

Underskrifter

Dato for aflevering: 28/05-2015

Gruppemedlemmer:

Kristian Boye Jakobsen (KB) 201370738

Lukas Hedegaard Jensen (LH) 201370768

Poul Overgaard (PO) 201370794

Jeppe Hofni (JH) 201370801

Mick Kirkegaard (MK) 201370970

Felix Blix Everberg (FB) 201371008

Vejleder: Arne Justesen

Indholdsfortegnelse

Kapitel 1 Termliste	7
Kapitel 2 Kravspecifikation	9
2.1 Indledning	9
2.2 Use cases	12
2.3 Aktører	12
2.4 Fully dressed use cases	13
2.5 Ikke-funktionelle krav	16
2.6 Web-skitser	17
2.7 E-mail beskeder	19
Kapitel 3 Baby Watch	21
3.1 Systemarkitektur	21
3.1.1 Domænemodel	21
3.1.2 System-states	22
3.1.3 BDD	23
3.1.4 IBD	23
3.2 Signalbeskrivelse	24
3.2.1 Allokering af logiske funktionaliteter	25
3.3 Grænseflader	26
3.3.1 Fysiske forbindelser	26
3.3.2 Grænseflade mellem Intelligent Lydmonitor og Controller	26
3.3.3 Grænseflade mellem Vuggesystem og Controller	27
3.3.4 Grænseflade mellem Controllers servere og ekstern internet enhed	29
Kapitel 4 Controller	31
4.1 Systemarkitektur	31
4.1.1 Softwarearkitektur	32
4.1.2 Hardware arkitektur	35
4.1.3 Signaltabel	36
4.2 Design	37
4.2.1 Softwaredesign	37
4.2.2 Hardwaredesign	41
4.3 Implementering	47
4.3.1 Softwareimplementering	47
4.3.2 Hardwareimplementering	51
4.4 Modultest	52
Kapitel 5 Intelligent lydmonitor	61
5.1 Forundersøgelse	61
5.1.1 Metode	61

5.1.2	Analysekode gennemgang	62
5.1.3	Analyser	65
5.1.4	Konklusion	80
5.2	Systemarkitektur	81
5.2.1	Hardware arkitektur	82
5.2.2	Software arkitektur	83
5.3	HW design og implementering	89
5.3.1	Mikrofon preamp	89
5.3.2	ADC forfiltrering	93
5.3.3	Spændingsregulering	97
5.4	SW design og implementering	99
5.4.1	Generelt	99
5.4.2	Prefilter	99
5.4.3	RecBuf	102
5.4.4	Analyzer	104
5.4.5	Categorizer	105
5.4.6	Statistician	105
5.4.7	TwoWireCom	105
5.5	Modultest	106
5.5.1	Modultest Prefilter	106
5.5.2	Modultest RecBuf	106
5.5.3	Modultest Analyzer	108
5.5.4	Modultest Categorizer	108
5.5.5	Modultest Statistician	108
5.5.6	Modultest TwoWireCom	108
Kapitel 6 Vuggesystem		111
6.1	Systemarkitektur	111
6.1.1	Hardware arkitektur	111
6.1.2	Grænsefladebeskrivelse	114
6.1.3	Software arkitektur	115
6.2	Design	118
6.2.1	Hardware design	118
6.2.2	Software design	125
6.3	Implementering	129
6.3.1	Hardware implementering	129
6.3.2	Software implementering	135
6.4	Modultest	140
6.4.1	Hardware modultest	140
6.4.2	Software modultest	143
Kapitel 7 Power Supply Unit		147
7.0.3	DC-DC konverter (LM2596S)	147
7.0.4	Spændingsregulatorer (LM317)	148
7.0.5	Relæ styring	150
7.0.6	Samlet PSU print	151

7.0.7	Implementering af PSU	152
7.0.8	Modultest	153
Kapitel 8	Integrationstest	155
Kapitel 9	Accepttestspezifikation	157
Kapitel 10	Bilag (CD-indhold)	163
Litteratur		165

Termliste 1

AASH Antal af samtidige hændelser

Baby Aktøren som står for at generere lyd til systemet

Baby Watch Navnet på systemet

BABYCON niveau En skala for babys humørtilstand (1, 2 og 3)

BABYCON3: Rolig Lyden indikerer at baby sover

BABYCON2: Urolig Lyden indikerer at baby er vågen

BABYCON1: Alarmrende Lyden indikerer at baby er i en tilstand der kræver tilsyn

Babypasser Brugeren som ønsker at benytte systemet

Betjeningspanel Et panel bestående af to knapper(En ON/OFF-knap og en Manuelstart-knap) og tre LED'er(En ON/OFF-LED, en "Manuel start-LED og en Wi-fi-LED).

