



Aansturing freesbank

Stage verslag
Sens2Sea

Olaf Goudriaan

Geert Mosterdijk
TINSTG05
Wessel Oele, Renee van Dorn
1071349
2025 - 2026

Contents

1 Context	2
1.1 Het bedrijf	2
1.2 Opdracht	2
2 Analyse	4
2.1 Huidige situatie	4
2.2 Sensor	5
2.3 Microcontroller	5
2.4 Requirements	6
3 Ontwerpen	7
3.1 Software ontwerpen	7
3.2 Elektrisch schema	9
4 Realisatie	9
4.1 Mechanica	9
4.2 Software	10
4.3 Test Rapport	13
5 Conclusie	13
6 Aanbevelingen	13
7 Bronnen	15
7.1 Datasheets	15
A Encoder onderzoek	16
A.1 Meet methodes	16
A.2 Encoder soorten	16
B elektrische schema's	19
C Afbeeldingen van de opstelling	21
D Testrapport	23
D.1 G-Code ontvangen en uitvoeren	23
D.2 X-richting bewegen	23
D.3 Y-richting bewegen	24
D.4 Z-richting bewegen	25
D.5 X-richting meten	25
D.6 Y-richting meten	26
D.7 Z-richting meten	26
D.8 Modulair	27
D.9 Conclusie	27

Voorwoord Ik dank hierbij iedereen die heeft geholpen bij mijn stage en de leuke stageopdracht. Ik ben veel gegroeid in dit half jaar in hard- en soft skills. Ik heb nieuwe ervaringen opgedaan, ben op plekken geweest waar je normaal niet kan en zou komen. En ik heb het gezellig gehad met de collega's. Ook heb ik veel nieuwe dingen geleerd zoals Freecad en regeltechniek. Daarnaast ben ik door het toepassen van de wiskundige formules beter geworden in wiskunde en natuurkunde. Ook zijn er onderdelen binnen gekomen die defect waren, een goede les om met tegenslag om te gaan. En veder denk ik een mooi prototype neergezet te hebben.

1 Context

1.1 Het bedrijf

De stage heeft plaatsgevonden bij het bedrijf Sens2Sea. Dit is een klein bedrijf dat gevestigd is in Rotterdam op een kavel in het Innovation Dock op het RDM-terrein. Sens2Sea ontwikkelt radartecnologieën voor sectoren zoals de scheepsvaart en andere maritieme sectoren. Het bedrijf wordt geleid door Geert Mosterdijk. Het bedrijf heeft maar een paar medewerkers. Deze zijn gespecialiseerd in diverse vakgebieden zoals elektronica en software. Verder lopen er bij Sens2Sea een aantal studenten van verschillende opleidingen stage. Het bedrijf werkt voor projecten samen met andere bedrijven. Deze bedrijven zijn tevens de stakeholders voor de projecten. Voor dit bedrijf geldt het volgende:

1. Sens2Sea. De opdracht is voor Sens2Sea, dit bedrijf bepaalt welke beslissingen gemaakt worden. Dit bedrijf heeft mensen in dienst die expertise hebben op gebied van elektronica en software. (Hoge invloed en hoog belang)
2. PKMarine. Dit bedrijf werkt samen met Sens2Sea en biedt expertise in regeltechniek. (Hoge invloed en laag belang)
3. CoE HRTech. Dit bedrijf financiert het project. (Lage invloed en hoog belang)

1.2 Opdracht

Sens2Sea is bezig met radartecnologie. Radar is een technologie die radiogolven gebruikt om, ten opzichte van de positie van de radar, de afstand, snelheid en richting van objecten te bepalen. Radar kan gebruikt worden in goed en slecht weer. Hierdoor is het uitstekend geschikt om objecten te detecteren in sectoren zoals de scheepsvaart. Ook kan de radartecnologie van Sens2Sea gebruikt worden om de zeebodem in kaart te brengen en om olievlekken te detecteren.

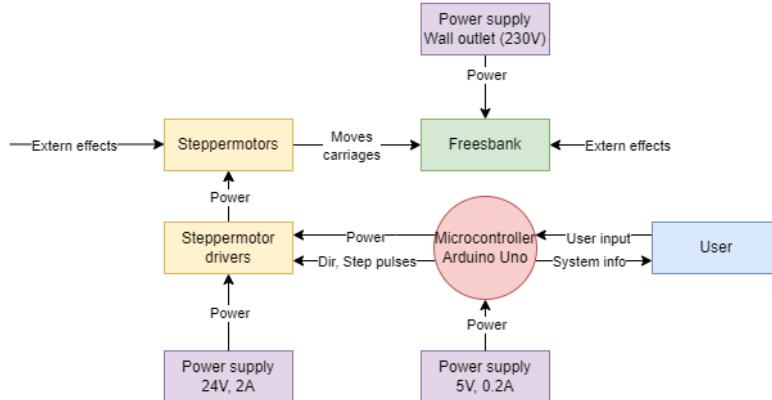
Sens2Sea is sinds kort bezig met FMCW (Frequency Modulated Continuous Wave) radartecnologie. Hiervoor wil Sens2Sea paraboolantennes gebruiken. Deze antennes zorgen ervoor dat de radargolven zodanig worden gebundeld dat de ontvangen radarreflecties worden versterkt. Dit zorgt ervoor dat het beeld een betere resolutie heeft. Deze antennes zijn niet te verkrijgen in alle maten omdat een antenne gemaakt wordt met een mal. Een mal is duur om te maken waardoor een antenne op maat een stuk duurder wordt dan een standaardmaat. Om deze reden wil Sens2Sea graag zelf dergelijke antennes kunnen maken. Er zijn hiervoor al testen geweest met antennes die in stukken geprint zijn met een 3D printer. Dit heeft redelijk goed uitgepakt, hoewel al snel bleek dat tijdens het hechten van de onderdelen, er een fout in de vorm ontstaat waardoor de antenne minder werkt. Daarom wil Sens2Sea de antennes in één stuk kunnen produceren met een 3D printer CNC machine combinatie. Deze machine moet in staat zijn antennes tot 4m x 1m x 1m kunnen produceren.

Er zijn verschillende opensource oplossingen zoals GRBL en Marlin om CNC-machines en 3D printers aan te sturen. Een 3D printer bouwt een model op door laagjes toe te voegen en een CNC machine die haalt laagjes weg om het model te krijgen. Deze oplossingen gebruiken de stappen van de stappenumotor om te bepalen waar de freeskop zich bevindt. Doordat er geen sensoren in het systeem zijn kan de microcontroller niet weten of de stappen echt gezet zijn en wordt de locatie minder nauwkeurig. Er zijn nog meer opensource oplossingen, maar geen van alle voldoen aan wat Sens2Sea wenst. Met name de nauwkeurigheid is een probleem. Het bedrijf denkt dat als de aansturing en de positiebepaling worden gescheiden de machine nauwkeuriger kan werken. Ook denkt Sens2Sea dat het systeem modular moet zijn zodat componenten als sensoren en actuatoren eenvoudig kunnen worden vervangen. Tenslotte is er de wens dat de freeskop van de machine op termijn over vijf assen moet kunnen bewegen. Deze assen zijn de bewegingsassen X, Y, Z en daarnaast de rotaties over de X en Y as.

De scope van het project omvat het scheiden van de aandrijving en positiebepaling van de freeskop. Daarnaast is er sprake van aanpassen of herschrijven van GRBL om deze te laten werken met de positie sensoren. Ook de hardware aansturen/aansluiten hoort bij de scope. Het aansturen of vervangen van de frees door de microcontroller, 3D print functie en het bewegen van de freeskop over de rotatie assen X en Y vallen buiten de scope van het project.

2 Analyse

2.1 Huidige situatie



Figuur 1: Freesbank huidige situatie

De huidige situatie bestaat uit een aantal componenten die te zien zijn in figuur 1. Er zijn al verschillende componenten aanwezig zoals de drivers van de motoren, motoren, freesbank met frees en een microcontroller. Hierin is de microcontroller een Arduino Uno met de opensource software GRBL.

De freesbank is gemaakt van hout en heeft drie assen en een handfrees. De buitenmaten zijn 300cm in de lengte(X), 152cm in de breedte(Y) en 60cm hoog(Z). Bij de maat van de hoogte is de lengte van de poten niet meegerekend. De werkruimte dat de freesbank kan bereiken is 251 cm in de lengte, 127.5 cm in de breedte en 8 cm in de hoogte. De freesbank kan in drie richtingen bewegen. De X-as is de lengterichting van de machine. Aan beide zijden bevinden zich rails. Over deze rails beweegt de carriage, het montageonderdeel dat over de X-as loopt waaraan de rest van de bewegingen zitten. Op de carriage is een tweede rails gemonteerd, waarover de montage van de Z-as kan bewegen. Deze beweging vindt zich plaats in de Y-richting. De Z-richting wordt bewogen met een schroef dat de frees beweegt over de Z-as. De X- en Y-as worden bewogen met tandriemen. De volledige constructie is te zien in Appendix C.

