

# Project Elektronica 4

## Projectplan

---

Academiejaar 2014-2015  
Semester 4

**2EA2**

Lorenz Put – s079368, [lorenz.put@student.ap.be](mailto:lorenz.put@student.ap.be)

Michiel Mulder – s079157, [michiel.mulder@student.ap.be](mailto:michiel.mulder@student.ap.be)

*Projectleider : Marc Smets*

## Inhoud

<b>1. Inleiding</b>	3
<b>2. Probleemstelling</b>	4
<b>3. Mindmap</b>	4
<b>4. Gantt – chart</b>	6
<b>5. Software analyse</b>	7
<b>6. Bijlagen</b>	8

## 1. Inleiding

Ons project in semester 3 bestond er in de auto op afstand te kunnen laten rijden. Voor de aansturing van de motoren was kennis van H-bruggen, mosfets en transistoren nodig. Voor de auto op afstand te kunnen laten rijden was kennis van een IR afstandsbediening en ontvanger vereist. In het algemeen was het ook belangrijk een basiskennis te hebben van de werking van de Arduino.

Het project in semester 4 bouwt voort op dat van semester 3. Het doel in dit semester is om de auto autonoom te laten rijden. Wat houdt dit in? Door middel van sensoren moet de auto hindernissen kunnen ontwijken. Er zijn 5 verschillende opdrachten die volbracht moeten/kunnen worden. De eerste opdracht bestaat erin om een parcours dat op voorhand bekend is af te leggen, m.a.w. de auto wordt voorgeprogrammeerd om een bepaald parcours af te leggen. Hier komen nog geen sensoren aan te pas. In opdracht 2 worden er wel sensoren gebruikt. In deze opdracht is het de bedoeling dat de auto een onbekend parcours af legt. Dit doen we met behulp van sensoren die obstakels detecteren. In opdracht 3 moet de auto een "speciaal" object kunnen detecteren en er vervolgens naar toe kunnen rijden. Als de auto het object bereikt heeft, moet hij dit laten weten. Dit kan door een muzikje te spelen, een ledje te laten branden en/of een vreugdedansje te doen. Opdracht 4 is een combinatie van opdracht 2 en 3. Er zal dus weer een object geïdentificeerd moeten worden. Deze is echter niet in een zomaar in een open ruimte, maar in een ruimte of parcours vol obstakels. De laatste opdracht bestaat uit opdracht 4 plus toevoeging van een tijdslimiet. Er zal dus naar een object gereden moeten worden via allerlei hindernissen binnen een bepaald tijdsbestek.

Hoe gaan we dit aanpakken? We beginnen met het opstellen van een projectplan en het presenteren hiervan. Daarna gaan we verder met het bestukken van de PCB en het aansluiten van de sensors. Nadat we de componenten hebben gerecupereerd van het matrixbord uit vorig semester, is het belangrijk dat we de componenten uitvoerig testen om na te gaan welke we kunnen hergebruiken. De volgende stap is dan om de sensors correct uit te kunnen lezen. Hier letten we goed op de nauwkeurigheid van het gebruik van de uitgelezen waarden. Verder combineren we deze data met de sturing van de auto en kunnen we dus objecten detecteren. Als laatste stap is het dan de bedoeling om de verschillende levels uit voorgaande paragraaf te implementeren.

