Lasmachine 

produktdocument



Student: <studentnaam>

<studentnummer>

Vak: Inleiding Software Engineering, D-B-INSE-O

Docent: <docentnaam>

<datum -2018>

**Documenthistorie**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Versie** | **Wie** | **Veranderingen** |
| ..-..-2018 | v0.1 | …. | Initiële versie. |
|  | v0.2 |  | Invulling definitiefase |
|  | v0.3 |  | Alle hoofdstukken, review collega |
|  | v0.4 |  | Review docent, voorbereiding definitieve versie v1.0 |
|  |  |  |  |
|  |  |  |  |

**Samenvatting**

Bij AWL-Techniek in Harderwijk worden geautomatiseerde lasmachines gebouwd. De producten worden gelast doormiddel van lasrobots. De producten die gelast worden zijn bijvoorbeeld autostoelen of hekwerken voor industriële doeleinden. De machines worden normaal gesproken door een PLC aangestuurd. Tijdens dit project is de besturing van een lasmachine in de taal C geprogrammeerd.

De besturing van de ‘lasmachine’ is geprogrammeerd in de taal C in QtCreator. Het project is een deel van een lasmachine omdat het anders te complex zou worden. Het gedeelte waar de software voor geprogrammeerd is zijn een lasrobot, handlingsrobot met de bijbehorende veiligheden. Dit gedeelte van de machine last een hekwerk, die in delen in de machine is gelegd, in elkaar doormiddel van de lasrobot. Doormiddel van de handlingrobot wordt het hekwerk uit de machine gehaald.

De documentatie is gemaakt doormiddel van het programma Doxygen, hiermee wordt doormiddel van de C code automatisch documentatie gemaakt.

De machine is gemakkelijk te bedienen voor de bediener doormiddel van een menu waaruit gekozen kan worden wat de machine gaat doen.

De code is overzichtelijk geprogrammeerd door gebruik te maken van headerfiles en commentaar.

De testen hebben aangetoond dat ???

**Voorwoord**

Voor u ligt het rapport ‘lasmachine’ in het kader van de module software engineering. Voor deze module begon had ik bijna geen ervaring met C-Programmeren. Dit is mij dan ook een beetje tegengevallen. Maar met de hulp van het leerteam en terugkoppeling van de leraar is het toch gelukt om tot een goed eindresultaat te komen. Voordat ik aan het project begon heb ik een planning gemaakt om de opdracht op een gestructureerde manier af te ronden. Door deze planning is het gelukt om de opdracht op tijd af te ronden.

Bij dezen wil ik de leraar Jos Onokiewicz bedanken voor het verzorgen van deze module en voor de hulp bij het maken van de C code en het rapport voor de eindopdracht. Tevens wil ik mijn leerteam bedanken voor de hulp.

**Inhoudsopgave**

[1. Inleiding 6](#_Toc532822727)

[2. Definitiefase 7](#_Toc532822728)

[2.1 Achtergrond project 7](#_Toc532822729)

[2.2 Functionele specificatie 8](#_Toc532822730)

[2.3 Technische specificatie 8](#_Toc532822731)

[3. Ontwerpfase 10](#_Toc532822732)

[3.1 Layered Architecture 10](#_Toc532822733)

[3.1.1 User Interface Layer 10](#_Toc532822734)

[3.1.2 System Control Layer 10](#_Toc532822735)

[3.1.3 Subsystems Layer 10](#_Toc532822736)

[3.1.4 Hardware Abstraction Layer 10](#_Toc532822737)

[3.2 State machine 11](#_Toc532822738)

[4. Realisatie en testen 12](#_Toc532822739)

[4.1 Ontwikkelomgeving 13](#_Toc532822740)

[4.2 Userinterface 14](#_Toc532822741)

[4.3 Acceptatie test 14](#_Toc532822742)

[4.4 Documentatie 15](#_Toc532822743)

[5. Eindresultaat en aanbevelingen 16](#_Toc532822744)

[Bijlage: 17](#_Toc532822745)

[C - Source Code 17](#_Toc532822746)

[User Manual 17](#_Toc532822747)

# Inleiding

Het project ‘Lasmachine’ is een project dat een lasmachine bij AWL-Techniek wordt geprogrammeerd in de taal C. Bij AWL-Techniek worden veel lasmachines voor verschillende industrieën gebouwd. Deze machines worden over de hele wereld verkocht.

