|  |  |
| --- | --- |
|  | **2015** |
|  | **STC01**  Alexander Ineke  Atze de Groot  Jorian Borst  Bart Camies  Joel Troost  https://cdn0.iconfinder.com/data/icons/science-10/450/robot-512.pngNathan Prins |

|  |
| --- |
| **[PlAn van aanpak]** |
| Plan van aanpak voor het project “Woordrobot” van de groep STC01 Windesheim 2015 |

1. **INHOUDSOPGAVE**
2. **INHOUDSOPGAVE**

**Inhoudsopgave…………………………………………………………………………………1**

**Plan van aanpak……………………………………………………………………………….2**

**Plan van aanpak: analyse………………………………………………………………….2**

**Globale tijdsplanning**

**Ontwerp**

**Benodigdheden**

**Analyse Algoritme(pseudo code + beschrijving)**

**Samenhang**

**Testplan**

**Gedetaileerde Tijdsplanning…………………………………………………………4**

**Evaluatie Ontwerp en realisatie**

**Reflectie(verbeterpunten+wat ging fout)**

**Uitleg bijlagen**

**HOUDSOP**

Eerst gaan we met het Team overleggen hoe we de robot precies willen maken. Hier moet uit komen wat er nodig is en wat we moeten doen de komen de weken. Dit doen we doormiddel van ervaringen van teamleden en voorbeelden van internet. Vervolgens maken we hier een ontwerp van en overleggen met de leraar of dat mag volgens de richtlijnen van de opdracht en of het goed is.

1. **PLAN VAN AANPAK**
2. **INHOUDSOPGAVE**

Hierbij is een Mindmap gemaakt en een ontwerp. Daarna kunnen we verder met het goedgekeurde ontwerp en maken we een (tijds)planning waarin alle taken besproken staan. Als laatste wordt de samenhang gemaakt (software+hardware allebei werkend) en het testplan uitgevoerd.

Voor de planning wordt Trello gebruikt om elke week opdrachten te verdelen over de groep en zo het werk delegeren.

**Analyse**

1. **INHOUDSOPGAVE**

Maken Ontwerp in beeld – Nathan Prins (week 4)

Uitleg: Een grafisch ontwerp maken van de robot zodat we een impressie krijgen van hoe het eruit moet zien en ook om er achter te komen of ons ontwerp goed is door het te laten zien aan de leraar.

Maken Ontwerp op papier – Atze de Groot (week 4)

Uitleg: Een ontwerp op papier zetten, hierdoor kunnen we weten wat we nodig hebben en hoe het in elkaar moet, verder kunnen we zo zien welke onderdelen we nodig hebben voorbepaalde functies en prioriteiten stellen in de opdracht.

Begin Testplan maken – Alexander Ineke (week 4)

Uitleg: Uiteindelijk moet er een Testplan komen aan de hand waarvan we kunnen nagaan of de schrijfrobot werkt naar behoren.

Globale Tijdsplanning – Joel (week 4)

Uitleg: Door aan iedere opdracht een aantal uren te geven kunnen we de opdrachten zo inplannen dat we optijd klaar zijn met de opdracht en alles ongehaast kan worden gemaakt, door het werk goed te verspreiden en op tijd te beginnen. Hierdoor krijgen we ook een beeld van hoeveel tijd we uiteindelijk kwijt zullen zijn.

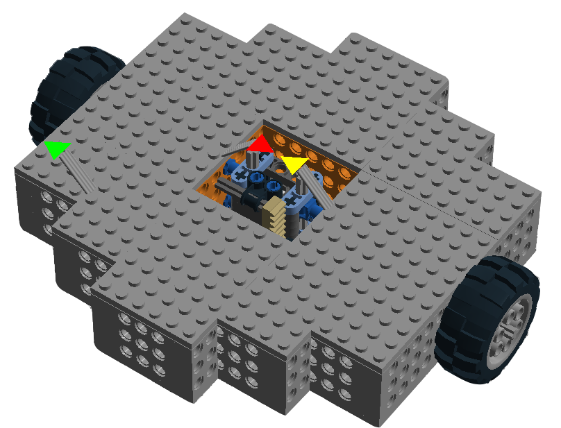
Documentatie – Bart Camies (week 4)

Uitleg: Bart is verantwoordelijk voor het samenvoegen van de documentatie tot 1 mooi bestand en hij moet ervoor zorgen dat dit ook opgestuurd wordt.

