

# GIP 6TW

## WHITEBOARD-VEGER



3<sup>de</sup> graad 2<sup>de</sup> jaar



Naam: Kobe Nuyts

Klas: *6 Technologische Wetenschappen*

Mentor: Yves Verbruggen

Schooljaar: *2024 – 2025*



# Inhoud

1	Inleiding .....	4
2	Wetenschappelijk deel .....	5
2.1	Opdracht en Onderzoeksdoel .....	5
2.1.1	De chemie van inkt .....	5
2.1.2	Metingen en analyse .....	5
2.2	Ontstaansgeschiedenis .....	5
2.2.1	De oorsprong van inkt .....	5
2.2.2	Atramentuminkt .....	6
2.2.3	De evolutie van drukinkten .....	6
2.2.4	Soorten inkt en hun toepassingen .....	6
2.2.5	Moderne ontwikkelingen en ecologische impact .....	6
2.3	Chemisch deel .....	8
2.3.1	Samenstelling whiteboard-inkten .....	8
2.3.2	Werking van whiteboard inkten op moleculair niveau .....	14
2.4	Regressieanalyse .....	19
2.4.1	Inleiding .....	19
2.4.2	Onderzoek en Analyse .....	19
2.4.3	Meetresultaten .....	21
2.4.4	Berekeningen .....	22
2.4.5	Interpretatie Berekende Regressie .....	25
2.4.6	Gevolgen Motorprestaties .....	27
2.4.7	Besluit .....	27
2.4.8	Sjabloon .....	28
3	Technisch deel .....	29
3.1	Opdracht .....	29
3.2	Functionele eisen .....	29
3.3	Onderdelen .....	30
3.3.1	Arduino Mega .....	30
3.3.2	TB6600 Stepper Driver .....	30
3.3.3	SL42S247A stappenmotor .....	33
3.3.4	SL42S234MA107 stappenmotor .....	34
3.3.5	LM2596 DC-DC Step-Down Converter .....	36
3.3.6	Knoppen .....	36
3.3.7	Voeding .....	39
3.3.8	Zekeringen .....	41
3.4	Schakelkast Overzicht .....	42
3.5	Maquetteschets .....	44
3.6	Modules .....	45
3.6.1	X-as module .....	45
3.6.2	Z-as modules .....	45
3.7	Technische tekeningen .....	46
3.8	Bedradingschema .....	49
3.9	Blokschema .....	50
4	Code: .....	51

4.1	Inleiding tot de Arduino Code .....	51
4.2	Flowchart .....	51
4.3	Overzicht van de Code .....	52
4.4	Wat doet de code? .....	52
4.4.1	Bibliotheken .....	52
4.4.2	Setup ( <b>setup()</b> ) .....	52
4.4.3	Hoofdprogramma ( <b>loop()</b> ) .....	52
4.4.4	Bewegingsfuncties ( <b>moveAxis()</b> ) .....	52
4.4.5	Stopfunctie (stopAllMotors()) .....	52
4.4.6	Pinconfiguratie .....	52
4.4.7	Code .....	55
4.4.8	Bestand code .....	61
5	Ondervonden problemen .....	62
6	Handleiding .....	66
6.1	Overzicht van het Bedieningspaneel .....	66
6.2	Belangrijke Veiligheidsinformatie .....	66
6.3	Bediening van de Bordveger .....	67
6.3.1	Handmatige Besturing .....	67
6.3.2	Geautomatiseerde Besturing .....	67
6.4	LED-indicator .....	68
6.5	Probleemoplossing (Snelle Tips) .....	68
7	Gebruikte programma's .....	69
8	Proces .....	70
9	Besluit .....	73
10	Dankwoord .....	75
10.1	Nederlands .....	75
10.2	Frans .....	76
	Engels .....	77
11	Bronnen .....	78

# 1 Inleiding

Mijn naam is Kobe Nuyts en ik zit momenteel in mijn zesde jaar Technologische Wetenschappen en engineering (6TWe) aan het GTI Mortsel. Gedurende de afgelopen zes jaar heb ik mij verdiept in deze richting en heb ik de kans gekregen om verschillende technische vaardigheden te ontwikkelen. Deze richting heeft me altijd geboeid omdat we de theorie meteen in de praktijk kunnen toepassen. Bovendien daagt het me uit om creatieve oplossingen te vinden voor technische problemen, wat ik erg interessant vind.

Voor mijn Geïntegreerde Proef (GIP) koos ik een project dat niet alleen technisch uitdagend is, maar ook praktisch nut heeft. Uiteindelijk koos ik ervoor om een automatische bordveger te maken voor mijn leerkracht wiskunde. Het idee kwam dan ook voort uit een opmerking van mijn leerkracht, die al lange tijd een oplossing zocht om whiteboards snel en efficiënt schoon te maken. Dit idee inspireerde me waaruit de keuze van mijn Geïntegreerde Proef verder kwam. In mijn ogen is een automatische bordveger een origineel en uitdagend concept dat tegelijkertijd een praktisch probleem oplost en een interessante uitdaging biedt.

Wat dit project voor mij extra interessant maakt, is dat het verschillende aspecten, die ik eerder heb geleerd in mijn richting, combineert. Ik moet niet alleen mechanische onderdelen ontwerpen en monteren, maar ook elektronica integreren en een gebruiksvriendelijke interface programmeren. Bovendien vereist het een grondige analyse van functionele eisen en technische specificaties. Met dit project kan ik dan ook mijn kennis over technologie gebruiken voor iets praktisch en laten zien wat ik tijdens mijn opleiding heb geleerd.

Mijn doel is om een automatische bordveger te creëren die zelfstandig kan functioneren met één druk op een knop. Daarnaast wil ik een modern ontwerp afleveren dat efficiënt werkt, veilig in gebruik is en eenvoudig bediend kan worden. Ik zie dit project niet alleen als een afsluiting van mijn schoolcarrière, maar ook als een kans om mezelf verder te ontwikkelen en voor te bereiden op mijn volgende studies in deze richting.

## 2 Wetenschappelijk deel

### 2.1 Opdracht en Onderzoeksdoel

In dit deel van mijn GIP onderzoek ik de eigenschappen en productie van inkt, met een focus op whiteboard-inkten. Ik zoek een antwoord om de centrale vraag:

*"Hoe werken whiteboard-inkten en welke chemische eigenschappen maken ze geschikt voor droog uitwissen?"*

#### 2.1.1 De chemie van inkt

Ik zal beginnen met te onderzoeken waar inkt vandaan komt en waarom bepaalde kleuren in het verleden zo belangrijk waren. Daarna ga ik bekijken welke soorten inkt er bestaan en of elke soort stift geschikt is om te schrijven op een whiteboard.

Voor whiteboard-inkten moet ik een overzicht maken van de belangrijkste stoffen die erin zitten. Ik zal tekeningen maken van de moleculen om te laten zien hoe ze eruitzien en uitleggen hoe ze in elkaar zitten.

Verder zal ik onderzoeken hoe whiteboard-inkt tegenwoordig gemaakt wordt. Ik ga uitzoeken welke processen daarbij een rol spelen, zoals het mengen van de juiste stoffen. Tot slot zal ik een kort overzicht geven van oplosmiddelen die geschikt zijn voor gebruik in whiteboard-inkten.

#### 2.1.2 Metingen en analyse

Eerst zal ik metingen uitvoeren en gegevens verzamelen. Deze gegevens zet ik vervolgens in overzichtelijke tabellen en bereken de gemiddelden om de resultaten te analyseren. Ook zal ik de nauwkeurigheid van de gegevens controleren.

Daarna moet ik grafieken maken om de data visueel weer te geven. Ik ga op zoek naar verbanden in de gegevens. Dit zal mij helpen om een duidelijk beeld van de gegevens te krijgen en uiteindelijk een sterke conclusie te trekken.

## 2.2 Ontstaansgeschiedenis

Inkt, een vloeibaar kleurmiddel dat essentieel is voor schrijven en drukken, heeft een rijke geschiedenis die duizenden jaren beslaat.

### 2.2.1 De oorsprong van inkt

De eerste vormen van inkt ontstonden rond 500 v.Chr. in China en Egypte. In China werd India-inkt ontwikkeld, gemaakt van beenderlijm en roet (ook wel zwartsel genoemd). Deze inkt werd in blokken gegoten en kon opnieuw vloeibaar worden gemaakt met een vochtige kwast. In Egypte werd inkt voornamelijk gebruikt voor hiërogliefen op papyrus. Hiernaast vind je een afbeelding van een traditionele inktblok en een voorbeeld van kalligrafie uit die periode:



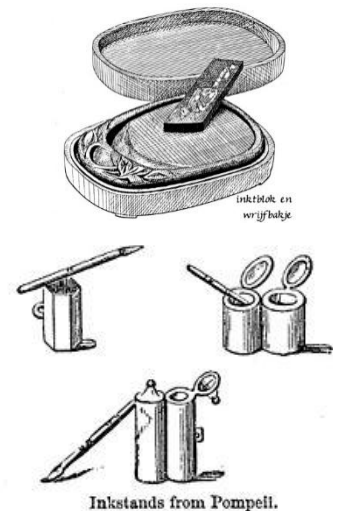
Figuur 1: Traditionele inktblok

Hier is een filmpje over hoe het vroeger gebruikt werd:  
<https://youtu.be/v6GiONcIiaA?si=asbIQAJOSIB1LIVC>

## 2.2.2 Atramentuminkt

Rond 400 na Chr. werd Atramentuminkt populair in Rome. Deze inkt, gemaakt van ijzerzouten gemengd met olie uit galnoten en verdikkingsmiddelen, bracht chemische innovaties met zich mee. Atramentuminkt kleurde pas bij blootstelling aan lucht en werd vaak aangevuld met kleurstoffen zoals sepia en indigo. Hoewel effectief, leidde de zure samenstelling tot inktvraat, een probleem dat in historische documenten zichtbaar is.

Video over hoe Atramentuminkt gemaakt wordt:  
<https://youtu.be/8njtIyUyyaY?si=vVMEJ0NHo-Kyg72T>



*Figuur 2: Atramentuminktblok en penseel*

## 2.2.3 De evolutie van drukinkten

Met de opkomst van de boekdrukkunst in Europa bleek Atramentuminkt ongeschikt vanwege het vlekken karakter. Nieuwe olieachtige inkten met terpentijnharsen werden ontwikkeld. In de 20e eeuw zorgden synthetische kleurstoffen en sneldrogende formules voor grote vooruitgang in inktproductie. Deze innovatie leidde niet alleen tot betere drukinkten, maar inspireerde ook andere toepassingen zoals balpeninkt, ontwikkeld door László Bíró.

## 2.2.4 Soorten inkt en hun toepassingen

Inksoorten zijn door de eeuwen heen diverser geworden. Naast traditionele inkten zoals Indiase inkt en vulpeninkt zijn er nu ook gespecialiseerde inkten zoals ecologische inkten en whiteboard-inkten. Hier is een overzicht van enkele belangrijke soorten:









1. **Traditionele inkt:** Indiase inkt en kalligrafie-inkten.
2. **Balpeninkt:** Geoptimaliseerd voor snelle droging en gebruiksgemak.
3. **Whiteboard-inkt:** Speciaal ontwikkeld voor droog wissen zonder sporen.
4. **Ecologische inkt:** Milieuvriendelijk en veilig.

## 2.2.5 Moderne ontwikkelingen en ecologische impact

Vandaag de dag wordt inkt niet alleen gebruikt voor schrijven, maar ook in printers, verpakkingen en kunst. De groeiende vraag naar milieuvriendelijke inkten heeft geleid tot nieuwe uitvindingen zoals plantaardige pigmenten en biologisch afbreekbare samenstellingen. Deze ontwikkeling benadrukt de voortdurende evolutie van inkt en de aanpassing aan moderne behoeften.

Hier is een tabel waarin de verschillende inktsoorten en hun eigenschappen worden samengevat:

Tabel 1: Soorten inkt samengevat

<b>Soort inkt:</b>	<b>Samenstelling:</b>	<b>Eigenschappen:</b>	<b>Toepassingen:</b>	<b>Afbeelding:</b>
<b>Indiase inkt</b>	Roet (zwartsel) en beenderlijm	Diepzwart, goed voor kalligrafie, niet uitwisbaar	Traditionele kalligrafie en schilderkunst	
<b>Atramentuminkt</b>	IJzerzouten, olie uit galnoten, kleurstoffen	Verkleurt bij luchtcontact, zuurhoudend, kan inktvraat veroorzaken	Historisch schrijven en documenten	
<b>Drukinkt</b>	Synthetische harsen, oliën, kleurstoffen	Stroperig, sneldrogend, vlek-vrij	Boekdrukkunst, drukpersen	
<b>Balpeninkt</b>	Olieachtig, oplosmiddelen, kleurstoffen	Sneldrogend, geschikt voor balpennen, blijft lang vloeibaar	Dagelijks schrijven	
<b>Whiteboard-inkt</b>	Oplosmiddelen, niet-permanente kleurstoffen	Droogt snel, makkelijk uitwisbaar, veilig voor gebruik op borden	Whiteboards in scholen en kantoren	
<b>Vulpeninkt</b>	Water, kleurstoffen, conserveermiddelen	Vloeibaar, geschikt voor vulpennen, vaak uitwisbaar	Schrijven met vulpennen	
<b>Tatoeage-inkt</b>	Pigmenten, oplosmiddelen, conserveringsmiddelen	Veilig voor huidgebruik, lang houdbaar	Tatoeages	
<b>Ecologische inkt</b>	Plant aardige of biologisch afbreekbare ingrediënten	Milieuvriendelijk, niet-giftig, vaak waterbasis	Kunst, schrijven, printen	



## 2.3 Chemisch deel

### 2.3.1 Samenstelling whiteboard-inkten

Whiteboardmarkers, ook wel droog-uitwisbare markers genoemd, zijn speciaal ontworpen om op gladde oppervlakken te schrijven en gemakkelijk te kunnen worden gewist. Daarom zijn ze ontworpen met een specifieke chemische samenstelling.

De inkt is opgebouwd uit 5 basiscomponenten met elk hun eigen specifieke functie. Deze componenten zijn terug te vinden in de informatiebladen over de materiaalveiligheid van Whiteboardmarkers:

Tabel 2: Componenten Whiteboardmarkers

	<b>Component:</b>	<b>Functie:</b>	<b>Percentage:</b>
1	Oplosmiddelen	Houden alle pigmenten en additieven in oplossing, zorgen voor vloeibaarheid en snelle verdamping.	60-70%
2	Pigmenten	Geeft kleur aan de Whiteboardmarker	10-20%
3	Harsen	Verspreiden pigmenten gelijkmatig op het oppervlak en zorgen voor hechting aan het whiteboard, vormt een film wanneer de inkt droogt.	7-9%
4	Polymeren (losmiddelen)	Vormen een beschermende laag en vergemakkelijken het wissen.	1-5%
5	Wax	Voorkomen uitdroging van de markerpunt en stabiliseren de inkt	<0,5%

Hieronder vindt u een gedetineerde uitleg en diepgang voor alle componenten:

#### 2.3.1.1 Oplosmiddelen

Oplosmiddelen zijn zeer belangrijk in whiteboard-inkten omdat ze alle pigmenten en componenten in oplossing houden en de inkt vloeibaar maken om deze te kunnen aanbrengen op het oppervlak van het bord en er vervolgens voor zorgt dat deze weer snel opdroogt.

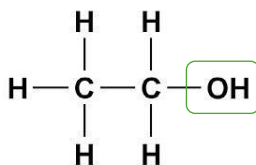
De meest gebruikte nodige basisoplosmiddelen zijn:

##### 1) **Ethanol** (C<sub>2</sub>H<sub>5</sub>OH)

- **Chemische structuur:** CH<sub>3</sub>CH<sub>2</sub>OH

- **Lewisstructuur:**

Dit molecuul heeft een hydroxylgroep (-OH) gebonden aan een ethanolkern.



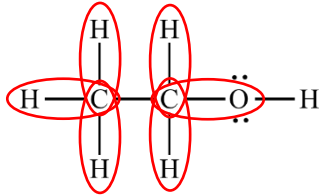
Figuur 4: Lewis Structuur ethanol

formula	CH <sub>3</sub> CH <sub>2</sub> OH
Hill formula	C <sub>2</sub> H <sub>6</sub> O
name	ethanol

Figuur 3: Chemische namen en formules Ethanol

- **Bindingen:**

- Covalente bindingen (CB) tussen de koolstof (C), Waterstof (H) en zuurstof (O) atomen. (C-H en C-O)



Figuur 5: Covalente bindingen ethanol

- Waterstofbruggen tussen de hydroxylgroepen (-OH) Van ethanolmoleculen.

	bond counts
C—C	1 bond
C—H	5 bonds
C—O	1 bond
H—O	1 bond

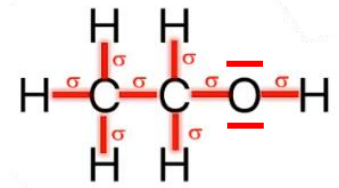
Figuur 6: Informatie bindingen Ethanol

- **Hybridisatie:**

Ethanol bestaat uit een ethaangroep (CH<sub>3</sub>CH<sub>2</sub>) en een hydroxylgroep (-OH). Deze hybridisatie van de atomen in ethanol bepaalt de moleculaire geometrie en fysisch-chemische eigenschappen. Hieronder leg ik de hybridisatie van elk relevant atoom gedetailleerd uit:

1) **Koolstofatomen (C) in de Ethaangroep:**

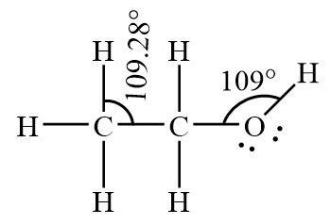
Elk koolstofatoom vormt **vier enkelvoudige covalente sigma bindingen** (met waterstof of andere koolstofatomen). Omdat er vier gebieden met elektronendichtheid zijn (vier bindingen) rond elk koolstofatoom. Het centrale koolstofatoom is sp<sup>3</sup>-gehybridiseerd dit zorgt voor een tetraëdrische geometrie met een bindingshoek van ongeveer 109,5°.



Figuur 7: Covalente sigma bindingen Ethanol

2) **Zuurstofatoom (O) in de Hydroxylgroep (-OH):**

Het Zuurstofatoom heeft zoals ook eerder vermeld twee bindingen. Eén met waterstof (O-H) en één met koolstof (C-O). Er zijn ook twee vrije elektronenparen op het zuurstofatoom. Hierdoor ontstaat er een **sp<sup>3</sup> Hybridisatie** wat zorgt voor een gebogen vorm (vergelijkbaar met die van water). En een Bindingshoek van ongeveer 109° (dit is minder dan ideaal tetraëdrisch door afstoting van de vrije elektronenparen).



Figuur 8: Bindingshoeken ethanol

Of

## 2) Isopropanol (C<sub>3</sub>H<sub>8</sub>O)

- **Chemische structuur:** CH<sub>3</sub>-CHOH-CH<sub>3</sub>

- **Lewisstructuur:**

(voor een correcte weergave worden de elektronenparen rondom het zuurstofatoom getekend omdat het zuurstofatoom twee vrije elektronenparen heeft.)

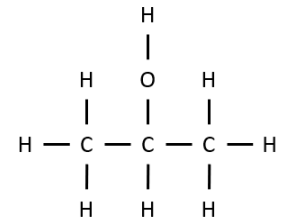
formula	(CH <sub>3</sub> ) <sub>2</sub> CHOH
Hill formula	C <sub>3</sub> H <sub>8</sub> O
name	isopropanol
IUPAC name	propan-2-ol

Figuur 9: Chemische namen en formules Isopropanol

- **Bindingen:**

Covalente bindingen (CB):

- Tussen koolstof (C) en waterstof (H) atomen: C-H bindingen in zowel de methylgroepen (CH<sub>3</sub>) als het centrale CH-gedeelte.
- Tussen koolstof (C) en zuurstof (O) atomen: C-O binding tussen het centrale koolstofatoom en de hydroxylgroep (-OH).
- Tussen zuurstof (O) en waterstof (H): O-H binding binnen de hydroxylgroep.



Figuur 10: Lewis Structuur Isopropanol

Waterstofbruggen:

- De -OH groepen kunnen waterstofbruggen vormen met de -OH groepen van aangrenzende isopropanolmoleculen

- **Hybridisatie:**

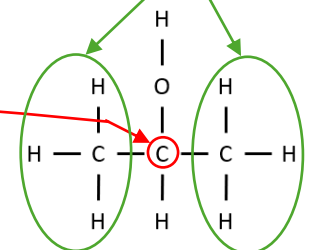
Isopropanol bestaat uit een ethaangroep waarin twee soorten groepen:

**a. Koolstofatomen (C) in de ethaangroep:**

- Methylgroepen (CH<sub>3</sub>):

Elk koolstofatoom in de methylgroep vormt **drie enkelvoudige covalente sigma bindingen** met waterstof en één binding met het centrale koolstofatoom. Hierdoor ontstaat er een **sp<sup>3</sup> Hybridisatie** en met een Tetraëdrische vorm met een bindingshoek van ongeveer 109,5°. Deze begrippen worden verder verklaart in 2.3.2

- Centraal koolstofatoom (verbonden met -OH en twee CH<sub>3</sub>): Dit atoom is eveneens omringd door vier elektronengebieden (drie bindingen met de twee methylgroepen en de -OH groep) en is daardoor sp<sup>3</sup>-gehybridiseerd. (Geometrie: Tetraëdrisch)



Figuur 11: Bindingen Isopropanol

**b. Zuurstofatoom (O) in de Hydroxylgroep (-OH):**

In totaal vormt het zuurstofatoom twee sigma bindingen één met waterstof (O-H) en één met het centrale koolstofatoom (C-O). Daarnaast bezit het ook twee vrije elektronenparen en is het sp<sup>3</sup>-gehybridiseerd. De vier elektronengebieden (twee bindingen en twee vrije elektronenparen) zorgen voor een tetraëdrische vorm, maar door afstoting van de vrije elektronenparen ontstaat er een **gebogen** moleculaire vorm. Met een bindingshoek van Ongeveer 104° (dit is iets minder dan de ideale tetraëdrische hoek).

### 2.3.1.2 Pigmenten:

## 1. Koperftalocyanine (CuPc)

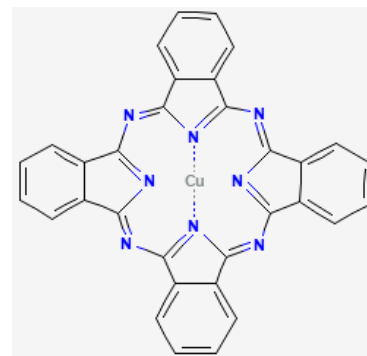
Koperftalocyanine (CuPc) is het belangrijkste blauwe pigment in whiteboard-inkten vanwege zijn **uitstekende kleursterkte, stabiliteit en niet-permanente hechting**. Hieronder bespreek ik de chemische structuur, bindingen en functionaliteit gedetailleerd:

- **Chemische Structuur:**  $C_{32}H_{16}CuN_8$

De structuur bestaat uit een macrocyclische verbinding (een grote ring) van vier verbonden bouwstenen (isoindoolgroepen), waarin een koperion ( $Cu^{2+}$ ) in het midden zit. Dit koperion is vastgehouden door vier stikstofatomen. De ring heeft een speciaal patroon van afwisselende enkele en dubbele bindingen, waardoor een stabiel elektronensysteem ontstaat.

formula	$C_{32}H_{16}CuN_8$
name	copper(II) phthalocyanine
alternate names	C.I. pigment blue 15   copper phthalocyanine   CuPc   phthalocyanine blue   pigment blue 15

Figuur 13: Chemische namen en formules Copper(II) phthalocyanine



Figuur 12: Structuur Copper(II) phthalocyanine

- **Bindingen en Hybridisatie:**
  - a. Stikstofatomen in de Ring
    - Hybridisatie:  $sp^2$ 
      - Elk stikstofatoom vormt **drie  $\sigma$ -bindingen** (twee met koolstof, één met koper).
      - **Eén vrij elektronenpaar** deelt zich in het geconjugeerde  $\pi$ -systeem.
    - Geometrie: **Trigonaal vlak**.
  - b. Koperion ( $Cu^{2+}$ )
    - Coördinatiebinding:  $Cu^{2+}$  wordt omringd door vier stikstofatomen in een **vierkant vlakke geometrie**.
    - Elektronenconfiguratie:  $d^9$  (gebruikt d-orbitalen voor bindingen).
  - c. Koolstofatomen
    - Hybridisatie:  $sp^2$  in de geconjugeerde ring.
    - Geometrie: Trigonaal vlak, met  **$120^\circ$  bindingshoeken**.



Figuur 14: Blauw CuPc Pigment

- **Waarom CuPc Perfect is voor Whiteboard-Inkten:**

- 1. Intense Kleur:**

- Zorgt voor een heldere inkt zonder veel pigment nodig te hebben.
- Absorbeert licht en geeft een sterke blauwe kleur

- 2. Makkelijk uit te wissen:**

- Hecht niet chemisch aan het whiteboard, alleen oppervlakkig.
- Lost niet op in water of ethanol, dus laat geen vlekken achter.

- 3. Duurzaam en stabiel:**

- Veilig in gebruik (niet giftig zoals andere pigmenten).
- Blijft goed onder invloed van zonlicht, zuurstof en hitte.

- 4. Goede mengbaarheid:**

- Werkt goed samen met polymeren (zoals PVA) en oplosmiddelen (zoals ethanol en isopropanol).

### 2.3.1.3 Polymeren:

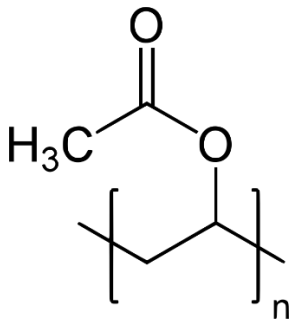
- 1) **Polyvinylacetaat (PVA)**

- **Functie en Toepassing:**

Polyvinylacetaat zorgt voor de **Filmvorming en Hechting** wordt gebruikt om een samenhangende, dunne film te vormen op het whiteboardoppervlak. Deze film zorgt ervoor dat de kleurstof zich goed hecht, maar niet permanent is, zodat de inkt eenvoudig verwijderd kan worden.

- **Chemische Structuur:**  $[\text{CH}_2\text{CH}(\text{O}_2\text{CCH}_3)]_n$

Het polymeer bestaat uit **herhaalde eenheden** van vinylacetaat met de structuur:



Figuur 15: Structuurformule van polyvinylacetaat

Hierbij staat  $\text{OCOCH}_3$  voor de acetaatgroep (acetylgroep verbonden met een zuurstofatoom).

- **Bindingen:**

De individuele eenheden zijn met elkaar verbonden door sterke covalente bindingen ( $\sigma$ -bindingen). En de De acetaatgroep is covalent gebonden aan het koolstofatoom van de herhalende eenheid.

- **Hybridisatie:**

Alle koolstofatomen zijn  $\text{sp}^3$ -gehybridiseerd, wat resulteert in een tetraëdrische geometrie. Die  $\text{sp}^3$ -hybridisatie zorgt voor flexibiliteit van de polymeerketen en dat is goed en belangrijk voor een goede filmvorming en een evenwichtige hechting op het whiteboardoppervlak.

#### 2.3.1.4 Waxen:

##### 1) Carnaubawas (E903)

- **Functie:**  
Voorkomt uitdroging van de markerpunt door een beschermende film te vormen. Deze wordt weer afgebroken als de whiteboardstift in contact komt met het bord. En het verbetert de glijbaarheid van de inkt op het whiteboard.
- Carnaubawas, ook wel bekend als Braziliaanse was, is een natuurlijke was afkomstig van de bladeren van de carnaubapalm (*Copernicia prunifera*), die voornamelijk in Brazilië groeit. Het heeft unieke eigenschappen die het zeer geschikt maken voor gebruik in whiteboard inkten.
- Moleculaire Samenstelling:  
Carnaubawas bestaat voornamelijk uit:
  - Vetzuuresters (80-85%)
  - vetalcoholen (10-15%)
  - Zuren (3-6%)
  - Koolwaterstoffen (1-3%)

De Werking en toepassing **molecuul orbitaal theorie op Carnaubawas wordt uitgelegd in 2.3.2.**

#### 2.3.1.5 Eigenschap van whiteboards:

het is gemakkelijk om op een whiteboard te tekenen door de vlekbestendigheid. Whiteboards worden gecoat met melaminehars. Dit is een mix van melamine ( $C_3H_6N_6$ ), en formaldehyde ( $CH_2O$ ). Als thermohardende kunststof kan melaminehars gemakkelijk op een plank worden gevormd. De kwaliteit van het whiteboard hangt af van de hoeveelheid hars die wordt aangebracht: meer maakt het bord beter bestand tegen vlekken, waardoor de levensduur wordt verlengd.