## 2. Probleemstelling

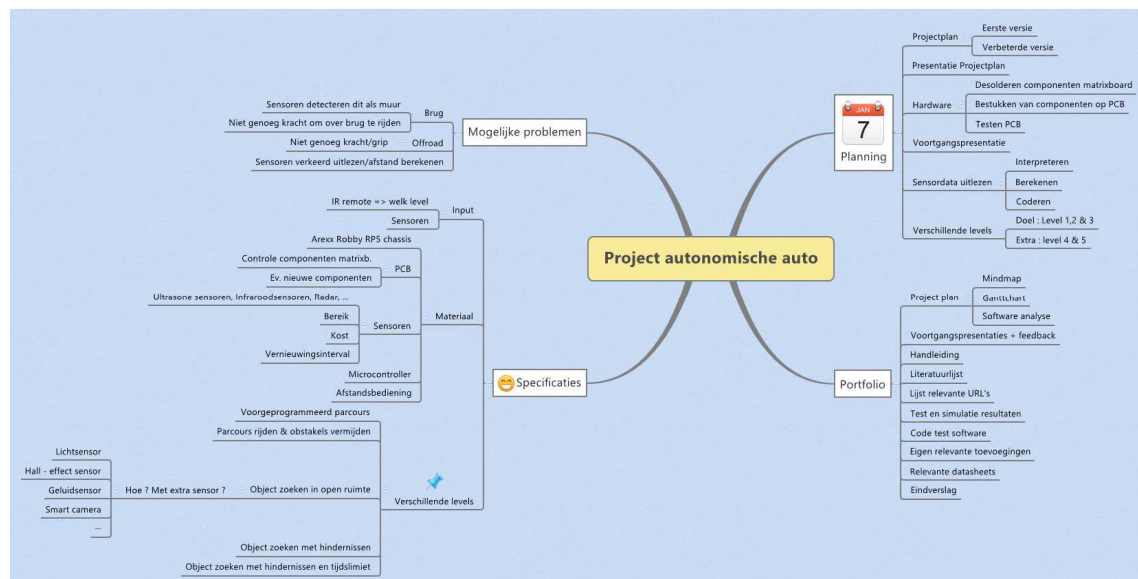
De sensoren vormen in project elektronica 4 een hele uitdaging. Ze moeten nauwkeurig uitgelezen kunnen worden op verschillende manieren. Dit wil zeggen dat ze tevens zeer nauwkeurig geplaatst moeten worden. Naast deze fysieke kwestie is de logica en interpretatie achter de uit te lezen waarden een belangrijk aspect. De waardes bepalen de afstand tussen de auto en de muren, obstakels, objecten en de grond. Hierop is het traject en de locatie van de auto gebaseerd, vandaar het belang. De data die we doorkrijgen van de sensoren zal doordacht gebruikt moeten worden in de software.

We vormen de vraag:

***Hoe laten we onze auto autonoom zijn weg en/of objecten vinden in een vast of onbekend parcours/ruimte, zonder dat deze hinder ondervindt?***

Met deze vraag in het hoofd beginnen we aan het project.

## 3. Mindmap



De mindmap (grotere versie vindt u in de bijlagen) die we hebben opgesteld is eigenlijk een visuele voorstelling van de brainstorm (vindt u ook terug in de bijlagen)

De mindmap bestaat uit 4 hoofdtakken namelijk: mogelijke problemen, specificaties, planning en portfolio.

De tak **mogelijke problemen** omschrijft zoals de naam zelf zegt de mogelijke problemen die kunnen opduiken tijdens het project. Voorbeelden hiervan zijn: de brug als muur detecteren, genoeg vermogen om over de brug en het offroad stuk te rijden en de sensoren juist uitlezen / afstand berekenen.

De tak **specificaties** bevat 3 subtakken namelijk input, materiaal en verschillende levels.

*Input* beschrijft wat er nodig is om het voertuig vanop een afstand aan te sturen. In dit geval is dat een IR remote waarmee we kunnen bepalen welk van de verschillende levels uitgevoerd moeten worden.

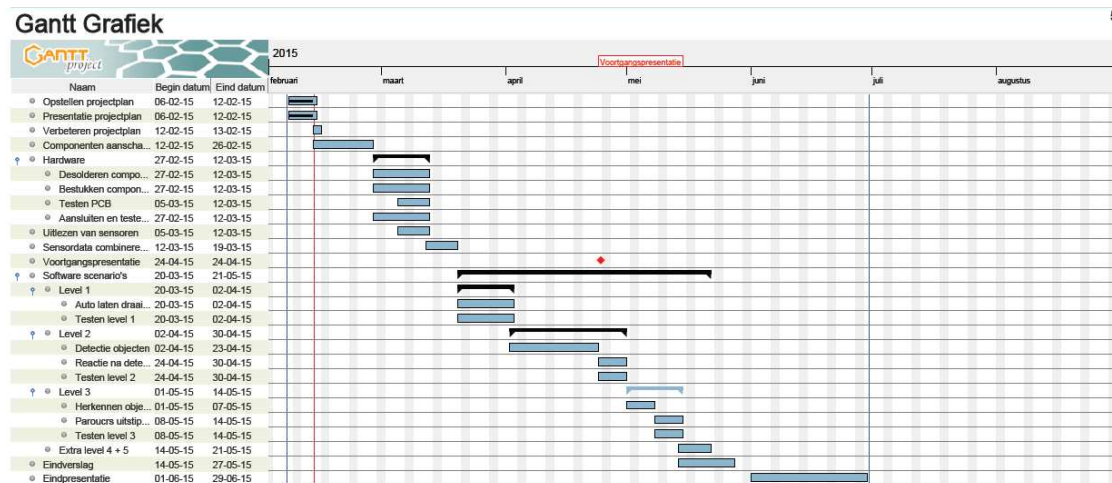
*Materiaal* omvat de zaken die nodig zullen zijn tijdens het vervolledigen van het project. Sommige dingen spreken voor zich zoals bijvoorbeeld de PCB. Over andere zaken moeten we eerst nog verder nadenken, zoals bijvoorbeeld welke soort sensoren we zullen gebruiken.

