



Høgskulen
på Vestlandet

Smarthusimplementasjon basert på HomeSeer

<Magnus F. Sande>
<Kristoffer M. Rødne>

28. mai. 2019

Dokumentkontroll

<i>Rapportens tittel:</i> Smarthusimplementasjon basert på HomeSeer HomeSeer-based smart home implementation	<i>Dato/Versjon</i> 28. mai. 2019/1.0
	<i>Rapportnummer:</i> BO19E-30
<i>Forfatter(e):</i> Magnus F. Sande Kristoffer M. Rødne	<i>Studieretning:</i> HKOM16/HEEL15
	<i>Antall sider m/vedlegg</i> 83
<i>Høgskolens veileder:</i> Thomas Aasebø	<i>Gradering:</i> Åpen
<i>Eventuelle Merknader:</i> Vi tillater at oppgaven kan publiseres.	

<i>Oppdragsgiver:</i> Høgskulen på Vestlandet	<i>Oppdragsgivers referanse:</i> Thomas Aasebø
<i>Oppdragsgivers kontaktperson(er) (inkludert kontaktinformasjon):</i> Thomas Aasebø, Thomas.Aasebo@hvl.no	

Revisjon	Dato	Status	Utført av
0.1	31.01.19	Forstudie	Magnus/Kristoffer
0.2	31.03.19	Midtveis	Magnus/Kristoffer
0.3	15.05.19	Utkast	Magnus/Kristoffer
0.4	27.05.19	Korrekturlesing	Magnus/Kristoffer
1.0	28.05.19	Endelig rapport	Magnus/Kristoffer

Forord

Denne oppgaven er skrevet i forbindelse med våre avsluttende bachelorutdanninger i kommunikasjon og elektronikk våren 2019. Oppgaven er gitt av Høgskulen på Vestlandet i forbindelse med oppretting av en tverrfaglig lab mellom helse og elektro. Det har vært et spennende og lærerikt prosjekt, som har gitt oss god innsikt i hvordan smarthussystemer fungerer.

Vi vil først og fremst takke vår veileder Thomas Aasebø for god veiledning, og høgskolen for lån av mesteparten av komponentene vi har brukt i oppgaven. Deretter vil vi takke fedrene våre, Kjetil og Tor Inge, for lån av komponenter og hjelp til utforming av demonstrerings-modell, samt øvrig familie for støtte gjennom hele semesteret. På grunn av oppgavens natur har vi måttet se oss til diverse forum, og vi vil i den sammenheng også takke brukerne på både Hjemmeautomasjon og HomeSeer sine forumer for deres idéer og kunnskap om systemet.

Sammendrag

Vi har fått i oppgave å prøve ut smarthussystemet HomeSeer for å finne ut om det er et bra system for skolen å bruke i en tverrfaglig lab i fremtiden. Et HomeSeer-system kan bli satt sammen på flere ulike måter, HomeSeer har selv 3 faste kombinasjoner av hardware og software, men det er også mulig å kun kjøpe softwaren og bruke egen datamaskin til å kjøre systemet. Kravene som ble stilt til oss var at vi skulle bestemme oss for en versjon av HomeSeer, samt at vi måtte implementere ulike sensorer, lysstyring og stemmestyring.

I denne oppgaven vil vi gå gjennom alt fra å sette opp HomeSeer på vår valgte maskin, Raspberry Pi, til hvordan man setter opp automatisk batterivarsel når batteriet på en av enhetene er under x antall prosent. Det sies fra HomeSeer sin side at man kan gjøre nesten hva man vil med systemet deres og det er ikke tull, men det er en liten hake; man må være over gjennomsnittet datakyndig for å få det til. Systemet i seg selv er relativt lett å sette seg inn i, men sammenlignet med andre smarthussystemer så er nok brukervennligheten lavere. Vi har derfor valgt å gå i detalj for hvordan man setter opp hele systemet, dette tas opp i kapittel 4.

Vår oppgave har gått ut på å finne ut mulighetene med systemet og hvilke begrensninger systemet har. Når det gjelder muligheter så nevnte vi over at det er egentlig egen kompetanse som setter begrensninger for hva systemet kan gjøre. Det er derimot noen begrensninger man som bruker ikke kan gjøre så veldig mye med. I kapittel 6 diskusjon, tar vi for oss systemet og diverse problemer vi har hatt. Et gjennomgående problem i bransjen er at produsentene av smarthushenheter ikke frigir oppdateringsfilene til enhetene sine, noe som gjør at man må ha produsentens egen «hovedkontroller» for å kunne oppdatere til siste versjon på enhetene sine. Dette ligger da ikke på HomeSeer, da det ville vært et problem uansett hvilket system man ville brukt. Et problem som derimot ligger på HomeSeer er at man med tar i bruk såkalte plug-ins. Plug-ins er utvidelser av systemet og er i veldig stor grad laget av privatpersoner. Dette fører da til at hvis utvideren slutter å bruke systemet, så vil oppdateringer av utvidelsen slutte, noe som igjen vil være en begrensning av systemet. Hvis man da er datakyndig kan man med litt arbeid lage sine egne plug-ins hvis man er redd for at utviderne skal slutte å komme med oppdateringer. Det at utvidelsene er skrevet av privatpersoner bærer også en viss risiko med seg, for det er ikke garantert at enhetene vil fungere med plug-in-en, da det ikke er produsenten selv som står for utvidelsen. Vi har vært borti et eksempel hvor vi fikk til å skifte farge og lysstyrke på en lypære mellom Philips Hue og Google Home. Når vi derimot gikk fra Google Home til HomeSeer, og så til Philips Hue, klarte vi kun å gjøre en av delene. Dette er et eksempel på hvordan ting fungerer direkte fra produsent, men når en tredjepart prøver på det samme så oppnås nødvendigvis ikke samme resultatet.

I kapittel 7 tar vi for oss hva vi har oppnådd med systemet, og hvilke ting man kan gjøre for å utvide oppgaven.

Innhold

Dokumentkontroll	2
Forord	3
Sammendrag	4
1 Innledning	11
1.1 Organisering av rapporten	11
1.2 Oppdragsgiver	11
1.3 Problemstilling	12
1.3.1 Oppdragsgivers problemstilling	12
1.3.2 Om smarthus	12
1.3.3 Eksisterende system	12
1.3.4 Vår oppgave	12
1.4 Hovedidé for løsningsforslag	12
1.5 Avgrensning	12
2 Kravspesifikasjon	14
3 Analyse av problemet	15
3.1 Software	15
3.1.1 HS3-Pi	15
3.1.2 HS3	15
3.1.3 HS3PRO	15
3.2 Hardware	16
3.2.1 HomeTrollers	16
3.2.2 Raspberry Pi	16
3.2.3 Datamaskin	16
3.3 Konklusjon	17
4 Realisering av valgt løsning	18
4.1 Oppsett av Pi og HomeSeer	18
4.1.1 Installasjon på SD-kort	18
4.1.2 Installasjon av RaZberry-kort	18
4.1.3 Førsteoppstart av HomeSeer	18
4.1.4 Regionsoppsett og -innstillinger	19
4.1.5 Øvrige innstillinger	22
4.2 Gjennomgang av menyer	23
4.3 Plug-ins	24

4.4	Oppkobling	24
4.4.1	Z-Wave.....	24
4.4.2	Hue	26
4.4.3	Sonos	27
4.4.4	Google Home.....	27
4.5	Device list / Home-meny	27
4.5.1	Filtre og organisering.....	28
4.6	Avanserte enhetsinnstillinger.....	29
4.6.1	Configuration.....	29
4.6.2	Advanced	30
4.6.3	Status Graphics.....	31
4.6.4	Plug-in-spesifikk.....	33
4.7	Ekstern tilgang.....	36
4.7.1	Secure shell (SSH) via kommandolinje	36
4.7.2	Tilgang via PuTTY	37
4.7.3	Filoverføring via SFTP (SSH File Transfer Protocol)	38
4.8	Virtuelle enheter	40
4.8.1	Oppsett.....	40
4.8.2	Profiler	41
4.9	Events og scripts.....	42
4.9.1	Triggere.....	42
4.9.2	Action.....	42
4.9.3	Sonos-event for lydeffekter.....	43
4.10	Laveste batteri.....	45
4.10.1	Scriptknapper	48
4.10.2	Events	50
4.11	Vårt system.....	52
5	Testing	54
5.1	Logikk.....	54
5.2	Plug-in.....	54
5.2.1	Chromecast.....	55
5.2.2	Yr Weather	55
5.2.3	WeatherXML.....	55
5.2.4	UltraNetatmo3	55

5.2.5	JowiHue	55
5.2.6	Sonos	55
6	Diskusjon	57
6.1	Redegjøring av fremdrift	57
6.2	Redegjøring av risikoanalyse	57
6.3	Problemer med Linux og oppsett	57
6.3.1	Pi-problemer.....	57
6.3.2	Z-Wave.....	58
6.3.3	ZigBee	59
6.4	Problemer hardware og software	59
6.4.1	Google Home	59
6.4.2	Jowi Hue	59
6.4.3	Stemmestyring av lys.....	60
6.5	Utfordringer med selvlagde ting	60
6.5.1	Scriptknapper	60
6.5.2	Laveste batteri - Script.....	61
6.5.3	E-post.....	62
6.5.4	Netatmo batteri.....	64
6.5.5	EnTur	65
7	Konklusjon	68
8	Referanser	70
9	Appendiks	73
Appendiks A	Forkortelser og ordforklaringer	73
Appendiks B	Prosjektledelse og styring.....	74
B.1	Prosjektorganisasjon	74
B.2	Fremdriftsplan	75
B.3	Risikoliste.....	76
Appendiks C	Brukerdokumentasjon.....	77
Appendiks D	Bill Of Materials	79
Appendiks E	Protokoller	82

Figur 1.1 Høgskulen på Vestlandet.....	11
Figur 4.1 Lokalisasjonsinnstillinger	19
Figur 4.2 Endring av lokasjon.....	20
Figur 4.3 Norsk lokasjon	20
Figur 4.4 Lokasjon HomeSeer	21
Figur 4.5 Lokasjonsendring HomeSeer	21
Figur 4.6 PuTTY error etter restart	22
Figur 4.7 Tilpasning av filterinnstillinger	23
Figur 4.8 Skjerm bilde av hjem-skjermen.....	23
Figur 4.9 Installere Z-Wave plug-in	24
Figur 4.10 Installert og aktivert Z-Wave-plug-in.....	24
Figur 4.11 Z-Wave «Controller Management».....	25
Figur 4.12 Legg til en node	25
Figur 4.13 Knappen er under batteridekselet	26
Figur 4.14 Philips-logoen er en knapp.....	26
Figur 4.15 Fjern eller legg til bridge og lyspærer.....	26
Figur 4.16 Prosentverdier for lysstyrke på pærene.....	27
Figur 4.17 To komponenter med tilhørende subkomponenter	28
Figur 4.18 Tilpasning av filter	28
Figur 4.19 Egenskaper for alle valgte komponenter	29
Figur 4.20 Hvordan komme inn på enhetsinnstillinger	29
Figur 4.21 Konfigurasjon.....	30
Figur 4.22 Avansert.....	30
Figur 4.23 Enhet med "Status" og "Control"-kolonner	31
Figur 4.24 Status Graphics.....	31
Figur 4.25 Statusverdier	31
Figur 4.26 Status Graphics norsk.....	32
Figur 4.27 Matrise	33
Figur 4.28 Nodeinformasjon.....	33
Figur 4.29 Enhetsinnstillinger Z-Wave.....	34
Figur 4.30 Parametere fra en motionsensor	35
Figur 4.31 Ledetekst i Windows	36
Figur 4.32 Terminal i MacOS	36
Figur 4.33 PuTTY SSH tilkobling	37
Figur 4.34 PuTTY ledetekst	37
Figur 4.35 WinSCP med SFTP-tilkobling	38
Figur 4.36 WinSCP utseende	39
Figur 4.37 WinSCP bilder mappe.....	40
Figur 4.38 Virtuell enhet lagring	40
Figur 4.39 Virtuell enhet konfigurasjon.....	41
Figur 4.40 Enkel event med if-setning og en kommando.....	42
Figur 4.41 Event som kjører et script	43
Figur 4.42 Velg for å laste opp lydfiler.....	43
Figur 4.43 Opplastingsmeny	44
Figur 4.44 Meny for linkgruppe i Sonos-plug-in-en.....	44

Figur 4.45 Å legge til høyttaler krever spesiell syntaks	44
Figur 4.46 Prioritert event	45
Figur 4.47 Laveste batteri-konfigurasjon	45
Figur 4.48 Laveste batteri Status Graphics 1.....	46
Figur 4.49 Laveste batteri Status Graphics 2.....	46
Figur 4.50 Laveste batteri Status Graphics 3.....	47
Figur 4.51 Laveste batteri Status Graphics 4.....	48
Figur 4.52 Laveste batteri før scriptknapper	48
Figur 4.53 Scriptknapp oppretting.....	49
Figur 4.54 Laveste batteri med oppdaterte knapper	49
Figur 4.55 Batteri - Varsel og rapport fullstendig.....	50
Figur 4.56 Batteri - Varsel og rapport kollapset	50
Figur 4.57 Batterirapport - opprinnelig, Gmail.....	51
Figur 4.58 Batterirapport - med batteritype, Gmail.....	51
Figur 4.59 Batteri - Oppdatering fullstendig	52
Figur 4.60 Batteri - Oppdatering kollapset.....	52
Figur 6.1 Linuxtools, brukernavn.....	58
Figur 6.2 Node – fjerning.....	58
Figur 6.3 Devicescriptbutton_List, fra sparkman	61
Figur 6.4 Devicescriptbutton_List, logg	61
Figur 6.5 Batterirapport - med batteritype, Gmail.....	61
Figur 6.6 Laveste batteri - opprinnelig	62
Figur 6.7 Laveste batteri – endringer	62
Figur 6.8 E-post, event.....	63
Figur 6.9 E-post, feilmelding.....	63
Figur 6.10 E-post, oppsett	64
Figur 6.11 E-post, logg	64
Figur 6.12 Netatmo batteri, event.....	65
Figur 6.13 Netatmo batteri, advanced	65
Figur 6.14 Netatmo batteri, hjemskjerm.....	65
Figur 6.15 Skyss	66
Figur 6.16 Kolumbus.....	66
Figur 6.17 Entur, Kolumbus hjemskjerm	66
Figur 6.18 Entur, Skyss hjemskjerm.....	67
Figur 9.1 Prosjektorganisering.....	74
Figur 9.2 Opprinnelig fremdriftsplan.....	75
Figur 9.3 Endelig fremdriftsplan	75

Tabell 3.1 HomeSeer software	15
Tabell 3.2 HomeSeer hardware	16
Tabell 8.1 Risikoanalyse.....	76
Tabell 8.2 Keyfob	77
Tabell 8.3 Stemmestyring.....	77
Tabell 8.4 Multisensor	78
Tabell 8.5 Dør- og vindussensor	78
Tabell 8.6 Stikkontakt	78

1 Innledning

1.1 Organisering av rapporten

Rapporten er grovt sett delt inn i tre deler. I første del tar vi for oss hvem arbeidsgiver er, bakgrunnen for oppgaven og kravspesifikasjonen. Her finner vi også en analyse av problemet, som inneholder informasjon om de ulike måtene det er mulig å sette opp et system med HomeSeer på. I andre del går vi grundig gjennom hvordan systemet fungerer og hvordan vi har satt opp systemet. Vi går gjennom grunnleggende oppsett av systemet og avansert oppsett med eksempler fra vårt system. Til slutt i del to går vi gjennom hvordan det endelige systemet vårt er satt opp, inkludert en kort gjennomgang av de ulike plug-in-ene vi har tatt i bruk. I tredje del vil vi først gjøre rede for fremdriftsplanen og risikoene til oppgaven. Vi vil deretter diskutere ulike problemer vi har hatt med systemet, før vi til slutt avslutter med en konklusjon med vår vurdering av systemet og mulighetene for videre arbeid.

Vi vil helt bakerst i rapporten ha appendikser som omhandler ordforklaringer, prosjektledelse og styring, brukerdokumentasjon, brukte komponenter og protokoller.

Vi henviser til kilder ved hjelp av referansestilen Institute of Electrical and Electronics Engineers (IEEE), som har følgende utseende: [1]. Figurer og tabeller vil ha format bestående av kapittelnummer først og deretter tabell/figurnummer. Dette vil ha følgende utseende: 1.1, 2.1, 3.1. Der hvor vi har brukt HomeSeer-terminologi vil det bli satt i hermetegn, slik som dette: «events», «functions».

1.2 Oppdragsgiver

Oppdragsgiveren vår er Høgskulen på Vestlandet. 1. Januar 2017 fusjonerte Høgskolen i Bergen, Høgskulen i Sogn og Fjordane og Høgskolen Stord/Haugesund og dannet Høgskulen på Vestlandet. Skolen ble med det en av de største utdanningsinstitusjonene i landet, med over 16 000 studenter [1] fordelt over fem campuser. Ambisjonene med denne fusjonen var å bli et universitet med en profil som er profesjons- og arbeidslivsrettet. Som student ved skolen kan man ta alt fra årsstudium til doktorgrad.

Instituttet som gav oss denne oppgaven var institutt for elektrofag, en enhet under fakultetet for ingeniør- og naturvitenskap. Fakultetet har 3 600 studenter fordelt på hvert av skolens fem campuser [2], mens institutt for elektrofag har studenter ved fire av dem. Utdanninger man kan ta på instituttet er: automatisering med robotikk, elektronikingeniør, elkraftteknikk, kommunikasjonssystem (master) [3].

Opgaven vår kommer fra institutt for elektrofag, campus Bergen. Campusen er foreløpig delt i to, med 6000 studenter på Kronstad og 2000 studenter i Møllendalsveien. Campus Kronstad er bygget på NSBs gamle verkstedtomt og var klar til bruk høsten 2014. I 2020 vil nok et nytt bygg være klar til bruk, og dermed vil man få alle studenter ved campus Bergen samlet på en og samme plass [4].



Figur 1.1 Høgskulen på Vestlandet

1.3 Problemstilling

1.3.1 Oppdragsgivers problemstilling

Høgskulen på Vestlandet har overtatt tomten til UiBs gamle odontologbygg på Årstad, og holder nå på å bygge ett nytt bygg der. Her er det tiltenkt en tverrfaglig lab tilhørende blant annet prosjektet Teknologi i helse og omsorg. Denne labben er en del av en etter- og videreutdanning for personer med bachelor innen helse eller sosialfaglig profesjonsutdanning. Ved endt videreutdanning skal studenten ha styrket sin kompetanse innen anskaffelser, bruk og konsekvenser av omsorgsteknologi [5]. Som et ledd i utviklingen av denne labben ønsker skolen at vi skal teste ut og dokumentere bruk av smarthussystemet HomeSeer.

1.3.2 Om smarthus

Det finnes ingen eksakt definisjon av smarthus og det kan dermed defineres på flere måter. Vi ser på smarthus som teknologien inni huset og ikke huset i seg selv, med andre ord kan et gammelt hus være et smarthus. Det er styring av hverdagslige ting ved hjelp av datamaskin, med formål om lavere energiforbruk, økt sikkerhet og bekvemmelighet. Man kan også si at det finnes smarthus av ulike grader. Noen ønsker kanskje at man bare skal overvåke om kaffetrakteren er avskrudd, mens andre ønsker et helautomatisert hjem. Teknologiene inne i huset kommuniserer med hverandre via ulike typer kommunikasjonsprotokoller, og blir knyttet opp mot omverdenen via internett. De mest utbredte kommunikasjonsprotokollene for smarthus er Z-Wave, ZigBee, Bluetooth og IP [6].

1.3.3 Eksisterende system

I dag finnes det mange forskjellige smarthussystemer. Man finner alt fra systemer med enkle brukergrensesnitt og plug-and-play-funksjonalitet til systemer hvor det kreves at man må sette seg ordentlig inn i systemet for å få gjort selv enklere ting. Man har også forskjellige typer smarthussystem der noen systemer er begrenset til for eksempel sikkerhet slik som alarmsystemer. Andre er mer frie og kan skreddersys til en brukers behov. Noen kjente smarthussystemer er Futurehome, Apple HomeKit, Fibaro Home Center og HomeSeer.

1.3.4 Vår oppgave

Oppgaven vår går ut på å teste smarthussystemet HomeSeer, et avansert smarthussystem som gir stor fleksibilitet og derfor potensiale for mye funksjonalitet. Det gjør at det kreves god kjennskap til systemet for å utnytte det, og det er derfor mye å lese seg opp på. På grunn av dette er HomeSeer ikke et system som passer for hvem som helst.

1.4 Hovedidé for løsningsforslag

Hovedidéen for løsningen er å teste ut diverse måter man kan bruke HomeSeer på. Vi har tenkt å bruke Z-Wave-sensorer, en Google Home smarthøytaler, Sonos-høytalere, og Philips Hue-belysning. Ved hjelp av disse skal vi automatisere funksjoner som i et hus vanligvis er manuelle. Det skjer ved at vi setter opp forskjellige utløsere (triggers), som, når de utløses, starter en eller flere hendelser (actions). Triggers og actions utgjør til sammen det vi kaller events.

1.5 Avgrensning

Denne oppgaven kan lett bli veldig stor hvis vi ikke avgrenser hva vi vil gjøre, og hva vi kan gjøre hvis vi har tid. Vi starter med et visst antall komponenter vi vil implementere i vårt hjemmesystem, ser vi at det går fort kan vi legge til flere komponenter. Ser vi derimot at det ble litt mer problemer enn først antatt vil vi gå vekk fra noen komponenter.

De kritiske leveringstidene i denne oppgaven er:

- Forstudie – 31.01.2019
- Midtveis presentasjon – 21.03.2019
- Oppgave innlevering – 31.05.2019
- Oppgave presentasjon – 10.05.2019
- Presentasjon – 06.06.2019
- EXPO – 13.06.2019

Vi har 5000 til rådighet, kommer til å bruke minst 3000 av de på hardware og software, men vi kommer nok til å bruke mer også, da alle komponentene vi skal ha inn i systemet vårt koster penger. Så alt i alt kommer vi nok til å bruke de 5000 vi har fått til rådighet.

2 Kravspesifikasjon

- Holde oss innenfor et budsjett 5000kr
- Styringsenhet, enten hardware eller software for HomeSeer
- Sensorer og aktuatorer
- Stemmestyring via Google Home
- Lysstyring via Philips Hue

Samlet forståelse av kravspesifikasjonen er at vi ved hjelp av en styringsenhet fra HomeSeer skal konfigurere ulike events. Eksempler på dette er stemmestyring som styrer lysene i et rom, åpning av dør fører til lyd fra lydenhet(er) og lignende.

3 Analyse av problemet

HomeSeer kommer med tre hardwareløsninger og tre softwareløsninger. Dersom man velger en software-basert løsning står man friere til å velge hardware selv, i motsetning til om man velger en ferdig hardware-pakke hvor software er inkludert.

3.1 Software

Software	HS3-Pi	HS3	HS3PRO
Pris	\$149,95	\$249,95	\$599,95
Plattform	Raspberry Pi	Win/Linux	Win/Linux
Z-Wave(integrasjon)	Ja	Ja	Ja
Google Home(integrasjon)	Ja	Ja	Ja
Tilgjengelige plug-ins	5 Linux	Alle	Alle(inkludert)
Fremtidige plug-ins	Mot betaling	Mot Betaling	Inkludert
HSTouch Designer	\$199,95	\$199,95	Inkludert

Tabell 3.1 HomeSeer software

Tabellen ovenfor sammenligner utvalgte forskjeller mellom de tre softwareløsningene. Plug-ins utvider funksjonaliteten til HomeSeer, der noen er gratis mens andre må man betale for. HSTouch Designer er et program som lar brukeren lage egne brukergrensesnitt til et HomeSeer-system. Du kan lage ett grensesnitt for nettbrett og et annet for mobil, som gir stor fleksibilitet. [7]

3.1.1 HS3-Pi

HS3-Pi er den billigste av de tre softwareløsningene HomeSeer tilbyr. Den koster \$149,95 og bruker Raspberry Pi som plattform [8]. Den store forskjellen på denne og de andre to er at man kun kan ha fem aktiverte plug-ins samtidig.

3.1.2 HS3

HS3 er den mellomste av softwareløsningene. Den koster \$249,95 og kan enten bruke Linux eller Windows som plattform [8]. Hovedforskjellen på denne og HS3-Pi er at man her kan aktivere så mange plug-ins man vil.

3.1.3 HS3PRO

HS3PRO er den dyreste av softwareløsningene. Den koster \$599,95 og kan enten bruke Linux eller Windows som plattform [8]. Den er ellers helt lik HS3 bortsett fra at man her har alle betalte plug-ins inkludert. Dette gjelder også alle fremtidige plug-ins, ikke bare de som er tilgjengelige i dag. Legg også merke til at man her får HSTouch Designer inkludert.

3.2 Hardware

3.2.1 HomeTrollers

HomeTrollers er hardwareløsningene HomeSeer selger. Det finnes tre varianter: Zee S2, SEL og S6 [9].

Modell	Zee S2	SEL	SEL-PRO	S6 PRO
Software	HS3 Pi	HS3	HS3PRO	HS3PRO
Pris	\$209,95	\$399,95	\$699,95	\$1 199,95
Operativsystem	Linux	Linux	Linux	Win 10
Z-Wave(integrasjon)	Ja(stick inkludert i pris)	Ja(stick må kjøpes)	Ja(stick må kjøpes)	Ja(stick må kjøpes)
Tilgjengelige plug-ins	5 Linux	Linux	Linux(inkludert)	Alle(inkludert)
Fremtidige plug-ins	Mot betaling	Mot Betaling	Inkludert	Inkludert
HSTouch Designer	199,95\$	199,95\$	Inkludert	Inkludert
Wifi	Ja		Ja	Ja
RAM	1GB	2GB	4GB	4GB
Lagring	8GB		32GB	120GB

Tabell 3.2 HomeSeer hardware

Z-Wave-integrasjonen skjer via en USB smartstick. Man må påberegne litt ekstra kostnad ved kjøp av denne.

3.2.1.1 Zee S2

Zee S2 er den billigste versjonen og kjører på en Raspberry Pi. Denne kan kun kjøre HS3-Pi-softwaren, og spesifikasjonene vil dermed følge det vi har spesifisert for softwaren tidligere. Operativsystemet som blir brukt her er Linux. Legg merke til at man får et integrert Z-Wave-kort ved kjøp av US-modellen, mens man ved EU-versjonen må kjøpe en Z-Wave USB-stick.

3.2.1.2 SEL

SEL er den andre typen hardware man kan få fra HomeSeer. Denne hardwaren kan kjøre både HS3 og HS3PRO, spesifikasjonene er avhengige av hvilken software man går for. Hovedforskjellen mellom SEL og SEL-PRO er altså softwaren de kjører og at man har 2GB mer RAM med PRO modellen. Dette resulterer selvsagt i at det er to forskjellige priser på SEL og SEL-PRO. Begge modellene kjører operativsystemet Linux. Begge SEL-modellene krever at man kjøper en Z-Wave USB-stick.

3.2.1.3 S6 PRO

S6 PRO er det tredje og siste hardware-alternativet man kan få fra HomeSeer. Her kjøres kun softwaren HS3PRO og operativsystemet er Windows 10. Det er også den desidert dyreste av hardwarene, men det hører med når både hardware og software spesifikasjonene er av ypperste klasse. For å kunne bruke Z-Wave-komponenter må man også her ha et Z-Wave-grensesnitt.

3.2.2 Raspberry Pi

Ved kjøp av software har man muligheten til å skaffe hardware selv. En av mulighetene man har da er å kjøpe seg en Raspberry Pi og installere softwaren på denne. Det er anbefalt å kjøpe HS3-Pi softwaren, men det er mulig å kjøre HS3(og dermed HS3PRO) også på en Raspberry Pi.

3.2.3 Datamaskin

Har man en gammel datamaskin stående kan man bruke denne som hardwaren til softwaren man kjøper. Man kan bruke både Linux- og Windows-datamaskiner, noe som gjør at det ikke er noen

restriksjoner på hvilken software man kan kjøpe. Det er verdt å merke seg at siden dette er sentralen i systemet, så burde det være en maskin som kan stå påskrudd døgnet rundt.

3.3 Konklusjon

Som man sikkert forstår så er det ekstremt mye å tenke på når man skal velge hvordan man skal sette sammen sitt eget system. Man kan gå for de predefinerte fra HomeSeer, eller man kan trikse og mikse litt selv med hardware og software. Her er det rett og slett smak og behag og du selv som velger hvordan systemet ditt skal se ut.

I vår oppgave har vi valgt å bruke en Raspberry Pi, med innebygget Z-Wave-kort og HS3-Pi-softwaren. Grunnen til vi har valgt som vi har gjort er av hensyn til tid og penger. Veileder Thomas hadde en Raspberry Pi med et internt Z-Wave-kort liggende på kontoret som vi kunne få bruke i denne oppgaven, og dermed kunne vi starte på oppgaven med en gang. Dette var også det billigste alternativet når det kom til hardware, da vi ikke trengte å bruke en krone på hardware. Softwaremessig gikk vi for det som var anbefalt for Raspberry Pi, både siden det var anbefalt, men også siden det var den rimeligste løsningen. Ut ifra hvordan vi har tenkt å løse oppgaven, så ser det ikke ut som vi trenger flere enn 5 plug-ins aktivert samtidig på systemet vårt, noe som gjorde valget meget enkelt.

