FSM Treadmill

Productdocument



|  |  |
| --- | --- |
| Student(en) | Jaap-Jan Groenendijk (548148)  Colin van Dreumel (2107036) |
| Vak | Inleiding Software Engineering, D-B-INSE-O |
| Docent | Olaf Roelofs |
| Datum | 29-11-2022 |

Documenthistorie

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Versie** | **Wie** | **Veranderingen** |
| 29-11-2022 | 0.1 | Colin van Dreumel  Jaap-Jan Groenendijk | Start document |
| 21-12-2022 | 0.2 | Jaap-Jan Groenendijk | Aanvulling van eisen en acceptatietesten |
| 10-01-2023 | 1.1 | Jaap-Jan Groenendijk | Laatste check, aanvullen testen |
| 10-01-2023 | 1.1 | Colin van Dreumel  Jaap-Jan Groenendijk | Checklist en evaluatie |
|  |  |  |  |
|  |  |  |  |

# Samenvatting

Door dhr. C. van Dreumel en dhr. J.J. Groenendijk is een Finite State Machine (hierna afgekort met FSM) op basis van een loopband.

Dit document beschrijft de ontwikkeling van de besturingssoftware voor een loopband. De software is ontworpen om gebruikers in staat te stellen om de conditie te verbeteren door te lopen op een loopband die naar wens kan worden ingesteld op snelheid en steilheid. Functionele eisen zijn opgesteld voor zowel gebruikers als beheerders, en technische eisen zijn opgesteld voor het ontwikkelen van de software. Het ontwerpproces van de software is beschreven, inclusief de architectuur en state chart. Het realisatie- en testproces is ook beschreven, waarbij de software gerealiseerd is en acceptatietesten zijn uitgevoerd. Tot slot wordt het eindresultaat en aanbevelingen voor verdere verbeteringen besproken.

De basisfunctionaliteit van het programma zijn afgerond en getest. Het programma heeft het beoogde doel van het simuleren van een loopband middels een finite state machine framework behaald. Het programma heeft ruimte voor verbeteringen, maar deze lijken vooral buiten de scope van dit project te liggen.

# Voorwoord

Dit project wordt uitgevoerd door Colin van Dreumel en Jaap-Jan Groenendijk. Jaap-Jan was in zijn vorige functie systeembeheerder, waar hij 5 jaar actief was in de ICT-dienstverlening. Colin heeft in een vorige functie ervaring opgedaan als applicatieontwikkelaar en heeft hierdoor al een respectabele kennis van meerdere programmeertalen. Beide hebben we technische ervaring vanuit een vorige functie, maar nog geen ervaring met het programmeren van microcontrollers in C. Door dit project uit te voeren als opdracht, zal de ervaring uitgebreid worden. We hopen hier dan ook veel van te leren.

In de toekomst zullen wij eerder starten met het maken van een planning. We hebben gemerkt dat we opdrachtonderdelen eerder hadden kunnen afronden als we eerder de moeite hadden genomen om samen een planning op te stellen en met elkaar door te nemen. Bij het maken van een strakke planning kunnen we onafhankelijk van elkaar werken in onze eigen tijd. Overleggen over de voortgang van het project en de planning wordt tijdens college-uren uitgevoerd en verloopt prima.

Inhoudsopgave

[Samenvatting 4](#_Toc122552172)

[Voorwoord 5](#_Toc122552173)

[1. Inleiding 7](#_Toc122552174)

[2. Definitiefase 8](#_Toc122552175)

[2.1 Kennismaking en doel van treadmill 8](#_Toc122552176)

[2.2 Functionele eisen 8](#_Toc122552177)

[2.3 Technische eisen 8](#_Toc122552178)

[2.4 Schets user interface 9](#_Toc122552179)

[3. Ontwerpen 10](#_Toc122552180)

[3.1 Architectuur 10](#_Toc122552181)

[3.2 State chart 10](#_Toc122552182)

[4. Realisate en testen 12](#_Toc122552183)

[4.1 Realisatie 12](#_Toc122552184)