De tak *verschillende levels* geeft weer wat de verschillende opdrachten zijn die uitgevoerd kunnen/moeten worden. Vooral bij opdracht 3 en verder moet er nog wat onderzoekwerk verricht worden om te bepalen hoe we een object in een ruimte gaan detecteren.

De tak **planning** omvat alle belangrijke zaken die niet vergeten mogen worden tijdens het opstellen van de Gantt-chart. Deze tak geeft dus een overzicht weer van alle geplande taken.

De tak **portfolio** bevat al het “papierwerk” dat uitgevoerd moet worden tijdens het project. De volledige portfolio moet afgegeven worden op het einde van het jaar dus het is zeer belangrijk dat dit tijdens het verdere verloop van het project zeer goed wordt bij gehouden.

## 4. Gantt – chart



De Gantt-chart geeft de tak planning van de mindmap weer in functie van de tijd. Op deze manier krijgen we een beter overzicht van de totale beschikbare tijd en kunnen we alle deelproblemen zo efficiënt mogelijk plannen.

Het eerste deel van de Gantt-chart focust zich vooral op het maken en presenteren van het projectplan. Extra info omtrent dit zou overbodig zijn, daarom wijden we hier ook niet verder over uit.

Verder hebben we dan nog componenten verzamelen en informatie vergaren. Tijdens deze fase is het vooral de bedoeling om beslissingen omtrent de componenten die we gaan gebruiken. De juiste beslissing kan echter alleen genomen worden indien er voldoende info vergaard is.

Deel 3 van de Gantt-chart focust zich op het hardware gedeelte. Dit omvat het lossolderen van de componenten op het matrixboard, het terug vast solderen van de componenten op de PCB, het uitvoerig testen van de PCB en het aansluiten en testen van de sensoren.

Het volgende hoofdpunt omvat het volledige software gedeelte van het project. Wat houdt dit in ? Als eerste stap is het de bedoeling om een algoritme te maken dat het mogelijk maakt om de eerst nutteloze sensordata om te zetten naar een eenheid waarmee we kunnen werken bijvoorbeeld : centimeter (afstand). Als dit gebeurt is kunnen we beginnen met effectief uitvoeren van de opdracht. Deze opdracht bestaat uit verschillende levels namelijk 1 tot en met 5. Voor een uitleg over de levels verwijst ik naar de inleiding. Voor ons is dit in zeer belangrijk deel. Daarom neemt dit hoofdpunt ook het grootste deel van de totale beschikbare tijd in beslag neemt.

In de laatste weken tijd die we nog hebben is het de bedoeling dat we ons eindrapport en presentatie samenstellen en voorbereiden.