4 Realisering av valgt løsning

Nå skal vi gå gjennom oppsett av systemet, alt fra grunnleggende førsteoppsett til mer spesifikke løsninger som gjelder mer for vårt system. Vi tar for oss blant annet installasjon av HomeSeer på en Raspberry Pi, Z-Wave-kort og komponenter, samt øvrige IP-baserte enheter.

4.1 Oppsett av Pi og HomeSeer

Ved førstegangsoppsett av HomeSeer på Raspberry Pi bruker vi mus, tastatur og skjerm direkte koblet til Pi-en. Senere bruker vi tredjepartsprogrammer for ekstern tilkobling, via lokalnettverket.

4.1.1 Installasjon på SD-kort

For å komme i gang med HomeSeer, må man først installere softwaren på et SD-kort som puttes inn i Raspberry Pi-en. Til det trenger man en plateavbildningsfil [10] som inneholder HomeSeer, og programmet [11] som skriver filen til SD-kortet. Etter å ha kjørt Etcher kan minnekortet settes i, og HomeSeer vil deretter starte opp automatisk.

4.1.2 Installasjon av RaZberry-kort

For å kunne bruke Z-Wave trenger man enten et utvidelseskort eller en USB-sender (dongle) som kan sende og motta Z-Wave-signaler. Vi har i vårt oppsett utvidelseskortet RaZberry, som installeres oppå Raspberry Pi-en.

Ut av boksen fungerer ikke RaZberry-kortet med resten av installasjonen, så man må skrive inn noen kommandoer i Linux-kommandolinjen på Raspberry Pi-en:

1. Tast inn kommandoen **`sudo apt-get update`**. Denne sjekker om det er pakkeoppdateringer som kan lastes ned, i tilfelle noe er utdatert.
2. Deretter tast inn **`sudo apt-get upgrade`**. Denne kommandoen installerer de eventuelle oppdateringene funnet i punkt 1.

Nå er selve systemet oppdatert, men vi har fortsatt ikke installert softwaren som muliggjør bruk av Z-Wave-kortet. For å fikse dette installerer vi softwaren som heter Z-Way, med hjelp fra en bruker på Hjemmeautomasjon.no [12]:

1. Tast inn kommandoen **`wget -q -O - razberry.z-wave.me/install/v2.2.4 | sudo bash`**.
2. Start så systemet på nytt med **`sudo reboot`**.

Nå skal det være mulig å bruke Z-Wave-kortet til å koble til komponenter.

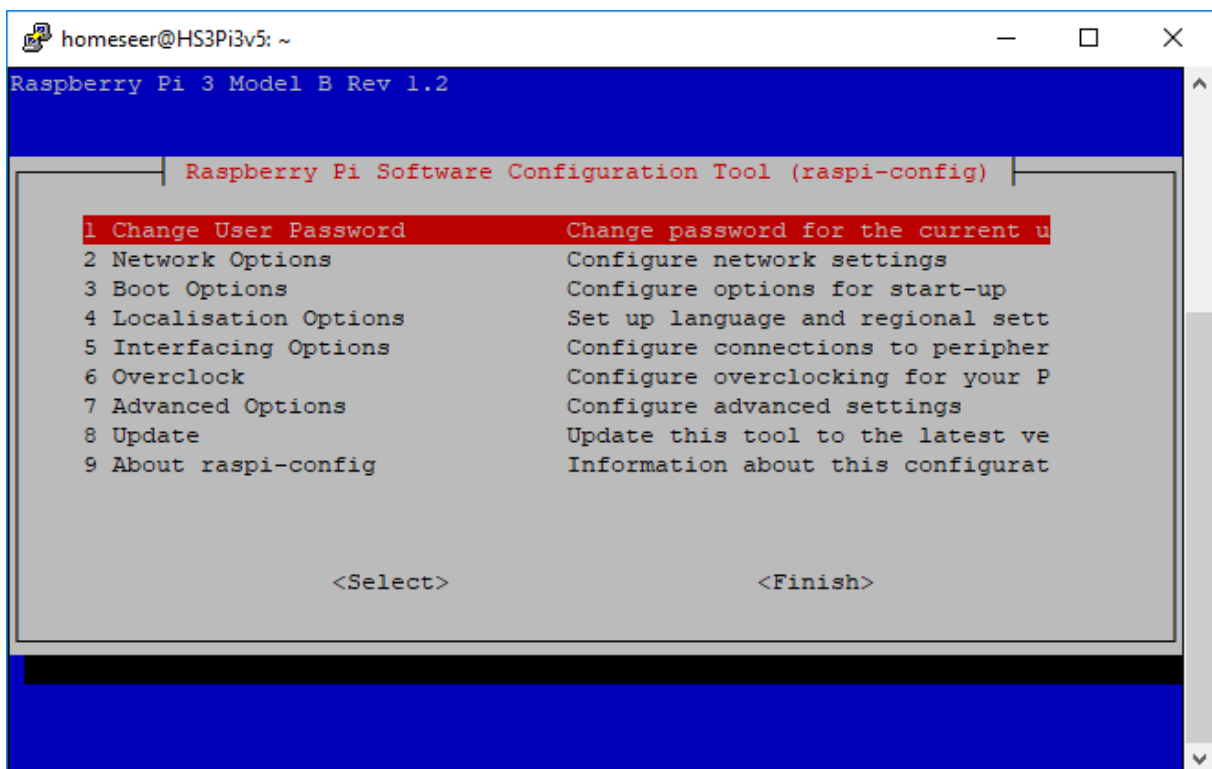
4.1.3 Førsteoppstart av HomeSeer

For at vi skal få til førsteoppsettet av HomeSeer er det viktig at vi er på samme nettverk som Pi-en. Når vi er på samme nettverk skal vi inn på følgende url: **`find.homeseer.com`**. Her vil vi finne én enhet vi kan trykke på. Vi vil da komme videre til et førsteoppsett av HomeSeer. Her vil vi bli fortalt at vi må lage brukernavn og passord og registrere softwaren (informasjonen man trenger blir sendt på mail). Det første vi gjør er å lage passordet til HomeSeer, altså innloggingspassordet til brukergrensesnittet. Det andre er passordet til «homeseer»-brukeren på Linuxsystemet. Når dette er gjort er vi klare til å begynne å sette opp systemet slik vi vil ha det, for nå er vi endelig kommet inn i HomeSeer.

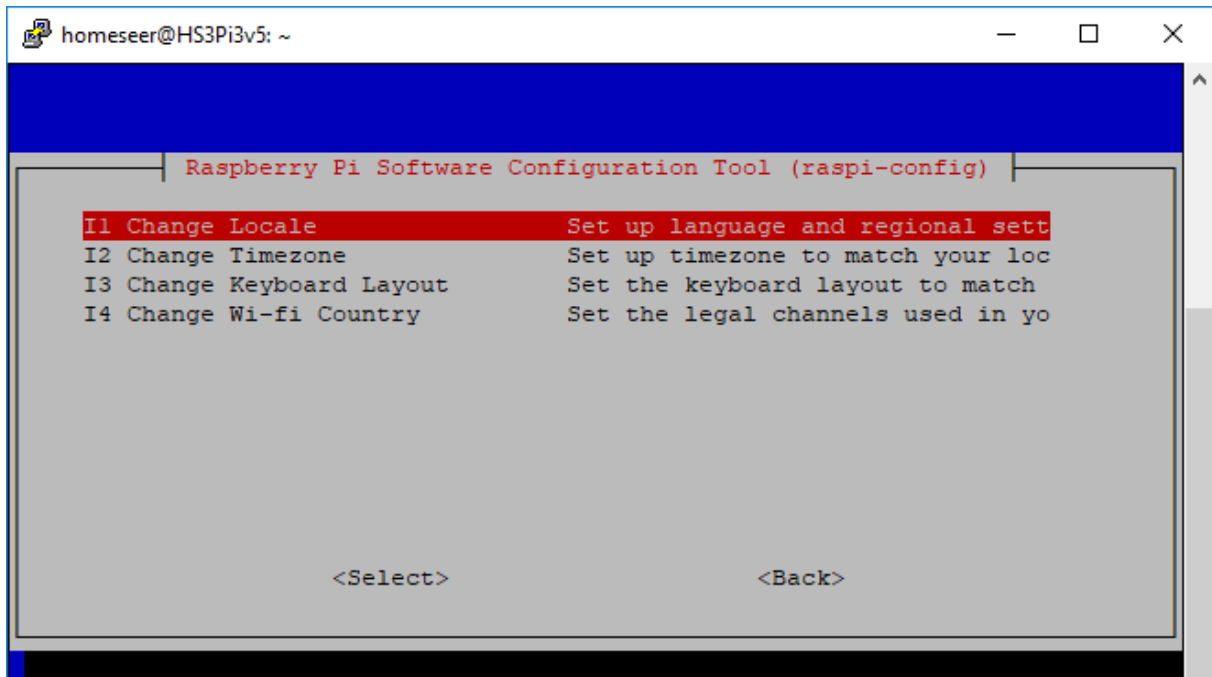
4.1.4 Regionsoppsett og -innstillinger

Pi-en kommer som standard med amerikansk regionsoppsett, noe som gjør at vi vil få alt av klokkeslett og lignende på amerikansk (AM/PM) og alt av datoer på amerikansk form. Dette er ikke noe problem i seg selv, men vi syntes det kunne være greit å få hele Pi-en over til et norsk regionsoppsett. På denne måten slipper vi å tenke oss om to ganger når vi for eksempel skal sette et tidspunkt eller dato for et event. For å gjøre dette må vi skrive inn noen linjer i Linux-kommandolinjen. Vi har valgt å gjøre det via PuTTY.

1. Kobler oss til Pi-en via PuTTY som vist i kapittel 4.7.2 i Figur 4.33
2. Skriver inn «homeseer» som brukernavn og det passordet vi lagte i forrige delkapittel
3. Skriver inn kommandoen ***sudo raspi-config***
4. Velger alternativ 4 – Localisation Options som vist i Figur 4.1
5. Velger alternativ 1 – Change Locale som vist i Figur 4.2

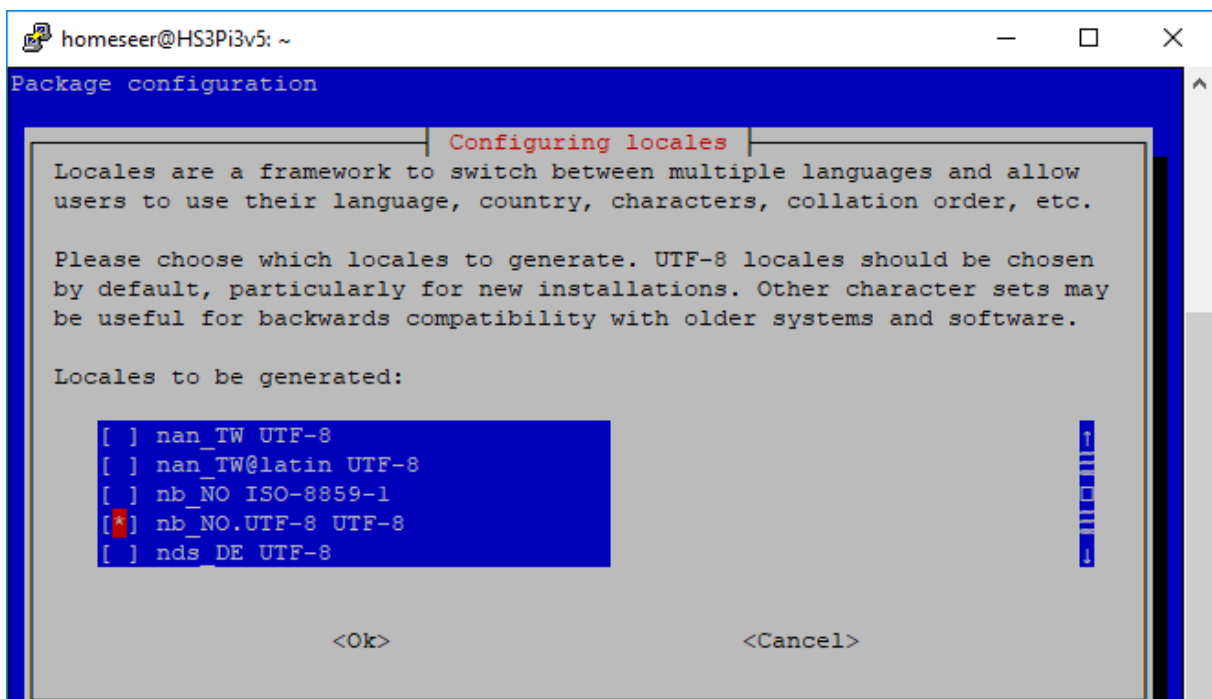


Figur 4.1 Lokalisasjonsinnstillinger



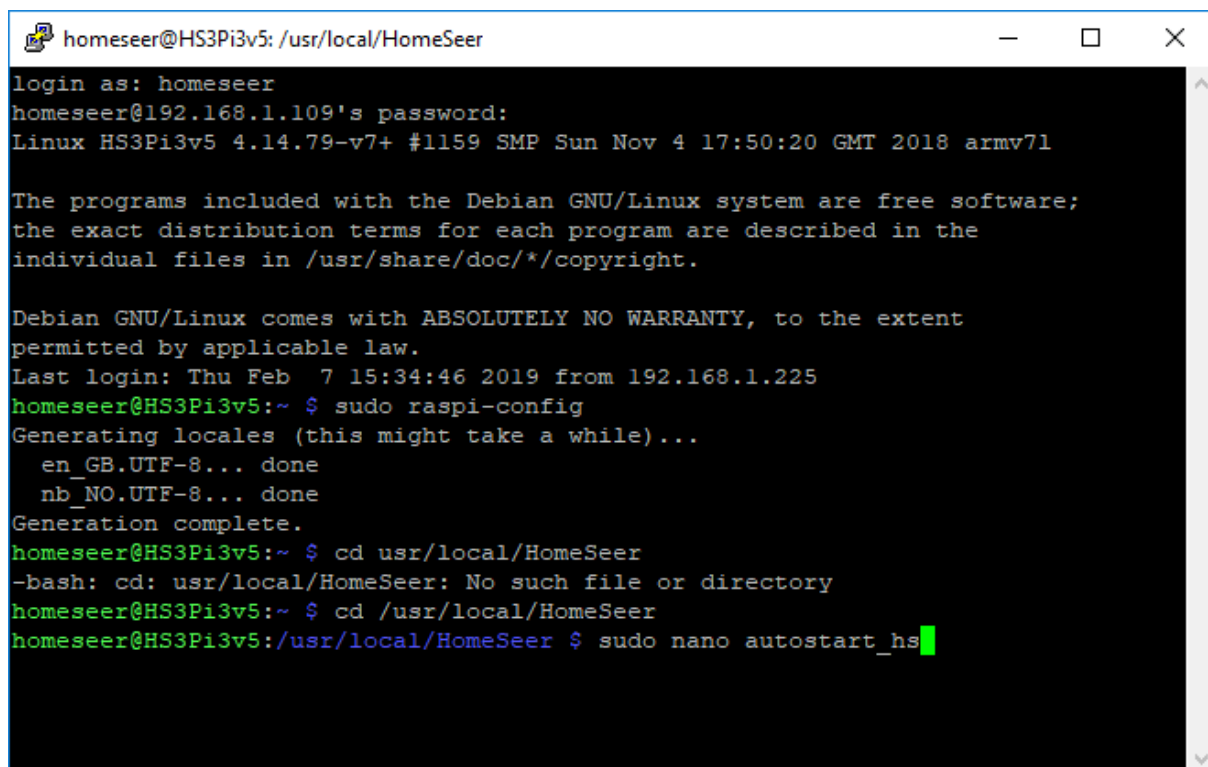
Figur 4.2 Endring av lokasjon

6. Blar til vi finner **nb_NO.UTF-8 UTF-8**. Trykke på mellomromstasten for å merke den og deretter tabulator for å kunne trykke OK.
7. Velger en_GB som default for systemet, kunne sikkert valgt norsk her, men vi valgte å ha en_GB som default.



Figur 4.3 Norsk lokasjon

8. Skriver inn kommandoen **cd /usr/local/HomeSeer** som vist i Figur 4.4
9. Skriver inn kommandoen **sudo nano autostart_hs** som vist i Figur 4.4



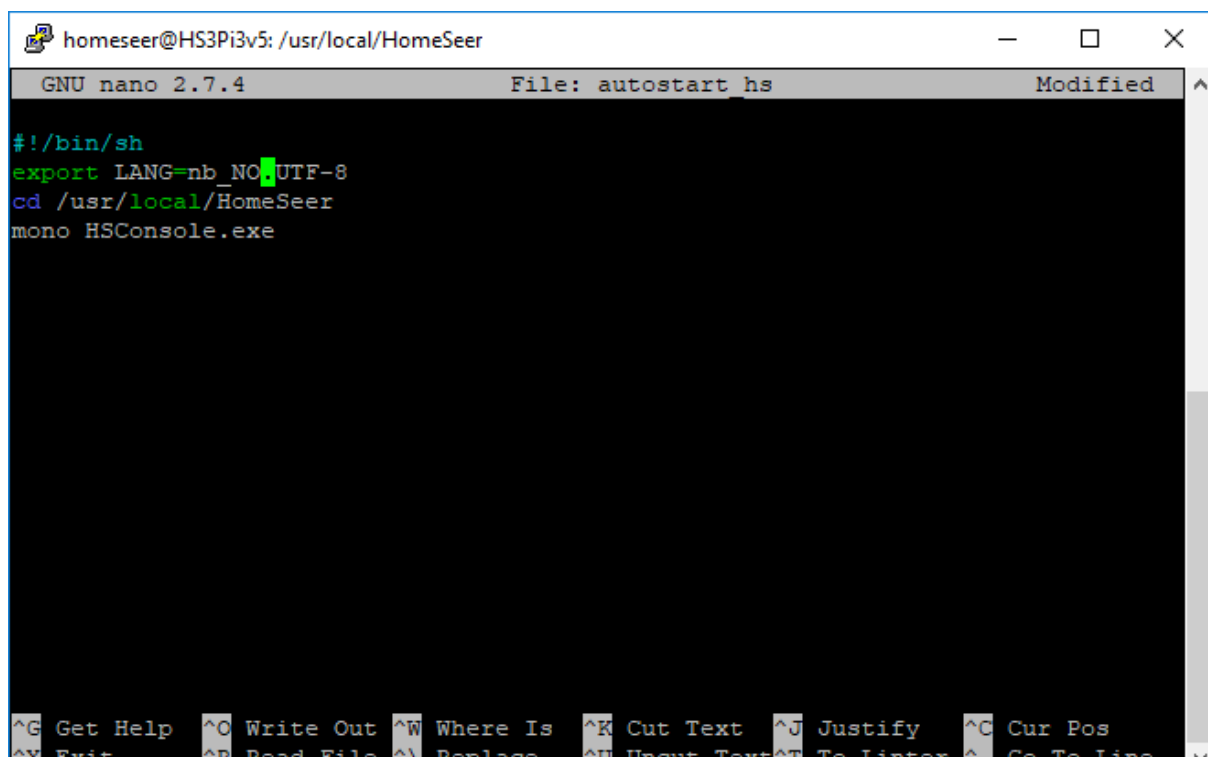
```
homeseer@HS3Pi3v5: /usr/local/HomeSeer
login as: homeseer
homeseer@192.168.1.109's password:
Linux HS3Pi3v5 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb  7 15:34:46 2019 from 192.168.1.225
homeseer@HS3Pi3v5:~ $ sudo raspi-config
Generating locales (this might take a while)...
  en_GB.UTF-8... done
  nb_NO.UTF-8... done
Generation complete.
homeseer@HS3Pi3v5:~ $ cd usr/local/HomeSeer
-bash: cd: usr/local/HomeSeer: No such file or directory
homeseer@HS3Pi3v5:~ $ cd /usr/local/HomeSeer
homeseer@HS3Pi3v5:/usr/local/HomeSeer $ sudo nano autostart_hs
```

Figur 4.4 Lokasjon HomeSeer

10. Får opp en skjerm som vist i Figur 4.5. Her skal vi endre LANG= til **nb_NO.UTF-8**. Denne har stod på **_US** for oss.
11. Trykker så på ctrl + x for å lagre, må deretter trykke y og enter for å komme tilbake til Linux kommandolinjen.



```
homeseer@HS3Pi3v5: /usr/local/HomeSeer
GNU nano 2.7.4 File: autostart_hs Modified
#!/bin/sh
export LANG=nb_NO.UTF-8
cd /usr/local/HomeSeer
mono HSConsole.exe

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line
```

Figur 4.5 Lokasjonsendring HomeSeer

12. Skriver inn kommandoen **sudo reboot** vist i Figur 4.6. Merk at vi her vil få en feilmelding som vist i figuren hvis man er på fjerntilkobling, noe som gir mening da vi restarter systemet.

```

homeseer@HS3Pi3v5: /usr/local/HomeSeer
login as: homeseer
homeseer@192.168.1.109's password:
Linux HS3Pi3v5 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l

The programs included with the
the exact distribution terms
individual files in /usr/shar

Debian GNU/Linux comes with A
permitted by applicable law.
Last login: Thu Feb 7 15:34:
homeseer@HS3Pi3v5:~ $ sudo ra
Generating locales (this might take a while)...
  en_GB.UTF-8... done
  nb_NO.UTF-8... done
Generation complete.
homeseer@HS3Pi3v5:~ $ cd usr/local/HomeSeer
-bash: cd: usr/local/HomeSeer: No such file or directory
homeseer@HS3Pi3v5:~ $ cd /usr/local/HomeSeer
homeseer@HS3Pi3v5:/usr/local/HomeSeer $ sudo nano autostart_hs
homeseer@HS3Pi3v5:/usr/local/HomeSeer $ sudo reboot
  
```

Figur 4.6 PuTTY error etter restart

Etter denne restarten av systemet vil man nå se norsk format på både klokkeslett og datoer inne i menyene på HomeSeer.

4.1.5 Øvrige innstillinger

Inne i HomeSeer er det noen innstillinger man gjerne vil endre for å gjøre ting enklere for oss etter hvert som systemet blir større og mer komplisert. I toppmenylinjen velger man først «Tools», og deretter «Setup».

Det første å sjekke er om systemet er skikkelig oppdatert, med «Check for update»-knappen øverst på siden. Deretter er det lurt å endre geografisk lokasjon, og sjekke at temperaturenheten er celsius, ikke fahrenheit.

I «Network»-fanen, sjekk at remote access er slått på (hvis det er ønskelig å kunne logge på eksternt). Her har man også muligheten til å sette en statisk ip-adresse til enheten vår slik at vi slipper å bruke myhs.homeseer.com for å kunne logge inn eksternt. Dette kan være nyttig også internt i huset, da man kan oppleve at enheten vil skifte IP-adresse (man bruker denne URL-en til å koble seg på via nettleser, på samme måte som man kobler seg på en ruter) hvis ruterens resettes. Hvis man velger å koble til via myhs.homeseer.com vil man kunne logge seg på ved hjelp av mailen man har registrert seg med og tilhørende passord.

«E-Mail»-fanen kan brukes til å sette opp varsel på e-post. Senere i oppgaven viser vi hvordan vi har brukt dette ved hjelp av events til å bli varslet når batterier i systemet vårt nærmer seg å bli tomme.

I «Custom»-fanen under «Web Site Settings», kryss av boksen der det står «Show Device Reference Number on Device Management Page (for developers)». Nå kan vi se en tallverdi ved siden av hver komponent i «Home»-menyen. Dette er nyttig å ha lett tilgjengelig når man jobber med scripts, som vi kommer tilbake til senere i oppgaven. I denne fanen kan man også endre navn på filtrene som er brukt i «Device Management». Som vist i Figur 4.7 har vi valgt å filtrere/sortere etter rom og funksjon.

Device Location Display Options	
Location 1 Label	Function
Use Location 2	<input checked="" type="checkbox"/>
Location 2 Label	Room
Display Location 1 First	<input type="checkbox"/>
Filter Display by Location	<input checked="" type="checkbox"/>

Figur 4.7 Tilpasning av filterinnstillinger

4.2 Gjennomgang av menyer

HomeSeer bruker et web-grensesnitt for konfigurasjon og oppsett av det som til slutt blir et smarthussystem. Det er satt opp som en tradisjonell nettside med et hovedvindu med en menylinje på toppen hvor man navigerer seg gjennom de ulike delene av HomeSeer.

Ref	Status	Room	Function	Name	Address	Last Change	Control
579	Dim up..	Living room	Remotes	Living room switch	ZLLSwitch-2	02.05.2019 13.37.07	
580	100%	Living room	Remotes	Battery Hue dimmer switch 1	SensorBattery-2	14.03.2019 14.26.18	
477	100%	JowiHue	JowiHue	Entertainment area 1	JowiHueGroup-3	30.04.2019 13.02.59	Off On Down Up Alert Blink Colour loop
478	8418	JowiHue	JowiHue	Entertainment area 1 (Hue)	JowiHueGroup-3H	02.05.2019 13.37.07	0020E2 Down Up
479	140	JowiHue	JowiHue	Entertainment area 1 (Sat)	JowiHueGroup-3S	02.05.2019 13.37.07	Down Up
480	366	JowiHue	JowiHue	Entertainment area 1 (CT)	JowiHueGroup-3K	02.05.2019 13.37.07	Down Up

Figur 4.8 Skjerm bilde av hjem-skjermen.

Under «Home» vises alle enhetene som er koblet til systemet. Denne listen finnes også under «View» → «Device Management». Her har man grunnleggende kontroller for de forskjellige enhetene.

«View»-fanen er der man finner det viktigste når det gjelder automatisering. Her kan man endre navn på enhetene, sortere og finne referansenummer som man tar med seg over i «Events» for å sette opp regler for hva som skal skje når gitte betingelser er oppfylte. Man har også tilgang til «Energy» som automatisk loggfører effektforbruk for de enhetene som støtter det (for eksempel smarte stikkontakter). «Log» viser alle actions i systemet, og er nyttig til feilsøking og overvåking.

«Tools» har innstillinger som angår selve systemet. Linux-kommandoer, omstart og måleenheter er ting som finnes her.

«Plug-ins» inneholder spesialfunksjonalitet som hver tilkobla enhet kan ha, avhengig av hvilke plug-ins man har installert. Måleenheter, enhetsgrupper og tilkobling av enheter foregår her.

4.3 Plug-ins

Plug-ins utgjør mye av grunnlaget for hvordan man faktisk kan bruke HomeSeer til noe nyttig, da de fungerer som en kommunikasjonsbro mellom HomeSeer og smartenheter. De er laget av både HomeSeer og tredjeparter, der noen er gratis mens andre må betales for. Generelt trenger hver enkelt komponent som kommuniserer via IP en egen plug-in, mens Z-Wave-baserte enheter bruker samme plug-in. Alle plug-ins som koster penger kan prøves i 30 dager før kjøp, noe som har vært avgjørende for hvilke plug-ins vi har valgt.

4.4 Oppkobling

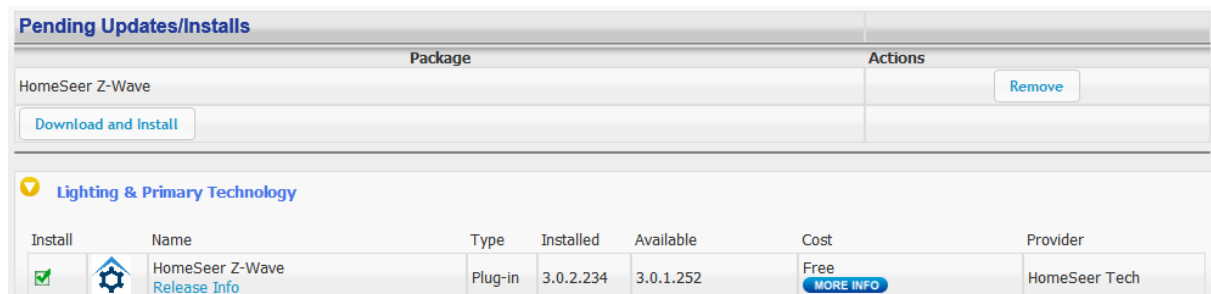
4.4.1 Z-Wave

Mange smart-enheter fra forskjellige produsenter bruker Z-Wave-protokollen, og denne støttes i HomeSeer ved bruk av en Z-Wave-plug-in. På HomeTroller-enhetene er denne installert som standard. Dersom man setter opp HomeSeer på egen maskin, må man installere plug-in-en manuelt. Det gjøres på følgende måte:

I toppmenyen på HomeSeer, gå til plug-ins og klikk «Manage».

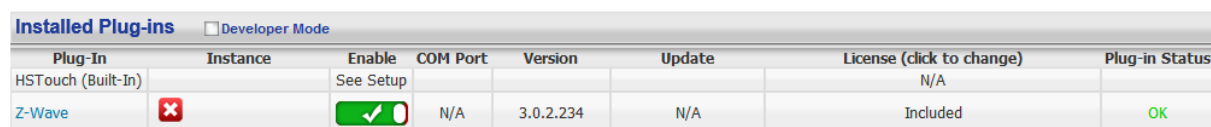
Her kan man se hvilke plug-ins som for øyeblikket er installert, og dersom man scroller ned på siden kan man bla gjennom ulike kategorier for ulike plug-ins. Vi er interesserte i den for Z-Wave, som ligger under «Lighting & Primary Technology» (den øverste).

