



HOGERE ZEEVAARTSCHOOL

SCHEEPSWERKTUIGGUNDE

BOUW & ONTWERP VAN EEN CNC- MACHINE

SMET VICTOR

Bachelorscriptie

Academic year: 2020-2021

Voorwoord

In het kader van het opleidingsonderdeel bachelor scriptie is er een project tot stand gekomen voor het ontwikkelen en bouwen van een CNC-machine die in staat is voor het bewerken van voornamelijk zachte materialen. Dit project is een zeer interessante manier om de opleidingsonderdelen zoals automatisatie, digitale en analoge technieken samen te bundelen en in de praktijk te kunnen verwezenlijken. De bedoeling is om enkele technische en praktische onderdelen van mijn kennis te kunnen weerspiegelen in dit project.

Ik wil graag de Hogere Zeevaartschool te Antwerpen bedanken die bereid was dit project te ondersteunen en te financieren waardoor het mogelijk was dit project te realiseren alsook mij de kennis te verschaffen die noodzakelijk waren voor een goede afloop van dit project. Hierbij wil ik ook mijn promotor Rik Floren bedanken die de uitdaging niet uit de weg ging om elke situatie bij te sturen waar nodig en het ter beschikking stellen van een Arduino gebruikt in het project. Even eens wil ik graag Vincent Leysen bedanken voor het projectidee.

Verder bedank ik nog Willem Maes voor het gebruik van de vloeistof verdeler waarop het project verder gebouwd is wat een evidente fundering van het project was.

Samenvatting (NL)

Het project zal bestaan uit het ombouwen van een lijm dispenser naar een volledig geautomatiseerde CNC-machine. Deze machine is in staat om zachte materialen te bewerken die een dikte kunnen aannemen Tot ongeveer 50mm. Door het gebruik van enkele microcontrollers zal het systeem G-code kunnen lezen en verwezenlijken. het CNC-toestel zal beschikken over 3 assen die elk bediend worden door een individuele stappenmotor.

Ook zal het systeem beschikken over een toerentalregelaar voor het verspanend gereedschap wat het mogelijk maakt om verschillende materialen te bewerken.

Het systeem zal bediend worden door Universal G-code sender. Dit is een software die in staat is om te communiceren met een microcontroller die compatibel is met de GRBL bibliotheek. De G-code is universeel en kan door elke CAD-software ontwikkeld en verwerkt worden.

Samenvatting (EN)

This project will contain the conversion of a fluid dispenser into a fully automatic CNC-machine. This machine is capable of processing mostly soft materials up to 50mm in thickness. Due to the use of microcontrollers the system will be capable of reading and processing G-code. The system will consist of 3 axis which are controlled by individual stepper motors as well as a controllable tool speed which makes it easier to process different types of material. The system will be run by UGS or Universal G-code sender. This software is capable of communicating with a microcontroller which is compatible with the GRBL library. De G-code is universal and can be developed in almost any CAD-software.

Inhoudsopgave

Lijst van figuren	vii
Lijst van tabellen.....	viii
Lijst van afkortingen	ix
verklarende woordenlijst	x
1. Inleiding.....	1
2. Wat Is een CNC-machine?	3
3. werking en genereren van G-code.....	5
3.1 Feed Rate.....	5
3.2 Voornaamste functies bij gebruik van G-code	6
4. Werking CNC-machine.....	7
5. Componenten	9
5.1 Microcontroller en development board - Arduino Mega 2560 & Arduino Uno	9
5.1.1 Werking microcontroller	9
5.1.2 Specificaties Arduino Mega 2560.....	10
5.1.3 Specificaties Arduino Uno.....	11
5.2 Stappen motoren	12
5.2.1 Werking stappenmotor	13
5.2.2 De Unipolaire stappenmotor	14
5.2.3 De Bipolaire stappenmotor	14
5.3 Motor Drivers.....	15
5.3.1 Microstepping	15
5.3.2 Instellen van de resolutie - A4988	16
5.3.3 PWM - Pulse width modulation.....	17

5.4 Einde eindeloopschakelaar	18
5.4.1 werking eindeloopschakelaar.....	18
5.5 Toerentalregelaar Frees	19
5.6 Voeding.....	20
6. Software.....	21
6.1 Arduino IDE.....	21
6.1.1 Arduino IDE structuur	21
6.1.2 Definiëren van de variabelen	21
6.1.3 Void Setup	22
6.1.4 Void Loop.....	23
6.2 GRBL	24
6.3 CAD Software	25
6.4 Universal G-code sender	26
6.4.1 UGS Toolbox	28
6.4.2 Zero en nullen.....	29
6.4.3 Hard limits and soft limits	29
6.4.4 UGS Firmware settings.....	30
7. Kalibreren van de CNC-machine	31
8. Aansluiting CNC-machine	37
8.1 aansluiting stappen motoren	37
8.2 aansluiting motor sturing	37
Bibliografie.....	40
Bijlagen.....	44

Lijst van figuren

Figuur 1: Voorbeeld G-code	3
Figuur 2: Zonder cover	7
Figuur 3: Originele installatie	7
Figuur 4: Arduino Mega 2560.....	12
Figuur 5: Schema stappenmotor	13
Figuur 6:Huidige Stappenmotor	13
Figuur 7: Unipolar vs. Bipolar.....	14
Figuur 8: CNC-shiel.....	16
Figuur 9: Puls breete modulatie	17
Figuur 10: Mosfet Motor Control	19
Figuur 11: Code Motor sturing	20
Figuur 12: Void Setup	22
Figuur 13: Void Loop	23
Figuur 14: Config.h.....	24
Figuur 15: voorbeeld G-code 2.....	25
Figuur 16: Voorbeeld UGS.....	27
Figuur 17: Toolbox.....	28
Figuur 18: firmware settings	30
Figuur 19: Aansluiting CNC-machine.....	38

Lijst van tabellen

Tabel 1: functies G-code.....	6
Tabel 2: specificatie Arduino Mega 2560.....	10
Tabel 3: Specificaties Arduino Uno	11
Tabel 4: Resolutie driver	16
Tabel 5: Voorbeelden variabelen.....	22
Tabel 6: bijlage figuur 15.....	26
Tabel 7: Step Pulse configuration.....	31

Lijst van afkortingen

CNC	Computer numerical control
CAD	Computer Aided Design
COM poort	Communication port
UGS	Universal G-code Sender
CPU	Central Processing unit
IDE	integrated development environment
UART	Universal asynchronous receiver-transmitter
SRAM	Static random access memory
EEPROM	Electrically erasable programmable read-only memory
PWM	Pulse Width modulation
LED	Light emitting diode

verklarende woordenlijst

Flash memory:	Een apparaat dat data behoudt als het uitgeschakeld wordt
clock speed:	De snelheid waarmee data verwerkt of geschreven kan worden per seconde.
Nullen:	Een toestel naar zijn gekalibreerde positie brengen om feedback te krijgen van zijn positie
Hardware:	Elektronische componenten die men fysiek kan aanraken
Software:	Een computerprogramma dat in staat is een specifieke taak uit te voeren.
open source:	Een ontwikkeling die door iedereen gebruikt kan worden zonder de bron te hoeven vermelden.
homing cycle:	Een cyclus die gestructureerd het nullen gaat uitvoeren.

1. Inleiding

In deze scriptie wordt de bouw en ontwerp van een CNC-machine onderzocht en uitgevoerd. De CNC-machine wordt verder gebouwd uit een bestaand project. Dit project was een vloeistof verdeler die pneumatisch aangestuurd werd. Ontworpen door Asymtek in het jaar 2000.

Het model betreft, Asymtek Automove 403. Hierop verder bouwend is deze omgevormd tot een CNC-machine die in staat is om aan de hand van G-code een werkstuk te realiseren.

De G-code is universeel en kan met talloze CAD software gegenereerd worden. Verder wordt ook de werking van het elektronische en theoretische aspect besproken.

Door behoud en recuperatie van een aantal onderdelen is het toestel in staat om werkstukken met een dikte tot 5 cm te bewerken met een nauwkeurigheid van 0.1mm

De CNC machine bestaat uit 3 stappenmotoren die de assen (X,Y en Z) zullen bewegen naar de gewenste positie. Op zijn beurt worden de stappenmotoren aangestuurd door een microcontroller met een CNC-shield en compatibele drivers. De verdere werking en componenten worden later besproken.

Het onderzoek zal bestaan uit de ontwikkeling van een CNC-machine, bewerking en het gebruik van de CNC-machine en zijn toepassing op de zeevaartsector.

2. Wat Is een CNC-machine?

CNC staat voor *Computer numerical control*. Kortweg zullen werktuigmachines die normaal handgestuurd bediend worden, geregeld worden door een computer die aan de hand van coördinaten de juiste handeling

uitvoeren. De coördinaten worden

bepaald door een input genaamd

G-code. G-code specificeert de

handelingen die het toestel moet

uitvoeren op een gestructureerde

manier. Deze instructies worden

opgemaakt in het ontwerpproces

van het werkstuk. Niet alleen

worden de coördinaten

gespecificeerd maar ook de

handelingen die vereist zijn, zoals het veranderen van frees of verplaatsing van het werkstuk.

Een CNC-machine is vergelijkbaar met een 3D-printer waarbij het verschil zit in het

wegnemen van materiaal of het toevoegen van materiaal. Een 3D-printer is hierin efficiënter

aangezien deze geen of gering materiaal verspilt. Ondanks de verspilling is het nog niet

mogelijk om alle materialen te printen omdat deze in vloeibare vorm aangebracht moeten

worden en tijd nodig hebben om te stollen. Voor het maken van werkstukken uit hardere

materialen is het met de huidige technologie enkel mogelijk met gebruik van een CNC-

machine. Een voordeel van de CNC-machine is dat het zeer tijdbesparend is, alsook

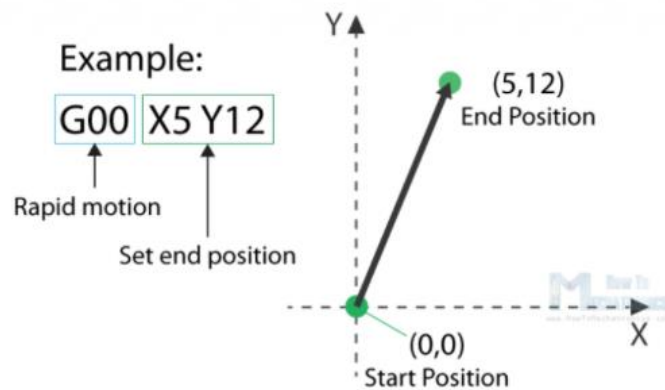
kostenbesparend door de automatisering van de uit te voeren handelingen. CNC-machines

worden vaak gebruikt bij productieprocessen die repetitief zijn waardoor elk werkstuk

identiek is en de menselijke fouten achterwege blijven. Aan de andere kant is er wel

mogelijkheid op mechanische gebreken. (Wikipedia, 2020)

G00 - Rapid Positioning



Figuur 1: Voorbeeld G-code

Bron: howtomechatronics.com

3. werking en genereren van G-code

G-code is de instructie om vanuit een theoretisch werkstuk een CNC-machine aan te sturen. De code wordt gegenereerd door CAD software (*Computer aided design*). In deze software wordt een werkstuk ontwikkeld en geprogrammeerd. Voorbeelden hiervan zijn Fusion 360, Autocad, Inkscape... .

