



OTRONIC ARDUINO UNO-STARTERKIT

Arduino UNO-starterkit: tutorial en 16 projecten

Stap voor stap: wat is een Arduino, wat zit er in je kit, hoe werkt Codey en 16 projecten om zelf te bouwen.

codey.online/nl/starter-kits/uno/tutorial



Inhoud

Welkom! Jij gaat het maken! 🚀	3
Wat is een Arduino UNO?	4
Wat zit er in je starterkit?	5
Breadboard	6
Jumperkabels	7
LED (lampje)	8
Weerstand 220 Ω	9
Drukknop	10
Actieve zoemer	11
Lichtsensoren (LDR)	12
Temperatuursensor DHT11	13
Afstandssensor HC-SR04	14
Bewegingssensor (PIR)	15
OLED-scherm 0,91"	16
Servomotor SG90	17
RGB LED-ring (8 LEDs)	18
Relais 5V	19
9V-batterijclip	20
Wat is vibe coding?	21
Maak je gratis account	22
Zo werkt Codey	23
Aansluiten en uploaden	24
Aan de slag! 🎉	25
Je eerste breadboard-schakeling	26
Knipperlicht	27
Verkeerslicht	28
Lichtknop	29
Deurbel	30
Regenboogring	31
Mini-weerstation	32
Parkeersensor	33
Nachtlampje	34
Bewegingsalarm	35
Slagboom	36
Reactiespel	37
Elektronische dobbelsteen	38
Parkeersensor lux	39
Afstandsmeter met lichtbalk	40
Slimme ventilator	41
Automatische slagboom	42

OTRONIC ARDUINO UNO-STARTERKIT

Welkom! Jij gaat het maken! 🚀

Met deze starterkit bouw je je eigen slimme apparaatjes. Een weerstation, een parkeersensor, een alarm... En het mooiste: je hoeft niet te kunnen programmeren. Codey helpt je.

Neem deze korte tutorial even door. Dan weet je wat er in je kit zit en wat je ermee kunt. Daarna ga je meteen aan de slag met je eerste project.

Zo gaat het:

- 2 **Wat is een Arduino en wat kun je ermee?**
- 3 **Alle onderdelen uit je kit, één voor één**
- 4 **Wat is vibe coding? Programmeren met AI**
- 5 **Je gratis account maken**
- 6 **Zo werkt Codey**
- 7 **Je Arduino aansluiten en je code uploaden**
- 8 **Aan de slag met 16 projecten**



🕒 Dit duurt ongeveer 15 minuten. Je kunt altijd overslaan en later terugkomen.

Wat is een Arduino UNO?

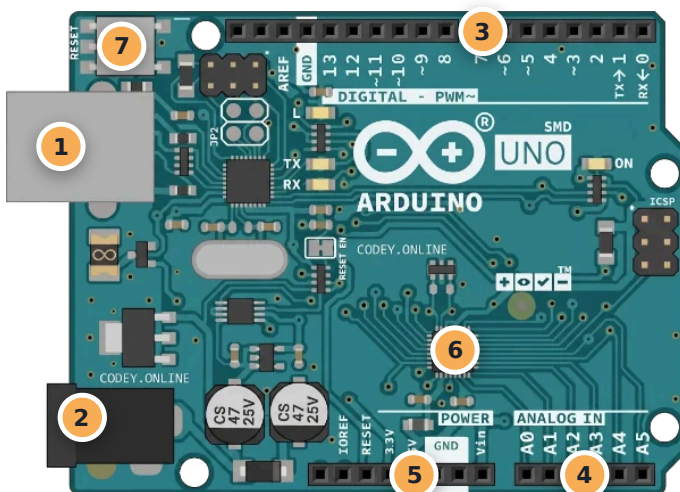
De Arduino is een klein computertje. Je vertelt hem wat hij moet doen, en dat doet hij. Steeds opnieuw, zonder moe te worden.

Zo denkt een Arduino



Voorbeeld: de afstandssensor ziet dat je dichtbij komt (input). De Arduino denkt: "te dichtbij!" En de zoemer gaat piepen (output). Zo werkt een parkeersensor in een auto.

Wat zit er op het bord?



- 1 USB-poort: hier komt de kabel naar je computer. Daar krijgt hij stroom én code door.
- 2 Stroomaansluiting: voor een batterij of adapter (7-12V), als je geen computer gebruikt.
- 3 Digitale pinnen D0 t/m D13: aan of uit. Voor lampjes, knoppen en zoemers.
- 4 Analoge pinnen A0 t/m A5: meten hoeveel. Bijvoorbeeld hoeveel licht er is.
- 5 Stroompinnen: 5V (plus) en GND (min) geven stroom aan je onderdelen.
- 6 De chip: het brein. Hier komt jouw code in te staan.
- 7 Resetknop: start je code opnieuw.

Twee woorden die je vaak gaat zien:

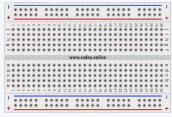
- **5V** is de plus. Hier komt de stroom vandaan (5 volt, net als een telefoonlader).
- **GND** (ground) is de min. Hier gaat de stroom weer terug. Elk onderdeel heeft een draad naar GND nodig.

Pinnen zijn de gaatjes aan de zijkant. Daar steek je draden in. Een pin kan een **ingang** zijn (hij luistert) of een **uitgang** (hij geeft stroom). Dat bepaal je in de code.

Wat zit er in je starterkit?

Je kit zit vol met onderdelen. Sommige voelen iets (sensoren), andere doen iets (lampjes, zoemer, motortje). Klik op Volgende, dan leer je ze één voor één kennen.

Basis: hiermee bouw je alles



Breadboard

Het bouwboard: bouwen zonder solderen

In je kit: 1x



Jumperkabels

Dupont-kabels: de draadjes die alles verbinden

In je kit: 40x



Weerstand 220 Ω

Remt de stroom af

In je kit: 10x

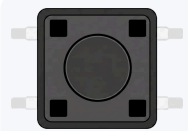


9V-batterijclip

Je project zonder computer laten werken

In je kit: 1x

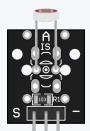
Input: deze onderdelen voelen of meten iets



Drukknop

Een knopje om op te drukken

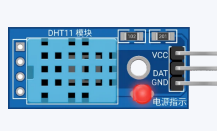
In je kit: 5x



Lichtsensormodule (LDR)

Meet hoe licht of donker het is

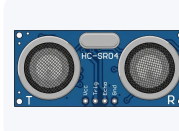
In je kit: 1x



Temperatuursensormodule DHT11

Meet de temperatuur en de luchtvochtigheid

In je kit: 1x



Afstandssensormodule HC-SR04

Meet afstand met geluid, net als een vleermuis

In je kit: 1x



Bewegingssensormodule (PIR)

Ziet of er iemand beweegt

In je kit: 1x

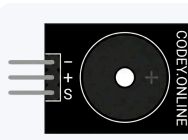
Output: deze onderdelen doen iets



LED (lampje)

Een klein lampje in 5 kleuren

In je kit: 10x



Actieve zoemer

Maakt piepjes

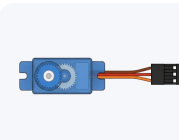
In je kit: 1x



OLED-schermje 0,91"

Een klein scherm voor tekst en getallen

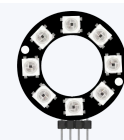
In je kit: 1x



Servomotor SG90

Een motortje dat naar een stand draait

In je kit: 1x



RGB LED-ring (8 LEDs)

8 slimme lampjes in elke kleur

In je kit: 1x

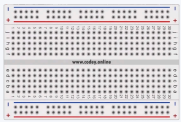


Relais 5V

Een schakelaar die je Arduino kan omzetten

In je kit: 1x

Onderdeel 1 van 15



Breadboard

Het bouwboard: bouwen zonder solderen

In je kit: 1x

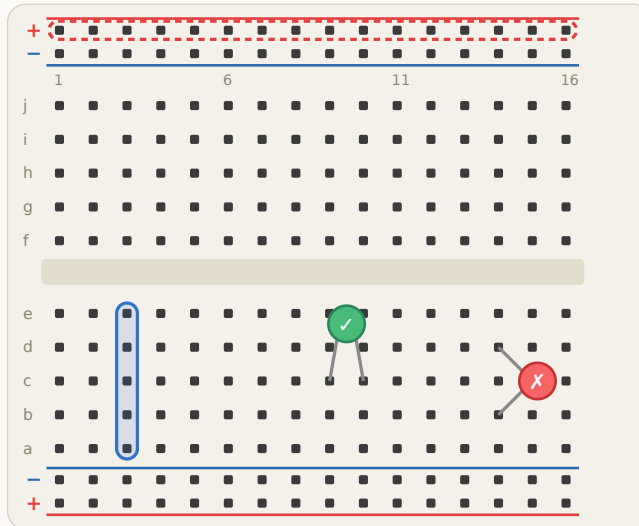
Wat is het?

Een breadboard is een plankje vol gaatjes. Je steekt er onderdelen en draden in. Onder de gaatjes zitten metalen strips die sommige gaatjes met elkaar verbinden. Zo maak je een schakeling zonder te solderen. Fout gedaan? Gewoon eruit trekken en opnieuw.

Zo sluit je het aan

Zo zit het in elkaar:

- **Rijtjes van 5:** in het middelste deel zijn steeds 5 gaatjes naast elkaar verbonden (a-e en f-j). Alles wat je in hetzelfde rijtje steekt, is met elkaar verbonden.
- **De gleuf in het midden** scheidt de twee kanten. Links (a-e) en rechts (f-j) zijn dus *niet* verbonden.
- **Stroomstrips (+ en -)** aan de lange zijken zijn over de hele lengte verbonden. Verbind de + met 5V en de - met GND van je Arduino. Nu heb je overal stroom.



- Stroomstrip: de hele lengte is verbonden
- 5 gaatjes in een rijtje zijn verbonden
- X Fout: beide pootjes in hetzelfde rijtje

- Gleuf: links en rechts zijn niet verbonden
- ✓ Goed: de pootjes zitten in verschillende rijtjes

Wat kun je ermee?

- Schakelingen uitproberen zonder solderen
- Meerdere onderdelen tegelijk stroom geven via de + en - strips
- Een LED met een weerstand netjes in serie zetten

Tip

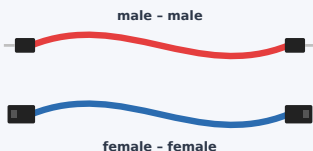
Zet de twee pootjes van een onderdeel nooit in hetzelfde rijtje van 5. Dan zijn ze met elkaar verbonden en gaat de stroom er gewoon langs. Zet een onderdeel liever over de gleuf heen.

🔧 **Probeer het meteen uit!** In de oefening "Je eerste breadboard-schakeling" bouw je een LED met een weerstand op het breadboard en laat je hem branden en knippen. →

Projecten met dit onderdeel

★ Je eerste breadboard-schakeling

Onderdeel 2 van 15



Jumperkabels

Dupont-kabels: de draadjes die alles verbinden

In je kit: 40x

Wat is het?

Met jumperkabels verbind je je Arduino met de onderdelen. Ze heten ook wel **Dupont-kabels**, naar de zwarte stekkertjes aan de uiteinden. Er zijn twee soorten in je kit:

- **Male-male** (pinnetje aan beide kanten): om in de Arduino en het breadboard te steken.
- **Female-female** (gaatje aan beide kanten): om op de pinnetjes van een module te schuiven.

Zo sluit je het aan

Steek de kabel stevig in het gaatje of over het pinnetje. Combineer je een male- en een female-kabel, dan heb je een male-female-kabel: handig om een module direct op je Arduino aan te sluiten.

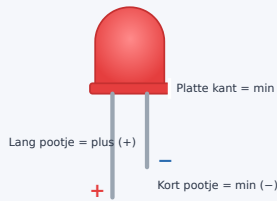
Wat kun je ermee?

- Je Arduino verbinden met het breadboard
- Sensormodules aansluiten
- Stroom (5V) en GND naar je onderdelen brengen

Tip

Gebruik vaste kleuren: rood voor 5V en zwart voor GND. Dan zie je meteen of de stroom goed zit.

Onderdeel 3 van 15



LED (lampje)

Een klein lampje in 5 kleuren

In je kit: 10x

Wat is het?

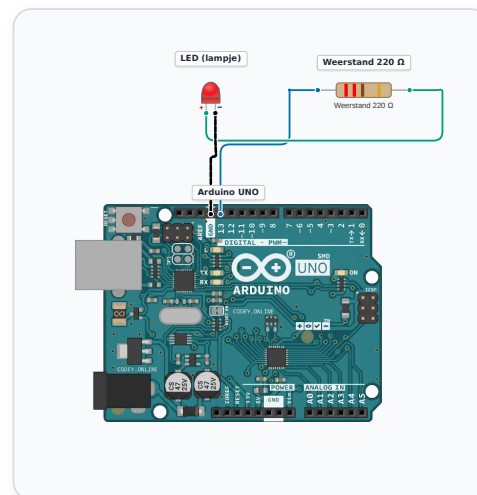
Een LED is een klein lampje dat heel weinig stroom gebruikt. Je kit heeft 10 LEDs in 5 kleuren. Een LED werkt maar één kant op, dus je moet hem goed om zetten.

Zo sluit je het aan

- Het **lange pootje** is de plus (+). Die gaat via een weerstand van 220 Ω naar een pin van je Arduino.
- Het **korte pootje** is de min (-). Die gaat naar GND. Aan die kant is de rand van het lampje ook plat.

Gebruik altijd een weerstand! Zonder weerstand krijgt de LED te veel stroom en gaat hij kapot.

Aansluitschema



- Arduino UNO · D13 → Weerstand 220 Ω · 1
- Weerstand 220 Ω · 2 → LED (lampje) · ANODE
- LED (lampje) · CATHODE → Arduino UNO · GND

Wat kun je ermee?

