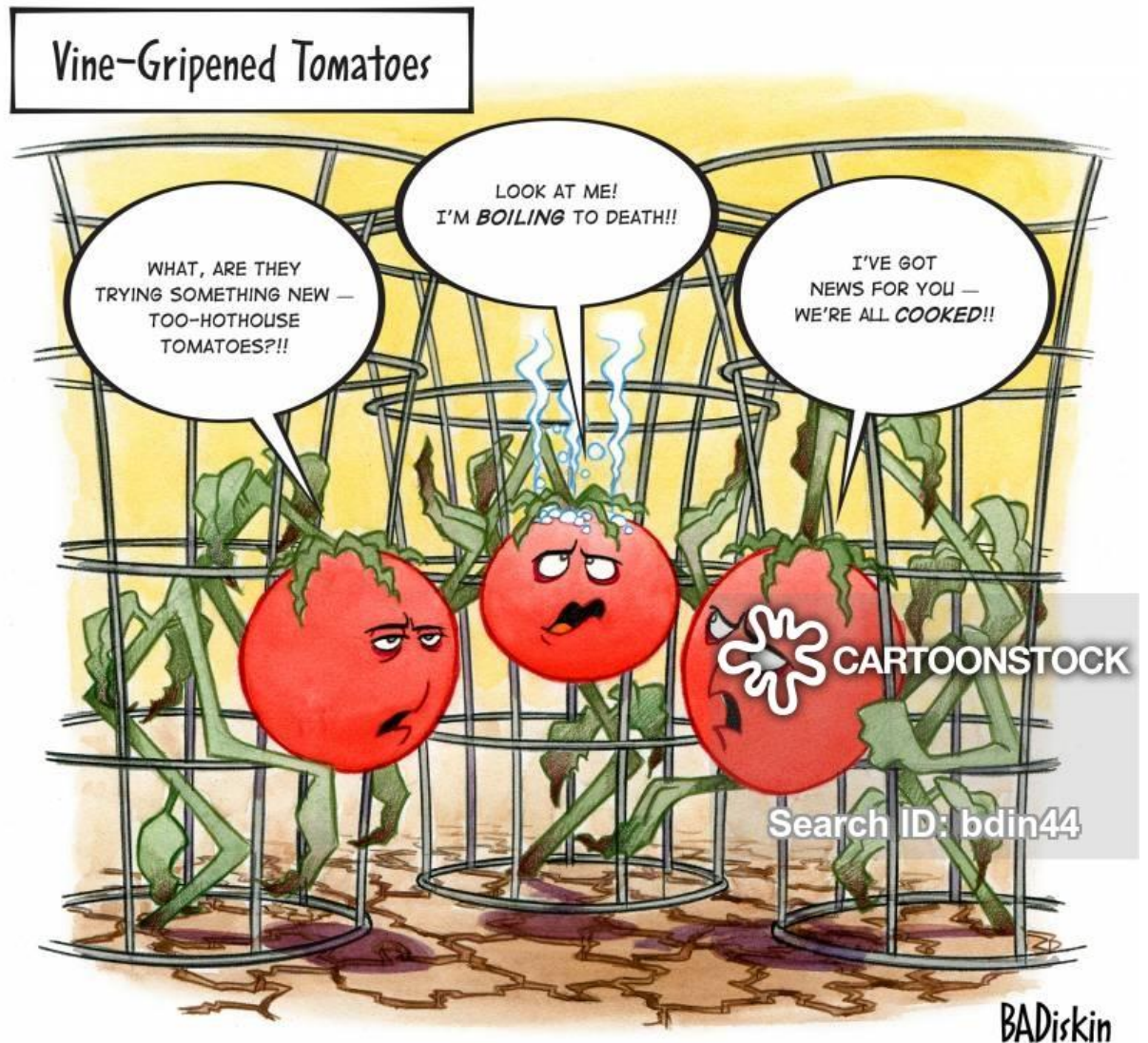


Tomato - The Automatic Greenhouse

Productdocument



**Student: Alwin Rodewijk
635653**

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

**Docent: Jos Onokiewicz
26-01-2020**

DOCUMENTHISTORIE

Datum	Versie	Wie	Veranderingen
21-11-2019	0.1	Alwin Rodewijk	Set up
16-12-2019	0.2	Alwin Rodewijk	Verschillende toevoegingen
24-01-2020	0.3	Alwin Rodewijk	Samenvatting t/m Eindresultaat aangevuld, gereed voor feedback
25-01-2020	1.0	Alwin Rodewijk	Feedback verwerkt
3-2-2020	1.1	Alwin Rodewijk	Feedback na <i>assessment</i>
7-2-2020	2.0	Alwin Rodewijk	Feedback verwerkt

SAMENVATTING

Tomato – The Automatic Greenhouse is een volledig automatische kas die voor gedefinieerde plant types kan zorgen. Een plant type wordt gedefinieerd a.d.h.v. een beschrijving en verzorgingseisen en is bijvoorbeeld een tomaten of aardbeien plant. De actieve plant is de plant waar momenteel voor gezorgd wordt in de kas.

