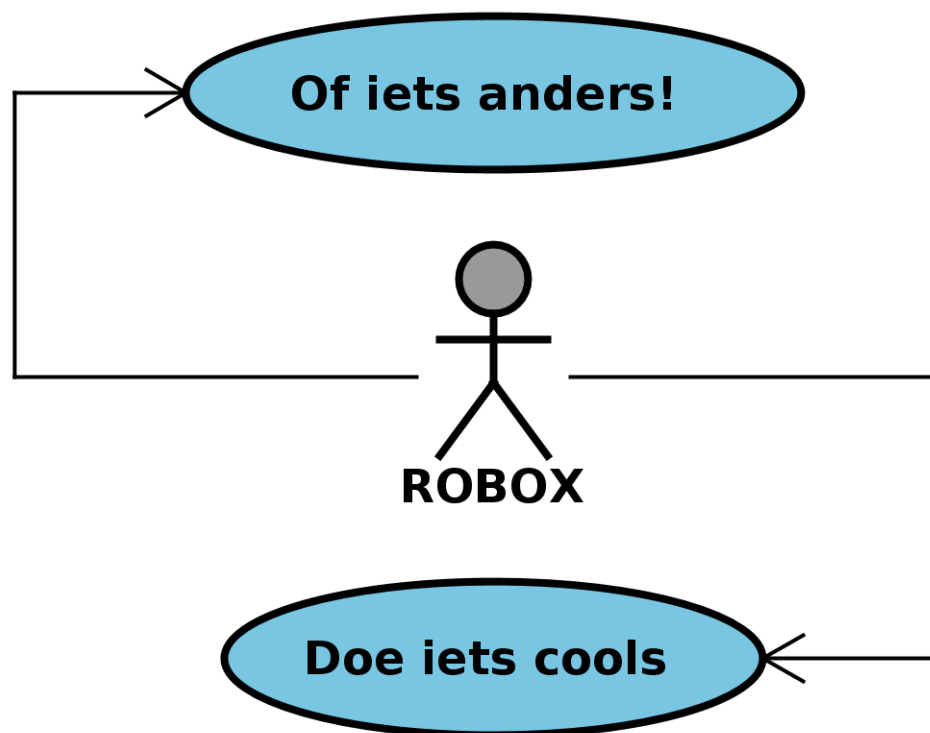


ROBOX - Firmware

Software requirements specification



Auteur	Luke van Luijn	Minor	Digital Media Productions (DMP)
Student nummer	587478	Docentbegeleider	Mario de Vries
Opleiding	HBO-ICT	Plaats	Nijmegen
Profiel	Embedded Software Development (ESD)	Datum	25-05-2022
Studiejaar	Jaar 3	Versie	1.0

Inhoudsopgaven

- 1 [Termen](#)
- 2 [Introductie](#)
 - 2.1 [Gebruikers, klassen en karakteristieken](#)
 - 2.2 [Ontwerp- en implementatie beperkingen](#)
 - 2.3 [Ontwikkelomgeving](#)
 - 2.4 [Product functionaliteiten](#)
- 3 [Usecase beschrijvingen](#)
 - 3.1 [UC0 - Aansturen van de robot](#)
 - 3.1.1 [UC0_0 - Aansturen door middel van hoeken](#)
 - 3.1.2 [UC0_1 - Aansturen door middel van coördinaten](#)
 - 3.1.3 [UC0_2 - Robot starten](#)
 - 3.1.4 [UC0_3 - Robot stoppen](#)
 - 3.1.5 [UC0_4 - Aansturen van de gripper](#)
 - 3.1.6 [UC0_5 - Robot pauzeren](#)
 - 3.2 [UC1 - Instellen van de configuratie](#)
 - 3.3 [UC2 - Reageren op requests](#)
 - 3.4 [UC3 - Uitvoeren Noodstop](#)
- 4 [Requirements](#)
 - 4.1 [Functionele requirements](#)
 - 4.2 [Niet functionele requirements](#)
 - 4.2.1 [QoS - Usability](#)
 - 4.2.2 [QoS - Reliability](#)
 - 4.2.3 [QoS - Performance](#)
 - 4.2.4 [QoS - Supportability](#)
- 5 [Literatuurlijst](#)