- Een knipperlicht
- Een verkeerslicht
- Een lampje dat laat zien of iets aan of uit staat

Tip

Doet je LED niets? Draai hem dan om. Hij werkt maar één kant op.

Projecten met dit onderdeel

★ Je eerste breadboard-schakeling

★ Knipperlicht

★ Verkeerslicht

★ Lichtknop

★★ Bewegingsalarm

★★ Reactiespel

Onderdeel 4 van 15

Rood, rood, bruin = 220 Ω

Weerstand 220 Ω

Remt de stroom af

In je kit: 10x

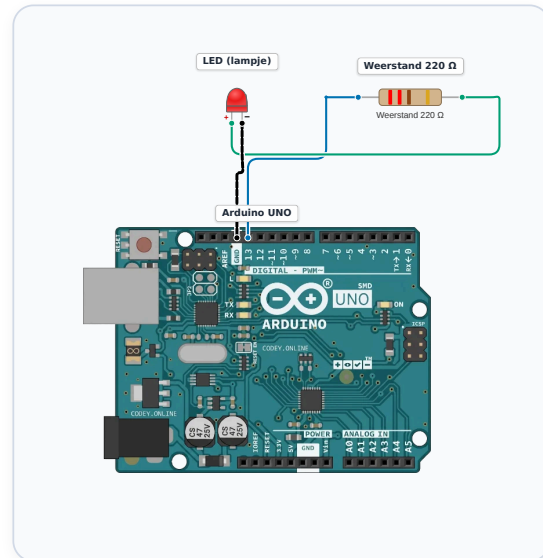
Wat is het?

Een weerstand remt de stroom een beetje af. Net als een kraan die je half dichtdraait. Zo krijgt een LED precies genoeg stroom en gaat hij niet kapot. De weerstanden in je kit zijn 220 ohm (Ω). Dat zie je aan de gekleurde ringetjes: rood, rood, bruin.

Zo sluit je het aan

Een weerstand heeft geen plus of min: je mag hem allebei de kanten op zetten. Zet hem tussen de pin van de Arduino en het lange pootje van de LED. De ene kant gaat naar de pin, de andere kant naar de LED.

Aansluitschema



- Arduino UNO · D13 → Weerstand 220 Ω · 1
- Weerstand 220 Ω · 2 → LED (lampje) · ANODE
- LED (lampje) · CATHODE → Arduino UNO · GND

Wat kun je ermee?

- Elke LED beschermen
- Stroom naar een onderdeel beperken

Tip

Eén weerstand per LED. Gebruik je drie LEDs? Dan heb je ook drie weerstanden nodig.

Projecten met dit onderdeel

★ Je eerste breadboard-schakeling

★ Knipperlicht

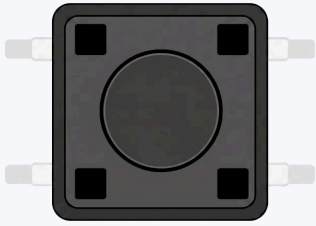
★ Verkeerslicht

★ Lichtknop

★★ Bewegingsalarm

★★ Reactiespel

Onderdeel 5 van 15



Drukknop

Een knopje om op te drukken

In je kit: 5x

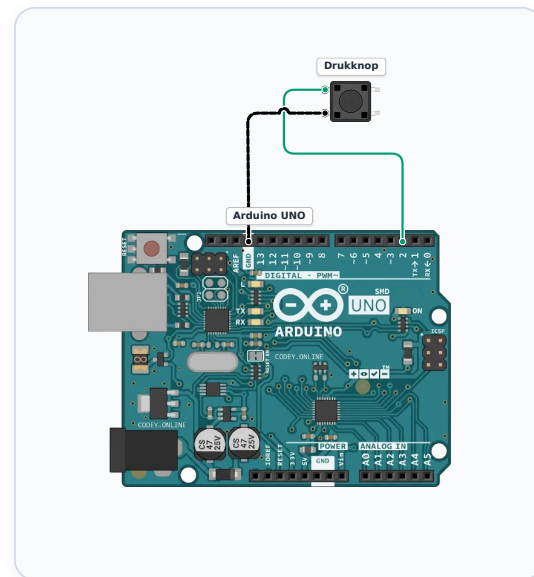
Wat is het?

Een drukknop maakt contact zolang je erop drukt. Laat je los, dan is het contact weer weg. Je Arduino kan voelen of de knop is ingedrukt. Je kit heeft 5 knoppen.

Zo sluit je het aan

De knop heeft 4 pootjes. Je hebt er maar 2 nodig: één naar een pin (bijvoorbeeld D2) en één naar GND. In de code zet je de pin op `INPUT_PULLUP`. Dan is de pin HIGH als je niet drukt en LOW als je wel drukt.

Aansluitschema



● Arduino UNO · D2 → Drukknop · 1

● Arduino UNO · GND → Drukknop · 2

Wat kun je ermee?

- Een lampje aan- en uitzetten
- Een deurbel
- Een spelletje: wie drukt het snelst?

Tip

Niet zeker welke pootjes? Neem twee pootjes die schuin tegenover elkaar zitten. Dan zit je altijd goed. Zet de knop over de gleuf van je breadboard.

Projecten met dit onderdeel

★ Lichtknop

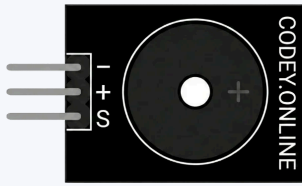
★ Deurbel

★★ Slagboom

★★ Reactiespel

★★ Elektronische dobbelsteen

Onderdeel 6 van 15



Actieve zoemer

Maakt piepjes

In je kit: 1x

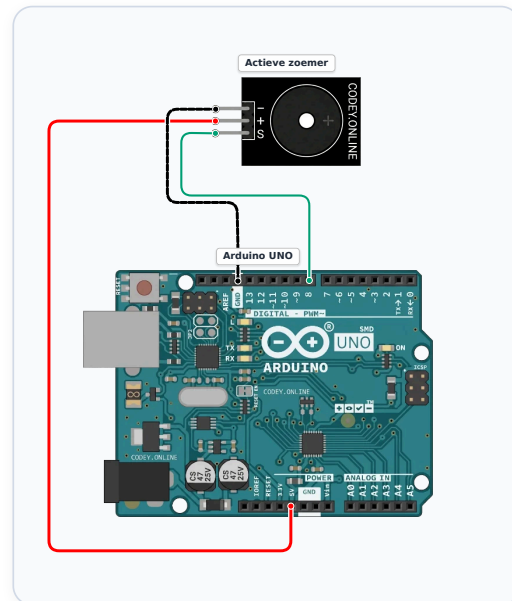
Wat is het?

De zoemer maakt een piepgeluid. Het is een **actieve** zoemer: hij piept vanzelf zodra hij stroom krijgt. Je hoeft hem alleen aan en uit te zetten, net als een lampje.

Zo sluit je het aan

- S (signaal) naar een pin, bijvoorbeeld D8
- + naar 5V
- - naar GND

Aansluitschema



● Arduino UNO · GND → Actieve zoemer · -

● Arduino UNO · 5V → Actieve zoemer · +

● Arduino UNO · D8 → Actieve zoemer · S

Wat kun je ermee?

- Een alarm
- Een parkeersensor die sneller piept als je dichterbij komt
- Een deurbel of een timer

Tip

Wil je een korte piep? Zet de zoemer aan, wacht 100 milliseconden en zet hem weer uit.

Projecten met dit onderdeel

★ Deurbel

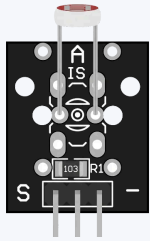
★★ Parkeersensor

★★ Bewegingsalarm

★★★★ Parkeersensor luxe

★★★★ Automatische slagboom

Onderdeel 7 van 15



Lichtsensorm (LDR)

Meet hoe licht of donker het is

In je kit: 1x

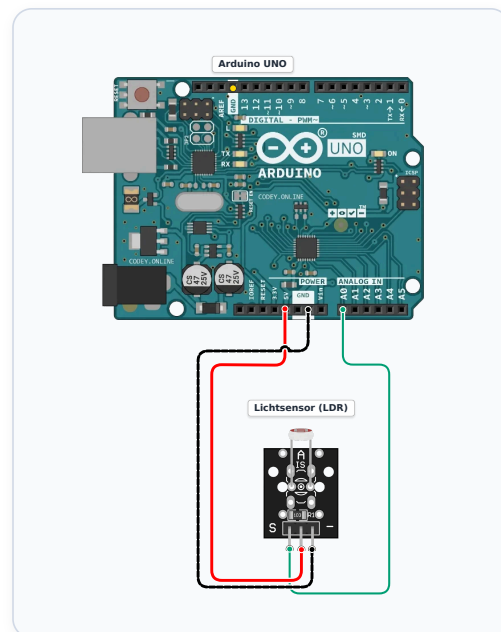
Wat is het?

Deze sensor meet hoeveel licht er is. Je Arduino leest een getal tussen 0 en 1023. Bij veel licht krijg je een ander getal dan in het donker. Zo weet je Arduino of het dag of nacht is.

Zo sluit je het aan

- S naar een analoge pin, bijvoorbeeld A0
- + naar 5V
- - naar GND

Aansluitschema



- Arduino UNO · A0 → Lichtsensor (LDR) · S
- Arduino UNO · 5V → Lichtsensor (LDR) · +
- Arduino UNO · GND → Lichtsensor (LDR) · -

Wat kun je ermee?

- Een nachtlampje dat vanzelf aangaat
- Een lichtmeter
- Een alarm dat afgaat als iemand de lade opendoet

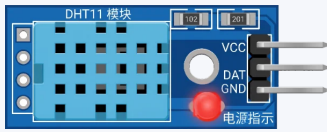
Tip

Kijk in de seriële monitor welk getal je ziet. Hou dan je hand boven de sensor. Zo zie je welk getal bij "donker" hoort.

Projecten met dit onderdeel

★★ Nachtlampje

Onderdeel 8 van 15



Temperatuursensor DHT11

Meet de temperatuur en de luchtvochtigheid

In je kit: 1x

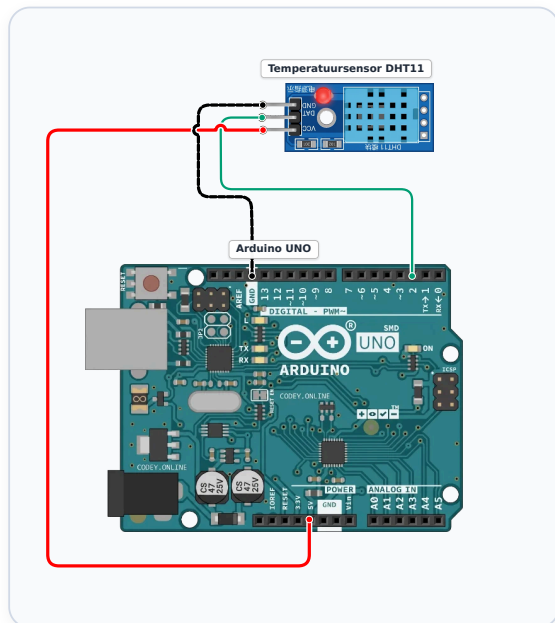
Wat is het?

De DHT11 meet hoe warm het is (in graden Celsius) en hoe vochtig de lucht is (in procent). Hij geeft om de 2 seconden een nieuwe meting.

Zo sluit je het aan

- **DAT** (data) naar een pin, bijvoorbeeld D2
- **VCC** naar 5V
- **GND** naar GND

Aansluitschema



- Arduino UNO · 5V → Temperatuursensor DHT11 · VCC
- Arduino UNO · GND → Temperatuursensor DHT11 · GND
- Arduino UNO · D2 → Temperatuursensor DHT11 · DAT

Wat kun je ermee?

- Een mini-weerstation
- Een ventilator die aangaat als het te warm is
- Een waarschuwing als het te vochtig is in de badkamer

Tip

Adem eens op de sensor. Dan zie je de luchtvochtigheid meteen stijgen!

Projecten met dit onderdeel

☆☆ Mini-weerstation

☆☆☆ Slimme ventilator

Onderdeel 9 van 15



Afstandssensor HC-SR04

Meet afstand met geluid, net als een vleermuis

In je kit: 1x

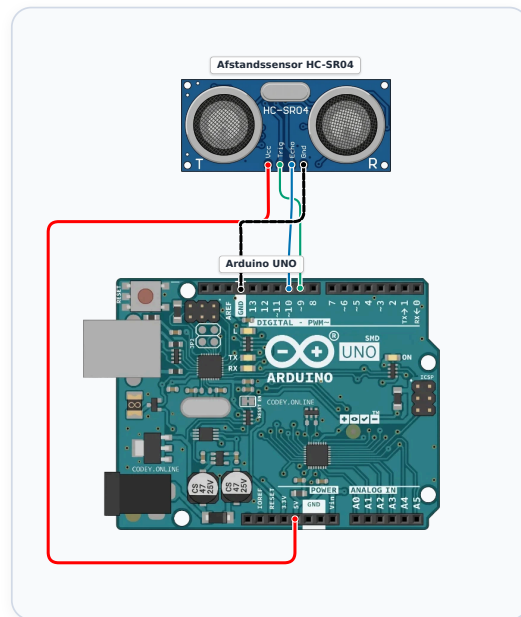
Wat is het?

Deze sensor stuurt een geluidje dat jij niet kunt horen. Het geluid botst tegen iets aan en komt terug. De sensor meet hoe lang dat duurt. Zo weet je Arduino hoe ver iets weg is: van 2 cm tot ongeveer 4 meter.

Zo sluit je het aan

- **Trig** (zenden) naar D9
- **Echo** (ontvangen) naar D10
- **Vcc** naar 5V
- **Gnd** naar GND

Aansluitschema



- Arduino UNO · 5V → Afstandssensor HC-SR04 · Vcc
- Arduino UNO · GND → Afstandssensor HC-SR04 · Gnd
- Arduino UNO · D9 → Afstandssensor HC-SR04 · Trig
- Arduino UNO · D10 → Afstandssensor HC-SR04 · Echo

Wat kun je ermee?