Blackfin Er en forkortelse af Blackfin ADSP-BF533 udviklingsboardet

Intelligent lydmonitor Del af system som står for indsamling/behandling af lyd

Manuel start Knap med funktion der manuelt starter Putte-tilstand.

Monitorerings-tilstand En tilstand hvor systemet styrer barnevognens vuggefunktion på baggrund af lyde analyseret af den Intelligent lydmonitor

Passiv monitorering Dækker over en tilstand hvor der vuges manuelt og monitoreres for BABYCON1

Putte-tilstand En tidsafgrænset tilstand hvor barnevognen vugger uden at blive styret af lydininput. Kaldes også "Manuel-start tilstand"

Regulerings MCU Beskriver micro controller enheden der styrer Vuggesystemet

Undtagelses-tilstand Dækker over tilstanden hvor barnevogn stopper og går i BABYCON1

Vugge-indsving Skal forstås som når barnevognen begynder en vuggesekvens af barnevognskurven så skal vuggesystemet give det en blid opstart

Wi-Fi-LED Diode der lyser når WiFi forbindelsen til netværk er afbrudt

Kravspecifikation 2

2.1 Indledning

Formålet med projektet er at lave en prototype af en intelligent babymonitor til barnevogne med dertilhørende vuggesystem, statushjemmeside og e-mail notifikation.

Systemet har tre tilstande; en ”manuel start”-tilstand, hvor barnet lægges til at sove, en monitorerings-tilstand og en undtagelses-tilstand. I ”**manuel start**”-tilstanden, skal systemet vugge babyen i et fastsat tidsinterval, hvorefter monitoreringen overtager. I **monitorerings-tilstanden** styres barnevognens vuggefunktion på baggrund af analysen af den aktuelle baby-lydoptagelse. I **undtagelses-tilstanden** skal barnevognen ikke vugge. En e-mail afsendes til den registrerede babypasser, og en alarmlyd afspilles på statushjemmeside indtil systemet resettes manuelt ude ved barnevognen.

Statushjemmesiden opdateres løbende på baggrund babyens tilstand. Tilstanden bliver på hjemmesiden kategoriseret i tre konditioner vist via en BABYCON statusbar:

- **BABYCON3**

På dette niveau kategoriseres lydsignalet fra babyen som roligt. Derfor skal barnevognen ikke vugges. Systemet indsamler lyd og afventer en ændring i lydsignalet, som vil medføre en ændring i BABYCON niveau.