De volgende aspecten worden gecontroleerd om de plant efficiënt te laten groeien:

- Omgevingstemperatuur
- Vochtniveau van de aarde
- Lichturen

De omgevingstemperatuur wordt beïnvloed op basis van de minimum- en maximumtemperatuur, zoals gedefinieerd voor de actieve plant.

Er wordt water gegeven wanneer het vochniveau van de aarde onder de helft van de gedefinieerde maximumwaarde komt.

Aan de hand van de lichturen worden de groeilichten aan of uit gezet zodat de plant de juiste hoeveelheid lichturen krijgt.

Er zijn twee modes gedefinieerd, de gebruikers- en servicemode. De gebruikersmode is ontworpen voor personeel dat klein onderhoud pleegt aan de installatie en informatie uitleest. De servicemode is ontworpen voor beheerders en servicemonteurs.

In gebruikersmode is het mogelijk om:

- De huidige status van de bovengenoemde onderdelen inzien
- De tijd gesimuleerd laten verlopen
- Inloggen met de service login

In servicemode is het mogelijk om:

- De huidige status van de bovengenoemde onderdelen inzien
- Een nieuw planttype definiëren en opslaan
- De actieve plant veranderen naar één van de opgeslagen plant types
- De hardwarecomponenten direct aansturen
- Uitloggen als service en teruggaan naar de gebruikersmode

Er wordt een logbestand bij gehouden. In dit logbestand is de volgende data opgenomen:

- De actieve plant
- Systeem *errors*
- Simulatiesysteem informatie

De bovenstaande onderdelen zijn getest aan de hand van acceptatietesten te vinden in bijlage 2. De testen zijn geslaagd.

VOORWOORD

Ik ben Alwin Rodewijk, een enthousiaste hardware engineer met een interesse in programmeren. Tijdens dit project heb ik me gefocust op de architectuur en de vormgeving daarvan.

Mijn voorkennis heb ik opgedaan tijdens hobby projecten waarin ik Arduino als mijn besturingsplatform heb gebruikt. Als hardware engineer kom ik niet vaak in aanraking met het schrijven van software. Daarentegen programmeer ik scripts om mijn werk te automatiseren of ondersteunen wanneer ik hiertoe de mogelijkheid zie. Zowel de Arduino projecten als de scripts zijn simpel, kort en missen structuur.

Dankzij mijn huidige werkgever heb ik voor het elektrisch tekenprogramma Eplan P8 de cursus 'Automation & scripting' gevolgd. Hier heb ik geleerd de tekenprogramma efficiënter te gebruiken met behulp van scripts geschreven in C#. Hier maak ik volop gebruik van om zelf efficiënter te werken en ik adviseer collega's wanneer ik een mogelijkheid tot verbetering in kwaliteit of tijd zie met behulp van scripts.

Er wordt bij mijn huidige werkgever ook veel gebruik gemaakt van Excel. Hierin heb ik een combinatie van formules en Visual Basic scripts complexe berekeningen gemaakt. Wanneer Excel niet voldoet door een te grote hoeveelheid data, heb ik Rstudio gebruikt voor het maken van berekeningen en grafieken. Een voorbeeld hiervan is een meting van de inductie pieken in een DC netwerk. Wanneer deze gedetecteerd werden moet een weerstand worden ingeschakeld om de pieken te onderdrukken. Door de hoge resolutie de lengte van de meting was de hoeveelheid data te groot om met Excel te verwerken. Ik heb op eigen initiatief Rstudio gekozen om de data te verwerken en een passend eindresultaat te maken.

Thuis maak ik als hobby elektrische ontwerpen, hier komt ook vrijwel altijd software aan te pas. Met Arduino als besturingsplatform heb ik verschillende projecten succesvol afgerond. Ik heb samen met mijn vriendin een *Spiderbot* gemaakt die op 6 poten kan lopen en dansen. Dit was een project met een grote *community* en de software bestond al. Omdat de originele communicatie niet werkte door defecte *bluetooth* modules heb ik deze vervangen door NRF24L01 modules. Dit had wel als gevolg dat de code die gebruikt werd voor de communicatie moest worden aangepast. Ondanks dit hebben we de robot werkend gekregen.

