

Handleiding aansturen van motoren met Arduino

WB2633 Project Advanced Engineering Design

Anton van Beek
TU Delft | Mechanical Engineering

Freek Broeren

Gepubliceerd op 2025-07-31
2026-07-28

Laatst bijgewerkt op

Inhoudsopgave

1 Inleiding	2
2 Useful links	2
3 Handleiding hardware	2
3.1 Stapgrootte Instellen	3
4 G-code naar de Arduino sturen	6
4.1 UGS Downloaden en installeren	7
4.2 UGS verbinden met de Arduino	7
4.3 Configuratie G-code	8
5 G-Code Programmeer handleiding	9
6 G-Code inlezen in UGS en Nulpunt vastleggen	12
7 Handleiding Solenoid en Electromagneet	13
8 Handleiding 775 DC motor	15
9 Installatie Arduino IDE App	16
9.1 Libraries uploaden	17
9.2 Sketch compileren en uploaden:	18
9.3 Motoren gaan draaien:	18
10 Installatie GRBL library / uploaden	19
11 Bijlage 1: Arduino programma voor stappenmotoren	20

1 Inleiding

In het AED project maken we gebruik van een Arduino om stappenmotoren en andere electronica aan te sturen. De Arduino is een programmeerbaar bordje wat je veel flexibiliteit geeft in hoe je wilt dat jouw machine gaat bewegen.

De Arduino's die we in dit project gebruiken, zijn standaard voorzien van GRBL-firmware. Hierdoor kunnen ze aangestuurd worden door middel van G-code, wat het makkelijk maakt om de stappenmotoren aan te sturen in xyz-coördinaten.

In deze handleiding beschrijven we eerst hoe de Arduino die je van ons ontvangt eruit ziet, hoe je controleert of die compleet is en wat er aan de hardware in te stellen is. Daarna beschrijven we hoe je G-code naar de arduino kunt sturen om daarmee stappenmotoren, solenoids en een electromagneet aan te sturen. Tenslotte zullen we ook beschrijven hoe je direct code naar de arduino kunt sturen met Arduino IDE. Dit is nuttig als je zelf de code voor je machine wilt schrijven of wanneer er toch iets mis blijkt te zijn met de voorgeïnstalleerde firmware.

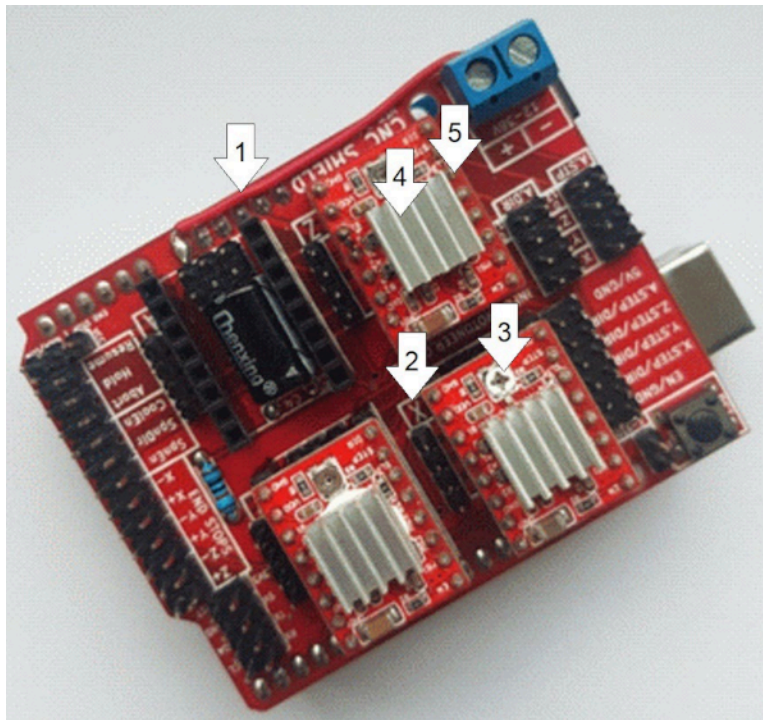
2 Useful links

Voor het schrijven van deze handleiding is gebruik gemaakt van onderstaande links.

- How to Setup GRBL & Control CNC Machine with Arduino <https://howtomechatronics.com/tutorials/how-to-setup-grbl-control-cnc-machine-with-arduino/>
- List of Most Important G-code Commands: <https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/>

3 Handleiding hardware

Arduino met CNC shield en stappenmotor drivers: Begin met het uitpakken van de Arduino UNO met daarop het CNC shield gemonteerd (rode printboard). Deze is getoond in Figuur 1. Controleer de aanwezigheid van de verbindingsdraad (1) die er voor zorgt dat de CNC shield spanning krijgt via de Arduino aansluiting.



Figuur 1: CNC shield met drie drivers. Elke driver heeft een koellichaam. 1. Rode draad 2. Op het CNC shield naast de drivers zijn de X, Y en Z poorten aangegeven 3. De drivers zijn met de kleine instelbare potmeter gericht naar de rode draad. 4. koellichaam op Driver maakt bijna contact met pinnen (5) wat kan resulteren in kortsluiting.

Op het CNC shield zijn drie drivers gemonteerd. Dit zijn de kleine rode printplaatjes met daarop een koellichaam. Hiermee kunnen de drie stappenmotoren individueel worden aangestuurd.

! Voorkom kortsluiting

Leg de Arduino board niet op een elektrisch geleidende ondergrond.

Controleer of de drie koellichamen op de stappenmotordrivers niet zodanig verschoven zijn dat deze contact maken met soldeerpunten in de omtrek.

