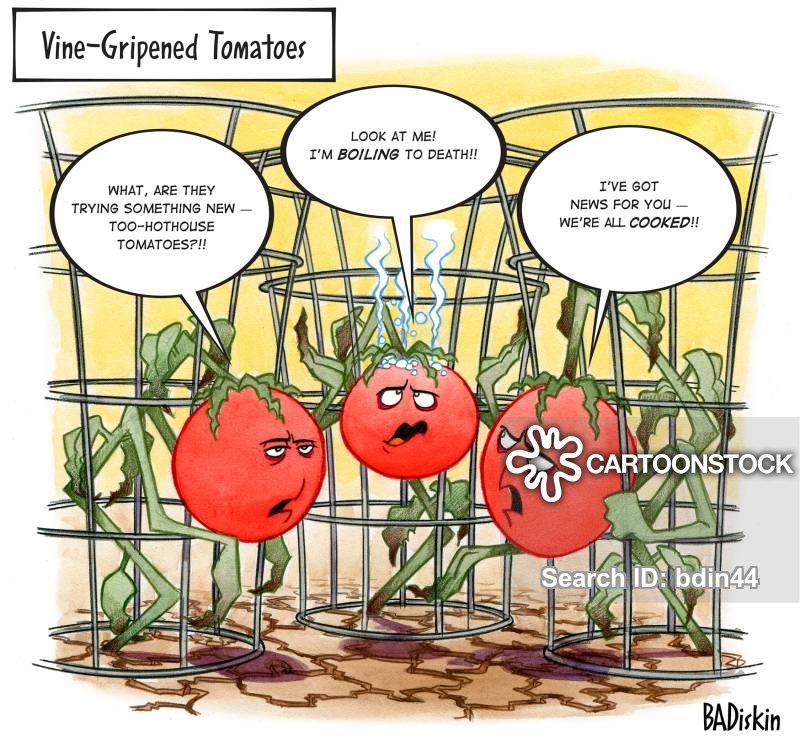
Tomato - The Automatic Greenhouse

productdocument

****

Student: Alwin Rodewijk

635653

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

Docent: Jos Onokiewicz

21-11-2019

Documenthistorie

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Versie** | **Wie** | **Veranderingen** |
| 21-11-2019 | 0.01 | Alwin Rodewijk | Startup |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Samenvatting

Het “Tomato” project is een volledig automatische kas die voor een type plant kan zorgen. Denk bij een type plant aan een tomaten plant, aardbeien plant, paprika plant, ed.

De temperatuur, vochtigheid van de grond en licht uren worden aan de hand van de parameters die bij een type plant horen beïnvloed. Hierdoor kan de plant optimaal groeien.

Met de service mode is de volgende handbediening mogelijk:

* Het huidige type plant veranderen.
* De parameters (temperatuur, water, licht) van de opgeslagen type planten aanpassen.
* Handbediening van de subsystemen (temperatuur, water, licht).

<Zakelijke samengevatte beschrijving van het project, eisen, wat is gerealiseerd, resultaat testen, belangrijkste aanbevelingen, vermijd de ik-stijl, moet op zichzelf leesbaar zijn zonder kennis van het project, niet verwijzen naar onderdelen in dit document, gebruik alinea's, opsommingen>

# Voorwoord

Na mijn MBO Mechatronica opleiding in de Elektro techniek beland te zijn ben ik thuis als hobby mij verder gaan ontwikkelen. Hieruit is een interesse in Hardware en Software ontstaan. 1 van mijn hobby projecten is een automatische kas bestuurd met een Arduino.

In dit project is al mijn voorkennis goed samengevat, de hardware is af en de software is in een staat waarin het hardware getest kan worden. Over het algemeen is dit het eindresultaat van mijn software gerelateerde projecten, werkende korte code.

Ook professioneel heb ik ervaring opgedaan met programmeren, maar wederom kleine projecten waarin code architectuur of documentatie niet aanbod is gekomen.

Dit project is dus een goede manier om ervaring op te doen met grotere gestructureerde code.