## 2.3.2 Werking van whiteboard inkten op moleculair niveau

De werking van whiteboardinkt (of whiteboardmarkers) kan op moleculair niveau verklaard worden met behulp van **moleculaire geometrie, molecuulorbitalen** en **polariteit**. Hieronder leg ik uit hoe deze bijdragen aan de werking van whiteboardinkt, en specifiek waarom de inkt goed schrijft op een whiteboard, maar ook weer gemakkelijk kan worden weggeveegd.

Allereerst een theoretische uitleg over de Moleculaire geometrie en polariteit van molecule, met daarna de toepassing van **Polariteit en hechting tussen inkt en het whiteboard oppervlak**.

### 2.3.2.1 Moleculaire geometrie

De Lewistheorie die eerder werd gebruikt om de verschillende componenten van whiteboard inkt voor te stellen, vertoont een aantal tekortkomingen. Een belangrijk aspect daarbij is dat ze de ruimtelijke vorm van het bestanddeel niet verklaart.

De geometrie van een molecule kan voorspeld worden uit de zogenaamde **Valentie-Schaal-Elektronen-Paar-Repulsie theorie (VSEPR)**.

Hierbij beschouw je de bindingselektronen en de vrije elektronenparen als negatief geladen wolken die elkaar afstoten. Ze blijven zo ver mogelijk van elkaar verwijderd en bepalen zo de ruimtelijke structuur en de bindingshoeken van de molecule.

De verschillende stappen in de VSEPR theorie zijn als volgt:

- 1) Beschouw een molecule (of ion) als opgebouwd uit een centraal atoom omgeven door andere atomen en schets hiervoor de Lewisstructuur (zoals aangegeven in het vorige deel)
- 2) dan zullen de elektronenparen rond het centraal atoom zo ver mogelijk uit mekaar zitten (gelijke ladingen stoten mekaar af). De vorm van de molecule (of ion) is daarvan een gevolg.
- 3) Hierbij moeten alle elektronenparen rond het centraal atoom in rekening gebracht worden: zowel bindende als niet-bindende paren.
- 4) De niet-bindende paren bepalen weliswaar mede de ruimtelijke plaats van de atomen in de molecule, doch de vorm van de molecule (ion) wordt uiteindelijk uitsluitend beschreven aan de hand van de aldus bekomen plaats van de atomen. Dit geeft aan dat niet-bindende e-paren meer plaats innemen dan bindende en dubbele bindingen. Hieronder een opsomming van de ruimtelijke rangschikking van de elektronenparen:
  - paren: lineair
  - paren naar de toppunten van een driehoek
  - paren naar de toppunten van een tetraeder (bv. De methylgroep in oplosmiddel Isopropanol)
  - paren: naar de toppunten van een trigonale bipiramide
  - paren naar de toppunten van een octaeder
- 5) Verder zegt de theorie dat:

"aangezien een niet-bindend paar (E) meer ruimte inneemt dan een bindend paar (B) de repulsie tussen elektronenparen toeneemt in de volgorde:  $BB < BE < EE$ ".

### 2.3.2.2 De molecuulorbitaaltheorie (MO theorie)

De molecuulorbitaaltheorie beschrijft de vorming van een atoombinding als de overlap van twee atoomorbitalen. Een normale atoomverbinding ontstaat uit de overlapping van twee half bezette atoomorbitalen. Hierbij ontstaat een bindend molecule orbitaal.

Een molecule is een verzameling van atoomkernen en elektronen, en deze laatste, bevinden zich in molecuulorbitalen.

Een oplossing van dit systeem is mogelijk via benaderingsmethoden.

Eén van die benaderingen is deze waarbij elk MO mathematisch voorgesteld wordt als een **Lineaire Combinatie van Atoom Orbitalen: de LCAO methode.**

Elk AO kan wiskundig gezien worden als een golffunctie. Moleculaire orbitaalberekeningen tonen aan dat de interactie tussen twee atoomorbitalen, leidt tot vorming van twee molecuulorbitalen. Eén heeft een lagere energie dan de AO's waaruit ze gevormd is. De elektronendensiteit is het hoogst tussen de kernen en veroorzaakt een bindend effect zodat ze omschreven wordt als **bindend MO.**

Het **anti-bindende MO** heeft een hogere energie dan de corresponderende AO's. De grootste elektronendensiteit valt te situeren in een gebied buiten de kernen waardoor deze niet meer zijn afgeschermd en elkaar zullen afstoten.

AO's overlappen of worden gecombineerd in verschillende richtingen:

- volgens de bindingsas: **sigma** molecuulorbitalen
- loodrecht op de bindingsas: **pi** molecuulorbitalen

De interactie tussen twee atoomorbitalen leidt tot de vorming van één bindend MO en één anti-bindend MO.

Met de MO-theorie kan de verdeling van elektronen in een molecuul worden voorspeld, wat kan leiden tot de voorspelling van moleculaire eigenschappen zoals vorm, magnetisme en bindingsorde.

Voorbeelden:

- p-p overlap: kan leiden tot zowel bindende als anti-bindende  $\pi$ -bindingen.
- d-d overlap: kan leiden tot zowel bindende als anti-bindende  $\delta$ -bindingen.
- s-s overlap: kan leiden tot een bindende  $\sigma$ -binding.
- **$\sigma$ -binding**: is een symmetrische binding langs de as tussen de twee atomen.
- **n-binding**: is een binding die loodrecht op de as tussen de twee atomen staat en vaak voorkomt bij dubbele en drievoudige bindingen.
- $\delta$ -binding: is een binding die gevormd wordt door d-orbitalen die loodrecht op elkaar staan.

#### **Bindingen en dubbele/drievoudige bindingen:**

De bindingshoek is de hoek tussen twee bindingen van eenzelfde atoom. Het geheel van bindingslengten en bindingshoeken bepaalt de ruimtelijke structuur van een molecule.

- Een enkelvoudige binding bestaat uit één  $\sigma$ -binding.
- Een dubbele binding bestaat uit één  $\sigma$ -binding en één  $\pi$ -binding (bv. O<sub>2</sub>).
- Een drievoudige binding bestaat uit één  $\sigma$ -binding en twee  $\pi$ -bindingen (bv. N<sub>2</sub>).



**Samengevat:** atoomorbitalen overlappen om moleculaire orbitalen te vormen, wat kan leiden tot bindende of anti-bindende bindingen, en hoe deze bindingen samenwerken om verschillende soorten dubbele en drievoudige bindingen te vormen.

Een vergelijking tussen een  $\sigma$ -binding en een  $\pi$ -binding levert volgend resultaat:

**1) sigma-binding:**

- axiale overlap gesitueerd tussen de kernen
- grote bindingsenergie
- symmetrische distributie rond de bindingsas
- slechts één binding tussen 2 atomen
- vrije rotatie rond de bindingsas

**2) pi-binding:**

- laterale overlap in een gebied buiten de kernen
- lagere bindingsenergie
- distributie boven en onder nodaal vlak
- één of twee bindingen mogelijk
- verhinderde rotatie

### 2.3.2.3 Polariteit van moleculen:

Een **polaire binding** ontstaat als twee atomen met verschillende EN-waarde een binding aangaan. De bindingselektronen verschuiven gedeeltelijk naar het atoom met de grootste EN-waarde. Hierbij ontstaat een dipool. Het meest elektronegatieve atoom krijgt een partieel negatieve lading en het minst elektronegatieve krijgt een partieel positieve lading. De verschuiving van de bindingselektronen is groter naarmate het verschil in EN-waarde toeneemt.

Elke partieel negatieve lading duid je in een Lewis structuur aan met het symbool  $\delta^-$ , een partieel positieve lading met  $\delta^+$ .

Bij een **apolaire binding** is het verschil in EN-waarde van de bindende atomen nul. De bindingselektronen zijn gelijk verdeeld tussen de twee atomen

De polariteit van een verbinding is niet enkel afhankelijk van het verschil in EN-waarde tussen de gebonden atomen. Een asymmetrische ladingsverschuiving in de molecule is een tweede noodzakelijke voorwaarde.

Om de symmetrie van de ladingsverschuiving van een molecule bij een covalente verbinding op te sporen, kun je met ladingsvectoren werken. Hiervoor teken je, volgens de bindingsrichting, ladingsvectoren tussen de verschillende bindingspartners. De zin van de ladingsvector is van de partieel positieve naar de partieel negatieve lading. De ladingsvector geeft de verschuiving van de bindingselektronen weer.

Het verschil in EN-waarde dient als maat voor de lengte van elke ladingsvector. De som van alle ladingsvectoren van een molecule is een maat voor de netto ladings- verschuiving. Moleculen waarvan deze som niet nul is, vertonen een asymmetrische ladingsverschuiving en bezitten dus een positieve en een negatieve kant. Een polaire molecule of dipoolmolecule is een molecule met een positieve en een negatieve pool. Een molecule die niet polair is, noem je een apolaire molecule.

Dus we kunnen hieruit besluiten dat:

- **POLAIRE** stoffen lossen goed op in POLAIRE solventen by; water
- **APOLAIRE** stoffen lossen goed op in APOLAIRE solventen

#### 2.3.2.4 Toepassing van Polariteit en hechting tussen inkt en het whiteboard oppervlak

Polariteit is cruciaal in de interactie tussen de inkt en het whiteboardoppervlak. Whiteboards zijn doorgaans gemaakt van een niet-polaire of zwak polaire coating, zoals geëmailleerd staal of kunststoffen zoals melamine of polyesterfilm. De whiteboardinkt is ontworpen om weinig polaire eigenschappen te hebben wanneer deze opdroogt. Dit betekent: zwakke intermoleculaire krachten tussen de inktfilm en het oppervlak en geen sterke waterstofbruggen of dipool-dipoolinteracties met het oppervlak. Hierdoor blijft de opgedroogde inkt oppervlakkig liggen zonder chemisch te binden met het whiteboard, wat het mogelijk maakt om het makkelijk te verwijderen.

#### 2.3.2.5 Molecuulorbitalen en vluchtigheid oplosmiddelen

De molecuulorbitalen helpen ons begrijpen waarom sommige componenten van de inkt snel verdampen en andere blijven liggen: Het oplosmiddel bevat moleculen met relatief hoge energie in hun hoogste bezette molecuulorbitaal (HOMO) en lage energie van hun laagste onbezette molecuulorbitaal (LUMO). Dit betekent: ze zijn vluchtig, omdat de overgang van HOMO naar LUMO weinig energie vereist. Ze verdampen dus snel bij kamertemperatuur. Dit helpt de inkt snel te drogen na het aanbrengen op het whiteboard. De polymeren in de inkt hebben grote molecuulorbitalen die weinig neiging hebben tot interactie met het oppervlak: hun LUMO is te hoog om gemakkelijk elektronen te accepteren van het whiteboardmateriaal en hun HOMO is te laag om elektronen af te staan. Dit draagt bij aan de lage hechting van de opgedroogde film.

#### 2.3.2.6 Droging en afveegbaarheid:

Door de combinatie van een vluchtig oplosmiddel (dat snel verdampt), niet-polaire of zwak polaire polymeren (die niet hechten), en de elektronische structuur van de moleculen (weinig neiging tot vorming van sterke bindingen), vormt de inkt een droge, broze film die mechanisch loskomt bij het wisselen, en daardoor gemakkelijk te verwijderen is.

Samengevat:

Tabel 3: Droging en afveegbaarheid van whiteboard-inkt

Eigenschap	Moleculaire verklaring
Slechte hechting aan whiteboard	Zwakke polariteit, weinig intermoleculaire interacties
Snelle droging	Vluchtige oplosmiddelen met geschikte molecuulorbitalen
Makkelijk te wisselen	Polymeren met lage hechting (door hun structuur en molecuulorbitalen)
Geen permanente binding	Geen chemische interactie tussen HOMO/LUMO van inkt en whiteboardmateriaal

#### Werking en toepassing molecuul orbitaal theorie op Carnaubawas

Carnaubawas bestaat voornamelijk uit lange ketens van vetzuren en alcoholen. Deze moleculen hebben zowel polaire als apolaire delen, wat hun unieke eigenschappen verklaart.

### Hydrofobiciteit:

De apolaire delen van de vetzuurketens in carnaubawas dragen bij aan de hydrofobiciteit. Volgens de MO-theorie ontstaan er bindende moleculaire orbitalen tussen de koolstof- en waterstofatomen in deze ketens, wat resulteert in een stabiele, apolaire structuur.

### Glans en Bescherming:

De polaire delen van de vetzuren en alcoholen kunnen interacties aangaan met andere moleculen in de inkt. De bindende moleculaire orbitalen tussen deze polaire groepen en andere componenten van de inkt zorgen voor een gladde, glanzende afwerking.

### Stabiliteit en Duurzaamheid:

De lange ketens van carnaubawas vormen sterke, stabiele bindingen door de overlap van p-orbitalen, wat resulteert in een hoge bindingsorde. Dit betekent dat er veel bindende moleculaire orbitalen zijn, wat de stabiliteit en duurzaamheid van de inkt verhoogt.

#### 2.3.2.7 Molecuulstructuren en VSEPR in oplosmiddelen van whiteboardinkt

Hierbij worden ook nog de molecuulstructuren en VSEPR van de oplosmiddelen meegegeven.

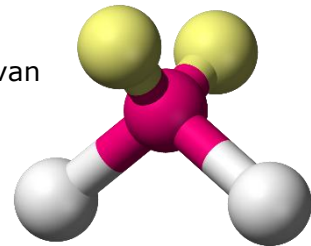
#### Ethanol & Isopropanol

Beide alcoholen bevatten een **-OH groep** die **polair** is.

VSEPR model voorspelt een **gebogen** structuur rond de zuurstof met bindingshoek van ongeveer  $104.5^\circ$ , vanwege 2 bindingen + 2 vrije elektronenparen

→  $AX_2E_2$  (zoals water).

Hun vluchtigheid komt doordat er relatief zwakke intermoleculaire krachten zijn (waterstofbruggen onderling, maar geen sterke binding aan oppervlak).



Figuur 16: Bol staaf model  $AX_2E_2$

Orbitaalinteracties (HOMO-LUMO):

- **Oplosmiddelen (ethanol, isopropanol):**  
HOMO bevat elektronen van het niet-bindende paar op zuurstof.  
Lage energie nodig voor excitatie → **vluchtige stoffen** (makkelijk verdampen).
- **Polymeren (zoals PVP):**  
HOMO zit op het zuurstof en stikstof; LUMO te hoog om effectieve donor-acceptorinteracties aan te gaan met het whiteboardoppervlak.  
Hierdoor blijven er **geen sterke donor-acceptorbindingen** achter op het oppervlak → makkelijk te wissen.

VSEPR-Uitleg Per Molecuul

#### 1) Ethanol ( $CH_3CH_2OH$ )

**Zuurstof:**  $AX_2E_2$  → **gebogen** structuur rond O (zoals in water), bindingshoek  $\sim 104.5^\circ$ .

**C-atomen:** tetraëdrisch ( $AX_4$ ) rond elk C, want elk C vormt 4  $\sigma$ -bindingen.

#### 2) Isopropanol ( $CH_3CHOHCH_3$ )

Zuurstof: idem, **gebogen**  $AX_2E_2$ .

Middelste C-atoom: verbonden met twee  $CH_3$ 's en een OH →  **$AX_4$  (tetraëdrisch)**

## 2.4 Regressieanalyse

### 2.4.1 Inleiding

Een belangrijk onderdeel van dit project is het goed laten werken van de motoren. Motoren worden warm als ze lang aanstaan en te veel warmte kan ervoor zorgen dat ze minder goed werken of zelfs stuk gaan.

Daarom is het belangrijk om te weten hoe de temperatuur van de motor verandert tijdens het gebruik. Deze analyse helpt te begrijpen hoe warm de motor wordt en of we maatregelen moeten nemen, zoals het toevoegen van koeling of het inlassen van pauzes. De regressieanalyse is dus een belangrijk deel van het project om ervoor te zorgen dat de whiteboard-veger goed en lang blijft werken.

Het hoofddoel van deze analyse is om precies te meten hoe de temperatuur van de motor oploopt terwijl hij werkt. Dit is nodig om te controleren of de motor binnen veilige temperaturen blijft. Zo voorkomen we problemen zoals minder goede prestaties, snelle slijtage of dat de motor helemaal stopt met werken.

### 2.4.2 Onderzoek en Analyse

#### 2.4.2.1 Onderzoeksvraag

De centrale vraag die deze analyse stuurt, is: "Hoe verandert de temperatuur van de motor (°C) naarmate hij langer in werking is?"

#### 2.4.2.2 Variabelen

De variabelen die in dit onderzoek zijn gedefinieerd, zijn:

- **Afhankelijke variabele (Y):** De temperatuur van de motor in graden Celsius (°C).
- **Onafhankelijke variabele (X):** De tijd dat de motor aanstaat (in minuten).

#### 2.4.2.3 Doel van de test

Ik wil uitzoeken of de motor na een tijdje te warm wordt. Als dat zo is, kan ik maatregelen nemen, zoals een koelsysteem toevoegen of pauzes tijdens het gebruik van de bordveger.

#### 2.4.2.4 Hoe ga ik dit testen?

##### 1. Experiment opzetten:

- Ik laat de motor draaien en meet de temperatuur elke minuut. De test gebeurt steeds onder dezelfde omstandigheden, zodat de resultaten betrouwbaar zijn.

##### 2. Metingen uitvoeren:

- Ik gebruik een infrarood-thermometer op een afstand van 2cm van de motor om de warmte van de motor te meten. Alle metingen noteer ik in een tabel.

##### 3. Gegevens analyseren:

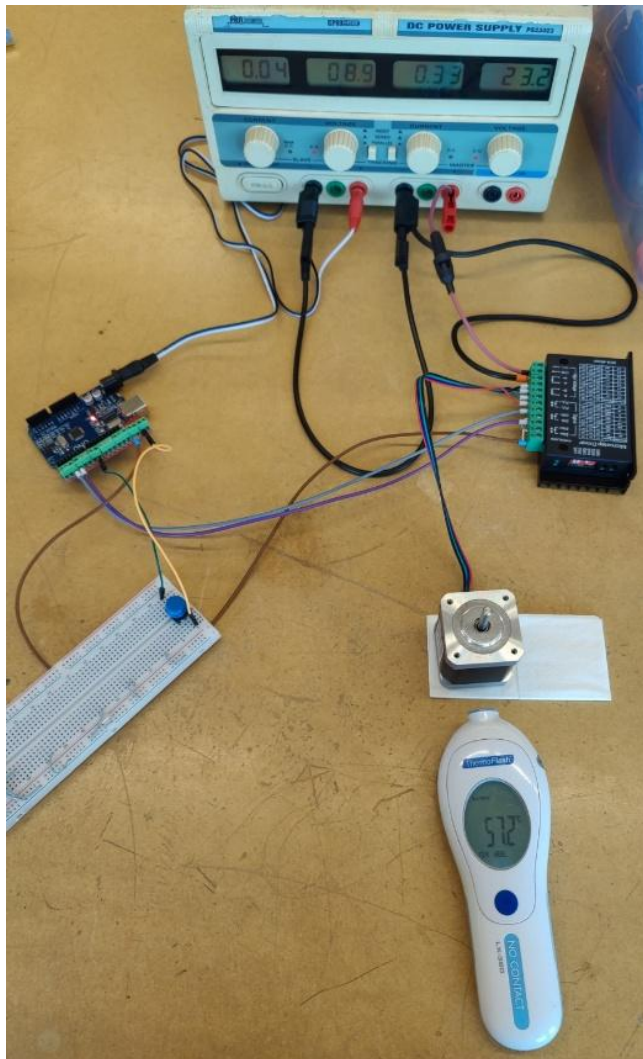
- Ik maak een grafiek in Excel waarin ik de tijd uitzet tegen de temperatuur. Met een trendlijn kijk ik of de temperatuur steeds blijft stijgen of op een bepaald punt stopt met stijgen. Ik zoek uit bij welke temperatuur de motor mogelijk problemen krijgt.

#### 2.4.2.5 Wat verwacht ik?

Waarschijnlijk zal de temperatuur starten bij kamertemperatuur (18-21 graden Celsius) en dan eerst snel stijgen en daarna langzamer toenemen of op een vast punt blijven. Dit zal dus waarschijnlijk een logistische Regressie worden. Als de motor te warm wordt, zal ik kijken of ik koeling moet toevoegen of de code moet aanpassen zodat de motor pauzes neemt.

#### 2.4.2.6 Uitwerking

- **Benodigdheden:**
  - Arduino (met testcode)
  - TB6600 motor driver
  - Stappen motor
  - infrarood-thermometer
  - chronometer of stopwatch
- **stappenplan:**
  1. bouw de test-opstelling
  2. positioneer de thermometer op 2cm van de motor en noteer de starttemperatuur
  3. start de motor en de chronometer
  4. meet en noteer om de minuut.



Figuur 17: Test opstelling Regressieanalyse

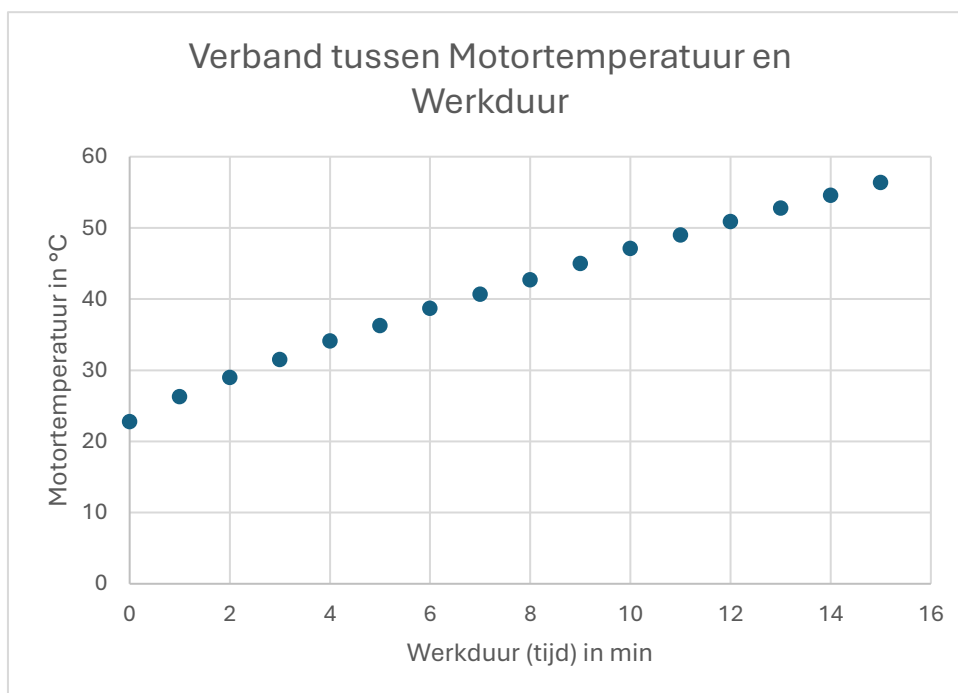
### 2.4.3 Meetresultaten

Toen de motor 15 minuten draaide, begon de temperatuur op 22,8 °C (kamertemperatuur) en steeg tot 56,4 °C. De temperatuur steeg in het begin vrij snel en nam daarna nog steeds toe, maar wel iets minder snel dan in de eerste minuten. Deze waarneming past bij de startfase en de daaropvolgende vertraging van een S-vormige groeicurve, waarbij de snelheid van de stijging afneemt naarmate het systeem een maximum nadert.

Tabel 4: Meetresultaten regressieanalyse

Motortemperatuur in °C	Werkduur (tijd) in min
22,8	0
26,3	1
29	2
31,5	3
34,1	4
36,3	5
38,7	6
40,7	7
42,7	8
45	9
47,1	10
49	11
50,9	12
52,8	13
54,6	14
56,4	15

Deze meetdata kan vervolgens grafisch worden voorgesteld in een Spreidingsdiagram:



Grafiek 1: Spreidingsdiagram meetresultaten

### 2.4.3.1 Meetafwijkingen

Tijdens het experiment zijn er enkele meetafwijkingen opgetreden die de nauwkeurigheid van de resultaten kunnen beïnvloeden. Omdat ik niet precies om de minuut kon meten, reken ik een maximale afwijking van twee seconden per meting in. Dit komt door het handmatig bedienen van de chronometer. Daarnaast kon de voedingsspanning van 24V en 9V niet met absolute precisie worden ingesteld, wat ook een bron van variatie is.

Deze meetonzekerheden kunnen ruis in de verzamelde gegevens veroorzaken. Hoewel deze afwijkingen op het eerste gezicht klein lijken, kunnen ze de relatie tussen motortemperatuur en tijd verstoren. Ze kunnen er ook voor zorgen dat een niet-lineaire trend, zoals de verwachte S-vormige curve, lineairder lijkt, vooral over een beperkt databereik. Lichte schommelingen in de spanning kunnen bijvoorbeeld temperatuurverschillen veroorzaken die niet alleen afhankelijk zijn van de tijd, wat de precisie van het regressiemodel beïnvloedt. Het is daarom belangrijk om experimentele fouten zoveel mogelijk te beperken voor een betrouwbare data-analyse.

## 2.4.4 Berekeningen

### 2.4.4.1 Het Gemiddelde

Het gemiddelde geeft een centrale waarde van de meetresultaten. Voor zowel de onafhankelijke variabele (tijd, X) als de afhankelijke variabele (temperatuur, Y) kan het gemiddelde worden berekend door alle waarden op te tellen en te delen door het aantal metingen (16 metingen).

- Het gemiddelde van de X-waarden (tijd) is:

$$\bar{x} = \frac{1}{16} \sum_{i=0}^{15} x_i = \frac{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15}{16} = 7,5 \text{ minuten}$$

- Het gemiddelde van de Y-waarden (temperatuur) is:

$$\bar{y} = \frac{1}{16} \sum_{i=0}^{15} y_i = \frac{22,8+26,3+29,0+31,5+34,1+36,3+38,7+40,7+42,7+45,0+47,1+49,0+50,9+52,8+54,6+56,4}{16} = 41,11875 \text{ }^{\circ}\text{C}$$

Het gemiddelde is een belangrijke basis voor het berekenen van andere regressieparameters, zoals de richtingscoëfficiënt van de regressielijn.

### 2.4.4.2 De Regressielijn

De regressielijn is een rechte lijn die de relatie tussen de gemeten punten zo goed mogelijk weergeeft. De vergelijking van deze lijn is:

$$y = ax + b$$

Waarbij 'a' de richtingscoëfficiënt is en 'b' het snijpunt met de Y-as.