Ved å huke av boksen til venstre ved siden av plug-in-en, dukker det øverst opp en «Download and Install»-knapp. Når den trykkes, lastes alle plug-ins som for øyeblikket er valgt ned.



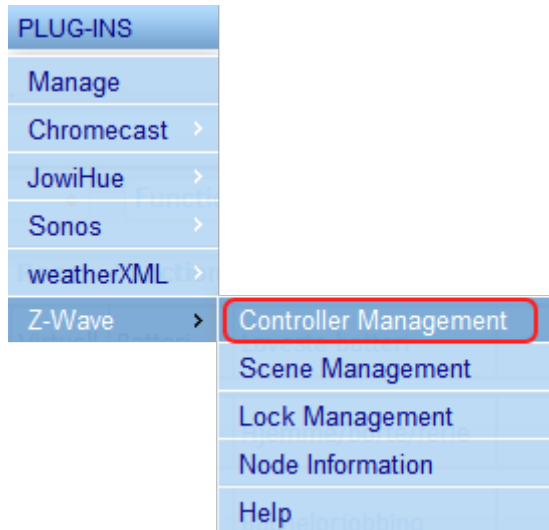
Figur 4.9 Installere Z-Wave plug-in

Når plug-in-en er installert, skal den dukke opp som aktivert under «Installed Plug-ins».

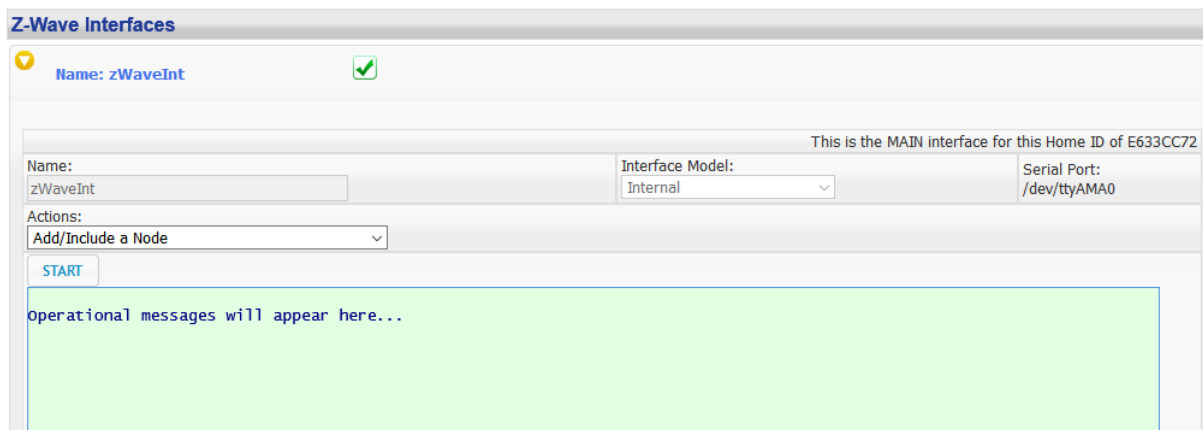


Figur 4.10 Installert og aktivert Z-Wave-plug-in.

Nå skal vi legge til en Z-Wave-enhet. Siden kommunikasjonen foregår ved hjelp av plug-in-en vi nettopp installerte, må vi gå via den for å opprette en tilkobling. Det gjør vi i «Controller Management» (Se Figur 4.11). Når vi da ekspanderer menyen for RaZberry-kortet (vårt Z-Wave-grensesnitt), kan vi i nedtrekksmenyen «Actions» velge «Add/Include a node», som vist i Figur 4.12.



Figur 4.11 Z-Wave «Controller Management»



Figur 4.12 Legg til en node

Når vi trykker på *Start* vil HomeSeer begynne å lete etter Z-Wave-enheter som skal legges til. Hvordan det gjøres vil være avhengig av merke og modell på smart-enheten, men fremgangsmåtene er stort sett like. Enheten som skal legges til må settes i søkemodus samtidig med HomeSeer. I dette eksempelet er en Fibaro Multisensor brukt. Den settes i søkemodus ved å trippel-trykke på knappen under batteridekselet (se Figur 4.13). Det vil så dukke opp en melding i tekstboksen om at enheten er lagt til HomeSeer, og da vil den være synlig i «Home»-menyen.



Figur 4.13 Knappen er under batteridekselet

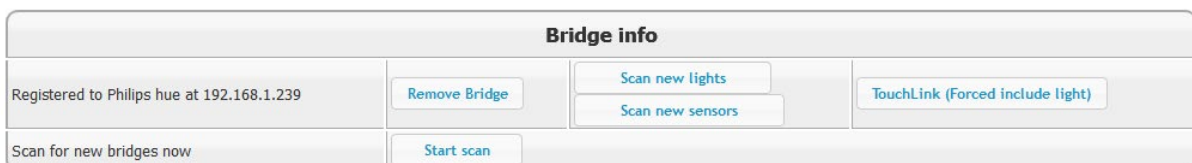
4.4.2 Hue

Philips Hue består av to hovedkomponenter for bruk med HomeSeer: smartlyspærene og en «bridge». Bridge-en fungerer som en mellommann mellom lyspærene og HomeSeer. Kommunikasjon mellom lyspære og bridge skjer via ZigBee Light Link (ZLL), mens HomeSeer snakker med bridge-en via ethernet. For at HomeSeer skal kunne snakke med bridge-en trenger vi også her en plug-in som er laget spesifikt for Philips Hue. Vi valgte å bruke JowiHue, som er en betalt plug-in som gir mange muligheter for styring. Før man kan bruke Hue sammen med HomeSeer, må denne installeres. Fremgangsmåte er den samme som vist i 4.4.1.

Når plug-in-en er installert, må den konfigureres for å etablere en kobling til bridge-en. Ved å navigere i toppmenyen til HomeSeer kan vi velge «Plug-ins» → «JowiHue» → «Configuration». Når vi skal legge til en bridge, trykker man på «Start scan». Når den har jobbet får brukeren beskjed om å trykke på bridge-knappen (Figur 4.14), og man kan deretter se at bridgen er lagt til (Figur 4.15).

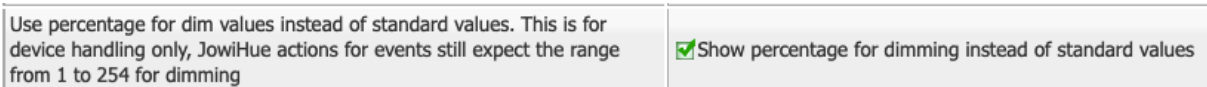


Figur 4.14 Philips-logoen er en knapp



Figur 4.15 Fjern eller legg til bridge og lyspærer

Når lyspærene har hatt strøm og bridge-en har fått skannet, dukker lyspærene opp i «device»-listen med både individuelle og gruppe-baserte innstillinger som standard. På den samme plug-in-siden huk av for å vise prosentert istedenfor absoluttverdier.



Figur 4.16 Prosentverdier for lysstyrke på pærene

4.4.3 Sonos

For å kunne kontrollere Sonos-høytalere trengs det også en egen plug-in. Denne er gratis, og støtter grunnleggende funksjoner som play/pause/frem/tilbake og så videre. Dessverre er det lite utvidet funksjonalitet på grunn av mangelfulle API-er. Dette gjør det problematisk å velge noe annet enn forhåndsinnhentede radiostasjoner ved hjelp av stemmestyring og/eller events. Valg av spillelister fra Spotify og andre streamingtjenester må gjøres på telefon, og kan deretter styres i HomeSeer med de grunnleggende kontrollene.

Plug-in-en installeres på samme måte som for de foregående komponentene. Tilkobling av Sonos til HomeSeer forutsetter at høytalene er satt opp og koblet til samme nettverk som HomeSeer-kontrolleren. Dette må gjøres på en smarttelefon med Sonos-appen. Når det er gjort vil plug-in-en finne høytalene automatisk.

4.4.4 Google Home

Google Home settes opp ved å bruke Google Home-appen på en Android- eller iOS-basert enhet. Ved å følge instruksjonene i appen, kobles Google Home Mini-en til nettverket. Google Home trenger ingen ekstra plug-in fordi støtte er innebygd i HomeSeer. Man må imidlertid koble Google-kontoen sammen med HomeSeer-kontoen for at det skal fungere. Guide finnes på HomeSeer sine hjemmesider [13].

4.5 Device list / Home-meny

Det er her man finner listen av komponenter og enheter som er lagt til i HomeSeer. Listen oppdateres automatisk etter hvert som komponenter blir lagt til.

Figur 4.17 viser et skjermbilde av hvordan menyen ser ut. En fysisk komponent kan ha mer enn en funksjon, og er i HomeSeer en sum av subkomponenter som til sammen utgjør denne. For eksempel har en multisensor temperaturmåler, lysstyrkemåler og bevegelsessensor. Hver av disse vil ha en egen subkomponent i HomeSeer. Alle subkomponentene til hver enhet er omsluttet av en ramme for å indikere at dette faktisk er samme fysiske enhet. I Figur 4.17 er det mange subkomponenter, og to rammer. Det vil si at det er i dette eksempelet to fysiske komponenter med henholdsvis sju og fem subkomponenter. Root-komponenten er den viktigste av subkomponentene. Den ligger øverst i hver ramme, og det er her man kan i detalj endre avanserte innstillinger som f. eks. parametere.

Device List Filtered, showing 88 of 173 Devices.

Display Filters: Room Function Device Type Show All

<input type="checkbox"/>	464		No Status	Living room	Door sensors	Fibaro Notification Sensor	E633CC72-014		
<input type="checkbox"/>	465		Status Was Reset	Living room	Door sensors	Application Status	E633CC72-014-Q12		Reset Application Status
<input type="checkbox"/>	466		64%	Living room	Door sensors	Battery	E633CC72-014-Q13	02.04.2019 17.03.29	
<input type="checkbox"/>	467		Open	Living room	Door sensors	Access Control Notification	E633CC72-014-Q14	26.03.2019 20.46.51	
<input type="checkbox"/>	468		Tampering Cover Removed	Living room	Door sensors	Home Security	E633CC72-014-Q15	04.03.2019 20.43.54	
<input type="checkbox"/>	469		ALARM	Living room	Door sensors	General Purpose Alarm	E633CC72-014-Q16	26.03.2019 20.46.51	
<input type="checkbox"/>	470		On-Open-Motion	Living room	Door sensors	Sensor Binary	E633CC72-014-Q17	26.03.2019 20.46.51	
<input type="checkbox"/>	417		No Status	Living room	Plugg	Fibaro Switch	E633CC72-009	27.02.2019 12.14.57	
<input type="checkbox"/>	418		515,440 kW Hours	Living room	Plugg	kW Hours	E633CC72-009-Q18	Today 14.56.21	Reset Accumulated Values
<input type="checkbox"/>	419		0	Living room	Plugg	Watts	E633CC72-009-Q19		
<input type="checkbox"/>	420		1183,4 W	Living room	Plugg	Power	E633CC72-009-Q20	Today 14.00.08	
<input type="checkbox"/>	421		On	Living room	Plugg	Switch	E633CC72-009-Q21	Today 06.00.09	Off On

Figur 4.17 To komponenter med tilhørende subkomponenter

4.5.1 Filtre og organisering

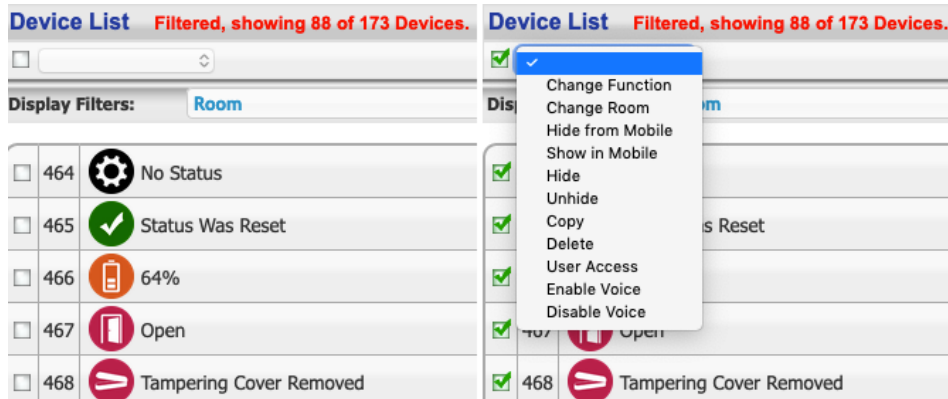
Filtre er nyttige for organisering i HomeSeer. I større oppsett kan man lett komme opp i tusenvis av subkomponenter, noe som gjør det viktig å kunne organisere og gruppere. Ovenfor komponentlisten kan man velge tre filtre, der de to første kan endres til det man ønsker. Vi har valgt å gruppere etter rom og funksjon. De kan endres ved å klikke på «Tools» → «Setup». Deretter velger man «Custom»-fanen, og endrer merkelappene som er vist i Figur 4.18.

Device Location Display Options

Location 1 Label	<input type="text" value="Function"/>
Use Location 2	<input checked="" type="checkbox"/>
Location 2 Label	<input type="text" value="Room"/>
Display Location 1 First	<input type="checkbox"/>
Filter Display by Location	<input checked="" type="checkbox"/>

Figur 4.18 Tilpasning av filter

I Figur 4.19, rett over «Display Filters» er en enslig avkryssingsboks, som vil huke av bare de komponentene nedover som er vist på skjermen. Det vil si om man har rom-filtrene «Stue» og «Kjøkken», og man bare viser komponentene fra «Kjøkken», er det kun kjøkken-komponentene som blir valgt. Dersom man markerer en eller flere komponenter vil en nedtrekksmeny bli tilgjengelig, og man kan gjøre endringer på alle de markerte komponentene istedenfor å ta de enkeltvis.



Figur 4.19 Egenskaper for alle valgte komponenter

4.6 Avanserte enhetsinnstillinger

Dersom man vil endre innstillinger for en spesifikk komponent og/eller man vil endre mer avanserte innstillinger, klikker man på navnet til subkomponenten. Her har man tilgang til opptil fire faner (hvor mange avhenger av hvilken komponent som er valgt).

Ref	Status	Room	Function	Name
<input type="checkbox"/> 417	No Status	Living room	Plugg	Fibaro Switch

Figur 4.20 Hvordan komme inn på enhetsinnstillinger

4.6.1 Configuration

Device Name, Floor og Room er alle parametere som kan endres på. Disse blir vist på hjemskjermen og de to sistnevnte er også med på filtreringen i systemet.

Voice Command er et felt hvor man kan fylle inn en spesifikk kommando for denne enheten, som igjen skal fungere hvis vi bruker en stemmeassistent. For eksempel hvis vi har valgt å kalle ett av lysene våre Gunnar i Voice Command, selv om lyset heter HUE1 under Device Name, så vil systemet reagere hvis vi sier «Turn on Gunnar». På denne måten kan vi forenkle hverdagen vår hvis vi bruker stemmeassistent. For at stemmeassistenten skal kunne overstyre valgt enhet, er det viktig at Voice Command er huket av.


Notes er en egen parameter hvor man kan føre sine egne notater for enheten. Her kan man for eksempel taste inn når den er kjøpt, hvilken batteritype den bruker eller lignende.

Modify First floor Living... 2/4/2019 3:42:54 PM Sunrise: 8:49 AM Sunset: 4:56 PM User: Kristoffer HomeSeer

HOME VIEW TOOLS PLUG-INS

Configuration Advanced Status Graphics Z-Wave

Device Name:	Fibaro Multisensor
Voice Command:	
Floor:	First floor New Floor
Room:	Living room New Room
Code:	<input type="text"/> <input type="text"/> Display Available Codes
Address:	E633CC72-007
Status Only Device:	<input type="checkbox"/>
Is Dimmable:	<input type="checkbox"/>
Is Light:	<input type="checkbox"/>
Hide device from mobile views:	<input type="checkbox"/>
Hide device from views:	<input type="checkbox"/>
Do not log commands from this device:	<input type="checkbox"/>
Voice command:	<input type="checkbox"/>
Confirm voice command:	<input type="checkbox"/>
Include in power fail recovery:	<input type="checkbox"/>
Use pop-up dialog for control:	<input type="checkbox"/>
Do not update device last change time if device value does not change:	<input checked="" type="checkbox"/>
User Access:	Users that have access: Any User Select Users
Notes:	+
Linked Device:	Not Linked



Figur 4.21 Konfigurasjon

4.6.2 Advanced

Inne på «Advanced»-fanen finner vi fire rader som er viktige for oss å ha kunnskap om, de resterende er info vi egentlig ikke bryr oss nevneverdig om med mindre vi har problemer. Reference ID er veldig viktig da hver enhet får sin referanse-ID, dette blir altså nummeret til enheten i vårt system. Slik vi ser i Figur 4.22 har denne enheten fått verdi 393. Denne ID-en er noe vi har bruk for hvis vi har lyst til å benytte scripting, noe vi skal vise litt senere i denne oppgaven. Videre ser vi Status og Value, disse går ofte hånd i hånd når det gjelder det som er foran likhetstegnet, mens det på høyre side kan variere basert på kontroll og status inne på «Status Graphics», men det kommer vi mer inn på i neste avsnitt. Den siste raden det kan være nyttig å ha kunnskap om er «Interface», da denne sier noe om hvilket grensesnitt som styrer enheten og hvordan enheten kom inn i systemet.


Configuration Advanced Status Graphics JowiHue

Reference ID	393
Technology Address	LCT015-2
Status	100 = ON
Value	100 = "100%"
String	
Supports Status	True
Dimmable	True
Interface	JowiHue
Extra Data Store	7 Named entries, 0 UnNamed entries.
Device Type Internal	Plug-In API, Type: Plug-In Type 0, SubType: Light
Device Type (String)	LCT015
Misc Settings	Show Values, Generate Voice Command
Device Image File	images/JowiHue/E27Color_large.png
Thumbnail Image File	images/JowiHue/E27Color_off.png
Relationship Status	Parent Root
Associated Devices	Living room Light One (hue), Living room Light One (sat), Living room Light One (CT)

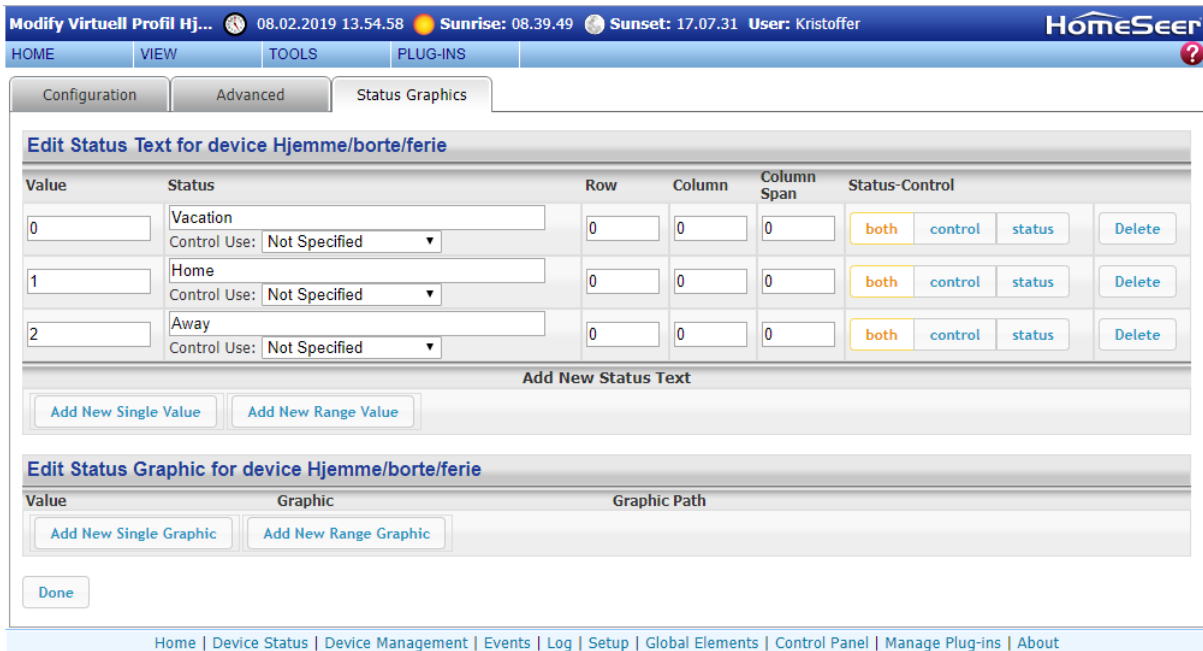
Figur 4.22 Avansert

4.6.3 Status Graphics

Hver enhet under «Device List» har en «Status»- og «Control»-kolonne. Disse brukes til å vise tilstanden til en komponent og til å kontrollere grunnleggende egenskaper, respektivt. I Figur 4.23 ser vi en enhet som viser om man er hjemme, borte, eller på ferie. Statusen, eller tilstanden vises til venstre (med bilde), og den kan manipuleres med knappene til høyre.

Status	Room	Function	Name	Address	Last Change	Control
 Vacation	Virtuell	Profil	Hjemme/borte/ferie		Today 13.54.52	Vacation Home Away

Figur 4.23 Enhet med "Status" og "Control"-kolonner



The screenshot shows the HomeSeer interface with the 'Status Graphics' tab selected. The page title is 'Edit Status Text for device Hjemme/borte/ferie'. It contains a table with columns: Value, Status, Row, Column, Column Span, and Status-Control. The table has three rows for 'Vacation', 'Home', and 'Away'. Each row has a 'Value' input field (0, 1, 2), a 'Status' dropdown menu, and a 'Control Use' dropdown menu (all set to 'Not Specified'). The 'Status-Control' column contains buttons for 'both', 'control', 'status', and 'Delete'. Below the table are buttons for 'Add New Single Value' and 'Add New Range Value'. There is also a section for 'Edit Status Graphic for device Hjemme/borte/ferie' with 'Add New Single Graphic' and 'Add New Range Graphic' buttons, and a 'Done' button at the bottom.

Figur 4.24 Status Graphics

I «Status Graphics»-fanen (se Figur 4.24) velger vi utseendet og navnet til «Status»-kolonnen, samt knappene som skal være i «Control»-kolonnen. Til venstre er det en kolonne som heter «Value». Dette er en liste med forskjellige verdier som en enhet kan ha. Hver verdi representerer en tilstand. Det kan være både «single»-verdier og «range»-verdier. I dette eksempelet tar vi for oss «single»-verdier (vi kommer tilbake til «range»-verdier i 4.10).

Vi ser på hvordan en «Status Graphics»-fane ser ut for en enhet med kun «single»-verdier. «Value» 0 har fått statusen Vacation, 1 har fått statusen Home, og 2 har fått statusen Away. Dette vil si at når vi har trykt på «Vacation»-knappen så vil feltene «Status» og «Value» i «Advanced»-fanen endre verdi til 0 (se Figur 4.25), og tilsvarende for de andre knappene og verdiene

Configuration	Advanced	Status Graphics
Reference ID	398	
Status	0 = OFF	
Value	0 = "Ferie"	
String		

Figur 4.25 Statusverdier

Dersom man velger engelske navn på statusene, er det av og til allerede lagt inn bilder i HomeSeer som matcher disse. De befinner seg i mappen `images/HomeSeer/status`. Om man vil ha norske navn på statusene, må man legge inn/referere til bilder manuelt slik som vi har gjort i Figur 4.26.




The screenshot shows two configuration panels for a device named 'Hjemme/borte/ferie'.

Edit Status Text for device Hjemme/borte/ferie

Value	Status	Row	Column	Column Span	Status-Control
0	Ferie Control Use: Not Specified	1	1	0	both control status
1	Hjemme Control Use: Not Specified	2	2	0	both control status
2	Borte Control Use: Not Specified	3	3	0	both control status

Buttons: Add New Single Value, Add New Range Value, Delete

Edit Status Graphic for device Hjemme/borte/ferie

Value	Graphic	Graphic Path
0		images/HomeSeer/status/vacation.png
1		images/HomeSeer/status/home.png
2		images/HomeSeer/status/away.png

Buttons: Edit, Delete

Figur 4.26 Status Graphics norsk

Nest til høyre har vi valg som omhandler «Status-Control». Dersom vi velger «status» vil dette være en verdi som blir rapportert og vist i HomeSeer, der vi ikke kan endre verdien direkte. For eksempel dersom et termometer sender 25 grader til den virtuelle enheten sin verdi, kan vi ikke endre verdien til den virtuelle enheten. Vi kan derimot sette tekst- og bildeverdier dersom termometeret har rapportert en viss verdi, knappen under «Control» på hjemskjermen vil ikke synes når dette valget er valgt. Når det gjelder «control» så blir kun verdien satt til enheten, mens status-teksten ikke blir å se i hjemskjermen. Merk her at hvis man har lagt til bilder så vil disse komme da disse også trigges av verdiene vi har valgt. Hvis «both» står på, vil man ha en knapp i hjemskjermen man kan trykke på for å kontrollere verdien til enheten, og når man trykker på denne vil man få opp den tilhørende status-teksten, denne kombinerer med andre ord både status og kontroll! Dette er ganske greit, da det ville vært utrolig tungvint hvis man kun kunne ha en av dem om gangen, og dermed måtte lage en statusrad for valgt verdi og en kontrollrad for samme valgte verdi.

Vi kan også endre på posisjonen til knappene under «Control». Dette gjøres i «Row», «Column» og «Column Span». Som standard er verdiene 0 0 0 på alle statusene våre, som gjør at knappene går fra venstre til høyre basert på stigende rekkefølge av verdiene. Man kan se på dette som en matrise, så om man endrer verdiene slik som i Figur 4.26 ser vi resultatet i Figur 4.27. Det ville også vært mulig å få dem rett under hverandre men på ulike rader ved å endre «Column» til samme verdi på alle knappene våre. «Column Span» brukes ikke her, men brukes til å optimalisere utseendet til knappen hvis det er for lang tekst i knappenavnet.



Figur 4.27 Matrise

4.6.4 Plug-in-spesifikk

De enhetene som er blitt lagt til via en plug-in vil få en fjerde fane når vi er inne og skal modifisere enhetene våre. Dette vil altså si at enheter lagt til via Z-Wave, JowiHue, Sonos, og andre plug-ins vil ha en fjerde fane hvor det står navnet til plug-in-en som enheten bruker. «Z-Wave» for de som er lagt til via Z-Wave, «JowiHue» for Philips Hue og så videre. For å kunne gjøre endringer i denne fanen må vi gå inn på rootkomponenten eller foreldre-enheten til gruppa. Vi kan fra enkelte plug-ins få tilgang til info om enheten hvis vi er i en subkomponent, men vi kan ikke gjøre noen endringer.

4.6.4.1 Z-Wave

Ved å trykke inn på en rootkomponent og så inn på Z-Wave-fanen vil vi få opp skjermen som vist i Figur 4.29. Her vil vi få opp info om noden hvis vi trykke på den gule pilen ved siden av «Information About Node 14». Som vi ser i Figur 4.28 får vi litt varierende informasjon herfra, det meste av informasjonen er kanskje ikke veldig nødvendig for Ola Nordmann å ha kontroll på, men det er greit å vite hvor man kan finne denne informasjonen. Noe av informasjonen man finner her kan man bruke i feilsøking av system og enheter.

Information About Node 14			
Network and Node ID	E633CC72, 14	Manufacturer	Fibaro System (ID=0x10F)
Product Type	0x701	Product ID	0x1001
Z-Wave Plus Node Type	ZWP Node	Z-Wave Plus Role Type	(Unknown)
Supports Status	No	Library Type	Enhanced Routing Slave
Z-Wave Lib Version	4,05 (Unknown)	Application Version	3.2
Hardware Version	1	Associated Device	Living room Door sensors Application Status Cmd Class: Application Status Cmd Class Version: 1
Associated Device	Living room Door sensors Battery Cmd Class: Battery Cmd Class Version: 1	Associated Device	Living room Door sensors Access Control Notification Cmd Class: Notification Cmd Class Version: 5
Associated Device	Living room Door sensors Home Security Cmd Class: Notification Cmd Class Version: 5	Associated Device	Living room Door sensors General Purpose Alarm Cmd Class: Sensor Alarm Cmd Class Version: 1
Associated Device	Living room Door sensors Sensor Binary Cmd Class: Sensor Binary Cmd Class Version: 1	Firmware Update Command Class Version	3
Association Groups	(5) = 1, 2, 3, 4, 5	Association Group 1	Max=1, Nodes=1
Association Group 2	Max=10, Currently Empty	Association Group 3	Max=5, Currently Empty
Association Group 4	Max=5, Currently Empty	Association Group 5	Max=5, Currently Empty
Wake-Up Queue Size and First Item	1 Items in Queue, First is (Unknown)	Neighboring Nodes	2 Neighbors: 9, 10

Figur 4.28 Nodeinformasjon

I Figur 4.29 ser vi videre at vi har en del knapper under «Functions», dette er knapper vi kan bruke når vi føler at noden vår ikke fungerer som den skal og vi vil prøve å finne ut hva som kan være galt. For å kunne bruke funksjonene disse knappene har står det at enheten må manuelt vekkes opp, dette gjøres ved å trykke på knappen vist i Figur 4.13, eventuelt finner man info om hvordan manuelt vekke en enhet i enhetens datablad.

