Elektro challenge 2025

Workshop materialen/motoren/hefmechanisme
14 februari 2025
Jos Scheelen RTC Antwerpen

RTC ANTWERPEN

Uitgangspunten

Controller:

- Arduino UNO (controller)

Presentaties:

- Google Education

Onderdelen

- Stuklijst elektromateriaal (POV)
- 3D Data STL code en G code (RTC Antwerpen)

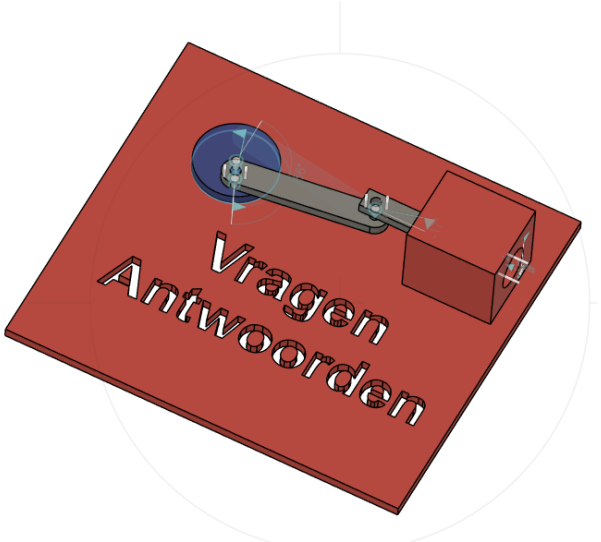
RTC ANTWERPEN

Technologische inhoud

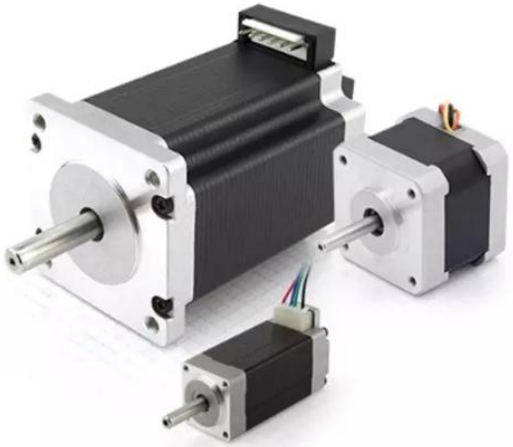
- Unipolaire Stepper Motor
- Bipolaire Stepper Motor
  Hoog koppel
- Bipolaire Stepper Motor
  Hoog toerental
- Servomotor
- DC-motor

- Presentatie technologie
- Demo
- Software specifieke code

RTC ANTWERPEN



Wat is een stepper motor?



Een stepper motor is een type elektromotor dat werkt door middel van stappen.

Een traditionele elektromotor werkt continu.

Een gehele omwenteling van 360 graden wordt opgedeeld in 200 stapjes, een stapje is dan 360 / 200 ... 1.8 graden.

Precieze controle over de positie.

RTC ANTWERPEN

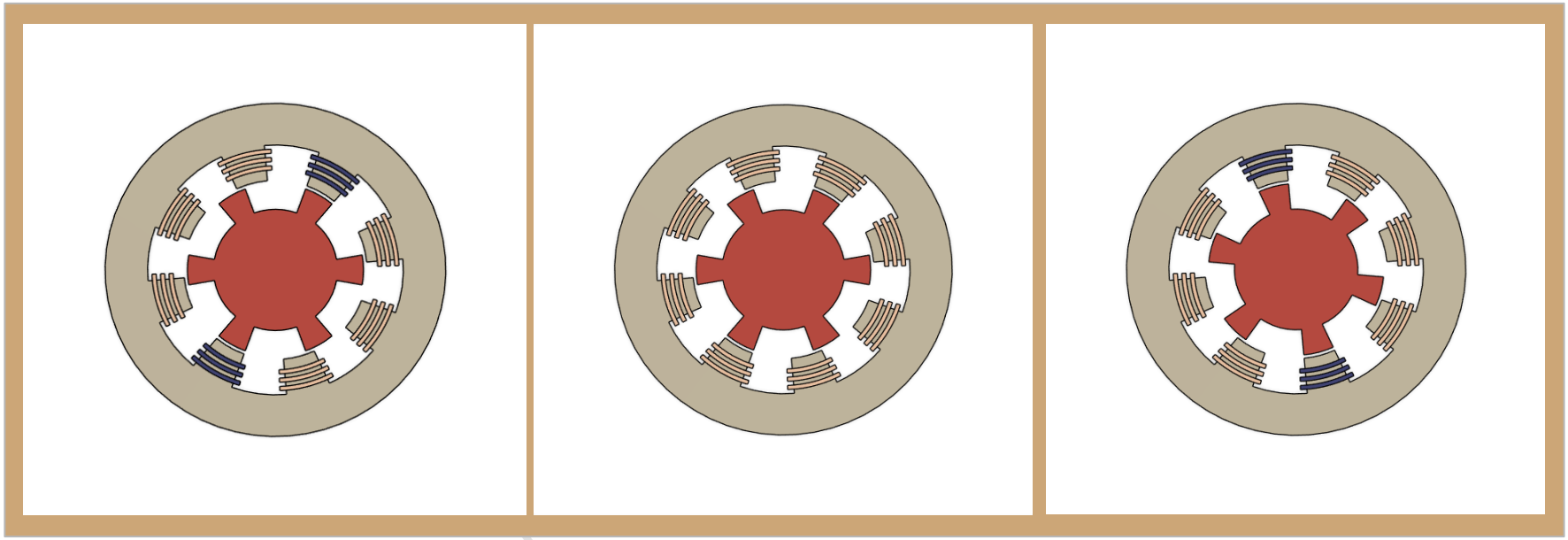
Kenmerken van een stepper motor?

- ***Discrete stappen*** 200 stappen voor één omwenteling
- ***Open-loop control***, geen terugkoppeling of meting van de positie
- ***Hoge precisie***, nauwkeurige positionering
- ***Hoge koppel bij lage snelheden*** belangrijk voor CNC en tractie

RTC ANTWERPEN

3D-printers, procescontrole aandrijving, robots, camera's, meetinstrumenten

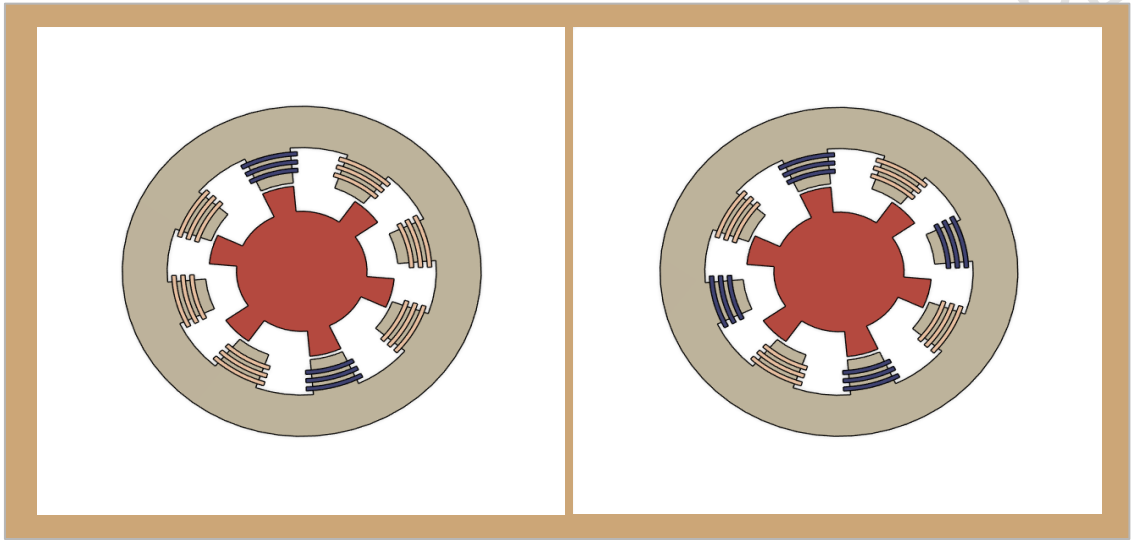
Principewerking van een stepper motor



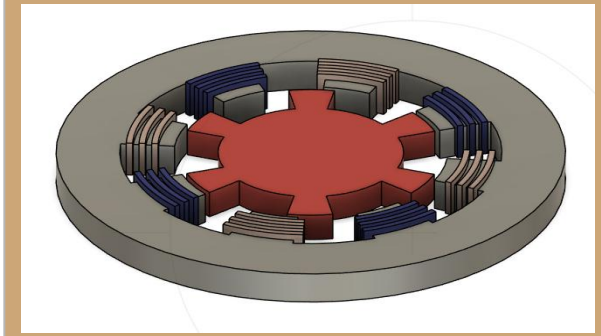
Magnetisatie spoel 1 *Geen magnetisatie* *Magnetisatie spoel 2*

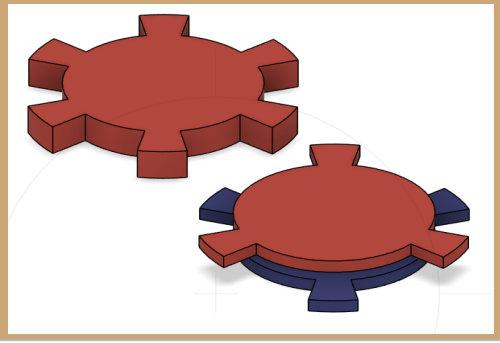
Geen 4 maar slechts 2 spoelen

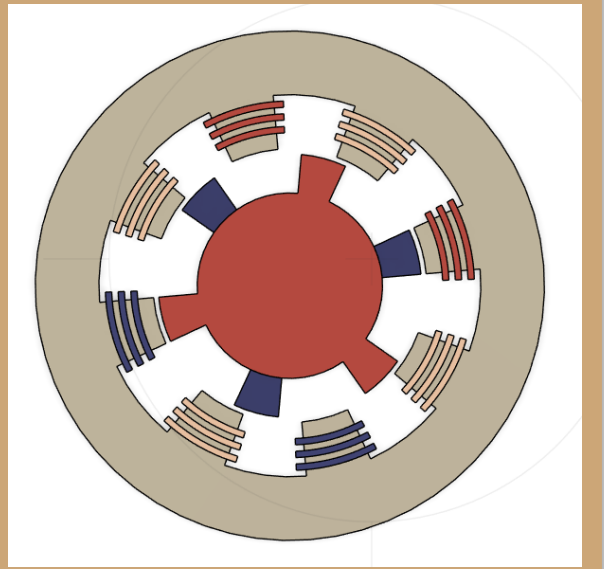
De spoelen worden niet individueel aangesloten maar zijn in serie geschakeld in twee groepen. De volgorde van de spoelen en de oriëntatie van de magnetische velden van de spoelen bepalen de werking van de motor.