De G-code kent vele varianten bepaald door het cijfer gevolgd naast de letter G (G27 bv.). In de basis vorm zal de G-code enkel de coördinaten van de frees bepalen. Naast de G-code is er ook nog M-code, M staat voor "*miscellaneous*". Dit wil zeggen dat de ontwerper van het werkstuk kan beslissen om de operatie te onderbreken en de machine een andere opdracht laat uitvoeren zoals wisselen van spaangereedschap. (Howtomechatronics, 2020)

In Figuur 1 zien we een voorbeeld van een G commando (G00 X5 Y12 F200).

- G00 = snel bewegen van de frees zonder materiaal weg te nemen
- X5 = verplaats de frees met 5 millimeter in de X richting
- Y12 = verplaats de frees met 12 millimeter in de Y richting
- F200 = dit is de *feed rate*, 200 millimeter per minuut

3.1 Feed Rate

De *Feed Rate* is de snelheid waarmee de machine door het werkstuk gaat bewegen en wordt uitgedrukt in millimeter per minuut.

Deze is zeer belangrijk, wanneer deze incorrect is ingesteld, kan er schade veroorzaakt worden aan het werkstuk of het verspanend gereedschap. De *feed rate* is afhankelijk van het materiaal van het werkstuk of het materiaal van het gereedschap.

naast de G, M en de F als functie zijn er talloze andere commando's die voor specifieke situaties gebruikt kunnen worden. met voornaamste \$H. Dit commando geeft de opdracht om het toestel te nullen.

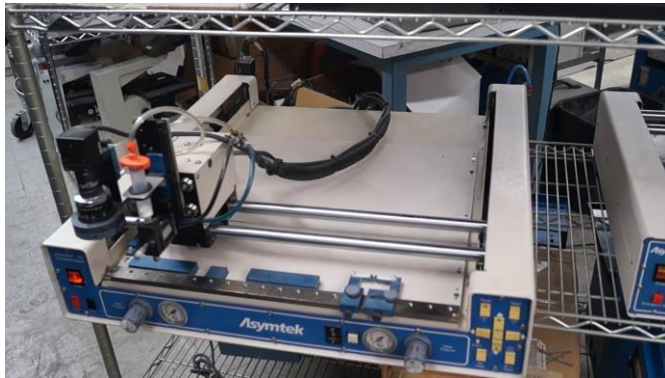
3.2 Voornaamste functies bij gebruik van G-code

Tabel 1: functies G-code
Bron: Victor Smet

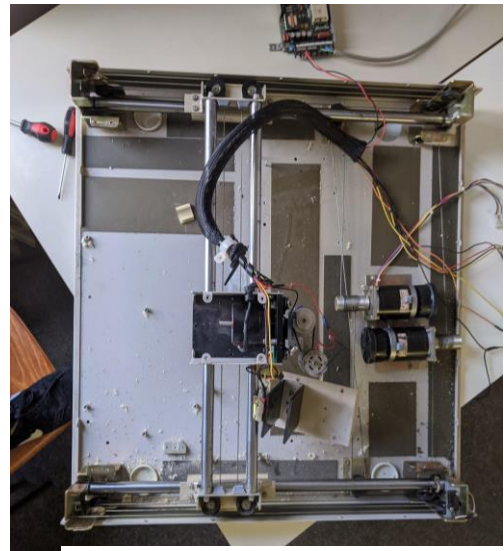
G00 rapid positioning	De frees zal zich zo snel mogelijk bewegen naar de opgegeven coördinaten
G01 Linear positioning	De frees zal zich Lineair bewegen naar opgegeven coördinaten met de snelheid bepaald door de <i>feed rate</i> .
G28 Homing	De frees zal zich met maximum <i>homing rate</i> naar het home punt van de installatie begeven
G09 Freeze	Stoppen van alle acties (<i>Freeze</i>)
F Feedrate	Bepaalt de snelheid waarmee de frees beweegt in millimeter per minuut
M01 Stop	Stopen van de frees
M03 Clockwise	De draairichting van de frees wordt met wijzers mee
M04 Counter clockwise	De draairichting van de frees wordt tegen wijzerzin
M06 Automatic Tool change	De freeskop zal vervangen worden door een ander verspanend gereedschap
H Tool offset	Definiëren van de Afstand tussen de frees en het werkstuk
S Speed	bepaalt de draaisnelheid van de frees

4. Werking CNC-machine

Om de CNC machine correct te laten functioneren zijn er een aantal componenten vereist. Deze worden verder in detail besproken in een latere sectie. Zoals eerder vermeld zal de machine handelingen uitvoeren op basis van een eerder gegenereerde G-code. Deze Code gaan we invoeren in een G-code verzender. Dit is software die in communicatie staat met de microcontroller via een COM poort. Een COM poort is een USB poort die in staat is om een input en output te transfereren. Voor dit project wordt UGS gebruikt. (*Universal G-code sender*). In deze software kan men gaan definiëren wat de instellingen van de CNC-machine zijn. UGS communiceert en met de microcontroller via de COM poort. Op de microcontroller bevindt zich een *CNC-shield* waarop de stappenmotoren en eindslag schakelaars aangesloten zijn. Zodra er een commando wordt gegeven in de software zullen de stappenmotoren bewegen naargelang de gevraagde actie indien het toestel correct gekalibreerd is. Naast de sturing van de stappenmotoren is er een tweede microcontroller geïmplementeerd die instaat voor het regelen van het toerental en draairichting van de frees aan de hand van een potentiometer en een drukknop respectievelijk. Het toerental zal geregeld worden door een potentiometer die gekoppeld is met de microcontroller en de driver. Verder is er een noodstop geplaatst om bij ongewenste situaties snel te kunnen ingrijpen.



Figuur 2: Originele installatie
Bron: Victor Smet



Figuur 3: Zonder cover
Bron: Victor Smet

5. Componenten

5.1 Microcontroller en development board - Arduino Mega 2560 & Arduino Uno

De microcontroller is een essentieel component van de CNC-machine aangezien alle data en feedback verwerkt wordt door de CPU die zich op de microcontroller board bevindt. Men kan stellen dat de microcontroller het hart van het systeem symboliseert. De dag van vandaag vindt men in zo goed als alle elektronische toepassingen wel een microcontroller terug omdat deze goedkoop en eenvoudig te programmeren is. Voor deze CNC-machine is er geopteerd voor het gebruik van de Arduino. De Arduino is een van de meest gebruikte development boards die werken met de ATmega microcontrollers aangezien deze werken met een eenvoudige programmeertaal aan de hand van de Arduino IDE software. Ook is Arduino “*open source*”, dit wil zeggen dat iedereen vrij is het model te kopiëren of aan te passen wat deze ook zeer goedkoop en gebruiksvriendelijk maakt. Arduino beschikt ook over een ruim aanbod van varianten die vooral verschillen in grootte en capaciteit. De kleinste variant is de Arduino micro en de grootste Arduino mega 2560, dewelke gebruikt wordt in dit project. (store.arduino.cc, 2021)(‘Arduino Mega 2560 Rev3 | Arduino Official Store’, z.d.)

5.1.1 Werking microcontroller

Een microcontroller beschikt over een aantal “pins” die functioneren als ingangen en uitgangen (*Inputs & Outputs*). Deze kunnen allemaal digitale signalen opnemen of uitsturen. Ook zijn er specifieke “pins” die in staat zijn om analoge data te verwerken of uit te sturen. De *in- en outputs* worden bepaald door de CPU (*central processing unit*) van de Arduino. De CPU kan men programmeren met de Arduino software waar men de in en uitgangen heeft gedefinieerd. Ook zal de CPU de geüploade code verwerken en handelen naargelang de beschikbare data of commando’s. Verder is het nog mogelijk om extra componenten en “*shields*” aan te sluiten op de Arduino om de mogelijkheden eindeloos uit te bereiden. In dit geval wordt er gebruik gemaakt van een *CNC-shield* waardoor de aansluiting van de motor drivers en eindslag schakelaars vereenvoudigd wordt. (Wikipedia, 2021)

5.1.2 Specificaties Arduino Mega 2560

De Arduino Mega is een 8-bit development board die beschikt over 54 Digitale in en uitgangen (waarvan 15 mogelijk als puls breedte modulatie) alsook 16 Analoge, 4 UART's. De microchip is een ATmega2560 met een kloksnelheid van 16 Mhz. Het board functioneert op 5V of 3.3V en is compatibel met meest van de “*shields*” ontworpen voor de Arduino Uno. (store.arduino.cc, 2021)

Tabel 2: specificatie Arduino Mega 2560
Bron: store.arduino.cc

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

5.1.3 Specificaties Arduino Uno

De Arduino Uno is het eerst ontwikkelde development board door Arduino en is gebaseerd op de ATmega328P. Het heeft 14 digitale in- en uitgangen waarvan 6 mogelijk als PWM. De CPU beschikt over een kloksnelheid van 16 Mhz en is compatibel met zo goed als elke module. (store.arduino.cc, 2021)

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

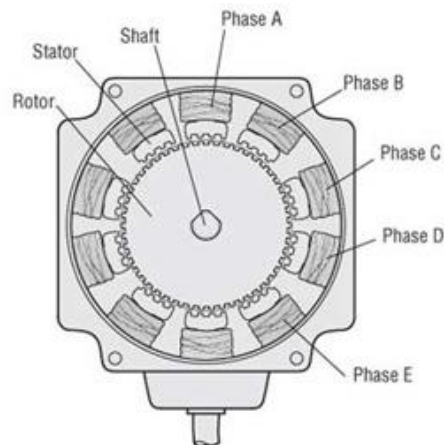
Tabel 3: Specificaties Arduino Uno
Bron: store.arduino.cc