[4.2 Acceptatietesten 12](#_Toc122552185)

[5. Eindresultaat en aanbevelingen 14](#_Toc122552186)

[5.1 Eindresultaat 14](#_Toc122552187)

[5.2 Aanbevelingen 14](#_Toc122552188)

# Inleiding

In dit document wordt beschreven hoe de besturingssoftware voor een treadmill is ontwikkeld. In hoofdstuk 1 wordt ingegaan op de inleiding van het project. Hoofdstuk 2 gaat in op de gestelde eisen aan de software en bevat onder andere kennismaking met de treadmill, functionele eisen voor zowel gebruikers als beheerders, en technische eisen voor het ontwikkelen van de software. Hoofdstuk 3 behandelt het ontwerpproces van de software, inclusief de architectuur en state chart. In hoofdstuk 4 wordt het realisatie- en testproces beschreven, waarbij de software gerealiseerd wordt en acceptatietesten worden uitgevoerd. Ten slotte, in hoofdstuk 5, wordt het eindresultaat en de aanbevelingen voor verdere verbeteringen besproken.

# Definitiefase

Dit hoofdstuk gaat in op de gestelde eisen aan de te realiseren besturingssoftware van een treadmill.

## Kennismaking en doel van treadmill

Een treadmill is een essentieel onderdeel van iedere sportschool. Gebruikers (sporters in een sportschool) van een treadmill kunnen lopen op een band die ronddraait. Het draaien van de band, de snelheid en de steilheid, is naar wens van de gebruiker in te stellen. Gebruikes van een treadmill willen door te lopen de conditie verbeteren of gezonder worden.



Figuur 1 - Een Star Trac Treadmill, de inspiratie voor dit project

## Functionele eisen

In dit document worden functionele specificaties onderscheiden die relevant zijn voor gebruikers en beheerders. Hieronder volgt een samenvatting van de belangrijkste punten:

Functionele specificaties die belangrijk zijn voor gebruikers:

* Fit worden door (veel) te lopen op een loopband.
* Starten/stoppen van de loopband met behulp van een start/stop-knop.
* Aanpassen van de snelheid via +/--knoppen.
* Aanpassen van de tilt via ↑/↓-knoppen.
* Pauze met behulp van een pauze-knop.
* Emergency stop met behulp van een sensor.

Functionele specificaties die belangrijk zijn voor beheerders/eigenaren:

* Diagnostische mogelijkheid met een servicecode om reparaties/aanpassingen uit te voeren.
* Mogelijkheid om firmware-updates uit te voeren.
* Mogelijkheid om het emergency-signaal te testen.

## Technische eisen

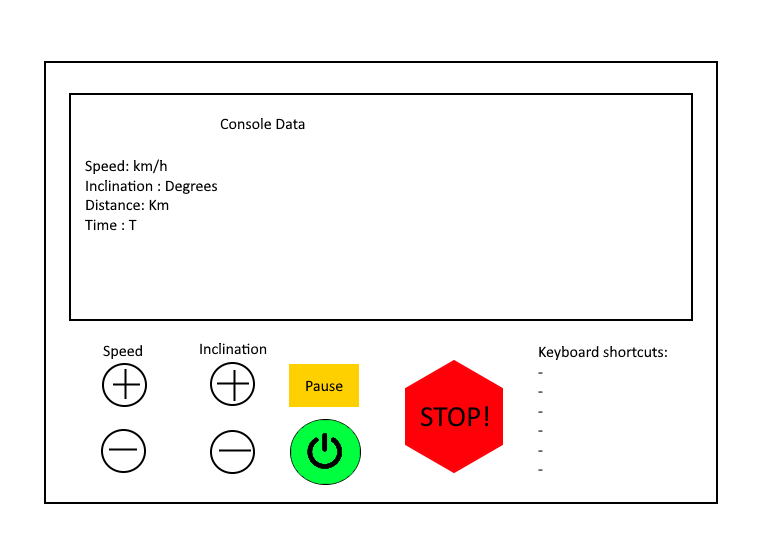
Voor dit project zijn er een aantal technische vereisten opgesteld die moeten worden gevolgd bij het ontwikkelen van de software. De vereisten zijn opgesteld aan de hand van de studiegids. Zo moet de code geschreven worden in de programmeertaal C en moet de QTCreator IDE worden gebruikt. De loopband zal moeten werken op 230 volt. De code voor het project zal op github worden gehost, zodat deze gemakkelijk toegankelijk is voor de ontwikkelaars.

