

Autovandingssystem

Projektrapport

Afgangsprojekt af:
Kenn Hedegaard Eskildsen
Karsten Schou Nielsen

Vejleder:
Bjarne Funch Skipper

Ingeniørhøjskolen, Aarhus Universitet
17. maj 2017

Navn	Studienummer	Underskrift
Kenn Hedegaard Eskildsen	201370904	
Karsten Schou Nielsen	201370045	

Abstract

Resumé

Resume

Indhold

Abstract	ii
Resumé	iii
Indhold	iv
Arbejdsopgaver	vi
1 Indledning	2
1.0.1 Læsevejledning.....	2
2 Projektformulering og systembeskrivelse	3
3 Kravspecifikation	4
4 Design og Implementering	5
4.1 Hardware Design og Implementering.....	5
4.1.1 Kontrolboks.....	5
4.2 Software Design og Implementering.....	8
4.3 Projektafgrænsning.....	9
5 Udviklingsprocess	10
5.1 Udviklingsmodeller.....	10
5.1.1 V-Model og ASE-Model.....	10
5.1.2 SysML.....	10
5.1.3 Versionsstyring.....	10
5.2 Projektstyring.....	11
5.2.1 Cadence OrCAD Capture.....	11
5.2.2 Cadence allegro.....	11
5.2.3 Atmel Studio 7.0.....	11
5.2.4 AWR Design Environment 11.....	11
5.2.5 WaveForms.....	12
5.2.6 PTC MathCad Prime.....	12
5.2.7 Maplesoft Maple.....	12
5.2.8 MathWorks MATLAB.....	12
5.2.9 Git.....	12
6 Produktion	13
6.1 Produktionsovervejelser.....	13
6.2 Markedsundersøger.....	13
6.3 EMC og Godkendelser.....	13
7 Resultater og Diskussion	14
8 Fremtidigt Arbejde	15
9 Erfaringer	16

10 Konklusion	17
Litteraturliste	18

Arbejdsopgaver

Arbejdsopgaver

Denne side er bevidst blank.

1 Indledning

Denne Projektrapporten er en opsummering 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 Autovandingssystemet. Autovandings-systemet kan via en kontrolboks tilkoblet brugernes vandforsyning og en sensor placeret i et ønsket gromedie videregive brugeren 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.

1.0.1 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 5. På denne måde er der struktur i hvordan der kan søges mere information i projektdokumentationen.

2 Projektformulering og systembeskrivelse

Projektformulering

3 Kravspecifikation

Krav

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 været 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

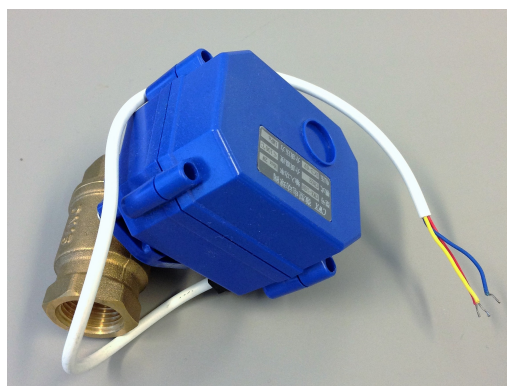
Mikroprocessor

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.

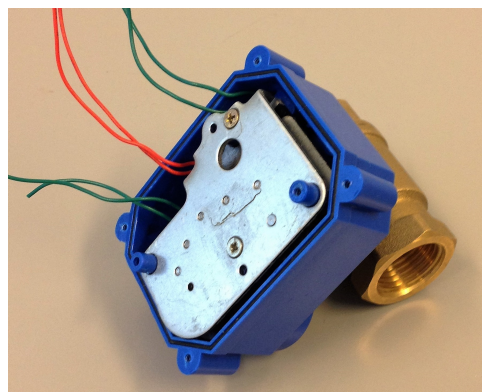
Af perifere enheder på kontrolboksen bruges der SPI til transceiveren, I2C til displayet samt en del analoge pins til kontrol af motorventil og tastatur. På sensoren bruges der også den indbyggede 10-bit SAR ADC.

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. 1 ses motorventilen som den kommer fra producenten. På Figur 2 ses motorventilen adskilt, hvor de grønne ledninger er til positions-kontakterne og de røde er til motoren.