Principewerking van een stepper motor





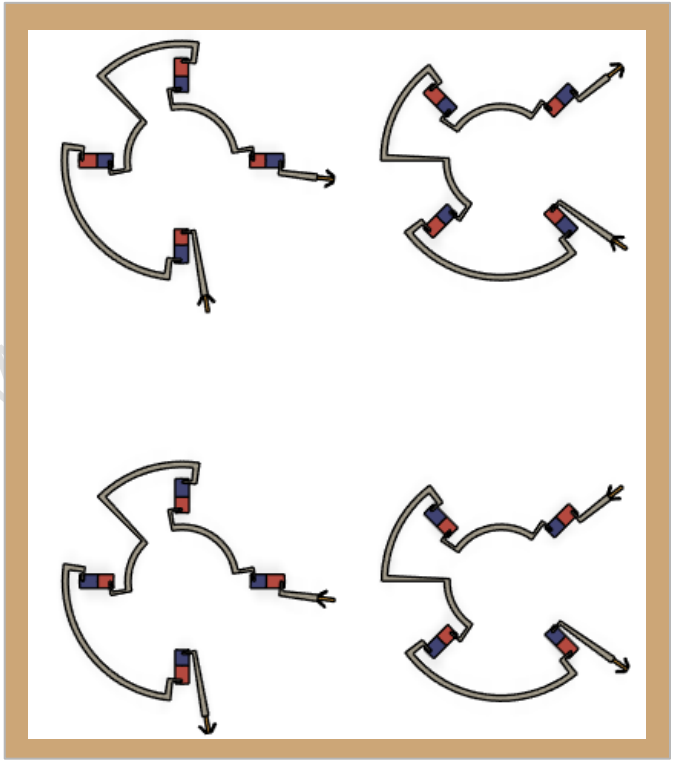


De rotor is een permanente magneet met een zuid- en noordpool.

Aansluiten van de spoelen, rechterhandregel

We moeten nu goed opletten hoe we de spoelen gaan aansluiten. Per pool zijn de spoelen identiek; de richting van de windingen en de aansluitingen zijn hetzelfde. Hoe verbinden we nu de verschillende spoelen?

Van de acht spoelen worden er afwisselend vier geactiveerd. Telkens wanneer we eenzelfde groep aansturen, keren we de stroomrichting om. Je kan dan de stroomrichting bepalen om de juiste magnetisatie te bereiken. Op basis hiervan bepaal je hoe je de spoelen moet aansluiten.

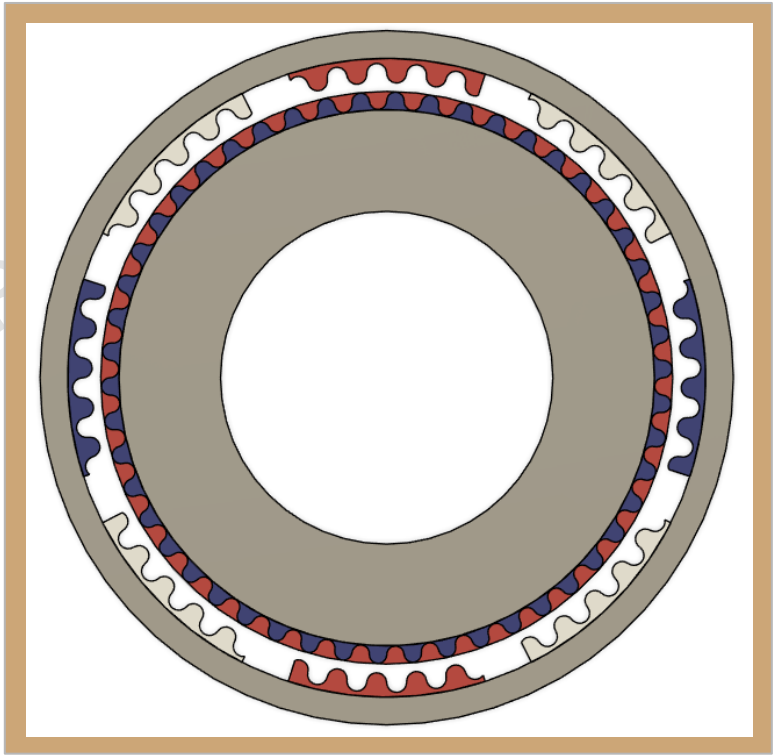


PTC ANTV

Getande polen

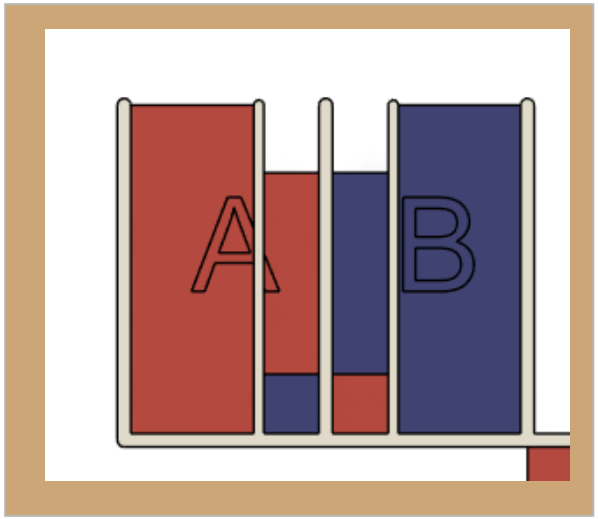
Achtereenvolgens gaan we opnieuw de spoelen bekrachtigen zoals in ons principeschema. Iedere zuidpool ligt nu perfect tegen een noordpool. Zelfs als de rotor één tand zou draaien, voldoen we aan die voorwaarde. Onderbreken we de bekrachtiging en bekrachtigen we de volgende spoelen, dan draait de rotor netjes ¼ tand en is de rotor weer stabiel.

Iedere puls: ¼ tand 50 tanden 200 stapjes voor 360 graden.



RTC ANTWER

Microstepping



Tot nu toe overlappen de stromen in de spoelen A en B elkaar niet. Indien we echter zowel spoel A als spoel B bekrachtigen, is de positie van de rotor niet bepaald. Als de spanning van A bijvoorbeeld 75% is en die van B 25% van de maximumwaarde op hetzelfde moment, dan staat de rotor niet precies voor A en niet precies voor B. De twee magnetische velden resulteren in een vectoriële som en de rotor richt zich naar de som van die twee velden. Met andere woorden, dit leidt niet tot een volledige stap, maar tot twee tussenstappen. De tussenstappen zijn niet zo nauwkeurig en de rotor is ook niet zo sterk gepolariseerd. Het koppel is minder.

RTC ALTMERREN

Controller

Door het spel van de timing, overlapping en sterkte van de stroom kun je de rotor in verschillende tussenposities plaatsen. Meestal wordt dit niet door de processor van je automatiserings-controller uitgevoerd, maar regelt een afzonderlijke chip de timing voor je. Je kiest dan:

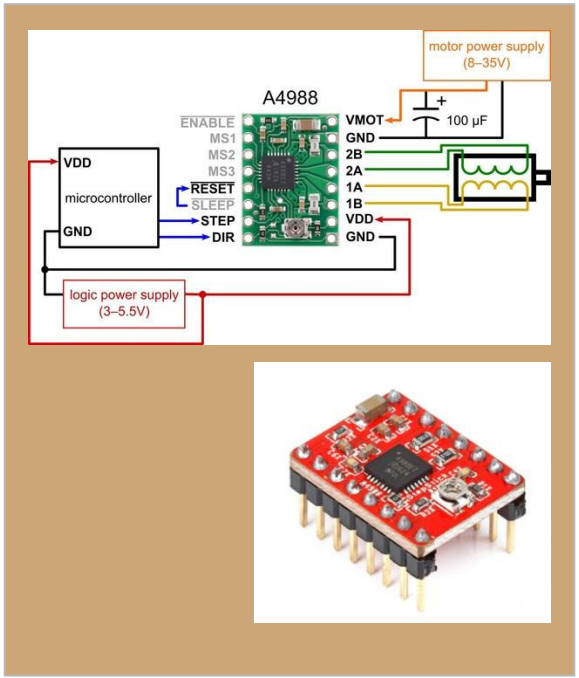
draairichting

microstepping 1, ½, ¼, ⅛ …

step nu!

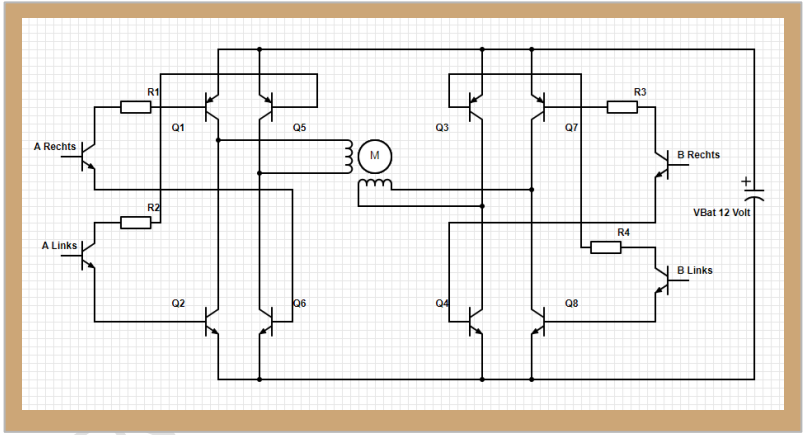
en eventueel een stroomloze situatie om de stappenmotor "vrij" te laten

De keuze tussen de controllers is zeer groot, afhankelijk van je toepassing.