Zorg bij demontage van componenten dat een eventueel aangesloten voeding is uitgeschakeld.

3.1 Stapgrootte Instellen

Stappenmotoren maken standaard vaak 200 stappen per omwenteling. De minimale stapgrootte is dan $\frac{360}{200} = 1,8$ graden per stap. Maar het is ook mogelijk om deze motoren aan te sturen met bijvoorbeeld $1/16^e$ van een stap waar-

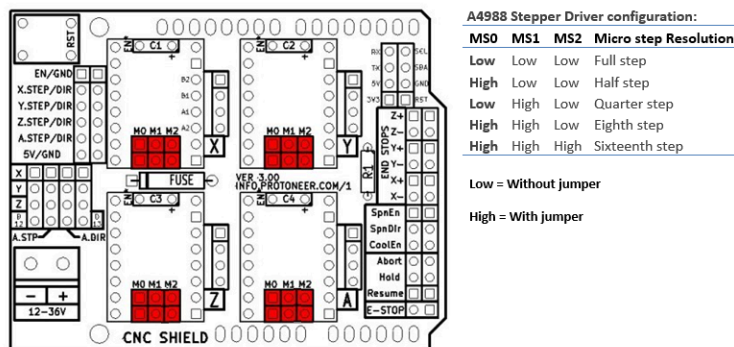
door de hoekverdraaiing $1,8/16$ graden per stap bedraagt. Dit noemen we microstepping.

Met een kleinere stapgrootte is zowel de resolutie als ook het maximaal motorkoppel hoger maar de maximale rotatiesnelheid lager. De kleinste stapgrootte (dit is de resolutie) die de stappenmotor kan maken wordt hardwarematig ingesteld door zogenaamde jumpers te plaatsen.

3.1.a Jumpers

Onder de drivers (kleine rode printplaatjes met koellichaam) kunnen jumpers worden gemonteerd. Dit zijn kleine blokjes met daarin een stukje geleidend materiaal. Hiermee kun je twee naastgelegen pinnen met elkaar verbinden.

Met de jumpers kunnen de stappenmotoren hardwarematig worden ingesteld op hele stappen van $\frac{360}{200} = 1,8$ graden per stap, maar het is ook mogelijk om te kiezen voor $1/2$, $1/4$, $1/8$ en $1/16$ stappen. Met halve stappen kan de hoekverdraaiing met kleinere stapjes worden ingesteld (hogere resolutie) waarbij het te leveren motorkoppel verdubbeld en de rotatiesnelheid halveert. In Figuur 2 hieronder zijn de posities waarop de jumpers ingesteld kunnen worden in het rood aangegeven met M0, M1 en M2.



Figuur 2: CNC shield zonder drivers. Per poort drie jumpers M0, M1 en M2

Zonder jumpers staat de motor ingesteld op Full step (default). Met alle jumpers gemonteerd staan de motoren ingesteld op $1/16$ stapgrootte. De jumpers (15 stuks) bevinden zich in een plastic zakje in de componenten box. Voor $1/16$ stapgrootte worden per motor 3 jumpers rechttop gemonteerd.

3.1.b Jumpers monteren

Om de jumpers te monteren moeten de drivers tijdelijk worden gedemonteerd. Dit doe je door ze voorzichtig vanaf de CNC shield naar boven te trekken. Begin met het uitnemen van de driver die het meest toegankelijk is.

Nu kun je de gewenste jumpers plaatsen volgens Figuur 2. Je jumpers zijn kleine blauwe blokjes die verbinding kunnen maken tussen twee naastgelegen pinnen.

De 6 jumpers die over zijn gaan terug in het zakje in de box. Het CNC shield wordt terug geplaatst op de Arduino. Met 1/16 stapgrootte staan de motoren nu ingesteld op $200 \cdot 16 = 3200$ stappen per omwenteling met een zo hoog mogelijk te leveren koppel. Met de software wordt straks ingesteld dat 3200 stappen nodig zijn voor 1 omwenteling. Als de motor is verbonden met een spindel met een spoed van $P = 1,5 \text{ mm}$ dan is de resolutie $1.5\text{mm}/3200$ is afgerond $0.5\text{mm}/\text{stap}$.

Feed forward: Doordat bij stappenmotoren het aantal stappen wordt ingesteld is de hoekverdraaiing van de rotor elk moment nauwkeurig bekend (feed forward). Bij een servomotor is dit anders, daarbij wordt de hoekverdraaiing met een sensor gemeten en teruggekoppeld (feed back).

Inschakelverschijnsel: Bij het inschakelen van een stappenmotor zal de rotor van de motor direct naar de dichtstbijzijnde voorkeurspositie verdraaien. Vanuit deze startpositie kan de hoekverdraaiing met grote nauwkeurigheid worden gekozen door het aantal stappen op te geven.

Houdkoppel: Het houdkoppel van een stappenmotor is met de motor gespecificeerd (0.4 Nm bij de Nema 17). Dit is het koppel waarmee de motoras belast kan worden zonder dat deze zal verdraaien.

Motorkarakteristiek: Stappenmotoren hebben een motorkarakteristiek die afhankelijk is van diverse factoren waaronder de processorsnelheid van de Arduino en de stapgrootte die is ingesteld. Wel is er vaak een houdkoppel bij full step en toerental nul gespecificeerd. Het maximum motorkoppel en toerental bij bedrijf kan experimenteel worden vastgesteld door vast te stellen wanneer slip optreedt.

Slip is het overslaan van stappen. De verplaatsing komt dan niet meer overeen met het opgegeven aantal stappen dat nodig is om de verplaatsing te realiseren.

