|  |  |  |  |
| --- | --- | --- | --- |
| ***Version*** | ***Kommentar*** | ***Dato*** | ***Personer*** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Systembeskrivelse

MoSCoW

Funktionelle krav

Ikke funktionelle krav

Aktørkontekst diagram



*Figur Aktørkontekst diagram*

Aktørbeskrivelse

|  |  |
| --- | --- |
| Aktørnavn: | Bruger |
| Anden reference | - |
| Type | Primær |
| Beskrivelse | Enhver som bruger systemet. Bruger laver både opsætning, kigger på data og påfylder vand |

|  |  |
| --- | --- |
| Aktørnavn: | Plante |
| Anden reference | - |
| Type | Sekundær |
| Beskrivelse | Planten inde I væksthuset. Plante kan repræsentere mange forskellige typer planter som systemet påvirker, ud fra givne parametre. |

|  |  |
| --- | --- |
| Aktørnavn: | Sensorer |
| Anden reference | - |
| Type | Primær |
| Beskrivelse | Sensorer aktøren indeholder sensorer af typen: Temperaturføler. Fugtighedsmåler.  Temperaturføleren måler temperaturen i væksthuset, fugtighedsmåleren måler fugtigheden i plantes jord. |

|  |  |
| --- | --- |
| Aktørnavn: | Aktuatorer |
| Anden reference | - |
| Type | Primær |
| Beskrivelse | Aktuatorer aktøren indeholder aktuatorer af typen: Motor. Varmelegeme. Blæser.  Motoren sørger for at lukke op for et spjæld så planten for tilført vand. Varmelegemet regulerer temperaturen i væksthuset. Blæseren regulerer bade temperaturen og fugtigheden I væksthuset |

Usecase diagram



*Figur 2 Usecase diagram*

Fully dressed usecases

|  |  |
| --- | --- |
| Navn | Usecase 1: første gangs opsætning |
| Mål | Systemet kører med valgte parametre |
| Aktør | Bruger |
| Start | Bruger starter systemet for første gang |
| Forudsætning | Systemet er tændt  Systemet har ikke noget program valgt. (Først gangs opsætning) |
| Resultat | Systemet har gemt de valgte parametre, og er gået I gang med at opretholde disse parametre. |
| Hoved scenarie | * Bruger indsætter plante i væksthus * Bruger isætter sensorer * Systemet beder bruger om at vælge program * Bruger vælger program   Ext. 1: Bruger vælger prædefineret program  Ext. 2: Bruger indsætter selv parametre manuelt |
| Udvidelser | Ext. 1: Bruger vælger prædefineret program  Gå til usecase 2  Ext. 2: Bruger indsætter selv parametre manuelt  Gå til usecase 3 |

*Tabel Usecase 1*

|  |  |
| --- | --- |
| Navn | Usecase 2: Vælg prædefineret program |
| Mål | Systemet kører med parametrene fra det prædefinerede program som bruger valgte |
| Aktør | Bruger |
| Start | Bruger vælger ændring af program |
| Forudsætning | Systemet er tændt og kører et program |
| Resultat | Systemet har gemt de valgte parametre, og er gået I gang med at opretholde disse parametre. |
| Hoved scenarie | 1. Bruger vælger prædefinerede program  2. Bruger godkender valg af program  3. Systemet kører valgt program |

*Tabel Usecase 2*

|  |  |
| --- | --- |
| Navn | Usecase 3: Vælg brugerdefineret program |
| Mål | Systemet kører med parametrene som brugeren manuelt har valgt. |
| Aktør | Bruger |
| Start | Bruger vælger ændring af program |
| Forudsætning | Systemet er tændt og kører et program |
| Resultat | Systemet har gemt de valgte parametre, og er gået I gang med at opretholde disse parametre. |
| Hoved scenarie | 1. Bruger vælger brugerdefineret program  2. Bruger vælger parameter  3. Bruger ændrer parameter  4. Gentag fra 2-4 indtil bruger er tilfreds  5. Bruger trykker godkend  6. Systemet gemmer program  7. Systemet kører program |

*Tabel Usecase 3*

|  |  |
| --- | --- |
| Navn | Usecase 4: Vis data |
| Mål | Systemet viser bruger data om systemet |
| Aktør | Bruger / Sensorer |
| Start | Systemet opdaterer sin data |
| Forudsætning | Systemet er tændt og kører et program  Ingen andre usecases er I gang |
| Resultat | Systemet viser bruger data |
| Hoved scenarie | 1. Systemet opdaterer sine data  2. Systemet viser disse data på brugergrænseflade |

*Tabel Usecase 4*

|  |  |
| --- | --- |
| Navn | Usecase 5: Kører valgt program |
| Mål | Systemet overholder valgte parametre |
| Aktør | Plante / Sensorer / Aktuatorer |
| Start | Systemet har ikke opdateret data I X tid |
| Forudsætning | Systemet er tændt og kører et program |
| Resultat | Systemet tilpasser miljøet I væksthuset til valgte parametre |
| Hoved scenarie | 1. Systemet indhenter data om miljø med sensor  2. Systemet sammenligner med valgte parametrene  3. Systemet tilpasser miljø med aktuatorer |

*Tabel Usecase 5*

Accepttestspecifikation

Usecase 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case under test | | UC1: First time setup | | |
| Scenarie | | Hovedscenarie | | |
| Prækondition | | Systemet er tændt, intet program valgt (Første gang systemet er tændt) | | |
| Step | Handling | Forventet observation/resultat | Faktisk observation/resultat | Vurdering (OK/FAIL) |
| 1. | Indsæt plante i væksthuset | Planten er inden i væksthuset. |  |  |
| 2. | Tilslut sensorer (Fugtighedssensoren) | Sensorerne er tilsluttet. |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case under test | | UC1: First time setup | | |
| Scenarie | | Extension: 1 | | |
| Prækondition | | Systemet er tændt, intet program valgt (Første gang systemet er tændt) | | |
| Step | Handling | Forventet observation/resultat | Faktisk observation/resultat | Vurdering (OK/FAIL) |
| 1. | Indsæt plante i væksthuset | Systemet går til den prædefineret tilstand |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case under test | | UC1: First time setup | | |
| Scenarie | | Extension: 2 | | |
| Prækondition | | Systemet er tændt, intet program valgt (Første gang systemet er tændt) | | |
| Step | Handling | Forventet observation/resultat | Faktisk observation/resultat | Vurdering (OK/FAIL) |
| 1. | Vælg brugerdefineret program | Systemet går til den brugerdefinerede tilstand |  |  |

