

Pakkeovervågning ved brug af Sun SPOTs

Peter Meyer Nielsen

Kongens Lyngby 2011
IMM-BSc-2011-21

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-BSc: ISSN 0909-3192

Summary

In this project the possibility of using wireless sensors to monitor a parcel transport is examined. The functionalities available in modern sensors are used to collect data during a transport. The collected information is used in an attempt to give the user a sensible and useful overview of the transport.

In the report it is studied which factors influence the quality and quantity of the monitoring, and to what degree these can be adjusted. In that connection the surveillance system is expanded, so that sensors can participate in a joint surveillance network.

As a result, the system is enhanced, so that sensors can communicate with each other during the transport, by which a significant increase in respectively the quality and quantity is achieved.

Resumé

I dette projekt undersøges muligheden for anvendelse af trådløse sensorer til overvågning af en pakketransport. De funktionaliteter, som moderne sensorer stiller til rådighed, anvendes i et forsøg på at tegne et fornuftigt og brugbart billede af transporten. Brugeren af systemet skal nemt kunne danne sig et overblik over et transportforløb, ud fra de informationer som sensorerne opsamler under transporten.

I løbet af rapporten undersøges hvilke faktorer, der spiller ind på kvaliteten og kvantiteten af overvågningen, og i hvilken grad der kan ændres på disse. I den forbindelse ønskes overvågningssystemet udvidet, så flere sensorer kan indgå i et fælles overvågningsnetværk.

Resultatet er en udvidelse af systemet, hvor sensorerne kan kommunikere med hinanden under transporten, hvorved en væsentlig forøgelse af henholdsvis kvalitet og kvantitet opnås.

Forord

Dette bachelorprojekt er skrevet som afslutning på softwareteknologilinjen ved institut for Informatik og Matematisk Modellering på Danmarks Tekniske Universitet med projektperiode fra februar til juli 2011.

Jeg vil gerne sige tak til min vejleder Hans Henrik Løvengreen for god rådgivning i løbet af projektperioden. Derudover vil jeg også gerne takke Emma Christine Krabbe Thykier og Matthias Huygens Haamann, der i samme projektperiode også udførte et projekt med Sun SPOTs, for udveksling og diskussioner om Sun SPOTs egenskaber.

Kongens Lyngby, juli 2011
Peter Meyer Nielsen

Indhold

Summary	i
Resumé	iii
Forord	v
1 Indledning	1
1.1 Problemformulering	2
1.2 Fremgangsmåde	2
1.3 Rapportstruktur	2
2 Baggrund om Sun SPOT	5
2.1 Hardware	5
2.2 Software	7
2.3 Version	7
3 Kravanalyse og specifikation	9
3.1 Overvågning	9
3.2 Administrering	12
I Individuel overvågning	15
4 Analyse	17
4.1 Sun SPOT	17
4.2 Overvågning	27
5 Design	31
5.1 Kommunikation	31

5.2	Vært	33
5.3	Sun SPOT	38
6	Implementering og test	41
6.1	Vært	41
6.2	Sun SPOT	42
6.3	Test	43
II	Kollektiv overvågning	47
7	Analyse og design	49
7.1	Baggrund om distribuerede systemer	49
7.2	Vært	50
7.3	Sun SPOT	51
8	Implementering og test	61
8.1	Sun SPOT	61
8.2	Test	63
9	Konklusion	67
9.1	Anvendelsesmuligheder og udvidelser	68
A	Batterianalyse	71
A.1	Batteriniveaudiagrammer	71
A.2	Afladningsdiagrammer	73
A.3	Overvågning	75
B	Radioanalyse	77
B.1	Signalstyrke	77
B.2	Overførselshastighed	80
C	Temperaturmåleranalyse	83
D	Tests for individuel overvågning	87
D.1	Initialisering	87
D.2	Ændring af måleindstillinger	88
D.3	Start overvågning	89
D.4	Importer målinger	90
D.5	Andre situationer	90

E	Tests for kollektiv overvågning	97
E.1	Kontrolleret terminering af målende Sun SPOT	97
E.2	Ukontrolleret terminering af målende Sun SPOT	98
E.3	Ukontrolleret terminering af ikke-målende Sun SPOT	99
E.4	Lang og kort udeblivelse og tilbagevenden af en målende Sun SPOT	99
E.5	Andre situationer	100
F	UML-diagrammer	103

KAPITEL 1

Indledning

Omkring 1998 startede den moderne udvikling af små trådløse sensorer [8]. På grund af deres store anvendelsesspektrum i forhold til pris, er de blevet populære, og anvendes til mange automatiserede opgaver i informationssamfundet. Sensorerne anvendes til overvågning af alt lige fra dæktryk til temperaturer i et rum. Anvendelsesmulighederne er, i takt med forøgelsen af funktioner, stadig stigende. Det vil være interessant selv at prøve kræfter med trådløse sensorer, da det kræver en anderledes tilgang og tankegang at udvikle software til, i forhold til almindelig softwareudvikling.

Jeg vil i dette projekt kigge nærmere på overvågning af pakker. Når en pakke overgår til en pakkedistributør, ved man som afsender ikke meget om hvad pakken udsættes for. For at kortlægge hvad en pakke er udsat for under transporten, anvendes sensorer med mulighed for at opfange relevante værdier. Opsamlet data kan anvendes til at lave statistiske sammenligninger af forskellige pakkedistributører og bidrage til en evaluering af transporten, herunder give en indikation af hvad pakken har været udsat for.

1.1 Problemformulering

I mit projekt vil jeg udvikle et system til overvågning af transportforløb ved brug af trådløse sensorer. Der skal således måles værdier, som kan indikere om pakken har været åbnet, henstillet udenfor eller er blevet tabt. Da DTU er i besiddelse af sensorer fra Sun Microsystems, kaldet Sun SPOTs, har jeg lånt disse til at bevise overvågningskonceptet.

Sensorerne skal kunne overvåge individuelt og kollektivt. Ved individuel overvågning, forstås brugen af en enkelt sensor i én pakke, der således overvåger transportforløbet. Ved kollektiv overvågning ønskes det undersøgt om brugen af flere sensorer i samme forsendelse, kan forbedre kvaliteten af overvågningen.

Som et led i overvågningen anvendes en stationær computer, som agerer administrator for sensorerne, og det er her målte værdier opsamles, lagres og visualiseres.

1.2 Fremgangsmåde

På grund af projektets udformning, har jeg valgt at opdele rapporten i to dele. Den ene del omhandler overvågning ved brug af en enkelt Sun SPOT, mens den anden beskriver udvidelsen til brug af flere Sun SPOTs.

Det er vigtigt at skabe et overblik over Sun SPOT'ens begrænsninger med hensyn til de funktionaliteter den tilbyder og som kan være relevante for implementeringen. Analysen af teknologien ligger derfor til grund for den videre udvikling, der resulterer i den endelige implementering, til overvågning med én Sun SPOT.

Efter de basale overvågningsfunktioner, som gør Sun SPOTs'ene i stand til at foretage individuelle overvågninger, er udviklet, påbegyndes videreudviklingen til overvågning ved brug af flere Sun SPOTs. Herunder er det vigtigt at få analyseret forskellige designmønstre, til varetagelse af den kollektive overvågning, før et egentligt designvalg for implementeringen vælges.

1.3 Rapportstruktur

Herunder beskrives indholdet af de enkelte kapitler i rapporten. Som tidligere skrevet, er rapporten opdelt i to dele om henholdsvis individuel og kollektiv

overvågning.

Den del af rapporten, der omhandler den individuelle overvågning indeholder følgende kapitler:

Analyse: Her analyseres de generelle funktionaliteter på Sun SPOT'en, der kan være relevante for implementeringen. Herpå følger en egentlig analyse af problemstillingen ved individuel overvågning

Design: Beskriver kommunikationsdesignet, der skal være mellem enheden, der administrerer Sun SPOTs'ene og sensorerne. Derudover diskuteres de generelle designvalg

Implementering og test: Her fremhæves de mere interessante implementeringsdele for henholdsvis den administrerende enhed samt Sun SPOTs'ene. Herefter foretages tests af interaktionen mellem enhederne

Rapportdelen om kollektiv overvågning indeholder følgende kapitler:

Analyse og design: Her diskuteres forskellige designløsninger, der kan anvendes til at opnå kollektiv overvågning. På baggrund af designvalget beskrives det egentlige design mere dybdegående

Implementering og test: De mest interessante implementeringsudvidelser i Sun SPOT-programmet nævnes. Derudover testes den kollektive overvågning, hvor flere enheder indgår i overvågningen

KAPITEL 2

Baggrund om Sun SPOT

Sun SPOT står for Sun Small Programmable Object Technology og er en trådløs netværksnode. Den blev i sin tid udviklet af Sun Microsystems, som nu ejes af Oracle.

Sun SPOTs giver mulighed for radiokommunikation mellem andre enheder. Hver enhed har sin egen unikke 16 cifrede IEEE adresse. I løbet af rapporten vil jeg omtale de forskellige Sun SPOTs, jeg har fået udleveret, ved de sidste fire cifre.

2.1 Hardware

Sun SPOT'en består som udgangspunkt af to hardwarekort: et bundkort og et applikationskort. En Sun SPOT uden applikationskort har følgende specifikationer:

- 180 MHz 32 bit ARM920T core - 512Kb RAM - 4Mb Flash
- 2.4 GHz IEEE 802.15.4 radio med integreret antenne
- USB interface

- Størrelse: 41 x 23 x 70 mm
- Vægt: 54 gram
- Batteri: 3.7 V genopladelig 750 mAh lithium-ion

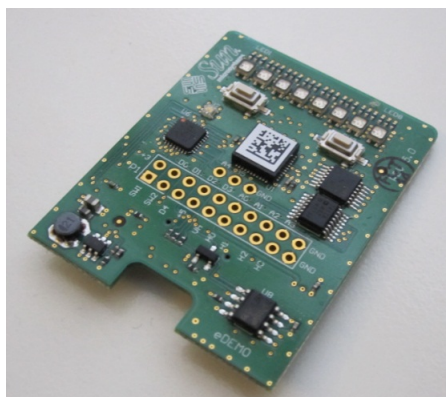
Et applikationskort udvider funktionerne som en Sun SPOT tilbyder. Som udgangspunkt er Sun SPOTs udstyret med et eDEMO-kort. Kortet er forbundet til bundkortet, hvilket gør bundkortet i stand til at behandle data opfanget af applikationskortet. Herunder nævnes de væsentligste specifikationer for eDEMO-kortet:

- Sensorer: 2G/6G 3D accelerometer, RGB-lyssensor og temperaturmåler
- 8 RGB LED'er
- 2 trykknapper
- 6 analoge input, 5 digitale I/O og 4 digitale output porte

Et Sun SPOT-sæt består af to Sun SPOTs og en såkaldt basestation. Basestationen er en simplificering af en Sun SPOT. Denne indeholder intet applikationskort eller batteri, hvorfor den er væsentligt mindre, og skal anvendes i forbindelse med en værtsenhed som for eksempel en computer.



Figur 2.1: Sun SPOT og dets bundkort



Figur 2.2: Applikationskortet

2.2 Software

På bundkortet forefindes Squawk Java Virtual Machine (forkortet JVM). JVM'en kører direkte på processoren uden et styresystem og gør det muligt at afvikle Java på enheden. JVM'en afvikler koden direkte fra flashhukommelsen, der sidder på bundkortet.

Alle enhedsdriverne på Sun SPOT'en er programmeret i Java, dette er bemærkelsesværdigt, da Java normalt er kendt for at være uafhængig af hardwaren. Derudover er der som udgangspunkt kun et begrænset antal Java-biblioteker på enhederne.

Koden, Sun SPOT'en afvikler, udvikles som udgangspunkt på en ekstern enhed, for eksempel en computer. I forbindelse med eksporteringen af softwaren til Sun SPOT'en, kan enten USB- eller radiokommunikation anvendes. USB-kommunikationen kan også anvendes til at overføre generelle data og beskeder fra Sun SPOT'en til en ekstern enhed. Ved brug af radiokommunikation anvendes Over-The-Air-teknologien (OTA), hvorigennem kommunikationen foregår trådløst mellem Sun SPOT'en og basestationen, der er tilsluttet computeren.

2.3 Version

I løbet af projektet har jeg anvendt følgende versioner af henholdsvis hardwaren og softwaren på samtlige udleverede Sun SPOTs:

Hardware: revision 6

Software: version 6, yellow-101117-1

KAPITEL 3

Kravanalyse og specifikation

Sun SPOTs'ene bruges til at varetage overvågningen af en pakke-transport. Før de sendes med en pakke, har brugeren mulighed for at administrere dem. Herigennem er det muligt at opsamle målinger. Jeg vil i dette kapitel se på de generelle krav til implementeringen. På baggrund af en analyse af forskellige krav, vælges de egentlige specifikationer for systemet.

Systemet kan som udgangspunkt befinde sig i to stadier, hvor Sun SPOTs'ene enten overvåger eller bliver administreret af en ekstern enhed. Kapitlet vil være opdelt efter disse to stadier.

3.1 Overvågning

I forbindelse med overvågningen skal forskellige overvågningsbegivenheder specificeres. Herunder analyseres kravene for henholdsvis måleforløbet, målinger, overvågning med flere Sun SPOTs og sikkerheden.

Måleforløb

Overvågningen er den periode, hvor en eller flere Sun SPOTs aktivt deltager i målingen af værdier under en pakke-transport. Det egentlige måleforløb startes igennem brugerens administrering af enhederne. Da en pakke sjældent sendes med posten direkte efter endt administrering, skal brugeren kunne vælge at udsætte starten på måleforløbet.

Et transportforløb udgøres af en transport af en pakke, der sendes fra A til B. Da afsenderen af pakken, kun har relationer til A og B, og ikke til transporten herimellem, er det kun i A og B en administrering af enhederne skal være mulig. Muligheden for at pause eller stoppe overvågningen under transporten er altså ikke til stede.

Sammenfattende stilles følgende krav til måleforløbet:

- Måleforløbet startes af brugeren med mulighed for at udsætte måleforløbet
- Først ved pakkens ankomst, er det muligt for brugeren igen at administrere enhederne
- Efter endt transportforløb kræves det, at brugeren fysisk besidder enheden for at administreringen kan påbegyndes

Målinger

Til at måle værdier bruges accelerometeret, lys- og temperaturmåleren på Sun SPOT'en. Tidspunktet en måling skal foretages, kan være bestemt af et tidsinterval eller ved overskridelse af en grænseværdi. I tidsintervaller kan Sun SPOT'en blot vågne op, tage en måling, gemme den og lukke ned igen. Til forskel kræver brugen af grænseværdier, at man ofte tager stikprøver til at sammenligne med grænseværdien. Denne proces er væsentligt mere krævende.

Både accelerationer og lysniveauændringer kan komme som hurtige udsving. Det er derfor nærliggende, at disse måles ved overskridelse af grænseværdier. Grænseværdien skal bidrage til, at ubrugelige målinger ikke anvendes. Der ønskes desuden at give brugeren et helhedsindtryk af transportforløbet. Derfor måles lysniveau og temperatur altid i faste intervaller. De målte værdier kan gemmes i et flygtigt eller ikke-flygtigt lager, hvorved data henholdsvis mistes eller ikke mistes, hvis enheden lukker ned. Da det tilstræbes at sikre al data, selvom enheden får problemer under transporten, vælges det ikke-flygtige lager.

Da det som udgangspunkt er meningen, at overvågningen skal give et helhedsindtryk af transporten, måles der kontinuerligt. Modsat kunne periodisk overvågning have været anvendt, således at kun specifikke dele af turen overvåges.

Sammenfattende stilles følgende krav til målingerne:

- Accelerationer og lysniveauer måles ved overskridelse af grænseværdier
- Lysniveauer og temperaturer måles ved faste intervaller
- Der måles kontinuerligt under hele transporten
- Målinger gemmes i et ikke-flygtigt lager

Overvågning med flere Sun SPOTs

Ved at forøge antallet af Sun SPOTs per pakke, ønskes kvaliteten og kvantiteten af overvågningen forøget. I et forsøg på at opnå en større målelængde, sammenlignet med brugen af én Sun SPOT, skal enhederne kunne indgå i et netværk og varetage måleforløbet sammen. Kvaliteten af overvågningen kan eksempelvis forbedres ved at øge antallet af samtidig overvågende Sun SPOTs eller lade enhederne overvåge hinanden. For at overvåge én pakke påkræves én Sun SPOT, det skal derfor ikke være muligt at overvåge flere pakker med en enkelt enhed.

Ved ændring af antallet af Sun SPOTs per pakke stilles følgende krav:

- Minimum én Sun SPOT per pakke
- Sun SPOTs'ene skal kunne indgå i en fælles overvågning, hvorved kvaliteten og kvantiteten af overvågningen skal forøges
- Antallet af overvågende enheder skal være nemt at skalere. Det forventes dog som udgangspunkt, at der anvendes i omegnen af fem Sun SPOTs per pakke
- Ved brug af flere enheder skal målelængden være minimum ét døgn

Sikkerhed

Der er forskellige sikkerhedsaspekter i forbindelse med overvågningen. Da der foregår trådløs kommunikation mellem enhederne, er der potentiale for, at uvedkommende personer lytter med eller ligefrem influerer systemet. Beskederne, der

sendes mellem enhederne, indeholder som udgangspunkt ikke hemmelige informationer, og det vurderes derfor ikke, at det er farligt at sende disse beskeder i klartekst. Til forskel kunne man vælge at kryptere beskederne, hvorved læsningen af beskedens informationer, væsentlig besværliggøres. Det er til gengæld vigtigt at sikre, at andre personer ikke kan have indflydelse på systemet. Dette kan opnås ved at kontrollere hvem en beskeds afsender er, og derved sikre sig at afsenderen er autoriseret. Denne fremgangsmåde udgør kun en mindre sikkerhed.

Desuden er der det fysiske sikkerhedsaspekt. Hvis en uautoriseret person, er i stand til at opnå fysisk adgang til enheden, kan han få direkte adgang til de opsamlede informationer. Dette kunne igen beskyttes ved kryptering, men vurderes dog ikke nødvendigt, da systemets sikkerhed ikke er det primære fokus i dette projekt.

Sammenfattende stilles følgende krav til sikkerheden i forbindelse med overvågningen:

- Sikre at netværkskommunikationen kun foregår mellem autoriserede enheder
- Sende netværksbeskeder i klartekst

3.2 Administrering

Administreringen udgør brugerens interaktion med en eller flere Sun SPOTs. En basestation og en computer skaber værten, som foretager administreringen. Det er herigennem brugeren foretager den egentlige styring, der blandt andet indbefatter opsamling af informationer eller ændring af måleindstillinger. De opsamlede informationer skal repræsenteres for brugeren på en måde, der hurtigt og nemt giver et overblik over transportforløbet. Det skal således være nemt at opdage hændelser, der kunne have beskadiget pakkens indhold. Man kunne således nøjes med kun at repræsentere brugeren for det data, der er mest interessant. Men da der ønskes et overblik over hele transporten, vises alt opsamlet data. Til dette formål bruges en graf til hver af måletyperne. Brugeren har desuden mulighed for at ændre måleindstillinger, som for eksempel hvor ofte der skal måles.

Som ved overvågningen, skal der stadig udvises en vis form for sikkerhed i forbindelse med administreringen. Det skal kun være autoriserede enheder, der kan influere systemet.

Til administreringen stilles følgende krav:

- Administrere opsamling af data fra flere Sun SPOTs og verificere deres korrekthed i forhold til grænseværdier
- Visualisere indsamlet data fra henholdsvis accelerometeret, lys- og temperaturmåler med mulighed for at aflæse eksakte værdier samt måletidspunkter
- Administrere alle Sun SPOTs, der har indgået i en overvågning, samtidig
- Ændre måleindstillinger: måleinterval, tærskelværdier, målestarttiden, måleinstrumenter der anvendes og antallet af målende Sun SPOTs
- Værten skal kunne håndtere udfald af basestationen eller Sun SPOTs'ene
- Sikre at kommunikationen kun foregår mellem autoriserede enheder

Del I

Individuel overvågning

Analyse

I denne del af rapporten beskrives udviklingen af den individuelle overvågning. Før design- og implementeringsfasen kan påbegyndes, analyseres Sun SPOT-teknologien. Det er vigtigt at få afgrænset hvilke af dens funktionaliteter, der kan anvendes til udførslen af kravspecifikationen. Efter analysen af teknologien, analyseres den egentlige problemstilling til overvågning ved brug af én Sun SPOT.

4.1 Sun SPOT

Sun SPOT er en eksperimentel teknologi, hvilket betyder det kan være svært at finde informationer om alle dens funktioner. Teknologien bærer desuden præg af kendte og ukendte fejl. På grund af disse mangler og fejl, analyseres teknologien med henblik på udførsel af kravspecifikationen.

Før den egentlige analyse kan begynde, afgrænses hvilke områder, der kan være relevante at undersøge i henhold til projektet. Emner, der analyseres nærmere i afsnittet:

- Radiokommunikation

- Måleinstrumenter
- Flashhukommelse
- Batteriet
- Den interne klokke

4.1.1 Radiokommunikation

I Sun SPOTs'ene er der integreret en radio baseret på IEEE 802.15.4 netværksstandarden. Hver Sun SPOT har sin egen unikke mac-adresse, som bruges til at bestemme hvem en besked sendes til.

Der tilbydes to forskellige radioprotokoller til overførsel af beskeder mellem enhederne: *Radiostream* og *Radiogram*.

Radiostream: Tilbyder en sikker forbindelse mellem to enheder. For at oprette forbindelsen skal begge enheder specificere modpartens mac-adresse, samt en fælles port.

Data sendes som en strøm af bytes. Bytes opbygges i en buffer og bliver først transmitteret når det vurderes nødvendigt. Brugeren oplever således ingen størrelsesbegrænsning på det data, der skal sendes, da protokollen blot sørger for at sende når det er nødvendigt. Brugeren kan desuden tvinge protokollen til at sende data hvis det ønskes

Radiogram: Til opsætning af en forbindelse mellem to enheder, skal den ene Sun SPOT kende modpartens mac-adresse, mens den anden enhed blot behøver at kende den fælles port.

Beskeden, der sendes mellem enhederne, kaldes for *Radiogram*, og udgøres af en pakke indeholdende informationer. Hver pakke har på forhånd en kendt størrelsesbegrænsning. Hvert *Radiogram* indeholder desuden et adressefelt, som beskriver henholdsvis modtagerens og afsenderens mac-adresse. Hos afsenderen defineres adressen, som pakken skal sendes til, mens den hos modtageren fortæller hvem afsenderen er. *Radiogram* kan desuden ikke garantere beskeders fremkomst i samme grad som *Radiostream*

I både *Radiogram* og *Radiostream* er der indbygget rutning. Det vil sige, at beskeder kan videresendes igennem enheder, hvorved sandsynligheden for fremkomst hos modtageren forbedres væsentligt. Det er muligt at specificere hvor mange enheder en pakke må have været transmitteret igennem, før den ankommer hos modtageren. Desuden er det indbygget i netværkslaget, at når en besked

sendes, skal afsenderen modtage en kvittering (engelsk: acknowledgement) fra modtageren, når beskeden er kommet frem.