5.2.1 Werking stappenmotor

Door het gebruik van een permanente magneet als rotor en een variabel aantal spoelen in de stator is een stappenmotor in staat om een bepaalde hoeveelheid toeren te maken en stil te staan op een gewenste positie. De spoelen die elk beschikken over een noord - en zuidpool zullen om de beurt bekrachtigd worden wat de tegenpool van de rotor zal doen aantrekken. In theorie kan men zoveel spoelen plaatsen als men wenst om de nauwkeurigheid te verbeteren. In de praktijk gebeurt dit niet en gaat men gebruik maken van een proces dat half stepping wordt genoemd. Dit wil zeggen dat er 2 spoelen tegelijkertijd worden bekrachtigd en de rotor zich gaat positioneren tussen deze 2 spoelen. Om de nauwkeurigheid nog verder te verbeteren gaat men gebruik maken van tanden op de rotor en stator. Op de stator bevinden zich minder tanden dan op de rotor waardoor deze altijd zullen verschuiven indien deze worden aangetrokken. De motor drivers zullen ervoor zorgen dat de juiste spoelen worden aangetrokken. Men kan stappen motoren ook nog opdelen in 2 groepen, namelijk de unipolaire en de bipolaire stappenmotor. De gebruikte stappenmotoren zijn van het type bipolair. (Wikipedia, 2021 & Microchip Technology, Stepping Motors Fundamentals, 2004)



Figuur 6: Huidige Stappenmotor
Bron: Victor Smet



Figuur 5: Schema stappenmotor
Bron: openlabpro.com

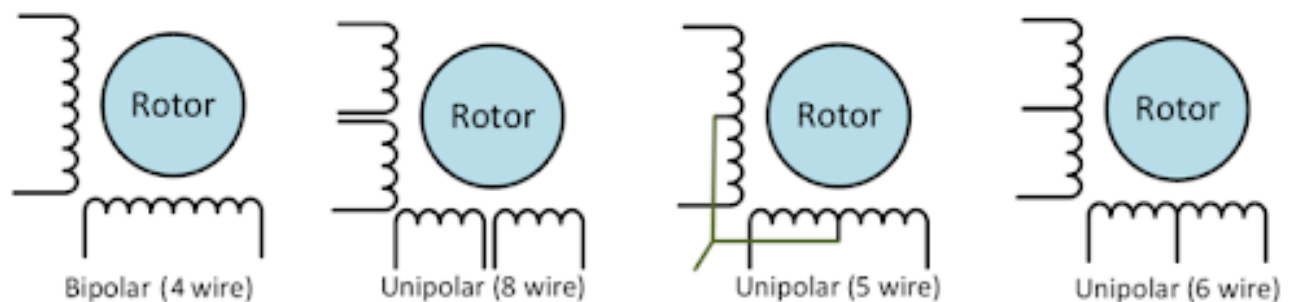
5.2.2 De Unipolaire stappenmotor

Dit type motor beschikt meestal over 4 spoelen die elk een gemeenschappelijke aansluiting hebben. De motor heeft dus 6 aansluitingen. Dit zorgt ervoor dat er telkens maar 2 halve wikkelingen bekrachtigd worden waardoor de rotor zich verplaatst. Door het gebruik van de 4 spoelen is deze motor minder compact in verhouding met de bipolaire stappenmotor. Verder is er nog een mogelijkheid waarbij de gemeenschappelijke aansluitingen samengevoegd worden waardoor er maar 5 aansluitingen zijn alsook een uitvoering met 4 spoelen die elk apart bekrachtigd worden met 8 aansluitingen.

5.2.3 De Bipolaire stappenmotor

De werking van de bipolaire stappenmotor is dezelfde als de unipolaire met het verschil dat deze maar beschikt over 2 spoelen die volledig bekrachtigd worden. Hierdoor heeft deze motor maar 4 aansluitingen (A+, A- en B+, B-), wat deze motor ook compacter maakt.

(Journal of automation and control, 2015 & wiki.stematschool, 2017)



Figuur 7: Unipolar vs. Bipolar
Bron: Wiki.stematschool.be

5.3 Motor Drivers

Voor het bedienen van de stappenmotoren is zoals eerder vermeld een driver nodig om de nauwkeurigheid van de stappen te kunnen verbeteren. De drivers zullen aangesloten worden op het *CNC-shield* van de microcontroller en gestuurd worden door het programma dat een nauwkeurige beschrijving geeft van de gewenste functie. De gebruikte driver is van het type A4988 en ondersteunt microstepping met modus van 1/1, 1/2, 1/4, 1/8 en 1/16 voor het toepassen van microstepping. De module is beveiligd tegen overstroom tot 2,5A en werkt tussen de 8 en 35V. Men kan de modus aanpassen door het verbinden van een brug tussen de gewenste aansluitingen. De drivers maken gebruik van PWM om de gewenste microstepping te bekomen.

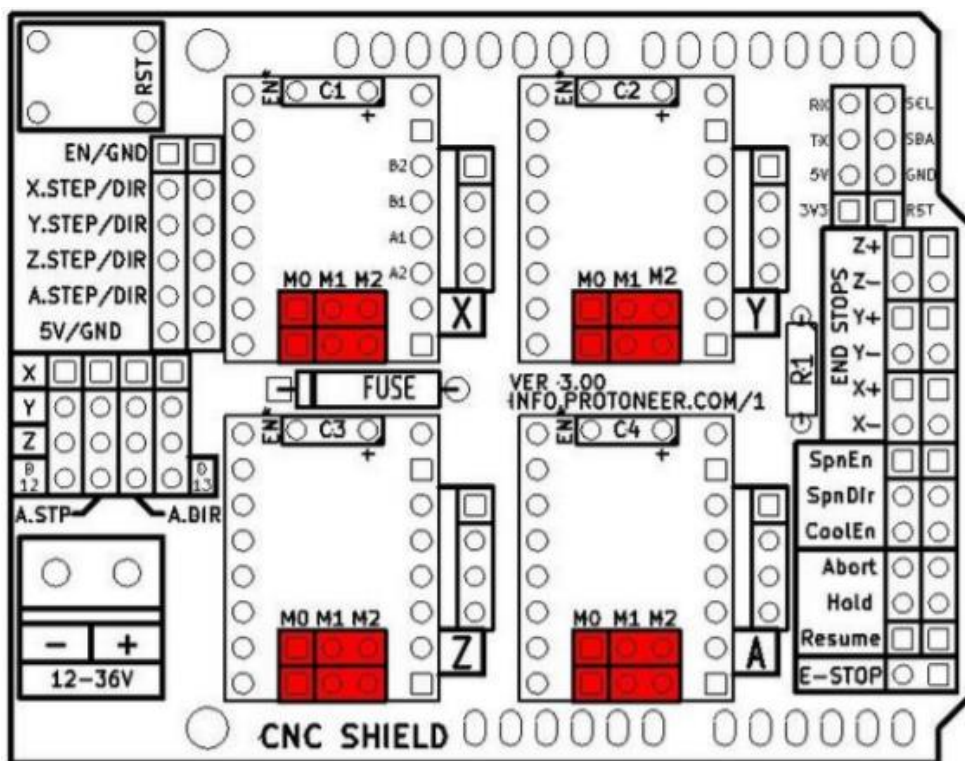
5.3.1 Microstepping

Het doel van microstepping is het bekomen van een vlottere operatie van het toestel en het vergroten van de nauwkeurigheid van de stappenmotoren. Nauwkeurigheid van stappenmotoren wordt ook wel de resolutie genoemd. De drivers zullen microstepping toepassen met een ratio van 1/8, wat wil zeggen dat de motor 8 pulsen zal krijgen voor deze 1 stap is verplaatst. Aangezien de motoren 1.8° bewegen per puls met een verhouding van 1 zullen deze nu 0.225 graden verplaatsen per puls. Zoals de voordelen die vernoemd zijn komen er ook nadelen kijken bij het gebruik van microstepping.

Het koppel van de motoren zal verminderen alsook de snelheid waarmee de motoren bewegen omdat de stappen dicht bij elkaar geplaatst zijn.

Het programma dat gebruikt wordt om de G-code te verzenden vereist dat de het aantal stappen per graden wordt omgezet in stappen per millimeter, dit wordt verder besproken in het onderdeel kalibratie. (Journal of Computer Science and Control Systems, 2011)

5.3.2 Instellen van de resolutie - A4988



Figuur 8: CNC-shield
Bron: Handsontec.com

Op Fig. 7 zien we de technische tekening van het *CNC-shield*. De rood gemarkeerde vlakken zijn aansluitingen die we kunnen overbruggen om de resolutie aan te passen van de drivers.

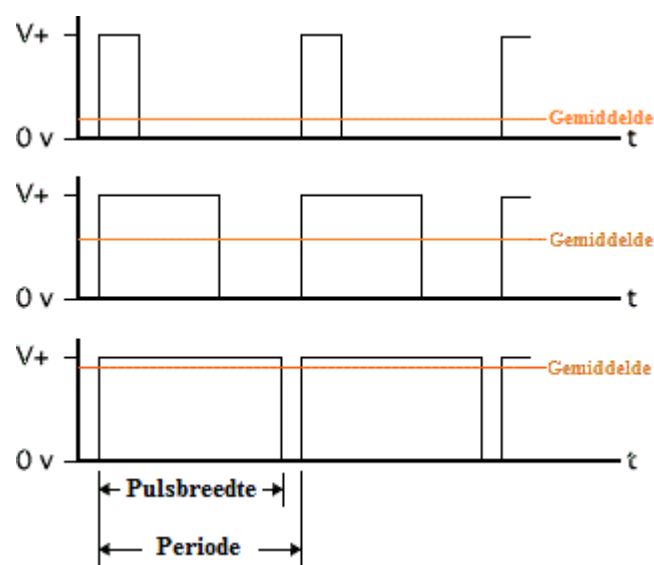
De correcte aansluiting van de resolutie vinden we terug in tabel 4. “Low” indiceert dat er geen connectie is gemaakt voor de betreffende brug en “high” indiceert dat er een klem is aangesloten op deze aansluiting. Voor het verkrijgen van een resolutie van $\frac{1}{8}$ moet er een verbinding gemaakt worden tussen MS0 en MS1. MS2 blijft open. (Handsontech, 2011)

Tabel 4: Resolutie driver
Bron: Handsontec.com

MS0	MS1	MS2	Resolutie
Low	Low	Low	1
High	Low	Low	1/2
Low	High	Low	1/4
High	High	Low	1/8
High	High	High	1/16