Usecase 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case under test | | UC2: Vælg prædefineret program | | |
| Scenarie | | Hovedscenarie | | |
| Prækondition | | Systemet er tændt, og UC1 er overstået | | |
| Step | Handling | Forventet observation/resultat | Faktisk observation/resultat | Vurdering (OK/FAIL) |
| 1. | Tryk på prædefinerede programmer | Planten er inden i væksthuset. |  |  |
| 2. | Tryk på et program | Systemet fortæller bruger at programmet er valgt |  |  |
| 3. | Systemet starter programmet | Programmet starter |  |  |

Usecase 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case under test | | UC3: Vælg brugerdefineret program | | |
| Scenarie | | Hovedscenarie | | |
| Prækondition | | Systemet er tændt, og UC1 er overstået | | |
| Step | Handling | Forventet observation/resultat | Faktisk observation/resultat | Vurdering (OK/FAIL) |
| 1. | Tryk på brugerdefineret program | Systemet viser en liste af parametre der kan ændres |  |  |
| 2. | Tryk på parameter | Systemet angiver at parameteren er valgt |  |  |
| 3. | Ændre parameter | Sysemet ændrer sig som forventet (+ / -) |  |  |
| 4. | Tryk godkend | Systemet angiver at programmet er gemt |  |  |

Usecase 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case under test | | UC4: Vis data | | |
| Scenarie | | Hovedscenarie | | |
| Prækondition | | Systemet er tændt og kører et program | | |
| Step | Handling | Forventet observation/resultat | Faktisk observation/resultat | Vurdering (OK/FAIL) |
| 1. | Kig på brugergrænsefladen | Systemet viser variablerne for miljøet inden i kassen |  |  |

Usecase 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case under test | | UC5: Kører valgt program | | |
| Scenarie | | Hovedscenarie | | |
| Prækondition | | Systemet er tændt og kører et program | | |
| Step | Handling | Forventet observation/resultat | Faktisk observation/resultat | Vurdering (OK/FAIL) |
| 1. | Gennemfør UC3 med kontrol parametre | Systemet kører valgt program med valgte parametre |  |  |
| 2. | Kontroller at ændrede parametre bevæger sig mod valgte værdier over en periode på X tid | Systemet ændrer miljøet inden i væksthuset så de bevæger sig mod de valgte værdier. |  |  |

Systemarkitektur

Indledning

Dette afsnit beskriver systemarkitekturen for projektet væksthuset, som er formuleret i projektbeskrivelsen og specificeret i kravspecifikationen.

Afsnittet indeholder beskrivelser af systemkomponenter,system arkitektur, HW-komponenter, SW-komponenter, i den givne rækkefølge.

Formål

Formålet med dokumentet er:

* At nedbryde systemet i overordnede HW- og SW-komponenter baseret på kravene specificeret i kravspecifikationen.
* At fastlægge grænsefladen mellem systemets overordnede komponenter.
* At identificere arbejdsopgaver for projektets design- og implementeringsfase.

Systemkomponenter

Der blev besluttet ud fra kravspecifikationfasen, hvilke komponenter der skal bruge og hvor de skal placeres.

Temperaturføler

Fugtighedssensor

LED

Blæser

Vandpumpe

Psoc4

Devkit8000

Varmelegeme

Hardware

BDD

Overordnede BDD



*Blokbeskrivelse*

|  |  |
| --- | --- |
| Bloknavn | Blokbeskrivelse |
| Strømforsyning | Forsyningsenhed til væksthusets hoved- og delsystemer;  forsyner væksthuset med spænding |
| Embedded Linux Inlejring | Enheden indeholder den grafiske brugergrænseflade, og sørger også for kommunikationen(UART) til kontrolenhedsblokken. |
| Sensor | Sensorblokken indeholde følgende sensorer; temperaturføler, fugtighedsmåler. |
| Aktuator | Aktuatorblokken indeholde følgende aktuatorer; blæser og motor blæseren regulere temperaturen og fugtigheden i væksthuset, motoren sørger for at lukke op for et spjæld så der kan tilføres vand til planten |
| User interface | HID samt GUI til væksthuset, her vælges parametre, og viser grafer og tekst til brugeren |

BDD Kontrolenhed



*Blokbeskrivelse*

|  |  |
| --- | --- |
| Bloknavn | Blokbeskrivelse |
| LED | LED blokken sørger for at der er lys i væksthuset, der varieres imellem blå og rød lys alt efter hvordan planten skal påvirkes. |
| Vandingssystem | Vandingssystemet sørger ved hjælp af en pumpe for at tilføre vand til planten. |
| Blæser | Blæseren sørger for at regulere temperaturen og fugtigheden i væksthuset. |
| Fugtighedsmåler | Fugtighedsmåleren måler fugtigheden i jorden i potteplanten. |
| Temperaturføler | Temperaturføleren måler lufttemperaturen inde i væksthuset. |
| Varmelegeme | Varmelegemet søger for at regulere temperaturen i væksthuset. |

IBD

Overordnede IBD



|  |  |
| --- | --- |
| Signaltype | Signalbeskrivelse |
| - | - |
| - | - |
| - | - |
| - | - |
| - | - |
| - | - |