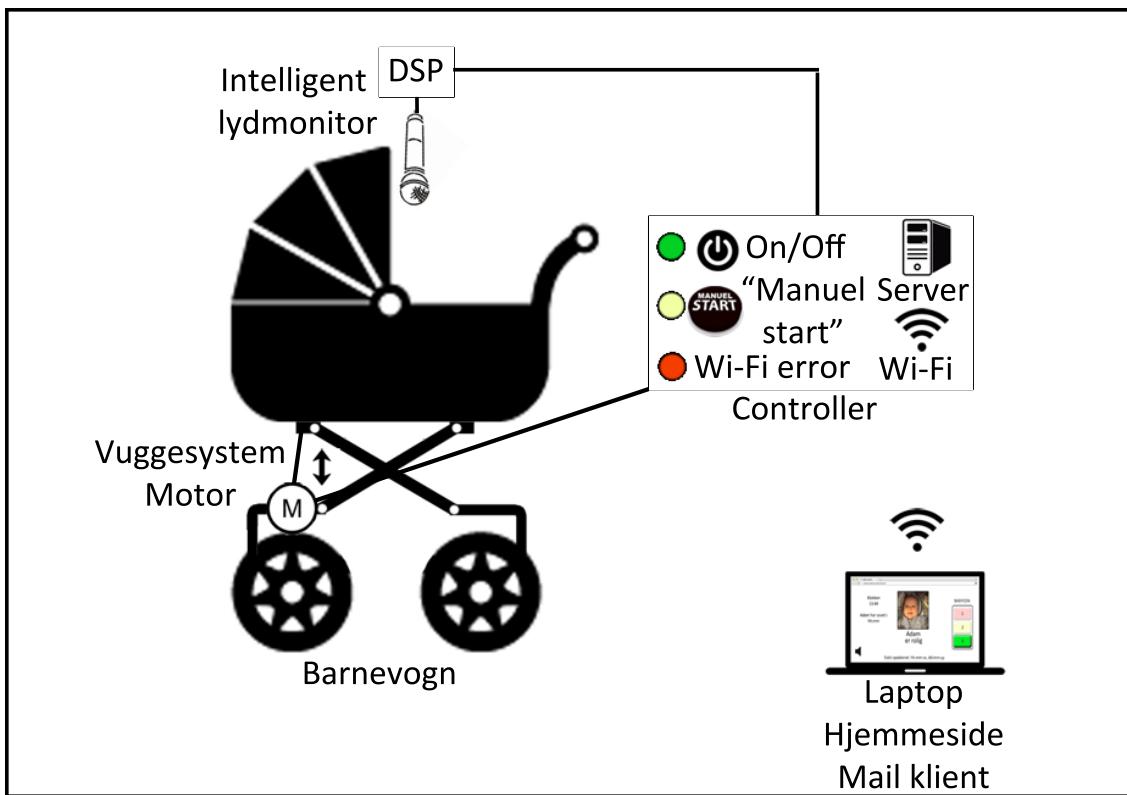
- **BABYCON2**

På dette niveau kategoriseres lydsignalet fra barnet som uroligt. Heraf skal barnevognen vugge gennem en sekvens af tre forskellige vugge-tilstande. Hvis babyen bliver beroliget af en bestemt vuggetilstand, detekteres dette af den intelligente lydmonitor. Dette vil medføre en ændring til BABYCON3. Hvis babyen afgiver en alarmerende lyd, vil det medføre en ændring til BABYCON1.

- **BABYCON1**

På dette niveau kategoriseres lydsignalet fra barnet som alarmerende. Undtagelses-tilstanden aktiveres.

Systemtegning



Figur 2.1. Skitse af systemet Baby Watch

Figur 2.1 illustrerer systemet. Barnevognen er udstyret med en controller som står for at styre de andre delsystem. På controlleren sidder et brugerpanel som sørger for at babypasseren fysisk kan interagere med systemet via knapper og LED'er. Den intelligente lydmonitor består af en mikrofon og en digital signal processer (DSP), som sender informationer til controlleren om babyens tilstand ud fra en processering af babyens lyde; herunder pitch og power. Vuggesystemet består af en motor og et selvregulerende system, som sørger for, at barnet altid vuges ud fra et vandret niveau. Vuggesystemet opererer med tre vuggetilstande (med hver deres frekvens og amplitude) på baggrund af kontolsignalet fra controlleren. Controlleren indholder også en http server med statushjemmesiden og en e-mail client. Controlleren har forbindelse via Wi-Fi til internettet. Babypasseren kan opsoge information om babyens tilstand på statushjemmesiden og modtager e-mails når systemet detekterer, at babyen er i en alarmerende tilstand eller hvis der opstår fejl i systemet.