1. Termen

Index	Term	Beschrijving
00	ROBOX/Robot	Met deze term wordt het fysieke apparaat bedoelt.
01	Microcontroller	Een microcontroller (Wikipedia-bijdragers, 2022a) is een kleine computer die het mogelijk maakt om direct met de input en output pinnen te communiceren.
02	Clock/klok snelheid	Hoe snel een processor (van bijvoorbeeld een microcontroller) een commando kan uit voeren per seconde. Een clocksnelheid van 16MHz staat dus voor 16.000.000 commando's per seconde (Wikipedia-bijdragers, 2020).
03	Computer	Hiermee wordt een 'normale' computer bedoelt, bijvoorbeeld een laptop of desktop.
04	Software/overkoepelende software	Hiermee wordt de bijhorende applicatie bedoelt.
05	Firmware	De software die op een apparaat, bijvoorbeeld een microcontroller, draaid.
06	Actor	De gebruiker van een systeem.
07	32 bits	Deze term geeft aan hoeveel geheugen een processor tegelijkertijd kan aanspreken. 32 bits kan dus 2^{32} waardes opslaan, een 8 bits architectuur kan maar 2^8 aantal waardes opslaan (Wikipedia-bijdragers, 2022a)
08	IC	'integrated circuit' een chip die meestal op een printplaat verwerkt is.
09	Run time data	De data velden die tijdens het afspelen van de applicatie veranderen.
10	Compile time data	De data velden die op het moment van compileren bepaald worden en verder niet veranderen.
11	Serieele bus	Een veegebruikt, maar ouderwets, protocol waarmee verschillende apparaten, bijvoorbeeld via USB, data kunnen uitwisselen (Wikipedia-bijdragers, 2022).
12	PWM	'Pulse width modulation' Een blokgolf signaal die een bepaalde waarde kan weergeven, in dit geval een 8 bits waarde (0/255).

2. Introductie

Dit is het software requirements specification document van het firmware aspect van het ROBOX project. De firmware van dit project zal draaien op een microcontroller in de robot zelf. Deze firmware is verantwoordelijk voor het correct interpreteren van de verschillende commandos afkomstig van de overkoepelende software draaiend op een verbonden computer. Verder is de firmware verantwoordelijk voor het waarborgen van de verschillende veiligheids eisen, eisen zoals tijdig stoppen, niet voorbij mini- en maximale waardes gaan en soortgelijke situaties.

2.1. Gebruikers, klassen en karakteristieken

Voor de firmware is er een enkele actor; Het overkoepelende systeem draaiend op een aangrenzende computer. De gebruiker van ROBOX zal geen directe interactie met de firmware hebben.

2.2. Ontwerp- en implementatie beperkingen

Gezien microcontrollers een beperkte hoeveelheid RAM en Geheugen hebben zal er tijdens de ontwikkeling rekening gehouden worden met deze beperkingen. De firmware zal dus waar mogelijk geen gebruik maken van dynamisch geheugen en zo veel mogelijk aspecten beperken op gebied van 'run time data' en de data maximaliseren op gebied van 'compile time data'.

2.3. Ontwikkelomgeving

De firmware zal ontwikkeld worden voor een moderne microcontroller met een degelijke clock snelheid. Een voorbeeld van een dergelijke microcontroller is bijvoorbeeld de [Teensy 4.0](#) (PJRC, n.d.), deze microcontroller beschikt over een 32 bits ARM IC met een clock snelheid van 600 MHz. Een voorbeeld van een niet geschikte microcontroller is bijvoorbeeld de [Arduino Uno](#) (Arduino, n.d.), deze microcontroller beschikt over een 8 bits ATmega IC met een clock snelheid van 16MHz.

De gebruikte microcontroller zal verder moeten beschikken over een seriële bus die aangespreekbaar is over USB. De communicatie met de overkoepelende software zal hier namelijk gebruik van maken.

2.4. Product functionaliteiten

De firmware zal verantwoordelijk zijn voor het aansturen en uitlezen van de verschillende hardware componenten. Hierbij is het belangrijk dat de componenten op correcte wijzen aangestuurd worden en mini- en maximalen waardes niet overschreden worden. Dit is belangrijk omdat er gewerkt zal worden met een fysiek apparaat dat zal beschikken over motors die krachtig genoeg zijn om het apparaat te vernielen.

3. Usecase beschrijvingen

In het onderstaande diagram zijn de verschillende use cases te vinden omtrent de firmware. In dit diagram is (schematische) samenhang weergegeven van de verschillende onderdelen van het systeem en hoe de gebruiker (de driver) gebruik kan maken van het systeem.