Aansluitdraden tussen stappenmotoren en CNC shield: Neem de flat cables uit de box. Er zijn twee flat cables met gelijke lengte en mogelijk eentje die wat korter is. De flat cables zijn wat aansluitingen betreft alle drie hetzelfde. De kortere flat cable is voor een motor die straks in de opstelling dichterbij de Arduino wordt gemonteerd. De witte connectoren van de flat cables passen op 1 manier in de stappenmotoren. Plug deze in de motoren. De andere zwarte connectoren kunnen op twee manieren op de CNC shield gemonteerd worden, rechts naast de driver. Als na het monteren blijkt dat een motor de verkeerde kant op draait dan kan de connector van de flat cable andersom worden gemonteerd. Plak eventueel een sticker met nummer 1, 2 of 3 op elk van de motorassen.

USB kabel verbinden met de computer: Om de voorbereiding van de hardware compleet te maken wordt de Arduino door middel van de geleverde USB kabel nu verbonden met de PC. Een groen lampje op de Arduino blijft nu branden. De Arduino IDE App kan nu op de PC worden geïnstalleerd waarna programma codes naar de Arduino gestuurd kunnen worden.

4 G-code naar de Arduino sturen

Om CNC machines aan te sturen wordt standaard gebruik gemaakt van G-code. Hiermee kun je direct de gewenste doelcoördinaten (in mm) en de snelheid (in m/s) doorgeven aan je machine. Dit maakt het veel makkelijker om de gewenste bewegingen te programmeren dan wanneer je direct de motoren aan zou moeten sturen. Daar komt wel voor terug dat je de aansturing goed moet calibreren zodat de machine ook echt 1 mm beweegt wanneer je dat opgeeft.

Daarnaast zijn er standaard G- codes om een cirkelbeweging te maken waarbij twee motoren tegelijk worden aangestuurd, de bewegingssnelheid in m/s tussentijds wordt aangepast, een extra actuator wordt aangestuurd enz.

Om G-code naar de arduino te sturen, maken wij gebruik van het open source programma UGS.

4.1 UGS Downloaden en installeren

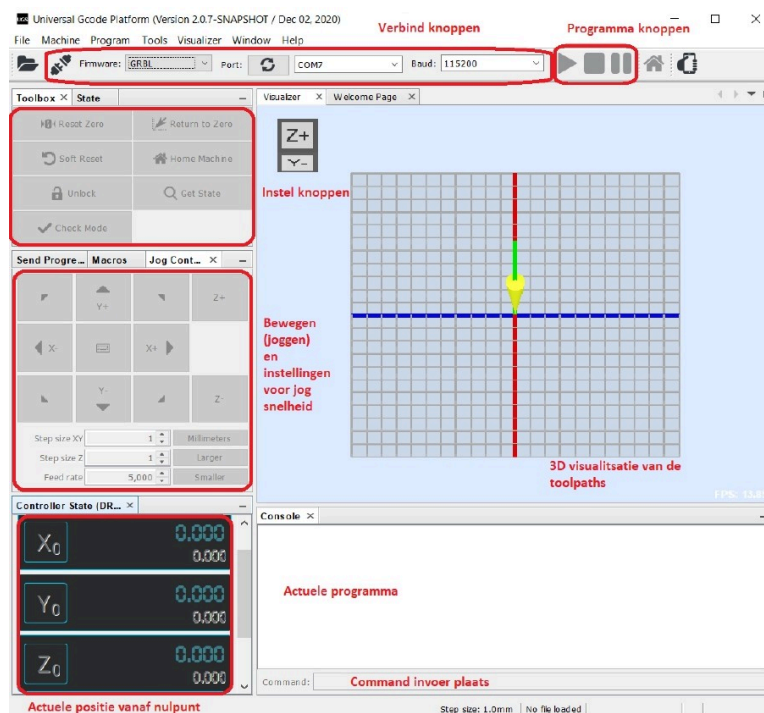
Met het open source programma UGS wordt de G-code geconfigureerd, waaronder het aantal stappen dat de stappenmotor moet maken per mm verplaatsing. Het UGS programma kan hier worden gedownload.

https://winder.github.io/ugs_website/

Kies "download" in de menu bar van de website en vervolgens "win64 ugs platform. Download het programma en unzip de file.

1. Download and extract the UGS Platform build from the downloads page.
2. Go to the ugsplatform/bin directory
3. On Windows run ugsplatform.exe or ugsplatform64.exe.

Het programma is nu geïnstalleerd en opgestart.



Figuur 3: Het UGS invoer scherm

4.2 UGS verbinden met de Arduino

Nu sluit je de arduino met de USB-kabel aan op je computer. Boven in de balk van UGS zie je drie velden staan: **Firmware**, **Port** en **Baud**. Controleer dat bij Firmware GRBL geselecteerd is en de Baud ingesteld is op 115200. Daarna kun je in het Port menu de juiste poort voor de Arduino kiezen. Wanneer je dit gedaan hebt, klik je op het stekkericoon links van deze velden om te verbinden met de

Arduino. Als dit lukt, wordt het icoontje oranje en zie je onder **Controller State** groot **IDLE** als status aangegeven.

In het scherm "Welcome Page", "Console" staat nu een hele lijst met parameters. Enkele van deze parameters worden nu geconfigureerd.

4.3 Configuratie G-code

De G-Code moet nu zo worden geconfigureerd dat een opgegeven aantal stappen waarmee de motoren worden aangestuurd overeen komt met een zekere x-, y- of z-verplaatsing in millimeters.

Bekijk eerst wat de huidige parameters zijn door `$$` in de regel onder het **Console** venster te typen en op Enter te drukken. In de console verschijnt dan een overzicht van de huidige parameters.