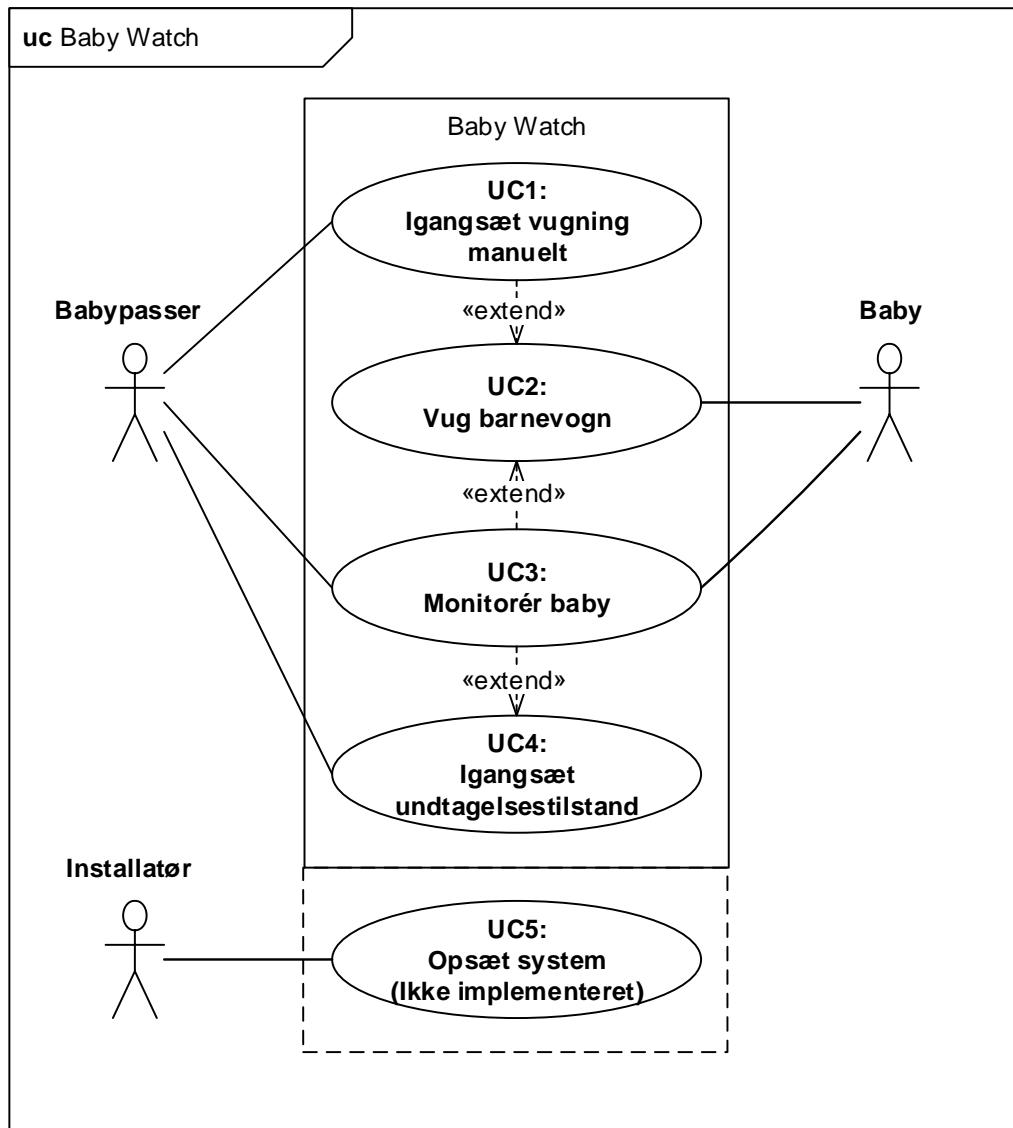
Figur 2.2. Controllerens knapper og lysdioder

Figur 2.2 viser Controllerens 2 knapper: ON/OFF-knap og "Manuel start"-knap. De 3 lysdioder er også illustreret. Den grønne indikerer at Baby Watch er ON, den gule indikerer at "Manuel start" er aktiveret og den røde indikerer manglende netværksforbindelse.

2.2 Use cases

I dette afsnit specificeres brugssituationer for systemet Baby Watch.

Kravspecifikationen er udfærdiget med basis i en use case-baseret tilgang. Den indeholder en beskrivelse over relevante aktører for systemet samt fully dressed use cases.



Figur 2.3. Usecase diagram for Baby Watch

Usecase diagrammet giver et overblik aktørerne og deres rolle i use cases.

2.3 Aktører

Aktør navn	Beskrivelse
Babypasser	En person som ønsker at benytte systemet til at berolige Baby til at falde i søvn samt monitorerer Babys tilstand elektronisk
Baby	Spædbarn som monitoreres og beroliges af system
Installatør	Tekniker, der opsætter systemet (optræder i den ikke-implementerede UC5)

2.4 Fully dressed use cases

UC1: Igangsæt vugning manuelt	
Mål	At igangsætte vugning af barnevogn manuelt
Initialisering	Babypasser
Aktører og Stakeholders	Babypasser(Primær)
Referencer	UC2, UC3, UC4
AASH	1
Efterfølgende tilstand	UC3: Monitorér baby igangsat
Hovedforløb	<ol style="list-style-type: none"> 1. Babypasser igangsætter manuel vugning ved tryk på "Manuel start"-knap på Baby Watch brugerpanel 2. Systemet starter vugning jf. UC2 3. Efter 2 minutter igangsætter systemet passiv monitorering af baby jf. UC3 4. Efter 5 minutter igangsætter systemet automatisk monitorering af baby jf. UC3 [Und: 4.a Alarmerende baby detekteret indenfor 5 min]
Undtagelser	<p>4.a Alarmerende baby detekteret efter 2 min</p> <ul style="list-style-type: none"> • Systemet igangsætter undtagelsestilstand jf. UC4