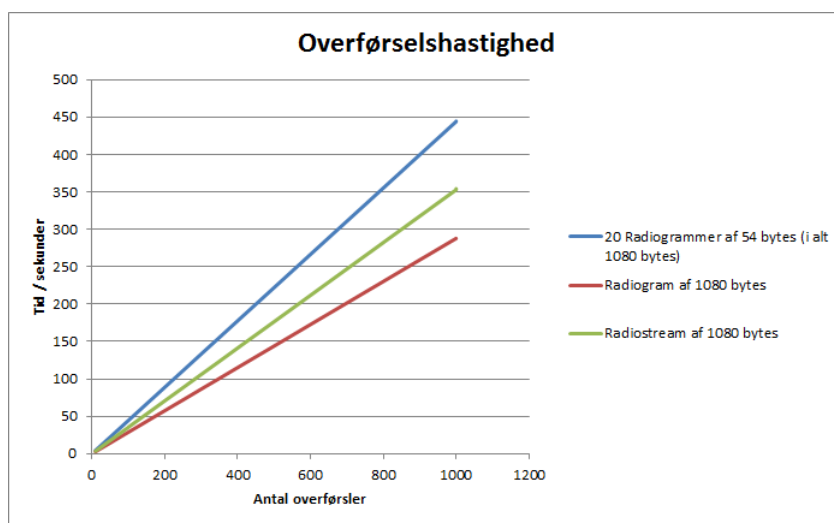
Radiogrammer og *Radiostreams* har en række felter, der kan indeholde informationer. Et udsnit af nogle af felterne ses på figur 4.1.

Radiogram/ Radiostream
Integer
Boolean
Double
UTF (String)
...

Figur 4.1: Struktur for et *Radiogram* eller en *Radiostream*

Radiogram tilbyder desuden at udsende en besked, som ikke kræver en kvittering. Denne funktion kaldes broadcasting, og udsender en besked til alle enheder på en bestemt port.

Udover de ovennævnte forskelle på *Radiostream* og *Radiogram*, er der også en væsentlig forskel på overførselshastigheden. Som det fremgår af figur 4.2 kan det som udgangspunkt bedst betale sig, at bruge *Radiostream*. Grunden til dette er



Figur 4.2: Overførselshastigheder ved brug af *Radiostream* og *Radiogram*

blandt andet, at der er en masse overhead på hver *Radiogram* i forhold til *Radiostream*. Til gengæld kan man spare dette overhead væk ved at pakke mere information per *Radiogram*, og derved kan der overføres større mængder hurtigere end ved *Radiostream*. Fremgangsmåden for overførselshastighedsanalysen fremgår af bilag B.2.

I bilag B.1 er der vedlagt resultater for test af signalstyrken. Det konkluderes ud fra testene, at de forskellige positioner Sun SPOTs'ene kan antage, når de afsender beskeder, ikke har nogen betydning for styrken en besked modtages med. Under mine tests, oplevede jeg til tider, at der var problemer med at afsende beskeder, hvis flere Sun SPOTs forsøgte at sende beskeder samtidig. Beskeden fremkom ikke hos modtageren, og den afsendende enhed opfangede heller ikke, at beskeden gik tabt. Dette kan forebygges nogenlunde, ved at tilføje en tilfældig forsinkelse på nogle millisekunder, før afsendelsen. I praksis opstår problemet kun i forbindelse med broadcasting. Ved normal udsendelse af beskeder, sendes beskederne én ad gangen, hvorved der ikke opstår interferens mellem beskederne. På trods af dette forbehold, virkede de to protokoller ganske pålidelige, og tages derfor begge i betragtning, når kommunikationsvalget skal tages.

4.1.2 Måleinstrumenter

Sun SPOT'ens applikationskort er som udgangspunkt udstyret med tre måleinstrumenter, henholdsvis et accelerometer, en lyssensor og en temperaturmåler. Det er vigtigt for projektet, at få afklaret hvilke begrænsninger disse instrumenter har, for at kunne klarlægge i hvilket omfang de kan inddrages i det færdige produkt.

4.1.2.1 Accelerometer

Accelerometeret, på applikationskortet, hedder LIS3L02AQ og er i stand til at måle accelerationer på henholdsvis X, Y og Z-aksen. Accelerationen måles enten på en ± 2 g eller en ± 6 g skala. På trods af en teoretisk maksimal acceleration på ± 6 g, har jeg ved flere lejligheder, i mit projektførløb, målt cirka $\pm 7,5$ g.

På grund af jordens tyngdekraft, vil Sun SPOT'en, hvis den ligger på en plan flade, måle accelerationer på Z-aksen på $+1$ g mens den på X og Y-aksen er 0 g. Ønsker man at undgå tyngdepåvirkningen, kan man få accelerometeret til at returnere den nuværende acceleration relativ til en tidligere måling. Derudover kan accelerometeret også returnere Sun SPOT'ens hældning på hver af de tre akser.

Under min analyse af accelerometeret stod det hurtigt klart, at dets præcision er utrolig ringe. Selvom det placeres på en helt stille flade, måles accelerationer på cirka $\pm 0,1$ g. Man skal altså være meget påpasselig ved anvendelse af accelerometeret, hvis man ønsker præcise data. På trods af dens upræcighed, kan accelerometeret fint anvendes til, at opfange mere kraftige bevægelser med større g-påvirkning.

I forbindelse med pakkeovervågning, er accelerationer på cirka $\pm 7,5$ g næppe nok, til at ødelægge indholdet af en pakke. Det bliver derfor svært til at bruge accelerometeret til direkte indikere om en genstand kunne være gået i stykker under transporten.

4.1.2.2 Temperaturmåler

Applikationskortet er udstyret med en temperaturmåler. Temperaturmåleren er en del af den analoge til digitale konverteringsmekanisme, der sidder på applikationskortet [5]. Dette bevirker, at den temperatur Sun SPOT'en måler, stammer fra luften inde i enheden i stedet for den fra omgivelserne. Temperaturen inde i Sun SPOT'en er væsentligt højere hvis enheden er tilsluttet USB eller har lavet beregninger i længere tid. Dette skyldes at de to handlinger afgiver varme. Ønsker man at få en idé om hvad temperaturen er udenom Sun SPOT'en, bør man derfor tage målinger efter enheden har været inaktiv et stykke tid.

Da det skal være muligt at måle temperaturer, der giver en indikation af omgivelserne omkring Sun SPOT'en, når enheden er aktiv, er det derfor nødvendigt at kalibrere temperaturmåleren. For at finde en kalibreringsfaktor for hver Sun SPOT, har jeg udført tests af alle Sun SPOTs'ene ved fem forskellige temperaturer. Hver Sun SPOT lå minimum 45 minutter i hvert miljø, og på baggrund af de sidste 15 minutters målinger, blev et gennemsnit udregnet. Gennemsnittet for de fem miljøer sammenholdt med en kontrolenheds temperaturmåling, ligger til grund for kalibreringsfaktoren for den enkelte Sun SPOT. Det kan desuden diskuteres hvorvidt kontrolenheden har målt mere præcist, men de temperaturer den målte virkede mest troværdige. Resultatet ses i figur 4.3. Derudover findes

Enhed	Måling 1	Måling 2	Måling 3	Måling 4	Måling 5
Kontrol	-16	5	9	10	18
1CFE	-10	8	15	12	23
2FB1	-9	9	16	16	23
2078	-11	9	16	15	24
3831	-8	8	14	14	22

Figur 4.3: Temperaturmålinger i forskellige miljøer. Alle værdier er i $^{\circ}\text{C}$

temperaturmålinger sammelignet med en korrigeringsfaktor i bilag C. Som det fremgår af bilaget, ligger de nye temperaturmålinger tættere på kontrolenheden, men løsningen er ikke perfekt.

Som med accelerometeret, er man nødt til at være opmærksom på, hvad det er man ønsker at bruge værdierne til. Hvis man har brug for præcise temperaturer, kan Sun SPOT'ens temperaturmåler ikke anbefales. Dog kan den fint anvendes til at give en indikation af temperaturen i det miljø den befinder sig i.

4.1.2.3 Lyssensor

På applikationskortet er også monteret en lyssensor til måling af lysniveauet. Lysniveauet måles i en ukendt enhed som multipliceret med 2, giver en tilnærmelse af belysningsstyrken i lux.

Det er svært at kontrollere om de målte værdier er korrekte, da det ikke har været muligt, at lave sammenligninger. Det er derfor ikke muligt at sige hvor præcis lyssensoren er i stand til at måle. Enheden måler desuden i intervallet 0 - 1500 lux.

Ved målinger indendørs, skal man være opmærksom på, at lys fra elektriske apparater ikke udsendes konstant, hvorfor enheden kan registrere lysniveauet på et tidspunkt, hvor det elektriske apparat ikke udsender lys. Derfor anbefales det at måle lysniveauet over tid.

Lyssensoren er i stand til at give en fin indikation af det lysniveau, der omgiver den og har således intet problem med at registrere hvis det er bælghmørkt, eller hvis solen står direkte ind på den. Den kan derfor fint inddrages i pakkeovervågningen, til at indikere om en kasse er åben eller lukket.

4.1.2.4 Generelt

Hvis man skal foretage mange målinger, men reelt kun anvender en brøkdelen af dem, fordi de andre ikke er relevante, konsumerer man hurtigt en stor mængde strøm. Dette kan undgås ved at bruge en hardware supporteret afbrydelse (engelsk: interrupt), som kun aktiveres når en defineret tærskelværdi (engelsk: threshold) overskrides. Derved behøver processoren ikke at være aktiv for at analysere om en måling har overskredet en vis værdi.

Der defineres henholdsvis en øvre og nedre grænse til lytteren (engelsk: listener)

af tærskelværdier. Når en af disse overskrides aktiveres enheden med mulighed for at aflæse værdien, der overskred grænsen. Alle tre sensorer understøtter tærskelværdislytteren. Jeg har dog ikke haft held med, at få det til at fungere optimalt på lys- og temperaturmåleren. På disse var det kun den øvre grænse der havde indfyldelse. På accelerometeret var der derimod ingen problemer.

4.1.3 Flashhukommelse

Der er 4 MB flashhukommelse på en Sun SPOT. Omkring 524.000 bytes kan anvendes af brugeren. Der er to mekanismer til at tilgå flashhukommelsen: *Record Management Store* og *FlashFile*.

Record Store: En *Record Store* er en listestruktur af bytes. Hvert input i listen har sit eget id. Der kan oprettes flere *Record Stores*, såfremt unikke navne anvendes. Navnet bruges desuden når data skal tilgås. Strukturen af alle *Record Stores* opretholdes altid. Dermed kan data også blive læst, selv hvis enheden har været uden strøm i en periode

FlashFile: Er en lavniveaus mekanisme, der bruges til at tilgå flashhukommelsen. Igennem denne skal brugeren selv tage højde for det fysiske layout af flashhukommelsen. Her skal man således selv definere metoder til allokering og frigørelse af hukommelsen. Sun SPOT tilbyder således biblioteker til for eksempel at allokere, læse og skrive i hukommelsen

Der er altså væsentligt forskel på de to mekanismer. Sun Microsystems anbefaler da også brugen af *Record Store*, da den tilbyder et fint interface til flashhukommelsen, hvorimod *FlashFile* kræver, at brugeren selv designer og implementere dette.

4.1.4 Batteriet

Sun SPOTs får strøm enten fra det tilhørende batteri eller en ekstern strømenhed som for eksempel USB. Da enhederne ofte bliver anvendt uden en ekstern strømenhed, er det vigtigt at identificere hvor lang tid en Sun SPOT er i stand til at køre på batteriet. Batteriets levetid afhænger af brugen af Sun SPOT'en. I Sun SPOT manualen [6] forefindes statiske strømforbrugsscenerier, hvor Sun SPOT'en eksempelvis kun har radioen tændt eller kun bruger applikationskortet. Værdierne er med til at give et fint overblik over hvilke dele af Sun SPOT'en,

der bruger meget eller lidt strøm. Men eftersom at batteriets levetid også afhænger af dens tidligere brug, har det været nødvendigt at undersøge batteriet lidt grundigere. I forbindelse med analysen af batteriet, er det også værd at stifte kendskab med batteri-interfacet `IPowerController`.

4.1.4.1 Implementering og udførelse

For at få et overblik over batteriets levetid har jeg opstillet tre statiske scenarier, der alle kører oftest muligt:

1. Udførelse af udregninger
2. Måling ved brug af henholdsvis accelerometer, lyssensor og temperaturmåler
3. Send og modtagelse af radiobesked

Ved brug af `IPowerController` er det muligt at få et estimat af forskellige relevante værdier for batteriet. Hvert minut gemmes en besked i flashhukommelsen indeholdende informationer om tiden samt spændingen og tilbageværende procent på batteriet.

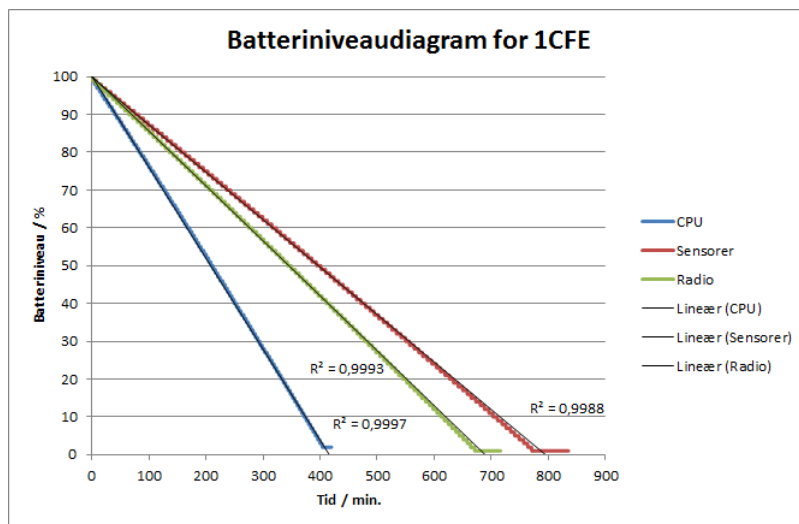
4.1.4.2 Resultat

Forsøget blev udført for alle fire udleverede Sun SPOTs. Et udsnit af resultatet ses på figur 4.4, mens det fulde resultat kan ses i bilag A.2. Det ses, at anvendelsen har stor betydning for batteriets levetid. Batteriet er for eksempel næsten i stand til at holde dobbelt så lang tid ved brug af sensorer fremfor processorkraft.

Sun SPOT	Processor	Sensorer	Radio
1CFE	420	835	716
2FB1	458	740	759
2078	423	780	680
3831	510	1168	734

Figur 4.4: Batterilevetiden ved brug af henholdsvis processor, sensorer eller radio. Alle værdier er i minutter

Udover at kigge på batteriets egentlige levetid, fik jeg også undersøgt anvendelsen af funktionen, som fortæller procentdelen af levetiden. Som det ses af



Figur 4.5: Batteriniveaudiagram for Sun SPOT 1CFE

batteriniveaudiagrammet for én Sun SPOT 4.5, er kurven for batteriniveauet som funktion af tiden tilnærmelsesvis lineær, hvilket betyder, at denne funktion fint kan anvendes til at vide hvornår Sun SPOT'en er ved at løbe tør for strøm. Tendensen er desuden den samme for de andre Sun SPOTs, hvilket fremgår af bilag A.1.

4.1.4.3 Tilstande

En Sun SPOT kan befinde sig i forskellige tilstande, alt afhængig af hvad den foretager sig. Tilstanden er en indirekte indikation af hvor meget strøm der bruges. Der findes tre hierarkiske tilstande, som ved hvert spring nedad i hierarkiet begrænser antallet af anvendte ressourcer på Sun SPOT'en. De tre tilstande kan i grove træk defineres som følgende:

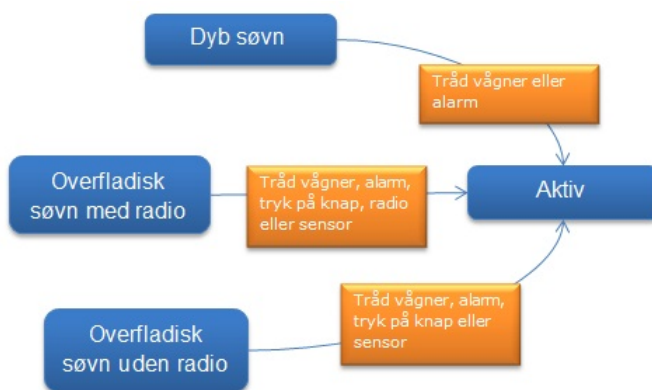
Aktiv: Har mindst en aktiv tråd

Overfladisk søvn: Har ingen aktive tråde, men har mindst én ressource tændt.
En ressource kan her være applikationskortet eller radioen (engelsk: shallow sleep)

Dyb søvn: Har ingen aktive tråde eller nogen ressourcer tændt (engelsk: deep sleep)

Der er en væsentlig forskel i den strømstyrke enheden bruger, alt efter hvilket stadie den befinder sig i. Hvis enheden, for eksempel, er i stand til at gå i dyb søvn, vil den kunne slukke for både processoren og applikationskortet.

For at vække en Sun SPOT fra et givent søvnstadie, kan forskellige fremgangsmåder anvendes alt efter stadiet. På figur 4.6 ses hvordan Sun SPOT'en vækkes fra forskellige søvnstadier.



Figur 4.6: Illustration over hvordan en sovende Sun SPOT vækkes

4.1.4.4 Strømbesparelse

Der kan desuden spares på strømmen ved at justere på Sun SPOT'ens hardwarekomponenter:

- I overfladisk søvn kan master clock'ens frekvens nedbringes hvorved strømstyrken mindskes. Jævnfør Sun SPOT manualen [6] kan der spares 10,8 mA, hvis frekvensen sænkes fra de oprindelige 60 MHz til 18,432 MHz. Ved at ændre frekvensen øges den typisk latenstid også til 265 ms i stedet for 81 ms. På nuværende tidspunkt understøttes denne funktion dog ikke, da ARM-processoren på Sun SPOTs'ene fejler under udførelsen
- Deaktivering af I/O-muligheder f.eks. USB og knapper
- Ændre sendestyrken for radioen

4.1.5 Den interne klokke

Hver Sun SPOT er udstyret med en intern klokke, der holder styr på, at for eksempel tråde sover den ønskede tid. Derudover bruges den til at fortælle hvilken dato det er eller klokken. Den interne klokke opdateres hver gang et program eksporteres til en Sun SPOT. Den nye tid hentes fra enheden der eksporteres fra.

Det er vigtigt at få afklaret i hvilken grad Sun SPOTs'enes klokker følges ad. Til undersøgelse af dette, eksporteres et program til hver Sun SPOT, der fortæller, igennem USB-kommunikationen, hvad klokken er på den enhed. Værdierne hentes i millisekunder.

For at få Sun SPOTs'ene til at sende en besked med klokken, kan man enten sende en forespørgsel via radio- eller USB-kommunikation. Desuden kan man vælge at bruge knapperne på enheden. Som udgangspunkt er der ingen af fremgangsmåderne, som sikrer, at forespørgslen modtages samtidigt på samtlige Sun SPOTs. Jeg forventer umiddelbart, at USB-kommunikationen er den hurtigste, og anvender derfor denne, i håb om, at den udsender forespørgslerne med kort mellemrum. Da den senere brug af den interne klokke ikke kræver stor nøjagtighed, er afvigelse af mindre grad desuden acceptable.

	Måling 1	Måling 2	Måling 3
12 timer	81,25	77,75	78,25
18 timer	79,75	69,25	69,00
24 timer	83,75	74,25	82,00

Figur 4.7: Standardafvigelser mellem fire Sun SPOTs ved tre på hinanden følgende målinger taget efter henholdsvis 12, 18 og 24 timer. Værdierne er i millisekunder

Som det fremgår af figur 4.7, er standardafvigelserne for alle værdierne i intervallet 69 - 83,75 ms., hvilket er ganske fint. Der tages desuden forbehold for, at USB-kommunikationen kan give en vis usikkerhed. Det noteres, at ingen af værdierne skilte sig nævneværdigt ud.

4.2 Overvågning

I dette afsnit vil jeg analysere problemstillingen omhandlende overvågning ved brug af én Sun SPOT. Da brugeren skal have mulighed for at interagere med Sun

SPOTs'ene på en mere sofistikeret måde, end hvad Sun SPOTs tilbyder igennem I/O-muligheder, er det naturligt, at der udvikles et eksternt program til at varetage dette. Der skal således udvikles henholdsvis et program til Sun SPOTs'ene og værtsenheden. Herunder vil jeg henholdsvis analysere administreringen af Sun SPOTs'ene samt karakteristika ved værts- og Sun SPOT-programmet.

4.2.1 Administrering af Sun SPOTs

For at administrere Sun SPOTs'ene igennem værtsprogrammet, skal der være en form for kommunikationsmulighed mellem de to. Kommunikationen skal henholdsvis bruges til at overføre opsamlet data fra Sun SPOTs'ene og lave handlingsforespørgsler til Sun SPOTs'ene. Værtsprogrammet er den styrende enhed, og det er således herfra forespørgsler udsendes.

På trods af, at systemet på nuværende tidspunkt kun skal understøtte overvågning ved brug af én Sun SPOT, skal det alligevel være muligt, at administrere flere Sun SPOTs ad gangen. Det skal derfor være muligt at interagere med én og flere Sun SPOTs ad gangen.

For at opnå kommunikation mellem værten og Sun SPOTs'ene skal der defineres en fælles protokol for begge enheder. Følgende funktioner skal udgøre administreringen af Sun SPOTs'ene:

- Initialisere Sun SPOTs
- Slette opsamlede informationer
- Starte overvågning
- Frigøre/låse Sun SPOT til en værtsenhed
- Importere informationer
- Ændre overvågningsindstillinger

4.2.2 Vært

Værtsprogrammet skal foruden administreringen af Sun SPOTs'ene, også varetage de opsamlede informationer. Ved overførslen fra en Sun SPOT til værten skal målinger verificeres i forhold til grænseværdier, således at ugyldige værdier ikke accepteres.