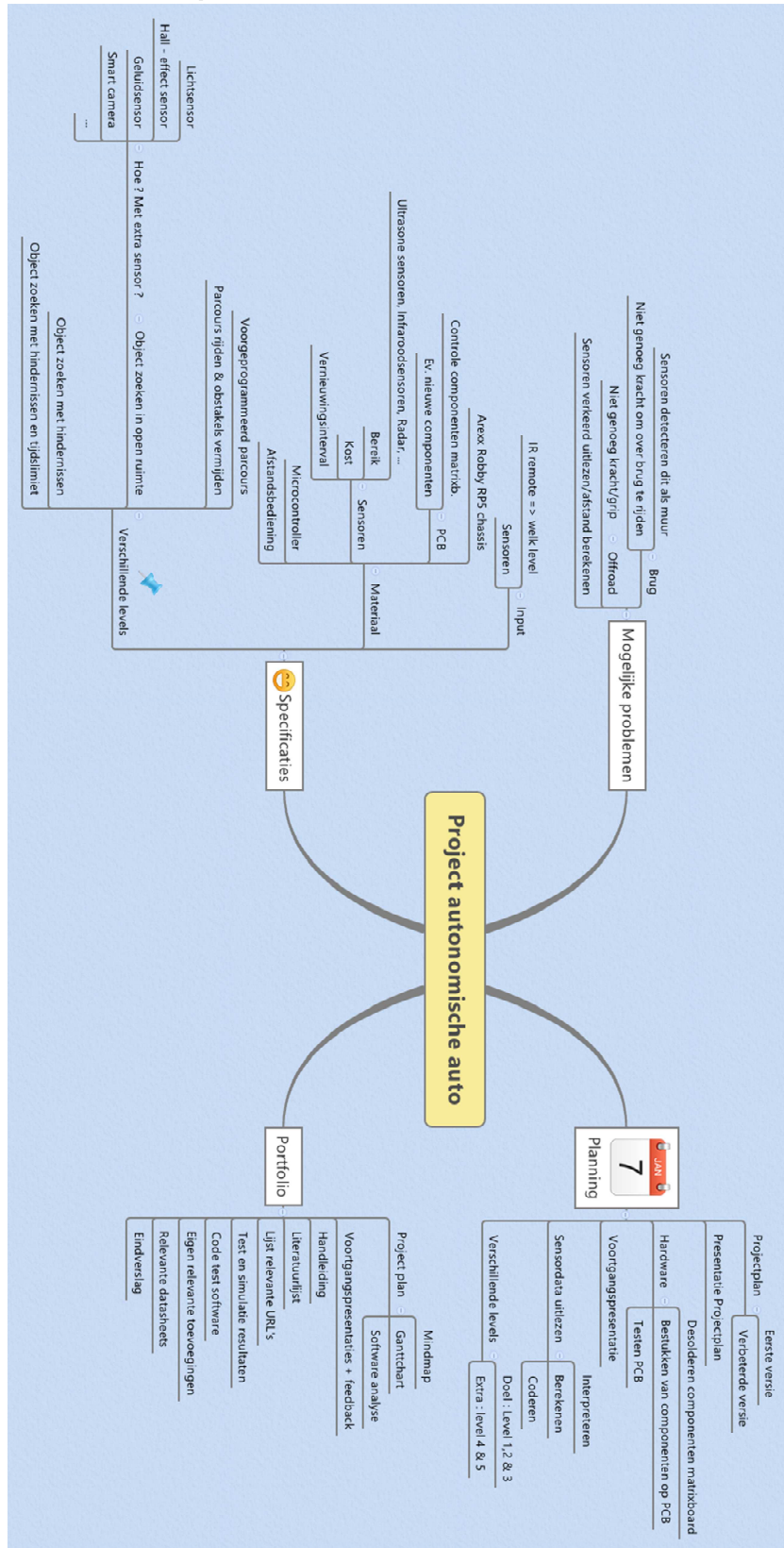
## 5. Software analyse

Om de software correct op te stellen zullen we gebruik maken van een software analyse, gebaseerd op het Nassi Shneidermann diagram. Dit betekent rekening houden met sequentie (opeenvolging van handelingen), selectie (keuzemogelijkheid) en iteratie (lus structuur).

1. **Sequentie**, wat te maken heeft met invoer en declaratie:
  - We gebruiken een eerste sensor als invoer. We kunnen de auto ook aansturen aan de hand van onze infrarood afstandsbediening, maar dit is niet het mikpunt van ons project.
  - Naast de eerste maken we gebruik van een tweede sensor.
  - (eventueel meerdere sensoren afhankelijk van de noodzaak.)
2. **Selectie**, wat te maken heeft met verwerking van gegevens tot informatie:
  - We vergelijken de afstanden van de sensoren hier, rekening houdend met afstand van de auto t.o.v. de muren of obstakels en de vloer.
  - Aangezien we nog geen concrete beslissing hebben wat de sensoren betreft is het nog niet mogelijk om specifieke berekeningen en formules toe te passen. Van zodra we een beslissing hebben genomen kunnen we hiermee aan de slag.
3. **Iteratie**, wat te maken heeft met de uitvoer:
  - De auto moet autonoom zijn weg kunnen zoeken in een vastgelegd parcours met obstakels (brug en offroad), zonder de kanten te raken.
  - De auto moet autonoom zijn weg kunnen zoeken in een onbekend hindernissenparcours, zonder de kanten te raken.
  - De auto moet in een tijdspanne van 3 minuten, in een open ruimte (minimum 5m<sup>2</sup>) een object kunnen herkennen en vinden. Indien de auto het object heeft gevonden moet deze een bepaald signaal van 'vreugde' uitdrukken. D.i. een liedje (vb.: "I am the one and only!"), flikkerende led, ...
  - De auto moet in een tijdspanne van 3 minuten, in een open ruimte (minimum 5m<sup>2</sup>) een object kunnen herkennen en vinden (inclusief vreugdedansje). Dit maal in een oppervlakte/parcours met hindernissen en/of obstakels.
  - De auto moet in een tijdspanne van 2 minuten, in een open ruimte (minimum 5m<sup>2</sup>) vijf objecten kunnen herkennen en vinden (inclusief vreugdedansje). Dit ook in een oppervlakte/parcours met hindernissen en/of obstakels.