Deze parameters kun je aanpassen door de nieuwe waarde via dezelfde regel naar de Arduino te sturen. Als we bijvoorbeeld de Step idle delay (parameter 1) aan willen passen naar de waarde 255, kan dat door `$1=255` op te geven. Je kunt dan weer met `$$` controleren dat deze parameter ook echt aangepast is.

Hieronder staan de parameters die voor dit project van belang zijn:

- `$1 = 255`

Dit zorgt ervoor dat de stappenmotoren altijd op "de rem" staan als ze niet bewegen.

- `$100 (X), $101 (Y), $102(Z)` in steps/mm

Met deze drie parameters worden het aantal stappen van de stappenmotoren per mm verplaatsing van de assen opgegeven. Als de stappenmotor op 1/16 microstepping ingesteld staat en deze drijft een spindel aan met een spoed P dan wordt de voeding (feed) $16 \cdot 200$ stappen/omw wat overeenkomt met een verplaatsing van P mm. Hieruit volgt de voeding $16 \cdot 200 / P$ (stappen per mm verplaatsing). Met $P=4$ mm levert dit een feed van 800 stappen per mm verplaatsing ofwel `$100 = 800.000`.

Als de overbrengingsverhouding niet kan worden berekend dan zou deze gemeten kunnen worden wat in het algemeen minder nauwkeurig is. De gemeten waarde wordt dan ingevoerd bij "Actual movement". Met deze invoer zal de wizard berekenen en mededelen welke

waarde gekozen moet worden voor het aantal stappen/mm. Met deze input waarde zal de machine daadwerkelijk verplaatsen volgens de op te geven verplaatsing.

- $\$110, \$111, \$112 - [X,Y,Z]**$ Maximum snelheid (mm/min).

Dit is de maximum snelheid van de assen als deze niet belast zijn. Als in het G-code programma een hogere snelheid gegeven is dan de Maximum snelheid, dan wordt de snelheid begrensd worden door de waarde van de maximum snelheid. De maximum snelheid van de machine kan het best bepaald worden door de snelheid van de machine stap voor stap te verhogen en na te gaan of de stappenmotor stappen begint over te slaan. Het overslaan van stappen wordt slip genoemd.

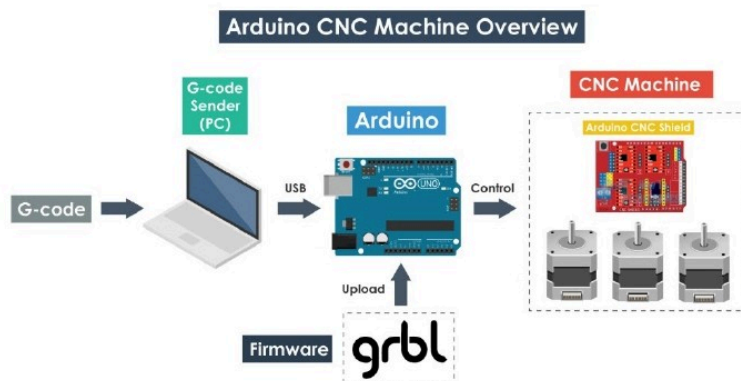
Als startwaarde wordt nu gekozen voor $v=100$ mm/min. Met $\$100=800.000$ ofwel 800 stappen per mm verplaatsing volgt dan $\$110 = 8000.000$.

- $\$120, \$121, \$122 - [X,Y,Z]**$ Versnelling in mm/sec²

Dit zijn de maximale versnellingen van de assen (sledes). Ook de maximale versnelling wordt bepaald door de waarde van de versnelling stap voor stap te verhogen en zo na te gaan wanneer slip op zal treden.

5 G-Code Programmeer handleiding

De GBRL library is op de Arduino geïnstalleerd. De verplaatsing is gekalibreerd en de maximum snelheid is begrensd. De volgende stap is het schrijven van een G-code programma. Dit is een tekstbestand. Het bestand krijgt ook wel de extensie (*.nc). Dit programma wordt vervolgens gerund met het UGS programma.



Figuur 4: Arduino CNC layout

Hieronder staan de basic G- en M codes die het meest gebruikt worden.

G0

Snelheid positioneerbeweging / Eilgang (in het Duits).

X,Y,Z

De coördinaten gemeten vanaf een ingesteld NULpunt.

G1

Een beweging met aanzetsnelheid (voedingssnelheid), dit is de snelheid bij een verspanende bewerking. De snelheid zelf wordt in dezelfde regel opgenomen met een F-code.

F

De aanzetsnelheid (mm/s). Staat in de regel met de G1-code.

M30

Een programma stop, laatste regel in het programma.

Hieronder staat een voorbeeld van een programma om een driehoek te frezen. Als nulpunt is gekozen voor het hoekpunt van de te bewerken plaat links onderin op een hoogte van $z = 20\text{ mm}$ vanaf de plaat. [Deze download je hier.](#)

```

1  G21 ; zet eenheden naar millimeters
2  G90 ; werk in absolute coördinaten
3  G94 ; snelheden in eenheid per minuut
4
5  G0 Z15.
6  G0 X0. Y0.
7  G0 X150. Y20.
8  G0 Z2.
9
10 G1 Z-0.5. F100.
11 G1 Y120.
12 G1 X50.
13 G1 X150. Y20.
14 G1 Z2.
15
16 G0 X0. Y0. Z15.
17
18 M30

```

Regels 5-8

In deze vier regels beweegt de frees vanuit het nulpunt met maximale snelheid naar Z=15mm boven het te bewerken werkstuk. Vervolgens beweegt deze met maximale snelheid naar de beginpositie (X150, Y20). Daarna tot 2 mm boven het werkstuk.