Aan de Z-as is een handfrees die op netstroom werkt gemonteerd. Wanneer er gefreesd wordt komt er materiaal los dat blijft liggen op de freesbank. Op de freesbank werken externe effecten zoals wrijving op de rails en uitrekking van de tandriemen.

De motoren die gebruikt worden voor de beweging van de assen zijn NEMA 17 motoren. Deze werken op 12-36V en de maximale stroomsterkte mag 2A zijn. De motoren zitten vast aan het frame met bevestigingspunten. De lengteas heeft motoren aan beide kanten van de freesbank. De hoogte en breedte assen worden aangedreven door één motor. De datasheet van deze motor is te vinden in '7.1 Datasheets'.

De drivers die gebruikt worden voor de motoren zijn DRV8824 drivers. Deze drivers werken op 8.2-45V en de maximale stroomsterkte mag 2.5A zijn op 24V. Deze drivers maken gebruik van een STEP(Stap) en DIR(Richting) pin. Door de DIR pin hoog en laag te maken verandert de polariteit van de aangesloten motor. Door de STEP pin hoog te maken zet de motor een stap. Deze drivers hebben ook de mogelijkheid

om te microsteppen. De techniek van microsteppen maakt gebruik van sinusvormige stromen om meer nauwkeurigheid en hogere resolutie te krijgen. De drivers zitten op een CNC shield dat 5V en 40mA nodig heeft. Deze stroom wordt geleverd door de microcontroller. De drivers krijgen voeding van een lab voeding. De datasheet van deze drivers en CNC shield zijn vinden in '7.1 Datasheets'.

De microcontroller is een Arduino Uno met de opensource software GRBL. Dit is het onderdeel dat input van de gebruiker krijgt en informatie over het systeem terugstuurt naar de gebruiker. De input die de gebruiker kan geven zijn systeem commando's waarmee de status van de machine wordt opgevraagd. Hiernaast kan de gebruiker G-Code sturen naar de microcontroller. De microcontroller wordt gevoed door een laptop met 5V en 200mA.

G-Code is een commando dat gebruikt wordt door freesmachines en 3D printers. Deze codes worden aangegeven met de letter G. Daarnaast worden ook M-Codes gebruikt om deze machines uit te voeren. Deze codes zijn hulpcodes om extra instructies te geven maar deze codes zijn even belangrijk als de G-Codes. In dit verslag zullen beide soorten G-Code genoemd worden.

Er zijn drie voedingen in de huidige situatie. De voeding voor de motor drivers is 24V 2A met een lab voeding. Het regelsysteem wordt gevoed door een laptop met 5V en 200mA. Hiernaast is er nog een voeding die de handfrees voedt met netstroom.

2.2 Sensor

Uit een interview met de opdrachtgever is gebleken dat de opdrachtgever encoders wil gebruiken. Dit mogen geen stappenmotoren zijn met encoders bevestigd omdat het bedrijf dan afhankelijk is van de fabrikant van de encoders. De opdrachtgever wil indien nodig de nauwkeurigheid van de machine verhogen met andere soorten sensoren.

Voor dit prototype is het belangrijk dat de positiebepaling gescheiden wordt van de aansturing van de motoren. Verder mag de prijs van de encoder niet boven de 60 euro per encoder uitkomen. Er is geen eis voor de nauwkeurigheid van de encoder omdat dit niet het belangrijkste is voor dit prototype.

Er zijn verschillende soorten encoders waarnaar onderzoek gedaan is. De nauwkeurigheid van een encoder wordt gemeten in Pulses Per Rotation(PPR). Dit betekent dat hoe hoger de PPR is, hoe nauwkeuriger de encoder is. Na het onderzoek zijn de opties voorgelegd aan de opdrachtgever en is er gekozen om een optische incrementale encoder(me38s6-c-(600)b5g2) te gebruiken. Dit omdat deze encoder relatief goedkoop is met een goede nauwkeurigheid ten opzichte van andere encoders. De datasheet van deze sensor is te vinden in '7.1 Datasheets'. Het onderzoek is te vinden in Appendix A.

2.3 Microcontroller

Het huidige systeem maakt gebruik van een Arduino Uno. Deze microcontroller heeft niet genoeg interrupt pinnen voor de encoders. Interrupt pinnen zijn speciale pinnen op een microcontroller. Deze pinnen onderbreken de code waardoor een ander stuk code uitgevoerd kan worden. Hierna gaat de microcontroller weer verder met waar deze mee bezig was. De huidige microcontroller heeft niet genoeg van deze pinnen. Verder heeft deze microcontroller niet genoeg pinnen voor de uitbreidingen die in de toekomst toegevoegd zullen worden. In overleg met de opdrachtgever is besloten om een Arduino Mega te gebruiken. Deze microcontroller heeft meer rekenkracht dan de Arduino Uno en heeft veel meer pinnen. Ook maakt het bedrijf al gebruik van de Arduino omgeving waardoor er al kennis van is.

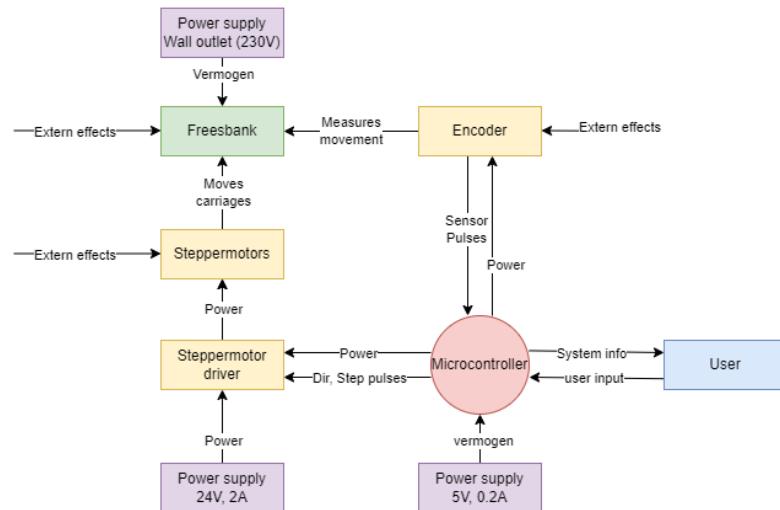
2.4 Requirements

Uit de analyse en interviews met de opdrachtgever zijn de volgende eisen opgesteld:

Requirement	User story	Acceptatie criteria	soort
De freeskop moet in de X-richting kunnen bewegen.	Als opdrachtgever wil ik dat de freeskop in de X-richting kan bewegen.	Als de freeskop in de X-richting kan bewegen.	Funct.
De freeskop moet in de Y-richting kunnen bewegen.	Als opdrachtgever wil ik dat de freeskop in de Y-richting kan bewegen.	Als de freeskop in de Y-richting kan bewegen.	Funct.
De freeskop moet in de Z-richting kunnen bewegen.	Als opdrachtgever wil ik dat de freeskop in de Z-richting kan bewegen.	Als de freeskop in de Z-richting kan bewegen.	Funct.
De positie van de freeskop op de X-as moet bepaald worden met een encoder.	Als opdrachtgever wil ik dat de positie van de freeskop op de X-as bepaald wordt met een encoder.	Als de positie van de freeskop op de X-as bepaald wordt met een encoder.	Funct.
De positie van de freeskop op de Y-as moet bepaald worden met een encoder.	Als opdrachtgever wil ik dat de positie van de freeskop op de Y-as bepaald wordt met een encoder.	Als de positie van de freeskop op de Y-as bepaald wordt met een encoder.	Funct.
De positie van de freeskop op de Z-as moet bepaald worden met een encoder.	Als opdrachtgever wil ik dat de positie van de freeskop op de Z-as bepaald wordt met een encoder.	Als de positie van de freeskop op de Z-as bepaald wordt met een encoder.	Funct.
De freesbank moet G-Code kunnen ontvangen en uitvoeren.	Als opdrachtgever wil ik dat de freesbank G-Code commando's kan ontvangen en uitvoeren.	Als de freesbank G-Code commando's kan ontvangen en uitvoeren.	Funct.
De software moet modulair zijn opgebouwd.	Als opdrachtgever wil ik dat de software van de freesbank modulair opgebouwd is zodat er gemakkelijk onderdelen aan toegevoegd kunnen worden.	Als de code van de freesbank zo gebouwd is dat er gemakkelijk onderdelen aan toegevoegd moeten kunnen worden.	Niet funct.