Het is belangrijk dat het programma compileert zonder fouten en dat de code tijdens de uitvoering niet crasht. De code moet grondig getest worden om ervoor te zorgen dat alle staten correct werken. Het is ook belangrijk dat de code functioneel correct is en dat de gebruikersinterface nuttige informatie bevat. De invoer van de gebruiker moet gevalideerd worden en de gebruiker moet de optie hebben om verkeerde invoer te corrigeren.

Er mag geen boilerplate code aanwezig zijn en de code moet zich bevinden in een enkele directory die hernoemd kan worden. Het is aan te bevelen om gebruik te maken van een modulaire opzet voor de code. De code moet het resultaat zijn van de FSM state chart. Bovendien moeten de coding guidelines van Jos Onokiewicz worden gebruikt en moet er een codeformatter worden gebruikt in QTCreator. De namen van functies, variabelen en constanten moeten informatief zijn.

## Schets user interface

Een digitale schets is opgesteld om een beeld te schetsen van de werking van de user interface. Gebruikers van een loopband zullen de knoppen in onderstaande illustratie (figuur 2) gebruiken om de loopband te bedienen.



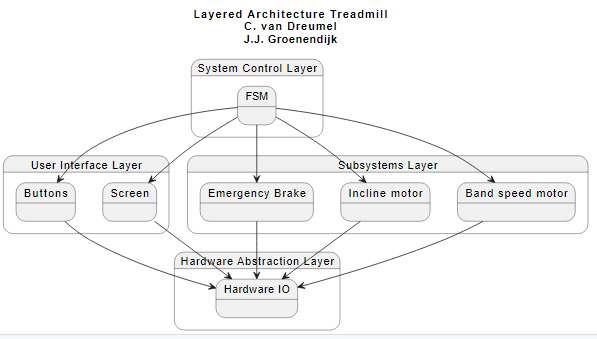
Figuur 2 - Voorbeeldschets user interface loopband

# Ontwerpen

Dit hoofdstuk gaat in op het ontwerp van een loopband met behulp van WMA Layered architecture & State chart.

## Architectuur

Onderstaande diagram (figuur 3) beschrijft de architectuur van een loopband met meerdere lagen. De bovenste laag is de gebruikersinterface laag, die bestaat uit het scherm en de knoppen. De volgende laag is de systeemcontrolelaag, die verantwoordelijk is voor de finite state machine (FSM). De laag daaronder is de subsystemenlaag, die bestaat uit de snelheidsmotor van de band, de kantelmotor en de noodrem. Ten slotte is er de hardware-abstractielaag, die verantwoordelijk is voor de hardware-io. De FSM communiceert met alle onderliggende lagen, zoals de subsystemenlaag en het scherm en de knoppen in de gebruikersinterface laag. De subsystemenlaag communiceert vervolgens met de hardware-abstractielaag.