- De richtingscoëfficiënt 'a' kan worden berekend met de formule:

$$a = \frac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n x_i^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

- En het snijpunt 'b' met de formule:

$$b = \bar{y} - a\bar{x}$$

Op basis hiervan kan ik de richtingscoëfficiënt en het snijpunt met de Y-as (b) berekenen en dan verder ook de vergelijking van de regressielijn:

1. Voor de richtingscoëfficiënt 'a' vul ik de formule in en kom ik uit op:

$$a = \frac{743,75}{340} = 2,1875$$

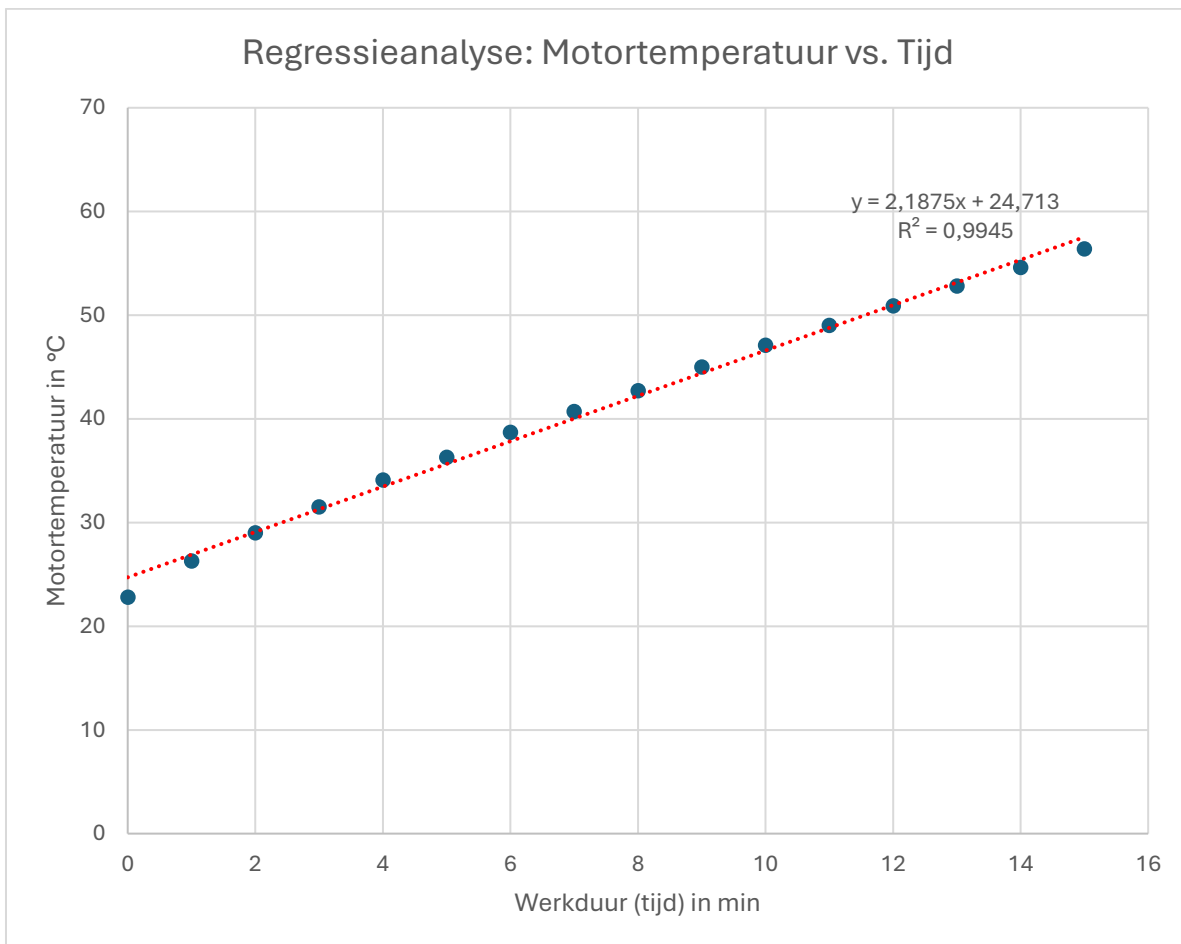
2. Ik doe hetzelfde voor het snijpunt met de Y-as (b), dat geeft:

$$b = \bar{y} - a\bar{x} = 41,11875 - (2,1875 \cdot 7,5) = 41,11875 - 16,40625 = 24,7125$$

Met deze twee waarden kan ik nu de vergelijking van de regressielijn opstellen:

$$Y = 2,1941X + 24,725$$

Deze lijn kan vervolgens worden toegevoegd aan een spreidingsdiagram. Dat geeft:



Grafiek 2: Spreidingsdiagram met regressielijn

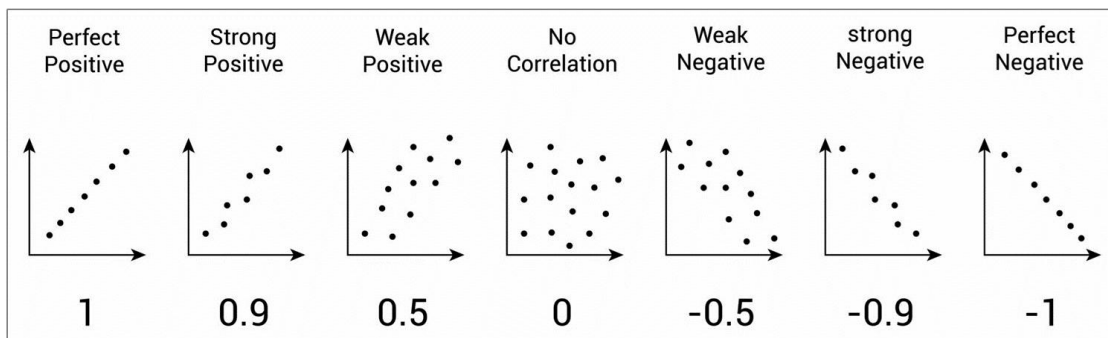
Als we deze waarden ingeven in Exel en het door de computer laten berekenen komen we op een gelijke waarden uit. Dit laat dus zien dat de berekeningen correct zijn. De kleine verschillen in de uitkomsten zijn af te leiden uit kleine afrondingsfouten tijdens het handmatig berekenen. De berekeningen in Exel zijn dus het meest correct.



### 2.4.4.3 De Correlatiecoëfficiënt

De correlatiecoëfficiënt ( $r$ ) is een getal tussen -1 en 1 dat aangeeft hoe sterk en in welke richting twee variabelen met elkaar samenhangen.

- Een waarde dicht bij +1 betekent een sterke positieve relatie (als X toeneemt, neemt Y ook toe).
- Een waarde dicht bij -1 betekent een sterke negatieve relatie (als X toeneemt, neemt Y af).
- Een waarde dicht bij 0 betekent een zwak of geen lineair verband.



Figuur 18: verschillende correlatiecoëfficiënten waarden

De determinatiecoëfficiënt ( $R^2$ ) is in enkelvoudige lineaire regressie gelijk aan het kwadraat van de correlatiecoëfficiënt ( $r$ ). Dit getal geeft aan welk deel van de veranderingen in de afhankelijke variabele verklaard kan worden door de onafhankelijke variabele via het model. De correlatiecoëfficiënt ( $r$ ) meet dus de sterkte en richting van de lineaire relatie en wordt berekend door:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

Nu kan ik deze formule invullen met mijn data en dat geeft:

$$r = \frac{\sum_{i=1}^{16} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{16} (x_i - \bar{x})^2 \sum_{i=1}^{16} (y_i - \bar{y})^2}} = \frac{743.75}{\sqrt{340 \times 1635.904375}} = 0,9972$$

Ook deze waarde kan ik in Exel narekenen, door de determinatiecoëfficiënt ( $R^2$ ) te laten berekenen en hiervan de vierkantswortel te nemen om de correlatiecoëfficiënt te krijgen:

- Determinatiecoëfficiënt (door Exel berekend):

$$R^2 = 0,9945$$

- Hieruit kan ik dan de correlatiecoëfficiënt berekenen:

$$r = \sqrt{R^2} = \sqrt{0,9945} = 0,9972$$

Na het berekenen met Exel blijkt de correlatiecoëfficiënt exact dezelfde waarde op te leveren. Dit bevestigt dat de berekening correct is.

Deze waarde van 0,9972 ligt dan ook heel dicht bij 1, wat duidt op een zeer sterke positieve lineaire relatie tussen de gemeten tijd en de motortemperatuur. Dit betekent dat naarmate de motor langer aanstaat, de temperatuur sterk en consistent toeneemt.

## 2.4.5 Interpretatie Berekende Regressie

Zoals vermeld, werd een lineair verband berekend, ondanks de verwachting van een logistisch verband. Dit betekent dat een lineair regressiemodel op de verzamelde temperatuur-tijd data is toegepast.

De berekende correlatiecoëfficiënt ( $r$ ) van 0,9972 en de determinatiecoëfficiënt ( $R^2$ ) van 0,9945 zijn zeer hoog. Dit toont dat de lineaire regressielijn de gemeten datapunten extreem goed volgt binnen het bereik van de metingen (0 tot 15 minuten). Een  $R^2$  van bijna 1 (0,9945) geeft aan dat bijna alle variatie in de motortemperatuur verklaard kan worden door de verstreken tijd, volgens dit lineaire model.

Echter, het is belangrijk om te begrijpen waarom dit lineaire model, ondanks de uitstekende pasvorm voor de gemeten data, toch 'anders' is dan wat theoretisch werd verwacht. Zoals eerder besproken, verwachtte ik dat de temperatuur van de motor een logistische curve zou volgen. Dit komt omdat de temperatuur eerst snel stijgt, maar dan afvlakt naarmate de motor een stabiele, maximale temperatuur bereikt.

### **Verklaring: Waarom neemt de temperatuur in de eerste 15 minuten lineair toe?**

In de eerste 15 minuten van de meting neemt de temperatuur van de motor vrijwel lineair toe. Dit fenomeen is te verklaren door de thermische eigenschappen van elektromotoren en het gedrag van warmteontwikkeling en -afvoer in het beginstadium van werking.

#### **1. Warmteproductie is aanvankelijk constant bij gelijkblijvende belasting**

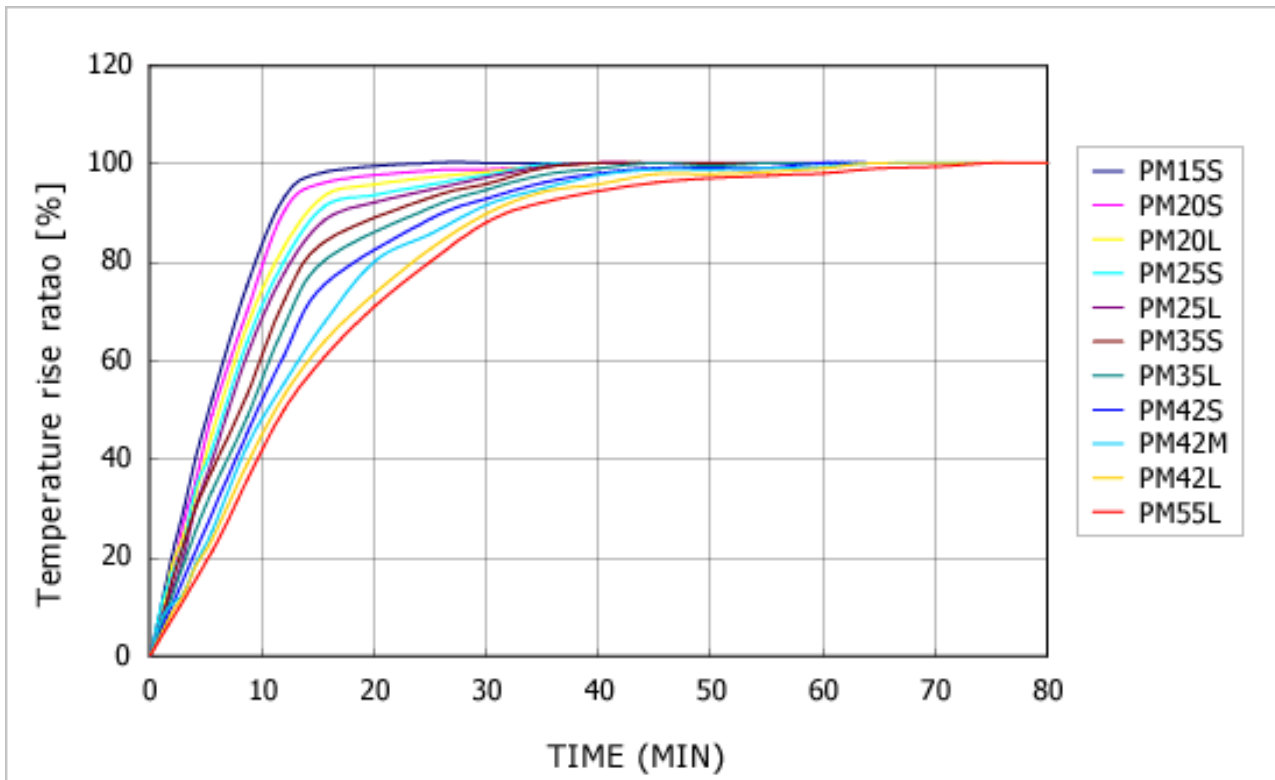
De warmte in een elektromotor ontstaat voornamelijk door elektrische verliezen ( $I^2R$ -verliezen in de wikkelingen), wrijving in de lagers en magnetische verliezen. Als de belasting en het toerental van de motor gedurende deze tijd constant blijven (zoals het geval is in de testomgeving), dan is ook de hoeveelheid geproduceerde warmte relatief constant. Dit resulteert in een constante vermogensinvoer van warmte in het systeem, wat leidt tot een lineaire temperatuurstijging zolang er nog geen evenwicht is tussen warmteproductie en -afvoer.

#### **2. Thermisch traag systeem: warmte stapelt zich op**

In het begin heeft de motor nog niet voldoende tijd gehad om warmte af te voeren aan de omgeving. De convectie (warmteafgifte aan de lucht) verloopt relatief traag en de geleiding van de warmte naar andere delen van het systeem kan in de testopstelling niet gebeuren. Dit komt omdat het normaal even duurt voordat de temperatuur van de buitenkant van de motor stijgt, aangezien warmte zich eerst moet verspreiden vanuit de wikkelingen naar de behuizing. Dit vertraagt de afkoeling en versterkt de lineaire opwarming in de beginfase.

#### **3. Pas later neemt de afkoeling relatief toe → afvlakking temperatuurcurve**

Naarmate de temperatuur stijgt, wordt het temperatuurverschil met de omgeving groter, en daardoor wordt de warmteafvoer via convectie efficiënter. Op dat moment begint de temperatuurstijging af te vlakken. In die latere fase zie je dus een duidelijk niet-lineair gedrag. Maar in de eerste 15 minuten is dat effect nog beperkt, waardoor de opwarming vrijwel rechtlijnig verloopt.



Grafiek 3: Opwarmingstijd stappenmotor

### Samengevat:

De lineaire temperatuurstijging in de eerste 15 minuten wordt veroorzaakt door:

- een constante warmteproductie bij constante belasting
- beperkte initiële warmteafvoer door thermische traagheid
- het ontbreken van thermisch evenwicht in de beginfase.

Wanneer een motor langere tijd aanstaat, bijvoorbeeld een stappenmotor of een gelijkstroommotor, wordt die motor steeds warmer. Dat komt doordat er bij het werken van de motor warmte ontstaat, vooral door elektrische weerstand en wrijving in de onderdelen. Als de motor niet genoeg kan afkoelen, blijft de temperatuur stijgen.

Een motor kan pas goed warmte afvoeren als het temperatuurverschil tussen de motor en de omgeving groot genoeg is. In het begin is dat verschil klein, dus raakt de motor zijn warmte nog niet snel kwijt. Daardoor warmt hij eerst snel op. Na een tijdje komt er een evenwicht, waarbij de motor evenveel warmte afgeeft als hij opwekt. Dan blijft de temperatuur ongeveer gelijk.

Dit gedrag is typisch voor elektromotoren zoals stappenmotoren of gelijkstroommotoren. Als ze lang aanstaan of zwaar belast worden, worden ze steeds warmer. Dit kan gevolgen hebben voor hun prestaties en levensduur.

## 2.4.6 Gevolgen Motorprestaties

Als een stappenmotor te warm wordt en dit probleem niet op tijd wordt opgelost, kan dit grote schade aanrichten. De belangrijkste gevolgen zijn:

- **Verlies van stappen:** door de hitte kunnen onderdelen binnenin uitzetten. Dit verandert de manier waarop de motor beweegt, waardoor hij soms stappen overslaat. Dit zorgt voor minder nauwkeurige bewegingen.
- **Minder goede prestaties en precisie:** bij toepassingen waar nauwkeurigheid belangrijk is, zoals bij een whiteboard-veger, kan de warmte ervoor zorgen dat de motor niet goed genoeg werkt.
- **Kortere levensduur:** als de motor vaak te heet wordt, slijten de onderdelen sneller. Vooral het isolatiemateriaal in de motor gaat sneller stuk, waardoor de motor minder lang meegaat.
- **Grotere kans op storingen:** te veel warmte kan zorgen voor ernstige problemen, zoals kortsluiting of beschadiging van de bedrading in de motor. Dit kan leiden tot grote schade.
- **Onstabiel systeem:** wanneer de motor slechter gaat werken door de hitte, kan dit ook andere onderdelen van het systeem beïnvloeden. Hierdoor kan het hele whiteboard-vegersysteem minder betrouwbaar worden.

De positieve temperatuurcoëfficiënt van weerstand betekent dat de elektrische weerstand van koperen draden groter wordt als de temperatuur stijgt. Dit zorgt voor een soort vicieuze cirkel: hoe warmer de motor wordt, hoe meer weerstand er is, en daardoor ontstaat er nóg meer warmte (als de stroom hetzelfde blijft). Dit effect laat zien hoe belangrijk het is om de motor goed te koelen. Als dat niet gebeurt, kan de motor te heet worden en slechter gaan werken of zelfs stukgaan.

Het is ook heel belangrijk om het juiste regressiemodel te kiezen om de temperatuur van de motor goed te kunnen inschatten. Dit helpt bij het maken van goede keuzes voor manieren om de motor af te koelen, zoals ventilatie of pauzes. Als je een eenvoudig recht model (lineair) gebruikt terwijl de temperatuur eigenlijk op een andere manier stijgt (bijv. logistisch), dan kun je grote fouten maken. Je kunt bijvoorbeeld denken dat de temperatuur langzaam blijft stijgen, terwijl die in werkelijkheid al bijna het maximum bereikt heeft. Hierdoor kun je te laat ingrijpen of niet genoeg doen om oververhitting te voorkomen.

## 2.4.7 Besluit

Uit dit onderzoek blijkt dat de temperatuur van de motor steeds stijgt naarmate hij langer aanstaat. In de eerste 15 minuten zien we dat de temperatuur bijna rechtlijnig toeneemt. Dit komt doordat de motor in het begin meer warmte opwekt dan hij kan afgeven aan de lucht. Pas later zal de temperatuur minder snel stijgen, maar dat is buiten het bereik van deze meting.

De berekende regressielijn ( $Y = 2,19X + 24,73$ ) past goed bij de meetgegevens. De correlatiecoëfficiënt van 0,9972 toont aan dat er een heel sterke en duidelijke relatie is tussen tijd en temperatuur. Dit wil zeggen dat we goed kunnen voorspellen hoe warm de motor wordt op basis van de tijd dat hij werkt.

Hoewel we verwachtten dat de temperatuur in een logistische curve zal vormen, zien we dat de eerste 15 minuten prima te beschrijven zijn met een rechte lijn. Dit betekent dat de motor in die periode nog niet in thermisch evenwicht is.

Deze analyse toont aan dat het belangrijk is om op te letten voor oververhitting. Als de motor te warm wordt, kunnen er problemen ontstaan, zoals minder goede werking, fouten in bewegingen of schade aan de motor. Daarom is het slim om maatregelen te nemen, zoals het toevoegen van koeling of het inlassen van pauzes.

Door deze regressieanalyse weten we nu beter wat er gebeurt met de temperatuur van de motor tijdens gebruik en kunnen we zorgen voor een veilige en betrouwbare werking van de whiteboard-veger.

Om het probleem van de warmte op te lossen kan ik als uitbreiding eventueel:

- **Het plaatsen van een temperatuursensor**  
Plaats een temperatuursensor dicht bij de motor en lees de temperatuur real-time uit. Stel in dat het besturingssysteem een waarschuwing of noodstop afgeeft zodra de motor bijvoorbeeld 56 °C bereikt, zodat je altijd tijdig ingrijpt.
- **Het monteren van actieve en passieve koeling**  
Monteer een klein ventilatortje dat pas inschakelt bij hogere temperaturen en bevestig een koellichaam van aluminium ribben op de motorbehuizing. Zo zorg je dat warme lucht actief wordt weggeblazen en dat warmte bovendien passief via het koellichaam wordt afgevoerd.
- **Het programmeren van gecontroleerde pauzes**  
Programmeer korte onderbrekingen na bijvoorbeeld elke minuut continu vegen, of automatisch zodra de temperatuur boven 50 °C uitkomt. Tijdens zo'n pauze laat je de motor tot stilstand komen of op een lage rustsnelheid draaien om af te koelen zonder dat de gebruiker hier handmatig op hoeft te letten.
- **Het aanpassen van de stroom via de TB6600-driver**  
Gebruik de instelmogelijkheid van de TB6600 om de stroom (current limit) te verlagen wanneer de motor stilstaat of weinig kracht hoeft te leveren. Schakel in je besturingssoftware tussen volle stroom bij actieve beweging en een lagere stroom in rust, om onnodige warmteontwikkeling te vermijden.
- **Het optimaliseren van materialen en behuizing**  
Vervang de PETG-montageplaat door metalen (bv: aluminium) onderdelen met een hogere warmtegeleiding. Voeg daarnaast ventilatieopeningen of luchtkanalen toe in de kap, zodat warme lucht gemakkelijk kan ontsnappen en nieuwe, koele lucht kan aanstromen.

## 2.4.8 Sjabloon



Sjabloon  
Regressieanalyse Mot

## 3 Technisch deel

### 3.1 Opdracht

De bordveger zal bestaan uit één geheel dat makkelijk bovenaan een Whiteboard kan bevestigd worden. Deze zal aan de beide verticale zijden een geleiding hebben voor het veeg-systeem van de bordveger. Het veeg-systeem zal twee aparte bewegingen kunnen maken: een op- en neergaande beweging over de gehele hoogte van het bord en een heen- en weergaande beweging over de gehele breedte van het bord. Het systeem zal een parkeerpositie hebben voor wanneer de veger niet operatief is. In deze positie zal het systeem de gebruiker niet hinderen en geen plaats van het Whiteboard in beslag nemen.

Het veeg-systeem moet XY-setup operationeel zijn. Dit betekent dat je de positie van de bordveger zelf kan instellen en je kan kiezen door middel van de X en Y coördinaten. Dit zal mogelijk gemaakt worden door stappen motoren en endstops die aangesloten en bestuurd zullen worden via een Arduino UNO en TB6600 Stappen Drivers.

Het gehele systeem zal ook beschikken over een noodstop die, wanneer ingedrukt, de motoren blokkeert zodat het veeg-systeem niet ongecontroleerd naar beneden zakt. Dit zal mogelijk gemaakt worden door een drukknop die het systeem blokkeert. De bordveger zal bestuurd worden door een gebruiksvriendelijke user interface en enkele knoppen voor basisacties uit te voeren.

### 3.2 Functionele eisen

#### **Afmetingen van het systeem:**

- Maximale hoogte: **1,2 m.**
- Maximale lengte: **1,5 m.**
- Maximale breedte: **15 cm.**

#### **Gewicht van het systeem:**

- Gewicht van het frame: **6,5 kg**
- Gewicht elektronica (motoren, drivers, etc.): **3 kg**
- ⇒ Het totale gewicht van het systeem zou dus +- **10 kg** moeten zijn.

#### **Precisie van het systeem:**

- Dankzij de zeer precieze motoren wil ik het systeem in stappen van 1 mm kunnen verschuiven, zodat ook de randen van het bord netjes geveegd kunnen worden.

#### **Snelheid van het systeem:**

- Ik zou graag een maximale snelheid van 500 mm/s halen. Ik denk dat dit goed haalbaar is, aangezien de meeste 3D-printers deze snelheid ook kunnen bereiken.

## 3.3 Onderdelen

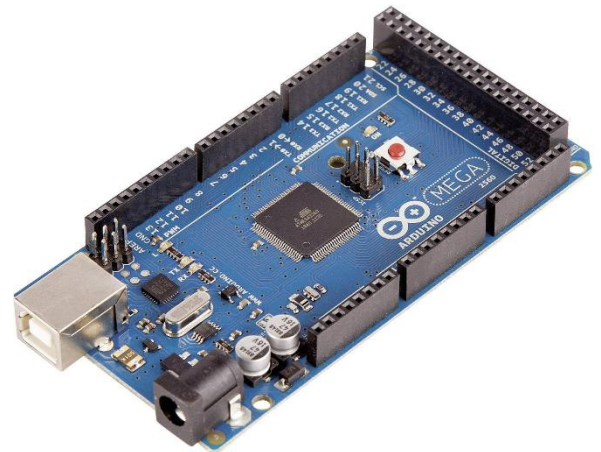
### 3.3.1 Arduino Mega

Het basisonderdeel van mijn GIP is een **Arduino Mega**. Dit is een microcontroller die veel gebruikt wordt in elektronische projecten. De Arduino Mega is populair en gemakkelijk te programmeren door vele online libraries en uitbreidingsmogelijkheden. Het is makkelijk om er basis sensoren en andere elektronische componenten mee aan te sturen, wat belangrijk is voor mijn project

---

[Datasheet Arduino Mega](#)

---



Figuur 19: Arduino Mega

### 3.3.2 TB6600 Stapper Driver

Voor mijn project ga ik de Arduino Mega combineren met **TB6600 Stappen Drivers**. Deze Drivers maken het mogelijk om de Arduino makkelijk te verbinden met de motoren. Door deze combinatie krijg ik een nauwkeurige en goede motorbesturing wat belangrijk is voor het stabiel te kunnen besturen van mijn veger.

#### 3.3.2.1 Werking TB6600:

De TB6600 maakt gebruik van een H-brug Circuitconfiguratie: de **TB6600 stappenmotor driver** is een krachtige module die **2-fase** en **4-fase stappenmotoren** kan aansturen. De module maakt gebruik van een **bipolaire H-brug** configuratie. Deze configuratie gebruikt **MOSFET-transistoren**, die zorgen voor **hoge prestaties, stabiele stroomverwerking**.

De TB6600 driver werkt via twee belangrijke besturingssignalen:

- **Puls (PUL):** elk puls-sigitaal triggert een stap van de motor.
- **Richting (DIR):** bepaalt de draairichting van de motor op basis van de aangelegde spanning.

Ik maak gebruik van een 2-fase bipolaire stappenmotor. Daarom is deze module een perfecte keuze.

#### Uitleg Circuitdiagram in H-brug Configuratie:

Het **H-brug schema** toont **vier MOSFET-transistoren: T1, T2, T3 en T4**. Deze componenten vormen de kern van de driver en regelen de **stroomtoevoer** die nodig is om de stappenmotor aan te drijven.

Daarnaast bevat de schakeling ook voor elke **MOSFET-transistor** een **flyback-diode (D1, D2, D3 en D4)**, deze zijn zeer belangrijk voor **bescherming tegen spanningspieken**, die schade kunnen toebrengen aan de motoren of andere elektronische componenten.

De aansluitingen **A+, A-, B+ en B-** worden aangesloten aan de stappenmotor, waardoor de motor nauwkeurig en gecontroleerd kan bestuurd worden.



Figuur 20: TB6600 Stappen Driver



## Hoe de H-brug de motor laat draaien:

De TB6600 driver schakelt de MOSFET-transistoren in de H-brug afwisselend in- en uit. Waardoor de motor draait.

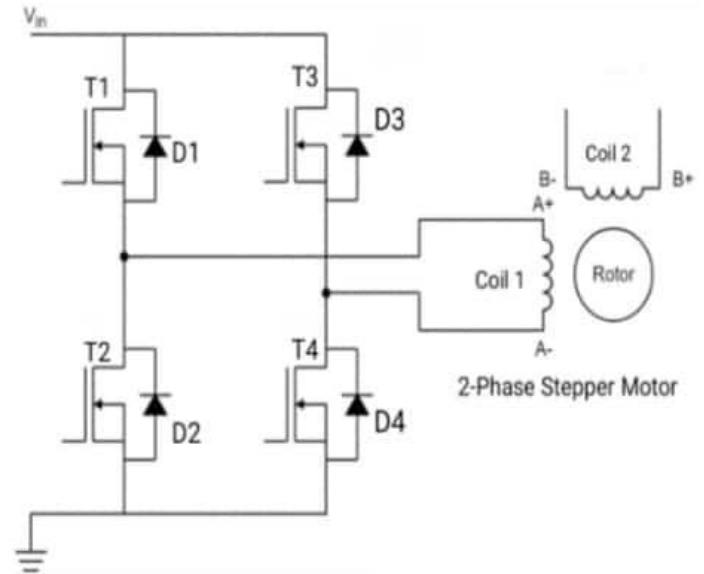
### ➤ **Rechtsom draaien:**

1. **T1 en T4** worden geactiveerd.
2. Stroom loopt van **A+ naar A-**.

### ➤ **Linksom draaien:**

- **T2 en T3** worden ingeschakeld.
- Stroom loopt van **A- naar A+**.

Deze wisselende stroomrichting zorgt voor een goede rotatie van de motor.



Figuur 21: H-brug schema TB6600

## Juiste werkmodus kiezen voor mijn toepassing:

De TB6600 heeft vier verschillende werkmodi, deze hebben tekens een verschillende verhouding in koppel, precisie en stapgrootte.

### Dit zijn de 4 werkmodi:

#### 1. **Golfmodus (Wave Mode):**

- Slechts één spoel wordt tegelijk geactiveerd.
- Hierdoor draait de motor in stappen van 90°.
- Eenvoudige werking, maar minder koppel.

#### 2. **Volstapmodus (Full-Step Mode):**

- Beide spoelen worden tegelijk geactiveerd.
- Hoger koppel en meer stabiliteit.
- Geschikt voor toepassingen die kracht en betrouwbaarheid nodig hebben.

#### 3. **Halfstapmodus (Half-Step Mode):**

- Combineert golfmodus en volstapmodus.
- De motor beweegt in stappen van 45°.
- Biedt een goede balans tussen precisie en koppel.

#### 4. **Microstapmodus (Microstep Mode):**

- Zorgt voor nog kleinere stappen.
- Werkt zeer precies om vloeiende overgangen te creëren.
- Hoogste precisie, ideaal voor toepassingen zoals CNC-machines en robotica.