3 Ontwerpen

Uit de wensen en eisen van de opdrachtgever ontstaat er een nieuw diagram.

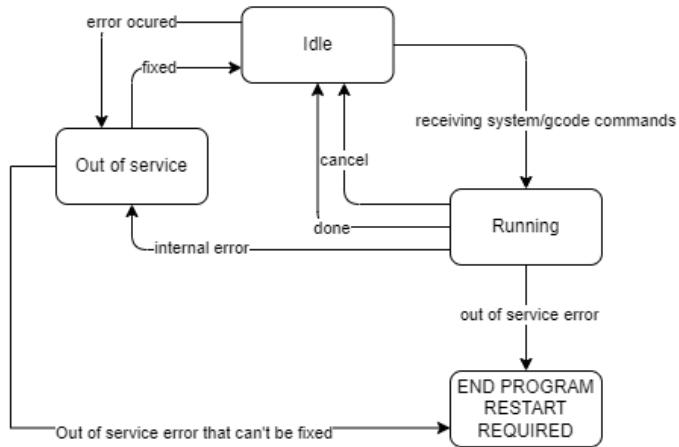


Figuur 2: Freesbank Nieuw ontwerp

In figuur 2 is het nieuwe ontwerp te zien waaraan een encoder is toegevoegd. Iedere as heeft zijn eigen encoder waardoor de bewegingen per as gemeten kan worden. De encoders moeten bevestigd worden op een getande riem dat op het frame is geplakt. Doordat deze riem op het frame geplakt is, kan deze niet uitrekken waardoor de nauwkeurigheid gewaarborgd blijft. De encoder gebruikt 5V en 40mA en krijgt deze voeding van de microcontroller. De encoders sturen pulsen naar de microcontroller als de as gedraait wordt. De microcontroller verwerkt deze pulsen. Verder is de microcontroller vervangen door een Arduino Mega. Deze microcontroller verstuurde de stap en richting signalen naar de motor drivers.

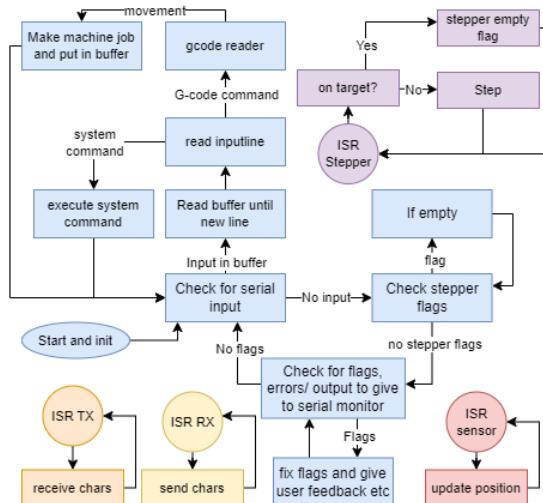
3.1 Software ontwerpen

Voor de software zijn verschillende ontwerpen gemaakt. Dit zijn de statemachine en flowchart van de software.



Figuur 3: Statemachine.

In figuur 3 is de statemachine te zien. Hierin is te zien in welke states het systeem zich kan bevinden. Het systeem start bij 'IDLE'. Hierna kan het systeem opdrachten krijgen waardoor de state 'RUNNING' wordt. De opdrachten kunnen klaar zijn of worden geannuleerd. Hierna wordt de state veranderd naar 'IDLE'. Er kunnen fouten optreden, als dit gebeurt zijn er 2 states. Als de fout niet kritiek is en het systeem deze zelf kan verhelpen verandert de state naar 'Out of service'. Als de fout niet opgelost kan worden door de machine moet er een handmatige restart komen en zal de state veranderen naar 'END PROGRAM RESTART REQUIRED'. Deze state gaat niet weg zonder dat er een restart is uitgevoerd.



Figuur 4: Software Flowchart.

In figuur 4 is een flowchart te zien dat verdeeld is in meerdere kleuren. Oranje en geel zijn interrupts voor de input van de gebruiker. Rood is de interrupt van de sensoren en paars is de interrupt die de motoren aanstuurt. Het blauwe gedeelte is de main loop die input, errors en het maken van opdrachten afhandelt. Als er een input van de gebruiker komt, wordt deze gelezen. Als het een systeem opdracht is zoals

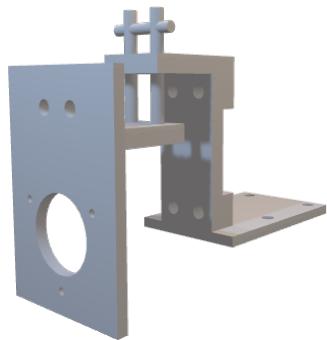
de status opvragen zal die meteen worden afgehandeld door 'execute system command'. In het andere geval wordt de G-Code gelezen en in de queue gezet. Als het G-Code commando meteen afgehandeld moet worden, wordt deze direct uitgevoerd. Mocht de gebruiker een verkeerde opdracht geven, wordt de opdracht genegeerd.

3.2 Elektrisch schema

Er zijn twee elektrische schema's gemaakt die te vinden zijn in Appendix B. Het eerste schema is gemaakt voor de aansturing op één as en bevat een Arduino Uno, een encoder, een DRV8824 motor driver, een 100uf condensator en een NEMA17 stappenmotor. Het tweede elektrische schema is voor de volledige beweging van drie assen en bevat een Arduino Mega, 3 encoders, 3 DRV8824 motor drivers, 3 100uf condensatoren en 3 NEMA17 stappenmotoren.

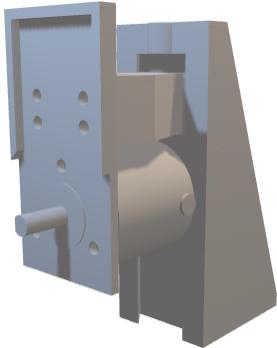
4 Realisatie

4.1 Mechanica



Figuur 5: Mechanica Iteratie 1

Omdat de encoder over een randriem van de Z-as moet lopen is een beugel nodig. De opstelling hiervoor is te zien in Appendix C, figuur 14 en figuur 15. De opdrachtgever wil dat er een ontwerp komt dat met een veer ervoor zorgt dat de encoder altijd op de tandriem blijft lopen. Hiernaast is het belangrijk dat het model wordt gemaakt in Freecad vanwege licentiebeperkingen. Voor deze beugel zijn meerdere iteraties gemaakt waarvan de eerste iteratie instabiel was en daardoor waren er veel beweging wanneer de freesbank bewoog. De oorzaak van dit probleem lag bij de staven tussen de twee bewegende delen die heel dun waren waardoor er veel spelting op zat. Daarnaast braken de staven af wanneer er een veer werd toegevoegd omdat de staven te dun waren. Ook had dit model het probleem dat deze niet goed paste omdat er geen rekening gehouden was met een metalen staaf waارlangs de Z-as beweegt.



Figuur 6: Mechanica Iteratie 2

In de tweede iteratie zijn deze problemen opgelost. Deze iteratie maakt gebruik van een driehoek vormige gleuf waarin het bewegende deel kan bewegen. Ook is er ruimte voor een spijker met de veer om het bewegende deel omlaag te duwen. Hierdoor zit er geen speling tussen de bewegende onderdelen en kan de encoder niet heen en weer kan bewegen bij trillingen. Dit ontwerp heeft echter het probleem dat de Z-as niet meer volledig naar beneden kan doordat het ontwerp te hoog is en recht onder het bewegende deel van de Z-as zit.



Figuur 7: Mechanica Iteratie 3

Voor de derde iteratie is er een plaat gemaakt dat over de gehele bovenkant van de Z-as zit. Deze plaat steekt aan de zijkant uit waarop iteratie 2 gemonteerd kan worden. Met dit ontwerp kan de Z-as volledig naar boven en naar beneden bewegen. Dit ontwerp zorgt ervoor dat de Y-as ongeveer 4 cm kleiner wordt, maar dit is volgens de opdrachtgever geen groot probleem voor het prototype.

4.2 Software

De software is opgebouwd uit de verschillende modules: **datacommunicatie**, **stepper**, **encoder** en **system**. Ook is er een bestand main.cpp waar het programma begint en een macros.h file waar alle macro's zich bevinden. De software is geschreven in C++ maar maakt geen gebruik van classes. De software maakt ook geen gebruik van de Arduino.h library. Hierdoor zijn alle Arduino functies niet beschikbaar en worden de pinnen rechtstreeks gezet door de registers te beïnvloeden. Ook interrupts worden op deze manier toegevoegd. Interrupts zijn señales die de processor vertellen dat die moet stoppen met waar die nu mee bezig is en de code die bij die interrupt hoort moet gaan uitvoeren. Nadat deze code is uitgevoerd hervat de processor de rest van de code.