In dit rapport is te lezen hoe het bouwen van de software in C voor een statemachine bij AWL-Techniek is uitgevoerd. De statemachine die is gerealiseerd is een lasmachine. De lasmachine bestaat uit verschillende onderdelen waaronder 2 robots, een bedieningspaneel met een start en stopknop, een noodstop en een deur om bij de robots te komen.

De opbouw van het rapport is als volgt:

* In hoofdstuk 2 wordt de definitiefase beschreven. In de definitiefase staan de functionele en technische specificaties waarin te lezen is hoe de machine werkt en welke producten zijn gebruikt.
* In hoofdstuk 3 is de ontwerpfase beschreven. Hierin is te lezen hoe er tot een technische oplossing voor de lasmachine is gekomen en zijn er de diagrammen te zien waarin te zien is hoe de machine is opgebouwd.
* In hoofdstuk 4 is informatie te lezen over de realisatiefase en het testen. In de realisatiefase staat hoe het programma is opgebouwd en is te zien hoe de gebruikersomgeving is. Er staat kort beschreven hoe de testen zijn uitgevoerd en waar de testen te vinden zijn.
* In hoofdstuk 5 staan het eindresultaat en de aanbevelingen. Er wordt teruggekeken naar het project en daarmee wordt er een conclusie beschreven.
* Aan het einde van het rapport bevind zich de bijlage, de bijlage bevat de gebruikershandleiding van de lasmachine.