Regels 10-14

Vervolgens beweegt de frees met voedingssnelheid 100 mm/min naar beneden tot 0.5 mm in de plaat. Daarna met dezelfde voedingssnelheid naar de hoekpunten van de driehoek en uiteindelijk omhoog tot 2 mm boven het werkstuk.

Regel 16

Na de bewerking beweegt de frees met maximale snelheid naar het Nulpunt (0,0,15).

Regel 18

Met M30 wordt het programma beëindigd.

Het programma hieronder is een voorbeeld om een staafje op de draaibank over een lengte van 4 mm diameter met 2 mm kleiner te maken. Deze download je hier

```

1  G21 ; zet eenheden naar millimeters
2  G90 ; werk in absolute coördinaten
3  G94 ; snelheden in eenheid per minuut
4
5  G0 X0 Y0.          (Nulpunt is hoekpunt werkstuk)
6
7  G1 Y0.2 F100.       (0.2 mm snedediepte)
8  G1 X4 F100.         (4 mm langsdraaien)
9  G1 Y0 F100.         (terug naar Y0)
10 G0 X0.              (in Eilgang naar X0)
11
12 G1 Y0.4 F100.       (0.4 mm snedediepte)
13 G1 X4 F100.         (4 mm langsdraaien)
14 G1 Y0 F100.         (terug naar Y0)
15 G0 X0.              (in Eilgang naar X0)
16
17 G1 Y0.6 F100.       (0.6 mm snedediepte)
18 G1 X4 F100.         (4 mm langsdraaien)
19 G1 Y0 F100.         (terug naar Y0)
20 G0 X0.              (in Eilgang naar X0)
21

```

```

22  G1 Y0.8 F100.      (0.8 mm snedediepte)
23  G1 X4 F100.        (4 mm langsdraaien)
24  G1 Y0 F100.        (terug naar Y0)
25  G0 X0.             (in Eilgang naar X0)
26
27  G1 Y1.0 F100.      (1.0 mm snedediepte)
28  G1 X4 F100.        (4 mm langsdraaien)
29  G1 Y0 F100.        (terug naar Y0)
30  G0 X0.             (Naar oorspronkelijke
hoekpunt X0)
31
32  M30

```

6 G-Code inlezen in UGS en Nulpunt vastleggen

Selecteer >File>Open en lees de ASCII file met de G-code in. Verbind de machine aan de computer.

Vervolgens is het belangrijk om het nulpunt vast te leggen. Dit kan op twee manieren:

1. Als de machine is uitgeschakeld kun je hem handmatig naar het nulpunt bewegen, de machine aanzetten en vervolgens in UGS de "Toolbox/Reset Zero" knop indrukken. De positie waar de machine op dat moment in staat wordt dan het nulpunt. Nadeel van deze werkwijze is dat als de machine aangezet wordt, de stappenmotoren naar de eerste hele stap springen. Het kan dus zijn dat het nulpunt hierdoor niet exact klopt.
2. De tweede manier is het nullen met UGS als de machine aanstaat. Beweeg met de "jog Controller" naar het bedoelde nulpunt en druk op de X0, Y0 en de Z0 knop.

Zorg dat de Visualizer zichtbaar is (Menubalk>Window>Visualizer). Druk de Play button in de menubalk en controleer het path dat de tool doorloopt.

Na deze testrun kan het echte verspanen beginnen. Vergeet bij het verspanen niet de DC motor aan te zetten.

7 Handleiding Solenoid en Electromagneet



Figuur 5: Voorbeeld van een solenoid (links) en een mosfet-module (rechts).

Een solenoid, ook wel een trekspoel, is in feite niets anders dan een gewikkelde koperdraad / spoel met daarin een ferromagnetische pin. Als de stroom door de spoel gaat wordt er een elektromagnetisch veld opgewekt waardoor een ferromagnetische kern in de spoel getrokken wordt. De drukveer zorgt voor het terugveren naar de afgebeeld positie. Aan de andere kant van de solenoid bij de dopmoer resulteert het magneetveld in een duwbeweging.

Wanneer we de ferromagnetische pin weghalen uit deze solenoid, houden we een electromagneet over. Dit is eigenlijk niets anders dan een spoel. Wanneer we hier stroom doorheen laten lopen zal er een electromagnetisch veld opgewekt worden.

Zowel een solenoid als een electromagneet gebruikt te veel stroom en vaak een andere spanning om met een Arduino te worden geschakeld. Een Arduino solenoid driver kan uitkomst bieden (MOS module of relais module). De MOS FET is een soort elektronische schakelaar die anders dan een relais weinig stroom vergt om te schakelen en wel net als een relais veel stroom kan geleiden in gesloten stand. Als de 5V en GND van de Arduino gebruikt zouden worden om de solenoid te voeden, dan blijft er mogelijk te weinig stroom over om motoren te laten draaien die eveneens via de Arduino gevoed worden. Daarom is een externe voeding nodig die door de MOS FET geschakeld wordt om de solenoid te bekrachtigen.