De overførte målinger skal visualiseres for brugeren for at overskueliggøre disse. Informationerne skal repræsenteres i individuelle grafer. Det vil sige at accelerationer, lysniveauer og temperaturer fremvises i hver sin graf. Udover den målte værdi skal tidspunktet for målingen også være repræsenteret. I grafen skal det desuden være muligt at aflæse eksakte værdier for hver enkelt måling. De opsamlede værdier skal kunne gemmes lokalt med henblik på senere genvisning.

4.2.3 Sun SPOT

Sun SPOT'ens måleinstrumenter bruges til at registrere henholdsvis accelerationer, lysniveauer og temperaturer. For at give et overblik over transportforløbet registreres lysniveau og temperatur i intervaller. Ved mindre intervaller tjekkes det desuden om lysniveauet har overskredet en tærskelværdi, og disse værdier registreres også. Derudover registreres accelerationer når de overskrider en tærskelværdi.

Opsamlede værdier gemmes på enheden med henblik på senere, at overføre dem til værtsenheden. Det er derfor vigtigt, at de lagres et ikke-flygtigt sted således, at de ikke forsvinder hvis Sun SPOT'en lukker ned. Udover den målte værdi skal tidspunktet for målingen også registreres.

En Sun SPOT skal besidde en vis form for sikkerhed med hensyn til kommunikationen til værten. Det må således ikke være muligt, at en Sun SPOT begynder at behandle forespørgsler fra andre værter, som derved potentielt kan slette informationer.

Jeg har nu defineret både kravene til systemet samt analyseret teknologien og problemstillingen. Ud fra denne viden kan systemets design nu beskrives. Kapitlet er opdelt i et afsnit om kommunikationen mellem Sun SPOTs'ene og værten og to afsnit omhandlende henholdsvis designet af værts- og Sun SPOT-programmet.

5.1 Kommunikation

Jeg vil i dette afsnit beskrive den generelle struktur, der ligger til grund for kommunikationen mellem værts- og Sun SPOT-programmet. Sun SPOT tilbyder to muligheder til at foretage kommunikation: USB eller radio. Ved anvendelse af én Sun SPOT er fordelene ved brug af USB mange. Her opnås blandt andet en større sikkerhed, i og med der kun kan interageres med enheden, hvis man fysisk besidder den.

Når systemet senere skal udvides er trådløs kommunikation nødvendig. I håb om at kunne genbruge en stor del af kommunikationsmetoderne, vælges det derfor, at al kommunikation foregår via radioen. Jeg vælger at bruge *Radiogram* til at varetage kommunikationen, da jeg har brug for broadcast-funktionen som

Radiostream ikke besidder. Derudover er behovet for en buffer ved afsending af beskeder ikke nødvendig, og desuden kan store beskeder sendes hurtigere med *Radiogrammer*.

Som skrevet i analysedelen, består et *Radiogram* af en række felter, som frit kan anvendes til at indeholde informationer. Herunder beskrives de felter, der anvendes i mit kommunikationsdesign:

Integer-feltet: Bruges til at identificere beskedens type. Det er altså i dette felt, beskedens formål er beskrevet. Feltet læses af protokollen for at bestemme hvad der skal udføres. Hvis beskeden indeholder informationer, kan dette også verificeres når beskedens formål er kendt

UTF-feltet: Anvendes i forbindelse med informationsbeskeder, og kan for eksempel indeholde informationer om temperaturmålinger

Det kommunikationsdesign, der varetager interaktionen mellem værten og Sun SPOTs'ene er desuden kortluntet. Hvis en af enhederne er ude af stand til at sende en pakke til modparten antages det, at denne ikke findes, og en ny forbindelse skal oprettes, for at de to på ny kan kommunikere. Til at bestemme om en modpart findes, anvendes det underliggende netværkslag i Sun SPOT'en. Det vil sige, at der for hver sendt besked, forventes en kvittering fra modtageren. Modtages denne kvittering ikke, antages det, at modparten ikke længere findes.

Beskederne mellem enhederne kan som udgangspunkt inddeles i tre kategorier:

- Broadcast-beskeder
- Kommandobeskeder
- Informationsbeskeder

5.1.1 Broadcast-beskeder

Som en indledende fase i kommunikationen, mellem værten og Sun SPOTs'ene, er der behov for, at de kender hinandens mac-adresse. Der skal altså opsættes referencer mellem værten og Sun SPOT'en og vice versa. Referencen opnås ved værtens udsendelse af en broadcast-besked. Som skrevet i analysedelen, forventer operationen ingen kvittering, og er derfor ideel til dette formål.

Hvis en Sun SPOT opfanger broadcast-beskeden sendes der et svar til værten og derved er der skabt en forbindelse mellem de to. Mac-adresserne opfanges fra de modtagne pakker.

5.1.2 Kommandobeskeder

I min netværksstruktur er det kun værten, der er en egentlig aktør. Sun SPOT'en svarer eller udfører handlinger på baggrund af de beskeder værten sender, og tager aldrig initiativ til selv at udføre opgaver.

Med kommandobeskeder forstås direkte kommandoer som Sun SPOT'en skal foretage handlinger ud fra. Blandt de vigtigste beskeder kan blandt andet nævnes:

- Slet hukommelse
- Start overvågning
- Frigør Sun SPOT
- Importer informationer

5.1.3 Informationsbeskeder

Informationsbeskeder afsendes både af værten og Sun SPOTs'ene. Disse beskeder indeholder en eller anden form for information til modparten. Denne type besked bruges til følgende:

- Overførsel af opsamlet informationer
- Ændre måleindstillinger på Sun SPOT'en

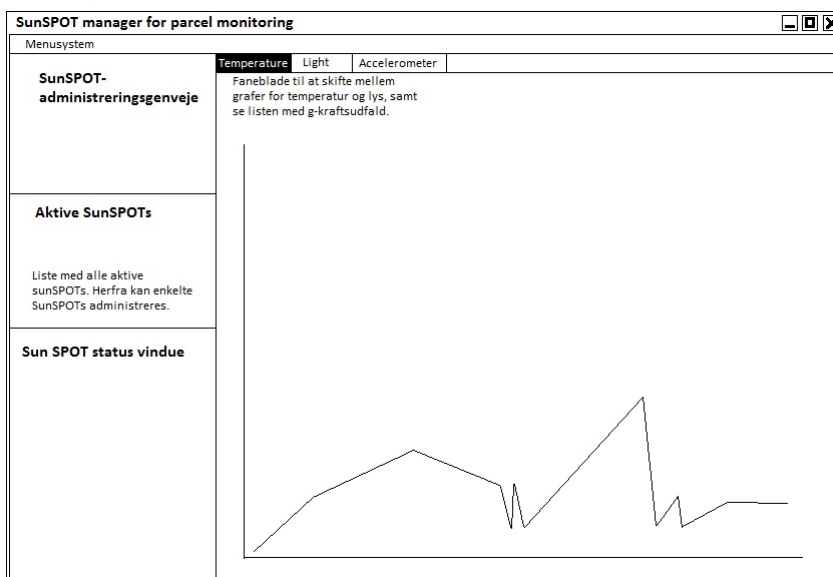
5.2 Vært

Værtsprogrammet er den del af systemet, der skal gøre brugeren i stand til, nemt og intuitivt at administrere Sun SPOTs'ene. Derudover skal det også bidrage til at give et hurtigt overblik over de indsamlede informationer.

Værtsprogrammet består af en grafisk brugerflade og er opbygget efter softwarearkitekturen Model-View-Control (MVC) [7] med brug af Javas observer. Da MVC-arkitekturen er det gennemgående design i værtsprogrammet, er det også oplagt at bruge dens naturlige afgrænsninger til at beskrive det egentlige design. Et klassediagram over programmet findes i bilag F.11.

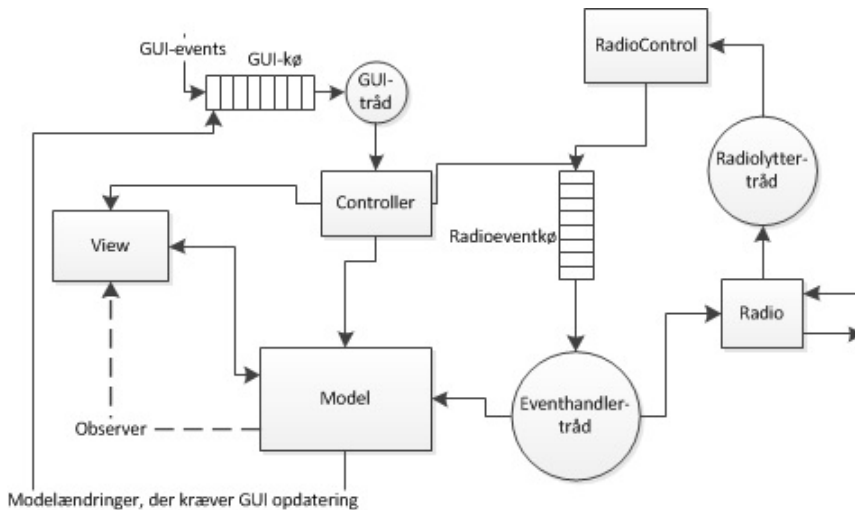
5.2.1 View

View-delen består af en grafisk brugergrænseflade (engelsk: graphical user interface, forkortet GUI), der hjælper til med at anskueliggøre brugerens muligheder, i forbindelse med interaktion med Sun SPOTs'ene samt visualisering af indsamlet data. View'et foretager ingen logiske operationer, men opdateres blot i forbindelse med ændringer i modellen eller controller'en. En skitse af brugerfladen ses på figur 5.1.



Figur 5.1: Skitse af brugerfladen

Da Java Swing anvendes til at lave brugerfladen, skal der tages højde for, at dens komponenter ikke er trådsikre. Det er derfor kun GUI-tråden, der må foretage de egentlige opdateringer af view'et. Derfor placeres en event i den interne GUI-kø, hver gang modellen opdateres, så det kun er GUI-tråden, der opdaterer view'et. Dette fremgår desuden af figur 5.2.



Figur 5.2: Flowchart over værtsprogrammet

5.2.1.1 Interaktion

I GUI'en repræsenteres Sun SPOTs'ene ved en liste af mac-adresser. Ved at vælge en individuel adresse, opnås mulighed for at interagere med netop denne Sun SPOT. Her er det således muligt for eksempel at ændre måleindstillinger eller slette data på den enkelte Sun SPOT. Over listen med aktive Sun SPOTs forefindes de samme interaktionsmuligheder. Disse interagerer dog med samtlige Sun SPOTs på én gang.

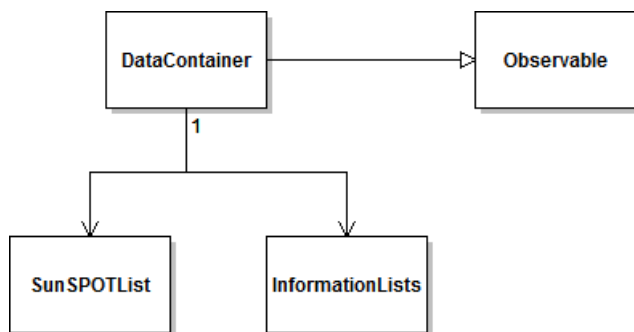
5.2.1.2 Informationsvisualisering

Til visualisering af importeret data anvendes det gratis tilgængelige JFreeChart [4]. Alt efter hvilke typer målinger der er foretaget, aktiveres de dertilhørende grafer. I grafen forefindes funktioner som zoom og aflæsning af eksakte værdier.

5.2.2 Model

Modellen i programmet repræsenterer det data, der ligger til grund for det, som ses i view-delen. Modellen består af en hovedklasse, der indeholder Sun SPOTs'ene og indsamlede informationer. Idet alle datarelaterede ændringer går

igennem denne hovedklasse, kan alle opdateringer gå herfra til view-delen. Som tidligere nævnt sker dette ved at placere en event i GUI-køen. Når eventen afvikles gør den brug af Java's `observer`-klasse, som kalder en opdatering af view'et.



Figur 5.3: Klassediagram over modellen

5.2.2.1 Sun SPOT

Sun SPOTs'ene repræsenteres som objekter. Objektet er en tilnærmelsesvis afspjeling af den fysiske Sun SPOT. Den er således repræsenteret ved dens aktuelle måleindstillinger, adresse og generelle status, som fortæller om enheden sender data eller er klar til at starte en ny overvågning.

Listen med Sun SPOTs'ene er desuden et kritisk område, da flere tråde samtidig kan tilgå denne. Dette skal der naturligvis tages højde for. Som udgangspunkt anvendtes et `HashMap` til håndtering af Sun SPOTs'ene, idet de er indekseret efter deres mac-adresse og derved nemt kan slås op. For at tage hånd om det parallelle problem, anvendes i stedet `ConcurrentHashMap`, der er sikkert ved brug af flere tråde.

5.2.3 Events

Events, der påvirker programmet, kan enten være modtagne beskeder fra radioen eller komme fra GUI'en. Events fra GUI'en opstår via brugerens interaktion med programmet. Der skildres her mellem to typer af events; de der kan afvikles lokalt og de der kræver kommunikation med Sun SPOTs'ene.

Alle GUI-events placeres som udgangspunkt i den interne GUI-kø. De events, der kan afvikles lokalt, bliver afviklet løbende af GUI-eventhandleren. Events, der kræver kommunikation, tager ofte længere tid at afvikle, da radiotransmissionen udgør en flaskehals. Hvis disse afvikles fra GUI-køen medfører det, at brugeren ikke kan anvende programmet i perioden hvor radioen anvendes. For at undgå dette problem udvikles en ny kø til events, der kræver kommunikation. Disse events flyttes fra GUI-køen til den nye kø, før de afvikles. Denne kø kaldes for radioeventkø og fremgår også af figur 5.2. Køen består af en hægtet liste, som sikrer at events afvikles i en retfærdig rækkefølge. Til behandling af events, udvikles en tilhørende eventhandler, som afvikler events løbende i en separat tråd. Da tråden til afvikling af events ikke skal køre hele tiden, anvendes en semafor indeholdende antallet af events, der venter på at blive afviklet. På den måde er tråden kun aktiv når der er events, der kan afvikles. Der skal desuden sikres mod flere trådes samtidig tilgang, derfor anvendes `ConcurrentLinkedQueue`.

Som tidligere nævnt, kan events også komme fra Sun SPOTs'ene i form af netværksbeskeder. Til håndtering af disse udvikles en tråd til tolkning af beskederne, hvorefter de placeres i radioeventkøen. Dermed afvikles både modtagne beskeder og beskeder, der skal sendes, fra radioeventkøen. Dette gøres fordi listen med Sun SPOTs kræver yderligere trådsikring end hvad `ConcurrentHashMap` kan tilbyde. Da radioeventkøen både indeholder indkommende og udgående beskeder, kan der forekomme forsinkelser, men da de fleste operationer ikke tager lang tid, bør der ikke opstå længere ventetider.

I radioeventkøen placeres der med jævne mellemrum, såfremt der er nogle Sun SPOTs, også en ping-besked. Udsendelse af denne besked kontrollerer, at alle Sun SPOTs stadig er kontaktbare. Ping-beskeder sættes i radioeventkøen af sin egen tråd.

Som det fremgår af figur 5.2, kan både GUI- og eventhandlertråden tilgå delte ressourcer i modellen. Det er som udgangspunkt kun listen med informationer, der kan tilgås af begge tråde og i den udstrækning det er nødvendigt, sikres gensidig udelukkelse, ved at låse listen i perioder. Der kan for eksempel opstå parallelitets problemer i forbindelse med den lokale lagring af grafer. Når brugeren ønsker at gemme graferne låses informationslisten. Herefter oprettes et snapshot af listen, og derefter frigives listen igen. Snapshottet udgør således graferne, der efterfølgende lagres.

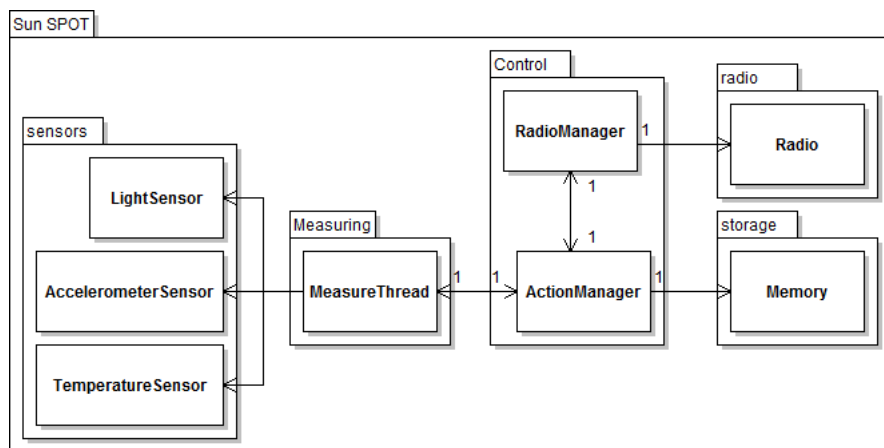
5.2.4 Controller

Controller-delen varetager de events, der kommer fra enten radioen eller GUI'en. Den sørger for, at events foretager de rigtige handlinger i enten view'et, modellen

eller begge. Herunder sikrer den, at events der enten kommer fra radioen eller som skal sendes med radioen, placeres i radioeventkøen.

5.3 Sun SPOT

Sun SPOT-programmet afvikles på Sun SPOT-enhederne. En overordnet opdeling af klasser og pakker ses på figur 5.4. En Sun SPOT kan befinde sig i to stadier hvor den enten måler eller lytter til radioen. Det vil sige, at når Sun SPOT'en foretager målinger, er det ikke muligt at kontakte den via radioen.



Figur 5.4: Klasse- og pakkediagram over Sun SPOT programmet

Til håndtering af modtagne netværksbeskeder anvendes der én tråd til både tolkning og afvikling af beskeder. Som ved afviklingen af beskeder på værten, kan der forekomme forsinkelser, men de bør dog være minimale. Før modtagne beskeder kan afvikles, venter de i den interne radiobuffer.

Sun SPOT'en måler lysniveau og temperatur i intervaller samt lysniveauet når det overskrider tærskelværdien. Accelerationer måles kun via hardwareafbrydelser, når grænseværdien overskrides. Værdierne gemmes i flashhukommelsen på Sun SPOT'en. Jeg har valgt at bruge *Record Store* som interface til hukommelsen. Jeg har taget dette valg, da det nemt og simpelt giver mig adgang til flashhukommelsen, til forskel fra *FlashFile*, hvor man selv skal implementere sin egen hukommelsesmekanisme. Hver Sun SPOT har tre *Record Stores* til målinger. Der er én *Store* til henholdsvis accelerationer, lysniveauer og tempera-

turer. Udover værdien, gemmes den korresponderende måletid også. Derudover er der én *Record Store* hvori adressen på basestationen, som Sun SPOT'en er tilknyttet, ligger. Før den egentlige overvågning kan påbegyndes, sikres det altid, at *Record Stores*'ene, der skal indeholde målinger, er tømte.

5.3.1 Strømbesparelse

En del af problematikken ved kun at bruge én Sun SPOT til at varetage overvågningen er, at den har en begrænset levetid. På trods af denne begrænsning skal det stadig være muligt at få brugbar viden om pakketransporten. Ved at give brugeren mulighed for at ændre forskellige måleindstillinger kan levetiden forøges. Det er således muligt at justere på følgende måleindstillinger:

- Hvilke værdier, der ønskes registreret (temperatur, lysniveau eller acceleration)
- Tærskelværdier (lysniveau og acceleration)
- Interval mellem målinger (temperatur og lysniveau)
- Udsættelse af måling, således at målingerne først starter efter et brugerdefineret antal minutter

Hver gang lysniveauet eller temperaturen skal registreres, er processoren nødt til at vågne op for at bede applikationskortet om værdier og derefter gemme disse. Eftersom det er processoren, der bruger mest strøm (jævnfør min batteri analyse [A](#)), kan der spares på batterilevetiden, ved ikke at anvende den mere end højst nødvendigt. Ved at lade brugeren ændre indstillingerne for målingerne, kan der derved spares meget strøm.

Accelerationer kommer som hurtige udsving, det er derfor nødvendigt at kontrollere disse så ofte det er muligt. For ikke at bruge processoren til at analysere målte accelerationer, anvendes i stedet en hardwareafbrydelse, som aktiverer processoren, når en tærskelværdi overskrides.

KAPITEL 6

Implementering og test

Ud fra de betragtninger, der er gjort i designkapitlet, kan den egentlige implementering udføres. I kapitlet bliver de mest interessante dele af implementeringen beskrevet mere detaljeret. Derudover vil emner, som er beskrevet mere dybdegående i designkapitlet, ikke blive nævnt yderligere. Til sidst beskrives henholdsvis fremgangsmåden for testningen og resultaterne.

6.1 Vært

De mere overordnede forudsætninger for kommunikationen med Sun SPOTs'ene fremgår af designkapitlet. Jeg skriver derfor kun om view'et og modellen i denne sektion, da jeg vurderer, at controller og events er tilstrækkeligt afdækket tidligere.

6.1.1 View

Til GUI-implementeringen anvendes Java Swing. Hovedklassen hedder `GUI` og er en `JFrame`. Denne udgøres af et `JPanel`, der repræsenterer det venstre panel

i brugerfladen, hvor brugeren for eksempel kan administrere Sun SPOTs'ene. Derudover består hovedklassen også af et `JTabbedPane` hvori hver af graferne, for de tre måletyper, udgør et faneblad.

Til visning af grafer anvendes, som tidligere nævnt, `JFreeChart`. Accelerationer fremvises som enkeltstående markeringer i en `xy`-graf, da der ikke er nogen sammenhæng mellem målingerne. Lysniveauer og temperaturer fremvises derimod som sammenhængende kurver i `xy`-grafer. `JFreeCharts` grafer nedarver `JPanel`, hvilket gør at de nemt kan indsættes i andre Swing komponenter.

Brugerfladen implementerer desuden Javas `Observer`-interface. Herigennem fås opdateringer fra modellen. Ved hver opdatering undersøger GUI'en, hvilke komponenter, der skal opdateres og udfører derefter den egentlige opdatering.

6.1.2 Model

Modellen udgøres primært af en liste med Sun SPOTs og en liste med målinger. `SunSPOTList`-klassen udgør datastrukturen, der indeholder Sun SPOTs'ene, og udvider (engelsk: extends) desuden `ConcurrentHashMap`. Dette giver mulighed for at bruge de mest normale `HashMap`-metoder. Klassen indeholder desuden også metoder til manipulation af Sun SPOTs.

`InformationLists`-klassen udgør listerne med målinger. Denne implementerer Java interfacet `Serializable`, som gør det muligt at serialisere informationerne med henblik på at lagre disse lokalt. Listerne for accelerationer, lysniveauer og temperaturer består af tre `HashMaps`, der er indekseret efter måletiden og desuden indeholder målingen.

