



Overdrachtsdocument DECO

18-06-2021

Maaïke Hovenkamp

Personal innovation 2021

Inleiding

Welkom bij de inleiding voor het overdrachtsdocument voor de sociale desktop companion DECO. DECO is een makkelijk programmeerbare robot voor op je bureau om je gezelschap te houden door de dag heen. DECO kan gebruikt worden om je te herinneren voor jezelf te zorgen door de dag heen en je te vermaken met een klein spelletje. Om ervoor te zorgen dat de robot makkelijk toegankelijk is, is er gekozen voor goedkopere onderdelen. Met amazon komt het totaal uit op +- 64 dollar maar lokale bronnen hebben vaak goedkopere onderdelen.

DECO is volledig ontworpen en geproduceerd door Maaïke Hovenkamp.

Inleiding	1
Artefacten	1
STL files voor DECO	1
Software	2
Software & hardware modellen	2
Documentatie	2
Bugs en beperkingen	3
Bugs	3
Beperkingen	3
Product backlog	3

Artefacten

Alle documentatie is te vinden op de Github pagina:

<https://github.com/MHovenkamp/DECO>

STL files voor DECO

Om de robot te printen zijn zes STL files geproduceerd:

1. Neck horizontal part, Deel 1 van neck gedeelte.
2. Neck vertical part, Deel 2 van neck gedeelte.
3. Lower body, voet van de robot met gat voor voeding en PIR-sensor
4. Upper body, Verlengstuk voor de voet met kom voor hoofd.
5. Lower head, bodem van hoofd passend in de kom van het lichaam.

6. upper head, Hoofd met gat voor OLED-screen en LIDAR module.

Er is een extra set aan STL file voor de upper body waarin dit onderdeel in 2 stukken is gesneden.

Software

Om de robot te besturen is er code geschreven in C++ op het framework van PlatformIO. PlatformIO is een cross-platform embedded development platform integreerbaar in visual studio code. De robot maakt gebruik van een paar libraries:

1. Adafruit BusIO, Oled screen
2. Adafruit GFX Library, Oled screen
3. Adafruit SSD1306, Oled screen
4. Adafruit_VL53L0X, LIDAR
5. Arduino_HTS221, interne luchtvochtigheid en temperatuur
6. Arduino_LPS22HB, interne BARO sensor

Om met de software te werken moet PlatformIO geïnstalleerd worden en de github pagina als project ingeladen worden. Alle libraries en build platformen worden automatisch geïnstalleerd bij de eerste build.

Software & hardware modellen

Voor de software zijn modellen gemaakt volgens de UML methode:

1. Object model
2. Task scheduling
3. Class diagram
4. State transition diagrams.

Er is een fritzing voor de kabel layout.

Deze bestanden zijn te vinden op de Github pagina.

Documentatie

Voor de DECO robot zijn verschillende verslagen geschreven.

1. Design document, Een overzicht van alle design keuzes en de daaruit volgende Use cases. In dit verslag zijn de keuzes voor hardware en software niet vastgelegd.
2. Technische verslag, In dit verslag wordt er een onderbouwde keuze gemaakt voor:
 - a. Code taal
 - b. Ontwikkelomgeving
 - c. Hardware

Bugs en beperkingen

Na vier maanden werken aan de DECO robot zijn de volgende bugs en beperkingen aanwezig:

Bugs

- De robot heeft momenteel een probleem met een tikkend geluid over de I2C bus. Een tik is hoorbaar.
- De robot verliest af en toe de functionaliteit over de servo motoren. Na de een keer de kabels te hebben bewogen werkt de robot weer. De kans dat het een slechte kabel is bestaat maar het kan ook een probleem zijn met de stroomlevering. Het probleem wordt significant minder als de servomotoren geen ground meer delen met het Oled scherm.
- De serial input van Arduino is niet extreem betrouwbaar waardoor file input vanaf de terminal na een bepaalde grootte niet meer succesvol werkt.

Beperkingen

- De robot heeft momenteel 6 threads lopen waardoor een getimedede task later kan zijn. Een interpreter commando kan een seconde achterlopen. Alle sensoren worden wel correct getimed.
- De codetaal is momenteel erg beperkt. Momenteel worden een aantal robot commandos ondersteunt en een hele minimale interpreter met integer support, if nodes en while nodes. Alles wordt geparsed en gelexed maar er kunnen meer specifieke errors worden toegevoegd en betrouwbaarheid kan hoger. Dit kan gedaan worden door bijvoorbeeld elk commando met een ; af te sluiten en hier op te parsen.

Product backlog

Hierbij de originele use cases, een volledig overzicht is te vinden in het design document.

1. Als gebruiker wil ik dat de robot mensen herinnert om elk half uur minimaal vijf minuten te bewegen.
2. Als gebruiker wil ik dat de robot mensen herinnert om elke twee uur een pauze te nemen.
3. Als gebruiker wil ik dat de robot mensen herinnert om elk uur een glas water te drinken.
4. Als gebruiker wil ik dat de robot een uitknop heeft waar de robot mee uitgeschakeld kan worden.

- 
5. Als gebruiker wil ik dat de robot gebruik maakt van piepjes, beweging en afbeeldingen om te communiceren.
 6. Als gebruiker wil ik dat de robot heeft een scherm waarop instructies kunnen worden getoond.
 7. Als gebruiker wil ik dat alle documentatie rond het gebruik van de robot in engels wordt geschreven.
 8. Als gebruiker wil ik dat de robot op een standaard USB-poort kan worden aangesloten voor stroom.
 9. Als gebruiker wil ik dat de robot geen gegevens deelt met derde partijen.
 10. Als gebruiker wil ik dat de robot geen gegevens over de gebruiker verzamelt afgezien van wat de gebruiker de robot geeft.
 11. Als gebruiker wil ik dat de robot geen scherpe randen of puntige objecten heeft.
 12. Als gebruiker wil ik dat de robot stationair is en niet zonder de hulp kan verplaatsen.
 13. Als gebruiker wil ik dat de robot de vooraf geplande acties uitvoert en hier niet van afwijkt.

De enige use case die niet uitgewerkt is een mogelijke uitknop voor de robot.

De originele backlog voor dit ontwerp is grotendeels afgerond maar ik wil in dit kopje wat verdere inspiratie geven voor mogelijke toekomstplannen voor DECO.

Verdere interpreter mogelijkheden:

1. Toevoegen van een printfunctie voor debuggen. Outputten van integer waarden en string variabelen.
2. Output van functies opvangen in integer nodes en hiermee kunnen rekenen.
3. Implementeren van boolean waardes, momenteel kunnen integer nodes hiervoor gebruikt worden maar een individuele variabele soort kan gewaardeerd worden.

Robot mogelijkheden

1. Meer vormen van interactie met de gebruiker, denk aan meer vormen van geluid.
2. De microcontroller van de robot kan een klein tensorflow model ondersteunen, De robot kan meer sociaal worden als hij zelf meer keuzes kan maken. Met een klein neurale netwerk zouden meer natuurlijke keuzes kunnen worden gemaakt.