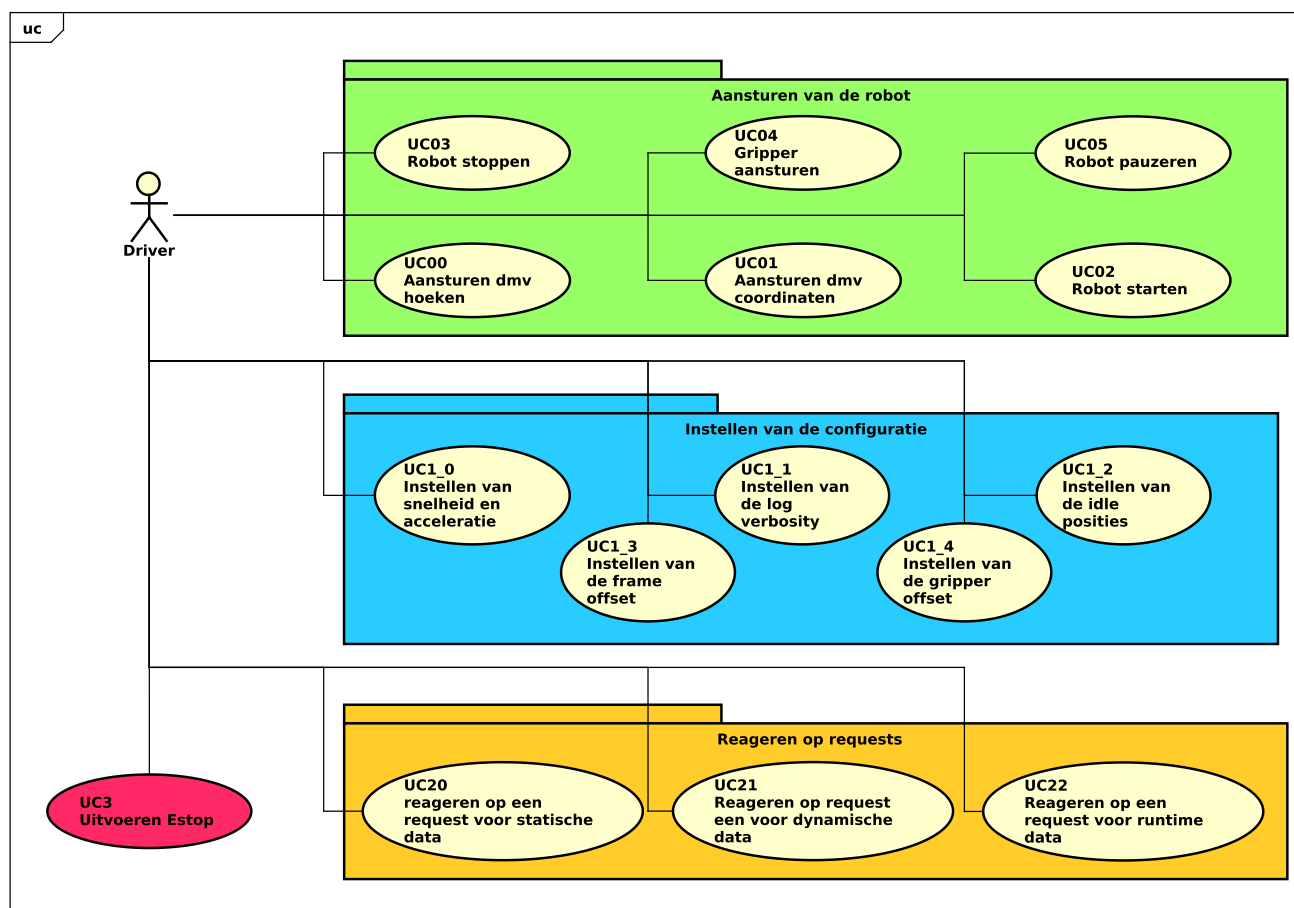


Diagram 1 - Use case diagram Firmware

In het diagram is te zien dat de verschillende use cases opgedeeld zijn in vier verschillende segmenten.

Het eerste segment 'Aansturen van de robot' is verantwoordelijk voor alle use cases die betrekking hebben tot het bewegen van de robot. Denk hierbij aan het bewegen naar een bepaalde positie of het pauzeren van de robot.

Het tweede segment, 'Instellen van de configuratie', regelt het instellen van de robot. Bijvoorbeeld het instellen van de snelheid en acceleratie.