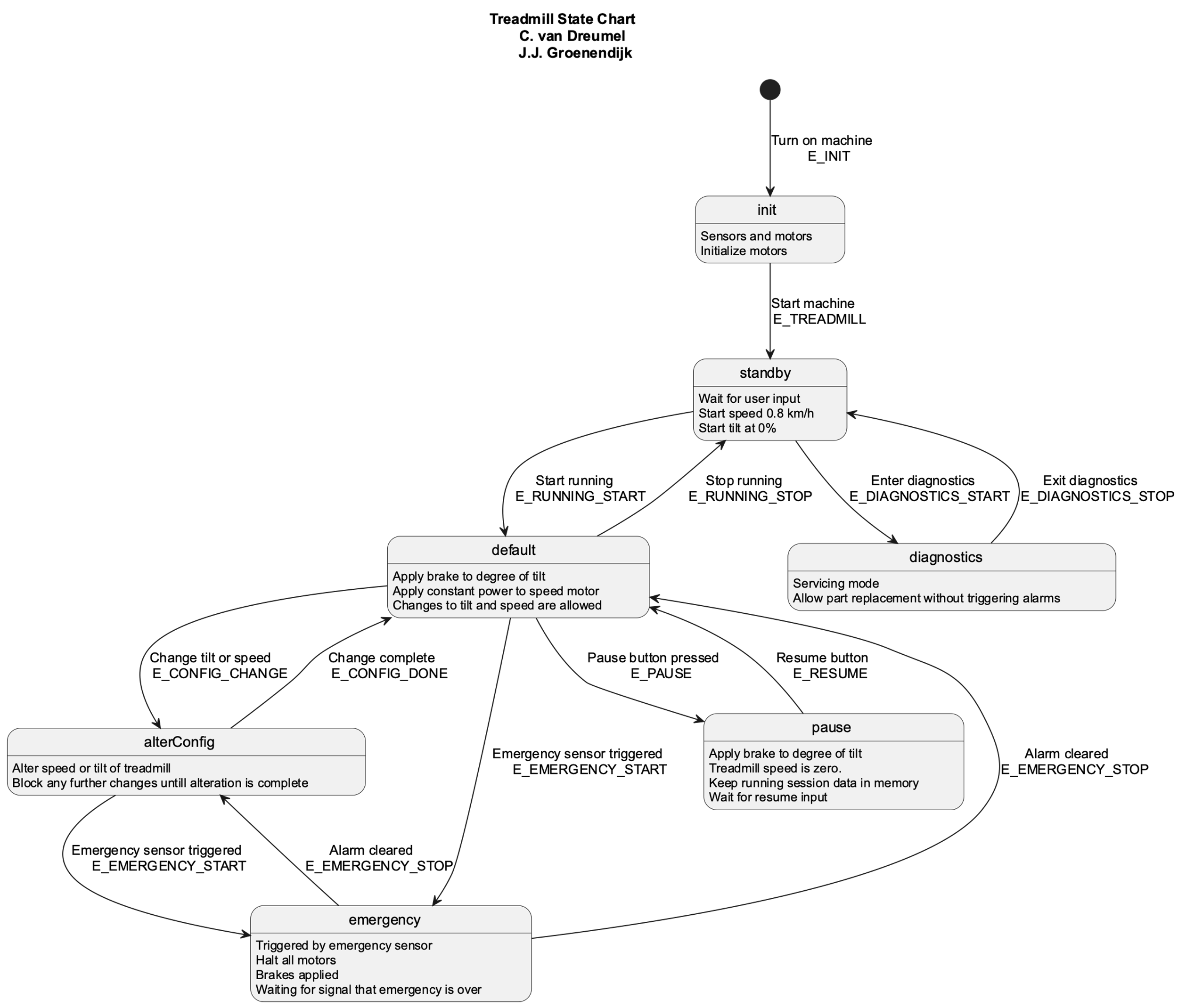


Figuur - Layered Architecture Treadmill

De meest recente versie van de broncode van het UML-diagram is hier te vinden:

<https://github.com/jjgroenendijk/fsm-treadmill/blob/main/uml/treadmill-architecture.uml>

