ADRES)) {
27         // Deze eeuwigdurende lus zorgt ervoor dat de applicatie niet verder gaat bij een fout
28         for (;;);
29     }
30     maakVerbindingMetWifi();
31     client.setInsecure(); // Gebruik geen beveiligde verbinding
32 }
33
34 void loop() {
35     gegevensOphalenEnWeergeven(); // Haal de tijd op van de server en geef deze weer
36     delay(1000 * 60 * 10); // Wacht 10 minuten (60000 milliseconden)
37 }
38
39 // Deze functie maakt verbinding met het netwerk
40 void maakVerbindingMetWifi() {
41     display.clearDisplay(); // Maak het scherm leeg
42     display.setTextColor(WHITE); // Schrijf witte letters
43     display.setTextSize(1); // Zet het tekstformaat op 1
44     display.setCursor(0, 0); // Begin met schrijven op positie 0,0
```

```

45     display.println("Verbinden met wifi"); // Schrijf tekst naar het scherm
46     display.display(); // Toon de teksten op het scherm
47
48     WiFi.mode(WIFI_STA); // Type wifi verbinding instellen
49
50     WiFi.begin(ssid, password); // Gebruik WiFi.begin om verbinding te maken
51     while (WiFi.status() != WL_CONNECTED) { // Herhalen tot er verbinding is
52         display.print("."); // Schrijf een punt op het scherm
53         display.display(); // Toon de teksten op het scherm
54         delay(500); // Wacht een halve seconde
55     }
56
57     display.clearDisplay(); // Maak het scherm leeg
58     display.setCursor(0, 0); // Begin met schrijven op positie 0,0
59     display.println("Verbonden met wifi"); // Schrijf tekst naar het scherm
60     display.print("SSID: "); // Schrijf tekst naar het scherm
61     display.println(WiFi.SSID()); // Schrijf het SSID naar het scherm
62     display.print("IP: "); // Schrijf tekst naar het scherm
63     display.println(WiFi.localIP()); // Schrijf het ip-adres naar het scherm
64     display.display(); // Toon de teksten op het scherm
65 }
66
67 // Deze functie haalt gegevens op van de server en geeft deze weer op de display
68 void gegevensOphalenEnWeergeven() {
69     // Uitvoeren als er een verbinding gemaakt is met de server
70     bool verbindingGeslaagd = client.connect(server, 443);
71     // Uitvoeren als er een fout is opgetreden
72     if (!verbindingGeslaagd) {
73         display.clearDisplay(); // Maak het scherm leeg
74         display.setTextSize(1); // Zet het tekstformaat op 1
75         display.setCursor(0, 0); // Begin met schrijven op positie 0,0 (links
76         boven)
77         display.print("Kan geen verbinding maken met server"); // Schrijf tekst naar het scherm
78         display.display(); // Toon de tekst op het scherm
79     }
80
81     // Een website opvragen bij de server
82     client.println("GET https://" + server + url + parameters + " HTTP/1.0");
83     client.println("Host: " + server);
84     client.println("Connection: close");
85     client.println();
86
87     // Wachten tot de server klaar is om de gegevens te versturen
88     while (client.connected()) {
89         String line = client.readStringUntil('\n');
90         if (line == "\r") {
91             break;
92         }
93     }
94
95     String gegevens = client.readString(); // De gegevens van de server uitlezen
96     deserializeJson(jsonObject, gegevens); // De gegevens omzetten naar een JSON object
97     String temperatuur = jsonObject["liveweer"][0]["temp"].as<String>(); // Temperatuur uitlezen
98     display.clearDisplay(); // Maak het scherm leeg
99     display.setCursor(4, 20); // Begin met schrijven op positie 4,20
100    display.setTextSize(3); // Zet het tekstformaat op 3
101    display.print(temperatuur); // Schrijf de huidige temperatuur naar het scherm
102    display.print(" C"); // Schrijf tekst naar het scherm
103    display.display(); // Toon de tekst op het scherm
104
105    client.stop(); // Stop de verbinding met de server
106 }
107
108
109
110

```

4. Vul net als in de voorgaande oefeningen jouw netwerknaam en wachtwoord in.
5. Voer op regel 23 jouw API-key en locatie in.
6. Zorg ervoor dat de [microcontroller verbonden is](#) (blz. 12) met de computer, dat je [het juiste bord](#) (blz. 11) en [de juiste poort](#) (blz. 11) geselecteerd hebt.