Tijdens dit project heb ik geleerd…

Alwin Rodewijk

<persoonlijke ervaringen, in ik-stijl, procesgericht, voorkennis, wat geleerd? wat doe je in de toekomst anders?>

<naam>

Inhoudsopgave

[Samenvatting 3](#_Toc530151323)

[Voorwoord 4](#_Toc530151324)

[1. Inleiding 6](#_Toc530151325)

[2. Definitiefase 7](#_Toc530151326)

[2.1 Kennismaking en doel van …….. 7](#_Toc530151327)

[2.2 Functionele eisen 7](#_Toc530151328)

[2.3 Technische eisen 7](#_Toc530151329)

[2.4 Schets user interface 7](#_Toc530151330)

[3. Ontwerpen 8](#_Toc530151331)

[3.1 Architectuur apparaat 8](#_Toc530151332)

[3.2 State chart apparaat 8](#_Toc530151333)

[4. Realisate en testen 9](#_Toc530151334)

[4.1 Realisatie 9](#_Toc530151335)

[4.2 Acceptatietesten 9](#_Toc530151336)

[5. Eindresultaat en aanbevelingen 10](#_Toc530151337)

[5.1 Eindresultaat 10](#_Toc530151338)

[5.2 Aanbevelingen 10](#_Toc530151339)

# Inleiding

Dit hoofdstuk gaat in op de opbouw van dit document.

<wat staat er in elk hoofdstuk? Heel korte omschrijving, iets meer info dan titel hoofdstuk, verwijs naar concrete hoofdstuknummers, gebruik lijstweergave>

# Definitiefase

Dit hoofdstuk gaat in op de gestelde eisen aan de te realiseren besturingssoftware van   
Tomato – The Automatic Greenhouse.

TODO:

//Technische beschrijving van het systeem

//Sub systemen uitwerken

//Menu uitwerken

//Time based acties

Opvragen van log bestand

Service menu:  
 File managment van plant type

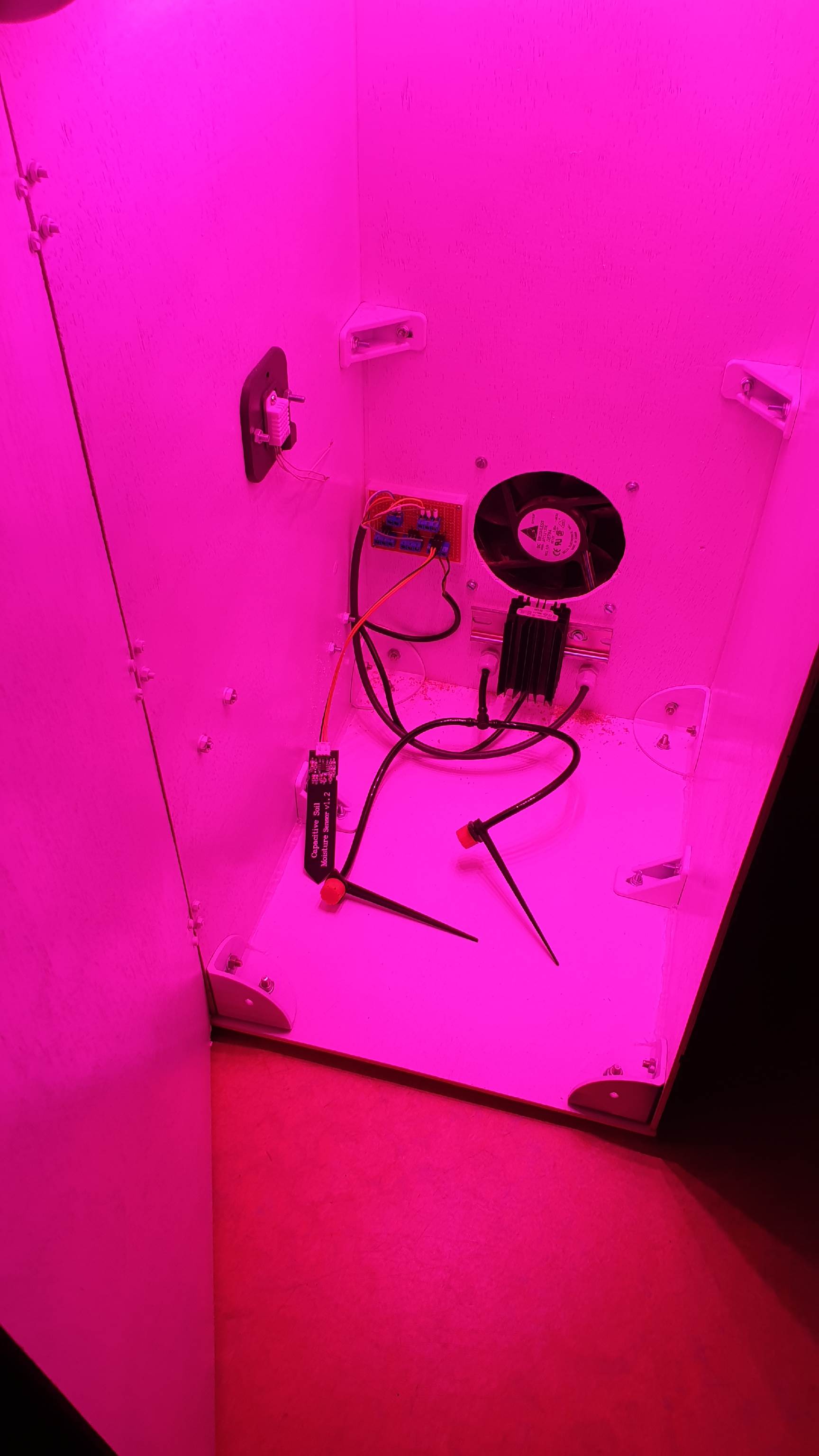
Hand bediening van subsystemen

## Kennismaking en doel van Tomato - The Automatic Greenhouse

Tomato – The Automatic Greenhouse is een volledig automatische kas waar verschillende soorten planten in kunnen groeien. De Tomato zorg voor je plant zonder dat je er iets voor hoeft te doen!

Er word rekening gehouden met de licht uren, de vochtigheid van de aarde waar de plant in staat en de temperatuur. Per soort plant kunnen deze eigenschappen veranderen, daarom is het mogelijk om een profiel voor een plant te maken waardoor je optimaal voor deze plant kan zorgen.

De Tomato maakt gebruik van een service login waardoor er veilig klein onderhoud kan worden gepleegd, maar ook door een service monteur kan worden bedient voor technisch onderhoud. BETER VERWOORDEN!!



## Functionele eisen

De functionele eisen zijn als volgt:

1. Er moet volledig automatisch voor een plant worden gezorgd.
2. Met een service login kan een ander soort plant worden geselecteerd.
3. Met een service login kan een nieuw soort plant worden aangemaakt en opgeslagen.
4. De plant het aantal licht uren per dag krijgen wat gespecifiseerd is voor die plant.
5. De plant zal voldoende water krijgen. Dit word gegarandeerd door een op een vast interval water te geven en daarnaast met een sensor te controleren of de aarde niet te droog word.
6. De temperatuur word tussen het minimum en maximum van de geselecteerde plant gehouden met behulp van een verwarmings element en een temperatuur sensor.
7. Er kan een log bestand worden gegenereerd om de gebeurtenissen van de huidige active sessie te bekijken. Oudere log bestanden worden opgeslagen en kunnen worden terug gevonden in de “Log\_files” map.
8. Aangezien de software via het terminal interface word bediend zal de tijd via een update actie vooruit worden gezet om de automatische acties te testen.

## Technische eisen

De technische eisen zijn als volgt:

1. De programmeertaal C moet worden gebruikt.
2. De ontwikkelomgeving QtCreator v5.13.2 moet worden toegepast.
3. De software word met het terminal window bestuurd.
4. Voor het opslaan van plant types word een .csv bestand gebruikt.
5. Er is gebruik gemaakt van GIT met het interface SourceTree voor het versie beheer.

## Schets user interface

|  |
| --- |
| C:\Users\Alwin Rodewijk\AppData\Local\Microsoft\Windows\INetCache\Content.Word\20191216_205718.jpg |
| Terminal HMI |

# Ontwerpen

Dit hoofdstuk gaat in op de …..

TODO:

Invullen

## Architectuur

…..

<overzicht subsystemen in lijstvorm>

* Water control
* Licht control
* Temperatuur control
* Service control
* Debug log

<UML + figuur titel + nr>

….

## State chart

…..

<UML + figuur titel + nr>

….

# Realisate en testen

Dit hoofdstuk bevat …

## Realisatie

<Korte weergave: Qt gebruik, instellingen, versies? Inhoud .pro file, Modulaire opbouw code, welke files? Directory structuur? Wat niet gemaakt? Bijv HAL, Bekende bugs? plantUML-gebruik, welke tools? is er codedocumentatie? Doxygen, Gebruikershandleiding beschikbaar?>

## Acceptatietesten

De volgende testen zijn …. <Volgorde? Voor wie? Hoe uitgevoerd?>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Action/input** | **Expected result** | **Pass Fail** | **Actual result if test has failed** |
| 1. |  |  |  |  |
| 2. |  |  |  |  |
| 3. |  |  |  |  |
| 4. |  |  |  |  |
| 5. |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

<Bovenstaande tabel mogelijk in landscape, onderzijde naar buiten toe>

<Samenvattende conclusie(s) over de testen, meer testen nog te doen? Waarom niet gedaan? >

# Eindresultaat en aanbevelingen

Dit hoofdstuk bevat ….

## Eindresultaat

<moet redelijk op zichzelf leesbaar zijn>

….

## Aanbevelingen

….

<wat nog te doen? verbeteringen? Bugs oplossen? Volgende versie? ….>