Er moet een verslag worden ingeleverd met een beschrijving van het project, de hardware en de software, welke problemen je hebt gehad, en hoe het uiteindelijke prototype afwijkt van de oorspronkelijke beschrijving.

Topverslag

# Beschrijving

In dit project is met behulp van een Arduino Uno een voertuig gemaakt waarmee een omgeving wordt verkend. De omgeving wordt in kaart gebracht door met dit voertuig naar de positie te gaan die het meest dichtbij is. Rijdt het voertuig tegen een obstakel aan, dan wordt dit waargenomen en opgeslagen in de interne kaart. Als de robot geen pad meer heeft om te verkennen dan zal hij stoppen met verkennen. De huidige kaart wordt dan naar het eeprom van de arduino geschreven. Deze kaart kan daarna uitgelezen worden op de laptop.

## Hardware

Het voertuig bestaat uit een Arduino Uno in combinatie met een motorshield (DFRduino Motor Drive). Deze zijn gemonteerd op een constructie waarop 2 motoren en een zwenkwieltje zitten. Op de voorkant van het voertuig zit een bumper die gekoppeld is aan druksensoren. Rijdt het voertuig ergens tegen aan, dan worden deze sensoren ingedrukt en krijgt de Arduino een signaal.

## Software

De software is verdeeld in verschillende klassen, elk met een eigen functie. Hieronder staat een overzicht van de gebruikte klassen.

### Map

Deze klasse is verantwoordelijk voor het bijhouden van de kaart. Een kaart bestaat uit een tweedimensionale array van chunks. Elke chunk heeft twee waarden: de eerste geeft aan of de chunk al behandeld is, de tweede geeft aan of deze chunk een obstakel bevat.

Naast methoden om deze waarden in de kaart te veranderen, bevat deze klasse ook twee methoden om de kaart te schrijven naar en te lezen van de eeprom. Op deze manier kan het voertuig zijn werk zonder via een serieële kabel vebonden te zijn, terwijl na het verkennen de kaart uitgelezen kan worden

### Motor

Deze klasse zorgt voor het aansturen van de motor.

### Pathfinder

Met behulp van de map zoekt de Pathfinder-klasse vakjes op die nog verkend moeten worden. Door middel van deze klasse weet de robot welke kant hij op moet rijden om naar het volgende vakje toe te verplaatsen.

Het pad word bepaald door elke vakje te bepalen wat de afstand is tot dit vakje. De afstand is uitgedrukt in het aantal verplaatste vakjes plus het aantal draaiingen van 90 graden. Dit wordt bepaald door voor elke vakje te kijken wat de kosten zijn voor de buren en hier 1 en de kosten voor een eventuele draaiing bij op te tellen. Als het minimum hiervan minder is dan de oude waarde, worden de huidige kosten en richting na aankomst veranderd. Dit gaat door tot er niets meer veranderd is. Op dit moment zijn voor elk vakje de kosten berekend. Het vakje met de minimale kosten waarbij dit vakje nog niet is verkend, wordt gekozen als doel.

Aan de hand van de onthouden richtingen kan de route zelf worden bepaald. Als het voertuig na aankomst op een vakje naar het noorden wijst, weet je dat hij hier is gekomen vanuit het zuiden. De stap om hier te komen is dus om beweging naar het noorden te doen.

### Test

Bij een goede applicatie horen testen. De bedoeling van deze klasse is dan ook om te testen of verscheidene methoden in verschillende klassen werken naar verwachting. Met name voor het pathfinding-algoritme is dit van belang, aangezien dit redelijk foutgevoelig is. Daarnaast worden de functies van de Map klasse en het lezen en schrijven van de kaart naar het EEPROM getest.

### Bumper

Deze klasse controleert of er een botsing is met een obstakel.

# Problemen

### Rotatie

De motor is gekalibreerd om een rotatie van om en nabij 90° te kunnen maken. Helaas is dit niet een perfecte hoek van 90 graden. Beide motoren draaien ook even snel en de wieltjes zijn niet perfect rond wat er voor zorgt dat er een kleine afwijking zit in de besturing. Als het programma te lang draait dan zal de afwijking te groot worden waardoor de robot niet meer precies weet waar hij is op het moment.

Er zijn componenten die dit probleem gedeeltelijk kunnen aanpassen. Zo zijn er

Pathfinding

Voeding (koeken/tosties  enzo)

# Resultaat