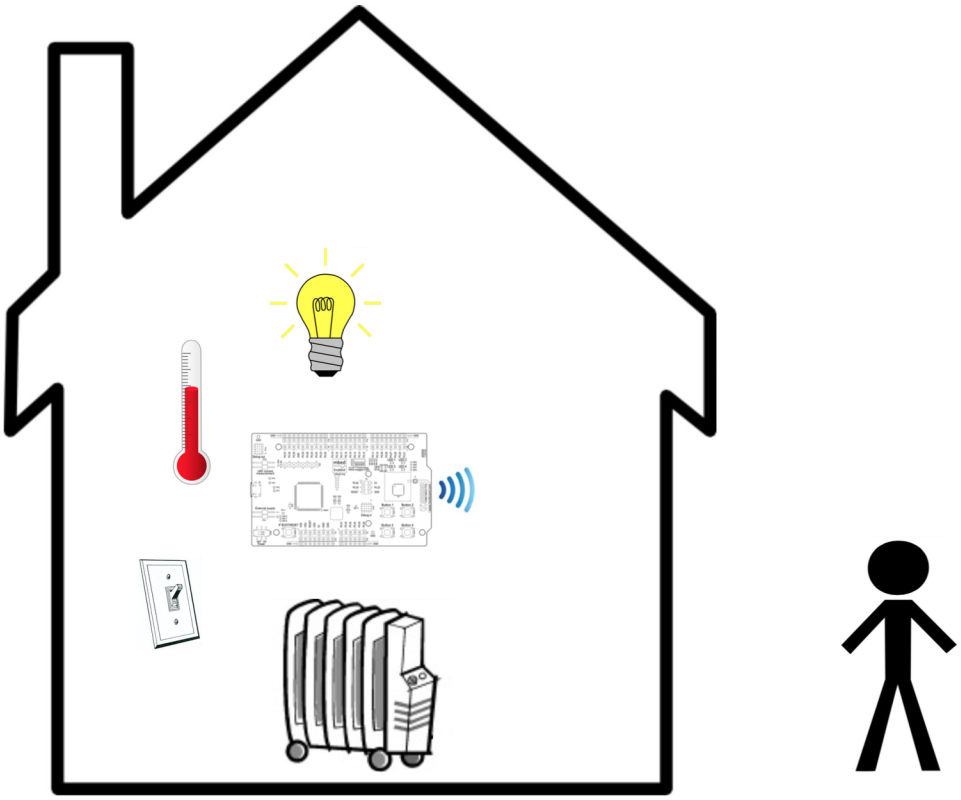
Toolkit for IoT Smart Home



P7 Afgangsprojekt
Institut for Elektroniske Systemer
Aalborg Universitet
7 Januar 2016

AALBORG UNIVERSITET

Institut for Elektroniske Systemer
Fredrik Bajers Vej 7
9000 Aalborg
Telefon 99 40 86 00
Fax 99 40 98 40
http://www.es.aau.dk

Titel:

Toolkit for IoT Smart Home

Projektperiode:

P7, Efterårssemesteret 2015
Afgangsprojekt

Projektgruppe:

714

Deltagere:

Lasse Kristensen

Vejleder:

Per Printz Madsen

Oplagstal: 3
Sidetal: 58 + Appendiks
Bilagsantal: 21 stk. på CD-ROM
Afsluttet den 07-01-2016

Synopsis:

Rapporten omhandler design og udvikling af et Toolkit for IoT Smart Home. Der er tale om et universalt system med mange anvendelses muligheder. Dette projekt fokuserer dog på to brugsscenarier som systemet skal kunne opfylde.
Opbygningen af systemet tager udgangspunkt i to udviklingskits fra Nordic Semiconductor, hvor formålet er at oprette en trådløs kommunikation imellem to enheder vha. Bluetooth Low Energy. Systemet er opbygget efter master og device princippet, hvor alle styrings funktionaliteterne er samlet på master enheden og alle eksterne komponenter er tilsluttet device enheden.

De to brugsscenarier er styring af en kontakt og en varmekontrol hvor en temperatur skal overvåges og reguleres efter behov. Udviklingen er sket med henblik på mulighed for fortsat videreudvikling, således at det hele tiden er kompatibelt med markedets efterspørgsel.

Forord

Denne rapport er udarbejdet af Lasse Kristensen og beskriver 7. semesters afgangsprojekt under uddannelsen til Diplomingeniør i Elektronik. Inspirationen til opgave er fundet i et samarbejde med virksomheden Homatic Engineering A/S. Projektet er udarbejdet i efterårssemesteret fra den. 1 September 2015 til den 7 Januar 2016.

Læsevejledning

Denne rapport henvender sig til vejledere, studerende og andre interesserede indenfor området Elektronik og Software udvikling og forudsætter, at læseren har en grundlæggende viden indenfor området.

Kildehenvisning vil gennem rapporten være nummerede henvisninger til en samlet litteraturliste bagerst i rapporten. Der vil ligeledes være henvisninger til appendiks og bilagsoversigt gennem rapporten. Appendiks findes bagerst i rapporten, imens bilag findes på en medfølgende cd, som er indsat på bagerste side i rapporten. Figurer, formler, tabeller og kode udsnit er nummeret i forhold til kapitel nummer.

Lasse Kristensen

Indholdsfortegnelse

1 Indledning

I forbindelse med et tidligere praktik forløb hos Homatic Engineering blev det aftalt, at jeg som afgangsprojekt skulle udvikle et universalt IoT toolkit. Ideen omkring projektet var, at det skulle være muligt at tilpasse systemet til forskellige brugsscenarier. For virksomheden, som primært arbejder med kundespecifikke projekter, ville dette være et godt produkt, som kunne tilpasses den enkelte kundes behov.

Smart Home er et forholdsvis nyt begreb indenfor husholdnings elektronikken. Det er et område, som der forventes meget af i de kommende år, og producenterne arbejder på højtryk for at være først med nye produkter. Smart Home produkter har dog i mange år været en del af de danske hjem. Det kan være, at du har en automatisk portåbner, som betjenes med en fjernbetjening, en timer på din kaffemaskine eller måske et alarmsystem.

Udover alle de smarte funktioner et Smart Home system kan bidrage med i hverdagen, kan der også være en økonomisk gevinst at hente. Ved at installere smarte kontakter i hjemmet kan du let afbryde strømmen til alle elektronik produkter om natten eller når du ikke er hjemme. Ser vi lidt ud i fremtiden, forventes det at priserne for el vil variere, alt efter tidspunkt på døgnet og belastning på el nettet. På denne måde kan man vha. sit Smart Home eksempelvis indstille sin opvaskemaskine til at starte klokken 3 om natten, hvor strømmen er billig frem for klokken 19, hvor der er en større belastning på el nettet og priserne derved vil være højere.

De flest Smart Home produkter der findes på markedet idag, er såkaldte stand alone produkter. Dette vil sige, at man som forbruger skal oprette forbindelse til hver enkel enhed for at kontrollere denne. Der vil indenfor dette område være en udvikling de kommende år, hvor kontrollen over alle husstandens Smart Home produkter kan foregå fra samme system. Dette vil give en bedre udnyttelse af produkterne som én samlet løsning, i stedet for tyve individuelle.

Ideen med udviklingen af et IoT Smart Home toolkit som dette er, at det skal være let at opsætte nye systemer. Det skal være let at ændre på sine opsætninger og eventuel kunne tilpasse systemet til nye komponenter eller nye brugsscenarier. På denne måde kan man let genbruge et system, i stedet for at udskifte det til et nyt.

Problemanalyse 2

Der vil i dette kapitel være en indledende analyse, af de teknologier som danner baggrunden for designet af toolkittet. Der vil også være beskrivelser omkring valget af udviklingskit, samt hvilke funktionaliteter det skal understøtte.

2.1 Introduktion til IoT Smart Home

I dette afsnit undersøges der mere uddybende hvad et IoT Smart Home system indeholder. Hvordan IoT (Internet of Thing) og Smart Home hænger sammen, samt hvordan de forskellige teknologier kan være med til effektiviserer det samlede elforbrug i Danmark.

2.1.1 Smart Home

Smart Home er kort fortalt en kontrol og informations teknologi for dit hjem. Det bruges ofte som overvågning eller kontrol af dine elektriske apparater. Eksempler på overvågning kan være dine røgalarmer, hvor du kan få en besked på din smartphone hvis der skulle opstå brand og derved hurtigt kan tilkalde brandvæsenet. Det kan også være at du har indstillet systemet til, at din hoveddør skal være låst i tidsrummet mellem klokken 22 om aftenen og klokken 8 om morgenen.

Man kan sige, at et Smart Home kan være med til, at automatiser styringen af dit hjem. Oftest i private hjem er funktioner som automatisk låsning af hoveddøren og tænd/sluk for lyset ekstra funktioner som ikke er en nødvendighed for at hverdagen kan fungere. Disse funktioner kan for folk med handicap eller funktionsnedsættelse være med til at forbedre deres livskvalitet.

Udviklingen indenfor hjemme elektronik produkter er gået hurtigt de seneste år. Det er nærmest et krav, at du fra din mobiltelefon eller tablet kan kontrollere dit lys, alarm system, TV og endda se om der er post i postkassen. Dette er blot nogle eksempler, på den udvikling der har været indenfor Smart Home produkter de seneste år. Der ses også ud fra tendensen, at flere og flere af husstandenes elektronik produkter skal kunne kommunikere trådløst med hinanden.

Der er idag mange producenter af Smart Home produker og af de store indenfor området kan der nævnes Samsung, Google og Apple. Især for Google har dette være et nyt område, hvor opkøb af virksomheder har været en del af deres strategi. Der er dog

ikke meget samarbejde imellem de store producenter, hvor en fælles standard for Smart Home produkter med fordel kunne opstilles. I stedet udvikler hver virksomhed deres egen teknologi, som kræver at alle enheder som tilsluttes skal være godkendt til netop denne platform.

Hvis der kigges på Homekit løsningen fra Apple, kræves der udover dine Homekit kompatible enheder også et AppleTV [1]. Denne fungere som husets masterenhed, og har til opgave at kommunikere med alle husets homekit enheder, inklusiv brugernes mobile enheder. Masterenheden er udover at være tilsluttet husets netværk også forbundet med internettet som åbner op for en masse nye funktionaliteter, men også stiller store krav til sikkerheden. Fordelen ved at kontrollen af alle enheder samles i masterenheden, er at brugerne kun behøver at tilkoble sig en enhed for at kontrollere dem alle.

Kigges der på funktionaliteterne for en masterenhed, er en vigtig del, at den indeholder styringsfunktionerne for kontrollen af det samlede toolkit. Dette kan med fordel udføres ud fra følgende to kriterier:

- Timed event
- Triggered event

Timed event vil sige at masterenheden er opsat til at kontrollere enheden indenfor et bestemt tidsinterval. Det kan være låsning af hoveddøren klokken 22 hver aften.
Triggered event er til gengæld afhængig af et input fra brugerne eller andre systemet. Dette kan være et tryk på en knap som efterfølgende skal slukke for en lampe.

En anden fordel ved at alle Smart Home produkter er sammenkoblet vha. masterenheden, er at man kan inddele hjemmets elektronikprodukter i grupper. På denne måde kan man kontroller flere enheder som om de var en enhed og opdele dem i grupper Et eksempel på dette kan være en gruppe kaldet *stue*, hvor man har adgang til alle lamper i stuen og kan tænde dem alle på samme kontakt. man kan kalde en gruppe for *spare gruppen* og derved let afbryde strømmen til alle elektronik som står i standby i de perioder hvor de ikke benyttes. Dette er funktioner som udover at være smarte for forbrugerne, også kan være med til at mindske det samlede energiforbrug for husstanden.

De enkelte enheder i et Smart Home kommunikere oftest trådløst med brugerne eller masterenhederne. De mest benyttede teknologier er WiFi og Bluetooth kommunikation. Der er dog både fordele og ulemper ved begge disse teknologier. Wifi forbindelsen giver mulighed for, at overførsel store datamængder som billeder og video imellem enhederne. Dog har WiFi forbindelsen et stort strømforbrug og stiller flere krav til sikkerheden, da de ofte er tilkoblet husstandens netværk.

Ser man derimod på Bluetooth og især udgaven kaldet Bluetooth Low Energy (BLE) er den meget populær blandt Smart Home produkter. Dette skyldes primært at den har et meget lavt strømforbrug og giver mulighed for at udvikle systemer med mange års batteritid. Overførsels hastighederne for BLE er dog væsentligt lavere end for WiFi forbindelsen. Dette er dog ikke noget problem, da der ofte sendes mindre data mængder imellem Smart Home produkter. Rækkevidden for BLE forbindelsen er i åbne arealer testet til over 100 meter, men forventes dog væsentligt reducerede i almindelige byggerier. [2]

2.1.2 IoT - Internet of Thing

IoT eller Internet of Thing har været under udvikling i mange år, men er først endeligt blevet navngivet i 1999. Det siges at den første IoT applikation blev udviklet i 1980'erne, hvor en Pepsi automat blev tilkoblet internettet. Programmørerne kunne se antallet og temperaturene for sodavandene i automaten, hvorefter de kunne vurdere, hvorvidt de ville tage turen ned til automaten efter en sodavand eller lade være [3].

IoT minder meget om det kendte M2M (Machine to Machine) kommunikation, som igennem mange år har været betegnelsen for denne teknologi. M2M har primært været benyttet til systemer, hvor der enten skal aflæses data fra en sensor eller sendes advarsler til en bruger. Kommunikationen imellem M2M enhederne kan foregå over mobilnettet, da data mængderne for disse systemet ofte er meget begrænset.

Udviklingen af små elektroniske enheder som automatisk kan transmittere data, er kraftigt voksende. Dette skyldes bl.a. at overvågning af sensorer Dette skyldes bl.a. en stigende efterspørgsel idet at der med M2M og IoT enheder ikke er grænser for hvad der kan overvåges. [4] [5] Eksempler på brugen af M2M og IoT enheder er:

- Aflæsning af el, vand og varme forbrug.
- Genbestilling af varer, når en ubemandet automat er ved at løbe tør.
- Tyveri alarmer
- Rottefælder
- Overvågning af kølecontainer under transport

Dette er blot et udpluk af de allerede eksisterende IoT enheder på markedet idag. Faktisk forventes det at der i år 2020 vil være op imod 25 milliarder IoT enheder på verdensplan [6].

2.1.3 Smart Grid

Der er i folketinget vedtaget en lov, der siger at i år 2020 skal det samlede CO2 udslip for Danmark nedsættes. Stramningerne for dette gælder på flere områder, som kan påvirke både virksomheder og privat personer. Indenfor el produktionen, er kravet at største delen af den el vi benytter i Danmark, skal produceres ved hjælp af vind og solenergi, fremfor kulkraftværker som der benyttes idag. Det er dog ikke nok blot at opsætte flere vindmøller og solcelleanlæg. For at udnytte den producerede el på bedst mulig måde er det nødvendigt at der indføres en intelligent styring af elnettet. Det er netop denne intelligente styring, der udgør det, der kaldes for Smart Grid. Første step under udviklingen af den intelligente styring, er at udskifte samtlige elmålere i danske hjem, så de kan fjernaflæses. En af fordelene ved dette er, at det er muligt at aflæse forbrugernes elforbrug på timebasis, istedet for på års basis som der benyttes mange steder idag.

Denne udbygning af elnettet er en stor udgift for elselskaberne og deres kunder. Men det forventes at der på sigt vil være en stor besparelse for forbrugerne, idet de nye elmålere vil åbne op for variable elpriser. En af årsagerne til at elpriserne kan variere er, at der ikke produceres samme mængde el hver time, hvor faktorer som vind og vejr spiller en stor rolle. Samtidig skal den intelligente styring være med til at give forbrugerne endnu bedre muligheder for selv at producere el. Her skal det være muligt for forbrugerne med egne

solcelle anlæg eller vindmøller at sende det overskydende producerede el videre ud på el nettet. Dette kan både være med til at nedsætte CO2 udledningen fra kulkraftværkerne, men også at give en ekstra indtægt til husstanden. [7] [8]

En af problematikkerne med produktion af el sammenlignet med andre energiformer, er at det ikke er muligt at opbevare el på lager. Der findes dog store batterilagrer til opbevaring af el, men dette er en dyr og krævende løsning. Dog forventes det at Smart Grid på sigt, vil kunne udnytte et voksende salg af el biler, hvor deres kraftige batterier kan bruges til opbevaring af strøm. Her kan man i perioder hvor der er overskudsstrøm på el nettet oplade bilernes batterier til de lavere elpriser. Derefter kan man udnytte denne strøm i de perioder hvor el nettet er meget belastet og el priserne er høje. På måder som dette, forventes det at de mål der er opsat på sigt vil kunne nås og vi derved sammen kan være med til at nedsætte det samlede CO2 udslip.

Implementeringen af Smart Grid er en langsigtet plan, som kan udbygges i mange etaper. I første omgange er det udskiftning af elmåler hos forbrugerne. Det er dog forventet at i 2020 skal halvdelen af elforbruget i Danmark være produceret med vindkraft og i 2025 skal det intelligente el net være udbygget med de første funktionaliteter.

2.1.4 Opsummering af IoT Smart Home

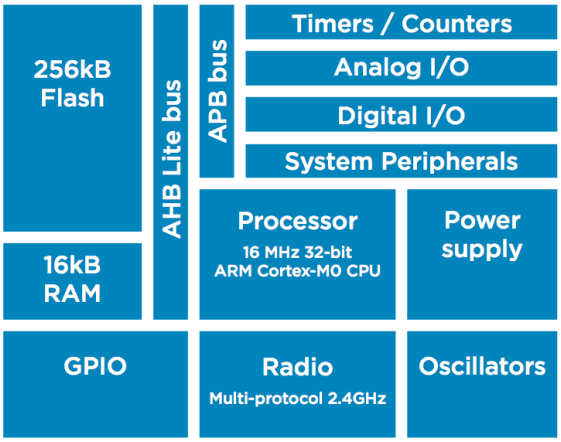
Smart Home, IoT og Smart Grid er alle teknologier, som der forventes meget af i de kommende år. Ses der på Smart Home er dette primært hjemme elektronik produkter. Dog indeholder de mange af de samme funktionaliteter som IoT enheder. Implementeringen af Smart grid teknologien i de danske hjem, vil kunne fungere i et samspil med de allerede eksisterende IoT og Smart Home produkter. Dette kan være central styring af elforbruget for hustandens elektronik, overvågning af det producerede el på solcelle anlægget eller styringen af huset varmeforbrug.