6.2 Sun SPOT

Sun SPOT-programmet består af én tråd, der enten varetager radiokommunikationen eller foretager målinger. Idet brugeren anmoder om, at Sun SPOT'en skal påbegynde sin overvågning, lukkes radioen og tråden fortsætter derefter i en løkke, der udgør overvågningen. Jeg vil ikke uddybe kommunikationen, da dette allerede er gjort i designafsnittet [5.1](#).

Overvågningen sker ud fra klassen `Properties`, hvori de brugerdefinerede måleindstillinger indeholdes. Klassen besidder altså værdierne som for eksempel måleintervaller og hvilke sensorer der skal anvendes.

Der er et gennemgående brug af Singleton mønstret i Sun SPOT-programmet [9]. Denne giver en global tilgang til ressourcer og parametre på Sun SPOT'en. Metoden `getInstance()` returnerer en instans af den ønskede klasse, der for eksempel kunne være bund- eller applikationskortet. Forsøger man at kalde metoden på en klasse, der ikke findes, oprettes den dynamisk. Singleton mønstret bruges således til at få adgang til underliggende enheder. Herunder ses to eksempler:

```
EDemoBoard.getInstance().getAccelerometer()  
Spot.getInstance().getSleepManager().enableDeepSleep()
```

Som det ses i eksemplet, kan der opnås adgang til accelerometeret og dyb søvn kan slås til, igennem Singleton. I programmet anvendes Singleton kun til hente referencer, som derefter kan genbruges.

6.3 Test

Herunder testes henholdsvis værts- og Sun SPOT-programmet. Der udføres primært funktionelle tests til verificering af kommunikationen mellem enhederne og de underliggende handlinger. På grund af den generelle klassestruktur, vurderes det ikke, at der er behov for at unit-teste.

Sun SPOT-miljøet tilbyder som udgangspunkt to fremgangsmåder til at teste Sun SPOT-programmet: Over-The-Air (OTA) eller print debugging.

OTA debugging: Giver mulighed for, igennem ens udviklingsmiljø, at tilføje en debugger til Sun SPOT'en. Basestationen bruges til at kommunikere med Sun SPOT'en, og det er her den egentligt overførsel af debugginginformationer findes. Debuggeren giver indblik i henholdsvis variable, tråde, kaldestakken (engelsk: call stack) og sessions

Print debugging: Gør det muligt at transmittere print-statements i ens Java-kode til ens udviklingsmiljø. Transmissionen sker enten trådløst via basestationen eller blot igennem USB. Udskriften fra hver Sun SPOT har sit eget vindue i udviklingsmiljøet, hvilket kan gøre det en smule uoverskueligt at holde styr på samtlige enheder

Til udførelse af de funktionelle tests, har jeg valgt at bruge print debugging via USB på Sun SPOTs'ene. Dette gøres, fordi print debugging er i stand til at

give mig de fornødne informationer i forbindelse med testene og desuden er væsentligt nemmere, at tilgå i forhold til OTA debugging. Da der kun kører en tråd på Sun SPOT'en, forventes det desuden, at konsoludskrifter kommer i kronologisk rækkefølge.

6.3.1 Generelle tests

Under udførslen af testene anvendtes op til fire af de udleverede Sun SPOTs. Samtlige tests kan ses i bilag D. Antallet af anvendte Sun SPOTs i de forskellige tests fremgår desuden i bilaget. Der testes for kommunikationen mellem enhederne og den tilhørende fejlhåndtering. Begge enheder kontrollerer som udgangspunkt beskeders oprigtighed før de anvendes. Da alle fejl må antages at stamme fra den kontrollerende enhed, værten, skal fejlmeddelelser også fremkomme her. Overordnet testes for følgende:

- At der kan skabes referencer mellem værten og Sun SPOTs'ene, med henblik på videre kommunikation. Det ønskes verificeret, at Sun SPOTs'ene kun kan være tilknyttet én basestation
- Ændring af måleindstillinger, med henblik på, at disse overføres og anvendes af Sun SPOTs'ene og at invalide værdier hverken godkendes af værten eller Sun SPOTs'ene
- Påbegyndelse af overvågning forløber som forventet. Herunder om Sun SPOT'en kontrollerer, at den er klar, før den starter
- Importering af målinger fra Sun SPOTs. Opfører GUI'en sig som forventet?

6.3.1.1 Testresultat

Generelt er resultaterne for testene som forventet. Kommunikationen mellem én og flere Sun SPOTs er som udgangspunkt ganske simpel. De egentlige problemer fremkommer først hvis beskeder ikke dukker op. Dette tager den underliggende netværksprotokol sig dog af, og sikrer, at der bliver gjort opmærksom på, hvis en besked ikke når frem. Sker dette mister værten referencen til Sun SPOT'en, og de to er nødt til at starte kommunikationen forfra. Der er altså ingen fejlretning hvad angår tab af beskeder. Dette er dog ikke katastrofalt, da værtsprogrammet altid administreres af en bruger, der kan observere fejlene som de opstår.

Derudover observeres det også i test 3 ved ændring af måleindstillinger [D.2.3](#), at Sun SPOT'en også selv verificerer beskeder før de anvendes. Desuden kontrollerer enheden også ting som, at flashhukommelsen er tom og om det er den rigtige basestation, der kontakter den. Dette fremgår af test 2 for start overvågning [D.3.2](#) og test 2 for initialisering [D.1.2](#).

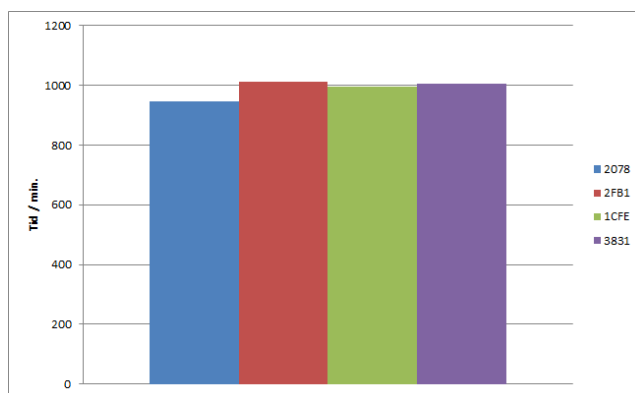
6.3.2 Felt-tests

Udover de funktionelle tests, er også større felt-tests blevet udført. Testene er primært blevet udført ved at placere en eller flere overvågende Sun SPOTs i en rygsæk, for derved at opnå en tilnærmelse af transporten i en pakke. Felt-testene har til formål at teste scenarier, hvor der er andre parametre, som har indflydelse på Sun SPOT'en.

Det er umiddelbart svært at verificere felt-testene, da disse ikke direkte giver en indikation om noget er gået galt. Målingerne kan dog bruges til at give et overblik over forløbet. Her kontrolleres det for eksempel om målingerne har de rigtige intervaller, og om det er muligt at hente alle informationer ud af flashhukommelsen. Desuden bruges felt-testene også til at give en indikation af hvor lang tid en Sun SPOT kan køre på batteriet. Før udførelsen af felt-testene var alle batterier fuldt opladte.

6.3.2.1 Testresultat

Som udgangspunkt forløb alt som forventet. Måleintervallerne for henholdsvis



Figur 6.1: Overvågningstider for de fire Sun SPOTs

lysniveau og temperatur er som forventet et fint omslag omkring det brugerspecificerede måleinterval.

På figur 6.1 ses en felt-test af de fire Sun SPOTs. Alle Sun SPOTs'ene anvendte standardmåleindstillingerne. Sammenholdt med den tidligere analyse af batteriet, virker de fundne måletider ganske troværdige.

Del II

Kollektiv overvågning

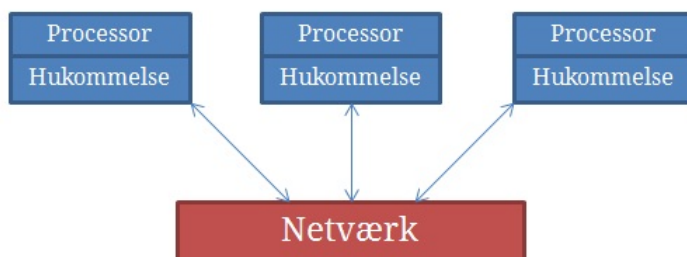
Analyse og design

Overvågning ved brug af én Sun SPOT er implementeret og virker tilfredsstillende. De tidligere implementerede metoder udgør de basale overvågningsfunktioner, som bruges i videreudviklingen af systemet til kollektiv overvågning. Denne del af rapporten omhandler videreudviklingen af systemet.

Først i kapitlet beskrives lidt grundlæggende baggrundsviden om distribuerede systemer, da denne viden bruges under udviklingen af systemet. Herefter beskrives designudvidelsen af værten kort. Desuden analyseres problemstillingen ved udvidelsen af systemet på Sun SPOTs'ene. Dernæst analyseres de forskellige designs, der kan anvendes til at varetage netværkskommunikationen mellem Sun SPOTs'ene under overvågningen og til sidst beskrives det endelige design.

7.1 Baggrund om distribuerede systemer

Et distribueret system består af flere selvstyrende enheder, der kommunikerer gennem et netværk. Enhederne samarbejder om at løse et fælles problem, og bruger netværkskommunikationen til at formidle informationer. Da enhederne er individer i et større netværk, er de i stand til løse det fælles problem sideløbende [1].



Figur 7.1: Eksempel på et distribueret system

Der er både fordele og ulemper ved distribuerede systemer. Herunder listes de mest generelle:

Fordele	Ulemper
Skalerbarhed: Netværket, der udgør det distribuerede system, er dynamisk. Det er derved muligt nemt at fjerne eller tilføje enheder til netværket	Viden: Der er ingen fælles viden om netværket. Hvis der skal dannes viden om netværket, skabes denne hos den enkelte enhed
Redundans: Flere enheder kan regne på det samme problem, hvorved den generelle robusthed af systemet forøges	Trådløst netværk: Udgør en usikker kommunikation, som kan give problemer i forhold til tab af beskeder og generelle netværksforsinkelser

7.2 Vært

I værtsprogrammet sker der ikke mange udvidelser, da de fleste funktioner allerede er understøttet. En væsentlig ændring, der dog skal nævnes, er import-funktionen. Tidligere har det kun været muligt at importere informationer fra én Sun SPOT ad gangen.

Brugeren har nu mulighed for at specificere hvilke Sun SPOTs det ønskes at importere fra. Når Sun SPOTs'ene har bekræftet, at de er klar til at overføre, kan den egentlige dataoverførsel starte. Det er således muligt at importere data fra Sun SPOTs, der ikke har indgået i den samme overvågning.

Et problem ved dette design er, at muligheden for overførsel af værdier med

samme målingstid er mulig. Grafen understøtter tidsdifferencer ned til ét millisekund. For at undgå problemer ved målinger med samme tider, frasorteres de værdier, der er mindst signifikante ved tidsdubletter. Frasorteringen sker ved følgende kriterier:

Acceleration: Her vælges den numerisk højeste værdi

Lys: Her vælges den højeste værdi

Temperatur: Her vælges den temperatur, der udgør det største udsving i forhold til de tidligere målte værdiers gennemsnit

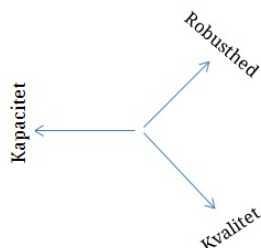
Kriterierne er valgt på baggrund af hvad jeg har vurderet vigtigst for overvågningen. Det er selvfølgelig muligt nemt at ændre disse.

7.3 Sun SPOT

I et forsøg på at forbedre overvågningstiden, robustheden og dækningen af overvågningen, anvendes flere Sun SPOTs til at overvåge. Enhederne skal indgå i et netværk, og overvåge i fællesskab. Overordnet arbejdes der med tre terminologier i forbindelse med den kollektive overvågning:

Robusthed: Robust over for udfald af sensorer eller interne problemer med hardwaren

Kapacitet: Den maksimale tid systemet kan overvåge. Denne er begrænset af henholdsvis batteriet og hukommelsen



Figur 7.2: Udledning af robusthed, kapacitet og kvalitet

Kvalitet: Kvaliteten af målingerne opretholdes ved for eksempel at lade flere Sun SPOTs overvåge samtidig, derved forbedres sandsynligheden for at alle måleudfald opfanges

Som udgangspunkt kan termerne ses som kontraster. Som det også ses af figur 7.2, udligninger de tre hinanden. Ønsker man for eksempel en stor kapacitet, er man nødt til at gå på kompromis med robustheden og kvaliteten af overvågningen. Modsat får man en ringe kapacitet og kvalitet, hvis man udelukkende fokuserer på robustheden af systemet. Der er ikke nødvendigvis ét valg, der er bedre end et andet. Valget skal vurderes i forhold til anvendelsen af systemet. Man er altså nødt til at gøre sig nogle overvejelser om designet, med hensyn til henholdsvis robusthed, kapacitet og kvalitet, før det endelige design kan vælges.

7.3.1 Mulige designs

Som tidligere nævnt, har den tilstand Sun SPOT'en befinder sig i, stor indflydelse på dens levetid. Denne viden kan derfor bruges som indgangsvinkel til designvalget. Hvis radioen anvendes, skal enheden befinde sig i overfladisk søvn eller aktivstadiet enten hele eller dele af overvågningen. Hvis radioen ikke anvendes kan dyb søvnstadiet anvendes. Med hensyn til strømbesparelse, kan det bedst betale sig at befinde i dyb søvn.

Jeg vil kigge på ét design, der forsøger at opnå en så stor kapacitet som mulig. Designet anvender på intet tidspunkt radioen under overvågningen.

Derudover vil jeg beskrive tre andre designs, der alle anvender radioen i større eller mindre grad, i løbet af overvågningen. Det første design anvender kun radioen til at foretage det endelige skift fra en målende til en ikke-målende Sun SPOT. Derefter vil jeg se på to designs, der fungerer som tilnærmelser af et distribueret system. I det distribuerede system fungerer hver Sun SPOT som en individuel enhed i et større netværk, hvor den selv skal skabe viden om de andre enheder i netværket. Denne viden skal løbende opdateres. På baggrund af den viden hver Sun SPOT besidder, kan de foretage ændringer i netværket. Systemet skal således forsøge, at have et netværk med korrekt antal målende enheder, og så vidt det er muligt, sørge for, at der hele tiden bliver målt. Der skal altså udarbejdes en protokol til kommunikation mellem alle Sun SPOTs'ene, som skal sikre en glidende overgang mellem målende enheder.

Der skal desuden tages stilling til lageringen af målingerne. Som udgangspunkt lagres disse i Sun SPOT'ens egen flashhukommelsen. Man kunne dog sikre

målingerne yderligere, i de designs der anvender radioen, ved at overføre dem fra den målende til den overtagende Sun SPOT i løbet af overtagelsesprocessen. Derved gøres overførslen af data til værtsenheden også lettere, da der blot skal overføres fra én enhed. Men da flashhukommelsen udgør en begrænsende faktor, vil dette kunne have indflydelse på kapaciteten af overvågningen, hvorfor det besluttes, at hver Sun SPOT altid beholder sine egne målinger.

Herunder beskrives de fire designs mere detaljeret. Ved hvert design er det desuden muligt at forøge kvaliteten af overvågningen, i og med brugeren har mulighed for at specificere hvor mange Sun SPOTs, der skal måle samtidig. Det vurderes desuden, fra analysen af den interne klokke 4.1.5, at denne fint kan anvendes i forbindelse med designs der er afhængige af klokken.

7.3.1.1 Fast overtagelsestid uden radio

Fast overtagelsestid uden radio forsøger at opnå en så stor strømbesparelse som mulig. Der udregnes et overtagelsestidspunkt for hver af Sun SPOTs'ene, der indgår i overvågningen. Overtagelsestidspunktet er det tidspunkt hvor det forventes, at den tidligere målende Sun SPOT har brug for at blive overtaget. Sun SPOTs'ene overlapper hinanden, uden brug af radiokommunikation. I overtagelsesperioden måler begge enheder.



Figur 7.3: Eksempel på fast overtagelsestid uden radio

Dette bevirker, at alle på nær den målende Sun SPOT befinder sig i dyb søvn, hvorved de næsten intet strøm bruger. I måleintervallerne vil den målende Sun SPOT befinde sig i overfladisk søvn, da den er nødt til at give strøm til accelerometeret.

Overtagelsestiden udregnes på baggrund af intervallerne mellem målingerne. Da måleintervallerne er faste, kan man ud fra disse forudsige hvornår Sun SPOT'en løber tør for strøm. Da accelerometeret ikke måler i intervaller, vil den teoretisk kunne dræne batteriet hvis der sker mange ryst. Da dette er uforudsigeligt er man som minimum nødt til at anvende en sikkerhedsmargen for overtagelsestiden. På trods af det, er fremgangsmåden ikke særlig robust. Skulle en Sun SPOT for eksempel få tekniske problemer undervejs opdages dette ikke, hvorved der

ingen målinger vil blive foretaget i en periode og derudover er det også svært at forudsige overtagelsestiden nøjagtigt.

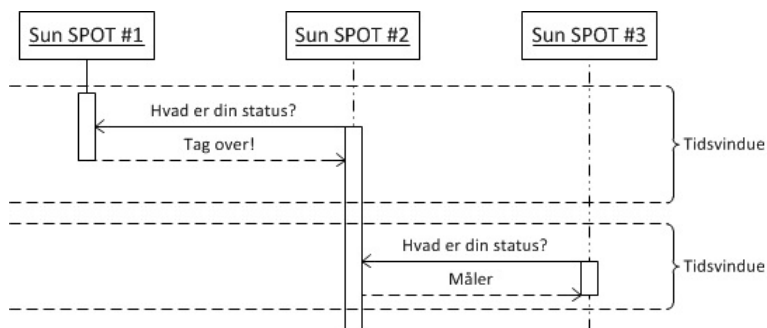
7.3.1.2 Fast overtagelsestid med radio

Fast overtagelsestid med radio fungerer som udgangspunkt på samme måde som fast overtagelsestid uden radio. Radioen åbnes dog ved alle overtagelsestidspunkter på alle Sun SPOTs'ene. Igennem radiokommunikation aftales det hvem, der overtager den målende Sun SPOT.

Det er altså ikke på forhånd givet hvem, der skal overtage den målende Sun SPOT, dette aftales først ved hver overtagelse. Igennem kommunikationen, ved overtagelsen, opnås en smule robusthed, da det her er muligt at kontrollere hvilke Sun SPOTs, der stadig er tændte og hvilke der er slukkede. Derved øges sandsynligheden for en succesfuld overtagelse, i forhold til fast overtagelsestid uden radio.

7.3.1.3 Tidssynkroniseret kommunikation

Ved tidssynkroniseret kommunikation åbner Sun SPOTs'ene radiokommunikationen op i små intervaller i løbet af overvågningen, for at finde ud af status på omkringliggende enheder. Når en Sun SPOT har brug for at blive overtaget, forsøger den i det næste intervalvindue, hvor radioen er åben, at få en anden til at overtage. Et eksempel på en overtagelse ses i figur 7.4. I sekvensdiagrammet er handlingssbare symboler for højere aktivitetsniveau end blot radioen, mens stiplede livslinjer betyder, at Sun SPOT'en kun har radioen tændt. Desuden beskriver et tidsvindue, hvad der rent faktisk sker i en specifik periode. I de



Figur 7.4: Eksempel på overtagelse ved brug af tidssynkroniseret kommunikation

andre perioder vides det ikke præcist hvad der sker. Disse betydninger vil være gældende for samtlige sekvensdiagrammer i rapporten.

Udover at blive bedt om at tage over, kan Sun SPOTs'ene også selv drage den konklusion. Hvis en Sun SPOT ved, at ingen andre enheder måler, tager den selv initiativ til at påbegynde overvågning.

Ved at lade Sun SPOTs'ene vågne op i mindre intervaller, for at lytte efter beskeder, opnås en større robusthed end ved for eksempel fast overtagelsestid og samtidig opnås der stadig en høj kapacitet. Et problem ved dette design er dog, at hvis der er for lang tid mellem radioen åbner, kan der opstå længere perioder uden målinger hvis en enhed går ned. Størrelsen på dette problem kan dog mindskes, ved at lade radioen være åben oftere.

7.3.1.4 Kontinuerlig kommunikation

Kontinuerlig kommunikation er en delvis viderebygning af tidssynkroniseret kommunikation. Denne bygger på princippet om, at enhederne skal have radioen tændt hele tiden, og derved opnå et højt robusthedsniveau. Der foregår en tovejskommunikation mellem henholdsvis målende og ikke-målende Sun SPOTs. Det vil sige, at de ikke-målende enheder i intervaller kontrollerer, at nok Sun SPOTs overvåger, og hvis ikke, overtager de. Modsat har de målende Sun SPOTs mulighed for at bede en ikke-målende Sun SPOT om at overtage på alle tidspunkter. Designet er det mest strømkrævende, men er også det med størst stabilitet. Som det fremgår af bilag A.9, falder levetiden for Sun SPOTs ikke meget ved brug af radioen under overvågning. Til gengæld viser bilag A, at radioen bruger mere strøm end sensorerne, hvorfor målende og ikke-målende enheder måske ender med at løbe tør for strøm samtidig.

7.3.2 Valg af design

Jeg har valgt at arbejde videre med kontinuerlig kommunikation som hoved-design. Af frygt for, at dette design ikke er strømbesparende nok, inkorporeres systemet også til at fungere med tilnærmelser af tidssynkroniseret kommunikation og fast overtagelsestid med radio. Herunder beskrives det egentlige design for kontinuerlig og tidssynkroniseret kommunikation. Fast overtagelsestid med radio-designet opnås ved at lade brugeren definere i hvilke intervaller Sun SPOTs'ene skal kommunikere med hinanden. Dette giver mulighed for, at enhederne først kommunikerer, når brugeren forventer, at de er ved at løbe tør for strøm, hvorved fast overtagelsestid med radio-designet er opnået.

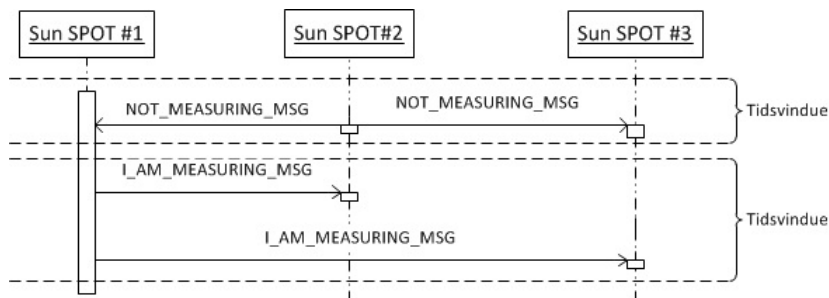
Netværksprotokollen, der opbygges omkring designet, skal være dynamisk. Sun SPOTs'ene skal kunne tilpasse sig den nuværende situation i netværket. Det vil sige, at systemet skal tilstræbe at overholde de specifikationer brugeren har givet, til alle tidspunkter. Hvis en bruger for eksempel har specificeret, at der skal være to målende Sun SPOTs, skal systemet, hvis der pludselig er tre målende Sun SPOTs, selv kunne gå tilbage til stadiet med to målende.

