Plant module

productdocument



Student: Gert Buursink en Luuk Gout

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

Docent: Olav Roelofs

12-01-2023

**Documenthistorie**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Versie** | **Wie** | **Veranderingen** |
| 11-01-2023 | v0.1 | …. | Initiële versie. |
| 12-01-2023 | v0.2 |  | Aanvulling testen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Inhoud

[1. Inleiding 4](#_Toc124431013)

[2. Definitiefase 5](#_Toc124431014)

[2.1 Achtergrond project 5](#_Toc124431015)

[2.2 Functionele specificatie 5](#_Toc124431016)

[2.3 Technische specificatie 5](#_Toc124431017)

[3. Ontwerpfase 6](#_Toc124431018)

[3.1 Layered Architecture 6](#_Toc124431019)

[3.1.1 User Interface Layer 6](#_Toc124431020)

[3.1.2 System Control Layer 6](#_Toc124431021)

[3.1.3 Subsystems Layer 6](#_Toc124431022)

[3.1.4 Hardware Abstraction Layer 7](#_Toc124431023)

[3.2 State machine 7](#_Toc124431024)

[4. Realisatie en testen 8](#_Toc124431025)

[4.1 Ontwikkelomgeving 8](#_Toc124431026)

[4.2 Acceptatie test 8](#_Toc124431027)

[4.3 Code 10](#_Toc124431028)

[4.3.1 States 10](#_Toc124431029)

[4.3.2 Events 10](#_Toc124431030)

[4.3.3 Event functions 10](#_Toc124431031)

[4.3.4 HAL functions 11](#_Toc124431032)

[5. Eindresultaat en aanbevelingen 12](#_Toc124431033)

# Inleiding

In dit document wordt de realisatie van dit project beschreven. Voor het project moesten we een proces met verschillende inputs en outputs simuleren. Hiervoor hebben we een Finite State Machine gemaakt. Voor de machine die we simuleren hebben we gekozen voor een autonome kamerplant. De Finite State Machine wordt geprogrammeerd in ANSI C met een door de HAN geleverde framework voor de FSM.

# Definitiefase

## Achtergrond project

De plant module is een autonome manier om een kamerplantje in leven te houden. Dit wordt gedaan doormiddel van actuatoren zoals een ventilator, verwarmingselement en een watersprayer. De temperatuur, luchtvochtigheid en Co2 gehalte worden gemonitord waarna de juiste actie wordt uitgevoerd. Statusindicatoren geven aan wanneer het systeem actief, idle of in error status is.

## Functionele specificatie

Om goed te functioneren dient het systeem aan de volgende specificaties te voldoen:

1. Het systeem moet verschillende sensoroutputs kunnen detecteren
2. Het system moet de juiste actie uitvoeren wanneer een input veranderd is en de waardes binnen bepaalde grenzen vallen (Bewateren, verwarmen of beluchten)
3. De juiste statusindicatie moet worden weergegeven (Goed, bezig, error)

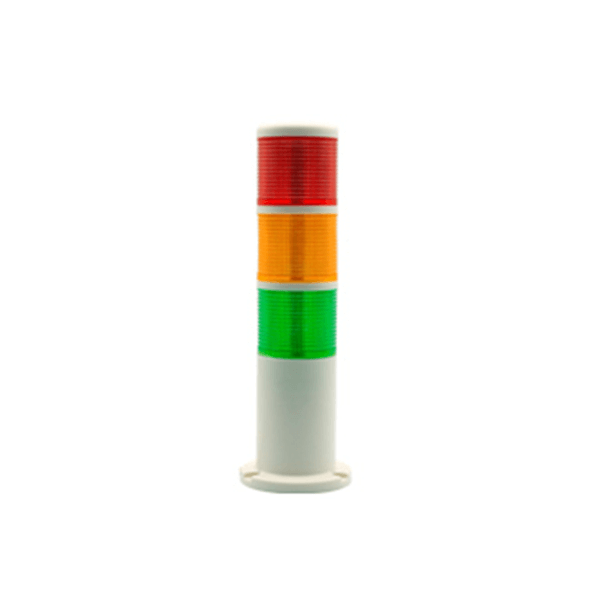
## Technische specificatie

1. Het system moet worden geprogrammeerd in C. met behulp van QtCreator
2. Er mag geen gebruik worden gemaakt van externe libraries.
3. Het programma moet gebruik maken van het aangeleverde framework.
4. De software moet worden gedocumenteerd doormiddel van Doxyen

De user interface bestaat uit drie signaallampen die de status van het systeem weergeven.