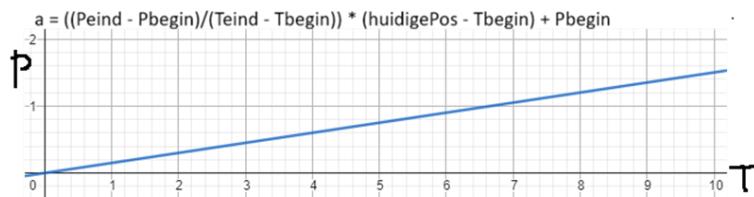
Main Deze module definieert het begin van het programma. Hierin bevindt zich de main loop waaruit alle andere functies van andere modules worden aangeroepen. Deze module initialiseert alle andere modules indien nodig en leegt de seriële buffer voor de zekerheid.

Macro's Deze header file heeft alle macro's die nodig zijn voor het bestand. Macro's zijn termen die door de compiler veranderd worden in de definitie van de macro. Dus als er een macro **TIJD_MS** 3 is, zal de compiler alle instanties van **TIJD_MS** veranderen in 3. Een belangrijk onderdeel van dit bestand is de custom datatypes. Omdat een int op een Arduino geen 32 bit maar 16 bit is, kloppen de standaard uint32_t datatypes niet. Hiervoor zijn deze opnieuw gedefinieerd als:**uchar**, **uint8**, **int8**, **uint16**, **int16**, **uint32**, **int32**. Floats en doubles zijn dezelfde grootte op een Arduino dus wordt er verder alleen float gebruikt. Hierin zijn ook alle pin en port definities te vinden.

Datacommunicatie Deze module zorgt voor de seriële verbinding met de gebruiker. De standaard Arduino library voldoet niet aan de benodigdheden voor een machine die snel en nauwkeurig moet werken. Dit komt omdat de functies van deze library blokkerend zijn. Dit betekent dat wanneer de buffer op de Arduino vol zit, omdat het niet snel genoeg verstuurd of ontvangen wordt, de Arduino wacht met alles wat gedaan moet worden inclusief bewegingen totdat er ruimte is in de buffer om de characters op te slaan. De implementatie van dit project werkt op dezelfde manier als de standaard library, maar deze gooit de characters weg als de buffer vol zit, waardoor de bewegingen nauwkeurig blijven. Deze module bevat ook verschillende high level implementaties om de data te sturen. Er kunnen door de **print()** functie intergers, chars, floats, strings, en nog meer verschillende datatypes geprint worden.

Stepper Deze module zorgt voor alle functies van de steppers. Eerst wordt een segment(block) klaar gemaakt. Dit block heeft een aantal eigenschappen, Hieronder vallen alle parameters meegegeven met de G-Code commando's. Hiervan worden een paar extra parameters berekend die nodig zijn per stap. Het stappen wordt gedaan door een interrupt die iedere aantal milliseconde uitgevoerd wordt. Deze snelheid is gedefinieerd in macros.h. Voor ieder segment wordt er een wiskundige formule opgesteld voor het pad dat de motoren moeten volgen. Iedere as heeft zijn eigen formule. Deze formules zorgen ervoor dat als de machine naar (X=10, Y=5, Z=2) moet, alle assen op hetzelfde moment aankomen. Er zijn twee verschillende modellen: een rechte beweging en een cirkelbeweging. De commando's zijn te vinden in de handleiding.

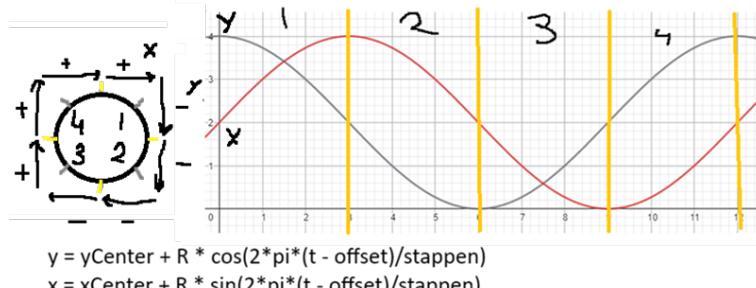
De rechte beweging is per as gedefinieerd met een lineaire formule. Deze formule gaat op de X-as (T) van t=0 naar t=eind. De eindtijd wordt bepaald door de totale afstand te delen door de afstand per puls van een stappenmotor. De Y-as(P) gaat van de beginpositie tot de eind positie van die as in de tijd van t=0 naar t=eind. Omdat de t=eind hetzelfde is bij iedere as zorgt deze constructie ervoor dat de assen op hetzelfde moment aankomen. Iedere tijdstap wordt er gekeken of de locatie achter ligt of voor ligt op de formule. Als de locatie achter ligt zal de motor een stap zetten.



Figuur 8: Software lineaire beweging.

De cirkelbeweging wordt gedaan met sinus en cosinus functies. De golven verschillen 1/4e cirkel met elkaar waardoor als deze gecombineerd worden er een cirkelbeweging ontstaat. Ook deze functie gaat van t=0 naar t=eind. Maar deze wordt verschoven met de begin hoek. Als de cirkel tegen de klok in gaat wordt de sinus en cosinus vermenigvuldigd met -1. Hierdoor lopen de golven de andere kant op waardoor de assen tegen de klok in gaan. De cirkel kan maar gebruik maken van twee assen, dus zijn de

commando's G17 tot G19 om te kiezen op welk vlak de machine moet werken. De standaard is op het XY-vlak(G17).



Figuur 9: Software cirkelbeweging.

Encoder Deze module zet de pulsen van de encoders om in een locatie. Deze module zet interrupts aan op de pinnen waaraan de encoders zijn verbonden. Als er een interrupt af gaat worden de beide pinnen uitgelezen. Door beide pinnen te gebruiken kan er onderscheid gemaakt worden tussen voor- en achteruit bewegen. In figuur 10 is de functie om de encoder uit te lezen te zien. Deze code maakt slim gebruik van bitwize operaties om het verschil te vinden. Dit stukje code wordt heel vaak uitgevoerd waardoor het belangrijk is dat de code snel uitgevoerd is. De encoders maken gebruik van de interrupt pinnen van de Arduino omdat deze pinnen de 2e hoogste prioriteit hebben op de Arduino. Deze pinnen staan ingesteld op 'any change' om de 'rising en falling edge' te detecteren.

```
static inline void readEncoderZ(void) {
    static uint8 encoderStateZ = 0;

    /** Get value of encoder pin b and masks everything but the lowest bit*/
    uint8 b = (PIND >> ENCODER_PIN_ZB) & 1;
    //PORTB ^= (1 << PBS);
    //println(encoderSteps[0]);
    /**
     * Read and mask value of encoder pin A, and convert it to 'binary graycode' (only 1 changing bit each time)
     * 0 & 0 = 0
     * 1 & 0 = 1
     * 1 & 1 = 2
     * 0 & 1 = 3
     */
    uint8 curr = (((PIND >> ENCODER_PIN_ZA) & 1) ^ b) | (b << 1);

    /**
     * Calculate movement direction using the current - previous. (83 is a bitwise modulo (%4) to avoid dividing)
     * possible values: 0 - 3
     * 0 = no movement;
     * 1 = step forward;
     * 2 = error (invalid or skiped step); (ignored)
     * 3 = step backwards;
     */
    uint8 diff = (curr - encoderStateZ) & 3;

    /**
     * if forward add 1, if backward subtract 1.
     * if diff = 1 the first part is true and it adds 1. second part is false so does not do anything
     * if diff = 3 the second part is true and it adds -1. first part is false so does not do anything
     */
    encoderSteps[Z_AXIS] += (diff == 1) - (diff == 3);

    /**
     * Save current state */
    encoderStateZ = curr;
}
```

Figuur 10: Software encoder interrupt.

System Deze module gebruikt de andere modules om alle opdrachten uit te lezen van de seriële buffer, deze om te zetten naar bruikbare informatie. Deze module zet de G-Code om naar een block en zet dit in de buffer van de stepper module. Hiernaast voert deze module ook alle systeem commando's uit zoals \$? om de huidige status uit te lezen.

4.3 Test Rapport

Na de testen die beschreven staan in Appendix D is te zien dat de freesbank in alle richtingen kan bewegen, G-Code kan uitvoeren, modulair is en de locatie kan bepalen op de Y-as.