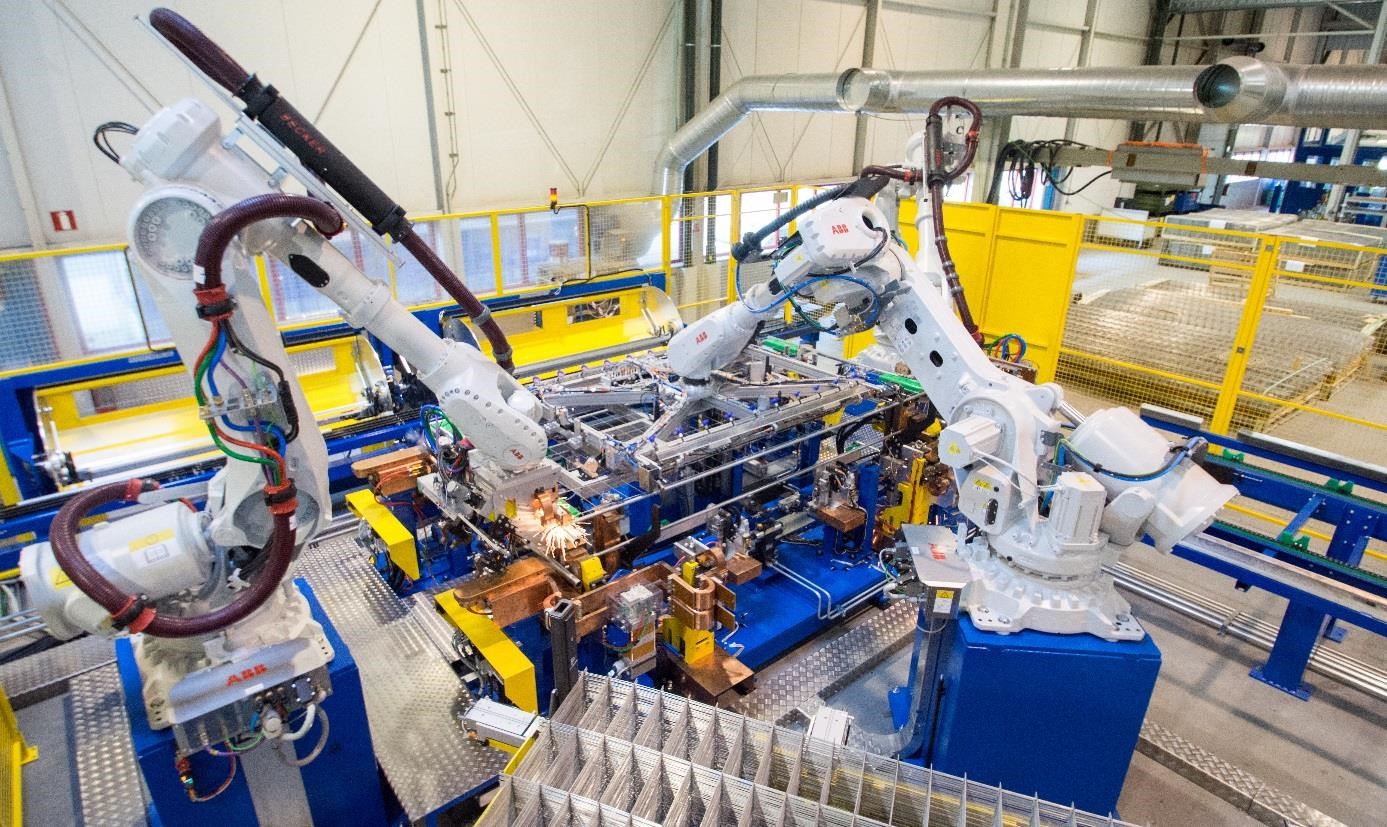
# Definitiefase

De lasmachine bestaat uit verschillende onderdelen en voert allerlei handelingen uit.

## Achtergrond project

De lasmachine wordt gebruikt voor het aan elkaar lassen van een metalen gaas aan een metalen frame. Hierdoor ontstaat een hekwerk. De delen worden door een operator in de machine gelegd, door een lasrobot aan elkaar gelast en door een handlingrobot uit de machine gehaald. Alleen de het frame van 200 x 200 cm en het gaas van 200 x 200 cm kunnen met de machine worden gelast. De robots zijn niet toegankelijk door een omkasting. In de omkasting zit een deur. Bij de operatorpositie aan de voorkant zit een bedieningspaneel met een noodstop, startknop, stopknop en een resetknop. De veiligheid is in dit project via de code geprogrammeerd om meldingen te ontvangen als er een onveilige situatie is ontstaan. Als de hardware van de machine ontwikkeld wordt zal de machine via veiligheidsrelais(speciale schakelingen voor een veiligheidscircuit in een machine) de veiligheid moeten regelen.

Figuur 1 is een afbeelding van de machine. De linker robot is de lasrobot en de rechter robot de handlingrobot.



*Figuur 1 De lasmachine met links de lasrobot en rechts de handlingrobot*

## Functionele specificatie

De machine behoort aan allerlei specificaties te voldoen. Denk hierbij aan het opstarten, stoppen en uitschakelen . Alle specificaties van de lasmachine zijn hieronder te lezen.

1. De lasrobot moet 2 delen aan elkaar kunnen lassen zodat een hekwerk ontstaat.
2. Simpele en begrijpelijke user interface
3. De machine heeft een opstartcyclus waarbij de machine geïnitialiseerd wordt.
4. Voor de veiligheid moet de deur dicht, met de deur open kan de machine het proces niet starten.
5. De noodstop moet ervoor zorgen dat de machine ten allen tijde kan stoppen in een onveilige situatie, hiermee moeten ook de robots stoppen.
6. Na een noodstop dienen de producten uitgenomen en de resetknop ingedrukt worden. De producten zijn niet meer te gebruiken omdat de las is onderbroken
7. De deur en noodstop zijn onderdeel van de veiligheid, na het onderbreken ervan moet de machine gereset worden. De software luistert hier alleen naar, de hardware voert de veiligheid uit doormiddel van volgens de machinerichtlijnen specificaties.
8. Als de delen ingelegd zijn is het nodig dat de startknop ingedrukt wordt om het proces te starten.
9. Als er na het lassen op de startknop gedrukt wordt, moet de handlingrobot het eindproduct uit de machine nemen.
10. De machine heeft een afsluitcyclus waarbij de machine apparatuur deinitialiseerd voordat de hele machine daadwerkelijk uitgaat.