IBD Kontrolenhed

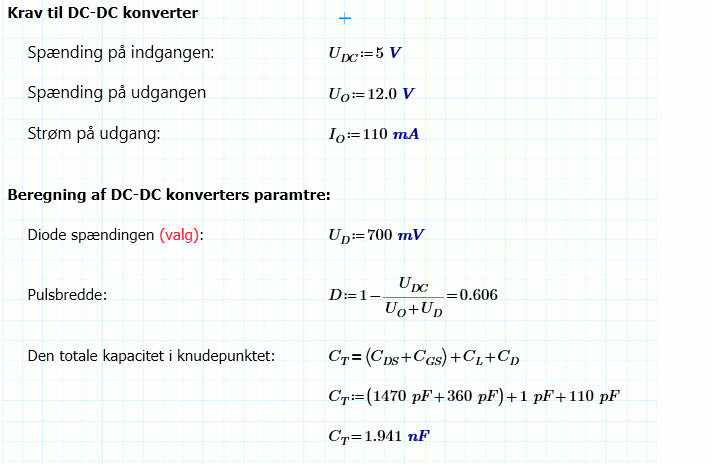


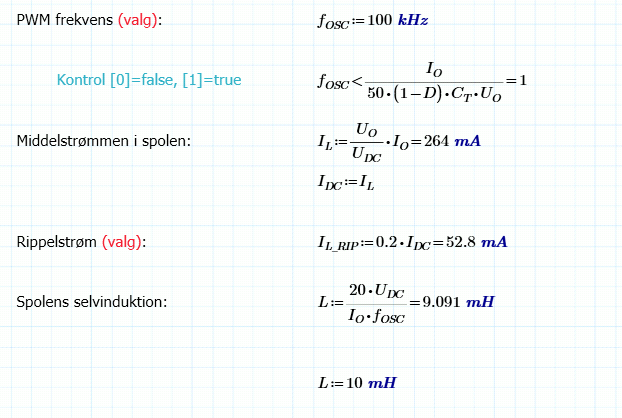
*Signalbeskrivelse*

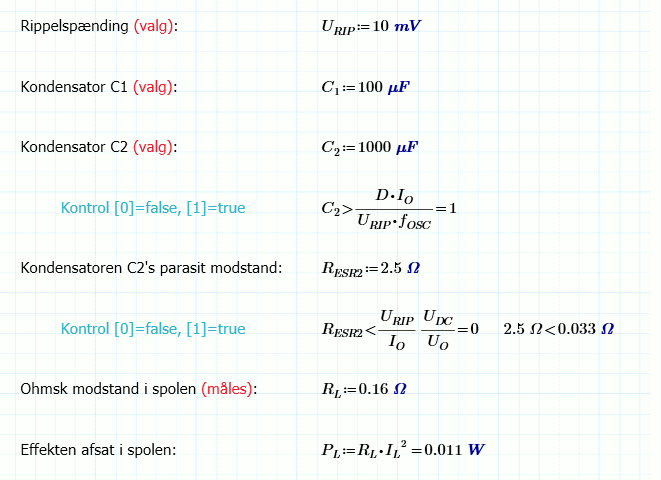
|  |  |
| --- | --- |
| Signaltype | Signalbeskrivelse |
| 5 VDC | Spændingsforsyning på 4,9-5,1 V |
| PWM | PWM 100KHz +/- 50Hz fra PSoC |
| SCL | I2C clocklinje (Signal type her) |
| SDA | I2C datalinje (Signal type her) |
| ADC Signal | DC spænding mellem xx-xx V som går til en ADC |
| MotKontrol | Signal der åbner et spjæld via en motor ( X VDC ved X Ampere) |

