



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
2.1.1 Oplossingen	5
2.2 Nieuwe functies	5
2.2.1 Software	6
2.3 Requirements	7
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 Aanbevelingen	13
6 Bronnen	13
6.1 Datasheets	14
A Encoder onderzoek	15
A.1 Meet methodes	15
A.2 Encoder soorten	15
B elektrische schema's	18
C Plaatjes systeem	20
D Testrapport	21
D.1 Er kan G-Code uitgevoerd worden door de freesbank.	21
D.2 De snijkop van de machine moet over de X, Y en Z as kunnen bewegen.	21
D.3 De positie van de snij kop op een as moet bepaald worden doormiddel van sensoren.	22
D.4 Conclusie	23

Voorwoord Ik dank hierbij iedereen die heeft geholpen met de 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 komen, heb het gezellig gehad met de collega's. Ook heb ik veel nieuwe dingen geleerd zoals Freecad en regeltechniek. Hiernaast ben ik door de toepassingen van de wiskundige formules beter geworden in wiskunde en natuurkunde. Ook zijn de onderdelen die kapot aan kwamen een les om daar goed mee om te gaan. En ik denk een mooi prototype neergezet te hebben.

naam file
aanpassen

1 Context

1.1 Het bedrijf

De stage heeft plaatsgevonden bij het bedrijf Sens2Sea. Dit is een klein bedrijf dat gevestigd is op een kavel in het Innovation Dock op het RDM terrein. Sens2Sea ontwikkelt radar technologieën 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 hun vakgebieden zoals: elektronica en software. Verder lopen er bij Sens2Sea een aantal studenten stage van verschillende opleidingen. 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 bieden op gebied van elektronica en software. (Hoge invloed en Hoog belang)
2. PKMarine. Dit bedrijf werkt samen met Sens2Sea en bied expertise in regeltechniek. (Hoge invloed, laag belang)
3. CoE HRTech. Dit bedrijf financiert het project. (Lage invloed en hoog belang)

1.2 Opdracht

Sens2Sea is bezig met radar technologie. Radar is een technologie die radiogolven gebruikt om, t.o.v. 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 radar technologie gebruikt worden om de zeebodem in kaart te brengen en om olievlekken te detecteren.

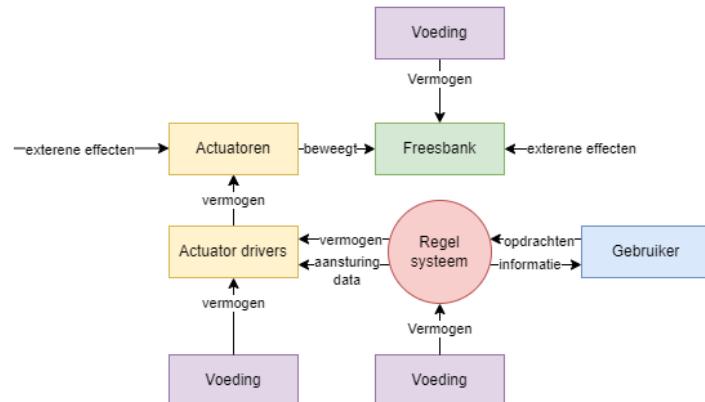
Sens2Sea is sinds kort bezig FMCW (Frequency Modulated Continuous Wave) radar technologie. Hiervoor wil Sens2Sea parabolantennes gebruiken. Deze antennes zorgen ervoor dat de radiogolven gebundeld worden. Door de bundeling van deze golven worden de signalen versterkt. Dit zorgt ervoor dat het beeld een betere resolutie heeft. Deze antennes zijn niet te verkrijgen in alle maten. Dit is omdat een antenne gemaakt wordt met een mal. Een mal maken is duur waardoor een antenne op maat een stuk duurder wordt dan een standaardmaat waardoor Sens2Sea graag zelf deze antennes wil maken. Er zijn hiervoor al testen geweest met antennes die in stukken geprint zijn met een 3D printer. Dit heeft goed uitgepakt, maar tijdens het hechten van de onderdelen kan er een fout in de vorm ontstaan waardoor de antenne minder werkt. Hierom wil Sens2Sea de antennes in één keer produceren met een 3D printer CNC machine combinatie. Deze machine moet 4m x 1m x 1m groot worden.

Er zijn opensource oplossingen voor CNC-machines en 3D printers. 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. GRBL is een voorbeeld van een opensource oplossing dat stappen telt om de gewenste locatie te bereiken. Hierdoor weet de microcontroller niet waar de snij kop zich exact bevindt. Hiernaast zijn er nog meer verschillende oplossingen. Deze oplossingen voldoen niet aan wat Sens2Sea wenst. Het bedrijf wenst dat de freesbank zijn aansturing en locatie bepaling gescheiden wordt. Hierdoor kan de machine nauwkeuriger werken. Ook wil Sens2Sea sensoren kunnen vervangen dus moet het systeem modulair worden. Hiernaast is er een wens om de machine over vijf assen te laten lopen. Dit zijn X, Y, Z en de rotaties over de X en Z as.

De scope van het project omvat het scheiden van de aandrijving en de positiebepaling van de snij kop. Daarnaast is er sprake van aanpassen of herschrijven van GRBL om deze te laten werken met de sensoren. Ook de hardware aansturen/aansluiten hoort bij de scope. Hier hoort het aansturen van de frees door de microcontroller niet bij.

2 Analyse

2.1 Huidige situatie



Freesbank: Huidige situatie

De huidige situatie bestaat uit een aantal functies die te zien zijn in het figuur: Freesbank: Huidige situatie. De functies die al aanwezig zijn, zijn de motor drivers, actuatoren, freesbank met frees en een regelsysteem. Hierin is het regelsysteem gebaseerd op GRBL en zoals er te zien is krijgt het regelsysteem geen terugkoppeling over waar de snij kop van de machine zich bevindt en of de stap succesvol gezet is.

verwijzing
datasheet?