Groen wordt weergegeven wanneer er niks aan de hand is: het systeem staat aan en heeft geen errors ondervonden.

De gele lamp brand wanneer het systeem bezig is met verwarmen, bewateren of beluchten. De rode lamp brandt bij een error state.



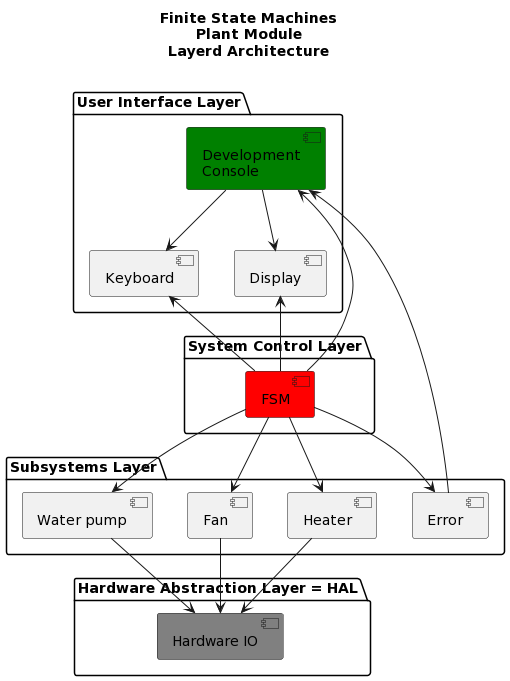
Figuur 1: Statusindicator (Goed (idle), bezig, error)

# Ontwerpfase

Dit hoofdstuk omschrijft het technisch ontwerp van de plant module.

## Layered Architecture

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



*Figuur 3 Layered architecture van de plant module*

### 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

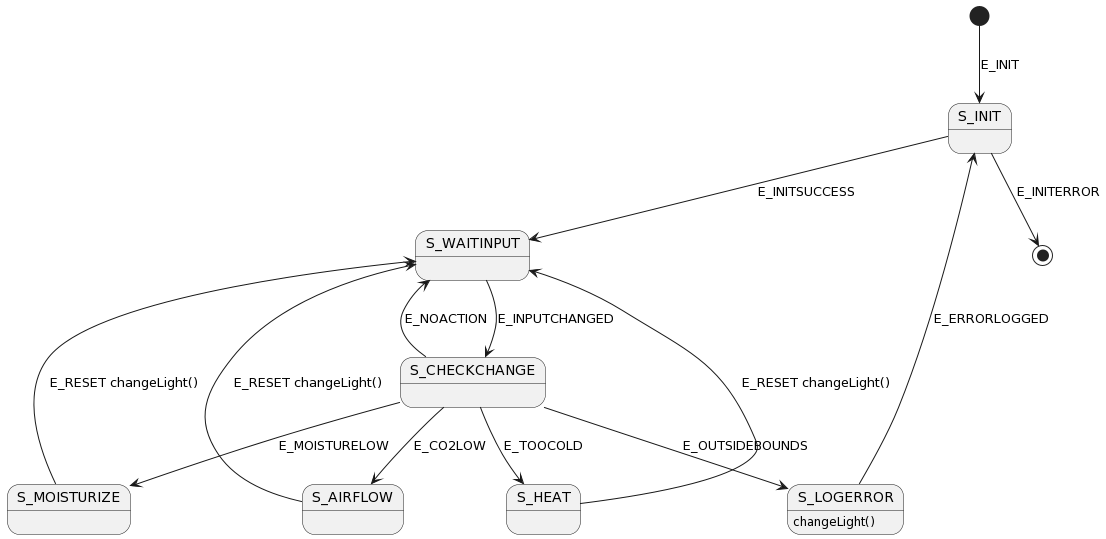
In de subsystems layer bevinden zich de functies die kunnen worden uitgevoerd door de FSM. Dit zijn de waterpomp, fan, heater en error subfuncties.

### Hardware Abstraction Layer

In de hardware abstraction layer worden de IO’s aangestuurd. In dit geval gaat het om de signaallampen en de actuatoren voor het verwarmen, bewateren en beluchten.

