

RoboX

EINDVERSLAG SMART SYSTEMS

Samuel Baudez, Jon Santacruz en Gianni Lenaerts
2EA1 | 2016-06-05

1.Inhoud

1.Inhoud	1
2. Voorwoord.....	2
3. Planning.....	3
3.1 Backlog	3
3.2 Sprint 1	4
3.4 Sprint 3	5
3.5 Sprint 4	5
4. Hardware	7
4.1 Arduino MEGA 2560	7
4.2 H-Brug.....	8
4.2.1 bouw.....	8
4.2.2 werking	9
4.3 Sensoren.....	10
4.3.1 Ultrasonische Sensoren	10
4.3.2 IR Sensoren.....	11
4.4 matrix board en PCB.....	12
4.5 frame	15
4.6 Bluetooth.....	16
5. Software	17
5.1 Arduino code	17
5.2 Android app.....	19
.....	19
6. Communicatie	19
6.1 Online	19
6.1.1 Skype	19
6.1.2 Trello.....	20
6.1.3 Github/Smartgit.....	20
6.2 Meetings.....	20
7. Besluit	20
8. Bronvermelding	21

2. Voorwoord

Als software- en hardware team hebben we ons de afgelopen maanden beziggehouden met een zeer uitdagend, maar leerrijk project. Na maanden code schrijven, verwijderen, herschrijven, hardwarecomponenten monteren en demonteren kon uiteindelijk ons robotje op eigen benen lopen of beter gezegd op eigen rupsbanden rijden.

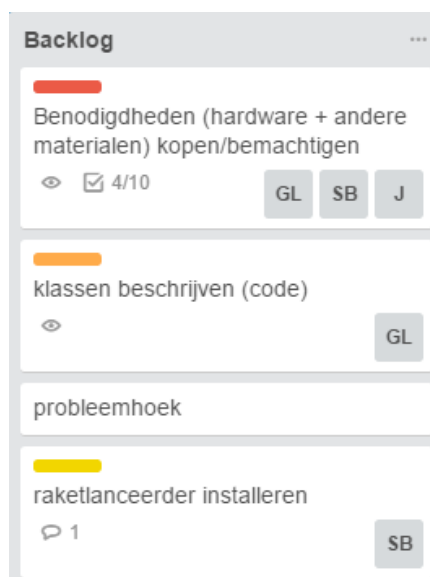
De naam van onze robot en project is uiteindelijk RoboX worden. De inspiratie kwam van het bedrijf “SpaceX”, een Amerikaans bedrijf voor ruimtetransport van de wereldberoemde CEO, Elon Musk. Dit was voor ons de meest geschikte naam, die best wel ‘catchy’ is, daarom besloot het hele team unaniem voor deze naam te kiezen.

In dit verslag zal er uitgelegd worden hoe we de robot hebben gemaakt, welke technologieën, communicatiemiddelen we toegepast hebben en hoe onze planning en meetings verliepen.

3. Planning

De planning van ons project werd vooral gestuurd via Trello. Hier werd de backlog, een lijst van ideeën, opgemaakt. Daarna werden deze punten stap voor stap verwerkt in de sprints. Het project bestond in totaal uit 4 sprints. Elke sprint had een deadline, waarin verschillende taken in orde moesten zijn. Op deze manier werd het hele project op een zeer efficiënte manier uitgevoerd.

3.1 Backlog



Snapshot van onze backlog.

In deze lijst werden al onze ideeën na het brainstormen verwerkt. De lijst bevatte alles wat de robot nodig had. Dit ging van de meest fundamentele onderdelen zoals de SN754410 motorcontroller tot de minst noodzakelijke onderdelen zoals een toeter of een raketlanceerinstallatie. Uiteindelijk werden niet alle zaken vanuit de backlog uitgevoerd. Zaken zoals de toeter, raketten enz. werden niet toegepast. Dit komt omdat de opdrachtgever de requirements voor de robot aanpaste. Deze moest namelijk kunnen tekenen. Door het gebruik van een agile methode was dit zeker geen probleem, omdat er in sprints werd gewerkt. Hierdoor konden we kort op de bal spelen en zaken zoals functionaliteiten vrij makkelijk veranderen. Dit in tegenstelling tot het klassieke

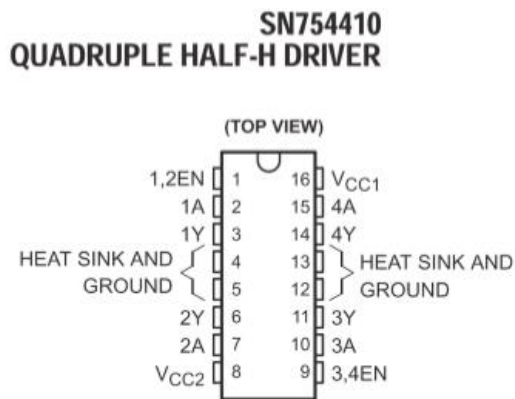
waterval ('*waterfall*') model.

Items in de backlog kregen ook een kleur naargelang de waarde van het item. Zo zal bijvoorbeeld een klasse voor de logica van de robot veel belangrijker zijn dan de code voor de led lampjes en zal deze taak de kleur rood krijgen.



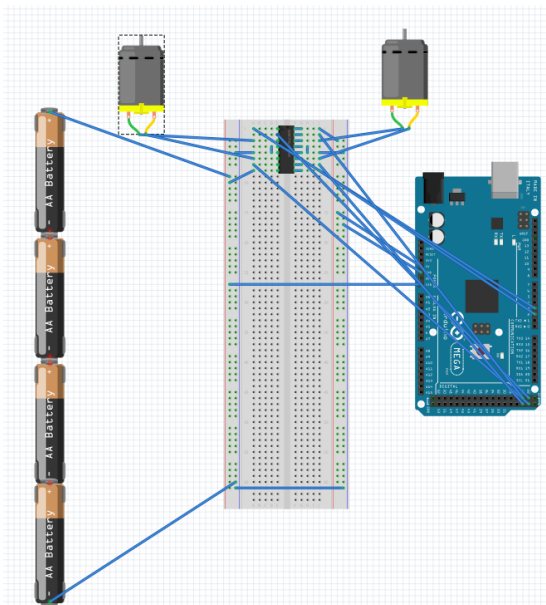
De verschillende prioriteiten en de kleur die erbij past.

3.2 Sprint 1



Schema van de quadrupel halve H-brug.

(beschikbaarheid, gebruiksgemak en vermogen). Zaken zoals batterijen, ledjes, weerstanden en de kracht van de motoren hebben we ook opgezocht.



Schema in Fritzing van onze motorcontroller, die gekoppeld is aan de Arduino MEGA.

Sprint 1 was vooral op hardware gericht. Dit komt omdat we eerst de componenten moesten opzoeken en samenstellen vooraleer we pas aan de software konden beginnen. Eerst en vooral werd de research gedaan. Hierbij werden de meest belangrijke items uit de backlog genomen en begon de research. Zo hebben we opgezocht wat de beste optie qua motorcontroller was, welke functies het moest verwezenlijken. Het SN754410 beantwoordde aan de belangrijkste criteria

Na de research hebben we de taken verdeeld, waarbij elke taak een prioriteit krijgt.

