****

**Project Elektronica 4**

Projectplan

**Academiejaar 2014-2015  
Semester 2**

***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 -** [**marc.smets@ap.be**](mailto:marc.smets@ap.be)

1. **Inhoudsopgave**

Inhoudsopgave

[**1.** **Inhoudsopgave** 2](#_Toc420707047)

[**2.** **Inleiding** 3](#_Toc420707048)

[**3.** **Probleemstelling** 4](#_Toc420707049)

[**4.** **Probleem oplossing** 5](#_Toc420707050)

[**4.1** **Projectplan & ganttchart** 5](#_Toc420707051)

[**4.1.1** **Taakverdeling** 5](#_Toc420707052)

[**4.1.2** **Verbeteren projectplan** 6](#_Toc420707053)

[**4.2** **Hardware** 6](#_Toc420707054)

[**4.2.1** **Informatie vergaren & componenten verzamelen** 6](#_Toc420707055)

[**4.2.2** **Desolderen matrixboard, bestukken en testen PCB** 7](#_Toc420707056)

[**4.2.3** **Plaatsing sensoren** 8](#_Toc420707057)

[**4.3** **Software** 8](#_Toc420707058)

[**4.3.1** **Uitlezen sensoren** 8](#_Toc420707059)

[**4.3.2** **Verschillende Levels** 9](#_Toc420707060)

[**4.3.2.1** **Level 1** 10](#_Toc420707061)

[**4.3.2.2** **Level 2** 10](#_Toc420707062)

[**4.3.2.3** **Level 3** 10](#_Toc420707063)

[**5.** **Dankwoord** 11](#_Toc420707064)

[**6.** **Bijlagen** 12](#_Toc420707065)

1. **Inleiding**

De opdracht van semester 3 bestond erin om de auto vanop afstand in alle richtingen te kunnen sturen. Om dit te bereiken was er de nodige kennis noodzakelijk. Voor de aansturing van de motoren moesten we ons verdiepen in de leerstof van H-bruggen, mosfets en transistoren. Hiernaast was ook de werking van IR afstandsbediening en ontvanger noodzakelijk en natuurlijk moesten we ook beschikken over een algemene en een uitgebreide kennis over aansluiten en programeren van een Arduino board.

Aangezien we in project elektronica 3 de volledige sturing in orde gebracht hebben, kunnen we hier in project elektronica 4 op verder bouwen. Onze opdracht voor dit semester bestond er namelijk in de auto autonomisch te laten rijden. Wat betekent dit nu juist? Met behulp van sensoren moet de auto een parcours kunnen rijden en eventuele hindernissen kunnen ontwijken.

De opdracht bestaat uit 5 verschillende delen die levels genoemd worden. Voor het eerste level zijn nog geen sensoren nodig. Dit level bestaat er namelijk in om een op voorhand bekend parcours af te leggen. De auto wordt m.a.w. voorgeprogrammeerd om een bepaald parcours af te leggen. Voor het 2de level is er wel nood aan sensoren. Bij dit level is het de bedoeling dat de gegevens verkregen van de sensors geïnterpreteerd worden en dat zo gedecteerd kan worden waar de hindernissen zich bevinden.Vervolgens moet de software dan beslissen over welke kant de auto moet opdraaien. Uiteindelijk moet de auto zonder hulp een onbekend parcours kunnen afleggen. Bij level 3 moet de auto een “speciaal” object kunnen detecteren, in ons geval is dit een spotlamp. Indien de auto het object bereikt heeft moet hij dit laten weten. Hoe is zelf te kiezen. Mogelijkheden zijn : een muziekje laten afspelen, een vreugdedansje doen en/of een ledje laten branden. Level 4 bestaat uit de combinatie van level 2 en 3. Er moet dus terug een object geïdentificeerd worden. Deze keer bevindt het object zich echter niet meer in een open ruimte, maar in een parcours met hindernissen. De laatste level is hetzelfde als level 4 alleen is er een tijdslimiet toegevoegd. De auto zal dus zo snel mogelijk naar het object moeten rijden terwijl de hindernissen nog steeds ontweken moeten worden

1. **Probleemstelling**

De sensoren vormden 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.  
In ons projectplan vormden we daarom 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?***