5.3.3 PWM - Pulse width modulation

PWM of puls breedte modulatie is een manier om aan de hand van een digitaal signaal een blok golf te genereren die kan dienen om informatie over te brengen of in deze toepassing een voeding te regelen van een toestel. Een bekende toepassing van PWM is een dimmer van een led. Een led is een gelijkspanningscomponent dat enkel aan of uit kan staan. Door op het signaal van de voedingsspanning een PWM toe te passen is het mogelijk om een blok golf te genereren die het signaal onderbreekt wanneer gewenst. De led zal hierdoor aan en uit geschakeld worden. Afhankelijk van de frequentie (indien snel genoeg) lijkt het of de sterkte van de led geregeld wordt terwijl de led in werkelijkheid maar de helft van de tijd brandt. (afhankelijk van de blok golf). Men spreekt hier van een duty cycle of arbeidscyclus dit is in feite de verhouding van het signaal. Men berekent hier de gemiddelde waarde van het signaal. Bij een arbeidscyclus van 50% zal de led dus half zo hard branden. Een gelijkaardig gebruik van de PWM wordt ook toegepast bij het bedienen van stappenmotoren. De spoelen van de motoren zullen dus niet heel de tijd bekrachtigd worden op volledig vermogen maar men zal het vermogen kunnen regelen door de voedingsspanning te bewerken waardoor de stappen verkleind kunnen worden. Men noemt dit ook wel een vermogensregeling. (Wikipedia, 2021)



*Figuur 9: Puls breedte modulatie
Bron: Wikipedia*

5.4 Einde eindeloopschakelaar

De eindeloopschakelaars zijn naast de Coördinaten van de G-Code het enigste input warden van het systeem, maar zeker belangrijk. Ze zorgen er namelijk voor dat de frees gekalibreerd blijft en dat bij eindeloop de motoren stoppen met draaien en niet overbelasten en daardoor te veel stroom trekken. Elke as beschik over een eindeloopschakelaar (X,Y en Z). Ook spelen de eindslag schakelaars een belangrijke rol bij het nullen van het freeshoofd. Als de CNC-Machine gestart wordt zal er een error melding weergegeven worden omdat het stuurprogramma de huidige coördinaten van het systeem niet kent. Hiervoor zal men ofwel het systeem manueel moeten ontgrendelen of kalibreren.

5.4.1 Werking eindeloopschakelaar

Een eindeloopschakelaar is een component dat veel gebruikt wordt bij automatisering van machines om een positie aan te geven of een beveiliging te integreren. Men heeft de mogelijkheid om de eindeloopschakelaar op 2 manieren aan te sluiten. Normaal open en normaal gesloten. Bij een normaal open aansluiting zal er geen signaal doorgelaten worden als de schakelaar niet geactiveerd is maar wel een signaal doorlaten als de schakelaar actief is. Dit geldt visa versa voor een normaal gesloten configuratie. Het signaal wordt gestuurd door het *CNC-shield* en vormt een terugkoppeling. Als de Arduino geen feedback krijgt van het uitgestuurde signaal zal deze aannemen dat de eindeloopschakelaar niet actief is. De eindeloopschakelaar beschik dus over 3 aansluitingen, 1 gemeenschappelijke die we als de negatieve zijde kunnen aanzien en 2 aansluitingen voor het configureren van een normaal open of gesloten systeem.

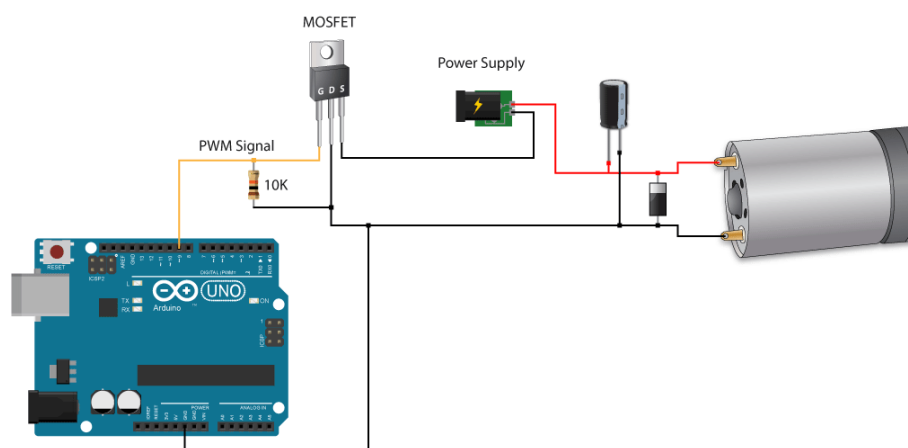
5.5 Toerentalregelaar Frees

Bij het gebruik van een CNC-machine is het gebruikelijk dat men het toerental van de frees kan aanpassen omdat men niet altijd met uniforme materialen werkt. Elk materiaal beschikt over verschillende hardheden en structureren. Daarom is er geopteerd om de draaisnelheid van de frees te kunnen regelen om zo verschillende materialen te kunnen bewerken. Omdat het lezen van de G-code en het sturen van de stappenmotoren 97% van het geheugen van een Arduino uno gebruikt, is er geopteerd voor het gebruik van een 2e Arduino die instaat voor de motorsturing van het freeshoofd.

In eerste instantie is er geopteerd voor een universele motor driver, namelijk de L298N. Deze driver is compatibel met de Arduino uno en de Arduino IDE. De driver kon motoren tot en met 2A voeden. Na enkele testen in de praktijk bleek dat de gebruikte motor 3-4A stroom trekt waardoor de driver snel overbelast werd waardoor temperaturen oplopen en het systeem niet meer functioneerde door een veiligheid van de L298N.

Om toch een sturing te kunnen realiseren is er een systeem ontworpen dat met behulp van een MOSFET het toerental zal regelen.

Een potentiometer zal hierbij een analoog ingangssignaal naar de Arduino sturen die op zijn beurt een PWM-signaal naar de MOSFET stuurt die de spanning naar de motor zal regelen dat in verhouding staat met het toerental van de motor. (Howtomechatronics.com, 2017)



*Figuur 10: Mosfet Motor Control
Bron: howtomechatronics.com*

```

#define enA 9
#define in1 6
#define in2 7
#define button 4
int rotDirection = 0;
int pressed = false;
void setup() {
    pinMode(enA, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(button, INPUT);
    // draairichting instellen
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    Serial.begin(9600);
}
void loop() {
    int potValue = analogRead(A0); // lezen van Potentiometer waarde
    int pwmOutput = map(potValue, 0, 1023, 0, 255); // omzetten van deze waardes
    analogWrite(enA, pwmOutput); // de gelze waarden uitsturen als PWM
    Serial.println(pwmOutput);
    delay(100);
}

```

*Figuur 11: Code Motor sturing
Bron: Victor Smet*

5.6 Voeding

Omdat de Arduino enkel 5V kan leveren met een beperkte stroom, vereist het project een externe voeding die leidt naar de drivers van de stappenmotoren. Hierdoor wordt het mogelijk om een hogere spanning en stroom te leveren (Max Motorstroom). De voeding zal een 230V AC signaal omzetten naar een gelijkspanning van 12V. Deze 12V wordt dan nogmaals opgedeeld naar de drivers op het *CNC-shield* die de spanning en stroom zullen regelen naar de motoren. Een 2e 12 volt voeding is vereist voor het voeden van een DC motor die ervoor zorgt dat de frees bekrachtigd wordt. Ook is het belangrijk dat de motorkring en de stuurkring geïsoleerd zijn om te voorkomen dat de componenten die werken op lage spanning beschadigd worden. De hoofdvoeding in het systeem is ook beveiligd met een noodstop zodat er op elk gegeven moment ingegrepen kan worden en het systeem uitgeschakeld kan worden.

6. Software

6.1 Arduino IDE

De software die gebruikt wordt voor het programmeren van een Arduino heet Arduino IDE wat staat voor Arduino *Integrated Development Environment*. De programmeertaal die hier gebruikt wordt is een afgeleide van C. Het evidente aan deze software is dat er eindeloze hulp terug te vinden is op forums of te vergelijken valt met reeds bestaande projecten.

De software herkent talloze functies die nodig zijn voor het programmeren van een structurele code die geüpload wordt naar de microcontroller op de Arduino. Verder is er een mogelijkheid om zogenaamde bibliotheken toe te voegen die de programmatie mogelijkheden eindeloos uitbreiden. Meestal gaat een component van een externe ontwikkelaar gepaard met een bibliotheek die de Arduino software compatibel maakt. Een bibliotheek die van toepassing is voor het programmeren van een CNC-Machine is GRBL.

6.1.1 Arduino IDE structuur

Om een correcte code te ontwikkelen voor het aansturen van de microcontroller vereist de software een vaste structuur die eenvoudig te lezen is door het Arduino board. De opbouw van deze structuur bestaat uit het definiëren van de gebruikte variabelen, de *void setup* en de *void Loop*.

6.1.2 Definiëren van de variabelen

Eens een code in de IDE is geschreven en geverifieerd door de software kan men deze uploaden naar de microcontroller. Wat de microcontroller als eerste gaat doen is het definiëren van de variabelen. Dit wil zeggen dat in deze fase van het uploaden alle ingangen en uitgangen worden beschreven en een waarde krijgen. afhankelijk van de het type variabele kan dit een binaire waarde zijn zoals hoog of laag maar ook is er de mogelijkheid om een getal als waarde te plaatsen. Afhankelijk van wat er van toepassing is zal er een bepaald voorzetsel geplaatst moeten worden dat bepaald welke waarde hier gebruikt wordt. Een toegewezen waarde kan zowel gelinkt zijn aan een hardware component als een fictieve waarde die gebruikt wordt in de software om bijvoorbeeld een berekening te maken.

(Arduino.cc, 2021)

Tabel 5: Voorbeelden variabelen
Bron: Arduino.nu

byte	Byte bewaart een 8-bit numerieke waarde zonder een decimale punt met een bereik van 0-255.
int	Integers zijn primaire data types om getallen te bewaren zonder een decimale punt een 16-bit waarde met een bereik van 32767 tot -32768
long	Datatype voor erg grote getallen, zonder een decimale punt (een soort uitgebreide integer) een 32-bit waarde met een bereik van 2,147,483,647 tot -2,147,483,648.