UC2: Vug barnevogn	
Mål	At vugge barnevognen i et tidsinterval
Initialisering	UC1, UC3
Aktører og Stakeholders	Baby(Sekundær)
Referencer	UC1, UC3, UC4
AASH	1
Efterfølgende tilstand	Barnvognen er i stilstand
Hovedforløb	<ol style="list-style-type: none"> 1. Systemet igangsætter vugge-indsvingning 2. Systemet følger prædefineret vuggesekvens jf. ikke-funktionelle krav (2.5) [Und: 2.a Fejl i vuggesystem] 3. Fra UC3 eller UC4 gives besked om stop af vugning 4. Systemet dæmper vugning til stilstand i vandret
Undtagelser	<p>2.a Fejl i vuggesystem</p> <ul style="list-style-type: none"> • Afsend Fejl i vuggesystem e-mail • Systemet stopper vugning • Systemet stopper monitorering • Systemet afventer genstart

UC3: Monitorér baby	
Mål	Analysere babytilstand og ager på baggrund af denne
Initialisering	UC1
Aktører og Stakeholders	Baby(Primær), Babypasser(Sekundær)
Referencer	UC1, UC2, UC4
AASH	1
Efterfølgende tilstand	Monitorér baby fortsættes
Hovedforløb	<ol style="list-style-type: none"> 1. Systemet optager lyd i tidsintervaller af 10 sekunder 2. Systemet analyserer indsamlet lyd 3. Systemet opdaterer BABYCON tilstand på baggrund af analyse 4. Systemet igangsætter vugning jf. UC2 (analyse viser BABYCON2) <ul style="list-style-type: none"> [Und: 4.a Vugning allerede igang (forrige analyse viste BABYCON2)] [Und: 4.b Analyse viser BABYCON3 (forrige analyse viste BABYCON2)] [Und: 4.c Analyse viser BABYCON3 (forrige analyse viste BABYCON3)] [Und: 4.d Analyse viser BABYCON1] 5. Systemet opdaterer BABYCON tilstand på hjemmeside <ul style="list-style-type: none"> [Und: 5.a Ingen netværksforbindelse] 6. Genstart UC3
Undtagelser	<ol style="list-style-type: none"> 4.a Vugning allerede igang (forrige analyse viste BABYCON2) <ul style="list-style-type: none"> • Systemet fortsætter til punkt 5 4.b Analyse viser BABYCON3 (forrige analyse viste BABYCON2) <ul style="list-style-type: none"> • Systemet stopper vugning jf. UC2 punkt 3 4.c Analyse viser BABYCON3 (forrige analyse viste BABYCON3) <ul style="list-style-type: none"> • Systemet fortsætter til punkt 5 4.d Analyse viser BABYCON1 <ul style="list-style-type: none"> • Systemet går i undtagelsestilstand jf. UC4 5.a Ingen netværksforbindelse <ul style="list-style-type: none"> • Wi-Fi-LED lyser

UC4: Igangsæt undtagelsestilstand	
Mål	At stoppe vugning og alarmere Babypasser
Initialisering	UC2, UC3
Aktører og Stakeholders	Babypasser(Sekundær)
Referencer	UC2, UC3, UC5
AASH	1
Efterfølgende tilstand	Vugning indstillet, e-mail afsendt og hjemmeside opdateret
Hovedforløb	<ol style="list-style-type: none"> 1. Systemet stopper vugning jf. UC2 punkt 3 2. Systemet opdaterer hjemmeside til BABYCON1 [Und: 2.a Ingen netværksforbindelse] 3. System afsender e-mail til Babypasser [Und: 3.a Ingen netværksforbindelse] 4. System afventer genstart fra Babypasser
Undtagelser	<p>2.a Ingen netværksforbindelse</p> <ul style="list-style-type: none"> • Wi-Fi-LED lyser <p>3.a Ingen netværksforbindelse</p> <ul style="list-style-type: none"> • Wi-Fi-LED lyser