- Een parkeersensor
- Een afstandsmeter
- Een slagboom die vanzelf opengaat

Tip

De twee "ogen" zijn de luidspreker en de microfoon. Richt ze recht op wat je wilt meten.

Projecten met dit onderdeel

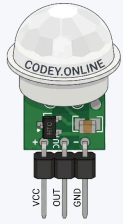
★★★ Parkeersensor

★★★★ Parkeersensor luxe

★★★★ Afstandsmeter met lichtbalk

★★★★ Automatische slagboom

Onderdeel 10 van 15



Bewegingssensor (PIR)

Ziet of er iemand beweegt

In je kit: 1x

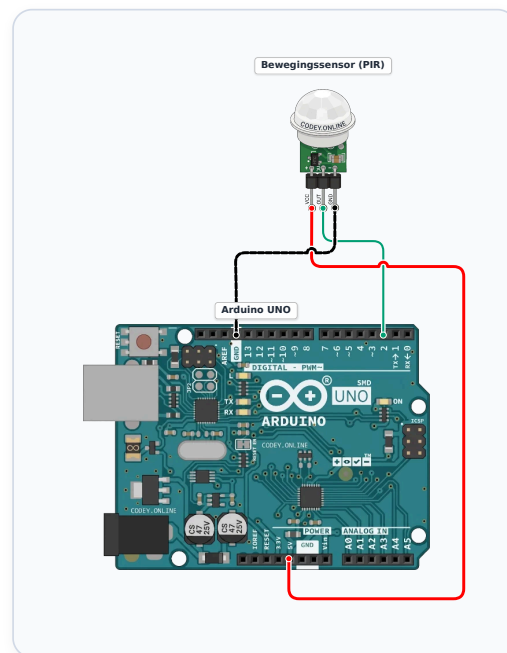
Wat is het?

Een PIR-sensor voelt de warmte van mensen en dieren. Beweegt er iets warmes voor de sensor? Dan wordt zijn uitgang HIGH. Zo werken ook de lampen die vanzelf aangaan als je langsloopt.

Zo sluit je het aan

- **OUT** naar een pin, bijvoorbeeld D2
- **VCC** naar 5V
- **GND** naar GND

Aansluitschema



● Arduino UNO · 5V → Bewegingssensor (PIR) · VCC

● Arduino UNO · GND → Bewegingssensor (PIR) · GND

● Arduino UNO · D2 → Bewegingssensor (PIR) · OUT

Wat kun je ermee?

- Een inbraakalarm
- Een lamp die aangaat als je binnenkomt
- Tellen hoe vaak iemand langsloopt

Tip

De sensor heeft na het opstarten een paar seconden nodig om te wennen aan de kamer. Sta dan even stil.

Projecten met dit onderdeel

★★ Bewegingsalarm

Onderdeel 11 van 15



OLED-schermpje 0,91"

Een klein scherm voor tekst en getallen

In je kit: 1x

Wat is het?

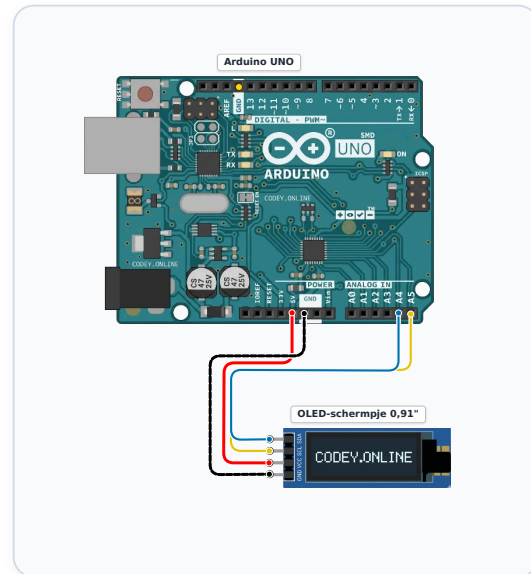
Een klein, scherp schermpje van 128 bij 32 pixels. Je kunt er tekst, getallen en simpele tekeningen op zetten. Bijvoorbeeld de temperatuur of een afstand.

Zo sluit je het aan

Het scherm praat via **I2C**: maar twee datadraden.

- **SDA** naar A4
- **SCL** naar A5
- **VCC** naar 5V
- **GND** naar GND

Aansluitschema



- Arduino UNO · A4 → OLED-schermpje 0,91" · SDA
- Arduino UNO · A5 → OLED-schermpje 0,91" · SCL
- Arduino UNO · 5V → OLED-schermpje 0,91" · VCC
- Arduino UNO · GND → OLED-schermpje 0,91" · GND

Wat kun je ermee?

- De temperatuur laten zien
- De afstand in centimeters tonen
- Een score of reactietijd tonen

Tip

Blijft het scherm zwart? Controleer of SDA en SCL niet omgedraaid zijn.

Projecten met dit onderdeel

★★★ Mini-weerstation

★★★ Reactiespel

★★★★ Parkeersensor luxe

★★★★ Slimme ventilator

Onderdeel 12 van 15



Servomotor SG90

Een motortje dat naar een stand draait

In je kit: 1x

Wat is het?

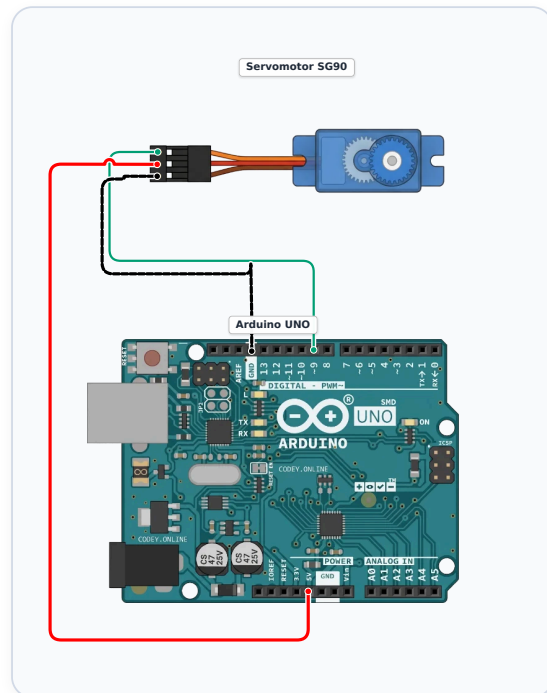
Een servo is een motortje dat je naar een precieze stand kunt draaien, van 0 tot 180 graden. Je zegt bijvoorbeeld "ga naar 90 graden" en hij gaat precies daarheen. Op het asje zet je een armpje.

Zo sluit je het aan

De servo heeft drie draadjes:

- **Oranje of geel** (signaal) naar een pin, bijvoorbeeld D9
- **Rood** naar 5V
- **Bruin** naar GND

Aansluitschema



● Arduino UNO · GND → Servomotor SG90 · GND

● Arduino UNO · 5V → Servomotor SG90 · VCC

● Arduino UNO · D9 → Servomotor SG90 · PWM

✨ Wat kun je ermee?

- Een slagboom
- Een wijzer die de temperatuur aanwijst
- Een zwaaiend robotarmpje

Tip

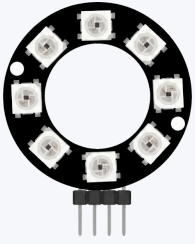
Draai het armpje niet met de hand. Dan kunnen de tandwieltjes binnenin kapotgaan.

Projecten met dit onderdeel

★★★ Slagboom

★★★★ Automatische slagboom

Onderdeel 13 van 15



RGB LED-ring (8 LEDs)

8 slimme lampjes in elke kleur

In je kit: 1x

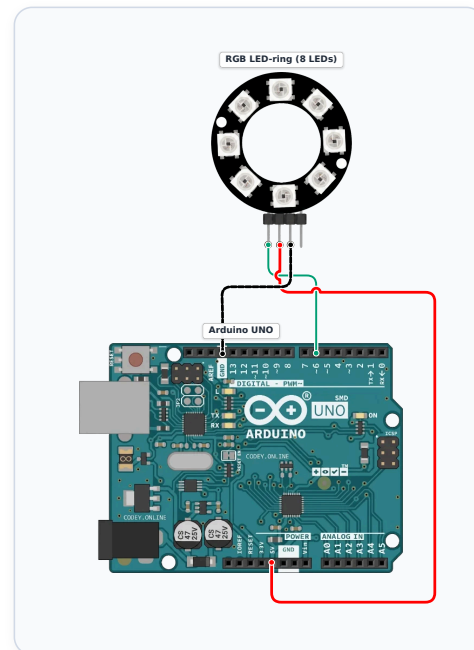
Wat is het?

Een ring met 8 slimme lampjes. Elk lampje kan elke kleur aannemen, en je stuurt ze allemaal apart aan. Met maar één datadraad! Superleuk voor lichteffecten.

Zo sluit je het aan

- **DI** (data in) naar een pin, bijvoorbeeld D6
- **5V** naar 5V
- **GND** naar GND
- **DO** laat je los (daar kun je nog een ring achter zetten)

Aansluitschema



- Arduino UNO · 5V → RGB LED-ring (8 LEDs) · 5V
- Arduino UNO · GND → RGB LED-ring (8 LEDs) · GND
- Arduino UNO · D6 → RGB LED-ring (8 LEDs) · DI

Wat kun je ermee?

- Een regenbooglamp
- Een lichtbalk die laat zien hoe dichtbij iets is
- Een dobbelsteen van licht

Tip

Zet de helderheid in de code niet te hoog. Dat is fijner voor je ogen en je Arduino.

Projecten met dit onderdeel

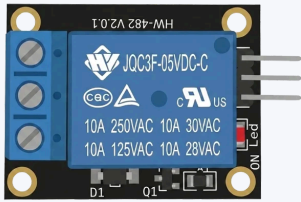
★ Regenboogring

★★ Nachtlampje

★★ Elektronische dobbelsteen

★★★★ Afstandsmeter met lichtbalk

Onderdeel 14 van 15



Relais 5V

Een schakelaar die je Arduino kan omzetten

In je kit: 1x

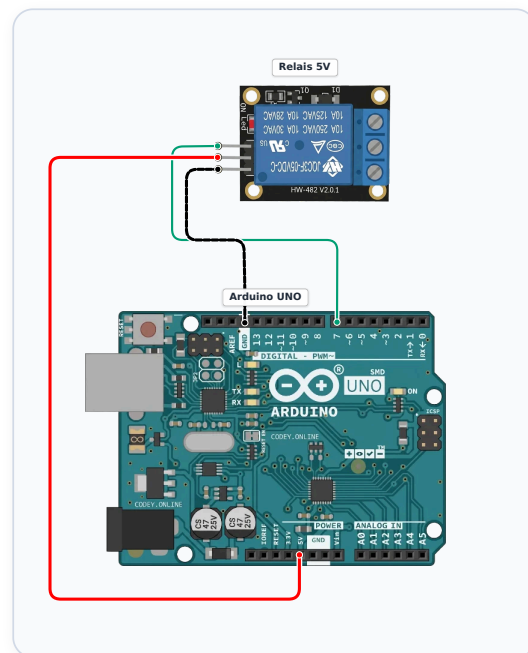
Wat is het?

Een relais is een schakelaar die je Arduino kan aan- en uitzetten. Je hoort hem "klikken". Met een relais kan je Arduino een ander apparaat aan- en uitzetten, bijvoorbeeld een ventilatortje of een lamp op een batterij.

Zo sluit je het aan

- **S** naar een pin, bijvoorbeeld D7
- **+** naar 5V
- **-** naar GND
- Het apparaat dat je wilt schakelen, sluit je aan op **COM** en **NO**. NO betekent: normaal uit, aan als het relais klikt.

Aansluitschema



● Arduino UNO · GND → Relais 5V · -

● Arduino UNO · 5V → Relais 5V · +

● Arduino UNO · D7 → Relais 5V · S

Wat kun je ermee?

- Een ventilator die aangaat als het warm is
- Een lamp op een timer
- Een pompje voor de planten

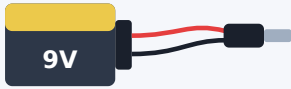
Tip

Veiligheid voorop: gebruik het relais alleen met batterijen of lage spanning (tot 12V). Nooit met het stopcontact (230V)!

Projecten met dit onderdeel

★★★★ Slimme ventilator

Onderdeel 15 van 15



9V-batterijclip

Je project zonder computer laten werken

In je kit: 1x

Wat is het?

Met deze clip zet je een 9V-batterij op je Arduino. Dan werkt je project ook zonder computer. Handig als je het wilt laten zien of ergens neerzetten.

Zo sluit je het aan

Klik de batterij op de clip en steek de ronde stekker in de zwarte stroomaansluiting van je Arduino. Je code staat nog op de Arduino, dus hij begint meteen.

Wat kun je ermee?

- Je parkeersensor in de garage hangen
- Je weerstation op de vensterbank zetten
- Je project aan vrienden laten zien

Tip

Haal de batterij eraf als je klaar bent. Dan gaat hij veel langer mee.

Wat is vibe coding?

Vroeger moest je eerst maanden leren programmeren. Nu vertel je gewoon in je eigen woorden wat je wilt maken. De AI schrijft de code voor je.

Dat heet **vibe coding**: jij hebt het idee, de AI doet het typwerk. Je zegt bijvoorbeeld: *"Maak een nachtlampje dat aangaat als het donker wordt."* Binnen een paar seconden staat de code klaar, met een aansluitschema erbij.

De AI van nu is heel goed in programmeren. Vaak zelfs beter en sneller dan een mens. Hij kent alle onderdelen, maakt minder typfouten en legt alles rustig uit.

Een project bouwen: vroeger en nu

Zelf programmeren

Uitzoeken, typen, fouten zoeken, opnieuw proberen...