RTC ANTWERPEN

Aansturing stepper motor

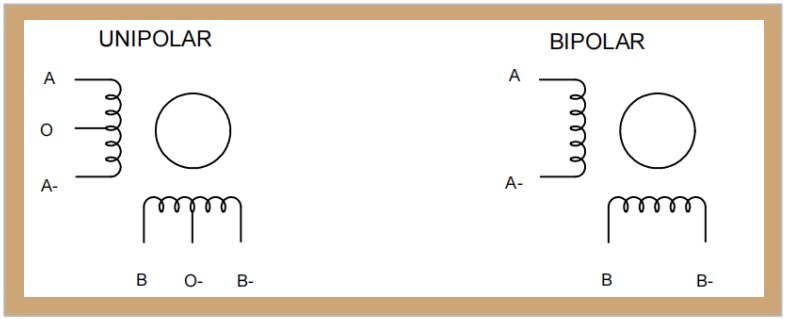


De stepper motor kan op verschillende manieren aangestuurd worden. Al deze manieren hebben meestal een gemeenschappelijk component: de "double H bridge".

Een double H-brug (of dubbele H-brug) is een configuratie van elektronische schakelingen. Een double H-brug bestaat uit twee H-bruggen. Elke H-brug heeft vier schakelaars (bijvoorbeeld transistors of MOSFET’s) die in een H-vorm zijn gerangschikt. Door deze schakelaars op de juiste manier te openen en te sluiten, kan de stroom in beide richtingen door een motorwikkeling worden gestuurd, waardoor de motor kan draaien of stappen in beide richtingen.

RTC ANTWERPEN

Unipolaire en bipolaire



Het verschil tussen een unipolaire en een bipolaire stepper motor zit voornamelijk in de manier waarop de windingen van de motor worden aangestuurd en hoe de stroom door deze windingen vloeit. Een unipolaire motor heeft windingen met een middenaftakking, waardoor elke winding in twee helften is verdeeld. Dit resulteert meestal in zes aansluitingen (of vijf bij een gemeenschappelijke middenaftakking). Een bipolaire motor heeft geen middenaftakking. Het heeft vier aansluitingen. Bij een unipolaire vloeit de stroom steeds in één richting door elke helft van de winding te vloeien. Bij de bipolaire moet de stroom in beide richtingen door de windingen kunnen vloeien, wat een complexer stuurcircuit vereist (meestal met een H-brug per winding).

RTC ANTWERPEN

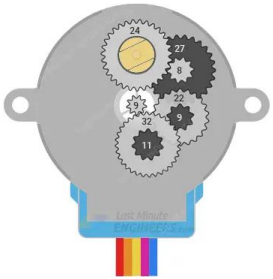
Unipolaire en bipolaire

Kenmerk	Unipolaire Motor	Bipolaire Motor
Aansluitingen	*Eén richting per winding*	*Beide richtingen nodig*
Stroomrichting	*Simpel*	*Complexer (H-bruggen)*
Complexiteit aansturing	*Lager*	*Hoger*
Koppel	*Lager*	*Hoger*
Toepassing	*Simpele systemen*	*Hogere prestatie nodig*

RTC ANTWERPEN

Kies een unipolaire motor als eenvoud en lage kosten belangrijk zijn, en een bipolaire motor als je meer kracht of efficiëntie nodig hebt.

Unipolaire Stepper Motor



Gear Ratios:

- 32 / 9
- 22 / 11
- 27 / 9
- 24 / 8

Multiplying the gear ratios:

$$\frac{32}{9} \times \frac{22}{11} \times \frac{27}{9} \times \frac{24}{8} = 64$$

This gives us a 64:1 gear ratio

Wikkeling met middenaftakking : Een unipolaire stepper motor heeft wikkelingen met een middenaftakking. Deze aftakking maakt het mogelijk om elke helft van een wikkeling afzonderlijk te gebruiken, waardoor de stroom slechts in één richting door de spoelen hoeft te stromen.

Eenvoudige aansturing : De unipolaire configuratie vereist minder complexe schakelingen dan een bipolaire motor. Elke wikkeling kan worden in- of uitgeschakeld met eenvoudige transistoren, zonder dat de stroomrichting hoeft te worden omgekeerd.

Meer bekabeling: Unipolaire motoren hebben meestal zes draden (of soms vijf), omdat ze zowel de middenaftakkingen als de uiteinden van de spoelen nodig hebben. Dit maakt de bedrading complexer dan bij bipolaire motoren. Je kan de ook geen gebruik maken van de middenaftakking en de motor aandrijven zoals een bipolaire motor.

Lager koppel in vergelijking met bipolaire motoren: Omdat slechts de helft van een wikkeling op een bepaald moment actief is, produceert een unipolaire stepper motor minder koppel dan een vergelijkbare bipolaire motor.

Veelgebruikte toepassingen: Unipolaire stepper motoren worden vaak gebruikt in eenvoudige toepassingen zoals printers, scanners en kleine robotica, waar een eenvoudige aansturing belangrijker is dan hoog koppel. *Voor onze toepassing is het gebruik voor de grijper, stuurinrichting aangewezen.*

RTC ANTWERPEN

Unipolaire Stepper Motor Arduino

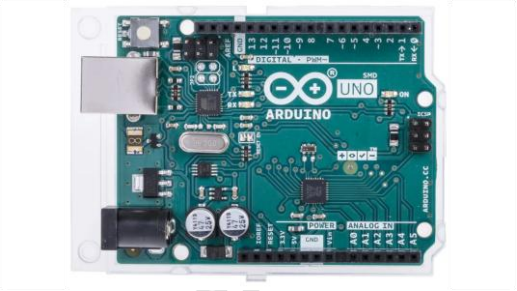
Arduino UNO: Deze microcontroller is erg populair, vooral in scholen en STEM-richtingen.

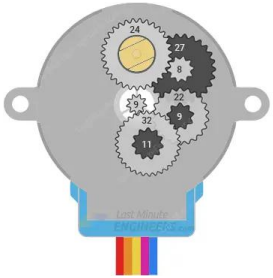
Joystick: Wordt gebruikt voor de bediening van de motor, maar kan desgewenst worden vervangen door WiFi- of Bluetooth-besturing.

I2C IO-expander PCF8574: Zorgt voor een onbeperkt aantal digitale IO's en beschermt je Arduino tegen foutief gebruik, overspanning en kortsluiting.

12 Volt voeding: Een gezamenlijke voeding voor zowel de Arduino als de controller van de steppermotor.

28BYJ-48 unipolaire steppermotor (met driverboard): Een populaire unipolaire steppermotor met tandwielreductie voor een hoger koppel.





Gear Ratios:

- 32 / 9
- 22 / 11
- 27 / 9
- 24 / 8

Multiplying the gear ratios:

$$\frac{32}{9} \times \frac{22}{11} \times \frac{27}{9} \times \frac{24}{8} = 64$$

This gives us a 64:1 gear ratio

Bipolaire Stepper Motor

Werkingsprincipe: Een bipolaire stepper motor is een type elektromotor die in discrete stappen draait. Dit wordt bereikt door afwisselend verschillende elektromagneten in de stator te activeren, waardoor de permanente magneet van de rotor zich stapsgewijs verplaatst.

Twee spoelen (wikkelingen): De motor heeft twee spoelen, maar geen middenaftakking (zoals bij unipolaire motoren). De stroomrichting door elke spoel wordt omgekeerd om het magnetische veld te manipuleren en de rotor te laten bewegen.

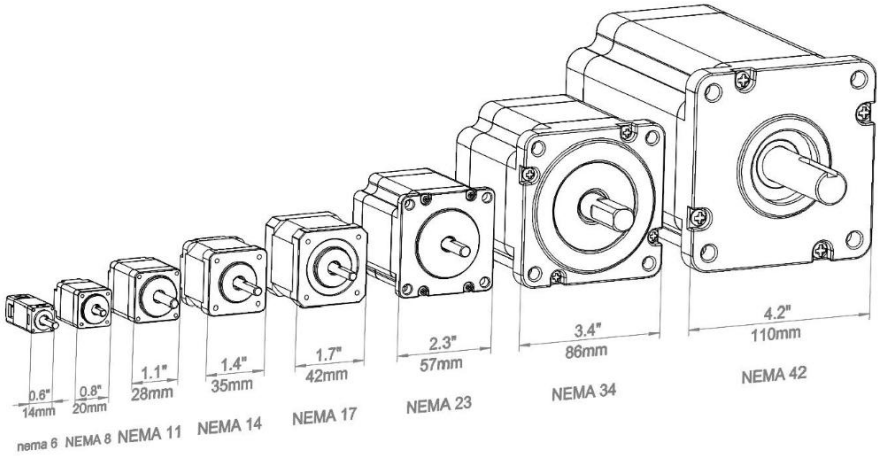
Dubbele polariteit: Omdat de stroom in beide richtingen door de spoelen moet kunnen stromen, vereist een bipolaire motor een **H-brug** schakeling of driver. Dit maakt het mogelijk om de polariteit van de spanning op de spoelen om te keren.

Voordelen: meer koppel: In vergelijking met unipolaire motoren biedt een bipolaire stepper motor meer koppel, omdat de hele wikkeling wordt gebruikt (in plaats van de helft, zoals bij unipolaire motoren).

Toepassingen: Bipolaire stepper motoren worden veel gebruikt in toepassingen waar precisie en koppel belangrijk zijn, zoals 3D-printers, CNC-machines, robots, en andere mechatronische systemen. Voor onze toepassing is het gebruik voor de *aandrijving en mechanische bewegingen van armen, kranen en onderdelen* waarvoor de unipolaire stepper motor onvoldoende vermogen heeft aangewezen.

RTC ANTWERPEN

Bipolaire Stepper Motor



RTC ANTWERPEN

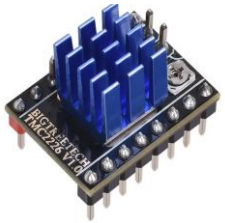
De NEMA-aanduiding verwijst naar de mechanische afmetingen van een motor. Een veelgebruikte variant is de **NEMA17**, die populair is bij de bouw van 3D-printers. De NEMA17 biedt een goed evenwicht tussen compacte afmetingen en beschikbaar vermogen, wat het een geschikte keuze maakt voor Arduino-toepassingen.

Binnen de NEMA17-reeks zijn motoren verkrijgbaar met verschillende vermogens en sterktes, en de dikte van de motor kan variëren. Wat echter constant blijft, is de buitenafmeting van **42 x 42 mm**.