Het derde segment 'Reageren op requests' is verantwoordelijk voor het opsturen van de data naar de driver. Het opsturen van de huidige positie is een voorbeeld hiervan.

Tot slot het vierde onderdeel; 'Uitvoeren Estop'. Het uitvoeren van een noodstop zal een andere uitvoering krijgen dan de onderdelen vewerkt in 'Aansturen van de robot', en heeft hierdoor ook een apart segment.

3.1. UC0 - Aansturen van de robot

In dit onderdeel zullen de verschillende use cases uit het onderdeel 'Aansturen van de robot' dieper toegelicht worden.

3.1.1. UC0_0 - Aansturen door middel van hoeken

Primary actor:	Driver
Brief description:	Het systeem reageert op een inkomend bericht omtrent het aansturen door middel van hoeken en stuurt de robot vervolgens adequaat aan.
Preconditions:	De robot heeft een 'start'-bericht ontvangen en verwerkt (bevind zich in een bekende positie).
Postconditions:	De robot zal zich bevinden in de ontvangen positie.
Actor action	System responsibility
1. Geeft een nieuwe positie op.	2. Verwerkt het ontvangen bericht.
	3. Kijkt of de posities binnen de uiterste waardes vallen.
	4. Kijkt of de robot klaar is om een beweging te maken.
	5. Beweegt de verschillende assen naar de gewenste hoeken.
Alternative flow 01:	
	<i>De waardes vallen niet binnen de uiterste waardes.</i>
	4a. Het systeem zal een log bericht versturen waarin vermeld wordt dat de positie niet uitgevoerd kan worden.
Alternative flow 02:	
	<i>De robot is niet klaar om een nieuwe beweging uit te voeren.</i>
	5a. Het bericht wordt toegevoegd aan een queue en op een later moment uitgevoerd worden.

Usecase 1 - Fully dressed usecase description - UC00

3.1.2. UC0_1 - Aansturen door middel van coördinaten

Primary actor:	Driver
Brief description:	Het systeem reageert op een inkomend bericht omtrent het aansturen door middel van hoeken en stuurt de robot vervolgens adequaat aan.
Preconditions:	De robot heeft een 'start'-bericht ontvangen en verwerkt (bevindt zich in een bekende positie).
Postconditions:	De robot zal zich bevinden in de ontvangen positie.
Actor action	System responsibility
1. Geeft een nieuwe positie op.	2. Verwerkt het ontvangen bericht.
	3. Berekent de hoeken op basis van de ontvangen coördinaten.
	4. Kijkt of de posities binnen de uiterste waarden vallen.
	5. Kijkt of de robot klaar is om een beweging te maken.
	6. Beweegt de verschillende assen naar de berekende hoeken.
Alternative flow 01:	
<i>De waarden vallen niet binnen de uiterste waarden.</i>	
	5a. Het systeem zal een log bericht versturen waarin vermeld wordt dat de positie niet uitgevoerd kan worden.
Alternative flow 02:	
<i>De robot is niet klaar om een nieuwe beweging uit te voeren.</i>	
	6a. Het bericht wordt toegevoegd aan een queue en op een later moment uitgevoerd worden.

Usecase 2 - Fully dressed usecase description - UC01

3.1.3. UC0_2 - Robot starten

Primary actor:	Driver
Brief description:	Het systeem reageert op een inkomend bericht omtrent het starten van de robot, vervolgens zal het systeem de stappen ondernemen om de robot te starten.
Preconditions:	Er is een seriele verbinding met de driver en de robot bevindt zich in een onbekende positie, of net opgestart, of gestopt.
Postconditions:	De robot bevindt zich in een bekende positie.
Actor action	System responsibility
1. Stuur een start commando.	2. Verwerkt het ontvangen bericht.
	3. Voert de homing sequence uit voor de verschillende actuatoren in de robot.
	4. Update alle data, omtrent positionering van de robot, naar de nieuwe waardes.
	5. Stuur een bericht naar de driver omtrent het voltooien van de homing sequence.