± 10 uur

Met Codey

Vertellen wat je wilt, uploaden, klaar.

± 5 minuten ↗

Zo werkt het



Jij vertelt wat je wilt maken.



Codey schrijft de code en tekent hoe je alles aansluit.



Jij bouwt het na en uploadt de code.



Wil je iets anders? Zeg het, dan past Codey het aan.

Je leert er ook nog van! Codey legt uit wat de code doet. Zo snap je steeds beter hoe het werkt.

Maak je gratis account

Om met Codey te werken heb je een account nodig. Dat is gratis en duurt maar een minuutje. Met je starterkit gebruik je Codey daarna onbeperkt.

1

Registreer

Klik op "Account maken" en vul je e-mailadres en een wachtwoord in. Heb je een Google-account? Klik dan op "Verder met Google", dan ben je meteen klaar.

2

Bevestig je e-mailadres

Je krijgt een mail van Codey Online. Klik op de link in die mail. Zie je geen mail? Kijk dan ook even bij je ongewenste mail.

3

Log in

Log in met je e-mailadres en wachtwoord. Je komt dan vanzelf terug in deze tutorial, bij de volgende stap.

Ik heb al een account: inloggen →

👉 Ga naar codey.online/uno om je gratis account te maken.



Ben je nog jong? Maak het account dan samen met je vader, moeder of een andere volwassene.

Zo werkt Codey

Codey is jouw programmeermaatje. Alles zit in één scherm. Dit zijn de belangrijkste knoppen.

The screenshot shows the Codey Online web interface. On the left is a sidebar with a chat window, project files, and navigation links. The main area is divided into a code editor (labeled 2) showing C++ code for an Arduino Uno, and a right-hand panel (labeled 1) containing instructions, a wiring diagram (labeled 4), and control buttons like 'Agent', 'Plan', 'Vraag', 'Seriële monitor', and 'Bedradingsschema'. At the bottom, there's a 'Mijn kit' section (labeled 6) and a 'BERICHT VERZENDEN' button. A 'MILESTONE' button (labeled 3) is also visible in the code editor area.

1 De chat

Hier praat je met Codey. Typ wat je wilt maken of veranderen, in gewone taal.

2 De code

Hier staat de code die Codey voor je schrijft. Je mag er zelf ook in typen.

3 Compileren & uploaden naar board

Eén knop voor alles: Codey controleert je code, maakt hem klaar (compileren) en stuurt hem via de USB-kabel naar je Arduino (uploaden). Is er een fout? Dan lost Codey die vaak zelf op.

4 Aansluitschema

Laat zien welke draad waar moet. Je ziet het in de chat; klik erop om het groot te zien.

5 Seriële monitor

Hier zie je de berichten die je Arduino stuurt, bijvoorbeeld de temperatuur of een afstand.

6 Mijn kit

Alle onderdelen uit je starterkit. Codey gebruikt alleen dingen die jij hebt.

Goed om te weten: met je starterkit gebruik je Codey onbeperkt in je kit-chats. Je kunt Codey alles vragen, ook "wat doet deze regel?" of "waarom werkt het niet?".

Aansluiten en uploaden

Je code staat klaar? Dan gaat hij nu naar je Arduino. Dat doe je in drie stappen.

Belangrijk: gebruik **Google Chrome** of **Microsoft Edge** op je computer. Alleen die browsers mogen met je Arduino praten via USB. Safari, Firefox en de meeste telefoons kunnen dat (nog) niet.

1 Sluit je Arduino aan

Steek de USB-kabel in je Arduino en in je computer. Er gaat een groen lampje (ON) branden.

2 Klik op Compileren & uploaden naar board

Codey controleert je code en maakt hem klaar (compileren). De eerste keer duurt dat soms een minuutje.

3 Kies je Arduino in het poortvenster

Er komt een venster met poorten. Kies je Arduino en klik op Verbinden. De lampjes RX en TX knipperen even: je code gaat erin (uploaden). Klaar!

Hoe heet mijn Arduino in het poortvenster?

Dat hangt af van je computer. Zie je er meer? Haal de kabel er even uit en kijk welke verdwijnt. Dat is je Arduino.

Windows

Een **COM-poort** met een nummer, bijvoorbeeld `Arduino Uno (COM4)` of `USB-SERIAL CH340 (COM5)`. COM1 is bijna nooit je Arduino.

Mac

Iets met `usbmodem` of `usbserial`, bijvoorbeeld `/dev/cu.usbmodem14101`, `cu.usbserial-1420` of `cu.wchusbserial1420`. Kies altijd de `cu.`-versie.

Linux

`/dev/ttyACM0` of `/dev/ttyUSB0`. Geen toegang? Voeg jezelf toe aan de groep `dialout` en log opnieuw in.

Chromebook

Chrome laat de naam van het apparaat zien, bijvoorbeeld `Arduino Uno`, `USB2.0-Serial` of `USB Serial`.

Seriële monitor: sommige projecten sturen berichten naar je computer, bijvoorbeeld de temperatuur. Die zie je als je in Codey op Seriële monitor klikt. Zo zie je wat je Arduino "denkt".

Lukt het niet? Geen paniek!

▼ Ik zie mijn Arduino niet in het poortvenster.

Probeer een andere USB-kabel. Sommige kabels kunnen alleen opladen en geen data sturen. Probeer ook een andere USB-poort op je computer.

▼ Ik zie CH340 of USB2.0-Serial, maar uploaden lukt niet (Windows/Mac).

Je Arduino gebruikt dan een CH340-chip. Installeer eenmalig de gratis CH340-driver (zoek op "CH340 driver") en start je computer opnieuw op.

▼ Uploaden mislukt halverwege.

Sluit andere programma's die de Arduino gebruiken, zoals de Arduino IDE of een tweede Codey-tabblad. Trek de kabel er even uit, steek hem terug en probeer opnieuw.

▼ Het werkt, maar mijn schakeling doet niets.

Controleer de draden met het aansluitschema. Zit 5V en GND goed? Vraag het daarna aan Codey: "Mijn LED gaat niet aan, wat kan het zijn?"

Aan de slag! 🎉

Kies een project en klik op **MAAK DIT**. Codey zet de code en het aansluitschema voor je klaar. Begin bij een project met één ster ★ als je nog nooit iets met Arduino hebt gedaan.



★ ⌚ 10 minuten

Je eerste breadboard-schakeling

Oefen met het breadboard: laat een LED branden en daarna knipperen.



★ ⌚ 10 minuten

Knipperlicht

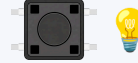
Je allereerste project: een LED laten knipperen.



★ ⌚ 15 minuten

Verkeerslicht

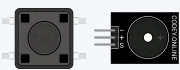
Drie LEDs die werken als een echt stoplicht.



★ ⌚ 15 minuten

Lichtknop

Druk op de knop en de lamp gaat aan. Nog een keer: uit.



★ ⌚ 10 minuten

Deurbel

Druk op de knop voor een vrolijk ding-dong.



★ ⌚ 10 minuten

Regenboogring

De LED-ring draait in alle kleuren van de regenboog.



★★ ⌚ 20 minuten

Mini-weerstation

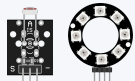
Temperatuur en luchtvochtigheid op je eigen schermje.



★★ ⌚ 20 minuten

Parkeersensor

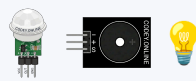
Hoe dichterbij, hoe sneller het piept. Net als in een auto.



★★ ⌚ 15 minuten

Nachtlampje

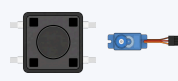
Wordt het donker? Dan gaat de lamp vanzelf aan.



★★ ⌚ 20 minuten

Bewegingsalarm

Loopt er iemand langs? Dan gaat het alarm af!



★★ ⌚ 20 minuten

Slagboom

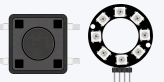
Druk op de knop en de slagboom gaat open.



★★ ⌚ 25 minuten

Reactiespel

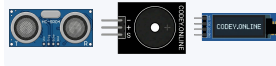
Hoe snel ben jij? Meet je reactietijd.



★★ ⌚ 15 minuten

Elektronische dobbelsteen

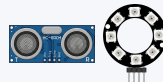
Druk op de knop en gooi een dobbelsteen van licht.



★★★ ⌚ 30 minuten

Parkeersensor luxe

Piepen én de afstand in centimeters op een schermje.



★★★★ ⌚ 25 minuten

Afstandsmeter met lichtbalk

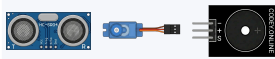
Hoe dichterbij, hoe meer lampjes: groen, oranje, rood.



★★★★ ⌚ 30 minuten

Slimme ventilator

Te warm? Dan zet het relais de ventilator aan.



★★★★ ⌚ 30 minuten

Automatische slagboom

Komt er een auto aan? Dan gaat de slagboom vanzelf open.

Heb je zelf een idee? Nog beter! Open Codey en vertel wat je wilt maken. Bijvoorbeeld: "Maak een lampje dat sneller knippert als het warmer wordt."

★ Beginner ⌚ 10 minuten

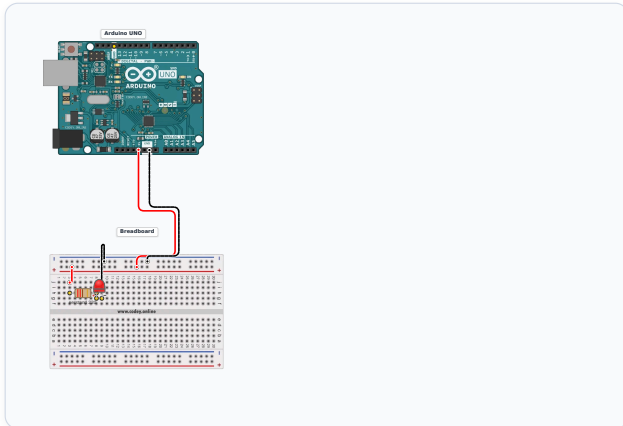
Je eerste breadboard-schakeling

Een oefening om te ervaren hoe een breadboard werkt. Je bouwt een LED met een weerstand op het breadboard. Eerst laat je hem branden met de 5V-strip, zonder code. Daarna laat je hem knipperen met pin 13.

Aansluitschema

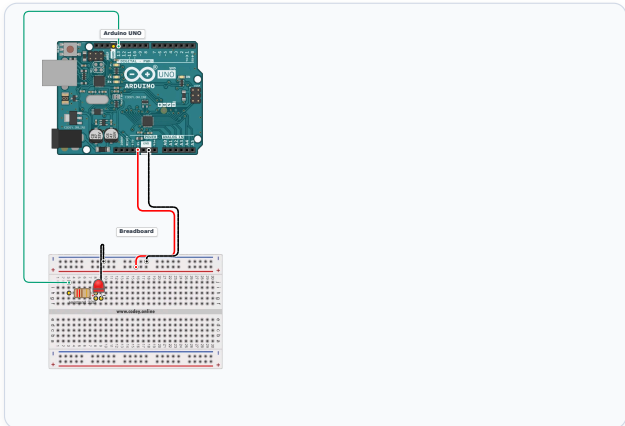
Stap A: laat de LED branden (zonder code)

Het rode draadje gaat van de +-strip naar rijtje 3. De stroom loopt dan: 5V → +-strip → weerstand → LED → --strip → GND. Brandt de LED? Dan heb je het breadboard goed gebruikt!



Stap B: laat de LED knipperen met pin 13

Haal het rode draadje tussen de +-strip en rijtje 3 weg. Zet er een draad voor in de plaats van pin 13 naar rijtje 3. Nu bepaalt de Arduino wanneer de LED stroom krijgt.



Wat heb je nodig?



Zo bouw je het

1. Verbind 5V van de Arduino met de +-strip bovenaan het breadboard (rode draad).
2. Verbind GND van de Arduino met de --strip ernaast (zwarte draad). Nu heb je twee stroomstrips.
3. Zet de weerstand van 220 Ω van gaatje H3 naar gaatje H8. Hij ligt dan in twee verschillende rijtjes.
4. Zet het lange pootje van de LED in G8 (zelfde rijtje als de weerstand) en het korte pootje in G9.
5. Verbind gaatje J9 met de --strip (zwart draadje). Zo kan de stroom terug naar GND.
6. Stap A: verbind de +-strip met gaatje J3 (rood draadje). De LED gaat branden!
7. Stap B: haal dat rode draadje weg en verbind pin 13 met gaatje J3 (groene draad).
8. Laat Codey de code maken: klik op MAAK DIT, of typ zelf in de chat: "Laat de LED op pin 13 elke seconde knipperen." Klik daarna op Compileren & uploaden naar board.

De code

```
// Je eerste breadboard-schakeling
// De LED op het breadboard gaat aan en uit. Hij krijgt stroom via pin 13.

const int LED_PIN = 13; // De groene draad gaat van pin 13 naar het breadboard

void setup() {
  pinMode(LED_PIN, OUTPUT); // Pin 13 is een uitgang: hij geeft stroom
}

void loop() {
  digitalWrite(LED_PIN, HIGH); // LED aan
  delay(1000); // 1 seconde wachten
  digitalWrite(LED_PIN, LOW); // LED uit
  delay(1000); // 1 seconde wachten
}
```

Wat doet de code?

- Pin 13 is een uitgang. Met **HIGH** geeft hij 5V, met **LOW** geeft hij niets.
- In **loop()** zetten we de LED aan, wachten we 1 seconde, zetten we hem uit en wachten we weer.
- De stroom loopt via het breadboard: pin 13 → rijtje 3 → weerstand → rijtje 8 → LED → rijtje 9 → --strip → GND.

Probeer eens...