Freesbank De freesbank is gemaakt van hout en heeft drie assen. De effectieve werkruimte bedraagt 251cm in de lengte(X-as), 127.5cm in de breedte(Z-as) en 8 cm in de hoogte(Y-as). De werkelijke grootte van de freesbank is 300cm(X) x 152cm(Z) x 60cm (Y). De poten van de machine is niet meegerekend in de Y-as grote. De freesbank bestaat uit drie assen met beweging. De X-as heeft aan beide kanten rails waarover een wagen rijdt. Deze wagen heeft een rails waarover een wagen rijdt. Op deze wagen zit een Schroef die de as beweegt als de schroef draait. De andere assen gebruiken timing pulleys op tandriemen voor de bewegingen. In appendix is dit te zien

nummer

Op de wagen van de hoogte-as is een handfrees gemonteerd. Deze frees werkt op netstroom. Als de frees snijdt, komt er materiaal los. Dit blijft liggen op de freesbank. De afzuiging van het materiaal het aansturen van de frees met de microcontroller of een nieuwe frees zoeken is buiten de scope van het project.

De freesbank heeft externe invloeden zoals wrijving op de rails en uitrekking van de tandriemen.

Actuatoren De motoren die worden gebruikt voor de beweging van de assen zijn NEMA 17 motoren. Deze werken op 12-36V en mogen maximaal 2A hebben. De motoren zitten vast aan het frame met bevestigingspunten. Aan de motoren zit een timing pulley bevestigd, waar een tandriem overheen loopt. De lengte as 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 '6.1 Datasheets'.

controleer
nummer

Drivers De drivers die worden gebruikt voor de motoren zijn DRV8824 drivers. Deze drivers werken op 8.2-45V en mogen maximaal 2.5A hebben op 24V. Deze drivers maken gebruik van een STEP en

DIR pin. Door de DIR pin hoog en laag te maken veranderd de polariteit over de aangesloten motor. Door de STEP pin hoog te maken zet de motor een stap. Deze drivers hebben ook de mogelijkheid om te microsteppen. Microsteppen maakt gebruik van sinusvormige stromen om meer nauwkeurigheid en hogere resolutie te krijgen. De drivers krijgen voeding van een lab voeding en het regelsysteem. De datasheet van deze driver is vinden in '6.1 Datasheets'.

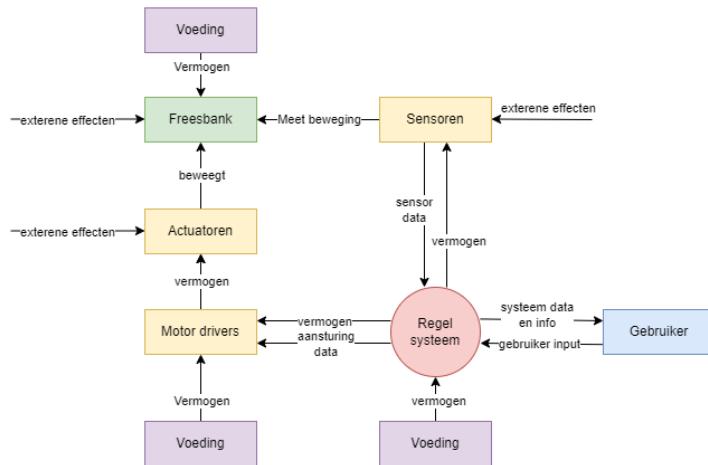
controleer nummer

Regelsysteem Het systeem wordt geregeld doormiddel van een Arduino Uno met een opensource regelsysteem GRBL. Dit is het onderdeel dat input van de gebruiker krijgt en informatie terugstuurt naar de gebruiker.

2.1.1 Oplossingen

Als eerste was de vraag er of een stappenumotor met ingebouwde encoders een oplossing zou zijn omdat dan de beweging logica van GRBL niet aangepast hoeft te worden. Dit is omdat deze encoders closed loop zijn zorgen ervoor dat de stappen motoren altijd op de juiste positie uitkomen. Dit is voorgelegd aan de opdrachtgever en die gaf aan dat dit niet de gewenste oplossing is, omdat er dan een afhankelijkheid van de fabrikant wordt gemaakt. Dit is omdat de fabrikant bepaald welke encoder op de stappenumotor komt. Hierdoor bepaald de fabrikant hoe nauwkeurig de freesbank kan zijn. Ook wil Sens2Sea het systeem gemakkelijk kunnen aanpassen om andere sensoren of actuatoren te gebruiken wanneer er meer nauwkeurigheid gewenst is.

2.2 Nieuwe functies



Freesbank: Nieuwe situatie

In het diagram is een nieuw component toegevoegd. Dit component zorgt voor de terugkoppeling van de beweging van de actuatoren naar het regelsysteem.