2.2 Valg af Mikroprocessor

Udgangspunktet for udviklingen af dette toolkit er en chip fra Nordic Semiconductor kaldet nRF51822. Denne er valgt da virksomheden Homatic, tidligere har haft gode erfaringer med netop denne chip. Der er her tale om et komplet system også kaldet et SOC (System on a Chip), hvilket vil sige at den indeholder CPU, RAM, flash, kommunikation og adskillige I/O porte. Firmaet Nordic Semiconductor er et Norsk firma som i over 25 år har specialiseret sig indenfor udviklingen af SOC systemer med integreret trådløs kommunikation og et lavt strømforbrug.

2.2.1 Nordic nRF51822

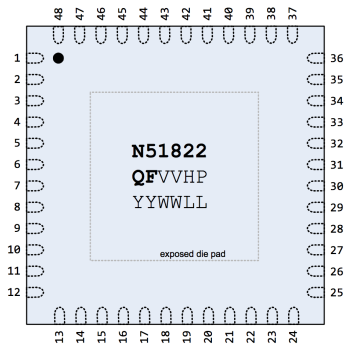
Nordic nRF51822 SOC er bygget op omkring en 32-bit ARM Cortex M0 CPU. Den fås i en udgave med enten 128 eller 256 kB flash hukommelse, 16 eller 32 kB RAM og en clock frekvens på 16 MHz. Kredsen er udviklet med henblik på trådløse applikationer, hvorfor den indeholder en 2.4 GHz Multi-protocol kommunikation. Dette vil sige at den understøtter både Bluetooth Low Energy og Nordics egne Gazell kommunikations protokol. På figur 2.1 vises en oversigt over opbygningen af den samlede nRF51822 kreds.



Figur 2.1. nRf51822 overblik

Som figuren viser, er processoren den centrale del af systemet, som de øvrige dele er bygget op omkring. Der ses også den omtalte Flash og RAM hukommelse som vha. en datakommunikations bus kan kommunikerer med enten processoren eller direkte med I/O portene. Systemet indeholder en GPIO (General Purpose Input/Output) som bruges til styring af de forskellige ind- og udgange på systemet. Udover dette ses en strømforsyning og en Oscillator som bl.a. bruges til styringen af Timers og Counters.

Nordic nRF51822 systemet kommer i en QFN48 indpakning som måler kun 7x7 mm og en figur af denne kan ses på figur 2.2. Den lille størrelse gør den meget anvendelig i udviklingen af Smart Home produkter. Som der ses på figuren har den 48 pin, som ved hjælp af en fleksibel GPIO kan bytte rundt på udgangene. Dette er med til at gøre den meget fleksibel i forbindelse med PCB design hvor man slipper for at trække lange baner rundt om nRF51822 chippen.



Figur 2.2. QFN48 Package

Det er muligt at benytte en forsyningsspænding fra 3.6 V og helt ned til 1.8 V til systemet. Dette giver gode muligheder for udviklingen af små trådløse systemer forsynet vha. en batterispænding.

2.2.2 Valg af Udviklingkit

For at gøre det lettere at udvikle systemer med nRF51822 chippen, vil der være benyttet et udviklingskit. Der findes flere forskellige udgaver, men fælles for dem alle er den lette tilgang til de forskellige pins på SOC systemet. Nogle udviklingskit har indbyggede strømforsyning enten vha. USB eller batteri, de kan have knapper, LED og en indbygget debugger som kan benyttes under udviklingen.

2.2.3 nRF51822 Go udviklingskit

Dette udviklingskit indeholder den omtale nRF51822 SOC, med 256 kB flash hukommelse og 16 kB RAM. Der er tale om et lidt ældre udviklingskit fra Nordic Semiconductor også kaldet for nRFgo eller PCA10004 og PCA10005 afhængig af antenne. Der er udlånt to af disse udviklingskit fra Homatic Engineering.

Udviklingskittene findes som nævnt i to udgaver, hvor den første har en indbyggede PCB antenne, og den anden har en ekstern antenne. Dette udviklingskit benyttes ofte sammen med et motherboard, som giver adgang til knapper, LED, USB og adskillige I/O porte. Dette er dog ikke tilgængelig i dette projekt, men udviklingskittene kan benyttes uden, vha. to tyve pins konnektorer placeret på kittenes underside. De to udviklingskit kan ses på figur 2.3 og 2.4.

Figur 2.3. nRf51822 PCB antenne

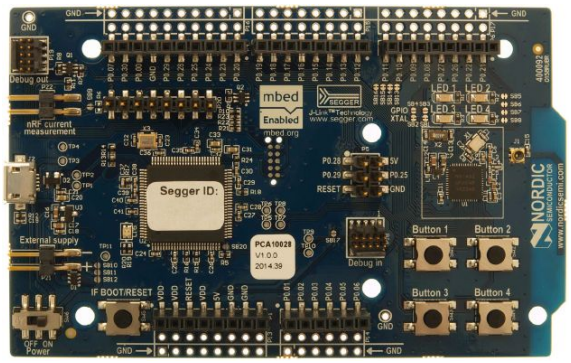
Figur 2.4. nRF51822 Extern antenne

2.2.4 nRF51-Development Kit

Et andet og nyere udviklingskit fra Nordic Semiconductor er kaldet nRF51-DK eller PCA10028. Dette kit indeholder til forskel en nRF51422 SOC istedet for den tidligere nævnt nRF51822. Årsagen til dette er at nRF51422 indeholde de samme specifikationer som nRF51822, på nær at den på kommunikations siden understøtter ANT protokollen. Men da der i dette projekt udvikles et system til nRF51822 systemet, vil ANT ikke blive benyttet eller yderligere beskrevet. Den indeholde en 256 kB flash hukommelse og 32 kB RAM, hvilken eventuelt kan begrænses i udviklingen til 16 kB, så det er muligt at implementere softwaren på nRF51822 systemet med disse specifikationer.

Udviklingskittet nRF51-DK kan ses på figur 2.5. Dette udviklingskit indeholder udover Nordic SOC kredsen, fire knapper og fire LED dioder. Der er mulighed for forsyning, vha. et batteri på undersiden og samtlige I/O porte er tilgængelige vha. konnektorer placeret i hver side på printet. Den har en integreret PCB antenne, som bruges i forbindelse med den 2.4 GHz kommunikations protokol. Den indbyggede Segger programmer og debugger gør det let at benytte udviklingskittet blot vha. en USB kabel. Denne fungerer ligeledes som

forsyning til systemet under udviklingen. Det er også muligt at forsyne udviklingskittet med en ekstern forsyning, blot den er imellem 1,8 og 3,6 V.



Figur 2.5. Nordic nRF51 Development Kit

2.2.5 Software Development Kit

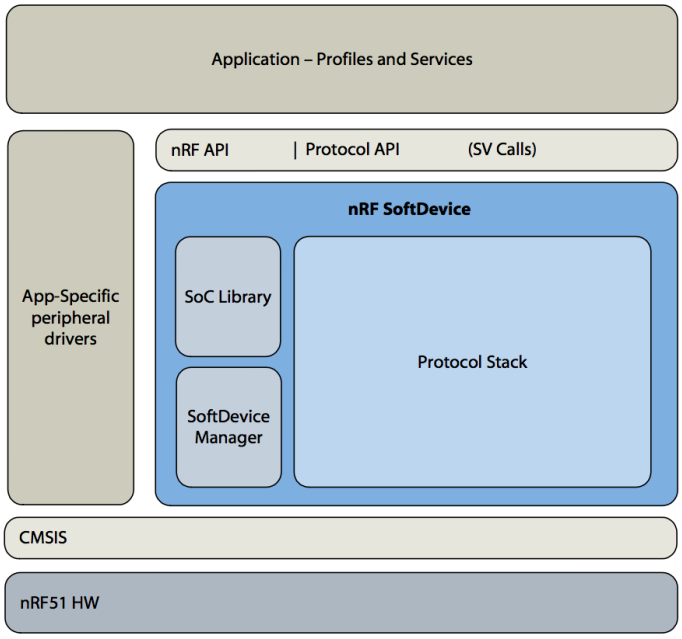
Et software Development Kit kaldes også for en SDK. Dette er et udviklingsværktøj, der gør det muligt for programmører at udvikle applikationer til forskellige systemer. I dette tilfælde er det udviklet af Nordic Semiconductor til deres nRF51822 kreds og giver adgang til udviklingen vha. et API (Application Programming Interface). Et API er en grænseflade som gør det muligt at udvikle software, som i dette tilfælde tilknytter sig til den hardware, som Nordic har stillet til rådighed for kredsen nRF51822.

Udover dette indeholder SDK pakken også drivere, biblioteker og applikations eksempler til udviklingskittet. Til udviklingen af softwaren i dette projekt er der benyttet udviklingsmiljøet Keil μvision 5. Dette er et system som anbefales at Nordic og som er tilgængelige i en gratis udgave.

2.2.6 nRF Softdevice

For at benytte den trådløse kommunikation på udviklingskittet har Nordic stillet en softdevice til rådighed. Dette er et stykke software, som indeholder kommunikations protokollerne for både Bluetooth Low Energy og deres egen Gazell kommunikation. Der er udviklet flere udgaver af denne softdevice, som hver indeholder forskellige funktionaliteter. Af softdevice kan nævnes S110, S120 og S130, som er henvendt til BLE applikationer. Imens S210 og S310 er henvendt aplikationer med ANT teknologien. Da der er tale om et toolkit til nRF51822, som ikke understøtter ANT teknologien, vil der kun blive arbejdet med en softdevice fra S1xx serien. På figur 2.6 ses arkitekturen for og omkring nRF softdevice.

Figure viser de enkelte del moduler som softdevicen indenholder. Nederst på figuren ses *nRF5 HW* hvilket repræsentere hardwaren for nRF51822 systemet. Ovenover denne ses *CMSIS* (Cortex Microcontroller Software Interface Standard) laget, som er udviklet af ARM og giver adgang til funktionaliteterne for deres Cortex M0 processor, uden først at skulle arbejde med en masse assembly kode. Herefter ses *nRF softdevice* som vha. *nRF API* og *Protocol API* kan kommunikerer med applikationen. Der ses ligeledes at applikationen



Figur 2.6. Softdevice arkitektur

også har direkte adgang til CMSIS og nRF51 HW delene vha. blokken *APP-Specifical peripheral drivers*.

Softdevicen er indelt i tre overordnede dele, hvor den første er *SoC Library*, som indeholder et API for de delte hardware ressourcer for systemet. Dernæst er der *Software Manager* som bl.a. har funktionerne til at aktivere og deaktiverer softdevicen. Sidst er der *Protocok Stack* som indeholder protokollerne til oprettelse af en Bluetooth Low Energy forbindelse. Denne softdevice er derved med til at forenkle opsætningen af BLE og Gazell applikationer på nRF51822 systemet.

Som nævnt tidligere er der tre udgaver af softdevicen fra Nordic.

- S110 anvendes til smart Home systemer, hvor en device skal oprette forbindelse til en masterenhed. Den anvendes ofte til funktioner som styring af en kontakt eller aflæsning af data fra en sensor.
- S120 anvendes til masterenheder, som skal kunne kommunikere med de enkelte smart Home produkter.
- S130 indeholder funktionaliteterne for både S110 og S120 softdevicen, og kan derved agere som enten device eller masterenhed.

Det er forventet at Nordic Semiconductor i de fremtidige udgaver, vil udfase S110 og S120 til fordel for kun at have softdevicen S130 til BLE applikationer. Udviklingskittet nRFgo beskrevet ovenfor understøtter desværre ikke S130 softdevicen, da det er en ældre udgave af nRf51822 chippen, hvorimod nRF51 udviklingskittet understøtter alle de ovennævnte softdevices.

2.3 Funktions Beskrivelser for Toolkit

Som nævnt tidligere tager dette projekt udgangspunkt i et projektforslag fra Homatic Engineering. Det var ønsket at udvikle et universelt IoT Smart Home toolkit med funktionaliteterne til, at kunne benyttes i forskellige brugsscenarier. Der vil i dette afsnit være en beskrivelse af de grundlæggende funktioner, som det er forventet at systemet skal indeholde. Beskrivelserne er lavet med udgangspunkt i det ønskede systemet nRF51822 fra Nordic Semiconductor, som også er beskrevet tidligere. Disse funktions beskrivelser vil videre i rapporten blive benyttet til opstilling af en kravspecifikation, design og implementering af toolkittet samt opstilling og udførsel af en accepttest.

Som nævnt er der tale om et universalt systemet, som overordnet set skal kunne tilkobles forskellige eksterne komponenter. Dette kan være en sensor, som måler en temperatur, eller en kontakt som bestemmer, om en lampe skal tændes eller slukkes. Men en ting er sikkert i begge disse situationer, og det er at systemet skal kunne håndtere disse input og sikre at den valgte funktion udføres. Når data fra en sensor aflæses, sker dette oftest på en lineær kurve, hvor temperaturen aflæses som en spænding fra eksempelvis 0-5 V. En af de funktioner systemet skal indeholde udover, at kunne aflæse den analoge spænding, er at skaler den målte spænding til en temperatur. Denne skalering er selvfølgelig ikke ens for alle sensorer, hvorfor det skal være muligt at tilføje og ændre i skalerings parametrene. En skalering kan udføres ud fra formel 2.1, hvor U_{in} er spændingen fra sensoren, a er omregningsfaktoren fra sensorens udgangssignal og T_o er skaleringen i forbindelse med den analoge indgang.

$$^{\circ}\mathrm{C} = U_{in} \cdot \frac{T_o}{a} \tag{2.1}$$

Som nævnt i beskrivelsen af Smart Home i afsnit 2.1.1, styres de fleste Smart Home funktionaliteter ud fra to kriterier, nemlig *Timed event* og *Triggered event*. Dette er funktioner som er uundværlige for toolkittet, og derfor skal være en meget central del af system designet. Set ud fra de tekniske krav til systemet, vil det sige at systemet skal indeholde Timer funktionaliteter, ligesom en mulighed for at modtage input fra enten en bruger eller en anden del af systemet.

Det endelige system kan bestå af to eller flere enheder, som skal kunne kommunikere trådløst med hinanden. Dette kan med fordel udføres med Bluetooth Low Energy eller Gazell kommunikations protokollerne, som begge er understøttet af udviklingskittene. Det er dog en nødvendighed at der udvikles én enhed som fungere som det centrale toolkit og har forbindelse til de øvrige enheder. Den centrale enhed skal fungere som styring for de andre toolkit, og indsamle de nødvendige data fra sensorer og lignende komponenter der er tilsluttet.

Det er en mulighed, at der indføres en logging funktionalitet på toolkittet. Dette giver mulighed for, at kunne tilgå de tidligere målte data, samt at kunne udføre statistiske beregninger på dataene, hvis dette skulle være nødvendigt. Logging funktionaliteten kan med fordel udføres på den centrale enhed, hvorved alle data ville være samlet på ét sted. Her ville det dog være nødvendigt, med en registrering af de enkelte data, så det er muligt at se, hvilken enhed det er modtaget fra.

For at gøre systemet så trådløst som muligt, skal det kunne forsynes med en batteri spænding. Dette giver muligheder for at benytte toolkittene på steder, hvor der ikke er adgang til en fast strømforsyning. Dog kan den centrale enhed med fordel placeres med mulighed for en fast forsyning, da den vil have et større strømforbrug. Årsagen til at centrale enheder kan have et højere strømforbrug er, at den skal være synlige for de øvrige enheder i større perioder. Dette vil medføre et væsentligt større strømforbrug. Det kræves dog at de øvrige enheder opbygges med mulighed for forsyning via en batterispænding.

Da der er tale om et universalt IoT Smart Home toolkit, skal det selvfølgelig være muligt at videreudvikle systemet, så det kan tilpasses de ønskede brugsscenarier. Det kan også være, at der skal tilføjes yderligere komponenter eller et ønske om at kommunikerer med Smart Home produkter fra andre producenter. En anden mulig videreudviklingsmulighed kan være at integrerer dette toolkit med Homekit løsningen fra Apple.

2.3.1 Opsummering af Funktioner for Toolkit

For at opsummere de ønskede funktioner for toolkittet, vil der herunder være opstillet en liste over de ønskede funktionaliteter. Som nævnt tidligere vil disse funktioner være med til at opstille en scenarie beskrivelse til det samlede toolkit i kravspecifikationen.

- Tilslutning af eksterne komponenter
- Skalering af sensor data
- Opbygning ud fra Timed og Triggered event
- Toolkit skal bestå af én central enhed og mindst en anden enhed
- Trådløs kommunikation mellem enhederne
- Skal indeholde en logging funktionalitet
- Mulighed for batteri forsyning, gælder dog ikke den centrale enhed
- Udvikles med henblik på videreudvikling og individuel tilpasning

2.4 Afgrænsning

Der vælges i dette projekt at afgrænse fra nogle af de tidligere nævnt funktionaliteter. Der vil dog være implementeret de standard funktioner som kræves for at toolkittet kan opfylde funktions beskrivelserne. Nedenfor er der opstillet en liste over de funktioner som der afgrænses fra i dette projekt.

- Minimere strømforbruget gennem udviklingen
- Sikkerhed for BLE forbindelsen
- Integration med Apple Homekit protokol

2.5 Problemformulering

Hvordan kan et Toolkit til IoT Smart Home designes med udgangspunkt i nRF51822 SOC fra Nordic Semiconductor?