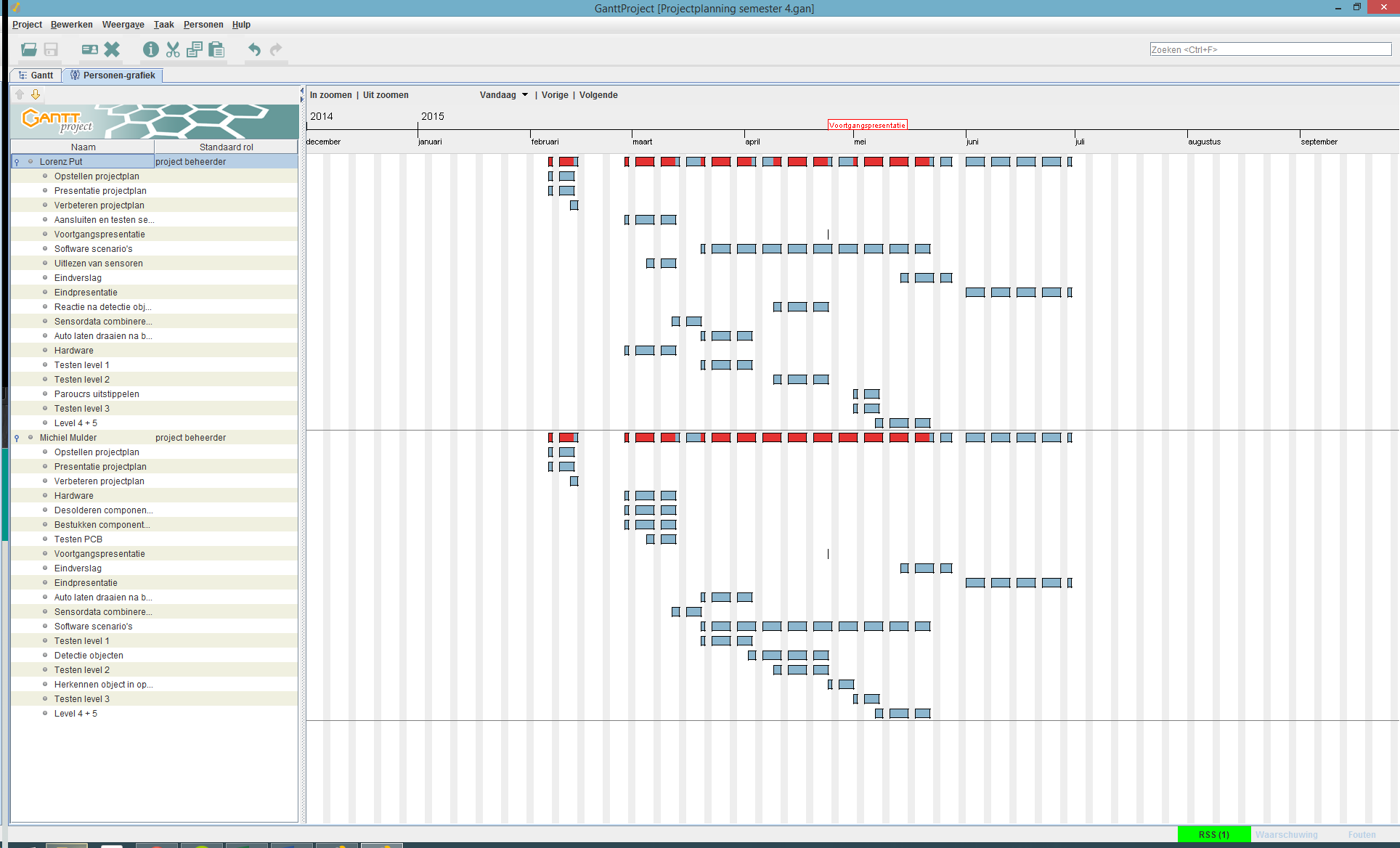
Met deze vraag in het achterhoofd zijn we begonnen aan het project.

1. **Probleem oplossing**
   1. **Projectplan & ganttchart**

Onze eerste projecturen hebben we gewijd aan het opstellen van het projectplan en Gantt-chart. De bedoeling hiervan was dat we op voorhand een overzicht maakten van alle stappen die uitgevoerd moeten worden om het project tot een goed einde te brengen. Deze taak verliep al een stuk vlotter dan bij project elektronica 3 omdat er al iets of wat van ervaring aanwezig was. Er waren echter nog altijd wel moeilijkheden omdat geen van ons twee al eens zo een project had gedaan en we dus niet precies wisten welke stappen er allemaal doorlopen zouden moeten worden

Hoewel met elkaar afspreken niet altijd een sinecure was en dat we daardoor af en toe wel wat achterliepen op schema, heeft het projectplan en de Gantt-Chart er toch voor gezorgd dat het grootste deel van onze opdracht binnen de opgegeven tijdspanne is afgeraakt.