Voor mijn toepassing heb ik gekozen voor microstep-regeling, omdat deze de beste balans biedt tussen soepelheid, precisie en voldoende koppel. De Z-asmotoren (twee TB6600 parallel) hebben een relatief hoge belasting en moeten sterk genoeg draaien, terwijl de X-asmotor kleiner en gevoeliger is. Daarom stel ik de Z-asmotoren in op full-step (één microstep), zodat ze op hun maximale koppel draaien, en regel ik de stroom in op 2,5 A. De X-asmotor wordt ingesteld op half-step (twee microsteps), omdat op full-step deze motor onvoldoende soepelheid levert. De stroom voor de X-asmotor is 1,5 A, passend bij de lagere belasting en gevoeligheid.



Tabel 5: DIP-switchconfiguratie normaalstroom

As	Motor(en)	Gewenste normaalstroom	SW1	SW2	SW3
Z-as	2 TB6600 parallel met grote motoren	2,5A	OFF	ON	ON
X-as	1 TB6600 met kleinere motor	1,5 A	ON	ON	OFF

Tabel 6: DIP-switchconfiguratie Microsteps

As	Motor(en)	Gewenste Microsteps	SW1	SW2	SW3
Z-as	2 TB6600 parallel met grote motoren	1 (full-step)	ON	ON	ON
X-as	1 TB6600 met kleinere motor	2/A (½-step) met A = 1,5A per fase	OFF	ON	ON

Dit is hoe ik de TB6600 zal instellen voor de stroom en stappenconfiguratie. Hiermee garandeer ik nu de juiste koppelafgifte en een goede balans tussen stabiliteit en precisie. In de toekomst kan ik de microsteps verder aanpassen (bijvoorbeeld overschakelen op 4 of 8 microsteps) om de besturing nog soepeler te maken en de positioneernaauwkeurigheid te verbeteren. De ingestelde stroom blijft echter vast, omdat de motoren en de driver hierop zijn afgestemd en ik dit niet meer kan wijzigen zonder extra hardware aanpassingen.



Figuur 22: DIP-switches TB6600

Microstep Driver									
Micro step	Pulse/rev	S1	S2	S3	Current(A)	PK Current	S4	S5	S6
NC	NC	ON	ON	ON	0.5	0.7	ON	ON	ON
1	200	ON	ON	OFF	1.0	1.2	ON	OFF	ON
2/A	400	ON	OFF	ON	1.5	1.7	ON	ON	OFF
2/B	400	OFF	ON	ON	2.0	2.2	ON	OFF	OFF
4	800	ON	OFF	OFF	2.5	2.7	OFF	ON	ON
8	1600	OFF	ON	OFF	2.8	2.9	OFF	OFF	ON
16	3200	OFF	OFF	ON	3.0	3.2	OFF	ON	OFF
32	6400	OFF	OFF	OFF	3.5	4.0	OFF	OFF	OFF

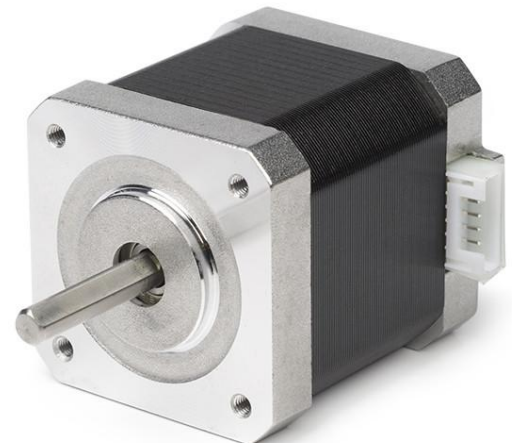
Figuur 23: TB6600 DIP-switch instellingen

### 3.3.3 SL42S247A stappenmotor

Voor mijn GIP maak ik gebruik van de **SL42S247A stappenmotor**. Dit is een Hybride stappenmotor speciaal gemaakt voor projecten waar nauwkeurige en gecontroleerde bewegingen belangrijk zijn. Dankzij de twee spoelen kan de motor in hele kleine en precieze stapjes laten bewegen, wat zorgt voor stabiele en betrouwbare resultaten. Bovendien is de motor erg sterk en kan hij ook grote gewichten aan, waardoor hij geschikt is om het gewicht van de profielen van mijn constructie naar boven te hijsen.

Hierdoor voldoet deze motor aan al mijn eisen en is hij perfect voor mijn project.

Ik gebruik 2 van deze motoren voor de Z-as.



*Figuur 24: SL42S247A stappen motor*

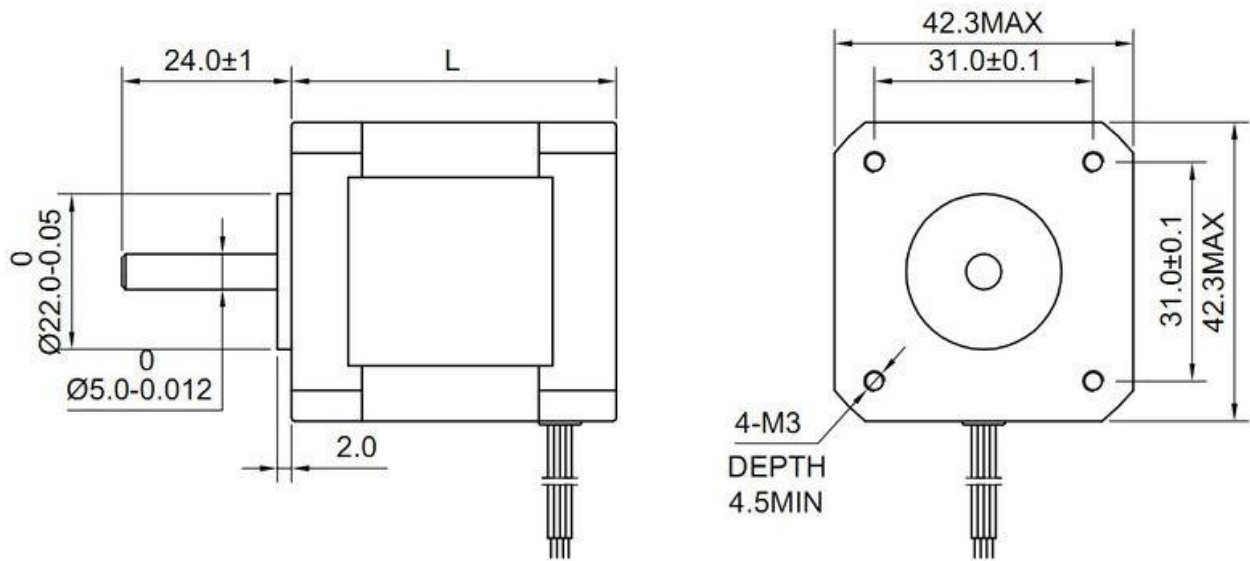
#### 3.3.3.1 Info/ Specificaties motor

- **Houdkoppel (5 kg-cm):** Dit is de maximale kracht die de motor kan uitoefenen wanneer hij stil staat. Met een houdkoppel van 5 kg-cm kan de motor bijvoorbeeld een gewicht van 5 kg op een afstand van 1 cm van de as vasthouden.
- **As diameter (5 mm):** De D-vormige as heeft een diameter van 5 mm. Dit is belangrijk om rekening mee te houden bij het zoeken van koppelingen en mechanische verbindingen zoals bv: tandwielen, riem pulleys...
- **Stap hoek (1,8°):** Bij elke stap roteert de motor 1,8 graden. Dit betekent dat de motor 200 stappen nodig heeft om één volledige omwenteling te maken, wat zorgt voor een goede precisie in positionering.
- **Voltage (2,7-3,3V) en Nominale stroom (1,8-2,5A):** Deze waarden geven aan welke elektrische waarden optimaal zijn voor de werking van de motor. Een spanning van tussen 2,7V tot 3,3V gecombineerd met een stroom van tussen 1,8A en 2,5A zorgt normaal voor de juiste prestaties zonder oververhitting of plotse uitval van de motor.
- **Gewicht (0,36 kg):** Met dit gewicht van 360 gram moet ik rekening houden bij mijn berekeningen.
- **Stappenmotor formaat/ bouwnorm (Nema 17):** De term "Nema 17" verwijst naar een standaardmatenclassificatie en is niet de naam van de motor waarin ik me voordien vergiste en foutief had genoteerd.

De afkorting "Nema" komt van de "National Electrical Manufacturers Association", en het getal "17" duidt op de grootte van de bevestigingsplaat (1,7 inch x 1,7 inch) of (42,3 mm x 42,3) wat dus ook de hoogte en breedte is van deze motor.

Dit betekent dat alle motoren die onder de Nema 17 specificatie vallen qua bevestigingsmaten en -vorm vergelijkbaar zijn, maar de interne eigenschappen zoals koppel, stroomverbruik en andere technische details kunnen variëren per type motor en fabrikant.

Dit geeft dus vooral aan hoe de motor gemonteerd kan worden, maar zegt op zich niets over het interne werkingsprincipe.



Figuur 25: Afmetingen Nema 17 motoren

---

[Datasheet SL42S247A stappen motor](#)

---

### 3.3.4 SL42S234MA107 stappenmotor

Voor de X-as gebruik ik een andere, minder sterke en minder krachtige motor, namelijk de SL42S234MA107. Deze motor is minder zwaar en heeft een lager houdkoppel dan de SL42S247A die ik gebruik voor het hijsen van de profielen. Hierdoor is hij beter geschikt voor lichtere taken zoals de horizontale verplaatsing over de X-as.

Verschillen met de SL42S247A:

- **Houdkoppel:** 2,1 kg-cm tegenover 5 kg-cm → minder krachtig, geschikt voor lichtere belasting.
- **Staphoek:** 0,9° tegenover 1,8° → dubbele resolutie, dus nauwkeuriger positioneren.
- **Stroomverbruik:** 1,2 A tegenover 1,8-2,5 A → minder warmteontwikkeling.
- **Lengte:** 34 mm tegenover 47 mm → compacter en minder zwaar.
- **Toepassing:** geschikt voor lichte bewegingen in de X-richting, niet voor zwaardere hijsbewegingen.



Figuur 26: SL42S234MA107 stappen motor

---

[Datasheet SL42S234MA107](#)

---

#### 3.3.4.1 Werking motor

##### 1. Basisprincipe

- **Wat is een stappenmotor?:**  
Een stappenmotor zet elektrische energie om in nauwkeurige mechanische

bewegingen. In tegenstelling tot standaard DC-motoren, kan een stappenmotor in kleine, discrete stappen draaien en zelfs een positie vasthouden.

- **Magnetisch veld:**

De werking berust op het wisselend activeren van elektromagnetische spoelen. Hierdoor ontstaat een roterend magnetisch veld dat de rotor die een magnetiseerbare kern bevat in kleine stappen laat meebewegen.

## 2. Constructie en Werking:

- **Stator (stilstaand deel):**

De stator bevat meerdere elektromagnetische spoelen die in een specifieke sequentie worden geactiveerd. Wanneer de spoelen worden bekrachtigd, ontstaat er een roterend elektromagnetisch veld.

- **Rotor (bewegend deel):**

De rotor is een ferromagnetische kern met duidelijk gedefinieerde noord- en zuidpolen (door cups met aparte tandjes) die bevestigd is aan de Shaft van de motor, deze reageert op het roterende magnetische veld van de stator en bij elke puls wordt de rotor aangetrokken of afgestoten, waardoor deze één stap draait. Hij "springt" van de ene positie naar de volgende. Dit resulteert in een nauwkeurige stapbeweging.

- **Opbouw:** Acht spoelen, gesplitst in twee groepen van vier.

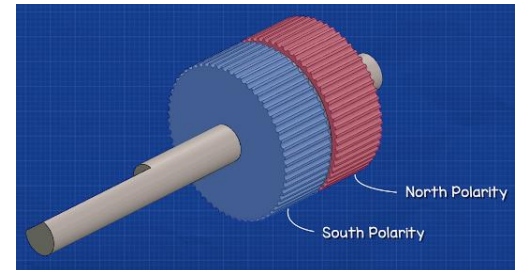
- **Tandverhouding:** De rotor heeft 50 tanden en de stator 48 tanden.

- **Stapgrootte:** Door het verschil in het aantal tanden is elke stap  $1,8^\circ$  ( $360^\circ/200$ ). dit betekent dat de stappenmotor 200 stappen moet zetten per volledige rotatie.

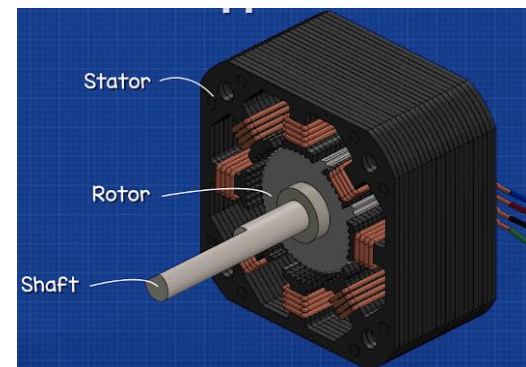
- **Driver en Besturing:**

Mijn motor driver (de Tb6600) stuurt de elektronische schakelaars die de spoelen in de juiste volgorde activeren. Dit zorgt voor een gecontroleerde pulsvolgorde die de rotor in kleine stappen laat bewegen.

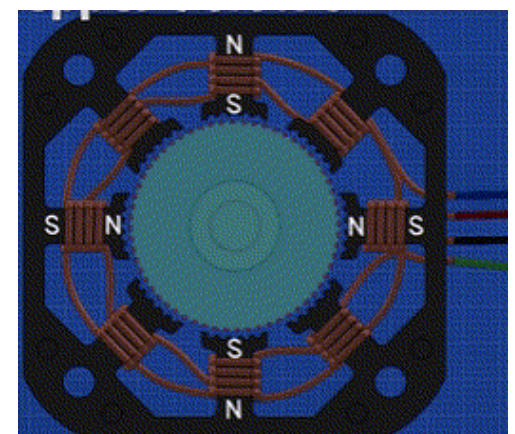
Elke keer als de spoelpolariteit verandert, draaien alleen de tanden die het dichtst bij de actieve (noord polige) spoelen liggen mee. Dit voorkomt dat alle tanden gelijktijdig uitlijnen, wat zorgt voor een zeer nauwkeurige en stabiele positiecontrole.



*Figuur 27: Rotor en Shaft Hybride stappenmotor*



*Figuur 28: Binnenkant Hybride stappenmotor*



*Figuur 29: verandering spoelpolariteit in stappen motor*

### 3.3.5 LM2596 DC-DC Step-Down Converter

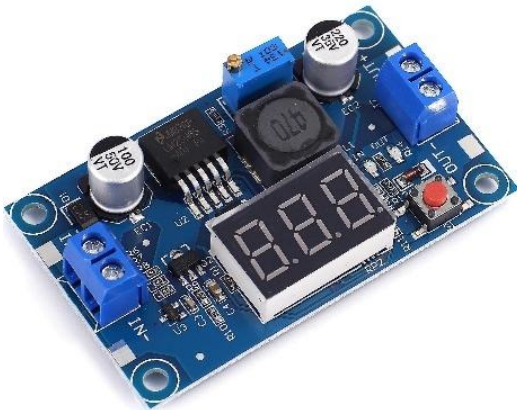
Voor mijn Arduino Mega te voeden gebruik ik een instelbare LM2596-module om de 12 V van mijn voedingsbron om te zetten naar 9 V voor de Vin van de Arduino Mega. De ingebouwde volt- en ampèremeter-display maakt het afregelen van de module makkelijk en laat meteen zien hoeveel de stroom de Arduino gebruikt.

Aansluiting:

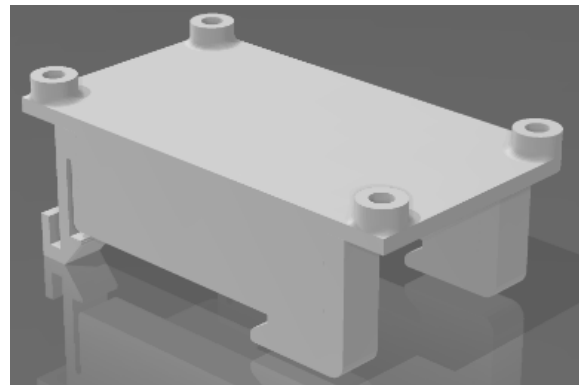
- Ingang: (VIN+, VIN-): 12V DC van de powersupply
- Uitgang: (VOUT+, VOUT-): ingesteld 9V DC naar de VIN-pin van de Arduino Mega

Montage:

De LM2596-converter zet ik vast in de kast met een houder die ik zelf heb ontworpen en 3D-geprint. Die houder klikt eenvoudig op een standaard DIN-rail. Zo blijft de module stevig zitten en blijft alles netjes op zijn plaats.



Figuur 31: LM2596 module



Figuur 30: LM2596 module houder

### 3.3.6 Knoppen

Voor de besturing van mijn whiteboard veger maak ik gebruik van verschillende knoppen en schakelaars om op een makkelijke en veilige manier kan besturen en laten werken.

Namelijk **eindschakelaars**, **drukknoppen** en een **noodstop**. Deze componenten zorgen ervoor dat het systeem correct en veilig werkt en kan reageren op verschillende situaties, zoals wanneer de veger het einde van het bord raakt (bewegingslimieten), besturing in de x en y richting van het bord (gebruikersinput). De noodstop schakelt de mijn bordveger uit wanneer hij ingedrukt wordt bij een onveilige situatie.

Hieronder volgt een uitleg van de verschillende knoppen en hun functie, en de elektrische aansluiting:



### 3.3.6.1 Eindschakelaars (limit switches)

- **Functie:** Ik gebruik 6 simpele eindschakelaars om het begin- en eindpunt van elke as aan te geven. De eindschakelaars worden gebruikt om de beweging van het whiteboard te beperken. Ze detecteren wanneer het bord een uiterste positie heeft bereikt en stoppen de motor om schade te voorkomen.
- **Aansluiting:** de eindschakelaars worden aangesloten op de ingangen van mijn microcontroller (de Arduino mega). Het zijn normaal open (NO) drukknoppen. Ze zijn dus ook **INPUT\_PULLDOWN**, zodat ze in ruststand LOW zijn en bij activering een HIGH-sigitaal geven.

Ik gebruik ook een extra eindstop om te controleren of mijn elektrische kast gesloten is: indien deze niet gesloten is zullen de motoren niet werken en het rode status lichtje gaat branden.



Figuur 32: Eindschakelaar

### 3.3.6.2 Drukknoppen

Voor de besturing van de veger worden meerdere drukknoppen gebruikt om de bewegingen te regelen en de veiligheid te waarborgen. Elke knop heeft een specifieke functie en zorgt mee voor een veilige werking van het systeem. Hieronder volgt een overzicht van de verschillende drukknoppen en hun functies:



Figuur 33: Drukknop

- 1) **Richtingsknoppen (omhoog, omlaag, links, rechts)**
    - Functie: Deze knoppen zorgen er voor dat de gebruiker in staat is om de veger in de gewenste richting te bewegen. Zodra een knop wordt ingedrukt, activeert deze de motoren om de veger in de juiste richting te sturen. Wanneer de knop wordt losgelaten, stopt de beweging direct om precieze controle te behouden.
  - 2) **Stopknop:**
    - Functie: De stopknop pauzeert onmiddellijk de werking van de veger. Dit zorgt ervoor dat de motoren in hun huidige positie blijven staan, waardoor onverwachte bewegingen worden voorkomen. De veger blijft stil staan totdat er opnieuw op een richtingsknop wordt gedrukt of de stopknop opnieuw wordt geactiveerd. Dit verhoogt de veiligheid en voorkomt ongewenste gevaarlijke situaties.
  - 3) **Homeknop:**
    - Functie: Met deze knop kan de gebruiker de veger terugbrengen naar de thuispositie. Wanneer de homeknop wordt ingedrukt, beweegt de veger automatisch naar een vooraf ingestelde rustpositie, zodat het systeem altijd een duidelijk startpunt heeft voor verdere bewegingen.
- **Aansluiting:** De knoppen worden aangesloten op digitale ingangen van mijn microcontroller (de Arduino mega). Eén zijde van de knop is verbonden met een ingang en de andere is verbonden met de gemeenschappelijke massa (GND).

Er worden ook pull-up weerstanden (deze zijn ingebouwd in de Arduino mega) gebruikt om een stabiel signaal te garanderen en zweefspanningen te vermijden.

- **Werking:** Wanneer de knop ingedrukt wordt, maakt het een elektrische verbinding tussen de twee aansluitpunten. Dit stuurt dan een signaal naar de microcontroller.

### 3.3.6.3 Noodstopshakelaar

- **Functie:** de noodstop is een belangrijke veiligheidsmaatregel. Wanneer deze wordt ingedrukt, onderbreekt hij onmiddellijk alle stroomtoevoer en stopt de besturing om alle bewegingen direct te stoppen.
- **Aansluiting:** de noodstop wordt in een Normally Closed (NC) configuratie gebruikt, wat betekent dat hij standaard een signaal doorlaat en bij activering de verbinding verbreekt. Dit zorgt ervoor dat bij een storing of losgekoppelde kabels het systeem automatisch stopt. Deze is in serie aangesloten met de voedingslijn van de besturing (de TB6600's en de Arduino mega). Dit zorgt voor een directe onderbreking van de motorstroom.



*Figuur 34: Noodstopshakelaar*



*Figuur 35: Noodstopshakelaar op mijn schakelkast*

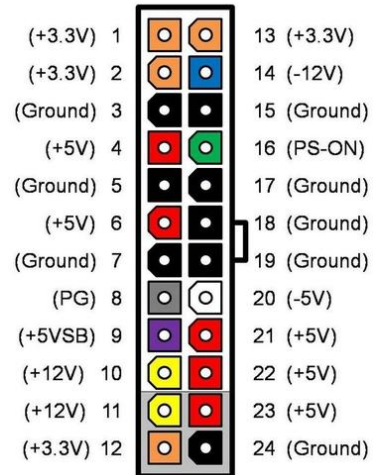
### 3.3.7 Voeding

Voor de stroomvoorziening van mijn whiteboard-veger gebruik ik een oude ATX pc voeding. Een ATX-voeding is een schakelende voeding (SMPS) die vanuit de 230 VAC-netspanning meerdere, vaste gelijkspanningen (+3,3V, +5V, +12V en -12V) levert en via een 3,3V PS\_ON sturing in- en uit schakelbaar is. In mijn toepassing voedt deze voeding mijn drie TB6600 modules en de step-down-converter voor de Arduino Mega.



Figuur 37: PC voeding

24 PIN CONNECTOR



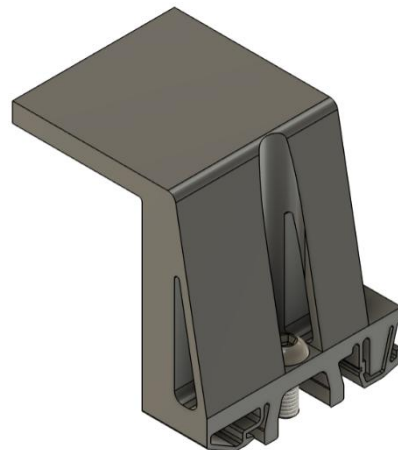
Figuur 36: PC voeding connector

#### 3.3.7.1 Montage

Voor de montage van de pc-voeding heb ik zelf een stevige beugel in Fusion 360 ontworpen en vervolgens 3D geprint. De beugel klikt vast op een standaard DIN-rail en wordt dan met een klemmschroef nog extra strak tegen de rail aangedrukt. Zodra de beugel op zijn plaats zit en niet verschuift, klik ik de voeding erin en span ik hem vast door de vijs aan te draaien. Op die manier klemt de voeding stevig en trillingvrij tussen de beugel en de rail.



Figuur 39: Voeding in de kast



Figuur 38: 3D model voeding houder

Om de voeding koel te houden, monteert ik een kleine 60mm ventilator aan de buitenzijde van mijn kast. Hiervoor heb ik ook een beschermkap geprint om te voorkomen dat je per ongeluk je vingers erin kunt steken.



### 3.3.7.2 Aansluitingen

Alle vier de 12V-kabels: drie naar de TB6600-drivers en één naar de LM2596-converter voor 9V naar de Arduino. laat ik netjes door kabelgoten lopen en bundel ik met tie-wraps. Tussen elke 12V-voeding en de drivers zitten glaszekeringen die bij een problemen de stroom direct onderbreekt. De verschillende GND-draden in de voeding zitten al intern met elkaar verbonden dus deze kan ik rechtstreeks op de negatieve ingang van de TB6600-drivers en de LM2596-converter aansluiten. Daarnaast heb ik een extra aardingsdraad van de pc-voeding naar een aardings-eindblok op de DIN-rail toegevoegd. Zo is de voeding en de kast goed geaard en blijft de hele installatie veilig.

### 3.3.7.3 Stroomberekening

Tabel 7: Stroomberekening onderdelen

Onderdeel:	Nominaal (A):	Piek (A)	Max bij 12 V (A)
Z-as motor 1 (TB6600)	2,5A	2,7A	2,7A
Z-as motor 2 (TB6600)	2,5A	2,7A	2,7A
X-as motor (TB6600)	1,5A	1,7A	1,7A
Arduino Mega	0,45A bij 12V	1A (max)	1A
<b>Totaal stroom:</b>			<b>≈ 8,1A</b>

Met een voeding die 16 A kan leveren en een geschat totaalverbruik van ongeveer 8 A, heb ik ruim twee keer zoveel capaciteit als nodig. Dat is meer dan voldoende om alle motoren en de Arduino veilig en zonder problemen van stroom te voorzien.

### 3.3.8 Zekeringen

In dit project maak ik gebruik van vier glaszekeringen om overstroom en kortsluiting te voorkomen en zo de TB6600-stappenmotordrivers en de Arduino Mega te beschermen. Iedere driver en de Vin van de Arduino Mega zijn voorzien van eenzekering met een nominale waarde net boven de maximale motorstroom voldoende om normale piekstromen toe te staan, maar laag genoeg om bij een fout snel door te slaan.

Tabel 8: Stroomberekening voor zekeringen motoren

As	Motortype	Nominale motorstroom (A)	TB6600 Gekozen stroom (A)	Piekstroom (A)
Z-as	SL42S247A (×2)	2,5A	2,5A	2,7A
X-as	SL42S234MA107-0524	1,2A	1,5A	1,7A

#### 3.3.8.1 Z-as motorstroom

Voor de Z-as gebruik ik twee identieke stappenmotoren van elk 2,5A met een piekstroom van 2,7 A. Deze worden parallel aangestuurd door twee TB6600-drivers. Om de motoren te beschermen bij overbelasting of kortsluiting, gebruik ik twee glaszekeringen van 3A.