Soort	Requirement	Uitslag
Funct	De freesbank moet G-Code kunnen ontvangen en uitvoeren.	Behaald
Funct	De freeskop moet in de X-richting kunnen bewegen.	Behaald
Funct	De freeskop moet in de Y-richting kunnen bewegen.	Behaald
Funct	De freeskop moet in de Z-richting kunnen bewegen.	Behaald
Funct	De positie van de freeskop op de X-as moet bepaald worden met een encoder.	Niet behaald
Funct	De positie van de freeskop op de Y-as moet bepaald worden met een encoder.	Behaald
Funct	De positie van de freeskop op de Z-as moet bepaald worden met een encoder.	Niet behaald
Niet. Funct	De software moet modulair opgebouwd zijn.	behaald

5 Conclusie

Het hoofddoel van deze stage was het scheiden van de aansturing en positiebepaling van de freesbank. Aan de hand van de testresultaten is te concluderen dat dit doel grotendeels is behaald. De positiebepaling op de Y-as is geslaagd. Op de X- en Z-as moeten er mechanische onderdelen komen om te bewijzen dat deze test voltooid is. De code is zo opgebouwd dat deze makkelijk is aan te passen om de positiebepaling en aansturing van deze assen te scheiden. Kortom de software is voorbereid.

6 Aanbevelingen

Tijdens dit project zijn meerdere onderdelen aan bod gekomen waar geen tijd voor was of buiten de scope lag.

Voor de X en Z assen moet er naar een manier gekeken worden om de encoders aan de assen te zetten.

In dit project is de handfrees gebruikt die al op de machine zat. Deze moet handmatig aangezet worden, maar dit is in de praktijk niet handig. Dus zou er een frees moeten komen die aangestuurd kan worden door de microcontroller. Ook maakt de frees materiaal los dat afgezogen moet worden. Hier moet ook over nagedacht worden.

De bekabeling van de freesbank is niet netjes. Voor het nieuwe ontwerp moet er gezorgd worden dat de kabels niet vast kunnen zitten achter andere onderdelen van de machine. Daarnaast moet ervoor gezorgd worden dat de bekabeling niet hangt om kabelbreuken tegen te gaan. Een kabelboom of kabelslang zou hier een oplossing voor kunnen zijn.

De motordrivers van het systeem zitten in dit ontwerp op een CNC-shield. De pinconfiguraties van de code is anders dan de configuratie van het CNC-shield. Hierdoor moeten losse kabels worden gebruikt en de bevestigingspunten voor deze kabels zitten onderop en bovenop de shield. Om deze reden en dat de freesbank in de toekomst 5 assen moet krijgt zou een PCB(Printed Circuit Board) maken te adviseren zijn.

Om de encoder uit te lezen worden op dit moment interrupt pinnen gebruikt op de Arduino. Er zijn ook modules die de pulsen van de encoders kunnen opvangen en daarmee de positie kunnen bepalen. Deze modules zouden alleen uitgelezen moeten worden door de microcontroller om de huidige locatie te krijgen. Dit bespaard de microcontroller rekenkracht. Of dit werkt zou onderzocht moeten worden.

Het regelsysteem heeft een gelimiteerd aantal plekken in de buffer voor G-Code commando's. Hierdoor raakt de buffer van de seriële lijn vol waardoor andere belangrijke commando's niet binnenkomen. Een SD kaart toevoegen aan het systeem waar de step files op staan zou dit verhelpen. Dit wordt al vaak toegepast op dit soort machines.

Het systeem moet gekalibreerd worden als deze gebruikt wordt. Dit kan gedaan worden door de freeskop van (X-, Y- en Z-) hoek naar de (X+, Y+ en Z+) hoek te bewegen en de encoderstappen tellen. Het aantal stappen per as kan door de lengte van de as gedeeld worden om de nauwkeurigheid te krijgen. Hiervoor moeten de switches die aangeven of het einde bereikt is aangesloten worden.

Het systeem maakt gebruik van stappemotoren. Deze motoren kunnen normale stappen zetten, maar ook microsteppen. In het systeem worden normale stappen gebruikt. Maar microsteppen zijn fijner en maken minder trillingen waardoor de nauwkeurigheid van de machine beter kan worden. Hier was tijdens het project geen tijd voor.

7 Bronnen

1. Wikipedia contributors, "Radar," Wikipedia, The Free Encyclopedia (Dutch version). [Online]. Available: <https://nl.wikipedia.org/wiki/Radar>. Accessed: Jan. 6, 2026.
2. Wikipedia contributors, "Radar," Wikipedia, The Free Encyclopedia. [Online]. Available: <https://en.wikipedia.org/wiki/Radar>. Accessed: Jan. 6, 2026.
3. Wikipedia contributors, "Incremental encoder," Wikipedia, The Free Encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/Incremental_encoder. Accessed: Jan. 6, 2026.
4. Wikipedia contributors, "Rotary encoder," Wikipedia, The Free Encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/Rotary_encoder. Accessed: Jan. 6, 2026.
5. Laumans Techniek, "Encoders voor industriële automatisering," Laumans Techniek. [Online]. Available: <https://www.laumanstechniek.nl/blog/kennisbank-3/encoders-voor-industriële-automatisering-20>. Accessed: Jan. 6, 2026.
6. Linear Motion Tips, "Microstepping Basics," Linear Motion Tips. [Online]. Available: <https://www.linearmotiontips.com/microstepping-basics/>. Accessed: Jan. 6, 2026.
7. Dynapar, "Magnetic Encoders," Dynapar Knowledge Center. [Online]. Available: <https://www.dynapar.com/knowledge/encoder-basics/encoder-technology/magnetic-encoders>. Accessed: Jan. 8, 2026.
8. Wikipedia contributors, "Gray code," Wikipedia, The Free Encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/Gray_code. Accessed: Jan. 8, 2026.
9. Wikipedia contributors, "G-Code," Wikipedia, The Free Encyclopedia. [Online]. Available: <https://nl.wikipedia.org/wiki/G-code>. Accessed: Jan. 30, 2026.
10. FreeCAD, FreeCAD: Your own 3D parametric modeler, FreeCAD Project. [Online]. Available: <https://www.freecad.org/>. Accessed: Jan. 30, 2026.

7.1 Datasheets

1. Texas Instruments, "DRV8824 Stepper Motor Controller IC," Datasheet. [Online]. Available: <https://www.ti.com/lit/ds/symlink/drive8824.pdf>. Accessed: Jan. 6, 2026.
2. PBC Linear, "Stepper Motor Support: NEMA 17 Data Sheet," PBC Linear. [Online]. Available: <https://media.pbclinear.com/pdfs/pbc-linear-data-sheets/data-sheet-stepper-motor-support.pdf>. Accessed: Jan. 6, 2026.
3. "Datasheet E38S6G5-600B-G24N," engautomacaoing, Scribd, [Online]. Available: <https://www.scribd.com/document/756625816/Datasheet-E38S6G5-600B-G24N>. Accessed on: Jan. 27, 2026.
4. Handson Technology, "3-Axis CNC/Stepper Motor Shield for Arduino User Manual," Handson Technology, Online. Available: <https://www.handsontec.com/dataspecs/cnc-3axis-shield.pdf>. Accessed on: Jan. 29, 2026.

A Encoder onderzoek

Om de positie van de freeskop van een freesbank te bepalen zijn verschillende mogelijkheden. Uit een interview met de opdrachtgever is gekomen dat encoders gebruikt moeten worden. Een encoder is een sensor met een roterende as. Als de as draait zet de encoder de beweging om in elektrische signalen. Hiermee kunnen microcontrollers de positie, snelheid en richting bepalen. De encoders zijn onder te verdelen in verschillende soorten: Incrementeel of absoluut. Ook zijn er verschillende meet methodes: Magnetisch of optisch. Er zijn ook encoders die lineair zijn, maar daarvan heeft de opdrachtgever aangegeven dat die niet bruikbaar zijn. Bij de keuze moet ook rekening gehouden worden met dat de prijs van een sensor die onder de 60 euro moet blijven.

A.1 Meet methodes

Er zijn verschillende manieren om de beweging te meten. Hiervoor zijn magnetisch en optisch de meest gebruikte technologieën.

Magnetische encoders maken gebruik van een magnetisch veld om de rotatie van de as te bepalen. Dit heeft het voordeel dat deze encoders robuuster zijn. Daarnaast zijn deze ook beter bestand tegen vervuiling, trillingen en extreme omgevingsomstandigheden. Het nadeel is dat sterke magnetische velden metingen van andere sensoren kunnen beïnvloeden. Er zijn drie grote groepen magnetische encoders.