Kravspecifikation 3

På baggrund af indledningen i kapitel 1 og problemanalysen i kapitel 2 vil der i dette afsnit være opstillet en kravspecifikation for det samlede IoT Smart Home toolkit. Kravspecifikationen vil være opdelt i tre dele, hvor det først er de overordnede krav til systemet. Herefter vil de tekniske krav være opstillet og sidst vil der blive opstillet krav ud fra de funktionsbeskrivelse der er for toolkittet i afsnit 2.3.

3.1 Overordnede Krav

De overordnede krav til det samlede toolkit kan ses på tabellen herunder og er opstillet ud fra funktionsbeskrivelsen i forrige afsnit.

Nr.	Parameter	Krav	Beskrivelse
1.1	Hardware	nRF51822 SOC	Toolkit skal opbygges ud fra nRF51822 kredsen fra Nordic Semiconductor
1.2	Hardware	Master og Device enheder	Det samlede toolkit skal bestå af mindst to enheder, hvor den ene er opsat som masterenhed og den anden som device
1.3	Kommunikation	Trådløs kommunikation	Skal kunne kommunikere trådløst med andre Toolkit
1.4	Digital I/O	Brug af digitale I/O porte	Det skal være muligt at tilslutte og aflæse data fra digital I/O porte
1.5	ADC	Tilslutning af analoge sensorer	Det skal være muligt at aflæse data fra analoge sensorer
1.6	Logging	Logging af data	Det skal være muligt at gemme de aflæste data

Tabel 3.1. Overordnede Krav

3.2 Tekniske Krav

I de tekniske krav kigges der mere individuelt på hvad der kræves for at de enkelt funktioner kan udføres.

Nr.	Parameter	Krav	Beskrivelse
2.1	Respons tid	Skal udføre kommando indenfor 200 mS	Det er forventet at en respons tid på mere en 200 mS, vil skabe en irritation hos brugerne.
2.2	Rækkevidde	Minimum 10 meters rækkevidde	Den trådløse kommunikationen mellem toolkit og anden enhed skal fungerer med en rækkevidde op til 10 meter.
2.3	Timed event	Skal indeholde timer til tidsstyring	Toolkit skal indeholde timer funktionaliteter til brug ved tidsstyring af toolkittets funktioner.
2.4	Triggered event	Skal kunne modtage triggered event	Toolkit skal have mulighed for at modtage triggered event.
2.5	Digital I/O	Skal undersøtte digitale I/O porte	Skal understøtte både digitale I/O porte
2.6	ADC	Tilslutning af anloge sensorer	Det skal være muligt at aflæse data fra analoge sensorer
2.7	Flash hukommelse	Logging vha. Flash hukommelse	Skal benytte den integrerede flash hukommelse til at gemme data.
2.8	2,4 GHz kommunikation	Kommunikation vha. BLE eller Gazell protokollerne	Kommunikation skal foregå ved enten at benytte Bluetooth Low Energy eller Gazell protokollen fra Nordic.
2.9	Batteri	2 års batteritid	Det skal være muligt at opnå 2 års batteritid for systemet.

Tabel 3.2. Tekniske krav

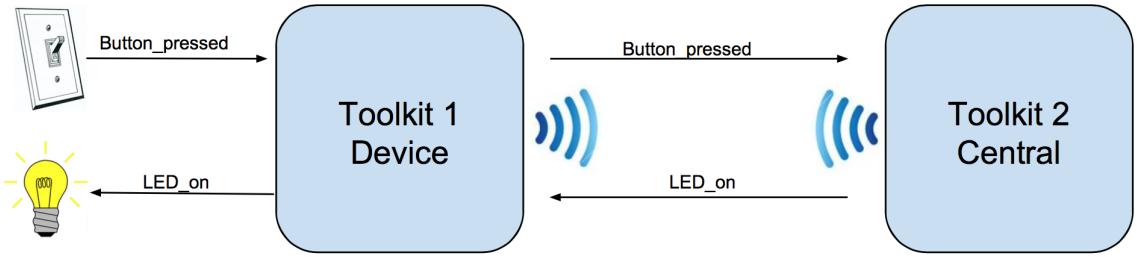
3.3 Krav ud fra Funktions Beskrivelser for Toolkit

Kravene i dette afsnit er opstillet ud fra funktions beskrivelserne i afsnit 2.3 og vil være opstillet som scenarie beskrivelser. Der vil være opstillet beskriver for brugsscenarierne kontaktstyring og varmekontrol. Ved udførsel af disse scenarie, vil alle de tidligere nævnte funktioner være benyttet, dog på nær logging funktionen.

3.3.1 Scenarie for Toolkit Kontaktstyring

Det første brugsscenarie er kontaktstyringen, hvor der stilles krav til toolkittet om at kunne kontrollere de digital I/O porte. Der kan på figur 3.1 ses en skitse af opstillingen for dette scenarie. Systemet skal opbygges sådan, at et input fra knappen, udløser et event på toolkit 1, som efterfølgende sender en kommando afsted til toolkit 2. Herefter skal toolkit

2 som indeholder styringsfunktionerne for det samlede system, sende en kommando til toolkit 1 om, at der skal tændes tændes for en lyskilde. Fordelen ved at styringsfunktioner er placeret på toolkit 2, er at der som en del af videreudviklingen kan tilkobles flere Device enheder. Her kan en af brugsmønstrene være, at om aftenen skal der tændes for lyskilde_1 og lyskilde_2, imens der om dagen kun tændes for lyskilde_1.



Figur 3.1. Scenarie for kontaktstyring

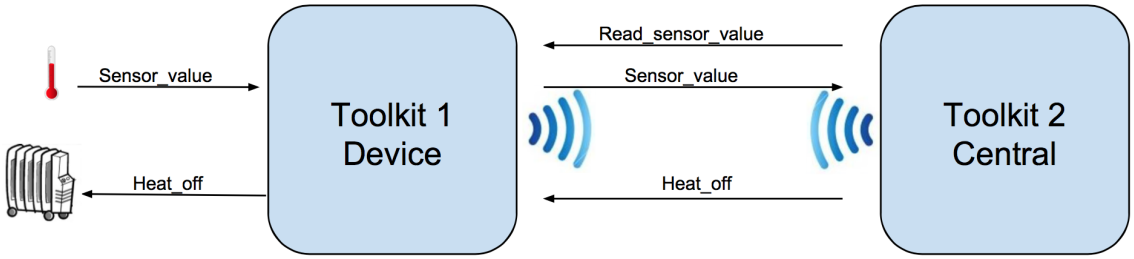
På tabel 3.3 er brugsscenarier for kontaktstyringen opdelt i mindre scenarier, som tilsammen vil danne grundlag for design og implementering af det samlede system.

Nr.	Test beskrivelse
3.1	Systemerne tændes og der oprettes en trådløs forbindelse mellem Toolkit 1 og Toolkit 2
3.2	Input fra en kontakt registreres og der afsendes en kommando til Toolkit 2
3.3	Toolkit 2 modtager kommandoen og afsender en ny kommando til Toolkit 1
3.4	Toolkit 1 modtager en kommando og tænder for lyskilden

Tabel 3.3. Scenarie beskrivelse for Toolkit kontaktstyring

3.3.2 Scenarie for Toolkit Varmekontrol

Det andet brugsscenarie som det forventes at toolkittet skal opfylde er varmekontrol. Her måles temperaturen med en sensor, hvorefter en varmekilde indstilles ud fra dette resultat og de forudbestemte temperatur grænseværdier. Der er på figur 3.2 vist en mulig opstilling for en varmekontrol vha. to Toolkit. Der ses på figuren hvordan Toolkit 2 giver besked til Toolkit 1 om at aflæse sensor værdien. Herefter vurdere Toolkit 2 om der skal tændes eller slukkes, for den varmekilden der er tilsluttet Toolkit 1. Det er ligeledes krævet, at der kan udføres en skalering på outputtet fra sensoren, så temperaturen kan aflæses i grader celsius. Endvidere forventes det at der kan måles temperaturer fra 0 - 50 °C.



Figur 3.2. Scenarie for varmekontrol

Ligesom ved kontaktstyringen i forrige sektion, er varmekontrollen også opdelt i mindre scenarier som kan benyttes gennem designet af systemet. Scenarierne giver ligeledes mulighed for at test det samlede systems funktionalitet i acceptesten.

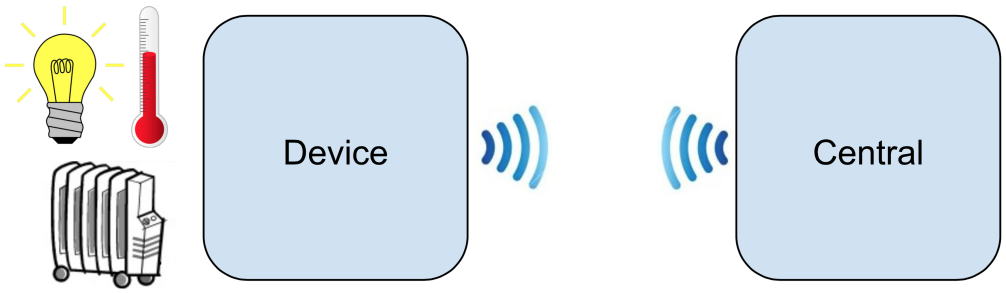
Nr.	Test beskrivelse
4.1	Systemerne tændes og der oprettes en trådløs forbindelse mellem Toolkit 1 og Toolkit 2
4.2	Der afsendes en kommando om at aflæse sensor værdien fra Toolkit 2
4.3	Toolkit 1 modtager en kommando og aflæser sensor værdien
4.4	Toolkit 1 afsender sensor værdien til Toolkit 2
4.5	Toolkit 2 modtager sensor værdi og laver en skalering til grader celsius.
4.6	Toolkit 2 vurdere ud fra den skalerede temperatur, om der skal slukkes eller tændes for varmekilden, og sender en kommando til Toolkit 1
4.7	Toolkit 1 modtager en kommando og slukker for varmekilden

Tabel 3.4. Manual for Toolkit varmekontrol

Design og Implementering 4

I dette kapitel designes og implementeres det samlede IoT Smart Home toolkit. Systemet vil være designet, ud fra de krav og funktions beskrivelser som er opstillet i kravspecifikationen i kapitel 3. Der vil gennem afsnittet være opstillet figurer og diagrammer som viser de enkelte dele af systemets funktionaliteter.

På figur 4.1 er der opstillet en illustration som viser det overordnede system. Der ses her, at systemet består af to enheder, en Device og en Central enhed, som kan kommnikerer trådløst med hinanden. Det vises også at Device enheden skal have mulighed for tilkobling af eksterne komponenter, som en lyskilde, en temperaturmåler og en varmekilde.



Figur 4.1. Det overordende system

De to enheder skal hver opbygges med udgangspunkt i nRF51822 systemet. Til dette kan de udviklingskit beskrevet i afsnit 2.2.2 med fordel benyttes. Designet af softwaren til de to enheder, vil være udviklet med udgangspunkt i eksempler og softdevices fra SDK pakken fra Nordic Semiconductor.

4.1 Modulopdeling

For at forenkle designet af de to toolkit kan de ønskede funktionaliteter deles op i mindre moduler. En af formålene med dette er, at identificere hvilke funktionaliteter hver af de to toolkit skal indeholde. Samtidig er det med til at danne et overblik over det samlede system, samt hvordan de enkelte dele af systemet skal samarbejde med de andre.

Identificering af disse funktionaliteter kan findes med udgangspunkt i de overordnede krav fra kravspecifikationen i kapitel 3. Kravene kan herefter benyttes til at definere de funktioner det enkelte toolkit skal indeholde, samt hvilke af funktionaliteterne der går

igen for begge enheder. Identificeringen af det enkelte toolkits funktioner kan findes ved, at opstille overordnede krav på en tabel. Herefter kan de funktioner som hvert toolkit skal indeholde markeres. En tabel for dette kan ses nedenfor på tabel 4.1.

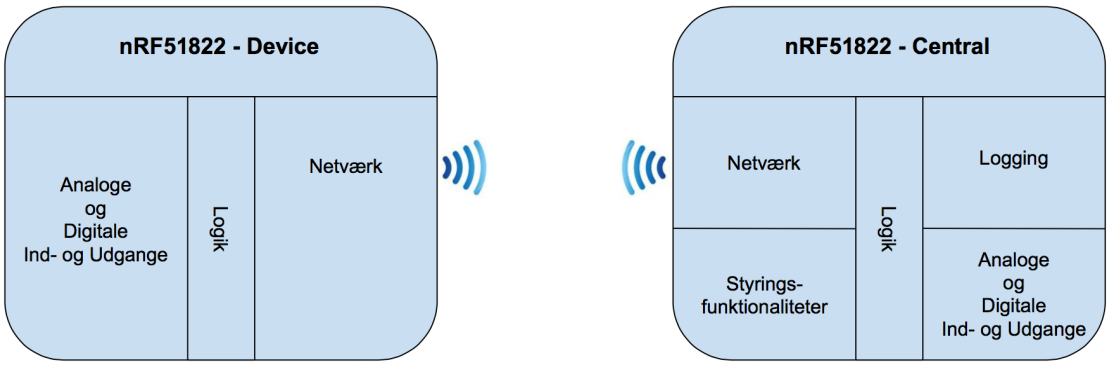
Krav	Device	Central
nRF51822	x	x
Netværk	x	x
Analoge indgange	x	(x)
Digitale ind- og udgange	x	(x)
Logging af data		x
Styringsfunktionaliteter		x

Tabel 4.1. Funktioner ud fra kravspecifikation

På tabellen ses det at nogle af de overordnede krav går igen på både Device og det Centrale toolkit. Det første er kravet om hvilken mikroprocessor, der skal benytte til projektet. Dette er fastlagt til nRF51822 systemet fra Nordic Semiconductor. Dernæst ses det at begge enheder skal indeholde en Netværks funktionalitet til kommunikation imellem de to enheder.

Da det er forventet at de eksterne komponenter skal tilsluttes Device enheden, stiller dette et krav om at der er adgang til både analoge og digitale ind - og udgange. Det kan dog være en fordel også at have adgang til disse forbindelser på den centrale enhed. Her kan en digital I/O port bl.a. benyttes til at tilslutte en LED for at vise status informationer for enheden eller en analog indgang til overvågning af strømniveauet for en batteriforsyning.

Der er på figur 4.2 opstillet en illustration, som viser hvilke funktionaliteter, hver af de to toolkit skal indeholde.



Figur 4.2. Modulopdeling af de to enheder

Som beskrevet i afsnittet valg af udviklingskit i problemanalysen er der to forskellige udviklingskit tilgængelig for dette projekt. Det første er en model kaldet nRF51-DK. Denne indeholder den nyeste version af nRF51x22 chippen, som giver funktionaliteterne for både nRF51422 og nRF51822 systemet. Dette udviklingskit er kompatibel med de nyeste versioner af SDK pakkerne fra Nordic, hvorfor software udviklet til dette kit vil benytte SDK pakken i version 10.0. Det andet udviklingskit hedder nRFgo og er et ældre udviklingskit fra Nordic Semiconductor og indeholder en ældre version af nRF51822

chippen som ikke understøtter de nyere udgaver af SDK pakkerne. Derfor vil udviklingen af software til dette kit tage udgangspunkt i SDK pakken i version 6.0.

Der vælges at det Centrale toolkit skal udvikles med udgangspunkt i nRF51-DK udviklingskittet, imens Device toolkittet skal udviklet ud fra nRFgo kittet. Det ville dog være fordelagtigt at udvikle de to systemet med udgangspunkt i samme SDK pakke, men da der kun er én nRF51-DK til rådighed er dette ikke muligt. Ydermere er et af kravene, at det skal være muligt at videreudvikle systemet, hvorfor det ikke vil være fordelagtigt at udvikle software til et forældet udviklingskit.

Nu hvor der benyttes forskellige SDK pakker til hver af de to toolkit, vil det betyder at udviklingen af software, ikke vil være ens på begge toolkit. Årsagen til dette er at de forskellige udgaver af softdevice og andre standard funktioner udvikles af Nordic Semiconductor, er ændret gennem de forskellige SDK versioner.

Der vil videre i dette kapitel, først være en beskrivelser af de funktionaliteter som går igen for begge toolkit, samt beskrivelser af de funktionaliteter hvor opsætningerne for de to SDK versioner er identiske. Sidste del af kapitlet vil indeholde beskrivelser af opsætningen og designet for henholdsvis Device og det Centrale toolkit.

4.2 Netværk

Blokken netværk indeholder alt kommunikation til og fra de to toolkit, og er en af de overordnede krav til det samlede system. Kommunikationen på nRF51822 systemet udføres vha. den indbyggede 2.4 GHz kommunikations protokol, som allerede er implementeret på udviklingskit. Her har man mulighed for at benytte enten Gazell kommunikations protokollen fra Nordic eller BLE (Bluetooth Low Energy) protokollen som trådløs kommunikation. Der vil i de næste afsnit være en uddybende beskrivelse af de to teknologier.

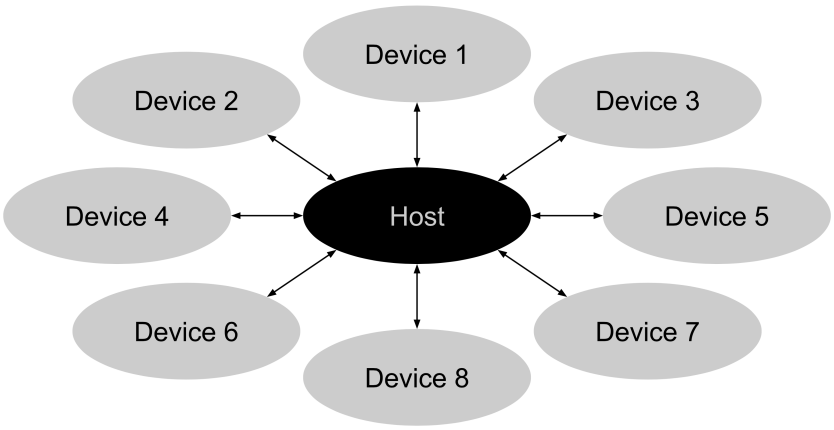
Udover den trådløs kommunikation, vil det også være en mulighed at implementerer en kablet kommunikation. Denne kan med fordel benyttes under udviklingen og test af softwaren som en debugger. Her vil der være tale om en UART (Universal Asynchronous Receiver/Transmitter) forbindelse, som sender data til en terminal på en computer. Det vil selvfølgelig også være en mulighed at benytte den trådløse kommunikation under udvikling som en debugger.