2.5 Ikke-funktionelle krav

Mikrofon

For at kunne opfange et tilstrækkeligt signal til analyse af babyens gråd, skal systemets mikrofon opfylde følgende krav:

- Mikrofonen skal have en max SPL rating på min. 120 dB.¹
- Mikrofonen skal have en jævn frekvens respons på maksimalt +/- 5 dB fra 40 Hz til 10 kHz

Vuggemekanisme

Nivellering:

- Vuggesystemet skal kunne nivellere planet, hvorpå babyen ligger, til vandret position indenfor 2°.
- Barnevognens understel må stå på et plan med op til 5° hældning.
- Når systemet er tændt, men ikke skal vugge, nivelleres planet, hvorpå babyen ligger, automatisk til vandret.

Ved vugning jf. UC2 gennemgår vugningen af barnet. Systemets vugge mekanisme skal overholde følgende krav for at sikre en blid vugning:

¹FIXME indsæt reference til studie om gråd volumen

- Vuggen skal kunne vippe planet, hvorpå babyen ligger, med op til 10° i hver retning fra dets vandrette udgangspunkt, med en fejlmargin på 2° .
- Vuggen skal kunne variere frekvensen hvormed der vugges fra 0 Hz til 2 Hz, med en fejlmargin på 0,2 Hz.
- Vuggen skal vende tilbage til vandret indenfor en vinkel på 2° når systemet lukkes ned.
- Vuggen skal have en begrænsning på vinkelfrekvensen ved 80° s^{-1} , med en fejlmargin på 10 %.
- Vuggens vinkel acceleration skal være begrænset ved 20° s^{-2} , med en fejlmargin på 10 %.

Vuggetilstande: Ved vugning jf. UC2 gennemgår vugningen af barnet en sekvens af tre vuggetilstande med et interval på 2 min.

1. Vugning foregår med en frekvens på 0,5 Hz og en amplitude på $10^\circ +/- 2^\circ$
2. Vugning foregår med en frekvens på 1 Hz og en amplitude på $6^\circ +/- 2^\circ$
3. Vugning foregår med en frekvens på 2 Hz og en amplitude på $4^\circ +/- 2^\circ$

Baby status

For at sikre at vurderingen af babyens status er pålidelig, samt rettidigt tilgængelig for brugeren skal systemet overholde følgende:

- Systemets BABYCON-statusbar(se illustration nedenfor) skal opdateres minimum hvert 10. sekund.
- Statushjemmesiden skal være opdateret senest 5 sekunder efter controlleren har opdateret babystatus.
- Når BABYCON-statusbaren opdateres til BABYCON1 niveau skal hjemmesiden afspillede en alarmlyd og BABYCON-statusbaren røde felt skal blinket med 2 Hz interval

2.6 Web-skitser

De følgende figurer 2.4 , 2.5 samt 2.6 skitser statushjemmesidens udseende. Hjemmesiden viser aktuel tid, samlet tid som barnet har været i ro, et billede af barnet med tilhørende navn og besked, BABYCON skala fra 1-3 og information omkring hvornår hjemmesiden sidst er opdateret.