Dit project is ook een voorbeeld van een hobby project. Het product dat in dit document wordt beschreven heb ik zelf gemaakt. De hardware is getest en functioneert met een Arduino als aansturing. De software is alleen nog niet in een stadium waarin de kas bruikbaar is. Met dit project heb ik hier verandering in gebracht.

Tijdens dit project heb ik geleerd om het *design pattern* de *finite state machine* te gebruiken voor het schrijven van gestructureerde code. Ik zie hier de meerwaarde van in voor het programmeren van grotere softwareprojecten. Het gebruik van een *finite state machine* heeft mij geholpen om dit project overzichtelijk te programmeren. Nu kan ik grotere softwareprojecten overzichtelijk houden.

Alwin Rodewijk

INHOUDSOPGAVE

Documenthistorie	2
Samenvatting.....	3
Voorwoord	4
Inhoudsopgave	5
1. Inleiding.....	6
2. Definitiefase	7
2.1 Kennismaking en doel	7
2.2 Functionele eisen	8
2.3 Technische eisen.....	8
2.4 Schets userinterface	9
3. Ontwerpen.....	10
3.1 Architectuur	10
3.2 State chart	13
4. Realisatie en testen.....	15
4.1 Realisatie.....	15
4.2 Acceptatietesten	15
5. Eindresultaat en aanbevelingen	16
5.1 Eindresultaat	16
5.2 Aanbevelingen.....	16
Bijlage 1 – Inhoud *.pro file	17
Bijlage 2 – Acceptatietesten	18

1. INLEIDING

In dit document beschrijft het proces dat geresulteerd heeft in de besturing software van de Tomato – The Automatic Greenhouse.

In hoofdstuk 2 'Definitiefase' wordt kennis gemaakt met het doel van dit project, waarna de functionele en technische eisen worden opgesteld. Ook wordt het *user interface* beschreven.

In hoofdstuk 3 'Ontwerpen' wordt de software beschreven. De structuur en gebruikte bestanden geven een duidelijk overzicht van hoe de software functioneert. Aan de hand van een *state chart* wordt aangegeven hoe een *Finite State Machine* is geïmplementeerd om de subsystemen te laten functioneren.

In hoofdstuk 4 'Realisatie en testen' wordt het eindresultaat beschreven en wordt gecontroleerd of er aan de gestelde eisen is voldaan. Met behulp van de Doxygen documentatie wordt een overzicht gegeven van de software en het functioneren daarvan. Aan de hand van acceptatietesten wordt gecontroleerd of de software aan de gestelde eisen voldoet.

Tot slot wordt in hoofdstuk 5 'Eindresultaat en aanbevelingen' het resultaat besproken. Op basis van hoofdstuk 4 wordt geconcludeerd dat het eindresultaat aan de gestelde eisen heeft voldaan. Vervolgens worden aanbevelingen voor verdere ontwikkeling van dit product gegeven.

2. DEFINITIEFASE

Dit hoofdstuk gaat in op de gestelde eisen aan de te realiseren besturingssoftware van Tomaat – The Automatic Greenhouse. Hierbij wordt kennis gemaakt met het doel van dit product. De functionele en technische eisen worden opgesteld. Waarna het userinterface aan de hand van een schets wordt omschreven.

2.1 Kennismaking en doel

Tomaat – The Automatic Greenhouse is een volledig automatische kas waar verschillende types planten in kunnen groeien. Er wordt rekening gehouden met de lichturen, de vochtigheid van de aarde en de temperatuur. Per type plant kunnen waarden voor deze aspecten worden ingesteld. De waarden van de actieve plant worden gebruikt om de kas te besturen. Hierdoor wordt gewaarborgd dat de plant optimaal groeit. In figuur 1 en 2 wordt het bestaande systeem weergegeven met foto's.

De volgende variabelen worden opgeslagen voor een type plant:

- De naam
- De maximumtemperatuur in graden Celsius
- De minimumtemperatuur in graden Celsius
- Het vocht percentage van de grond in procenten
- De lichturen per dag



Figuur 1 – Hardwarecomponenten in de kas met groei licht aan.



Figuur 2 - Hardware componenten buiten de kas.

2.2 Functionele eisen

De functionele eisen voor het automatische systeem zijn als volgt:

1. Er wordt volledig automatisch voor een plant gezorgd.
2. De plant krijgt het aantal licht uren per dag wat gespecificeerd is voor dat type plant.
3. De plant krijgt voldoende water. Dit wordt gegarandeerd door de plant water te geven wanneer het vochniveau onder de helft van het maximum komt.
4. De temperatuur blijft tussen het minimum en maximum van de actieve plant met behulp van een verwarmingselement en een temperatuursensor.