#### 3.3.8.2 X-as motorstroom

De stappenmotor voor de X-as verbruikt normaal 1,2A. Maar op de tb6600 kies ik 1,5A als normaalstroom dus hiervoor is eenzekering van 2 A geschikt. Dit biedt ook voldoende marge voor de piekstromen tot 1,7A

#### 3.3.8.3 Arduino Mega stroom:

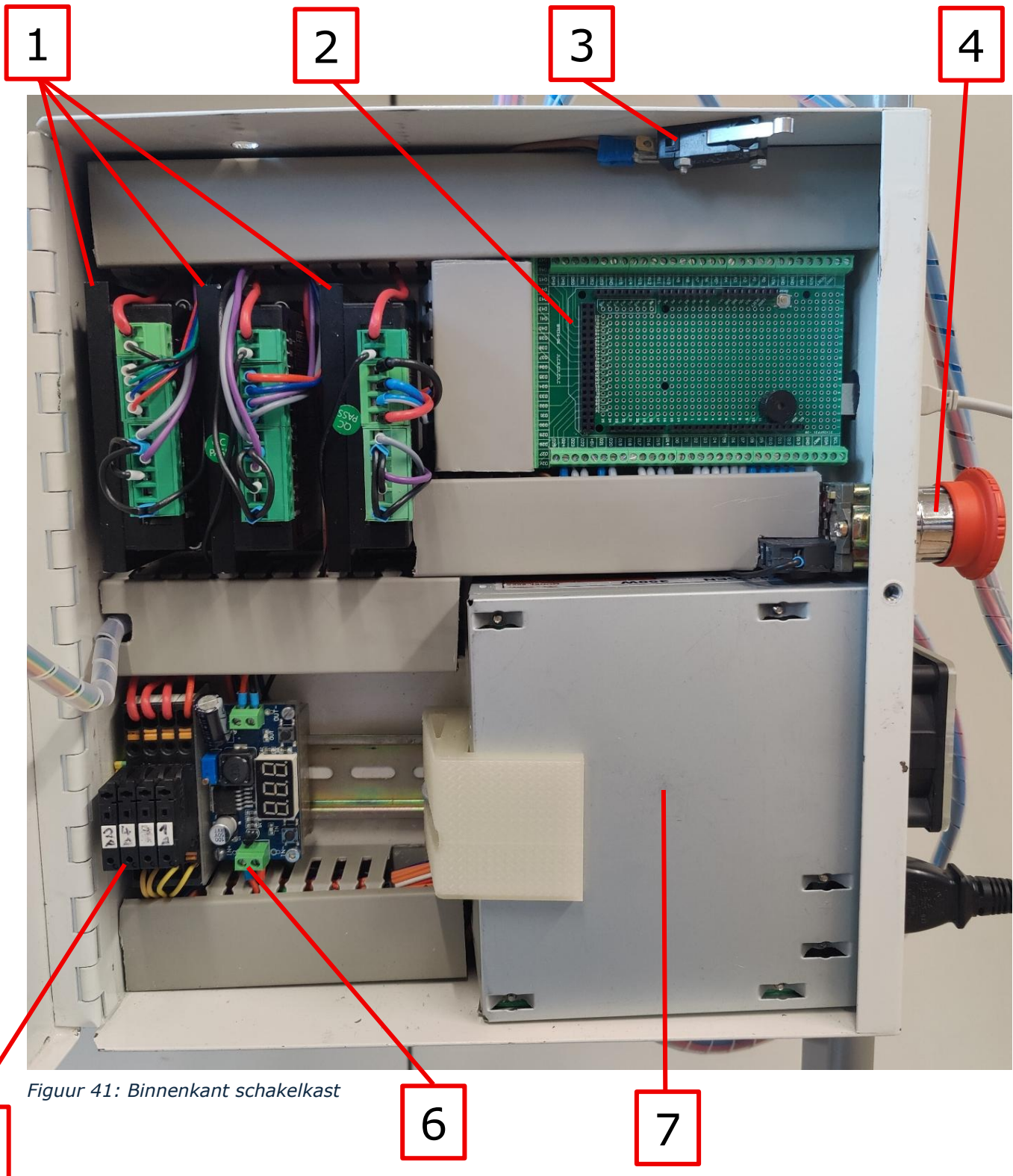
De Arduino Mega zelf verbruikt relatief weinig stroom (ongeveer 0,5 A). Hiervoor gebruik ik eenzekering van 0,6 A. De Arduino Mega heeft ook een ingebouwde polyfuse (zelfherstellendezekering), maar deze werkt alleen wanneer je de Arduino via de USB-poort van stroom voorziet. Aangezien ik dit niet doe, gebruik ik een apartezekering.



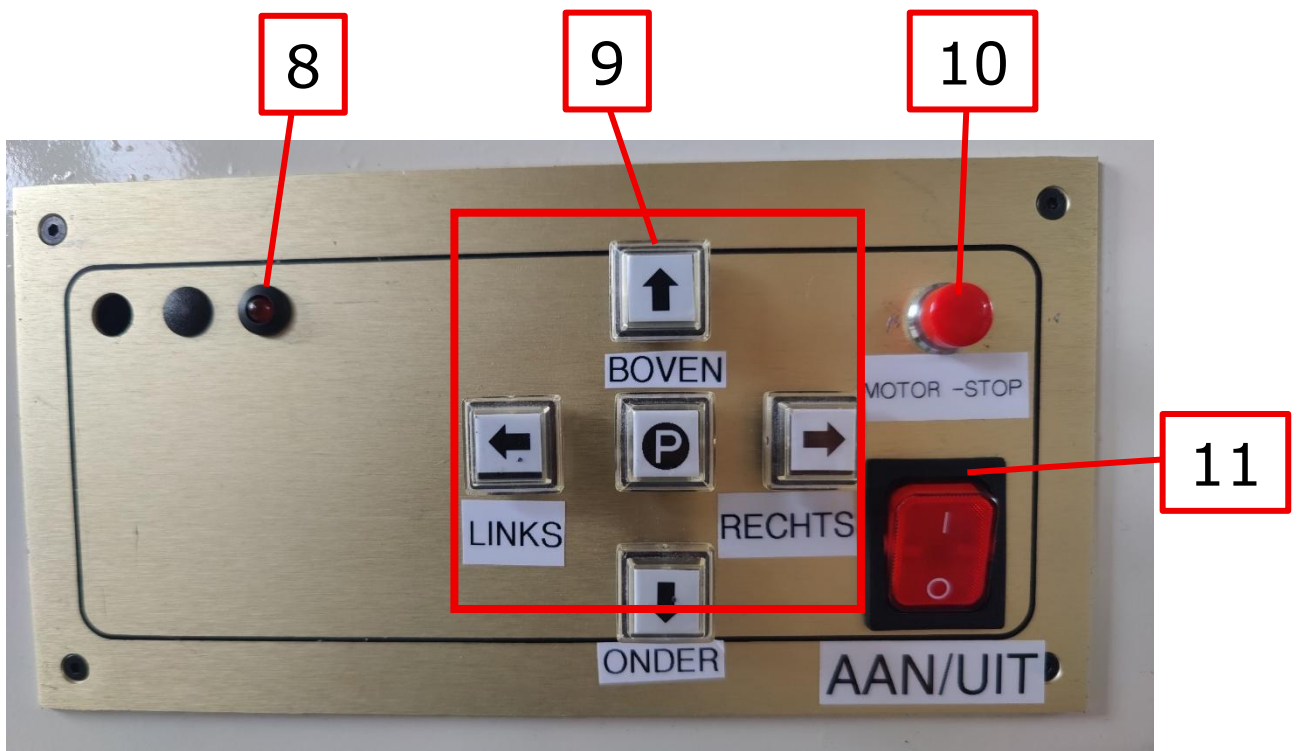
Figuur 40: Glaszekering

### 3.4 Schakelkast Overzicht

Hieronder ziet u mijn schakelkast, waarin elk onderdeel is voorzien van een nummer. In de bijbehorende tabel vindt u per nummer zowel de naam als een korte functiebeschrijving, zodat u in één oogopslag ziet wát zich waar in de kast bevindt en welke rol het onderdeel vervult.



Figuur 41: Binnenkant schakelkast

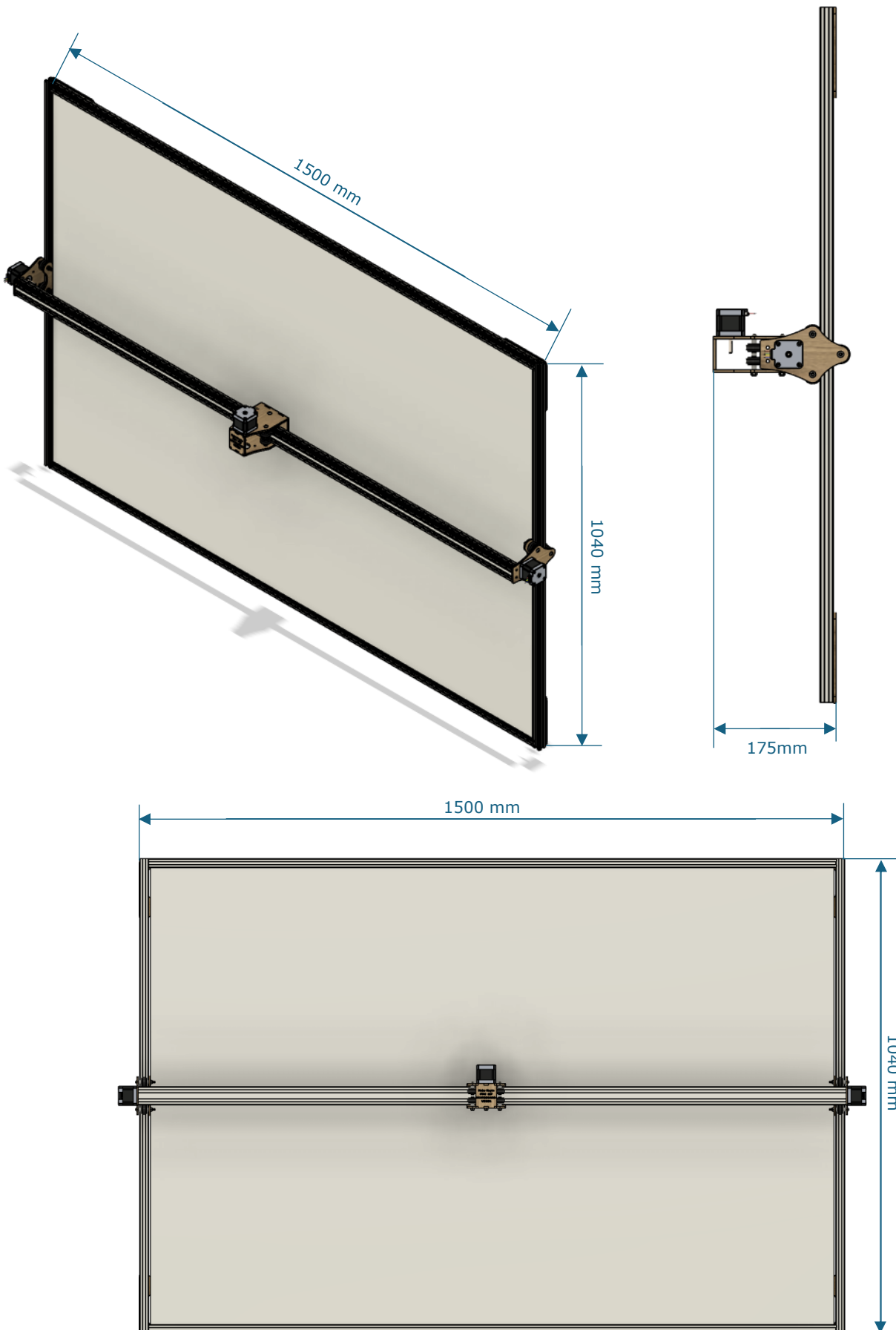


Figuur 42: bedieningspaneel bordveger

Tabel 9: Naam en functie onderdelen schakelkast

Nummer:	Naam onderdeel:	Functie:
1	TB6600 Stapper Driver (3.3.2)	Stuurt de stappenmotoren aan en regelt stroom en pulsen.
2	Arduino Mega (3.3.1)	Stuurt alles aan en voert de code uit.
	Case close knop (3.4.1.1)	Controleert of de kast gesloten is. Als de kast open is, werken de motoren niet en gaat een rood lampje branden.
4	Noodstopshakelaar (3.4.1.3)	Onderbreekt onmiddellijk de voeding (stroom) bij noodsituaties.
5	Zekeringen	Beschermen de onderdelen en bekabeling tegen overbelasting en kortsluiting.
6	LM2596 (DC-DC converter)	Zet de binnenkomende spanning om van 12V naar 9V voor de Arduino.
7	Powersupply	Voorziet al de onderdelen van stroom.
8	Status led	Geeft de status visueel weer bv: case open, motoren gestopt...
9	Besturingsknoppen (3.4.1.2)	Handmatige bediening: vooruit, achteruit, links, rechts...
10	Motor Stopknop (3.4.1.2)	Zet de motoren stil zonder ze spanningsloos te maken.
11	Aan/uit-hoofdschakelaar	Zet de volledige kast in- of uit.

### 3.5 Maquetteschets



Figuur 43: Maquetteschets whiteboardveger



## 3.6 Modules

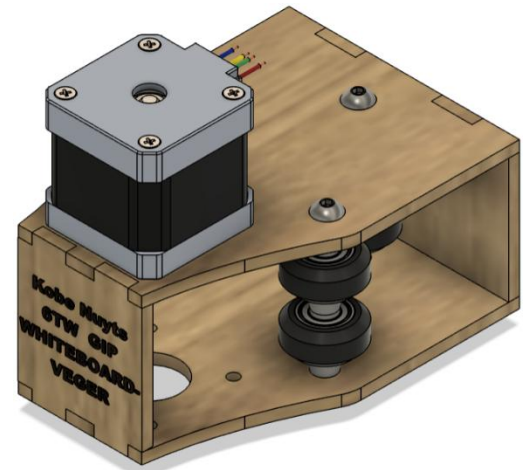
De tekeningen van de modules heb ik in Fusion 360 gemaakt, omdat ik goed met dit programma kan werken en ik met deze tekeningen zowel makkelijk kan lasersnijden en 3D-printen. Voor het lasersnijden werk ik met Adobe Illustrator om de ontwerpen klaar te maken voor de lasersnijder. Voor de 3D prints klaar te maken (te slicen) gebruik ik Ultimaker Cura. Voor materialen gebruik ik triplex platen van 4 mm, omdat dit materiaal licht, stevig en gemakkelijk te bewerken is. De 3D-geprinte onderdelen maak ik uit PETG (voor sterke, duurzame stukken) en PLA (voor onderdelen die minder zwaar worden belast). Dankzij deze combinatie van hout en kunststof kan ik een heel stabiele constructie bouwen, waarbij elk onderdeel precies wordt gemaakt en perfect op elkaar aansluit.

### 3.6.1 X-as module

De Veger- of X-as module bestaat grotendeels uit gelaserd triplex voor het frame. Dit houten onderdeel is stevig, snel en nauwkeurig uit te snijden.

Het mechanisme zelf wordt aangedreven door een stappenmotor waar een riem over wordt gespannen, die zorgt voor de lineaire beweging langs de X-as. Dankzij de stappenmotor kan de module precies worden gepositioneerd en bewogen, wat belangrijk is voor een soepele en nauwkeurige werking.

De ondersteunende onderdelen zoals de geleiders en wielen worden aangekocht omdat deze niet zelf te maken zijn en ze zeker sterk en precies gemaakt moeten worden.

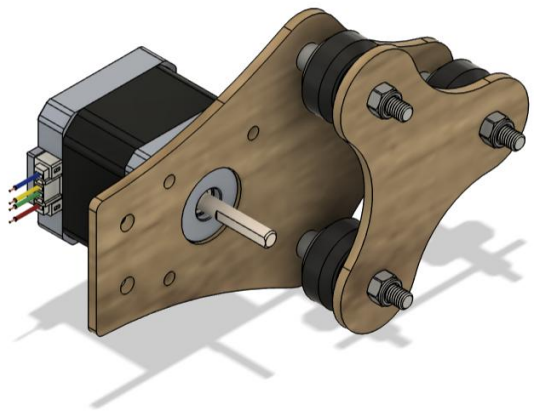


Figuur 44: X-as module Bordveger

### 3.6.2 Z-as modules

De Z-as modules zijn op dezelfde manier opgebouwd als de Y-as module, maar bevinden zich aan de zijkanten van de constructie. Het frame bestaat uit triplex, wat snel te snijden is en tegelijkertijd ook stevig is.

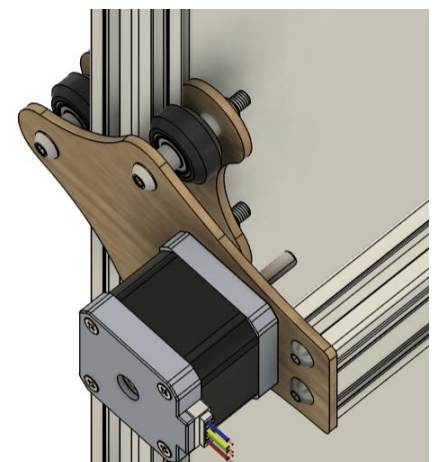
Het mechanisme wordt net als bij de X-as aangedreven door een stappenmotor met een riem, die zorgt voor de lineaire beweging langs de Z-as. Door deze configuratie kan de module heel precies worden gepositioneerd en bewogen, wat belangrijk is voor een soepele en nauwkeurige verticale verplaatsing.



Figuur 45: Z-as module Bordveger

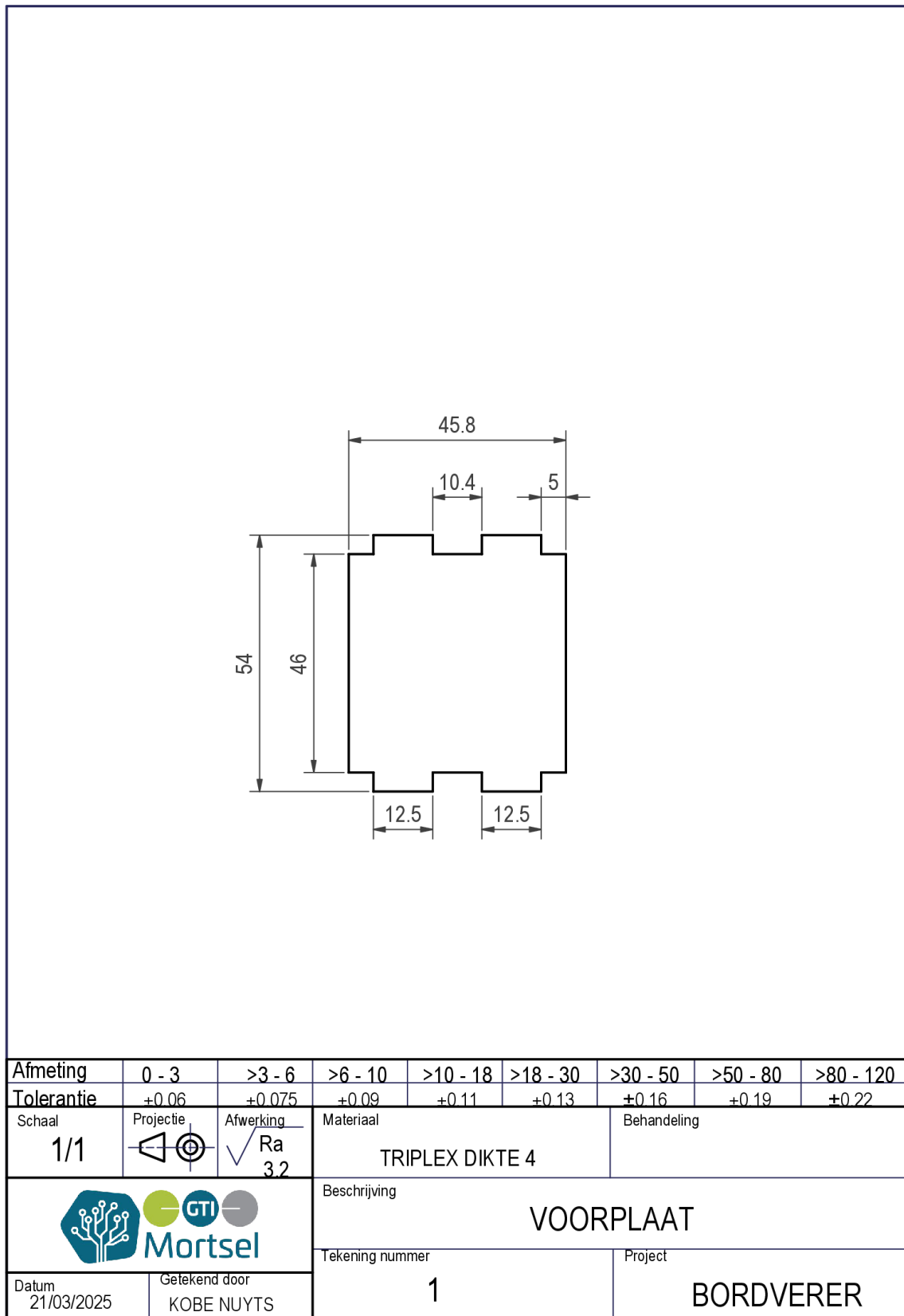
Er zijn 2 dezelfde Z-as modules die tegelijkertijd aan beide kanten werken en met elkaar verbonden zijn met een 20x40 X-profiel. Dit is ook het X-profiel waarover de Y-as module zal bewegen.

De ondersteunende onderdelen, zoals geleiders en wielen, worden aangekocht om zeker te zijn dat ze sterk en nauwkeurig genoeg zijn. Deze onderdelen zorgen voor een stabiele geleiding en een lange levensduur, waardoor de Z-as soepel en betrouwbaar blijft werken.

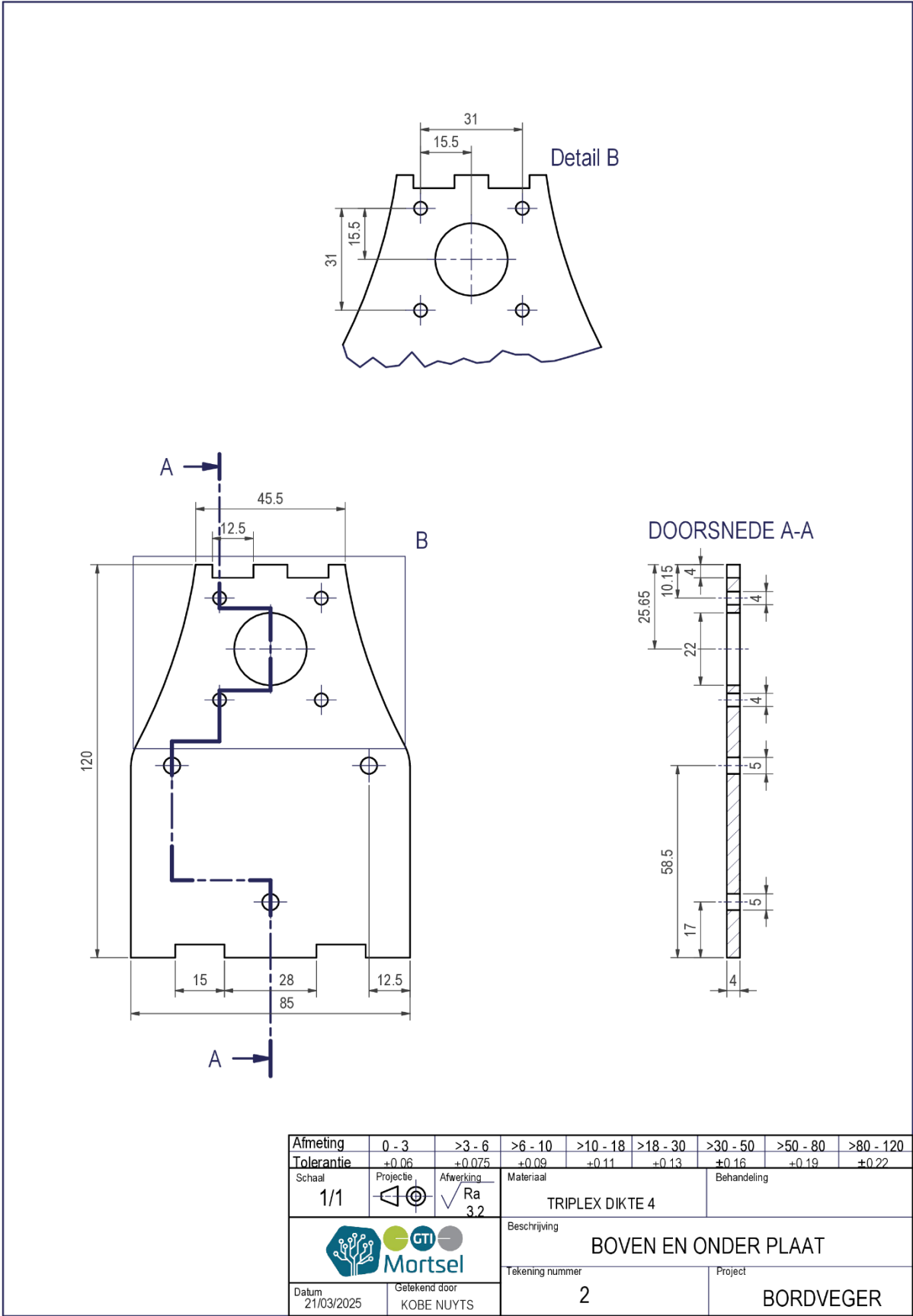


Figuur 46: Z-as module Bordveger

### 3.7 Technische tekeningen

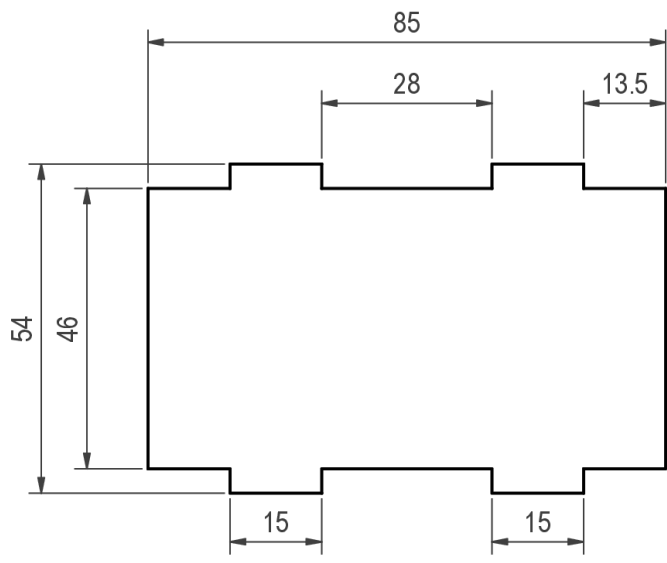


Figuur 47: Voorplaat X-as module



Figuur 48: Boven en onder plaat X-as module

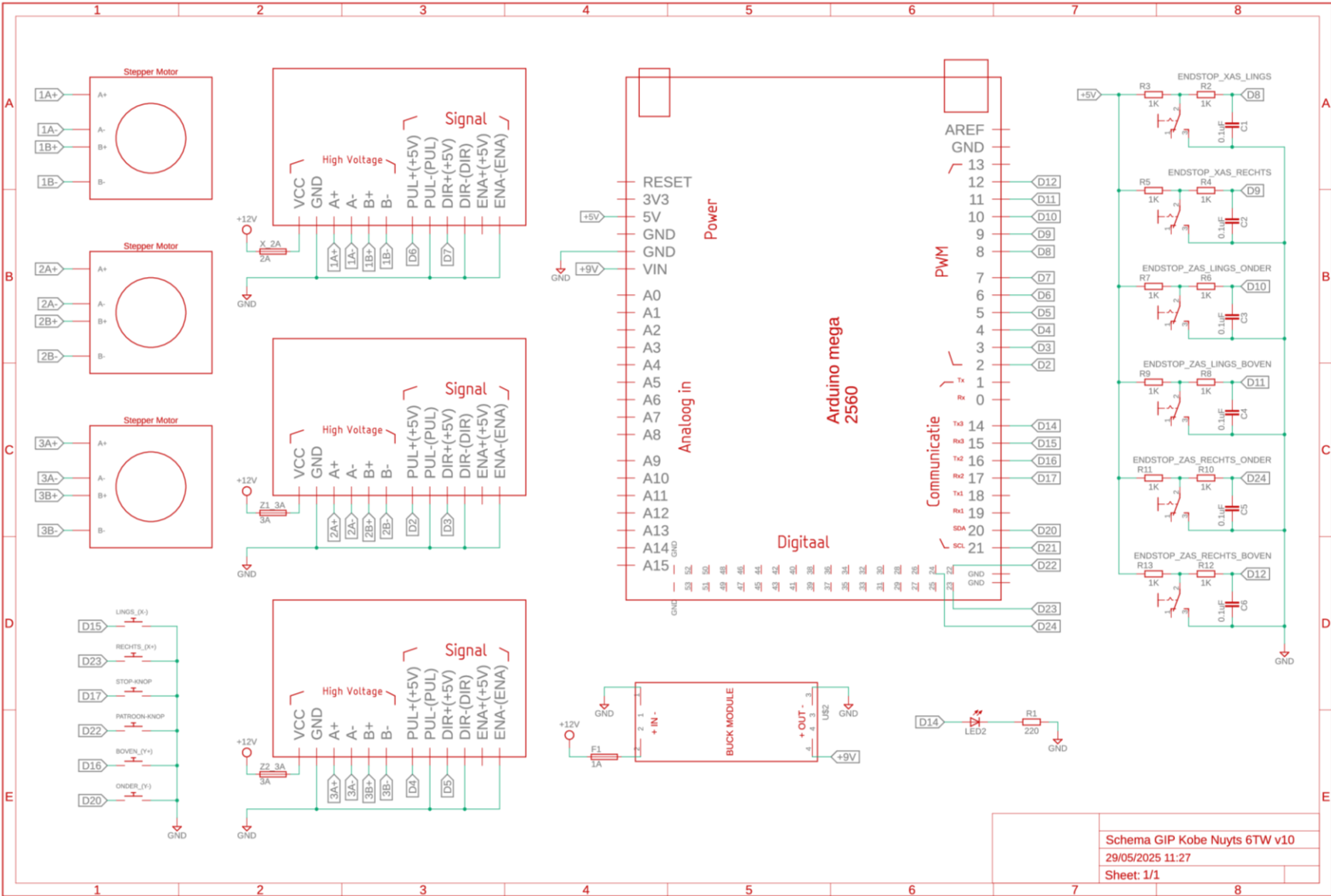




Afmeting	0 - 3	>3 - 6	>6 - 10	>10 - 18	>18 - 30	>30 - 50	>50 - 80	>80 - 120
Tolerantie	+0.06	+0.075	+0.09	+0.11	+0.13	+0.16	+0.19	+0.22
Schaal	Projectie	Afwerking	Materiaal	Behandeling				
1/1		√ Ra 3.2	TRIPLEX DIKTE 4					
			Beschrijving					
			ACHTERPLAAT					
			Tekening nummer			Project		
Datum			3			BORDVERER		
21/03/2025			Getekend door					
			KOBE NUYTS					

