

Autovandingssystem

Projektrapport

Afgangsprojekt af:
Kenn Hedegaard Eskildsen
Karsten Schou Nielsen

Vejleder:
Bjarne Funch Skipper

Ingeniørhøjskolen, Aarhus Universitet
18. maj 2017

Navn	Studienummer	Underskrift
Kenn Hedegaard Eskildsen	201370904	
Karsten Schou Nielsen	201370045	

Abstract

Resumé

Resume

Indhold

Abstract	ii
Resumé	iii
Indhold	iv
Arbejdsopgaver	v
1 Indledning	2
1.1 Læsevejledning	2
2 Projektformulering	3
3 Kravspecifikation	4
4 Design og Implementering	7
4.1 Hardware Design og Implementering.....	7
4.1.1 Kontrolboks	7
4.2 Software Design og Implementering	10
4.3 Projektafgrænsning	11
5 Udviklingsprocess	12
5.1 Udviklingsmodeller	12
5.1.1 V-Model og ASE-Model	12
5.2 Projektstyring	13
5.3 udviklingsværktøjer	13
6 Produktion	15
6.1 Produktionsovervejelser	15
6.2 Markedsundersøger	15
6.3 EMC og Godkendelser	15
7 Resultater og Diskussion	16
8 Fremtidigt Arbejde	17
9 Erfaringer	18
10 Konklusion	19
Litteraturliste	20

Arbejdsopgaver

Arbejdsopgaver

Denne side er bevidst blank.

1 Indledning

Denne Projektrapporten er en opsummeringen af projektdokumentationen og dækker hovedsageligt over de udviklingsmæssige overvejelser og beslutninger som projektgruppen har foretaget undervejs i udviklingen. Rapporten indeholder beskrivelse af udviklingen og design af 1. overordnede iteration af Autovandings-systemet. Autovandings-systemet kan via en kontrolboks tilkoblet brugenes vandforsyning og en sensor placeret i et ønsket gromedie videregive brugerens informationer om hhv. fugtighed og temperatur i mediet, der kommunikeres trådløst imellem kontrolboks og sensor. På baggrund af data fra sensoren kan systemet selv foretage vanding af gromediet afhængig af brugerens ønske og indstillinger.

1.1 Læsevejledning

Projektdokumentationen til denne rapport er skrevet kronologisk i forhold til de givne faser i ASE Modellen [1], på nær accepttestspecifikationen, som er udarbejdet i forlængelse af kravspecifikationen. Selve accepttesten er dog udført i slutningen af forløbet, deraf placeringen sidst i dokumentet. Den samme rækkefølge er ført i denne rapport under kapitel 4 fra side 7. På denne måde er der struktur i hvordan der kan søges mere information i projektdokumentationen.

2 Projektformulering

Som befolkningen vokser bliver det nødvendigt at bruge jordens ressourcer mere effektivt. Drikkevand er en ressource som i nogle områder er truet pga. forurening og overforbrug. Ved kun at bruge den nødvendige mængde af de nuværende drikkevandsressourcer vil der blive mere vand til vores efterkommer og vi vil have større mulighed for stadig at sikre rent drikkevand i hancerne. I Danmark vandes der i mange haver med rent drikkevand og her er der en mulighed for at kunne optimere brugen og samtidig sikre optimale grobetingelser for planterne. Ved at installere et automatisk fugtbaseret havevandingsanlæg sikres der at der ikke overvandes men samtidig også at planterne ikke mangler vand. Det automatiske fugtbaseret havevandingsanlæg består samlet af en sensor, en kontrolboks og en motorventil. Systemet virker ved at sensoren er placeret ved et gromedie og sender et trådløst signal indeholdende jordfugtigheden samt en overfladetemperatur, batteristatus og lysintensitet til kontrolboksen. Kontrolboksen åbner eller lukker herefter for den givne ventil afhængig af hvilken fugtighed kontrolboksen er præindstillet til og brugerne har mulighed for at aflæse fugtigheden og temperatur på kontrolboksen. Bruger skal selv tilslutte en haveslange til ventilen og kan derfor også selv bestemme hvilken type af havevander der skal tilkobles. En anden stor fordel ved systemet er at brugerne nu også har mulighed for at være væk fra sin bopæl eller fritidshus igennem længere tid, uden at skulle have andre til at tilse sin have.