1. **Magnetic Gear Tooth Sensor or Pickup**, Deze encoder heeft een magnetische sensor en een ferromagnetisch tandwiel. Dit betekent dat het tandwiel in staat is om een magnetisch veld te geleiden. Het voordeel hiervan is dat deze goedkoop zijn, alleen zijn deze gelimiteerd door het aantal tanden van het tandwiel. Dit limiteert de resolutie naar 120 of 240 PPR(Pulses Per Rotation).
2. **Magneto-Resistive**, Deze encoder genereert een sinus golf door weerstand te meten op een wiel met afwisselende magnetische polen of een film met weerstanden. De afwisselende magnetische polen bieden een hogere nauwkeurigheid dan het film met weerstanden. Deze encoders zijn moeilijker om te integreren in een systeem.
3. **Hall-Effect magnetic**, Deze encoder maakt gebruik van het Hall-Effect. De encoder bevat een laag van een halfgeleidermateriaal dat verbonden is met een voeding. Als een magnetische pool langs de Hall-Effect sensor komt, wordt er een hoge voltage gegenereerd. De frequentie en amplitude van de verstoring in het magnetisch veld kan worden gebruikt om de snelheid en verplaatsing te bepalen.

Optische encoders zijn encoders die gebruik maken van optische signalen om stappen te tellen. Dit doen deze encoders met behulp van een schijf met patronen dat vastzit aan de as van de encoder. Door een lichtsensor worden deze patronen omgezet in elektrische pulsen. Deze sensoren bieden een hoge resolutie en veel nauwkeurigheid. Hierdoor zijn deze encoders goed op plekken te gebruiken waar veel nauwkeurigheid verwacht wordt. Het nadeel van deze encoders is dat deze gevoeliger zijn voor stof, vuil en trillingen.

A.2 Encoder soorten

Incrementele encoders genereren pulsen als de as beweegt. Op de schijf zijn er één of meerdere rijen met patronen. Deze rijen heten kanalen en als er twee of meer kanalen zijn, is het mogelijk om de richting

te bepalen. Deze patronen zijn 1/4 patroon van elkaar verschoven. Hierdoor heeft een encoder met twee kanalen een patroon in de signalen. Dit betekent ook dat deze hetzelfde aantal draden heeft als kanalen. Doordat de encoder een patroon volgt moet de microcontroller zelf de positie bijhouden tenopzichte van een beginpunt. Hierdoor gaat de positiebepaling verloren als er een stroomonderbreking is.

Het aantal patronen per kanaal bepaalt de nauwkeurigheid. Als een encoder meerdere kanalen heeft vergroot dat de nauwkeurigheid. Bij een encoder met twee kanalen gaat de nauwkeurigheid maal vier. Bijvoorbeeld bij 600 PPR (Pulses Per Rotation) op één kanaal, is de nauwkeurig op twee kanalen $600 \times 4 = 2400$ PPR.

Het patroon van de signalen volgt graycode. Dit is te zien in figuur 11. Graycode is een ander soort binaire code dat bij elke ophoging en verlaging één bit verandert. De signalen moeten opgevangen worden door een microcontroller via polling of interrupts. Een interrupt vertelt de microcontroller dat er een verandering is in de pin waarop de interrupt staat. Polling is het continu kijken of de status verandert. Interrupts zijn hierbij handiger omdat er dan geen data verloren kan gaan door een te langzame sample snelheid.

A	B
0	0
1	0
1	1
0	1

Figuur 11: Encoder output table.

De signalen moeten worden opgevangen door een microcontroller. Dit kan via polling of interrupts. Interrupts stoppen het programma en voeren hun code uit, waarna het programma weer verder kan gaan. Polling is een manier van input lezen door iedere keer te kijken of de data veranderd is. Interrupts zijn hiervoor beter omdat er bij polling data verloren kan gaan door een te lage sample snelheid. De andere kant is dat een microcontroller maar een bepaald aantal interrupts per seconde kan verwerken.

Absolute encoders zijn encoders die elke positie op de schijf een unieke code geven. Hierdoor weet de encoder exact waar die zich bevindt in de rotatie. Hierdoor gaat de positie niet verloren als de spanning onderbroken is. De single-turn variant kan niet het aantal rotaties meten. Hierom zijn er multi-turn varianten. Absolute encoders moeten worden uitgelezen met protocollen. Op een absolute encoder bepaalt het aantal bits hoe nauwkeurig de encoder is. Een multi-turn absolute encoder heeft n aantal bits per rotatie en n aantal bits voor de aantal rotaties.

Absolute encoders zijn duurder dan incrementele encoders. Van absolute encoders zijn de multi-turn encoders varianten weer duurder dan single-turn encoder.

	Absolute	Incrementeel
Kosten	Duur	Goedkoper
Uitlezen	Protocol	Pulsen tellen
Positie onthouden zonder stroom	Ja	Nee

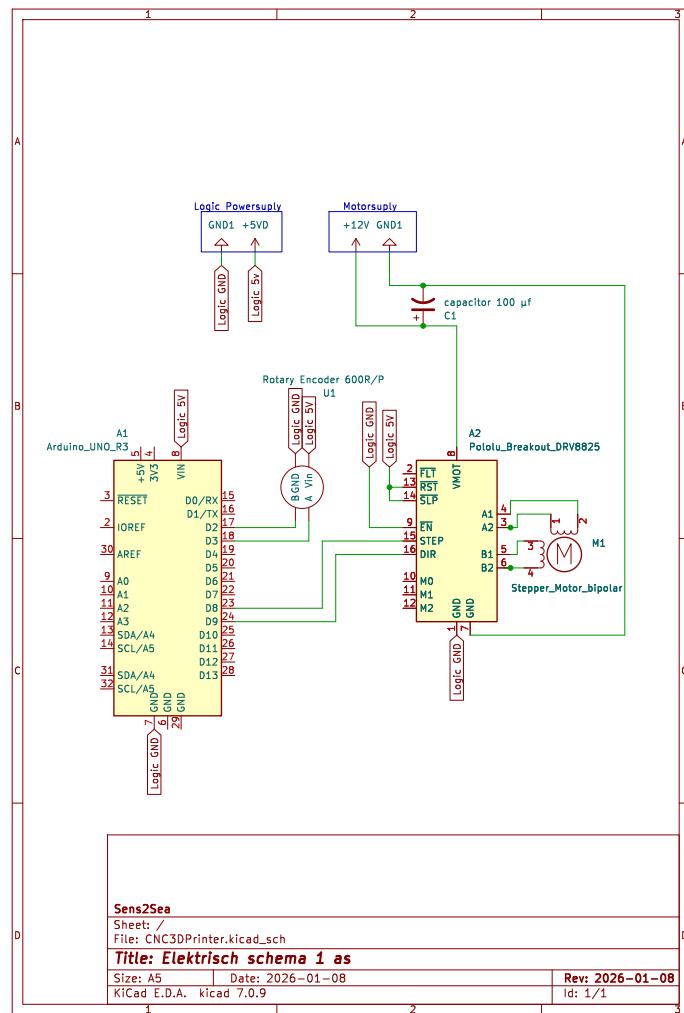
Om zeker te weten of een incrementele encoder kan werken door middel van interrupts, is er een testje gedaan. Door de interrupt pin te verbinden met een andere pin die steeds hoog en laag gemaakt wordt is er getest hoeveel interrupts de microcontroller kan verwerken in een seconde. Deze test geeft een indicatie hoeveel de microcontroller kan verwerken. Hierbij wordt de tijd gemeten door de tijd van de seriële monitor.

Op tijdstip 1: 09 : 36 : 47.772 zijn er 848785 interrupts verwerkt.
Op tijdstip 2: 09 : 36 : 53.204 zijn er 1484065 interrupts verwerkt.
Dit betekent dat er in 5.432 seconde 635280 interrupts zijn verwerkt.
 $5.432/635280 \approx 116951$ interrupts per seconde.