Plan van Aanpak – Jorian Borst (week 4)

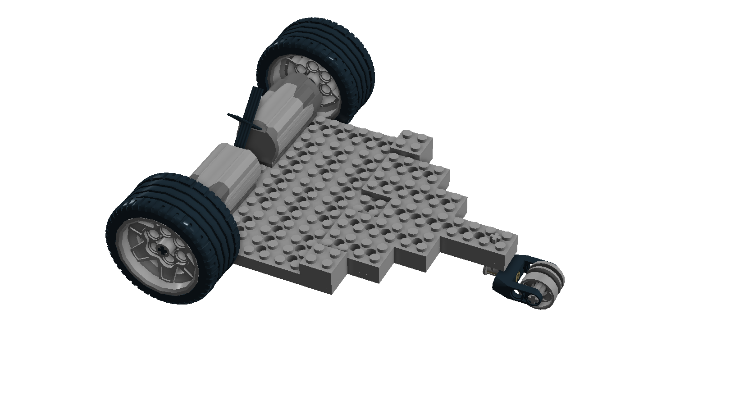
Uitleg: Door het plan van aanpak krijgt de groep een beeld van wat de bedoeling is van deze opdracht en wat de groepsleden moeten doen. Hierdoor weten ze altijd wat ze moeten doen en wat er gedaan hoort te zijn, Hiermee voorkom je verwarring.

Ontwerp

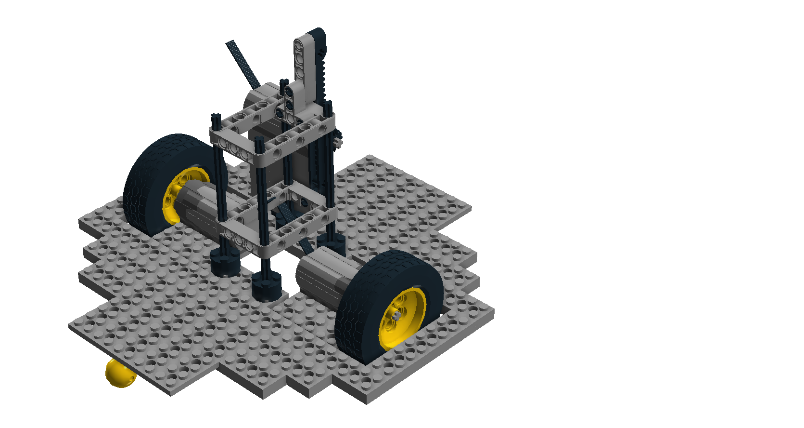


Uiteindelijke versie (3)

Inspiratie

We moesten voor de schrijfrobot opdracht natuurlijk een ontwerp maken van de body. We hebben in totaal 3 versies bedacht en zijn uiteindelijk op deze (zie afbeelding boven) uitgekomen.

Bij onze eerste versie hadden we een heel simpel ontwerp, een waar de wielen gevolgd werden door een zwenkwiel. De problemen die wij daarbij bedachten waren ten eerste dat het wiel onder het chassis vast kom komen zitten en daardoor helemaal niet meer kon roteren en ten tweede, dat er nu een staart achter de robot zwaaide die strakke rotaties moeilijk zou maken door de massa achter de robot. Versie 1.

Wij wilde in onze tweede versie alle problemen oplossen door het meer de vorm te geven van een Roomba. Hierdoor draaide alle massa om het middelpunt van de as van de wielen. Niet alleen maakt dat strakke bochten waar de pen op 1 punt blijft staan heel makkelijk, dit zorgt er ook voor dat hij heel goed in balans is en het dus niet nodig is om een zwenkwiel te gebruiken, maar de mogelijkheid open komt om twee glijdpunten te gebruiken.

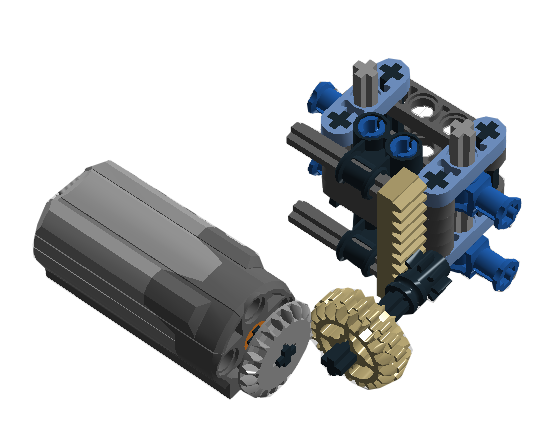
Als u de afbeelding rechts vergelijkt met de afbeelding van de Roomba ziet u grote overeenkomsten in de vorm, zowel de rondheid als de platheid.