3 Kravspecifikation

I dette afsnit er de opstillede krav for Autovandingssystemet beskrevet. Disse krav er opstillet ud fra projektformuleringen og ud fra forestillinger om relevante test af systemet. Funktionelle krav er opstillet ud fra Use Cases og ikke-funktionelle krav er opstillet ud fra relevante målbare størrelser.

I figur 1 på side 6 ses Use Cases for systemet samt en kort beskrivelse af dem. De funktionelle krav for systemet er udledt ud fra disse og beskrevet herunder. For at se den komplette beskrivelse af kravene for systemet henvises til 2 Kravspecifikation på side 5 i dokumentationen.

- **UC1: Tænd system** sørger for opstart af systemet.
- **UC2: Aflæs fugtighed og temperatur** sørger for aflæsning af fugtighed og temperatur fra sensoren
- **UC3: Indstil ønsket fugtig** benyttes til, på kontrolboksen at indstille den ønskede fugtighed i det medie hvori sensoren er placeret
- **UC4: Indstil åbningstid** giver brugeren mulighed for at indstille hvor længe der skal vandes ad gangen
- **UC5: Indstil vandingsinterval** giver brugeren mulighed for at indstille intervallet imellem vandingerne
- **UC6: Åbn/luk for ventil manuelt** giver brugeren mulighed for at åbne og lukke ventilen manuelt
- **UC7: Indstil vandingstidspunkt** giver brugeren mulighed for at indstille hvornår på dagen der ønskes at vande
- **UC8: Par sensor og kontrolboks** giver brugeren mulighed for parre sensor og kontrolboks

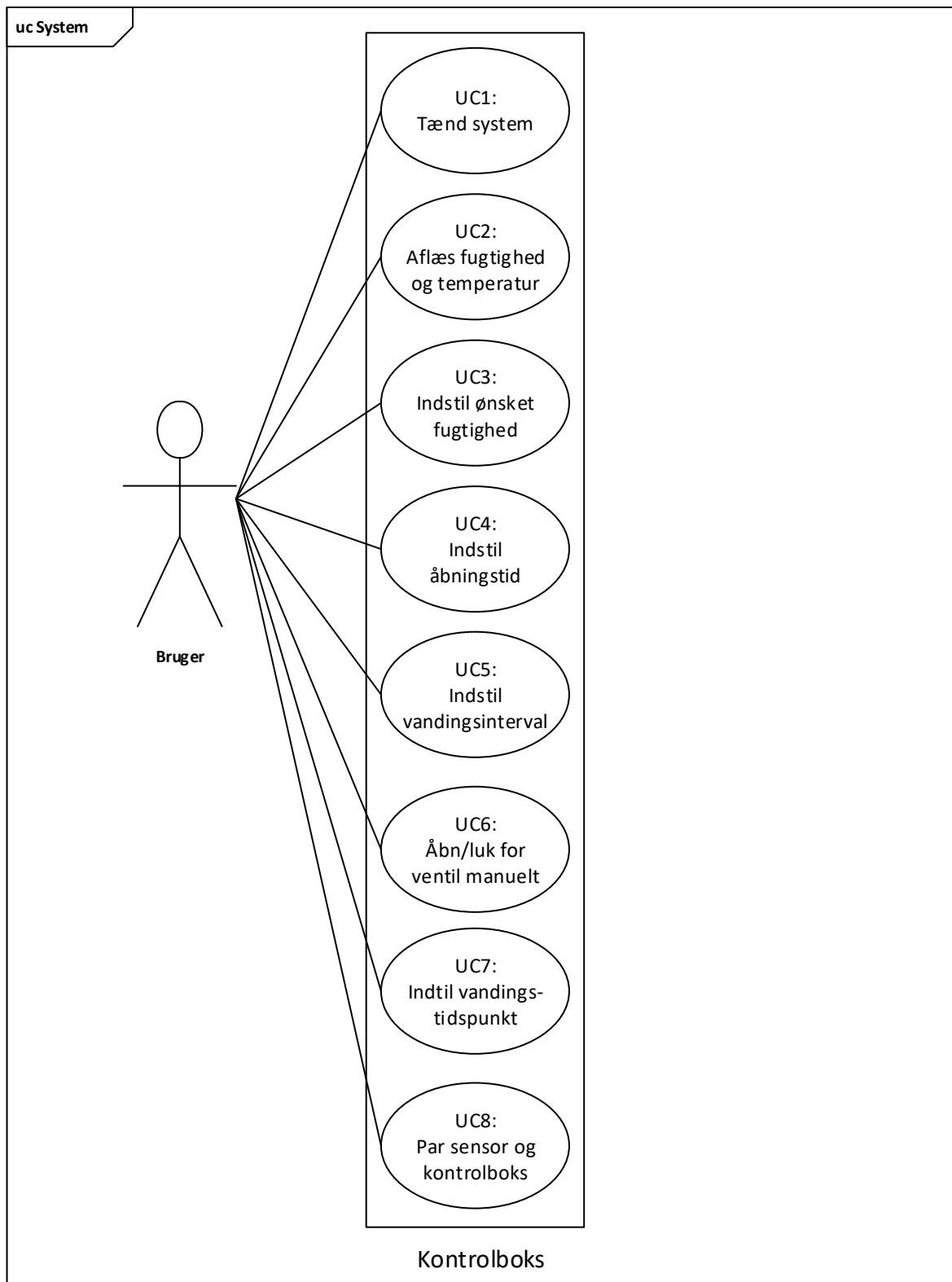
I tabel 1 opstilles de funktionelle krav til systemet, disse krav skal være opfyldt for at systemet betragtes som fuldt funktionelt. Kravene listes med et unik ID således de kan spores tilbage igennem hele dokumentationen. Kravenes ID er givet ved:

- MC: *Mechanical Constraints*: Dette er de mekaniske krav til systemet
- EL: *Electrical Constraints*: Dette er de elektriske krav til systemet
- CM: *Communications Constraints*: Dette er de krav der vedrører kommunikation i systemet
- UX: *User Experience*: Dette er de bruger-relaterede krav til systemet

ID:	Krav:	Prioritet:
MC_01	Systemet skal bestå af en sensor samt en kontrolboks	<i>Must</i>
MC_02	Sensoren skal måle temperatur, jordfugtighed samt lysintensitet	<i>Must</i>
MC_03	Kontrolboksen skal vise jordfugtighed og temperatur på et display	<i>Would</i>
MC_04	Kontrolboksen skal indeholde et tastatur	<i>Would</i>
MC_05	Kontrolboksen skal drive en indbygget motorventil til at åbne og lukke for vandet	<i>Would</i>
MC_06	Systemet skal kunne pre-indstilles til en specifik afgrøde	<i>Could</i>
MC_07	Systemet skal kunne måle PH-værdien i jorden	<i>Could</i>
MC_08	Systemet skal kunne betjenes fra en smartphone-applikation	<i>Would</i>
MC_09	Systemet skal kunne tilsluttes flere sensorer	<i>Would</i>
EL_01	Sensoren skal være batteridrevet	<i>Must</i>
EL_02	Kontrolboksen skal kunne forsynes med en 3. parts 5V AC/DC adaptor	<i>Must</i>
CM_01	Kontrolboksen skal kunne udveksle data med sensoren via en trådløs forbindelse	<i>Must</i>
CM_02	Sensor og kontrolboks skal parres manuelt	<i>Must</i>
CM_03	Kommunikation fra sensor til kontrolboks skal virke ved minimum 30m	<i>Should</i>
UX_01	Bruger skal have mulighed for at åbne/lukke for ventilen manuelt på kontrolboksen	<i>Must</i>
UX_02	Bruger skal kunne indstille en ønsket jordfugtighed på kontrolboksen	<i>Must</i>
UX_03	Bruger skal kunne indstille en åbningstid for ventilen	<i>Must</i>
UX_04	Bruger skal kunne vælge automatisk vandingstidsrum	<i>Must</i>
UX_05	Kontrolboksen skal kunne måle om slangen er sprunget fra, og give fejlmelding	<i>Should</i>
UX_06	Bruger skal kunne indstille en tidsbaseret vandingssekvens	<i>Should</i>

Tabel 1: Funktionelle krav

De ikke-funktionelle krav findes i Kapitel 2 Kravspecifikation i dokumentationen.