Som tidligere nævnt er der en tovejskommunikation mellem målende og ikke-målende Sun SPOTs. På den måde har alle Sun SPOTs'ene mulighed for at danne sig et overblik over netværket, herunder hvilke andre Sun SPOTs, der måler, ikke-måler eller er lukket ned. Det er vigtigt, at det er den individuelle Sun SPOT, der danner sin egen opfattelse af netværket. Dette medvirker til, at hver Sun SPOT kan drage sine egne konklusioner og ikke blot agere ud fra hvad den bliver fortalt. Hvis for eksempel to Sun SPOTs, A og B, har hvert sit billede af det omkringliggende netværk, og A siger, at B skal stoppe med at måle, vil B kun udføre denne handling, hvis den selv kan verificere, at det rent faktisk er påkrævet, at den lukker ned.

7.3.2.1 Netværksoverblik

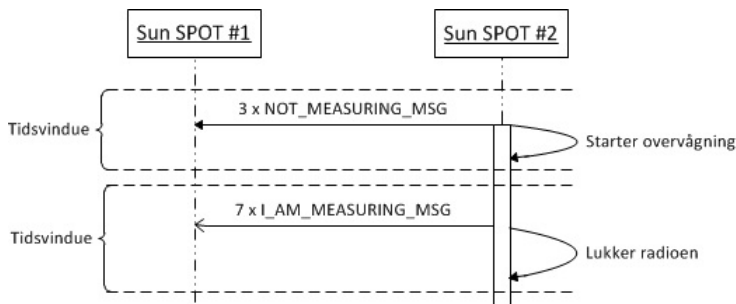
Brugeren har mulighed for at specificere hvorvidt han ønsker at bruge kontinuerlig eller tidssynkroniseret kommunikation. Forskellen er, at tråden til tolkning og håndtering af netværksbeskeder i kontinuerlig kommunikation er aktiv hele tiden, hvorfor det altid er muligt at kommunikere. I tidssynkroniseret kommunikation er det, til forskel, kun muligt at kommunikere i de intervalperioder brugeren har bestemt. Det er desuden kun muligt at kommunikere mellem Sun SPOTs'ene og altså ikke med basestationen. Derudover kommunikeres der direkte mellem Sun SPOTs'ene. Det vil sige, at den indbyggede rutning, som jeg omtalte i analysen [4.1.1](#), er slået fra.

Samtidig med tråden, der tolker og håndterer netværksbeskeder, afvikles en tråd, der varetager udsendelsen af Sun SPOT-beskeder. Alle Sun SPOTs udsender en statusbesked, som fortæller om Sun SPOT'en henholdsvis måler eller er inaktiv. En statusbesked har to formål; at give den anden Sun SPOT information om hvad afsenderen foretager sig og finde ud af om modtageren er lukket ned. Hvis den afsendende Sun SPOT ikke får nogen kvittering fra statusbeskeden, ved den, at modtageren ikke var i stand til at modtage beskeden. Denne viden kan bruges til at drage konklusioner om modtageren. Det vil som udgangspunkt være naivt at drage nogle konklusioner ud fra et enkelt udfald, hvorfor der i stedet opbygges en tæller for antallet af ikke-modtagne beskeder i træk. Tælleren kan nå to niveauer, hvor det første niveau bevirker, at en anden Sun SPOT overtager dens målinger. Det næste niveau medfører, at Sun SPOT'en antages at være



Figur 7.5: Eksempel på udsendelse af statusbeskeder. Sun SPOT #1 måler, mens Sun SPOT #2 og #3 ikke måler

slukket, og derfor ikke kontaktes igen, medmindre den selv gør opmærksom på, at den rent faktisk er vågen. Hvis en målende Sun SPOT ikke kvitterer for en statusbesked, genfremsendes en ny statusbesked hurtigt efter. Dette gøres med henblik på hurtigt at kunne overtage, hvis den målende Sun SPOT skulle være lukket ned. Det er desuden ikke kun statusbeskeder, der bevirker tælleren, alle Sun SPOT-beskeder har indflydelse.



Figur 7.6: Eksempel på udsendelse af statusbeskeder. Sun SPOT #1 var tidligere målende, men er nu ukontaktbar. Sun SPOT #2 forsøger at kontakte #1

7.3.2.2 Overtagelse

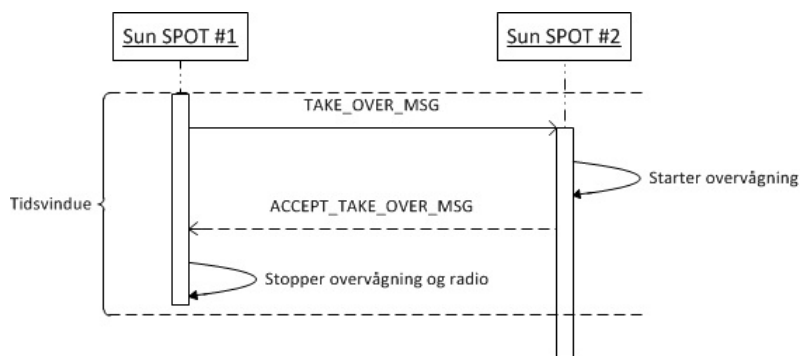
Som tidligere nævnt kan en Sun SPOT vælge at overtage en anden enhed, hvis den vurderer, at enheden ikke længere måler. Denne type overtagelse er ikke glidende, men skyldes derimod en fejl. Jeg vil her se på den type overtagelse, hvor en Sun SPOT bliver bedt om at overtage. Der er som udgangspunkt tre parametre, der har indflydelse på hvornår en målende Sun SPOT anmoder om at blive overtaget; hukommelsen, batteriet og måleinstrumenterne.

Hukommelsen: Denne har en begrænsning på cirka 524 KB. Da hver Sun SPOT indeholder de informationer den selv har opsamlet, er det naturligt, at der sker et skift når hukommelsen er tæt på at være fyldt. Desuden skiftes der også Sun SPOT, hvis det gentagne gange, ikke har været muligt at gemme værdier i hukommelsen

Batteriet: Denne udgør også en naturlig begrænsning for hvornår der skal overtages. Da jeg tidligere har verificeret, at den indbyggede funktion til at hente batteriniveauet i Sun SPOT'en fungerer fint, kan denne anvendes til at sikre, at der bliver overtaget, før enheden løber tør for strøm

Måleinstrumenterne: Hvis det gentagne gange ikke har været muligt at få målinger fra enten accelerometeret, lys- eller temperaturmåleren, vil Sun SPOT'en forsøge at blive overtaget

For at undgå, at der opstår perioder hvor der ikke bliver målt under overtagelsesprocessen, skal der tilknyttes en slags kvittering på, at en anden Sun SPOT har overtaget. Et eksempel på en succesfuld overtagelse ses i figur 7.7.



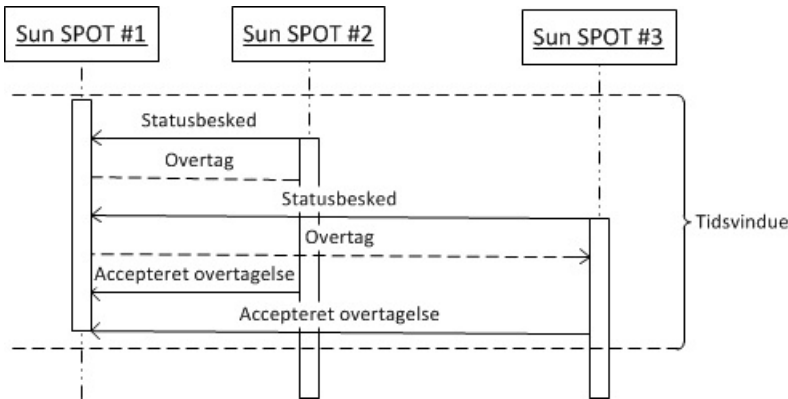
Figur 7.7: Eksempel på en succesfuld overtagelse

Som det fremgår af sekvensdiagrammet af overtagelsen mellem to Sun SPOTs, skal den overtage Sun SPOT altid starte overvågning, før Sun SPOT'en, der ønsker at blive overtaget, kan lukke ned. Derved fås en kontinuerlig overvågning af transporten.

7.3.2.3 Mindskelse af redundans

Der er en række scenarier der skal tages højde for i forbindelse med kommunikationen i netværket. Da der er tale om et netværk, der er afhængig af radiobeske-

der, kan det være svært at forudsige beskedernes ankomsts rækkefølge. Netop dette faktum gør, at flere Sun SPOTs i forbindelse med overtagelsen kan få besked på, at de skal overtage. Derved opstår et scenarie hvor flere Sun SPOTs pludselig forsøger at overtage, mens der i virkeligheden kun var én der skulle overtage. Scenariet er vist som et sekvensdiagram i figur 7.8. Problemet løses ved at tildele den første Sun SPOT, der forsøger at overtage, en token. En token er en slags billet, der er gyldig i en mindre periode. Der kan kun udstedes én token ad gangen, og denne er således adgangsgivende til overtagelsen af Sun SPOT'en. Derved er der kun én Sun SPOT, der bliver bedt om at overtage.



Figur 7.8: Eksempel på en forkert overtagelse

I forbindelse med en overtagelse er det vigtigt, hurtigt at tilkendegive, at Sun SPOT'en er begyndt at måle. Hvis ikke de omkringliggende Sun SPOTs får dette at vide hurtigt, kan de danne sig et billede af et netværk hvor der ikke bliver overvåget. Derved starter de selv op for at varetage overvågningen, hvorved for mange Sun SPOTs er målende.

Det er svært at sikre sig mod samtlige scenarier, hvorfor det, som tidligere nævnt, er vigtigt, at systemet er dynamisk og derved, for eksempel, selv er i stand til at justere på antallet af målende Sun SPOTs. I den forbindelse er det vigtigt at der findes en fælles opfattelse af hvilke Sun SPOTs, der har højere overtagelsesprioritet end andre, således at det er den rigtige Sun SPOT, der stopper med at måle. Prioriteringen kan for eksempel beregnes ud fra den unikke mac-adresse hver Sun SPOT besidder.

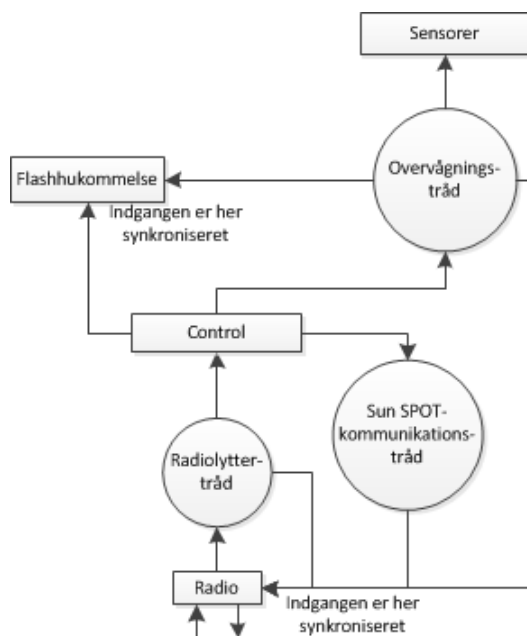
7.3.2.4 Generelt

Ved overvågning med én Sun SPOT, var der kun brug for én tråd, til enten at varetage overvågningen eller kommunikationen med basestationen. I designet til kollektiv overvågning anvendes der yderligere to tråde, så der i alt er tre tråde:

Generel radiokommunikation: Sikre modtagelse og håndtering af beskeder fra andre Sun SPOTs

Sun SPOT-kommunikation: Varetager udsendelsen af statusbeskeder til omkringliggende Sun SPOTs

Overvågning: Sikre overvågningen af de tre måletyper



Figur 7.9: Flowchart for Sun SPOT-programmet

Som det fremgår af figur 7.9, har samtlige tråde mulighed for at kommunikere gennem radioen. Den generelle radiokommunikation skal som udgangspunkt kunne svare på netværksbeskeder, og Sun SPOT-kommunikationen skal udsende statusbeskeder. Derudover skal overvågningstråden have mulighed for at udsende beskeder. Dette skyldes, at der i forbindelse med overvågningen kan opstå problemer med eksempelvis måleinstrumenterne, som kræver at, Sun SPOT'en hurtigt bliver overtaget. De tre tråde beskrives mere detaljeret i næste kapitel.

KAPITEL 8

Implementering og test

I dette kapitel vil jeg beskrive implementeringen af netværksprotokollen, der udgør kommunikationen mellem Sun SPOTs'ene. Jeg vil ikke uddybe udvidelsen af værtsprogrammet, da jeg vurderer, at dette er gjort tilstrækkeligt i designkapitlet 7.2. Herefter vil jeg beskrive testene af Sun SPOT-kommunikationen og kommentere resultaterne.

8.1 Sun SPOT

Fra tidligere kun at have én tråd, kan Sun SPOT-programmet nu have op til tre sideløbende tråde. De tre tråde varetager henholdsvis radiokommunikationen samt behandling af netværksbeskeder, overvågning og udsendelse af statusbeskeder.

8.1.1 Radiokommunikation

Tråden til kommunikationen opretholdes stadigvæk under transporten. I tids-synkroniseret kommunikation er radiotråden dog kun aktiv i intervaller. For at

opnå dette, manipuleres med modtagerforbindelsens timeout-parameter. Denne parameter sættes altid til at være tiden, der skal tilbagelægges før radiointervallet slutter. Derved bruges den `TimeoutException` der smides (engelsk: throws) til at lukke radioen. Der tages desuden også højde for forlængelse og forkortelse af radiointervallet i specialtilfælde, hvor en besked for eksempel tager længere tid at behandle.

Som et led i den generelle radiokommunikation, kontrolleres det hvorvidt der er grundlag for, at radioen stadig er åben. Hvis det for eksempel vurderes, at alle omkringliggende Sun SPOTs er lukket ned, er der ikke behov for at bruge ressourcer på at holde radioen åben.

8.1.2 Overvågning

Overvågningen af transporten udgøres, som i individuel overvågning, af sin egen tråd. Til forskel fra tidligere, kontrollerer denne nu de parametre, der har indflydelse på hvorvidt der skal overtages. Som en del af hver intervalmåling af lysniveau og temperatur, kontrolleres det således om der skal overtages.

8.1.3 Udsendelse af statusbeskeder

Udsendelsen af statusbeskeder varetages af sin egen tråd. Beskederne udsendes i brugerdefinerede intervaller. Udsendelsen er baseret på hvorvidt en Sun SPOT er målende eller ikke-målende. Hvis en Sun SPOT måler, kontrolleres det om den har behov for at blive overtaget, ellers udsendes der blot en regulær statusbesked, hvor den fortæller de andre enheder, at den måler. Er en Sun SPOT ikke-målende, kontrollerer den først hvorvidt den faktisk kan overtage. Det vil sige at flashhukommelsen for eksempel ikke må være fyldt. Hvis Sun SPOT'en ikke kan overtage, lukker den ned og underretter de andre enheder i netværket. Hvis enheden kan overtage, udsendes statusbeskeder og baseret på disse, vurderes det efterfølgende om der er behov for at påbegynde overvågning.

8.1.4 Parallele problemer og valg af datastruktur

Flere af trådene er kun aktive i intervaller. Det vil sige, at de en stor del af tiden skal befinde sig i et sovende stadie. Til det formål bruges trådens `sleep`-funktion. Systemet har mulighed for at stoppe trådene. Dette sker ved at sætte et flag i tråden, som fortæller at dette er dens sidste iteration og derefter kalde

`interrupt` på tråden. Er tråden allerede i gang med at afvikle kode, gøres dette færdigt. Der foretages således ingen direkte afbrydelser under kodeafviklingen, da det underliggende hardware ikke altid tåler dette. Efter endt afbrydelse fusioneres de to tråde, for at sikre, at tråden er endeligt stoppet, før hovedtråden fortsætter.

Da Sun SPOTs har et begrænset udvalg af Java-biblioteker tilgængelige, er der som udgangspunkt ikke mange datastrukturer at vælge imellem. For at få en dynamisk datastruktur som for eksempel den *ArrayLists* tilbyder, har jeg udviklet en simpel dynamisk listestruktur. Strukturen bruges til at indeholde Sun SPOTs, der indgår i overvågningen. Strukturen tilgås af de tre tråde, der henholdsvis varetager afsendingen af Sun SPOT-statusbeskeder, overvågningen og modtagelse af beskeder. Strukturen kan kun ændres i initialiseringsfasen af beskeder fra basestationen. Det vil sige, at når trådene for afsending af statusbeskeder og overvågningen er aktive, kan strukturen ikke ændres, men blot læses. Der er derfor ikke anledning til at sikre den mod flere tråde.

8.2 Test

Herunder testes kommunikationen mellem Sun SPOTs'ene under transporten. Der er som udgangspunkt ikke forskel på de grundlæggende funktionaliteter i henhold til de tre tidligere omtalte designs. Der kan opstå situationer hvor enhederne, der indgår i et netværk, der anvender kontinuerlig kommunikation, kan overtage en målende Sun SPOT hurtigere end i to andre designs. Da dette er den eneste reelle forskel, er der ikke behov for at foretage separate tests af hvert enkelt design.

8.2.1 Fremgangsmåde

Som nævnt under testen af den individuelle overvågning 6.3, findes der to forskellige måder at debugge Sun SPOTs på. Da testning af netværksprotokollen mellem Sun SPOTs'ene har været det vigtigste at undersøge, har jeg undladt at bruge OTA debugging. Dette gøres, da det må forventes, at OTA debugging influere kommunikationen, som netop er den jeg ønsker at debugge. I stedet bruges print debugging over USB.

Ved brug af tidsstempler på konsoludskrifter, er det muligt at verificere hvornår forskellige netværksbeskeder afsendes og modtages. Tidsstemplerne er ikke nødvendigvis akkurate, og man kan forvente en smule afvigelse. Klokkeren, der an-

vendes til at genere tidsstempler på Sun SPOT'ene, opdateres hver gang man eksporterer sit program og tælles i millisekunder.

Sun SPOT 2078		Sun SPOT 3831	
Tid	Handling	Tid	Handling
(1307451698570) (1307451698638)	Measured temperature... Measured light...		
(1307451718252)	Received: Not measuring (35) from 0014.4F01.0000.3831	(1307451718231)	Sending: Not measuring (35) to 0014.4F01.0000.2078
(1307451718273)	Sending: Take over (33) to 0014.4F01.0000.3831	(1307451718304)	Received: Take over (33) from 0014.4F01.0000.2078
		(1307451718651)	Measuring...
(1307451718685)	Received: I am measuring (36) from 0014.4F01.0000.3831	(1307451718662)	Sending: I am measuring (36) to 0014.4F01.0000.2078
		(1307451718788) (1307451718856)	Measured temperature... Measured light...
(1307451718890)	Received: Accept take over (34) from 0014.4F01.0000.3831	(1307451718868)	Sending: Accept take over (34) to 0014.4F01.0000.2078
(1307451718908)	Stopped measuring		
(1307451718920)	Sending: Shutdown (39) to 0014.4F01.0000.3831	(1307451718950)	Received: Shutdown (39) from 0014.4F01.0000.2078
(1307451718953)	Closed SunSPOT communication	(1307451718964)	Closed SunSPOT communication
(1307451718973)	Closed connection	(1307451718984)	Closed connection

Figur 8.1: Eksempel på print debugging af to Sun SPOTs

Til verificering af kommunikationen foretages en manuel gennemgang af konsoludskrifterne. For overskuelighedens skyld, har jeg i eksemplet opstillet korresponderende handlinger overfor hinanden. Da Sun SPOT-programmet består af flere forskellige tråde, behøver konsoludskrifterne ikke nødvendigvis være kronologiske. Derfor kan tidsstemplerne bruges til at verificere den egentlige rækkefølge af handlingerne. Generelt kontrolleres det om kommunikationen mellem enhederne virker tilfredsstillende. Det vil sige, at beskederne kommer i den korrekte rækkefølge og at de rent faktisk fremkommer. Desuden bekræftes det, at de interne handlinger på Sun SPOTs'ene stemmer overens med hvad der forventes.

8.2.2 Generelle tests

Testning af netværksprotokollen er omfattende, fordi systemet kan befinde sig i mange obskure situationer. Det har derfor været vigtigt at få udført tests af de tænkelige fejlscenarier samt funktionerne til mindskelse af redundans. Jeg har haft fire udleverede Sun SPOTs og har brugt disse igennem mine tests. I mange af testene har det været nødvendigt at manipulere fejlscenarier frem. For at få Sun SPOT'en til at tro, at der er noget galt med enten hukommelsen, måleinstrumenterne eller batteriet, sættes et flag med indikation herom, når

man trykker på en af knapperne på applikationskortet. Hvis Sun SPOT'en i en periode ikke skal kunne kommunikere med de andre enheder, skabes en afstand mellem dem, der er stor nok til, at beskederne ikke kommer frem. Testudførelserne findes i bilag E. Overordnet testes der på følgende:

- Kontrolleret overgang fra en målende til en ikke-målende Sun SPOT. Verificering af den tidligere nævnte overtagelsesproces. Derudover sikres det, at Sun SPOT'en er i stand til at detektere fejl, der kræver, at den bliver overtaget
- Ukontrolleret terminering af målende Sun SPOT, hvorved det bekræftes at Sun SPOTs'ene i netværket opfanger, at der ikke bliver målt, og derfor selv påbegynder overvågning
- Ved ukontrolleret terminering af ikke-målende Sun SPOT tjekkes det, at målende og ikke-målende enheder foretager de korrekte korresponderende handlinger, som for eksempel lukning af radioen
- Henholdsvis kort og lang udeblivelse af en målende Sun SPOT og derpå tilbagevenden. Det kontrolleres, at når en målende enhed er udeblevet en periode, bliver dens overvågning varetaget af en anden enhed. Derudover skal overvågningsopsætningen kunne gå tilbage, hvis en målende enhed, igen bliver kontaktbar

8.2.2.1 Testresultat

Overordnet set blev alle testene udført med forventede resultater. Dette skyldes blandt andet, at mange af situationerne ligger til grund for protokollen. Derudover ses det af enkelte af testene, at de mere komplekse fejlsituationer, der kan opstå i netværket, med tiden går i sig selv. Disse skyldes oftest, at enhederne har forskellige forestillinger af hvordan netværket ser ud. Denne forestilling bør dog teoretisk kun være forvrænget i en kortere periode, hvorefter systemet bør vende tilbage til en korrekt opsætning. Desuden ses det i test E.4.2, at der opstår en situation hvor der i stedet for at være én målende Sun SPOT er to. Situationen skyldes antagelsen om, at en anden enhed ikke findes, når den har været ukontaktbar 10 gange i træk.