6.1.3 Void Setup

In de setup gaat men bepaalde functies uitvoeren die maar eenmalig zijn voor het systeem opstart, reset of upload. De meeste functies zijn toewijzen van statussen (HIGH, LOW) of start waardes aan het systeem geven. Ook gaat men bepaalde componenten hier opstarten of een communicatie openen. (Serial begin...). Voorbeelden hiervan zijn

- Definiëren van in of outputs
- Starten van de seriële monitor
- Instellen van de communicatie snelheid (*baud rate*)
- Eventuele I2C protocollen opstarten

```
void setup() {
  Serial.begin(9600);           // stel de seriële monitor in

  stepper.setMaxSpeed(1000.0);  // stel de maximale motorsnelheid in
  stepper.setAcceleration(100.0); // stel de acceleratie in
  stepper.setSpeed(200);        // stel de huidige snelheid in

  stepper.moveTo(degToSteps(90)); // geef motor instructie om de motor
```

Figuur 12: Void Setup
Bron: Victor Smet

6.1.4 Void Loop

In de loop gaat men alles implementeren wat eindeloos herhaald wordt tot de Arduino uitgezet of gereset word. De loop gaat simpelweg vergelijkingen maken en op basis van de uitkomst van deze vergelijkingen dergelijke acties uitvoeren of juist stoppen. Omdat een code zo degelijk groot kan worden gaat men verwijzen naar bepaalde bibliotheken die ervoor zorgen dat een enkele regel verwijst naar een externe code waar dit bepaalde commando leesbaar wordt voor de microcontroller.

```
void loop() {  
  
    // Als de stepper niet beweegt en geen afstand meer hoeft af te leggen  
    if (!stepper.isRunning() && stepper.distanceToGo() == 0) {  
  
        // Beweeg de stepper naar het aantal graden op de positie stepIndex van steps[]  
        stepper.moveTo(degToSteps(steps[stepIndex]));  
  
        // Verhoog de index als de beweging klaar is  
        stepIndex++;  
  
        // Als we alle posities in steps[] gehad hebben  
        if (stepIndex > stepsCount) {  
  
            // Begin overnieuw bij 0  
            stepIndex = 0;  
        }  
    }  
  
    stepper.run();  
}
```

*Figuur 13: Void Loop
Bron: Victor smet*

6.2 GRBL

GRBL is een bibliotheek die het voor Arduino mogelijk maakt om op een eenvoudige manier G-Code te communiceren via een COM poort van de computer naar de microcontroller. Hierdoor is de code die gebruikt wordt in dit project heel beperkt en wordt enkel de GRBL bibliotheek gedefinieerd. In de GRBL bibliotheek kunnen we wel aanpassingen maken die specifiek anders zijn voor elke machine. Bijvoorbeeld het aantal assen die nodig zijn voor het aansturen van het systeem en de volgorde van het nullen etc... De aanpassingen worden gemaakt in een tekst document genaamd config.h (configuratie). Eens hier een aanpassing wordt gemaakt is het belangrijk dat dit document wordt opgeslagen en de code opnieuw wordt geüpload naar de Arduino aangezien deze gelinkt is aan de bibliotheek. Een ander voordeel aan het GRBL format is dat het volledig opensource en gratis is.

In figuur 13 kan je duidelijk zien waar de nullen cycli gedefinieerd staat voor 3 assige machines. Deze is namelijk aangepast om na het nullen van de Z-as eerst de X-as te nullen en dan de Y-as. In originele toestand werden na de Z-as X en Y tezamen genuld wat voor enkele problemen zorgde. (Wikipedia.fr, 2016)

```
// NOTE: Defaults are set for a traditional 3-axis CNC machine. Z-axis first to clear, followed by X & Y.
#define HOMING_CYCLE_0 (1<<Z_AXIS) // REQUIRED: First move Z to clear workspace.
// #define HOMING_CYCLE_1 ((1<<X_AXIS)|(1<<Y_AXIS)) // OPTIONAL: Then move X,Y at the same time.7
#define HOMING_CYCLE_1 (1<<X_AXIS)
#define HOMING_CYCLE_2 (1<<Y_AXIS)
// #define HOMING_CYCLE_2 // OPTIONAL: Uncomment and add axes mask to enable

// NOTE: The following are two examples to setup homing for 2-axis machines.
// #define HOMING_CYCLE_1 ((1<<X_AXIS)|(1<<Y_AXIS)) // NOT COMPATIBLE WITH COREXY: Homes both X-Y in one cycle.

// #define HOMING_CYCLE_0 (1<<X_AXIS) // COREXY COMPATIBLE: First home X
// #define HOMING_CYCLE_1 (1<<Y_AXIS) // COREXY COMPATIBLE: Then home Y

// Number of homing cycles performed after when the machine initially jogs to limit switches.
// This help in preventing overshoot and should improve repeatability. This value should be one or
// greater.
#define N_HOMING_LOCATE_CYCLE 1 // Integer (1-128)
```

Figuur 14: Config.h
Bron: Victor Smet

6.3 CAD Software

Voor het realiseren van een G-code is het meest eenvoudige om de G-code manueel uit te schrijven en de coördinaten manueel in te vullen. Eens de gewenste werkstukken gecompliceerder worden, is het uitschrijven van een G-code zeer tijdrovend en niet eenvoudig. Hiervoor wordt meestal een CAD software voor aangeraden. CAD staat voor *computer-aided design*. Dit wil zeggen dat men aan de hand van een computer software, voorwerpen of apparaten gaat ontwerpen die enkel theoretisch zijn maar wel realistisch en accuraat. Voor de meeste CAD-software geldt ook de mogelijkheid om het ontwerp om te zetten in een G-code die enorm uitgebreid is. Afhankelijk van het gebruikte CAD-programma zullen er opties zijn zoals in hoeveel lagen het werkstuk gemaakt moet worden. Verandering van verspanend gereedschap tijdens de operatie om dergelijke bewerkingen uit te voeren. CAD-software wordt in zo goed als alle ontwikkelingsprocessen gebruikt. Enkele voorbeelden hiervan zijn.

- AutoCAD
- Inkscape
- FUSION 360
- FreeCAD

Voor het gebruik van deze CNC-machine zijn er enkele sketches ontwikkeld in het CAD-programma Inkscape. Dit is een van de meest eenvoudige programma's en zeer gebruiksvriendelijk. Voor het genereren van een G-code is er echter wel een extra plug-in vereist die dit mogelijk maakt. Eens een G-code gegenereerd is, dient deze enkel nog geüpload te worden naar de software die gaat communiceren met de microcontroller wat in dit geval UGS is (*Universal G-code Sender*).

Vierkant 30x30

```
G00 X0 Y0 F70
G01 Z-0.5 F5
G01 X0 Y3 F5
G01 X3 Y0
G01 X0 Y-3
G01 X-3 Y0
G00 Z1 F70
M30
%
```

Figuur 15: voorbeeld G-code 2
Bron: Victor Smet

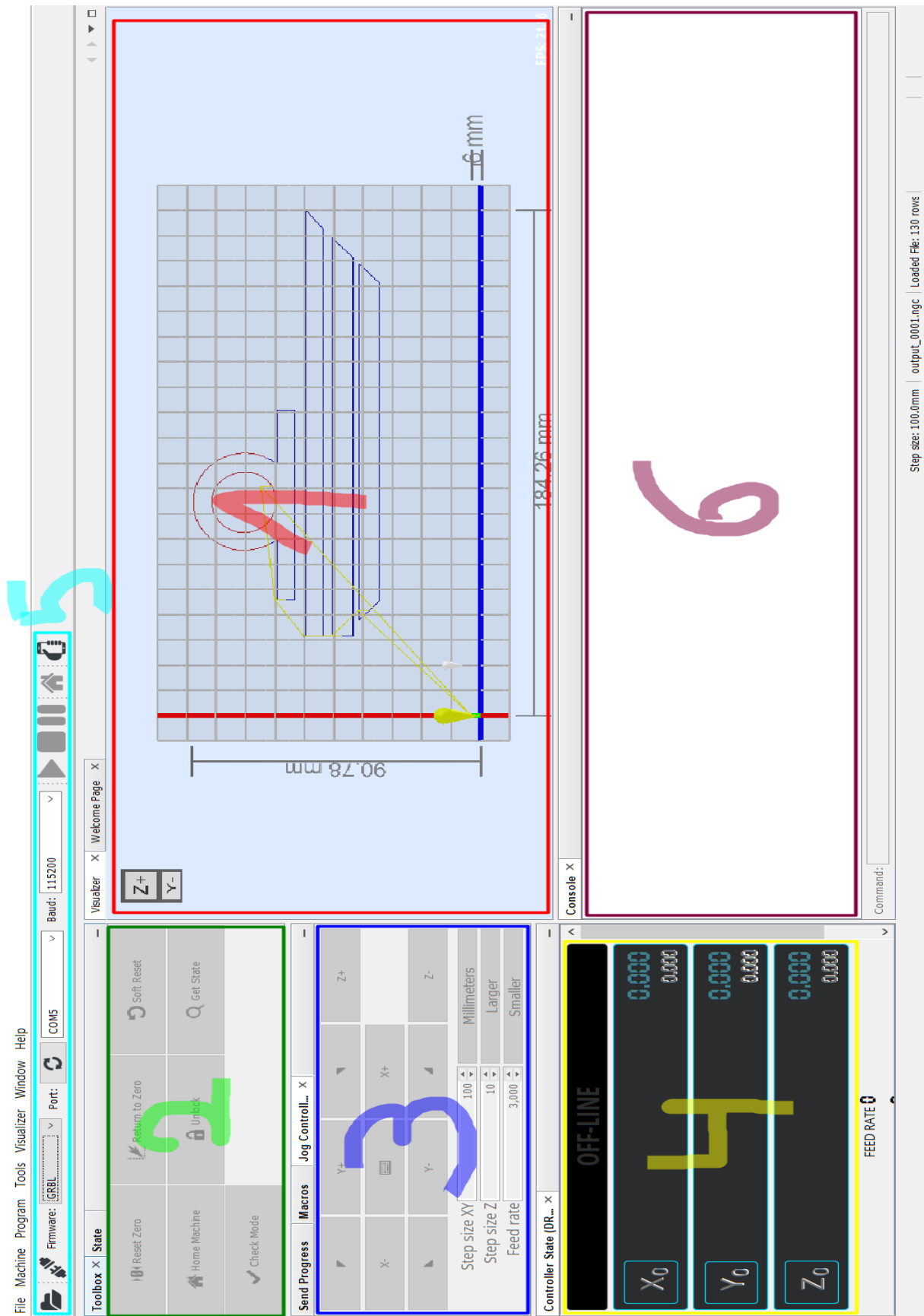
6.4 Universal G-code sender

Universal G-code sender is een open source platform die het sturen van G-codes via GRBL ondersteunt. Via dit platform wordt de G-code geüpload en krijgt men ook al een voorbeeld van het ontwerp te zien. Deze software wordt niet enkel gebruikt voor het uitsturen van G-code maar ook voor het kalibreren van de machine door middel van een geïmplementeerde setup wizard die stap voor stap de belangrijke kalibratiepunten afgaat.