4.2.1 Gazell Link Layer

Gazell Link Layer eller blot Gazell er en kommunikations protokol, udviklet af Nordic Semiconductor og benytter deres 2.4 GHz kommunikations sender og modtager. Det er udviklet ligesom Nordics øvrige produkter, til at have et lavt strømforbrug som derved giver mulighed for anvendelse i mange forskellige typer af trådløse applikationer.

Gazell fungere ved hjælp af Host og Device metoden. Dette vil sige at en enhed er opsat som Host og den anden som Device. Det er enheden opsat som Host som skal lytte efter Device enheden. På denne måde kan man bygge systemer hvor Device enhederne, kan være meget strømbesparende. Dette skyldes at Gazell protokollen derfor kun behøver at være aktiv i det tidsrum hvor det er nødvendigt at sende data, hvorefter den igen kan gå i

dvaletilstand. Host enheden derimod vil være knap så strømbesparende og være afhængig af en større strømkilde. På figur 4.3 ses et overblik over opkoblingen, hvor det også vises at der maksimalt kan tilsluttes otte Device enheder for hver Host [11].



Figur 4.3. Figur over Host og Device [11]

Opbygningen af Gazell protokollen gør den meget anvendelig til Smart Home produkter. Dog kan den maksimale kapacitet på otte Device enheder for hver Host, være i underkanten på længere sigt, hvor det er forventet at en stor del af husstandens elektronik produkter er tilsluttet. Det er dog muligt at udvide den maksimale kapacitet, da en device enhed kan have forbindelse til flere Host enheder på en gang. På denne måde kan mere avancerede systemer opbygges, som giver mulighed for et uendeligt antal Device og Host enheder tilsluttet. En anden begrænsning med Gazell er at det kun kan benyttes imellem enheder fra Nordic Semiconductor [11].

Det er dog muligt at implementere to kommunikations protokoller på nRF51822 systemet. Dette kan give mulighed for at benytte både Gazell og BLE protokollen, men de kan dog ikke benyttes på samme tid. I dette projekt vælges der dog kun at implementere én kommunikationsprotokol, hvorfor Gazell ikke benyttes. Dette skyldes bl.a. at der er opstillet et krav om at det skal være muligt at videreudvikle det samlede systemet, så det kan være kompatibelt med Smart Home produkter fra andre producenter.

4.2.2 Bluetooth Low Energy

Bluetooth er idag en af de mest brugte teknologier til at kommunikere trådløst imellem enheder. Det blev udviklet tilbage i 90'erne i et samarbejde imellem Intel, Ericsson, Nokia og senere også Toshiba og IBM. Bluetooth er opkaldt efter den danske konge Harald Blåtand, som var konge fra 958-987 og er kendt for at have reageret over både Danmark, Sverige og Norge [9]. På figur 4.4 ses en tegning af Runestenen lavet af Jim Kardach fra Intel, hvor Harald Blåtand holder en computer i den ene hånd og en mobiltelefon i den anden. Dette billede blev brugt under en præsentation af Bluetooth teknologien og viser hvordan den trådløse teknologi har til hensigt at forbinde mobiltelefonen og computeren med hinanden [10].

Efter lanceringen af den først version af Bluetooth teknologien er det gået hurtigt og idag er Bluetooth version 4.0 standard i alle nyere computer, mobiltelefoner og tablet. Denne

Figur 4.4. Figur af Harald Blåtand lavet af Jim Kardach [10]

version kaldes også BLE (Bluetooth Low Energy) eller Bluetooth Smart. BLE indeholder alle de funktioner som idag er krævet af moderne elektronik produkter. Især to ting er med til at gøre den meget populær, det første er det meget lave strømforbrug som er optimalt til batteri forsynede enheder. Det andet er dens evne til at arbejde sammen med andre allerede eksisterende BLE enheder.

En anden fordel ved at benytte teknologier som BLE til Smart Home produkter er med hensyn til sikkerheden. En af disse grunde er at den ikke har en direkte forbindelse til husets trådløse router og internettet. Dette vil sige at man er nød til at være indenfor dens rækkevidde for at forsøge at logge på. Der er dog også implementeret en 128 bit AES data kryptering på forbindelsen, som dog ikke vil være videre beskrevet i denne rapport.

Forskellen på BLE og de tidligere version af Bluetooth er udover strøm forbruget også en mere begrænset overførelses hastighed på 1 Mbit/s mod tidligere op til 3 Mbit/s. Rækkevidden er angivet til over 100 m, men ved normal brug forventes denne rækkevidde at være væsentligt reduceret. Almindelig Bluetooth er dog også benyttet idag, da den højere data overførselshastighed bl.a. giver mulighed for trådløs overførsel af lyd til musikanlæg og trådløse hovedtelefoner. Der vil i denne rapport ikke være yderligere beskrivelser af de tidligere Bluetooth versioner, da de ikke er tilgængelige på de benyttede udviklingskit.

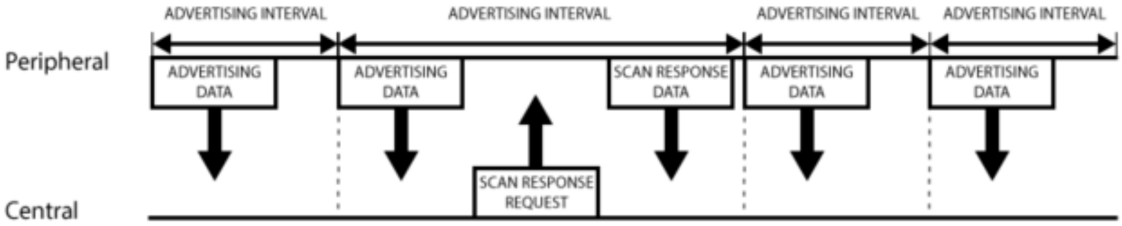
Opbygningen af BLE

BLE er opbygget ligesom Gazell protokollen, hvor der er en master og en device enhed. Device enheden er igen den strømbesparende enhed, som kan være tilsluttet en temperatur sensor, kontakt eller lignende.

En af fordelene ved BLE er, at det er understøttet af alle nyere smartphone og tablet på markedet. Dette gør det meget interessant for smart Home producenterne, da det er forventet at det er disse enheder som skal kontrollere smart Home produkterne.

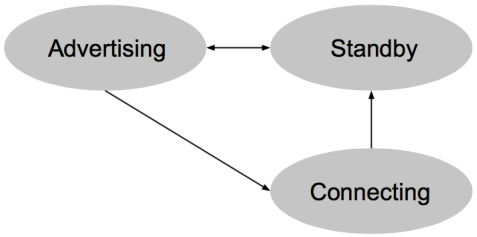
På figur 4.5 ses funktionen *advertising* for BLE, hvor Peripheral er device som kan være en kontakt og Central som kan være en smartphone. På figuren ses hvordan Peripheral

sender en pakke kaldet *advertising data* og derefter afventer svar fra den Centrale enhed. På figuren ses også hvordan denne *advertising data* pakke sendes med et fast intervaller også kaldet *advertising interval*. Dette interval er ikke fast defineret og kan derfor indstilles efter behov, hvilket kan være med til at formindske det samlede strømforbrug. Efterfølgende når Central enheden ønsker at oprette en forbindelse, sendes der en *scan response request* pakke tilbage til Peripheral enheden.

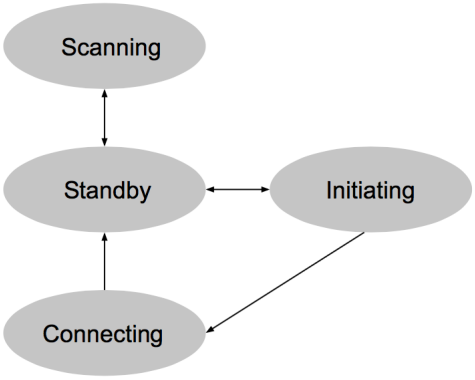


Figur 4.5. Figur over BLE advertising [12]

BLE forbindelserne kan under brug opdeles i stadier. For Peripheral eller device enhederne er der tre stadier, imens der for Central eller master enhederne er fire stadier. Disse kan ses på figurerne 4.6 og 4.7. Der ses her hvordan device enhederne kun har mulighed for at advertise og derved gøre sig synlig for master enhederne. Det er herefter master enhedernes opgave vha. scanningen af finde den rette device og oprette en forbindelse. Dette sker ved, at master enheden skifter fra scanning tilbage til standby for derefter at starte en initailisering inden forbindelsen oprettes imellem de to enheder [13].



Figur 4.6. Stadier for device enhederne

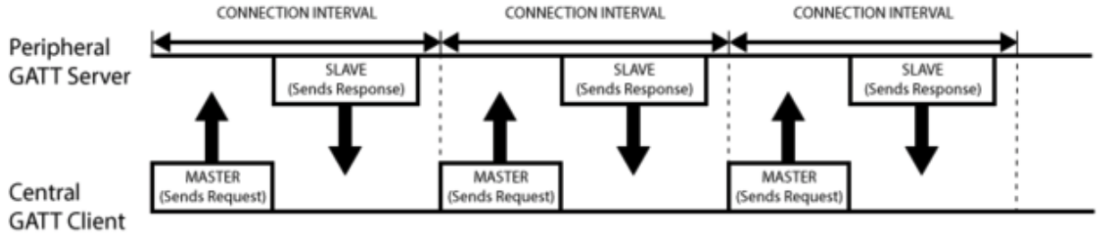


Figur 4.7. Stadier for master enhederne

En anden mulighed for at sende data fra en device enhed til en eller flere master enheder er ved at benytte den såkaldte *Broadcasting* metode. Dette er en måde hvorpå device enheden benytter advertising data pakken, som kan ses på figur 4.5. Denne metode er meget anvendelig hvis en bruger eller et system blot skal modtage data fra en enhed, men ikke nødvendigvis behøver at sende data tilbage. Metoden er ofte brugt på bl.a. museer, hvor gæsterne kan få information om kunstværkerne direkte på deres smartphone, blot ved at nærme sig det ønskede kunstværk. Dette vil dog ikke være benyttet i dette projekt,

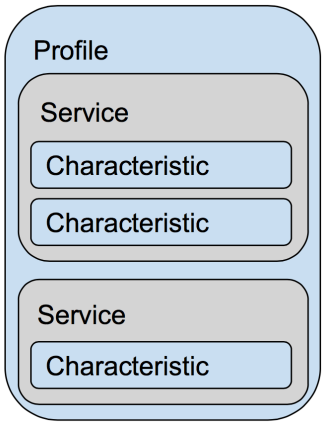
da de opstillede funktions scenarier fra kravspecifikationen siger, at der skal oprettes en forbindelse imellem to enheder.

Generic Attribute Profile eller GATT er en del af BLE opbygningen, og sørger for at to enheder kan kommunikere med hinanden. GATT bliver dog først en nødvendighed når en forbindelse imellem en device og master er oprettet. Dette vil sige, at ud fra figur 4.6 og 4.7 er der oprettet en forbindelse imellem de to enheder vha. connecting blokken. Det skal lige huskes at en device kun kan oprette forbindelse til en master, imens masteren godt kan have forbindelser til flere device enheder på samme tid. Når en forbindelse oprettes, er det første der sker, at device enheden vil foreslå et interval som også kaldes for *connection interval*. Herefter vil master enheden, ved starten af hvert interval, sende en forespørgsel til devicen for at se om der er data tilgængelig. På figur 4.8 kan en data transaktion imellem en master og device enhed ses.



Figur 4.8. Figur af GATT data transaktion [13]

Det sidste der vil være nævnt omkring opbygningen af BLE er en Profil. En BLE profil er overordnet set en applikation, som kan indeholde forskellige funktioner, eksempelvis at oprette en forbindelse til en anden BLE enhed. En profil indeholder typisk forskellige Services og Characteristics, og der er på figur 4.9 opstillet en illustration som viser dens opbygning. En profil kan enten være udviklet af en producent og indeholde standard funktionaliteter, eller være bruger defineret og indeholde specialle funktioner tilpasset det udviklede system.



Figur 4.9. Figur af profile, services og characteristic [13]

Services i en profil har bl.a. til opgave at opdele dataene, for efterfølgende at overføre dem til data pakker kaldet characteristics. Dette kan være en temperarur målt med en

sensor, som opdeles i data pakker, inden de sendes over en BLE forbindelse til en anden enhed.

Der vil gennem udviklingen af dette projekt være benytte standard BLE opsætninger fra Nordic Semiconductor. Dette skyldes at Nordic har tilpasset deres softdevices til at udnytte nRF51822 system bedst muligt, både hvad angår et minimalt strømforbrug men også med hensyn til at sikre et stabilt system.

4.2.3 UART

En anden form for kommunikation der er muligt med udviklingskittene er, at oprette en UART forbindelse til en terminal på en computer. På denne måde kan der let vises data og meddelelser for systemet under brug. På et system som dette vil UART forbindelsen til en computer dog oftest benyttes i udviklingsfaserne, da en computer ikke vil være tilsluttet toolkittene under normal brug.

Udviklingskittet nRF51-DK som i dette systemet benyttes til den Centrale enhed, kan vha. den indbyggede Segger programmer og debugger benyttes som forbindelse mellem udviklingskittet og en computer når en UART forbindelse oprettes. Dette er dog ikke muligt med udviklingskittet som benyttes til Device enhederne, da der benyttes en ekstern programmer og debugger som ikke indeholde denne funktionalitet. Der vil derfor ikke oprettes en UART forbindelse imellem Device enheden og en computer i dette projekt. Dog kan der under udvikling enten tilsluttes en ekstern UART forbindelse til en computer eller der kan sendes meddelelser til en computer via BLE forbindelsen til det Centrale toolkit.

4.2.4 Opsætning af UART Forbindelsen

Opsætningen af UART forbindelsen vil i dette afsnit være beskrevet med udgangspunkt i det til Centrale toolkit, hvilket skyldes at opsætningen for de to systemer minder meget om hinanden og at der er tale om en standard funktionalitet. På kode udsnittet nedenfor ses funktionen *uart_init()* som benyttes under opsætningen af softwaren.

```
1  //UART Initialization
   static void uart_init(void)
3  {
       uint32_t                     err_code;
5      const app_uart_comm_params_t comm_params =
       {
7          RX_PIN_NUMBER,
           TX_PIN_NUMBER,
9          RTS_PIN_NUMBER,
           CTS_PIN_NUMBER,
11         APP_UART_FLOW_CONTROL_ENABLED,
           false,
13         UART_BAUDRATE_BAUDRATE_Baud38400
       };
15
       APP_UART_FIFO_INIT( &comm_params,
17                         UART_RX_BUF_SIZE,
                           UART_TX_BUF_SIZE,
19                         uart_event_handle,
```

```
                        APP_IRQ_PRIORITY_LOW,
21                      err_code);
    APP_ERROR_CHECK(err_code);
23 }
```

Som kodeudsnittet viser, bliver der under opsætningen af UART forbindelsen opsat forskellige parameter. Dette er bl.a. parameter der bestemmer hvilken udgange på systemet der skal benyttes, hvilken overførsels hastighed skal data sendes med, også kaldet for baudraten osv. Udover disse parameter, kan det også ses at der opsættes en FIFO (First In First Out). Denne har til funktion at håndterer de forskellige data der sendes og modtages med UART forbindelsen.

For efterfølgende at benytte denne UART forbindelse til afsendelse af data, kan et eksempel for dette ses på kodeudsnittet nedenfor. Her ses det hvordan en *adc_sample* sendes vha. en standard C funktion, til en tilsluttet terminal på en computer via UART forbindelsen.

```
1  // Sending data with the UART
   printf("%d\r\n", (int)adc_sample);
```

Til modtagelse af data fra UART forbindelsen benyttes endnu en funktion fra Nordic. Funktionen hedder *app_uart_get(&data_array[index])* som henter dataene fra UART forbindelsen, for herefter at gemme dem i et data array til videre brug programmet.

Denne UART forbindelse er primært tiltænkt brug under udviklingen, og ikke under normal brug, hvorfor det vil være en fordel at kunne slå funktionen til og fra. Dette kunne med fordel være en funktion for toolkittet, hvor det er muligt at sætte systemet i udviklingsmode eller produktionsmode hvor meddelelser ikke sendes over UART forbindelsen. Udviklingen af denne funktion vil være beskrevet under afsnittet omkring det Centrale Toolkit i afsnit 4.6.

4.3 Opsætning af Ind- og Udgange for Toolkittene

En anden funktionalitet som går igen for både Device og det Centrale toolkit, er brugen af analoge og digitale ind- og udgange. Dette er funktionaliteter, som gør det muligt for de to systemer at kommunikere med eksterne komponenter. Opsætningerne for dette er stort set identisk for de to udviklingskit som benyttes i projektet, hvorfor der vil være en fælles beskrivelse af opsætningerne for henholdsvis de analoge indgange, samt de digitale ind- og udgange.

4.3.1 Opsætning af Analog Input

Et af kravene til systemet er tilkoblingen af eksterne sensorer. Dette kan udføres vha. de analoge indgange på de to udviklingskit. Opsætningen af de analoge indgange er nødvendige, i forbindelse med scenariet hvor en varmekontrol skal udvikles. Her skal det være muligt at aflæse en temperatur fra en sensor tilsluttet toolkittet. Denne værdi kan efterfølgende videre behandles af de øvrige moduler i systemet.