Versie 2.

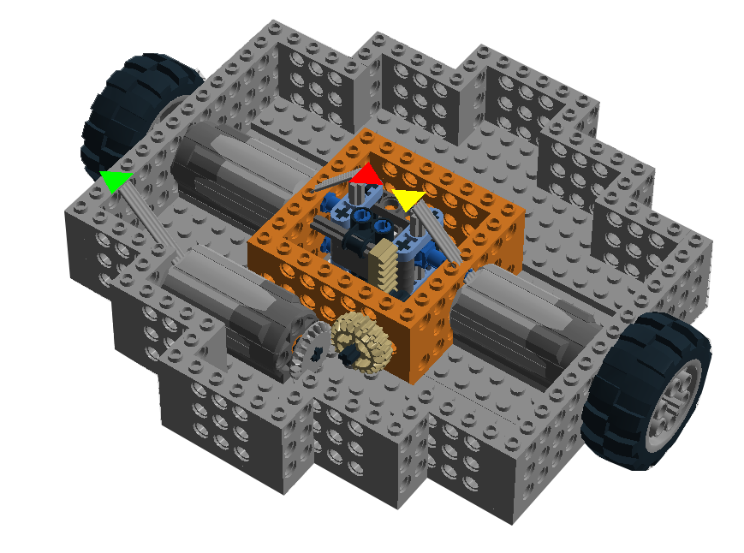
Roomba

De pen

Wij vonden het ook een leuke uitdaging om een mechanisme toe te voegen dat de pen omhoog en omlaag kon bewegen

Hier voor hebben wij het mechanisme hiernaast gebruikt.

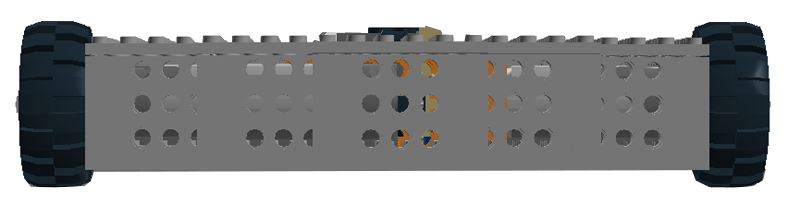
Dit mechanisme past perfect in een kleine ruimte en zorgt er voor dat we de pen net genoeg omhoog en omlaag kunnen bewegen om hem van het blad af te halen.



Links ziet u hoe het mechanisme in het chassis past.

Het ontwerp zou in theorie goed werken aangezien er weinig wrijving te voelen was en de motor zou meer dan genoeg power hebben om dit systeem te laten werken.

Een van de grootste voordelen van dit ontwerp is, is dat het zo plat is, zelfs met al deze functionaliteit pas het compleet binnen het chassis. (Zie afbeelding hieronder)



Jammer genoeg hadden wij in praktijk, door wat onvoorziene gebeurtenissen en te nauwe planning, niet de tijd om het helemaal uit te werken.

Voordelen:

* Hij is zeer snel en kan door het draaien om de as, zonder staart, heel rap keren. In gebouwd penmechanisme en licht.
* Hij is ook zeer klein en efficiënt met de ruimte die het heeft.

Nadelen :

Omdat lego door wrijving in elkaar blijft en wij een dunne bodem hebben met weinig lagen, is het huidige chassis zwak. Mocht dit zelfde chassis gemaakt worden met aluminium dan zou het waarschijnlijk nóg lichter en vele malen steviger zijn. Op dit moment is er nog weinig ruimte voor de elektronica, nu zou dit geen probleem zijn als we ons eigen printplaat konden ontwerpen, maar aangezien we dat niet kunnen moeten ze op de bovenkant gemonteerd worden.

Analyse Algoritme.

Allereerst zijn er een aantal eisen waar we met onze robot, dus ook met ons algoritme, ons aan moeten houden:

• De robot moet binnen 3 minuten het woord schrijven.

• Op het eerste 7-segmentsdisplay verschijnt een letter “L” (van letter) en op het tweede 7- segmentsdisplay verschijnt een nummer van de letter waarmee de robot bezig is. ( 1 voor eerste letter, 2 voor tweede letter enz. ). Dus achtereenvolgens L1, L2, L3, L4 enz. Dit hoeft alleen tijdens het 10 cm naar voren rijden vanaf de referentielijn! Tijdens de andere trajecten blijft het display gedoofd.

• De robot parkeert zichzelf in één van de vier hoeken wanneer het woord geschreven is.