7. Klik op de **Uploaden** knop  om het programma te uploaden naar de microcontroller.

Wanneer alles goed is gegaan verschijnt na enige tijd de huidige temperatuur op jouw locatie. Iedere tien minuten wordt de waarde ververs.

Krijg je niet de temperatuur te zien? Probeer dan eens een andere plaatsnaam, wellicht wordt jouw plaats niet ondersteund.

Gaat er nog iets mis? Bekijk dan de tips onder [Problemen oplossen](#) (blz. 57).

# 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](#) (blz. 11) en [het aansluiten van de microcontroller](#) (blz. 12) 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 (zie [afbeelding 0.1](#) (blz. 3)) 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. 3)).

## **De buzzer piept niet**

- Let goed op of je de buzzer wel in de juiste oriëntatie in het breadboard hebt geprikt. Het korte pootje is de minpool en moet verbonden zijn met de GND, het lange pootje is de pluspool en moet verbonden zijn met een digitale pin op de microcontroller.

# Gefeliciteerd!

Je hebt nu geleerd hoe je informatie van het internet op de display kunt tonen. Uit de gebruikte API's kun je natuurlijk ook nog andere gegevens tonen dan wij nu hebben gedaan. Misschien lukt het je wel om het volgende gegeven te tonen wanneer je op het knopje drukt.

Op het internet zijn er ook nog heel veel meer API's te vinden dus met alles wat je hebt geleerd in deze kit zijn de mogelijkheden oneindig!

We zijn benieuwd wat jij allemaal gaat tonen op jouw display en vinden het heel leuk als je dit met ons deelt via de email of social media kanalen.

## Inspiratie

Voor nog meer inspiratie en projectideeën kun je ons volgen:



[instagram.com/awesome.makes](https://www.instagram.com/awesome.makes)



[twitter.com/awesome\\_makes](https://twitter.com/awesome_makes)



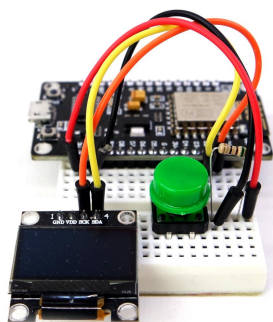
[facebook.com/AwesomeMakes](https://www.facebook.com/AwesomeMakes)



[linkedin.com/company/awesome-makes](https://www.linkedin.com/company/awesome-makes)

## Meer bouwen?

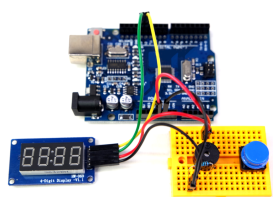
Heb je de smaak te pakken en wil je meer bouwen? Bekijk dan eens de andere Awesome Crates op [www.awesomemakes.com](http://www.awesomemakes.com)



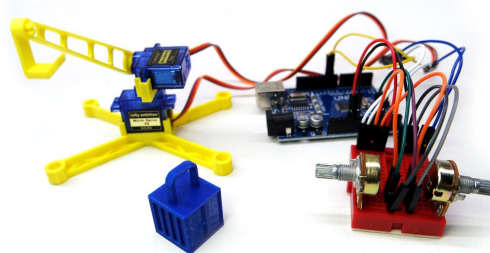
*IOT Smart Display*



*IOT Mood Light*



*Countdown Timer*



*Bestuurbare Hijskraan*