KU LEUVEN kulak

Groep Wetenschappen & Technologie

X0B54a – Probleemoplossen en ontwerpen, deel 2

Ontwerpproces van een zelf-rijdende wagen

Eindverslag

Team 4: Team R2D2

Matthijs Deforche, Karl Van Holder, Thomas Varheust, Kobe De Weerdt, Yaron Verhulst
Kevin Truyaert, Benjamin Maveau, Martijn Boussé

Academiejaar 2020 – 2021

Inhoud

0.1 Inleiding

Het principe van zelfrijdende wagens is vandaag de dag geen item meer dat alleen in de toekomst mogelijk is. Er zijn al verschillende bedrijven volop aan de slag met de ontwikkeling van zelfrijdende wagens. Een ideaal voorbeeld hiervan is de autofabrikant Tesla. Vele ongevallen worden nog steeds veroorzaakt door de mens, dus lijken zelfrijdende wagens hiervoor een ideale oplossing. Om dit mogelijk te maken is het van essentieel belang om betrouwbare zelfrijdende wagens te ontwikkelen. Dit semesterproject laat ons toe om kennis te maken met de werking en principes van kleine robotwagentjes.

Het concept van *Smart city* ofwel slimme stad is een concept dat we in dit project zullen gebruiken. *Smart city* is een stad waar alles bestuurd en beheerd wordt door informatietechnologie. Mobiliteit is een heel belangrijk deelaspect van het concept. Men wilt het verkeer zo efficiënt mogelijk in de stad regelen zodat er zo weinig mogelijk conflicten, ongevallen, filevorming enzovoort zijn. Dit semester zullen we een zelfrijdend robotwagentje ontwerpen dat zelfstandig door een modelstad zal rijden. Het moet in staat zijn om verkeerslichten te kunnen interpreteren, juiste handelingen te kunnen uitvoeren zoals afslaan of rechtdoor rijden en het zal obstakels zoals voorliggers moeten kunnen ontwijken. Een vlotte verkeerssituatie in *Smart city* is cruciaal en dit is dan ook het doel van dit project. Onze auto zonder problemen en vlot laten rijden in de modelstad net zoals in het concept.

0.2 Klantenvereisten

De constructie van onze wagen moet aan een budget voldoen van 3500 eenheden. Er is een materiaallijst beschikbaar die we kunnen raadplegen om onderdelen te bestellen.

Onze auto moet een geïmplementeerd traject kunnen volgen via een lijn die op de ondergrond is aangebracht. Deze verschilt qua intensiteit met de ondergrond zodat de auto de lijn kan herkennen. Het traject bevat ook meerdere kruispunten met verkeerslichten. Onze auto moet deze kunnen interpreteren en gepast reageren op het rode of groene licht. Om te kunnen stoppen bij het rode licht zal onze auto een dikke stoplijn moeten kunnen detecteren.

Om het traject te volgen zal onze auto dus moeten kunnen rijden en draaien. Ook stoppen is belangrijk. Onze auto moet namelijk voorliggers kunnen detecteren en op tijd stoppen om een botsing te vermijden. Er wordt verwacht dat het volledige traject foutloos uitgereden kan worden op een aanvaardbaar tempo.

Voor onze auto het traject op kan, zullen we ook een noodstop moeten uitvoeren. Dit zal via commando's van een computer moeten gebeuren.

0.3 Ontwerpspecificaties

De verkeerslichten kunnen interpreteren: stoppen bij een rood licht, doorrijden bij een groen licht.
De hoogte van het verkeerslicht gemeten vanaf de grond tot aan het middelpunt van het verkeerslicht is 75 mm. Het verkeerslicht heeft een knipperfrequentie van 1 Hz. Een technische tekening van het kruispunt is te zien op Figuur 1.1.

Een geïmplementeerd traject kunnen volgen / stoplijn detecteren.
De ondergrond van het traject zal ofwel donker ofwel helder zijn. De kleur van de lijnen die de auto moet volgen zal afhangen van wat de kleur van de ondergrond is. Het zal het omgekeerde zijn waardoor het verschil duidelijk is. Er zijn ook nog verschillen in de soorten lijnen die op de ondergrond aangebracht zullen worden. De breedte van een volglijn is 25 mm en de breedte van een stoplijn is 50 mm. De kruispunten liggen op 1000 mm van elkaar en het bovenaanzicht van een kruispunt is te zien op Figuur 1.2.

Commando's van een computer kunnen volgen.
Vooraleer onze auto het traject mag oprijden, moeten we eerst op afstand een noodstop kunnen uitvoeren.

De auto moet aan een bepaald budget voldoen.
Er is een budget van 3500 eenheden beschikbaar. Dit budget kunnen we gebruiken om materiaal aan te kopen op de site: http://www.irkulak.be/po2/. Verder kunnen we het budget ook nog gebruiken om onze eigen ontwerpen te 3D-printen.

Het volledige traject foutloos kunnen afleggen met een aanvaardbaar tempo.
De wagen heeft een maximumbreedte van 250 mm, een maximumhoogte van 300 mm en een minimumhoogte van 75 mm.

0.4 Teamverantwoordelijkheden

Naam	Deforche Matthijs	De Weerdt Kobe	Van Holder Karl
Penningmeester		x	
Teamleider			x
Notulist			
3D-modelleren			
Programmeren	x		x
Verantwoordelijke bouwen			x
Verantwoordelijke aankopen		x	
Presentatie	x	x	

Naam	Thomas Varheust	Yaron Verhulst
Penningmeester		
Teamleider		
Notulist	x	
3D-modelleren	x	x
Programmeren		
Verantwoordelijke bouwen	x	
Verantwoordelijke aankopen		
Presentatie		x

0.5 Ontwerp

0.5.1 Onderdelen

We hebben keuze tussen een Raspberry Pi en een NI MyRio als microcontroller voor de auto en kiezen voor de NI MyRio omdat het programmeren op die controller voor onze toepassing simpeler is dan op de Raspberry Pi. We kunnen op deze manier namelijk voor het volledige programmeerwerk LabVIEUW gebruiken wat zorgt voor een naadloze integratie aangezien we sowieso LabVIEW gebruiken als interface. Daarnaast laat de MyRio ook analoge sensoren toe. Eén van die sensoren is de afstandssensor die een bereik van tot 80cm heeft, tegenover de 10cm van de digitale variant die we ter beschikking hebben, dit laat ons toe de voorliggers eerder te detecteren en op die manier beter en veiliger onze snelhied bij te stellen. We gebruiken voor de andere sensoren ook de analoge versies om het programmeren simpel te houden, dit houdt concreet in dat we slechts op één manier de implementatie voor het inlezen van de sensoren moeten voorzien. Aangezien de NI MyRio een power input nodig heeft van 6-16 Volt, kan de powerbank (met een output van 5V) niet gebruikt worden. Door TWEE Lithium-ion batterijen van 3,6 Volt te kiezen en deze in serie te schakelen is het wel mogelijk om een geschikte spanning voor de power-input te verkrijgen.