Ook al de feedback van het systeem wordt gelezen door deze software en weergegeven op het scherm. Dit gaat van de huidige kalibratie instellingen van de machine tot de live opvolging van het freeshoofd in het werkstuk. op fig.15 en tabel 6. zien we de voorstelling van de UGS software.

*Tabel 6: bijlage figuur 15
Bron: Victor Smet*

1 Visualizer	In dit deel wordt het model van het werkstuk weergegeven tegenover de assen waarop de frees zich op huidig ogenblik bevindt.
2 Toolbox	In dit vak kunnen de commando's staan die men kan uitvoeren en sturen naar het toestel.
3 Jog Controller	De jog controller laat ons toe de nauwkeurigheid van het toestel aan te passen alsook een manuele bediening uit te voeren.
4 Controller State	Door de data in vak 4 te bekijken kunnen we zien hoe ver de frees al bewogen is tegenover de vorige positie en de nul positie van de machine.
5 connection & configuration	Bij opstart van het toestel gaan we met deze instellingen de seriële connectie met de microcontroller openen en de communicatiesnelheid bepalen.
6 console	In de console vinden we de uitgestuurde G-code terug alsook de instellingen van de microcontroller.

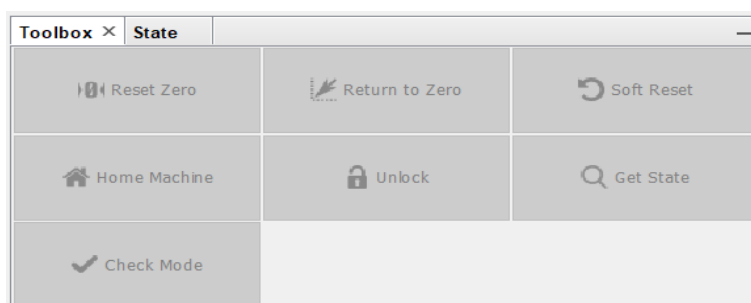


Figuur 16: Voorbeeld UGS
Bron: Victor Smet

6.4.1 UGS Toolbox

Zoals eerder vermeldt vinden we in de toolbox van UGS een aantal belangrijke commando's terug die we kunnen uitvoeren om het toestel te kalibreren of bij opstarten van het toestel. Aangezien deze functies vrij tot zeer belangrijk zijn voor de basiswerking van het toestel worden deze verder in detail besproken. (DIY machining, 2016)

- Reset Zero: als men deze functie activeert gaat men een nieuw theoretisch nulpunt instellen op de positie waar de frees zich op huidig ogenblik bevindt.
- Return to Zero: eens een theoretisch nulpunt is bepaald, kan men op elk ogenblik het toestel naar deze positie sturen op voorwaarde dat het toestel correct gekalibreerd is.
- Soft Reset: door het gebruik van een soft reset gaat het nulpunt alsook de instellingen van de microcontroller terug naar de originele waarde en wordt het theoretisch nulpunt de huidige positie van de frees.
- Home machine: door deze knop te activeren wordt het toestel genuld. Dit wil zeggen dat de frees zal bewegen naar zijn werkelijk beginpunt, aangeduid door de eindeloopschakelaars.
- Unlock: unlock zal ervoor zorgen dat het toestel vrij is om te bedienen met een manuele input. Als dit niet het geval is, zal de software in alarm status gaan waardoor het niet mogelijk is de stappenmotoren de bewegen.
- Get State: zoals letterlijk vertaald zal door het activeren van de get state knop de positie van het toestel geprint worden in de console.
- Check Mode: als er zich een foutmelding in het systeem bevindt, gaat de optie check mode deze foutmelding weergeven in de console.



*Figuur 17: Toolbox
Bron: Victor Smet*

6.4.2 Zero en nullen

Om mogelijke verwarring uit te sluiten is het belangrijk het verschil tussen nul positie en een zero positie duidelijk te stellen. Bij een nul positie spreken we over een vaste positie op het toestel die onveranderbaar is. (enkel door het verplaatsen van de eindslag schakelaars) In het Engels noemen we dit de home positie of thuis positie. Een zero positie is verplaatsbaar en kan men vergelijken met een theoretische nul positie. De zero zal ook enkel werken als het toestel correct gekalibreerd is in tegenstelling tot de thuis positie. Door gebruik te maken van de reset zero zoals eerder besproken, kan men deze positie plaatsen waar men maar wil zolang het binnen het werkgebied van het toestel blijft. Het werk oppervlak bedraagt 500x500 mm

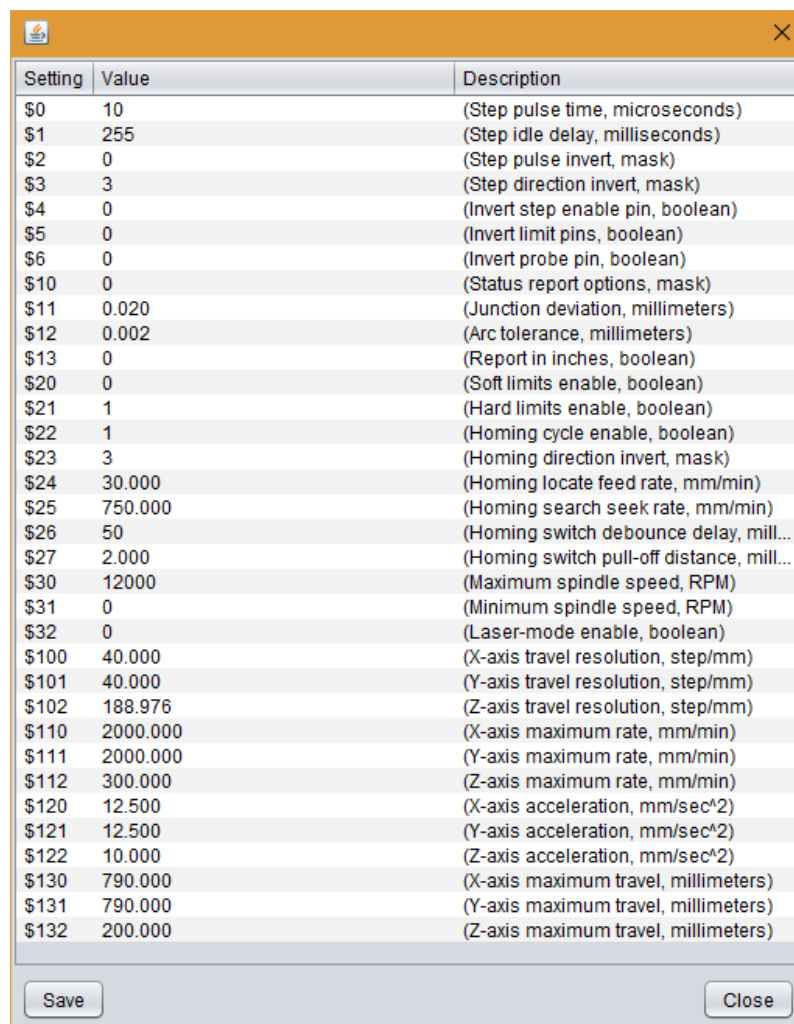
6.4.3 Hard limit and soft limit

Hard limit en soft limit zijn minstens zo belangrijk als het begrijpen van de nul en zero positie. Een hard limit is meer gelinkt aan het nullen van het toestel en wil zeggen dat de machine nooit over deze limieten kan gaan. De harde limieten bevinden zich aan de uiteinden van het werkvlak en worden begrensd door de eindloop schakelaars.

Een soft limiet daarentegen kan men ook weer plaatsen waar men wenst zolang het toestel correct gekalibreerd is en de gewenste limieten niet groter zijn dan de hard limiet en zich ook nog in het werkvlak begeven. Een soft limiet is dus een Theoretische grens die bepaald wordt door de afstand die het toestel al heeft afgelegd. terwijl een hard limiet geen rekening houdt met de voorgaande positie van het toestel, maar enkel geactiveerd wordt als de eindloop actief is.

6.4.4 UGS Firmware settings

Bij opstarten van het toestel gaat UGS in eerste instantie de instellingen of *firmware settings* van de microcontroller en GRBL uitlezen. Deze zullen dan weergegeven worden in de console van de *universal G-code sender*. Men kan deze instellingen ook op elk moment opvragen door het commando \$\$ in te voeren in de console. Ook kan men via de console of het optie menu onder machine de setupwizard aanpassen door een gewenste waarde te schrijven achter het juiste adres. bv: \$100= 40.00 (nu zal de resolutie aangepast zijn naar 40 stappen per mm). Het gebruik van volgende commando's is belangrijk voor het kalibreren van het toestel. (willysgaragenorway.no, 2017)



Setting	Value	Description
\$0	10	(Step pulse time, microseconds)
\$1	255	(Step idle delay, milliseconds)
\$2	0	(Step pulse invert, mask)
\$3	3	(Step direction invert, mask)
\$4	0	(Invert step enable pin, boolean)
\$5	0	(Invert limit pins, boolean)
\$6	0	(Invert probe pin, boolean)
\$10	0	(Status report options, mask)
\$11	0.020	(Junction deviation, millimeters)
\$12	0.002	(Arc tolerance, millimeters)
\$13	0	(Report in inches, boolean)
\$20	0	(Soft limits enable, boolean)
\$21	1	(Hard limits enable, boolean)
\$22	1	(Homing cycle enable, boolean)
\$23	3	(Homing direction invert, mask)
\$24	30.000	(Homing locate feed rate, mm/min)
\$25	750.000	(Homing search seek rate, mm/min)
\$26	50	(Homing switch debounce delay, mill...
\$27	2.000	(Homing switch pull-off distance, mill...
\$30	12000	(Maximum spindle speed, RPM)
\$31	0	(Minimum spindle speed, RPM)
\$32	0	(Laser-mode enable, boolean)
\$100	40.000	(X-axis travel resolution, step/mm)
\$101	40.000	(Y-axis travel resolution, step/mm)
\$102	188.976	(Z-axis travel resolution, step/mm)
\$110	2000.000	(X-axis maximum rate, mm/min)
\$111	2000.000	(Y-axis maximum rate, mm/min)
\$112	300.000	(Z-axis maximum rate, mm/min)
\$120	12.500	(X-axis acceleration, mm/sec^2)
\$121	12.500	(Y-axis acceleration, mm/sec^2)
\$122	10.000	(Z-axis acceleration, mm/sec^2)
\$130	790.000	(X-axis maximum travel, millimeters)
\$131	790.000	(Y-axis maximum travel, millimeters)
\$132	200.000	(Z-axis maximum travel, millimeters)

Figuur 18: firmware settings

Bron: 3dtekforum.com

7. Kalibreren van de CNC-machine

Zoals eerder vermeldt is kalibreren van de CNC-machine een belangrijke vereiste voor de meeste functies. Het kalibreren doet men via de software door de waardes van de firmware aan de passen waarbij men de limieten kan aanpassen en bepalen alsook de resolutie van het toestel en talloze andere toepassingen. (DIY Machining, 2016)

1. \$: het printen van huidige instellingen in de console
2. \$0: step pulse length (μsec), bepaalt de lengte van de pulsen die naar de stappenmotoren worden gestuurd. Dit is bij voorkeur zo kort mogelijk maar hangt af van het type stappenmotor dat gebruikt wordt. Een goede gemiddelde waarde is $10\mu\text{sec}$
3. \$1: step idle delay (msec). Dit is de tijd waarbij de motoren bekrachtigd blijven na een actie te hebben uitgevoerd. Als men deze waarde naar 255 zet zullen de motoren altijd bekrachtigd blijven en dus niet verplaatsbaar zijn.
4. \$2: step pulse configuration. Door rekening te houden met een bepaalde tabel 7. kan men de assen draaien indien nodig voor bijvoorbeeld de configuratie omgekeerde machines of foutief geïnstalleerde stappen motoren. Een andere mogelijkheid om dit te veranderen is de aansluiting van de stappenmotoren op de drivers om te draaien.