* + 1. **Taakverdeling**

Om het werk te verdelen hebben we de lijn van het vorige semester doorgetrokken, wat wil zeggen: Lorenz gericht op de software en Michiel gericht op de hardware. Natuurlijk is deze opstelling voornamelijk bedoeld als rode draad die een houvast biedt. We beperken ons dus niet slechts tot de eigen vooropgestelde taak. Het voordeel is hier dat ieder specifiek onderzoek kan verrichten en beiden de handen gelijktijdig uit de mouwen kunnen steken zonder elkaar in de weg te lopen. Wanneer de een hulp nodig heeft kan de ander een frisse kijk op de zaak werpen en eventuele problemen uit de wereld helpen.  
  


* + 1. **Verbeteren projectplan**

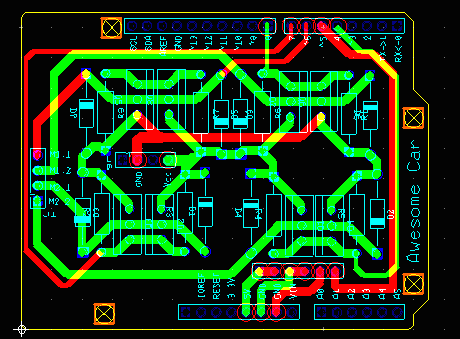
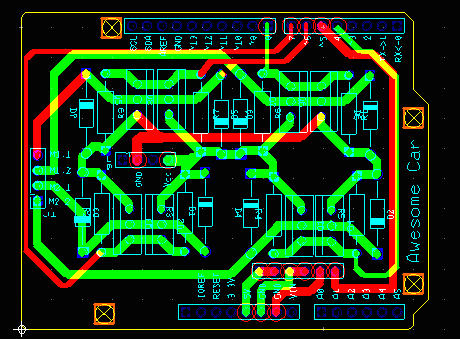
Na de eerste presentatie van onze eerste schets van het projectplan en alvorens aan de slag te gaan hebben we ons projectplan verder op punt gesteld. Dit hield in: de taken iets nauwkeuriger verdelen en de scenario’s van de levels beter in kaart brengen. Het laatste hebben we voorgesteld in de vorm van Nassi Schneidermann-schema’s.

* 1. **Hardware**
     1. **Informatie vergaren & componenten verzamelen**

Bij het hardware gedeelte bouwden we verder op de funderingen die we in het vorige semester tijdens het eerste deel van het project gelegd hadden. De H-bruggen getest en werkend en een verse, componentloze PCB hadden we dus al. De kersverse PCB bestukken en informatie betreffende de sensoren verzamelen lag ons nog voor de boeg.  
We begonnen met het overwegen van de verschillende sensoren. In het voorgaande semester hadden we reeds gewerkt met infraroodsensoren, maar niet met ultrasoon sensoren. Aangezien de infraroodsensoren problemen kunnen ondervinden bij werking in het licht en de ultrasone kleuronafhankelijk zijn kozen wij voor de laatste.

* + 1. **Desolderen matrixboard, bestukken en testen PCB**

Zoals hierboven vermeld zaten we met een kersverse, lege PCB geprint naar ons eigen ontwerp uit het vorige semester. Dit was voor ons een spannende zaak omdat er veel fout kan gaan bij het ontwerpproces. Van breadboard naar matrixboard, naar printed circuit board. Aangezien er toch een aantal regels bestaan bij het ontwerpen van een PCB en de banen op een doordachte manier geplaatst moeten worden kan er veel mislopen. We wilden er zeker van zijn dat er geen fouten bij de componenten zouden liggen kozen we ervoor alle componenten nieuw aan te schaffen. We hebben deze dus niet van het matrixbord verwijderd zodat we ten eerste het matrixbord nog als back-upplan zouden hebben en ten tweede dat we er zeker van konden zijn dat er geen componenten stuk zouden kunnen gaan bij het desolderen. (Alsook waren er enkele pootjes van componenten reeds stevig gebogen en geplooid.) Alle componenten zijn getest en gemeten alvorens deze te bevestigen op de PCB.  
Eenmaal alle componenten zorgvuldig aangebracht waren kregen we het bij het testen even benauwd: de auto reed slechts met 1 wiel (rupsband) tegelijk. Hierop volgde dus veel gepeins, trouble-shooting en herbekijken van de PCB-ontwerpen. Het probleem lag bij een verkeerde verbinding in het ontwerp van de PCB te zien op onderstaande afbeeldingen.