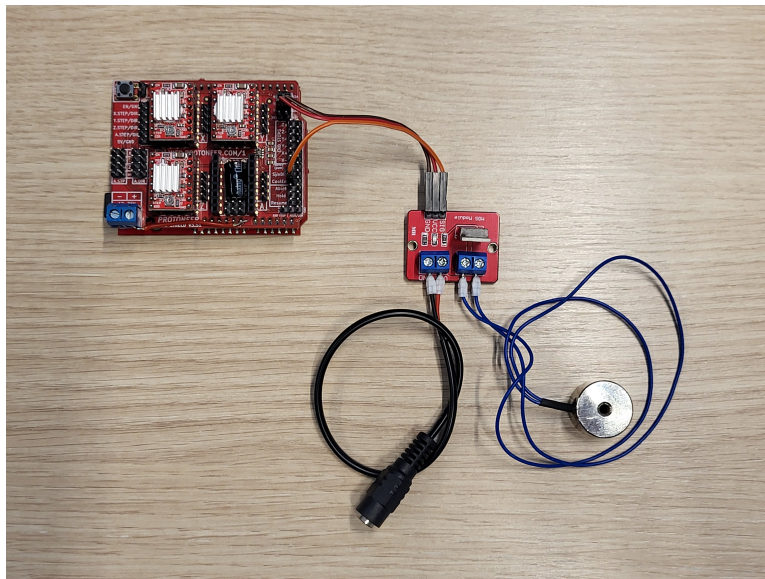
De MOS module heeft drie pinnen die aangesloten moeten worden op de Arduino: **SIG**, **VCC** en **GND**. De **VCC** en **GND** pinnen kunnen aangesloten worden op de 5V en GND pinnen die rechts boven in Figuur 2 te zien zijn. Hiermee voorzien we de MOS module van spanning. Via de **SIG** pin ontvangt de MOS module het signaal dat deze open gezet moet worden.

Alleen wanneer dit signaal positief is, kan er een stroom lopen door de outputs. Voor het aansluiten van de SIG pin zijn er twee opties op het CNC shield: `SpnEn` en `CoolEn`. Ook deze vinden we aan de rechterkant vna Figuur 2 Origineel zijn deze bedoeld om een spindel en koelvloeistof aan te sturen, maar deze kunnen ook voor andere dingen gebruikt worden.

Om de Solenoid of Electromagneet aan te sluiten op de MOS module, sluit je de draden hiervan aan op de poorten met labels `V+` en `V-`. Voor deze onderdelen maakt het niet uit welke kabel op welke poort aangesloten worden.

Tenslotte moet de MOS module ook een 12V spanning ontvangen. Dit is de spanning die doorgegeven wordt aan je electromagneet of solenoid wanneer de MOS module een positief signaal ontvangt. Hiervoor sluit je de DC contrastekker met de rode draad aan op `Vin` en de zwarte draad op `GND`.

Een voorbeeld van een volledig aangesloten electromagneet zie je in Figuur 6.



Figuur 6: Een electromagneet kan via een MOS-module aangesloten worden op de arduino. Hier is dat gedaan op de `CoolEn` pin van de CNC shield.

Wanneer de Solenoid, via een MOS module, op deze pinnen aangesloten is, kun je ze via de volgende pin-nummers en G-Codes aansturen:

Pin naam	Pin nummer	G-Code	
SpnEn	11	M3 S1000 of M4 S1000 M5	Start Stop
CoolEn	A3	M8 M9	Start Stop

Een alternatief voor de MOS FET is een relais module, waarbij een relais via een transistor wordt aangestuurd. De transistor is dan de schakelaar die met weinig stroomgebruik door de Arduino wordt aangestuurd. De hoofdstroom naar de Solenoid loop dan via het relais.

```
int solenoidPin = 11;           //We kiezen hier
                                //voor de SpnEn pin

void setup() {
  // put setup code here, to run once:
  pinMode(solenoidPin, OUTPUT); //Sets the pin as
  an output
}

void loop() {
  // put code here, to run repeatedly:
  digitalWrite(solenoidPin, HIGH); //Switch Solenoid
  ON
  delay(1000);                    //Wait 1 Second
  digitalWrite(solenoidPin, LOW); // Switch Solenoid
  OFF
  delay(1000);                    //Wait 1 Second
}
```

8 Handleiding 775 DC motor

Voor het aandrijven van de hoofdspil van een draaibank is een type 775 DC motor beschikbaar.

Verder zijn er 2 voedingen van elk 12V beschikbaar. Een van beide voedingen is al gebruikt om aan te sluiten op Arduino PCB. Met de extra 12V voeding kan een 24V power supply gecreëerd worden door beide voedingen in serie te schakelen.

De 12V aansluiting blijft dan in gebruik voor de aansluiting op de Arduino PCB terwijl de 24 V aansluiting gebruikt kan

worden voor de 775 DC motor. In de Faculty Workshop zal gedurende het project als alternatief ook een regelbare 24V voeding beschikbaar zijn. Ook tijdens het testen zelf zal een 24V labvoeding aanwezig zijn. Het in en uitschakelen van de 775 DC motor kan handmatig, dus hoeft niet via de Arduino geschakeld te worden.

Aansluiten van de DC motor via een relais

https://www.youtube.com/watch?v=RGGxlqCHL_c

Verdere uitleg over het CNC shield

<https://www.youtube.com/watch?v=fyH5hU6ctvw>

9 Installatie Arduino IDE App

! Belangrijk

Op de arduino's die je in dit project gebruikt, staat standaard al GRBL geïnstalleerd. Wanneer je zelf vanuit Arduino IDE code ernaar upload, overschrijf je dit en zul je daarna niet meer met UGS verbinding kunnen maken.

Je kunt GRBL weer op de arduino zetten met deze instructies: Paragraaf 10

Installatie Arduino IDE App: De Arduino zal nu instructies moeten krijgen om de motoren in beweging te brengen. Deze instructies worden in de vorm van programma code met de "Arduino IDE App" geüpload naar de Arduino. De meest recente versie van Arduino IDE App kan hier gedownload worden: <http://arduino.cc/en/Main/Software>

Geadviseerd wordt om de Arduino IDE App te installeren terwijl de Arduino verbonden is met de PC.