Een steppermotor is een zelfremmende motor. Zolang de spanning op de spoelen niet verandert, blijft de motor stil staan. Als je de as toch wilt verdraaien, moet je een minimale kracht uitoefenen, uitgedrukt in kilogram-centimeter. Bijvoorbeeld: een houdkoppel van **4,8 kg-cm** betekent dat de motoras stil blijft staan wanneer een hefboom van 10 cm lengte een gewicht van 480 gram draagt.

Voor een NEMA17 varieert deze waarde doorgaans tussen **1,33 en 5,0 kg-cm**. Kies het juiste houdkoppel voor je toepassing. Let wel: hoe groter het houdkoppel, hoe meer stroom door de motor vloeit.

Bipolaire Stepper Motor Driver



De Arduino Uno kan een bipolaire stepper motor niet rechtstreeks aansturen. Bij een bipolaire stepper motor moet niet alleen de stroomsterkte worden gevarieerd, maar ook de polariteit, in tegenstelling tot een unipolaire stepper motor. Daarom is het noodzakelijk om een stepper-driver-module te gebruiken.

Er zijn verschillende modellen stepper-drivers beschikbaar, elk met hun eigen specifieke eigenschappen. Een belangrijke eigenschap is de maximale stroom die de driver kan leveren. Let op: de maximale stroom die door een stepper motor mag worden gestuurd, wordt bepaald door de fabrikant. Een te hoge stroom kan de motor oververhitten en beschadigen.

Vrijwel alle stepper-drivers zijn uitgerust met een regelpotentiometer waarmee de stroom kan worden ingesteld. Praktisch gezien is het eenvoudig om de snelheid van de stepper motor te variëren tussen 0 en de maximale waarde. Stel vervolgens de stroom zo af dat de motor soepel en zonder problemen functioneert.

Een populaire A4988 driver levert maximaal 2 ampere terwijl een arduino motor shield V2 maximaal slechts 1.2 ampere kan leveren. Een TMC5160 levert dan weer continue 3 ampere met pieken tot 32 ampere. Kies dus welke driver bij je type motor past en geschikt is voor je toepassing.

RTC ANTWERPEN

Servo Motor

Precieze controle van positie, snelheid en kracht: Een servomotor kan nauwkeurig worden aangestuurd om een specifieke positie te bereiken, wat essentieel is voor toepassingen zoals robots, CNC-machines

Gebruik van een feedbacksysteem: Een servomotor bevat een encoder of potentiometer die de actuele positie meet en terugkoppelt naar de besturing. Dit maakt continue aanpassingen van de positie mogelijk met een hoge nauwkeurigheid ook na spanningsuitval wat een belangrijk verschil is met een stepper motor.

Compact en krachtig: Ondanks hun kleine formaat leveren servomotoren toch een redelijk koppel en prestaties, wat ze ideaal maakt voor situaties waar ruimte beperkt is.

Werkt met een servocontroller: De motor wordt aangestuurd door een servocontroller die ingesteld wordt door een microcontroller. Voor kleine motoren kan de microcontroller de servomotor rechtstreeks aansturen via een PWM signaal.

Breed toepassingsgebied: Servomotoren worden gebruikt in industriële automatisering, robotica, medische apparatuur, modelbouw (zoals RC-auto’s en -vliegtuigen). Voor onze toepassing is het gebruik voor de stuurinrichting of in mindere maten de grijper een aangewezen.

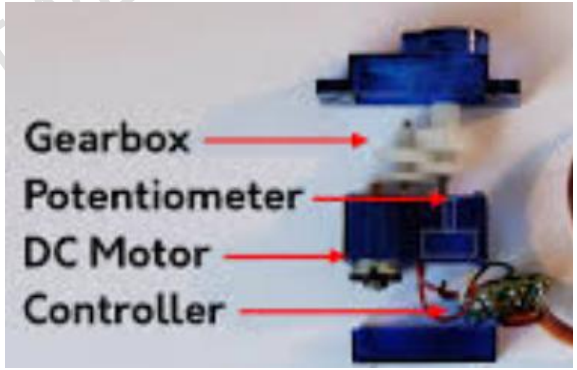
RTC ANTWERPEN

Servo Motor feedbacksysteem

In tegenstelling tot een stappenmotor moet een servomotor zijn positie onthouden na een spanningsonderbreking. Het gebruik van een eindeloopschakelaar is hiervoor dus niet geschikt. Een mogelijke oplossing is het gebruik van een potentiometer, waarvan de weerstandswaarde afhankelijk is van de hoekverdraaiing, of een optisch systeem.

In ons voorbeeld maken we gebruik van een ingebouwde potentiometer.

- goedkoop
- lineaire meting
- snel real-time sturing
- uitlezen van analoge spanning : eenvoud
- levensduur potentiometer kan "kraken"
- beperkt tot minder dan 360° meestal 270° maximum
- temperatuursafhankelijk (stof, vocht)
- nauwkeurigheid beperkt



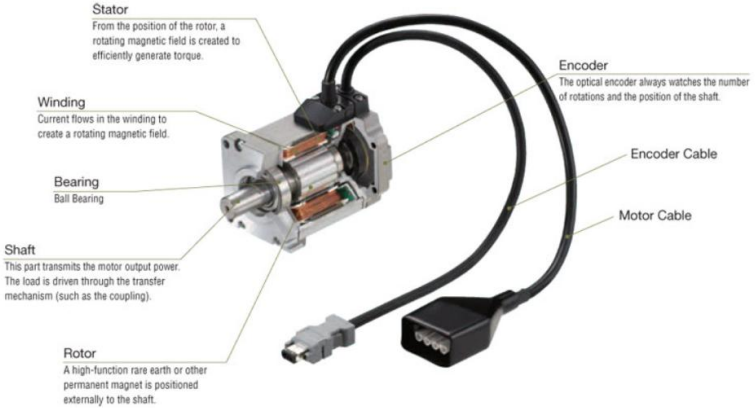
RTC ANTWERPEN

Servo Motor feedbacksysteem

Een tweede mogelijkheid is het gebruik van een optische encoder.

In ons voorbeeld maken we gebruik van een ingebouwde potentiometer.

- nauwkeuriger
- precies
- oneindige rotatie (niet beperkt tot 270°)
- hoge snelheid mogelijk
- geen ruis, drift temperatuursafhankelijkheid
- gevoelig voor vuil en stof
- optische meetsysteem
- verbruikt meer energie dan een potentiometer meting



RTC ANTWERPEN

Servo Motor tandwieloverbrenging

- Verhoogd koppel, een servomotor heeft vaak een hoog koppel maar een lage snelheid, waardoor de motor zwaardere lasten kan bewegen zoals kleppen en ventielen.
- Nauwkeurige positionering
- Zonder overbrenging zou de motor meer stroom nodig hebben om een last te verplaatsen.
- Een kleine motor kan hierdoor evenveel kracht leveren als een grotere motor zonder overbrenging.
- Beperking van inertie en trillingen een tandwieloverbrenging helpt om plotselinge bewegingen te dempen, waardoor de motor soepeler en stabieler werkt.

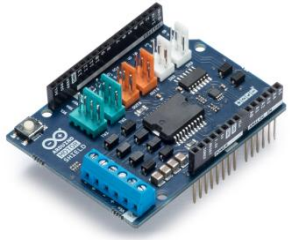
RTC ANTWERPEN

DC Motor

Definitie: Een DC-motor (gelijkstroommotor) is een elektrische motor die gelijkstroom (Direct Current, DC) omzet in mechanische energie.

Werking van een DC-motor:

- Rotor (anker): Het bewegende deel waarin een elektrisch magnetisch veld opwekt wordt.
- Stator: Het stationaire deel met permanente magneten.
- Commutator: Zorgt ervoor dat de stroom in de spoelen van de rotor omkeert, wat continu draaien mogelijk maakt.



Toepassingen van een DC-motor: Wordt vaak gebruikt in toepassingen die een regelbare snelheid en hoog koppel vereisen, zoals: elektrische voertuigen, kleine huishoudelijke apparaten. speelgoed en robotica.

Voordelen van een DC-motor: Eenvoudige snelheidsregeling door de spanning te variëren. Hoger startkoppel, ideaal voor toepassingen met zware belastingen bij het starten.

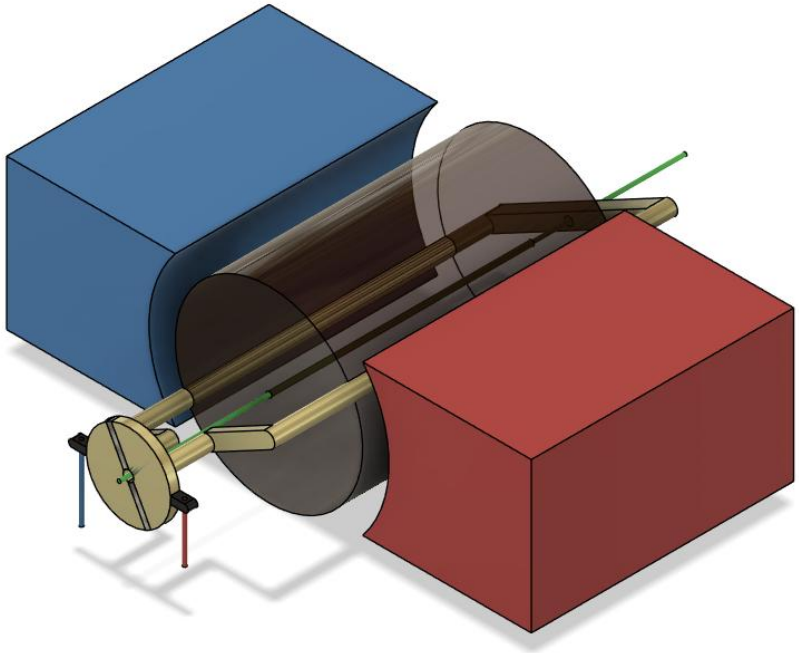
Soorten DC-motoren: Er zijn verschillende typen, afhankelijk van hoe het magnetisch veld wordt opgewekt:

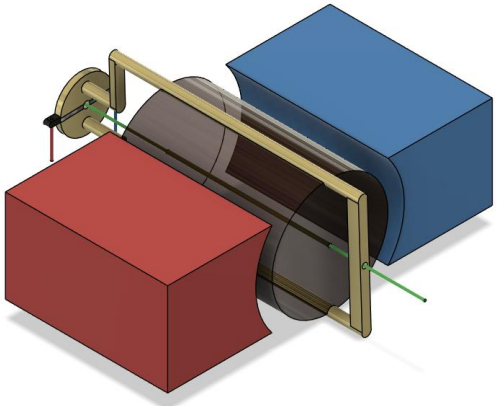
- Serie-geschakelde DC-motor: hoge kracht bij lage snelheden.
- Parallel-geschakelde DC-motor: constante snelheid, ongeacht belasting.
- Permanent-magneet DC-motor: compact en efficiënt, vaak gebruikt in kleine apparaten.