We hebben er al voor gezorgd dat we een kleine schakeling hadden, waaraan de arduino (MEGA 2560) werd gekoppeld. Hierna kon ons autootje al enkele basis manoeuvres, zoals vooruit en achteruit rijden.

Ook in sprint 1 zal het basisframe met input pluggen voor de voeding voltooid zijn.

3.3 Sprint 2

De tweede sprint werd opgesplitst in sprint 2.1 en 2.2. De reden voor deze beslissing is omdat we in sprint 2.1 vooral de basis klaar wouden hebben zoals besturing van de auto via de seriële input (putty.exe + usb kabel) en de basic hardware.

Het software gedeelte was vooral het code voor de basic manoeuvres en snelheden voor de robot. Elke richting kreeg een commando die we dan aanstuurden via de pc. De voor- en achterlampjes werden ook al hierin verwerkt, zo konden de rechter led lampjes aangaan als de auto rechts afslaat en vice versa. Als het autootje stopte moesten de achterste lampjes aangaan enz.

Hierna werd alles in libraries gestoken. Door gebruik te maken van libraries werd de hele code overzichtelijker en eenvoudiger om aan te passen.

De hardware was vooral het basisframe met de outputs in elkaar steken, lampjes installeren, solderen van de bedradingen aan de batterij en testen of alles optimaal werkte.

In sprint 2.2 zal de basis van 2.1 verder worden afgewerkt. Zo kon het autootje bestuurd worden door een Android applicatie in plaats van een kabel dat aan een laptop hing. Hier hebben we gebruik gemaakt van bluetooth technologie om de auto te laten rijden door middel van een controle paneel in de app. In 2.2 werd ook het pcb design ontworpen, een shield gemaakt voor de arduino en kon de bluetooth module getest worden.

Er werd ook al gewerkt aan de sensoren (deze was nog low priority op de moment).

3.4 Sprint 3

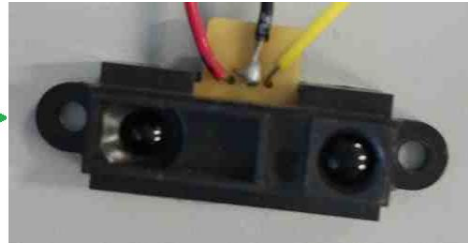
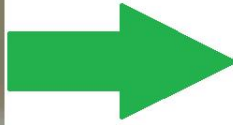
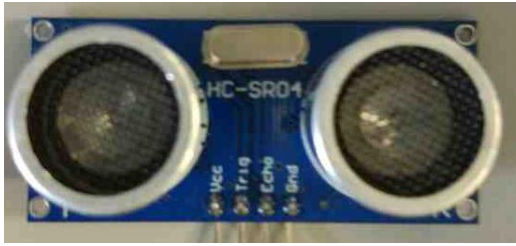
Sprint 3 speelde zich af na de presentaties van het kerstexamen. Hier pasten we weer dezelfde strategie toe. We deelden onze sprint 3 weer op in 3.1 en 3.2. Het verschil met sprint 2 is dat we 3.1 hebben gebruikt voor de software en 3.2 voor de hardware.

In 3.1 werd zoals in sprint 2 de basis verwerkt. Hier schreven we de basiscode voor de sensoren en verwerkten deze in libraries. Ook werd de code van de servomotor geschreven en verwerkt in een library.

3.2 was vooral op hardware gericht. Dit bestond uit het testen van de sensoren en de servomotor.

3.5 Sprint 4

De vierde en laatste sprint stond volledig in het teken van de opdracht “Margin of error”. Sensoren en de servomotor kregen de aandacht voor het hardware deel. Bij het software deel werd er gefocust op de logica voor het tekenpatroon.



Links: ultrasone sensor/ Rechts: IR sensor

In de loop van sprint 3 ondervonden we moeilijkheden met de sensoren. Dit kwam doordat we ultrasone sensoren gebruikten. Met deze sensoren kwamen we zeer onnauwkeurige resultaten uit. Door deze problemen zijn we overgeschakeld naar IR sensoren (infra rood sensoren).

Nadat we de nieuwe sensoren getest hadden, konden we ze implementeren op het autootje. De code werd in een library gestoken en de servo motor code voor de applicatie werd ook voltooid om een stift op en neer te laten bewegen.

De volgende taak was de logica schrijven om het wagentje autonoom te laten rijden.

De laatste taak van sprint 4 was het tijdelijke matrix bord vervangen door de PCB, het schrijven van een flowchart en een sequentiële diagram.

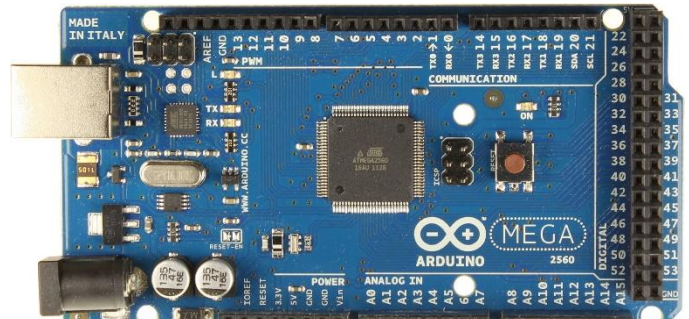
4. Hardware

4.1 Arduino MEGA 2560

Voor dit project hebben we gekozen voor een Arduino MEGA 2560. Deze Arduino heeft vele voordelen voor ons project:

De vele digitale I/O pinnen die belangrijk zijn om zaken aan te sturen zoals onder andere de auto zelf, de leds, de pen enz.

Natuurlijk zijn de analoge pinnen ook zeer belangrijk. Hiermee kon de Arduino zaken aansturen zoals de snelheid, licht intensiteit van de leds enz. De MEGA heeft ook het voordeel van een snellere chip aan boord te hebben.



Arduino MEGA 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Technical specs

Extra info over de Arduino MEGA:

<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

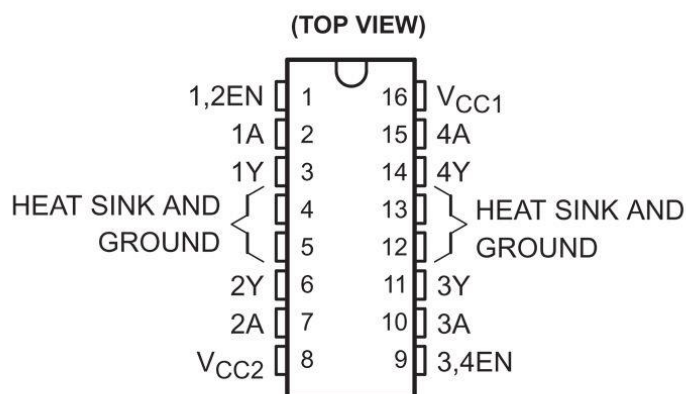
4.2 H-Brug

4.2.1 bouw

Als motor voor ons autootje hebben we gekozen voor een SN754410 Quadruple Half-H DRIVER.