⚠ Admin-rechten

Voor het installeren van de Arduino IDE App heb je mogelijk administrator rechten op de PC nodig

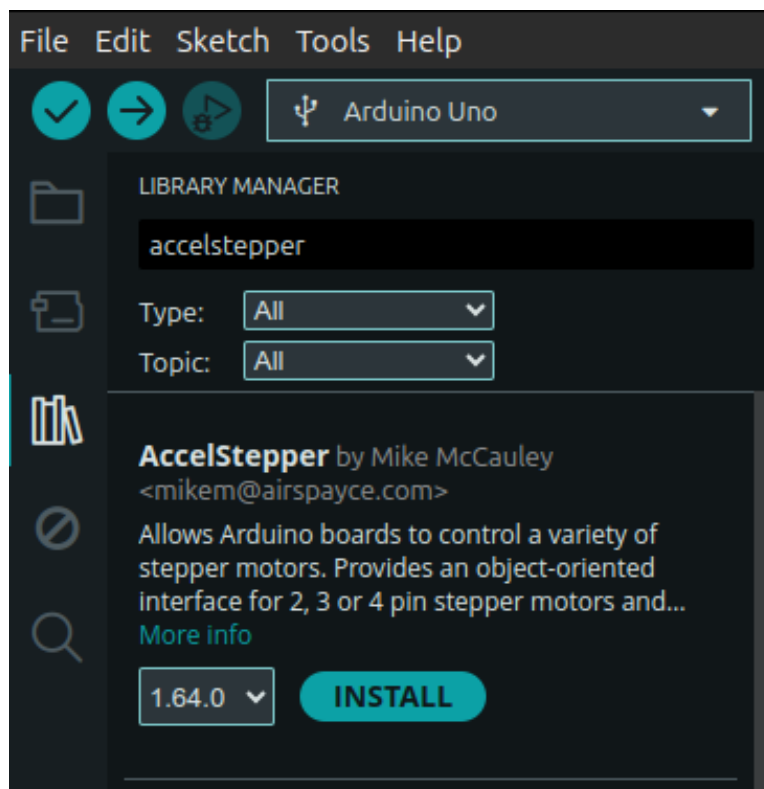
Selecteer na installatie in het bovenste venster van de Arduino IDE App de "Arduino Uno & COM poort". De Arduino is nu ook softwarematig verbonden met de Arduino IDE

App op de PC. Met de Arduino IDE App is het nu mogelijk programma's te uploaden naar de Arduino zelf.

9.1 Libraries uploaden

Arduino noemt een lijst met Arduino commando's ofwel een Arduino programma een sketch. Een sketch is een tekstbestand. Een voorbeeld van een sketch is gegeven in Paragraaf 11.

Voor het met de Arduino aansturen van devices, bijvoorbeeld stappenmotoren of servomotoren, kan gebruik gemaakt worden van een standaard set commando's die verzameld zijn in een zogenaamde library. Er zijn talloze libraries te vinden op internet. In het voorbeeldprogramma "sketch", in bijlage 1, wordt gebruik gemaakt van de library voor het aansturen van stappenmotoren genaamd `AccelStepper.h`. Deze library is standaard beschikbaar in Arduino IDE via `Manage Libraries`. Dit menu kun je openen door in de linkerbalk op het derde icoontje (die ziet eruit als vier boeken naast elkaar) te klikken. Daar kun je zoeken in een grote verzameling van beschikbare libraries.



Figuur 7: De library `AccelStepper` is in Arduino IDE via `Manage Libraries` te downloaden en installeren.

In dit menu kun je ervoor kiezen om de libraries die je wilt gebruiken te installeren door middel van de grote blauwe **INSTALL** knop. Daarna kun je deze gebruiken in je sketches.

9.2 Sketch compileren en uploaden:

De sketch gegeven in bijlage 1, met de commando's in de library Accelstepper, kan nu met de Arduino IDE App worden gecompileerd en overgebracht worden naar de Arduino met de "Upload" button linksboven in het scherm van de Arduino IDE app.

9.3 Motoren gaan draaien:

Als de 12V voeding is aangesloten op de Arduino dan zal een paar seconden na het uploaden het programma gaan lopen. Als de 12V voeding nog niet aangesloten was dan kan deze nu worden aangesloten en zal het programma direct daarna gaan lopen. De stappenmotoren zullen nu wisselend rechtsom en linksom blijven draaien tot de voeding weer is afgekoppeld. De USB kabel kan in principe worden losgekoppeld terwijl de motoren volgens het programma blijven draaien. Als de motoren nu synchroon met elkaar linksom en rechtsom draaien dan is de hardware in orde en de installatie van de software gelukt. Zoals eerder beschreven kan de draairichting van een motor worden aangepast door de connector van de flat cable op de printplaat om te draaien.

! Storing

Als een van de motoren niet goed functioneert dan is er mogelijk een hardware defect. Om er achter te komen wat er defect is kunnen motoren, kabels en controllers om de beurt onderling verwisseld worden om zo vast te stellen welk onderdeel mogelijk defect is. Een defect onderdeel kan worden ingewisseld bij de afhaalbalie van de werkplaats. Mocht mogelijk de voeding, Arduino of CNC shield defect zijn dan kunnen die om de beurt worden uitgewisseld met leencomponenten die beschikbaar zijn bij de afhaalbalie van de werkplaats. In plaats van dit Arduino programma is het meer gebruikelijk om stappenmotoren aan te sturen met een programma dat bestaat uit zogenaamde G-codes. Een G-code programma is wat makkelijker maar er zijn wat voorbereidende stappen nodig zoals een zogenaamde GRBL library uploaden.