Figur 1: Use Cases for autovandingssystem

4 Design og Implementering

Efter at kravspecifikationen var på plads gik gruppen i gang med implementeringen af både hardware og software. Hardwaren blev lavet først da det herefter var nemmere at skrive softwaren. Dog har der være flere versioner af softwaren samtidig med at hardwaren er blevet ændret. Eks. er der blevet skrevet software til at teste den resistive-målemetode af jordfugten og lavet automatisk kalibrering af kredsløbet til fasemåling af den kapacitive-målemetode. Disse software funktioner er ikke blevet implementeret eller dokumenteret da hardwaren siden er blevet ændret. For et fuld overblik over disse ændringer henvises til projektdokumentationen under afsnittet implementering. Der gives herunder en kort beskrivelse af hver blok og hvordan de er blevet designet og implementeret.

4.1 Hardware Design og Implementering

4.1.1 Kontrolboks

Mikroprocessor

Mikroprocessoren til kontrolboksen er valgt til at være en Atmel ATMEGA8L i TQFP pakke. Samme mikroprocessor er valgt til sensoren. Valget stod mellem Microchip's PIC serie eller Atmels ATMEGA serie og da der var størst kendskab til programmering af Atmel var det den producent der blev valgt. Fordelen ved Atmel er samtidig at programmet til at programmere mikroprocessoren i, AtmelStudio, er gratis og at der er nogle gode biblioteker som er godt dokumenteret. Microchip kan derimod godt være svære at gennemskue med hensyn til hvad der kræves betaling for.

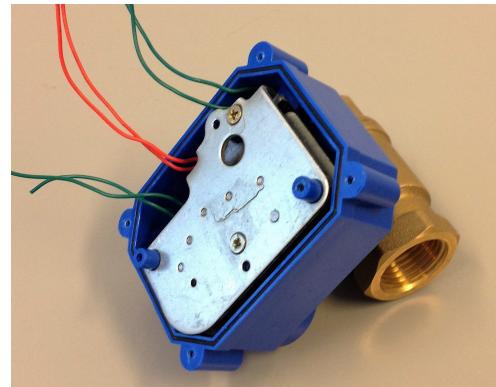
Af perifere enheder på kontrolboksen bruges der SPI til transceiveren, I2C til displayet samt en del analoge pins til kontrol af motorventil og tastatur. På sensoren bruges der også den indbyggede 10-bit SAR ADC.

Motorventil

Der blev hos den kinesiske producent Flow-Control bestilt nogle motorventiler hjem. Disse blev i første omgang valgt grundet deres lave pris, omkring 150kr stykket og har en tilslutning som fylder kravet om den skal kunne monteres på et 1/2"vandrør. Ventilen blev adskilt og det sås at der sat et lille print med et relæ til styring af ventilen. Styringen var lavet således at når den enten var i fuldt lukket eller åben position trak den intet strøm. 2 ses motorventilen som den kommer fra producenten. På Figur 3 ses motorventilen adskilt, hvor de grønne ledninger er til posotions-kontakterne og de røde er til motoren.