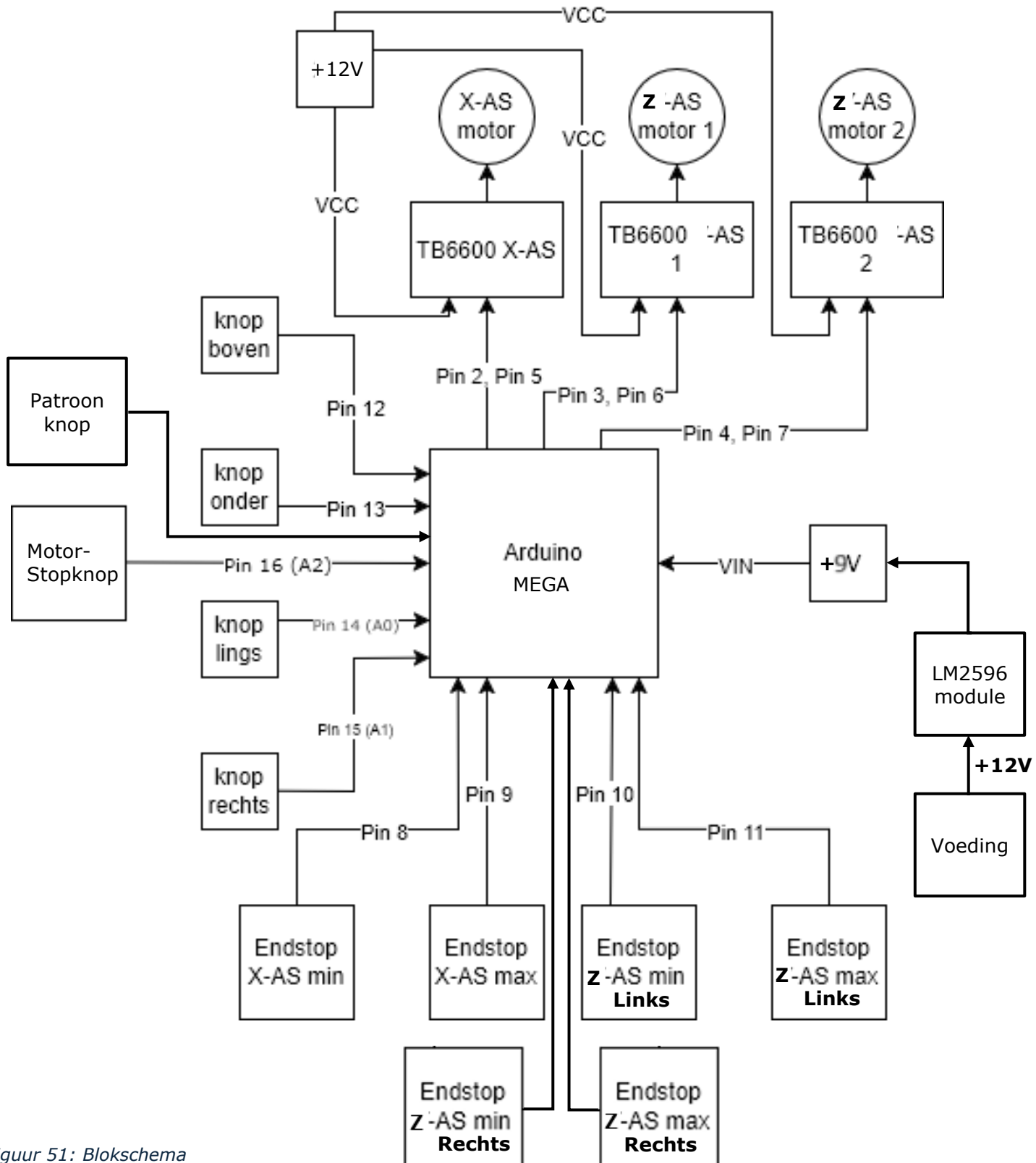
Figuur 49: Achterste plaat X-as module

### 3.8 Bedradingschema



Figuur 50: Volledig Bedradingschema

### 3.9 Blokschema



Figuur 51: Blokschema

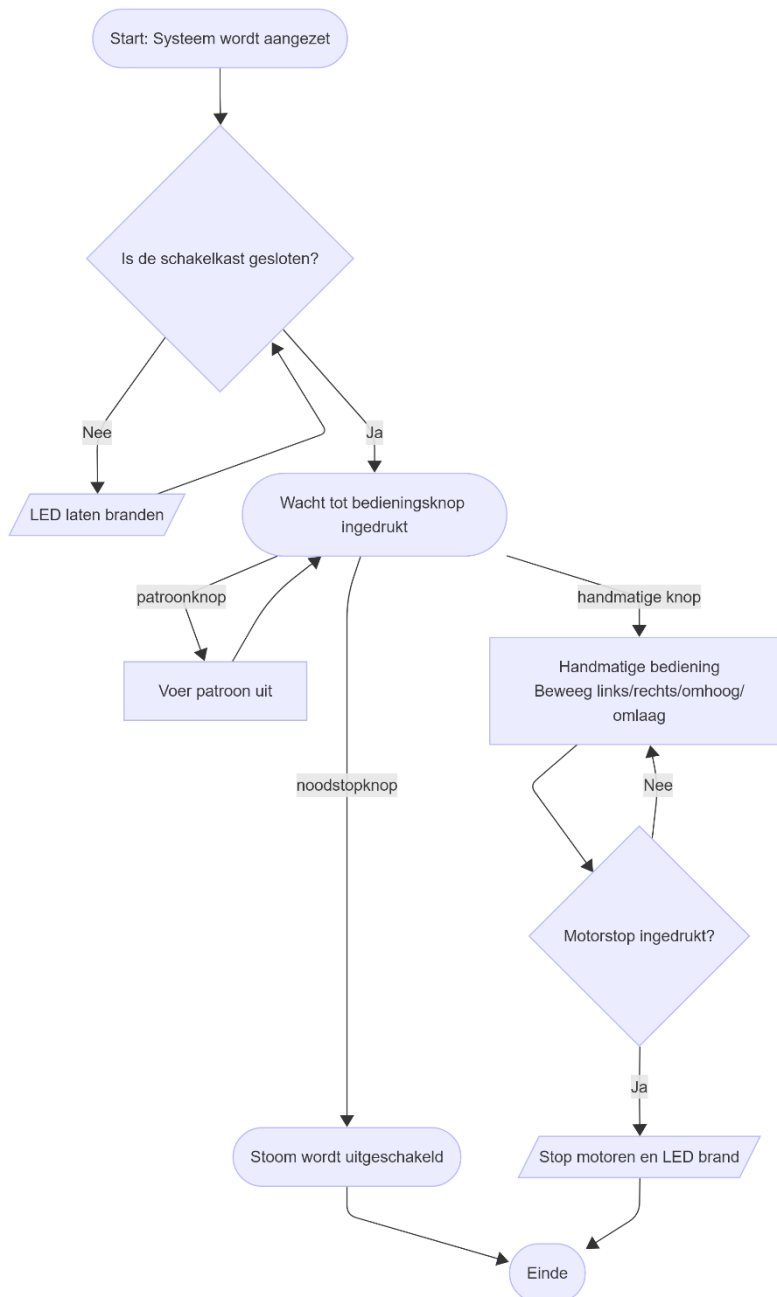
## 4 Code:

### 4.1 Inleiding tot de Arduino Code

De kern van mijn bordveger wordt gevormd door de Arduino code. Deze code is verantwoordelijk voor het aansturen van de motoren, het uitlezen van de knoppen en veiligheidssensoren, en het uitvoeren van de logica die de bordveger bedient.

### 4.2 Flowchart

Om de werking van deze code beter te begrijpen, heb ik een simpele flowchart opgesteld. Deze flowchart laat de verschillende stappen en beslissingen die de code neemt zien, van het opstarten van het systeem en de veiligheidscontroles, tot de handmatige besturing en de reactie op noodstops. De code en de flowchart werken samen om een duidelijk beeld te geven van hoe de bordveger functioneert:



Figuur 52: Flowchart code

## 4.3 Overzicht van de Code

Deze Arduino-code bestuurt een CNC-achtig systeem dat verticaal met het oppervlak van het bord staat met twee Z-motoren (die parallel werken) en één X-motor via TB6600 stepper drivers.

## 4.4 Wat doet de code?

### 4.4.1 Bibliotheken

Ik maak enkel gebruik van de AccelStepper-bibliotheek voor motorbesturing.

### 4.4.2 Setup (**setup()**)

- Stelt de maximale snelheid en acceleratie in.
- Configureert de knoppen en eindstoppen.

### 4.4.3 Hoofdprogramma (**loop()**)

- Controleert of een bewegingsknop wordt ingedrukt en stuurt de motoren aan:
  - **Omhoog (Y+)** → Beide Y-motoren bewegen vooruit.
  - **Omlaag (Y-)** → Beide Y-motoren bewegen achteruit.
  - **Links (X-)** → X-motor beweegt naar links.
  - **Rechts (X+)** → X-motor beweegt naar rechts
- Motorstop-knop stopt alle motoren.

### 4.4.4 Bewegingsfuncties (**moveAxis()**)

- Beweegt een de X-motor of beide Z-motoren tegelijk.
- Wacht tot de beweging voltooid is voordat de code verdergaat.

### 4.4.5 Stopfunctie (**stopAllMotors()**)

- Stopt alle motoren

### 4.4.6 Pinconfiguratie

#### 4.4.6.1 Motoren:

- **X-as motor:**

Tabel 10: Pinconfiguratie X-as motor

TB6600 (X-as)	Arduino MEGA
PUL+ (Stap)	Pin 6
DIR+ (Direction)	Pin 7
ENA+ (Enable)	<b>Niet nodig om aan te sluiten</b>
PUL- (GND)	GND
DIR- (GND)	GND
ENA- (GND)	GND

- **Z-as motoren (2 parallel werkende motoren):**

Tabel 11: Pinconfiguratie Z-as motor 1

TB6600 (Z1-motor)	Arduino MEGA
PUL+ (Stap)	Pin 2
DIR+ (Direction)	Pin 3
ENA+ (Enable)	<b>Niet nodig om aan te sluiten</b>
PUL- (GND)	GND
DIR- (GND)	GND
ENA- (GND)	GND

Tabel 12: Pinconfiguratie Z-as motor 2

TB6600 (Z2-motor)	Arduino MEGA
PUL+ (Stap)	Pin 4
DIR+ (Direction)	Pin 5
ENA+ (Enable)	<b>Niet nodig om aan te sluiten</b>
PUL- (GND)	GND
DIR- (GND)	GND
ENA- (GND)	GND

Bij de stappenmotoren zelf worden ze 2 spoelen aangesloten op de A+ A- B+ B- aansluitingen van de TB6600.

#### 4.4.6.2 Knoppen:

- **Eindstoppen (limit switches):**

Er zijn vier eindstoppen nodig om de grenzen van de assen te detecteren. De eindstoppen en knoppen zijn INPUT\_PULLUP, zodat ze actief LOW zijn.

Tabel 13: Pinconfiguratie Eindstops

Eindstop	Arduino MEGA	Aansluiting
X-min	Pin 8	Eén kant naar GND, andere naar pin 8
X-max	Pin 9	Eén kant naar GND, andere naar pin 9
Z-min-links	Pin 10	Eén kant naar GND, andere naar pin 10
Z-max-links	Pin 11	Eén kant naar GND, andere naar pin 11
Z-min-rechts	Pin 12	Eén kant naar GND, andere naar pin 12
Z-max-rechts	Pin 24	Eén kant naar GND, andere naar pin 24

De eindstoppen werken met pull-up weerstanden (deze zijn ingebouwd in de Arduino Mega). Dit betekent dat de ingang normaal HIGH is, en naar LOW schakelt wanneer de knop wordt ingedrukt.

- **Drukknoppen (voor de besturing):**

Ik gebruik 6 drukknoppen voor de besturing van de motoren, een Lings rechtsboven en onder knop en een noodstop die alle motoren stopt en in die positie houdt.

Tabel 14: Pinconfiguratie drukknoppen

Functie	Arduino UNO
Boven: <b>Up (Y+)</b>	Pin 16
Onder: <b>Down (Y-)</b>	Pin 20
Lings: <b>Left (X-)</b>	Pin 15
Rechts: <b>Right (X+)</b>	Pin 23
Motorstop-knop	Pin 17
Patroon-knop	Pin 22

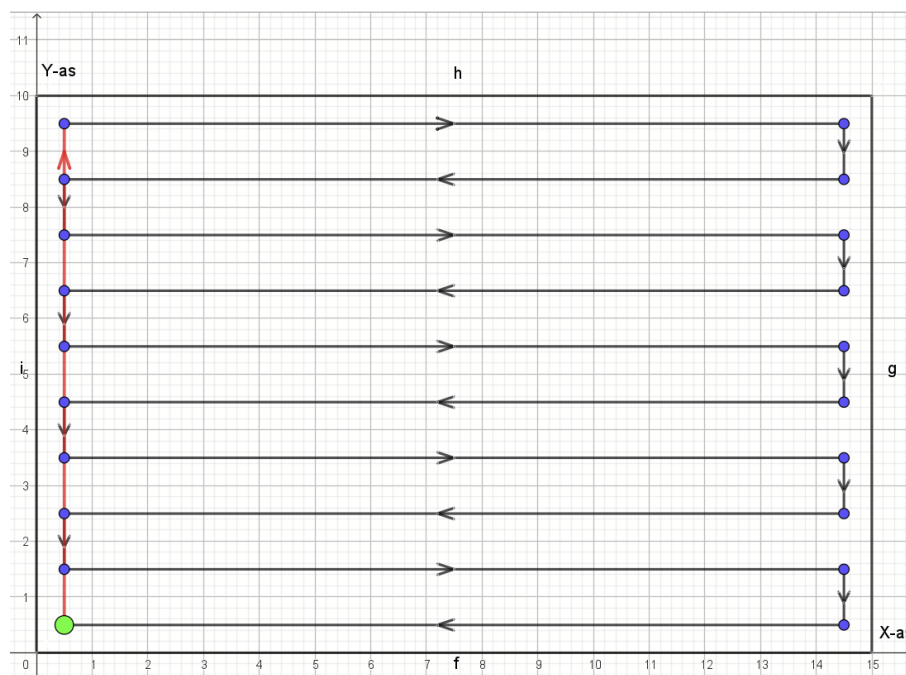
Ook Elke drukkноп wordt met één kant op GND aangesloten, en de andere kant op de Arduino-pin. De Arduino gebruikt interne pull-up weerstanden waardoor de knop normaal HIGH is en LOW wordt als erop gedrukt wordt.

Op pin 21 is de case close knop aangesloten die checkt als mijn schakelkast open of gesloten is. En de motoren blokkeert indien niet gesloten.

- **Patroon knop (uitbreiding):**

Als hier tijd voor is wil ik later nog uitbreiden en een patroon knop toevoegen die een patroon afgaat en heel het board proper veegt door op een knop te drukken.

Het patroon Start in de linker onder hoek (groene bol) en volg eerst de rode lijn en daarna de richting van de zwarte pijlen en lijnen en die bij elke blauwe bol van kant veranderd (naar onder/ boven Lings rechts) en stopt terug bij de groene bol.



Figuur 53: Foto patroon

## 4.4.7 Code

```
// Inclusief de AccelStepper bibliotheek.
// Deze bibliotheek maakt het eenvoudig om stappenmotoren te besturen,
// inclusief functies voor acceleratie, vertraging en positionering.
#include <AccelStepper.h>

// Definieert het driver type voor de stappenmotordrivers.
// '1' staat voor een driver die aparte STEP en DIR (richting) pinnen gebruikt,
// zoals de TB6600 die hier wordt gebruikt.
#define DRIVER 1

// Pin toewijzingen voor de TB6600 stappenmotordrivers op de Arduino Mega.
// Elke stappenmotor heeft een PUL-pin (voor het sturen van stappen) en een DIR-
// pin (voor de draairichting).
#define Z2_PUL_PIN 2 // Digitale pin 2: PUL-pin voor de tweede Z-as motor (Z2)
#define Z2_DIR_PIN 3 // Digitale pin 3: DIR-pin voor de tweede Z-as motor
(Z2)
#define Z1_PUL_PIN 4 // Digitale pin 4: PUL-pin voor de eerste Z-as motor (Z1)
#define Z1_DIR_PIN 5 // Digitale pin 5: DIR-pin voor de eerste Z-as motor
(Z1)
#define X_PUL_PIN 6 // Digitale pin 6: PUL-pin voor de X-as motor
#define X_DIR_PIN 7 // Digitale pin 7: DIR-pin voor de X-as motor

// Pin toewijzingen voor de bedieningsknoppen.
// Deze knoppen worden gebruikt voor handmatige besturing en systeemfuncties.
#define BUTTON_UP_PIN 16 // Digitale pin 16: Knop voor beweging omhoog
(Z-as)
#define BUTTON_DOWN_PIN 20 // Digitale pin 20: Knop voor beweging omlaag
(Z-as)
#define BUTTON_LEFT_PIN 15 // Digitale pin 15: Knop voor beweging naar
links (X-as)
#define BUTTON_RIGHT_PIN 23 // Digitale pin 23: Knop voor beweging naar
rechts (X-as)
#define BUTTON_STOP_PIN 17 // Digitale pin 17: Motor-stopknop, stopt alle
beweging van de motoren
#define BUTTON_CASE_CLOSE_PIN 21 // Digitale pin 21: Veiligheidsknop die
aangeeft of de behuizing gesloten is

// Pin toewijzing voor de status-LED.
// Deze LED geeft feedback over de systeemstatus (bijv. stop actief, fout of
behuizing open...).
#define LED_PIN 14 // Digitale pin 14: indicatie-led

// Snelheidsinstellingen voor de stappenmotoren.
// Deze waarden bepalen hoe snel en geleidelijk de motoren bewegen.
#define X_MAX_SPEED 4000 // Maximale snelheid in stappen per seconde voor
de X-as. Hoe hoger, hoe sneller.
#define X_ACCELERATION 200 // Acceleratie in stappen per seconde per seconde
voor de X-as. Hoe hoger, hoe sneller de motor op snelheid komt.

#define Z_MAX_SPEED 200 // Maximale snelheid in stappen per seconde voor
de Z-assen (Z1 en Z2). Lager dan X voor meer controle en kracht.
#define Z_ACCELERATION 100 // Acceleratie in stappen per seconde per seconde
voor de Z-assen. Lager dan X voor soepelere beweging.

// Debounce instellingen voor knoppen.
// Debouncing voorkomt dat een enkele druk op een knop als meerdere snelle
drukken wordt geregistreerd.
```



```

unsigned long lastStopPress = 0; // Tijdstempel van de laatste keer dat de
Stopknop werd ingedrukt.
const unsigned long debounceDelay = 200; // Minimale tijd (in milliseconden)
tussen twee geldige knopdrukken.

// Maak AccelStepper objecten aan voor elke stappenmotor.
// Deze objecten beheren de beweging van de motoren.
AccelStepper stepperX(DRIVER, X_PUL_PIN, X_DIR_PIN); // Object voor de X-as
motor
AccelStepper stepperZ1(DRIVER, Z1_PUL_PIN, Z1_DIR_PIN); // Object voor de eerste
Z-as motor
AccelStepper stepperZ2(DRIVER, Z2_PUL_PIN, Z2_DIR_PIN); // Object voor de tweede
Z-as motor

// Statusvariabelen die de huidige toestand van het systeem bijhouden.
bool stopActive = false; // Vlag die aangeeft of de motorstop actief is (true =
actief, false = inactief).
// caseClosed is TRUE als de knop NIET is ingedrukt (d.w.z. dat de behuizing
'open' of onveilig is).
bool caseClosed = true; // Vlag die de status van de behuizing aangeeft.

// Variabelen om de vorige status van de knoppen bij te houden voor de "single
print" functie.
// Dit voorkomt dat er constant berichten naar de seriële monitor worden
gestuurd als een knop ingedrukt blijft.
// Het bericht zal pas veranderen als er een andere knop word ingedrukt of de
status veranderd.
bool lastStopButtonState = HIGH; // Vorige staat van de Motorstop-knop
(HIGH = niet ingedrukt, LOW = ingedrukt).
bool lastCaseCloseButtonState = HIGH; // Vorige staat van de Case Close-knop.
bool lastUpButtonState = HIGH; // Vorige staat van de BOVEN-knop.
bool lastDownButtonState = HIGH; // Vorige staat van de ONDER-knop.
bool lastLeftButtonState = HIGH; // Vorige staat van de LINKS-knop.
bool lastRightButtonState = HIGH; // Vorige staat van de RECHTS-knop.
bool lastConflictState = false; // Vlag die aangeeft of er een handmatig
bewegingsconflict is (meerdere knoppen tegelijk zijn ingedrukt).

// Systeemstatusvariabelen:
// Dit definieert de mogelijke staten waarin het systeem zich kan bevinden.
enum SystemState {
    IDLE, // Het systeem is inactief en wacht op commando's.
    MANUAL_CONTROL // Het systeem wordt handmatig bestuurd via de richting
knoppen.
};
SystemState currentSystemState = IDLE; // De huidige staat van het systeem
begint altijd in IDLE.

// Functie prototypes: declaraties van functies die later in de code worden
gedefinieerd.
// Dit vertelt de compiler dat deze functies bestaan voordat ze daadwerkelijk
worden geïmplementeerd.
void stopAllMotors(); // Stopt alle stappenmotoren onmiddellijk.
void stopMotion(); // Zet de snelheid van alle motoren op nul, waardoor ze
stoppen.

// Functie voor de X-as motor: Beweegt de X-as motor continu.
// 'm' is de motor die moet bewegen, 'dir' is de richting (bijv. 1 voor links, -
1 voor rechts).
void moveX_AS(AccelStepper &m, int dir);

// Functie voor de Z-as motoren: Beweegt beide Z-as motoren tegelijk continu.

```

```

// 'm1' is de eerste Z-as motor, 'm2' is de tweede Z-as motor, 'dir' is de
// richting (bijv. 1 voor omhoog, -1 voor omlaag).
void moveZ_AS(AccelStepper &m1, AccelStepper &m2, int dir);

//De setup() functie wordt één keer uitgevoerd wanneer de Arduino start of
//reset.
//Hier worden alle initiële configuraties gedaan, zoals pinmodusen en seriële
//communicatie.
void setup() {
  // Start seriële communicatie met een baudrate van 9600.
  // Dit maakt het mogelijk om berichten naar de seriële monitor op de computer
  // te sturen.
  Serial.begin(9600);
  Serial.println("Setup gestart..."); // Stuurt een bericht naar de seriële
  // monitor.

  // Configureer de maximale snelheid en acceleratie voor elke stappenmotor.
  // Deze instellingen worden gebruikt door de AccelStepper bibliotheek om de
  // beweging te regelen.
  stepperX.setMaxSpeed(X_MAX_SPEED);
  stepperX.setAcceleration(X_ACCELERATION);
  stepperZ1.setMaxSpeed(Z_MAX_SPEED);
  stepperZ1.setAcceleration(Z_ACCELERATION);
  stepperZ2.setMaxSpeed(Z_MAX_SPEED);
  stepperZ2.setAcceleration(Z_ACCELERATION);

  // Configureer de knoppenpinnen als INPUT_PULLUP.
  // INPUT_PULLUP activeert de interne pull-up weerstanden van de Arduino.
  // Dit betekent dat de pin standaard HIGH is, en LOW wordt wanneer de knop
  // wordt ingedrukt (verbonden met GND).
  pinMode(BUTTON_UP_PIN, INPUT_PULLUP);
  pinMode(BUTTON_DOWN_PIN, INPUT_PULLUP);
  pinMode(BUTTON_LEFT_PIN, INPUT_PULLUP);
  pinMode(BUTTON_RIGHT_PIN, INPUT_PULLUP);
  pinMode(BUTTON_STOP_PIN, INPUT_PULLUP);
  pinMode(BUTTON_CASE_CLOSE_PIN, INPUT_PULLUP);

  // Configureer de LED-pin als OUTPUT.
  // Een OUTPUT-pin kan worden gebruikt om spanning te leveren (HIGH) of te
  // verwijderen (LOW).
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW); // Zet de LED bij start UIT.

  Serial.println("Setup voltooid."); // Stuurt een bericht naar de seriële
  // monitor dat de setup klaar is.
}

//De loop() functie wordt continu herhaald nadat setup() is voltooid.
//Dit is de hoofdloop van het programma waar alle logica voor besturing en
//statusupdates plaatsvinden.
void loop() {
  unsigned long now = millis(); // Haalt de huidige tijd in milliseconden op
  // sinds de Arduino startte.

  // 1. Lees knop statussen (met enkele afdruk)
  // Deze sectie leest de status van de Stopknop en de Case Close-knop.
  // De logica zorgt ervoor dat berichten slechts één keer worden afgedrukt bij
  // een statusverandering.

  // Lees de huidige status van de Stopknop.
  // De knop is 'ingedrukt' als de pin LOW is (actief laag).

```

```

bool curStop = (digitalRead(BUTTON_STOP_PIN) == LOW);
// Controleert of de knop nu is ingedrukt en eerder niet was ingedrukt.
if (curStop && !lastStopButtonState) {
    stopActive = !stopActive; // Wisselt de status van 'stopActive' (aan/uit).
    // Print de nieuwe status van de stopfunctie naar de seriële monitor.
    Serial.print("Motor-stop getoggled: ");
    Serial.println(stopActive ? "ACTIEF" : "INACTIEF");
}
lastStopButtonState = curStop; // Update de vorige staat van de Stopknop.

// Lees de staat van de Case Close knop.
bool physicalCaseCloseButtonState = (digitalRead(BUTTON_CASE_CLOSE_PIN) ==
LOW);
// Controleert op verandering van de Case Close knop status voor deze in de
seriële monitor te kunnen printen.
if (physicalCaseCloseButtonState && !lastCaseCloseButtonState) {
    Serial.println("Case Close knop ingedrukt (behuizing gesloten).");
} else if (!physicalCaseCloseButtonState && lastCaseCloseButtonState) {
    Serial.println("Case Close knop losgelaten (behuizing open).");
}
lastCaseCloseButtonState = physicalCaseCloseButtonState; // Update de vorige
staat van de Case Close knop.
// caseClosed is TRUE als de knop NIET is ingedrukt (d.w.z. dat de behuizing
open of onveilig is).
// Hier wordt de status van caseClosed bepaald: HIGH betekent dat de knop niet
is ingedrukt (staat open).
caseClosed = (digitalRead(BUTTON_CASE_CLOSE_PIN) == HIGH);

// Lees de huidige staat van de handmatige bewegingsknoppen.
bool currentUp = (digitalRead(BUTTON_UP_PIN) == LOW); // BOVEN-knop status
bool currentDown = (digitalRead(BUTTON_DOWN_PIN) == LOW); // ONDER-knop status
bool currentLeft = (digitalRead(BUTTON_LEFT_PIN) == LOW); // LINKS-knop status
bool currentRight = (digitalRead(BUTTON_RIGHT_PIN) == LOW); // RECHTS-knop
status

// 2. Conflictcontrole (alleen voor handmatige bewegingsknoppen)
// Deze sectie controleert of meerdere handmatige bewegingsknoppen
tegelijktijd worden ingedrukt.
// Dit wordt beschouwd als een conflict en blokkeert de beweging tot de
knoppen weer zijn losgelaten.
int manualMovementButtonCount = 0; // Teller voor het aantal ingedrukte
bewegingsknoppen.
if (currentUp) manualMovementButtonCount++;
if (currentDown) manualMovementButtonCount++;
if (currentLeft) manualMovementButtonCount++;
if (currentRight) manualMovementButtonCount++;

// conflict is waar als meer dan één handmatige bewegingsknop tegelijk is
ingedrukt.
bool conflict = (manualMovementButtonCount > 1);
// Print het conflictbericht slechts één keer wanneer het conflict begint.
if (conflict && !lastConflictState) {
    Serial.println("Conflict: meerdere handmatige bewegingsknoppen ingedrukt");
}
lastConflictState = conflict; // Update de vorige conflictstatus.

// 3. LED- en motor-stop logica (hoogste prioriteit)
// Deze sectie controleert de algemene veiligheidsstatus en activeert de LED
en motoren stop.
// De LED brandt als de stop actief is, de behuizing open is (onveilig), of er
een conflict is.