Desuden blev det også verificeret, at ved brug af enten kontinuerlig eller tids-synkroniseret kommunikation befandt enhederne sig i henholdsvis overfladisk eller dyb søvn. Denne observation gøres ud fra aktivitets-LED'en, der sidder på Sun SPOT'en.

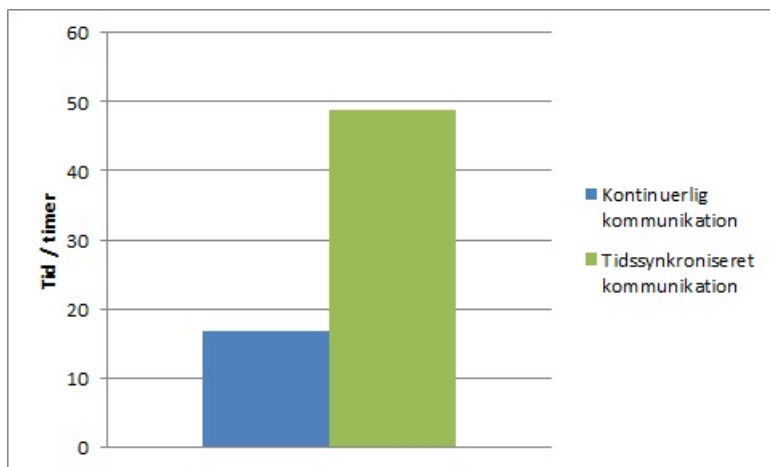
8.2.3 Felt-test

Under testene af individuel overvågning, blev det bekræftet at de generelle overvågningsfunktioner fungerede tilfredsstillende. I denne test er det derfor vigtigere at kigge på de nye aspekter ved kollektiv overvågning. I felt-testene tjekkes det om der kan opnås en forøgelse af den samlede målelængde. Derudover kontrolleres det også, at der ikke er længere perioder i overvågningen, hvor der ikke bliver målt.

Der blev udført tests af henholdsvis tidssynkroniseret og kontinuerlig kommunikation. Standardmåleindstillinger anvendtes i begge tests. Før udførelsen af testene var alle fire Sun SPOTs desuden fuldt opladte.

8.2.3.1 Testresultat

Begge tests blev udført med forventede resultater. Måletiderne var alle fornuftige. Der var ingen længere perioder, hvor der ikke blev overvåget. Testene giver et godt indtryk i forskellen på målelængden. Af figur 8.2 fremgår det, at tidssynkroniseret kommunikation, fordi den kun har radioen åben i intervaller, opnår en målelængde, der er næsten tre gange så lang som kontinuerlig kommunikation. Jævnfør testene ved brug af én Sun SPOT 6.3.2, er det muligt at forøge målelængden væsentligt.



Figur 8.2: Felt-tests ved brug af kontinuerlig og tidssynkroniseret kommunikation

Konklusion

Jeg har i dette projekt undersøgt hvorvidt Sun SPOTs kan bruges til overvågning af pakketransporter, og herunder give et fornuftigt helhedsindtryk af et transportforløb. Den første fase bestod af udviklingen af de simple overvågningsfunktioner. Til dette formål anvendtes de funktionaliteter Sun SPOT stiller til rådighed. I løbet af denne fase, stod det klart, at Sun SPOTs'enes hardware har nogle væsentlige afvigelser, der gør, at overvågningssystemet mere er et bevis af et koncept, end et markedsdygtigt produkt. Blandt andet accelerometerets begrænsning og afvigelse samt temperaturmålerens placering har medført, at de målte værdier ikke er brugbare nok.

Det blev vurderet, at der var behov for at udvikle et eksternt program til at varetage administreringen af Sun SPOTs'ene. Programmet giver en nem og intuitiv administrering af enhederne. Derudover vises de indsamlede informationer som grafer i programmet. Det vurderes, at værdifremvisningen giver et hurtigt og brugbart overblik over et transportforløb.

Ved felt-testen af det simple overvågningsystem, hvor hver Sun SPOT overvåger for sig selv, stod det klart, at kapaciteten for en enkelt enhed, ikke er tilstrækkelig til at dække en gennemsnitlig pakketransport. I et forsøg på at øge kapaciteten, men også kvaliteten og robustheden af overvågningen, forsøgte flere Sun SPOTs inkluderes i én overvågning.

I den forbindelse blev systemet udvidet, med mulighed for, at sensorerne overvåger hinanden. Derved blev en generel forøgelse af kvalitet og kapacitet opnået. Overvågningsnetværket, der udgøres af sensorerne i en pakke, er endvidere robust overfor ukontrollerede udfald af Sun SPOTs.

De vigtigste resultater i forbindelse med det færdige system kan opsummeres til:

- Opsamle værdier fra accelerometeret, lys- og temperaturmåleren og lagre dem sikkert. Værdierne kan til en vis grad give et brugbart billede af transporten
- Gøre brugeren i stand til, nemt og intuitivt, at administrere Sun SPOTs'ene. Desuden er det muligt hurtigt at danne et fornuftigt indtryk af transporten
- Ved at forøge antallet af Sun SPOTs per pakke, er det muligt at opnå en væsentlig forøgelse af kvaliteten, kapaciteten og robustheden

Sun SPOTs'ene giver fin mulighed for at demonstrere overvågningskonceptet, og vil til en vis grad kunne anvendes i forbindelse med en faktisk pakketransport. Det er lykkedes, at lave et system, der giver brugeren mulighed for at tilpasse overvågningen til hans præferencer, hvorved det for eksempel er muligt at opsætte et intenst overvågningsforløb eller opnå stor overvågningskapacitet.

9.1 Anvendelsesmuligheder og udvidelser

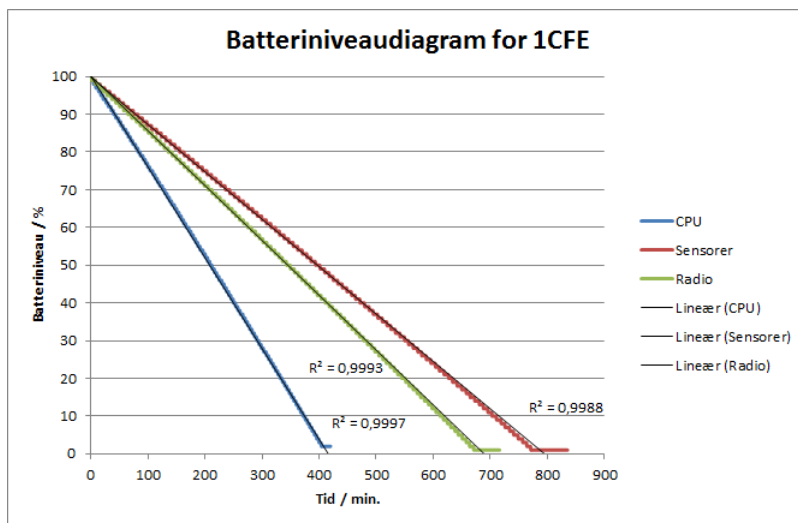
Som udgangspunkt er der tale om et system til overvågning af pakketransporter. Teoretisk set har systemet dog et større anvendelsesspektrum. Pakker er ikke den eneste type forsendelse der kan overvåges. Systemet kan anvendes i alle forsendelseshenseender, for eksempel transport af containere, bagage osv. Derudover er der også mulighed for anvendelse i mere faste omgivelser, som for eksempel et rum, hvor lysniveau og temperatur ønskes registreret med jævne mellemrum.

Da hardwaren udgør den væsentligste begrænsning i forhold til systemet, ville det være nærliggende at udskifte henholdsvis accelerometeret og temperaturmåleren, med bedre instrumenter. Hvad angår softwaren, ville det være interessant at lade radioen lytte efter værtsbeskeder hele tiden, så brugeren altid har mulighed for at administrere Sun SPOTs'ene, også selvom de overvåger. Derudover kunne man indføre planlægning af hvornår der skulle overvåges, og

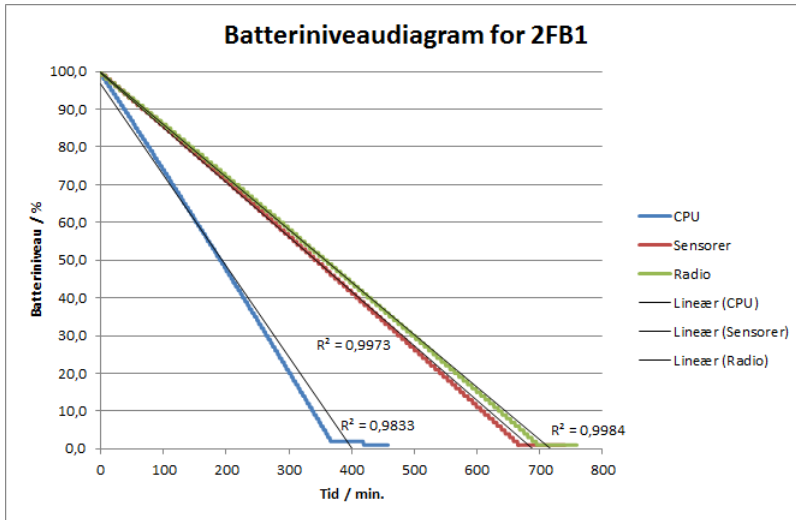
derigennem give mulighed for at slukke ned for enkelte af måleinstrumenterne under dele af transporten.

Batterianalyse

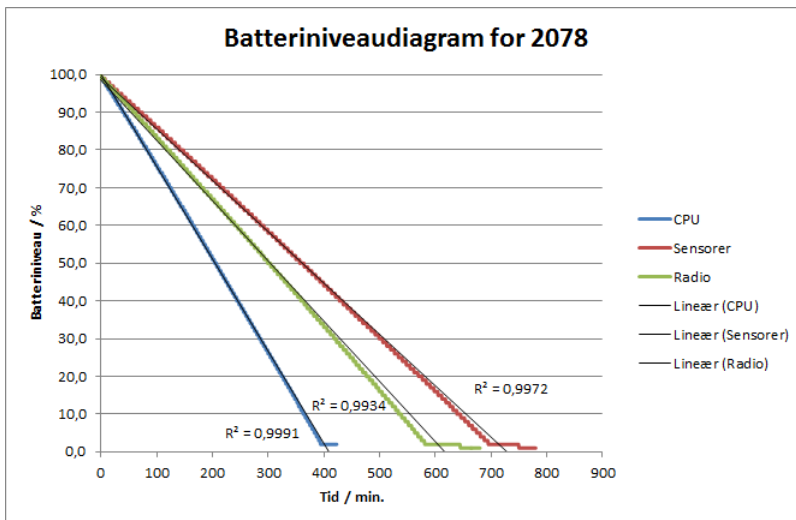
A.1 Batteriniveaudiagrammer



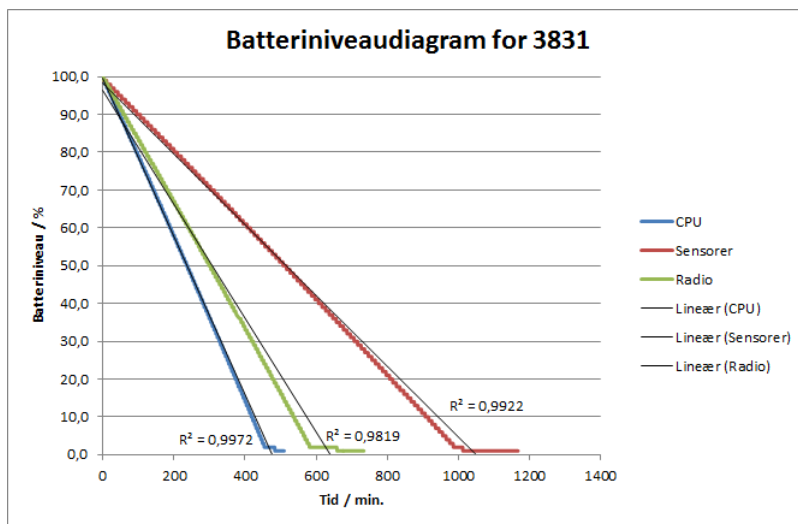
Figur A.1: Batteriniveaudiagram for Sun SPOT 1CFE



Figur A.2: Batteriniveaudiagram for Sun SPOT 2FB1

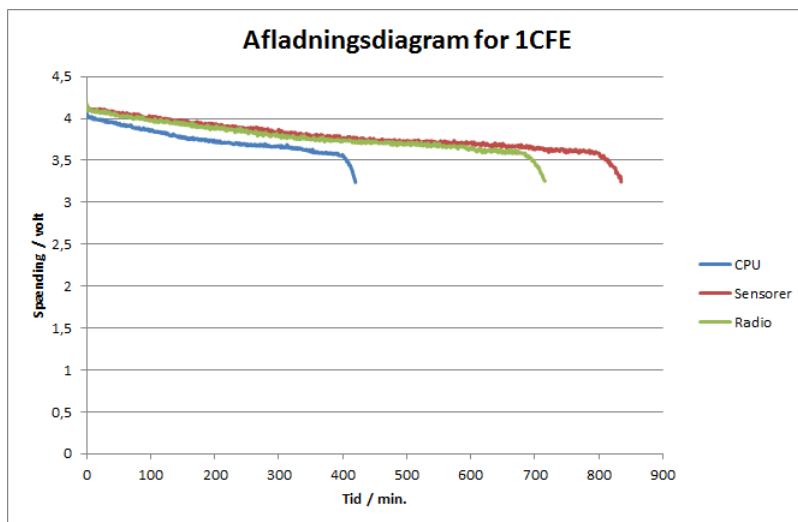


Figur A.3: Batteriniveaudiagram for Sun SPOT 2078

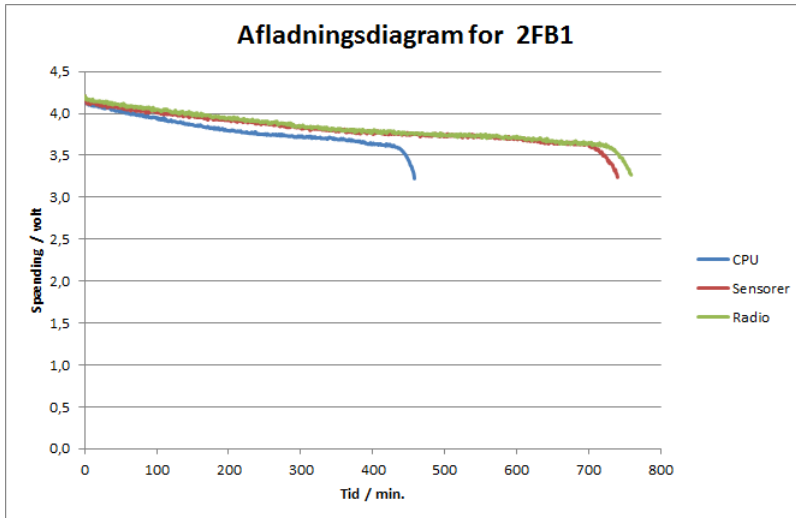


Figur A.4: Batteriniveaudiagram for Sun SPOT 3831

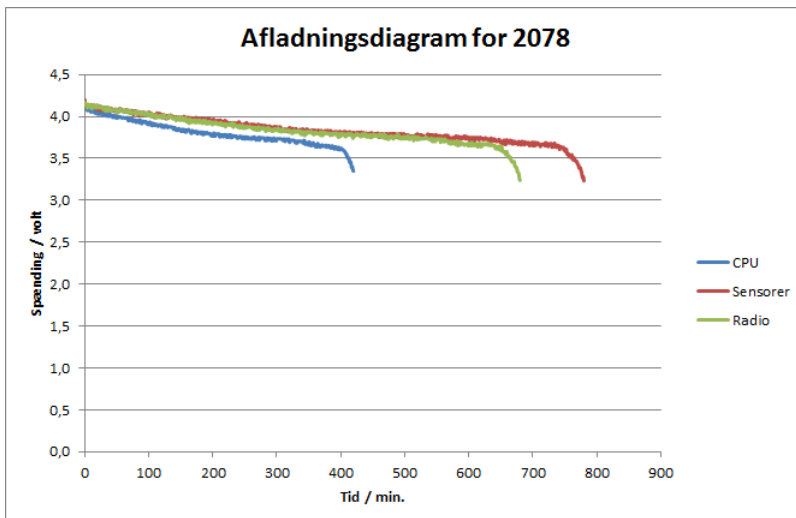
A.2 Afladningsdiagrammer



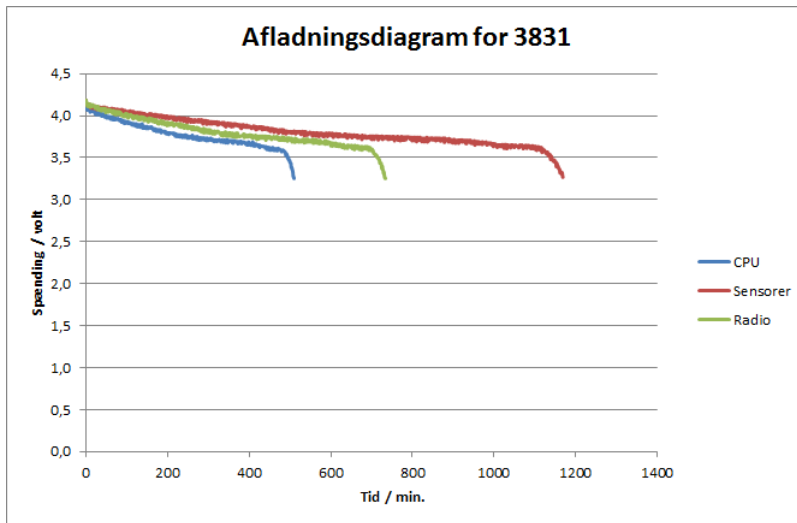
Figur A.5: Afladningsdiagram for Sun SPOT 1CFE



Figur A.6: Afladningsdiagram for Sun SPOT 2FB1

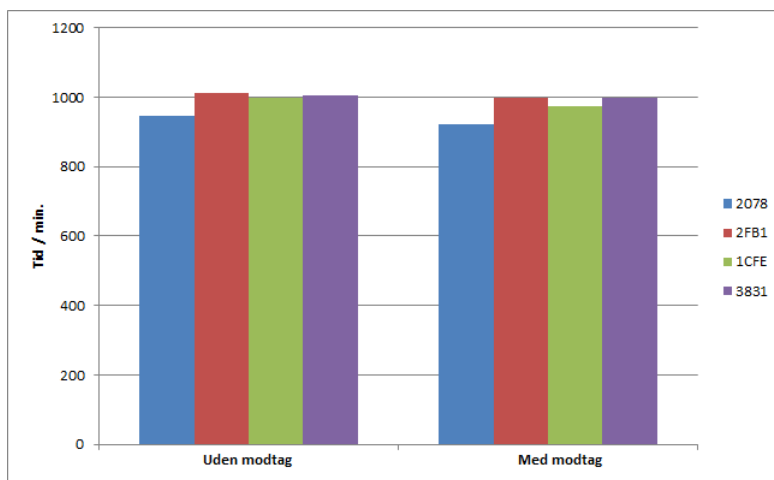


Figur A.7: Afladningsdiagram for Sun SPOT 2078



Figur A.8: Afladningsdiagram for Sun SPOT 3831

A.3 Overvågning



Figur A.9: Overvågning med og uden radio-modtag

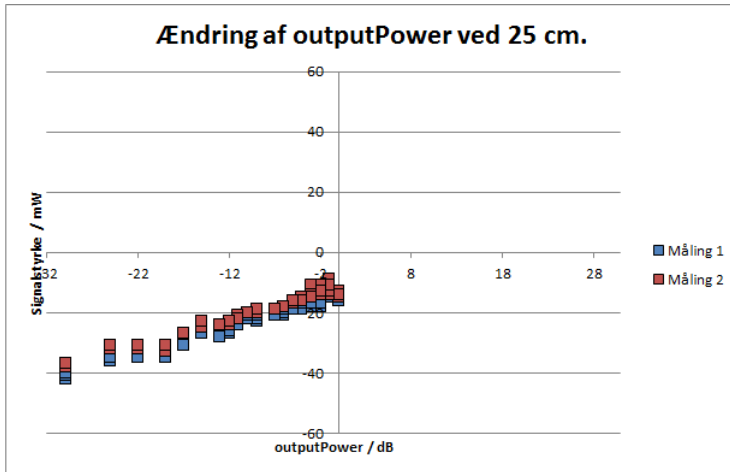
BILAG B

Radioanalyse

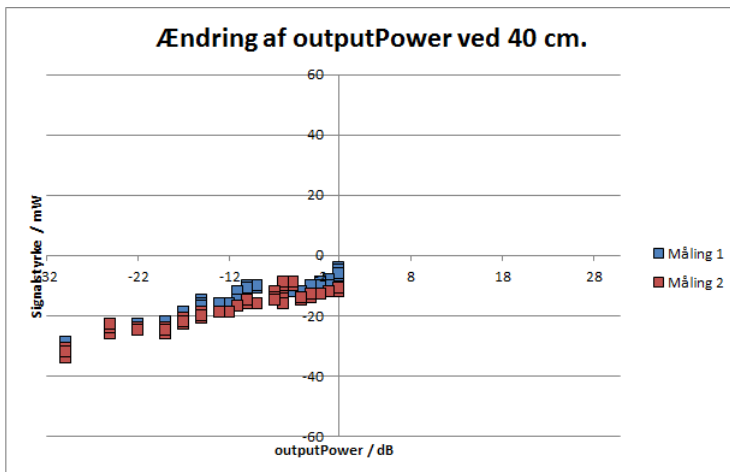
B.1 Signalstyrke

Til udførelse af testene for signalstyrken anvendtes to Sun SPOTs. Hver Sun SPOT blev implementeret med enten et program til at sende eller modtage beskeder. Til radiokommunikationen anvendtes *Radiogram*, da disse indeholder en værdi for styrken som de modtages med. Herpå overførtes 100 beskeder fra den ene enhed til den anden, og aflæsningen af disses signalstyrke ligger til grund for diagrammerne.