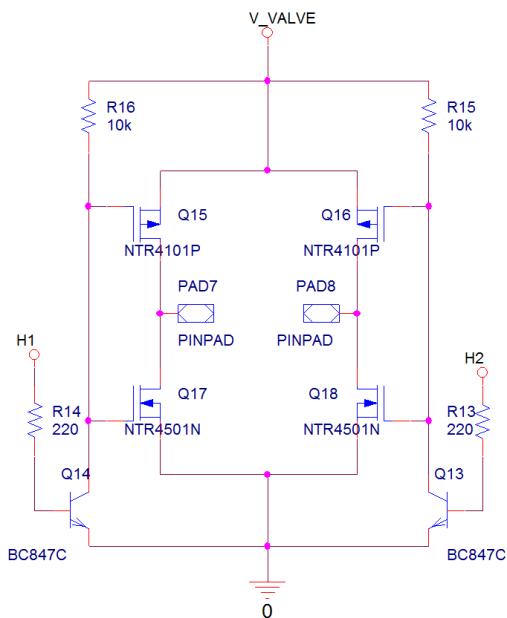


Figur 2: Motorventil



Figur 3: Motorventil adskilt

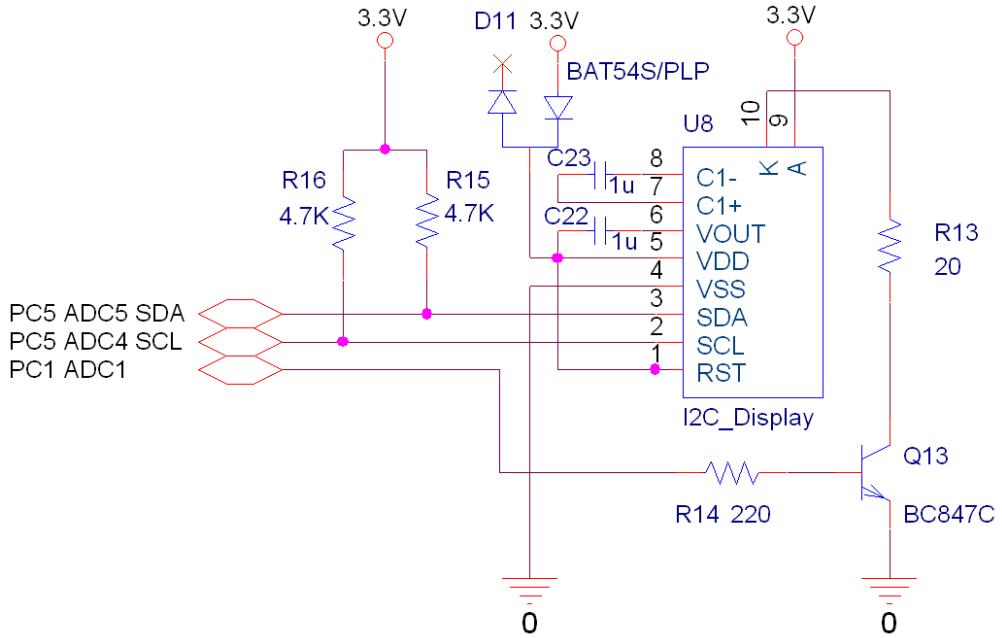
Der blev dog i første omgang designet en relæstyring af ventilen. Men den viste sig at være dyr at producere samtidig med at relæet træk en del strøm når det var sluttet. Herefter blev der designet en H-bro som bruger minimal tomgangsstrøm.



Figur 4: Diagram over H-bro'en

Display

Af krav til displayet var at det skulle køre på 3.3V, kommunikere over I2C bus og have 16X2 karakterer og have baggrundsbelysning. Ved undersøgelse af flere displays blev Newheaven's display *NHD-C0216CiZ-FSW-FBW-3V3* udvalgt. Dette skyldes primært at display'et har indbygget DC-DC step-up converter således at displayet kan forsynes ned til 2.8V. Normalt kører LCD displays på 5V, hvilket er over forsyningskravet på 3.3V. Opsætning og kommunikation sker via I2C som kravet foreskriver. Diagrammet for display'et ses på Figur 5.