Opsætningen af de analoge indgange kan udføres vha. standard opsætninger til de to udviklingskit. Dog kan der med fordel justeres på tre parameter, afhængig af hvilken sensor der tilsluttes udviklingskittet. Den første af disse paramter er ADC (Analog to Digital Converter) opløsningen. Her kan der benyttes en opløsning på enten 8, 9 eller 10 bit, alt efter hvilken præcision der er ønsket for målingen. Den næste parameter er skaleringen af input signalet. Denne skalering er med til at bestemme den maksimale spænding, der må tilsluttes den analoge indgang på udviklingskittet. Som skalering kan der vælges $\frac{1}{1}$, $\frac{2}{3}$ eller $\frac{1}{3}$ af reference spændingen. Den sidste parameter er selvfølgelige referencespændingen som skal vælges.

Et eksempel på dette kan være hvor systemet er opsat med en skalering på $\frac{1}{3}$ og en reference spænding på 1,2 V. Her må den maksimale spænding på den analoge indgang højst må være 3,6 V. Dette er også tæt på den maksimale spænding der må tilsluttes indgangene på nRF51 kredsen. Den maksimale grænse går ved VDD spændingen + 0,3 V, hvor VDD er forsyningsspændingen for hele kredsen, som maksimalt må være 3,6 V [14].

For at vise opsætningen af den analoge indgange er der nedenfor indsat et kodeudsnit over dette. Der ses her på linje fire, at der vælges en konfiguration kaldet *NRF_ADC_CONFIG_DEFAULT*. Dette er en standard opsætning, hvor der benyttes en 10 bit opløsning til den indbyggede ADC, en skalering på $\frac{1}{3}$ samt en reference spænding på 1.2 V.

```
// ADC initializing
void adc_init(void)
{
  const nrf_adc_config_t nrf_adc_config = NRF_ADC_CONFIG_DEFAULT;

  // Initializing  and configure the ADC
  nrf_adc_configure( (nrf_adc_config_t *)&nrf_adc_config);
  nrf_adc_input_select(NRF_ADC_CONFIG_INPUT_2);
  nrf_adc_int_enable(ADC_INTENSET_END_Enabled << ADC_INTENSET_END_Pos);
  NVIC_SetPriority(ADC_IRQn, NRF_APP_PRIORITY_HIGH);
  NVIC_EnableIRQ(ADC_IRQn);
}

// ADC interrupt handler
void ADC_IRQHandler(void)
{
  nrf_adc_conversion_event_clean();
  adc_sample = nrf_adc_result_get();

  // trigger  next ADC conversion
  nrf_adc_start();
}
```

På kode udsnittet kan det ligeledes ses, at opsætningen består af to funktioner, nemlig en *adc_init* som opsætter den analoge indgang og en *ADC_IRQHandler* som opsætter en interrupt rutine. Denne interrupt rutine har til funktion, at aflæse dataene når de er klar, hvorefter de videre kan benyttes af systemet. Efter hver aflæsning af data, ses det nederst i kode udsnittet at en ny adc aflæsning startes.

4.3.2 Opsætning af Digital I/O Porte

Det andet krav omkring tilslutning af eksterne komponenter er tilslutning vha. de digitale ind- og udgange på udviklingskittene. Dette giver funktionaliteter til tilslutning af kontakter, knapper, LED og lignende komponenter. Denne funktionalitet benyttes under udviklingen af de to scenarier opstillet i kravspecifikationen omkring en kontaktstyring og en varmekontrol.

Det er forholdsvis simple funktioner der benyttes under opsætningen af de digitale ind- og udgange på udviklingskittene. Dette kan ses på kode eksemplet indsat nedefor, hvor *LED_4* opsættes som en udgang. Herefter er der indsat en kode som skiftevis sætter udgangen høj og lav, som derved tænder og slukker for *LED_*4 med et interval på 500 ms.

```
// Port_init
  nrf_gpio_cfg_output(LED_4);

while(true)
{
  // LED on
  nrf_gpio_pin_clear(LED_4);
  nrf_delay_ms(500);

  // LED off
  nrf_gpio_pin_set(LED_4);
  nrf_delay_ms(500);
}
```

På kode eksemplet nedenfor ses opsætningen af den digital indgang *BUTTON_1*, som en pull down opsætning. Efterfølgende er der indsat en funktion som får en LED til at lyse hvis der trykkes på en knap tilsluttet *BUTTON_1*.

```
// Port_init
  nrf_gpio_cfg_input(BUTTON_1, BUTTON_PULL);

// Button
void button_pressed()
{
  if (nrf_gpio_pin_read(BUTTON_1) == 0)
  {
    nrf_gpio_pin_clear(LED_4);
  }
  else
  {
    nrf_gpio_pin_set(LED_4);
  }
}
```

4.4 Logik

I dette afsnit beskrives logik modulet, som har funktionaliteterne til at styre de øvrige moduler. Derved kan det bl.a. siges, at det er den, som bestemmer hvornår og hvordan, systemet skal oprette forbindelse til andre enhed eller hvilke data der skal sendes. Det er vigtigt, at de funktioner som udvikles til logik funktionaliteterne for både Device og den Centrale enhed, opfylder de opstillede krav fra kravspecifikationen.

Styringen af systemet skal foregå ud fra de tidligere nævnte *Timed* og *Triggered event*. Dette er vigtige funktionaliteter, som bestemmer hvornår de enkelte funktioner skal udføres. Der er dog nogle hardware og software krav, som er nødvendige for at systemet kan udføre disse handlinger. Først kan der nævnes at der skal implementeres timer funktionaliteter til styringen af de tidsbestemte funktioner. Derudover skal der benyttes en event funktionalitet, som skal kunne starte funktioner ud fra input fra udefrakommende hændelser.

En anden funktionalitet er at styre de eksterne komponenter, som er tilkoblet toolkittet. Dette kan enten være komponenter tilkoblet direkte til I/O portene på udviklingskittet eller trådløse komponenter tilsluttet gennem netværks blokken. For at systemet kan understøtte de forskellige komponenter, er det vigtigt at opsætningen af I/O portene udføres korrekt og tilknyttes de rigtige funktioner i softwaren. Der er i Kravspecifikationen i kapitel 3 er der nævnt et par brugsscenarier, som stiller krav til, at systemet kan kontroller både digitale og analoge I/O porte. Ligeledes er det vigtige at systemet er designet så funktionerne som styringen af en kontakt udføres indenfor 200 ms.

Når eksterne komponenter som en temperaturmåler tilsluttes toolkittet, aflæses temperaturen ofte i en lineær graf fra. eks. 0-5 V. Denne værdi giver ikke en brugbar temperatur for en eventuel bruger. Derfor er det nødvendigt med funktionaliteter som kan omregne disse data til de rigtige værdier. Det er her vigtigt, at systemet understøtter forskellige sensorer, da disse omregninger kan være meget forskellige. Der kan være situationer hvor dataene fra sensoren ikke er lineære og en mere avancerede skalering er nødvendig.

Toolkittet skal også indeholde funktioner, som kan udføres ud fra resultaterne af bl.a. sensor værdierne. Dette kan være funktionen til varmekontrol, som er et af kravene fra kravspecifikationen. Her skal en temperatur overvåges og ud fra to grænseværdier skal systemet bestemme om der skal tændes eller slukkes for en varmekilde. Det kan også være en funktioner, hvor det er nødvendigt at kontrollere en ekstern enhed vha. et PWM signal. Dette kan give mulighed for at tilslutte endnu flere eksterne komponenter til toolkittet og derved giver flere muligheder for videreudviklingen af systemet.

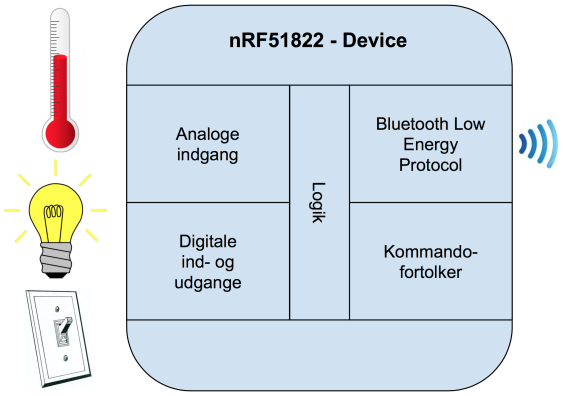
For at opsummere de ønskede funktioner til logik funktionaliteterne for toolkittet er de listet herunder.

- Timer funktionaliteter
- Interrupt funktionaliteter
- Brug af ind- og udgange
- Behandling af sensor data
- Skalering af sensor data

- Udføre funktioner ud fra sensor værdi
- Benytte netværk funktionaliteter

4.5 Device Toolkit

Systemet udviklet gennem dette projekt vil som nævnt tidligere bestå af to enheder, en Device og en Master toolkit. I denne sektion vil designet af Device enheden være beskrevet. Device enheden har til funktion at kontrollere de eksterne komponenter og integrerer dem som en del af toolkittet. Den indeholder også kommunikations funktionaliteterne til vha. BLE at oprette en forbindelse til det centrale toolkit. For at vise de overordnede krav til Device enheden er der på figur 4.10 opstillet en illustration som viser de funktionaliteter det skal indeholde.



Figur 4.10. Figur over device systemet

På figuren ses at funktionerne som de analoge indgang og de digitale ind- og udgange er standard funktioner som allerede er beskrevet i de tidligere afsnit. For at benytte disse funktioner skal der udvikles en logik funktionalitet speciel til denne device enhed. Der ses ligeledes, at BLE kommunikations protokollen også er en standard Nordic funktion, men da der benyttes forskellige softdevices til de to toolkit, vil opsætningen være beskrevet senere i dette afsnit. Det er her vigtigt at der benyttes en softdevice som indeholder Device funktionaliteterne. Sidste blok på illustrationen viser en kommandofortolker, som benyttes når den centrale enhed sender kommandoer til denne enhed.

Det er besluttet i kravspecifikationen i kapitel 3, at systemet skal understøtte to brugsscenarier, nemlig kontaktstyring og varmekontrol. ud fra disse scenarier kan de eksterne komponenter udvælges og systemet kan designes op omkring disse. De brugsscenarier som er udvalgt, indeholder funktioner som sikre at det samlede toolkit kan understøtte mange standard komponenter på markedet. Det er ligeledes forventes at de fleste brugsscenarier for et universalt toolkit som dette, vil være en kombination af standard funktionaliteter.

For at udspecificere de valgte brugsscenarier, kan de enkelte scenarier opdeles i mindre del moduler. I scenariet hvor en kontaktstyring udvikles er det første der kræves en digital indgang. Denne er tilkoblet en knap, som sender et signal til indgangen når der trykkes på knappen. Dernæst kræves det at der er opsat en digital udgang som kan styre en

lyskilde. I det andet brugsscenarie hvor en varmekontrol skal udvikles, skal der benyttes en temperatur sensor. Denne har et udgangssignal givet som en spænding, og kræver fra systemet en analog indgang. Varmekontrollen skal yderligere kunne styre en varmekilde. Denne kræver ligesom styringen af en lyskilde en digital udgang. Grænsefladerne for de eksterne komponenter tilsluttet Device toolkittet er også vist på tabel 4.2.

Kontaktstyring		
Ekstern komponent		Udviklingskit
Knap	$\longrightarrow$	Digital indgang
Lyskilde	$\longleftarrow$	Digital udgang

Varmekontrol		
Ekstern komponent		Udviklingskit
Sensor	$\longrightarrow$	Analog indgang
Varmekilde	$\longleftarrow$	Digital udgang

Tabel 4.2. Komponent grænseflader for Device Toolkit

Nu hvor de overordnede grænsefladerne imellem de eksterne komponenter og toolkittet er defineret, kan software designet for de enkelte del systemer udvikles.

4.5.1 Opsætning af Standard Funktionaliteter

Device enheden benytter flere af de standard funktionaliteter beskrevet i de foregående afsnit. På kode udsnittet nedenfor ses et lille overblik over hvilke funktioner der benyttes for denne del af systemet.

```
1  leds_init();       // LED/Digital output initialising
   buttons_init();    // Button/Digital input initialising
3  adc_init();        // Analog input initialising
```

Der vil i dette projekt være opsat fire digital udgange, som er forbundet til LED dioder. De har til opgave at vise status for de forskellige funktioner for toolkittet. Der er på tabel 4.3 opstillet en oversigt over de forskellige status LED dioder, samt hvad de symboliserer.

BLE advertising	Rød LED
BLE forbundet	Grøn LED
Kontakt funktionalitet	Blå LED
Varmekilde	Rød LED

Tabel 4.3. Status LED setup

4.5.2 Opsætning af BLE på Device Udviklingskit

Som nævnt i afsnit 2.2.6 benyttes der en softdevice til implementeringen af BLE (Bluetooth Low Energy) funktionaliteten på udviklingskittene. I denne del af systemet skal der opsættes Device funktionaliteter for enheden, hvorfor softdevice S110 benyttes. Denne enhed er opsat som device da der skal oprettes forbindelse til en master enhed. I dette

projekt vil master enheden være det andet toolkit, som gennem rapporten også kaldes det Centrale toolkit og være beskrevet i afsnit 4.6.

Opsætningen af BLE forbindelsen på nRFgo udviklingskittet tager udgangspunkt i et eksempel fra den benyttet SDK pakke. Dette eksempel indeholder funktionaliteterne til at oprette forbindelse en UART forbindelsen til en master enhed over BLE. På kode udsnittet nedenfor ses de standard funktionskald som benyttes under opsætningen af BLE forbindelsen på Device enheden.

```
1  // BLE initialize.
   ble_stack_init();
3  gap_params_init();
   services_init();
5  advertising_init();
   conn_params_init();
```

De fem funktioner fra kode udsnittet giver tilsammen alle nødvendige funktionaliteter til at benytte BLE på udviklingskittet. Årsagen til at der tages udgangspunkt i et eksempel fra Nordic er, at denne opsætning af BLE er udviklet specifikt til nRF51 kredsen og dennes softdevice. Opsætningen er udviklet så der er mulighed for at ændre på forskellige parametre som navn, UUID (Universally Unique Identifier) osv. Opsætningen er yderligere udviklet med hensyn til et lavt strømforbrug.

Den første funktion fra kode udsnittet *ble_stack_init()* har til opgave at opsætte softdevicen til systemet og sørge for at BLE er tilgængelig. Dernæst er der *gap_params_init()* som sørger for opsætningen af alle nødvendige parameter for BLE som bl.a. enhedens navn. Næste funktion *services_init()* opsætter de services som bliver brugt i forbindelse med softwaren. Her kan nævnes en funktion til håndteringen af de data der modtages fra enten BLE eller en computer over UART forbindelserne. Funktionen *advertsing_init()* sørger for at BLE forbindelsen kan advertise, altså gøre sig synlige overfor en masterenhed som scanner netværket for BLE enheder. Sidste funktion *conn_params_init()* sørger for opsætning af de nødvendige parameter og for at der kan oprettes en forbindelse til en master enhed.

De funktioner i softwaren som Nordic stiller til rådighed via deres eksempler, vil videre i projektet være benyttet som lukkede funktioner og vil ikke være yderligere beskrevet. Dog vil de funktioner hvor der ændres på standard opsætningen fra eksemplerne blive yderligere beskrevet.

Brug af BLE på Device Udviklingskit

BLE forbindelsen kan benyttes på samme måde som en UART forbindelse til at sende data til en anden enhed. Nedenfor er der vist et kode eksempel over funktionen, som benyttes til at sende data fra Device enheden. Som der ses er der mulighed for at ændre på tre parameter indstillinger for funktionen. Den første parameter er *&m_ble_uart_c* som er en pointer til en UART klient struktur, som er en af standard funktionerne fra Nordic. Næste paramter er *data_array* som er det data array der ønskes at sende fra Device enheden. Sidste parameter input er *index* som er længden på det data array som

ønskes at sende. Formatet på de data som skal sendes med denne funktion er af typen uint8_t.

```
ble_uart_c_write_string(&m_ble_uart_c, data_array, index);
```

4.5.3 Design af Kontakt Funktionaliteter

Styringen af en kontakt kræver ikke det helt store af systemet. Når først den pågældende pin på toolkittet er opsat som en udgang, kan den sættes høj eller lav. Der kan dog designes en simpel funktion, som kan modtage parameter for hvilken pin der er tale om på toolkittet, samt om det skal sættes højt eller lavt. Denne kan med fordel bruges på de forskellige udgang der benyttes på enheden. På kode udsnittet nedenfor kan denne funktion ses.

```
1  void switch_function(int pin, int state)
   {
3    if(state == 1)
     {
5      nrf_gpio_pin_set(pin);
     }
7    else
     {
9      nrf_gpio_pin_clear(pin);
     }
11 }
```

4.5.4 Design af Knap Funktionaliteter

Til funktionen kontaktstyring er det nødvendigt, at en af de digitale I/O porte er opsat som en indgang. Opsætningen af denne er udført som beskrevet i afsnit 4.3.2 omkring opsætning af digitale ind- og udgange. Da udviklingskittet til Device enheden ikke indeholder knapper, er der opsat en ekstern knap som en pull-up opsætning.

Designet af denne knap tryk funktionalitet kan ses på kode udsnittet nedenfor. Under designet af denne blok er der som en hjælp indført en funktion som tænder og slukker en LED når der trykkes på knappen. I det endelige system vil det blive afsende en kommando over BLE forbindelsen om at der er trykket på knappen.

```
1  /**** BUTTON PRESSED****/
   void button_pressed(int btn, int led)
3  {
     uint8_t btn_nstate;
5    uint8_t btn_ostate = nrf_gpio_pin_read(btn);

7    while(true)
     {
9      uint8_t btns = 0;
       btn_nstate = nrf_gpio_pin_read(btn);
11
       if((btn_ostate == 1) && (btn_nstate == 0) )
13     {
```

```
            if (ledOn == 0)
15          {
              /* Turn LED ON */
17            switch_function(led,1);
              ledOn = 1;
19          }
            else
21          {
              /* Turn LED OFF */
23            switch_function(led,0);
              ledOn = 0;
25          }
          }
27        btn_ostate = btn_nstate;
        }
29    }
```

4.5.5 Design af Sensor Funktionaliteter

Designet af sensor funktionaliteterne for Device toolkittet kan udføres som nævnt tidligere vha. den analoge indgang på nRF51822 kredsen. Opsætningen af systemet er beskrevet som en af de grundlæggende funktioner udviklet til Nordic SDK pakken. Det skal lige huskes at der til denne Device enhed benyttes et ældre udviklingskit, hvorfor der kan være lidt forskel på opsætningen og brugen af den analoge indgang på de to typer af udviklingskit.