Deze elektische schakeling laat ons toe om een apparaat aan te sturen met motoren zoals in ons project de wielen van de robot.

SN754410 QUADRUPLE HALF-H DRIVER



FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low-level

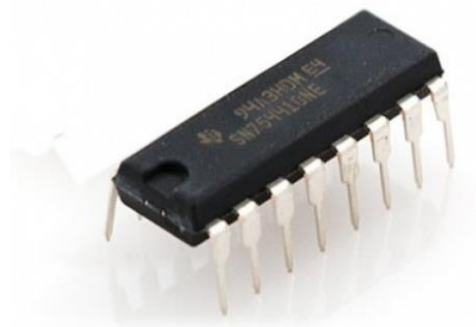
X = irrelevant

Z = high-impedance (off)

† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

Onze H-brug bestaat uit 16 pinnen:

- *EN*: deze analoge pinnen krijgen een waarde die kan variëren van 0 tot 255
- *A*: dit zijn input signalen die net zoals de enable een high of low zal zijn. (3 toestanden: HIGH, LOW, X)
- *Y*: dit zijn de outputs, dit wordt HIGH, LOW of Z (high-impedance en dus betekend dit dat dit off zal zijn)
- *Pin 4,5,12,13*: deze pinnen worden gebruikt als de ground

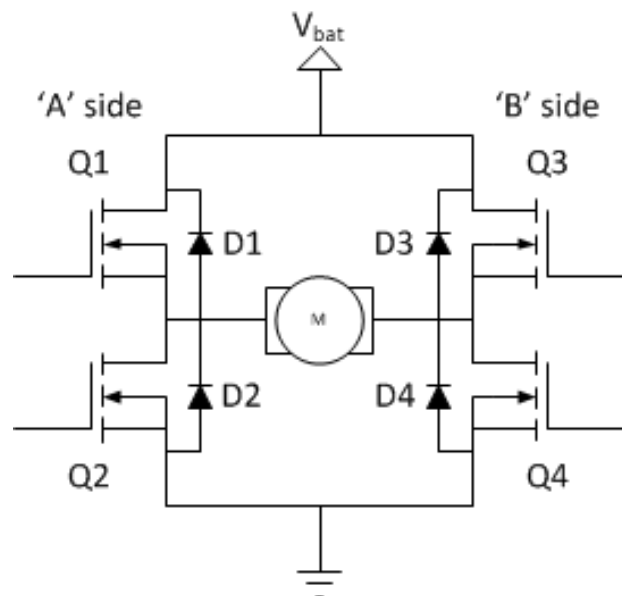


Algemeen gezien is de SN754410 een vrij eenvoudige chip om mee te werken en bestaat uit 4 halve H-bruggen.

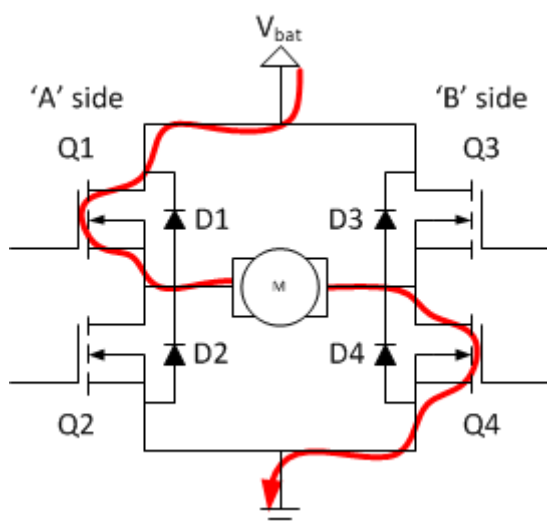
Een H-brug bestaat uit 4 'schakelaars', met de load in het midden. Vandaar de naam H-brug, omdat het de vorm heeft van een H. De 'schakelaars' (Q1..Q4) zijn vaak bipolair of FET transistoren, in sommige applicaties zijn dit IGBT's (high voltage circuits).

De dioden (D1..D4) noemt men 'catch diodes' en zijn vaak van een 'Schottky' type. Aan de bovenkant van het circuit hang meestal de power supply (batterij) en aan de onderkant bevindt zich de ground.

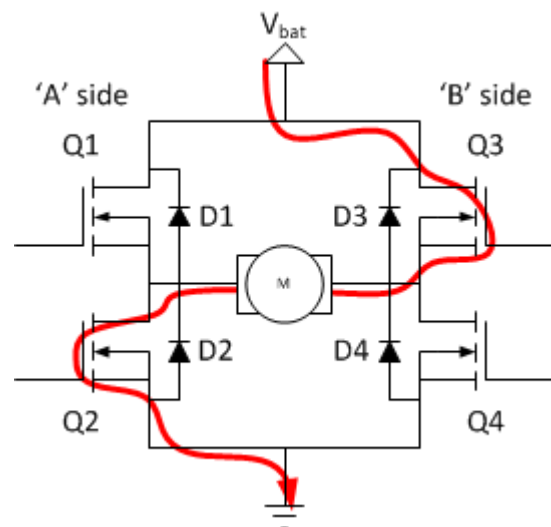
Onze 4 schakelaars kunnen dus onafhankelijk aan en uit gezet worden, maar hier hangen uiteraard beperkingen aan vast.



4.2.2 werking



De basis werking van een H-brug is vrij eenvoudig. Als de schakelaar Q1 en Q4 aan staan (dicht). Zal de stroom door de motor vloeien waardoor de motor 'energie' krijgt. Hierdoor zal de motor vooruit rijden. Als de



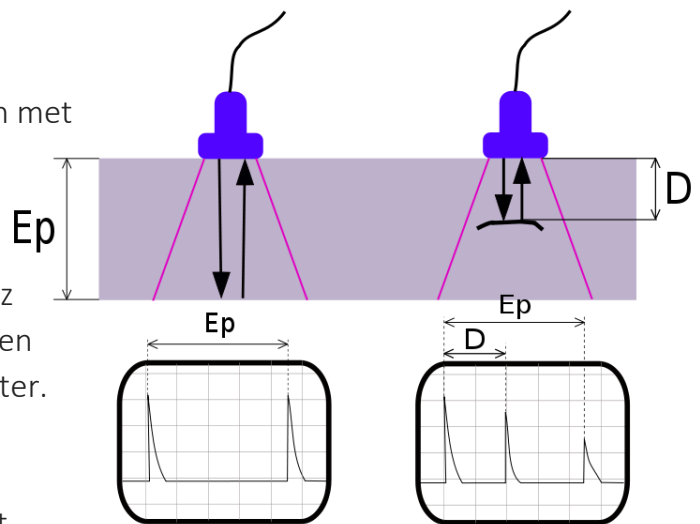
schakelaars Q3 en Q2 aan staan (dicht), zal de stroom langst de andere kant in de motor vloeien. Hierdoor zal de motor in de tegengestelde richting energie krijgen en zal dus in reverse draaien (achteruit rijden).

4.3 Sensoren

Robox is voorzien van drie sensoren (vooraan/links/rechts). We hebben geëxperimenteerd met twee typen sensoren: ultrasone sensoren en infrarood sensoren.