Op de linkse afbeelding zien we het oorspronkelijke ontwerp waar we per ongeluk een baan verkeerd aangesloten hebben op het foute deel van de foute H-brug. Op de rechtse afbeelding is te zien hoe we het probleem hebben aangepakt. (Probleem: rode baan vertrekkende vanaf digitale pin 6, rechts bovenaan.) We waren verheugd om te zien dat na deze aanpassing de auto keurig functionneerde.

* + 1. **Plaatsing sensoren**

We hebben er voor gekozen om 3 ultrasoon sensoren te gebruiken om de auto zich te kunnen laten situeren in zijn omgeving: 1 vooraan, en 2 opzij. Een probleem dat we hebben ondervonden is de dode hoek die bestaat tussen de voorste sensor en deze aan de zijkanten. Dit hebben we zo goed mogelijk proberen op te lossen door de afstanden voor correctie van de koers in te stellen op waardes die een botsing in de dode hoek voorkomen. Dit houdt in: de afstanden aan de zijkanten ver genoeg zodat de dode hoek niet tot aan de muur geraakt. Dit gebeurde door verschillende afstanden te testen, zowel vooraan als opzij.  
De sensor vooraan hebben we ook met een kleine hoek omhoog geplaatst zodat de auto de brug niet als obstakel aanschouwt.

* 1. **Software**
     1. **Uitlezen sensoren**

Voor het opbouwen van de software is er in verschillende stappen gewerkt. We zijn vrij eenvoudig begonnen met het uitlezen van de sensors. We zijn gestart met het uitlezen van de ultrasoonsensor. Hiervoor hebben we beroep gedaan op voorbeeldcode op het internet omdat we hier zonder hulp niet in geslaagd zouden zijn. Het programma zet eerst de trigpin op laag, wacht dan 2 mircoseconden en zet dan trigpin op hoog, daarna wacht het programma 10 mircoseconden alvorens de trigpin terug op laag te zetten. Na dat dit gebeurt is wordt er gemeten hoe lang het duurt alvorens we het signaal terug ontvangen op de echopin. Als laatste stap wordt er op de duur een berekening toe gepast zodat we de afstand bekomen. Wat gebeurt er juist in de ultrasoonsensor ? Terwijl dat de trigpin op hoog staat wordt er een pulstrein verstuurd vanaf de ultrasoonsensor. Deze pulstrein weerkaatst op een object en komt dan terug. Dit tijdsverschil tussen de 2 bepaald dan afstand tussen de sensor en het object.

Om het blackbox principe toe te passen hebben we deze code verplaatst naar een biliotheek. Hierdoor staat er minder code in het hoofdbestand en wordt het overzichtelijker.

Vervolgens zijn we begonnen met uitlezen van de infraroodsensor. Deze sensor wordt in het finale design niet gebruikt, maar vermits hier toch enige tijd mee gewerkt is, wordt er toch wat uitleg over gegeven. Voor het uitlezen hebben we gebruik gemaakt van onze kennis uit mircocontrollers 3. In dit opleidingsonderdeel is er namelijk al met de infraroodsensor gewerkt en hebben we zelf een formule ontwikkeld om de afstand te berekenen. In het programma wordt eerst een analogread gedaan op de distancepin van de infraroodsensor. Daarna wordt de bekomen waarde omgezet naar volt. Hiermee kunnen we dan de afstand tussen het object en de sensor bepalen. Ook hier hebben we onze code overgebracht naar een bibliotheek. Dit om dezelfde reden als hierboven reeds vermeld.

* + 1. **Verschillende Levels**

Nadat de mogelijkheid om de sensordata uit te lezen er was, konden we beginnen aan de verschillende levels te integreren. Tijdens dit schrijven is level 3 nog niet volledig functioneel en zijn de levels 4 en 5 niet aanwezig wegens tijdsgebrek. Over level 3 gaan we in ons verslag toch een uitleg geven omdat we hier reeds veel tijd in gestoken hebben.