De functionele eisen voor het acties die de gebruiker kan uitvoeren zijn als volgt:

1. In servicemode kan een ander soort plant worden geselecteerd.
2. In servicemode kan een nieuw soort plant worden aangemaakt en opgeslagen.
3. De tijd kan via de 'time' actie vooruit worden gezet om de automatische acties uit te voeren. Ieder half uur wordt gecontroleerd of de installatie een actie moet uitvoeren om de plant correct te verzorgen.

De functionele eisen omtrent het opslaan en uitlezen van bestand zijn als volgt:

1. Er wordt een log bestand gegenereerd om de gebeurtenissen op te slaan.
 - Log bestanden worden opgeslagen teruggevonden in de 'files/log' map met als naam 'hhmm_DDMMYYY_log.txt'.
 - De meldingen in dit bestand zijn als volgt opgebouwd: '<type melding>: <tijd> - <optioneel: relevante waarde> - <omschrijving>'.
2. De installatie onthoudt welke plant de actieve plant is. Wanneer het programma wordt afgesloten en daarna opnieuw gestart wordt de vorige actieve plant gebruikt als huidige actieve plant.
3. Er kunnen meerdere plant soorten worden opgeslagen in een passend bestands formaat.

2.3 Technische eisen

De technische eisen zijn als volgt:

1. De programmeertaal C met compiler 'MinGW 7.3.0 32-bit for C' wordt gebruikt.
2. De ontwikkelomgeving 'Qt Creator 4.11' wordt gebruikt.
3. Doxygen v1.8.17 in combinatie met Graphviz v3.38 wordt gebruikt voor het genereren van code documentatie.
4. Er is gebruik gemaakt van [GIT](#) met het interface SourceTree voor het versiebeheer.
5. De software wordt met het *terminal window* bestuurd.
6. Voor het opslaan van gedefinieerde plant types wordt een csv-bestand gebruikt.
7. Voor het log bestand wordt een txt-bestand gebruikt.

2.4 Schets userinterface

Het userinterface is vormgegeven in het *terminal window*. Er wordt een menu weergegeven waarin de huidige status van het systeem en de plant zijn weergegeven, vanaf nu te noemen 'systeem informatie'. Het *terminal window* heeft als gebrek dat weergegeven informatie alleen kan worden geüpdatet door het opnieuw te printen naar het *terminal window*. In figuur 3 staat een schets van de systeem informatie.

Door het gebruik van een toetsenbord kan de gebruiker acties uitvoeren. Deze acties zijn worden beschreven in het hoofdstuk '3.1 Architectuur'.

Bij elke actie van het automatische systeem wordt de huidige tijd en de bijbehorende waarde weergegeven in het *terminal window*.

TOMATO - THE AUTOMATIC GREENHOUSE			
Current values:		Active plant	
Water level	43	Water level max	60%
Temperature	13C	Max temperature	20C
Heater	on	Min temperature	10C
Light	on	Light hours	8
Time	12:30	Operation mode	

Enter 'help' to display the help menu.			

—			

Figuur 3 – Schets Terminal window

3. ONTWERPEN

Dit hoofdstuk gaat in op het ontwerp van de software van de Tomato – The Automatic Greenhouse. Hierbij wordt in paragraaf ‘3.1 Architectuur’ beschreven hoe de software is ontworpen. In paragraaf ‘3.2 State chart’ wordt het ontwerp van de *finite state machine* besproken.