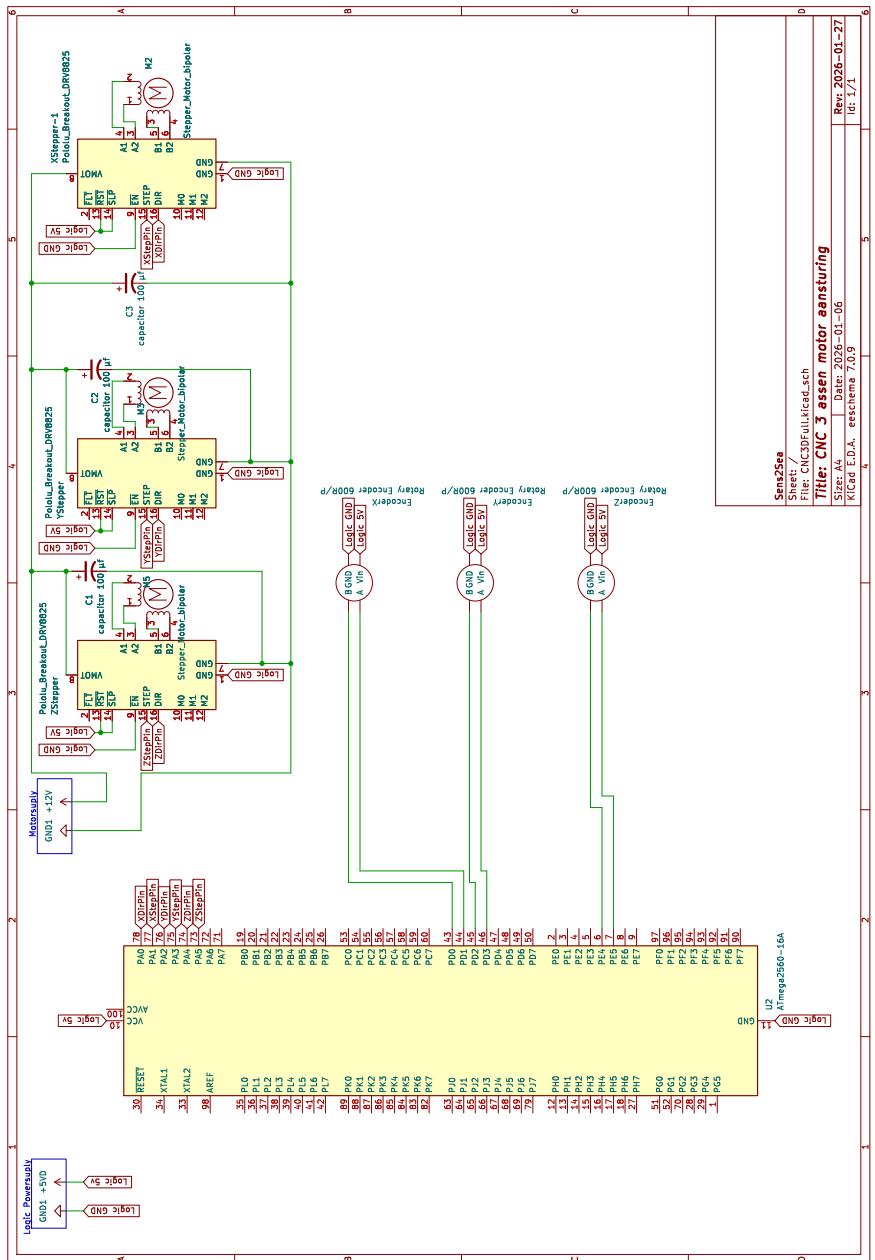
Een pulley van 16 tanden op een tandriem met tanden van 1mm tanden geeft een afstand van 2mm per tand. Dit is $2 * 16 = 32$ mm per rotatie. Indien er 20000 interrupts per as gebruikt kunnen worden door de sensoren kan de freesbank in theorie $20000/2400 * 32 \approx 266.6$ mm/s bewegen. Dit is in theorie voldoende voor de werking van de freesbank. Kortom deze soorten encoders zijn een optie.

De opties zijn voorgelegd aan de opdrachtgever en deze heeft gekozen voor een betaalbare optische incrementele encoder. Dit is uiteindelijk de 'me38s6-c-(600)b5g2' rotary encoder geworden. Deze encoder heeft 600 PPR per kanaal. Dus 2400 PPR in twee kanalen.

B elektrische schema's



Figuur 12: Elektrisch schema één as.



Figuur 13: Elektrisch schema drie assen.

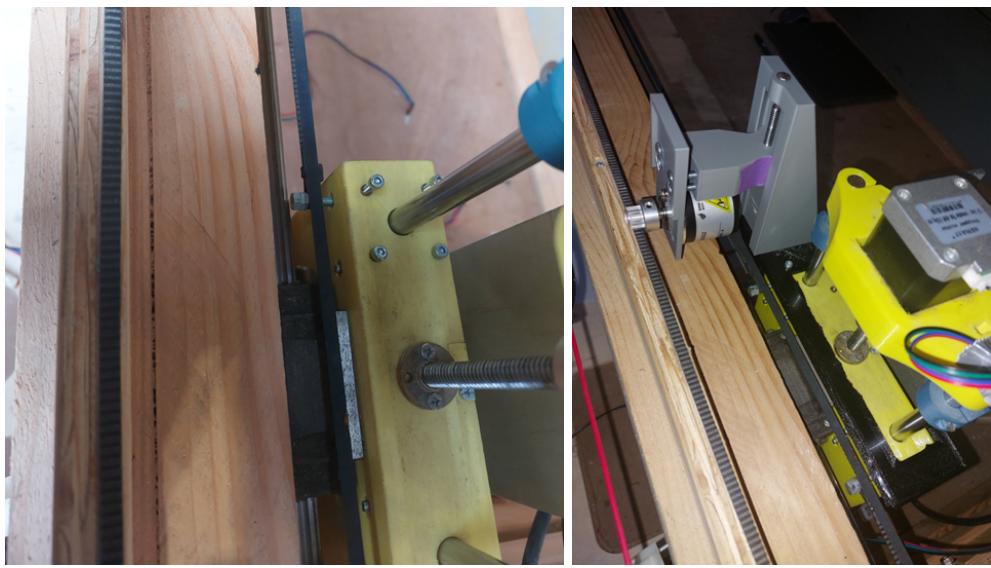
C Afbeeldingen van de opstelling



Figuur 14: Zijaanzicht freesbank.



Figuur 15: Vooraanzicht freesbank.



Figuur 16: Links de Y-as zonder encoder, Rechts de Y-as met encoder en mechanische onderdelen.

D Testrapport

In dit rapport worden alle testen voor de eisen van het prototype weergegeven. Het prototype moet verschillende dingen kunnen doen. Deze zijn gedefinieerd in de requirements en zijn het volgende:

1. De freesbank moet G-Code kunnen ontvangen en uitvoeren;
2. De software moet modulair zijn opgebouwd;
3. De freeskop moet in de X-richting kunnen bewegen;
4. De freeskop moet in de Y-richting kunnen bewegen;
5. De freeskop moet in de Z-richting kunnen bewegen;
6. De positie van de freeskop op de X-as moet bepaald worden met een encoder;
7. De positie van de freeskop op de Y-as moet bepaald worden met een encoder;
8. De positie van de freeskop op de Z-as moet bepaald worden met een encoder.

D.1 G-Code ontvangen en uitvoeren

Voor deze requirement moet het prototype G-Code kunnen ontvangen en uitvoeren. Deze test is voldaan als de machine G-Code commando's kan uitvoeren die de gebruiker stuurt via de seriële monitor.

Deze test kan alleen op de machine worden uitgevoerd. Hiervoor zijn de volgende dingen nodig:

1. Microcontroller met code dat G-Code kan ontvangen en uitvoeren.

Om de test uit te voeren moeten de volgende stappen worden uitgevoerd:

1. Verbind de microcontroller met een laptop met seriële monitor;
2. Start de seriële monitor en stuur het commando G1 X10;
3. Wacht totdat de machine 'OK.' terug stuurt;
4. Herhaal dit voor alle andere commando's die beschikbaar zijn.

Waarnemingen en conclusie Na het sturen van ieder commando stuurt de machine 'OK.' terug. Dit geeft aan dat de test geslaagd is.

D.2 X-richting bewegen

Voor deze requirement moet het prototype in de X-richting kunnen bewegen. Deze test is voldaan als de machine in de X-richting beweegt. Voor deze test moet de machine G-Code kunnen uitvoeren.

Deze test kan alleen op de machine worden uitgevoerd. Hiervoor zijn de volgende dingen nodig:

1. Microcontroller met code dat G-Code kan ontvangen en uitvoeren en assen kan laten bewegen.

2. Stappenmotor met driver.
3. Bedrading.

Om de test uit te voeren moeten de volgende stappen worden uitgevoerd:

1. Bedraad alle componenten zoals aangegeven is in het elektrische schema;
2. Verbind de microcontroller met een laptop met seriële monitor;
3. Start de seriële monitor en stuur het commando G1 X10;
4. Wacht totdat de machine 'OK.' geeft en de machine gaat bewegen;
5. Als de X-as is bewogen stuur dan het commando G1 X0 en wacht totdat de machine terug is bewogen naar het beginpunt.

Waarnemingen en conclusie Na het sturen van G1 X10 commando stuurt de machine 'OK.' terug en gaat de as bewegen. Na het sturen van G1 X0 beweegt de machine terug naar het beginpunt. Dit geeft aan dat de test geslaagd is.

D.3 Y-richting bewegen

Voor deze requirement moet het prototype in de Y-richting kunnen bewegen. Deze test is voldaan als de machine in de Y-richting beweegt. Voor deze test moet de machine G-Code kunnen uitvoeren.