- Laat de LED twee keer zo snel knipperen
- Voeg een tweede LED toe op het breadboard die om de beurt knippert
- Laat de LED drie keer kort knipperen en dan even uit blijven

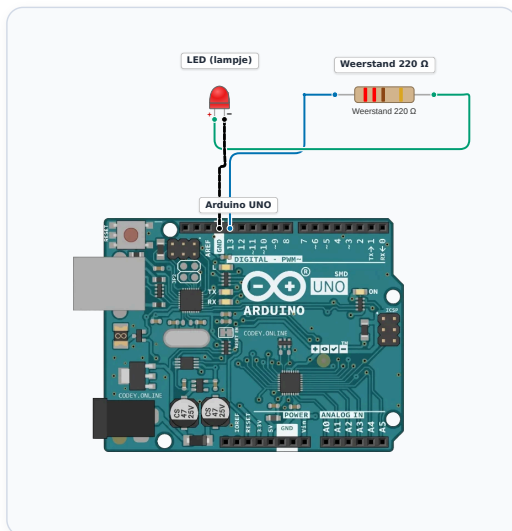
🔧 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/breadboard-led en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.

★ Beginner ⌚ 10 minuten

Knipperlicht

Je eerste project! Een LED gaat steeds aan en uit, net als een knipperlicht. Zo leer je hoe de Arduino een uitgang aan- en uitzet.

Aansluitschema



Wat heb je nodig?



1x Arduino UNO



1x LED (lampje)

1x Weerstand 220 Ω USB-kabel +
Jumperkabels

Zo bouw je het

1. Zet de LED in het breadboard, met de pootjes in twee verschillende rijtjes.
2. Zet een weerstand van 220 Ω vanaf het rijtje van het lange pootje naar een ander rijtje.
3. Verbind dat andere rijtje met een draad met pin 13 van je Arduino.
4. Verbind het korte pootje met een draad met GND.

De code

```
// Knipperlicht
// Een LED gaat steeds aan en uit, net als een knipperlicht.

const int LED_PIN = 13; // De LED zit op pin 13 (via de weerstand van 220 ohm)
const int WACHTTIJD = 500; // Hoe lang de LED aan of uit blijft (in milliseconden)

void setup() {
  // setup() draait één keer als de Arduino start
  pinMode(LED_PIN, OUTPUT); // Pin 13 is een uitgang: hij geeft stroom
}

void loop() {
  // loop() draait steeds opnieuw, voor altijd
  digitalWrite(LED_PIN, HIGH); // LED aan
  delay(WACHTTIJD); // Even wachten
  digitalWrite(LED_PIN, LOW); // LED uit
  delay(WACHTTIJD); // Even wachten
}
```

Wat doet de code?

- In `setup()` zeggen we dat pin 13 een uitgang is.
- In `loop()` zetten we de LED aan, wachten we even, zetten we hem uit en wachten we weer.
- `loop()` begint daarna gewoon opnieuw. Zo blijft de LED knipperen.

Probeer eens...

- Laat de LED twee keer zo snel knipperen
- Maak er een SOS-signaal van: kort-kort-kort, lang-lang-lang, kort-kort-kort
- Laat de LED rustig fel en zacht worden, alsof hij ademt



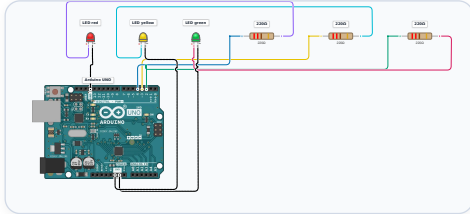
Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/blink en klik op **MAAK DIT**. Dan staan de code en het aansluitschema meteen klaar.

★ Beginner ⌚ 15 minuten

Verkeerslicht

Je bouwt een echt verkeerslicht met een rode, een gele en een groene LED. Ze gaan om de beurt aan, net als op straat.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Zet de drie LEDs naast elkaar in het breadboard.
2. Geef elke LED een eigen weerstand van 220 Ω naar zijn pin: rood naar 4, geel naar 3, groen naar 2.
3. Verbind alle korte pootjes met de min-strip (–) van het breadboard.
4. Verbind de min-strip met GND van je Arduino.

De code

```
// Verkeerslicht
// Drie LEDs doen precies wat een echt stoplicht doet: groen, oranje, rood.

const int ROOD = 4;    // Rode LED op pin 4
const int ORANJE = 3;  // Gele/oranje LED op pin 3
const int GROEN = 2;   // Groene LED op pin 2

void setup() {
  pinMode(ROOD, OUTPUT);
  pinMode(ORANJE, OUTPUT);
  pinMode(GROEN, OUTPUT);
}

// Zet één lampje aan en de andere twee uit
void zetLicht(int aan) {
  digitalWrite(ROOD, aan == ROOD ? HIGH : LOW);
  digitalWrite(ORANJE, aan == ORANJE ? HIGH : LOW);
  digitalWrite(GROEN, aan == GROEN ? HIGH : LOW);
}

void loop() {
  zetLicht(GROEN); // Groen: je mag rijden
  delay(5000);
  zetLicht(ORANJE); // Oranje: pas op, het wordt rood
  delay(2000);
  zetLicht(ROOD); // Rood: stoppen!
  delay(5000);
}
```

Wat doet de code?

- Elke LED heeft een eigen pin: rood op 4, geel op 3 en groen op 2.
- De functie `zetLicht()` zet één lampje aan en de andere twee uit.
- In `loop()` gaan we van groen naar oranje naar rood, met een wachttijd ertussen.

Probeer eens...

- Laat het oranje lampje drie keer knipperen voordat het rood wordt
- Voeg een knop toe: als je drukt, springt het licht sneller op rood (een voetgangersknop)
- Maak er een nachtstand van waarin alleen het oranje lampje knippert

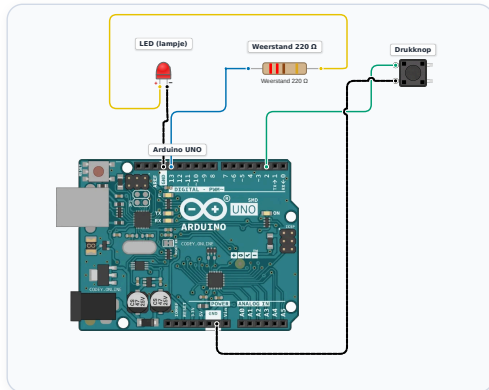
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/traffic-light en klik op **MAAK DIT**. Dan staan de code en het aansluitschema meteen klaar.

★ Beginner ⌚ 15 minuten

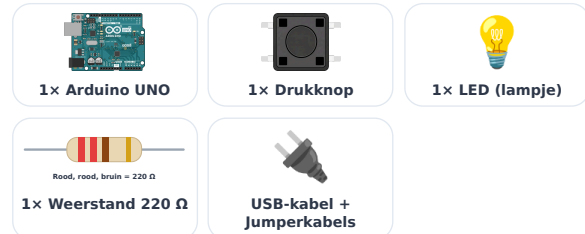
Lichtknop

Je maakt een lichtknop. Eén keer drukken: de LED gaat aan. Nog een keer drukken: de LED gaat uit. Zo leer je hoe de Arduino een knop "voelt".

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Zet de drukknoop over de gleuf van je breadboard.
2. Verbind één pootje met pin 2 en een pootje aan de andere kant met GND.
3. Zet de LED met een weerstand van 220 Ω op pin 13, het korte pootje naar GND.

De code

```
// Lichtknop
// Druk één keer op de knop: de LED gaat aan. Nog een keer: de LED gaat uit.

const int KNOP_PIN = 2; // De drukknoop zit op pin 2 (andere kant naar GND)
const int LED_PIN = 13; // De LED zit op pin 13 (via de weerstand)

bool ledAan = false; // Onthoudt of de LED nu aan of uit is
bool vorigeKnop = HIGH; // Hoe de knop de vorige keer was

void setup() {
  pinMode(KNOP_PIN, INPUT_PULLUP); // Ingang met ingebouwde weerstand: los = HIGH, ingedrukt = LOW
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  bool knop = digitalRead(KNOP_PIN);

  // Alleen reageren op het moment dat je de knop indrukt
  if (vorigeKnop == HIGH && knop == LOW) {
    ledAan = !ledAan; // Omdraaien: aan wordt uit, uit wordt aan
    digitalWrite(LED_PIN, ledAan ? HIGH : LOW);
    delay(50); // Kort wachten tegen "stuiteren" van de knop
  }

  vorigeKnop = knop;
}
```

Wat doet de code?

- De knop zit op pin 2 met `INPUT_PULLUP`. Los is de pin HIGH, ingedrukt LOW.
- We onthouden of de LED aan of uit is in `ledAan`.
- Alleen op het moment dat je de knop indrukt, draaien we `ledAan` om.

Probeer eens...

- Laat de LED knipperen in plaats van aan blijven
- Laat de LED na 10 seconden vanzelf weer uitgaan
- Tel hoe vaak er op de knop is gedrukt en laat het zien in de seriële monitor

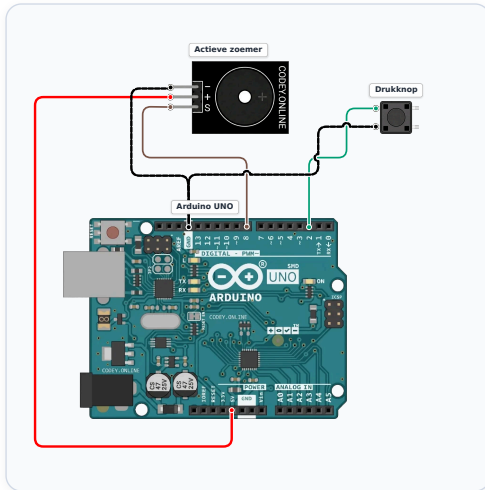
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/light-switch en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.

★ Beginner ⌚ 10 minuten

Deurbel

Je maakt je eigen deurbel. Druk op de knop en de zoemer zegt "ding-dong"!

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Zet de drukknop over de gleuf van het breadboard: één pootje naar pin 2, een pootje aan de andere kant naar GND.
2. Sluit de zoemer aan: S naar pin 8, + naar 5V en – naar GND.

De code

```
// Deurbel
// Druk op de knop en de zoemer laat een vrolijk "ding-dong" horen.

const int KNOP_PIN = 2; // Drukknop op pin 2 (andere kant naar GND)
const int ZOEMER_PIN = 8; // Actieve zoemer: S op pin 8

void setup() {
  pinMode(KNOP_PIN, INPUT_PULLUP); // Los = HIGH, ingedrukt = LOW
  pinMode(ZOEMER_PIN, OUTPUT);
}

// Laat de zoemer een tijdje piepen
void piep(int hoeLang) {
  digitalWrite(ZOEMER_PIN, HIGH);
  delay(hoeLang);
  digitalWrite(ZOEMER_PIN, LOW);
}

void loop() {
  if (digitalRead(KNOP_PIN) == LOW) { // Iemand drukt op de bel!
    piep(300); // "Ding"
    delay(150);
    piep(600); // "Dong"
    delay(1000); // Even wachten, zodat de bel niet blijft rinkelen
  }
}
```

Wat doet de code?

- De knop zit op pin 2 en de zoemer op pin 8.
- De functie `piep()` zet de zoemer even aan en dan weer uit.
- Als de knop is ingedrukt, laten we een korte en een lange piep horen.

Probeer eens...

- Speel een vrolijk melodietje van vijf piepjes
- Laat er ook een LED knipperen als iemand aanbelt
- Tel hoe vaak er is aangebeld en toon het in de seriële monitor

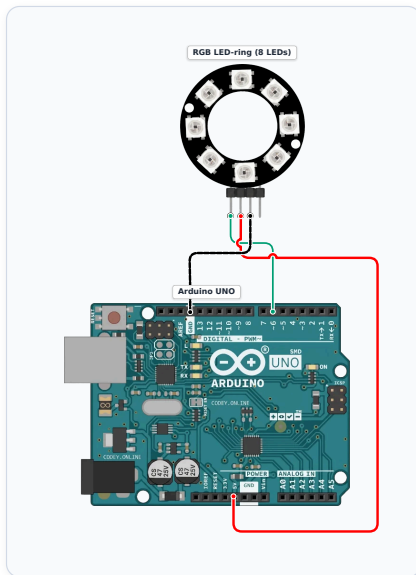
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/doorbell en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.

★ Beginner ⌚ 10 minuten

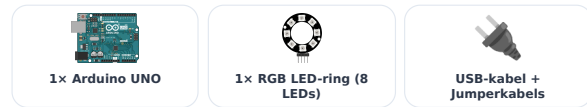
Regenboogring

Met de LED-ring maak je een draaiende regenboog. Elk lampje krijgt een andere kleur, en alle kleuren schuiven rond.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Sluit de LED-ring aan: DI naar pin 6, 5V naar 5V en GND naar GND.
2. Dat is alles! DO laat je los.

De code

```
// Regenboogring
// De LED-ring laat alle kleuren van de regenboog rondjes draaien.

#include <Adafruit_NeoPixel>

const int RING_PIN = 6; // DI van de ring zit op pin 6
const int AANTAL_LEDS = 8; // De ring heeft 8 lampjes
const int SNELHEID = 20; // Hoe kleiner, hoe sneller het draait

Adafruit_NeoPixel ring(AANTAL_LEDS, RING_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  ring.begin();
  ring.setBrightness(40); // Niet te fel: 0 = uit, 255 = superfel
  ring.show();
}

void loop() {
  // Schuif de regenboog steeds een stukje op
  for (long kleur = 0; kleur < 65536; kleur += 512) {
    for (int i = 0; i < AANTAL_LEDS; i++) {
      // Elk lampje krijgt een iets andere kleur
      long lampKleur = kleur + (i * 65536 / AANTAL_LEDS);
      ring.setPixelColor(i, ring.gamma32(ring.ColorHSV(lampKleur)));
    }
    ring.show();
    delay(SNELHEID);
  }
}
```

Wat doet de code?

- We gebruiken de bibliotheek `Adafruit_NeoPixel` om de ring aan te sturen.
- `ColorHSV()` maakt een kleur uit de regenboog. Elk lampje krijgt een stukje verder in de regenboog.
- Door de kleur steeds een stukje op te schuiven, lijkt de regenboog rond te draaien.