Vanaf de derde week is er begonnen aan het schrijven van de software. We hadden het idee om 1 groot programma te schrijven en dan vanop afstand te kunnen bepalen welk level er uit gevoerd wordt. De bedoeling was om onze reeds geïntegreerde IR afstandsbediening te gebruiken.Dus als de gebruiker op het cijfer 1 duwt zou level 1 uitgevoerd moeten worden. Dit geldt ook voor level 2 en 3. Daarnaast zouden we de manuele controle willen behouden dus als de gebruiker op \* duwt zou hij de auto terug moeten kunnen besturen met de pijltjes op de afstandsbediening. Verder was er het plan om de auto draadloos te kunnen doen stoppen. Dit plan kwam er omdat we het zeer onhandig vonden om elkeer de batterij van de arduino of de batterijen van de auto uit te moeten trekken. Dit verliep echter niet van een leien dakje en daardoor waren we genoodzaakt om ons idee een beetje af te zwakken. Momenteel is het mogelijk om de auto te laten starten in een bepaald level door op het juiste cijfer te duwen en de auto te laten stoppen tijdens het uitvoeren van een level.

Hoe gebeurt dit alles ? In de loop functie van de arduino gebruiken we if statements die de ontvangen IR codes vergelijken met de codes voor de verschillende knoppen. Dus als de code 16738455 binnen komt van de IR receiver weten we dat er op 1 geduwt is en wordt er een boolean in dit geval voor level 1 op true gezet. Deze booleans worden dan ook nog eens gecheckt in een latere reeks if statements en indien een van deze statements waar is wordt het desbreffende level uitgevoerd. De reden waarom we booleans werken is omdat anders de levels maar 1 keer uitgevoerd worden en dit niet volstaat. Vooral voor level 2 en 3.

Door problemen met de hardware waardoor de software grotendeels niet getest kon worden, is er altijd maar verder geschreven aan de software waardoor we op den duur met een groot, ongetest programma zaten. Eens de hardware operationeel was konden we het grote programma testen. Natuurlijk werkte dit niet van de eerste keer en waren we genoodzaakt om ons groot programma terug op te splitsen in kleinere delen en deze één voor één te testen alvorens deze terug te integreren in het hoofdprogramma. Dit euvel zorgde wel voor een hoop tijdsverlies.

* + - 1. **Level 1**

Level 1 bestaat erin om de auto een op voorhand bekend parcours te laten rijden. Dit wordt gedaan door programmatorisch te bepalen hoe lang de auto rechtdoor moet rijden, naar links/rechts moet draaien. Om de duur van het manoeuvre te bepalen gebruiken we delays. Deze delays zorgen ervoor dat het programma gedurende bepaalde tijd “slaapt”. Vermits dat de correcte pinnen op hoog of laag gezet worden alvorens de delay wordt uitgevoerd blijven ze dus in die status zo lang de delay loopt.

* + - 1. **Level 2**

Level 2 bestaat erin om met behulp van de gegevens van de sensoren een parcours te rijden terwijl eventuele hindernissen ontweken worden. het algoritme dat hiervoor gebruikt wordt bestaat uit verschilende if en else structuren en gaat in de volgende alinea stap voor stap uitgelegd worden.

Het algoritme begint met de voorste sensor uit te lezen. Indien de waarde hiervan niet kleiner als 10 cm word er gekeken of de afstand links en rechts niet groter is dan 7. Is één van deze 2 waar dan draait de auto de andere richting uit om de rijrichting te corrigeren. Als deze if statements allemaal niet waar zijn, rijdt de auto vooruit.

Indien de afstand vooruit wel kleiner is als 5 wordt er gecheckt of de afstand links en rechts groter is dan 7. Indien dit niet het geval is, wordt er gecheckt of de afstand achteruit groter is dan 10. Is dit ook niet waar dan gaat de led op pin 13 van de arduino branden en stopt de auto. Is dit wel waar rijdt de auto een bepaalde afstand achteruit en wordt er terug gekeken of de afstand links en rechts nu groter zijn dan 7 anders wordt het proces herhaald. Indien de afstand achteruit niet groter is dan 10 stopt de auto en gaat de led op pin 13 van de arduino branden.