4.3.1 Ultrasone Sensoren

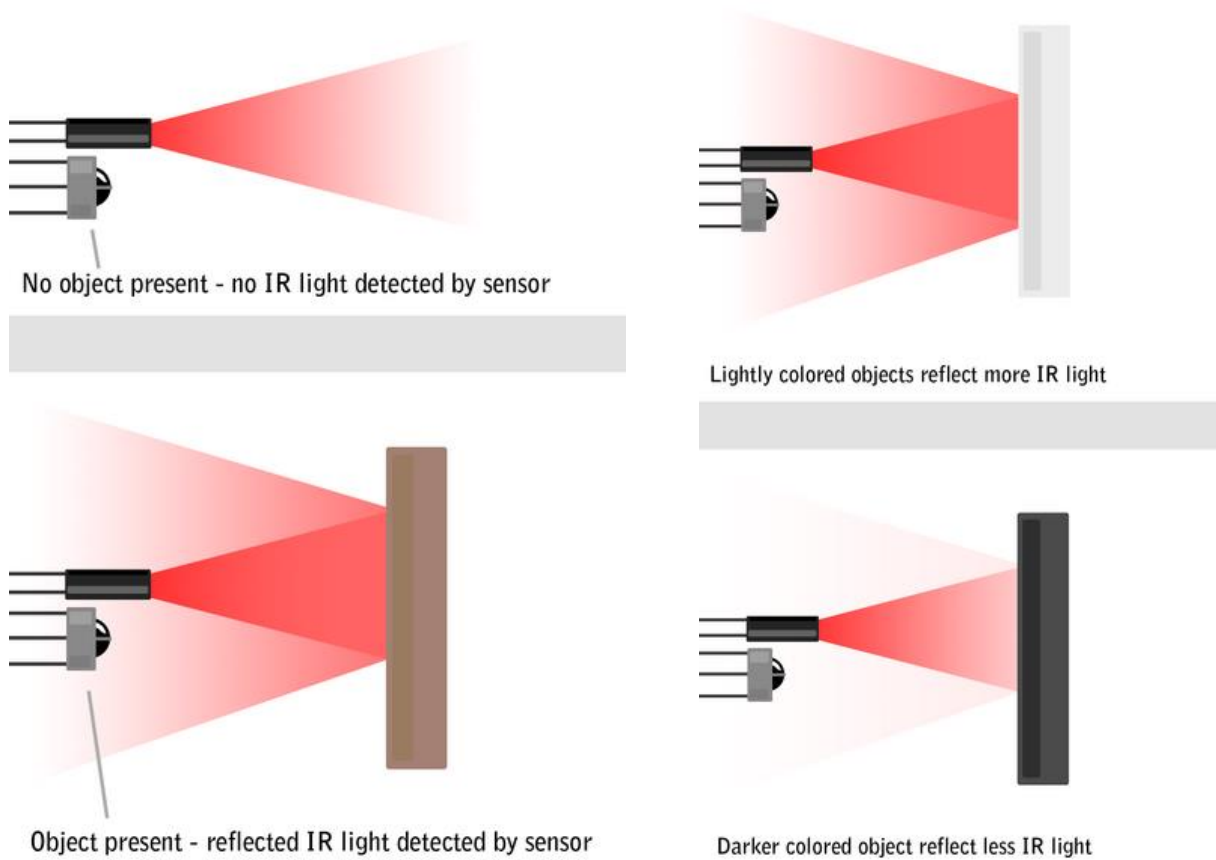
Ultrasoonsensoren zijn sensoren die werken met behulp van geluidsgolven op frequenties hoger dan waarneembaar voor het menselijk oor. Dit gaat om golven met een frequentie hoger dan 18 kHz tot 200 kHz. Bij dergelijke sensoren wordt een ultrasone golf verzonden door een transmitter. Door reflectie op een voorwerp of materie krijgen we een reflectie of echo van deze geluidsgolf. Door het tijdsinterval tussen het verzenden en het ontvangen van de geluidsgolf te meten kan men afstanden bepalen. Afhankelijk van de fysieke afstand tussen transmitter en receiver moet men nog een meetkundige bewerking uitvoeren, zoals triangulatie, om de correcte afstand te bekomen. Ook dient men rekening te houden met het medium waarin de ultrasone golf wordt verzonden (lucht, water, etc.). Elke medium heeft een aparte snelheid waarmee de ultrasone golf zich door het medium verplaatst. Bij lucht bedraagt deze 340m/s bij kamertemperatuur. Deze snelheid is alsook afhankelijk van de temperatuur.



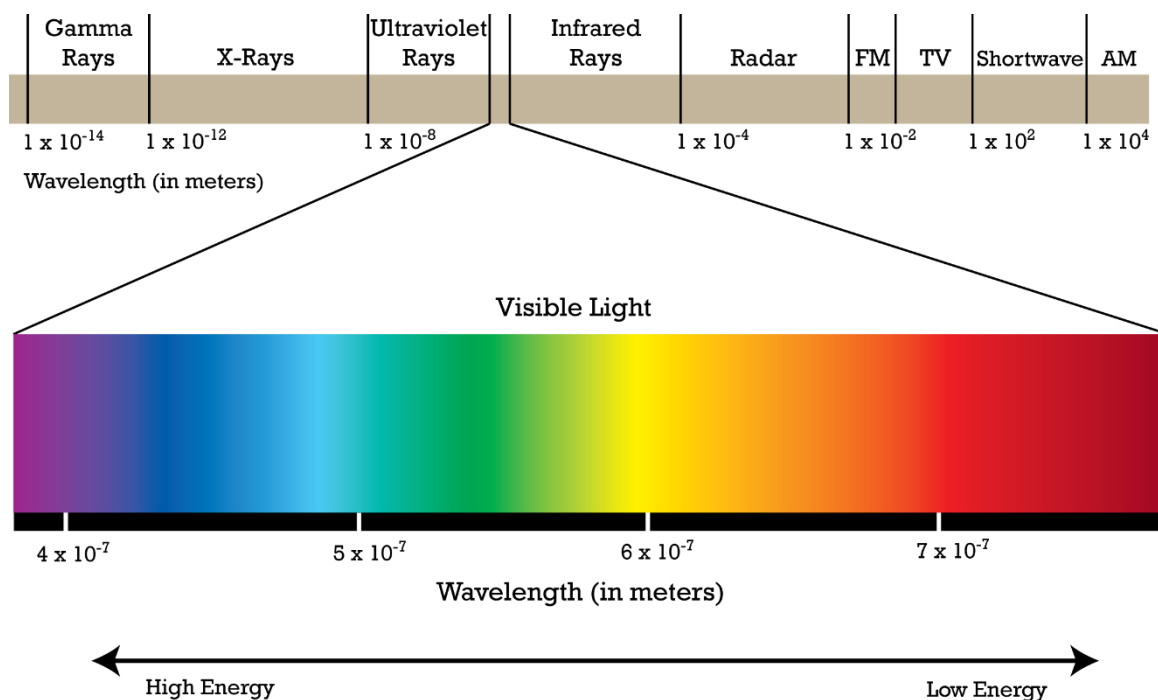
Bij het testen van onze sensoren, kwamen we vaak op zeer onnauwkeurige resultaten uit. Zo zou de sensor niet altijd werken en vaak de verkeerde resultaten weergeven. Door het aanpassen van onze code en het verplaatsen van de sensoren kregen we al iets betere resultaten, maar nog altijd ondermaats.