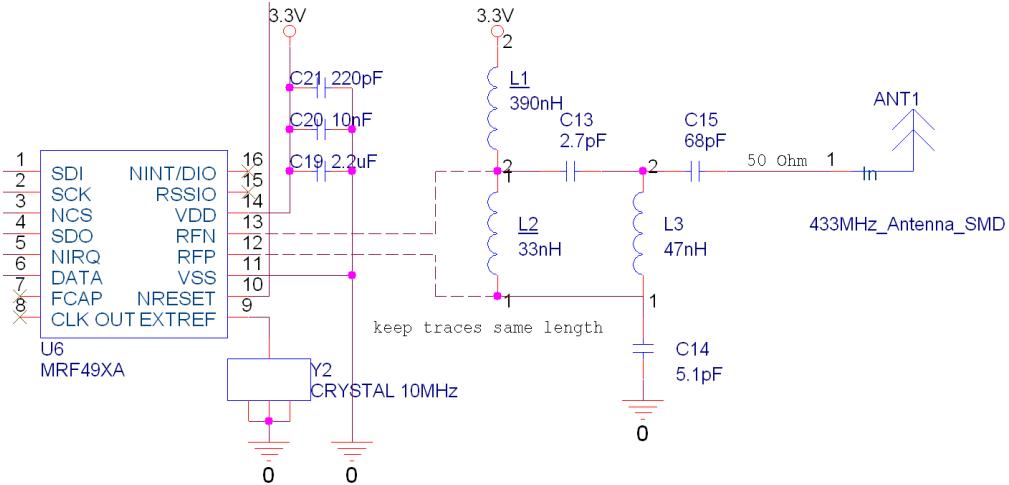
Figur 5: Diagram over I2C Display

D11 sidder for at sænke forsyningsspændingen til omkring 3V da forsyningen ikke må overskride 3.3V. C22 sidde over pin5 og pin6 som udgangskondensator for Step-up konverteren, C23 sidder over pin7 og pin8 som en integreret del af step-up konverteren og disse værdier er valgt ud fra databladet anbefalinger. Derudover benyttes Q13 for at kontrollere LED'en til baggrundsbelysningen hvormed dette kan styres fra software. SDA og SCL linjerne har pull-up modstande påsat som en del af standardopsætning for I2C-bussen.

HF Transceiver

Der er valgt at benytte en transceiveren MRFX49A fra Microchip, denne er valgt da den opfylder designkravene. Derudover skal denne kreds ikke forprogrammeres som mange andre RF-kredse skal, den kan derimod sættes op ved initialisering fra mikroprocessoren som krævet af systemet. Chippen er som skræddersyet til dette system da den er designet til batteri-applikationer. Den har en indbygget "low battery voltage detection" som afgiver et interrupt til mikroprocessoren når batterispændingen når under et forudindstillet threshold som sættes internt i transceiveren. Dette interrupt bliver brugt på sensoren til at enable boostkonverteren, mere info kan finde i implementeringsafsnittet for transceiveren på sensoren i dokumentationen. Derudover har den også en indbygget wake-up timer således at chippen kan sættes i sleep-mode, herved minimeres strømforbruget til nogle få μ A, den kan herefter vækkes op igen efter en forudindstillet tidsperiode. Transceiveren kan kommunikere ved 433/868/915 MHz og ligger derfor i et frekvens område som ofte benyttes i consumer-elektronik. Transceiveren benytter sig af FSK modulation.

På Figur 6 ses implementering af transceiveren på kontrolboksen.



Figur 6: Diagram af MRF49XA på kontrolboksen

Som udgangspunkt er databladets anbefalinger fulgt. Balun-kredsløbet er designet på baggrund af referencedesignet i databladet. Her er det vigtigt at holde sig til at lavtolerance-komponenter for at holde sig tæt på de 50Ω udgangsimpedans som det kræves af antennen. Det skal dog nævnes at der i databladet er byttet om på værdierne til balun-kredsløbet ved 433MHz og 868MHz i tabellerne under diagrammerne. Det er derfor svært at vide hvilke værdier der er de rigtige. Fejlen blev først opdaget tilsidst i projektet da rækkevidden ikke kunne komme længere op end ca 10m. Det er derfor nærliggende at tro det er de forkerte værdier som er blevet monteret.