Usecase 3 - Fully dressed usecase description - UC02

3.1.4. UC0_3 - Robot stoppen

Primary actor:	Driver
Brief description:	Het systeem reageert op een inkomend bericht omtrent het stoppen van de robot, vervolgens zal het systeem de stappen ondernemen om de robot te stoppen.
Preconditions:	De robot heeft een 'start'-bericht ontvangen en verwerkt (bevind zich in een bekende positie).
Postconditions:	De robot bevind zich in de 'idle'-positie en de motoren zijn uitgeschakeld.
Actor action	System responsibility
1. Stuur een stop commando.	2. Verwerkt het ontvangen bericht.
	3. Kijkt of de robot klaar is om een beweging te maken.
	4. Beweegt de assen naar de verschillende 'idle'-posities.
	5. Deactiveert de verschillende motoren in de robot.
	6. Verwijdert alle ge-queuede bewegings commando's uit het systeem.
Alternative flow 01:	
	<i>De robot is niet klaar om een nieuwe beweging uit te voeren.</i>
	4a. Het bericht wordt toegevoegd aan een queue en op een later moment uitgevoerd worden.
	Zodra het bericht uitgevoerd wordt zal er gestart worden bij stap 4.

Usecase 4 - Fully dressed usecase description - UC03

3.1.5. UC0_4 - Aansturen van de gripper

Primary actor:	Driver
Brief description:	Het systeem reageert op een inkomend bericht omtrent het aansturen van de gripper.
Preconditions:	De robot heeft een 'start'-bericht ontvangen en verwerkt (bevindt zich in een bekende positie).
Postconditions:	De gripper is aangestuurd met de gevraagde PWM waarde.
Actor action	System responsibility
1. Stuurt een nieuwe gripper waarde.	2. Verwerkt het ontvangen bericht.
	3. Kijkt of de waarde binnen de uiterste vallen.
	4. Kijkt of de robot klaar is om een beweging te maken.
	5. Activeert de gripper met de opgegeven PWM waarde.
Alternative flow 01:	
	<i>De waarde valt niet binnen de uiterste.</i>
	4a. Er zal een bericht naar de driver gestuurd worden waarin vermeld wordt dat de opgegeven waarden niet uitgevoerd kunnen worden.
Alternative flow 02:	
	<i>De robot is niet klaar om een beweging te maken.</i>
	5a. Het bericht wordt toegevoegd aan een queue en zal op een later moment uitgevoerd worden.

Usecase 5 - Fully dressed usecase description - UC04

3.1.6. UC0_5 - Robot pauzeren

Primary actor:	Driver
Brief description:	Het systeem reageert op een inkoment bericht omtrent het pauzeren van de robot. Het systeem zal de robot pauzeren voor de aangegeven tijd.
Preconditions:	De robot heeft een 'start'-bericht ontvangen en verwerkt (bevind zich in een bekende positie).
Postconditions:	De robot heeft gepauzeerd voor de aangegeven tijd.
Actor action	System responsibility
1. Stuurt een 'pauzeer'-bericht	2. Verwerkt het ontvangen bericht
	3. Kijkt of de robot klaar is om een beweging te maken.
	4. Pauzeert de robot voor het aangeven aantal milliseconden.
Alternative flow 01:	
	<i>De robot is niet klaar om een nieuwe beweging te maken.</i>
	4a. Het bericht wordt toegevoegd aan een queue en op een later moment zal het bericht uitgevoerd worden.

Usecase 6 - Fully dressed usecase description - UC05

3.2. UC1 - Instellen van de configuratie

De usecases die vallen onder dit onderdeel zijn allemaal soortgelijk in uitvoering. Door deze gelijksoortigheid is er gekozen om een enkele 'fully dressed' uitwerking te maken.

Primary actor:	Driver
Brief description:	Het systeem verwerkt een configuratie bericht.
Preconditions:	Er is een seriele verbinding met de driver.
Postconditions:	De configuratie is verwerkt in het systeem.
Actor action	System responsibility
1. Stuurt een configuratie bericht.	2. Verwerkt het bericht.
	3. De bijhorende waarde in het systeem wordt aangepast naar de waarde vermeld in het ontvangen bericht.

Usecase 7 - Fully dressed usecase description - Instellen van configuratie

3.3. UC2 - Reageren op requests

De usecases die vallen onder dit onderdeel zijn allemaal soortgelijk in uitvoering. Door deze gelijksoortigheid is er gekozen om een enkele 'fully dressed' uitwerking te maken.

Primary actor:	Driver
Brief description:	Het systeem reageert op een inkomend data request bericht. Het systeem zal een reponse opzetten en terug sturen naar de driver.
Preconditions:	Er is een seriele verbinding met de driver.
Postconditions:	Een response met de corresponderende data is verstuurd naar de driver.
Actor action	System responsibility
1. Stuurt een data request bericht	2. Verwerkt het bericht.
	3. Stelt een bericht samen met de bijhorende waardes.
	4. Verstuurd het bericht naar de driver.