Als persoonlijke touch kozen we om geen MakerBeams te gebruiken. In plaats daarvan hebben we dus een chassisplaat nodig en het beste voor ons is het 'rechthoekig zwart chassis' opdat we een confortabele marge hebben om alle onderdelen op de auto te plaatsen. Ook kunnen we de wagen stabieler maken door het zwaartepunt te verlagen wanneer we de onderdelen meer kunnen uitspreiden. Een alternatief van de materiaallijst als chassisplaat zou het 'universeel chassis' zijn maar het 'rechthoekig zwart chassis' kost 50 eenheden minder en is voor de rest vergelijkbaar.

Voor wielen en motoren gaan we voor een *ball caster*, wanneer we deze monteren met bijgevoegde *vulring* komt hij even hoog uit als de grootste wielen uit de lijst. Daardoor ligt de chassisplaat perfect horizontaal. Deze twee wielen drijven we aan met twee van de 'Micro Metal Gear Motor 50:1 HP'. Deze heeft een drie millimeter as, wat compatibel is met onze gekozen wielen en heeft een gemiddeld vermogen, voldoende om de wagen te doen stoppen en opnieuw te versnellen. Omdat deze motoren niet meer beschikbaar waren wanneer wij onze bestellingkonden plaatsen kozen we voor de 'Micro Metal Gear Motor 100:1 HP', deze is redelijk gelijkaardig maar heeft een ander vermogen. Samen met deze motoren kozen we twee motorbeugels om de motoren aan het chassis te kunnen bevestigen en de 'Dual Drive DRV8833' zodat we beide motoren tegelijk kunnen aansturen. Om eerst de wagen te testen voor we gaan solderen op de printplaat kozen we nog een breadbord, aangezien we maar drie sensoren en twee motoren hadden kozen we voor het kleinere formaat omdat dit beter is voor ons budget.

0.5.2 Assemblage

Om de poorten van de myRIO-microcontroller vrij te houden, hebben we de myRIO boven de wielen geplaatst. Omdat de myRIO relatief zwaar is ten opzichte van de andere onderdelen, komt het zwaartepunt van de wagen hoger te liggen. Om te voorkomen dat het zwaartepunt te hoog zou liggen, plaatsen

we de batterijen centraal op het chassis. De reflectiesensor wordt vooraan in het verlengde van het chassis bevestigd via de "reflectie sensor houder". Dit is een onderdeel dat we zelf ge-3D-print hebben omdat we ervoor gekozen hebben om geen makerbeams te gebruiken. De technische tekening van de *reflectie sensor houder* is terug te vinden in Bijlage 1.6. Omdat de afstandssensor maar objecten waarneemt vanaf 80mm hebben we besloten deze op 35mm van de voorkant van het chassis te plaatsen. Zo kan deze efficiënter gebruikt worden. De kleurensensor wordt op 75mm boven de grond, 25mm horizontale afstand van de reflectiesensor en op 65mm van de symmetrieas van het chassis geplaatst omdat het zich zo op de ideale positie bevindt om het verkeerslicht uit te lezen wanneer de auto stilstaat. De afstandssensor en kleurensensor worden geïntegreerd in de wagen via het *Skelet van de wagen*, dit is een ge-3D-print onderdeel die ook de myRIO ondersteunt en waarvan een afbeelding te zien is in Figuur 1.4 in Bijlage 1.6. De sensoren maken verbinding met de myRIO via de printplaat, deze is via plakband bevestigd op de myRIO zodat er voldoende ruimte is om overzichtelijk alles te bevestigen. Het elektrisch circuit is te zien in Bijlage 1.7. Het 3D-model van de wagen is te zien in Figuur 1.3 in Bijlage 1.6.

0.6 Software

0.6.1 flowchart moet nog aangevuld worden

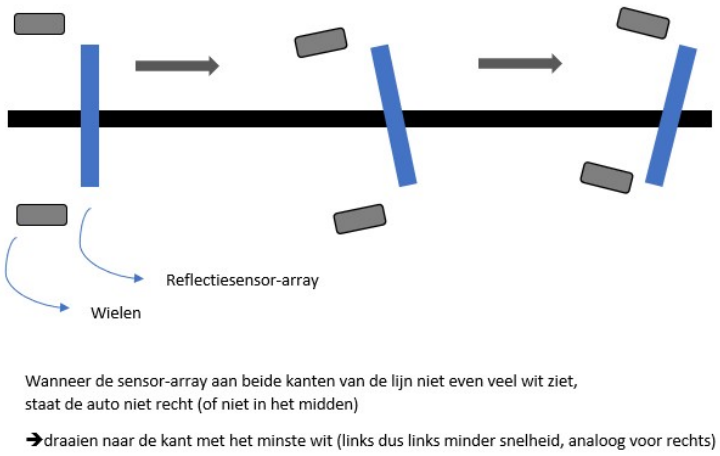
we moeten eerst met behulp van flowchart/stroomdiagram het algemene idee uitleggen vd software dit moet nog geschreven worden

0.6.2 Lijnvolgalgoritme

Het lijnvolgalgoritme werkt volgens een eenvoudig principe dat voorgesteld wordt in Figuur 1. Het parcours bestaat uit donkere lijnen op een witte ondergrond. Bij het volgen van deze lijnen maken we gebruik van de 'QTR-8A analoge reflectie sensor array'. De aansluiting hiervan op de microcontroller kan je zien in Bijlage 1.7. De sensor heeft acht lichtgevoelige sensoren. We hebben ons lijnvolgalgoritme gebaseerd op het feit dat men aan de ene kant van de sensor meer zwart ziet dan aan de andere wanneer die niet evenwijdig met de lijn rijdt. Wanneer men meer zwart ziet aan de linkerkant van de sensor, moet men naar links draaien (dit door de linkermotor trager te doen draaien). Wanneer men echter meer zwart ziet aan de rechterkant van de sensor, moet men naar rechts draaien (dit door de rechtermotor trager te doen draaien). Als men aan beide zijden even veel zwart detecteert, moet men rechtdoor blijven rijden. Dit algoritme komt tot een einde wanneer beide kanten even veel zwart ervaren en deze waardes tezamen veel groter zijn. Dit betekent dat er een stopstreep is, en men dus moet stoppen.

0.6.3 Licht herkennen

Dit algoritme treedt pas in werking als men een stopstreep detecteert. Dit is zo wanneer de acht sensoren van de reflectiesensor zwart detecteren (zoals in vorige paragraaf beschreven). Met behulp van een while-loop gaan we kijken of het licht groen is dat de kleurensensor detecteert. We gaan gebruik maken van een polling rate van 1 seconde zodat we de myRIO niet overbelasten. Zolang



Figuur 1: Het lijnvolgalgoritme laat ons robotwagentje toe om zijn traject te corrigeren indien nodig. Wanneer er links meer wit te zien is, zal het algoritme de wagen naar rechts laten draaien zodat hij de lijn correct volgt. Dit door de linkse motor sneller te laten

het rood is blijft de while-loop lopen, vanaf de kleurensensor groen ziet, stopt die while-loop en kan het traject verdergezet worden.