Probeer eens...

- Laat de regenboog twee keer zo snel draaien
- Maak er een politielicht van dat afwisselend rood en blauw knippert
- Laat één lampje rondjes draaien, als een lichtje dat rondrent

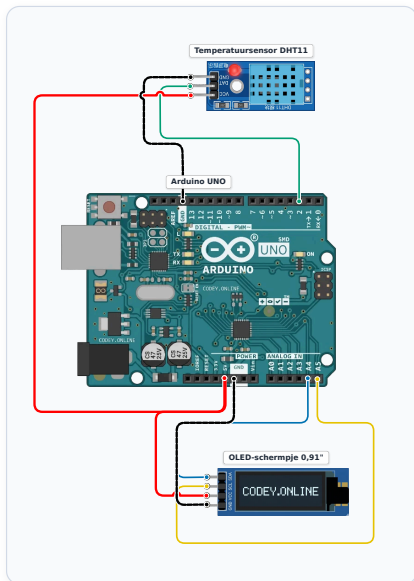
🔧 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/rainbow-ring en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.

★★ **Gevorderde beginner** ⌚ 20 minuten

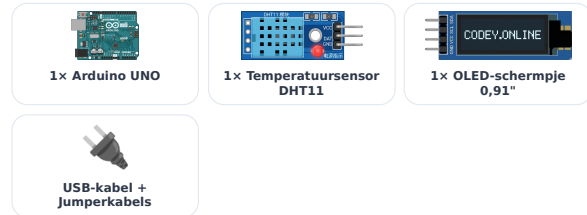
Mini-weerstation

Je maakt een mini-weerstation. De DHT11 meet de temperatuur en de luchtvochtigheid, en het OLED-schermje laat ze zien.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Sluit de DHT11 aan: DAT naar pin 2, VCC naar 5V en GND naar GND.
2. Sluit het OLED-scherm aan: SDA naar A4, SCL naar A5, VCC naar 5V en GND naar GND.
3. Tip: twee onderdelen hebben 5V en GND nodig. Gebruik de + en – strips van je breadboard.

De code

```
// Mini-weerstation
// De DHT11 meet de temperatuur en de luchtvochtigheid. Het OLED-schermje laat
// het zien.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

const int DHT_PIN = 2; // DAT van de DHT11 zit op pin 2
DHT dht(DHT_PIN, DHT11);

Adafruit_SSD1306 scherm(128, 32, &Wire, -1); // OLED van 128 x 32 pixels (SDA =
A4, SCL = A5)

void setup() {
  Serial.begin(9600);
  dht.begin();
  if (!scherm.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // 0x3C is het adres van het
  scherm
    Serial.println(F("Scherm niet gevonden. Kijk de draden van SDA en SCL
na."));
    while (true); // Stoppen: zonder scherm
  }
  scherm.setTextColor(SSD1306_WHITE);
}

void loop() {
  float temperatuur = dht.readTemperature(); // In graden Celsius
  float vocht = dht.readHumidity(); // In procent

  scherm.clearDisplay();
  scherm.setTextSize(2);
  scherm.setCursor(0, 0);

  if (isnan(temperatuur) || isnan(vocht)) { // Lukt het meten niet?
    scherm.setTextSize(1);
    scherm.print(F("Geen DHT11 gevonden"));
  } else {
    scherm.print(temperatuur, 1);
    scherm.print((char)247); // Het graden-teken
    scherm.println(F("C"));
    scherm.print(vocht, 0);
    scherm.print(F("% vocht"));

    Serial.print(F("Temperatuur: "));
    Serial.print(temperatuur);
    Serial.print(F(" C Vocht: "));
    Serial.print(vocht);
    Serial.println(F(" %"));
  }

  scherm.display(); // Pas nu komt alles echt op het scherm
  delay(2000); // De DHT11 kan maar eens per 2 seconden meten
}
```

Wat doet de code?

- De DHT11 zit op pin 2. Met `readTemperature()` en `readHumidity()` lezen we de waarden.
- Het OLED-scherm praat via SDA (A4) en SCL (A5). Met `print()` zetten we tekst op het scherm.
- Elke 2 seconden komt er een nieuwe meting. Lukt het meten niet? Dan zie je een melding.

Probeer eens...

- Laat ook de temperatuur in Fahrenheit zien
- Laat een LED branden als het warmer is dan 25 graden
- Onthoud de hoogste temperatuur van vandaag en laat die ook zien

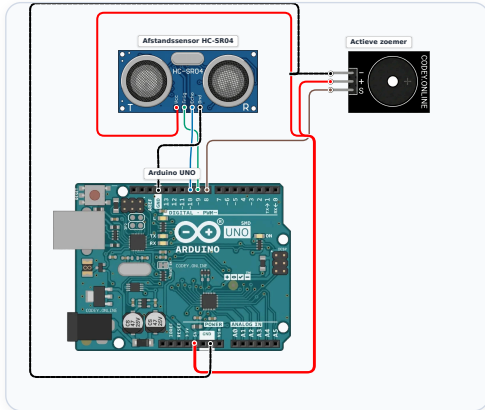
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/weather-station en klik op **MAAK DIT**. Dan staan de code en het aansluitschema meteen klaar.

★★★★★ **Gevorderde beginner** ⌚ **20 minuten**

Parkeersensor

Je bouwt een parkeersensor, net als in een auto. Hoe dichterbij de sensor komt, hoe sneller de zoemer piept. Heel dichtbij? Dan klinkt er één lange toon.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Sluit de HC-SR04 aan: Trig naar pin 9, Echo naar pin 10, Vcc naar 5V en Gnd naar GND.
2. Sluit de zoemer aan: S naar pin 8, + naar 5V en – naar GND.
3. Gebruik de + en – strips van je breadboard voor 5V en GND.

De code

```
// Parkeersensor
// Hoe dichterbij de sensor komt, hoe sneller de zoemer piept.
// Net als in een auto!

const int TRIG_PIN = 9; // Trig van de HC-SR04 op pin 9
const int ECHO_PIN = 10; // Echo van de HC-SR04 op pin 10
const int ZOEMER_PIN = 8; // S van de zoemer op pin 8

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(ZOEMER_PIN, OUTPUT);
}

// Meet de afstand in centimeters
long meetAfstand() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Stuur een kort geluidje (dat
  // jij niet hoort)
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long tijd = pulseIn(ECHO_PIN, HIGH, 30000); // Hoe lang duurt
  // het voor de echo terugkomt?
  if (tijd == 0) return 999; // Geen echo: niets

  in de buurt
  return tijd / 58; // Tijd omrekenen
  // naar centimeters
}

void loop() {
  long afstand = meetAfstand();
  Serial.print(F("Afstand: "));
  Serial.print(afstand);
  Serial.println(F(" cm"));

  if (afstand < 5) {
    digitalWrite(ZOEMER_PIN, HIGH); // Heel dichtbij: één lange
    // toon
    delay(100);
  } else if (afstand < 50) {
    digitalWrite(ZOEMER_PIN, HIGH); // Korte piep
    delay(50);
    digitalWrite(ZOEMER_PIN, LOW);
    delay(afstand * 10); // Dichterbij = kortere pauze
    // = sneller piepen
  } else {
    digitalWrite(ZOEMER_PIN, LOW); // Ver weg: stil
    delay(100);
  }
}
```

Wat doet de code?

- `meetAfstand()` stuurt een geluidje en meet hoe lang het duurt voor de echo terugkomt. Dat rekenen we om naar centimeters.
- Verder dan 50 cm: de zoemer is stil.
- Dichterbij: de pauze tussen de piepjes wordt korter. Onder de 5 cm piept hij aan één stuk door.

Probeer eens...

- Laat de parkeersensor al vanaf 1 meter beginnen met piepen
- Voeg een rode LED toe die gaat branden als het te dichtbij is
- Laat de afstand zien in de seriële monitor, met een balkje van sterretjes

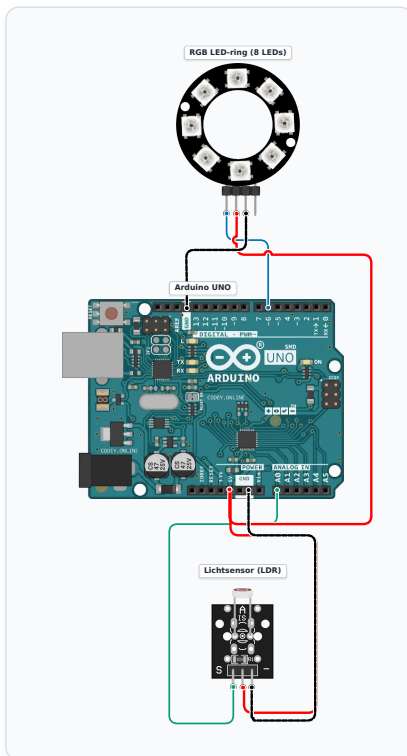
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/parking-sensor en klik op **MAAK DIT**. Dan staan de code en het aansluitschema meteen klaar.

★★ **Gevorderde beginner** ⌚ 15 minuten

Nachtlampje

Je maakt een slim nachtlampje. Wordt het donker, dan gaat de LED-ring zacht branden. Wordt het licht, dan gaat hij weer uit.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Sluit de lichtsensor aan: S naar A0, + naar 5V en – naar GND.
2. Sluit de LED-ring aan: DI naar pin 6, 5V naar 5V en GND naar GND.
3. Open de seriële monitor en kijk welk getal je ziet bij licht en bij donker. Pas `DONKER_GRENS` daarop aan.

De code

```
// Nachtlampje
// Wordt het donker? Dan gaat de LED-ring zacht branden. Wordt het licht? Dan
gaat hij uit.

#include <Adafruit_NeoPixel.h>

const int LICHT_PIN = A0; // S van de lichtsensor op A0
const int RING_PIN = 6; // DI van de LED-ring op pin 6
const int AANTAL_LEDS = 8;

// Bij deze waarde vinden we het "donker". Kijk in de seriële monitor welk getal
jij ziet
// bij licht en bij donker, en zet hier een getal tussenin.
const int DONKER_GRENS = 500;

Adafruit_NeoPixel ring(AANTAL_LEDS, RING_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  Serial.begin(9600);
  ring.begin();
  ring.setBrightness(60);
  ring.show();
}

void loop() {
  int licht = analogRead(LICHT_PIN); // Een getal van 0 tot 1023
  Serial.print(F("Lichtsensor: "));
  Serial.println(licht);

  // Bij deze sensor wordt het getal hoger als het donkerder wordt.
  // Werkt het bij jou net andersom? Draai dan > om naar <.
  bool donker = licht > DONKER_GRENS;

  for (int i = 0; i < AANTAL_LEDS; i++) {
    if (donker) {
      ring.setPixelColor(i, ring.Color(255, 140, 40)); // Warm geel licht
    } else {
      ring.setPixelColor(i, 0); // Uit
    }
  }
  ring.show();
  delay(200);
}
```

Wat doet de code?

- De lichtsensor zit op A0. `analogRead()` geeft een getal van 0 tot 1023.
- Is het getal hoger dan `DONKER_GRENS`? Dan vinden we het donker.
- In het donker geven we alle lampjes een warme gele kleur. Anders gaan ze uit.

Probeer eens...

- Laat het lampje langzaam feller worden naarmate het donkerder wordt
- Laat de kleur langzaam veranderen, als een rustgevende sfeerlamp
- Laat het lampje na 1 minuut vanzelf weer uitgaan

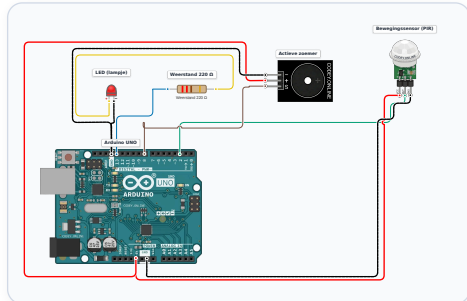
🔧 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/night-light en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.

★★ **Gevorderde beginner** ⌚ **20 minuten**

Bewegingsalarm

Je bouwt een inbraakalarm. Ziet de bewegingssensor iemand? Dan gaat de zoemer af en knippert de rode LED.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Sluit de PIR-sensor aan: OUT naar pin 2, VCC naar 5V en GND naar GND.
2. Sluit de zoemer aan: S naar pin 8, + naar 5V en – naar GND.
3. Zet de rode LED met een weerstand van 220 Ω op pin 13, het korte puntje naar GND.

De code

```
// Bewegingsalarm
// Loopt er iemand langs? Dan knippert de rode LED en gaat de zoemer af.

const int PIR_PIN = 2;    // OUT van de bewegingssensor op pin 2
const int ZOEMER_PIN = 8; // S van de zoemer op pin 8
const int LED_PIN = 13;   // Rode LED op pin 13 (via de weerstand)

void setup() {
  Serial.begin(9600);
  pinMode(PIR_PIN, INPUT);
  pinMode(ZOEMER_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  Serial.println(F("Sensor warmt op... even stilstaan!"));
  delay(10000); // De sensor heeft even nodig om te wennen aan de kamer
  Serial.println(F("Alarm staat aan."));
}

void loop() {
  if (digitalRead(PIR_PIN) == HIGH) { // Beweging gezien!
    Serial.println(F("Beweging! ALARM!"));
    for (int i = 0; i < 5; i++) { // 5 keer piepen en knipperen
      digitalWrite(LED_PIN, HIGH);
      digitalWrite(ZOEMER_PIN, HIGH);
      delay(150);
      digitalWrite(LED_PIN, LOW);
      digitalWrite(ZOEMER_PIN, LOW);
      delay(150);
    }
  }
  delay(100);
}
```

Wat doet de code?

- De PIR-sensor zit op pin 2. Ziet hij beweging, dan is de pin HIGH.
- Bij het opstarten wachten we 10 seconden. Zo kan de sensor wennen aan de kamer.
- Bij beweging piepen en knipperen we 5 keer.