Hierdoor zijn we overgeschakeld naar IR sensoren.

4.3.2 IR Sensoren



IR sensor werkt door middel van het detecteren van licht (golflengte) dat zich bevindt in het infra rood spectrum. Door het gebruiken van een infra rood led, straalt de sensor deze infra rood stralen uit. Als er een object in de buurt komt van deze led, zal



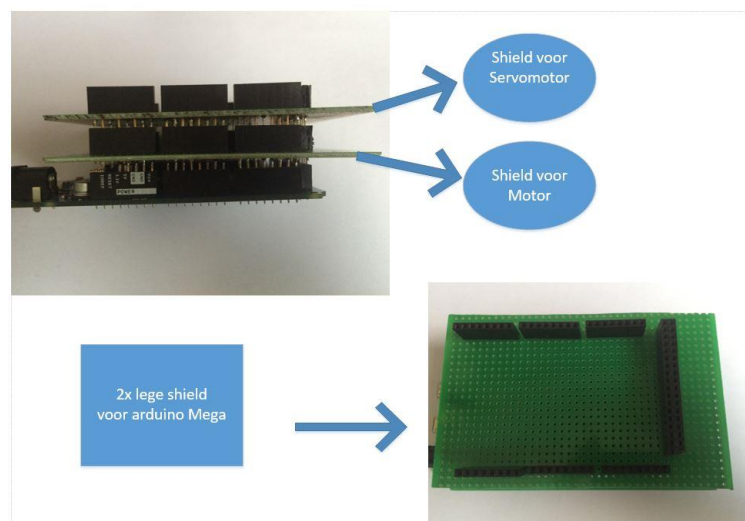
dit object de stralen terugkaatsen. De licht sensor zal deze terug gebotste signalen opvangen. Dit inkomend signaal zal een heel andere intensiteit hebben als het verzonden signaal. Zo kan de sensor meten hoeveel afstand er zich bevindt tussen het object en de sensor.

Doordat we gebruik maken van een infra rood sensor zal de sensor vooral lichte oppervlakken detecteren. Bij donkere oppervlakken zal het terugkaatsend licht minder sterk zijn.

4.4 matrix board en PCB

Fase één, het brainstormen (Samuel/Gianni/John) voor het eerste PCB prototype. Tijdens deze fase is men tot het idee gekomen om meerdere shields te maken. Ons concept van fase één was om meerdere shields op elkaar te bouwen. Deze shields kunnen samen werken en worden aangestuurd door een arduino MEGA. De reden dat men hiervoor koos, was dat er op het eerste zicht de benodigde werkruimte op de PCB boord niet geschat kon worden. En ook hadden wij in het achterhoofd dat er indien hardware fouten optraden, deze aangepast kon worden zonder invloed te hebben van de werkende shield.

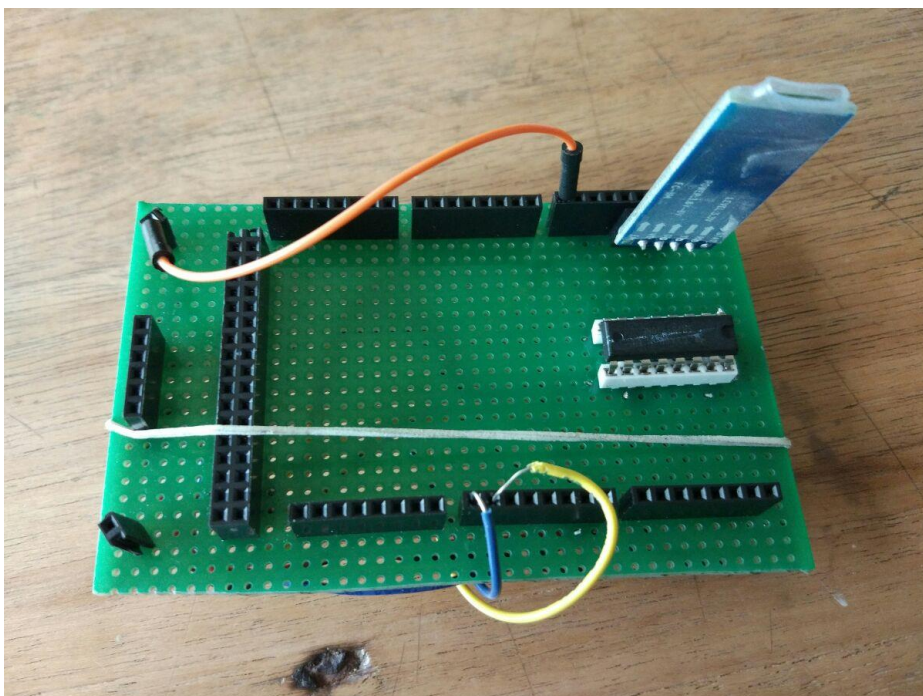
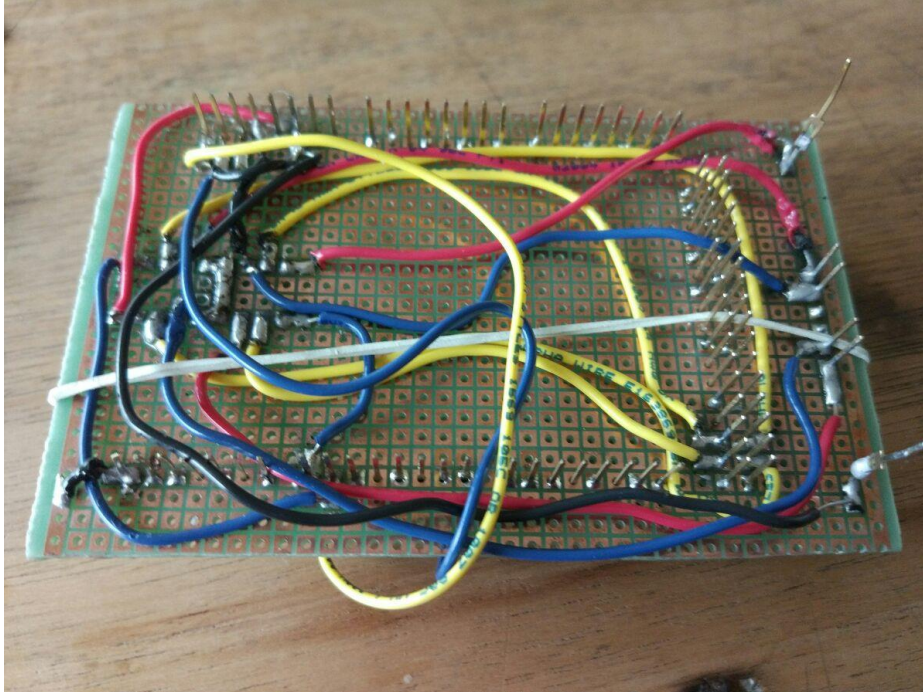
In het slechtstse geval bijvoorbeeld als het fout gesoldeerd werd en deze niet meer herstelbaar was, kon men deze vervangen door een nieuwe shield.