0.6.4 Andere wagen herkennen en versnellen/vertragen

Met de afstandssensor kunnen we de afstand tot een voorliggende wagen detecteren. Wanneer de afstand te klein is, gaat de wagen vertragen. Als echter bij het vertragen de kritieke minimale snelheid overschreden wordt (dus nog minder dan de minimale), dan stopt onze wagen (dit is wanneer een andere wagen voor ons stilstaat). Wanneer de snelheid boven een bepaalde waarde is, neemt hij de 'ideale snelheid' aan.

0.6.5 Draaien op kruispunt

We maken een ondersheid tussen drie gevallen; linksaf, rechtsaf en rechtdoor. Bij linksaf moeten we eerst 375mm rechtdoor rijden, dan 90graden naar links draaien om de as van de wagen en tenslotte meer dan 375 mm rechtdoorrijden om dan weer de lijn te volgen.

Bij rechtsaf moet men eerst 125mm rechtdoorrijden, dan 90graden naar rechts draaien om de as van de wagen en tenslotte meer dan 125mm rechtdoorrijden om dan weer de lijn te volgen. Bij rechtdoor moet men meer dan 500mm rechtdoorrijden om dan vervolgens de lijn te volgen. bij deze drie gevallen moet men tijdens het uitvoeren ervan rekening houden of er al dan niet een voorligger zich binnen de zeer kritische gevarenzone bevindt (dit wil zeggen dat de ander wagen voor de een of andere reden stilstaan op het kruispunt)

0.6.6 Handmatige Besturing

Om de controle manueel over te nemen sturen we de motoren rechtstreeks aan. Dit stellen we voor met behulp van twee sliders in LabVIEW. Later zullen we aan die sliders concreet een invulling voor geven om op de test te gebruiken.

0.6.7 Implementatie traject

Vanaf het moment dat we ons traject kennen, kunnen we aan de hand van de vorige algoritmes ons parcours samenstellen. De flowchart van ons programma is terug te vinden in bijlage 1.8.

0.6.8 Concrete implementatie

Het hele LabVIEW-programma, gebruikmakend van de voorgenoemde functies splitsen we op in twee stappen:

De eerste stap is wat de auto moet doen tijdens het rijden. Hier moet de auto tegelijk de lijn op de grond volgen, een goede afstand tot zijn voorligger behouden (wat dus inhoudt dat die de voorligger moet detecteren en aan de hand van de afstand tot die voorligger zijn snelheid moet aanpassen) en de auto moet ook nog eens comfortabel kunnen stoppen,en dus niet te snel rijden, wanneer die een stopstreep detecteert. De input voor de snelheid komt van de motoren, de input voor de afstand tot de voorligger komt van de afstandssensor, de lijn en stopstreep detecteren we met de reflectiesensorarray en tot slot het verkeerslicht (dat besproken zal worden in de volgende stap) detecteren we met de kelurensensor.

De tweede stap begint bij de stopstreep van bij het eindd van stap één. Wanneer de auto gestopt is aan de stopstreep moet die een verkeerslicht detecteren en daar gepast opreageren, nadat het groen licht geworden is moet die dus over het kruispuntdraaien. Daarna moet die een bocht nemen of rechtdoor rijden aan de hand van een vooraf opgesteld parcours. Na deze stappen is het algoritme klaar en begint hij weer met stap één.

0.6.9 Implementatie traject

Vanaf het moment dat we ons traject kennen, kunnen we aan de hand van de vorige algoritmes ons parcours samenstellen. De flowchart van ons programma is terug te vinden in bijlage 1.8.

Hoofdstuk 1

Bijlagen

1.1 Planning

1.2 Overzicht ontwerpspecificaties

1.3 Teamverantwoordelijkheden

Verantwoordelijkheidsstructuur

Penningmeester: Kobe De Weerdt

Teamleider: Karl Van Holder

Notulist: Thomas Varheust

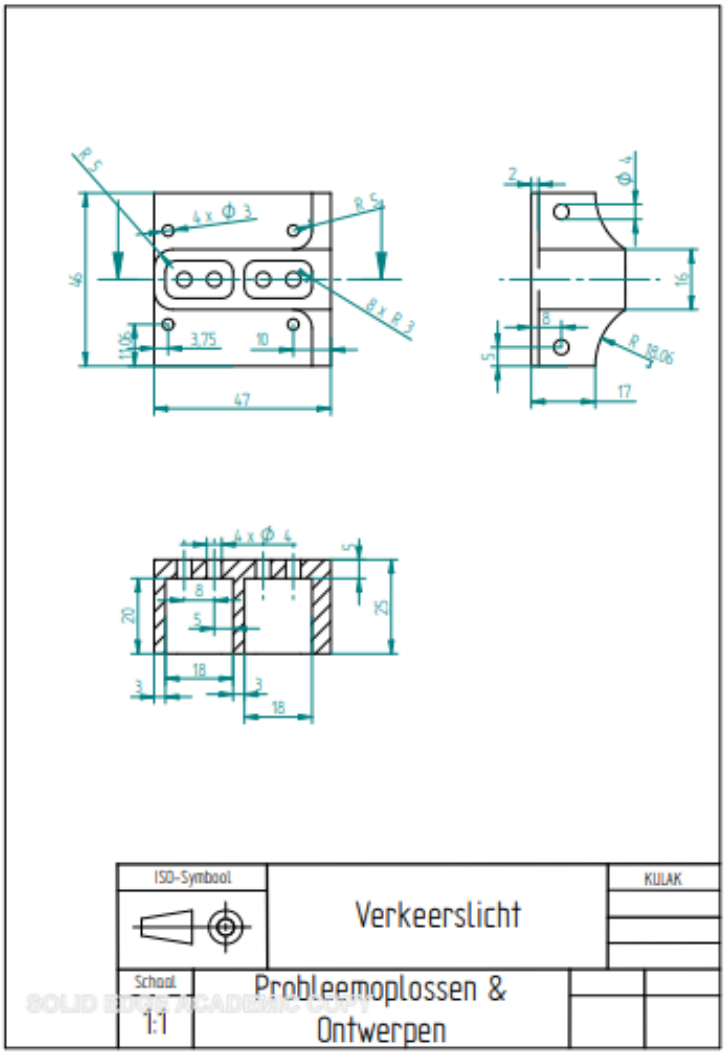
3D-modelleren + technische tekeningen: Yaron Verhulst + Thomas Varheust