Blæser hardware

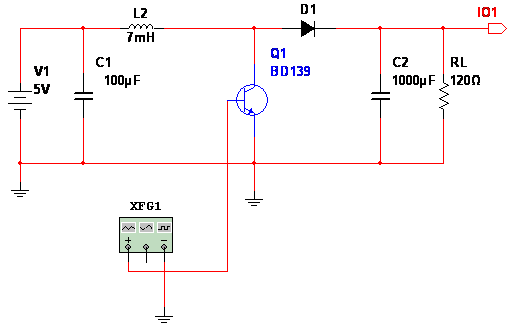
Beregninger

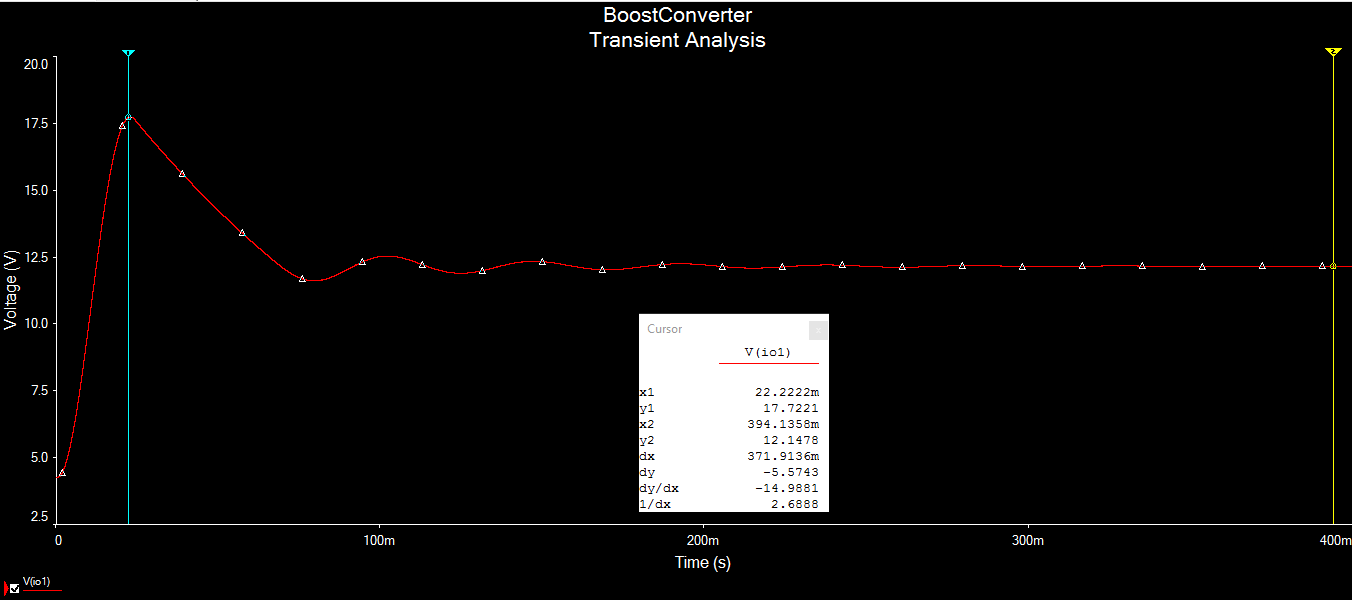






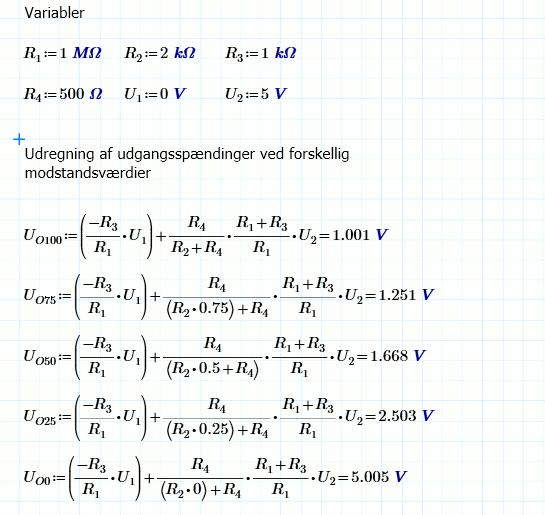
Multisim simulering

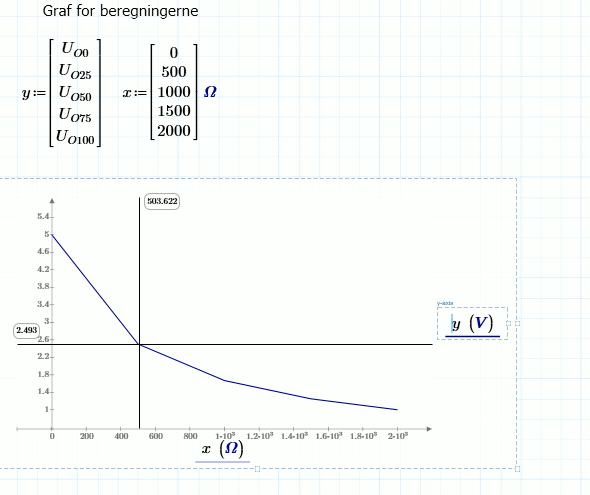




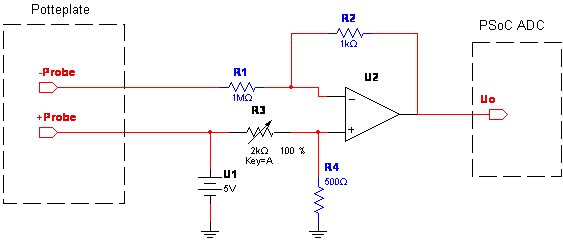
Fugtighedsmåler

Beregninger





Multisim simulering



Strøm forsyning

Systemet består af flere enheder der skal have strøm for at fungere. Derfor skal der laves en strømforsy-ning. Alle enheder skal modtage et DC signal som input. Af den grund er der valgt at bruge en spændingsregulator, der kan regulere spændingen ned, desuden er der brugt diode bro system til at skifte AC til DC. Strømforsyningen skal regulere DC signalet ned til de ønskede spændinger.

Herunder er vist de inputspændinger som hver enhed har brug for.

|  |  |  |
| --- | --- | --- |
| **Enheds navn** | **Strøm (I)** | **Spænding (U)** |
| Psoc | 100-150mA | 5V |
| Blæser | 110 mA | 5 v +/- 0.3V |
| Pumpe | 3A | 12V |
| Lys | 1.2A | 12V |
| Varmelegeme |  | 5V |
| Motor |  | 12V |
| Temperatur målere |  | 3,3 - 5v (fra Psoc) |
| Fugtigheds måler | 100mA | 3,3 – 5V |

Det vil sige, der er brug for to kredsløb 1 med 12V DC og 1 kredsløb med 5V DC.

Kredsløbet i Figur nr er en slags strømforsyning som skal have 28V DC som inputspænding og kan levere forskellige outputspændinger ved at justere på R2’s værdi.



**Figur nr : Kredsløbet kan levere spænding fra 1.2V-25V**

Grunden til dette kredsløb er valgt, er fordi det er nemt at implementere, da det kun er R2’s værdi, som skal ændres for at få den ønskede spænding på outputtet.

Denne vil blive brugt til 12VDC kredsløbet

**Bregning for R2**

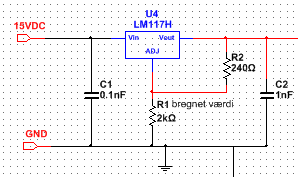
I Data sheet for LM317/LM117 er der opstillet et formel for hvordan output kan beregnes.

Formlen er vist herunder:

  
IADJ har en konstant værdi på 50uA.

Da det er nu bestemt at hvilke spændinger der er brug for kan der hurtig beregnes værdien for R2. Dvs. output spænding på strømforsyning svarer til input spænding til enhederne.