Tijdens het bouwen van de prototype shield van roboX zijn wij tot de conclusie gekomen dat er in feite minder werkruimte nodig was dan we hadden voorzien op het PCB board. De docent geeft ons een mededeling gegeven met als feedback dat alles in één shield kon gebeuren. Met deze conclusie en de mededeling van de docent hebben wij besloten om toch alles op slechts één shield te implementeren.

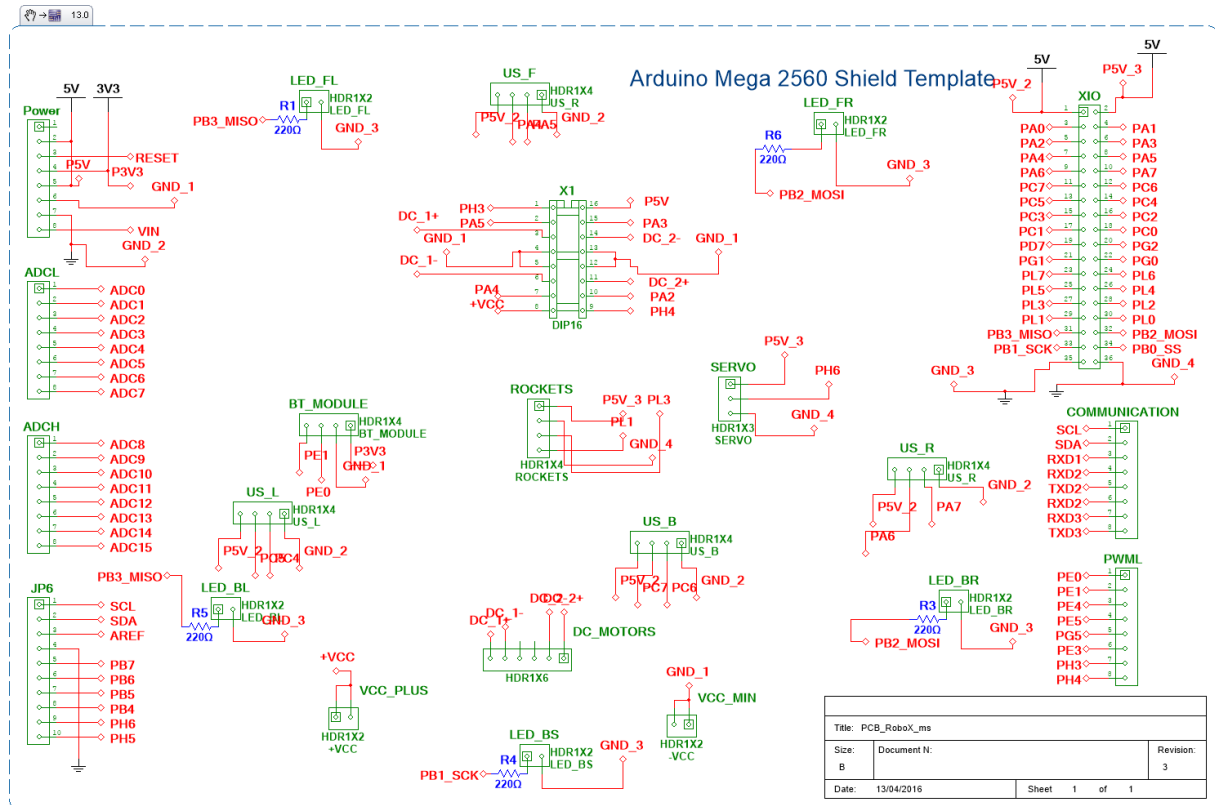
Deze shield is nog voorzien van lange pinnen die ingeplugd kunnen worden op het transparante frame van roboX. Dat maakte ons eerste prototype als het ware plug-and-play.

Het overgebleven lege shield werd als reserve onderdeel bewaard.

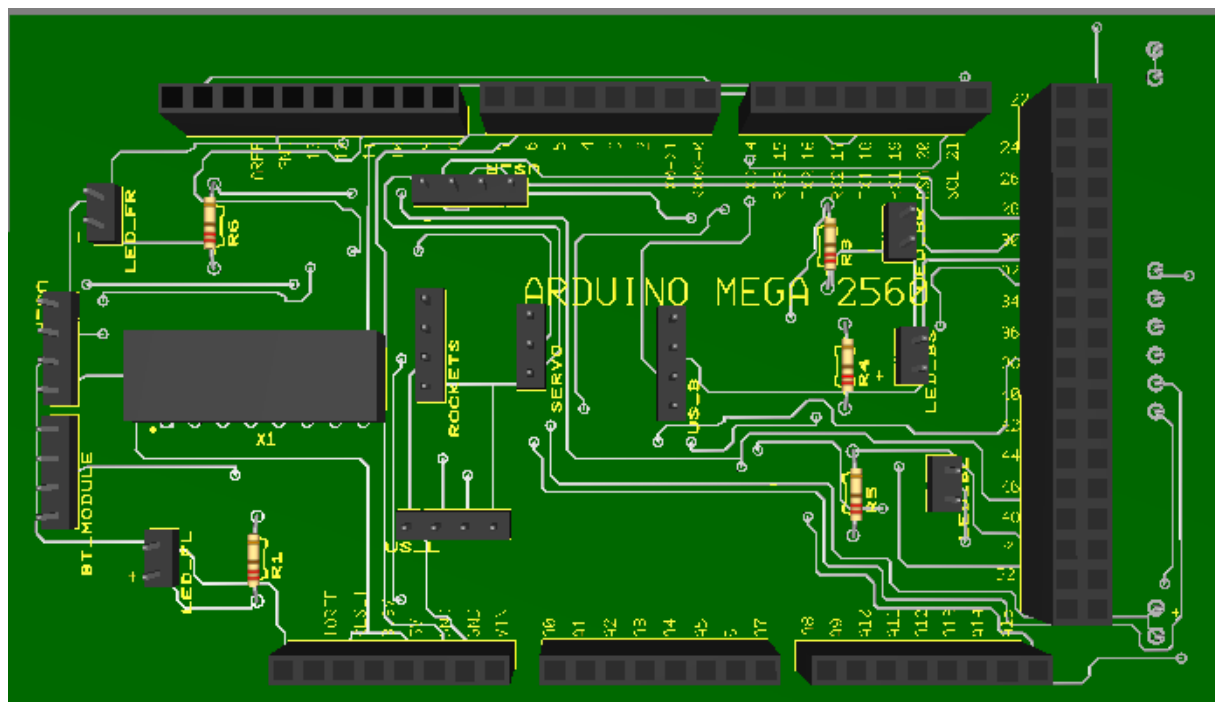


Het PCB design

Bij het ontwerp van het definitieve PCB design werden er connectoren voorzien voor elke mogelijke toevoeging die we tot dan toe in gedachten hadden. Helaas zijn er veel connectoren ongebruikt gebleven door een wijziging van de opdracht (en van sensoren).