## Technische specificatie

De lasmachine bestaat uit allerlei onderdelen en is met verschillende programma’s ontwikkeld. De volgende programma’s en merken zijn voor de lasmachine gebruikt.

1. Programmeertaal: C
2. Programmeersoftware: QtCreator
3. UML diagrammen editor: PlantUML
4. Robotmerk: ABB
5. Deurschakelaar: Euchner
6. Randapparatuur: Siemens

In figuur 2 is de user interface van de machine te zien. De gebruikers interactie is gedaan doormiddel van toetsenbord knoppen zoals:

1: Om deel 1 in de machine te leggen 2: Om deel 2 in de machine te leggen s: Noodstop d: Deur open

x: Programma uitschakelen



*Figuur 2 Schets van de user interface*

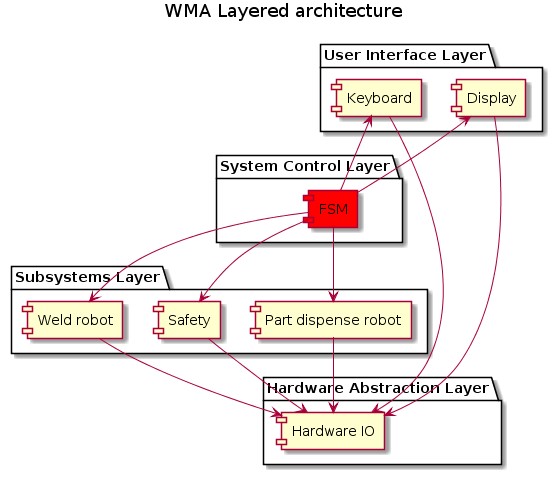
# Ontwerpfase

In dit hoofdstuk wordt het technisch ontwerp beschreven. Als eerste de layered architecture van de lasmachine. Hiermee wordt de hardware/software structuur weergegeven waarbinnen de lasmachine geprogrammeerd en getest is.

Hierna volgt de UML van de statemachine en de uitleg hoe de FMS is geprogrammeerd.

## Layered Architecture

In figuur 3 is te zien hoe het systeem in lagen is opgebouwd.



*Figuur 3 Layered architecture van de lasmachine*

### User Interface Layer

In deze laag kan de bediener de status van de machine bekijken en de machine aansturen doormiddel van drukknoppen.

### System Control Layer

In de FSM (finite state machine) zit de besturing van de machine, de states worden hier uitgevoerd. De machine kan zich in een state tegelijk bevinden en gaat naar een andere state doormiddel van een event. Ook worden in de FMS functies uitgevoerd.