Tabel 7: Step Pulse configuration
Bron: DIYmachining.com

Setting Value	Reverse X	Reverse Y	Reverse Z	Setting Value	Reverse X	Reverse Y	Reverse Z
0	NO	NO	NO	4	NO	NO	YES
1	YES	NO	NO	5	YES	NO	YES
2	NO	YES	NO	6	NO	YES	YES
3	YES	YES	NO	7	YES	YES	YES

5. \$3: Axis direction. Draait de richting van de alle assen om zonder de bedrading aan te passen.
6. \$4: Step enable invert. Om de drivers in te schakelen is er een signaal vereist. In geval van een incompatibel CNC-shield maakt deze instelling het mogelijk het enable signaal te inverteren.

7. \$5: limit pins invert. Deze instelling verwijst naar de configuratie van de eindeloopschakelaars. In een normaal open configuratie zullen deze een pull down weerstand gebruiken die zich op het CNC-shield bevinden. Indien deze zijn aangesloten met een normaal gesloten configuratie kan met \$5 de configuratie omgedraaid worden. zodat de pull down weerstand op het CNC-shield altijd gebruikt wordt en er geen signaal storingen plaatsvinden.
8. \$6: Probe Pin Invert. Zoals de instelling van de eindeloop schakelaars is het ook mogelijk om voor de Z-as een probe te installeren in plaats van een eindeloopschakelaar. Door het inverteren van \$6 kan men er hier ook voor zorgen dat de pull down weerstand altijd in gebruik is.
9. \$10: Status report. Een status report zal communicatie leggen met de microcontroller en data opvragen zoals de positie van het toestel. Deze resultaten zullen ook uitgeprint worden in de console.
10. \$11: Junction deviation. Als er 2 bewegingen tegelijk worden uitgevoerd door 2 verschillende assen, zal de snelheid trager worden om te vermijden dat er stappen overgeslagen worden door de motoren waardoor de accuraatheid van het systeem verminderd.
11. \$12: Arc tolerance. Zoals de junction deviation zal Arc tolerance de accuraatheid verbeteren maar in plaats van dit te doen bij hoeken zal deze hier rekening mee houden bij grote bochten.
12. \$13: Feedback Units. Omdat GRBL opensource is en dus door iedereen gebruikt kan worden, heeft men de mogelijkheid om de eenheden om te zetten in inches. als \$13=1 zal de eenheid in inches staan als deze waarde 0 is, zal de eenheid in mm staan.

13. \$20: Soft limits enable. Door deze waarde hoog of laag te definiëren, kunnen we bepalen of het toestel rekening gaat houden met de opgegeven soft limieten als deze zijn opgegeven.
14. \$21: Hard limits enabled. Door deze waarde hoog of laag te definiëren, kunnen we bepalen of het toestel rekening gaat houden met de opgegeven hard limieten als het systeem beschikt over eindeloopschakelaars.
15. \$22: Homing cycle enabled. Door deze waarde aan te passen kunnen we bepalen of de homing cycle uitgevoerd wordt bij het opstarten van het systeem om een referentie te krijgen van de toestand van het toestel.
16. \$23: homing cycle direction. Door de eindeloopschakelaars aan te sluiten op de negatieve assen zal het nullen altijd naar de negatieve kant van het werkvlak uitlopen. Dit kan men omdraaien door de waarde van \$23 laag te schrijven.
17. \$24: homing feedrate (mm/min). Als het toestel de eindslag schakelaars 1x geactiveerd heeft, zal het deze actie herhalen om de juiste nul positie te vinden. De homing feedrate zal hierbij de snelheid van het toestel bepalen voor het tweede keer activeren van de eindeloopschakelaars.
18. \$25: Homing seekrate (mm/min). Voor het eerste maal activeren van de eindeloopschakelaars tijdens het nullen zal de homing seekrate de snelheid bepalen van deze actie.
19. \$26: Homing Debounce (msec). Na het nullen van het toestel is er een mogelijkheid om een vaste afstand van de eindeloopschakelaars te nemen waardoor het toestel niet tegen de eindeloopschakelaars aanzit na het einde van het nullen. De homing debounce bepaalt hoe lang de tijd bedraagt tussen het nullen en het weg verplaatsen van de eindeloopschakelaars.

20. \$27: Homing Pull-off (mm): Deze waarde in millimeter zal de afstand tussen de eindeloopschakelaars en het freeshouder bepalen na het nullen van het toestel.
21. \$100: X Steps/mm: Afhankelijk van de resolutie zal deze instelling het aantal stappen bepalen dat de motor moet uitvoeren voor het freeshouder 1 millimeter te doen verplaatsen tegenover de X-AS. Men kan dit bepalen door volgende formule:
(steps/mm = stappen per toer * microsteps (resolutie van de driver) / millimeter per toer).
22. \$101: Y Steps/mm: Afhankelijk van de resolutie zal deze instelling het aantal stappen bepalen dat de motor moet uitvoeren voor de freeshouder 1 millimeter te doen verplaatsen tegenover de Y-AS. Dezelfde formule is hier van toepassing.
23. \$102: Z Steps/mm: Afhankelijk van de resolutie zal deze instelling het aantal stappen bepalen dat de motor moet uitvoeren voor de freeshouder 1 millimeter te doen verplaatsen tegenover de Z-AS. Dezelfde formule is hier van toepassing.
24. \$110: X-max rate (mm/min). Deze instelling zal de maximale snelheid of feedrate van de X-as bepalen. Als deze waarde te hoog is, zal de motor niet kunnen volgen en vastlopen. Indien de waarde wel laag genoeg is om te kunnen volgen, moet men nog steeds rekening houden met het materiaal waarmee men werkt. Als dit een hard materiaal is, kan men de feedrate best verlagen.
25. \$111: Y-max rate (mm/min). Deze instelling zal de maximale snelheid of feedrate van de Y-as bepalen.
26. \$112: Z-max rate (mm/min). Deze instelling zal de maximale snelheid of feedrate van de Z-as bepalen.

27. \$120: X-max acceleration (mm/sec²). Door deze waarde aan te passen zal de snelheid vanuit stilstand tot de maximum feedrate bepaald worden voor de X-as.
28. \$121: Y-max acceleration (mm/sec²). Door deze waarde aan te passen zal de snelheid vanuit stilstand tot de maximum feedrate bepaald worden voor de Y-as.
29. \$122: Z-max acceleration (mm/sec²). Door deze waarde aan te passen zal de snelheid vanuit stilstand tot de maximum feedrate bepaald worden voor de Z-as.
30. \$130: X - Max travel (mm). Deze waarde zal de soft limit van de X-as bepalen en bepalen hoeveel afstand het toestel kan afleggen alvorens deze in alarm status gaat.
31. \$131: Y - Max travel (mm). Deze waarde zal de soft limit van de Y-as bepalen en bepalen hoeveel afstand het toestel kan afleggen alvorens deze in alarm status gaat.
32. \$132: Z - Max travel (mm). Deze waarde zal de soft limit van de Z-as bepalen en bepalen hoeveel afstand het toestel kan afleggen alvorens deze in alarm status gaat.