Usecase 8 - Fully dressed usecase description - Reageren op requests

3.4. UC3 - Uitvoeren Noodstop

Primary actor:	Driver
Brief description:	Het systeem ontvangt een noodstop bericht en zal vervolgens direct alle beweging van de robot staken.
Preconditions:	Er is een seriele verbinding met de driver.
Postconditions:	De robot is direct gestopt en alle motoren zijn uitgeschakeld.
Actor action	System responsibility
1. Stuurt een 'noodstop'-bericht.	2. Identificeert het bericht als 'noodstop'.
	3. Schakeld direct alle motoren uit.
	4. Verwijdert alle opgeslagen bewegingsberichten uit de queue.

Usecase 9 - Fully dressed usecase description - UC

4. Requirements

In dit hoofdstuk zullen de verschillende (niet-) functionele requirements beschreven worden van firmware applicatie.

4.1. Functionele requirements

Index	Prioriteit	Requirement	Voldaan
FR-01	Must	Het systeem moet de mogelijkheid bieden tot het opvragen van de maximale positie per motor.	✓
FR-02	Must	Het systeem moet de mogelijkheid bieden tot het opvragen van de minimale positie per motor.	✓
FR-03	Must	Het systeem moet de mogelijkheid bieden tot het opvragen van de huidige snelheid per motor.	✓
FR-04	Must	Het systeem moet de mogelijkheid bieden tot het opvragen van de huidige acceleratie per motor.	✓
FR-05	Must	Het systeem moet de mogelijkheid bieden tot het opvragen van de huidige positie per motor.	✓
FR-06	Must	Het systeem moet de mogelijkheid bieden tot het opvragen van de huidige gripper PWM waarde.	✓
FR-07	Must	Het systeem moet de mogelijkheid bieden tot het bewegen van de robot aan de hand van verschillende hoeken.	✓
FR-08	Must	Het systeem moet de mogelijkheid bieden tot het bewegen van de robot aan de hand van een cartesische positie (x,y,z).	✓
FR-09	Must	Het systeem moet de mogelijkheid bieden tot het instellen van de huidige gripper PWM waarde.	✓
FR-10	Must	Het systeem moet de mogelijkheid bieden tot het stoppen van de robot.	✓
FR-11	Must	Het systeem moet de mogelijkheid bieden tot het direct stoppen van de robot (noodstop).	✓
FR-12	Must	Het systeem moet commando's kunnen opslaan en later uitvoeren.	✓

Index	Prioriteit	Requirement	Voldaan
FR-13	Should	Het systeem moet de mogelijkheid bieden tot het instellen van globale snelheid.	✓
FR-14	Should	Het systeem moet de mogelijkheid bieden tot het instellen van globale acceleratie.	✓
FR-15	Should	Het systeem moet de mogelijkheid bieden tot het instellen van de snelheid per motor.	✓
FR-16	Should	Het systeem moet de mogelijkheid bieden tot het instellen van de accleratie per motor.	✓
FR-17	Should	Het systeem moet de mogelijkheid bieden tot het (de)activeren van de <i>error</i> berichten.	✓
FR-18	Should	Het systeem moet de mogelijkheid bieden tot het (de)activeren van de <i>warning</i> berichten.	✓
FR-19	Should	Het systeem moet de mogelijkheid bieden tot het (de)activeren van de <i>info</i> berichten.	✓
FR-20	Should	Het systeem moet de mogelijkheid bieden tot het (de)activeren van de <i>debug</i> berichten.	✓
FR-21	Should	Het systeem moet de mogelijkheid bieden tot het instellen van de <i>*frame height offset</i> .	✓
FR-22	Should	Het systeem moet de mogelijkheid bieden tot het instellen van de <i>*gripper height offset</i> .	✓
FR-23	Should	Het systeem moet de mogelijkheid bieden tot het instellen van de <i>*idle position</i> per motor.	✓
FR-24	Should	Het systeem moet de mogelijkheid bieden tot het opvragen van de huidige <i>*frame height offset</i> .	✓
FR-25	Should	Het systeem moet de mogelijkheid bieden tot het opvragen van de huidige <i>*gripper height offset</i> .	✓
FR-26	Should	Het systeem moet de mogelijkheid bieden tot het opvragen of een motor momenteel actief is (aan het bewegen).	✓