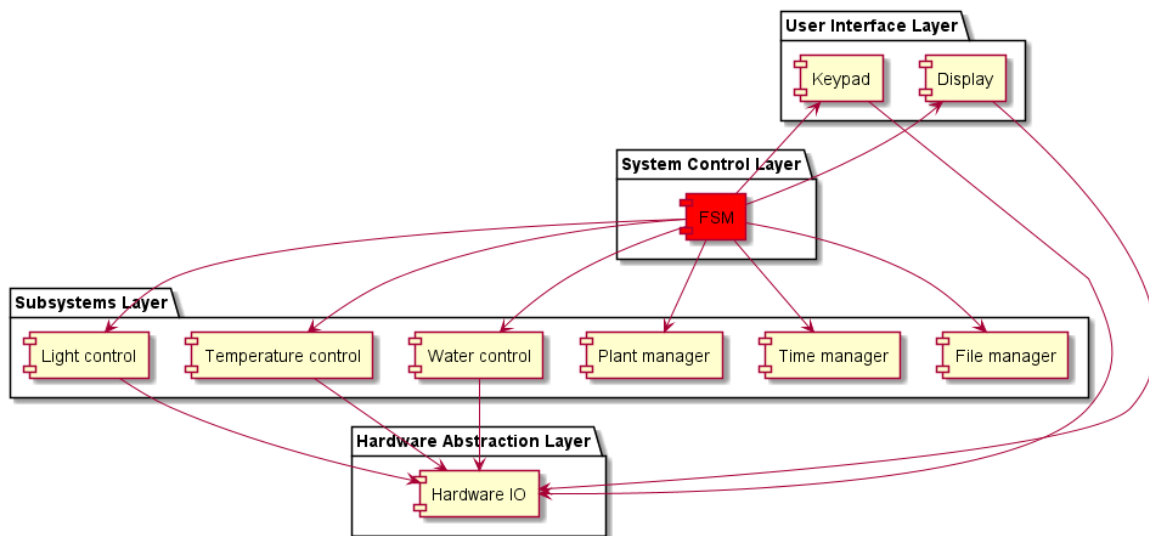
3.1 Architectuur

De volgende subsystemen bestaan binnen dit systeem.

- *User Interface Layer*:
 - *Display*: voor het weergeven van informatie op het terminal interface.
 - *Keyboard*: voor het ophalen van de gebruikersinput. Dit wordt gedaan in de vorm van tekst, een numerieke waarde of een gebruiker/service acties.
- *System Control Layer*:
 - *Finite State Machine*: voor bijhouden van de *state* waarin de installatie momenteel verkeerd en het afhandelen van *events*.
- *Subsystems Layer*, voor het automatisch systeem:
 - *Water control*: voor het monitoren van de vochtigheid van de grond, water geven aan de plant en de waterpomp aan of uit zetten.
 - *Light control*: voor het aan- en uitzetten van de groei lampen.
 - *Temperature control*: voor het monitoren van de temperatuur en het aan- of uitzetten van de verwarming.
- *Subsystems Layer*, voor bijhouden van de informatie voor het systeem:
 - *Plant manager*: voor het bijhouden van de huidige actieve plant.
 - *Time manager*: voor het bijhouden en simuleren van de tijd, die gebruikt wordt in deze installatie om de automatische acties uit te voeren.
 - *File manager*: voor het maken van het logbestand, het toevoegen van een nieuwe plant aan het ‘plant.csv’ bestand en het opslaan en uitlezen van de huidige actieve plant.
- *Hardware Abstraction Layer*:
 - *Hardware IO*: voor het aansturen van de *hardware* componenten.

In figuur 4 wordt de *layered architecture* weergegeven die van toepassing is op dit systeem. In bijlage 2 ‘Inhoud *.pro file’ is het overzicht van de bestanden die bij de subsystemen horen.

TAG layered architecture - 1 -



Figuur 4 – Overzicht *layered architecture*.

Voor de gebruikersinput zijn acties gedefinieerd. De acties die mogelijk zijn in gebruikers- en servicemode zijn te vinden in het help menu beschikbaar in deze modus.

Het overzicht van acties luidt als volgt:

- Algemene acties:
 - *help*: (in beide modus beschikbaar) om het help menu weer te geven. Afhankelijk van of de actie is uitgevoerd in het gebruikers of service menu, worden de beschikbare actie met omschrijving weergegeven.
 - *update*: (in beide modus beschikbaar) om het *terminal window* leeg te maken waarna de systeem informatie opnieuw wordt weergegeven.
 - *time*: (alleen in gebruikersmode) om verloop van tijd te simuleren. Na het invullen van deze actie wordt er gevraagd naar het aantal te simuleren minuten.
- Acties voor het wisselen tussen modes:
 - *service*: (alleen in gebruikersmode) om naar servicemode te gaan.
 - *user*: (alleen in servicemode) om de service mode uit te zetten en terug te keren naar gebruikersmode. Het *terminal window* wordt leeg gemaakt, waarna de systeem informatie wordt weergegeven.
- Acties die de automatisch systemen handmatig aansturen:
 - *water*: (alleen in servicemode) om handmatig het waterproces aan te zetten.
 - *light*: (allen in servicemode) om het licht respectievelijk aan- of uit te zetten.
 - *heater*: (alleen in servicemode) om de verwarming respectievelijk aan- of uit te zetten.
 - *pump*: (alleen in servicemode) om de pomp respectievelijk aan- of uit te zetten.

- Acties die betrekking hebben tot de plant:
 - *add*: (alleen in servicemode) om een planttype toe te voegen. De gebruiker wordt gevraagd om de eigenschappen van de plant één voor één in te vullen, waarna het nieuwe planttype wordt opgeslagen in het bestand 'plants/plants.csv'.
 - *change*: (alleen in servicemode) om de actieve plant te veranderen. Na deze actie te hebben ingevuld, wordt er gevraagd een keuze te maken uit de weergegeven planten. Deze keuze wordt vervolgens verwerkt en opgeslagen in het bestand 'plants/activePlants.txt'.