Sensor Uit een interview met de opdrachtgever is gebleken dat de opdrachtgever encoders wil gebruiken en deze wil bevestigen op een getande riem die op het frame is geplakt. Hierdoor kan de getande riem niet uittrekken. De nauwkeurigheid van de encoder is niet het belangrijkste omdat het project gaat over het scheiden van de aandrijving van de frees en positiebepaling van de snij kop. Verder doen de kosten

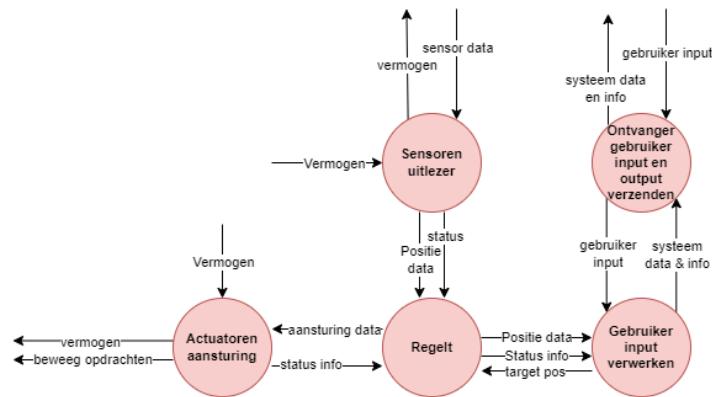
ertoe, maximaal 60 euro per encoder. Ook is er belang bij dat de huidige positie opvragen snel kan aangezien een freesbank een high performance machine is. Er zijn verschillende soorten encoders die verschillend werken. De nauwkeurigheid van een encoder wordt gemeten in Pulses Per Rotation(PPR). Dit geeft aan dat hoe hoger de PPR is de hoger de nauwkeurigheid omdat er meer pulsen gegeven worden in dezelfde rotatie. Na het onderzoek zijn de opties voorgelegd aan de opdrachtgever en is er gekozen om een optische incrementale encoder te gebruiken. Er zijn ook externe invloeden op de encoder zoals vuil dat in de encoder komt, dat de metingen kan verstören. De datasheet van deze sensor is te vinden in '6.1 Datasheets'.

controleer
nummer

Regelaar Als regelaar is een microcontroller nodig. De keuze van de encoder zorgt ervoor dat de microcontroller 6 interrupt pinnen nodig heeft. Verder moet deze rond de specificaties zijn als een Arduino Uno. Dit is omdat opensource oplossingen zoals GRBL gebaseerd zijn op een Arduino Uno. Verder werkt het bedrijf veel met Arduino Uno's. Hierom is voorgesteld om een Arduino Mega te gebruiken. Deze microcontroller heeft meer rekenkracht dan de Arduino Uno en heeft dus voldoende rekenkracht. Hiernaast heeft deze microcontroller ook de benodigde 6 interrupt pinnen voor de drie verschillende assen. Daarnaast bieden de vele pinnen mogelijkheden om het systeem uit te breiden.

Na de opties voorgelegd te hebben aan de opdrachtgever zijn er optische incrementale encoders (me38s6-c-(600)b5g2) besteld. Hiernaast is er ook een Arduino Mega besteld.

2.2.1 Software



Freesbank: Nieuwe software situatie

De software op de microcontroller bestaat uit verschillende onderdelen zoals te zien is in het figuur. Hoe de voeding binnen de microcontroller loopt is weggelaten. De datastromen zijn ook te zien in het figuur.

Actuatoren aansturing is het onderdeel wat de actuatoren aanstuurt. Dit onderdeel beïnvloed de pinnen om de motor driver aan te sturen.

Sensoren uitlezer is het onderdeel dat de sensoren uitleest. Hierna zet de sensor de data om naar bruikbare positie data.

Gebruiker input en output is het onderdeel dat de data ontvangt en verstuurd van de gebruiker.

Gebruiker input verwerker is het onderdeel dat de input van de gebruiker verwerkt en omzet naar bruikbare opdrachten. Ook zendt dit onderdeel opdrachten naar de regelaars. Ook handelt dit onderdeel alle andere opdrachten voor het systeem af. Dit zijn opdrachten zoals status opvragen en terugsturen, andere opdrachten voor de bewegingen die niet afgehandeld door de regelaars.

De regelaars zijn de onderdelen die per as bepalen hoe de actuatoren bewogen moeten worden. Dit doen ze door de positie data van de sensor uitlezer te vergelijken met de positie waar de actuator moet zijn.

2.3 Requirements

Uit de analyse en aan de hand van meerdere interviews met de opdrachtgever zijn de volgende requirements opgekomen.

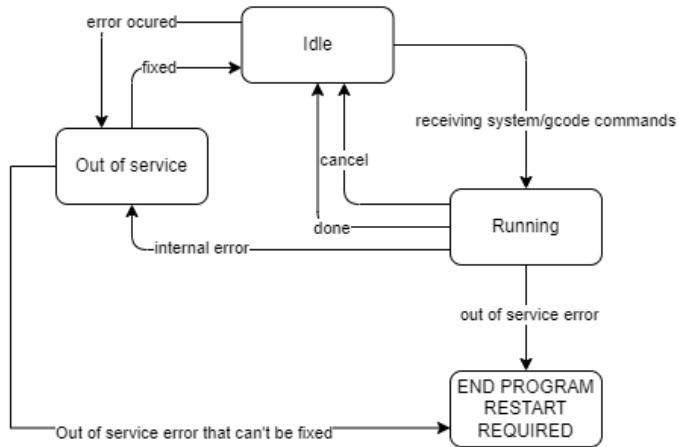
Requirement	User story	Acceptatie	soort
De snij kop van de machine moet over de X, Y en Z as kunnen bewegen.	Als opdrachtgever wil ik dat de machine zich over de X, Y en Z as kan bewegen.	Als de kop zich over de X, Y en Z as kan bewegen.	Funct.
De positie van de snij kop op een as moet bepaald worden doormiddel van sensoren.	Als opdrachtgever wil ik dat de machine de positie van de kop bepaald doormiddel van sensoren op de nauwkeurigheid te garanderen.	Als de positie van de snij kop op een as bepaald wordt door een sensor.	Funct.
Er kan G-Code uitgevoerd worden door de freesbank.	Als opdrachtgever wil ik een G-Code bestand laten uitvoeren door de freesbank.	Als de machine G-Code kan uitvoeren.	Funct.
De code van het prototype moet modulair zijn waardoor gemakkelijk delen vervangen kunnen worden, zoals actuatoren en sensoren.	Als opdrachtgever wil ik gemakkelijk onderdelen kunnen vervangen en toevoegen aan de machine, zonder dat de hele code omgebouwd moet worden.	Als er in de code van de machine componenten toegevoegd, vervangen of verwijderd kunnen worden.	Niet Funct.