Probeer eens...

- Voeg een knop toe om het alarm aan en uit te zetten
- Tel hoe vaak er beweging was en laat het zien in de seriële monitor
- Laat het alarm langer en harder klinken bij de tweede keer beweging

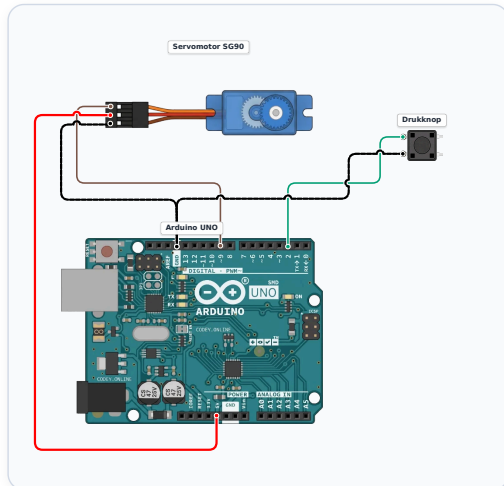
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/motion-alarm en klik op **MAAK DIT**. Dan staan de code en het aansluitschema meteen klaar.

★★ **Gevorderde beginner** ⌚ 20 minuten

Slagboom

Je bouwt een slagboom, zoals bij een parkeergarage. Druk op de knop: de servo draait de slagboom omhoog, wacht even en gaat weer dicht.

Aansluitschema



Wat heb je nodig?

1x Arduino UNO

1x Drukknop

1x Servomotor SG90

USB-kabel + Jumperkabels

Zo bouw je het

1. Zet de drukknoop over de gleuf van je breadboard: één pootje naar pin 2, een pootje aan de andere kant naar GND.
2. Sluit de servo aan: oranje naar pin 9, rood naar 5V en bruin naar GND.
3. Plak een rietje of ijsstokje op het armpje van de servo: dat is je slagboom!

De code

```
// Slagboom
// Druk op de knop: de slagboom (servo) gaat omhoog, wacht even
// en gaat weer dicht.

#include <Servo.h>

const int KNOP_PIN = 2;    // Drukknop op pin 2 (andere kant
// naar GND)
const int SERVO_PIN = 9;   // Oranje/gele draad van de servo op
// pin 9

const int DICT = 0;        // Hoek van de servo als de slagboom
// dicht is
const int OPEN = 90;       // Hoek van de servo als de slagboom
// open is

Servo slagboom;

void setup() {
  pinMode(KNOP_PIN, INPUT_PULLUP);

  slagboom.attach(SERVO_PIN);
  slagboom.write(DICT);    // Begin met de slagboom dicht
}

void loop() {
  if (digitalRead(KNOP_PIN) == LOW) { // Knop ingedrukt
    // Langzaam open draaien, graad voor graad
    for (int hoek = DICT; hoek <= OPEN; hoek++) {
      slagboom.write(hoek);
      delay(15);
    }
    delay(3000); // 3 seconden open: de
    // auto rijdt erdoor
    // Langzaam weer dicht
    for (int hoek = OPEN; hoek >= DICT; hoek--) {
      slagboom.write(hoek);
      delay(15);
    }
  }
}
```

Wat doet de code?

- We gebruiken de bibliotheek `Servo`. Met `write()` zeg je naar welke hoek de servo moet.
- Na een druk op de knop draaien we graad voor graad van 0 naar 90. Zo gaat het rustig.
- Na 3 seconden draaien we weer terug naar 0: de slagboom is dicht.

Probeer eens...

- Laat de slagboom sneller open- en dichtgaan
- Laat een zoemer piepen zolang de slagboom beweegt
- Laat een rode LED branden als de slagboom dicht is en een groene als hij open is

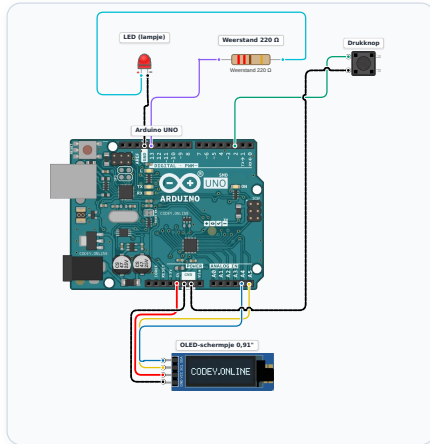
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/barrier en klik op **MAAK DIT**. Dan staan de code en het aansluitschema meteen klaar.

★★ **Gevorderde beginner** ⌚ 25 minuten

Reactiespel

Hoe snel ben jij? Wacht tot de LED aangaat en druk dan zo snel mogelijk op de knop. Het scherm laat je tijd zien in milliseconden. Daag je vrienden uit!

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Zet de drukknop over de gleuf van het breadboard: één pootje naar pin 2, een pootje aan de andere kant naar GND.
2. Zet de LED met een weerstand van 220 Ohm op pin 13, het korte pootje naar GND.
3. Sluit het OLED-scherm aan: SDA naar A4, SCL naar A5, VCC naar 5V en GND naar GND.

De code

```
// Reactiespel
// Wacht tot de LED aangaat en druk dan zo snel mogelijk op de knop.
// Het scherm laat zien hoe snel je was!

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const int KNOP_PIN = 2; // Drukknop op pin 2 (andere kant naar GND)
const int LED_PIN = 13; // LED op pin 13 (via de weerstand)

Adafruit_SSD1306 scherm(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

unsigned long besteTijd = 0; // Je snelste tijd tot nu toe

// Zet een tekst groot op het scherm
void toon(const __FlashStringHelper* boven, unsigned long getal) {
  scherm.clearDisplay();
  scherm.setTextSize(1);
  scherm.setCursor(0, 0);
  scherm.print(boven);
  if (getal > 0) {
    scherm.setTextSize(2);
    scherm.setCursor(0, 14);
    scherm.print(getal);
    scherm.print(F(" ms"));
  }
  scherm.display();
}

void setup() {
  pinMode(KNOP_PIN, INPUT_PULLUP);
  pinMode(LED_PIN, OUTPUT);
  randomSeed(analogRead(A0)); // Zorgt voor echt willekeurige wachttijden

  scherm.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  scherm.setTextColor(SSD1306_WHITE);
}

void loop() {
  toon(F("Druk op de knop..."), 0);
  while (digitalRead(KNOP_PIN) == HIGH); // Wachten tot je de knop indrukt
  while (digitalRead(KNOP_PIN) == LOW); // ...en weer loslaat
  toon(F("Klaar? Let op de LED!"), 0);

  delay(random(2000, 5000)); // Een geheime wachttijd tussen 2 en 5 seconden

  if (digitalRead(KNOP_PIN) == LOW) { // Te vroeg gedrukt? Valsspeler!
    toon(F("Te vroeg! Opnieuw."), 0);
    delay(2000);
    return;
  }

  digitalWrite(LED_PIN, HIGH); // NU!
  unsigned long start = millis();
  while (digitalRead(KNOP_PIN) == HIGH); // Wachten op jouw druk
  unsigned long tijd = millis() - start;
  digitalWrite(LED_PIN, LOW);

  if (besteTijd == 0 || tijd < besteTijd) {
    besteTijd = tijd;
    toon(F("NIEUW RECORD!"), tijd);
  } else {
    toon(F("Jouw tijd:"), tijd);
  }
  delay(3000);
}
```

Wat doet de code?

- Na een druk op de knop wachten we een geheime tijd tussen 2 en 5 seconden (`random()`).
- Dan gaat de LED aan en starten we een stopwatch met `millis()`.
- Als je drukt, rekenen we uit hoeveel milliseconden het duurde. Je beste tijd onthouden we.

Probeer eens...

- Maak er een spel voor twee spelers van, met elk een eigen knop
- Laat een zoemer piepen bij een nieuw record
- Speel 5 rondes en laat daarna je gemiddelde tijd zien

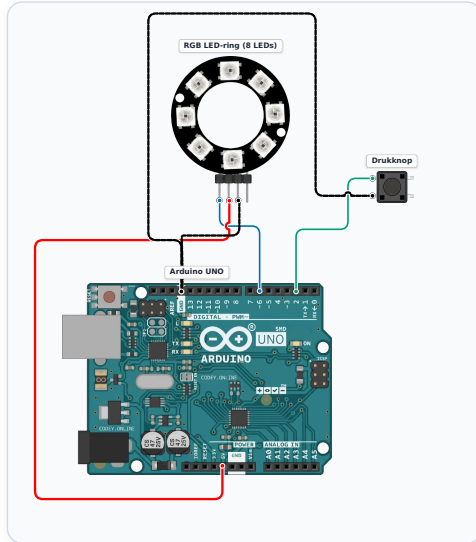
🔧 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/reaction-game en klik op **MAAK DIT**. Dan staan de code en het aansluitschema meteen klaar.

★★ **Gevorderde beginner** ⌚ **15 minuten**

Elektronische dobbelsteen

Je maakt een dobbelsteen van licht. Druk op de knop: de lampjes rollen even, en dan zie je hoeveel je hebt gegooid. 1 tot 6 lampjes!

Aansluitschema



Wat heb je nodig?



1× Arduino UNO



1× Drukknop



1× RGB LED-ring (8 LEDs)



USB-kabel + Jumperkabels

Zo bouw je het

1. Zet de drukknoop over de gleuf van het breadboard: één pootje naar pin 2, een pootje aan de andere kant naar GND.
2. Sluit de LED-ring aan: DI naar pin 6, 5V naar 5V en GND naar GND.

De code

```
// Elektronische dobbelsteen
// Druk op de knop: de LED-ring rolt even en laat dan 1 tot 6
lampjes branden.

#include <Adafruit_NeoPixel.h>

const int KNOP_PIN = 2;    // Drukknoop op pin 2 (andere kant naar
GND)
const int RING_PIN = 6;    // DI van de LED-ring op pin 6
const int AANTAL_LEDS = 8;

Adafruit_NeoPixel ring(AANTAL_LEDS, RING_PIN, NEO_GRB + NEO_KHZ800);

// Laat een aantal lampjes branden in één kleur
void toonGetal(int getal, uint32_t kleur) {
  ring.clear();
  for (int i = 0; i < getal; i++) {
    ring.setPixelColor(i, kleur);
  }
  ring.show();
}

void setup() {
  pinMode(KNOP_PIN, INPUT_PULLUP);
  randomSeed(analogRead(A0)); // Zorgt dat de dobbelsteen echt
willekeurig is
  ring.begin();
  ring.setBrightness(50);
  ring.show();
}

void loop() {
  if (digitalRead(KNOP_PIN) == LOW) {
    // Even "rollen": snel wisselende getallen, steeds een beetje
    langzamer
    for (int i = 0; i < 15; i++) {
      toonGetal(random(1, 7), ring.Color(0, 80, 255));
      delay(40 + i * 10);
    }
    int worp = random(1, 7); // Een getal van 1
    tot en met 6
    toonGetal(worp, ring.Color(0, 255, 60)); // Het echte getal
    in groen
    delay(500);
  }
}
```

Wat doet de code?

- `random(1, 7)` kiest een willekeurig getal van 1 tot en met 6.
- Eerst laten we snel wisselende getallen zien in het blauw. Zo lijkt het alsof de dobbelsteen rolt.
- Daarna laat `toonGetal()` het echte getal zien in het groen.

Probeer eens...

- Geef elk getal een eigen kleur
- Laat een zoemer kort piepen als de dobbelsteen stilligt
- Maak er een dobbelsteen van 1 tot 8 van

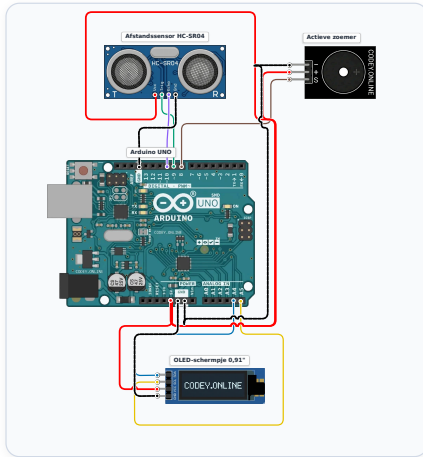
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/dice en klik op **MAAK DIT**. Dan staan de code en het aansluitschema meteen klaar.

★★★★ Uitdaging ⌚ 30 minuten

Parkeersensor luxe

De luxe versie van de parkeersensor! Hij piept sneller als je dichterbij komt, en het OLED-scherm laat de afstand zien in centimeters. Met een balkje dat langer wordt als je dichterbij komt.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Sluit de HC-SR04 aan: Trig naar pin 9, Echo naar pin 10, Vcc naar 5V en Gnd naar GND.
2. Sluit de zoemer aan: S naar pin 8, + naar 5V en – naar GND.
3. Sluit het OLED-scherm aan: SDA naar A4, SCL naar A5, VCC naar 5V en GND naar GND.
4. Drie onderdelen hebben 5V en GND nodig. Gebruik de + en – strips van je breadboard.