De robot moet dus aan deze eisen voldoen, plus de robot moet een woord kunnen schrijven binnen het veld, dit gaan wij bereiken door een Arduino te programmeren; deze Arduino:

• Werkt op een afgeleide taal (de Arduino taal) van C, de reference te vinden op Arduino.cc.

Door niet werkende Sensors hebben we wel besloten de code ervoor te laten werken maar zonder de daadwerkelijke sensors te werken. Het algoritme doen we doormiddel van het uitelkaar trekken van de String die we invoeren d.m.v. char substr = content.charAt(i);

Voor elke letter hebben we dan een Actie die het Algoritme moet uitvoeren (zie Bijlage)

Deze Code werkt door het precies uit meten hoe een letter gereden moet worden. De Pseudo code hier naast laat zien hoe dat ongeveer gaat.

1. 90 graden naar links draaien

2. 10 cm naar voren rijden

3. 75 graden naar rechts draaien

4. pen neerzetten

5. +- 25 cm naar voren rijden

6. 30 graden naar rechts draaien

7. +-7 cm naar voren rijden

8. 110 graden naar rechts draaien

9. +- 5 cm naar voren rijden

10. +- 5cm naar achter rijden

11. 110 graden naar links draaien

12. +- 18 cm naar voren rijden

13. pen omhoog

14. 15 graden naar rechts draaien

15. 10 cm naar voren rijden

16. 90 graden naar links draaien

17. volgende letter

In een bijlage mapje staat elke letter uitgewerkt + code en berekeningen.

Het voordeel hiervan is dat je een letter makkelijk aanpakt en dat het relatief snel klaar is. Het werkt perfect als je altijd op dezelfde ondergrond werkt. Je kunt makkelijk kiezen hoe je de letters schrijft als er iets niet voldoet aan de regels.

Deze code zou perfect zijn als je met bijvoorbeeld sensors de spaken telt zodat je precies weet hoeveel elk wiel rijdt. Door tijdsgebrek is dit er niet van gekomen maar de letters zijn perfect gereden.

De Nadelen hiervan zijn:

* Erg afhankelijk van ondergrond.
* Niet zo precies als PWM

1. **GEDETAILLEERDE TIJDSPLANNING**
2. **INHOUDSOPGAVE**

|  |  |  |  |
| --- | --- | --- | --- |
| Week 1(week 4) | Week 2 (week 5) | Week 3 (week 6) | Week 4 (week 7) |
| In de eerste week is het eerste ontwerp gemaakt en een plan van aanpak  Het doel hiervan is dat we precies weten wat we moeten doen en wat we gaan maken. (eind week 1 Opdracht 1 en 2 af) | In week 2 moet het Algoritme beschreven worden en gemaakt zodat het theoritisch werkt. Verder denken we na denken hoe je het ontwerp en het algoritme goed samen laat werken. | In week 3 wordt de robot gemaakt en en testplan. Een gedetailleerde tijdsplanning.  Een herzien ontwerp wordt gemaakt om verbeteringen toe te passen | De laatste week moeten we realiseren testen en verbeteren.  Dit doen we door de Robot non stop worden te laten schrijven en het algoritme te verfijnen |

**4.Benodigdheden**

1. **INHOUDSOPGAVE**

Analyse van wat we nodig hebben voor de woordrobot:

- 2 wielen

- Kogelwiel

- Tandwielen

- Lego technic (stokjes vooral)

- 2 motoren

- Lego

- Pen

- 2 motorshields

Aan de achterkant doen we 2 wielen die worden aangestuurd door een motor en aan de voorkant doen we een kogelwiel. We gebruiken een kogelwiel aan de voorkant omdat de robot dan gemakkelijk kan sturen. Dit is ook zo bij de 2 achterwielen, want als we willen draaien dan laten we maar 1 wiel van de robot draaien. In het midden van de robot zit de pen en die is op een soort hefboom van tandwielen aangesloten. Deze zorgt ervoor dat de pen omhoog en omlaag kan wanneer dat nodig is. Op de twee wielen is een motorshield aangesloten en één op de pen. We hebben voor dit ontwerp gekozen omdat we maar 2 motorshields krijgen van school en dit leek ons de beste manier om de motorshields te benuttigen. Het midden stuk gaan we gewoon bouwen van lego en legotechnic. Verder sluiten we ook een lijnsensor aan die detecteerd waar de dikke en de dunne lijnen zitten van het speelveld. Dit is om te voorkomen dat de robot van het speelveld afrijd en om gelijk aan de dikke lijn te schrijven, anders zou de robot de letters schuin kunnen schrijven.