Vi ser videre i Figur 4.29 at vi har en rad som heter «Firmware Update». Denne brukes til å kjøre ut fastvareoppdateringer til enheter som kjører Z-Wave Plus, men det forutsetter at man har denne fastvareoppdateringen tilgjengelig, filer de fleste produsenter foreløpig holder for seg selv. Men man vil forhåpentligvis med tiden klare å få styrt markedet til at alle produsenter må frigi oppdateringsfilene slik at forbrukere uavhengig av hvilken hjemmesentral man bruker kan få kjørt ut disse oppdateringene.

Den nederste raden i Figur 4.29 viser assosiasjoner. Her kan vi assosiere noder slik at en node kan styre en annen basert på informasjonen den får. Dette er ikke noe vi har benyttet oss av i vårt system, men det er verdt å få med seg at denne muligheten finnes.

4.6.4.1.1 Parameter

Parametere er innstillinger og tilpasninger vi kan gjøre på enheten vår. Parameterne inneholder et nummer, en lengde og en verdi. Man finner info om hva de ulike parameterne til en enhet gjør ved å se i databladet til enheten. For å kunne endre parameterne til en enhet må vi inn i fanen «Settings», og som man ser i Figur 4.29 må vi vekke opp enheten vi vil endre parameter på.

The screenshot displays the HomeSeer web interface for configuring a Z-Wave node. The top navigation bar includes 'HOME', 'VIEW', 'TOOLS', 'PLUG-INS', and a 'Z-Wave' tab. The main content area is titled 'Information About Node 7' and contains several sections: 'Options' with a dropdown for 'Node Number' set to 7 and a 'Polling Interval' field; 'Functions' with a note that the device must be woken up manually and buttons for 'Rescan', 'Remove Bad Node', 'Replace Node', 'Test Connectivity', 'Optimize', 'Full Optimize', and 'Audit This Node'; 'Firmware Update'; 'Settings' for a 'Fibaro Motion Sensor Version 3.2' with a red error message 'Device did not respond, aborting reading parameters, make sure device is awake and try again'; and 'Associations'. A 'Done' button is located at the bottom of the configuration area.

Figur 4.29 Enhetsinnstillinger Z-Wave

Når en enhet har blitt vekket opp vil vi få opp en liknende skjerm som vist i Figur 4.29. Som man ser kommer noen parametere opp i listeform med beskrivende tekst. Hvis dette er en av parameterne man vil endre så kan man endre de ganske direkte her. Vi vil anbefale å dobbeltsjekke i datablad om teksten i grensesnittet er lik teksten i databladet, slik man vet at man endrer korrekt parameter. Er man i tvil kan man alltid benytte seg av «Set Configuration Parameters» (se nederst på Figur 4.30). Her ser man at man kan velge parameternummer, verdi og størrelse. Denne informasjonen finner man i datablad om hvert av parameterne. Det er verdt å merke seg at man kun kan endre en parameter om gangen. Det innebærer at når man har satt en parameter, så må man trykke på «Set» for at endringen skal skje. Det er også verdt å merke seg at endringene må skje relativt raskt slik at enheten er våken når vi endrer parameteren, hvis ikke blir den sendt ved neste oppvåkning. Ved full reset av enheten vil alle parametere gå tilbake til fabrikkinnstillinger.

The screenshot shows the 'Z-Wave' configuration page for 'Node 7'. It is divided into several sections:

- Options:** Node Number: 7, Polling Interval: 0 Hours 0 Minutes 0 Seconds.
- Functions:** Rescan, Remove Bad Node, Replace Node, Test Connectivity, Optimize, Full Optimize, Audit This Node. A note states: 'Note: The device must be woken up manually to use the functions below.'
- Firmware Update:** (Section header)
- Settings:** Fibaro Motion Sensor Version 3,2
 - Motion Sensitivity (8-255) (lower=more sensitive): Value: 15
 - Motion Blind Time (0-15): Value: 15
 - Motion Detection Pulse Counter: Value: 2 Pulses
 - Motion Detection Window Time: Value: 12 Seconds
 - Motion Detection Alarm Cancellation Delay (1-32767) (seconds): Value: 15
 - Tamper Sensitivity (0-121) (0.08g-2g every 0.01g): Value: 20
 - Illumination Report Threshold (0-32767 (lux)): Value: 100
 - Illumination Report Interval (0-32767): Value: 3600
 - Temperature Report Threshold (0-255 (.1-25.5 C 0.1 step)): Value: 10
 - Temperature Measure Interval (0-32767): Value: 900
 - Temperature Report Interval (0-32767): Value: 0
 - LED Brightness (0-100%): Value: 20
 - Set wake-up interval (min=1 seconds max=18 hours, 12 minutes, 15 seconds): 0 Days 12 Hours 0 Minutes 0 Seconds
- Set Configuration Parameters:** Use this feature to change Z-Wave configuration parameters that are not included in the settings above. Fields for Parameter Number, Value (Decimal), and Value Size (1 Byte).

Figur 4.30 Parametere fra en motionsensor

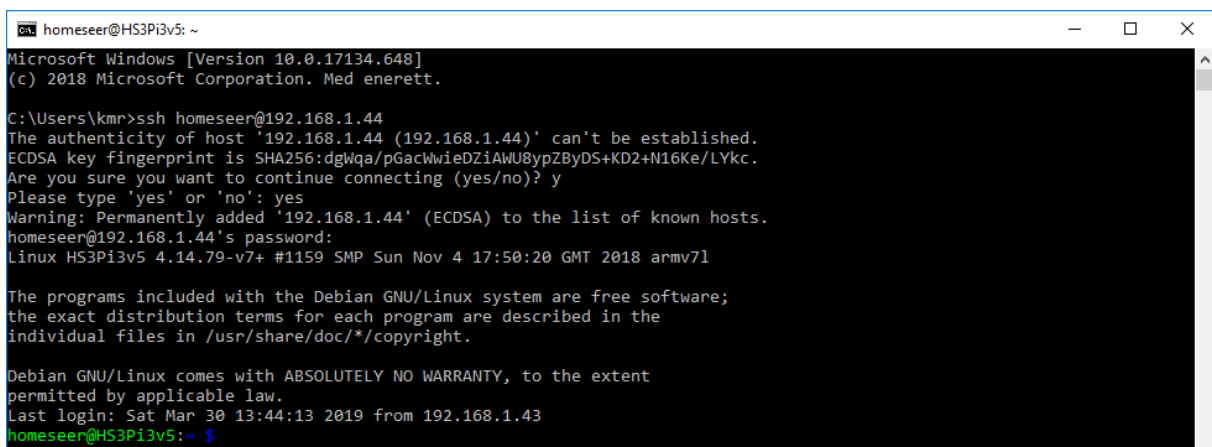
4.7 Ekstern tilgang

En måte å kjøre kommandoer på Pi-en er ved å sette inn tastatur og skjerm direkte i enheten, men en mer praktisk måte er kanskje via fjerntilgang. Vi skal i dette kapittelet gå gjennom ulike måter vi har brukt til fjernstyring av Pi-en

4.7.1 Secure shell (SSH) via kommandolinje

Vi kan få fjerntilgang via Terminal for MacOS og via Ledetekst for Windows. Fremgangsmåten er tilnærmet lik for begge to og vil dermed bli beskrevet samtidig.

1. Åpne Terminal/Ledetekst
2. Skriv inn **ssh brukernavn@ip-adresse** . Brukernavnet er «homeseer» som default, dette finner vi inne i web-grensesnittet til HomeSeer ved å gå til «Tools» → «System» → «Linux tools». IP-adressen er den lokale IP-en til Pi-en (den samme bruker vi når vi skal inn på web-grensesnittet).
3. Blir så bedt om å skrive inn passordet, som er det vi i 4.1.4 satte til denne brukeren.



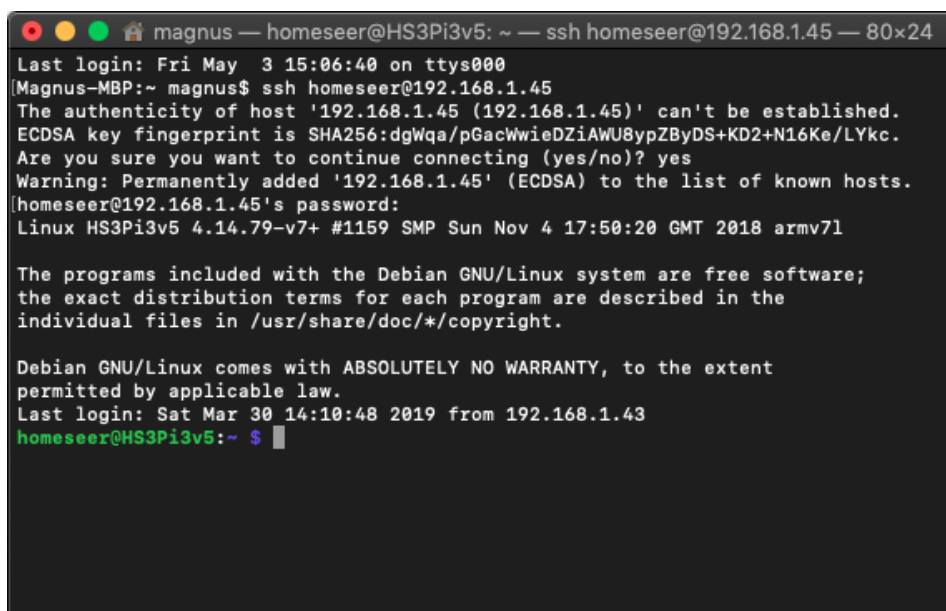
```
homeseer@HS3Pi3v5: ~
Microsoft Windows [Version 10.0.17134.648]
(c) 2018 Microsoft Corporation. Med enerett.

C:\Users\kmr>ssh homeseer@192.168.1.44
The authenticity of host '192.168.1.44 (192.168.1.44)' can't be established.
ECDSA key fingerprint is SHA256:dgWqa/pGacWwieDZiAWU8ypZByDS+KD2+N16Ke/LYkc.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.1.44' (ECDSA) to the list of known hosts.
homeseer@192.168.1.44's password:
Linux HS3Pi3v5 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 30 13:44:13 2019 from 192.168.1.43
homeseer@HS3Pi3v5:~$
```

Figur 4.31 Ledetekst i Windows



```
magnus — homeseer@HS3Pi3v5: ~ — ssh homeseer@192.168.1.45 — 80x24
Last login: Fri May 3 15:06:40 on ttys000
[Magnus-MBP:~ magnus$ ssh homeseer@192.168.1.45
The authenticity of host '192.168.1.45 (192.168.1.45)' can't be established.
ECDSA key fingerprint is SHA256:dgWqa/pGacWwieDZiAWU8ypZByDS+KD2+N16Ke/LYkc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.45' (ECDSA) to the list of known hosts.
[homeseer@192.168.1.45's password:
Linux HS3Pi3v5 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l

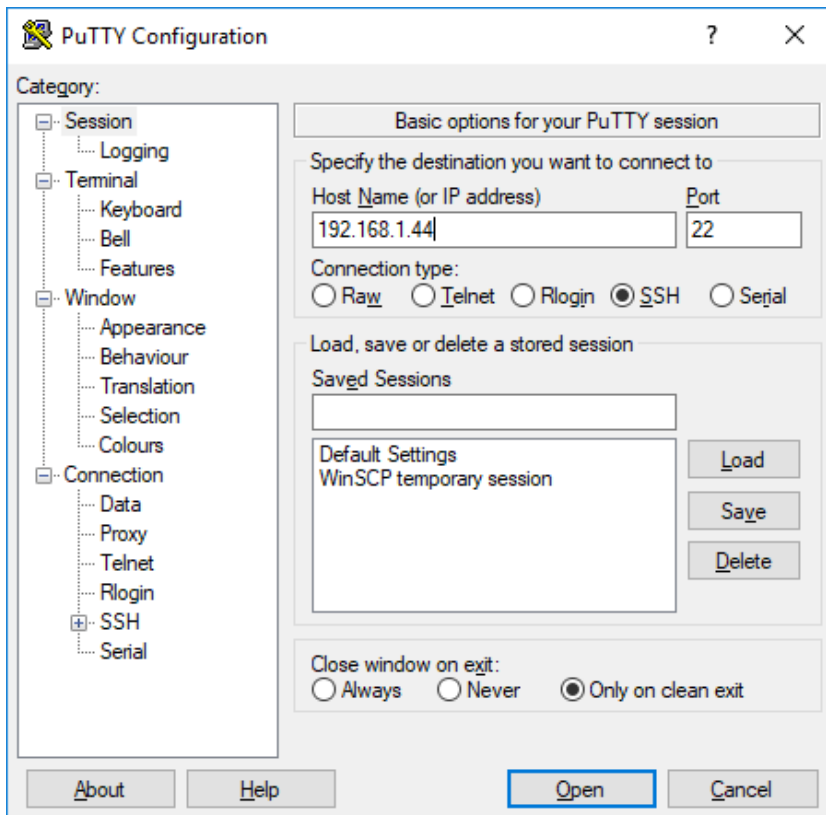
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 30 14:10:48 2019 from 192.168.1.43
homeseer@HS3Pi3v5:~$
```

Figur 4.32 Terminal i MacOS

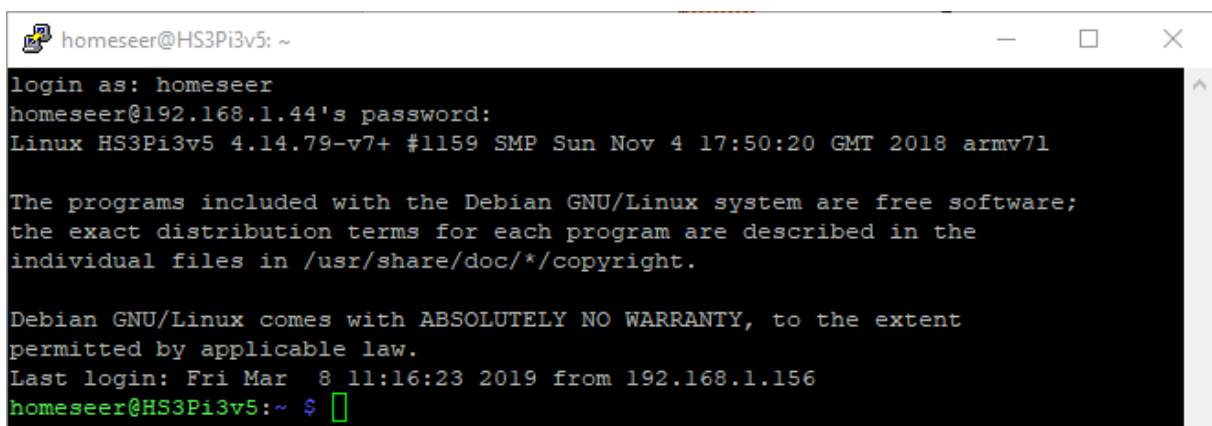
4.7.2 Tilgang via PuTTY

Det finnes andre metoder for å få ekstern tilgang, for eksempel ved å bruke PuTTY. [14]. Måten dette fungerer på er at vi skriver inn den lokale IP-adressen til vår Pi. Vi må bruke tilkoblingstype SSH og portnummer 22. Når vi har skrevet inn disse detaljene trykker man på «Open».



Figur 4.33 PuTTY SSH tilkobling

Vi kommer så inn i et ledetekstliknende vindu, her blir vi først bedt om å skrive inn brukernavn, og deretter passord. Her benytter vi «homeseer» som brukernavn og passordet som ble laget i 4.1.4. Vi er nå inne på Linuxsystemet og kan dermed begynne å bruke Linuxkommandoer på samme måte som tidligere i oppgaven.



Figur 4.34 PuTTY ledetekst

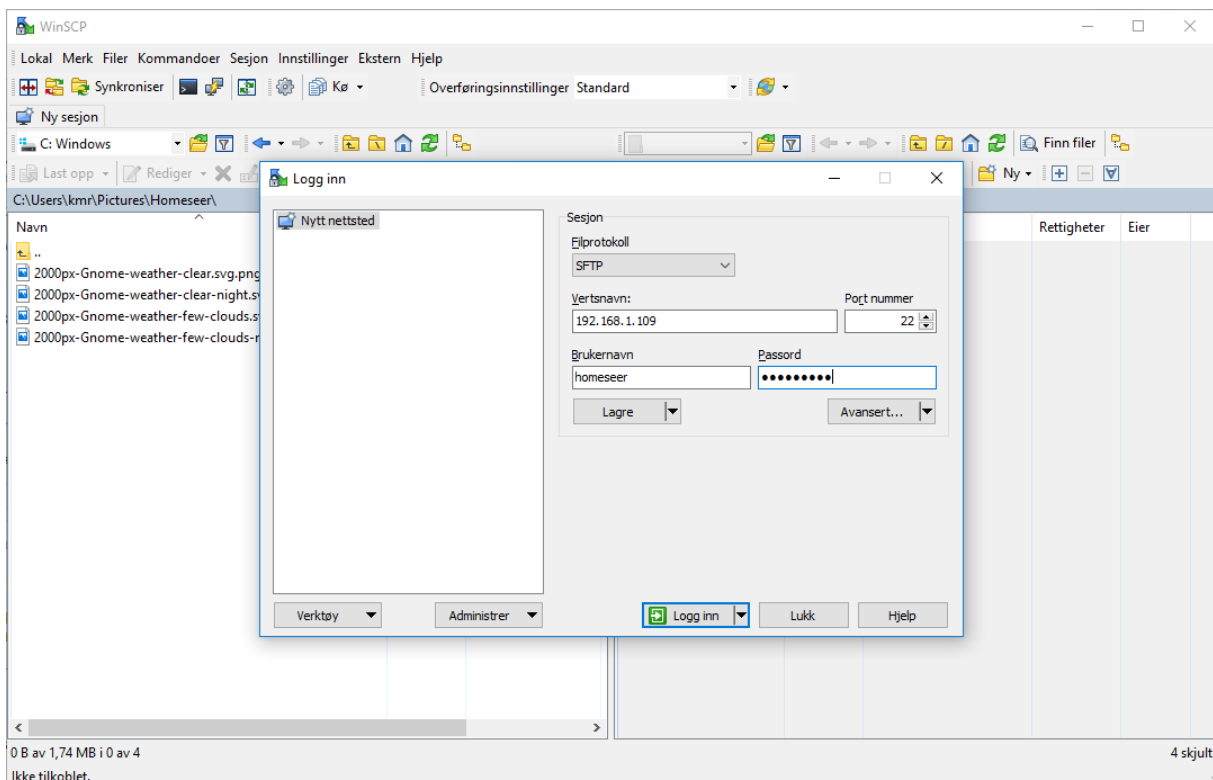
4.7.2.1 Root-tilgang

For å kunne få root-tilgang må man gjøre følgende [15]:

1. Logg inn med samme brukernavn og passord som i 4.7.1
2. Skriv kommandoen ***sudo passwd root***. Denne kommandoen gjør at vi lager et passord til root-brukeren.
3. Blir bedt om å skrive inn passordet til brukeren vi er logget inn med.
4. Så blir man bedt om å skrive et passord til root-brukeren. For enkelhetens skyld har vi valgt å ha samme passord på begge brukerne.
5. Man må deretter verifisere det nye passordet.
6. Passordet er nå laget. For å verifisere dette går vi ut av PuTTY, for så å logge inn på nytt bare med «root» som brukernavn, og det nye passordet.

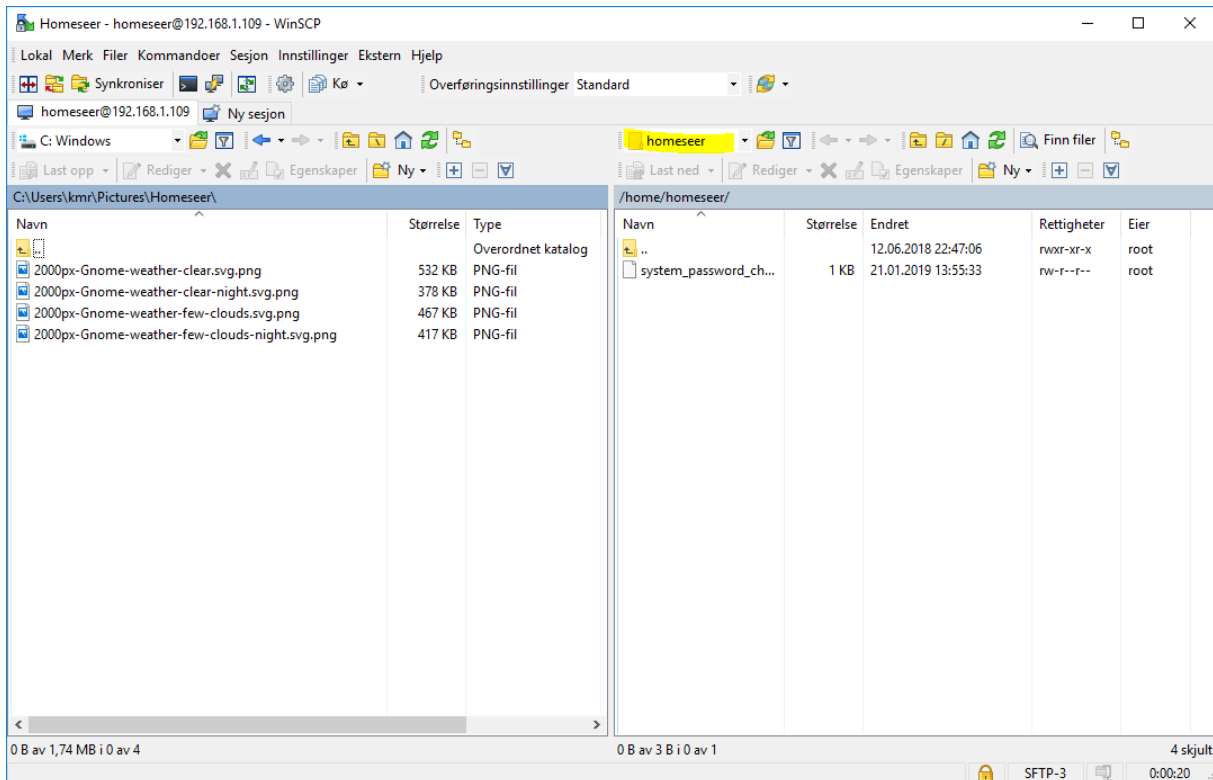
4.7.3 Filoverføring via SFTP (SSH File Transfer Protocol)

For å overføre filer til vår Raspberry Pi har vi valgt å bruke en Windows-PC og programmet WinSCP [16]. Vi trenger en måte for å overføre filer til Pi-en, siden vi vil benytte oss av egne script, og egne bilder 4.7.2 og portnummer som nevnt i 4.7.2, men istedenfor SSH bruker vi SFTP. Vi må også skrive inn brukernavn og passord (her bruker vi samme brukernavn og passord som vi laget til Linux i 4.1.3). Man kan også benytte seg av root-tilgangen vi laget i delkapittelet over.



Figur 4.35 WinSCP med SFTP-tilkobling

Vi er nå kommet inn i Linuxsystemet med WinSCP. Til venstre ser vi filene og filbanen på PC-en vi benytter til filoverføringen, mens vi til høyre ser Linuxsystemet. Det som er merket i gult er mappen vi befinner oss i på det gitte tidspunktet. For å gå opp et mappenivå kan vi enten trykke på mappa med pil på seg, eller trykke på det som er markert i gult i figuren og velge mappenivået man vil inn i.

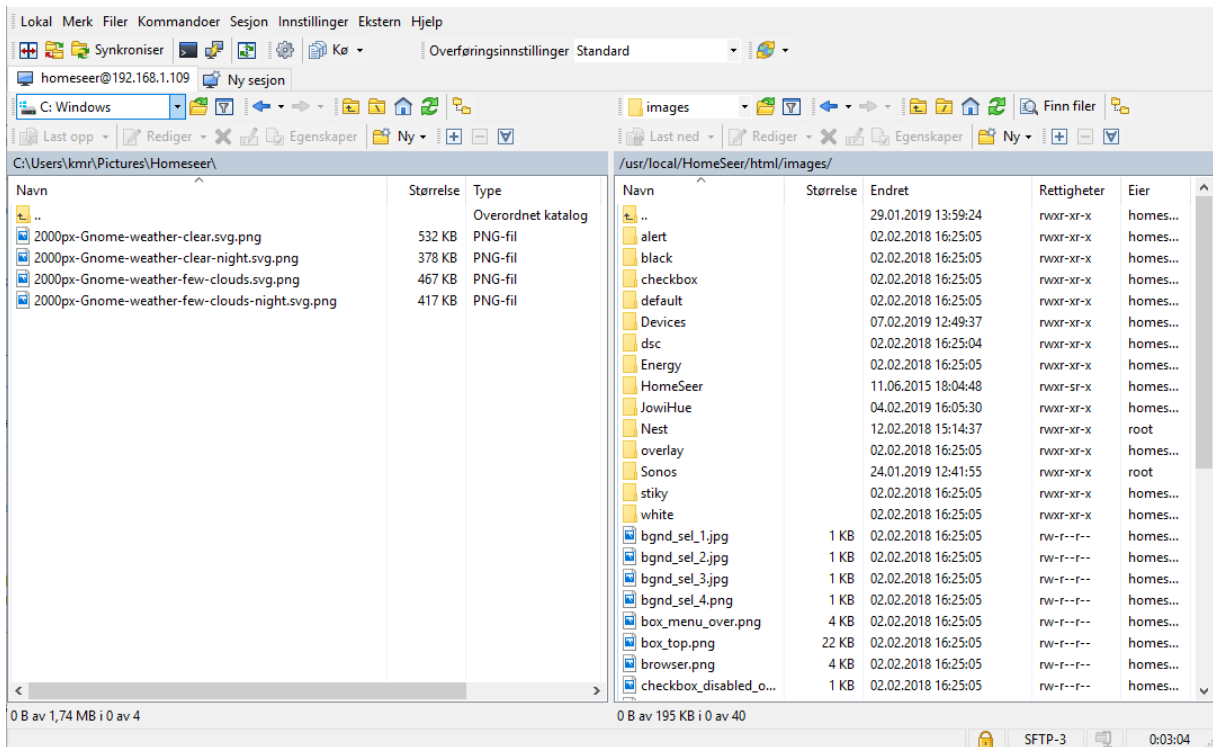


Figur 4.36 WinSCP utseende

Hvis vi vil legge til bilder som vi skal bruke som grafisk statusvisning av enheter må vi legge de til i følgende filbane:

usr/local/HomeSeer/html/images

Merk her at vi legger dem i images og ikke Images, altså liten i. Måten vi legger de til på er at vi finner filbanen hvor filene befinner seg på PC-en på venstre side av skjermen, og filbanen nevnt over på høyre side av skjermen. Når dette er gjort drar og slipper man filene fra venstre (PC) til høyre (Linux). Og selvsagt hvis det er filer vi vil ha fra Linux over til PC drar vi filene fra høyre til venstre.



Figur 4.37 WinSCP bilder mappe

En annen filbane det er verdt å merke seg er:

`/usr/local/HomeSeer/scripts`

Dette er filbanen hvor vi kommer til å legge scriptene. Vi har benyttet denne filbanen veldig mye da vi har brukt PC og Visual Studio til å redigere i script vi har fått feilmeldinger på, for så å laste dem opp til filbanen over for å prøvekjøre dem på nytt.

Som vi ser på bildene over er det ulike eiere av de ulike filene. Det vil si at for å kunne endre enkelte filtyper så må vi ha høyere tilgang enn det HomeSeer-brukeren har. Vi må dermed benytte oss av root-tilgang som beskrevet i 4.7.2.1.

4.8 Virtuelle enheter

Er det funksjonalitet man savner i HomeSeer kan man ofte fikse denne funksjonaliteten ved å lage en virtuell enhet, for så å bruke denne i events. I dette kapittelet skal vi gå inn på hvordan vi kan sette opp virtuelle enheter, og med eksempel på forskjellige bruksområder av disse.