De code

```
// Parkeersensor luxe
// Net als de gewone parkeersensor, maar nu zie je de afstand ook op het OLED-scherm.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const int TRIG_PIN = 9; // Trig van de HC-SR04 op pin 9
const int ECHO_PIN = 10; // Echo van de HC-SR04 op pin 10
const int ZOEMER_PIN = 8; // S van de zoemer op pin 8

Adafruit_SSD1306 scherm(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

// Meet de afstand in centimeters
long meetAfstand() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long tijd = pulseIn(ECHO_PIN, HIGH, 30000);
  if (tijd == 0) return 999; // Geen echo: niets in de buurt
  return tijd / 58;
}

// Zet de afstand groot op het scherm, met een balkje eronder
void toonAfstand(long afstand) {
  scherm.clearDisplay();
  scherm.setTextSize(2);
  scherm.setCursor(0, 0);
  if (afstand >= 999) {
    scherm.print(F("----"));
  } else {
    scherm.print(afstand);
    scherm.print(F(" cm"));
  }
  // Balkje: hoe dichterbij, hoe langer
  int lengte = map(constrain(afstand, 0, 100), 0, 100, 128, 0);
  scherm.fillRect(0, 24, lengte, 8, SSD1306_WHITE);
  scherm.display();
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(ZOEMER_PIN, OUTPUT);
  scherm.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  scherm.setTextColor(SSD1306_WHITE);
}

void loop() {
  long afstand = meetAfstand();
  toonAfstand(afstand);

  if (afstand < 5) {
    digitalWrite(ZOEMER_PIN, HIGH); // Heel dichtbij: één lange toon
    delay(100);
  } else if (afstand < 50) {
    digitalWrite(ZOEMER_PIN, HIGH); // Korte piep
    delay(50);
    digitalWrite(ZOEMER_PIN, LOW);
    delay(afstand * 10); // Dichterbij = sneller piepen
  } else {
    digitalWrite(ZOEMER_PIN, LOW);
    delay(100);
  }
}
```

Wat doet de code?

- `meetAfstand()` meet de afstand in centimeters met de HC-SR04.
- `toonAfstand()` zet het getal groot op het scherm en tekent een balkje met `fillRect()`.
- De zoemer piept sneller naarmate de afstand kleiner wordt.

Probeer eens...

- Laat "STOP!" op het scherm zien als je dichterbij 5 cm bent
- Laat de afstand in meters zien als iets verder dan 1 meter weg is
- Voeg de LED-ring toe die van groen naar rood kleurt

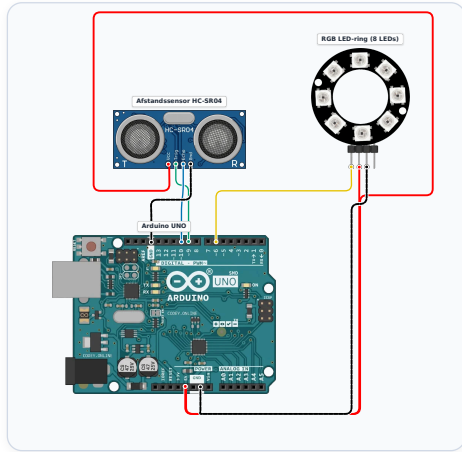
🔧 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/parking-sensor-deluxe en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.

★★★★ Uitdaging ⌚ 25 minuten

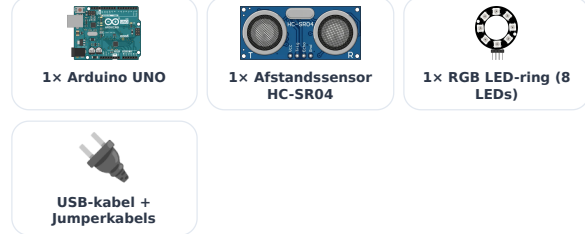
Afstandsmeter met lichtbalk

Je maakt een lichtbalk die laat zien hoe dichtbij iets is. Ver weg: geen lampjes. Hoe dichterbij, hoe meer lampjes er gaan branden. Eerst groen, dan oranje en heel dichtbij rood.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

- Sluit de HC-SR04 aan: Trig naar pin 9, Echo naar pin 10, Vcc naar 5V en Gnd naar GND.
- Sluit de LED-ring aan: DI naar pin 6, 5V naar 5V en GND naar GND.

De code

```
// Afstandsmeter met lichtbalk
// Hoe dichterbij je komt, hoe meer lampjes er branden: groen, dan
// oranje, dan rood.

#include <Adafruit_NeoPixel.h>

const int TRIG_PIN = 9; // Trig van de HC-SR04 op pin 9
const int ECHO_PIN = 10; // Echo van de HC-SR04 op pin 10
const int RING_PIN = 6; // DI van de LED-ring op pin 6
const int AANTAL_LEDS = 8;
const int MAX_AFSTAND = 80; // Verder dan dit (in cm): alle lampjes uit

Adafruit_NeoPixel ring(AANTAL_LEDS, RING_PIN, NEO_GRB + NEO_KHZ800);

long meetAfstand() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long tijd = pulseIn(ECHO_PIN, HIGH, 30000);
  if (tijd == 0) return 999;
  return tijd / 58;
}

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  ring.begin();
  ring.setBrightness(40);
  ring.show();
}

void loop() {
  long afstand = meetAfstand();
  Serial.print(F("Afstand: "));
  Serial.print(afstand);
  Serial.println(F(" cm"));

  // Hoeveel lampjes moeten er branden? Dichtbij = 8, ver weg = 0
  int aantal = map(constrain(afstand, 0, MAX_AFSTAND), 0, MAX_AFSTAND, AANTAL_LEDS, 0);

  for (int i = 0; i < AANTAL_LEDS; i++) {
    if (i < aantal) {
      if (i < 3) ring.setPixelColor(i, ring.Color(0, 255, 0));
      // Groen
      else if (i < 6) ring.setPixelColor(i, ring.Color(255, 120, 0));
      // Oranje
      else ring.setPixelColor(i, ring.Color(255, 0, 0));
      // Rood
    } else {
      ring.setPixelColor(i, 0);
    }
    // Uit
  }
  ring.show();
  delay(100);
}
```

Wat doet de code?

- `meetAfstand()` meet de afstand met de HC-SR04.
- `map()` rekent de afstand om naar een aantal lampjes: 0 tot 8.
- De eerste 3 lampjes zijn groen, de volgende 3 oranje en de laatste 2 rood.

Probeer eens...

- Laat alle lampjes rood knipperen als je dichterbij 10 cm bent
- Voeg de zoemer toe die piept bij rood
- Maak de meter gevoeliger: alle lampjes aan bij 40 cm in plaats van 80 cm

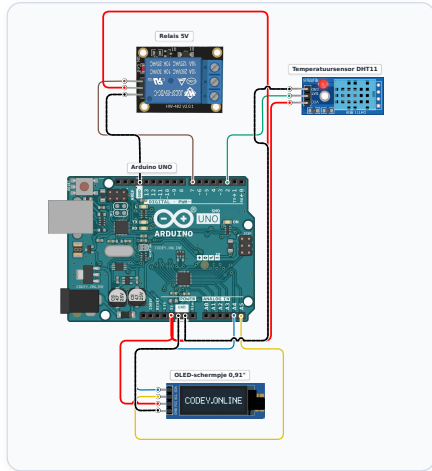
🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/distance-light-bar en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.

★★★ Uitdaging ⌚ 30 minuten

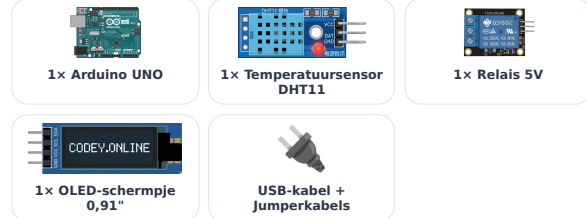
Slimme ventilator

Je bouwt een thermostaat. Wordt het warmer dan 25 graden, dan klikt het relais aan. Zo kan je Arduino bijvoorbeeld een ventilatortje op een batterij aanzetten. Op het scherm zie je de temperatuur en of de ventilator aan staat.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Sluit de DHT11 aan: DAT naar pin 2, VCC naar 5V en GND naar GND.
2. Sluit het relais aan: S naar pin 7, + naar 5V en – naar GND.
3. Sluit het OLED-scherm aan: SDA naar A4, SCL naar A5, VCC naar 5V en GND naar GND.
4. Wil je echt iets schakelen? Gebruik dan alleen een batterij of lage spanning op COM en NO, nooit het stopcontact.

De code

```
// Slimme ventilator (thermostaat)
// Wordt het te warm? Dan schakelt het relais aan, bijvoorbeeld voor een ventilator.
// Op het scherm zie je de temperatuur en of de ventilator aan staat.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

const int DHT_PIN = 2;      // DAT van de DHT11 op pin 2
const int RELAIS_PIN = 7;   // S van het relais op pin 7
const float AAN_BOVEN = 25.0; // Boven deze temperatuur gaat de ventilator aan
const float UIT_ONDER = 24.0; // Onder deze temperatuur gaat hij weer uit

DHT dht(DHT_PIN, DHT11);
Adafruit_SSD1306 scherm(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

bool ventilatorAan = false;

void setup() {
  Serial.begin(9600);
  pinMode(RELAIS_PIN, OUTPUT);
  digitalWrite(RELAIS_PIN, LOW); // Begin met het relais uit
  dht.begin();
  scherm.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  scherm.setTextColor(SSD1306_WHITE);
}

void loop() {
  float temperatuur = dht.readTemperature();

  if (!isnan(temperatuur)) {
    // Twee grenzen, zodat het relais niet steeds aan en uit klikt
    if (temperatuur > AAN_BOVEN) ventilatorAan = true;
    if (temperatuur < UIT_ONDER) ventilatorAan = false;
    digitalWrite(RELAIS_PIN, ventilatorAan ? HIGH : LOW);
  }

  scherm.clearDisplay();
  scherm.setTextSize(2);
  scherm.setCursor(0, 0);
  if (isnan(temperatuur)) {
    scherm.print(F("DHT11 ?"));
  } else {
    scherm.print(temperatuur, 1);
    scherm.print((char)247);
    scherm.print(F("C"));
  }

  scherm.setTextSize(1);
  scherm.setCursor(0, 22);
  scherm.print(ventilatorAan ? F("Ventilator: AAN") : F("Ventilator: uit"));
  scherm.display();

  Serial.print(F("Temperatuur: "));
  Serial.print(temperatuur);
  Serial.println(ventilatorAan ? F(" C -> AAN") : F(" C -> uit"));

  delay(2000);
}
```

Wat doet de code?

- De DHT11 meet de temperatuur. Boven 25 graden zetten we het relais aan.
- Pas onder 24 graden gaat het relais weer uit. Met die twee grenzen klikt hij niet steeds aan en uit.
- Het OLED-scherm toont de temperatuur en "Ventilator: AAN" of "uit".

Probeer eens...

- Voeg twee knoppen toe om de grens-temperatuur hoger en lager te zetten
- Laat een LED branden zolang de ventilator aan staat
- Laat de ventilator ook aangaan als de luchtvochtigheid boven de 70% komt

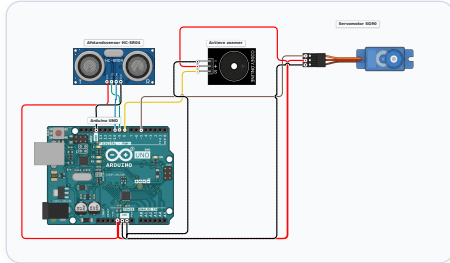
🔧 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/thermostat en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.

★★★★ Uitdaging ⌚ 30 minuten

Automatische slagboom

De slimme versie van de slagboom! De afstandssensor ziet dat er een auto aankomt. Dan piept de zoemer en gaat de slagboom vanzelf open. Is de auto weg? Dan gaat hij weer dicht.

Aansluitschema



Wat heb je nodig?



Zo bouw je het

1. Sluit de HC-SR04 aan: Trig naar pin 9, Echo naar pin 10, Vcc naar 5V en Gnd naar GND.
2. Sluit de servo aan: oranje naar pin 5, rood naar 5V en bruin naar GND.
3. Sluit de zoemer aan: S naar pin 8, + naar 5V en – naar GND.
4. Plak een rietje of ijsstokje op de servo als slagboom en zet de sensor ervoor.

De code

```
// Automatische slagboom
// Komt er een auto aan (iets dichters dan 15 cm)? Dan piept de zoemer en
// gaat de slagboom open.

#include <Servo.h>

const int TRIG_PIN = 9;      // Trig van de HC-SR04 op pin 9
const int ECHO_PIN = 10;     // Echo van de HC-SR04 op pin 10
const int SERVO_PIN = 5;     // Oranje/gele draad van de servo op pin 5
const int ZOEMER_PIN = 8;    // S van de zoemer op pin 8
const int DICHTBIJ = 15;     // Vanaf deze afstand (cm) gaat de slagboom
// open

const int DICHT = 0;         // Hoek als de slagboom dicht is
const int OPEN = 90;        // Hoek als de slagboom open is

Servo slagboom;

long meetAfstand() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long tijd = pulseIn(ECHO_PIN, HIGH, 30000);
  if (tijd == 0) return 999;
  return tijd / 58;
}

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(ZOEMER_PIN, OUTPUT);
  slagboom.attach(SERVO_PIN);
  slagboom.write(DICHT);
}

void loop() {
  long afstand = meetAfstand();

  if (afstand < DICHTBIJ) {
    Serial.println(F("Auto gezien! Slagboom open."));
    digitalWrite(ZOEMER_PIN, HIGH); // Kort piepje
    delay(100);
    digitalWrite(ZOEMER_PIN, LOW);

    slagboom.write(OPEN);
    delay(1000);
    // Open blijven zolang de auto er nog staat
    while (meetAfstand() < DICHTBIJ) {
      delay(200);
    }
    delay(2000); // Nog even wachten tot de auto
    // echt weg is
    slagboom.write(DICHT);
    Serial.println(F("Slagboom dicht."));
  }
  delay(100);
}
```

Wat doet de code?

- De HC-SR04 meet steeds de afstand. Dichter dan 15 cm? Dan staat er een auto.
- De zoemer piept kort en de servo draait naar 90 graden: open.
- Zolang de auto er staat, blijft de slagboom open. Daarna wachten we 2 seconden en gaat hij dicht.

Probeer eens...

- Tel hoeveel auto's er langs zijn gekomen en laat het zien in de seriële monitor
- Voeg een rode en een groene LED toe als stoplicht bij de slagboom
- Laat de slagboom alleen opengaan als er ook op een knop wordt gedrukt

🚀 Bouw dit project met Codey: open codey.online/nl/starter-kits/uno/projects/automatic-barrier en klik op MAAK DIT. Dan staan de code en het aansluitschema meteen klaar.