3 Ontwerpen

3.1 Software ontwerpen

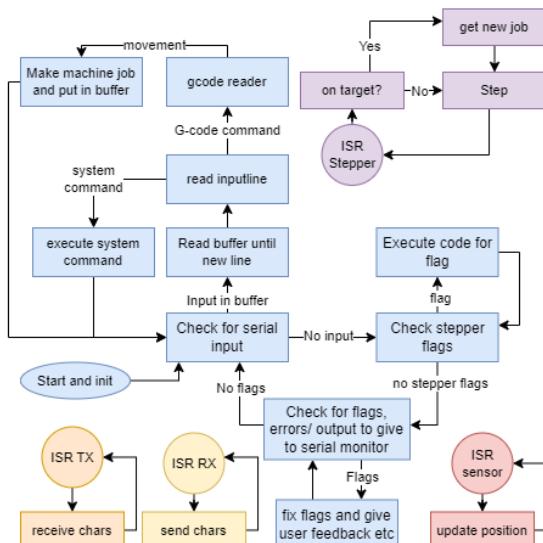
Voor de software zijn verschillende ontwerpen gemaakt. Dit zijn de state machine en flowchart.

State machine In software diagram A is de state machine 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. Indien het systeem een fout heeft dat deze zelf kan verhelpen veranderd de state naar 'Out of service'. Als de fouten niet te verhelpen zijn moet er een restart komen door een mens en zal de state 'END PROGRAM RESTART REQUIRED' worden. Deze state gaat niet weg zonder dat er een restart is geweest.



Software diagram A: State machine.

Flowchart In software diagram B is een flowchart te zien. De flowchart is verdeeld in meerdere kleuren. Oranje en geel zijn interrupts voor de input van de gebruiker. Rood is de interrupt van de sensoren en paars 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 die gelezen. Als het een systeem commando is zoals de status opvragen zal die meteen worden afgehandeld door 'execute system command'. Anders wordt de G-Code gelezen en in de queue gezet. Als het G-Code commando meteen afgehandeld moet worden, zal dat gedaan worden.



Software diagram B: Flowchart.

3.2 Elektrisch schema

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

check

4 Realisatie

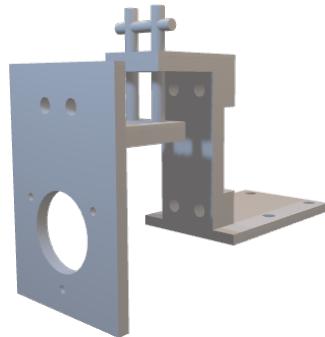
spelling check

4.1 Mechanica

De encoder moet lopen over een randriem die vastzit op de Z-as. Hiervoor is een beugel nodig. De opdrachtgever wil dat er een ontwerp komt dat met een veer ervoor zorgt dat de encoder altijd op de tandriem blijft lopen. Ook moet het model gemaakt worden in Freecad. Voor deze beugel zijn meerdere iteraties gemaakt. De eerste iteratie was instabiel waardoor er veel beweging was wanneer de freesbank bewoog. De oorzaak van dit probleem lag bij dat de staven tussen de twee bewegende delen heel dun waren. Hiernaast waren ze zo dun dat deze afbraken wanneer er een veer toegevoegd was. Ook had dit model een probleem dat deze niet goed paste omdat er geen rekening gehouden was met een metalen staaf waارlangs de Y-as beweegt.

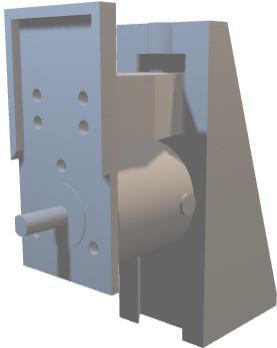
add picture link

link freecad



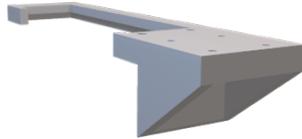
Mechanica: Iteratie 1

In de tweede iteratie zijn problemen opgelost. Deze iteratie maakt gebruik van een driehoek vormige gleuf waar het bewegende deel doorheen past. Ook is er ruimte voor een spijker met de veer om het bewegende deel omlaag te duwen. Hierdoor zit er geen speling op het ontwerp waardoor de encoder niet heen en weer beweegt bij plotselinge bewegingen. Dit ontwerp heeft een probleem dat de Y-as niet meer volledig naar beneden kan. Dit is omdat het ontwerp hoog is en recht onder het bewegende deel zit.



Mechanica: Iteratie 2

Voor de derde iteratie is er een plaatje gemaakt die vast zit op de plek waar iteratie één en twee vast zaten. Maar deze iteratie zorgt ervoor dat de bevestigingspunten verschuiven naar de zijkant. Hierdoor kan iteratie twee op dit plaatje geschroefd worden. Met dit ontwerp kan de Y as volledig naar boven en beneden bewegen. Dit ontwerp zorgt ervoor dat de Z-as ongeveer 4 cm kleiner wordt, maar dit is volgens de opdrachtgever niet erg voor het prototype.