4.8.1 Oppsett

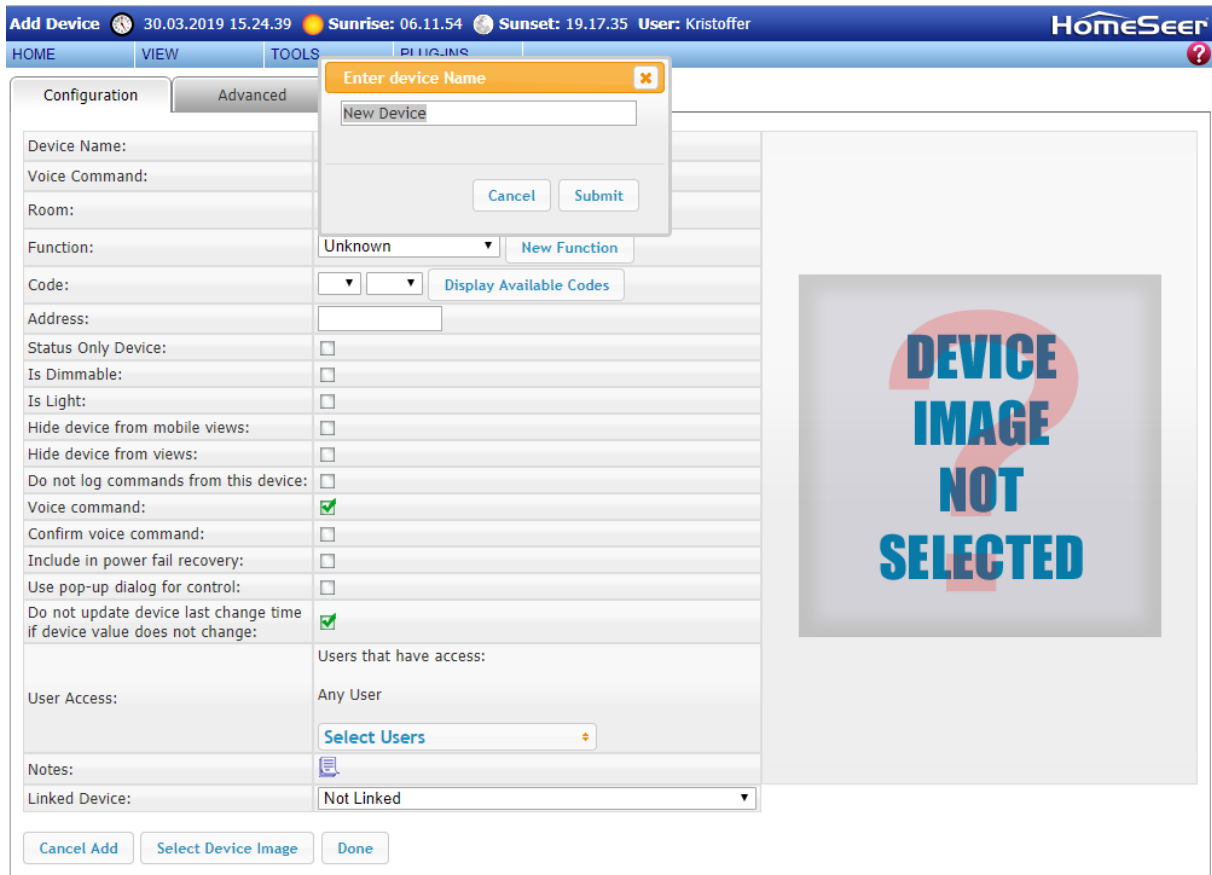
For å sette opp en virtuell enhet trykker vi på pluss-tegnet som vist i Figur 4.38



Figur 4.38 Virtuell enhet laging

Da får vi opp en skjerm vist i Figur 4.39. Her må vi velge navn på enheten, rom, funksjon og så videre. Det er ikke kritisk om vi ikke vet hva vi vil bruke enheten til foreløpig, da vi kan endre på disse

innstillingene senere også. Vi har gått gjennom de ulike valgene man kan gjøre per enhet tidligere, de samme gjelder her og vi kommer ikke til å gå mer i dybden på disse i dette kapittelet.



Figur 4.39 Virtuell enhet konfigurasjon

Innstillingene på denne enheten vil være valgt spesifikt for hvordan vi ønsker å bruke denne, og er dermed ikke retningsgivende for hvordan alle virtuelle enheters innstillinger skal være.

4.8.2 Profiler

Vi har i denne oppgaven benyttet oss av virtuelle enheter for å sette profiler som man kan bruke til å styre ulike events, og til å lese av og skrive ut laveste batteristatus.

4.8.2.1 Husstatus

Den profilen vi har satt opp som «hovedbryter» her er husstatusen, altså om man er hjemme, borte eller på ferie. Denne er tiltenkt å stå som hjemme når vi er i vanlig hverdag, borte når man er på helgetur, og ferie når man reiser på en lengre ferie. Denne enheten kan styres ved hjelp av knapper, slik det er enkelt å sette hvilken status huset skal ha.

4.8.2.2 Tidsstatus

Den andre profilen, Tidsstatus, er en profil som sier hvilken periode på døgnet det er. Denne har vi satt opp slik at den skifter mellom morgen, dag, kveld eller natt ved gitte klokkeslett.

4.9 Events og scripts

I HomeSeer er events det som muliggjør automatisering. Oppsett av events minner om betingelsessetninger fra tradisjonell programmering der man har en eller flere betingelser som må være oppfylte for at noe skal skje.

4.9.1 Triggere

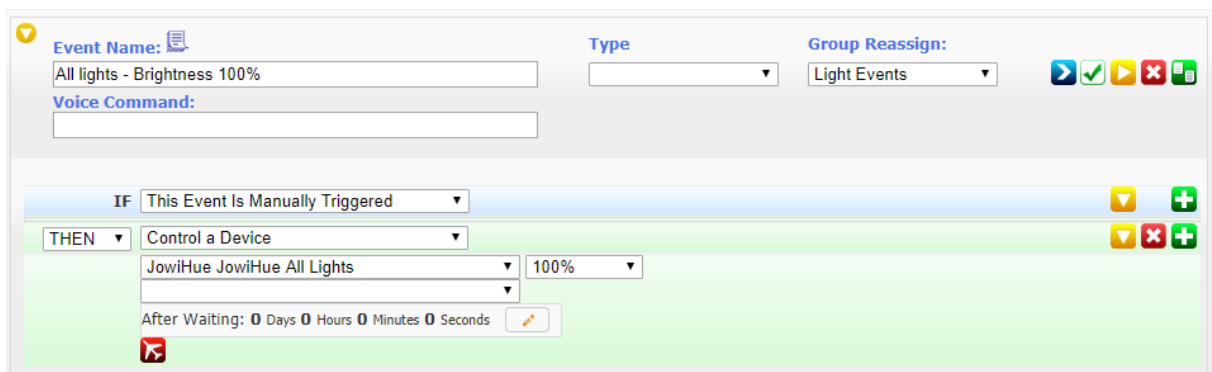
Triggere er betingelsene som settes for at noe skal skje. I motsetning til i programmering der man skriver betingelsene sine selv, velger man i HomeSeer disse fra nedtrekksmenyer. Hva som ligger i nedtrekksmenyene avhenger i stor grad av hvilke enheter og plug-ins man har koblet til systemet. På grunn av dette er programmeringserfaring ikke nødvendig for å lage fungerende betingelser, men fordi det er veldig logikk-basert vil dette være en fordel å ha kjennskap til/erfaring med det.

4.9.2 Action

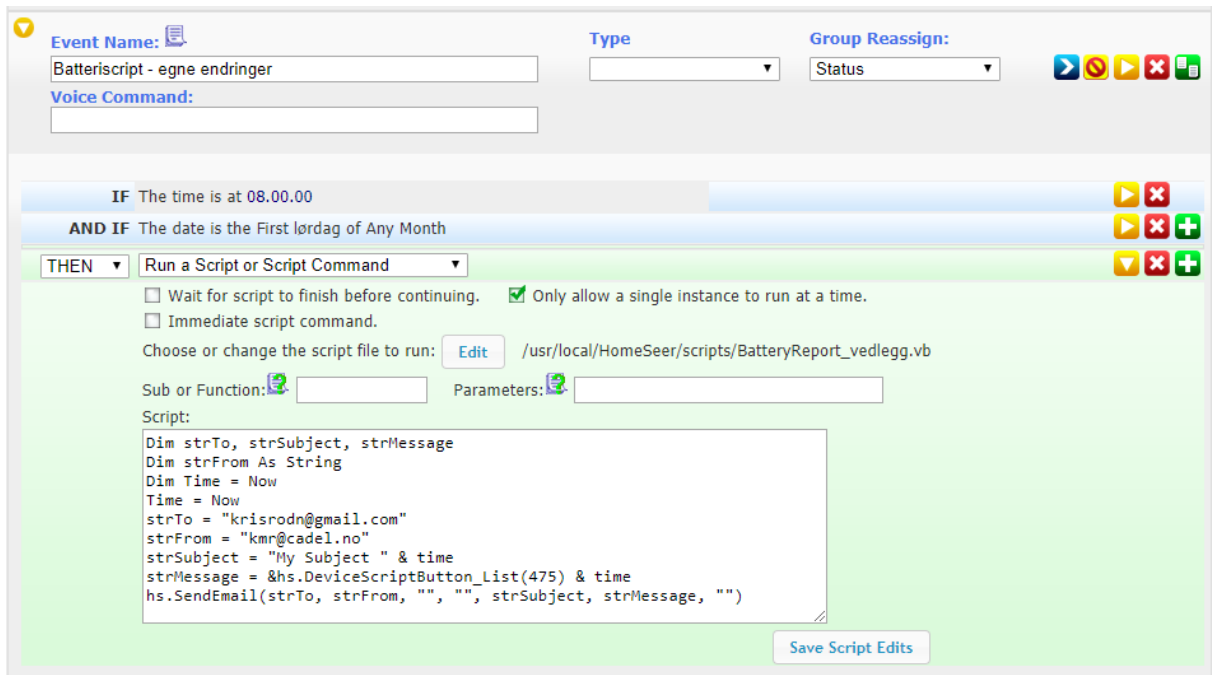
Når man setter opp events må man også velge hva som skal skje når betingelsene er oppfylte. Det vil her også være ulike valg avhengig av hvilke plug-ins man har installert.

4.9.2.1 Scripts

Scripts er handlinger som muliggjør funksjonalitet som ikke er tilgjengelig via de eksisterende nedtrekksmenyene. På denne måten kan man skreddersy events slik at man får nøyaktig den funksjonaliteten man vil ha. Språket som er brukt i HomeSeer er VB.NET. Man kan velge å kjøre én kodelinje, eller man kan lage et større script og laste opp manuelt.



Figur 4.40 Enkel event med if-setning og en kommando



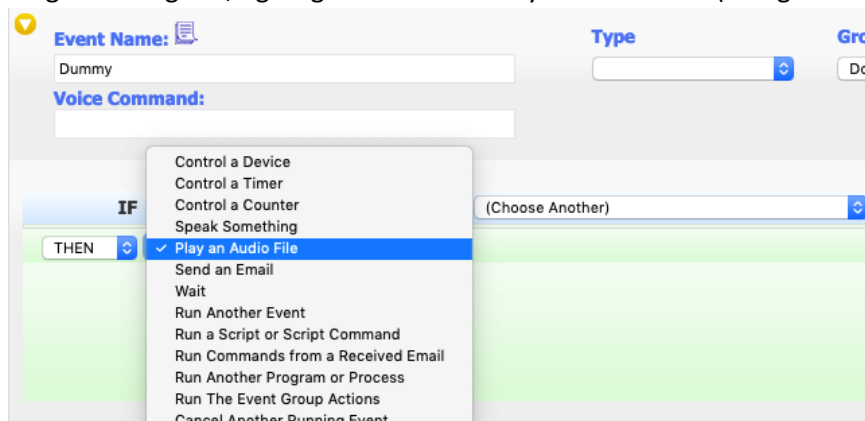
Figur 4.41 Event som kjører et skript

4.9.3 Sonos-event for lydeffekter

Når man setter opp et system med høyttalere, kan det i noen tilfeller være praktisk å ha en lydeffekt som spilles av dersom en spesifikk ting skjer. Ved å bruke den gratis Sonos-plug-in-en kan man spille av lydfiler som man har lastet opp til HomeSeer.

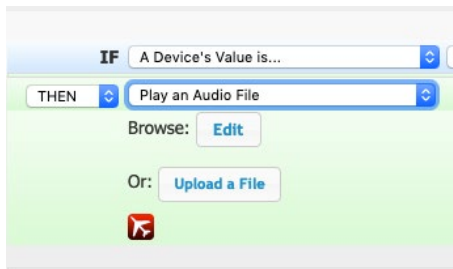
Dette kan gjøres på forskjellige måter. Dersom man ikke vil benytte seg av et eksternt program som WinSCP, kan man laste opp lydfiler direkte i HomeSeer på følgende måte:

- Opprett en dummy-event
- Velg en betingelse, og velg hendelse til «Play an audio file» (se Figur 4.42)



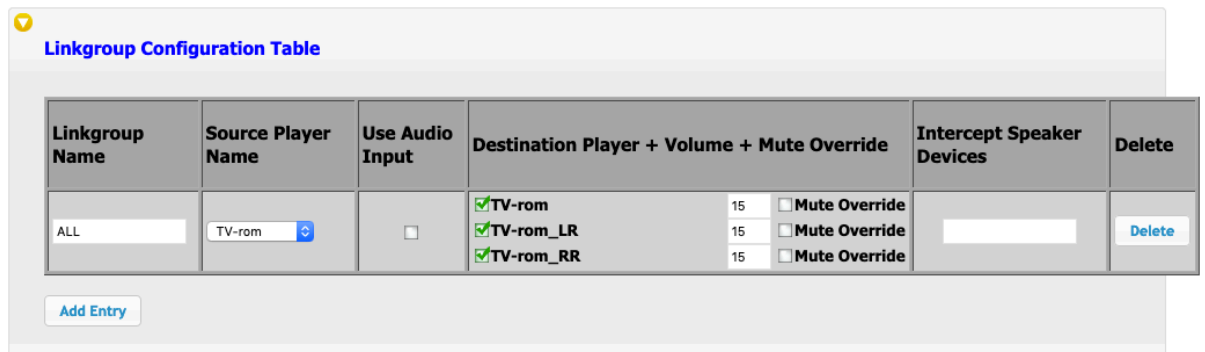
Figur 4.42 Velg for å laste opp lydfiler

- Trykk «Upload a file», og finn lydfilen du vil laste opp. Dette er testet med MP3-filer, 128-320kbps, CBR, 44.1-48 khz sampling rate. 96khz fungerer ikke.




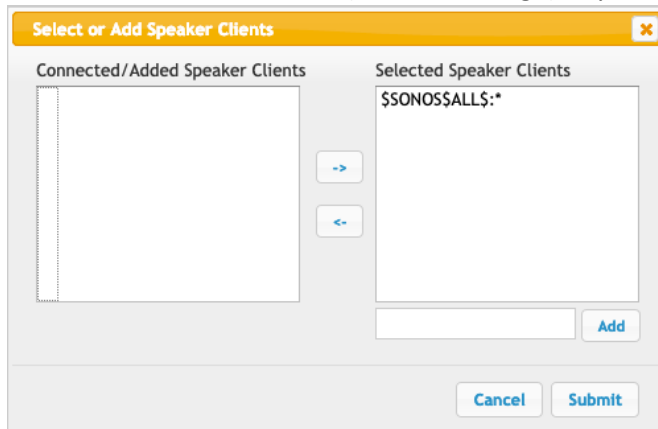
Figur 4.43 Opplastingsmeny

- Etter å ha lastet opp filen vil det stå på skjermen hvor lydfilen ligger på systemet. I vårt eksempel ligger den under: /usr/local/HomeSeer/Media/MGS.mp3
- Kopier denne adressen. I stedet for "Play audio file", velg "Speak something". Lim inn adressen i feltet som dukker opp.
- Siden vi bruker Sonos til å spille av lyden, må vi fortelle HomeSeer at vi vil sende lyden dit. Først går man til plug-in-siden «Plug-Ins» → «Sonos» → «Config». Her må man sette opp en linkgruppe. I vårt tilfelle kaller vi den «ALL» (husk dette navnet).



Figur 4.44 Meny for linkgruppe i Sonos-plug-in-en

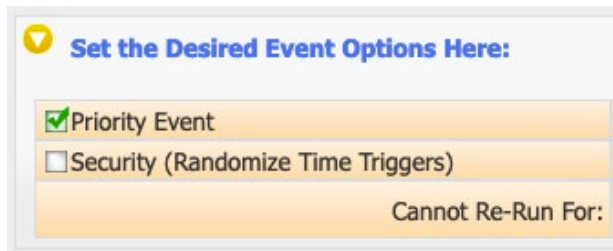
- Gå så tilbake til events, og finn frem eventen med lydfilen som skal spilles av. Trykk så på dette ikonet for å få opp avanserte alternativer: 
- Trykk på «Edit speaker client list».
- I tekstboksen nede til høyre spesifiserer man hvor man vil sende lyden. Syntaksen for Sonos-plug-in-en er «\$SONOS\$<linkgruppenavn>\$». I vårt tilfelle kalte vi gruppen «ALL», så vi skriver inn «\$SONOS\$ALL\$» (uten hermetegn). Trykk så «Add», og deretter «Submit».



Figur 4.45 Å legge til høyttaler krever spesiell syntaks

- Nå skal lyden spilles av når eventen kjøres.

- Av og til kan det oppstå problemer med lyden, ved at det noen ganger er det en god del forsinkelse (et par minutter, noen ganger sekunder), eller ingen lyd i det hele tatt. Dette fikses ved å huke av for «Priority event» under event options:



Figur 4.46 Prioritert event

4.10 Laveste batteri

Vi lager virtuell enhet som nevnt tidligere, her velger vi å navngi enheten Laveste Batteri, da dens oppgave er å opplyse om batteristatusen til den enheten med lavest batteri. Vi velger å plassere den i rommet «Virtuell», hvor vi har alle våre manuelt lagde virtuelle enheter, og tilegner den funksjon «Batteri». Disse tilegnelsene kan som nevnt endres senere hvis man finner en bedre/mer hensiktsmessig måte å filtrere enhetene på. I Figur 4.47 ser man at vi har valgt å ha alle avhakninger åpne, det er spesielt viktig at «Status Device» er åpen, da vi skal lage noen knapper senere i kapitlet.

Figur 4.47 Laveste batteri-konfigurasjon

Den siste fanen for manuelt lagte virtuelle enheter er «Status Graphics». Som vi ser i Figur 4.48 legger vi til to «Range»-verdier og en «Single»-verdi. Vi velger å ha en rekkeviddeverdi fra 0-100 med suffiks %, hvor vi inkluderer verdier. Denne verdien skal vi bruke til å vise den laveste batteriprosenten i systemet vårt. Den neste verdien er fra 101-254. Dette er alle ugyldige verdier når det kommer til

batteriprosent og har dermed fått prefiksen «Ugyldig verdi». «Single»-verdien 255 gir vi navnet/statusen «Lavt batteri varsel». Verdt å merke seg at vi under «Status-Control» velger å la alle verdiene våre være status. Dette er siden vi ikke skal kunne styre noen av disse verdiene manuelt.

Value	Status	Row	Column	Column Span	Status-Control
Start: 0 End: 100	Prefix: <input type="text"/> Suffix: % Dec Places: 0 <input checked="" type="checkbox"/> Include Values <input type="checkbox"/> Has Additional Data <input type="checkbox"/> Has Scale Value Offset: 0	0	0	0	both control status Delete
Start: 101 End: 254	Prefix: Ugyldig verdi Suffix: <input type="text"/> Dec Places: 0 <input type="checkbox"/> Include Values <input type="checkbox"/> Has Additional Data <input type="checkbox"/> Has Scale Value Offset: 0	0	0	0	both control status Delete
255	Lavt batteri varsel Value Offset: 0	0	0	0	both control status Delete

Figur 4.48 Laveste batteri Status Graphics 1

Her kan vi kan også legge til egne bilder som nevnt tidligere. Det har vi tenkt å gjøre her, slik vi får en fin grafikk på hvordan statusen til det laveste batteriet er. Vi ser på Figur 4.49, Figur 4.50 og Figur 4.51 at verdi 0-3 tilsvarer grafikk med et flatt batteri, 4-36 viser batteri som er 25% fullt, 37-64 viser 50% fullt, 65-89 viser 75%, 90-100 viser 100% fullt, og verdier 101-255 viser en grafikk som tilsvarer at det er feil på batteriet.

Value	Graphic	Graphic Path
Start: 0 End: 3	<input checked="" type="radio"/>	/images/HomeSeer/status/custom.gif Edit Delete
Start: 4 End: 36	<input checked="" type="radio"/>	/images/HomeSeer/status/custom.gif Edit Delete
Start: 37 End: 64	<input checked="" type="radio"/>	/images/HomeSeer/status/custom.gif Edit Delete
Start: 65 End: 89	<input checked="" type="radio"/>	/images/HomeSeer/status/custom.gif Edit Delete
Start: 90 End: 100	<input checked="" type="radio"/>	/images/HomeSeer/status/custom.gif Edit Delete
Start: 101 End: 255	<input checked="" type="radio"/>	/images/HomeSeer/status/custom.gif Edit Delete

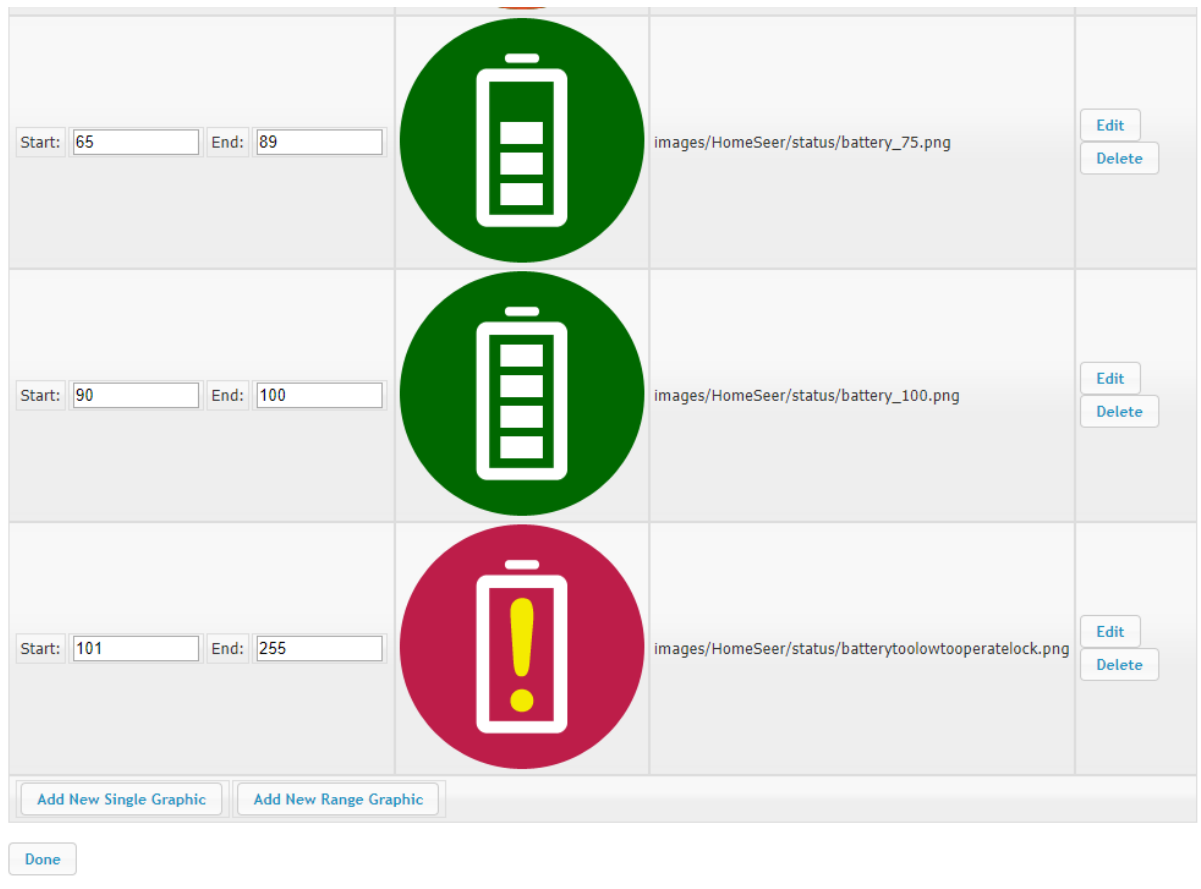
Add New Single Graphic Add New Range Graphic

Done

Figur 4.49 Laveste batteri Status Graphics 2

Edit Status Graphic for device Laveste batteri			
Value	Graphic	Graphic Path	
Start: <input type="text" value="0"/> End: <input type="text" value="3"/>		images/HomeSeer/status/battery_0.png	Edit Delete
Start: <input type="text" value="4"/> End: <input type="text" value="36"/>		images/HomeSeer/status/battery_25.png	Edit Delete
Start: <input type="text" value="37"/> End: <input type="text" value="64"/>		images/HomeSeer/status/battery_50.png	Edit Delete
Start: <input type="text" value="65"/> End: <input type="text" value="89"/>		images/HomeSeer/status/battery_75.png	Edit Delete

Figur 4.50 Laveste batteri Status Graphics 3



Figur 4.51 Laveste batteri Status Graphics 4

Når vi nå går på hjem-skjermen vil det se ut som Figur 4.52 viser. Vi ser her at det er en av- og en på-knapp, som vil forsvinne når vi lager knappene som vist i neste delkapittel. Vi ser også at den ligger på 0%, fordi verdien på enheten nå er satt til 0. Vi har enda ikke satt opp speilingen av laveste batteristatus i systemet til denne enheten. Dette kommer vi også tilbake til i de neste delkapitlene.

Status	Room	Function	Name	Address	Last Change	Control
<input type="checkbox"/> 0%	Virtuell	Batteri	Laveste batteri			<input type="button" value="On"/> <input type="button" value="Off"/>

Figur 4.52 Laveste batteri før scriptknapper

4.10.1 Scriptknapper

Som nevnt tidligere har vi tenkt å lage noen knapper som skal ha hver sin funksjon. Måten vi gjør det på er å først legge til de scriptene vi skal bruke i script-mappa på Pi-en (for å se hvordan man gjør det, se kapittel 4.7.3). Når det er gjort skal vi benytte oss av kommandoen «DeviceScriptButton_AddButton» som vi finner i databladet HS3Help. Ved hjelp av databladet kan vi finne ut hvordan vi kan benytte oss av denne kommandoen og hva de ulike parameterne betyr.

Vi går til Tools → Control Panel. Her vil vi få opp et vindu hvor vi kan skrive inn en umiddelbar scriptkommando. For å lage knappene skriver vi følgende, en kommando av gangen:

```
&hs.devicescriptbutton_addbutton(<refID>,"Oppdater",1001,"LowBatteryCheck.vb","","",1,1,1)
```

```
&hs.devicescriptbutton_addbutton(<refID>,"Send Rapport",1002,"BatteryReport.vb","","",1,2,1)
```

<refID> er referanseID-en til enheten, slik som nevnt tidligere. OBS! <> skal ikke være med i skriptet

"Oppdater"/ "Send Rapport" er teksten på knappen

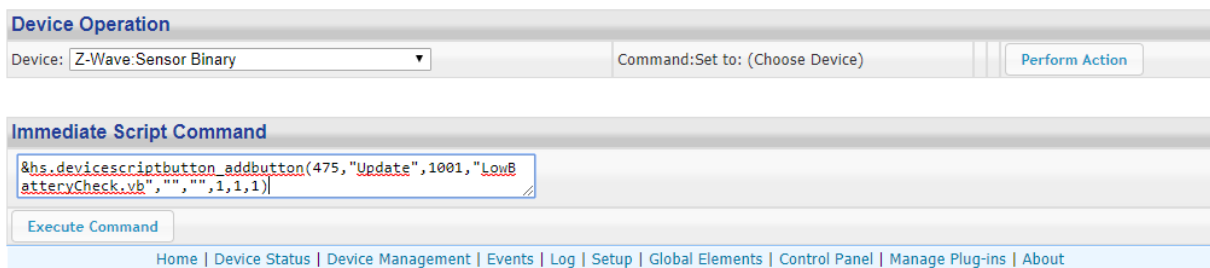
1001/1002 er en unik verdi for hver knapp på enheten

LowBatteryCheck.vb/BatteryReport.vb er navnet på scriptene vi vil kjøre

Den første "" inneholder funksjonen som skal kjøres, er den tom kjøres main

Den andre "" er en valgfri variabel som legges til i skriptet som parameter

1,1,1/1,2,1 er hvordan knappene er plassert på hjemskjermen. Se kap.4.6.3 for utdypende informasjon.



Figur 4.53 Scriptknapp oppretting

Status	Room	Function	Name	Address	Last Change	Control
<input type="checkbox"/> 0%	Virtuell	Batteri	Laveste batteri			Oppdater Send Rapport

Figur 4.54 Laveste batteri med oppdaterte knapper

Når scriptet er skrevet inn trykker vi på «Execute Command». Når vi har gjort dette med begge kommandoene har «on»- og «off»-knappene forsvunnet og blitt erstattet med «Oppdater» og «Send Rapport». Det er også verdt å merke seg her at vi står under device «Z-Wave:Sensor Binary», som vi ser ut i fra Figur 4.53. Det har ingenting å si hvilken enhet vi har stående her, da vi spesifiserer hvilken enhet vi refererer til ved hjelp av «refID» i selve scriptkommandoen.

Vi kommer til å gå mer inn på selve scriptene i kapittel 6.5 og forklare grundig hva hvert av de gjør.

4.10.2 Events

For at scriptene skal fungere slik vi vil de skal, må vi lage noen events. Vi lager ett event for varsling og rapportering, og ett for oppdatering.

4.10.2.1 Varsel og rapport

Dette eventet skal gjøre slik at det blir sendt en mail til valgt e-post adresse med informasjon om batteristatusen til alle batterier i systemet hvis en av følgende kriterier blir møtt:

- Hvis ett av batteriene i systemet har lavere batteriprosent enn 10
- Hvis ett av batteriene i systemet har en batteriprosentverdi høyere enn 100
- Hvis klokka er 08.00 den første lørdagen i hver måned.