*Ligningen løses for x vha. CAS-værktøjet WordMat.*

Dvs. R2 skal være 2044Ω for at få 12V på output.  


**Spændingsregulator (15V DC – 5V DC)**

Da de fleste komponenter arbejder med kun 5V DC, skal de 15V DC reguleres ned til 5V. Til dette formål anvendes en spændingsregulator af typen MC7805. Se Figur.

**

Fordelen ved sådan en spændingsregulator er at de kan tåle kortslutninger der sker på udgang og at det vil automatisk afbryde kontakten til udgang. Hvis der sker en store overbelastning, som afgiver effekt i form af varm, så vil den vente til den bliver kølet ned til en passende temperatur og slår kontakten til igen.

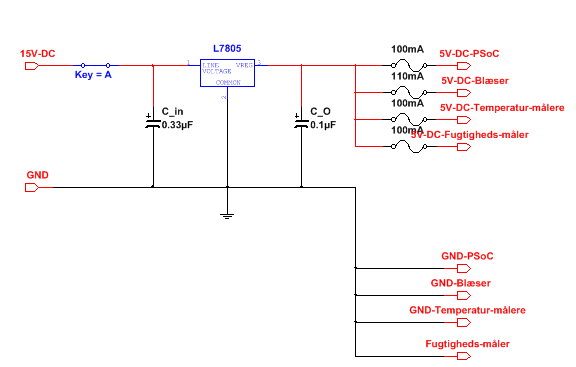
Ifølge databladet for spændingsregulatoren, må der maksimalt afsættes en effekt på 2W uden kølerplade. Derfor er der fortaget en beregning vha. Ohms lov for at finde frem til den totale effekt der afsættes.

Den total strømforbrug er ud fra enheds skemaet 4,2A. V\_in er 15V DC der kommer fra forsyning og de 5V DC er spændingen på udgang af spændingsregulatoren.

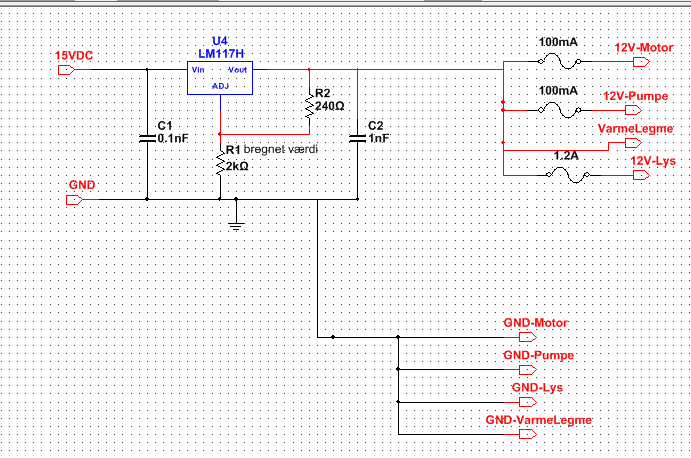
Formel for beregningen af total effekt

W

Den total effekt er beregnet til at være 42 W, derfor skal der anvendes en køleplad til at nedkøling af spændingsregulatoren.

Den færdiglavet diagram for de to kreds løb ser således ud   


5VDC kreds løb

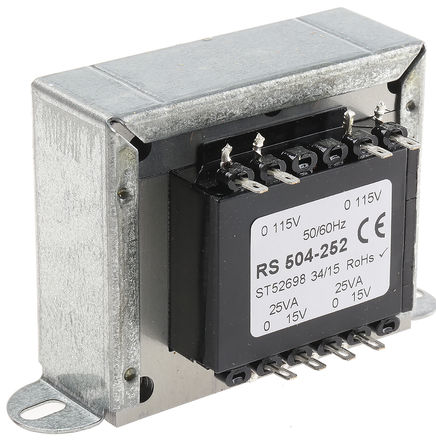


12VDC kreds løb

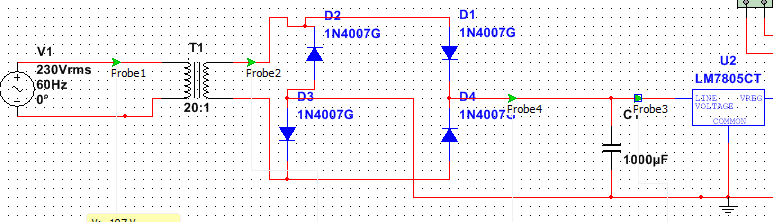
Som der kan ses på Figur nr, er der blevet brugt sikringer til at beskytte de forskellig dele i printpladen imod de høj strøm. Sikringerne er valgt ved at aflæse den maks. strøm, der må løbe igennem de forskellige komponenter, ude fra deres datablade.

**Voltage Transformer (VT)**

Spændingen fra stik kontakten er 230 VAC, da vores enheder skal max bruge 12 VDC er det nødvendligt med en transformer.  
Der vil blive brugt sådan en transformer model 504-252 fra RS Pro



Desuden er der brugt en diode bro for at ens arte spændingen fra AC til DC, som det kan ses af figur nr



# LED driver:

For at planter kan gro skal der tilføres energi i form af lys. Vi ønsker ikke at spilde energi på at simulere hele solens lysspektrum men kun den del de fleste planter bruger. Da planter er grønne og derved reflektere grønt lys vil det være spild af energi at belyse planten med grønt lys.

## Beregninger

## C:\Users\Peter\Dropbox\4. semester\Semesterprojekt 3 - Væksthus\05. Implementering & Design\Hardware\Belysning\CE, CCG 2.PNG

V\_R1 = 0,7V

V\_Q1BE = 1,4V

VCC = 12V

VEE = 5V

V\_Q1BE er base emitter spændingen over Q1.

I1 er strømmen gennem dioden

I2 er strømmen ind i basen på Q1

I1 = 0,7V/1Ω =700mA

R2 = (5V-1,4V)/1mA = 3,6 Ω

1mA er valgt da β for Q1 er på 1000 og I1 er på kun 700mA

Tab i Q1

Ved røde LED’er vil spændingsfaldet over LED’erne være 7,2 V.