## State chart

In onderstaande diagram (figuur 4) wordt de werking van een loopband weergegeven. Er zijn verschillende staten waarin de loopband zich kan bevinden, zoals S\_INIT, S\_STANDBY, S\_DEFAULT, S\_DIAGNOSTICS, S\_ALTERCONFIG, S\_EMERGENCY en S\_PAUSE. Als de loopband aangezet wordt, gaat hij naar de S\_INIT-staat, waarbij de sensoren en motoren worden geïnitialiseerd. Als de machine gestart wordt, gaat hij naar de S\_STANDBY-staat, waarbij hij wacht op gebruikersinvoer. Als het hardlopen begint, gaat de loopband naar de S\_DEFAULT-staat, waarbij de rem op het juiste tiltniveau wordt aangebracht en een constante kracht naar de snelheidsmotor wordt geleverd. Veranderingen in de snelheid en het tiltniveau zijn hier mogelijk. Als het hardlopen stopt, gaat de loopband terug naar de S\_STANDBY-staat. Als de gebruiker de diagnosemodus inschakelt, gaat de loopband naar de S\_DIAGNOSTICS-staat, waarbij onderhoud uitgevoerd kan worden zonder alarmsignalen te triggeren. Als de gebruiker de snelheid of het tiltniveau wil veranderen, gaat de loopband naar de S\_ALTERCONFIG-staat, waarbij verdere veranderingen worden geblokkeerd totdat de verandering voltooid is. Als er een noodsituatie optreedt, bijvoorbeeld als een noodsensor geactiveerd wordt, gaat de loopband naar de S\_EMERGENCY-staat, waarbij alle motoren gestopt worden en de remmen worden aangebracht. Als de noodsituatie voorbij is, gaat de loopband terug naar de S\_DEFAULT-staat. Als de gebruiker de pauzeknop indrukt, gaat de loopband naar de S\_PAUSE-staat, waarbij de snelheid op nul wordt gezet en de hardloopgegevens in het geheugen worden bewaard. Als de gebruiker op de hervatknop drukt, gaat de loopband terug naar de S\_DEFAULT-staat.

Figuur 4 - State Chart diagram van een loopband

Een bijgewerkte versie van de broncode voor de state chart UML is hier te vinden:

<https://github.com/jjgroenendijk/fsm-treadmill/blob/main/uml/treadmill-state-chart.uml>

# Realisate en testen

Dit hoofdstuk bevat de test omgeving voor de loopband en de realisatie van het project.

Om de loopbnd met alle juiste eisen te voldoen is er een test uitgevoerd met vershillende scenario's om te kijken hoe de machine reageert.

## Realisatie

Voor het realiseren van de fsm-software is gebruik gemaakt van QTCreator als ontwikkelomgeving (IDE). Github is gebruikt als hostingplatform voor de code. Versiebeheer is uitgevoerd met behulp van git. De git integratie in Qtcreator werd hierin als zeer handig ervaren. De inhoud van het .pro-user-bestand is genegeerd en toegevoegd aan het .gitignore-bestand. Voor het maken van de UML-diagrammen is de website planttext.com gebruikt. Code-documentatie en uitleg is uitgevoerd met behulp van ChatGPT. Voor het realiseren van de fsm-software is ook het fsm-framework 0.2 gebruikt. De directorystructuur van het verstrekte framework is behouden.

Instellingen van Qt Creator zijn voor zover mogelijk op standaard gelaten. Alleen de configuratie van het geleverde FSM-framework is overgenomen. Het FSM-framework is door hoge school HAN geleverd.

Code documentatie is grotendeels opgesteld door commentaar in de code. Het commentaar in de code is met behulp van Doxygen verwerkt naar een een stukje documentatie in html formaat. De output van Doxygen HTML bevat meer dan 3000 kleine bestanden en is omwille van beheerbaarheid in een zip bestand geplaatst. Gebruikers, beheerders en programmeurs kunnen deze documentatie vinden door bestand html.zip uit te pakken en index.html te openen.

Binnen het project wordt gebruik gemaakt van verschillende header en source files. Deze bestanden bevinden zich in de app folder. Header bestanden hebben een .h extensie. C bestanden hebben een .C extensie.

## Acceptatietesten

Acceptatietests zijn tests die worden uitgevoerd om te bepalen of een product of systeem aan de eisen voldoet die zijn opgesteld in de acceptatiecriteria. Deze tests zijn vaak gericht op het testen van de functionele eisen van het product of systeem, zoals de gebruikersinterface, de prestaties, de beveiliging en de integratie met andere systemen. Het doel van acceptatietests is om ervoor te zorgen dat het product of systeem voldoet aan de verwachtingen van de gebruikers en aan de acceptatiecriteria die zijn opgesteld door de opdrachtgever of de organisatie.