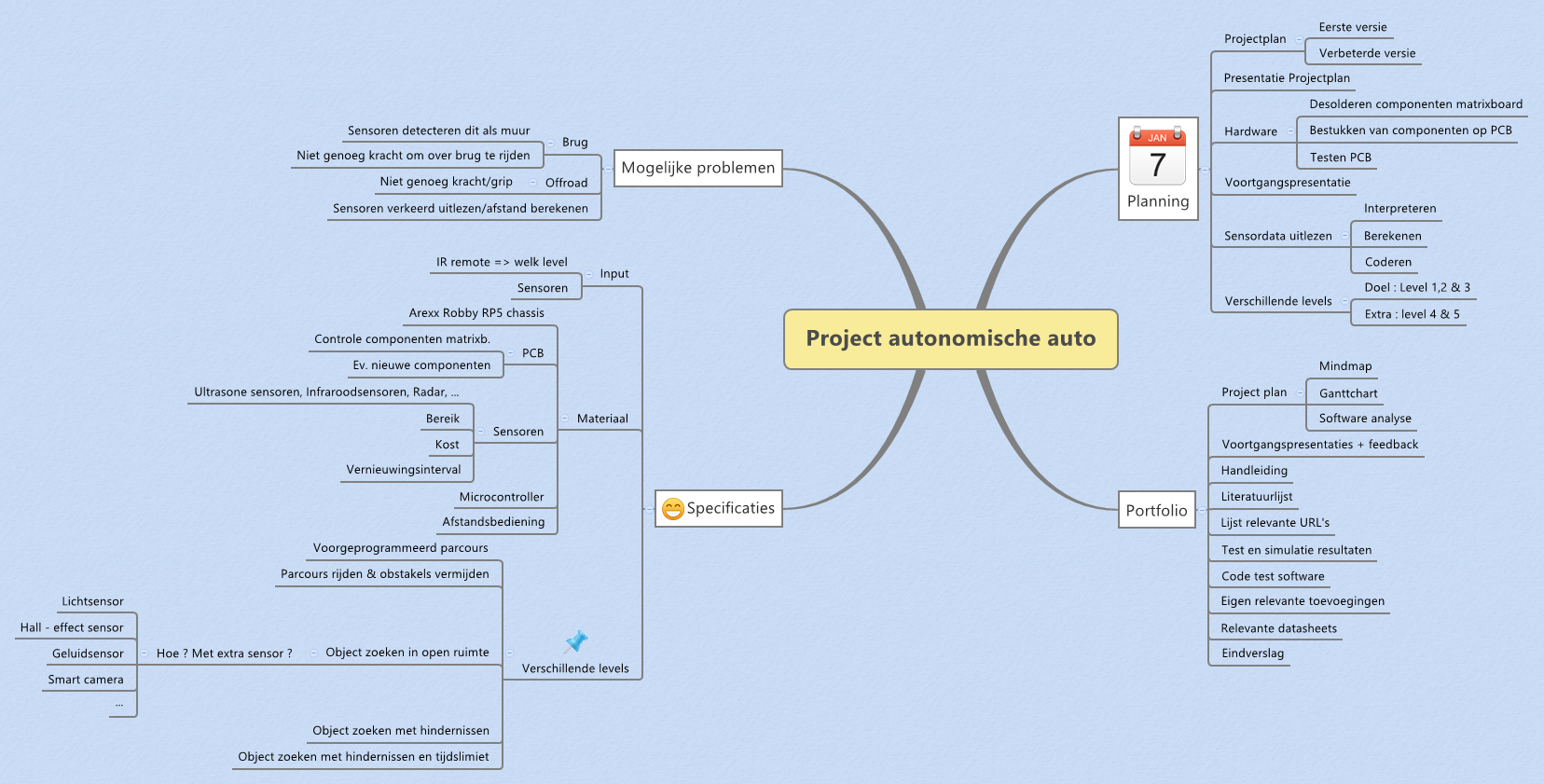
Indien de afstand links of rechts wel groter is dan 7 wordt er gekeken waar de afstand het grootst is. Deze kant wordt dan opgedraaid.

* + - 1. **Level 3**

Level 3 bestaat erin om een bepaald object te zoeken en vinden in een open, vierkante ruimte. Hiervoor hebben we een lamp gekozen die we zoeken aan de hand van 2 light dependent resistors of LDR’s. We hebben hiervoor een LDR links vooraan geplaatst en een andere rechts vooraan en 2 korte plastiek buisjes er over geplaatst om overbodig licht te elimineren, waardoor we de spotlamp nauwkeuriger kunnen vinden. Wanneer er links of rechts een sterkere waarde wordt uitgemeten zal de auto deze kant op draaien en zo, links en rechts vergelijkend de lamp vinden wanneer het signaal krachtig genoeg is. Hier zijn we een beetje afgeweken van de vooropgestelde analyse.