The screenshot shows the configuration for an event named "Batteri - Varsel og rapport". The event type is set to "Status" and the group is "Status". The configuration is as follows:

- IF** This device had its value set and is less than...
 - Virtuell Batteri Laveste batteri is less than in the range... (value)%
 - and is 10%
- OR IF** This device had its value set and is greater than...
 - Virtuell Batteri Laveste batteri is greater than in the range... (value)%
 - and is 100%
- OR IF** The time is this:
 - Time: 08.00.00
- AND IF** The date is a specific day of the week of the month.
 - Today is the First of Any Month
- THEN** Control a Device
 - Use 'clear' in the Set Display field to CLEAR the current device string setting.
 - Virtuell Batteri Laveste batteri | Send Rapport
 - Set Device String to:
 - After Waiting: 0 Days 0 Hours 0 Minutes 0 Seconds

Figur 4.55 Batteri - Varsel og rapport fullstendig

The screenshot shows the same event configuration as Figure 4.55, but in a collapsed view. The configuration is summarized as follows:

- IF** Virtuell Batteri Laveste batteri was set and has a value that is less than 10%
- OR IF** Virtuell Batteri Laveste batteri was set and has a value that is greater than 100%
- OR IF** The time is at 08.00.00
- AND IF** The date is the First lørdag of Any Month
- Then** Set Device Virtuell Batteri Laveste batteri to Send Rapport

Figur 4.56 Batteri - Varsel og rapport kollapset

Figur 4.55 viser hvordan det vil se ut når vi setter opp dette eventet og Figur 4.56 viser hvordan det vil se ut nå vi har minimert alle faner i eventet. Basert på den tiden vi har prøvd ut systemet har vi valgt at en batterirapport en gang i måneden vil være mer enn nok, dessuten vil vi ved hjelp av eventet vi går igjennom i 4.10.2.2 kunne se batteriprosenten til det laveste batteriet i systemet vårt på hjemskjermen. Det er også mulig å kjøre dette scriptet ved å trykke på knappen «Send Rapport» hvis man ønsker en rapport akkurat nå.

I scriptet batterirapport har vi lagt til en ekstra kolonne i rapporten, slik at vi kan legge ved batteritype i tillegg til infoen som allerede er i scriptet. Dette gjorde vi siden vi synes det var noe som manglet i den opprinnelige rapporten vist i Figur 4.57. Grunnen til vi følte at dette var viktig informasjon var at hvis man har et system med hundrevis av batteridrevne enheter er det ikke lett å vite hvilken batteritype hver enhet bruker. Dette vil da være en grei og oversiktlig måte å få denne informasjonen på. I vårt lille system har vi allerede fire forskjellige batterityper, man kan tenke seg hvor mange forskjellige typer det ville vært i et større system.

Battery Report 🖨️ 📧

krisrodn@gmail.com 15:50 (for 5 minutter siden) ☆ ↶ ⋮

til meg

Ref ID	Battery %	Device Location / Name	Last Updated
[256]	96%	Living room Multisensors Battery	28.02.2019 02:40:03
[444]	80%	Misc Remotes Keyfob - Battery	28.02.2019 15:39:58
[454]	66%	Hallway Multisensors Battery	28.02.2019 13:28:49
[466]	0%	Node 14 Z-Wave Battery	01.01.0001 00:00:00

[4 Battery Devices] - Report created 28.02.2019 15:50:36

Figur 4.57 Batterirapport - opprinnelig, Gmail

For å beskrive hvordan vi har lagt til denne informasjonen, må vi begynne med å beskrive hvordan scriptet finner enheter å putte i lista vist i Figur 4.57. Scriptet filtrerer vekk alle enheter som ikke har «Battery» i navnet sitt, og på den måten får vi navigert oss inn på alle subkomponenter som omhandler batteri. Inne på disse subkomponentene legger vi til batteritypen i notater. Det vi videre gjorde var å legge til parameteren for disse notatene inn i scriptet under den nye kolonnen vi har laget. Når alt dette er gjort vil rapporten bli seende ut som i Figur 4.58

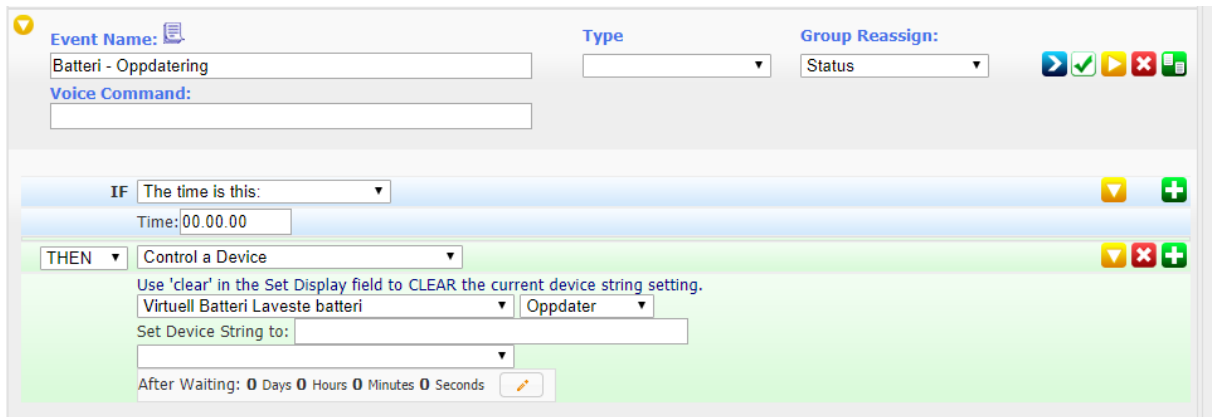
Ref ID	Battery %	Battery type	Device Location / Name	Last Updated
[256]	96%	Unknown	Living room Multisensors Battery	01.03.2019 02:38:50
[444]	80%	Unknown	Misc Remotes Keyfob - Battery	28.02.2019 15:39:58
[454]	66%	Batteritype: CR123A	Hallway Multisensors Battery	01.03.2019 13:28:43
[466]	90%	Unknown	Node 14 Z-Wave Battery	01.03.2019 02:17:29

[4 Battery Devices] - Report created 01.03.2019 18:10:05

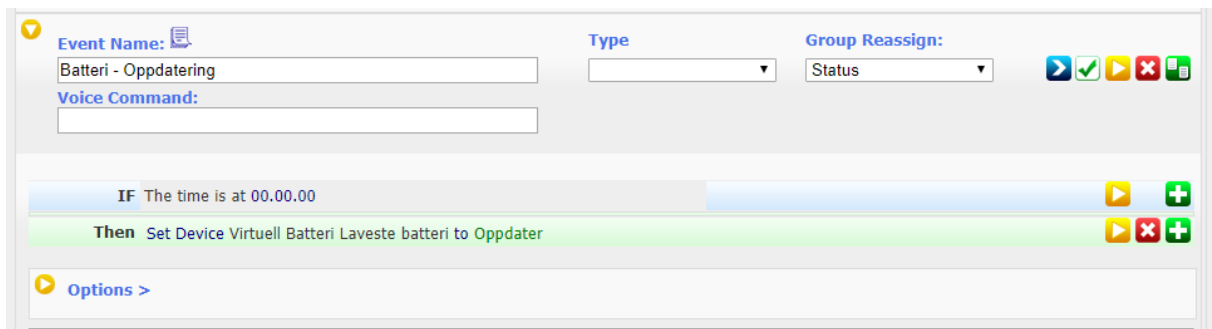
Figur 4.58 Batterirapport - med batteritype, Gmail

4.10.2.2 Oppdatering

Dette eventet skal oppdatere statusen til enheten vist i Figur 4.54. Vi ser at statusen er 0%, grunnen til dette er at vi ikke har kjørt gjennom skriptet enda og den som standard har stått som «off» som er det samme som 0. Etter eventet er kjørt vil denne prosenten oppdatere seg til den med lavest verdi i systemet. Dette eventet fungerer slik at den hver dag klokka 00.00.00 vil oppdatere statusen til enheten til den laveste batteriprosenten ved å kjøre scriptet.



Figur 4.59 Batteri - Oppdatering fullstendig



Figur 4.60 Batteri - Oppdatering kollapset

Det er vår erfaring av batteriprosenten ikke endres ekstremt mye i løpet av en dag, og vi har dermed valgt å kun kjøre oppdatering en gang daglig. Men hvis man absolutt vil oppdatere statusen kan vi også her trykke på «Oppdater»-knappen.

4.11 Vårt system

På grunn av oppgavens natur, har vi valgt å sette søkelys på et begrenset antall fysiske komponenter der vi viser flere typer funksjonalitet avhengig av hvilke betingelser som er oppfylte. På denne måten kan man se mye av potensialet til HomeSeer, samtidig som vi holder oss innenfor budsjettet. Virtuelle komponenter har vært en viktig del av det ferdige systemet. Vi har merket at kompatibiliteten for stemmestyring er mangelfull selv om denne også har stort potensiale.

Av Z-Wave-komponenter har vi valgt en fjernkontroll, en multisensor, en stikkontakt og en dørsensor i vår endelige løsning. Vi har brukt profiler (ved hjelp av virtuelle komponenter) slik at vi har kunnet få forskjellig respons fra systemet avhengig av hvilke profiler som er i bruk. For eksempel vil det skje en ting dersom bevegelsessensoren aktiveres om morgenen, mens en annen ting vil skje dersom det er bevegelse om natten. «Hjemme/borte/ferie»-profilen er også med på betingelser i events. Fjernkontrollen vår, en Fibaro Keyfob, brukes slik at vi enkelt kan demonstrere ulike deler av systemet. Med denne kan vi endre hvilke profiler som brukes, kontrollere musikk, og sjekke været, med mer (se Appendiks C).

Philips Hue brukes mye i forbindelse med de andre komponentene i systemet vårt. Ulike lyssignaler og farger blir brukt avhengig av hvilken sensor eller hendelse som skjer i systemet, slik at vi får vist

hvordan de kan brukes. Samtidig gir de også gode muligheter som visuelle indikatorer på at systemet reagerer.

Stemmestyringen med Google Home bruker vi mest i forbindelse med kontrollering av lys, da støtten for komponenter ofte er knotete. Selv med lysene har vi hatt problemer (se kapittel 6.4), men grunnleggende kommandoer er på plass (se Appendiks C for eksempler).

Det bør nevnes at i systemet er det lagt vekt på at det lett skal kunne demonstreres hvilke funksjoner som finnes. Av den grunn har for eksempel Keyfob-en knapper for å stille inn på hvilken tid på døgnet det er, og lys slås på/av i løpet av få sekunder. Dette vil ikke være naturlig å ha i et reelt system.

5 Testing

På grunn av måten systemet fungerer på, har testingen fungert med hver enkelt komponent der ingen enkeltkomponent vil ha noe å si for resten av systemet om den skulle feile. Unntaket er ved hardwarefeil på Raspberry Pi-en eller Z-Wave-kortet, men dette vil stort sett fungere ordentlig eller ikke i det hele tatt. Her er det også liten sjanse for brukerfeil, med unntak av logiske brister i events når de blir satt opp. Også her vil ikke en enkelt event ha noe å si for de andre delene av systemet.

5.1 Logikk

Mesteparten av den logiske testingen har blitt gjort når vi har satt opp ulike events. Vi har da satt opp ulike triggers og conditions for å teste om systemet gjør det vi vil i de ulike testene. Et eksempel på dette var da vi satte opp en veggplugg for å styre en varmeovn. Det første vi måtte gjøre var å finne ut hva som skulle være oppfylt for at ovnen skulle slå seg på, og vi kom da frem til følgende:

- Måtte være temperatur (målt av en multisensor) på mindre enn 21grader
- Og hovedstatusen måtte være satt til Hjemme
- Og tidsstatusen måtte ikke ha verdi natt

Vi gikk et par runder for å komme frem til løsningen som er nevnt over. Først startet vi kanskje med at tidsstatusen måtte være morgen eller kveld, men da utelukket vi jo dag. Vi kom dermed til slutt frem til at det mest logiske ville være at ovnen skulle skru seg på så lenge det var mindre enn 21 grader og hovedstatusen var satt til hjemme og det ikke var natt. Vi kom frem til dette da vi ikke ville at ovnen skulle stå på og varme om natta da ingen er våkne.

Denne tankegangen var også det som fikk oss til å sette logikken på tidsstatusene våre. Nattstatusen starter klokken 11 og morgenstatusen starter 6 på morgenen. Dette gjør dermed at denne tidsstatusvariabelen direkte innsatt i veggpluggeventet fører til at det blir «kaldt» på natta og at når man står opp i 7-8 tiden så er det varmt igjen.

Men vi må jo selvsagt ha en motpart til å skru på ovnen, å skru den av. Dette hadde følgende krav:

- Måtte være temperatur (målt av en multisensor) på mer enn 24grader
- Eller hovedstatusen var satt til ikke hjemme
- Eller tidsstatusen var satt til natt

Vi ser her at vi har satt motsatt «fortegn» på hovedstatus og tidsstatus, samt at vi har endret fra og-setninger til eller-setninger. Her vil altså pluggen skru seg av hvis statusen ikke er satt til hjemme, eller hvis tidsstatusen er satt til natt eller det er mer enn 24 grader.

Dette var bare ett av eksemplene på hvordan vi har tenkt når vi har satt opp logiske events og teste dem for smutthull. Det er ganske mye man må tenke på for at systemet skal fungere akkurat slik man vil, og dermed er det veldig viktig at man tenker nøye gjennom hva man vil systemet skal gjøre, og luke ut alle logiske brister. Man kan med fordel sette opp et flytdiagram eller lignende slik man får kontroll over tankene sine, vi merket at det var fort gjort å gå seg «vill» i logikken.

5.2 Plug-in

Vi har testet ut litt diverse plug-ins i løpet av arbeidet med oppgaven, og vi skal nå gå kort gjennom de plug-in-ene vi har prøvd ut og prøve å komme med positive og negative sider med de ulike. Noe som er et generelt «problem» med dette systemet er at mange av plug-in-ene er laget av tredjeparter og i

hovedsak privatpersoner, som kan føre til at en plug-in man har brukt plutselig ikke får flere oppdateringer siden tredjeparten har sluttet å kjøre oppdateringer på plug-in-en. Problemet kan løses ved at man lager plug-ins selv, men dette krever da selvsagt at man er datakyndig og liker å programmere.

5.2.1 Chromecast

Chromecast er en plug-in vi har prøvd i hovedsak for å få til Text To Speech (TTS), noe som vil si at vi ikke har prøvd ut andre deler av denne plug-in-en. Grunnen til at vi har valgt å ta i bruk Chromecast er at den bruker Googles stemmeassistent ved bruk av TTS, noe som gjør at det høres mye mer naturlig ut enn med HomeSeers innebyggede TTS.

5.2.2 Yr Weather

Yr Weather har vi brukt for å ha muligheten til å bruke fremtidsvarsel og nåværende vær som en av betingelsene i events. Plug-in-en fungerte helt greit, men et minus med den var at den kostet penger, noe lignende plug-ins på markedet ikke gjør. Det positive med denne plug-in-en er at statusene vil alle være på norsk, noe som gjør at det vil høres og se mye mer naturlig ut hvis hele systemet inkludert TTS er på norsk.

5.2.3 WeatherXML

WeatherXML er en av de andre plug-in-ene som omhandler vær, men imotsetning til Yr Weather så er denne gratis. Det negative med denne er at all tekst er på engelsk, noe som gjør det unaturlig å bruke hvis man vil ha et fullnorsk system. Ellers er bruksområdet av denne plug-in-en lik YR Weather.

5.2.4 UltraNetatmo3

UltraNetatmo3 er en plug-in til værstasjonene til Netatmo. Vi tok i bruk en innendørsmodul og en utendørsmodul. Den fungerte meget bra og har litt flere målinger enn det en multisensor har. Med denne fikk vi både fuktighet, CO₂-nivå, støymåling med mer, noe som kan gjøre at vi kan styre events med enda høyere presisjon. Det er også en liten overlapp til de andre vær-plug-in-ene da vi kan bruke utendørsmodulen til å lese av utetemperaturen. Denne koster også penger, men vil være verdt pengene hvis man har flere enheter fra Netatmo. Vi valgte å ikke ta i bruk denne i den endelige løsningen vår på grunn av plassmangel, altså at vi har andre plug-ins vi så var bedre å bruke i et endelig system enn denne.

5.2.5 JowiHue

JowiHue er den plug-in-en som brukes til å styre lysene våre, det er også verdt å få med seg at man med denne plug-in-en og en annen plug-in kan styre andre lys en kun Philips sine. Dette kommer vi ikke til å gå noe særlig inn på da vi ikke har testet det ut, men det er vertfall en mulighet. Det som er positivt med denne plug-in-en per dags dato er at utvikleren er veldig aktiv på forumet og prøver å rette opp i det som kommer av feilmeldinger fra brukere. Vi kan også lage mange forhåndsdefinerte scener og lysinnstillinger via plug-in-en, noe som vi igjen kan bruke i eventene våre. Det negative er måten plug-in-en håndterer stemmestyringen opp mot Google Home, men det vil forhåpentligvis bli løst med tiden.

5.2.6 Sonos

Sonos plug-in-en har vi brukt for å styre vår Sonos høyttaler. Vårt inntrykk av denne plug-in-en er at den er veldig kaotisk satt opp og at det er vanskelig å vite hva som er hva av enhetene som kommer inn via plug-in-en. Denne kunne absolutt ha blitt gjort litt mer brukervennlig, men på den annen side

så vil man sitte med nesten like mye kontroll som man vil med Sonos sin egen app til mobil og pc-enheter, det kreves bare at man har tungen i rett munnen når man holder på med plug-in-en. Planen vår var å bruke Sonos-høytalerne som hovedhøytaler for systemet vårt, men dette viste seg å være problematisk på grunn av manglende kompatibilitet mellom Sonos og Google Home. Dette er rett rundt svingen og det er i skrivende stund startet betaarbeid mellom Sonos og Google.

6 Diskusjon

I dette kapitlet vil vi først ta for oss fremdriftsplanen vi satte oss før arbeidet med rapporten startet og risikoene vi mener kunne skadet arbeidet vårt. Videre vil vi gå inn på problemer og utfordringer vi har hatt med både system, hardware/software og selvlagde scripts.

6.1 Redegjøring av fremdrift

Vi har delt opp fremdriftsplanen vår i tre deler; 1. Forstudie, 2. Arbeid frem mot midtveis og 3. Arbeid frem mot EXPO. Vi har i løpet av oppgaven måttet endre litt på fremdriftsplanen vår. Den første delen fullførte vi med null avvik, det er derimot i del to og tre vi har sett at det har blitt litt endringer i forhold til den opprinnelige fremdriftsplanen vist i Figur 9.2 kontra den endelige vist i Figur 9.3. De generelle forskjellene i disse to delene er at det har gått mer tid på arbeid med rapport og innhenting av kompetanse enn planlagt, samt at det har gått relativt fort å sette opp events og komponenter. Dette har da ført til at tiden vi sparte inn på komponenter og events er blitt «spist opp» igjen av arbeid med rapporten. En av hovedforskjellene mellom opprinnelig og endelig fremdriftsplan er at del to er blitt betydelig mindre, noe som igjen har ført til at del tre er blitt tilsvarende større. Grunnen til denne endringen er at midtveispresentasjonen ble flyttet to uker frem i tid i forhold til den opprinnelige planen. Vi har også hatt to uker ferie i denne perioden, noe som kan finnes igjen i timelisten vår, men dette har også vært med i den opprinnelige planen. Vi har også flyttet frem noe av EXPO arbeidet i den endelige planen vår, grunnen til dette var at vi ville ha med vår endelige EXPO løsning i oppgaven, noe som da ville bety at denne løsningen måtte være på plass før vi kunne gjøre ferdig oppgaveskrivingen.

6.2 Redegjøring av risikoanalyse

Med tanke på at denne oppgaven i hovedsak er utprøving av systemet for Høgskulen på Vestlandet, så har vi vurdert det til at det ikke er like kritisk hvis vi ikke blir «ferdige» med utprøvinga, kontra en oppgave som for eksempel skal ha klart et ferdig produkt. Dette bærer da risikoanalysen vår i B.3 litt preg av. Hoved risikoene går på hardware/software-problemer, sykdom og interne konflikter. Når det gjelder hardware er det ikke så mye vi får gjort hvis hardwaren blir ødelagt, annet enn å måtte kjøpe komponenter på nytt og dermed muligens overskride budsjettet. Hvis softwaren slutter å fungere er det ikke så mye vi kan få gjort i løpet av oppgaven annet enn å prøve lignende software. Hvis denne derimot ikke fungerer som planlagt vil vi måtte lage egne plug-ins, noe som ville gått utover denne oppgavens rammer. Sykdom og interne konflikter er risikoer i ethvert prosjekt, men på grunn av oppgavens kjerne så vil ikke dette bli noe stort problem.

6.3 Problemer med Linux og oppsett

6.3.1 Pi-problemer

Det første problemet vi støtte på var ved overføring av plateavbildingsfilen fra PC/Mac til minnekortet. Vi fikk det ikke til ved bruk av Mac, så vi benyttet oss dermed av PC. Vi brukte som forklart tidligere programmet balenaEtcher [11] til denne overføringen. Vi fikk en feilmelding når filen var lastet over fra PC til minnekort, men denne feilmeldingen kom som følge av at minnekortet nå var gjort om til en lesbar Linux-fil, noe som gjorde at Windows ikke klarte å lese hva som stod på kortet lenger. Dette var altså ikke en reell feil, men det er greit å være klar over hvis man skal gjøre det samme selv ved en senere anledning.

Det andre «problemet» vi støtte på var at man ikke trengte å skrive inn brukernavn og passord for å komme seg inn på systemet fra find.homeseer.com eller IP-en til enheten så lenge man var på samme

nettverk. Dette anså vi som et mulig sikkerhetsproblem, da man er meget utsatt hvis en hacker først kommer seg inn på nettverket ditt. Det viste seg å være en enkel løsning på problemet, og det var å huke av en boks under «Tools»→«Setup»→«Network», hvor det står at man må logge inn med brukernavn og passord selv om man er på det lokale nettverket.

Det tredje problemet vi møtte på var når vi skulle inn på selve Linuxsystemet. Vi hadde lagd et passord, men problemet var at vi ikke visste det tilhørende brukernavnet. Vi prøvde brukernavnet vi lagde i førsteoppsettet, noe som kun fungerte i web-API-et. Vi prøvde ID tilsendt på e-post, vi prøvde mailadressen vi registrerte enheten på, men ingenting fungerte. Vi fant ut hva brukernavnet var ved å gå inn på «Tools»→«System»→«Linuxtools», her så vi at standard brukernavn var «homeseer», slik som vist i Figur 6.1.

Enter system user to access Linux tools (default is homeseer):

Enter system password to access Linux tools:

Login

Figur 6.1 Linuxtools, brukernavn

6.3.2 Z-Wave

Hvis man prøver å legge til en node uten at den blir lagt til i systemet kan det tyde på at noden har vært i bruk i et annet nettverk tidligere og ikke blitt resatt før man prøver å koble den til det nye systemet. Måten man fikser problemet på er som nevnt å resette noden som vist i datablad, for så å importere noden på nytt.

Hvis man har problemer med en node kan det være naturlig å fjerne noden og legge den til på nytt igjen etter på. Vedvarer problemet, kan man følge rådet over og resette noden for så å importere den inn i systemet på ny. Man fjerner en node på samme måte som man legger til en node, bare at man nå velger Remove/Exclude a Node istedenfor Add/Include a Node i nedtrekksmenyen, se Figur 6.2.

The screenshot shows the HomeSeer web interface for Z-Wave controller configuration. The top navigation bar includes 'HOME', 'VIEW', 'TOOLS', and 'PLUG-INS'. The main content area is titled '+ Z-Wave Networks and Options' and 'Z-Wave Interfaces'. A specific interface named 'zWaveInt' is selected, with a green checkmark indicating it is the main interface for Home ID E633CC72. The configuration fields show 'Name: zWaveInt', 'Interface Model: Internal', and 'Serial Port: /dev/ttyAMA0'. The 'Actions' dropdown menu is set to 'Remove/Exclude a Node'. Below the configuration fields, there are 'ABORT' and 'CLOSE' buttons. A terminal window at the bottom displays the following log output:

```
Please wait...
Activate the 'Remove from Network' function on the device...
The controller found a node to be removed...
Remove Node from Network is removing a SLAVE node...
Node 12 was removed. Removing corresponding device(s) in HomeSeer.
Reloading (importing) node information...
Getting Z-wave node list for network E633CC72...
Found 6 Z-wave nodes
Syncing Z-wave nodes with device list...
Synchronize nodes finished. Number of device nodes to be created/added = 0
Saving network information to file: ZNet_E633CC72_2019-02-28_13.16.44.Zwave. Please wait...
Done!
```

Figur 6.2 Node – fjerning

Vi opplevde også problemer med en av multisensorene våre. Den mistet kontakten med sentralen kort tid etter inkludering, selv om den var under 6m fra sentralen samt at den var 20cm fra en veggplugg. Vi prøvde å optimalisere noden, fjerne og legge til på nytt, men ingenting fungerte. Så det endte opp med at vi fjernet hele noden.

Den ene veggpluggen vår kjører på en gammel versjon (25.25), mens den andre kjørte en nyere versjon (3.2). Dette er i seg selv ikke noe problem, men å oppdatere Fibaro-enheter i HomeSeer ikke rett frem. Selv om det er et valg å oppdatere fastvare når man er inne i Z-Wave-fanen på enheten, så er man avhengig av å ha .hex-filen. Denne filen får vi ikke tak i fra Fibaro, og det virker ikke som Fibaro er interesserte i å dele disse oppdateringsfilene til produktene sine heller. Eneste måten man kan oppdatere enhetene sine på er ved å bruke Fibaro sin egen hub, som selvfølgelig koster en del. Heldigvis er det relativt smertefritt å oppdatere ved å bruke Fibaro sin egen hub og deretter inkludere i vårt eget system igjen [17].

6.3.3 ZigBee

Vi prøvde å koble til en plugg fra Nookbox til Philips Hue bridge-en vår, men dette fungerte ikke. Etter litt søking fant vi ut at dette skyldtes at Philips Hue benyttet seg av ZLL, mens pluggen fra Nookbox benyttet seg av Zigbee Home Automation (ZHA). ZigBee har nå laget en ny protokoll som de nye enhetene skal kjøre, ZigBee 3.0.

6.4 Problemer hardware og software

6.4.1 Google Home

Stemmestyringen er kanskje det som har vært mest problemer med i løpet av prosjektet. Den har vist seg å være den mest ukonsekvente av komponentene vi har brukt. Det hender at den ikke forstår kommandoer for å kjøre visse events, selv om den har kjørt dem uten problemer tidligere. Det har også vært en unaturlig flyt i bruk av stemmestyring fordi man må si «aktiver» for å kjøre events. For eksempel «Hey Google, aktiver skift profil til hjemme», istedenfor «Hey Google, skift profil til hjemme». Man må også passe på at events ikke har samme stemmekommando som Google sine innebygde, ellers er det sjanse for at Google Home ikke vet hvilken kommando brukeren mener. Linking av Google Home til HomeSeer var også et problem da vi prøvde det mens Google sine servere var nede. Dette var naturlig, men det var vanskelig å feilsøke og vi fant ikke ut av det før serverne var oppe igjen. Et viktig punkt for at Google Home i det hele tatt skulle fungere, er at den oppkoblede Google-kontoen må ha nett- og appaktivitet slått på. Uten dette vil ingen stemmestyring fungere, og man får bare feilmelding om at man må sjekke Google Home-appen uten at det er spesifisert hva som er feil.

6.4.2 Jowi Hue

Vi hadde noen innkjøringsproblemer med lyspærene men det er usikkert hva som var grunnen. Lyspære-bridgen eller JowiHue viste tegn til å ha hengt seg opp, da den registrerte en lyspære som fortsatt lå i esken som på. Ingen lys-events ville kjøre heller. Løsningen var å kjøre en full reset, ved å bruke reset-knappen på lyspære-bridgen, samt sletting av filene JowiHue.db under \data\JowiHue og HSPI_JowiHue.exe.config fra HS3 base-mappa. For sikkerhets skyld slo vi JowiHue av og på under plug-ins, og kjørte omstart av HomeSeer. Etter dette fungerte alt som normalt.

6.4.3 Stemmestyring av lys

På grunn av det som sannsynligvis er manglende eller mangelfulle API-er kunne vi ikke kontrollere lys slik som vi ville med stemmestyring. Når vi ville stille inn på for eksempel lysstyrke på en lyspære eller gruppe, kunne vi be den sette lysstyrken til 50%. Dette ville da skje, men samtidig stilte HomeSeer også verdien til alle andre subkomponenter, inkludert farge, til «50». Siden fargeverdiene i HomeSeer går fra 0-65535, vil lyset alltid skifte til en type rødfarge dersom man ber om å skifte lysstyrke. Løsningen ble derfor å deaktivere stemmestyring for farge, slik at lysstyrke kunne stilles. Selv da ville den ikke fungere helt som den skulle, da den ofte satte lysstyrken til 99%, selv om vi ba den om 100%.