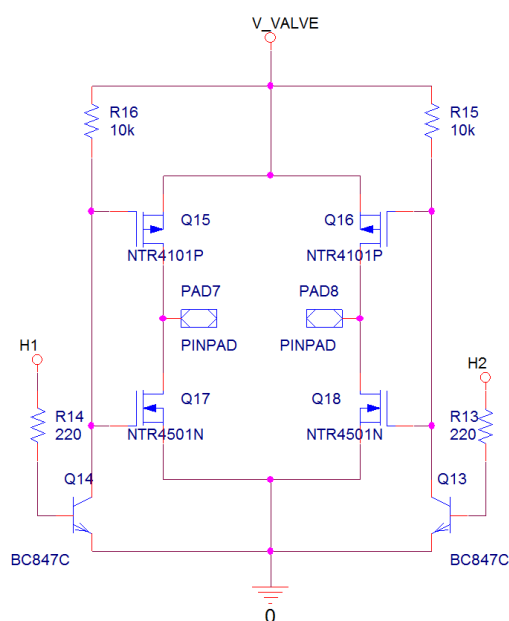


Figur 1: Motorventil



Figur 2: Motorventil adskilt

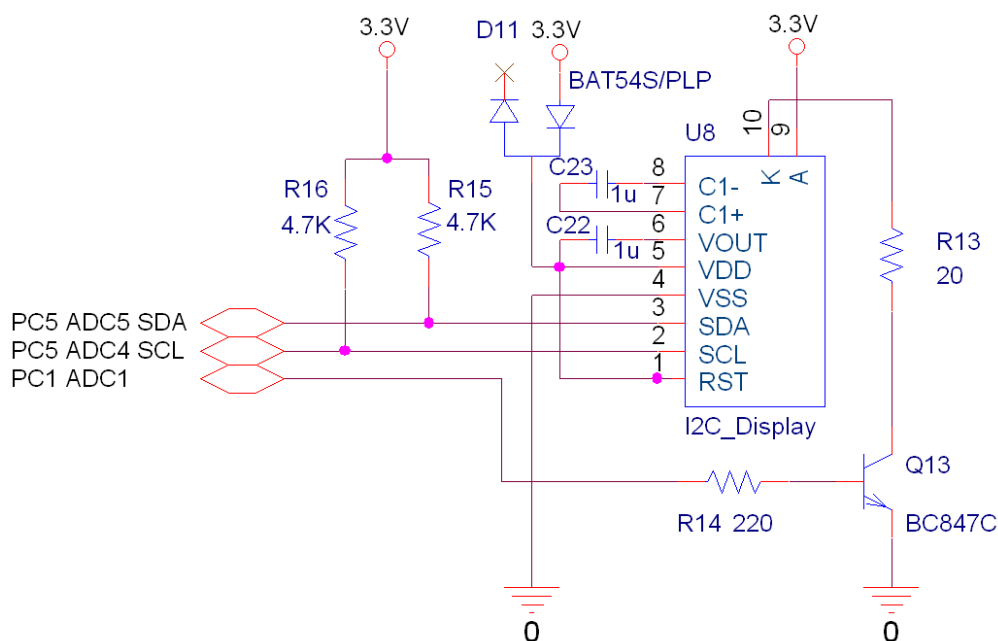
Der blev dog i første omgang designet en relæstyring af ventilen. Men den viste sig at være dyr at producere samtidig med at relæet træk en del strøm når det var sluttet. Herefter blev der designet en H-bro som bruger minimal tomgangsstrøm.



Figur 3: 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 Newhaven'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 4.



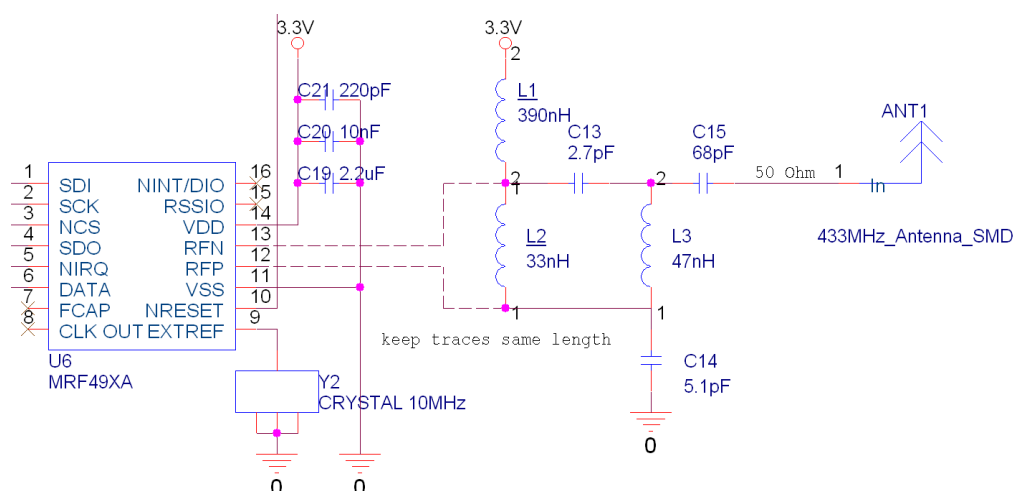
Figur 4: Diagram over I2C Display

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 sidde 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 μA , 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 5 ses implementering af transceiveren på kontrolboksen.