Mechanica: Iteratie 3

4.2 Software

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

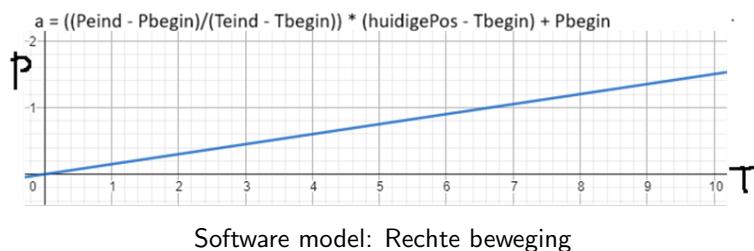
main Deze module definieert het begin van het programma. Hier bevindt zich de main loop waaruit alle andere functies van andere modules worden aangeroepen. Deze module initialiseert alle andere modules indien nodig en leegt de seriële buffer voor de zekerheid.

macro's Deze header file heeft alle macro's die nodig zijn voor het bestand. Macro's zijn woorden 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** 3 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. Hier 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 dat snel en nauwkeurig moet werken. Dit komt doordat de functies van deze library blokkerend zijn. Dit betekent dat wanneer de buffer op de Arduino vol zit omdat het niet snel genoeg verstuurd wordt of ontvangen wordt, de Arduino wacht met alles wat gedaan moet worden inclusief bewegingen totdat er weer characters op de buffer kunnen. 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.

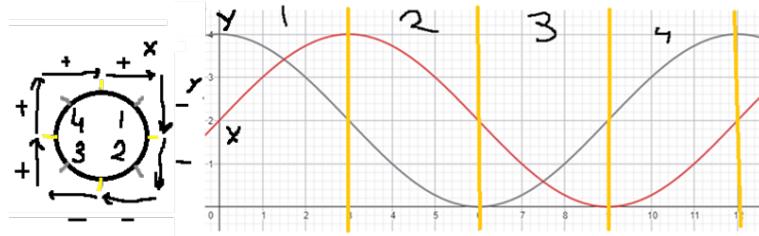
stepper Deze module zorgt voor alle functies van de steppers. Eerst wordt een segment(block) klaar gemaakt. Dit block heeft een aantal eigenschappen. Hieronder vallen alle parameters meegegeven met de G-Code commando's. Hiervan worden een paar extra parameters berekend die nodig zijn per stap. Het stappen 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: Rechte beweging, Cirkelbeweging. De commando's zijn te vinden in de handleiding.

De rechte beweging is per as 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 stappen motor. De Y-as(P) gaat van de beginpositie van die as 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.



Software model: Rechte beweging

De cirkelbeweging wordt gedaan met sinus en cosinus functies. Deze functies hebben een 4ste cirkel verschil met elkaar waardoor als deze gecombineerd worden er een cirkel beweging ontstaat. Ook deze functie gaat van t=0 naar t=eind. Maar deze word verschoven met de begin hoek. Als de cirkel tegen de klok in gaat wordt de 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 beweging kan maar op twee assen, dus zijn de commandos (G17-G19) om te kiezen op welk vlak de machine moet werken. De standaard is op het XZ vlak(G19).



Software model: Cirkelbeweging

encoder Deze module zet de pulsen van de encoders om in een locatie. Deze module zet interrupts aan op alle pinnen van de machine. 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. Hieronder is de functie om de encoder uit te lezen te zien. Deze code maakt slim gebruik van bitwise operaties om het verschil te vinden. Dit stukje code wordt heel vaak uitgevoerd waardoor het belangrijk is dat dit heel snel is. De encoders maken gebruik van de interrupt pinnen van de Arduino omdat deze 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 = (PINB >> ENCODER_PIN_ZB) & 1;
    //PORTB ^= (1 << PB5);
    //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 = (((PINB >> ENCODER_PIN_ZA) & 1) ^ b) | (b << 1);