## State machine

De state machine is ontworpen in PlantUML. In dit UML diagram worden de verschillende states weergegeven van de plant module.



Figuur 2 UML diagram SFM

Na het initialiseren van het systeem is de S\_WAITINPUT state actief. Hierin wordt er gewacht op een sensorinput. Wanneer een input veranderd is wordt er geschakeld naar de S\_CHECKCHANGE state waarin er wordt welke input is veranderden of deze inputwaarde binnen de juiste thresholds valt. Wanneer hier een error voorkomt wordt de S\_INIT state opnieuw actief en gaat een errorlampje aan. Op basis van de verandering wordt S\_MOISTURIZE, S\_AIRFLOW of S\_HEAT actief. Ten slotte gaan we na een reset terug naar S\_WAITINPUT. 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.

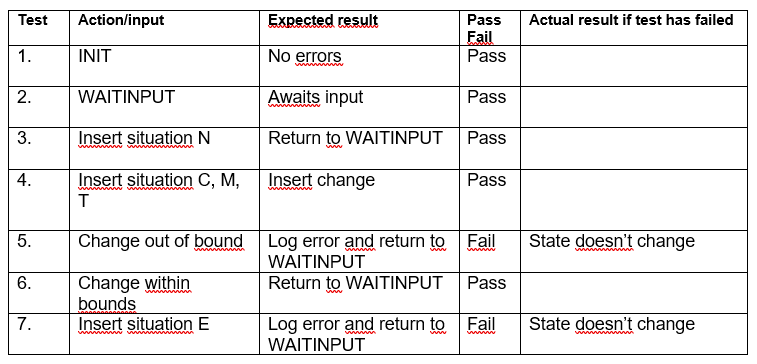
# Realisatie en testen

## Ontwikkelomgeving

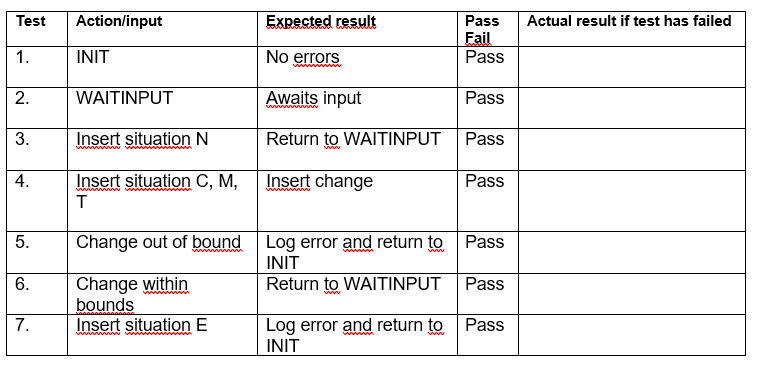
De code is geschreven in C in QtCreator. De UML diagrammen zijn gemaakt in de browserversie van PlantUML.

## Acceptatie test

11-1-2023: Initiele test



12-1-2023: Test 2



## Code

De code is overzichtelijk en bevat duidelijke comments

### States

De FML bevat de volgende states:

S\_NO (initialisatie)

S\_START

S\_INIT

S\_WAITINPUT

S\_CHECKCHANGE

S\_LOGERROR

S\_AIRFLOW

S\_MOISTURIZE

S\_HEAT

### Events

E\_NO

E\_INIT

E\_INITERROR

E\_INPUTCHANGED  
E\_NOACTION

E\_NORMALCHANGE  
E\_OUTSIDEBOUNDS

E\_ERRORLOGGED  
E\_CO2LOW  
E\_MOISTURELOW

E\_TOOCOLD

E\_RESET

### Event functions

event\_t EF\_WAITINPUT(void);

event\_t EF\_CO2LOW(void);

event\_t EF\_MOISTURELOW(void);

event\_t EF\_TOOCOLD(void);

### HAL functions

void ChangeLight(int);

void LogError(void);

void OpenWindow(void);

void Moisturize(void);

void HeatPlant(void);

# Eindresultaat en aanbevelingen

De plantmodule functioneert naar behoren. De documentatie zou in de toekomst nog kunnen worden gemaakt met Doxygen. De S\_WAITINPUT en S\_CHECKCHANGE states zijn redundant en zouden kunnen worden samengevoegd tot één state.