Figuur 1PCB multisim



4.5 frame

Als goep hebben wij aan een frame bedacht dat ervoor zorgt geen draden door elkaar te mengen.

Het frame is een transparant deksel dat input pluggen bevat. Het frame zorgt ervoor dat de draden van de motoren, voeding voor de motoren en de voeding van de arduino onder het deksel blijven. Tijdens het sleutelen of bij het upgraden van roboX konden de draden niet meer storen. Het voordeel hiervan is dat wij op deze manier de mogelijke fouten minimaliseren en indien er toch een fout optreedt, deze sneller vinden.



Het deksel bevat aan het ene zijde een input aansluiting waar de arduino MEGA ingeplugd kan worden, deze aansluitingen zijn voorzien voor zowel de linkse(A) als de rechtse(B) motor en heeft aan de buiten zijde een voeding aansluiting. De voeding aansluiting zorgt ervoor dat de motoren worden gevoed.



Aan de andere kant van het deksel zijn ook input pluggen voorzien waarbij een extra voeding aangesloten kan worden, deze voeding zal een 9 volt batterij zijn. Deze batterij heeft als doel de arduino mega te voeden. In principe is het laden van de batterij mogelijk zonder het deksel te verwijderen (batterij lader aansluiten aan de input pluggen).

De input pluggen hebben wij als groep bedacht, deze is gemaakt uit jumper draden die in de labo lokaal ter beschikking zijn. De arduino jumper kabel man-vrouw is gekozen voor deze toepassing en is aangesloten als volgt:

Man(pin zijde van de kabel)= gesoldeerd met de draden van de motoren en batterijen.



Vrouw (plug zijde van de kabel)= Deze uiteinde van de jumper kabel wordt gebruik als input plug op het frame van RoboX.

Het bevestigen van deze pluggen op het frame van RoboX en het vast lijmen gebeurde buiten de labo sessies met eigen apparatuur.

4.6 Bluetooth

Vanaf het begin wilden we een robot die vanop afstand bestuurbaar was. We hebben niet geaarzeld om meteen een bluetooth module aan te schaffen om er zo snel mogelijk mee te beginnen. Het grootste voordeel van de HC-06 module is dat er voor de arduino geen verschil is tussen commando's die afkomstig zijn van een USB kabel of een draadloze bluetooth module.



Figuur 2HC-06 BT module

5. Software

5.1 Arduino code

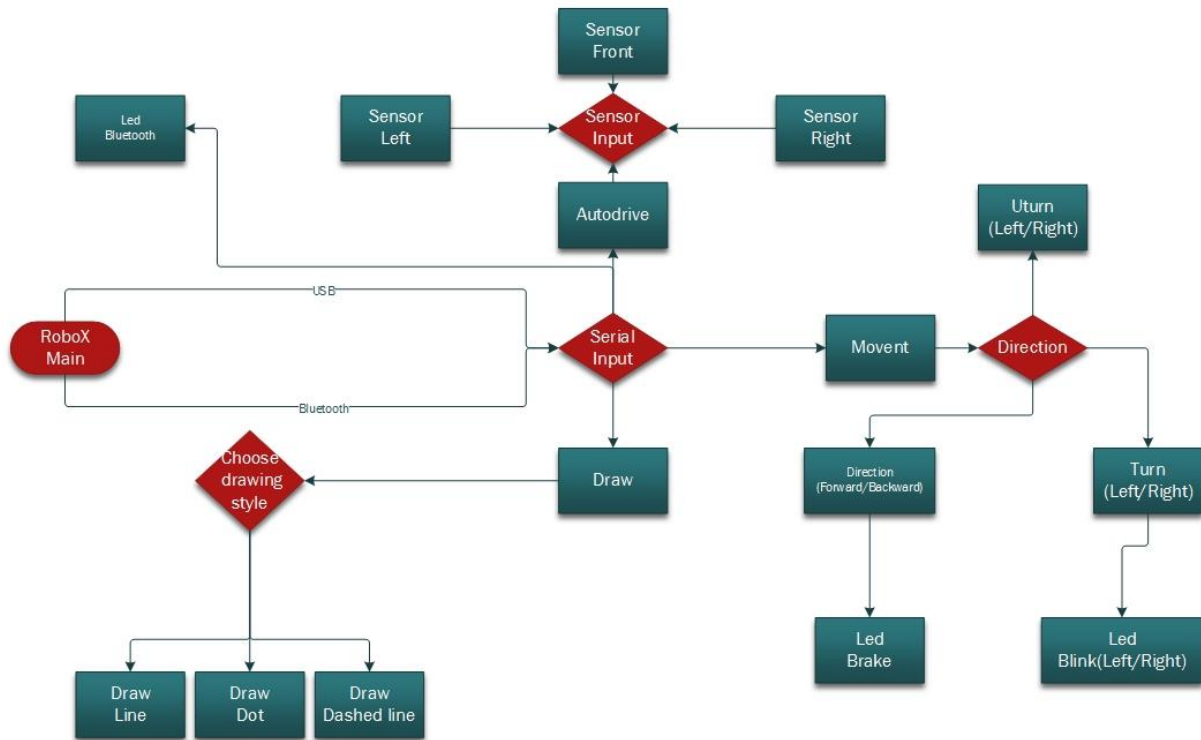
Zoals in de flowchart wordt weergegeven bestaat onze code uit 6 grote klassen (1 main + 5 libraries). Je heb de Main klasse van hieruit worden de commando's (afkomstig van de app) uitgelezen en worden de verschillende libraries aangeroepen.

De grote klassen die in libraries zitten zijn dus: Movement, Draw, Autodrive en Leds (Led Bluetooth).