Deze test kan alleen op de machine worden uitgevoerd. Hiervoor zijn de volgende dingen nodig:

1. Microcontroller met code dat G-Code kan ontvangen en uitvoeren en assen kan laten bewegen;
2. Stappenmotor met driver;
3. Bedrading.

Om de test uit te voeren moeten de volgende stappen worden uitgevoerd:

1. Bedraad alle componenten zoals aangegeven is in het elektrische schema;
2. Verbind de microcontroller met een laptop met seriële monitor;
3. Start de seriële monitor en stuur het commando G1 Y10;
4. Wacht totdat de machine 'OK.' geeft en de machine gaat bewegen;
5. Als de Y-as is bewogen stuur dan het commando G1 Y0 en wacht totdat de machine terug is bewogen naar het beginpunt.

Waarnemingen en conclusie Na het sturen van G1 Y10 commando stuurt de machine 'OK.' terug en gaat de as bewegen. Na het sturen van G1 Y0 beweegt de machine terug naar het beginpunt. Dit geeft aan dat de test geslaagd is.

D.4 Z-richting bewegen

Voor deze requirement moet het prototype in de Z-richting kunnen bewegen. Deze test is voldaan als de machine in de Z-richting beweegt. Voor deze test moet de machine G-Code kunnen uitvoeren.

Deze test kan alleen op de machine worden uitgevoerd. Hiervoor zijn de volgende dingen nodig:

1. Microcontroller met code dat G-Code kan ontvangen en uitvoeren en assen kan laten bewegen;
2. Stappenummer met driver;
3. Bedrading.

Om de test uit te voeren moeten de volgende stappen worden uitgevoerd:

1. Bedraad alle componenten zoals aangegeven is in het elektrische schema;
2. Verbind de microcontroller met een laptop met seriële monitor;
3. Start de seriële monitor en stuur het commando G1 Z10;
4. Wacht totdat de machine 'OK.' geeft en de machine gaat bewegen;
5. Als de Z-as is bewogen stuur dan het commando G1 Z0 en wacht totdat de machine terug is bewogen naar het beginpunt.

Waarnemingen en conclusie Na het sturen van G1 Z10 commando stuurt de machine 'OK.' terug en gaat de as bewegen. Na het sturen van G1 Z0 beweegt de machine terug naar het beginpunt. Dit geeft aan dat de test geslaagd is.

D.5 X-richting meten

Voor deze requirement moet het prototype in de X-richting de afgelegde afstand kunnen bepalen doormiddel van een encoder. Deze test is voldaan als de afstand die de machine heeft afgelegd wordt bepaald door een encoder. Voor deze test moet de machine G-Code kunnen uitvoeren.

Deze test kan alleen op de machine worden uitgevoerd. Hiervoor zijn de volgende dingen nodig:

1. Microcontroller met code dat G-Code kan ontvangen en uitvoeren en de afstand kan bepalen met een encoder;
2. Encoder;
3. Bedrading.

Om de test uit te voeren moeten de volgende stappen worden uitgevoerd:

1. Bedraad de encoder met de microcontroller zoals aangegeven is in het elektrische schema;
2. Verbind de microcontroller met een laptop met seriële monitor;
3. Start de seriële monitor en stuur het commando \$? om de huidige informatie op te halen;
4. Beweeg de as met de hand en stuur het commando \$? om de huidige informatie op te halen;
5. Kijk of de het eerste getal veranderd is.

Waarnemingen en conclusie Er is geen encoder te bevestigen op deze as. Om deze reden is de test dus niet voltooid.

D.6 Y-richting meten

Voor deze requirement moet het prototype in de Y-richting de afgelegde afstand kunnen bepalen doormiddel van een encoder. Deze test is voldaan als de afstand die de machine heeft afgelegd wordt bepaald door een encoder. Voor deze test moet de machine G-Code kunnen uitvoeren.

Deze test kan alleen op de machine worden uitgevoerd. Hiervoor zijn de volgende dingen nodig:

1. Microcontroller met code dat G-Code kan ontvangen en uitvoeren en de afstand kan bepalen met een encoder;
2. Encoder;
3. Bedrading.

Om de test uit te voeren moeten de volgende stappen worden uitgevoerd:

1. Bedraad de encoder met de microcontroller zoals aangegeven is in het elektrische schema;
2. Verbind de microcontroller met een laptop met seriële monitor;
3. Start de seriële monitor en stuur het commando \$? om de huidige informatie op te halen;
4. Beweeg de as met de hand en stuur het commando \$? om de huidige informatie op te halen;
5. Kijk of de het eerste getal veranderd is.

Waarnemingen en conclusie Na het sturen van \$? geeft de machine overal 0 aan. Na het bewegen geeft de machine 17.2 aan dan 0 op de 1e plaats. Deze test is dus voltooid.

D.7 Z-richting meten

Voor deze requirement moet het prototype in de Z-richting de afgelegde afstand kunnen bepalen doormiddel van een encoder. Deze test is voldaan als de afstand die de machine heeft afgelegd wordt bepaald door een encoder. Voor deze test moet de machine G-Code kunnen uitvoeren.

Deze test kan alleen op de machine worden uitgevoerd. Hiervoor zijn de volgende dingen nodig:

1. Microcontroller met code dat G-Code kan ontvangen en uitvoeren en de afstand kan bepalen met een encoder;
2. Encoder;
3. Bedrading.

Om de test uit te voeren moeten de volgende stappen worden uitgevoerd:

1. Bedraad de encoder met de microcontroller zoals aangegeven is in het elektrische schema;
2. Verbind de microcontroller met een laptop met seriële monitor;

3. Start de seriële monitor en stuur het commando `$?` om de huidige informatie op te halen;
4. Beweeg de as met de hand en stuur het commando `$?` om de huidige informatie op te halen;
5. Kijk of de het eerste getal veranderd is.

Waarnemingen en conclusie Er is geen encoder te bevestigen op deze as. Om deze reden is de test dus niet voltooid.

D.8 Modulair

Voor deze requirement moet de code modulair zijn. Deze test is voldaan als er gemakkelijk een as toegevoegd kan worden.

Deze test kan alleen op de machine worden uitgevoerd. Hiervoor zijn de volgende dingen nodig:

1. Microcontroller;
2. Stappenmotoren(3) met drivers(3);
3. Encoder;
4. Bedrading.

Om de test uit te voeren moeten de volgende stappen worden uitgevoerd:

1. Volg de handleiding om de code aan te passen zodat de X en Z as stappen tellen en de Y werkt met de input van de encoder;
2. Verbind de microcontroller met een laptop met seriële monitor;
3. Start de seriële monitor en stuur het commando `G1 X10 Y10 Z10;`
4. Wacht totdat de machine 'OK.' geeft en de machine gaat bewegen.

Waarnemingen en conclusie Alle drie de motoren gaan bewegen en wanneer de locatie opgevraagd wordt met `$?` geeft het systeem het volgende aan: (0.0, 10.0, 0.0; 0.0, 10.0, 0.0; 0.0, 0.0, 0.0). Dit betekent dat de test geslaagd is.

D.9 Conclusie

Na testen is de conclusie dat de freesbank in alle richtingen bewogen kan worden, G-Code kan uitvoeren, modulair is en de locatie kan bepalen op de Y-as.

Soort	Requirement	Uitslag
Funct	De freesbank moet G-Code kunnen ontvangen en uitvoeren.	Behaald.
Funct	De freeskop moet in de X-richting kunnen bewegen.	Behaald
Funct	De freeskop moet in de Y-richting kunnen bewegen.	Behaald
Funct	De freeskop moet in de Z-richting kunnen bewegen.	Behaald
Funct	De positie van de freeskop op de X-as moet bepaald worden met een encoder.	Niet behaald
Funct	De positie van de freeskop op de Y-as moet bepaald worden met een encoder.	Behaald
Funct	De positie van de freeskop op de Z-as moet bepaald worden met een encoder.	Niet behaald
Niet. Funct	De software moet modulair zijn opgebouwd.	behaald

Changelog

Datum	verandering
1-12-2025	Begonnen met stageverslag maken van losse documenten
10-12-2025	Feedback verwerkt van studenten
05-01-2026	Document gereorganiseerd en analyse verbeterd
06-01-2026	Stakeholders verplaatst, spelling, onderzoek verplaatst / herschreven.
08-01-2026	Bijlage toegevoegd.
14-01-2026	Ontwerpen toegevoegd en aan realisatie gewerkt
27-01-2026	Realisatie ingevuld, testrapport toegevoegd. Spellingcheck
29-01-2026	Revision
30-01-2026	Revision
1-2-2026	Final revision