Index	Prioriteit	Requirement	Voldaan
FR-27	Should	Het systeem moet de mogelijkheid bieden tot het opvragen van de huidige positie van de gripper in het cartesiaans coördinaten systeem (x,y,z).	✓
FR-28	Should	Het systeem moet de mogelijkheid bieden tot het pauzeren van de robot voor een bepaald aantal milliseconden.	✓
FR-29	Should	Het systeem moet de gebruiker laten weten wanneer een aangevraagde positie niet bereikbaar is. Bijvoorbeeld door een log bericht.	✓
FR-30	Should	Het systeem moet de gebruiker laten weten wanneer een aangevraagde gripper waarde niet mogelijk is, bijvoorbeeld door een log bericht.	✓
FR-31	Should	Het systeem moet de gebruiker laten weten wanneer een aangevraagde snelheid niet mogelijk is, bijvoorbeeld door een log bericht.	✓
FR-32	Should	Het systeem moet de gebruiker laten weten wanneer een aangevraagde acceleratie niet mogelijk is, bijvoorbeeld door een log bericht.	✓
FR-33	Should	Het systeem moet de gebruiker laten weten wanneer een aangevraagde <i>*idle positie</i> niet mogelijk is, bijvoorbeeld door een log bericht.	✓
FR-34	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de verschillende pin configuraties.	✓
FR-35	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de microstepping resolutie per motor.	✓
FR-36	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de <i>*steps per degree</i> waarde per radiale actuator.	✓
FR-37	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de <i>*steps per millimeter</i> waarde per lineaire actuator.	✓
FR-38	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de status van <i>error</i> berichten.	✓
FR-39	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de status van <i>warning</i> berichten.	✓

Index	Prioriteit	Requirement	Voldaan
FR-40	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de status van <i>info</i> berichten.	✓
FR-41	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de status van <i>debug</i> berichten.	✓
FR-42	Could	Het systeem moet de mogelijkheid bieden tot het opvragen van de <i>*idle positions</i> per motor.	✓

Tabel 1 - Functionele requirements

* *Frame height offset* staat voor het verschil tussen de laagst mogelijke positie van de robot en de daadwerkelijke ondergrond waar het apparaat zich op bevind.

* *Gripper height offset* staat voor het verschil in hoogte tussen het laagste deel van de gripper en het hoogste (bewegende) deel van de robot.

* *Idle positions* zijn de posities die de robot aanneemt wanneer er een stop commando is ontvangen.

* *Steps per degree* is het aantal stappen de motor moet maken voor het een graad is gedraaid.

* *Steps per millimeter* is het aantal stappen de motor moet maken voor het een millimeter is veranderd in hoogte.

4.2. Niet functionele requirements

In dit onderdeel worden de niet functionele requirements verder toegelicht. De niet functionele requirements zijn onderverdeeld op basis van quality of service (QoS). De verschillende QoS elementen; usability, reliability, performance en supportability, zijn in de onderstaande tabel uitgewerkt.

4.2.1. QoS - Usability

Index	Prioriteit	Onderdeel	Requirement	Voldaan
NFR-01	Must	Safety	Het systeem zal logs met de prioriteit: ERROR naar de gebruiker sturen, indien de gebruiker dit niet specifiek heeft gedeactiveerd.	✓
NFR-02	Must	Safety	Het systeem zal logs met de prioriteit: WARNING naar de gebruiker sturen, indien de gebruiker dit niet specifiek heeft gedeactiveerd.	✓
NFR-03	Must	Safety	Het systeem zal logs met de prioriteit: INFO naar de gebruiker sturen, indien de gebruiker dit niet specifiek heeft gedeactiveerd.	✓
NFR-04	Must	Safety	Het systeem zal logs met de prioriteit: DEBUG naar de gebruiker sturen, indien de gebruiker dit niet specifiek heeft gedeactiveerd.	✓
NFR-05	Must	Quality	Alle publieke methodes in het systeem zullen voorzien worden van beschrijvend doxygen (Doxygen, n.d.) commentaar.	✓
NFR-06	Must	Quality	Commentaar in code zal geschreven worden in de voertaal: engels.	✓
NFR-07	Should	Quality	De verschillende onderdelen in het systeem zullen worden onderverdeeld in namespaces.	✓
NFR-08	Should	Quality	De verschillende header documenten (*.h) zullen worden gegroepeerd op basis van namespace.	✓
NFR-09	Could	Quality	De applicatie zal voorzien worden van een installatiehandleiding.	✗
NFR-10	Could	Quality	De installatiehandleiding zal geschreven worden in de voertaal; engels.	✗
NFR-11	Could	Quality	De applicatie zal voorzien worden van een gebruikershandleiding.	✗