De volgende testen zijn een korte reeks aan toetsingen van basisfunctionaliteiten. Deze testen zijn opgesteld door de auteurs van het programma, niet door door een onafhankelijk test team. De testen zijn uitgevoerd tijdens het opstellen en afronden van het document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Action/input** | **Expected result** | **Pass Fail** | **Actual result if test has failed** |
| 1. | Code compileert correct | Ja | Pass | N/A |
| 2. | Programma crasht tijdens uitvoering | Nee | Fail | Task failed successfully. Programma crasht niet. |
| 3. | Alle states van het FSM-model komen terug in het programma | Ja | Pass | N/A |
| 4. | User input wordt gevalideerd. | Ja | Fail | Alle input wordt geaccepteerd |
| 5. | Code voldoet aan coding guidelines | Ja | Pass | N/A |
| 6. | Code legt zichzelf uit met commentaar | Ja | Pass | N/A |
| 7. | Gebruiker kan user input aanpassen indien nodig | Ja | Pass | N/A |
| 8. | System error bits worden gebruikt? | Ja | Fail | Error bits worden niet gebruikt |
| 9. | Programma komt niet in een soft lock (bricked) tijdens de uitvoering | Ja |  |  |

Bovenstaande testen zijn een kleine collectie voor het valideren van de basiswerking van het programma. Aan de hand van de behaalde resultaten is te stellen dat het programma het doel heeft bereikt en zonder crashes kan draaien. Tot op heden zijn nog geen situaties ontdekt waarin het programma onder normale omstandigheden zal crashen. Dit biedt nog geen uitsluitsel voor de werking in de toekomst wanneer andere gebruikers aan de slag gaan met dit programma of wanneer wijzigingen worden gedaan in de toekomst.

In versie 1.1 zitten nog enkele visuele bugs die het gebruik moeilijker kunnen maken, maar normaal gebruik niet zullen verhinderen. Over de visuele bugs wordt meer geschreven in de aanbevelingen. In theorie kunnen testen opgesteld worden voor het visuele aspect voor het programma, maar dat ligt vooralsnog buiten de scope van dit project.

# Eindresultaat en aanbevelingen

Dit hoofdstuk bevat een korte samenvatting van de behaalde testresultatenen en aanbevelingen op basis van de resultaten. Ook bevat dit hoofdstuk enkele persoonlijke aanbevelingen van de auteurs van dit programma.

## Eindresultaat

Het programma fsm-treadmill v1.1 simuleert succesvol een loopband. Het programma heeft ruimte voor verbeteringen, maar is in de basis compleet. Het programma is volgens de auteurs van dit document eenvoudig in gebruik en biedt ruimte voor het doorontwikkelen van een loopband.

## Aanbevelingen

Openstaande taken die verder uitgevoerd zijn discutabel en vereisen teamoverleg voordat deze opgepakt worden. Mogelijke vervolg taken zijn:

* Overleg over prioriteiten van opvolgende taken.
* Bugs oplossen.
* Visuele bugs oplossen.

Op het moment van afronden van dit project lijken enkele verbeterpunten voor het project als volgt te zijn:

* Code efficiënter maken.
* Code korter maken door herhaalde code te vervangen door enkele functies.
* Commentaar laten controleren door een tweede of derde partij en waar nodig uitbreiden.

Bij het afronden van het project zijn enkele bugs vastgesteld die niet opgelost zijn. Deze bugs zijn als volgt:

* Momenteel zijn er nog een paar visuele bugs met het DPS-show systeem. Aanbevolen om de stats voor DPS-show in een aparte functie te zetten zonder /n.
* Afstand wordt niet correct berekend in de change config-functie door wijziging van variabelen. Aanbevolen om updateDis te triggeren met config-wijziging.

Wensen van de auteurs voor een toekomstige versie zijn bij een evaluatie vastgesteld en als volgt opgesomd:

* Bugs oplossen
* Compatibiliteit met Linux
* Gebruikersinterface verbeteringen met kleuren.