## 6. Bijlagen

### 6.1 Mindmap





## 6.2 Brainstorm

### ALGEMEEN:

- opstellen projectplan
- presentatie projectplan
- verbeteren projectplan
- eindverslag
- eindpresentatie

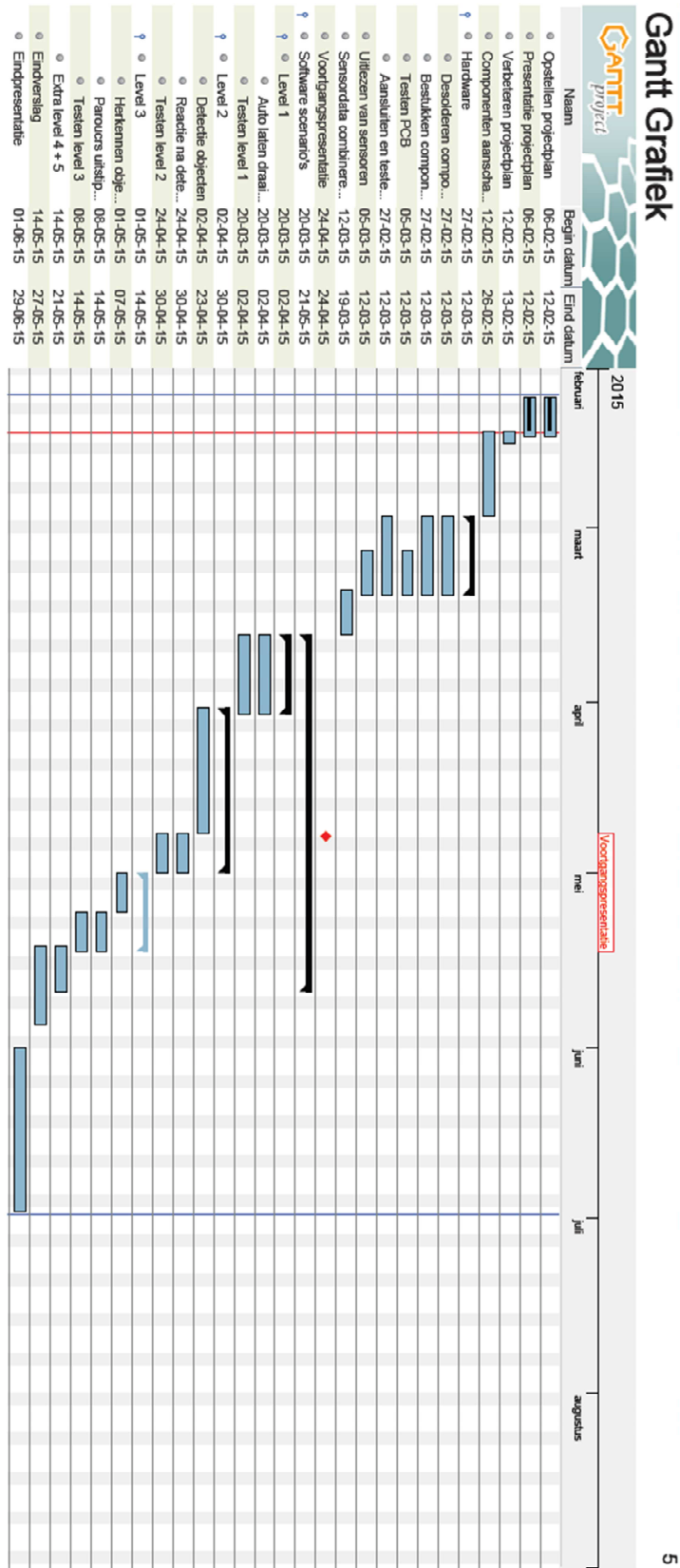
### HW:

- desolderen componenten matrixboard
- bestukken componenten op PCB
- PCB combineren met arduino
- testen PCB
- aansluiten sensoren

### SW scenario's:

- uitlezen van sensoren
- sensordata combineren met sturing auto
- level 1
  - auto laten draaien na bepaalde tijd
- level 2
  - detectie objecten
  - reactie bij detectie objecten
- extra : level 3
  - extra sensor ? (lichtsensor, hall ??)

6.3 Gantt – Chart





## **6.4 Software analyse**