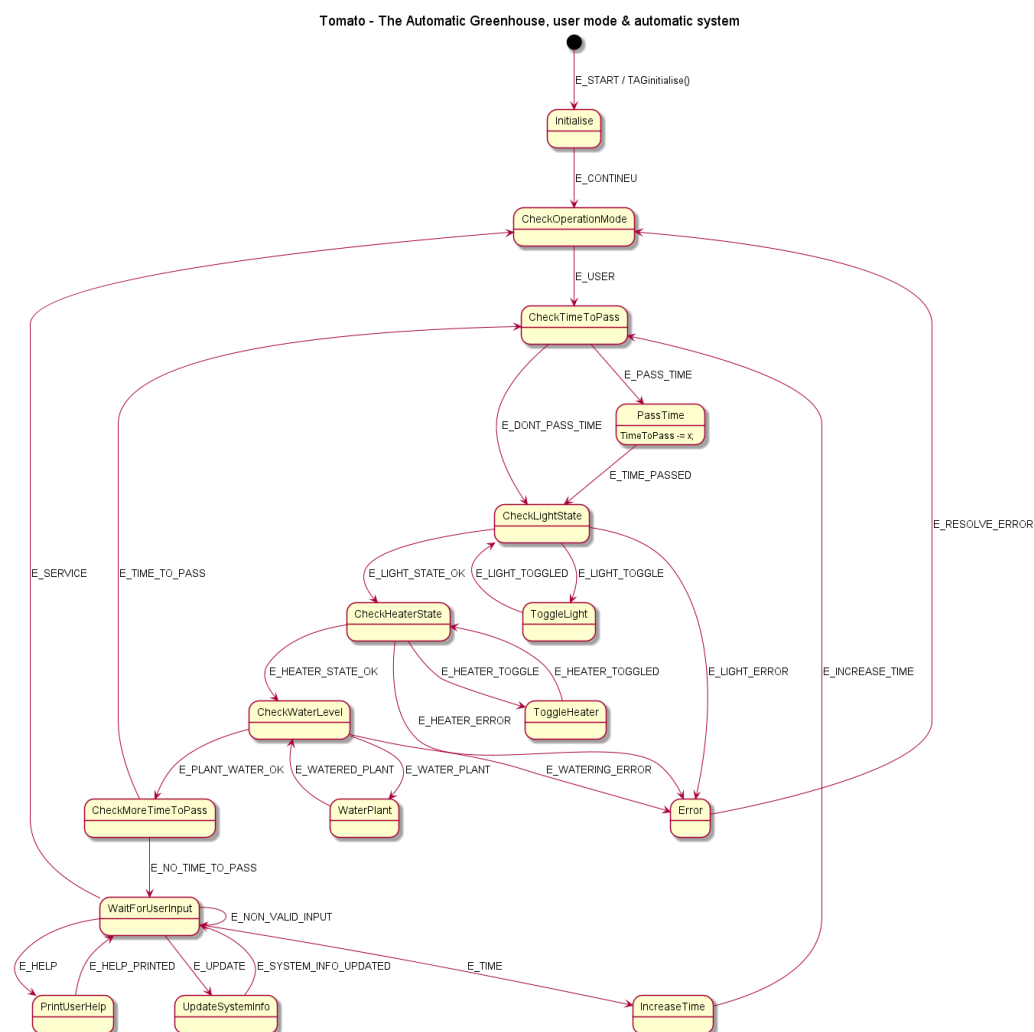
3.2 State chart

Er is een *finite state machine* gebruikt voor het realiseren van de besturing. De *states* en *events* staan beschreven in de Doxygen documentatie.

De onderstaande diagrammen zijn opgedeeld om deze overzichtelijk te houden. De volgende *states* zijn hetzelfde in beide diagrammen en verbinden de 2 diagrammen met elkaar:

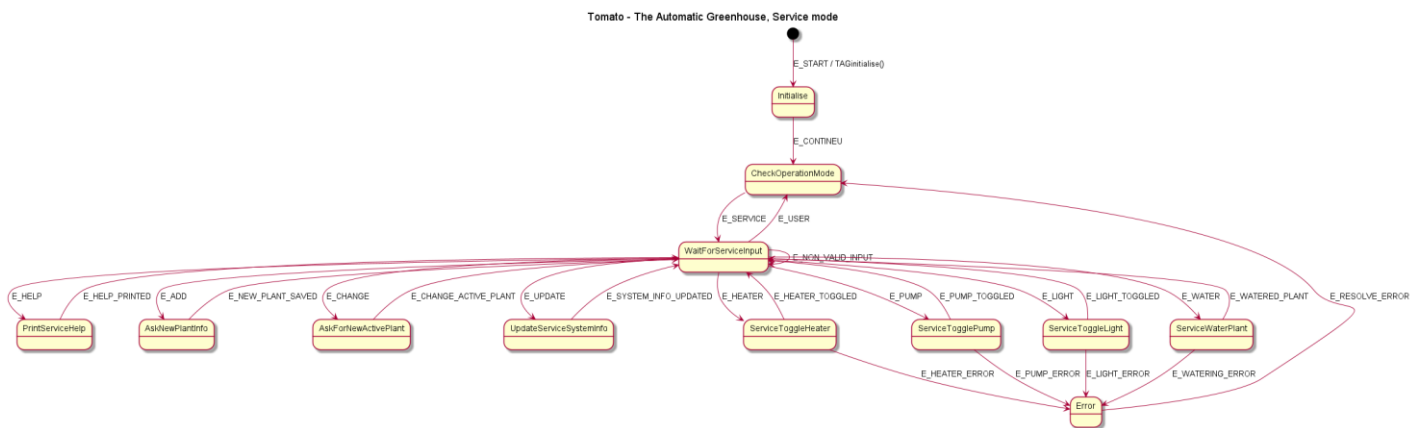
- Initialise
- CheckOperationMode
- Error

Het diagram in figuur 5 beschrijft het deel van de *finite state machine* dat het automatische systeem en de gebruikers input afhandelt.



Figuur 5 – State diagram voor het automatische systeem en de gebruikers input

Het diagram in figuur 6 beschrijft het deel van de *finite state machine* dat service input afhandelt.



Figuur 6 – State diagram gebruikt voor het afhandelen van service input

4. REALISATE EN TESTEN

Dit hoofdstuk bevat de omschrijving van de realisatie en acceptatietesten van dit project. Eerst wordt besproken wat er gerealiseerd is en hoe dit is vorm gegeven. Vervolgens wordt aan de hand van acceptatietesten gecontroleerd of het gewenste resultaat is behaald.

4.1 Realisatie

Er is gebruik gemaakt van QT Creator 4.10 als programmeer omgeving; van Plant_UML voor het maken van de UML diagrammen; en van DoxyGen voor het genereren van de documentatie van de code. In de Doxygen documentatie vindt u de handleiding die bij deze installatie hoort.

De code is modulair opgezet in verschillende header- en c-files terug te vinden in bijlage 1 'Inhoud *.pro file'

De folder structuur is als volgt:

- Tomato/
 - Software files.
 - files/
 - log/
 - Gegenerateerde log files.
 - Plants/
 - activePlants.txt
 - plants.csv
- Tomato_docs/
 - De Doxygen files nodig voor het generen van de software documentatie.

De volgende bekende bugs bestaan momenteel nog in dit project:

- Er wordt momenteel niet gecontroleerd op karakters die niet functioneren in een .csv best tijdens de invoer van de plant naam.
- Wanneer er een systeem error voorkomt kan de software hier niet zelf van herstellen.

4.2 Acceptatietesten

Met de acceptatietesten is gecontroleerd of de geproduceerde software functioneert zoals in dit document beschreven. De hieronder staan de testen in volgorde zoals ze zijn uitgevoerd in QT Creator 4.10.0. Deze testen zijn uitgevoerd door Alwin Rodewijk op 25-01-2020.

In bijlage 2 'Acceptatietesten' zijn de resultaten van de acceptatietesten te vinden. Met deze resultaten wordt bevestigen dat de software voldoet aan de eisen opgesteld in '2.2 Technische eisen'.

De volgende punten moeten ook getest worden om de software te verbeteren:

- Er moet worden getest met welke karakter het csv formaat om kan gaan.
- Er moet worden getest of de maximum waarde van de gebruikte variabelen kan worden overschreden.

5. EINDRESULTAAT EN AANBEVELINGEN

Dit hoofdstuk omschrijft het resultaat van dit project. Er wordt ingegaan op het volledige proces vanaf het opstellen van de eisen tot de realisatie. Daarna worden aanbevelingen gedaan voor toekomstige ontwikkelingen van dit project.

5.1 Eindresultaat

De software functioneert correct, wordt geconcludeerd op basis van de uitgevoerde acceptatie testen. Er kan volledig automatisch voor een plant verzorgd worden volgens gedefinieerde parameters. Er wordt water passend gegeven. Het licht wordt correct aan of uit gezet. En de verwarming wordt respectievelijk aan of uit gezet wanneer de temperatuur moet worden aangepast.