Index	Prioriteit	Onderdeel	Requirement	Voldaan
NFR-12	Could	Quality	De gebruikershandleiding zal geschreven worden in de voertaal; engels.	✗

Tabel 2 - niet functionele requirements - QoS Usability

4.2.2. QoS - Reliability

Index	Prioriteit	Onderdeel	Requirement	Voldaan
NFR-13	Must	Reliability	Het systeem zal voorzien zijn van mini- en maximale positionele waardes per motor.	✓
NFR-14	Must	Reliability	Het systeem zal rekening houden met de mini- en maximale positionele waardes door deze niet te overschreiden.	✓
NFR-15	Must	Reliability	Het systeem zal voorzien zijn van mini- en maximale snelheids waardes per motor.	✓
NFR-16	Must	Reliability	Het systeem zal rekening houden met de mini- en maximale snelheids waardes door deze niet te overschreiden.	✓
NFR-17	Must	Reliability	Het systeem zal voorzien zijn van mini- en maximale acceleratie waardes per motor.	✓
NFR-18	Must	Reliability	Het systeem zal rekening houden mde mini- en maximale accelerate waardes door deze niet te overschreiden.	✓
NFR-19	Must	Safety	Het systeem zal binnen 100 milliseconden tot stilstand komen in het geval van een noodstop.	✓
NFR-20	Could	Reliability	Het systeem zal altijd hetzelfde pad volgen vanaf een positie A naar een positie B	✓

Tabel 3 - niet functionele requirements - QoS Reliability

4.2.3. QoS - Performance

Index	Prioriteit	Onderdeel	Requirement	Voldaan
NFR-21	Must	<i>Timeliness</i>	Het systeem zal binnen 100 milliseconden en beweginscommando uitvoeren indien er geen andere beweginscommando's in de wachtrij staan.	✓
NFR-22	Must	<i>Timeliness</i>	Het systeem zal binnen 20 milliseconden een response hebben verstuurd wanneer er een data request ontvangen is.	✓

Tabel 4 - niet functionele requirements - QoS Performance

4.2.4. QoS - Supportability

Index	Prioriteit	Onderdeel	Requirement	Voldaan
NFR-23	Must	<i>Quality</i>	Het product zal ontwikkeld worden in de visual studio code extensie: PlatformIO.	✓
NFR-24	Should	<i>Quality</i>	De configureerbare aspecten van het systeem zullen een omschrijvende documentatie krijgen per onderdeel.	✗
NFR-25	Should	<i>Quality</i>	Het product zal voorzien worden van een document waarin beschreven staat welke eventuele bugs of fouten er aanwezig zijn.	✗
NFR-26	Should	<i>Quality</i>	Het product zal voorzien worden van een document waarin beschreven staat welke toekomstige verbeterpunten er zijn.	✗
NFR-27	Could	<i>Quality</i>	Het product zal voorzien worden van omschrijvende documentatie.	✓

Tabel 5 - niet functionele requirements - QoS Supportability

5. Literatuurlijst

Index	Source
1	Wikipedia-bijdragers. (2022a, mei 25). Microcontroller. Wikipedia. geraadpleegd op May 25, 2022, van https://en.wikipedia.org/wiki/Microcontroller
2	Wikipedia-bijdragers. (2020, september 15). Kloksnelheid. Wikipedia. geraadpleegd op May 25, 2022, van https://nl.wikipedia.org/wiki/Kloksnelheid
3	Wikipedia-bijdragers. (2022a, april 24). 32-bit computing. Wikipedia. geraadpleegd op May 25, 2022, van https://en.wikipedia.org/wiki/32-bit_computing
4	Wikipedia-bijdragers. (2022, february 10). Universal serial bus. Wikipedia. geraadpleegd op May 25, 2022, van https://nl.wikipedia.org/wiki/Universal_serial_bus
5	PJRC. (n.d.). Teensy® 4.0. Teensy 4.0. Retrieved May 25, 2022, from https://www.pjrc.com/store/teensy40.html
6	Arduino. (n.d.). Arduino Uno Rev3. Arduino Official Store. Retrieved May 25, 2022, from http://store.arduino.cc/products/arduino-uno-rev3
7	Doxygen. (n.d.). Doxygen Manual: Documenting the code. Retrieved May 25, 2022, from https://www.doxygen.nl/manual/docblocks.html