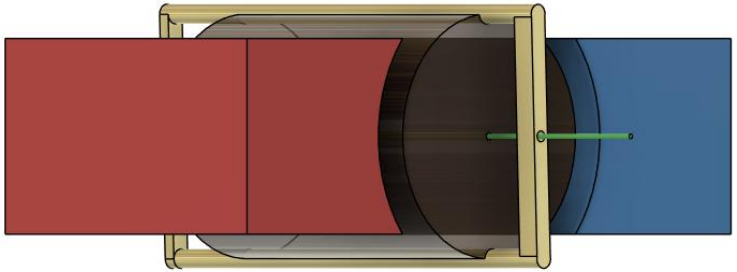
RTC ANTWERPEN

DC Motor

- noord- en zuidpool (stator)
- Wire (rotor)
- Borstels met commutator
- As
- Geleiders + en -







Voordelen en nadelen DC motor permanente magneet met koolstofborstels.

Voordelen

- **Eenvoudige constructie en betrouwbaarheid**: De motor heeft geen externe veldwikkelingen nodig, omdat de permanente magneet in de stator het magnetisch veld genereert. Dit maakt de motor eenvoudig en betrouwbaar in gebruik, vooral voor basistoepassingen.
- **Kleine omvang**: Dankzij het gebruik van permanente magneten zijn deze motoren meestal compact en licht van gewicht.
- **Energiezuinig statorgedeelte**: Het statorgedeelte verbruikt geen energie, wat bijdraagt aan de efficiëntie.
- **Geschikt voor basistoepassingen**: Deze motoren worden vaak gebruikt in toepassingen zoals speelgoed, pompen en kleine apparaten.
- **Gemakkelijk te regelen**: De snelheid en draairichting van de motor kunnen eenvoudig worden aangepast door de aangelegde spanning en polariteit te variëren.
- **Kosteneffectief**: Door de eenvoudige constructie en het gebruik van permanente magneten zijn de productiekosten laag.

Nadelen

- **Beperkt koppel en vermogen**: Het maximale koppel en vermogen worden beperkt door de sterkte van de permanente magneten. Hierdoor zijn deze motoren minder geschikt voor toepassingen met een hoge belasting.
- **Slijtage van borstels en commutator**: Dit leidt tot een kortere levensduur en vereist meer onderhoud.
- **Beperkte snelheid**: Bij hoge snelheden treedt slijtage sneller op, en door de commutatie kunnen vonken ontstaan, wat de prestaties beperkt.
- **Temperatuurgevoeligheid van permanente magneten**: Bij hoge temperaturen kunnen de magneten hun magnetische eigenschappen verliezen, wat kan leiden tot prestatieverlies.
- **Lagere efficiëntie**: Door wrijvingsverliezen in de borstels en de commutator is de efficiëntie lager in vergelijking met borstelloze motoren.
- **Minder nauwkeurig bij lage snelheden**: Het regelen van de motor op lage snelheden kan lastig zijn.

KTS ANTWERPEN

Samengevat DC motor

Een brushed DC-motor met een permanente magneet in de stator is een goede keuze voor eenvoudige, kosteneffectieve toepassingen met lage tot middelmatige eisen aan koppel en snelheid. Echter, voor toepassingen die hogere efficiëntie, langere levensduur of hogere prestaties vereisen, kunnen alternatieven zoals borstelloze DC-motoren (BLDC), motoren met elektromagnetische stators of stepper motors geschikter zijn.

RTC ANTWERPEN

DC Motor Arduino

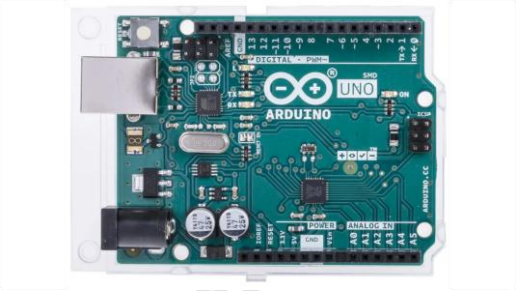
Arduino UNO: Deze microcontroller is erg populair, vooral in scholen en STEM-richtingen.

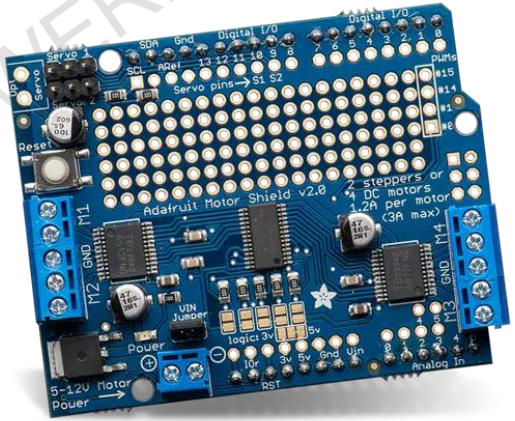
Joystick: Wordt gebruikt voor de bediening van de motor, maar kan desgewenst worden vervangen door WiFi- of Bluetooth-besturing.

Adafruit Motor Shield V2: Wordt gebruikt voor het aansturen van elektromotoren. 4 DC motoren of 2 stepper motoren. I2C controle interface (32 mogelijke adressen, stacked)

12 Volt voeding: Een gezamenlijke voeding voor zowel de Arduino als de controller van de steppermotor.

DC-Motor 6V - Type 140 (met driverboard): Een populaire mini DC motor





RTC ANTWERPEN

DC Motor Arduino Motor Shield V2

RTC Antwerpen heeft in de afgelopen schooljaren deze module beschikbaar gesteld binnen de BITL-projecten voor het aansturen van motoren. Inmiddels is er ook de Arduino Motor Shield Rev3. Dit is echter geen opvolger van het Motor Shield V2, maar eerder een beperkte versie!

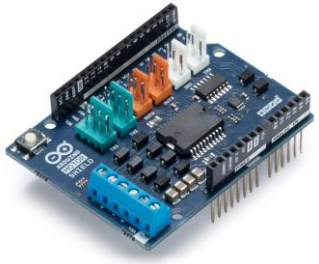
Arduino Motor Shield Rev3:

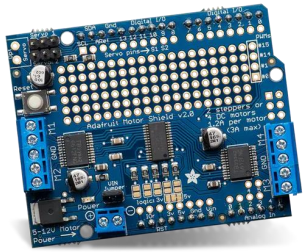
- Geschikt voor eenvoudige toepassingen met weinig motoren.
- Ideaal als je een snel en eenvoudig te begrijpen setup wilt zonder afhankelijkheid van I2C.

Adafruit Motor Shield V2:

- Geschikt voor complexere projecten met meerdere motoren.
- Efficiënter, uitbreidbaar en biedt meer precisie bij het aansturen van motoren.

*Door het gebruik van I2C ben je **onafhankelijk** van de Arduino UNO tijdens de uitvoering van het programma, wat een belangrijk voordeel is. De controller functioneert volledig autonoom.*



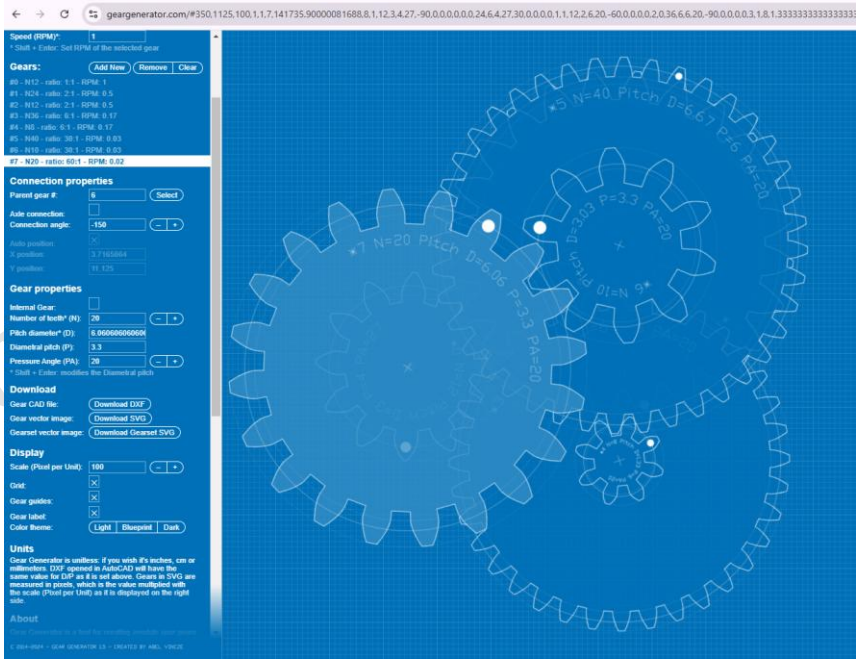


Kenmerk	Arduino Motor Shield Rev3	Adafruit Motor Shield V2
Protocol	PWM (directe pinbesturing)	I2C
Aantal motoren	2 DC of 1 stepper	4 DC of 2 steppers
Stroomcapaciteit	2A per kanaal	1.2A per kanaal
Efficiëntie	Lager (L298P H-brug)	Hoger (TB6612 MOSFET-drivers)
Voeding	Gedeeld of extern	Gescheiden extern
Software	Geen specifieke bibliotheek	Adafruit-bibliotheek
Uitbreidbaarheid	Niet uitbreidbaar	Tot 32 shields via I2C

RTC ANTWERP

Gear Generator webbased ontwerp van tandwielen

- Opstelling getrouw ingeven
- Ingeven tandwiel reducties
- Eventueel aanpassen van diametrical pitch
- Nakijken van as locaties
- Controleer de werking door aanpassen van animatie snelheid



RTC ANT

Vormgeving tanden.