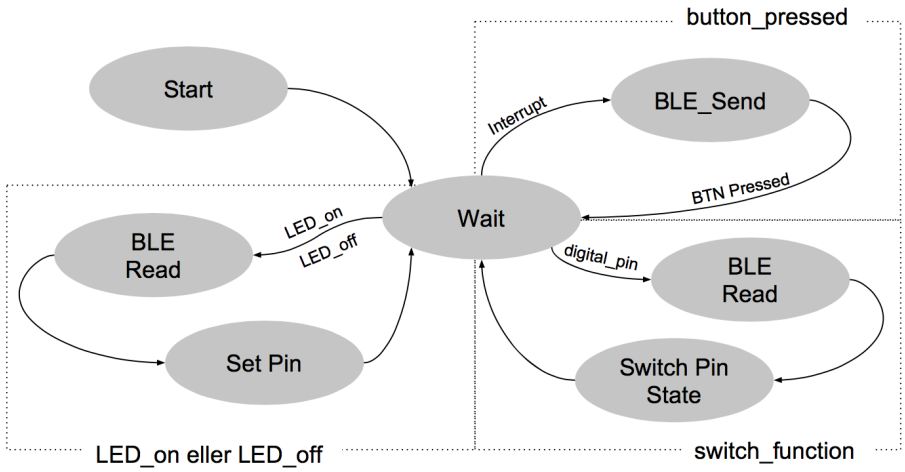
Udviklingskittet som benyttes til denne Device enhed, er ofte benyttet sammen med et motherboard som giver adgang til samtlige I/O porte på kredsen. Som tidligere beskrevet kan de forskellige I/O porte tilgås vha. to konnektorer på undersiden af udviklingskittet. Dette giver dog nogle problemer i forhold til adgangen til de analoge udgange, hvorfor denne sensor funktionalitet i stedet vil være udviklet på det Centrale toolkit i afsnit 4.6.5

4.5.6 Design af Logik funktionaliteter

Logik funktionaliteterne for Device toolkittet indeholder styringen af de funktioner, som ovenfor er udviklet til hver af de eksterne komponenter. Det er også Logik modulet funktion at sørge at behandle de kommandoer der via BLE forbindelsen modtages fra den Centrale enhed.

På den overordnede figur for Device enheden på figur 4.10 ses der også et modul ved navn kommandofortolker. Dette modul har sammen med Logik blokken til opgave at sørge for at de forskellige funktioner udføres korrekt.

For at give et overblik over den samlede funktionalitet der skal designes for Device enheden i dette projekt, er der på figur 4.11 opstillet et tilstandsdiagram. Dette diagram viser de forskellige tilstande der kræves til kontakt og knap funktionaliteterne. Der ses også på dette diagram at Device enheden er opbygget med udgangspunkt i *Triggered event*. Dette skyldes at alle styringsfunktioner er placeret på det Centrale toolkit, og at dette derfor blot skal udføre de kommandoer der modtages fra enten en bruger eller det andet toolkit.



Figur 4.11. Tilstandsdiagram over device funktionalitet

Som beskrevet i dette afsnit består Device enhedens funktionaliteter af enkle eksterne komponenter tilsluttet toolkittet. Derfor tager designet af logik blokken også udgangspunkt i simple funktionaliteter, for hver af de eksterne komponenter. Der skal designes enkle funktionskald, som vil være en del af modulet kommandofortolker. Det er dennes opgave at modtage kommandoer fra BLE forbindelsen og sikre at de rigtige funktioner på Device enheden kaldes. Den skal ligeledes designes, således at der i andre situationer kan sammenkoble flere funktioner for de eksterne komponenter. Det er et af kravene til systemet om muligheden for videreudvikling af toolkittet.

Read BLE Message

En vigtig funktion er at Toolkittet kan modtage BLE kommandoer fra den Centrale enhed. Til dette benyttes en funktion kaldet *nus_data_handler*, som er en standard BLE funktion, som i dette projekt er tilpasset dette ønskede formål. Til søgning efter kommandoer i de modtagne data, benyttes funktioner *strstr(...)*, som også kan beskrives som at finde en nål i en høstak. Denne funktioner fungere ved at man indtaster det ønskede søge ord eller kommando hvorefter det valgte data kontrolleres. På kode udsnittet nedenfor er der vist et eksempel på en sådan funktionalitet fra det benyttede software. Her ses det at første parameter er dataene som skal søges igennem og sidst er der søge ordet.

```
1  if(strstr((char*)(p_data), "digital_pin"))
   {
3    // function
   }
```

Send BLE Message

En anden funktionalitet som er nødvendig for at kravene overholdes, er at sende data over BLE. Da der i netværks afsnittet allerede er beskrevet hvordan en standard besked sendes over BLE, vil der i dette afsnit istedet designes en funktion, som kan modtage en parameter med de data som skal sendes over BLE. På denne måde kan funktionen bruges

forskellige steder i programmet, hvor det er nødvendigt at sende data til den Centrale enhed.

Der er nedenfor indsat et kode udsnit på en sådan funktioner. Her ses det på parameterene at beskeden som sendes skal være af typen *uint8_t* samt at længden af meddelelsen også skal indtastes.

```
/**** SEND DATA OVER BLE ****/
void ble_send(uint8_t * message, int length_)
{
  uint8_t * send = message;
  ble_nus_send_string(&m_nus, send, length_);
}
```

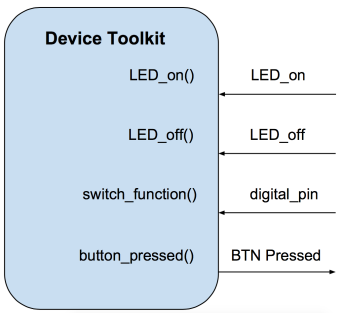
4.5.7 Opsummering af Device Toolkit

For at opsummer designet og implementeringen af Device toolkittet er der opstilles en tabel for at give et overblik. På tabellen indeholder første række de kommandoer som benyttes til kommunikation med den Centrale enhed. Dette er kommandoer som enten sendes eller modtages over BLE forbindelsen, og benytter de nævnte funktioner i anden række på tabellen.

Kommandoer	Funktioner på Device enhed	Beskrivelse
LED_on	LED_on()	Sætter LED pin aktiv
LED_off	LED_off()	Sætter LED pin passiv
digital_pin	switch_function()	Skifte LED pin til modsatte tilstand
BTN Pressed	button_pressed()	Sender besked til den centrale enhed om at der er trykket på knappen

Tabel 4.4.

Som sidste del af design fasen for Device enheden er der på figur 4.12 indsat en illustration over de ovennævnte kommandoer og funktioner.

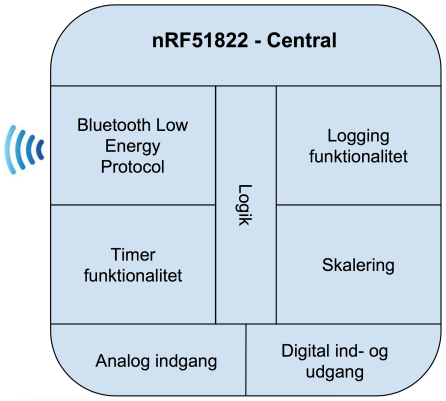


Figur 4.12. Illustration over Device toolkit

4.6 Central Toolkit

Designet og implementeringen af den Centrale enhed, tager udgangspunkt i udviklingskittet nRF51-DK. Dette udviklingskit benytter som tidligere nævnt SDK pakken 10.0, hvilket betyder at funktionskaldende udviklet af Nordic Semiconductor vil være anderledes end dem benyttet gennem designet af Device enheden.

Den Centrale enheds overordnede opgave er at kontrollere de forbundne Device enheder. Hertil er der gennem kravspecifikationen opstillet funktioner og funktionaliteter som er nødvendige for at netop dette er opfyldt. For at nævne disse funktionaliteter er der først BLE forbindelsen som sikre at der kan oprettes forbindelse til de øvrige Device toolkit. Dernæst er der en logging funktion som skal benyttes til at gemme dataene på systemet. Der er en skalering som sikre at de måle data som modtages over BLE forbindelsen, kan konverteres til de rigtige enheder. Der er en timer funktion som benyttes til styringen af hvornår de forskellige funktioner skal udføres. For at forbindelse de forskellige funktionaliteter skal en logik modul designes. Der er på figur 4.13 opstillet en illustration som viser disse funktionaliteter for det Centrale toolkit.



Figur 4.13. Figur over central systemet

kommunikationsmodulet på denne enhed er som allerede nævnt en BLE forbindelse opsat med UART funktionaliteter.

4.6.1 Opsætning af Standard funktionaliteter

Ligesom med Device enheden er der en række standard funktionaliteter som benyttes på dette toolkit. På kodeudsnittet herunder ses de fire opsætnings funktionskald.

```
leds_init();      // LED/Digital output initialising
buttons_init();   // Button/Digital input initialising
adc_init();       // Analog input initialising
uart_init();      // UART initialising
```

4.6.2 Opsætning af BLE på det Centrale Udviklingskit

BLE funktionaliteterne for den Centrale enhed er opbygget omkring softdevice S120 eller S130 fra Nordics SDK pakke. Årsagen til at begge softdevices er nævnt er at de begge

understøtter de funktionaliteter der kræves for den Centrale enhed. Det forventes at Nordic på længere sigt vil udfase softdevicerne S110 og S120 for blot at stille S130 til rådighed for BLE kommunikationen.

Opsætningen af den centrale enhed er lidt anderledes end den er for Device enheden. Dette skyldes, at Device enheden har til opgave at advertise, altså gøre sig selv synlig for andre enheder. Herefter er det op til den Centrale enhed at oprette forbindelsen til Device enheden. På kode udsnittet nedenfor ses de funktionskald som giver opsætningen af BLE funktionaliteterne for det benyttede udviklingskit.

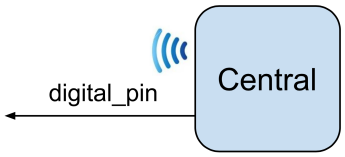
```
// BLE Central Initialising
ble_stack_init();
ble_db_discovery_init();
scan_start();
```

Som vist på kode udsnittet ovenfor opsættes de forskellige BLE funktionaliteter for det benyttet udviklingskit. Som det sidste startes en scanning for at se om der er andre BLE enheder i nærheden. Hvis dette er tilfældes vil den Centrale enhed forsøge at oprette forbindelse til denne. Dette gøre ud fra ud fra nogle forudbestemte parameter bl.a. UUID (Universally Unique Identifier) som er en identifikationen for Device enheden. Når der er oprettet forbindelse imellem en Central og en Device, kan den benyttes ligesom en almindelig UART forbindelse.

For at afsende data fra den centrale enhed sker på samme måde som beskrevet under Device enheden. Dette vil derfor ikke blive yderligere beskrevet.

4.6.3 Design af Kontakt Styring

Funktionen kontakt styring er en simpel funktion, som overordnet set blot skal afsende en kommando til Device enheden om at sætte en digital udgang høj eller lav. Det forventes at opsætningen af den digitale udgang er udført på Device enheden. Som en illustration over denne funktionalitet er dette indsat på figur 4.14. Her ses det hvordan det er påkrævet at den Centrale enhed at sende kommandoen *digital_pin* til Device enheden for at aktiver eller deaktiver den digital udgang.



Figur 4.14. Kontakt funktion

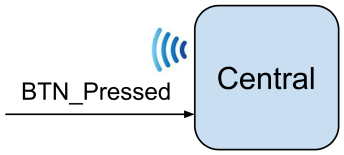
Designet af denne kontakt softwaren kan ses på kode udsnittet nedenfor. Her ses det hvordan en streng af formatet uint8_t oprettes med kommandoen *digital_pin*. Denne sendes efterfølgende over BLE forbindelsen til Device enheden.

```
// Function for the Switch functionality
void switch_control()
{
  uint8_t data1[] = "digital_pin";
```

```
    int length1 = sizeof(data1);
    while (ble_nus_c_string_send(&m_ble_nus_c, data1, length1) != NRF_SUCCESS)
    {
        // repeat until sent
    }
}
```

4.6.4 Design af Knap Input

Funktionen for knap funktionaliteten er stort set modsat den for kontakt styringen. Her skal der modtages en kommando fra Device enheden om at der er trykket på knappen. Herefter skal det Centrale toolkit udføre den valgte kommando. På figur 4.15 er der opstillet en figur som viser den overordnede funktion, hvor kommandoen *BTN_Pressed* modtages.



Figur 4.15. Knap funktion

Designet af softwaren for modtagelse af kommandoer fra Device enheden, er en del af funktionaliteten for en data handler. Denne vil være beskrevet som en del af designet af logik funktionaliteten. Der vil dog være indsat et kode udsnit herunder, som viser søgningen af kommandoen, samt et funktionskald til styring af en kontakt.

```
// Function for reading Button Press
if(strstr((char*)(p_data), "BTN_Pressed"))
{
  switch_control();    // Function for switch functionality
}
```

4.6.5 Design af Sensor Funktioner

Opsætningen af sensor og analoge funktionaliteter for toolkittet er tiltænkt videreudvikling, hvor det evt. kan overvåge en batterispænding. Dog vil der i dette projekt være tilsluttet en temperatur sensor, som benyttes i forbindelse med kravet om at der skal udvikles en varmekontrol. Denne temperatur sensor er tilsluttet en af de analoge indgange på udviklingskittet og opsætningen af denne indgang er beskrevet tidligere i afsnit 4.3.1.

Til måling af temperatur benyttes en sensor ved navn LM35. Dette er en sensor som kan benyttes uden en kalibrering og har en præcision på $10\frac{mV}{°C}$. Hvis dette afbilledes på en graf vil udgangsspændingen være vist som en lineær kurve, hvor 10 °C vil svare til en udgangsspænding på 100 mV. Da det er forventet, at der kan måles temperaturer imellem 0 - 50 °C vil dette give en maksimal udgangsspænding på 500 mV ved 50 °C.

Ud fra valget af sensor kan paramterne for den analoge indgang vælges. Som reference spænding benyttes de standard 1,2 V, dog med en prescaling på $\frac{1}{1}$ i forhold til

indgangsspændingen. Dette vil sige at den maksimale indgangsspænding for den analoge indgang vil være 1,2 V. Grunden til at der vælges en lav maksimal grænse, er at den ligger tættere på den maksimale udgangsspænding fra temperatur sensoren. Det er dog nødvendigt at forstærke udgangssignalet fra temperatur sensoren, så den ved 50 °C levere 1,2 V på den analoge indgang.

Forstærkningen af temperatursensoren udføres ved, at indsætte en ikke inverterende operationsforstærker mellem sensor udgang og analog indgang. Forstærkningen kan udregnes til $\frac{1200mV}{500mV} \approx 2,4$ gange. Opsætningen af den samlede sensor og operationsforstærker kan ses som appendiks i afsnit A.1.

Under opsætningen af den analoge indgange kan opløsningen for ADC'en også vælges. Her kan den som tidligere nævnt, vælges til en opløsning på enten 8, 9 eller 10 bit. Der vælges dog til dette projekt en opløsning på 8 bit som giver en præcision på $\frac{50}{2^8} \approx 0,2$ °C. Denne præcision er tilstrækkelig til et projekt som dette, hvor det blot er en varmekilde der skal kontrolleres.

Aflæsningen af sensor dataene er implementeret som en del af standard opsætningen for den analoge indgang. Her benyttes en ADC interrupt handler til at aflæse værdien og gemme den i en streng kaldet *adc_ sample*. Dette kan ses i afsnittet om opsætningen af den analoge indgang i afsnit 4.3.1.

Da det i et endeligt system er tiltænkt at denne temperatur måling skal foregå på en Device enhed, vil det være nødvendigt at sende disse data over BLE til den Centrale enhed. Dette kan med fordel gøres ved at benytte en funktion som den indsat nedenfor, hvor dataene *adc_ sample* afsendes over BLE.

```
/*** Send sensor data **
*/
void send_data()
{
  uint8_t str[4];
  sprintf((char*)str, "%d", (int)adc_sample);
  ble_nus_send_string(&m_nus, str, strlen((char*)str));  // Send data with BLE
}
```

Nu hvor opsætningen af den analoge indgang og sensor kredsen er designet, kan funktionen for skalering af dataene designes.

