MJTL: PROJECT TURTLE

Door: Leo Jenneskens, Tim IJntema, Monisha Wielkens en Jip Galema.

De opdracht

Voor dit project was het de bedoeling om een zelfrijdende auto te maken. Dit zou gebeuren in de vorm van een lego mindstorm robot. Deze robot 'moest' een aantal dingen kunnen. Daarnaast was er een lijstje dingen die eigenlijk ook gedaan moesten worden. Als laatste was er een lijstje punten die je kon doen als er tijd over was. Onze robot (zie afbeelding hieronder) kan alles wat we moesten hebben, dit hield in dat het voertuig: Een lijn kan volgen, stopt als er een obstakel aanwezig is, tijdens het rijden geluid maakt, stopt op commando van de mobiele telefoon en weer rijdt op commando. Daarnaast hebben we ook een groot deel van de andere punten voor de casus voltooid. Zo kan onze robot bijvoorbeeld kruispunten herkennen en links, rechts of rechtdoor gaan op het kruispunt.

Voordelen en nadelen van ons ontwerp

De voordelen van ons ontwerp hebben vooral te maken met de lichtsensoren. We hebben de twee sensoren naast elkaar gezet. Hierdoor kunnen we heel goed kruispunten herkennen (als immers de twee sensoren allebei zwart geven dan staat de robot op een kruispunt). Daarnaast hebben we ook een kapje om de sensoren heen gebouwd, hierdoor komt er minder licht van buitenaf binnen. Een nadeel van ons ontwerp is het achterste draaiwieltje, deze wordt namelijk best onder druk gezet door het gewicht van de robot, het sturen lukt goed maar dit kan over tijd misschien voor problemen zorgen

Problemen waar we tegen aan liepen

Er waren bij dit project een aantal problemen waar we tegenaan liepen. Zo kwamen we erachter dat het lastig is om t-splitsingen te herkennen met de robot, gelukkig zitten er in het parkour geen t-splitsingen. Daarnaast was er een probleem met de licenties van robotC, hierdoor konden we een hele tijd in de derde week nier werken. Ook werkt de bluetooth app vrij slecht, hij verbindt maar heel kort en hij blijft de hele tijd dingen sturen. Daarnaast was het lastig om robot code te testen, omdat we allemaal robot code wouden testen.

Hoe werkt het?

Het geluid

We hebben een lijstje gevonden met verschillende muziek functies gevonden en konden met playSoundFile *();* de gewenste muziek afspelen. Het rso-bestand (muziek bestand omgezet uit een WAV-bestand) was alleen niet boven de motoren te horen. Daarom zijn we overgestapt op de functie playTone(); maar dat resulteerde in een hele lange lijst omdat elke toon individueel met eigen wait time aangegeven moet worden. Uiteindelijk zijn we op het rmd bestand (omgezet midi-bestand) gestuit, die muziek als een lijst playTone functies afspeelt. De muziek werkt nu volgens onderstaande codes.

```
task music(){
        playSoundFile("supermario.rmd");
        wait1Msec(10000);
}
task main(){
        startTask(music);
        while(1){
                //motorcodes
        }
}
```

Lijn volgen

Eén van de hoofd opdrachten het zorgen dat de robot een lijn volgt. De robot moet over een lijn rijden en hierbij op de lijn blijven en niet van deze lijn afgaan. Hoe wordt op de lijn gebleven? Door aan elke kant van de lijn een sensor te hebben die vele malen per seconde de licht waarde uit meet. De waardes die de sensoren meten geven aan hoe groot deel van het onderliggende vlak zwart is en dus hoever de robot over de lijn zit. Door de gemeten waardes als een variabele in een wiskundige formule te gooien wordt berekent hoe de motoren moeten worden aangestuurd. Het gebruik van een formule is noodzakelijk voor het goed verlopen van de bochten in de lijn om te zorgen dat de robot niet te veel en niet te weinig stuurt. Echter in verschillende omgevingen kunnen de licht waardes variëren, hier is aan gedacht door het gebruik van een globale variabelen voor de maximale en de minimale Lichtwaardes zodat de robot kan berekenen wat de beste waarde is voor het volgen van de lijn.

Bluetooth

De onderdelen om bluetooth werkende te krijgen waren door de docenten geleverd. Dit hebben wij gekopieerd en gewijzigd zodat het zou werken in onze code. We hebben nu 2 bluetooth functies in een eigen header file De eerste bluetooth functie is een hele simpele functie die bij aanroep de bluetooth input uitleest. Wanneer deze is gegeven stopt hij de input in een string die , door de pointer verwijzing die meegegeven is bij de aanroep, ook buiten de bluetooth functie beschikbaar is. Deze functie heet check_bluetooth(). De tweede is wat uitgebreider. Hij word aangeroepen vanuit tak main() als de input van de bluetooth check die daar word gedaan gelijk is aan "B". Dit is de knop voor een noodstop aangezien de motoren in een keer op 0 worden gezet. In de functie zelf, genaamd bluetooth_control(), Word er opnieuw gewacht op een bluetooth input. Aan de hand van deze bluetooth input word er een bepaalde handeling uitgevoerd. Behalve rechts, links, enz., is er ook nog de optie om de code af te sluiten of verder te gaan met de normale code (de knoppen "C" en "A").

Object detectie

Object detectie is een onderdeel dat in de follow line functie verwerkt zit. Een object dat minder dan 30 cm voor de robot staat, is een object waar de robot voor stopt. Dit doet hij langzaam afremmend om te zorgen dat inzittenden niet door elkaar worden geschud. Vervolgens draait de robot rustig om met een versnelling.
Wanneer de robot de lijn weer ziet vertraagt hij weer met draaien en hierna rijdt hij verder. De andere kant op. De sonar op de voorkant van de robot detecteert het object. Dit doet hij continu vanwege de oneindige while loop in de task main functie. Hierdoor weet de robot altijd wanneer hij moet stoppen.