### Subsystems Layer

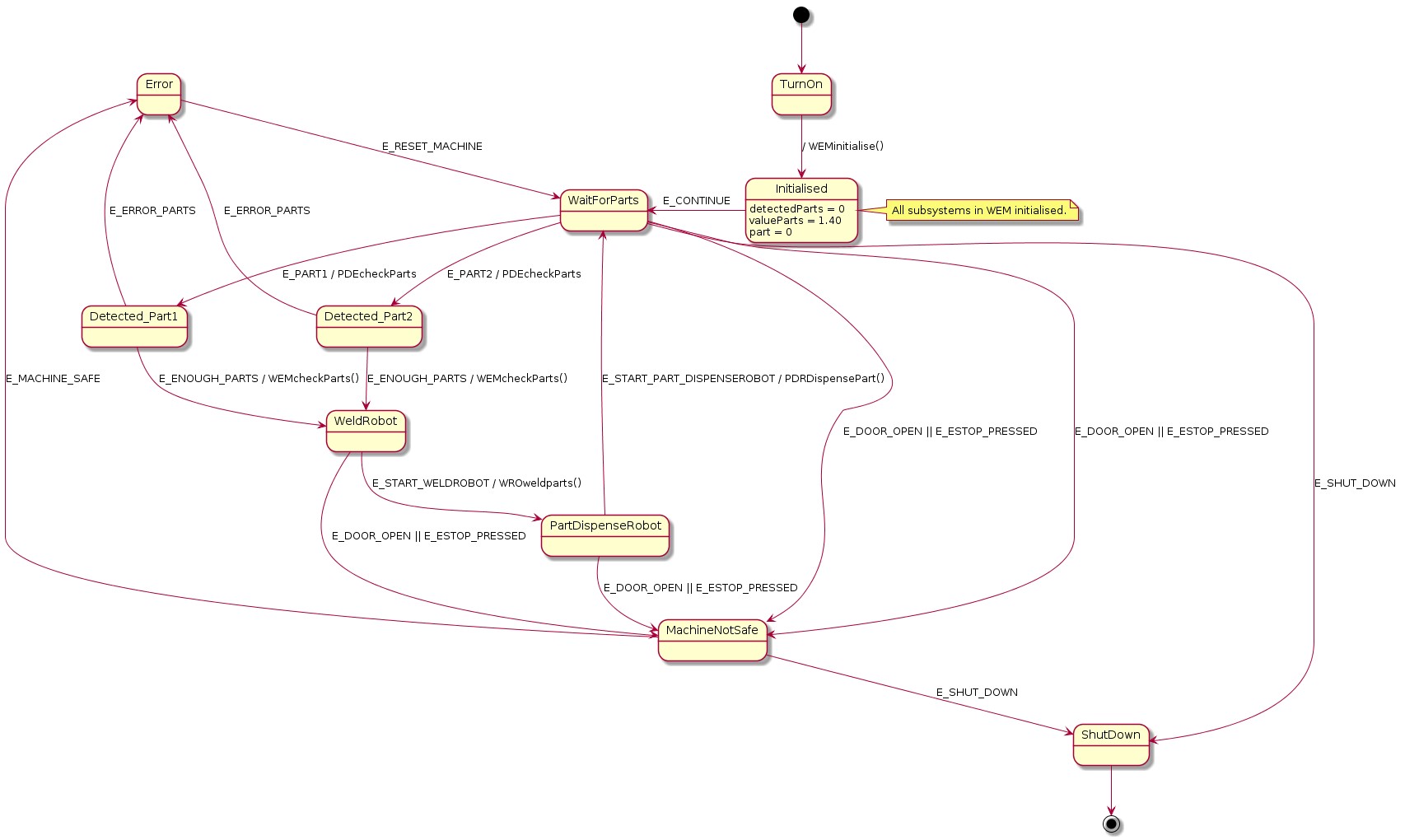
In deze laag bevinden zich de systemen die aangestuurd worden vanuit de FSM. Bijvoorbeeld de beide robots.

### Hardware Abstraction Layer

Hier wordt de daadwerkelijke hardware aangestuurd en ontvangen. Deze wordt aangestuurd vanuit de subsystemen, maar kan ook rechtstreeks via de User Interface Layer aangestuurd worden om bijvoorbeeld te testen.

## State machine

Het ontwerp van de statemachine is in een UML diagram weergegeven. Het UML diagram van de lasmachine is te zien in figuur 4. In het UML diagram is te lezen welke states de lasmachine heeft en welke events (bijvoorbeeld een toets op het toetsenbord) er voor nodig zijn om in de state te komen.