4.6.6 Design af Skalerings Funktionalitet

Dataene som aflæses på den analoge indgang eller som modtages fra en Device enhed, kræver en skalering inden videre brug i systemet. For dette projekt skal udgangsspænding omregnes til en temperaur i grader celsius. Der er i funktionsbeskrivelserne i afsnit 2.3 vist en formel, som med fordel kan benyttes til designet af denne funktionalitet. På formlen nedenfor kan denne udregning også ses.

$$°\text{C} = U_{in} \cdot \frac{T_o}{a} \tag{4.1}$$

Værdierne for T_o og a kan findes ud fra opsætningen af den analoge indgang og sensoren fra foregående afsnit. Udregningen af T_o tager udgangspunkt i den valgte reference spænding samt opløsningen fra ADC samplingen. Der er her valgt en reference spænding på 1,2 V og en opløsning på 8 bit. Udregningen af T_o værdien kan ses nedenfor på formel 4.2.

$$T_o = \frac{V_{ref}}{oplsning} = \frac{1200mV}{2^8} \approx 4,69\frac{mV}{bit} \tag{4.2}$$

Udregningen af a tager udgangspunkt i skaleringen fra sensoren og operationsforstærkeren. Hvor sensoren har en specifikation på 10 $\frac{mV}{°C}$ og operationsforstærkeren har en forstærkning på 2,4 gange. Udregningen af denne faktor kan ses nedenfor på formel 4.3.

$$a = 10\frac{mV}{°C} \cdot 2,4 = 24\frac{mV}{°C} \tag{4.3}$$

Da faktorerne T_o og a skal kunne ændres i tilfælde hvor der benyttes andre sensorer, er de i softwaren defineret øvest. På denne måde er det let at korrigere eventuelle ændringer på denne skalering. Dette kan også ses på kodeudsnittet nedenfor.

```
#define ADC_SCALING_FACTOR   4.69      // Scaling factor Vref/ADC resolution (1,2 V/ 2^8)
#define SENSOR_FACTOR         24       // Sensor scaling * OPAMP (10 mV * 2,4)
```

Implementeringen af skalerings funktionen i softwaren, kan ses på kodeudsnittet nedenfor. Der ses hvordan formlen er implementer og den skalerede temperatur gemmes som *adc_sample_scaled* til videre brug for designet af varmekontrollen.

```
  // Scale ADC value
static void adc_scale(void)
{
  adc_sample_scaled = adc_sample * ADC_SCALING_FACTOR / SENSOR_FACTOR;
}
```

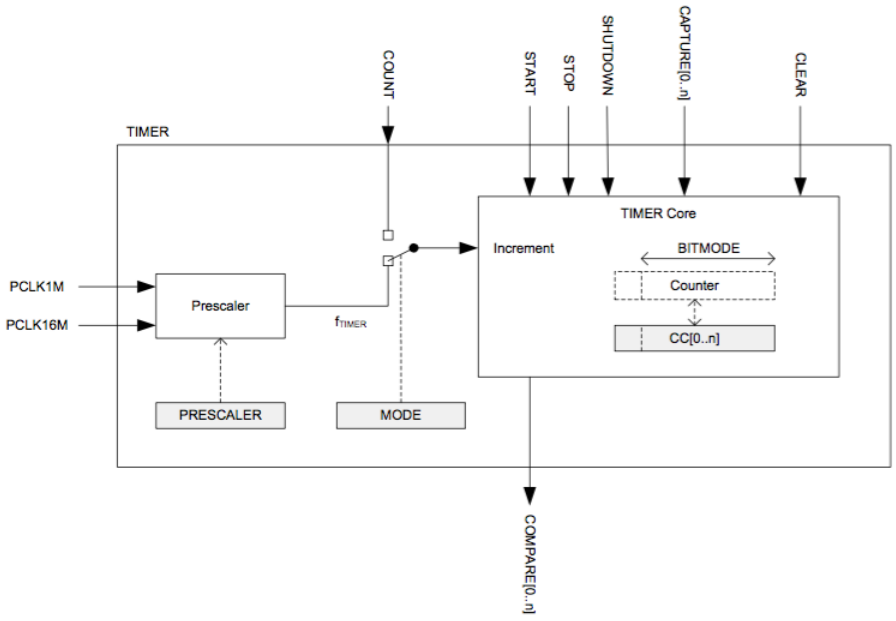
Et af kravene til det samlede toolkit er muligheden for videreudvikling. Der ses i dette afsnit at der kan ændres på de benyttede skaleringsfaktorer. Det er også muligt at tilføje nye faktorer for derefter at kunne tilføje endnu en sensor som kan benyttes sammen med temperatur sensoren.

4.6.7 Timer Funktionaliteter

Der er til systemet opstillet et krav om at styringen af toolkittet skal kunne udføres ud fra Timed event. For at dette kan udføres er det derfor nødvendigt at implementere timer funktionaliteter. For udviklingskittet er der flere muligheder for at implementerer dette, hvor den ene metoder er at bruge den indbyggede 32kHz krystal. Fordelen ved at benytte en så lav frekvens er, at den er udviklet med henblik på at et meget lavt strømforbrug,

hvorfor Nordic også betegner den som en ultra lav energi timer. Det er dog ikke muligt at benytte denne timer til bruger applikationer, hvis der benyttes en softdevice i systemet.

En anden mulighed er at benytte Timer funktionen vist på figur 4.16. Denne timer benytter en 16 MHz clock frekvens, som kan ned skaleres vha. en indbyggede prescaler. På figuren ses det at de to bokse *Prescaler* og *Mode* er grålige, hvilket betyder at der her er mulighed for at tilpasse timeren efter behov. Prescaler kan benyttes som benyttes til at nedsætte frekvensen, imens mode kan benyttes til bestemme om timeren skal være i timer eller count mode. Forskellen på disse er at timer mode benytter den førnævnte 16 MHz clock frekvens, imens der for Count mode skal implementeres en ekstern trigger funktion. [14]



Figur 4.16. Opbygningen af timer funktion [14, p. 99]

På figur 4.16 ses også de funktioner som kan benyttes til styringen af blokken kaldet *Timer Core*. Denne blok indeholder de grundlæggende funktioner for timeren og her ses også blokken *Counter*. Det er denne som vha. en sammenlignings funktioner benyttes som trigger, til de ønskede funktionkald.

For at benytte timeren i Time mode, skal en Prescale værdi først vælges. Der er her mulighed for, at vælge en værdi mellem 0 og 9. Udregning af frekvensen f_Timer kan ses på formel 4.4, hvor en Prescaler værdi på 9 er valgt. Dette giver en frekvens på 31,25 KHz, som er tilstrækkelig for dette projekt, da der ikke er tidskritiske funktioner i designet af varmekontrollen.

$$f_{Timer} = \frac{16MHz}{2^{Prescaler}} = \frac{16MHz}{2^9} = 31250Hz \tag{4.4}$$

Opsætningen og brugen af denne timer funktion, kan ses på kodeudsnittet nedenfor. Her ses det på linje 4 og 6 hvordan der først vælges Timer mode og en Prescaler værdi på 9. Af

andre vigtige parameter for timeren kan der på linje 8 ses, hvilken værdi der benytte som sammenlignings grundlag. Her vælges det at hver gang counteren når til 31250, udløses en event vha. en *TIMER_IRQHandler*. Der er i denne interrupt handler implementeret en funktion som hver 10. gang trigger en skalering af sensor dataene. Dette vil sige at der for hver 10 sekund kaldes funktionen *adc_scale*.

```
1  /** TIMER2 setup */
   void start_timer(void)
3  {
   NRF_TIMER2->MODE = TIMER_MODE_MODE_Timer;
5  NRF_TIMER2->TASKS_CLEAR = 1;
     NRF_TIMER2->PRESCALER = 9;
7    NRF_TIMER2->BITMODE = TIMER_BITMODE_BITMODE_16Bit;
     NRF_TIMER2->CC[0] = 31250;
9
   // Enable interrupt on Timer 2, for CC[0] and compare match events
11 NRF_TIMER2->INTENSET = (TIMER_INTENSET_COMPARE0_Enabled <<
       TIMER_INTENSET_COMPARE0_Pos);
   NVIC_EnableIRQ(TIMER2_IRQn);
13
   NRF_TIMER2->TASKS_START = 1;    // Start TIMER2
15 }

17
   /** TIMER2 peripheral interrupt handler. This interrupt handler is called whenever there is a
       TIMER2 interrupt
19  */
   void TIMER2_IRQHandler(void)
21 {
   if ((NRF_TIMER2->EVENTS_COMPARE[0] != 0) && ((NRF_TIMER2->INTENSET &
       TIMER_INTENSET_COMPARE0_Msk) != 0))
23 {
     NRF_TIMER2->EVENTS_COMPARE[0] = 0;  //Clear compare register 1 event
25   NRF_TIMER2->TASKS_CLEAR = 1;          //Reset counter
     if(timer_event == 10)
27   {
       adc_scale();
29     timer_event = 0;
     }
31   else
     {
33     timer_event++;
     }
35 }
   }
```

Som en videreudvikling af timer funktionaliteterne kan det udbygges med en ur funktionalitet. Dette giver mulighed for at udvikle funktioner som ikke blot måler temperaturen indenfor et bestem interval, men som også kan skelne mellem de forskellige tidspunkter på døgnet. Man kan på denne måde indføre funktioner som at temperaturen i rummet er lavere om natten end om dagen.

Videre igen kan timer funktionaliteterne udvides med funktioner som kalender, hvor der kan skelnes imellem hverdag og weekend. Det kan være, at man ønsker en lavere

temperatur i formiddagstimerne på hverdage hvor man er på arbejde, end i weekenderne hvor man har fri.

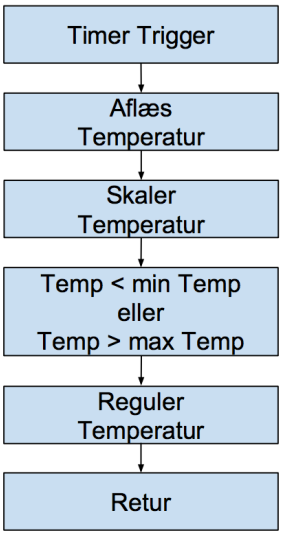
System Tilstande

En anden metode at benytte timer, ur og kalender funktioner er ved at udvikle tilstande for systemet. Tilstande kan benyttes som en samling af funktioner som er opsat til bestemte værdier. Et eksempel på en tilstand for toolkittet kan være for varmekontrollen, hvor der oprettes en tilstand for dag og nat. For dagen er temperaruren valgt til 22 °C imens den om natten er valgt til 19 °C. Tilstande behøver dog ikke at være tilknyttet timer funktionaliteterne, det kan også være muligt at benytte dem i forbindelse med event udløst af et brugerinput.

4.6.8 Design af Varmekontrol

Designet af en varmekontrol til toolkittet er et af de to scenarier beskrevet i kravspecifikationen i kapitel 3. Designet heraf tager udgangspunkt i de øvrige funktioner beskrevet gennem dette afsnit omkring designet af den Centrale enhed. For at opsummere varmekontrollen skal den overvåge en temperatur og ud fra to definerede grænseværdier bestemme om varmekilden skal tændes eller slukkes.

For at vise hvordan de forskellige funktioner samarbejder omkring varmekontrollen, er der på figur 4.17 opstillet et diagram over dette. Der ses her hvordan funktionen kaldes vha. en trigger fra Timeren, hvorefter temperaturen aflæses og skaleres. Herefter sammenlignes den skalerede temperatur med min og max grænsen for temperatur hvorefter det vurderes om der skal reguleres vha. varmekilden. Herefter afsluttes funktionen, indtil timeren igen trigger et event.



Figur 4.17. Diagram over varmekontrollen

Designet af de fire første funktioner på digrammet er allerede udviklet og implementeret i softwaren. Der mangler dog blokken regulering af temperaturen. I dette modul skal den skalerede temperatur sammenlignes med de to grænseværdier. Fordelen ved at implementer

en nedre og en øvre grænse er at det er muligt at indstillet et interval for varmekontrollen. Der indføres her en funktionalitet hvor der eksempelvis kan tændes for varmekilden ved en temperatur på under 22 °C og først slukkes når temperaturen er over 24 °C. Ønskes et lavere interval og derved en højere frekvens for om varmekilden er tændt eller slukkes, vælges der to grænseværdier tættere på hinanden.

Der er nedenfor indsat et kode udsnit som viser varmekontrollen. Der ses her hvordan de to grænseværdier *MIN_ TEMP* og *MAX_ TEMP* er defineret som faste værdier. De to værdier er defineret øverst i koden, så værdierne let kan tilpasses de ønskede behov. Videre vil det være en mulighed at tilpasse funktionen, så de to grænseværdier let kan ændres af eventuelle bruger af toolkittet.

For at tænde og slukke for varmekilden, som i dette projekt er vist som en rød LED, sendes der kommandoer til Devicen enheden over BLE forbindelsen. Der er i forbindelse med dette, implementeret en *heat_ state* værdi som viser om varmekilden allerede er tændt eller slukket. Dette er indført for at begrænse de dataer der sendes over BLE forbindelse.

```
  /** Heating control */
2 static void heat_control(void)
{
4   if (adc_sample_scaled < MIN_TEMP)
      {
6       if (heat_state == 0)
        {
8         uint8_t data1[] = "digital_set";     // Turn heat ON
          int length1 = sizeof(data1);
10        while (ble_nus_c_string_send(&m_ble_nus_c, data1, length1) != NRF_SUCCESS)
          {}
12        heat_state = 1;
        }
14    }
  else if (adc_sample_scaled > MAX_TEMP)
16    {
        if (heat_state == 1)
18      {
          uint8_t data1[] = "digital_clear";  // Turn heat OFF
20        int length1 = sizeof(data1);
          while (ble_nus_c_string_send(&m_ble_nus_c, data1, length1) != NRF_SUCCESS)
22        {}
          heat_state = 0;
24      }
      }
26  if (DEV_MODE == 1)
  {
28    printf("Temperature is: %.2f\r\n", adc_sample_scaled);
  }
30}
```

4.6.9 Logging Funktionalitet

En anden funktionalitet der er opstillet i kravspecifikationen er logging af data. Denne funktioner er ikke en del af de to scenarier der er beskrevet, og er primært tiltænkt

videreudvikling af toolkittet. Det er den indbyggede flash hukommelse på nRF51822 kredsen som benyttes når data skal gemmes på enheden. Der vil i dette projekt ikke være implementeret logging funktionaliteter på toolkittet, hvorfor det ikke vil være yderligere beskrevet.

4.6.10 Design af Logik Funktionalitet for Central Toolkit

Nu hvor de enkelte del moduler af den Centrale enhed er designet kan de sidste logik funktionaliteterne udvikles. Da de fleste funktioner allerede er opsat til automatisk at starte, enten vha. timer eller triggered event, vil der i dette afsnit være en kort beskrivelse af at sætte systemet i udviklingsmode.

Udviklingsmode er en funktion hvor man kan til og fra koble UART funktionaliteter. Funktionen er implementeret til at man øverst i koden kan sætte en værdi til 1 eller 0, alt efter om det skal opsætte som udviklingsmode eller produktionsmode. Et eksempel på brugen af funktionen kan ses i kode udsnittet fra varmekontrollen, hvor den skalerede temperatur udskrives på terminalen på en computer.

```
if (DEV_MODE == 1)
{
  printf("Temperature is: %.2f\r\n", adc_sample_scaled);
}
```

Det er også muligt at benytte denne UART forbindelse til debugging fra Device enheden. Her kan man sende de ønskede data over BLE forbindelsen, for efterfølgende at udskrive disse på samme måde som kode udsnittet ovenfor.

4.7 Opsummering af Design og Implementering

Der er gennem dette afsnit designet og implementeret kontaktstyrings og varmekontrols funktionaliteter på det samlede toolkit. Der benyttes i disse scenarier funktioner, som gør det muligt at benytte det samlede system til videreudvikling af kundespecifikke projekter. Systemet vil i næste kapitel blive testet op imod de krav, som er opstillet i kravspecifikationen.

Accepttest 5

Der vil i dette kapitel være opstillet en accepttest for det samlede toolkit. Formålet med denne test er at undersøge om de udviklede toolkit opfylder de krav og scenarier, der er beskrevet i kravspecifikationen i kapitel 3. Opbygningen af accepttesten vil tage udgangspunkt i de tabeller fra kravspecifikationen, hvorfor også nummeringen af de enkelte krav vil gå igen.

5.1 Accepttest for de Overordnede Krav

Den første del af accepttesten består de overordnede krav til det samlede toolkit. Dette er funktioner som var en del af udgangspunktet for projektet.

Nr.	Parameter	Enhed	Forventet Resultat	Resultat
1.1	Hardware	Device og Central	Forventet at det samlede toolkit er opbygget omkring nRF51822 kredsen fra Nordic	Opfyldt
1.2	Hardware	Device og Central	Det samlede toolkit skal bestå af en device og en master central	Opfyldt
1.3	Kommunikation	Device og Central	Der skal være trådløs kommunikation mellem de to toolkit	Opfyldt
1.4	Digital I/O	Device og Central	Der skal være mulighed for at benytte digitale ind- og udgange på begge toolkit	Opfyldt
1.5	ADC	Device og Central	Det skal være muligt at benytte de analoge indgange op begge toolkit	Delvist opfyldt
1.6	Logging	Central	Det skal være muligt at logge måle data på den centrale enhed	Ikke opfyldt

Tabel 5.1. Accepttest for de overordnede krav

Som resultaterne for testen viser opfylder toolkittet ikke alle de opstillede krav. Dette skyldes bl.a. problemer med tilgange til de analoge indgange på udviklingskittet benytte

som Device enhed. Ligeledes er logging funktionaliteten ikke implementeret på systemet, hvorfor den heller ikke er opfyldt.

5.2 Accepttest for de Tekniske Krav

Den næste del af accepttesten tager udgangspunkt i de tekniske krav som er opstillet for toolkittet.

Nr.	Parameter	Enhed	Forventet Resultat	Resultat
2.1	Respons tid	Samlet Toolkit	Skal registrer et knap tryk og tænde en LED indenfor 200 ms	Ikke testet
2.2	Rækkevidde	Samlet Toolkit	Der skal være en rækkevidde på min 10 meter mellem de to enheder	Opfyldt
2.3	Timed event	Central	Toolkittet skal indeholde timer funktionalitet for tidsstyrede event	Opfyldt
2.4	Triggered event	Device og Central	Toolkittet skal indeholde funktionaliteterne for triggered event	Opfyldt
2.5	Digital I/O	Device og Central	Der skal være mulighed for at benytte digital ind- og udgange	Opfyldt
2.6	ADC	Device og Central	Der skal være mulighed for at benytte de analoge indgange	Delvist opfyldt
2.7	Flash hukommelse	Central	Det skal være muligt at logge data vha. Flash hukommelsen på nRF51822 kredsen	Ikke opfyldt
2.8	2,4 GHz kommunikation	Device og Central	Det skal være muligt at kommuniker trådløst vha. den indbyggede 2,4 GHz kommunikation	Opfyldt
2.9	Batteri	Device og Central	Det skal være muligt at forsyne de to toolkit fra en batterispænding	Delvist opfyldt

Tabel 5.2. Accepttest for de tekniske krav

Der ses her at respons tiden for systemet ikke er testen, hvorfor det ikke vides om denne er opfyldt. Systemet opfylder dog de fleste krav, igen på nær den analoge indgange på Device enheden og logging funktionaliteterne. Kravet omkring en batteri forsyning er delvis opfyldt. Dette skyldes at der kun er testet på udviklingskittet til den centrale enhed og ikke på Device enheden. Det dog muligt at forsyne denne med en batterispænding, blot den er mellem 1,8 og 3,6 V.