De tanden van twee gekoppelde tandwielen moeten identiek zijn. Ze worden zodanig ontworpen dat ze perfect in elkaar grijpen en met minimale weerstand weer loskomen. De nauwkeurigheid waarmee de tanden in elkaar passen, wordt de **drukhoek** (*pressure angle*) genoemd. Wanneer deze wordt aangepast, verandert de vorm van de tanden.

Een hogere drukhoek resulteert in scherpere tanden, waardoor de tandwielen soepeler draaien. Echter, dit leidt ook tot snellere slijtage. Wanneer de slijtage te hoog wordt, kunnen de tandwielen los van elkaar gaan bewegen.

PTC ANTWE

Aantal tanden verdeeld op de omtrek

Voorbeeld verhouding tanden A / B = 12 / 36

Dan verkrijg ik een diameter van 2.4 op 7,2 … ook eenzelfde verhouding.

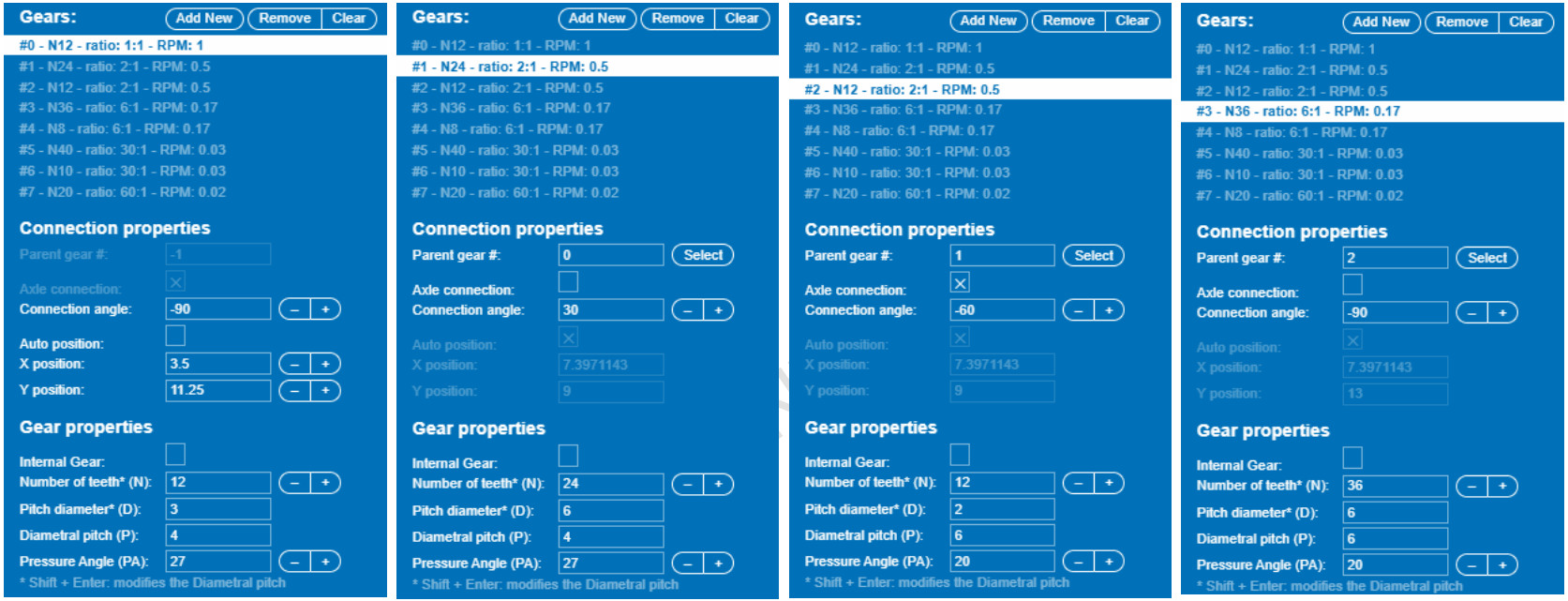
De diametral pitch bepaald het aantal tanden per inch gemeten op de diametral cirkel, de cirkel waarop de tanden ingrijpen.

De diametral pitch voor aansluitende tanden is steeds gelijk.

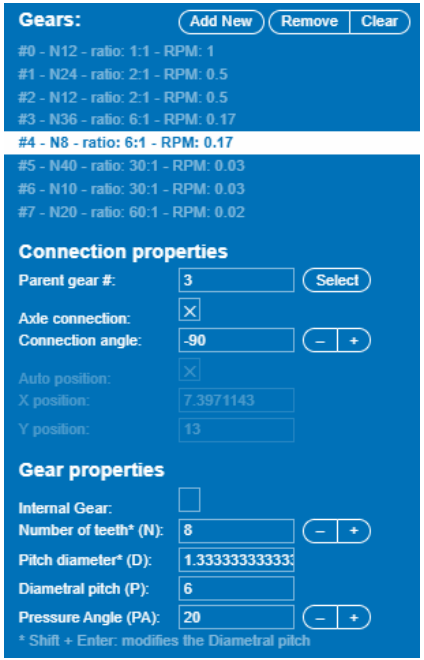
Voor onze toepassing maakt dat niet uit we scalen de tandwielen toch verder in Fusion 360 tot we de gewenste afmetingen bekomen.

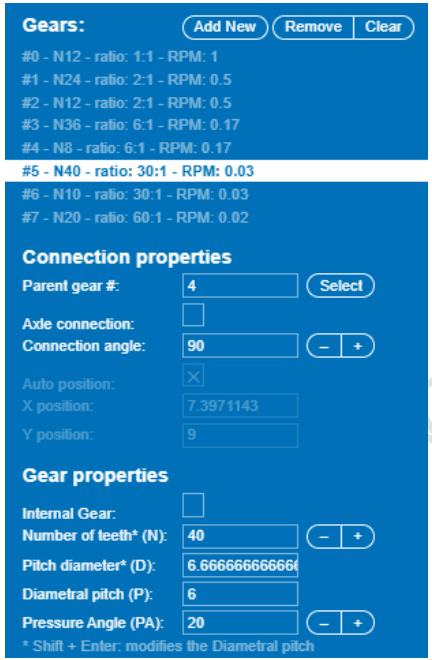
RTC ANTWE

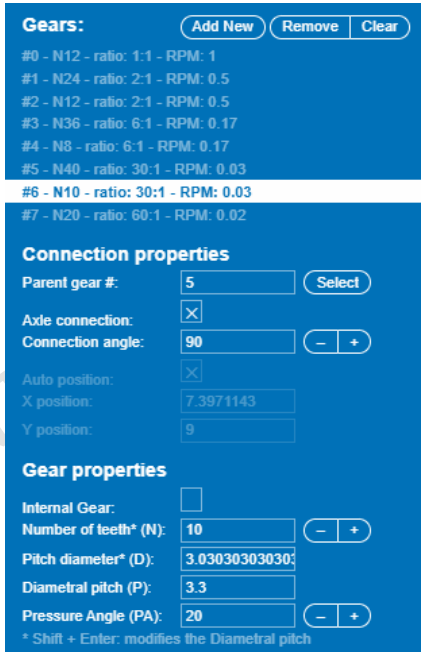
Noteer de vier belangrijkste parameters.

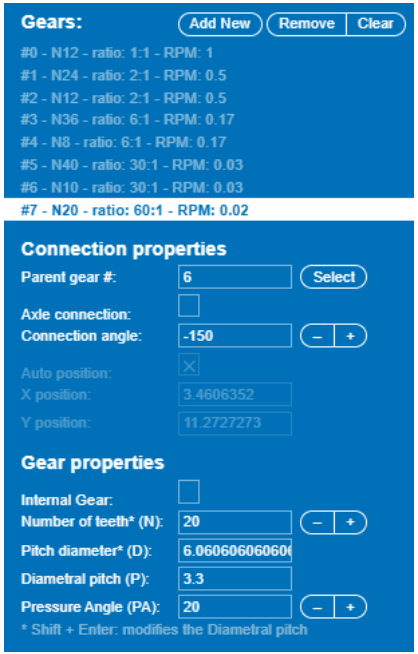


Noteer de vier belangrijkste parameters.









Tandwielberekening voor een klok

We moeten een reductie van 60 realiseren, aangezien er 60 minuten in één uur zitten. De stepper maakt één omwenteling per minuut, wat overeenkomt met **200 stappen per minuut**, oftewel een **vertraging van 3,33... ms per stap**. Door een extra reductie toe te passen, kunnen we afrondingsfouten vermijden. Een mogelijke reductie is **200 / 180**.

- Het tandwiel met **180 tanden** roteert **9 keer per minuut**.
- Het tandwiel met **200 tanden** roteert dan met een snelheid van: 200/180 × 9 = 10 omwentelingen per minuut
- Hierdoor wordt het **aantal stappen per minuut 2000**, wat neerkomt op **0,1 seconde per stap**.
- Om 60 minuten te tellen tot één uur is een tandwielreductie van **1/60** nodig, maar dit is moeilijk te realiseren. We kunnen deze reductie als volgt herschrijven: 1 / 60=1 / (2×3×5×2) Als we de draairichting willen behouden, moet het aantal overbrengingen **even** zijn. De combinatie **2-2-3-5** ligt het dichtst bij elkaar. Bij de combinatie **3-4-5 oneven** keert de draairichting om, wat we niet willen.
- **Een tandwiel met 12 tanden en een tandwiel met 24 tanden** geven een reductie van **2**.
- **Een tandwiel met 8 tanden en een tandwiel met 40 tanden** geven een reductie van **5**.