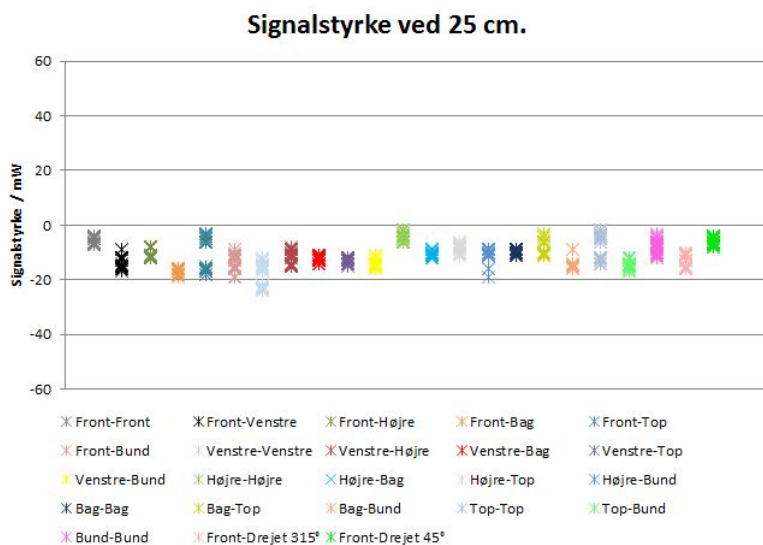
Da det forventes at Sun SPOTs'ene, når de er placeret i en pakke, ikke har mere end 40 cm. mellem hinanden, er der kun lavet to tests ved henholdsvis 25 og 40 cm. De første to diagrammer viser signalstyrken som funktion af `outputPower`, som sænker styrken en pakke udsendes med. De to sidste diagrammer viser signalstyrken som funktion af positionen to kommunikerende Sun SPOTs antager.



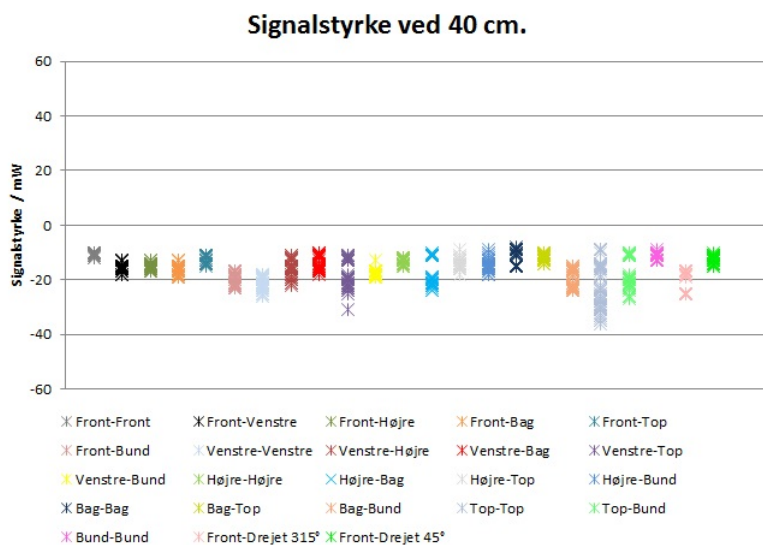
Figur B.1: Analyse af radiosendestyrke ved 25 cm.



Figur B.2: Analyse af radiosendestyrke ved 40 cm.



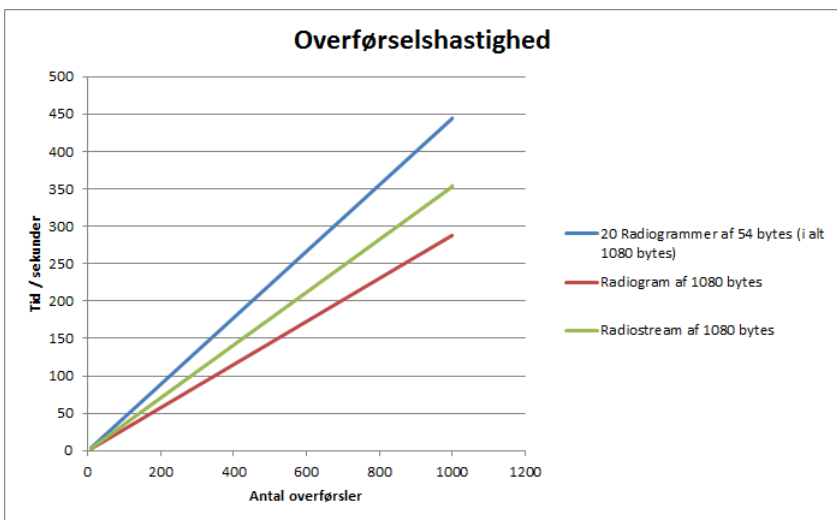
Figur B.3: Analyse af radiostyrke ved 25 cm.



Figur B.4: Analyse af radiostyrke ved 40 cm.

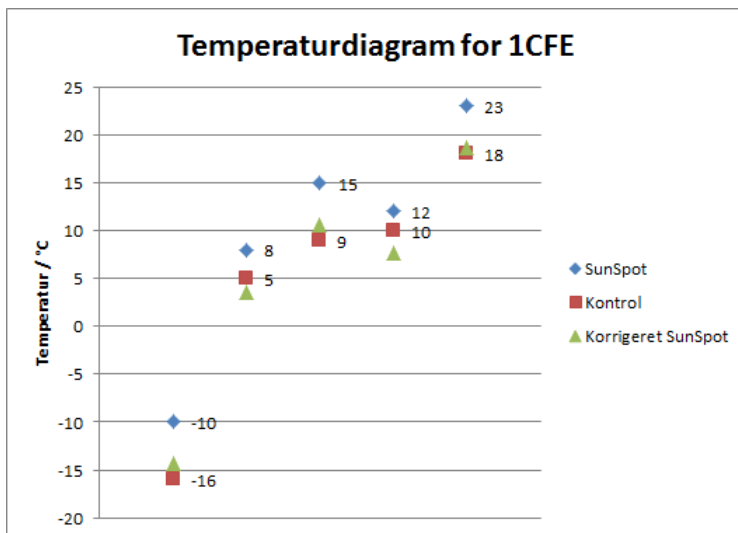
B.2 Overførselshastighed

Til udførelse af testen for overførselshastighed anvendtes to Sun SPOTs. Hver Sun SPOT blev implementeret med enten et program til at sende eller modtage beskeder. Hver transmission mellem enhederne bestod af at sende 1080 bytes. Dette blev gjort for henholdsvis *Radiogram* og *Radiostream*. Ved *Radiogram* blev der henholdsvis lavet en test hvor der sendtes 20 *Radiogrammer* af 54 bytes, hvilket er 1080 bytes, og en test med ét *Radiogram* på 1080 bytes. Der blev udført tests med henholdsvis 10, 100 og 1000 af sådanne sammenhængende transmissioner.

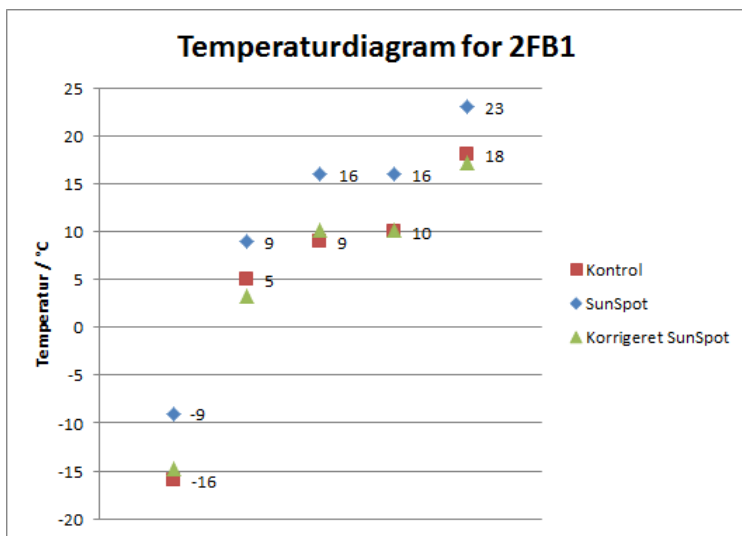


Figur B.5: Analyse af overførselshastighed

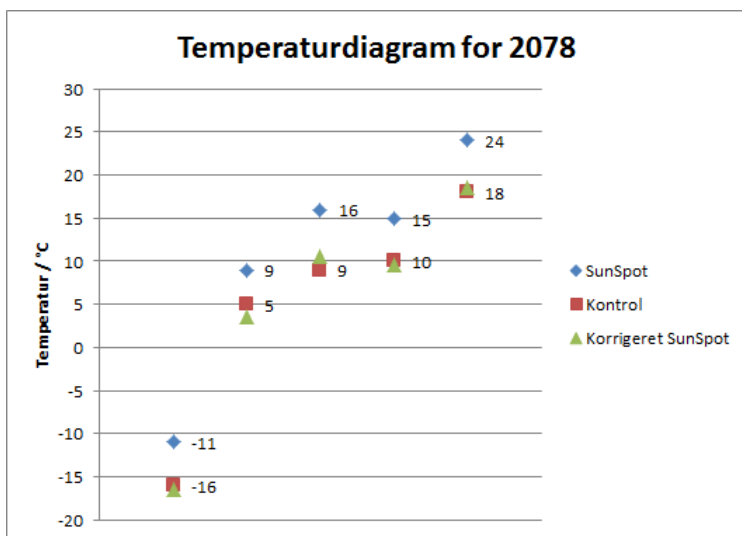
Temperaturmåleranalyse



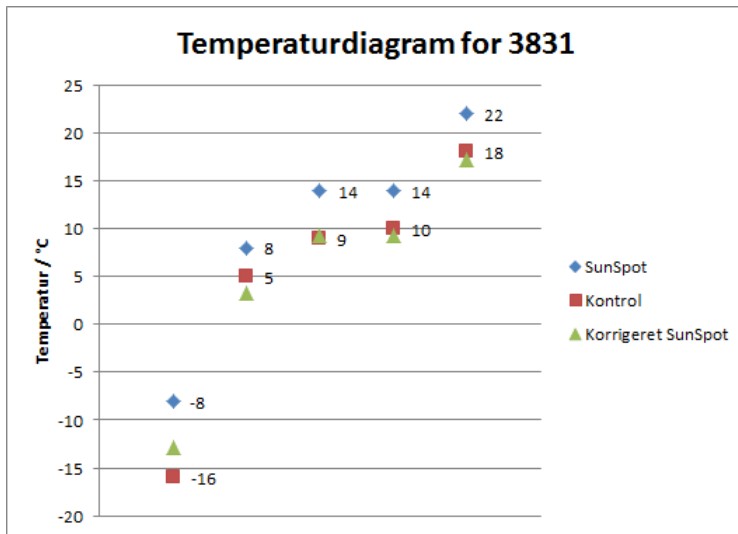
Figur C.1: Analyse af temperaturmåleren for Sun SPOT 1CFE. Korrigeringsfaktor: $-4,4^{\circ}\text{C}$



Figur C.2: Analyse af temperaturmåleren for Sun SPOT 2FB1. Korrigeringsfaktor: $-5,8^{\circ}\text{C}$



Figur C.3: Analyse af temperaturmåleren for Sun SPOT 2078. Korrigeringsfaktor: $-5,4^{\circ}\text{C}$



Figur C.4: Analyse af temperaturmåleren for Sun SPOT 3831. Korrigeringsfaktor: $-4,8^{\circ}\text{C}$

BILAG D

Tests for individuel overvågning

D.1 Initialisering

D.1.1 Test 1

Opstilling: Fire Sun SPOTs: ikke-initialiseret og alle tilknyttet den samme basestation

Scenario: Igennem brugerfladen initialiserer brugeren alle omkringliggende Sun SPOTs

Forventning: Alle Sun SPOTs'enes adresser visualiseres i brugerfladens liste over enheder

Resultat: Som forventet

D.1.2 Test 2

Opstilling: To Sun SPOTs: ikke-initialiseret og tilknyttet hver sin basestation

Scenarie: Igennem brugerfladen initialiserer brugeren alle omkringliggende Sun SPOTs

Forventning: Kun den Sun SPOT, der er tilknyttet basestationen bliver initialiseret. Den anden Sun SPOT bevirker, at der fremkommer en fejlmeddelelse i brugerfladen

Resultat: Som forventet. Se figur [D.1](#)

D.1.3 Test 3

Opstilling: Én Sun SPOT: ikke-initialiseret. Ingen basestation tilsluttet

Scenarie: Igennem brugerfladen initialiserer brugeren alle omkringliggende Sun SPOTs

Forventning: I brugerfladen fremkommer en fejlmeddelelse, der fortæller, at ingen basestation er tilsluttet

Resultat: Som forventet

D.2 Ændring af måleindstillinger

D.2.1 Test 1

Opstilling: Én Sun SPOT: initialiseret

Scenarie: Der ændres på samtlige måleindstillinger for Sun SPOT'en. Alle værdier er lovlige

Forventning: Efter ændringen af måleindstillingerne forventes det, at disse er ændret både ved genåbning af brugerfladevinduet for ændring af måleindstillinger, men samtidig også på Sun SPOT'en

Resultat: Som forventet

D.2.2 Test 2

Opstilling: Én Sun SPOT: initialiseret

Scenarie: Accelerationstærskelværdien ændres til en ulovlig værdi for Sun SPOT'en

Forventning: Før de nye værdier bliver sendt til Sun SPOT'en, skal værten nå at opfange fejlen og fortælle brugeren herom

Resultat: Som forventet. Se figur [D.2](#)

D.2.3 Test 3

Opstilling: Én Sun SPOT: initialiseret

Scenarie: Sendes forkert type måleindstillinger til Sun SPOT

Forventning: Sun SPOT'en gør opmærksom på, at de måleindstillinger den modtog, er forkerte, og at den derfor anvender sine standardindstillinger i stedet

Resultat: Som forventet. Se figur [D.3](#)

D.3 Start overvågning

D.3.1 Test 1

Opstilling: Én Sun SPOT: initialiseret. Flashhukommelsen på Sun SPOT'en er tømt

Scenarie: Start overvågning

Forventning: Efter Sun SPOT'en er blevet bedt om at starte, forsvinder den fra brugerfladen. Desuden observeres det på Sun SPOT'en, at den rent faktisk måler

Resultat: Som forventet

D.3.2 Test 2

Opstilling: Én Sun SPOT: initialiseret. Flashhukommelsen på Sun SPOT'en er ikke tømt

Scenarie: Start overvågning

Forventning: Efter Sun SPOT'en er blevet bedt om at starte, fremkommer en meddelelse i brugerfladen, der fortæller, at hukommelsen ikke er tømt på Sun SPOT'en. Desuden begynder Sun SPOT'en ikke overvågningen

Resultat: Som forventet

D.4 Importer målinger

D.4.1 Test 1

Opstilling: Én Sun SPOT: initialiseret

Scenarie: Importer data fra Sun SPOT

Forventning: Ved importens begyndelse ses en statuslinje (engelsk: progress bar) for procent data overført. Efter alt data er overført, aktiveres de dertilhørende grafer

Resultat: Som forventet. Se figur [D.4](#) og [D.5](#)

D.5 Andre situationer

D.5.1 Test 1

Opstilling: To Sun SPOTs: initialiseret (A/B)

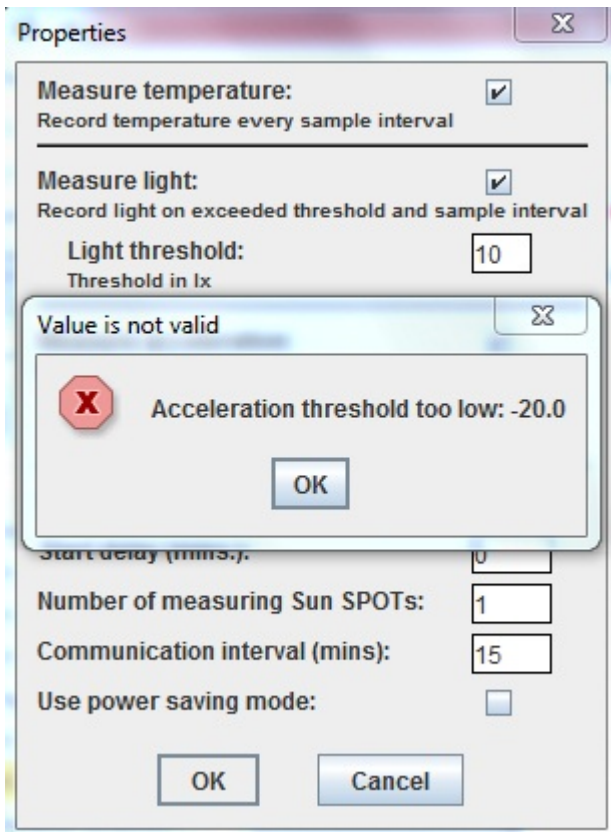
Scenarie: A kommer udenfor rækkevidde af basestationen, mens både A og B er initialiseret

Forventning: Værten opfanger, at A ikke længere findes, hvilket fortælles til brugeren, igennem brugerfladen

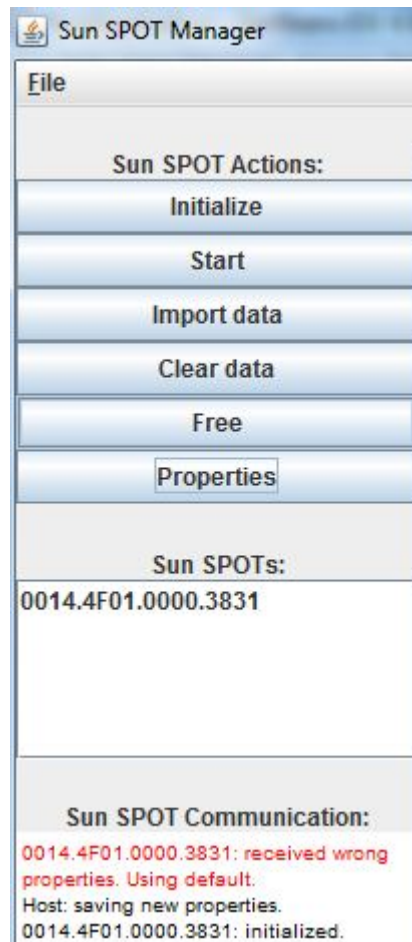
Resultat: Som forventet. Se figur [D.6](#)



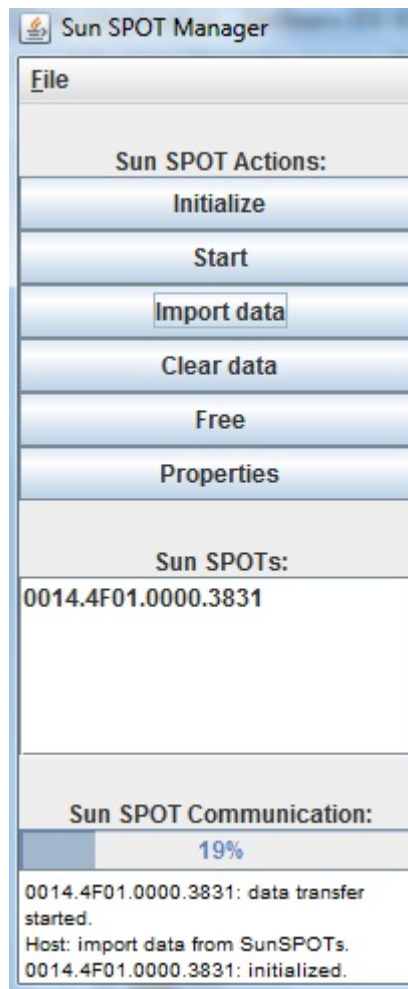
Figur D.1: Screenshot fra test [D.1.2](#)



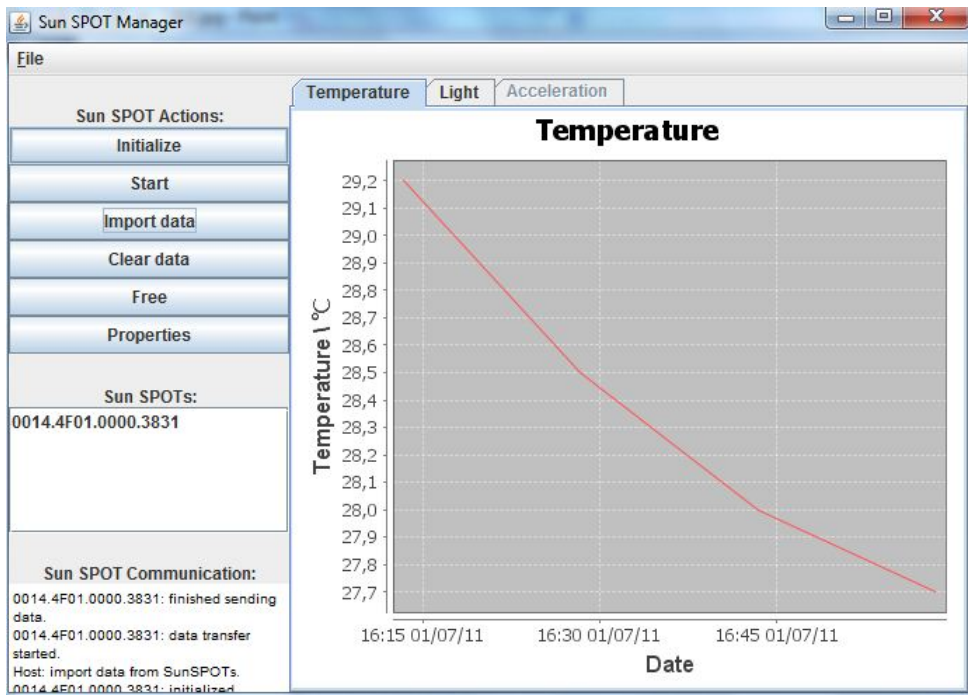
Figur D.2: Screenshot fra test [D.2.2](#)



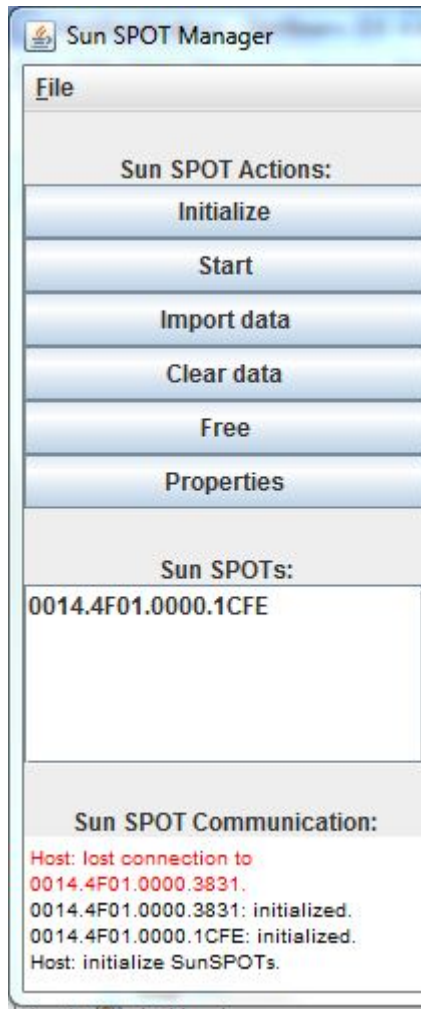
Figur D.3: Screenshot fra test [D.2.3](#)



Figur D.4: Screenshot fra test D.4.1



Figur D.5: Screenshot fra test D.4.1

Figur D.6: Screenshot fra test [D.5.1](#)

BILAG E

Tests for kollektiv overvågning

E.1 Kontrolleret terminering af målende Sun SPOT

E.1.1 Test 1

Opstilling: Tre Sun SPOTs: en målende (A) og to inaktive (B/C)

Scenarie: A tvinges til at tro, at der er noget galt med hukommelsen, og ønsker derfor at blive overtaget hurtigst muligt. Den udsender en anmodning om at blive overtaget til enten B eller C

Forventning: At enten B eller C overtager målingen og at det sker i korrekt rækkefølge, det vil sige at den overtagende Sun SPOT skal starte overvågning før den terminerende Sun SPOT lukker

Resultat: Som forventet. Se sekvensdiagram [F.1](#)

E.1.2 Test 2

Opstilling: To Sun SPOTs: en målende (A) og en inaktiv (B)

Scenarie: A udsender besked om at den ønsker at blive overtaget

Forventning: Sikre test 1's forventninger samt lukning af Sun SPOT-kommunikation og radioen ved overtagelse

Resultat: Som forventet. Se sekvensdiagram [F.2](#).