Programmeren: Matthijs Deforche + Karl Van Holder

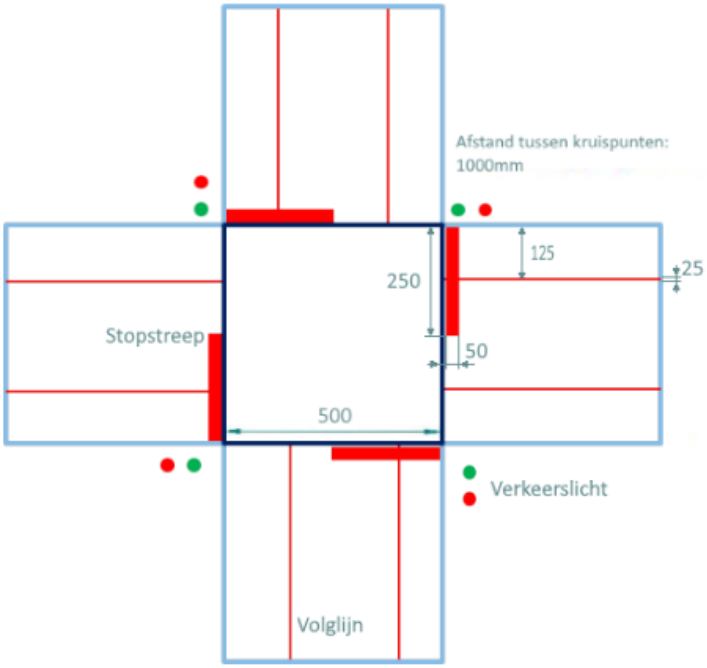
Verantwoordelijke bouwen: Yaron Verhulst + Karl Van Holder

Verantwoordelijke aankopen: Kobe De Weerdt

Presentatie: Matthijs Deforche + Kobe De Weerdt



Figuur 1.1: Technische tekening verkeerslicht, opgehaald van [1].



Figuur 1.2: Bovenaanzicht kruispunt met relevante maten en items, aangepast vanuit [2].

1.4 Takenstructuur

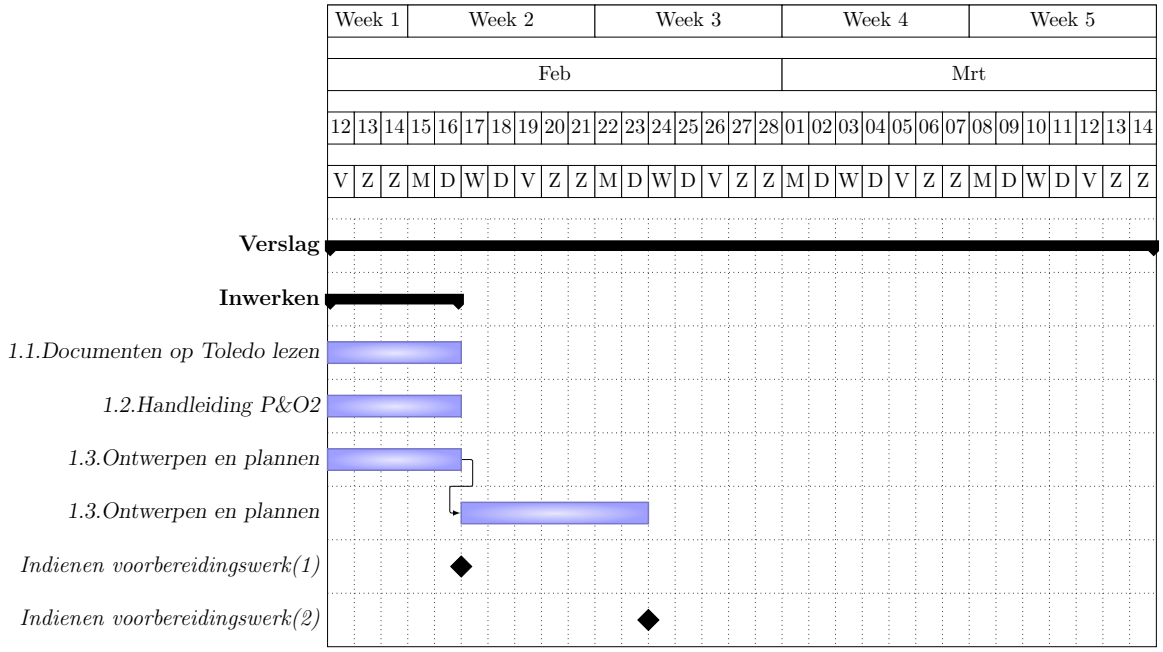
Tabel 1: Takenstructuur

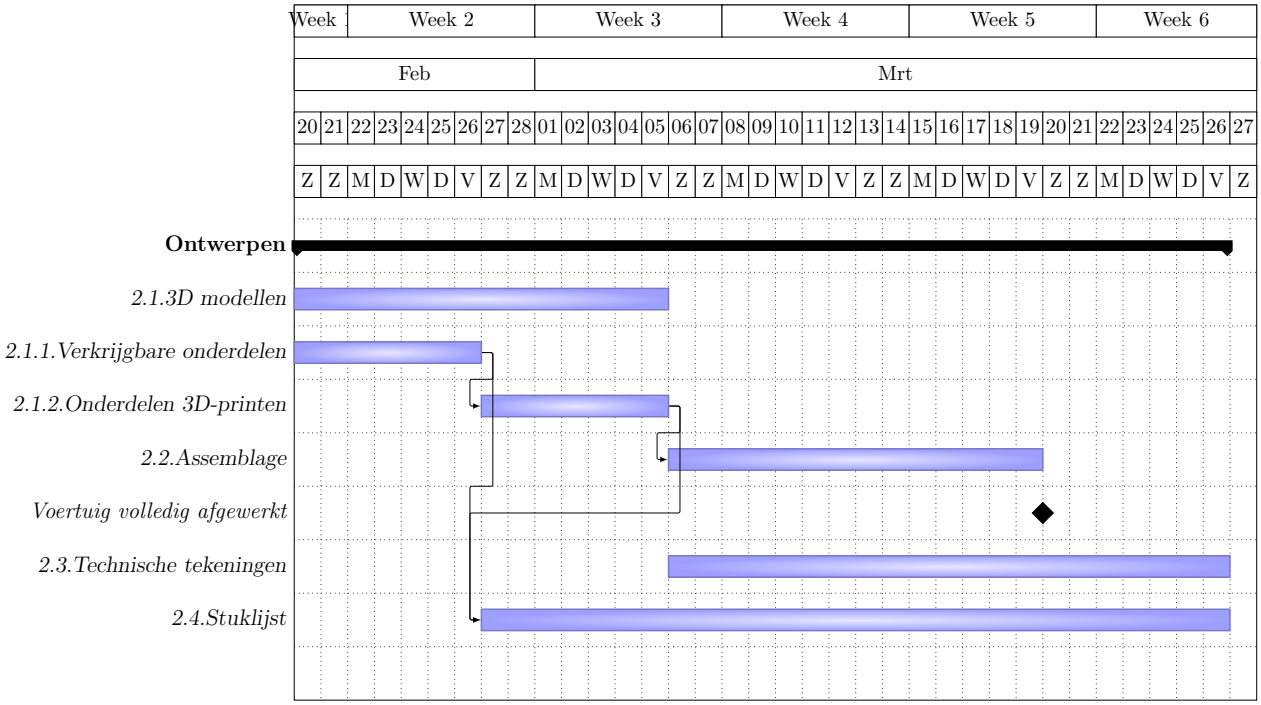
Code	Taak	Status
1	Inwerken	OK
1.1	Documenten op Toledo lezen	OK
1.2	Ontwerpen en plannen	OK
1.2.1	Materiaallijst maken	OK
1.2.2	Teamkalender maken	OK
1.2.3	Klantenvereisten opstellen	OK
1.2.4	Overzicht ontwerpspecificaties	OK
1.2.5	Takenstructuur	OK
1.2.6	Gantt-grafiek	OK
2	Ontwerpen met behulp van de computer	niet OK
2.1	3D modellen (Solid parts)	OK
2.1.1	Wiel	OK