*Figuur 4 UML diagram lasmachine*

De eerste state waar de machine in komt na het opstarten is initialised. In deze state wordt de lasmachine geïnitialiseerd. De gedetecteerde delen wordt op 0 gezet omdat er nog geen delen in de machine leggen en de waarde van de 2 delen bij elkaar wordt hier ingesteld.

Na het initialiseren komt de machine in de state WaitForParts. In deze state wacht de machine tot er een deel wordt ingelegd. Vanuit deze state kan de machine uitgeschakeld worden. Ook is het mogelijk om, als er een onveilige situatie ontstaat, de noodstop te bedienen waardoor de machine naar de state MachineNotSafe gaat. Naar de state MachineNotSafe gaat de machine ook als de deur geopend wordt. Als de machine delen detecteert wordt er doormiddel van de functie WMAcheckparts bepaald of er genoeg delen in de machine liggen of dat de verkeerde delen in de machine liggen. De functie werkt door de waarde van de delen bij elkaar op te tellen. Als de 2 delen bij elkaar de valueParts is liggen er genoeg delen in de machine. Zolang de waarde onder de valueParts is, liggen er niet genoeg delen in de machine. Als de waarde boven de valueParts is dan liggen er teveel delen in de machine en gaat de machine naar de error state.

De states weldRobot en partDispenseRobot zijn eenvoudige states waarin de robots gestart kunnen worden doormiddel van de startknop. Vanuit deze state kan de noodstop bediend worden in een onveilige situatie. In de praktijk zal dan de robot stoppen met bewegen, bij deze statemachine is dat niet realiseerbaar omdat het niet communiceert met de robot.

Als de noodstop bediend is of de deur is geopend gaat de machine naar de state MachineNotSafe, om uit deze state te komen is ervoor gekozen om de gebruiker te laten kiezen met een toets op het toetsenbord of de machine veilig is. In de praktijk zal dit automatisch gedaan worden als de noodstop gedeactiveerd is of de deur wordt gesloten. Nadat de machine veilig is verklaard zal de machine naar de state Error gaan omdat de machine reset moet worden. Vanuit de state MachineNotSafe kan de machine uitgeschakeld worden.

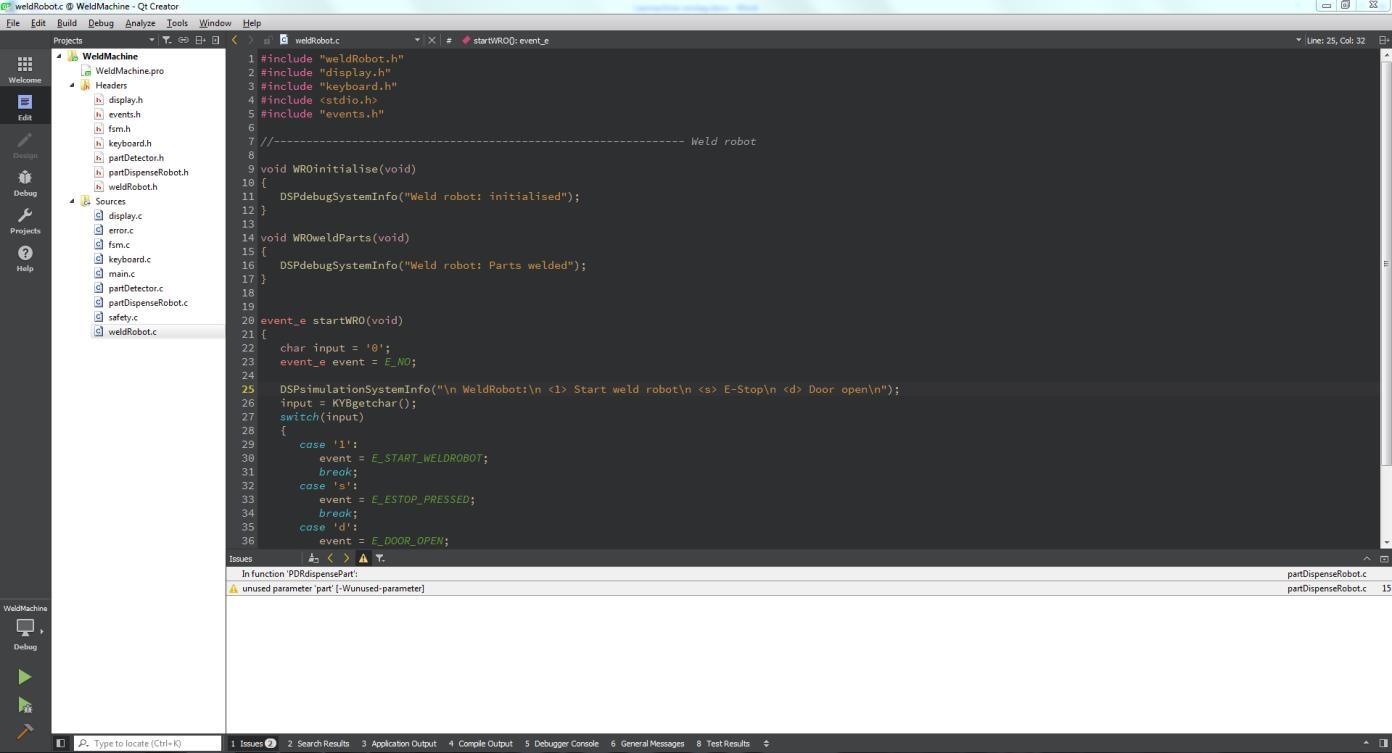
De state Error zorgt ervoor door de toets ‘r’ op het toetsenbord dat de machine reset wordt. Omdat de gebruiker de delen uit de machine moet halen in de state Error wordt de waarde van de delen die in de machine lagen op 0 gezet.