6.5 utfordringer med selvlagde ting

6.5.1 Scriptknapper

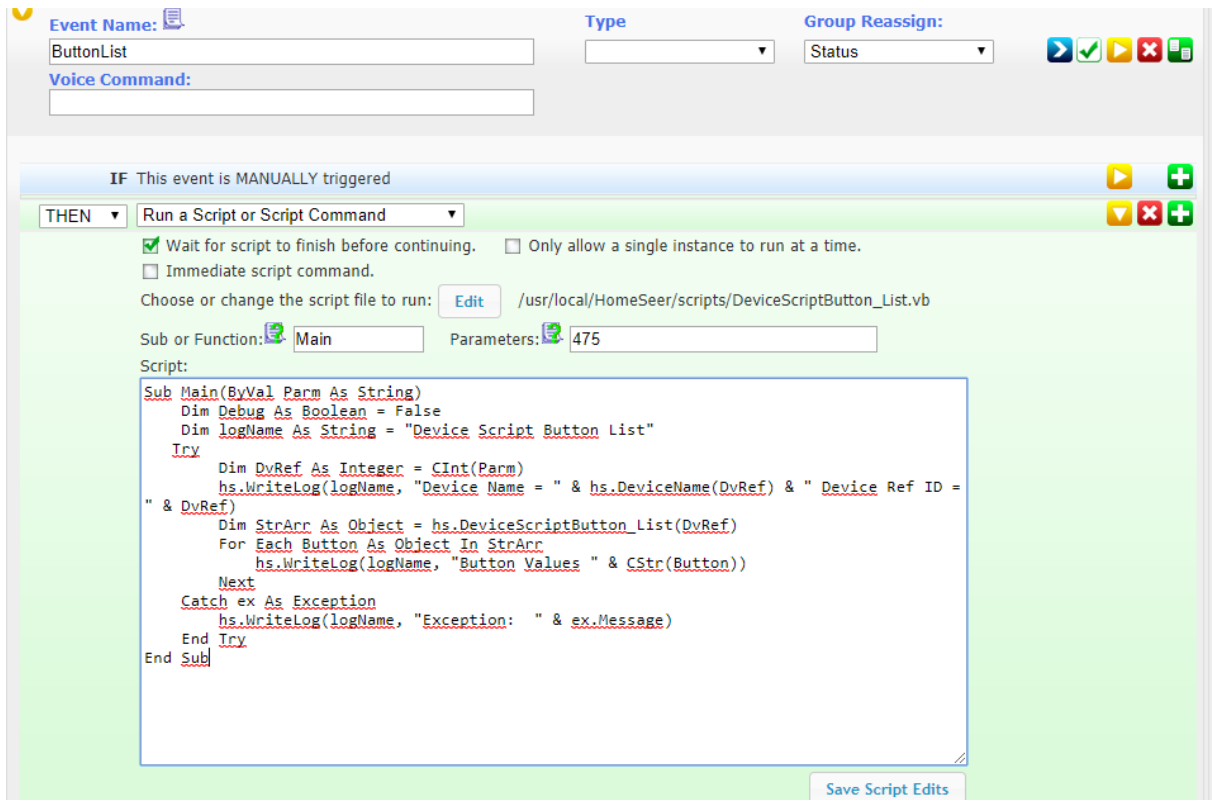
Som nevnt i kapittel 4.10.1 laget vi egne knapper til batteristatus-enheten vår, og som vi så der satte vi en unik «knappe-ID» til hver av knappene våre. Det viste seg å være en umulig jobb å endre noen parametere i en allerede eksisterende knapp. Hvis vi ville få til dette måtte vi ha slettet knappen og laget den på nytt med de parameterne vi ville hatt. For å slette en knapp måtte vi ha husket denne «knappe-ID-en» vi satte og deretter kjørt følgende kommando samme plass som vi lagde knappen:

```
&hs.devicescriptbutton_deletebutton(RefID,unik knappeverdi)
```

Men det er ikke sikkert at man alltid husker hvilke verdier de ulike knappene har, så vi leita oss fram i databladet og fant følgende kommando:

```
DeviceScriptButton_List
```

Vi forsøkte å kjøre denne på samme måten som både opprettelse og sletting av knapper, men vi fikk det ikke til å fungere. Vi spurte på forumet til HomeSeer og der fikk vi først hjelp av et par stykker som sa at vi måtte lage et eget script og dermed eget event for dette. Løsningene deres fungerte ikke for oss siden vi kjørte scriptet i Linux (og dermed med Mono), mens de kjørte sine i Windows med en kompiler som er mye mer tilgivende. Til slutt fikk vi hjelp av en i forumet [18] med løsning som vist i Figur 6.3 og med en output i loggen som vist i Figur 6.4.



Figur 6.3 Devicescriptbutton_List, fra sparkman

mar-19 09.18.14	Device Script Button List	Button Values Send Rapport,1002,BatteryReport.vb,,,1,2,1
mar-19 09.18.14	Device Script Button List	Button Values Oppdater,1001,LowBatteryCheck.vb,,,1,1,1
mar-19 09.18.14	Device Script Button List	Device Name = Virtuell Batteri Laveste batteri Device Ref ID = 475
mar-19 09.18.14	Z-Wave	Device: Hallway Plugg Switch Set to OFF
mar-19 09.18.08	Event	Running script and waiting: /usr/local/HomeSeer/scripts/DeviceScriptButton_List.vb("Main","475")
mar-19 09.18.08	Event	Event Trigger "Status ButtonList"
mar-19 09.18.08	Event	Event Status ButtonList triggered by the event page 'Run' button.

Figur 6.4 Devicescriptbutton_List, logg

Vi kunne selvsagt ha benyttet oss av kommandoen:

DeviceScriptButton_DeleteAll

Men da hadde alle scriptknapper blitt fjernet fra enheten, og dette ville vært veldig lite gunstig hvis vi hadde hatt flere scriptknapper på enheten.

6.5.2 Laveste batteri - Script

Scriptene vi har benyttet oss av i 4.10 er ikke selvlagde, men er et resultat av andres harde arbeid, samt vår lille touch på dem.

Ref ID	Battery %	Battery type	Device Location / Name	Last Updated
[256]	96%	Unknown	Living room Multisensors Battery	01.03.2019 02.38.50
[444]	80%	Unknown	Misc Remotes Keyfob - Battery	28.02.2019 15.39.58
[454]	66%	Batteritype: CR123A	Hallway Multisensors Battery	01.03.2019 13.28.43
[466]	90%	Unknown	Node 14 Z-Wave Battery	01.03.2019 02.17.29

[4 Battery Devices] - Report created 01.03.2019 18.10.05

Figur 6.5 Batterirapport - med batteritype, Gmail

I Figur 6.5 ser vi også at det står «Unknown» på noen av enhetene, noe vi har lagt inn i scriptet. Det vi la inn da var at hvis den ikke fant noen notater, så skulle det stå «Unknown» i denne kolonnen.

I dette scriptet sliter vi litt med sorteringa, da vi helst skulle sett at den sorterte basert på batteriprosent og ikke «RefID». Men vi har valgt å ikke legge alt for stor vekt på, da det viktigste for oss med scriptet var å få den informasjonen vi ville ha og ikke hvordan sorteringen av denne informasjonen ble.

Det er også litt problemer med formateringen av denne tabellen som er vist i Figur 6.5. Den ser veldig fin og oversiktlig ut når vi åpner med Gmail-app på telefonen eller i Gmail på pc, men når vi åpner med native e-post leser på telefon og pc så vil formateringen bli helt feil. Vi tror det har noe med at vi ikke har klart å «låse» kolonnene våre i scriptet, men er ikke helt sikre da ingen av oss er noe stødige på verken .net programmering eller html(som er tabellformatet). Denne feilen fant vi utav med en tilfeldighet, da vi åpnet denne mailen i en annen leser enn gmail sin. Vi har derfor ikke brukt noe tid på å prøve å rette opp denne feilen.

Vi gjorde også små endringer i scriptet «lowbatterycheck». Disse endringene gikk ut på å endre rekkefølgen på hvordan room og function kommer frem i scriptet, samt å legge på ekstra informasjon, se forskjellen på Figur 6.6 og Figur 6.7.

Status	Room	Function	Name	Address	Last Change	Control
<input type="checkbox"/>  66% - Multisensors Battery	Virtuell	Batteri	Laveste batteri		Today 17.15.40	Oppdater Send Rapport

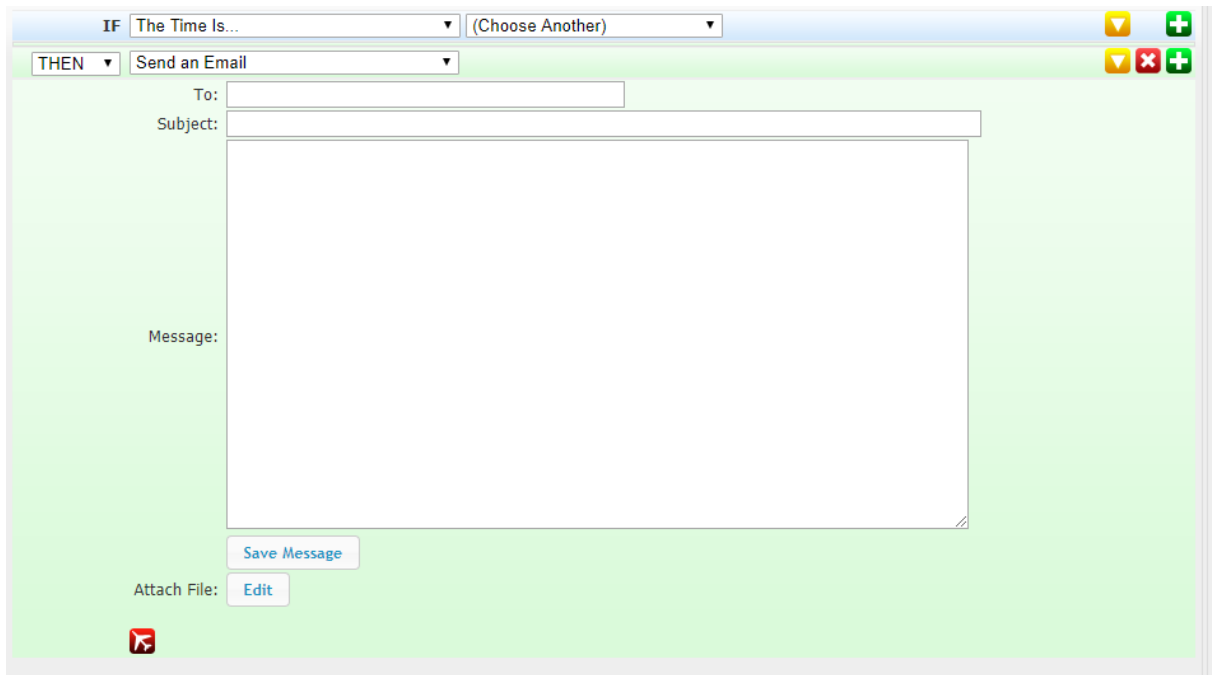
Figur 6.6 Laveste batteri - opprinnelig

Status	Room	Function	Name	Address	Last Change	Control
<input type="checkbox"/>  66% - Hallway Multisensors Battery	Virtuell	Batteri	Laveste batteri		Today 17.24.04	Oppdater Send Rapport

Figur 6.7 Laveste batteri – endringer

6.5.3 E-post

Vi har tidligere i oppgaven kort nevnt at vi under «Tools»→«Setup»→«E-mail» kan sette opp mail for å sende varsel eller lignende ut fra systemet, noe vi fant ut etter litt prøving og feiling. Vi tenkte først at vi kunne sette det per event ved å sette e-mail adresse inn i «To»-feltet i Figur 6.8. Dette fungerte ikke, vi fikk bare opp en lignende feilmelding som i Figur 6.9.



Figur 6.8 E-post, event

feb-28 15.52.58	Error	Cannot connect to POP or Imap email server, no host was specified in setup.
feb-28 15.52.50	Warning	Sending email To: krisrodn@gmail.com aborted after 4 tries
feb-28 15.52.50	Error	Exception sending email via (To:krisrodn@gmail.com, cc:, bcc:, Sub:Battery Report, Attach:) = Connection refused
feb-28 15.52.40	Z-Wave	Device: Hallway Plugg Power Set to 633,4 (W)
feb-28 15.52.39	Error	Exception sending email via (To:krisrodn@gmail.com, cc:, bcc:, Sub:Battery Report, Attach:) = Connection refused
feb-28 15.52.39	Z-Wave	Device: Hallway Plugg Power Set to 384,5 (W)
feb-28 15.52.39	Z-Wave	Device: Hallway Plugg Switch Set to ON
feb-28 15.52.28	Error	Exception sending email via (To:krisrodn@gmail.com, cc:, bcc:, Sub:Battery Report, Attach:) = Connection refused
feb-28 15.52.27	Z-Wave	Device: Hallway Plugg Power Set to 1,2 (W)
feb-28 15.52.26	Z-Wave	Device: Hallway Plugg Power Set to 27,7 (W)
feb-28 15.52.26	Z-Wave	Device: Hallway Plugg Switch Set to OFF
feb-28 15.52.17	Error	Exception sending email via (To:krisrodn@gmail.com, cc:, bcc:, Sub:Battery Report, Attach:) = Connection refused
feb-28 15.52.17	BatteryReport	Battery report created.
feb-28 15.52.17	Event	Running script in background: /usr/local/HomeSeer/scripts/BatteryReport.vb
feb-28 15.52.17	Event	Event Trigger "Status Batteriscript"
feb-28 15.52.17	Event	Event Status Batteriscript triggered by the event page 'Run' button.

Figur 6.9 E-post, feilmelding

Vi prøvde også å kun fylle ut de fire første linjene i scriptet «BatteryReport.vb» men da fikk vi også feilmeldinger som vist i Figur 6.9.

Til slutt fant vi ut at vi kunne gå inn på Setup og deretter E-mail. Her kunne vi velge litt diverse e-post innstillinger. Vi har valgt å benytte oss av Gmail som var et eget alternativ her. Det er også mulig å bruke Imap/POP, og man må da følge oppsettet fra e-post leverandøren for Imap/POP. Da vi hadde gjort som vist i Figur 6.10, fungerte både det å sende e-post direkte fra events som vist i Figur 6.8 og å sende e-post via script. Loggen så da ut som vist i Figur 6.11.

Setup 28.02.2019 16.03.00 Sunrise: 07.44.31 Sunset: 18.00.56 User: Kristoffer

HOME VIEW TOOLS PLUG-INS HOMESEER

General Network E-Mail Voice Cameras Custom Labs

E-Mail Settings General

Use Gmail

Gmail Username krisrodn

Gmail Password ****

E-Mail Settings Sending

SMTP Server

SMTP Server Port (default=25) 25

SMTP Username

SMTP Password ****

Mail Domain (enter if your username does not include @domain part)

Default From address

Use SSL

Default "TO" E-Mail address

Default E-Mail Subject

Default E-Mail Message

E-Mail Settings Receiving

Check E-Mail

Check Frequency (minutes) 1

Receiving Server (POP)

Use SSL

Server Port (default=110) 110

POP Username

POP Password ****

Home | Device Status | Device Management | Events | Log | Setup | Global Elements | Control Panel | Manage Plug-ins | About

Figur 6.10 E-post, oppsett

feb-28 16.02.01	email	Email successfully sent to krisrodn@gmail.com
feb-28 16.02.01	Z-Wave	Device: Hallway Plugg Power Set to 1,2 (W)
feb-28 16.02.01	Z-Wave	Device: Hallway Plugg Switch Set to OFF
feb-28 16.02.00	Z-Wave	Device: Hallway Plugg Power Set to 33,8 (W)
feb-28 16.01.59	BatteryReport	Battery report created.
feb-28 16.01.59	Event	Running script in background: /usr/local/HomeSeer/scripts/BatteryReport.vb
feb-28 16.01.59	Event	Event Trigger "Status Batteriscript"
feb-28 16.01.59	Event	Event Status Batteriscript triggered by the event page 'Run' button.

Figur 6.11 E-post, logg

I denne sammenheng kommer spørsmålet om man skal skaffe seg en egen e-post for HomeSeer-en sin. Hvis man velger å lage en egen mail til hjemmesystemet velger vi å tro at systemets Wife Acceptance Factor (WAF) vil da øke. Da vil man få vite hvem man får mail fra og hvorfor man får denne mailen.

6.5.4 Netatmo batteri

Vi fikk prøvd ut en værstasjon fra Netatmo i denne oppgaven, for å se hvilke muligheter det var med disse enhetene. De fungerte omtrent på samme måte som multisensoren når det kommer til temperaturavlesninger. Men en stor forskjell mellom multisensoren og værstasjonen var at værstasjonen ikke leverte batteristatus i prosent, de leverte batteristatusen på en skala fra 1-10000. Dette førte dermed til en liten klinsj med scriptet «BatteryRaport.vb», som skriver ERROR på alle verdier over 100. Måten vi løste det på var å lage et event som benyttet seg av følgende kommando:

&hs.setdevicevaluebyref(refID, (hs.devicevalueex(refID) / 100), True)

Det denne kommandoen gjør er å sette verdien til valgt enhet, basert på en annen enhet sin verdi. Det finurlige vi har gjort i her er å sette vår enhets verdi til sin egen verdi delt på 100.

Event Name: Netatmobatteri

Type: [Dropdown]

Group Reassign: Status

Voice Command: [Text Box]

IF Environmental UltraNetatmo3 Plugin Batteri changes and becomes Any Value

THEN Run a Script or Script Command

Wait for command to finish before continuing. Only allow a single instance to run at a time.

Execute immediate script command: `&hs.setdevicevaluebyref(508, (hs.devicevalueeex(508)) / 100`

Then Wait 10 Seconds

Set the Desired Event Options Here:

Priority Event Include in Powerfailure Recovery Include as Voice Command

Security (Randomize Time Triggers) Remove This Event After Triggering Do Not Log This Event

Cannot Re-Run For: 0 Days 0 Hours 0 Minutes 1 Seconds

Figur 6.12 Netatmo batteri, event

Som vi ser i Figur 6.12 vil vi kjøre kommandoen nevnt over hver gang Netatmo-batteriet endrer verdi. Noe som i utgangspunktet vil skape en loop som vil overlaste systemet, det er derfor vi har lagt til at eventet ikke kan kjøre på nytt før om 1 sekund. På denne måten unngår vi at systemet looper, da endringen skjer momentant, deretter settes eventet på «pause» i 10 sekunder til delingen på 100 har skjedd. I «Advanced» under enheten vil det da se ut som i Figur 6.13. Vi har også måttet endre litt på Status Graphics slik at batteriet blir vist slik vi ønsker å ha det, og slik vi ser i Figur 6.14.

Configuration	Advanced	Status Graphics	UltraNetatmo3
Reference ID	508		
Technology Address	02:00:00:32:bc:30-Battery-Outdoor		
Status	61,44 = Dim		
Value	61,44 = "Low"		
String			

Figur 6.13 Netatmo batteri, advanced

<input type="checkbox"/>	 61%	Environmental	UltraNetatmo3 Plugin	Battery	02:00:00:32:bc:30-Battery-Outdoor	Today 19.19.22	
--------------------------	---	---------------	----------------------	---------	-----------------------------------	----------------	--

Figur 6.14 Netatmo batteri, hjemskjerm

6.5.5 EnTur

For å sette opp denne kollektivtransport oversikten har vi fulgt guiden til Moskus på Hjemmeautomasjon sitt forum [19]. I sammenheng med denne oversikten møtte vi på et par problemer. Den første og største var at vi ikke fikk til å kjøre den ene rutinen «CallsFromStopplace» i scriptet «EnTur.vb». Det er fremdeles uklart for både oss og trådstarter Moskus hvorfor det ikke fungerer for oss, men fungerer for flere andre. Vi har våre mistanker om at det er mono (kompileren til Linux) som er roten til dette problemet.

Det andre umiddelbare problemet vi møtte på var at ved å følge guiden slavisk, ville vi få feilmelding hvis vi brukte NSR-en (den unike ID-en til en holdeplass) til bybanen. Etter litt feilsøking fant vi ut at

problemet var litt inni rutinen «TripsBetweenStops», her står [bus] som parameter og ikke tram som er parameteren for bybane. Det vi da fant ut var at det letteste ville være å legge inn en rutine for hver fremkomstmåte. Altså at vi har en som heter «TripsBetweenStopsBus», en som heter «TripsBetweenStopsTram», en som heter «TripsBetweenStopsTrain» og så videre. Etter at vi endret function til rutinen «TripsBetweenStopsTram» fungerte det helt fint med bybanen.

Det tredje «problemet» vi møtte på var formatteringen, denne var forskjellig fra Bergen til Stavanger, hvor Moskus er fra. Etter litt forskning fant vi ut at dette var på grunn av infoen tilgjengelig på Entur sin journeyplanner var forskjellig fra, Figur 6.15 - Skyss, og Figur 6.16 - Kolumbus [20].

```
{
  "data": {
    "stopPlace": {
      "id": "NSR:StopPlace:29806",
      "name": "Edvard Griegs vei",
      "estimatedCalls": [
        {
          "realtime": false,
          "aimedArrivalTime": "2019-03-08T08:01:00+0100",
          "aimedDepartureTime": "2019-03-08T08:01:00+0100",
          "expectedArrivalTime": "2019-03-08T08:01:00+0100",
          "expectedDepartureTime": "2019-03-08T08:01:00+0100",
          "actualArrivalTime": null,
          "actualDepartureTime": null,
          "date": "2019-03-08",
          "forBoarding": true,
          "forAlighting": true,
          "destinationDisplay": {
            "frontText": "Sentrum"
          },
          "quay": {
            "id": "NSR:Quay:51237"
          },
          "serviceJourney": {
            "journeyPattern": {
              "line": {
                "id": "SKV:Line:51",
                "name": "Birkelandsskiftet - Kokstad - Fjøsanger - Sentrum",
                "transportMode": "bus"
              }
            }
          }
        }
      ]
    }
  }
}
```

Figur 6.15 Skyss

```
{
  "data": {
    "stopPlace": {
      "id": "NSR:StopPlace:27298",
      "name": "Lurahammaren",
      "estimatedCalls": [
        {
          "realtime": false,
          "aimedArrivalTime": "2019-03-08T08:11:00+0100",
          "aimedDepartureTime": "2019-03-08T08:11:00+0100",
          "expectedArrivalTime": "2019-03-08T08:11:00+0100",
          "expectedDepartureTime": "2019-03-08T08:11:00+0100",
          "actualArrivalTime": null,
          "actualDepartureTime": null,
          "date": "2019-03-08",
          "forBoarding": true,
          "forAlighting": true,
          "destinationDisplay": {
            "frontText": "UIS øst - Lurahammaren - Sandnes"
          },
          "quay": {
            "id": "NSR:Quay:47066"
          },
          "serviceJourney": {
            "journeyPattern": {
              "line": {
                "id": "KOL:Line:1007_237",
                "name": "6",
                "transportMode": "bus"
              }
            }
          }
        }
      ]
    }
  }
}
```

Figur 6.16 Kolumbus

<input type="checkbox"/>	521	14.33 → 14.43: Rute 6 14.48 → 14.58: Rute 6 15.03 → 15.13: Rute 6 15.18 → 15.28: Rute 6 15.33 → 15.43: Rute 6	Misc	Transport	Transport3		Today 14.19.15
--------------------------	-----	---	------	-----------	------------	--	-------------------

Figur 6.17 Entur, Kolumbus hjemskjerm

<input type="checkbox"/>	521		14.37 → 15.51: Rute Sentrum - Solheimsviken - Sentrum 15.00 → 15.58: Rute Oasen - Smiberget - Lægdene - Mannsverk	Misc	Transport	Transport3		Today 14.35.31	
--------------------------	-----	---	--	------	-----------	------------	--	-------------------	--

Figur 6.18 Entur, Skyss hjemskjerm

Det ser mye ryddigere ut på Figur 6.17 enn Figur 6.18. Grunnen til det er «name» informasjonen vi finner på Figur 6.15 og Figur 6.16. Vi ser at for Kolumbus er det bare tallet 6 her, mens det for Skyss er hele ruta. Måten vi kan endre det på er at vi for Skyss skulle benyttet oss av «id», med «SKY:line» filtrert bort. Dette er ikke noe vi har brukt tid på, men det er vertfall greit å vite at muligheten er der!

7 Konklusjon

Etter å ha arbeidet mye med systemet, er det tydelig at HomeSeer er en løsning som passer best for spesielt interesserte. Med softwaren har man frihet til å velge egen hardware, men da må man installere softwaren selv og sjekke at hardwaren fungerer som den skal, slik som vi måtte med RaZberry-kortet. Selv om man velger en ferdig hardware- og software-løsning fra HomeSeer er det likevel ikke sikkert at systemet oppfører seg slik som brukeren ønsker. Da kan det hende at brukeren må lese dokumentasjonen fordi alt kan ikke fikses i web-grensesnittet.

Hvor mye arbeid som kreves for å sette opp systemet avhenger av hva brukeren vil oppnå. HomeSeer-softwaren er veldig fleksibel, og det er få begrensninger på hva man kan få ut av systemet. Man kan få et fungerende system ved å koble til noen få smart-enheter og deretter sette opp noen enkle events, uten noen spesielle forkunnskaper. Dersom man vil implementere spesiell funksjonalitet, endre parametere, eller endre innstillinger i det underliggende Linux-systemet krever dette mer kunnskap. Hvis det er ønskelig kan en kyndig bruker skrive egne scripts eller egne plug-ins for å utnytte systemet til det fulle.

Generelt har HomeSeer god støtte for sensorer og hardware fra andre produsenter. Mye av grunnen til dette er det plug-in-baserte systemet, som lar brukere og andre tredjeparter lage plug-ins selv, som brukes til å koble til enhetene som ønskes. Som regel fungerer det som forventet, men av og til oppstår det problemer slik vi viste med Google Home og Philips Hue. På grunn av dette systemet har man heller ingen garantier på at alt skal fungere, og langsiktig støtte er uviss.

Når systemet er satt opp slik som brukeren vil ha det, er det lite vedlikehold som skal til. I perioden vi har brukt systemet har vi ikke opplevd at noe plutselig har sluttet å fungere. Systemet har ikke stoppet eller hengt seg opp, og komponentene har fungert som forventet under testperioden. En del av komponentene våre kjører på batteri, og disse må naturlig nok skiftes ut regelmessig. Batteritiden har vi opplevd som meget god, men når de må skiftes ut demper batteri-scriptet en del av ulempene som kommer med dette.

I HomeSeer er det så mange muligheter for videre implementasjon av systemet av vi ikke kan ta for oss alt. Vi har under valgt å komme med fire forslag til videre implementering av ting vi mener har stort potensiale.

Vi så at det ikke finnes noen lett måte å få på TV-er via HomeSeer, med mindre man har tatt i bruk en Chromecast. Vårt forslag til videre arbeid vil dermed være todelt; første del vil være å fysisk lage en IR fjernkontroll med Z-Wave sender i seg, og andre del kan være å lage plug-in til HomeSeer slik at man enkelt kan ta i bruk denne Z-Wave fjernkontrollen.

Mye av målet med et hjemmesystem er å spare penger på sikt, det vil i aller høyeste grad også gjelde HomeSeer. Med tanke på at vi i alle norske hjem skal ha avansert målesystem (AMS) i sikringsskapene våre, vil vi kunne bruke disse koblet opp mot Tibber og HomeSeer slik at vi kan bruke enheter mindre når strømmen er dyr. Det finnes en Tibber plug-in til HomeSeer, noe som gjør at man kan bruke Tibberpriser til å styre når enheter skal være av og på. Vi har ikke prøvd dette i vårt system, men etter lesning på Hjemmeautomasjons forum er det mange som har prøvd å ta det i bruk i sitt HomeSeer-system.

Med tanke på at formålet med oppgaven er klargjøring til en helse-lab, vil det kanskje være naturlig at man i videre arbeid prøver å ta i bruk enheter som kan benyttes av helsepersonell i form av lesning av vitale verdier og lignende. Her kan det hende at det mangler en del plug-ins, noe som kan gjøre det aktuelt med en ny oppgave som handler om å lage plug-in til enheter som kan være med å hjelpe helsepersonell. Til å begynne med kan man kanskje prøve FitBit, som det finnes en plug-in for. Se hvordan denne fungerer og om informasjonen som kommer fra FitBit-en er tilfredsstillende for helsepersonellet, er den det vil man dermed ha en idé for hvordan man selv kan lage en plug-in for andre enheter.

En stor del av HomeSeer sin forse, nemlig HS Touch, har ikke vi prøvd ut i vår oppgave. Grunnen til at vi ikke har testet det ut er at det lå utenfor budsjettet vårt, samt at oppgaven ville blitt litt for stor for to personer med denne utvidelsen. Vi mener dette er noe som kan og burde bli brukt hvis Høgskulen på Vestlandet skal ta i bruk systemet i labbene sine.

Vi synes at systemet har vært greit å jobbe med, og vi har ikke hatt noen større utfordringer under arbeidsperioden. Vi har også sett at det er mye vi enda ikke har hatt mulighet til å teste ut, på grunn av hvor åpent og fleksibelt systemet er. Fordi mange av ulempene med systemet i stor grad er kunnskapsbaserte, tror vi at det kan være et bra system for Høgskulen på Vestlandet å ta i bruk.