5.3 Accepttest for Kontaktstyring Scenariet

De to sidste dele af acceptesten tager udgangspunkt i de opstillede scenarie beskrivelser, som det er forventet, at systemet kan opfylde. På tabellen nedenfor er scenariet opdelt i mindre dele, som hver enkelt vil blive teste. Det er dog forventer, at alle punkter opfylder kravene, for at man kan sige at kontaktstyringen virker efter hensigten.

Nr.	Test beskrivelse	Resultat
3.1	Systemerne tændes og der oprettes en trådløs forbindelse mellem Toolkit 1 og Toolkit 2	Opfyldt
3.2	Input fra kontakt aflæses og der afsendes en kommando til Toolkit 2	Opfyldt
3.3	Toolkit 2 modtager kommando og sender en ny kommando til Toolkit 1	Opfyldt
3.4	Toolkit 1 modtager kommando og tænder for lyskilde	Opfyldt

Tabel 5.3. Accepttest for kontaktstyring scenariet

Resultaterne viser, at det samlede toolkit opfylder alle kravene omkring kontaktstyring scenariet.

5.4 Accepttest for Varmekontrol Scenariet

Den sidste test del af accepttesten er scenariet for varmekontroller. Her skal systemet overvåge en temperatur, og ud fra resultatet enten tænde eller slukke for en varmekilde. Testen af dette scenarie vil dog på forhånd ikke kunne opfyldes helt, da der ikke er implementeret en sensor på Device enheden. Dette skyldtes problemer med den analoge indgang på udviklingskittet. Der er derfor i stedet tilsluttet en sensor til den analoge indgang på den centrale enhed, hvorfor værdien aflæses direkte fra denne enhed.

Nr.	Test beskrivelse	Resultat
4.1	Systemerne tændes og der oprettes en trådløs forbindelse mellem Toolkit 1 og Toolkit 2	Opfyldt
4.2	Der afsendes en kommando om at aflæser sensor værdi fra Toolkit 2	Delvist Opfyldt
4.3	Toolkit 1 modtager kommando og aflæser sensor værdi	Delvist opfyldt
4.4	Toolkit 1 afsender sensor værdi til Toolkit 2	Delvist opfyldt
4.5	Toolkit 2 modtager sensor værdi og laver en skalering til grader celsius	Opfyldt
4.6	Toolkit 2 vurdere ud fra den skalerede temperatur om der skal slukkes elller tændes for varmekilden og sender en kommando til Toolkit 1	Opfyldt
4.7	Toolkit 1 modtager kommando og slukker for varmekilden	Opfyldt

Tabel 5.4. Accepttest for varmekontrol scenariet

Det kan dog ses på tabellen at det samlede toolkit opfylder kravet om at kontrollere en varmekilde ud fra den målte temperatur. Dog er temperaruren aflæst vha. den analoge indgang på den Centrale toolkit istedet for på Device enheden.

Konklusion 6

Ideen til projektet var at udvikle et IoT Smart Home toolkit, med udgangspunkt i nRF51822 kredsen fra Nordic Semiconductor. Der var tale om et system med mulighed for individuel tilpasning til forskellige brugsscenarier. Gennem udviklingen i dette projekt blev to brugsscenarier opstillet, kontaktstyring og en varmekontrol. Disse scenarier var en del af kravene til det endelige system, og var udgangspunktet for designet af toolkit funktionaliteterne gennem rapporten.

Det endelige toolkit er opbygget ud fra ideen om at integrere de funktionaliteter, det forventes at nutidens IoT og Smart Home produkter skal indeholde. Som beskrevet i afsnittet omkring introduktionen til IoT Smart Home, består mange af de eksisterende produkter af simple funktioner såsom at tænde en kontakt, aflæse sensor værdier eller reagere på et brugerinput. Dette er alle funktioner, som er en del af designet for toolkittet beskrevet i dette projekt.

Det endelige system er designet vha. to udviklingskit som er opsat som henholdsvis device og master. Der er benyttet to forskellige udviklingskit fra Nordic, der begge er kompatible med nRF51822 kredsen. De indeholder dog forskellige versioner af nRF51822 kredsen, hvorfor de er udviklet med forskellige software versioner. Dette har gennem udviklingen givet lidt problemer, da opsætningen og brugen af de forskellige funktioner har ændret sig på de to udgaver. Af denne årsag er sensor funktionaliteterne ikke implementeret på device enheden, men i stedet opsat på den centrale enhed, som understøtter de nyeste versioner af Nordic softwaren.

Et af kravene til systemet var at implementere en logging funktionalitet. Dette er dog ikke blevet implementeret gennem projektet, men vil være en vigtig funktion for videreudviklingen af toolkittet.

Det endelige system skulle bestå af to enheder, hvor den ene skulle fungere som den centrale enhed og indeholde alle styringsfunktionerne. Den anden enhed skulle være en device enhed og indeholde tilslutninger af eksterne komponenter, såsom LED, knapper og sensorer. Device enheden opfyldte desværre ikke alle kravene om tilslutningen af en sensor, da det benyttede udviklingskit gav problemer med adgangen til de analoge porte. I stedet blev sensor funktionaliteterne flyttet over på de analoge indgange på udviklingskittet, som benyttes til den centrale enhed.

Perspektivering 7

Man kan sige at et IoT Smart Home Toolkit, som er blevet udviklet i dette projekt, aldrig vil være helt færdig udviklet. Dette skyldes at der altid kan implementeres nye funktioner eller nye brugsscenarier. Der vil derfor i dette afsnit være beskrevet forskellige videreudviklingsmuligheder for systemet.

- Et af de første punkter for en videreudvikling af dette system, vil være at anskaffe endnu et udviklingskit af modellen nRF51-DK. Dette vil give mulighed for at udvikle systemet med udgangspunkt i samme SDK pakke og derved udnytte de samme funktioner på begge udviklingskit. Det ville også være muligt at designe én kode, som indeholder både device og central funktionaliteterne vha. S130 softdevicen. På denne måde har man kun ét program som skal udvikles, tilpasses og vedligeholdes.

- Systemet kunne med fordel designes til at have et lavt strømforbrug, hvor der er mulighed for at slukke for sensorerne, når de ikke bruges. En anden strømbesparende faktor kunne være, at der kun oprettes forbindelse til den centrale enhed når der skal afsendes data.

- Man kunne udvide systemet med integration af enten en smartphone applikation eller knapper og et display til opsætning og styring af de forskellige funktioner på systemet.

- Et af de vigtige punkter for videreudviklingen af systemet er muligheden for let at integrere nye komponenter. Her tænkes især på sensorer og skaleringen af disse måledata. Det kan også være en fordel at implementere PWM funktioner, som åbner op for helt nye komponent tilslutninger.

- Man kan udvide de eksisterende ADC funktioner til at kunne overvåge batterispændingen for systemet, for på den måde at kunne advare brugerne ved lavt batteriniveau.

- En sidste videreudvikling er selvfølgelig integrationen af Homekit løsningen fra Apple. Dette ville åbne op for et helt nyt marked for et toolkit som dette, da antallet af Iphone og Ipad brugere er meget stort i Danmark.

Bilagsoversigt 8

ALLE BILAG ER AT FINDE PÅ VEDLAGTE CD

Datablade

- OPAMP - TLC071C.pdf
- Temperatur sensor - LM35.pdf

Litteratur

- nRF51 Series Compatibility Matrix v2.1.pdf
- nRF51 Series Reference manual v3.0.pdf
- nRF51-DK - User Guide v1_0.pdf
- nRF51822 Development Kit User Guide v1.4.pdf
- nRF51822 Product Specification v3.1.pdf
- PCA10004 Schematic And PCB.pdf
- PCA10005 Schematic And PCB.pdf
- PCA10028 Schematic And PCB.pdf
- PROD BRIEF nRF51-DK v1.0.pdf
- PROD BRIEF nRF51822 v2.3.pdf
- S110 SoftDevice Specification v1.3A.pdf
- S120 SoftDevice Specification v0.5.pdf
- S130 SoftDevice Specifikation v1.0.pdf

Rapport

- Toolkit for IoT Smart Home.pdf

SDK Pakker

- nRF51_SDK_10
- nrf51_sdk_v6_0_0_43681

Software

- ble_device
- ble_master
- Info

Litteratur

[1] Apple HomeKit, HomeKit,
http://www.apple.com/ios/homekit/,
Apple, 2015.

[2] Bluetooth, Low Energy,
https://www.bluetooth.com/what-is-bluetooth-technology/
bluetooth-technology-basics/low-energy,
Bluetooth SIG, 2015.

[3] Internet of Things (IoT), Internet of Things (IoT),
http://whatis.techtarget.com/definition/Internet-of-Things,
Margaret Rouse, Juni 2014.

[4] Internet of Things, Internet of Things: Derfor vil alt ændre sig,
http://www.computerworld.dk/art/226786/internet-of-things-derfor-vil-alt-aendre-sig-nu,
Computerworld, 22 maj 2013.

[5] Machine-to-Machine, Hvad er M2M - Machine-toMachine kommunikation?,
http://meremobil.dk/2014/07/hvad-er-m2m-machine-machine-kommunikation/,
meremobil.dk, 10 juli 2014.

[6] Intenet of Things, Du kan næsten putte alt ind under Internet of Things lige nu,
http://www.version2.dk/artikel/du-kan-naesten-putte-alt-ind-under-internet-things-lige-nu-
296377,
Version2, 27 juli 2015.

[7] Smart Grid, Smart Grid i Danmark,
http://energinet.dk/SiteCollectionDocuments/Danske%20dokumenter/El/Det%20intelligente
%20elsystem%20-%20SmartGrid%20i%20Danmark%20rapport.pdf,
Energinet.dk og Dansk Energi, 2010.

[8] Smart Grid, Smart Grid - fremtidens intelligente elnet,
https://www.bolius.dk/smart-grid-fremtidens-intelligente-elnet-22038/,
Bolius, 21 maj 2014.

[9] Danmarks historien, Harald Blåtand ca. 958-987,
http://danmarkshistorien.dk/leksikon-og-kilder/vis/materiale/harald-blaatand-ca-958-987/,
Inge Skovgaard-Petersen, Danmark, 21 maj 2012.

[10] Blutooth history, A short history of Bluetooth,
https://www.nordicsemi.com/eng/News/ULP-Wireless-Update/A-short-history-of-Bluetooth,
Nordic Semiconductor, 14 Juli 2014.

[11] Gazell protokol, Gazell link layer user guide,
https://developer.nordicsemi.com/nRF51_SDK/nRF51_SDK_v6.x.x/doc/6.1.0/s110/html/a00113.html,
Nordic Semiconductor, 29 August 2014.

[12] GATT, Bluetooth Smart - GATT,
https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt,
Kevin Townsend, 5 April 2015.

[13] Bluetooth Smart, Introduction to Bluetooth Low energy,
https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction,
Kevin Townsend, 17 April 2014.

[14] nRF51 Reference Manual, nRF51 Series Reference Manual
Nordic Semiconductor, Version 3.0 20 Oktober 2014.

Målejouraler A

A.1 Sensor Setup

Der vil her være beskrevet opsætningen for sensor kredsen, som er benyttet igennem projektet. Som beskrevet tidligere i afsnittet omkring design sensor funktioner i afsnit 4.3.1, er det nødvendig at forstærke signalet fra temperatur sensoren. Der er ligeledes fundet frem til at signalet skal forstærkes med en faktor på 2,4 gange. Til dette vil en ikke inverterende operationsforstærker være benyttet.

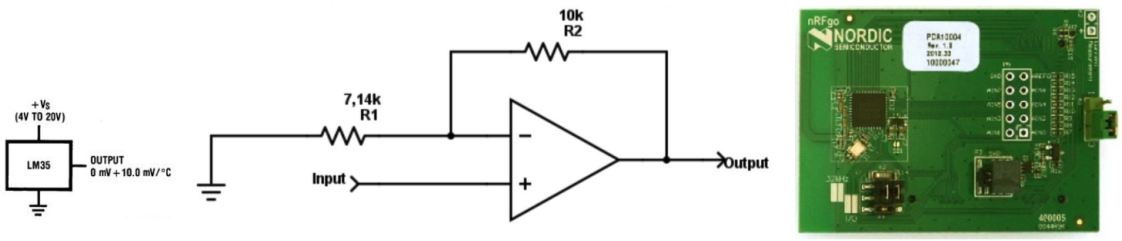
For at opnå en forstærkning på 2,4 gange kan opsætningen af operationsforstærker kredsen udregnes som følgende.

$$Forstaerkning = 1 + \frac{R2}{R1}$$

$$2,4 = 1 + \frac{10K\Omega}{R1}$$

$$R1 \approx 7,14k\Omega$$

Den samlede opsætning for temperatur sensor og operationsforstærker kan ses nedenfor på figur A.1. Der ses også hvordan operationsforstærkeren er opsat med de to modstande på henholdsvis $7,14k\Omega$ og $10K\Omega$. Der er til både temperatur sensoren og operationsforstærkeren benyttet en forsyningsspænding på 5 V.



Figur A.1. Diagram over sensor opkobling

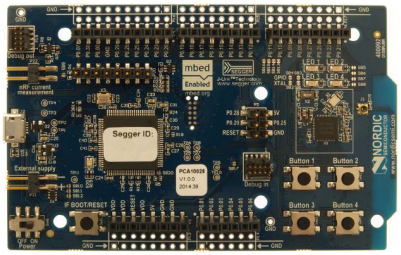
A.2 Test af Digitale I/O Porte

Formål

Formålet med denne test er at opsætte de digitale ind og udgange på Nordic kredsen.

Testopstilling

Der er her benyttet de indbyggede knapper og LED for at teste de digitale I/O porte på Nordic udviklingskittet.



Figur A.2. Nordic nRF51 Development Kit

Anvendt Udstyr

Instrument	Fabrikat	Type/Model	Bemærkninger
SOC	Nordic Semiconductor	nRF51-DK	Benyttes som nRF51822
Multimeter	Fluke	115	

Benyttede Kode Eksempel

Kode udsnittet nedenfor er opsat så LED_2 er tændt, når knappen holdes nede.

```
int main(void)
{
  // Button and LED setup
  nrf_gpio_cfg_output(LED_2);
  nrf_gpio_cfg_input(BUTTON_2, BUTTON_PULL};

  uint8_t btn_nstate;
  uint8_t btn_ostate = nrf_gpio_pin_read(BUTTON_2);

  while(true)
  {
    uint8_t btns = 0;
    btn_nstate = nrf_gpio_pin_read(btn);

    if ((btn_ostate == 1) && (btn_nstate == 0) )
    {
      nrf_gpio_pin_clear(LED_2);
    }
    else
    {
      nrf_gpio_pin_set(LED_2);
    }
    btn_ostate = btn_nstate;
```

```
24    }
}
```

Testprocedure

Testen er udført ved at implementer koden på udviklingskittet nRF51-DK, hvorefter knappen med van BUTTON_2 er trykket på.

Resultat

Testusikkerhed/Fejlkilder

Fejlkilder kan være hvis opsætningen af ind- eller udgangene ikke stemmer overens med den tilsluttede eksterne komponent.

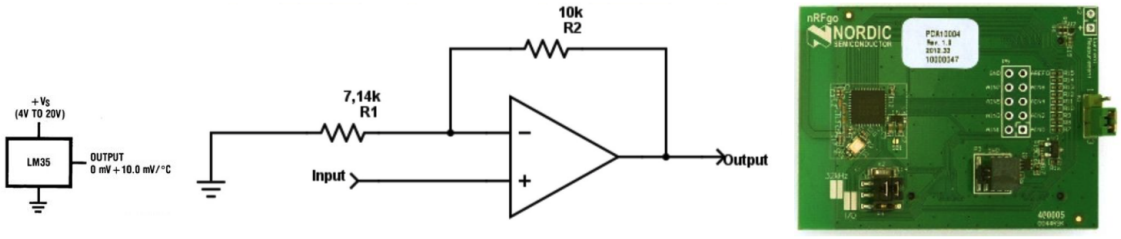
A.3 Test af Sensor funktionalitet

Formål

Formålet med denne test er at opsætte en analog indgang på udviklingskittet, for herefter at aflæse resultatet. Der vil ikke være implementeret nogen skalerings funktionen, hvorfor det blot vil være værdien fra AD konverteren.

Testopstilling

Der er benyttet testopstillingen fra appendiks A.1.



Figur A.3. Diagram over sensor og operationsforstærker opkoblingen

Anvendt Udstyr

Instrument	Fabrikat	Type/Model	Bemærkninger
SOC	Nordic Semiconductor	nRF51-DK	Benyttes som nRF51822
OPAMP	Texas Instruments	TLC071C	Opsat med en forstærkning på 2,4
Sensor	National Semiconductor	LM35	

Benyttede Kode Eksempel

Til denne test er der benytte den standard opsætning af den analoge indgang, og resultatet er gemt i adc_sample. For at udskrive denne data, er der opsat en BLE kommunikation, som kan sende datene til en terminal på en computer. På kode udsnittet nedenfor sendes dataene derfor over BLE.

```
1 /??? Send sensor data ?? ?/
  void send_data()
3 {
    uint8_t str  [4];
5   sprintf  ((char?)str  ,  "%d", (int)adc_sample);
    ble_nus_send_string(&m_nus, str, strlen((char?)str));   // Send data with BLE
7 }
```

Testprocedure

Når funktionen fra kodeudsnittet ovenfor kaldes, sendes den aflæste værdi over en UART forbindelse til en terminal på en computer.

Resultat

Den opfylder ønsket om at konvertere den analoge spænding til en digital værdi, til senere brug i designet.

Testusikkerhed/Fejlkilder

Fejlkilder kan være hvis forstærkningen på operationsforstærkeren ikke er præcis, ud fra en afvigelse på de benyttede modstande.