    /**
     * Calculate movement direction using the current - previous. (&3 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;
}
```

Software: Encoder functie

system Deze module gebruikt de andere modules om alle opdrachten uit te lezen uit de buffers, om te zetten naar bruikbare info en in de buffer voor de stappen motoren zetten. Deze module zet de G-Code om naar een block en zet in de buffer. Hiernaast voert deze module ook alle systeem commando's uit zoals \$? om de huidige status uit te lezen.

4.3 Test Rapport

Zoals beschreven staat in het test rapport in appendix , zijn de testen behaald en voldoet het prototype welke aan de requirements. Hieronder zijn deze nog eens te zien.

Soort	Requirement	Uitslag
Funct	De positie van de snij kop op een as moet bepaald worden doormiddel van sensoren.	Behaald.
Funct	De snijkop van de machine moet over de X, Y en Z as kunnen bewegen.	Behaald
Funct	Er kan G-Code uitgevoerd worden door de freesbank.	Behaald
Niet. Funct	De code van het prototype moet modulair zijn waardoor gemakkelijk delen vervangen kunnen worden, zoals actuatoren en sensoren.	behaald

spelling check

5 Aanbevelingen

Bij deze opdracht zijn meerdere onderdelen aan bod gekomen waar dit project niet aan toe gekomen is.

1. Er zijn modules waarmee de encoders makkelijker uit te lezen zijn. Hier moet onderzoek naar gedaan worden.
2. rotzooi afzuigen frees
3. calibratie
4. zorgen volledige werkruimte benutten.
5. microcontroller en voeding niet mee laten bewegen met de freesbank, dus de alleen de kabels voor de stroom.
6. zorgen dat de machine geen beweegings commandos aanneemt als de buffer vol is.
7. pcb maken voor alle steppermotoren

aanbevelingen toevoegen

6 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.

6.1 Datasheets

1. Texas Instruments, "DRV8824 Stepper Motor Controller IC," Datasheet. [Online]. Available: <https://www.ti.com/lit/ds/symlink/drv8824.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/E38S6G5-600B-G24N>. Accessed on: Jan. 27, 2026.

code in bijlage linken

A Encoder onderzoek

Voor de positie van de snij kop van een freesbank te bepalen zijn verschillende mogelijkheden. Uit een interview met de opdrachtgever is gekomen dat deze encoders wilt gebruiken. Een encoder is een sensor met een roterende as. Als deze as draait zet de encoder de beweging om in elektrische signalen. Hierdoor 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 onder de 60 euro moet blijven.

A.1 Meet methodes

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

Magnetisch Magnetische encoders maken gebruik van een magnetisch veld om de rotatie van de as te bepalen. Dit heeft het voordeel dat deze robuuster zijn. Hiernaast 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 ook drie grote groepen magnetische encoders.

1. **Magnetic Gear Tooth Sensor or Pickup**, Deze encoder heeft een magnetische sensor en een ferromagnetische tandwielen. Dit betekent dat het tandwielen 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. Dit limiteert de resolutie naar 120 of 240 PPR.
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 halfgeleider materiaal 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.

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

A.2 Encoder soorten

Incrementeel, Incrementele encoders genereren pulsen als de as beweegt. Op de schijf zijn er 1 of meerdere rijen met patronen, die 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 krijg je bij 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 ten opzichte van een referentie punt,

A	B
0	0
1	0
1	1
0	1

Figure 1: Encoder sequentie

moet de microcontroller zelf de positie bijhouden. Hierdoor gaat de positiebepaling verloren als er een stroomonderbreking is.

Het aantal patronen per kanaal bepaalt de nauwkeurigheid. Indien een encoder meerdere kanalen heeft vergroot de nauwkeurigheid. bij 600 PPR (pulses per rotation) op één kanaal, heeft twee kanalen $600*4=2400$ PPR.