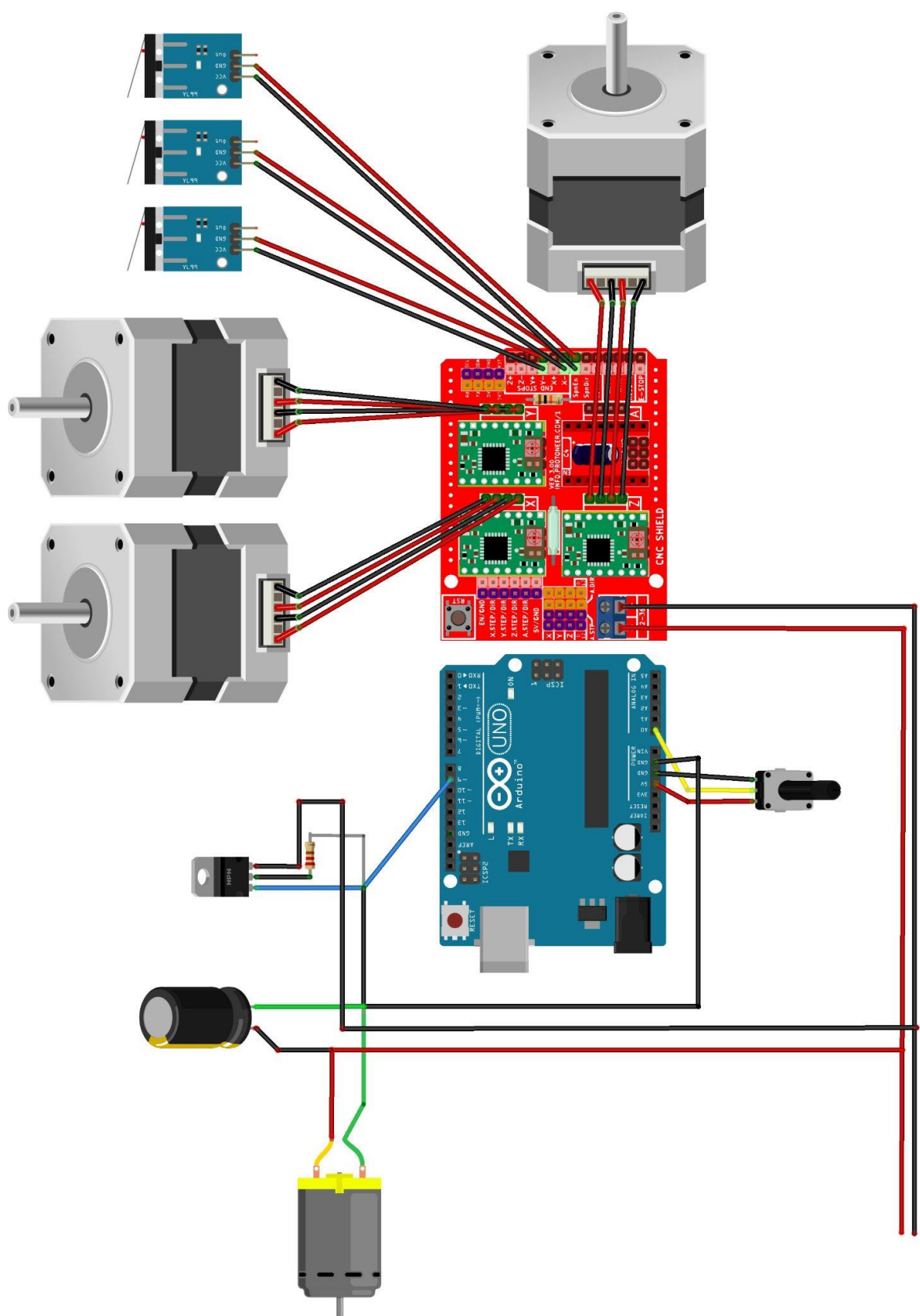
8. Aansluiting CNC-machine

8.1 Aansluiting stappen motoren

Zoals eerder vermeld zal dit project gebruik maken van 2 Arduino's. Een Arduino zal instaan voor het bedienen van de stappenmotoren en altijd verbonden zijn met de computer om de communicatie tussen UGS en de microcontroller te kunnen realiseren. Op deze Arduino bevindt zich het CNC-shield waarop de 3 stappenmotoren zijn aangesloten alsook de eindeloopschakelaars van elke as. De Eindeloopschakelaars zijn aangesloten op X-, Y- en spindel enable. Door een fout in ontwikkeling is de aansluiting van de eindeloopschakelaar van Z- verplaatst naar spindel enable. Vandaar deze onlogische aansluiting. Verder zal er een voeding van 230AC-12DC aanwezig zijn de het CNC-shield zal voeden en op zijn beurt ook de stappenmotoren. (victor smet, 2021)

8.2 Aansluiting motor sturing

Voor het regelen van de motor die de frees zal bedienen zal er een 2e Arduino aanwezig zijn die aangesloten is op een potentiometer. Deze potentiometer zal ervoor zorgen dat men het toerental kan regelen en een analogoog signaal sturen naar de Arduino op pin A0. De Arduino zal aan de hand van de code in fig. 10 een PWM signaal uitsturen naar een mosfet die aangesloten is op de gemeenschappelijke 12 volt voeding. Dit PWM signaal zal het toerental van de motor gaan bepalen. De Arduino die gebruikt wordt, zal niet gevoed worden door de computer maar een externe voeding krijgen via de 12V die beschikbaar is in de CNC-machine. (Victor smet, 2021)



fritzing

Figuur 19: Aansluiting CNC-machine
Born: Victor smet

9. Het gebruik van CNC in de toekomst van de scheepvaart

Rond het gebruik van 3D-print technieken is al een hele tijd speculatie op het gebied van de scheepvaart. Men was tot op kort nog niet zeker of het mogelijk was om gecompliceerde stukken die moeten voldoen aan bepaalde vereisten te printen. RAMLAB een Nederlands startup bedrijf dat sponsoring krijgt van Port of Rotterdam en nauw samenwerkt met de Universiteit van Delft heeft in 2017 het tegendeel bewezen. RAMLAB heeft een schroef ontwikkeld en gefabriceerd door het gebruik van additieve 3D-print technieken. Deze ontwikkeling heeft veel publiciteit gemaakt en veel interesse opgewekt om gebruikte technieken grootschaliger te gaan gebruiken. De grootste bedreiging die de ontwikkeling nog tegenhoudt op grote schaal is de huidige bevoorraadingsketen. In tegendeel zijn de voordelen wel bewezen zoals de snelle afwerking en mogelijkheid om meerdere stukken exact te reproduceren zonder ervaren personeel.

Volgens Wilhelmsen is de grote kost van het vervangen van onderdelen momenteel het opslagen, vervoeren en logistiek het probleem. De oplossing hiervoor zou het 3D-printen van onderdelen over verschillende “HUB’s” over heel de wereld waar benodigde onderdelen vanuit deze “HUB’s” rechtstreeks naar de schepen kunnen gaan waar nodig.

Het analyseren van onderdelen zou hiervoor ook een belangrijke rol spelen, hiermee zal er een betere replica gemaakt kunnen worden door gebruik van verschillende materialen.

Omdat er verschillende materialen gebruikt worden is het ook niet direct mogelijk om de vervangstukken aan boord van het schip te maken.

Wilhelmsen spreekt van een opslag van een 80 tal verschillende materialen om optimale werkstukken te bekomen. Een 2^e vereisten is het personeel dat de machine moet bedienen zal zeer ervaren moeten zijn met een specifieke opleiding in machine engineering.

Door het bewijs van een complexe productie van een propeller hoopt Wilhelmsen naar de toekomst op grootte schaal onderdelen te ontwikkelen.

Conclusie

Het project is volledig functioneel met alle aspecten besproken in deze scriptie. De CNC-machine is in staat om zo goed als elke G-code geprogrammeerd door een CAD software te kunnen realiseren zolang deze binnen de capaciteit van de machine blijft. Het is mogelijk om werkstukken te fabriceren met een hoogte tot 50mm en een lengte en breedte van 400mm indien de materiaalkeuze bestaat uit een zacht materiaal zoals PDF, PVC en zelfs aluminium als men goed rekening houdt met de *feed rate* van het toestel.

Bij het gebruik van het toestel moet men wel rekening houden dat het nullen van de Z-as in het oog gehouden wordt om contact met het werkstuk te vermeiden. Naar de toekomst toe kan de eindeslag schakelaar verplaatst worden of een ander type eindeslag beveiliging geïnstalleerd worden.

Hierbij wil ik nog eens mijn promotor Rik Floren bedanken voor de hulp en ander kijk naar het project. Alsook Vincent Leysen voor het projectidee dat mij enorm aansprak.

Het project heeft me een andere kijk gegeven op het produceren van werkstukken die in het verleden manueel gefabriceerd moesten worden met een draaibank en frees machine.

De CNC-machine staat nu ter beschikking van de Hogere Zeevaartschool Antwerpen om naar de toekomst toe werkstukken te realiseren die mogelijks gebruikt worden voor andere projecten en afstudeer werken.

Bibliografie

Arduino Mega 2560 Rev3 | Arduino Official Store. (z.d.). Geraadpleegd 31 mei 2021, van

<https://store.arduino.cc/arduino-mega-2560-rev3>

Arduino Mega Tutorial—Pinout and Schematics. Mega 2560 Specifications. (2018, 8

februari). *Electronic Circuits and Diagrams-Electronic Projects and Design*.

Geraadpleegd van <https://www.circuitstoday.com/arduino-mega-pinout-schematics>

Arduino Reference—Arduino Reference. (z.d.). Geraadpleegd 31 mei 2021, van

<https://www.arduino.cc/reference/en/>

Arduino Uno Rev3 | Arduino Official Store. (z.d.). Geraadpleegd 31 mei 2021, van

<https://store.arduino.cc/arduino-uno-rev3>

C, J. O. (2011). *Jc S C S*. 4.

Computer numerical control. (2020, 20 mei). *Wikipedia*. Geraadpleegd van

https://nl.wikipedia.org/w/index.php?title=Computer_numerical_control&oldid=56316232

Dejan. (2017, 8 augustus). L298N Motor Driver—Arduino Interface, How It Works, Codes,

Schematics. *HowToMechatronics*. Geraadpleegd van

<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>

Dejan. (2020, 6 mei). G-code Explained | List of Most Important G-code Commands.

HowToMechatronics. Geraadpleegd van <https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/>

First DIY CNC build (part 12)—Configure \$\$ GRBL on ARDUINO uno TUTORIAL. (2017,

14 oktober). *Willys Garage Norway*. Geraadpleegd van

<http://willysgaragenorway.no/2017/10/14/first-diy-cnc-build-part-12-configure-grbl-on-arduino-uno-tutorial/>

Grbl. (2020, 6 juni). *Wikipédia*. Geraadpleegd van

<https://fr.wikipedia.org/w/index.php?title=Grbl&oldid=171738367>

GRBL - Motion Control for Makers. (2016, 13 augustus). *DIY Machining*. Geraadpleegd van

<https://diymachining.com/grbl/>

Grbl/grbl. (2021). C, Grbl CNC controller. Geraadpleegd van <https://github.com/grbl/grbl>

Microcontroller. (2021, 18 januari). *Wikipedia*. Geraadpleegd van

<https://nl.wikipedia.org/w/index.php?title=Microcontroller&oldid=58050500>

Pulsbreedtemodulatie. (2021, 23 januari). *Wikipedia*. Geraadpleegd van

<https://nl.wikipedia.org/w/index.php?title=Pulsbreedtemodulatie&oldid=58102166>

Shrivastava, S., Rawat, J., & Agrawal, A. (2012). Controlling DC Motor using

Microcontroller (PIC16F72) with PWM. *International Journal of Engineering Research*, 4.

Stappenmotor. (2021, 19 maart). *Wikipedia*. Geraadpleegd van

<https://nl.wikipedia.org/w/index.php?title=Stappenmotor&oldid=58535828>

Stappenmotor—STEM wiki. (z.d.). Geraadpleegd 31 mei 2021, van

<http://wiki.stematschool.be/readonly/index.php?title=Stappenmotor>

Virgala, I., Kelemen, M., Gmitterko, A., & Lipták, T. (2015). Control of Stepper Motor by

Microcontroller. *Journal of Automation and Control*, 5.

W.jones, D. (2004). Stepping Motors Fundamentals. *Stepping Motors Fundamentals*, 22.

Mission Zero – ABB – Power sustainability

3D printing. Parts of a bigger future, 48.

Bijlagen

1. Control of Stepper Motor by Microcontroller
2. GRBL Settings - Pocket Guide
3. 3-Axis CNC/Stepper Motor Shield for Arduino
4. Controlling DC Motor using Microcontroller (PIC16F72) with PWM