```

```

if (stopActive || caseClosed || conflict) {
    digitalWrite(LED_PIN, HIGH); // Zet de LED AAN.
    stopAllMotors(); // Stopt alle motoren onmiddellijk.
} else {
    digitalWrite(LED_PIN, LOW); // Zet de LED UIT als er geen actieve stop,
    onveilige behuizing of conflict is.
}

// 4. Hoofdlogica op basis van systeemstatus (alleen als niet gestopt of in
// conflict)
// Alle motorbesturingslogica wordt alleen uitgevoerd als er geen actieve
// stop, geen onveilige behuizing en geen handmatig conflict is.
if (!stopActive && !caseClosed && !conflict) {
    // Handmatige besturing heeft voorrang als richtingknoppen worden ingedrukt.
    if (currentUp || currentDown || currentLeft || currentRight) {
        // Controleert welke richtingknop is ingedrukt en start de continue
        beweging.
        if (currentUp) {
            // Beweegt beide Z-assen omhoog. (+1 is de richting voor omhoog)
            moveZ_AS(stepperZ1, stepperZ2, 1);
        } else if (currentDown) {
            // Beweegt beide Z-assen omlaag. (-1 is de richting voor omlaag)
            moveZ_AS(stepperZ1, stepperZ2, -1);
        } else if (currentLeft) {
            // Beweegt de X-as naar links. (+1 is de richting voor links)
            moveX_AS(stepperX, 1);
        } else if (currentRight) {
            // Beweegt de X-as naar rechts. (-1 is de richting voor rechts)
            moveX_AS(stepperX, -1);
        }
        currentSystemState = MANUAL_CONTROL; // Zet de systeemstatus naar
        handmatige bediening.
    } else { // Geen handmatige knop ingedrukt.
        // Als het systeem eerder in handmatige besturing was, stopt het nu en
        gaat het naar IDLE.
        if (currentSystemState == MANUAL_CONTROL) {
            stopMotion(); // Stopt de motoren (vertraging naar 0
            snelheid).
            currentSystemState = IDLE; // Zet de systeemstatus terug naar IDLE.
        }
    }
}

// 5. Enkele afdruk voor bewegingsknoppen, rekening houdend met systeemstatus
// Deze sectie print berichten naar de seriële monitor wanneer een handmatige
// bewegingsknop wordt ingedrukt.
// Het onderscheidt of de beweging is toegestaan of geblokkeerd door
// veiligheidsredenen.
if (currentUp && !lastUpButtonState) {
    if (stopActive || caseClosed || conflict) {
        Serial.println("Kan niet bewegen (Omhoog): Stop actief, behuizing
        onveilig, of conflict.");
    } else {
        Serial.println("Beweeg omhoog");
    }
}
if (currentDown && !lastDownButtonState) {
    if (stopActive || caseClosed || conflict) {
        Serial.println("Kan niet bewegen (Omlaag): Stop actief, behuizing
        onveilig, of conflict.");
    } else {

```

```

        Serial.println("Beweeg omlaag");
    }
}
if (currentLeft && !lastLeftButtonState) {
    if (stopActive || caseClosed || conflict) {
        Serial.println("Kan niet bewegen (Links): Stop actief, behuizing onveilig,
of conflict.");
    } else {
        Serial.println("Beweeg links");
    }
}
if (currentRight && !lastRightButtonState) {
    if (stopActive || caseClosed || conflict) {
        Serial.println("Kan niet bewegen (Rechts): Stop actief, behuizing
onveilig, of conflict.");
    } else {
        Serial.println("Beweeg rechts");
    }
}

// Update de last-staat voor de volgende loop.
// Dit is essentieel voor de single print in de seriële monitor.
lastUpButtonState = currentUp;
lastDownButtonState = currentDown;
lastLeftButtonState = currentLeft;
lastRightButtonState = currentRight;

// 6. Altijd run() de steppers om beweging of vertraging mogelijk te maken
// De run() functie van AccelStepper moet continu worden aangeroepen in de
loop()
// om de motoren te laten bewegen, accelereren, vertragen en stoppen.
stepperX.run();
stepperZ1.run();
stepperZ2.run();
}

//Beweegt één stappenmotor continu in een opgegeven richting.
//Deze functie is specifiek voor de X-as motor.
//'m' Is een referentie naar het AccelStepper object van de motor die moet
bewegen.
//'dir' Is de bewegingsrichting (1 voor positief, -1 voor negatief).
void moveX_AS(AccelStepper &m, int dir) {
    // Gebruikt de juiste maximale snelheid op basis van welk stepper-object wordt
doorgegeven.
    m.setSpeed(dir * X_MAX_SPEED); // Stelt de snelheid in voor de X-as.
}

//Beweegt twee stappenmotoren tegelijk continu in een opgegeven richting.
//Deze functie is specifiek voor de Z-assen (Z1 en Z2).
//'m1' Is een referentie naar het AccelStepper object van de eerste Z-as motor.
//'m2' Is een referentie naar het AccelStepper object van de tweede Z-as motor.
//'dir' Is de bewegingsrichting (1 voor positief, -1 voor negatief).
void moveZ_AS(AccelStepper &m1, AccelStepper &m2, int dir) {
    // Deze functie is specifiek voor de Z-assen, dus gebruikt altijd Z_MAX_SPEED.
    m1.setSpeed(dir * Z_MAX_SPEED); // Stelt de snelheid in voor de eerste Z-as
motor.
    m2.setSpeed(dir * Z_MAX_SPEED); // Stelt de snelheid in voor de tweede Z-as
motor.
}

//Stopt alle stappenmotoren onmiddellijk.

```

```
//Deze functie roept de stop() methode van AccelStepper aan, die de motor  
onmiddellijk stopt.  
void stopAllMotors() {  
  stepperX.stop();  
  stepperZ1.stop();  
  stepperZ2.stop();  
}  
  
//Zet de snelheid van alle motoren op nul.  
//Deze functie zet de snelheid op 0, waardoor de AccelStepper bibliotheek de  
motoren geleidelijk laat vertragen tot stilstand.  
void stopMotion() {  
  stepperX.setSpeed(0);  
  stepperZ1.setSpeed(0);  
  stepperZ2.setSpeed(0);  
}
```

#### 4.4.8 Bestand code



Arduino\_Code\_Gip\_K  
obe\_Nuyts\_6TW.ino

## 5 Ondervonden problemen

Bij het maken van mijn eindwerk ben ik heel wat problemen tegengekomen. Hieronder zal ik beschrijven welke problemen ik allemaal heb ondervonden, hoe ik ze heb opgelost en welke lessen ik er uit heb kunnen leren.

### 5.1.1.1 *Vertraagde levering van onderdelen:*

Door onduidelijke communicatie met mijn leerkracht en doordat de schoolrekeningen tijdelijk geblokkeerd werden door de stad Mortsel, kwam de betaling niet tijdig door. Hierdoor kwamen mijn onderdelen pas zeer laat aan.

- Geleerde les: ik moet voortaan duidelijk en tijdig communiceren met mijn leerkracht en concreet aangeven welke onderdelen ik nodig heb om leveringsproblemen te voorkomen.

### 5.1.1.2 *Onstabiele plankjes voor de Z-as:*

De oorspronkelijke plankjes van 4 mm hout bleken niet stijf genoeg om de Z-as stabiel te houden.

- Oplossing: ik heb de houten plankjes vervangen door plankjes van 8 mm. Deze plankjes heb ik ge3D-print uit PETG, omdat PETG een veel stijver en sterker materiaal is voor dit doel.
- Geleerde les: materiaalkeuze is cruciaal; vooraf testen of berekenen van de benodigde stijfheid kan toekomstige aanpassingen besparen.

### 5.1.1.3 *Defecte voltage regulator (LM2596):*

Plots werkte de spanningsregelaar (van 12 V → 9 V voor de Arduino Mega) niet meer. Na het nameten van het circuit bleek één van de condensatoren beschadigd en één van de pinnen losgekomen.

- Oplossing: ik heb een reservecapacitor met dezelfde specificaties gevonden en deze vervangen en erop gesoldeerd. De spanningsregelaar functioneerde nadien weer naar normaal

### 5.1.1.4 *Kapotte TB6600-stepperdriver:*

Door een foutieve aansluiting van de voedingskabels op de TB6600 ontstond een kortsluiting, waardoor de TB6600-driver beschadigd raakte.

- Oplossing: ik heb alle kabels opnieuw aangesloten, gebruik gemaakt van kabeleindhulzen voor al mijn kabels én zekeringen geplaatst tussen iedere driver en de Arduino Mega.
- Geleerde les: goed informeren over de onderdelen, zorgvuldige bekabeling, beschermende kabels en zekeringen voorkomen fouten.

### 5.1.1.5 Zwevende pinnen door elektrische ruis:

Tijdens het testen van de eindstops merkte ik iets vreemds. De seriële monitor gaf constant meldingen van mijn eindstops die "geactiveerd" en "gedeactiveerd" werden, alsof ze de hele tijd aan en uit sprongen. Dit gebeurde zomaar, zonder dat ik de schakelaars aanraakte. Ik ontdekte al snel dat dit probleem begon zodra ik mijn externe PC-voeding aanzette.

Dit fenomeen heet een "zwevende pin". Het betekent dat de draad naar de eindstop als een kleine antenne werkt en elektrische ruis opvangt. Deze ruis komt van de PC-voeding, die dicht bij mijn elektronica zit. De ruis is eigenlijk een soort ongewenste, snelle schommelingen in de stroom. Zelfs met de ingebouwde beveiliging van de Arduino (pull-up weerstanden) was de ruis te sterk. De Arduino dacht daardoor de hele tijd dat de eindstop snel werd in en uitgedrukt.

```
X-Max Eindstop: GEDEACTIVEERD (losgelaten)
X-Max Eindstop: GEACTIVEERD (ingedrukt)
X-Max Eindstop: GEDEACTIVEERD (losgelaten)
X-Max Eindstop: GEACTIVEERD (ingedrukt)
X-Max Eindstop: GEDEACTIVEERD (losgelaten)
X-Max Eindstop: GEACTIVEERD (ingedrukt)
X-Max Eindstop: GEDEACTIVEERD (losgelaten)
X-Max Eindstop: GEACTIVEERD (ingedrukt)
X-Max Eindstop: GEDEACTIVEERD (losgelaten)
X-Max Eindstop: GEACTIVEERD (ingedrukt)
X-Max Eindstop: GEDEACTIVEERD (losgelaten)
X-Max Eindstop: GEACTIVEERD (ingedrukt)
X-Max Eindstop: GEDEACTIVEERD (losgelaten)
```

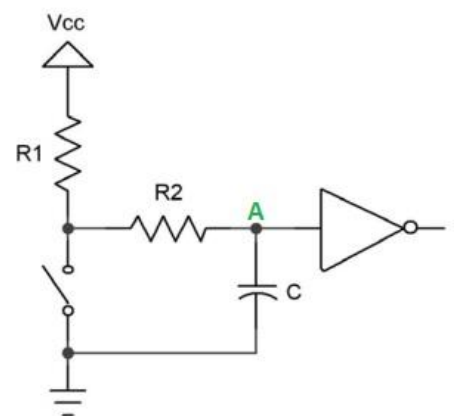
Figuur 54: Probleem in de seriële monitor

- **Oplossing:** om dit op te lossen, heb ik eerst voor elke eindstop geprobeerd om externe pull-up weerstanden van 10k toe te voegen tussen de pinnen van de eindstops en de +5V van de Arduino maar dit verhielp het probleem niet. Nadien heb ik geprobeerd kleinere weerstanden van 4.7 kOhm toe te voegen maar ook dit veranderde het probleem niet...

Na verder onderzoek kwam ik uit op een RC-filter (resistor-condensator filter). Deze filter zorgt ervoor dat de ruis wordt weg gefilterd en dat de eindstop alleen reageert als hij echt wordt ingedrukt.

De RC-filter bestaat uit twee belangrijke onderdelen:

1. **De pull-up weerstand (R1):** Deze weerstand verbind ik tussen een kant van mijn eindschakelaar en de +5V aansluiting van de Arduino. De weerstand zorgt ervoor dat de pin stevig op "hoog" (5V) blijft wanneer de eindstop niet is ingedrukt.
2. **De keramische condensator (C):** Deze condensator heb ik parallel over de eindstop-schakelaar aangesloten (dus tussen de Arduino-pin en de ground). In mijn circuit zit hier ook nog een 1 kOhm weerstand (R2) tussen een kant van mijn eindschakelaar en de condensator, die samen met de condensator en de andere pull-up weerstand de ruis nog beter filtert. De condensator werkt als een soort kleine batterij die snelle ruispieken opvangt en afvlakt. Hierdoor krijgt de Arduino een veel "schoner" signaal.



Figuur 55: Een RC debouncer

De andere pinnen van de eindstops heb ik ook rechtstreeks met de gemeenschappelijke ground verbonden.



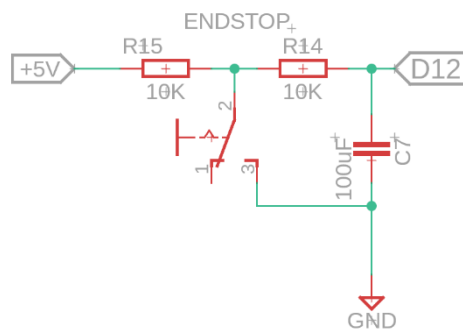
➤ Hoe een RC-filter werkt:

De weerstand en de condensator werken samen om een "tijdsfilter" te maken. De tijd die de filter nodig heeft om te reageren ( $\tau$ ) is de weerstand (R) keer de condensator (C):  $\tau = R \times C$  Dit betekent dat de filter heel snel reageert op echte schakelacties, maar te traag is om de supersnelle ruispieken door te laten. De ruis wordt als het ware "gladgestreken".

Ik heb de RC-filter eerst geprobeerd met een 100 $\mu$ F (microfarad) condensator en 10kOhm weerstanden. Dit verhielp inderdaad het probleem met de zwevende pin, maar ik merkte dat de reactietijd van de eindstop hierdoor trager werd en het bij de werking van de motoren terug begon te zweven. Dit was dus nog niet ideaal. Dit kan ik ook zien als ik mijn tijdsconstante van dit circuit bereken:

$$\tau = 100000\Omega \times 0,0001F = 1 \text{ second.}$$

Het betekent dus dat de filter 1 seconden nodig heeft om te reageren en een signaal terug te sturen.

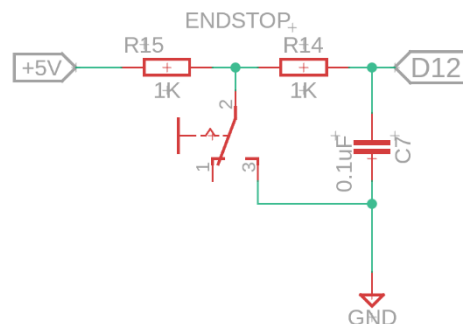


*Figuur 56: Eerste circuit*

Daarom heb ik een verder geoptimaliseerd circuit gemaakt met een 0.1  $\mu$ F condensator en 1 kOhm weerstanden (R1 en R2). Deze combinatie bleek perfect te werken! De 1 kOhm weerstanden zijn sterker en trekken de pin nog stabiel naar 5V, waardoor de ruis geen kans meer krijgt. De 0.1  $\mu$ F condensator is groot genoeg om de ruis te filteren, maar klein genoeg om de reactietijd van de eindstop razendsnel te houden. Voor mijn gekozen onderdelen is de tijdsconstante:

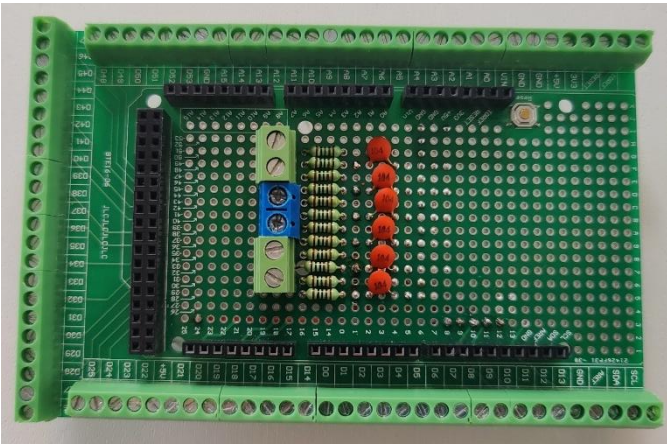
$$\tau = 1000\Omega \times 0,0000001F = 0,0001 \text{ seconden}$$

ofwel 100 microseconden. Dit betekent dat de filter veel sneller reageert op de echte schakelacties, maar te traag is om de supersnelle ruispieken door te laten. De ruis wordt dus nog steeds goed gladgestreken.

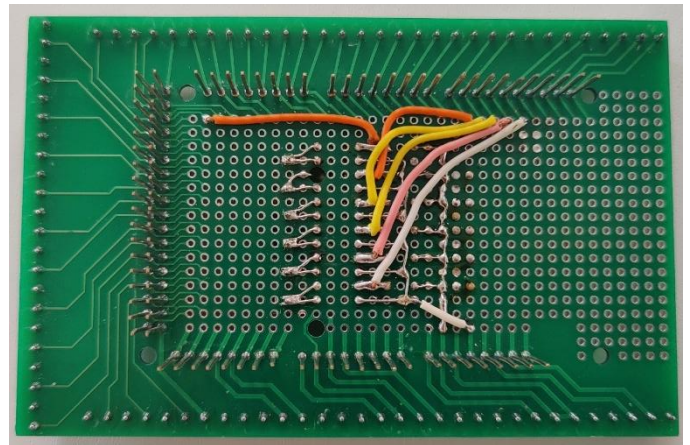


*Figuur 57: Aangepaste nieuwe circuit*

Om een zo goed mogelijke en stevige verbinding te garanderen, heb ik het hele circuit 6 keer mooi langs elkaar (voor elke eindstop één) op een breadboard gesoldeerd. Ook een kant van de knoppen worden met screw terminals verbonden en de andere kant direct aan de gemeenschappelijke ground, wat zorgt voor een stevige en veilige aansluiting.



*Figuur 58: Bovenaanzicht gesoldeerde circuit*



*Figuur 59: onderaanzicht gesoldeerde circuit*

#### Besluit:

Deze fout heeft me veel geleerd. Elektrische ruis is een echt probleem, vooral als je externe voedingen gebruikt die niet speciaal voor deze kleinere elektronica zijn gemaakt. De ingebouwde bescherming van de Arduino is dan vaak niet genoeg.

Door steeds te proberen en te testen met verschillende onderdelen (eerst 10 kOhm en 4.7 kOhm weerstanden, daarna verschillende condensatoren en 1 kOhm weerstanden), heb ik uiteindelijk de beste oplossing gevonden. Dit zorgt voor het goed wegfilteren van ruis en dat mijn eindstops precies en betrouwbaar werken.

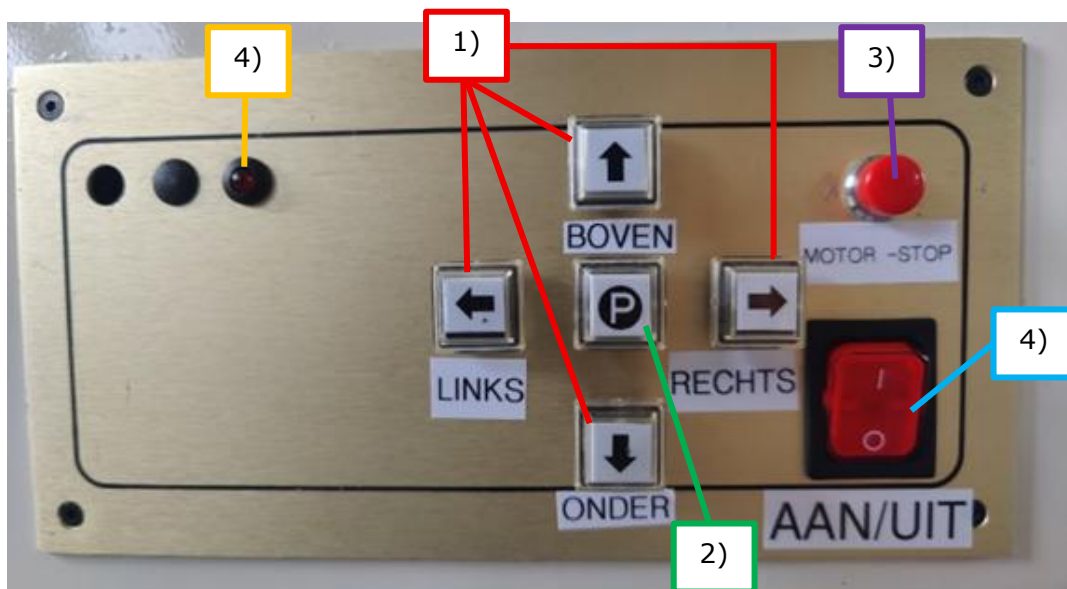
## 6 Handleiding

Welkom bij de handleiding voor de bordveger! Deze handleiding helpt u de verschillende functies van het bedieningspaneel te begrijpen en de bordveger veilig en efficiënt te gebruiken.

### 6.1 Overzicht van het Bedieningspaneel

Het bedieningspaneel is ontworpen voor de controle van de X- en Z-assen van de bordveger. Het paneel bevat de volgende knoppen en indicatoren:

- 1) **Richting knoppen (Pijlen):** Voor handmatige beweging van de veger.
- 2) **Patroon-knop:** Voor het starten van een voorgeprogrammeerd patroon.
- 3) **MOTOR-STOP knop (Rood):** Voor het direct stoppen van alle bewegingen.
- 4) **AAN/UIT-schakelaar:** De hoofdschakelaar voor de stroomtoevoer.
- 5) **LED-indicator (Rood):** Geeft de status van het systeem aan (fout, stop, openstaande behuizing).



Figuur 60: Bedieningspaneel bordveger

### 6.2 Belangrijke Veiligheidsinformatie

Voordat u de bordveger bedient, is het cruciaal om de volgende veiligheidsaspecten te begrijpen:

- **Behuizing Veiligheid (case close schakelaar):** De case close schakelaar in de kast dient als een veiligheidsschakelaar voor de behuizing.
  - Wanneer de schakelkast open staat, zal de rode LED branden en zijn alle motorbewegingen geblokkeerd. Dit is een belangrijke veiligheidsmaatregel om te voorkomen dat de veger beweegt wanneer de behuizing open is of niet correct gesloten.
  - Zorg er altijd voor dat de kast volledig en goed gesloten is (met de blauwe schroef) voordat u de veger probeert te bewegen. De rode LED moet dan uit zijn.



Figuur 61: (blauwe) Sluitschroef schakelkast

- **MOTOR-STOP knop:** Deze rode knop is een soort noodstop. Druk deze in om alle motorbewegingen onmiddellijk te stoppen. De stroom naar de motoren wordt hierbij niet onderbroken, waardoor de motoren hun positie behouden en niet oncontroleerbaar naar beneden vallen. Gebruik deze knop in geval van gevaar of ongewenst gedrag. De rode LED zal dan branden. Druk nogmaals om de stop op te heffen.
- **Conflict Handmatige Besturing:** Druk nooit meerdere richtingsknoppen tegelijk in (bijv. BOVEN en LINKS). Dit zal een conflict veroorzaken, de rode LED laten branden en alle bewegingen blokkeren totdat alle knoppen zijn losgelaten.

## 6.3 Bediening van de Bordveger

### 6.3.1 Handmatige Besturing

Gebruik de pijlknoppen voor directe controle over de beweging van de veger:

- **BOVEN (Pijl omhoog):** Beweegt de veger omhoog (Z-as).
  - **Eindstopbeveiliging:** De veger stopt automatisch als hij de bovenste eindstops (links en rechts boven) bereikt. Hij kan dan alleen nog omlaag bewegen.
- **ONDER (Pijl omlaag):** Beweegt de veger omlaag (Z-as).
  - **Eindstopbeveiliging:** De veger stopt automatisch als hij de onderste eindstops (links en rechts onder) bereikt. Hij kan dan alleen nog omhoog bewegen.
- **LINKS (Pijl naar links):** Beweegt de veger naar links (X-as).
  - **Eindstopbeveiliging:** De veger stopt automatisch als hij de linker eindstop bereikt. Hij kan dan alleen nog naar rechts bewegen.
- **RECHTS (Pijl naar rechts):** Beweegt de veger naar rechts (X-as).
  - **Eindstopbeveiliging:** De veger stopt automatisch als hij de rechter eindstop bereikt. Hij kan dan alleen nog naar links bewegen.

### 6.3.2 Geautomatiseerde Besturing

De bordveger kan een voorgeprogrammeerd patroon uitvoeren:

- **Patroon-knop (kort indrukken):** Druk kort op de Patroon-knop om het geautomatiseerde veegpatroon te starten. De veger zal een reeks bewegingen uitvoeren (omlaag, rechts, omhoog, links) om het hele bord te vegen.
  - **Voorwaarden:** Het patroon start alleen als het systeem niet gestopt is, de behuizing veilig is en er geen handmatig bewegingsconflict is.
  - **Pauzeren:** U kunt het patroon op elk moment pauzeren door de patroonknop, tijdens het patroon in werking is, in te drukken. Wanneer u de patroonknop vervolgens terug indrukt zal het patroon verder gezet worden
  - **Onderbreken:** U kunt het patroon op elk moment onderbreken door op de rode MOTOR-STOP knop te drukken.

- **Homing (Naar Startpositie):**

Het systeem kan zichzelf naar een gedefinieerde "thuispositie" (homing) bewegen met behulp van de eindstops:

- **Patroon-knop (5 seconden of langer indrukken):** Houd de Patroon-knop voor 5 seconden of langer ingedrukt om de homing-procedure te starten. De veger zal uit zichzelf eerst naar de linker X-as eindstop bewegen en vervolgens naar de onderste Z-as eindstops (links en rechts onder) om zijn nul positie te vinden.
  - **Voorwaarden:** Homing start alleen als het systeem niet gestopt is, de behuizing veilig is en er geen handmatig bewegingsconflict is.
  - **Onderbreken:** U kunt de homing-procedure op elk moment onderbreken door op de rode MOTOR-STOP knop te drukken.

## 6.4 LED-indicator

De rode LED op het bedieningspaneel geeft belangrijke statusinformatie:

- **LED AAN:** wanneer de led brandt en dus aanstaat kan dit meerdere dingen betekenen namelijk:
  - De MOTOR-STOP knop is actief.
  - De schakelkast staat open of is niet goed gesloten.
  - Er is een conflict in de handmatige besturing (meerdere richtingsknoppen tegelijk ingedrukt).
    - In al deze gevallen zijn alle motorbewegingen geblokkeerd.
- **LED UIT:** Het systeem is klaar voor gebruik, er is geen actieve stop, de behuizing is veilig en er zijn geen conflicten.

## 6.5 Probleemoplossing (Snelle Tips)

- **Veger beweegt niet:**
  - Controleer of de schakelkast open staat of niet goed gesloten is.
  - Controleer of de rode MOTOR-STOP knop ingedrukt is.
  - Zorg ervoor dat u niet meerdere richtingsknoppen tegelijk indrukt.
  - Controleer of de veger niet al tegen een eindstop aan zit in de gewenste bewegingsrichting.
- **Vreemde bewegingen of geluiden:** Druk onmiddellijk op de rode MOTOR-STOP knop en controleer alle kabels en mechanische onderdelen.

Mocht u verdere vragen hebben of problemen ondervinden die niet in deze handleiding worden beschreven, raadpleeg dan de technische documentatie van mijn boekwerk of neem contact met mij op.

## 7 Gebruikte programma's

Hier zijn de programma's die ik heb gebruikt voor mijn GIP, met een korte uitleg erbij:

- **Adobe illustrator:**  
Dit programma heb ik gebruikt om tekeningen te maken voor de laser cutter.
- **Arduino IDE:**  
Met de Arduino IDE heb ik code geschreven en geüpload naar mijn Arduino Uno. Hiermee kon ik mijn microcontroller programmeren om bijvoorbeeld mijn koppen in te lezen en motoren aan te sturen.
- **Autodesk autoCAD:**  
AutoCAD heb ik gebruikt om 3D-ontwerpen om te zetten naar 2D-tekeningen. Dit was nodig voor onderdelen correct te kunnen uitsnijden in de laser cutter.
- **Autodesk Fusion 360:**  
Dit programma heb ik gebruikt om 3D-ontwerpen te maken. Hiermee kon ik onderdelen modelleren die ik later wilde 3D-printen of laser cutten.
- **Autodesk inventor:**  
Met Inventor heb ik mijn 3D-ontwerpen voorzien van maataanduidingen. Dit was belangrijk om alles precies te laten passen en correct te laten maken.
- **Ulimaker Cura:**  
Cura heb ik gebruikt om mijn 3D-bestanden te slicen. Dit programma zet het 3D-model om naar een bestand dat de 3D-printer kan lezen, met de juiste instellingen zoals laaghoogte en printsnelheid.
- **Bambu Studio:**  
Bambu Studio heb ik gebruikt als slicer voor de Bambu Lab 3D-printer. Hiermee kon ik mijn 3D-modellen voorbereiden op het printen met specifieke instellingen zoals laaghoogte en printsnelheid.
- **Microsoft Word:**  
Word heb ik gebruikt om mijn GIP-verslag te schrijven en vorm te geven. Hierin verzamelde ik alle informatie, afbeeldingen en uitleg over mijn project.
- **Microsoft Excel:**  
Excel heb ik gebruikt voor mijn berekeningen, bestellijst, regressieanalyse en het opstellen van mijn planning.