2.1.2	Motor	OK
2.1.3	Motorbeugel	OK
2.1.4	Chassis	OK
2.1.5	Ballcaster	OK
2.1.6	Printplaat	OK
2.1.7	Kleursensor	OK
2.1.8	Afstandssensor	OK
2.2	Assemblage (Assembly)	OK
2.3	Technische tekeningen (Drawing)	niet OK
2.4	Stuklijst	niet OK
3	Software	niet OK
3.1	Sturen/snelheid regelen	OK
3.2	Lijnvolgalgoritme	OK
3.3	Verkeerslichtinterpretatie	OK
3.4	Voorliggerdetectie	OK
3.5	Stoppen	OK
3.6	Vertragen	OK
3.7	Handmatig besturen	OK
3.8	Inputs vormgeven	niet OK
3.9	Outputs vormgeven	niet OK
3.10	Testen	niet OK
4	Rapportering	niet OK
4.1	Tussentijds verslag	OK
4.1.1	Nalezen	OK
4.2	Tussentijdse presentatie	niet OK
4.2.1	Structuur	OK
4.2.2	Presentatie maken	niet OK
4.2.3	Nalezen	niet OK
4.2.4	Inoefenen	niet OK
4.3	Eindverslag	niet OK
4.3.1	Structuur	niet OK
4.3.2	Nalezen	niet OK
4.4	Eindpresentatie	niet OK
4.4.1	Structuur	niet OK
4.4.2	Presentatie maken	niet OK
4.4.3	Nalezen	niet OK
4.4.4	Inoefenen	niet OK