1. **Dankwoord**

Wij willen graag Marc Smets en Patrick Van Houtven bedanken voor de informatie die zij ons meegegeven hebben in de voorbije semesters, om dit project tot stand te kunnen brengen. Eveneens danken wij hen voor de ervaring die wij dankzij dit project hebben kunnen opdoen. In het bijzonder ook dank aan Marc Smets om ons als projectleider bij te staan.  
Ook danken wij graag Jeroen Doggen voor de kennis die we hebben gekregen om te werk te kunnen gaan met microcontrollers.

1. **Bijlagen**
   1. Mindmap
   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

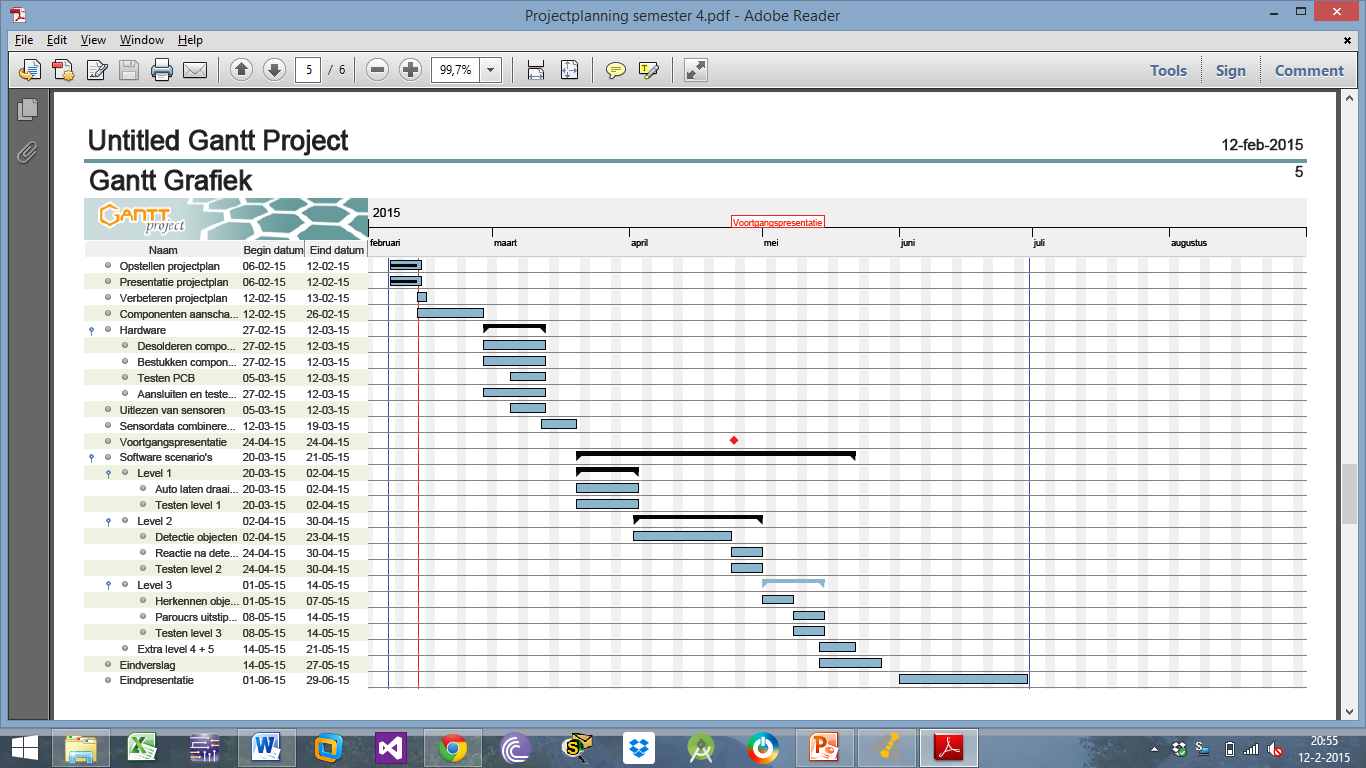
- level 2

-- detectie objecten

-- reactie bij detectie objecten

- extra : level 3

-- extra sensor ? (lichtsensor, hall ??)

* 1. Gantt – Chart
  2. Software analyse