5.2 Aanbevelingen

Om de machine beter te laten functioneren zijn de volgende aanbevelingen gedaan:

1. Momenteel wordt er niet gereageerd op events die de subsystemen genereerd in de *service mode*. Omdat de handbediening in de *service mode* wordt gebruikt kan er niet worden gereageerd op deze *events*. Er moeten meldingen worden weergegeven wanneer een *event* wordt gegenereerd.
2. Er moeten foutmeldingen worden gegenereerd wanneer er vanuit de *hardware layer* onverwachte waarden worden geleverd.
3. Wanneer een actie een vervolg input nodig heeft (denk bijvoorbeeld aan '*time*' en '*add*') moet het mogelijk zijn om deze vraag/actie te annuleren. Hiervoor is een nieuwe actie vereist.
4. De systeem informatie moet worden verversd terwijl de tekst in het *terminal window* blijft staan.

Om de software beter te laten functioneren zijn de volgende aanbevelingen gedaan:

1. Er moet worden gecontroleerd of er verboden karakters zijn bij het invoeren van de plant naam. Het csv formaat zal hierbij mogelijk een beperkende factor zijn.
2. Het systeem moet, wanneer het opnieuw opstart, in dezelfde staat verkeren als wanneer hij voor het laatst aanstond. Hiervoor moeten, naast de actieve plant, de huidige waarden van de subsystemen worden opgeslagen.
3. Het controleren van de acties moet verbeterd worden, omdat het momenteel foutgevoelig is. Dit kan gedaan worden met behulp van een *hash table* of *dictionary*.

BIJLAGE 1 – INHOUD *.PRO FILE

```
TEMPLATE = app
CONFIG += console
CONFIG -= app_bundle
CONFIG -= qt

SOURCES += \
    display.c \
    file_manager.c \
    fsm.c \
    keyboard.c \
    light_control.c \
    main.c \
    plant_manager.c \
    temperature_control.c \
    time_manager.c \
    water_control.c

HEADERS += \
    display.h \
    events.h \
    file_manager.h \
    fsm.h \
    keyboard.h \
    light_control.h \
    plant_manager.h \
    time_manager.h \
    time_manager.h \
    temperature_control.h \
    water_control.h
```

BIJLAGE 2 – ACCEPTATIETESTEN

Test	Action/input	Expected result	Pass /Fail	Actual result if test has failed
1.	Start het programma in QT 4.10.0. Wanneer deze is gestart, type ' <i>help</i> ' en druk op enter.	Het help menu wordt weergegeven.	Pass	n.v.t.
2.	Typ ' <i>time</i> ' in druk op enter. Voer het getal '500' in en druk op enter.	Er worden automatische acties uitgevoerd om de huidige actieve plant te verzorgen. Vijfhonderd minuten worden gesimuleerd.	Pass	n.v.t.
3.	Typ ' <i>update</i> ' en druk op enter.	De systeem informatie staat weergegeven met up-to-date waardes onder ' <i>Current values:</i> '	Pass	n.v.t.
4.	Typ ' <i>service</i> ' en druk op enter. Herhaal dit vervolgens met ' <i>update</i> ' en ' <i>help</i> '.	In de systeem informatie onderaan ' <i>Active plant</i> ' staat nu ' <i>service mode</i> '. Het service helpmenu wordt weergegeven.	Pass	n.v.t.
6.	Typ ' <i>light</i> ' en druk op enter. Herhaal dit vervolgens met ' <i>pump</i> ', ' <i>heater</i> ' en ' <i>water</i> '.	Er wordt weergegeven dat het licht, de pomp en de verwarming van state zijn veranderd ten opzichte van de state die is weergegeven onder ' <i>Current values:</i> '. Er wordt weergegeven dat er water is gegeven.	Pass	n.v.t.
7.	Typ ' <i>add</i> ' en druk op enter. Vul vervolgens de gevraagde input in.	Er wordt weergegeven dat er een planttype is toegevoegd.	Pass	n.v.t.
8.	Typ ' <i>change</i> ' en druk op enter. Kies het hiervoor aangemaakte planttype uit de lijst.	Het aangemaakte planttype kan gekozen worden. Er wordt weergegeven dat de actieve plant is veranderd.	Pass	n.v.t.
9.	Typ ' <i>user</i> ' en druk op enter.	De systeem informatie wordt weergegeven. ' <i>User mode</i> ' wordt nu weergegeven. Het nieuw aangemaakte planttype met de ingevulde waardes staat nu onder ' <i>Active plant</i> '.	Pass	n.v.t.
10.	Sluit het terminal window en start vervolgens het programma.	De systeem informatie wordt weergegeven. De actieve plant is van hetzelfde type en heeft dezelfde waardes als aan het eind van de vorige test.	Pass	n.v.t.