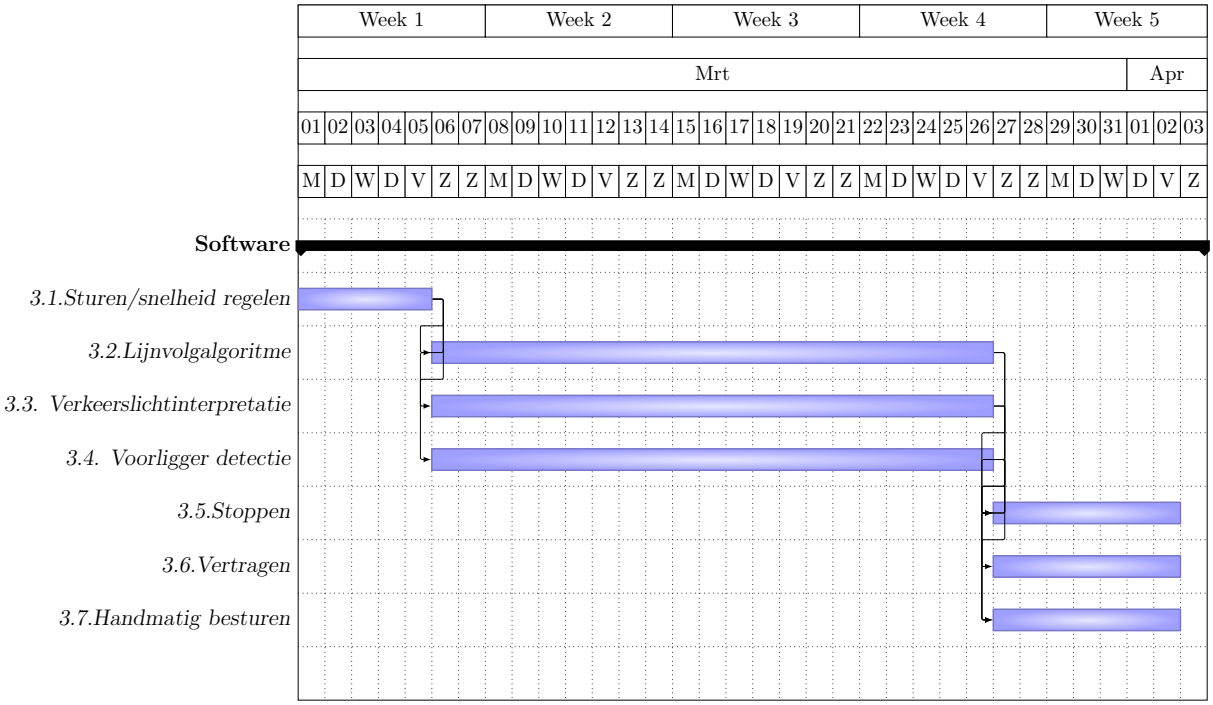
Tabel 1.1: Takenstructuur

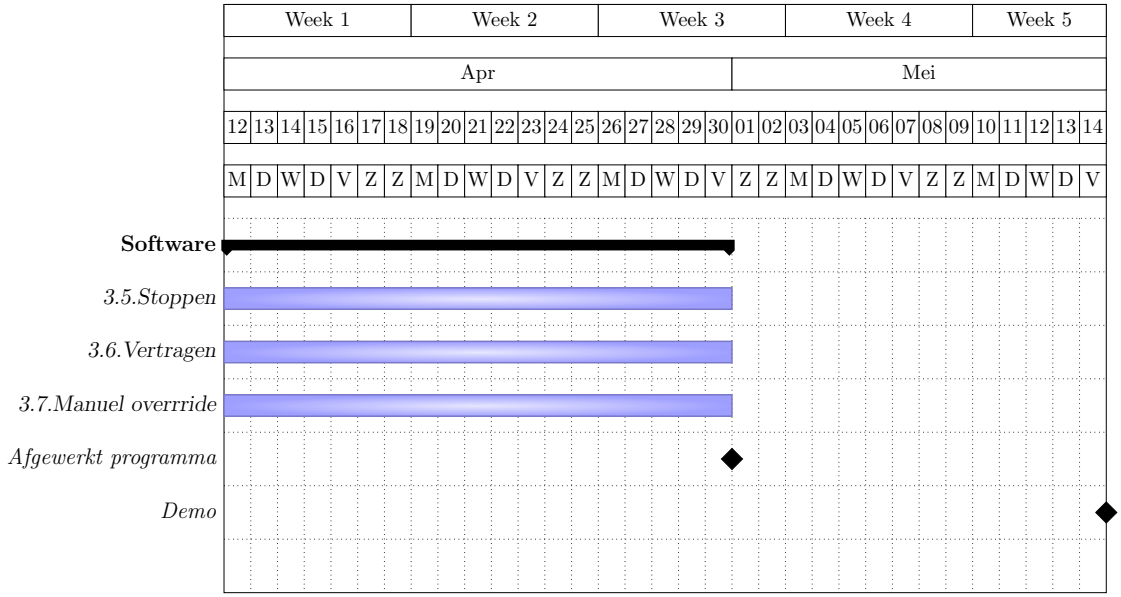
1.5 Gantt-grafiek & Teamkalender

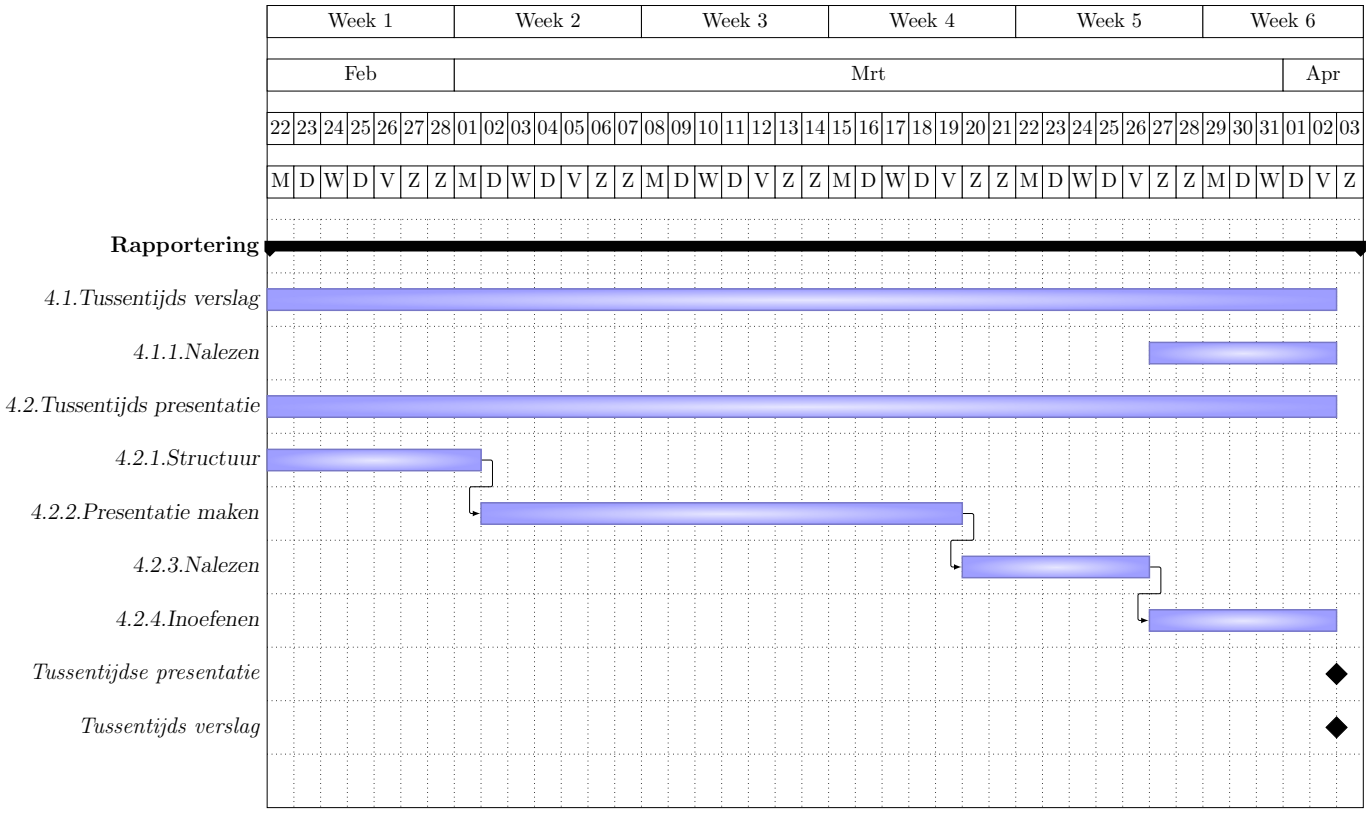
Week	Data	Maandag	Dinsdag	Woensdag	Donderdag	Vrijdag
1	08/02-12/02					
2	15/02-19/02		1.3			
3	22/02-26/02		1.4			
4	01/03-05/03	4.2.1				2.1.2/3.1
5	08/03-12/03					
6	15/03-19/03					2.2/4.2.2
7	22/03-26/03					2.3/2.4/4.2.3/3.2/3.3/3.4
8	29/03-02/04					tussentijdse presentatie en verslag
Paasvakantie1	05/04-09/04					
Paasvakantie2	12/04-16/04					
9	19/04-23/04	4.4.1				
10	26/04-30/04					3.5/3.6/3.7
11	03/05-07/05					demonstratie/4.4.2
12	10/05-14/05					4.4.3
13	17/05-21/05				eindverslag	eindpresentatie