4.2 Software Design og Implementering

Software design

4.3 Projektafgrænsning

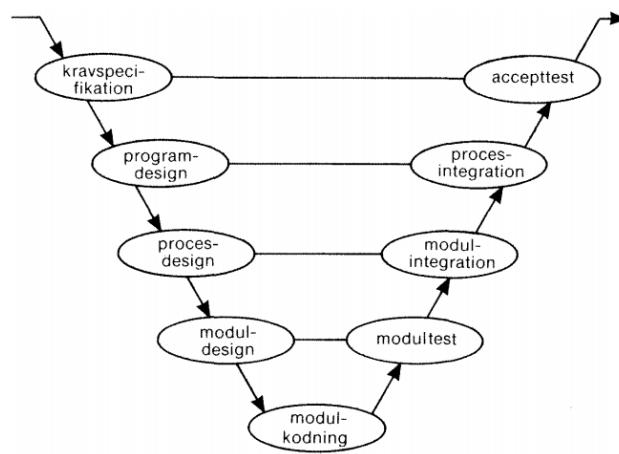
Projekt afgrænsning

5 Udviklingsprocess

5.1 Udviklingsmodeller

5.1.1 V-Model og ASE-Model

Under udvikling af dette projekt at det valgt at følge V-Modellen [1]. Denne model definere forløbet for udviklingsprojektet og eigner sig godt til et iterativt projekt, da der let kan køres flere iterationer under implementeringsfasen. Se Figur 7 for denne model.



Figur 7: V-model

I dette projekt er modul- og process-integration dog udeladt, og projektet har holdt sig til modul samt accepttests. Derudover er der, som det også fremgår af den iterative beskrivelse i projektdokumentationen udført løbende modultests under implementering af hardware og software.

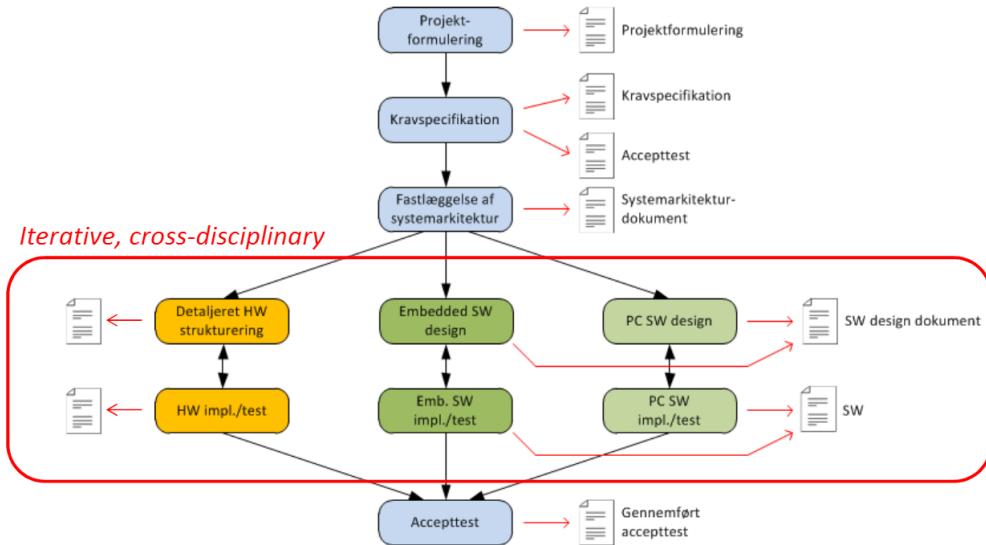
Ud over V-Modellen, er ASE-Modellen [1] taget i brug som en vejledning til gennemførelse af projektet. ASE-modellen kobler hver fase i udviklingsprojektet med en dokument, disse dokumenter udgår herefter projektdokumentationen. Derudover kan der med denne model efter defineret Systemarkitektur udviklet hardware og software hver for sig i et givent antal iterationer, dette samles så til sidst i Accepttesten for hele systemet. Oversigt over ASE-modellen ses på figur 8 på 13.

SysML

Projektet har anvendt SysML primært i systemarkitektur-fasen, for at beskrive systemet bedst muligt ud industristandarden for opbygning logiske blokke. Muligheden for derefter at beskrive overordnede blokke har givet mulighed for forskellige abstraktionsniveauer i udviklingen af projektet. Veldefinerede grænseflader mellem blokke ved hjælp af SysML har også bidraget til en mere klart defineret designfase og implementeringsfase.