# Realisatie en testen

De manier waarop de lasmachine is gerealiseerd is in onderstaande hoofstukken te lezen. Om met de lasmachine aan alle juiste eisen van de klant te voldoen is er een test uitgevoerd. Hierbij zijn alle scenario’s getest die er zijn en wordt er gekeken hoe de machine hier op reageert.

## Ontwikkelomgeving

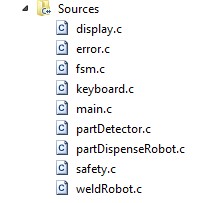
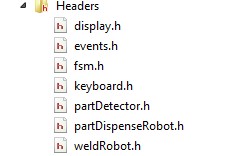
Het project ‘lasmachine’ is ontwikkeld met het programma QtCreator. In figuur 5 is te zien hoe de ontwikkelomgeving eruit ziet.



*Figuur 5 Ontwikkelomgeving van QtCreator*

In het linker gedeelte staat de .pro file met daaronder de header files en sources. Aan de rechterkant is de ontwikkelomgeving waar de code in geprogrammeerd wordt. Onderin staan de issues wat een belangrijke functie is van QtCreator om te kunnen debuggen.

In figuur 6 zijn de headerfiles te zien die bij de lasmachine horen en in figuur 7 de sources.



*Figuur 6 Headerfiles Figuur 7 Sources*

Hieronder staat de tekstfile van de .pro file:

TEMPLATE = app

CONFIG += console

CONFIG -= app\_bundle

CONFIG -= qt

CONFIG += c11

SOURCES += main.c \

fsm.c \ display.c \ keyboard.c \ partDetector.c \ partDispenseRobot.c \ weldRobot.c \ safety.c \

error.c

HEADERS += \

fsm.h \ display.h \ events.h \ keyboard.h \ partDetector.h \ partDispenseRobot.h \ weldRobot.h

## Userinterface

De userinterface zoals in figuur 8 staat afgebeeld bestaat uit een gedeelte waarin te zien is in welke state de machine zich bevindt. Onder dit gedeelte staan instructies voor de gebruiker. Deze instructies zijn bijvoorbeeld het indrukken van een toets of een handeling aan de machine zoals delen uitnemen.