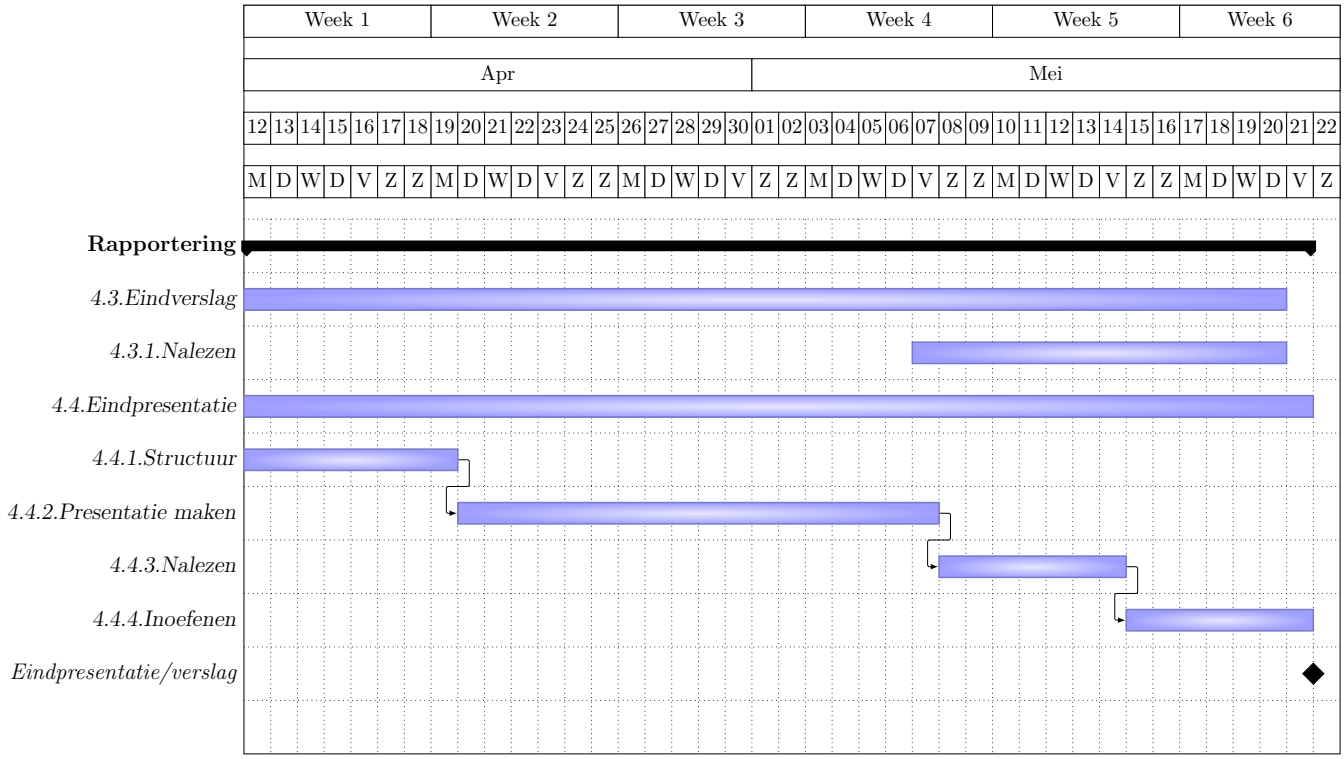

Week 1
Week 2
Week 3
Week 4
Week 5
Feb
Mrt
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 01 02 03 04 05 06 07 08 09 10 11 12 13 14
V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z
Verslag
Inwerken
1.1.Documenten op Toledo lezen
1.2.Handleiding P&O2
1.3.Ontwerpen en plannen
1.3.Ontwerpen en plannen
Indienen voorbereidingswerk(1)
Indienen voorbereidingswerk(2)


Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Feb
Mrt
20 21 22 23 24 25 26 27 28 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z
Ontwerpen
2.1.3D modellen
2.1.1.Verkrijgbare onderdelen
2.1.2.Onderdelen 3D-printen
2.2.Assemblage
Voertuig volledig afgewerkt
2.3.Technische tekeningen
2.4.Stuklijst

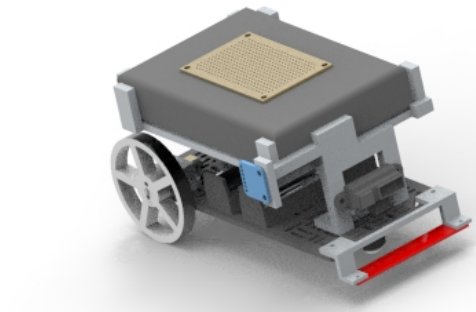

Week 1
Week 2
Week 3
Week 4
Week 5
Mrt
Apr
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 01 02 03
M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z
Software
3.1.Sturen/snelheid regelen
3.2.Lijnvolgalgoritme
3.3. Verkeerslichtinterpretatie
3.4. Voorligger detectie
3.5.Stoppen
3.6.Vertragen
3.7.Handmatig besturen


Week 1
Week 2
Week 3
Week 4
Week 5
Apr
Mei
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 01 02 03 04 05 06 07 08 09 10 11 12 13 14
M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V
Software
3.5.Stoppen
3.6.Vertragen
3.7.Manuel overrride
Afgewerkt programma
Demo

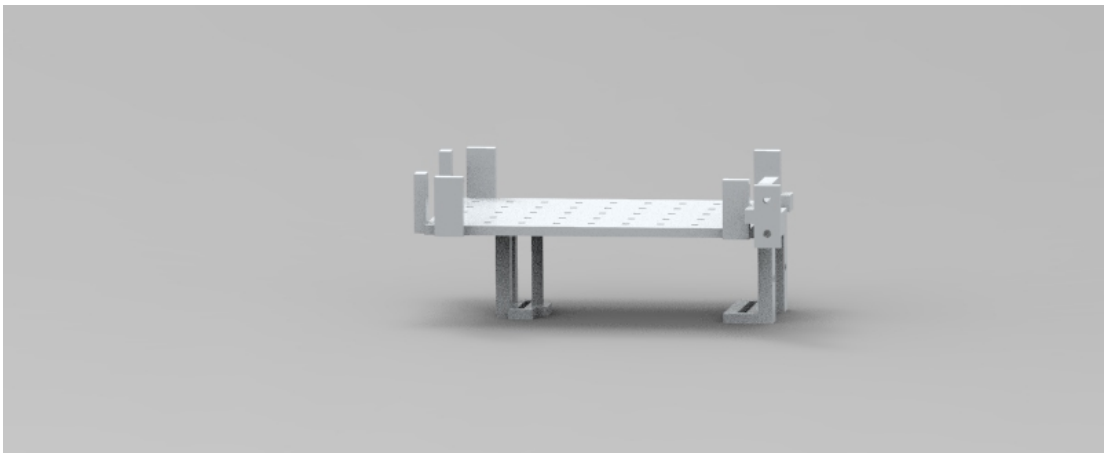

Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Feb
Mrt
Apr
22 23 24 25 26 27 28 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 01 02 03
M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z
Rapportering
4.1.Tussentijds verslag
4.1.1.Nalezen
4.2.Tussentijds presentatie
4.2.1.Structuur
4.2.2.Presentatie maken
4.2.3.Nalezen
4.2.4.Inoefenen
Tussentijdse presentatie
Tussentijds verslag


Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Apr
Mei
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z M D W D V Z Z
Rapportering
4.3.Eindverslag
4.3.1.Nalezen
4.4.Eindpresentatie
4.4.1.Structuur
4.4.2.Presentatie maken
4.4.3.Nalezen
4.4.4.Inoefenen
Eindpresentatie/verslag

1.6 Visualisatie ontwerp



Figuur 1.3: CAD-ontwerp van de wagen, samengesteld uit *.par-onderdelen*, gezien vanuit de rechterbovenhoek.