Versionsstyring

Der er anvendt versionshistorik på dokumenter i projektdokumentationen samt projektrapport i for git, og for resten af projektet i form af dropbox. Væsentlige ændringer i eks. design har givet anledning til versions-aændring, hvilket hjælper med at holde styr på hvilke større ændringer projektet har gennemgået.



Figur 8: ASE-model

5.2 Projektstyring

Tidsplan - Mødestruktur - Arbejdsfordeling

5.3 udviklingsværktøjer

I dette afsnit vil de forskellige udviklingsværktøjer, som er blevet anvendt under dette projekts design-, implementerings- og integrationsproces, blive gennemgået.

Cadence OrCAD Capture er brugt til at designe alle hardwarediagrammer til kontrolboks og sensor. Programmet blev valgt da dette er industristandard, samt at der i projektgruppen er et rigtig godt kendskab til dette arbejdsmiljø. Der har ligeledes været mulighed for at oprette et stort antal af egne komponenter der er brugt til designs. Derudover Cadence pSpice-modulet blevet brugt i forbindelse med alle simuleringer. OrCAD integrerer derudover fuldt ud med det valgte layout-tool.

Cadence Allegro er valgt som Layout-tool da der i projektgrupper er stor erfaring med dette program. Derudover integrerer det fuldt ud med det brugte design-tool, og mulighed derudover crossprobing som markant nedsætter arbejdstiden.

Atmel Studio 7.0

Atmel Studio 7.0 er benyttet som udviklingsmiljø til al software. Dette miljø er skab af fabrikanten til den valgte mikrokontroller. Derfor var det oplagt at skrive i dette tool.

AWR Design Environment 11

AWR Design Environment 11 designet af native Instruments er lavet til beregner på højfrekvente kredsløb, her er det benyttet til at lave beregninger på mikrostrip-linjerne til antennerne.

WaveForms

WaveForms er brugt til at teste de forskellige hardwareenheder samt input/output porte af de forskellige embeddede enheder. WaveForms er et program der integrerer med Analog Discovery-enheten, som begge er produceret af Analog Devices, og er et multi-funktions instrument med mulighed for bland andet at agere oscilloskop og funktionsgenerator. Fordelen med Analog Discovery

er den mobilitet der gives under udviklingen.

PTC MathCad Prime

PTC MathCad Prime er benyttet til at lave diverse beregninger.

Maplesoft Maple

Maplesoft Maple er benyttet til at lave diverse beregninger.

MathWorks MATLAB

MathWorks MATLAB er benyttet til at lave diverse beregninger samt diverse plot og dataanalyser.

Git

Git er et versionsstyringsværktøj til vedligeholdelse af diverse dokumenter. Som repository host er der valgt Github grundet stabilitet og projektgruppens tidligere arbejde med dette miljø. Projektgruppen har valgt at lægge kildekoden til bl.a. denne rapport og Projektdokumentationen på Git for netop at opnå en kraftfuld versionsstyring af hele projektet.

6 Produktion

6.1 Produktionsovervejelser

6.2 Markedsundersøger

6.3 EMC og Godkendelser

7 Resultater og Diskussion

Resultater og Diskussion

8 Fremtidigt Arbejde

Eftersom at denne prototype kun er 1. iteration af det samlede system er der en del planer for fremtidigt arbejde, det er derfor valgt at dele op i enkelte afsnit afhængig af hvad det omhandler.

9 Erfaringer

Erfaringer

10 Konklusion

Konklusion

Litteraturliste

- [1] Introduction to System Engineering Compendium

Type: *kompodium*

Forfattere: *Forskellige forfattere*

Udgivelsessted: *Arhus Universitet*

Udgivelsesår: *Dato ukendt*

- [2] xxx

Type: *xx*

Forfattere: *xx*

Udgivelsessted: *xxx*

Udgivelsesår: *xx*

Link: *xxx/*