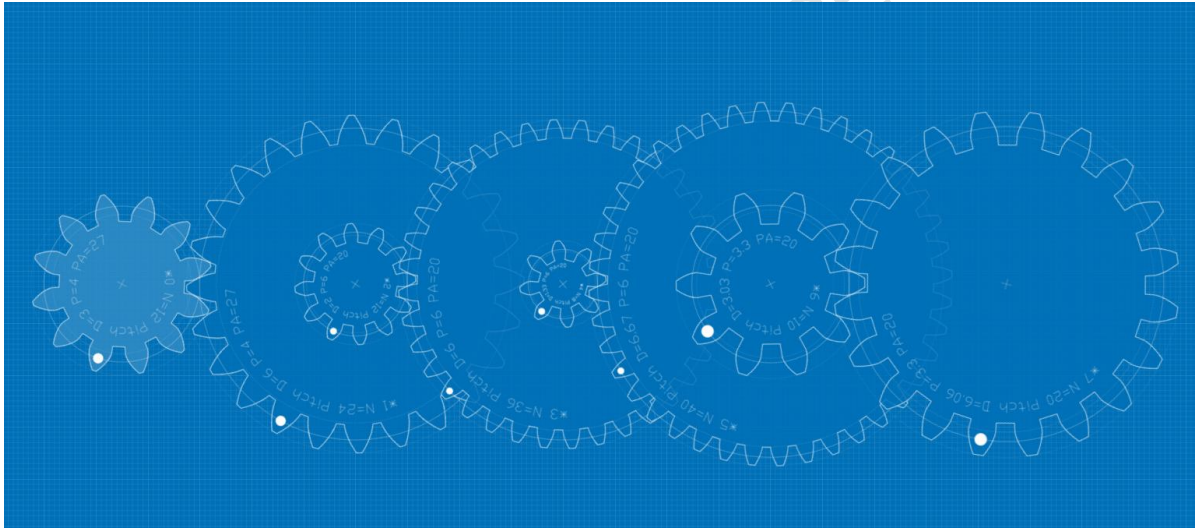
TWEEPEN

Gear	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	
Aantal tanden N	12	24	12	36	8	40	10	20	
Pitch Diameter D	3	6	2	6	1,3333	6,6666	3,0303	6,0606	
Diametrical Pitch P	**4**	**4**	**6**	**6**	**6**	**6**	**3,3**	**3,3**	
Pressure Angle PA	27	27	20	20	20	20	20	20	
Reductie		2		3		5		2	**60**

Plaats de tandwielen horizontaal

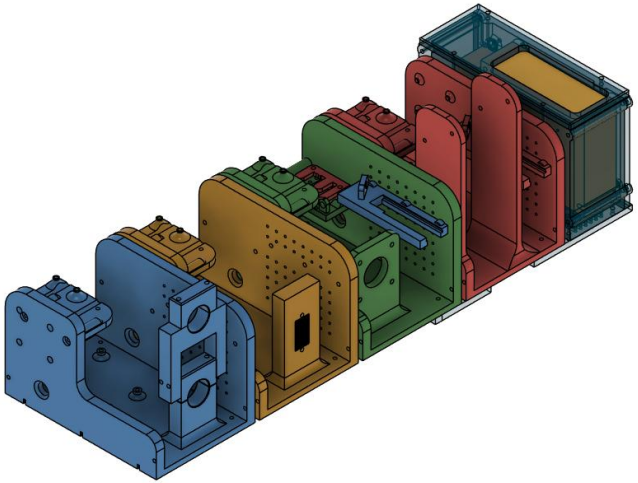
Hoek op 0 graden plaatsen en nakijken van as locaties!

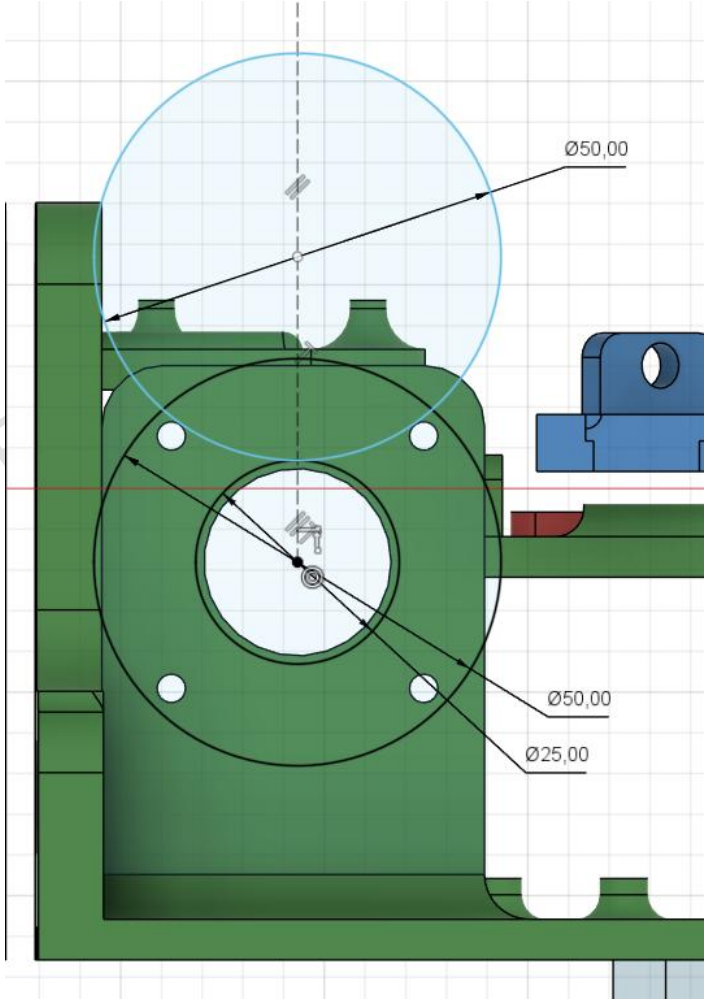
Download GearSet SVG … inlezen in Fusion 360.



Fusion 360

Teken de stepper motor bevestiging, assen …
asafstand van 75 / 2 = **37.5**





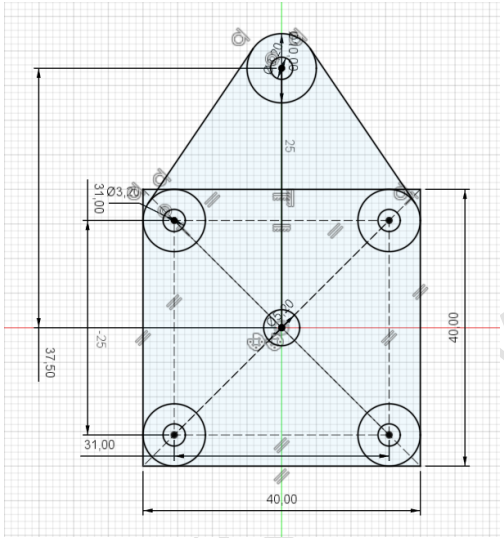
iC ANTWERP

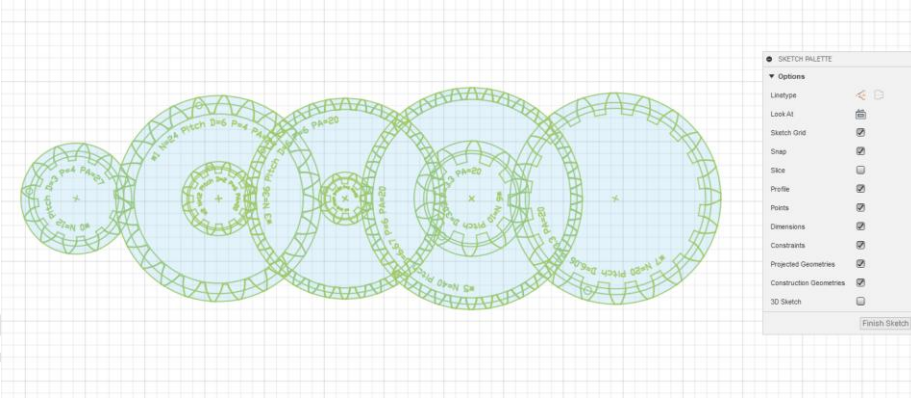
Voorbeeld gear reductie "uurwerk"

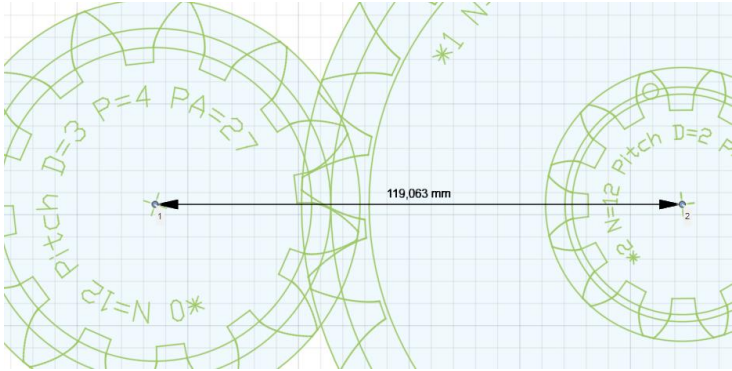
As-afstand gewenst is 37,5 mm. As-afstand gemeten tandwiel 0 en 1 is … 119,063 mm

Schaal factor

- ***0 1 = 37,5 / 119,063***
- ***2 3 = 37,5 / 105,833***
- ***4 5 = 37,5 / 105,833***
- ***6 7 = 37,5 /120,265***