V\_LED1 = 7,2V

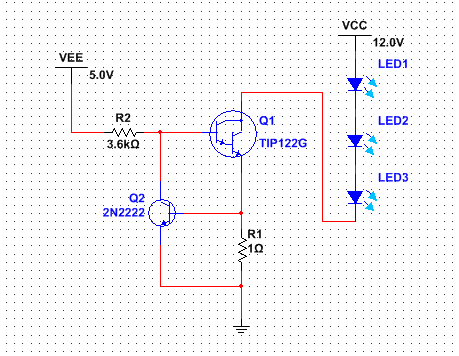
V\_R1 = 0,7V

I1 = 700mA

V\_Q1 = 12V – 0,7V – 7,2V = 4,1V

P\_Q1 = V\_Q1 \* I1 = 4,1V \* 0,7A = 2,87W

**Endeligt design**

****

Temperature Føler

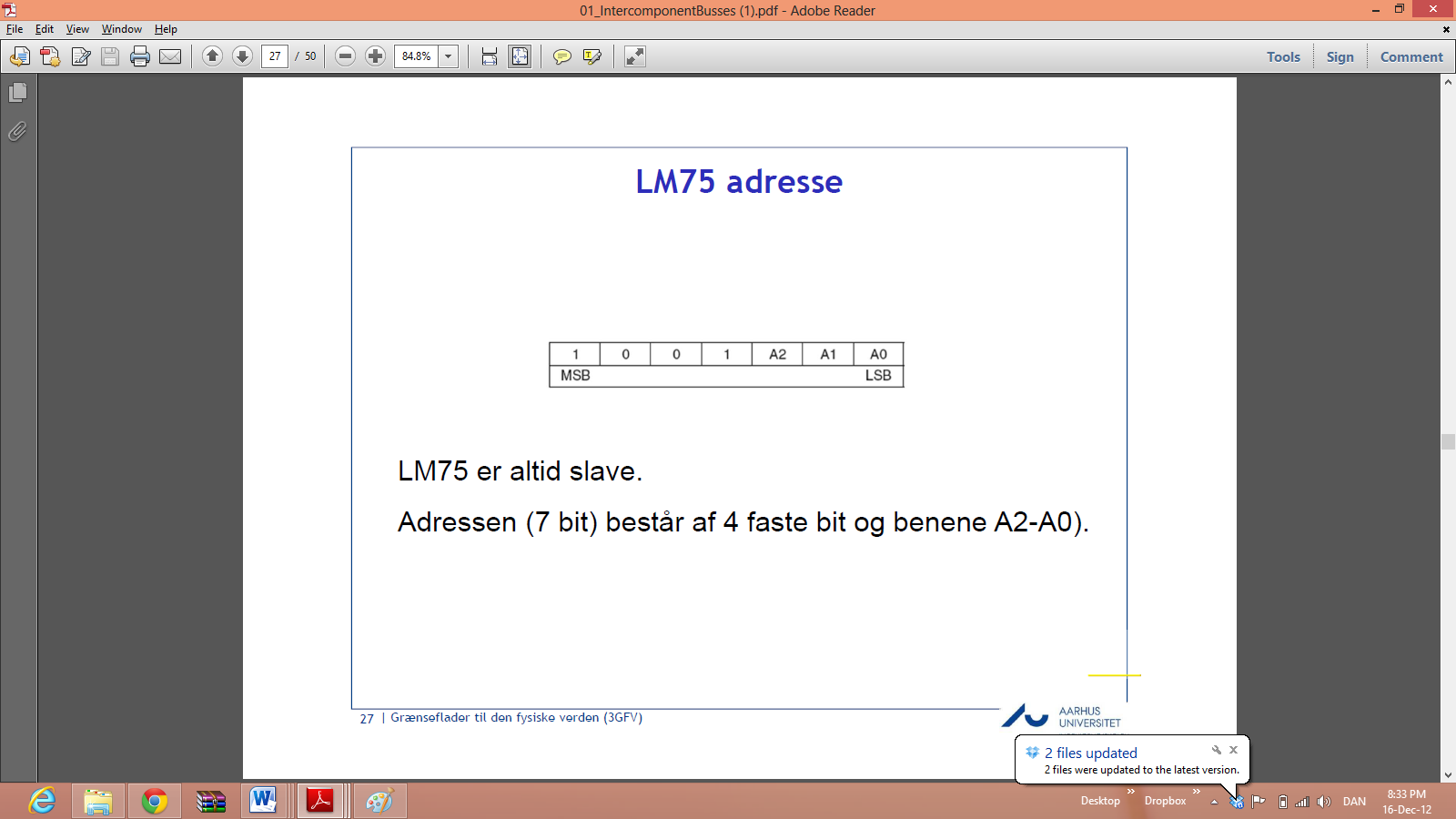
Der blev valgt at montere en temperatursensor, LM75m den bliver brugt til at registrere den aktuelle temparatur til systemet. Komponenten kan måle indenfor intervallet -50 til 125C, intervallet er meget større end de stillede krav. Den kan også måle en opløsning af 1C, hvilket opfylder kravene som blev valgt.

Denne komponent har en I2C interface, som kommunikere med PsoC.

LM75 bruges som en temperaturføler. Den skal aflæse temperatur og sende signalet videre til PsoC’n. Da den benytter I2C skal chippet derfor være en slave og psoc’n skal være master. Fordelene ved at bruge LM75 er den er selv en slave. Psoc’n skal kunne adressen for LM75’s adresse for at kunne aflæse fra den. LM75 har en I2C but som kan sende dataer videfre til Psoc’n.