E.2 Ukontrolleret terminering af målende Sun SPOT

E.2.1 Test 1

Opstilling: To Sun SPOTs: en målende (A) og en inaktiv (B)

Scenarie: A terminerer ukontrolleret, hvorved B ikke ved, at der ikke bliver målt

Forventning: Efter B har forsøgt at kontakte A tre gange påbegynder den overvågning. Desuden skal radioen være åben indtil det 10. mislykkedes kontaktforsøg

Resultat: Som forventet. Se sekvensdiagram [F.3](#)

E.2.2 Test 2

Opstilling: Tre Sun SPOTs: en målende (A) og to inaktiv (B/C)

Scenarie: A terminerer ukontrolleret, hvorved ingen af de andre enheder ved, at der ikke bliver målt

Forventning: Den af to inaktive enheder der først opnår tre på hinanden følgende timeouts fra A påbegynder overvågningen

Resultat: Som forventet. Se sekvensdiagram [F.4](#)

E.3 Ukontrolleret terminering af ikke-målende Sun SPOT

E.3.1 Test 1

Opstilling: To Sun SPOTs: en målende (A) og en inaktiv (B)

Scenarie: B terminerer

Forventning: Efter den målende Sun SPOT har forsøgt at kommunikere med den inaktive enhed 10 gange slukkes radioen og Sun SPOT-kommunikationen

Resultat: Som forventet. Se sekvensdiagram [F.5](#)

E.3.2 Test 2

Opstilling: Tre Sun SPOTs: to målende (A/B) og en inaktiv (C)

Scenarie: C terminerer

Forventning: Efter begge målende Sun SPOTs har forsøgt at kommunikere med C 10 gange slukker de begge radioen og Sun SPOT-kommunikationen, da to målende Sun SPOTs ikke behøver at kunne kommunikere med hinanden

Resultat: Som forventet. Se sekvensdiagram [F.6](#)

E.4 Lang og kort udeblivelse og tilbagevenden af en målende Sun SPOT

E.4.1 Test 1

Opstilling: To Sun SPOTs: en målende (A) og en inaktiv (B)

Scenarie: A udebliver i en kort periode hvor B opbygger fem timeouts

Forventning: Når B tre på hinanden følgende gange ikke kunne kontakte A, overtager den overvågningen og lader radioen være åben. Ved A's tilbagevenden begynder de to Sun SPOTs igen kommunikationen og finder ud af, i fællesskab, hvem der skal fortsætte med at måle

Resultat: Som forventet. Se sekvensdiagram [F.7](#)

E.4.2 Test 2

Opstilling: To Sun SPOTs: en målende (A) og en inaktiv (B)

Scenarie: A udebliver i en lang periode hvor B opbygger 10 timeouts

Forventning: Når B tre på hinanden følgende gange ikke kunne kontakte A, overtager den overvågningen og lader radioen være åben. Efter 10 timeouts, stopper B sit forsøg på at skabe kontakt og lukker radioen. Samtidig med dette, får A samme indtryk af netværket og lukker derfor også sin radio. Derved overvåges af både A og B

Resultat: Som forventet. Se sekvensdiagram [F.8](#)

E.5 Andre situationer

E.5.1 Test 1

Opstilling: Tre Sun SPOTs: to målende (A/B) og en inaktiv (C)

Scenarie: A er ukontaktbar. C får en opfattelse af, at der er behov for at den tager over, mens B ikke mener der er behov for overtagelse

Forventning: C overtager overvågningen. B fortæller nu C, at den skal lukke ned, da den mener, at der er for mange målende Sun SPOTs. C vælger dog at ignorere dette, da den har et andet billede af netværket. Senere får B også opfattelsen at A ikke længere måler

Resultat: Som forventet. Se sekvensdiagram [F.9](#)

E.5.2 Test 2

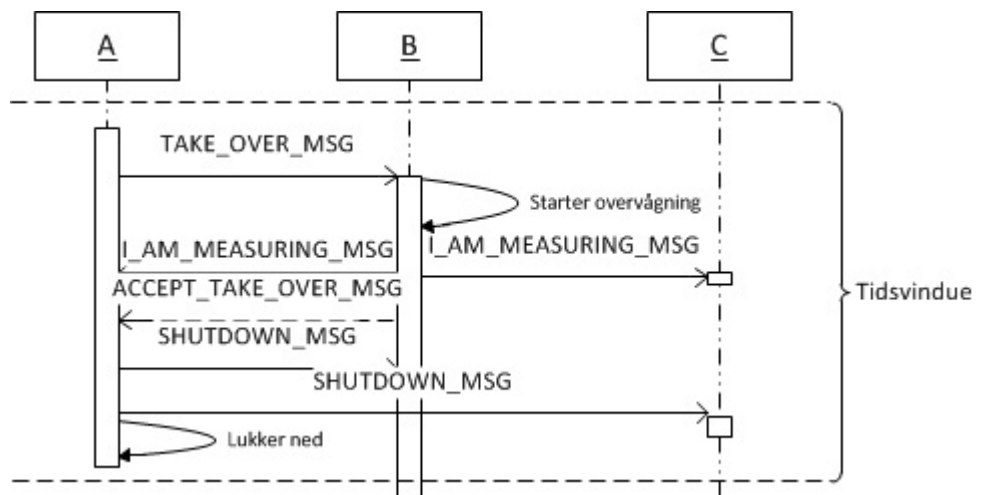
Opstilling: Fire Sun SPOTs: to målende (A/B) og to inaktive (C/D)

Scenario: A er ukontaktbar og B ønsker at blive overtaget

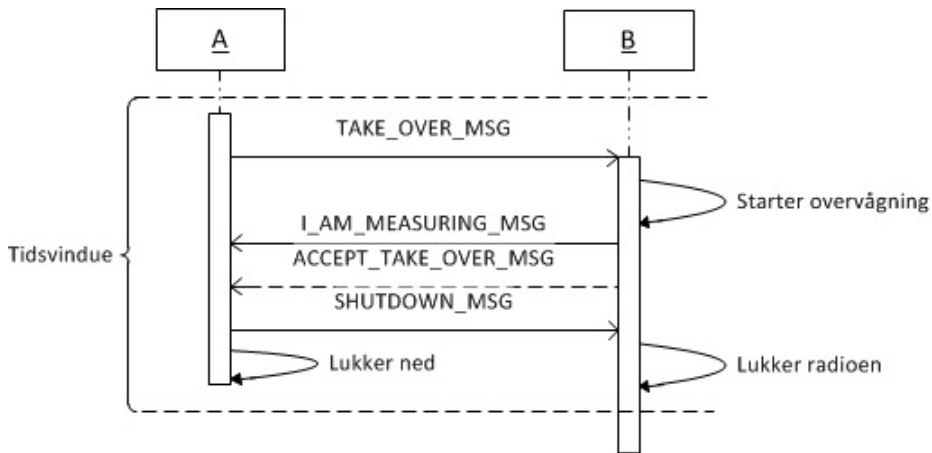
Forventning: B overtages hurtigst muligt af enten C eller D. Den af C og D, der efterfølgende ikke måler, overtager A når den ikke har svaret i en kort periode

Resultat: Som forventet. Se sekvensdiagram [F.10](#)

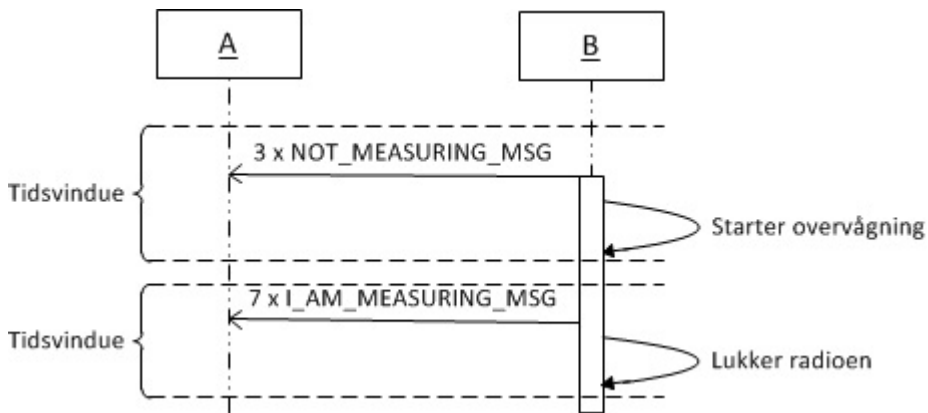
UML-diagrammer



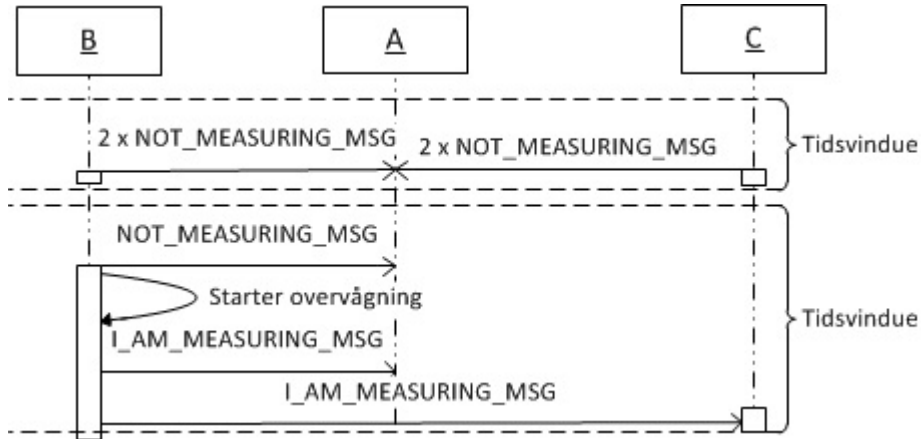
Figur F.1: Sekvensdiagram for test [E.1.1](#)



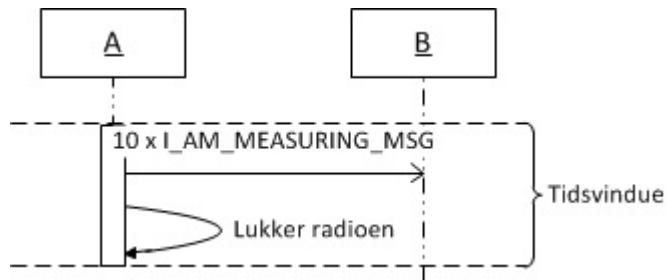
Figur F.2: Sekvensdiagram for test E.1.2



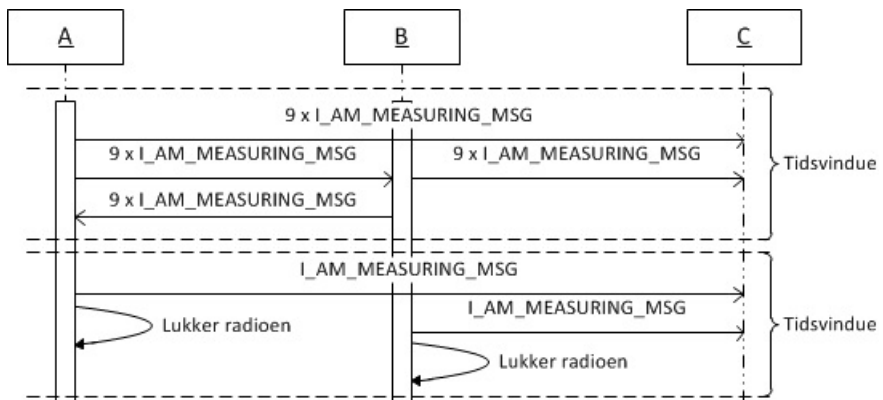
Figur F.3: Sekvensdiagram for test E.2.1



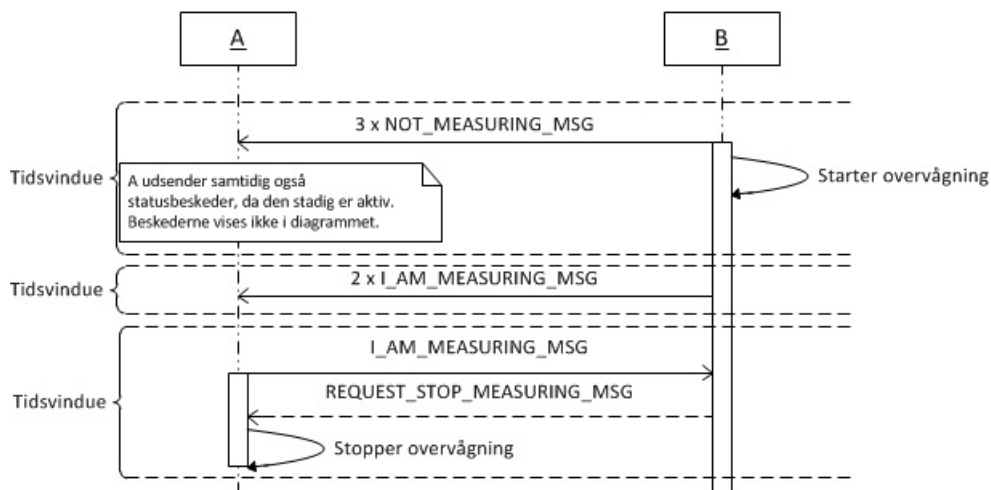
Figur F.4: Sekvensdiagram for test E.2.2



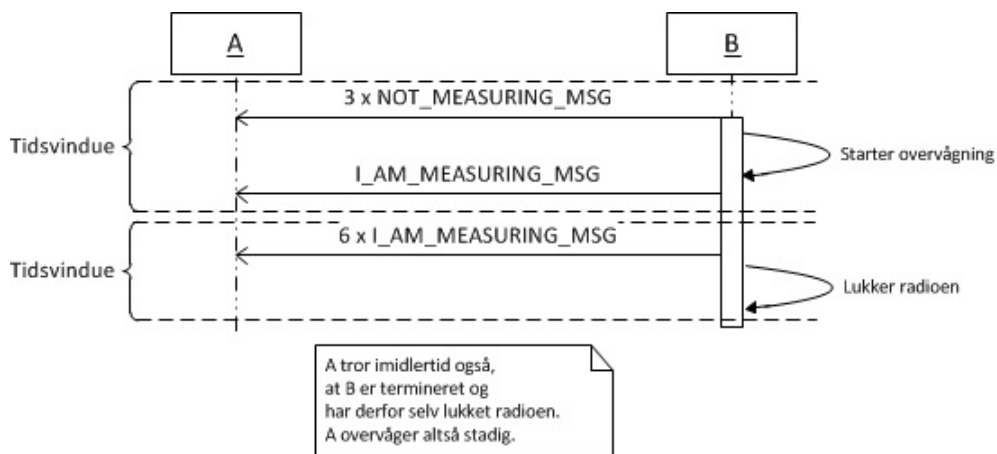
Figur F.5: Sekvensdiagram for test E.3.1



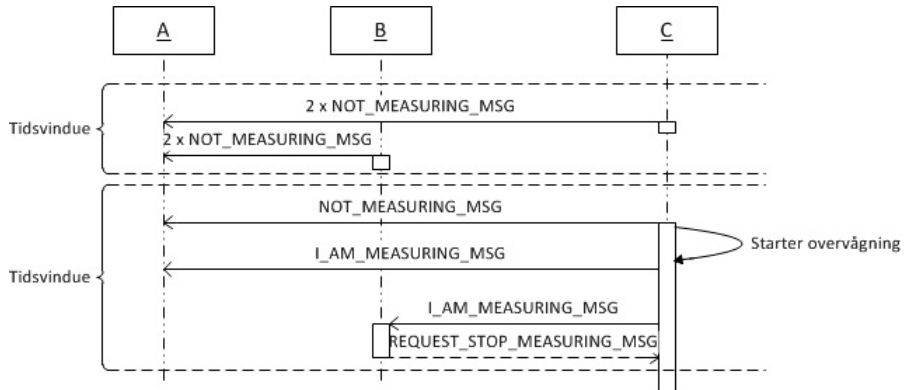
Figur F.6: Sekvensdiagram for test E.3.2



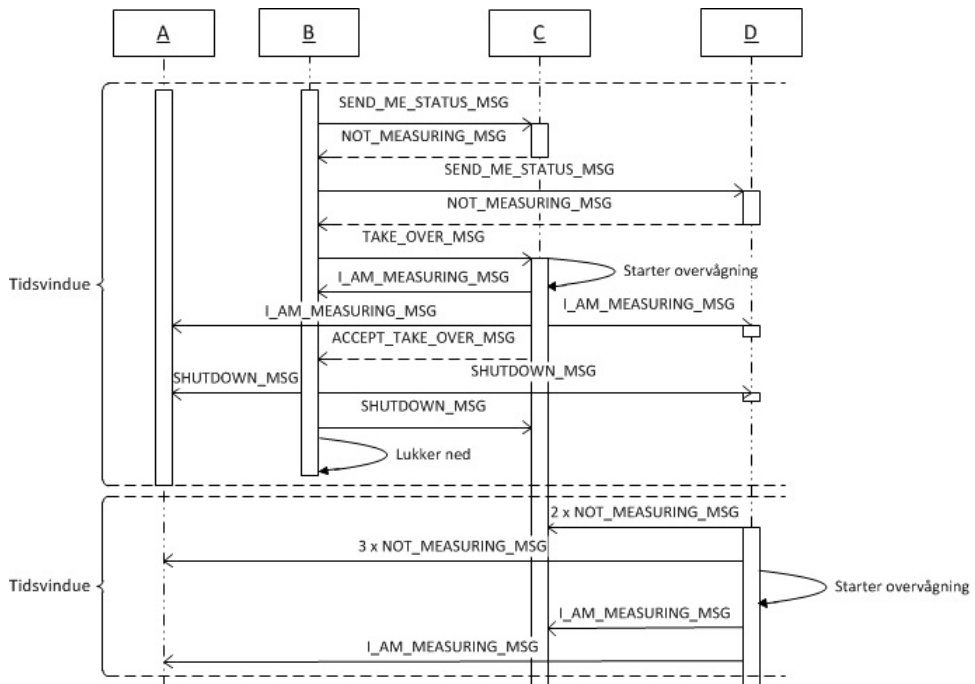
Figur F.7: Sekvensdiagram for test E.4.1



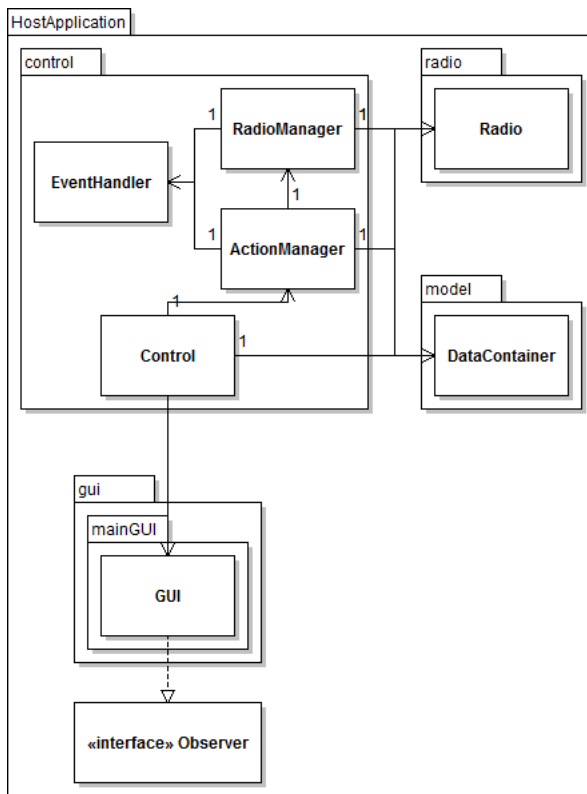
Figur F.8: Sekvensdiagram for test E.4.2



Figur F.9: Sekvensdiagram for test E.5.1



Figur F.10: Sekvensdiagram for test E.5.2



Figur F.11: Klasse- og pakkediagram for værtsprogrammet

Litteratur

- [1] Gregory R. Andrews. *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, 2000.
- [2] Martin Fowler. *UML Distilled*. Addison-Wesley, 3. edition, 2004.
- [3] Ron Goldman. Using the spot accelerometer. <http://www.sunspotworld.com/docs/AppNotes/AccelerometerAppNote.pdf>, september 2010.
- [4] JFreeChart. <http://www.jfree.org/jfreechart/>.
- [5] Sun Labs. Sun spot - theory of operation. <http://http://sunspotworld.com/docs/Yellow/SunSPOT-TheoryOfOperation.pdf>, juni 2009.
- [6] Sun Labs. Sun spot - programmer's manual. <http://sunspotworld.com/docs/Yellow/SunSPOT-Programmers-Manual.pdf>, november 2010.
- [7] Wikipedia. Model-view-controller. <http://en.wikipedia.org/wiki/Model-view-controller>, maj 2011.
- [8] Wikipedia. Sensor node. <http://en.wikipedia.org/wiki/Motes>, april 2011.
- [9] Wikipedia. Singleton pattern. http://en.wikipedia.org/wiki/Singleton_pattern, juni 2011.