8 Referanser

- [1] Høgskulen på Vestlandet, «Om Høgskulen på Vestlandet,» 22 01 2019. [Internett]. Available: <https://hvl.no/om/>. [Funnet 15 03 2019].
- [2] Høgskulen på Vestlandet, «Fakultet for ingeniør- og naturvitskap (FIN),» 11 02 2019. [Internett]. Available: <https://hvl.no/om/organisering/fin/>. [Funnet 15 03 2019].
- [3] Høgskulen på Vestlandet, «Institutt for elektrofag,» 26 06 2019. [Internett]. Available: <https://hvl.no/om/organisering/fin/institutt-for-elektrofag/>. [Funnet 15 03 2019].
- [4] Høgskulen på Vestlandet, «Campus Bergen,» 14 02 2019. [Internett]. Available: <https://hvl.no/studentliv/studentliv-bergen/campus-bergen/>. [Funnet 15 03 2019].
- [5] Høgskulen på Vestlandet, «Teknologi i helse og omsorg,» [Internett]. Available: <https://hvl.no/studier/studieprogram/2019h/v-tho/>. [Funnet 15 03 2019].
- [6] G. Writer, «Smart Home Protocols – Thread, Zigbee, Z-Wave, KNX and More,» 15 06 2017. [Internett]. Available: <https://www.iotforall.com/smart-home-protocols/>. [Funnet 15 03 2019].
- [7] HomeSeer, «HS3Touch Designer,» [Internett]. Available: <https://homeseer.com/hs3touch-designer/>. [Funnet 15 03 2019].
- [8] HomeSeer, «home-automation-software,» [Internett]. Available: <https://homeseer.com/home-automation-software/>. [Funnet 15 03 2019].
- [9] HomeSeer, «Compare Smart Home Controllers,» [Internett]. Available: <https://homeseer.com/compare-smart-home-controllers/>. [Funnet 15 03 2019].
- [10] HomeSeer, «HomeSeer,» 21 Januar 2019. [Internett]. Available: http://www.homeseer.com/updates3/hspi3_image_091218.zip. [Funnet 21 Januar 2019].
- [11] balena, «balenaEtcher,» 21 Januar 2019. [Internett]. Available: <https://www.balena.io/etcher>. [Funnet 21 Januar 2019].
- [12] Oddvar, «Hjemmeautomasjon,» 25 01 2018. [Internett]. Available: <https://www.hjemmeautomasjon.no/forums/topic/2684-homeseer-p%C3%A5-rpi-med-razberry-2/>. [Funnet 23 01 2019].
- [13] HomeSeer, «Google Home Integration,» [Internett]. Available: <https://homeseer.com/google-home-integration/>. [Funnet 23 01 2019].
- [14] PuTTY, «PuTTY,» 21 Januar 2019. [Internett]. Available: <https://www.putty.org/>. [Funnet 21 Januar 2019].

- [15] cass2115, «HomeSeer,» 25 09 2018. [Internett]. Available: <https://forums.homeseer.com/forum/homeseer-products-services/system-software-controllers/hometroller-sel/95232-how-to-get-root-access-to-hometroller?p=1249576#post1249576>. [Funnet 08 02 2019].
- [16] WinSCP.net, «WinSCP,» 21 Januar 2019. [Internett]. Available: <https://winscp.net/eng/index.php>. [Funnet 21 Januar 2019].
- [17] ZoRaC, «Hjemmeautomasjon,» 11 Juli 2017. [Internett]. Available: <https://www.hjemmeautomasjon.no/forums/topic/1014-oppdatere-fibarofirmware/?do=findComment&comment=19657>. [Funnet 14 Mars 2019].
- [18] sparkman, «forum.homeseer,» 20 Mars 2019. [Internett]. Available: <https://forums.homeseer.com/forum/homeseer-products-services/system-software-controllers/hs3-hs3pro-software/hs3-hs3pro-discussion/81567-how-to-create-a-low-battery-level-event?p=1293724#post1293724>.
- [19] Moskus, «Hjemmeautomasjon,» 14 Januar 2019. [Internett]. Available: <https://www.hjemmeautomasjon.no/forums/topic/4338-script-enturvb-f%C3%A5-inn-data-fra-entur/>. [Funnet 07 Mars 2019].
- [20] Entur, «JourneyPlanner,» Entur, 2019. [Internett]. Available: <https://api.entur.org/doc/shamash-journeyplanner/>. [Funnet 07 Mars 2019].
- [21] Meteorologisk institutt, «Samarbeidspartnere,» [Internett]. Available: <https://hjelp.yr.no/hc/no/articles/206550249-Samarbeidspartnere>. [Funnet 15 03 2019].
- [22] Silicon Laboratories, «Smart home products with Z-Wave inside work together, use just one app to connect and control your smart home from anywhere,» 2019. [Internett]. Available: <https://www.z-wave.com/learn>. [Funnet 15 03 2019].
- [23] P. Lamkin, «Z-Wave explained: What is Z-Wave and why is it important for your smart home?,» 22 01 2019. [Internett]. Available: <https://www.the-ambient.com/guides/zwave-z-wave-smart-home-guide-281>. [Funnet 15 03 2019].
- [24] Z-Wave Alliance, «Z-Wave Plus™ Certification,» [Internett]. Available: https://z-wavealliance.org/z-wave_plus_certification/. [Funnet 26 03 2019].
- [25] ZigBee Alliance, «ZigBee Home Automation Download,» [Internett]. Available: <https://www.zigbee.org/non-menu-pages/zigbee-home-automation-download/>. [Funnet 26 03 2019].
- [26] ZigBee Alliance, «ZigBee Light Link Download,» [Internett]. Available: <https://www.zigbee.org/non-menu-pages/zigbee-light-link-download/>. [Funnet 26 03 2019].

[27] ZigBee Alliance, «ZigBee 3.0,» [Internett]. Available: <https://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>. [Funnet 26 03 2019].

9 Appendiks

Appendiks A **Forkortelser og ordforklaringer**

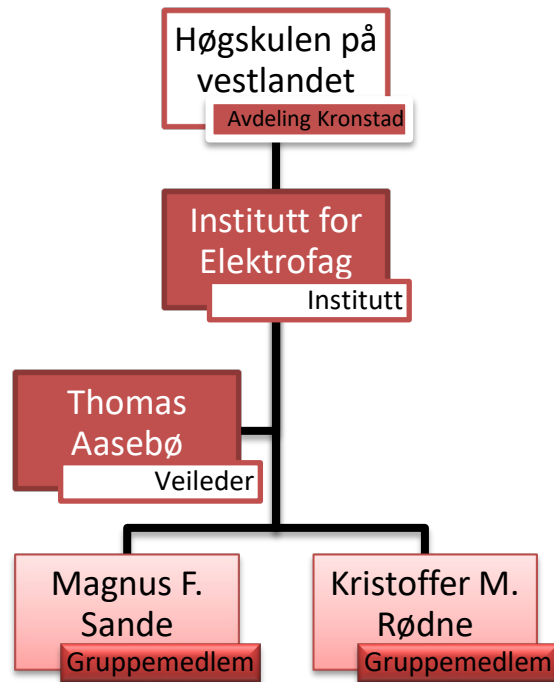
AMS	Avansert målesystem
BLE	Bluetooth Low Energy
IEEE	Institute of Electrical and Electronics Engineers
OTA	Over-the-air
SFTP	SSH File Transfer Protocol
SSH	Secure shell
TTS	Text-to-speech
WAF	Wife Acceptance Factor
ZBA	ZigBee Building Automation
ZHA	ZigBee Home Automation
ZLL	ZigBee Light Link

Appendiks B Prosjektledelse og styring

Siden vi skal arbeidet med et hjemmesystem, tenkte vi det ville være greiest å være hjemme hos en av oss. Det er også lettere å ha alt av komponenter stående på en plass, slik vi slepper og drasse dem til/fra skolen hver gang vi møtes og vi sparer tid i form av at vi slepper å sette alt opp hver gang.

Når det gjelder innkjøp generelt til denne oppgaven, er det Thomas Aasebø som vil stå for dette. Vi gir beskjed til Thomas hva vi trenger/vil ha og han fikser det.

B.1 Prosjektorganisasjon



Figur 9.1 Prosjektorganisering

Ansvarsfordeling og arbeidsfordeling vil være relativt flytende da vi bare er to på gruppen. Men vi har startet med at Magnus tar seg av det praktiske, mens Kristoffer har tatt seg av skriving av oppgaven. Dette kan endre seg utover i oppgavejobbingen, men vi vil selvsagt begge to holde på med både det praktiske og selve oppgaveskrivingen.

B.2 Fremdriftsplan

Planlagt arbeid for hovedaktivitet					Uke	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Planlagt arbeid for aktivitet					Man	7/1	14/1	21/1	28/1	4/2	11/2	18/2	25/2	4/3	11/3	18/3	25/3	1/4	8/4	15/4	22/4	29/4	6/5	13/5	20/5	27/5	3/6	10/6
Utført arbeid					Sen	13/1	20/1	27/1	3/2	10/2	17/2	24/2	3/3	10/3	17/3	24/3	31/3	7/4	14/4	21/4	28/4	5/5	12/5	19/5	26/5	2/6	9/6	16/6
#	Aktivitet	Planlagt arbeid	Utført arbeid	Start Dato	Slutt Dato	Framdrift	Ansvarlig	SU/VF																Páske				
1	Forstudie			9/1	31/1			~100																				
1.1	Oppstartsmøte med veileder	2	2	9/1	9/1	100%	MFS/KMR	2t																				
1.2	Forskning	50	50	9/1	21/1	100%	MFS/KMR	~50 timer																				
1.3	Rapportskriving	20	20	21/1	31/1	100%	KMR	~20 timer																				
1.4	Hovedoppsett av system	8	8	21/1	31/1	100%	MFS	~8 timer																				
1.5	Oppsett av komponenter	20	20	23/1	-	100%	MFS/KMR	~20 timer																				
1.6	Kjøp av software			28/1	28/1	100%	MFS/KMR	♦																				
1.7	Forstudie innlevering			31/1	31/1	0%	MFS/KMR	♦																				
2	Arbeid frem mot midtveis			1/2	1/4	0%	MFS/KMR	~400																				
2.1	Oppsett av komponenter	20		-	1/4	0%	MFS/KMR	~20																				
2.2	Oppsett av events	240		1/2	-	0%	MFS/KMR	~240																				
2.3	Rapportskriving	120		1/2	-	0%	MFS/KMR	~120																				
2.4	Arbeid med midtveispresentasjon	20		28/3	31/3	0%	MFS/KMR																					
2.5	Midtveis presentasjon			1/4	5/4	0%	MFS/KMR	♦																				
3	Arbeid frem mot EXPO			2/4	13/6	0%	MFS/KMR	~450																				
3.1	Rapportskriving	160		-	30/5	0%	MFS/KMR	~160																				
3.2	Oppsett av events	220		-	12/6	0%	MFS/KMR	~220																				
3.3	Arbeid med presentasjon seminar	20		7/5	9/5	0%	MFS/KMR																					
3.4	Bachelor oppgave seminar			10/5	10/5	0%	MFS/KMR																					
3.5	Bachelor oppgave innlevering			31/5	31/5	0%	MFS/KMR	♦																				
3.6	Arbeid med oppgave presentasjon	20		1/6	5/6	0%	MFS/KMR																					
3.7	Bachelor oppgave presentasjon			5/6	12/6	0%	MFS/KMR																					
3.8	Arbeid med EXPO	30		7/5	13/6	0%	MFS/KMR	~10																				
3.9	EXPO			13/6	13/6	0%	MFS/KMR	~20																				
4	Timer	950	100																									

Figur 9.2 Opprinnelig fremdriftsplan

Planlagt arbeid for hovedaktivitet					Uke	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Planlagt arbeid for aktivitet					Man	7/1	14/1	21/1	28/1	4/2	11/2	18/2	25/2	4/3	11/3	18/3	25/3	1/4	8/4	15/4	22/4	29/4	6/5	13/5	20/5	27/5	3/6	10/6
Utført arbeid					Sen	13/1	20/1	27/1	3/2	10/2	17/2	24/2	3/3	10/3	17/3	24/3	31/3	7/4	14/4	21/4	28/4	5/5	12/5	19/5	26/5	2/6	9/6	16/6
#	Aktivitet	Planlagt arbeid	Utført arbeid	Start Dato	Slutt Dato	Framdrift	Ansvarlig	SU/VF																Páske				
1	Forstudie			9/1	31/1			~100																				
1.1	Oppstartsmøte med veileder	2	2	9/1	9/1	100%	MFS/KMR	2t																				
1.2	Forskning	50	50	9/1	21/1	100%	MFS/KMR	~50 timer																				
1.3	Rapportskriving	20	20	21/1	31/1	100%	KMR	~20 timer																				
1.4	Hovedoppsett av system	8	8	21/1	31/1	100%	MFS	~8 timer																				
1.5	Oppsett av komponenter	20	20	23/1	-	100%	MFS/KMR	~20 timer																				
1.6	Kjøp av software			28/1	28/1	100%	MFS/KMR	♦																				
1.7	Forstudie innlevering			31/1	31/1	100%	MFS/KMR	♦																				
2	Arbeid frem mot midtveis			1/2	1/4	100%	MFS/KMR	~310																				
2.1	Oppsett av komponenter	80	71,5	-	1/4	100%	MFS/KMR	~71,5																				
2.2	Oppsett av events	130	128	1/2	-	100%	MFS/KMR	~128																				
2.3	Rapportskriving	80	76,5	1/2	-	100%	MFS/KMR	~76,5																				
2.4	Arbeid med midtveispresentasjon	20	21,5	28/3	31/3	100%	MFS/KMR	~24,5																				
2.5	Midtveis presentasjon			3	1/4	5/4	100%	MFS/KMR																				
3	Arbeid frem mot EXPO			2/4	13/6	60%	MFS/KMR	~520																				
3.1	Rapportskriving	350	305	-	30/5	100%	MFS/KMR	~305																				
3.2	Oppsett av events	100	96	-	12/6	100%	MFS/KMR	~96																				
3.3	Arbeid med presentasjon seminar	20	8	7/5	9/5	100%	MFS/KMR																					
3.4	Bachelor oppgave seminar			10/5	10/5	100%	MFS/KMR	♦																				
3.5	Bachelor oppgave innlevering			31/5	31/5	100%	MFS/KMR	♦																				
3.6	Arbeid med oppgave presentasjon	20	8	1/6	5/6	0%	MFS/KMR																					
3.7	Bachelor oppgave presentasjon			5/6	12/6	0%	MFS/KMR																					
3.8	Arbeid med EXPO	30	20	7/5	13/6	0%	MFS/KMR	~10																				
3.9	EXPO			13/6	13/6	0%	MFS/KMR	~20																				
4	Timer	930	841,5																									

Figur 9.3 Endelig fremdriftsplan

B.3 Risikoliste

Gjerne i tabellform (kort, konkret og oversiktlig).

Risiko	Sannsynlighet	Alvor	Konsekvens	Tiltak
Ikke ferdig innen tidsfrist	Lav	Høy	Ikke bestått bacheloroppgave	Følge fremdriftsplan, produktiv jobbing
Interne konflikter	Lav	Middels	Forsinkelse av fremdriftsplan, dårlig arbeidsmiljø	God kommunikasjon
Ødelagt minnekort	Lav	Middels/Høy	Arbeidet vi har gjort kan forsvinne	Ta backup jevnlig
Ødelagt hardware	Lav	Lav/Middels	Må kjøpe nytt	Finnes ikke tiltak mot hardware feil og mangler
Sykdom	Lav	Middels	Forsinkelse av fremdriftsplan	Kortere oppgave, eventuelt ta igjen det tapte når man er frisk igjen
Feilvurdering av arbeidsmengde	Middels	Middels/Høy	Oppgaven ikke blir 100% fullført	Jobbe mer enn det fremdriftsplan sier.
Lisensproblem	Lav	Høy	Ikke fungerende system	Ringte kundestøtte
Ikke fungerende software	Middels	Middels	Systemet kan slutte å fungere som planlagt, kompatibiliteten med enkelte komponenter svekkes	Prøve annen lignende software, lage egne plug-ins

Tabell 8.1 Risikoanalyse

Appendiks C Brukerdokumentasjon

Keyfob

Knapp	Funksjon
Square – 1-click	Brightness – 5%
Square – 2-click	Brightness – 10%
Square – 3-click	Brightness – 100%
Square – Hold	Off
Circle – 1-click	Colour – Yellow
Circle – 2-click	Colour – White
Circle – 3-click	Colour – Red
Circle – Hold	
Cross – 1-click	Volume down
Cross – 2-click	Volume up
Cross – 3-click	Play NRKmp3
Cross – Hold	Stop playing
Triangle – 1-click	Plug on
Triangle – 2-click	Plug off
Triangle – 3-click	Temperatur multisensor
Triangle – Hold	
Minus – 1-click	Morning
Minus – 2-click	Day
Minus – 3-click	Evening
Minus – Hold	Night
Plus – 1-click	Go out
Plus – 2-click	Forecast today
Plus – 3-click	Forecast tomorrow
Plus – Hold	Coming home

Tabell 8.2 Keyfob

Stemmestyring

Komponent	Lokasjon	Funksjonalitet
Lyspære 1	«Living room»	Skifte farge
Lyspære 2	«Hallway»	Skifte lysstyrke
Stikkontakt	«Hallway»	Slå av/på

Tabell 8.3 Stemmestyring

Kommandoeksempler

For å kunne stemmestyre må man først si «Ok Google» eller «Hey Google», etterfulgt av hva man vil gjøre.

Lyspære 1

- «Bytt farge i living room til grønn»
- «Bytt farge på living room til blå»
- «Turn the living room orange»

Lyspære 2

- «Slå av hallway light»

- «Slå på hallway light»
- «Demp hallway»
- «Gjør hallway lysere»
- «Gjør hallway lysere med 20%»
- «Turn off hallway light»
- «Dim the hallway light»
- «Set the hallway brightness to 50%»

Stikkontakt

- «Turn off veggplugg»
- «Turn on veggplugg»
- «Skru av veggplugg»
- «Skru på veggplugg»

Multisensor

Betingelser	Funksjonalitet
Det er natt (kl 00.00-06.00)	Lys blir gult ved bevegelse, skrur av etter 5 sekunder.
Det er morgen (06.00-09.00)	Lys blir rosa når registrert lux-verdi er mindre enn 25. Skru seg av når den overstiger 50.

Tabell 8.4 Multisensor

Dør- og vindussensor

Betingelser	Funksjonalitet
Det er natt, <u>og</u> sensoren åpnes	Sonos spiller av lyd, lys blinker rødt, TTS sier «Alarm».
Det er morgen, <u>og</u> sensoren åpnes	Lys slås på, TTS sier vinduet er åpent, stikkontakt slås av.

Tabell 8.5 Dør- og vindussensor

Stikkontakt

Betingelser	Funksjonalitet
Dersom det ikke er natt, <u>og</u> man er hjemme	Stikkontakt slås på
Dersom det er natt, <u>eller</u> man ikke er hjemme	Stikkontakt slås av

Tabell 8.6 Stikkontakt

Appendiks D **Bill Of Materials**

Fibaro Wall Plug

Hardware: FGWPF-102

Firmware: 2.5

Kommunikasjon: Z-Wave

Fibaro Wall Plug er en smartbryter med flere smarte funksjoner. Man kan som navnet sier skru enheten av og på, det er også mulighet for strømovervåkning noe som igjen gir muligheten til både nåværende og historisk strømmåling. Med disse funksjonene kan vi enkelt styre når vi vil at enheten skal skrus av/på, enten om det går på tid, strømmålingen eller ut i fra input fra andre sensorer/komponenter i systemet vårt. Et pluss til denne sensoren er at den er barnesikret, noe som gjør at det er vanskelig å få i seg strøm.

Hardware: FGWPF-102 ZW5

Firmware: 3.2

Kommunikasjon: Z-Wave Plus

Vi har to veggplugger. Den ene er av typen FGWPF-102 ZW5, mens den andre er av typen FGWPF-102. Hovedforskjellen er firmwaren som kjøres og hvilken kommunikasjonsprotokoll som brukes.

Fibaro Door/Window Sensor

Hardware: FGK-10x

Firmware: 3.2

Kommunikasjon: Z-Wave Plus

Fibaro Door/Window Sensor består av to komponenter. En batteridrevet hovedsensor, og en tilhørende magnet. Den fungerer ved at den viser status som åpen når det ikke er kontakt med magneten, og lukket når den har det. Det er også mulighet for å koble en binær sensor til hovedsensoren, noe som gjør at man kan ha temperaturmåling også ved hjelp av denne sensoren. En av flere bruksområder til denne sensoren kan for eksempel være å skru av en panelovn når vi åpner dør/vindu og de er åpne lenger en 30 sekunder.

Fibaro Motion Sensor

Hardware: FGMS-001

Firmware: 3.2

Kommunikasjon: Z-Wave Plus

Fibaro Motion Sensor er en batteridrevet sensor som består av flere smarte funksjoner. Vi kaller den multisensor i systemet vårt, av den grunn at den har 3 hovedbruksområder. Den detekterer bevegelse som ligger i navnet til sensoren. Sensoren kan også måle temperatur og lysintensiteten i rommet den befinner seg. Disse 3 funksjonene gjør at dette er en enhet som egner seg veldig godt til lysstyring. Det er også en siste funksjon som går på seismisk aktivitet, men den er ikke veldig aktuell her i Norge.

Fibaro Keyfob

Hardware: FGKF-601

Firmware: 3.2

Kommunikasjon: Z-Wave Plus

Fibaro Keyfob er en liten, programmerbar fjernkontroll. Den er IP54-sertifisert, og består av 6 knapper som reagerer på 1, 2, og 3 trykk, samt at man holder knappen inne. Det gir totalt 24 ulike muligheter, i tillegg til at man kan programmere tastesekvenser. Det er mulig å lage til sammen 6 sekvenser av tastetrykk, altså at man for eksempel trykker firkant, rounding, kryss for å sette systemet i feriemodus.

Google Home Mini

Hardware: HOA

Firmware: Over-the-air (OTA), automatisk oppdatering

Kommunikasjon: 802.11b/g/n/ac - (2,4/5 GHz) Wi-Fi, Bluetooth® 4.1

Google Home Mini er en smarthøytaler med stemmeassistent. Det vil si at man kan stille denne assistenten spørsmål, og den vil svare etter beste evne. I denne oppgaven skal vi fokusere på å stemmestyre vårt smartsystem ved hjelp av Google Home Mini. Vi kommer også til å bruke den som høytaler i enkelte tilfeller. Integrert i HomeSeer, så plug-in er ikke nødvendig.

Philips Hue Starter kit E27

Philips Hue Bridge

Hardware: BSB002

Firmware: 1901181309

Kommunikasjon: Wi-Fi og ZLL

Dette er hjernen til Philips Hue-lysstyringen. Her snakker broen med lyspærer og brytere via ZigBee LL, mens den snakker med vårt hjemmesystem via ethernet.

Philips Hue White and color ambiance lightbulb 10W A60 E27

Hardware: LCT015

Firmware: 1.46.13_r26312

Kommunikasjon: ZLL

Philips Hue White and color ambiance lightbulb er lyspærer med E27-sokkel hvor man kan styre farge og lysstyrke på lysene.

Philips Hue Dimmer switch

Hardware: RWL021

Firmware: 5.45.1.17846

Kommunikasjon: ZLL

Philips Hue Dimmer switch er en bryter som kan skru av og på lyset, samt endre lysstyrken. Bryteren har 5 programmerbare funksjoner som alle bruker on-knappen. I motsetning til andre brytere er denne sekvensielt programmerbar, det vil si at den har en kjent sekvens på 5 funksjoner som den bytter på i kronologisk rekkefølge.

Sonos One

Hardware: 1.21.1.8-2

Firmware: 10.0

Kommunikasjon: 802.11b/g (2.4 GHz) Wi-Fi, Ethernet, Bluetooth Low Energy (BLE).

Sonos One er en kompakt smarthøytaler. Den kommuniserer over Wi-fi, men vi trenger en plug-in for å bruke den sammen med HomeSeer. Sonos One støtter for øyeblikket kun Alexa stemmestyring, men vil med tiden også støtte Google sin stemmeassistent.

Netatmo Urban Weather Station

Innendørsmodul

Hardware: NWS03IN

Firmware: 137

Kommunikasjon: 802.11 b/g/n (2.4GHz) Wi-Fi, RF: 868.9-869.9 Mhz

Innendørsmodulen måler temperatur, luftfuktighet, lufttrykk, CO₂ nivå og lydnivå. Alle disse målingene kan vi bruke videre til triggers eller conditions for ting vi vil skal skje i smartsystemet vårt. Modulen snakker med andre moduler, i vårt tilfelle utendørsmodulen, over radiofrekvens 868.9-869.9MHz. Innendørsmodulen registrerer dataene fra utendørsmodulen og laster begge datasettene opp til skyen over Wi-Fi. Det er denne skyen vi henter informasjon fra via en plug-in i vårt smartsystem.

Utendørsmodul

Hardware: NWS03OUT

Firmware: 46

Kommunikasjon: RF: 868.9-869.9 Mhz

Utendørsmodulen måler temperatur og luftfuktighet. Den er også med i et nettverk med andre utendørsmoduler, slik at man får opp maks temperatur, min temperatur og værprognoser for området ditt. Dette blir faktisk brukt av yr for å lage mer korrekte prognoser på områder hvor de ikke har sensorer selv. [21]

Appendiks E **Protokoller**

Z-Wave

Z-Wave er en trådløs kommunikasjonsprotokoll som baserer seg på meshnettverkstopologi. Det vil si at hver ikke-batteridrevne enhet i nettverket vil fungere som en signalrepeater [22]. Det betyr dermed at jo flere enheter man har i nettverket sitt, desto sterkere blir det. Protokollen ble unnfanget i Danmark i 1999 av firmaet Zensys [23] og ble laget for overvåkning, kontroll og statuslesning av enheter. Den utvikles nå av Silicon Labs sammen med Z-Wave Alliance, som er et konsortium som består av over 700 bedrifter fra hele verden.

Z-Wave har ulik frekvens alt etter hvor i verden man er. Vi i Norge følger standarden som er satt for Europa, som er på 868,42 Mhz. Det at den har ulik frekvens basert på hvor i verden man er, gjør at man må være obs når man kjøper Z-Wave enheter slik at man får enheter som er kompatible med systemet og frekvensen man kjører. Det at den ligger i denne frekvensen, gjør også at den er lite utsatt for interferens av andre signaler i hjemmet. Protokollen fungerer toveis noe som gjør at enheter og sentral kan snakke frem og tilbake med hverandre. Mellom Z-Wave enheter kan det maksimalt være 30m for at de skal klare å snakke sammen.

Z-Wave Plus

Z-Wave Plus er et sertifiseringsprogram laget for å gjøre det lettere for forbrukere å finne eller identifisere produkter som støtter neste-generasjons Z-Wave hardwareplattform. [24] Dette er den femte generasjonen av Z-Wave-plattformen, som har en rekke forbedringer sammenlignet med forrige generasjon, blant annet når det kommer til batterilevetid, rekkevidde, båndbredde m.m. Z-Wave Plus-symbolet gjør de sertifiserte produktene lett gjenkjennelige. Med Z-Wave Plus har man også oppdatering av software OTA, noe man ikke har med vanlig Z-Wave protokoll.

ZigBee

ZigBee-alliansen er også et konsortium som består av over 300 selskaper. I motsetning til Z-Wave er ZigBee ikke en egen standard, men et varemerke. Alliansen har ulike ZigBee-profiler slik at produktprodusenter kan lage produkter som fungerer sammen, basert på disse profilene. De ulike profilene snakker ikke sammen, bortsett fra ZigBee 3.0 som er kompatibel med samtlige.

ZigBee opererer med en frekvens på 2,4 GHz, noe som gjør at denne protokollen kan være veldig utsatt for interferens med andre signaler i hjemmet, spesielt da av WiFi som også operere i 2,4Ghz båndet. Fordelen med ZigBee kontra Z-Wave er at det kan være opp mot 100m mellom hver enhet for at man skal ha kontakt med hverandre. En ulempe er som sagt at den er mer utsatt for interferens, og det vil dermed kanskje ikke være optimalt å kjøre sikkerhetskomponenter med ZigBee, men heller med Z-Wave. Vi har i vårt system kun lys med ZigBee, da det ikke er kritisk om disse komponentene opplever interferens.

ZHA

Rettet mot hjemmeautomasjon. Profilen er lagt opp slik at nettverket vanligvis «sover» til brukeren velger å sende en kommando via f. eks. en fjernkontroll. [25]

ZLL

Ment som en enkel profil spesielt utviklet for lys- og lampesystemer. Den inkluderer funksjoner som ikke er i HA-profilen. [26]

ZigBee 3.0

Samler-profilen som kan koble sammen til alle typer ZigBee-nettverk. Den er både for- og bakoverkompatibel. «Legacy»-enheter (dvs. alt som ikke er 3.0), vil kunne bli med i et 3.0-nettverk, og 3.0-enheter kan legges til i «Legacy»-nettverk. ZigBee 3.0 er ment som en erstatning for alle de andre profilene nevnt ovenfor. [27]