Het patroon van de signalen is in graycode. Dit patroon is te zien in figuur x. Graycode is een ander soort binaire code dat bij elke ophoging en verlaging één bit veranderd. De signalen moet je opvangen met een microcontroller via polling of interrupts. Interrupts zijn handiger omdat er dan geen data verloren kan gaan door een te kleine sample snelheid.

De signalen moeten worden opgevangen door een microcontroller. Dit kan via polling of interrupts. Interrupts stoppen het programma en voeren hun code uit, hierna kan het programma weer verder 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, 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. Absolute encoders onthouden niet standaard het aantal rotaties die zijn gedaan. Hiervoor zijn multi-turn varianten. Absolute encoders moeten worden uitgelezen met een protocol zoals SPI. Op een Absolute encoder bepaald het aantal bits hoe nauwkeurig de encoder is. Een multi-turn absolute encoder heeft x aantal bits per rotatie en x aantal bits voor het aantal rotaties.

Absolute encoders zijn duurder dan incrementele encoders. Van absolute zijn de multi-turn encoders 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 krijgen in een seconde. Deze test geeft een idee hoeveel de microcontroller kan hebben. Hierin wordt de tijd gemeten door de tijd van de serial monitor.

Op tijdstip 1: 09:36:47.772 waren er 848785 interrupts geweest.

Op tijdstip 2: 09:36:53.204 waren er 1484065 interrupts geweest.

Dit betekent dat er in 5.432 seconde 635280 interrupts zijn geweest.

$$5.432/635280 \approx 116951 \text{ interrupts per seconde.}$$

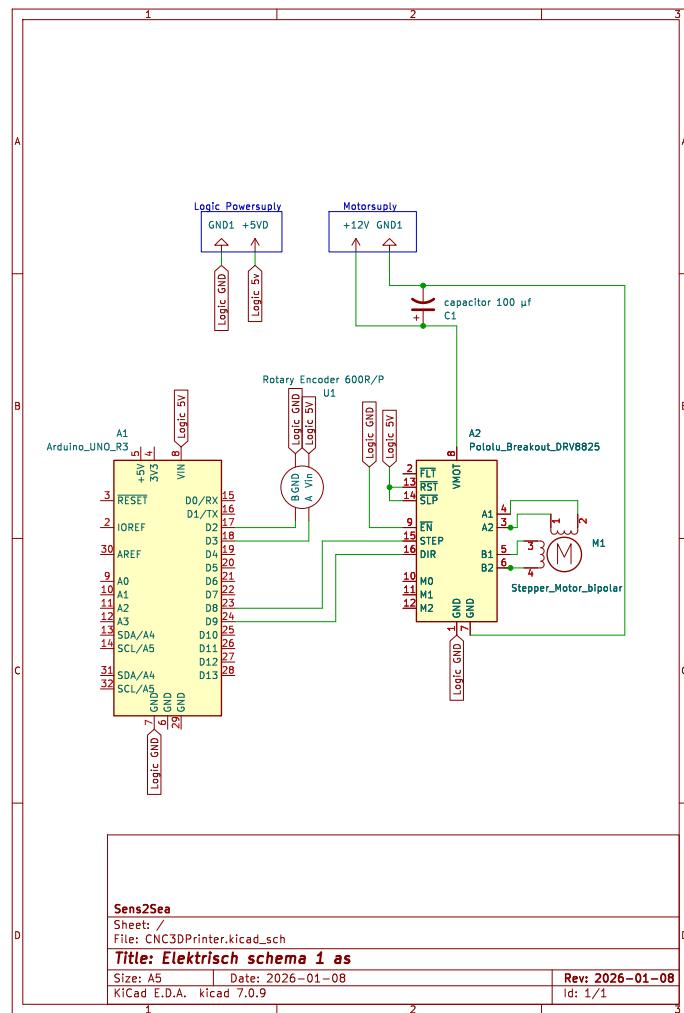
Een pulley van 16 tanden op een tandriem van 1mm tanden geeft 2mm op een tand. $2 * 16 = 32\text{mm}$ per rotatie. Indien er 20000 interrupts per as gebruikt kunnen worden door de sensoren kan de freesbank

figuur nummer

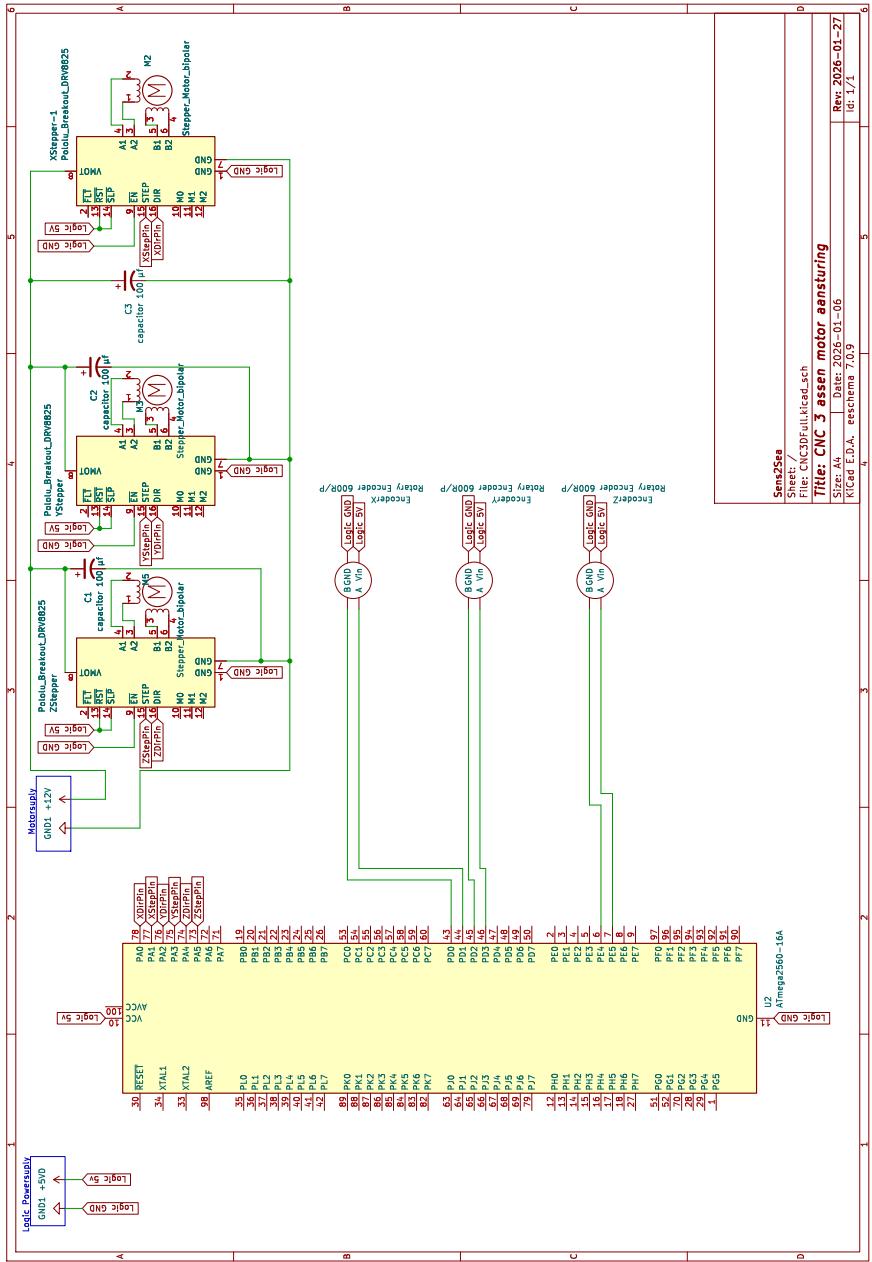
in theorie $20000/2400 * 32 \approx 266.6\text{mm/s}$ bewegen. Dit is genoeg dus deze soorten encoders zijn een optie.

Na de opties voorgelegd te hebben aan de opdrachtgever heeft de opdrachtgever 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 dual channel.

B elektrische schema's

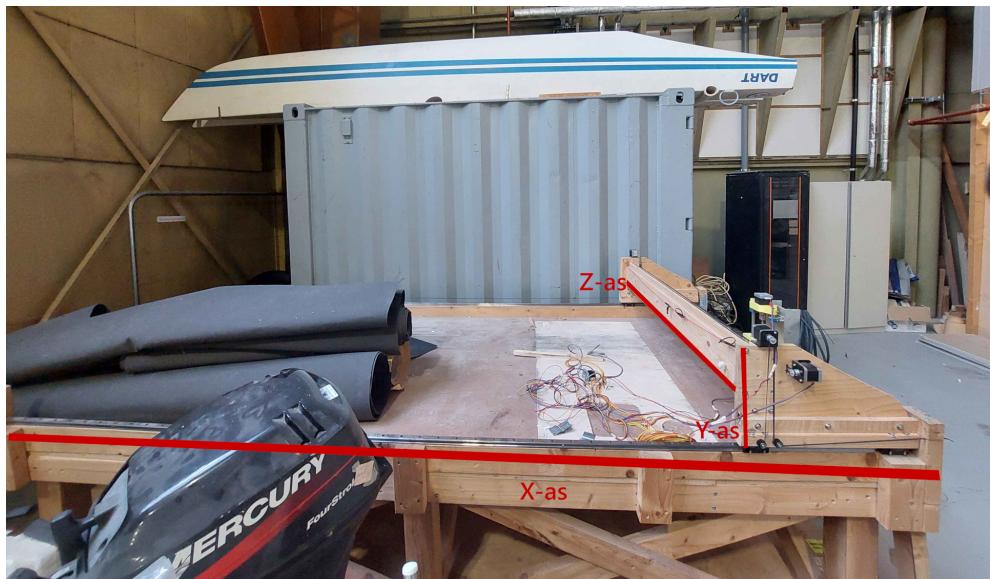


Elektrisch schema één as



Elektrisch schema drie assen

C Plaatjes systeem



Zij aanzicht



Voor aanzicht

D Testrapport

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

1. De snijkop van de machine moet over de X, Y en Z as kunnen bewegen.
2. De positie van de snij kop op een as moet bepaald worden doormiddel van sensoren.
3. Er kan G-Code uitgevoerd worden door de freesbank.
4. De code van het prototype moet modulair zijn waardoor gemakkelijk delen vervangen kunnen worden, zoals actuatoren en sensoren.

De vierde requirement is niet functioneel en is behaald als de code makkelijk is aan te passen voor de verschillende testen.

D.1 Er kan G-Code uitgevoerd worden door de freesbank.

Deze test heeft de volgende acceptatie criteria: De test is voldaan als de machine G-Code commandos uitvoert.

Testopstelling Deze test kan alleen op de machine uitgevoerd worden. Benodigd hiervoor is:

1. Freesbank een as of meer verbonden met stappen motor(s).
2. Microcontroller met code om G-Code commando's te ontvangen en uit te voeren.

Instructies

1. Verbind alle stappemotoren met de microcontroller op de manier aangegeven op het elektrische schema(1 motor is genoeg).
2. Ze op de microcontroller de code om de machine te laten bewegen.
3. Voer het G-Code commando uit om een rechte lijn te bewegen op de aangesloten as.
4. Wacht totdat de machine klaar is met bewegen en laat de machine terug bewegen.

Verwacht resultaat Het verwachte resultaat is dat de machine gaat bewegen wanneer deze een G-Code command krijgt en OK. stuurt.