10 Installatie GRBL library / uploaden

Het is nu mogelijk met de Arduino IDE App libraries te importeren en te uploaden waarmee vervolgens programma's in JavaScript, Python en andere programmeertalen kunnen worden ingelezen. In deze handleiding wordt de GRBL library gebruikt. GRBL staat voor G-code Language en fungeert als een vertaler van G-code opdrachten naar bewegingscommando's van CNC machines.

Importeren GRBL Library: Om de GRBL library te importeren in de Arduino IDE App wordt eerst onderstaande zipfile gedownload.

<https://github.com/gnea/grbl/releases/tag/v1.1h.20190825>

Isoleer in deze file de map "GRBL" en maak hier een zip file van. Deze grbl.zip moet naar de Arduino geüpload worden. Open hiervoor de Arduino IDE App en voer de volgende stappen uit.

1. Sketch > Include Library > Add .ZIP library¼ (zie figuur 3).
2. File > Examples > grbl > grblUpload (Navigeer naar grbl.zip op de PC en selecteer deze). In het onderste venster van de Arduino IDE App staat nu "Library installed". De GRBL library is nu ingelezen in de Arduino IDE App.
3. Selecteer vervolgens onder Sketch "Upload" om de GRBL library naar de Arduino te uploaden. Als dit gelukt is staat er in de notificatie van de IDE App, zie onderste regel helemaal rechts, "done uploading".

De GRBL library blijft op de Arduino staan en wordt automatisch opgestart als de Arduino wordt aangesloten. Het uploaden van de GRBL library is dus eenmalig.

Configureren GBRL Library: Na het uploaden van de GRBL library zal deze nog geconfigureerd moeten worden. Hierbij wordt de GBRL library zodanig aangepast dat een opgegeven aantal stappen van de stappenmotor overeenkomt met een zekere verplaatsing in millimeters. Een open source programma dat hiervoor gebruikt kan worden is "Universal G-code Sender (UGS)". Sluit de Arduino IDE App af en ga verder met de installatie van UGS.

11 Bijlage 1: Arduino programma voor stappenmotoren

Dit programma is ook als download beschikbaar: [files/sketch-AccelStepper.ino](#)

Meer informatie over de library Accelstepper

```
#include <AccelStepper.h> // Stepper driver library

// Dit zijn de pinnen die de Arduino gebruikt om de
// stappenmotoren aan te sturen
#define XSTEP 2
#define YSTEP 3
#define ZSTEP 4

#define XDIR 5
#define YDIR 6
#define ZDIR 7

#define ENABLEPIN 8

#define LEDPIN 13 // Dit is de onboard led

#define STEPS (float)200
#define DEGREES (float)360

AccelStepper stepperX(1, XSTEP, XDIR); //
// initialiseren van de stapper op poort x
AccelStepper stepperY(1, YSTEP, YDIR); //
// initialiseren van de stapper op poort y
AccelStepper stepperZ(1, ZSTEP, ZDIR); //
// initialiseren van de stapper op poort z

void setup() // Deze routine wordt 1 keer gerund aan
// het begin van het programma
{
    // Met pinMode zet je een pin op input/output
    pinMode(LEDPIN, OUTPUT); //Nu zetten we de
    // LEDPIN(13)
    pinMode(ENABLEPIN, OUTPUT); // en ENABLEPIN(8) op
    // output.

    // Met Serial kan je communiceren met de Serial
    // monitor in Arduino IDE op je pc.
    Serial.begin(9600); // Gebruik dit commando
    // eenmalig om de verbinding te maken.
    Serial.println("hoi"); // Nu kan je Serial.print
    // en Serial.println gebruiken.
```

```

    // Met digitalWrite kan je pinnen HIGH en LOW
    maken. Dit gebruiken we nu om de stappenmotoren in
    te schakelen.
    digitalWrite(ENABLEPIN,LOW); // LOW= de motoren
    zijn actief. HIGH= de motoren zijn vrij te bewegen.

    // Tot slot laten we de led kort aangaan om te zien
    dat void setup() klaar is.
    digitalWrite(13,HIGH);
    delay(1000); // Met delay kan je de Arduino X aantal
    miliseconden laten wachten.
    digitalWrite(13,LOW);

    // Hier stellen we een arbitraire snelheid en
    versnelling in als voorbeeld
    stepperX.setMaxSpeed(2000);
    stepperY.setMaxSpeed(2000);
    stepperZ.setMaxSpeed(2000);

    stepperX.setAcceleration(4000);
    stepperY.setAcceleration(4000);
    stepperZ.setAcceleration(4000);
}

void loop()
{
    // In deze void loop hebben we een voorbeeld
    programma dat heen en weer draait
    stepperX.moveTo(1000); // De doelpositie in aantal
    stappen.
    stepperY.moveTo(1000);
    stepperZ.moveTo(1000);
    while(stepperX.distanceToGo()+stepperY.distanceToGo()+stepperZ.distanceToGo() )
    {
        stepperX.run(); // Beweging in gang zetten tot
        doelpositie is bereikt.
        stepperY.run();
        stepperZ.run();
    }

    stepperX.moveTo(-1000); // Doelpositie in stappen
    terug bewegen
    stepperY.moveTo(-1000);
    stepperZ.moveTo(-1000);
    while(stepperX.distanceToGo()+stepperY.distanceToGo()+stepperZ.distanceToGo() )
    {
        stepperX.run(); // Beweging in gang zetten tot
        doelpositie is bereikt.
        stepperY.run();
    }
}

```

```
    stepperZ.run();  
  }  
}
```