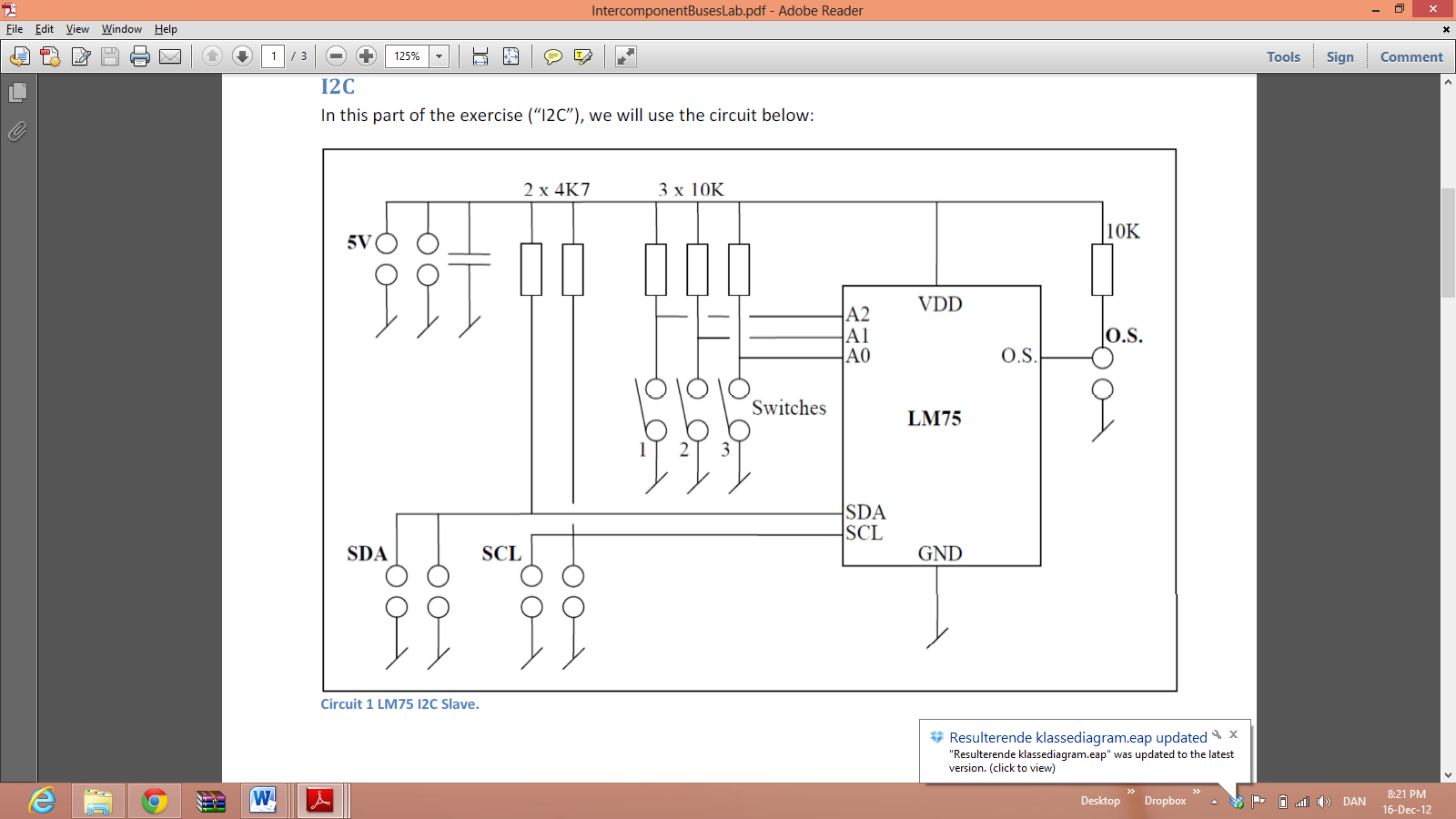
LM75 har en adresse som består af 7bit, de sidste 4 bit er faste, mens de tre første kan blive indstillet manuelt ved switches som er inkluderet med i printet, de kan bruges til at ændre på bit værdiern. Men de indstillet til at være aktiv lave.

Den nedenstående diagram viser adressen til LM75. A0-A2 er de bit som kan indstilles manuelt vha. Switchene.



Figur 2 LM75 Adresse

Pull-up modstandene til switchene og til SDA, SCL og OS. Det er ansvarlig for at forhindre modtagelse af 5V som kommer fra spændingsforsyning direkte til ground, der de ville være en kortslutning.