Figur 5: 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 $50\ \Omega$ 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

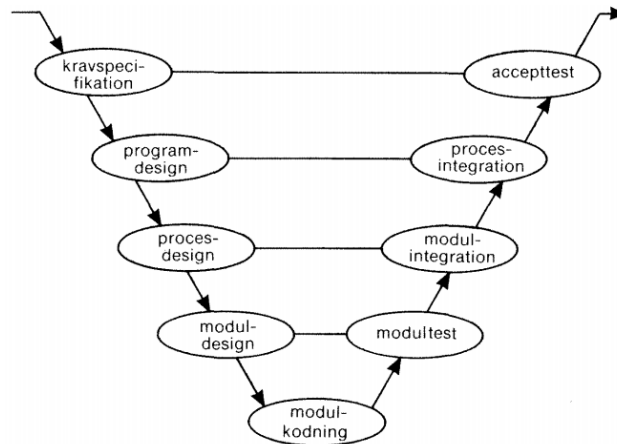
Projekt afgrænsning

5 Udviklingsprocess

5.1 Udviklingsmodeller

5.1.1 V-Model og ASE-Model

Under udvikling af dette projekt er det valgt at følge V-Modellen [1]. Denne model definerer forløbet for udviklingsprojektet og egner sig godt til et iterativt projekt, da der let kan køres flere iterationer under implementeringsfasen. Se Figur 6 for denne model.



Figur 6: 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.

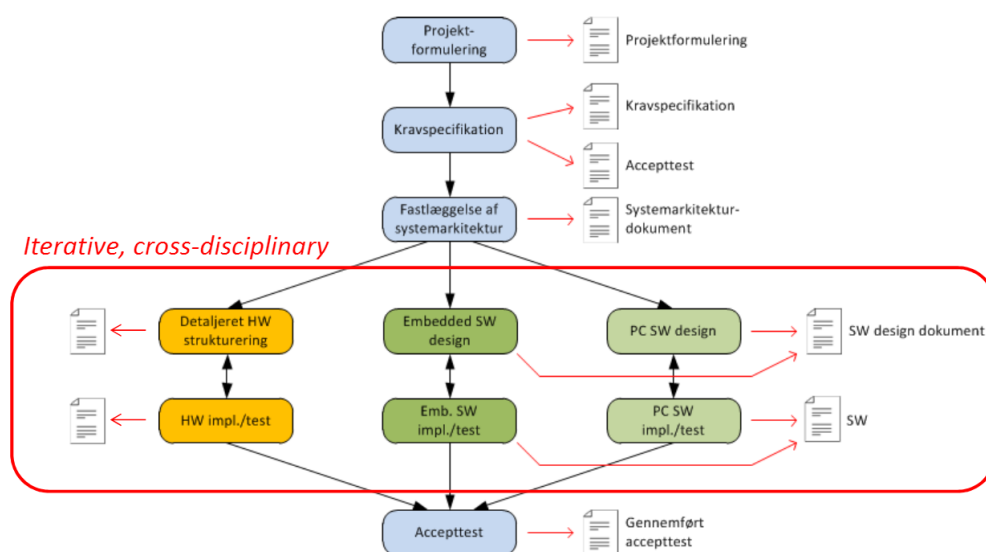
Ud over V-Modellen, er ASE-Modellen [1] taget i brug som en vejledning til gennemførelse af projektet. ASE-modellen kobler hver fase i udviklingsprojektet med en dokument, disse dokumenter udgår herefter projektdokumentationen. Derudover kan der med denne model efter defineret Systemarkitektur udviklet 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 7 på 11.

5.1.2 SysML

Projektet har anvendt SysML primært i systemarkitektur-fasen, for at beskrive systemet bedst muligt ud i 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.

5.1.3 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-ændring, hvilket hjælper med at holde styr på hvilke større ændringer projektet har gennemgået.



Figur 7: ASE-model

5.2 Projektstyring

Tidsplan - Mødestruktur - Arbejdsfordeling

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

5.2.1 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 integrer derudover fuldt ud med det valgte layout-tool.

5.2.2 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.

5.2.3 Atmel Studio 7.0

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

5.2.4 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.

5.2.5 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-enheden, som begge er produceret af Analog Devices, og er et multi-funktions instrument med mulighed for blandt andet at agere oscilloskop og funktionsgenerator. Fordelen med Analog Discovery er den mobilitet der gives under udviklingen.

5.2.6 PTC MathCad Prime

er benyttet til at lave diverse beregninger.

5.2.7 Maplesoft Maple

er benyttet til at lave diverse beregninger.

5.2.8 MathWorks MATLAB

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

5.2.9 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

Fremtidigt arbejde

9 Erfaringer

Erfaringer

10 Konklusion

Konklusion

Litteraturliste

- [1] Introduction to System Engineering Compendium

Type: *kompendium*

Forfattere: *Forskellige forfattere*

Udgivelsessted: *Arhus Universitet*

Udgivelsesår: *Dato ukendt*

- [2] xxx

Type: *xx*

Forfattere: *xx*

Udgivelsessted: *xxx*

Udgivelsesår: *xx*

Link: *xxx/*