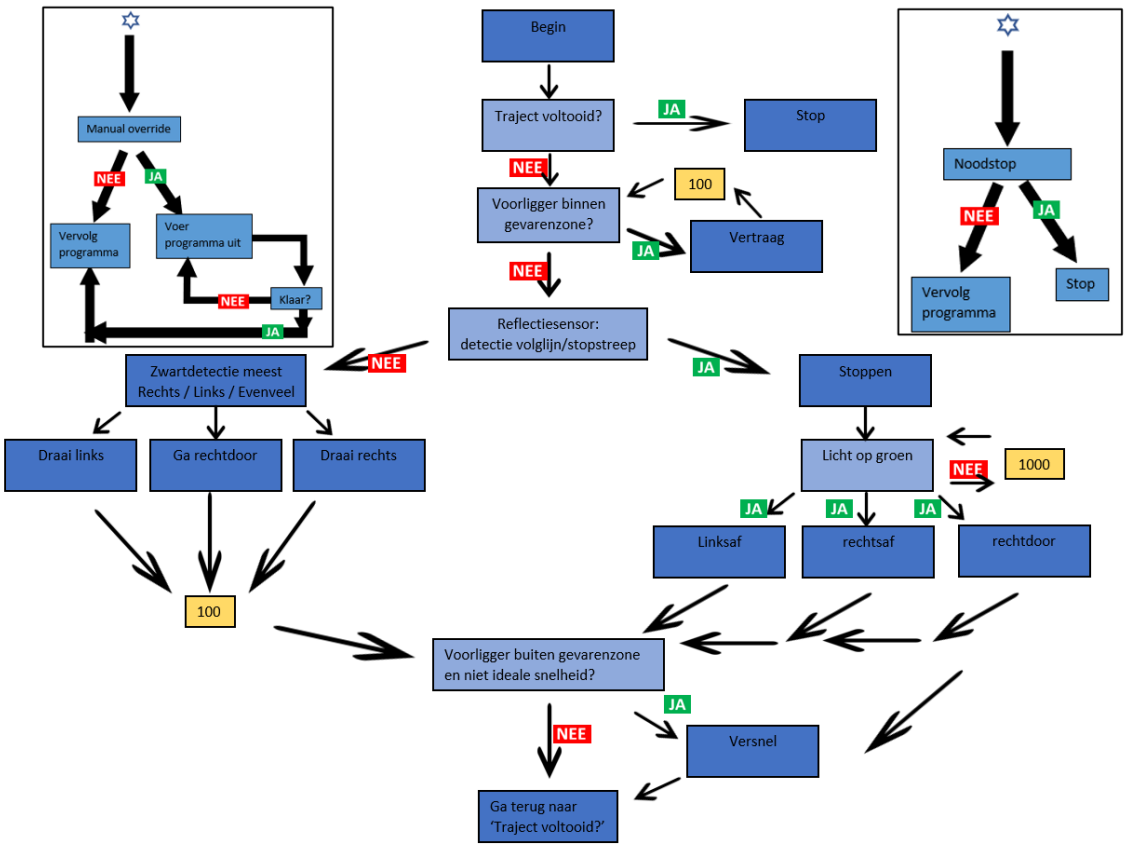


Figuur 1.4: Het 3D-geprinte onderdeel *Skelet van de wagen*

.

1.7 Elektrisch circuit

1.8 Flowcharts



Figuur 1.5: De flowchart van ons algoritme.

1.9 Financieel Rapport

Hardware

Onderdeel	prijs per stuk	aantal	totale prijs
Micro Metal Gear Motor 100:1 HP	160	2	320
Dual Drive DRV8833	70	1	70
Optische afstandsensor (analoog)	160	1	160
TCS34725 Kleur sensor BOB	150	1	150
QTR-8A analoge reflectie sensor array	150	1	150
Oplaadbare LITHIUM-ION	90	2	180
NI MyRio	240	1	240
Breadboard Tiny	40	1	40
Printplaat	50	1	50
Totaal: 1360			

Tabel 1.2: uitgaven hardware

Ontwerp

Onderdeel	prijs per stuk	aantal	totale prijs
Ball Caster	60	1	60
Wiel 60x8mm zwart	35	2	70
Robot Chassis Rechthoekig Zwart	70	1	70
Totaal: 200			

Tabel 1.3: uitgaven ontwerp

Assemblage

Onderdeel	prijs per stuk	aantal	totale prijs
Micro metal gear motor beugel	2	25	50
Skelet van de wagen	N.A	1	N.A
Reflectie sensor houder	N.A	1	N.A
Totaal: 50			

Tabel 1.4: uitgaven assemblage

Onkosten

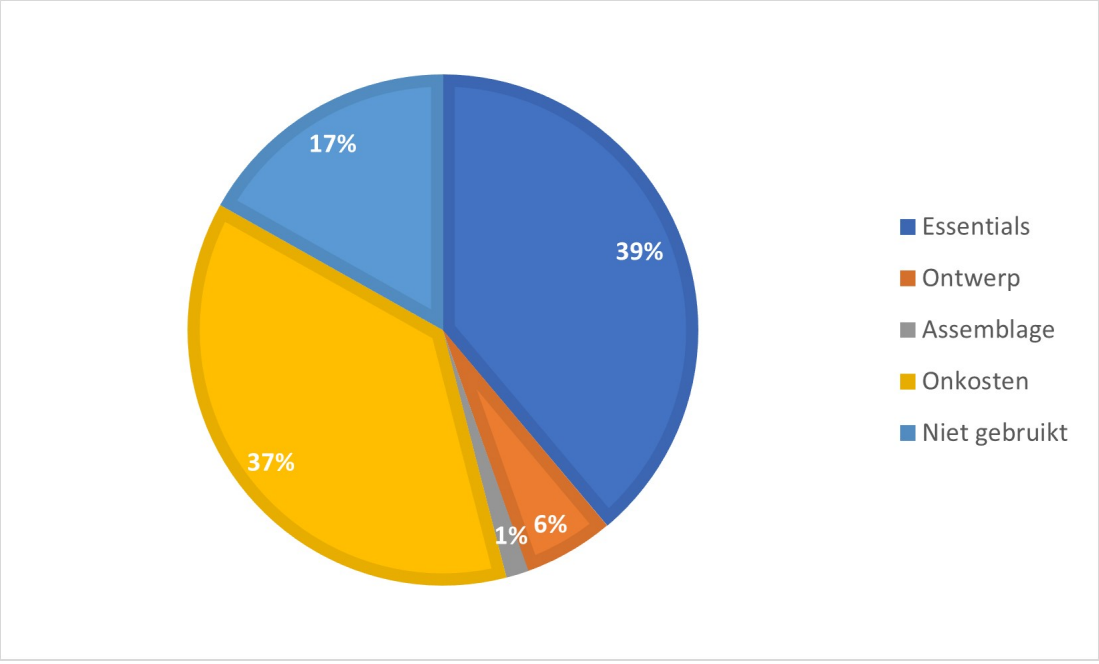
Niet gebruikt

Totaal: 590

kost	waarde
Bieding	1300

Totaal: 1300

Tabel 1.5: uitgaven onkosten



Figuur 1.6: Visualisatie van de besteding van ons budget

Bibliografie

[1] Kevin Truyaert Benjamin Maveau. Opgave teamopdracht probleemoplossen en ontwerpen 2.

[2] Martijn Boussé Benjamin Maveau, Kevin Truyaert. P&o2 : Smart city.

Team 4: Team R2D2
Groep Wetenschap & Technologie
KU Leuven Kulak
Etienne Sabbelaan 53, 8500 Kortrijk