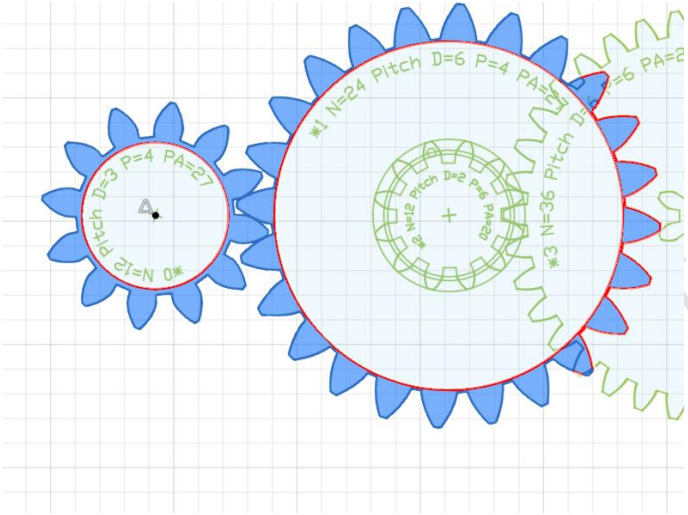


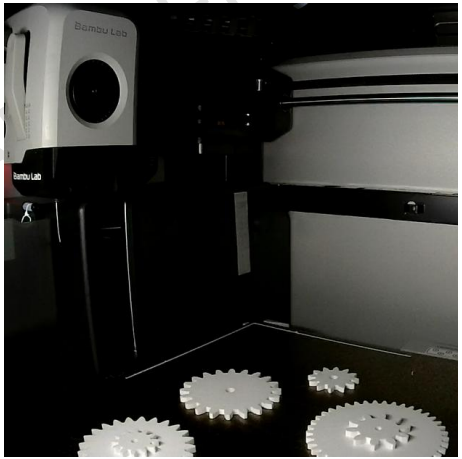


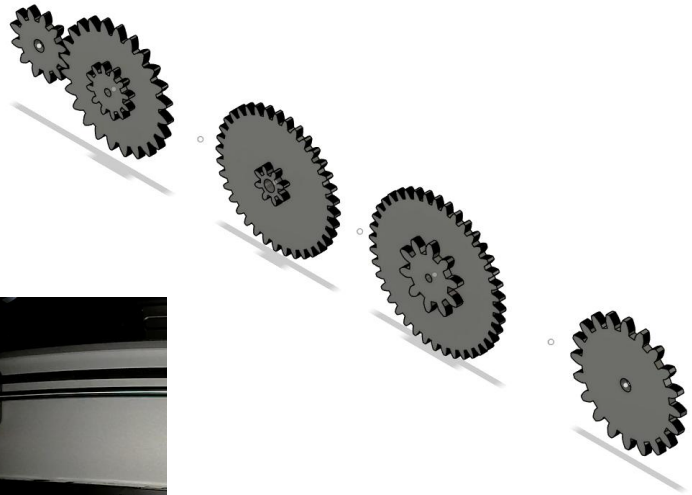


Selecteer en expandeer vanuit de tekening.

Pas tekening aan, selecteer tandwielen per paar en scaling per paar. Tandwielen verplaatsen tot ze exact passen.

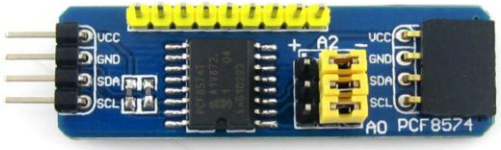






Software dem = PCF8574

IO expander met interrupt.




```
#include <Adafruit_PCF8574.h>

Adafruit_PCF8574 pcf;

void setup() {
  while (!Serial) { delay(10); }
  Serial.begin(115200);
  Serial.println("Adafruit PCF8574 LED blink test");
  if (!pcf.begin(0x20, &Wire)) {
    Serial.println("Couldn't find PCF8574");
    while (1);
  }
  for (uint8_t p=0; p<8; p++) {
    pcf.pinMode(p, OUTPUT);
  }
}

void loop() {
  for (uint8_t p=0; p<8; p++) {
    pcf.digitalWrite(p, LOW);
    delay(1000);
    pcf.digitalWrite(p, HIGH);
  }
}
```

DC motor Arduino

```
#include <Wire.h>
#include <Adafruit_MotorShield.h>

Adafruit_MotorShield AFMS = Adafruit_MotorShield();

Adafruit_DCMotor *DCMotor3 = AFMS.getMotor(3);
Adafruit_DCMotor *DCMotor4 = AFMS.getMotor(4);

bool debugFlag = true;

int xAValue = 0; // gemeten X waarde joystick A
int yAValue = 0; // gemeten Y waarde joystick A
#define VRXjoyA A0 // A0 verbonden met analog out joystick
#define VRYjoyA A1 // A1 verbonden met analog out joystick
#define VRxACenter 498 // center waarde voor joystick A in de X waarde naar rechts
#define VRxANoise 20 // noise waarde voor joystick A in de X waarde naar rechts
#define VRxALeftMax 1023 // maximale waarde voor joystick A in de X waarde naar rechts
#define VRyACenter 502 // center waarde voor joystick A in de Y waarde naar rechts
#define VRyANoise 20 // noise waarde voor joystick A in de Y waarde naar rechts
#define VRyALeftMax 1023 // maximale waarde voor joystick A in de Y waarde naar rechts

int xMinSpeed = 0;
int xMaxSpeed = 9;

int yMinSpeed = 0;
int yMaxSpeed = 100;

```

We maken gebruik van een Arduino Motorshield V2 met library *Adafruit_MotorShield.h* de library is standaard aanwezig in de IDE.

De Wire library wordt gebruikt voor de I2C communicatie tussen de arduino en het shield.

Het object **AFMS** van de klasse Adafruit_MotorShield fungeert als een interface tussen je Arduino en het Adafruit Motor Shield. Hiermee kun je motoren configureren en aansturen op een eenvoudige manier.

Adafruit_MotorShield() is de constructor.

Er wordt een pointer (*) aangemaakt die verwijst naar een object van de klasse Adafruit_DCMotor.

De maximum speed mag op 255 geplaatst worden. We zullen zien dat de waarde 100 al ruimschoots voldoende is.

DC motor Arduino

```
void setup() {
  Serial.begin(115200);
  AFMS.begin();

  DCMotor3->setSpeed(150);
  DCMotor3->run(FORWARD);
  DCMotor3->run(RELEASE);

  DCMotor4->setSpeed(150);
  DCMotor4->run(FORWARD);
  DCMotor4->run(RELEASE);
}
```

Enkel voor een initiële setup maar heeft geen verdere functionaliteit.

We gebruiken eventueel de serial line voor debuggen.

RTC ANTWERPEN

DC motor Arduino

```
42  void loop() {
43    uint8_t i;
44
45    xAValue = analogRead(VRXjoyA);
46    yAValue = analogRead(VRYjoyA);
47
48    if (xAValue < (VRxACenter - VRxANoise)) {
49        DCMotor3->run(BACKWARD);
50        if (debugFlag) {Serial.println("xAValue 1 = " + String(xAValue));}
51        int SpeedValue = map(xAValue, VRxACenter - VRxANoise, 0, xMinSpeed, xMaxSpeed);
52        DCMotor3->setSpeed(SpeedValue);
53    }
54    else if (xAValue > (VRxACenter + VRxANoise)) {
55        DCMotor3->run(FORWARD);
56        if (debugFlag) {Serial.println("xAValue 2 = " + String(xAValue));}
57        int SpeedValue = map(xAValue, VRxACenter + VRxANoise, VRxALeftMax, xMinSpeed, xMaxSpeed);
58        DCMotor3->setSpeed(SpeedValue);
59    }
60    else {
61        DCMotor3->run(RELEASE);
62    }
63
64    if (yAValue < (VRyACenter - VRyANoise)) {
65        DCMotor4->run(BACKWARD);
66        int SpeedValue = map(yAValue, VRyACenter - VRyANoise, 0, yMinSpeed, yMaxSpeed);
67        DCMotor4->setSpeed(SpeedValue);
68    }
69    else if (yAValue > (VRyACenter + VRyANoise)) {
70        DCMotor4->run(FORWARD);
71        int SpeedValue = map(yAValue, VRyACenter + VRyANoise, VRyALeftMax, yMinSpeed, yMaxSpeed);
72        DCMotor4->setSpeed(SpeedValue);
73    }
74    else {
75        DCMotor4->run(RELEASE);
76    }
77  }
78
```

Alle code is wel duidelijk. De `map()` functie in Arduino wordt gebruikt om een getal van één bereik naar een ander bereik te herberekenen (of "hermapping"). Het is handig wanneer je bijvoorbeeld een sensorwaarde wilt omzetten naar een schaal die beter past bij je toepassing, zoals een potentiometerwaarde omzetten naar een motorschaal.

map(**value, fromLow, fromHigh, toLow, toHigh**)

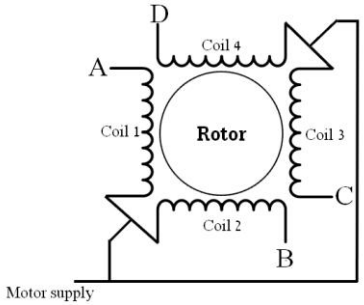
Parameters

1. `value`: Het getal dat je wilt herberekenen.
2. `fromLow`: De ondergrens van het oorspronkelijke bereik.
3. `fromHigh`: De bovengrens van het oorspronkelijke bereik.
4. `toLow`: De ondergrens van het nieuwe bereik.
5. `toHigh`: De bovengrens van het nieuwe bereik.

WERPEN

Unipolaire Stepper Motor Arduino

```
void stepper(int xw) {
  for (int x = 0; x < xw; x++) {
    delay(1);
    switch (Steps) {
      case 0:
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, HIGH);
        break;
      case 1:
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, HIGH);
        digitalWrite(IN4, HIGH);
        break;
      case 2:
```



Step	A	B	C	D
0	*0*	*0*	*0*	*0*
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	0	0
6	1	1	0	0
7	1	0	0	0
8	1	0	0	1
0	*0*	*0*	*0*	*0*

RTC ANTWERPEN

Stepper met Joystick gebruik microstepping 1 min per rotatie met tandwieloverbrenging.

```
void loop() {
// Lezen van de joysticks
  xAValue = analogRead(VRXjoyA);
  if (debugFlag) {Serial.println("xAValue = " + String(xAValue));


  if (xAValue > 600){
    OrientxA = 1;
    pcfE.digitalWrite(1, LOW); // M0
    pcfE.digitalWrite(2, LOW); // M1
    pcfE.digitalWrite(3, LOW); // M2
    Stepper_Enable(true);
    NumberOfStepsM1(1, OrientxA);
    Stepper_Enable(false);
```

```
// Subroutines
String NumberOfStepsM1(int Steps, int RNotL) {
  if ((RNotL == 1) and(OrientPreviousxA != 1)) {
    pcfE.digitalWrite(0, LOW); // rechts
    OrientPreviousxA = 1;
  }
  if ((RNotL == 2) and(OrientPreviousxA != 2)) {
    pcfE.digitalWrite(0, HIGH); // links
    OrientPreviousxA = 2;
  }
  if ((RNotL == 1) or(RNotL == 2)) {
    digitalWrite(stepPinM1, HIGH);
    delay(stepDelay);
    digitalWrite(stepPinM1, LOW);
    delay(stepDelay);
  }
  int increment = round(pow(2, (5 - MicroSteppingValueM1)));
  if (RNotL == 1) {
    positionM1 = positionM1 + increment;
    return (" " + String(Steps) + " Steps Rechts");
  }
```