*Figuur 8 Userinterface*

## Acceptatie test

Om de machine goed op te leveren aan de klant zijn de testen uitgevoerd. In figuur 9 is een testtabel te zien van de uitgevoerde testen.

De volgorde van het testen is als volgt:

* Test 1: Opstarten van de applicatie
* Test 2: Input van de gebruiker
* Test 3: Foute input van de gebruiker
* Test 4: Juiste delen in de machine
* Test 5: Verkeerde delen in de machine 
* Test 6: Afsluiten van de applicatie

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project name: WeldMachine** | | | | **Date:** 29.12.2017 | | |
| **Test case ID: 001** | | **Name:** | |  | | |
| **Function/module/program under test:**  **WeldMachine** | | | |  | | |
| **Test** | **Action/input** | | **Expected result** |  | **Pass**  **Fail** | **Actual result if test has failed** |
| 1. | Application start properly | | Displayed main manu |  | Pass |  |
| 2. | Application accept user input (all variants) | | Application accepted all defined imputs |  | Pass |  |
| 3. | Application handle user input errors | | Error message shown on the screen |  | Pass |  |
| 4. | function calculate parts works properly | | 1.40.  Part 1 = 1  Part 2 = 0.40 |  | Pass |  |
| 5. | Application handle wrong parts | | If value > 1.40  Wrong parts in machine  displayed |  | Pass |  |
| 6. | Application ends properly | | Shutdown screen displayed, deinitialisation of subsystems shown |  | Pass |  |

*Figuur 9 Test tabel*

In de tabel is te lezen dat de testen voldoen aan de eisen van de machine.

## Documentatie

De code documentatie is gemaakt met Doxygen. Doxygen maakt een HTML-pagina met daarin de code en de user-manual. Doormiddel van commentaar in de C-code wordt de pagina automatisch gecreëerd. Doxygen is te downloaden op [www.doxygen.org.](http://www.doxygen.org/)

# Eindresultaat en aanbevelingen

Het doel van de gerealiseerde besturing is een lasmachine aansturen doormiddel van een gebruiker. In deze lasmachine is veiligheid een belangrijk aspect. Er is aan alle eisen die gesteld zijn voldaan.

Voor de lasmachine zijn de volgende eisen gerealiseerd:

1. Opstart cyclus
2. Input van de gebruiker doormiddel van een toetsenbord
3. Een state waarin de lasmachine wacht op de ingelegde delen
4. State waarin weldRobot aangestuurd kan worden
5. State waarin partDispenseRobot aangestuurd kan worden
6. Veiligheidsstate waarin de machine terecht komt als de deur open is of als de noodstop is ingedrukt
7. Errorstate waarin de machine terecht komt als er een error voorkomt
8. Afsluit cyclus

De testen van de machine zijn allemaal juist doorlopen. De input van de gebruiker werkt goed en als de gebruiker een verkeerde input invoert dan wordt het teruggekoppeld op het display. De functie die de delen bij elkaar optelt werkt ook correct en zorgt ervoor dat er teruggekoppeld wordt of er voldoende, onvoldoende of de verkeerde delen in de machine liggen.

De safety in de machine werkt in de software op een juiste manier en geeft een melding op de display dat er wat aan de hand is met de safety. De afsluit cyclus werkt ook goed en de machine de-initialiseert hiermee zodat de machine op een juiste manier uitgeschakeld kan worden.

Als er in de toekomst verder wordt gegaan met deze machine dan zou het communi­ceren met de robot onderzocht moeten worden. De robots worden in de ontwikkel­omgeving van het robotmerk geprogrammeerd. Er zou dus een verbinding met deze applicatie en de robot applicatie gerealiseerd moeten worden.

De safety werkt nu softwarematig om meldingen te ontvangen als er een onveilige situatie ontstaat, maar als de machine hardware gebouwd wordt, moet de veiligheid ook hardware matig de robots onderbreken.