## 8 Proces

Het maken van mijn automatische whiteboardwisser was een ingewikkeld maar leuk project. Ik zal hier uitleggen hoe ik dat stap voor stap heb aangepakt.

### 1. Het idee en de eerste tekeningen:

- Het begon met een idee voor een automatische whiteboardwisser die ik makkelijk kon gebruiken met de druk van één knop. Iets wat mijn leerkracht al lang wou.
- Toen maakte ik direct tekeningen, zoals 3D-schetsen. Zo zag ik al voor me hoe het eruit zou zien en hoe het zou werken. Ik dacht na over hoe goed het moest werken en hoe veilig het moest zijn.

### 2. Onderzoek naar inkt:

- Een best groot deel van mijn project was het uitzoeken hoe whiteboardinkt precies werkt. Ik heb gekeken uit welke stoffen het bestaat en waarom je het zo makkelijk kunt uitvegen.
- Dit hielp me om te begrijpen hoe mijn wisser het beste de inkt van het bord kan krijgen.

### 3. Opzoeken en testen:

- Voordat ik de onderdelen bestelde, heb ik veel tijd gestoken in het opzoeken en vergelijken van verschillende opties. Ik wilde zeker weten dat ik de allerbeste en meest geschikte motoren, knoppen, drivers en andere elektronica vond voor mijn project.
- Ook heb ik de onderdelen getest om te kijken of ze deden wat ik ervan verwachtte en of ze sterk genoeg waren. Dit was belangrijk om te zorgen dat de wisser goed zou werken.

### 4. Regressieanalyse (Motoren testen op temperatuur):

- Ik heb een regressieanalyse gedaan om te kijken hoe warm de motoren werden als ze lang werkte. Dit was superbelangrijk, want ik wilde niet dat ze oververhit geraakten en stuk gingen.
- Ik heb de temperaturen heel precies bijgehouden en die gegevens daarna in grafieken gezet. Zo kon ik goed zien hoe de temperatuur steeg. Op basis daarvan kon ik beslissen of ik misschien een ventilator moest toevoegen of een pauze in de code moest zetten, zodat de motoren tussendoor even konden afkoelen.

### 5. Wat moet het kunnen? En welke spullen heb ik nodig?

- Daarna heb ik precies opgeschreven wat de wisser moest kunnen. Denk aan de afmetingen, hoeveel het mocht wegen, hoe precies het moest bewegen en hoe snel.

Op basis daarvan heb ik alle onderdelen gezocht en besteld. Dat waren bijvoorbeeld de Arduino Mega (het 'brein' van het project), de motoren, allerlei knoppen en een oude PC-voeding voor de stroom.

- **Probleem en oplossing:** het duurde superlang voordat mijn spullen binnenkwamen. Dat kwam door misverstanden en met problemen met de rekeningen op school. Ik heb daaruit geleerd hoe belangrijk het is om heel duidelijk te zijn en alles op tijd te regelen.

### 6. De bouwtekeningen en voorbereidingen:

- Vervolgens heb ik precieze tekeningen gemaakt van alle onderdelen: de X-as (die van links naar rechts beweegt), de Z-as (die omhoog en omlaag gaat) en het gehele frame.



- Ik gebruikte computerprogramma's zoals Fusion 360 en Illustrator om deze tekeningen klaar te maken voor de lasersnijder (voor de houten delen) en de 3D-printer (voor de plastic onderdelen). Zo wist ik zeker dat alles perfect zou passen.
- **Probleem en oplossing:** de eerste houten planken voor de Z-assen waren niet stevig genoeg. Ze bewogen te veel. Ik heb ze toen vervangen door veel sterkere en dikkere planken uit plastic, die ik met de 3D-printer heb gemaakt. Ik heb geleerd dat het kiezen van de juiste materialen superbelangrijk is!

#### 7. Elektriciteit aansluiten en code schrijven:

- Alle elektrische onderdelen werden gemonteerd en bekabeld, precies zoals op mijn schema stond. Ik zorgde ervoor dat de kabels netjes lagen en alles goed geaard was.
- Belangrijke toevoegingen: Ik heb zekeringen toegevoegd bij de motordrivers en de Arduino. Dit is een extra beveiliging tegen te veel stroom of kortsluiting.
- Daarna schreef ik de code voor de Arduino. Deze code vertelt de wisser hoe de motoren moeten bewegen, hoe hij moet reageren op de knoppen en hoe hij veilig kan werken.
- **Probleem en oplossing:** de eindschakelaars gaven soms valse signalen door elektrische "ruis" van de voeding. Ik heb dit opgelost door het circuit aan te passen en RC-filters toe te voegen. Hierdoor werken de eindschakelaars nu betrouwbaar en zonder problemen of ruis.

#### 8. Alles aansluiten en testen:

- Nadat alles was aangesloten en de code erin stond, zette ik alles in elkaar en testte ik het de veger. Ik controleerde of alle bewegingen klopten, of de knoppen werkten en of de noodstop goed reageerde. En paste de code aan bij de stuken die niet goed werkten.

#### 9. De gebruikershandleiding maken:

Nadat mijn whiteboardwisser klaar was, vond ik het superbelangrijk dat iedereen hem makkelijk en veilig kon gebruiken. Daarom heb ik een duidelijke handleiding geschreven. Hierin staat precies hoe de wisser werkt en waar je op moet letten.

- **Uitleg van de bediening:** ik heb eerst een overzicht gemaakt van alle knoppen en wat ze doen: de pijltjestoetsen voor zelf bewegen, de 'patroonknop' voor de automatische schoonmaak of om naar de startpositie te gaan, de rode 'STOP'-knop voor noodgevallen, en de aan/uit-schakelaar.
- **Veiligheid van de veger:** in de handleiding leg ik ook de veiligheid uit. Er zit bijvoorbeeld een schakelaar op de kast die controleert als de kast open staat en de motoren blokkeert en een rood lampje laat branden. De rode 'STOP'-knop is ook heel belangrijk: die zet alles meteen stil als er iets misgaat. Ik waarschuw ook dat je nooit meerdere knoppen tegelijk moet indrukken, want dan kan er een storing komen.
- **De werking van de wisser:** ik beschrijf hoe je de wisser handmatig kunt bedienen (omhoog, omlaag, links, rechts). De wisser stopt trouwens automatisch bij de randen van het bord. Ook leg ik uit hoe je de automatische schoonmaak start met de 'patroonknop' en hoe je de wisser naar de startpositie stuurt door die knop langer in te houden. Het rode LED-lampje geeft aan of alles in orde is of dat er een storing is. Tot slot staan er 'snelle tips' in voor als er problemen zijn.



## 10. Het Eindwerk Boekwerk (Mijn GIP-document):

Naast het bouwen van de wisser zelf, is het belangrijk om alles wat ik gedaan en geleerd heb goed vast te leggen. Dat heb ik gedaan in een compleet verslag, mijn GIP-document. Dit is het 'boekwerk' van mijn hele project.

- **Inhoud:** in dit document leg ik stap voor stap uit hoe ik het project heb aangepakt. Ik beschrijf mijn hele proces gedetailleerd, van het eerste idee tot de laatste tests en oplossingen voor problemen. Elk onderdeel, elke keuze en elke test die ik heb gedaan, staat erin uitgelegd.
- **Programma's en bronnen:** ik heb ook een lijst opgenomen van alle computerprogramma's die ik heb gebruikt. Denk aan programma's voor 3D-tekeningen (zoals Fusion 360 en Inventor), programma's om tekeningen voor de lasersnijder te maken (zoals Adobe Illustrator en AutoCAD) en programma's om de code voor de Arduino te schrijven (Arduino IDE) en de 3D-printer aan te sturen (Ultimaker Cura). Uiteraard heb ik ook Word gebruikt om het hele document te schrijven. En natuurlijk staat er een lijst in van alle bronnen (websites, boeken, artikelen) die ik heb gebruikt voor mijn onderzoek, bijvoorbeeld over de chemie van inkt of over hoe motoren werken. Zo kan iedereen precies zien hoe het project is opgebouwd en waar alle informatie vandaan komt.

## 9 Besluit

Voor deze Geïntegreerde Proef (GIP) was het de bedoeling om een automatische whiteboard-veger te ontwerpen en te maken. Dit project bestond uit twee belangrijke delen: grondig onderzoek en het schrijven van het bijbehorende verslag en het daadwerkelijk bouwen van de whiteboard-veger.

Het schrijven van het verslag was een grote klus die veel tijd en inzet vroeg. Voordat ik kon beginnen met schrijven, moest ik veel informatie opzoeken over verschillende onderwerpen. Het wetenschappelijke deel ging over de chemie van inkt, vooral whiteboard-inkten. Ik heb de geschiedenis, de samenstelling en hoe inkt op moleculair niveau werkt, uitgebreid beschreven. Door dit onderzoek heb ik ondervonden wat het verschil is tussen gewone alcoholstiften en whiteboardstiften, vooral hoe ze droog en zonder sporen kunnen worden weggeveegd.

Dit deel bevatte ook een analyse van de motortemperatuur, waarbij ik keek naar het verband tussen de tijd dat de motor draaide en de temperatuur. De resultaten lieten zien dat de temperatuur in de eerste 15 minuten duidelijk lineair steeg. Dit is belangrijk om te weten voor de betrouwbaarheid en levensduur van de motoren. De berekende correlatiecoëfficiënt van 0,9972 bevestigde een zeer sterk verband tussen tijd en temperatuur, wat betekent dat we goed kunnen voorspellen hoe warm de motor wordt.

Hoewel het theoretische deel van het technisch verslag minder werk kostte omdat ik al veel wist over de gebruikte onderdelen en hoe ze werken, en er online meer informatie te vinden is over technische onderdelen dan over de chemische kant van inkt, was het fysieke bouwen van de technische aspecten juist een grotere uitdaging. In dit deel beschreef ik de keuze van onderdelen zoals de Arduino Mega, TB6600 stappenmotor drivers en de specifieke stappenmotoren. Ook de elektrische schema's, de bedrading en de opbouw van de schakelkast werden uitgebreid behandeld.

Het bouwen van de whiteboard-veger was net zo'n grote uitdaging als het schrijven van het verslag. Het proces begon met het gedetailleerd uitwerken van alle onderdelen en hoe ze samenwerken. Voor het elektrische deel werden eerst berekeningen gedaan en schema's getekend. Vervolgens werden alle elektronische onderdelen mooi in mijn schakelkast gemonteerd en met elkaar verbonden, met gebruik van DIN-rails en kabelgoten om het allemaal proper en stevig vast te zetten.

Tijdens het bouwen van de whiteboard-veger zijn er veel problemen voor gekomen. Een van de problemen was de onstabiele mechanische constructie van de Z-as, doordat de oorspronkelijke houten plankjes van 4 mm niet stijf genoeg bleken. Dit is gelukkig snel opgelost door ze te vervangen door stevigere, 3D-geprinte PETG-onderdelen van 8 mm. Op elektrisch gebied ondervond ik ook problemen, zoals een defecte LM2596 spanningsregelaar, die ik hersteld heb door de beschadigde condensator te vervangen. Ook raakte een TB6600-stepperdriver kapot door een slechte aansluiting: dit is opgelost door alle kabels opnieuw en netjes aan te sluiten, kabeleindhulzen te gebruiken en zekeringen te plaatsen. Verder waren er problemen met zwevende pinnen bij de eindstops door elektrische ruis van mijn voeding, wat verholpen is met een RC-filter voor elke eindstop. Deze uitdagingen zijn tijdens de bouw aangepakt en opgelost.

De functionele eisen stelden dat de veger een whiteboard van maximaal 1,2 meter hoog en 1,5 meter lang moest kunnen vegen, met een precisie van 1 mm en een maximale snelheid van 500 mm/s. De afmetingen en nauwkeurigheidseisen zijn behaald. De aanwezigheid van zes eindstops en een noodstop draagt bij aan de veiligheid en functionaliteit.

Op basis van wat ik heb geleerd en bereikt, zijn er nog verschillende manieren om de whiteboard-veger te verbeteren en uit te breiden. Om te zorgen dat de motoren niet te warm worden, zou ik een temperatuursensor kunnen toevoegen. Deze kan de temperatuur meten en waarschuwen of de veger stoppen als het te warm wordt (bijvoorbeeld bij 56 °C). Ik zou ook

ventilatie of koellichamen kunnen plaatsen om de motoren beter te koelen. Verder kan ik de software aanpassen zodat de veger automatisch pauzes neemt of minder stroom gebruikt als hij stilstaat, om oververhitting te voorkomen. Tenslotte zou het gebruik van andere materialen, zoals metaal, en het toevoegen van ventilatiegaten in de behuizing de veger nog sterker en efficiënter kunnen maken.

Deze GIP heeft me enorm veel geleerd. Ik heb onder andere ondervonden hoe belangrijk duidelijke en tijdige communicatie is bij het regelen van onderdelen. Ook heb ik ervaren dat de materiaalkeuze cruciaal is, aangezien de oorspronkelijke plankjes voor de Z-as niet stijf genoeg bleken. Ik heb geleerd over het zorgvuldig bekabelen en het plaatsen van zekeringen om schade aan onderdelen te voorkomen, zoals bij de TB6600-stepperdriver. Een belangrijke les was ook het omgaan met elektrische ruis en het effectief toepassen van RC-filters om zwevende pinnen bij de eindstops te verhelpen, wat de betrouwbaarheid van het systeem aanzienlijk verbeterde. Hoewel er veel uitdagingen waren, ben ik erg tevreden met het eindresultaat. De ervaring heeft mijn technische vaardigheden aanzienlijk verbeterd en heeft me inzicht gegeven in het belang van zorgvuldige planning en het oplossen van problemen. Ik ben trots op wat ik heb bereikt en zie dit project als een waardevolle voorbereiding op mijn toekomstige studies.

# 10 Dankwoord

## 10.1 Nederlands

Het maken van deze Geïntegreerde Proef, de automatische whiteboard-veger, was een groot en leerzaam project. Dit project was niet gelukt zonder de belangrijke hulp en begeleiding van veel mensen. Ik wil iedereen die heeft geholpen dan ook hartelijk bedanken.

Ik wil in het bijzonder mijn leerkrachten bedanken voor hun kennis, geduld en alle uren begeleiding:

- **Meneer S. Van den Bosch**, mijn leerkracht elektriciteit en elektronica, voor al zijn hulp bij de elektrische delen van dit project, het zoeken en bestellen van de onderdelen.
- **Meneer Y. Verbruggen**, mijn leerkracht mechanica en begeleider van deze GIP, voor al zijn hulp bij het mechanische deel en het altijd zorgen voor de juiste schroeven.
- **Mevrouw C. Bakkaus**, mijn leerkracht chemie en wiskunde, voor haar deskundige hulp bij het chemische deel en de regressieanalyse, die heel belangrijk waren voor de wetenschappelijke basis van mijn werk.
- **Meneer T. Goethals**, voor zijn waardevolle hulp bij de taal en opbouw van mijn verslag en zijn constante steun en tips gedurende het hele schooljaar.
- **Mevrouw A. Jacobs**, voor het lasersnijden, wat heel belangrijk was voor de nauwkeurigheid van de onderdelen.

Daarnaast wil ik ook zeker de volgende personen in het bijzonder bedanken:

- **Meneer T. Cornelissens**, voor zijn hulp en ideeën bij het begin van het eerste ontwerp van de bordveger.
- De **GIP-jury**, voor hun nuttige opmerkingen, goede tips en de aanpassingen die hieruit voortkwamen, waardoor de kwaliteit van mijn project veel beter werd.
- **Meneer M. Spaepen**, voor het mogelijk maken van langere werktijden op school en het openen van de school op een vrije dag, wat heel belangrijk was voor de afwerking van mijn project.
- **Meneer D. Vaes**, voor het regelen van de houten planken die gebruikt zijn voor het lasersnijden.
- Mijn **klasgenoten en meerden leerlingen**, en in het bijzonder Szymon uit 5MC en Jan uit 5TW, voor hun hulp en tips bij het gebruik van de 3D-printers.

Tot slot wil ik mijn ouders bedanken voor hun steun, geduld en aanmoediging gedurende dit hele project. Ook mijn kleine broertje wil ik speciaal bedanken voor zijn hulp bij het maken van de schakelkast in de weekends en na school. Mijn oudere zus wil ik bedanken voor haar deskundig advies over het chemisch gedeelte.

Dankzij jullie allemaal is dit project succesvol afgerond.

## 10.2 Frans

La réalisation de ce projet intégré, le racleur automatique pour tableau blanc, a été une expérience enrichissante. Ce projet n'aurait pas abouti sans l'aide et les conseils précieux de nombreuses personnes. Je tiens donc à remercier chaleureusement tous ceux qui m'ont aidé. Je tiens tout particulièrement à remercier mes professeurs pour leurs connaissances, leur patience et toutes les heures qu'ils m'ont consacrées :

- **Monsieur S. Van den Bosch**, mon professeur d'électricité et d'électronique, pour toute son aide dans les parties électriques de ce projet, la recherche et la commande des pièces.
- **Monsieur Y. Verbruggen**, mon professeur de mécanique et accompagnateur de ce GIP, pour toute son aide dans la partie mécanique et pour avoir toujours fourni les vis appropriées.
- **Madame C. Bakkaus**, mon professeur de chimie et de mathématiques, pour son aide experte dans la partie chimique et l'analyse de régression, qui ont été très importantes pour la base scientifique de mon travail.
- **Monsieur T. Goethals**, pour son aide précieuse dans la rédaction et la structure de mon rapport, ainsi que pour son soutien constant et ses conseils tout au long de l'année scolaire.
- **Madame A. Jacobs**, pour la découpe au laser, qui a été très importante pour la précision des pièces.

Je tiens également à remercier tout particulièrement les personnes suivantes :

- **Monsieur T. Cornelissens**, pour son aide et ses idées au début de la conception initiale du balai.
- **Le jury GIP**, pour ses commentaires utiles, ses bons conseils et les modifications qui en ont résulté, qui ont permis d'améliorer considérablement la qualité de mon projet.
- **Monsieur M. Spaepen**, pour avoir permis de prolonger les heures de travail à l'école et d'ouvrir l'école pendant un jour de congé, ce qui a été très important pour la finalisation de mon projet.
- **Monsieur D. Vaes**, pour avoir fourni les planches de bois utilisées pour la découpe au laser.
- **Mes camarades de classe et plusieurs élèves**, en particulier Szymon de 5MC et Jan de 5TW, pour leur aide et leurs conseils sur l'utilisation des imprimantes 3D.

Enfin, je tiens à remercier mes parents pour leur soutien, leur patience et leurs encouragements tout au long de ce projet. Je tiens également à remercier tout particulièrement mon petit frère pour son aide dans la fabrication du boîtier de commande pendant les week-ends et après l'école. Je remercie ma grande sœur pour ses conseils avisés sur la partie chimique.

Grâce à vous tous, ce projet a été mené à bien.

## Engels

Creating this thesis, the automatic whiteboard eraser, was a major and educational project. This project would not have been possible without the significant help and guidance of many people. I would therefore like to express my sincere thanks to everyone who helped. I would particularly like to thank my teachers for their knowledge, patience and all the hours of guidance:

- **Mr S. Van den Bosch**, my electricity and electronics teacher, for all his help with the electrical parts of this project, finding and ordering the parts.
- **Mr Y. Verbruggen**, my mechanics teacher and supervisor of this GIP, for all his help with the mechanical part and always providing the right screws.
- **Ms C. Bakkaus**, my chemistry and maths teacher, for her expert help with the chemical part and the regression analysis, which were very important for the scientific basis of my work.
- **Mr T. Goethals**, for his valuable help with the language and structure of my report, and his constant support and tips throughout the school year.
- **Ms A. Jacobs**, for the laser cutting, which was very important for the accuracy of the parts.

In addition, I would also like to thank the following people in particular:

- **Mr T. Cornelissens**, for his help and ideas at the start of the first design of the board wiper.
- **The GIP jury**, for their useful comments, good tips and the resulting adjustments, which greatly improved the quality of my project.
- **Mr M. Spaepen**, for allowing me to work longer hours at school and opening the school on a day off, which was very important for the completion of my project.
- **Mr D. Vaes**, for arranging the wooden planks used for laser cutting.
- **My classmates and other students**, and in particular Szymon from 5MC and Jan from 5TW, for their help and tips on using the 3D printers.

Finally, I would like to thank my parents for their support, patience and encouragement throughout this project. I would also like to give special thanks to my little brother for his help in making the switch box on weekends and after school. I would like to thank my older sister for her expert advice on the chemical part.

Thanks to all of you, this project has been successfully completed.

# 11 Bronnen

---

*Geschiedenis van inkt.* (2020, Oktober 9). IsGeschiedenis.  
<https://isgeschiedenis.nl/nieuws/geschiedenis-van-inkt>

---

Colordo, D. (2024, October 18). Geschiedenis en vervaardiging van Chinese inkt: kunst en wetenschap. *Cultura 10*. <https://nl.cultura10.com/de-geschiedenis-van-Chinese-inkt/>

---

Wikipedia-bijdragers. (2024, September 16). *Inkt*. Wikipedia.  
<https://nl.wikipedia.org/wiki/Inkt>

---

Pad, O. (2020, May 26). *Geschiedenis van inkt*. Avondtuur!  
<https://www.avondtuur.nl/geschiedenis-van-inkt/>

---

*NEMA17-04 Bipolar Stepper Motor.* (2021). [https://joy-it.net/files/files/Produkte/NEMA17-04/NEMA17-04\\_Datasheet.pdf](https://joy-it.net/files/files/Produkte/NEMA17-04/NEMA17-04_Datasheet.pdf)

---

De Ledesma, A. P., & De Ledesma, A. P. (2024, 7 mei). *Whiteboard Markers: A Deep Dive into Ink Chemistry and Performance | My Beautiful Adventures*. My Beautiful Adventures |.  
<https://mybeautifuladventures.com/2024/05/07/whiteboard-markers-a-deep-dive-into-ink-chemistry-and-performance/>

---

NOBO Whiteboard Marker Ink. (z.d.). In Esselte BV, *VEILIGHEIDSINFORMATIEBLAD* (pp. 1–8).  
[https://www.123inkt.nl/pdf/Nobo\\_Whiteboard\\_markers\\_NL.pdf](https://www.123inkt.nl/pdf/Nobo_Whiteboard_markers_NL.pdf)

---

Wikipedia-bijdragers. (2024, 22 December). *Ethanol*. Wikipedia.  
<https://nl.wikipedia.org/wiki/Ethanol>

---

*Illustrated Glossary of Organic Chemistry - Ethanol.* (z.d.).  
<https://www.chem.ucla.edu/~harding/IGOC/E/ethanol.html>

---

National Geographic. (2016, 17 November). *What's in Dry-Erase Markers? | Ingredients with George Zaidan (Episode 10)* [Video]. YouTube.  
<https://www.youtube.com/watch?v=xKsi3i4zTfs>

---

Wikipedia-bijdragers. (2024a, november 25). *Hybridisatie (scheikunde)*. Wikipedia.  
[https://nl.wikipedia.org/wiki/Hybridisatie\\_\(scheikunde\)](https://nl.wikipedia.org/wiki/Hybridisatie_(scheikunde))

---

詹忆源, 张杏金, & Ltd, S. S. N. S. C. (2006, 22 November). *CN1962772A - Ink for white board marker and method for manufacturing same - Google Patents*.  
<https://patents.google.com/patent/CN1962772A/en>

---

Wikipedia contributors. (2024, 25 september). *Polyvinyl acetate*. Wikipedia.  
[https://en.wikipedia.org/wiki/Polyvinyl\\_acetate](https://en.wikipedia.org/wiki/Polyvinyl_acetate)

---

TB6600 Stepper Motor Driver – Maker Hardware. (z.d.).  
<https://makerhardware.net/knowledge-base/electronics/tb6600-stepper-motor-driver/>

---

Limited, A. (z.d.). *TB6600 Stepper Motor Controller: Pinout, Arduino Interface, and How It Works*. [https://www.allelcoelec.com/blog/TB6600-Stepper-Motor-Controller-Pinout-Arduino-Interface-and-How-It-Works.html?srsId=AfmBOopwIXhVXmQ4\\_NgHW9EGqBbVU4uao153XtSuyS7RLkAWharj0xHM#5.%20H-Bridge%20Circuit%20Configuration%20for%20the%20TB6600](https://www.allelcoelec.com/blog/TB6600-Stepper-Motor-Controller-Pinout-Arduino-Interface-and-How-It-Works.html?srsId=AfmBOopwIXhVXmQ4_NgHW9EGqBbVU4uao153XtSuyS7RLkAWharj0xHM#5.%20H-Bridge%20Circuit%20Configuration%20for%20the%20TB6600)

---

The Engineering Mindset. (2021, 26 juni). *How Stepper Motors Work - Electric motor* [Video]. YouTube. <https://www.youtube.com/watch?v=09Mpkjcr0bo>

---

Evans, P. (2022, 23 januari). *Hybrid Stepper Motor Basics*. The Engineering Mindset.  
<https://theengineeringmindset.com/hybrid-stepper-motor-basics/>

---

The Engineering Mindset. (2021b, december 6). *Hybrid Stepper Motor Basics* [Video]. YouTube. <https://www.youtube.com/watch?v=Da63f16KKDo>

---

Ganssle, J. & The Ganssle Group. (2004). *A Guide to Debouncing* (p. 12).  
<https://my.eng.utah.edu/~cs5780/debouncing.pdf>

---

*Temperature (PM stepping motors) | Engineering Information | MinebeaMitsumi Product Site*. (z.d.). MinebeaMitsumi Product Site.  
[https://product.minebeamitsumi.com/en/technology/rotary/pm\\_motor\\_cat-1\\_002-03.html](https://product.minebeamitsumi.com/en/technology/rotary/pm_motor_cat-1_002-03.html)

---

Minebea Co., Ltd. (2010). *Selection of PM Motors* (pp. 1–15).  
[https://product.minebeamitsumi.com/en/technology/rotary/\\_icsFiles/afieldfile/2020/10/02/pm\\_select\\_3\\_1.pdf](https://product.minebeamitsumi.com/en/technology/rotary/_icsFiles/afieldfile/2020/10/02/pm_select_3_1.pdf)

---

Ltd, J. W. M. C. (2025, 15 februari). What are the effects of overheating a stepper motor? - Knowledge. *Wantmotor*. <https://www.wantmotor.com/info/what-are-the-effects-of-overheating-a-stepper-102798173.html>

---

Haydon Kerk Motion Solutions / Pittman Motors. (z.d.). *TEMPERATURE EFFECTS ON MOTOR PERFORMANCE*. [https://www.haydonkerkpittman.com/-/media/ametkhaydonkerk/downloads/technical-documents/pittman-technical-documents/temperature\\_effects\\_on\\_dc\\_motor\\_performance.pdf](https://www.haydonkerkpittman.com/-/media/ametkhaydonkerk/downloads/technical-documents/pittman-technical-documents/temperature_effects_on_dc_motor_performance.pdf)

---



*Temperature Effects on DC Motor Performance*. (z.d.).  
<https://www.haydonkerkpittman.com/learningzone/whitepapers/temperature-effects-on-dc-motor-performance>

---

*DrufelCNC - CNC motion control software*. (z.d.).  
<https://drufelcnc.com/?c=blog&p=DipSwitches>

---

*Wat is Carnaubawas?* (2018, 20 september). *Wat Is Carnaubawas?*  
<https://www.greelane.com/nl/science-tech-math/wetenschap/what-is-carnauba-wax-607371/>

---

Libretexts. (2022, 12 september). *8.5: Molecular Orbital Theory*. Chemistry LibreTexts.  
[https://chem.libretexts.org/Bookshelves/General\\_Chemistry/Chemistry\\_1e\\_%28OpenSTAX%29/08%3A\\_Advanced\\_Theories\\_of\\_Covalent\\_Bonding/8.05%3A\\_Molecular\\_Orbital\\_Theory](https://chem.libretexts.org/Bookshelves/General_Chemistry/Chemistry_1e_%28OpenSTAX%29/08%3A_Advanced_Theories_of_Covalent_Bonding/8.05%3A_Molecular_Orbital_Theory)

---

Wikipedia-bijdragers. (2020, 17 april). *Carnaubawas*. Wikipedia.  
<https://nl.wikipedia.org/wiki/Carnaubawas>

---