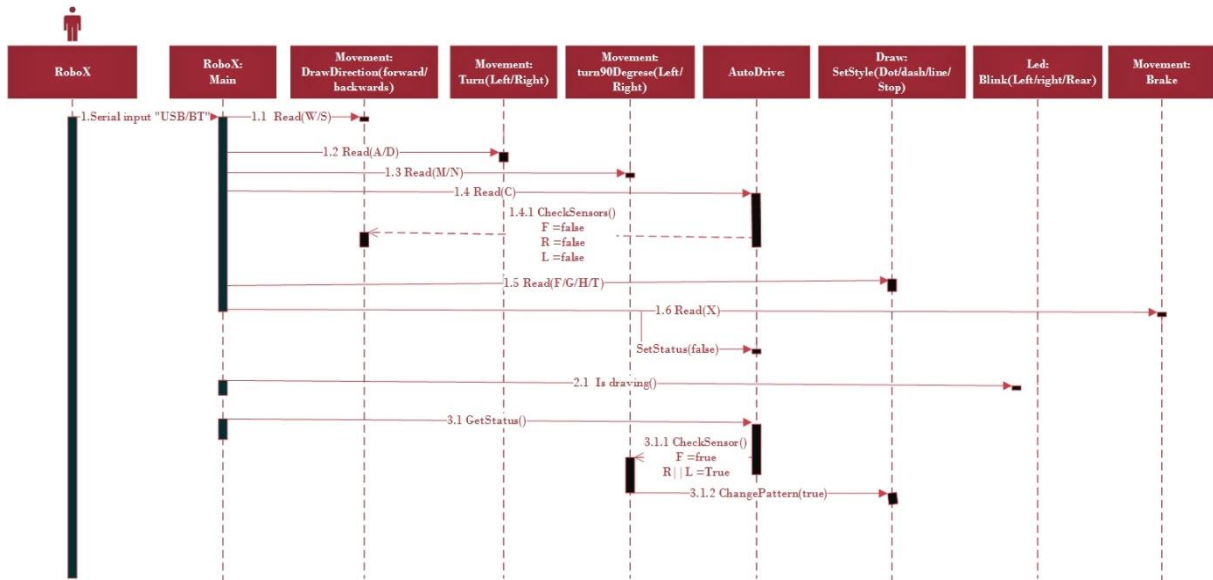
In de movement klasse zal afhankelijk van de commando's een bepaalde beweging worden uitgevoerd. Nadat deze beweging is uitgevoerd, zal de klassen verwijzen naar de Led klasse. Zo zullen bijvoorbeeld de rechter leds pinken als het autootje rechts aflat. Bij de autodrive zal ons autootje de sensor klassen aanroepen. Deze klasse zal de verschillende waarde van de sensoren meegeven aan de autodrive klasse. Dit is bijvoorbeeld: vooraan TRUE & rechts TRUE. Dit betekend dat onze auto zowel vooraan als recht een obstakel waarneemt. Hierdoor zal de auto links afdraaien. Hierbij gaat ons autootje de movement klassen weer aanroepen en zal hij de commando's meegeven. In plaats dat de user commando's meegeeft (door middel van de app), zal nu de auto dit zelf doen.

De draw klasse alleen worden aangeroepen als de user op de draw buttons drukt (zie afbeelding Android app) of wanneer ons autootje in autodrive modus staat. Hierbij zal het voertuig een bepaalt tekenpatroon aannemen.

Flowchart



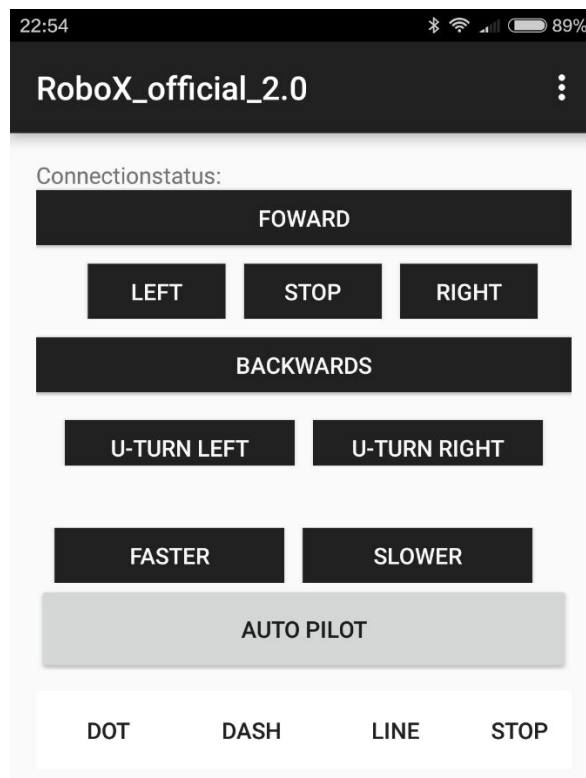
Sequentiedigram



5.2 Android app

Om het apparaat te besturen hebben we een android applicatie gemaakt. Hierbij krijgt de user een controle paneel, met verschillende knoppen op. De user kan dus het autootje volledig zelf besturen, op autopilot zetten en tekenen.

Met de knoppen 'DOT', 'DASH' en 'LINE' kan de user een mooi tekenpatroon kiezen.



6. Communicatie

6.1 Online

Vandaag de dag is het bijna onmogelijk om geen goede communicatiemiddelen te vinden. Dit gaat van gewone planning tot instructies geven en een online gesprek houden. We hebben de afgelopen maanden gebruik gemaakt van 3 communicatiemiddelen: Skype, Trello, Github/Smartgit.

6.1.1 Skype

Skype is een technologie dat we het meest gebruikt hebben, omdat dit een zeer eenvoudig en handige 'tool' is om direct een online conversatie te beginnen. Je kan zo allerlei zaken vragen en binnen de minuut kunnen je collega's antwoorden op je vraag.

De dag van vandaag wordt Skype zelfs door verschillende bedrijven gebruikt en terecht. Het is een zeker aan te raden tool voor directe online communicatie.

6.1.2 Trello

Trello werd vooral gebruikt om onze planning in orde te krijgen. Deze online service is zeer handig om op een agile wijze te kunnen werken aan een project.

Je kan je project zo onderverdelen in verschillende sprints en deze stap voor stap afwerken, aangezien elke sprint nog eens is onderverdeeld in stukken. Dit is het positieve aan Trello, je project is niet meer één enorme brok werk. Er komt structuur en overzicht in je werk en je kan taken op een zeer handige manier onderverdelen over je teamgenoten.

6.1.3 Github/Smartgit

Dit hebben we vooral gebruikt om code te delen, bewerken en verwijderen. Via Smartgit kan je op een zeer handige manier code van en op Github zetten, waarbij er automatisch code wordt samengevoegd (merging code) wordt toegepast.

6.2 Meetings

Uiteraard kan je geen project maken door af te spreken in het echt. We hebben de afgelopen maanden daarom veel samengekomen tijdens de lessen om aan ons robotje te sleutelen. Tijd is vaak het grootste gevaar in een eindproject zoals dit. Daarom hebben we ook enkele keren een ‘meeting’ gehouden bij iemand thuis.

De eerste keer zijn we bij Samuel samengekomen. De reden voor onze meeting was de fine-tuning van ons autootje en de voorbereiding voor ons mondelinge presentatie. Je kan zo’n zaken namelijk niet online doen, omdat er een duidelijke takenverdeling moet zijn.

De tweede meeting die we gehouden hebben was bij Gianni thuis. Hier hebben we gesleuteld aan de sensoren en het automatisch laten rijden van de robot.

De laatste meeting zal voor de eindpresentatie zijn. Hier moeten we zien dat we alles in orde hebben om zo de presentatie tot een goed einde te brengen.

7. Besluit

Het hele project roboX was een zeer leerzame en toffe ervaring. We hebben op een vrij leuke en volledige autonome robotje kunnen maken dat automatisch kon rijden en tekenen.

Dit project heeft ons vooral meer inzicht gegeven in het programmeren en het werken met libraries. Qua hardware was dit ook een zeer leerzaam project.

De presentatie voor de kunstacademie was ook een zeer plezierige en unieke ervaring die ons nog lang gaat bijblijven.

8. Bronvermelding

<http://www.pion.cz/en/article/electromagnetic-spectrum>

<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

<https://chioszrobots.com/2013/12/06/sn754410-h-bridge-motor-driver-stepper-motor-1a-compatible-l293d/>

<http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>

https://nl.wikipedia.org/wiki/Ultrasoon_sensor

http://education.rec.ri.cmu.edu/content/electronics/boe/ir_sensor/1.html