Waarnemingen en conclusie De machine stuurt OK. terug en begint te bewegen naar het punt. Na het volgende commando stuurt de machine weer OK. en beweegt die terug. Dit betekent dat deze test voldaan is.

D.2 De snijkop van de machine moet over de X, Y en Z as kunnen bewegen.

Deze test heeft de volgende acceptatie criteria: De test is voldaan als de machine over de X, Y en Z as kan bewegen.

Testopstelling Deze test kan alleen op de machine uitgevoerd worden. Ook is hiervoor nodig dat G-Code uitgevoerd kan worden. Benodigt hiervoor is:

1. Freesbank met drie assen verbonden met stappen motors.
2. Microcontroller met code voor drie assen en G-Code commando's te ontvangen en uit te voeren.

Instructies

1. Verbind alle stappemotoren met de microcontroller op de manier aangegeven op het elektrische schema.
2. Ze op de microcontroller de code om de machine te laten bewegen.
3. Voer het G-Code commando uit om een rechte lijn te bewegen naar ($X=10, Y=2, Z=5$).
4. Wacht totdat de machine klaar is met bewegen en laat de machine naar het 0 punt bewegen.
5. Wacht totdat de machine klaar is met bewegen en laat de machine met een halve cirkel naar ($X=10, Y=0, Z=0$) bewegen met centerpunt ($X=5, Z=0$)

Verwacht resultaat Het verwachte resultaat is dat de machine eerst naar ($X=10, Y=2, Z=5$) beweegt, daarna terug gaat naar het 0 punt en hierna een halver cirkel maakt naar ($X=10, Y=0, Z=0$)

Waarnemingen en conclusie De machine beweegt naar ($X=10, Y=2, Z=5$) en stopt. Hierna beweegt de machine terug naar het 0 punt. De machine beweegt met een halve cirkel naar $X=10$. Dit betekend dat deze test voldaan is.

D.3 De positie van de snij kop op een as moet bepaald worden doormiddel van sensoren.

Deze test heeft de volgende acceptatie criteria: De test is voldaan als de machine de locatie kan bepalen op een as doormiddel van sensoren.

Testopstelling Deze test kan alleen op de machine uitgevoerd worden. Ook is hiervoor nodig dat G-Code uitgevoerd kan worden. Benodigt hiervoor is:

1. Freesbank met een as verbonden met stappen motor en encoder.
2. Microcontroller met code voor een of meer assen, encoder locatie bepaling en G-Code commando's te ontvangen en uit te voeren.

Instructies

1. Verbind alle stappemotoren met de microcontroller op de manier aangegeven op het elektrische schema(1 is genoeg).
2. Verbind een encoder van de as met een stappemotor met de microcontroller zoals aangegeven op het elektrische schema.
3. Ze op de microcontroller de code om de machine te laten bewegen en locatie bepalen doormiddel van een encoder.

4. Voer het G-Code commando uit om een rechte lijn over de as.
5. Wacht totdat de machine klaar is met bewegen en vraag de locatie op.
6. Controleer of de locatie is veranderd naar de juiste richting en dat de locatie overeen komt met de locatie uit het commando.

Verwacht resultaat Het verwachte resultaat is dat de machine beweegt over de as en dat de encoder de bewegingen meet. Verwacht wordt dat de encoder het getal aangeeft dat in het commando is ingevoerd.

Waarnemingen en conclusie De machine beweegt over de as en de positie telt mee op. De machine stopt als de encoder de locatie uit het commando bereikt. Dit betekend dat deze test voldaan is.

D.4 Conclusie

Na testen is de conclusie dat alle requirements gehaald zijn.

Soort	Requirement	Uitslag
Funct	De positie van de snij kop op een as moet bepaald worden doormiddel van sensoren.	Behaald.
Funct	De snijkop van de machine moet over de X, Y en Z as kunnen bewegen.	Behaald
Funct	Er kan G-Code uitgevoerd worden door de freesbank.	Behaald
Niet. Funct	De code van het prototype moet modulair zijn waardoor gemakkelijk delen vervangen kunnen worden, zoals actuatoren en sensoren.	behaald

Changelog

Datum	verandering
1-12-2025	Begonnen met stage verslag 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.