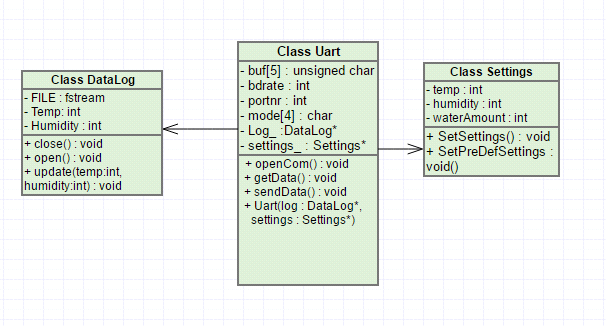
Figur 3 LM75 Temparatur føler

Design

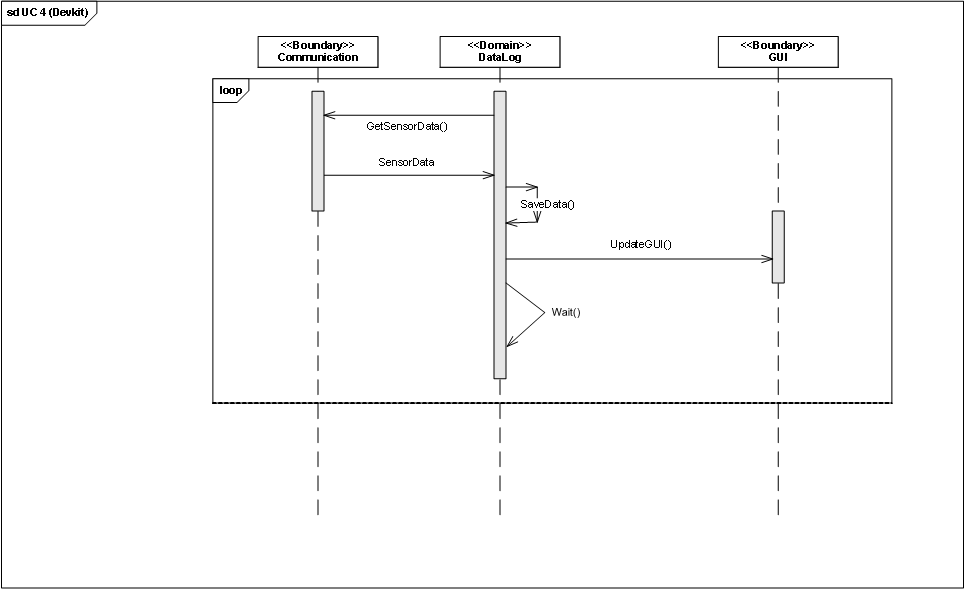
Software design

Devkit:

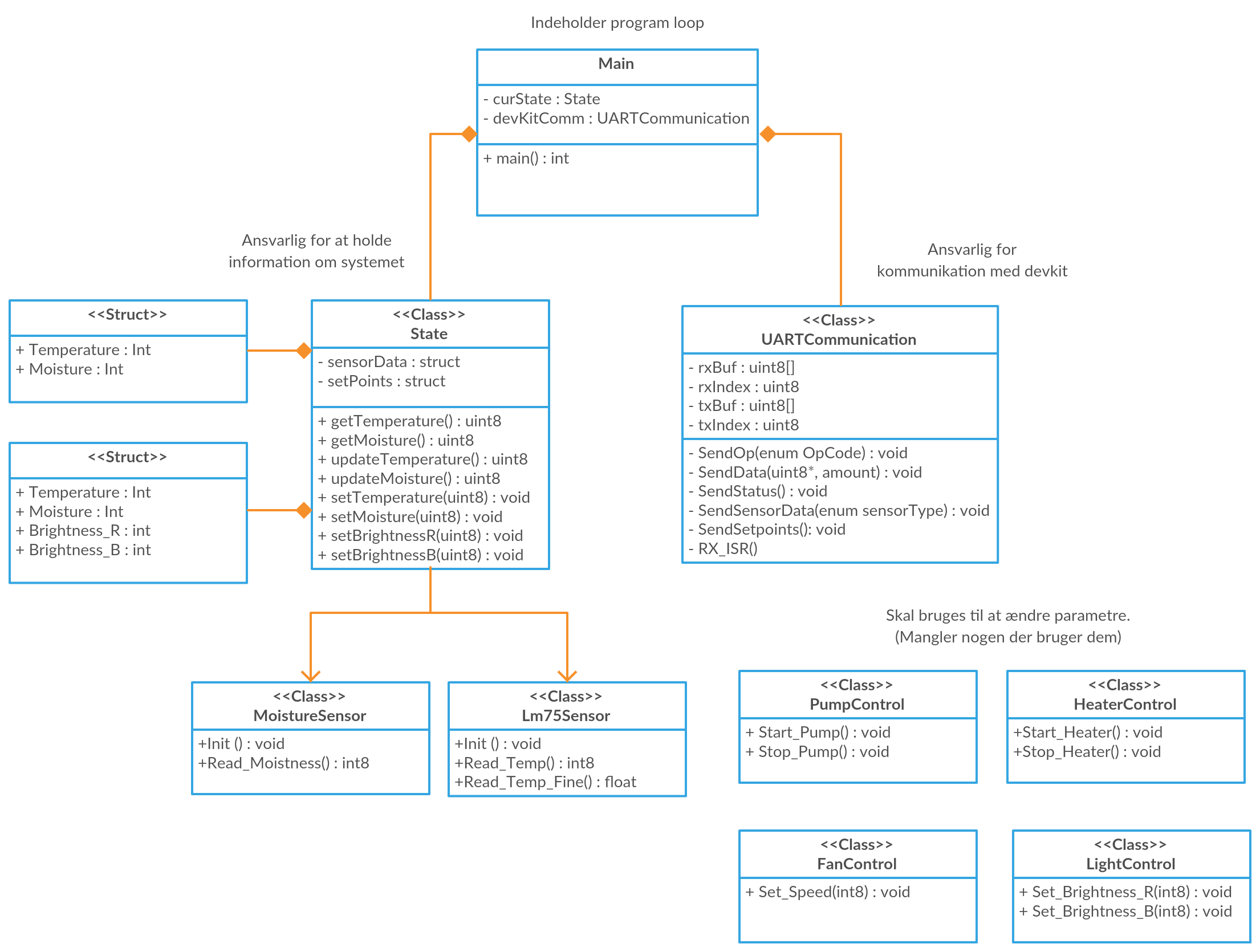
UML Klasse Diagram foreløbig(minus GUI):



SD til UC5 devkit:



Klasse diagram over systemet



Figur 4 Klasse Diagram

**Klasse diagram for hele Psoc software**

State klassen indeholder information om systemet, både sensor data og setpoints.

Den er også ansvarlig for indhenting af data fra sensorer.

Main er program loopet. Den kommer ikke til at indeholde meget mere end opsætning

og program loopet.

UARTCommunication er ansvarlig for kommunikation med Devkit. Den holder styr på en

buffer.

Der mangler en klasse til at håndtere regulering af systemet, samt en timer der

bestemmer hvor hvor ofte der bliver reguleret.

Der mangler også at bliver taget en beslutning om hvordan dette skal gøres

eksempelvis bare sekventielt, med en form for scheduler eller andet