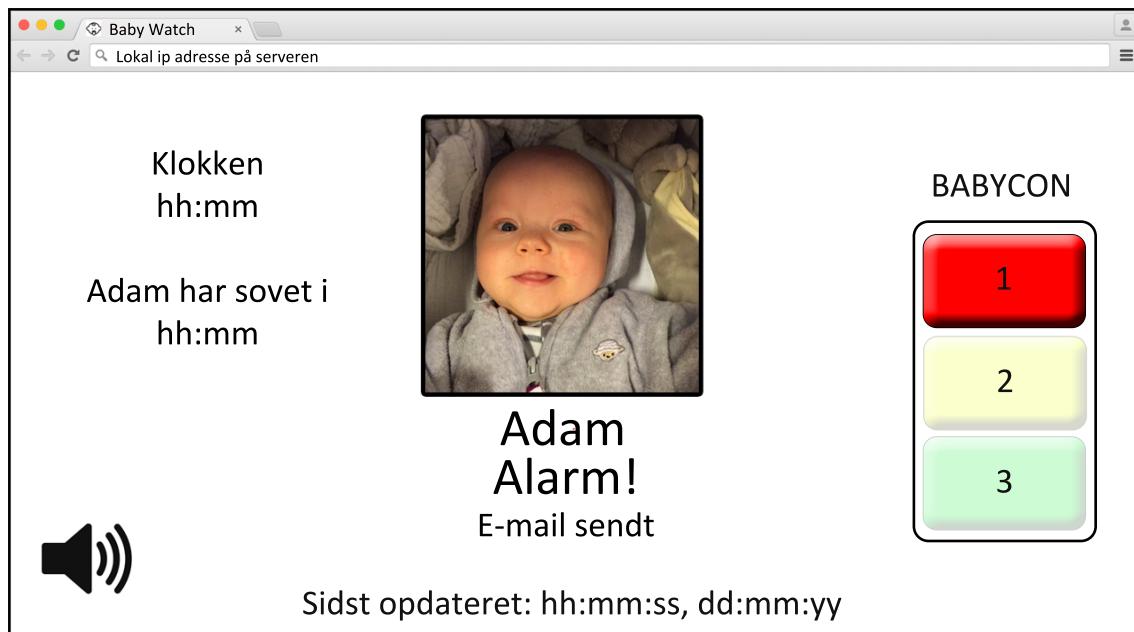
Figur 2.4. BABYCON3

Figur 2.4 illustrerer hjemmesiden når BABYCON niveauet er 3, det niveau hvor Babyen er rolig.



Figur 2.5. BABYCON2

Figur 2.5 illustrerer hjemmesiden når BABYCON niveauet er 2, det niveau hvor Babyen er urolig, men ikke nok til at udløse alarm til Babypasser.



Figur 2.6. BABYCON1

Figur 2.6 illustrerer hjemmesiden når BABYCON niveauet er 1. BABYCON1 er niveauet hvor Babyen er alarmerende utilfreds. Babypasseren modtager en e-mail og skal selv trøste Baby. På dette niveau skal hjemmesiden afspille en alarmlyd og det BABYCON røde felt skal blinke.

2.7 E-mail beskeder

BabyWatch: Alarm!

Emne: BabyWatch: Alarm!

Tekst: Baby Watch har registreret en alarmerende baby (Vugning er stoppet)

BabyWatch: Fejl!

Emne: BabyWatch: Fejl!

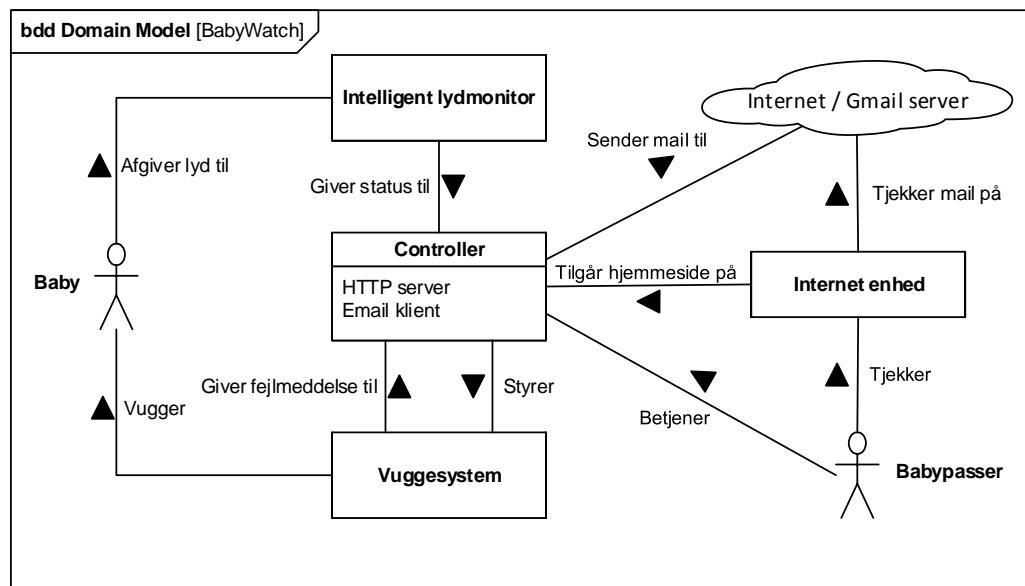
Tekst: Fejl i vuggesystem, afventer genstart.

Baby Watch 3

3.1 Systemarkitektur

Det følgende afsnit beskriver den overordnede systemarkitektur for Baby Watch. Afsnittet består af forskellige diagrammer og tabeller med tilhørende forklaringer. Diagrammerne er opbygget efter SysML standarden.

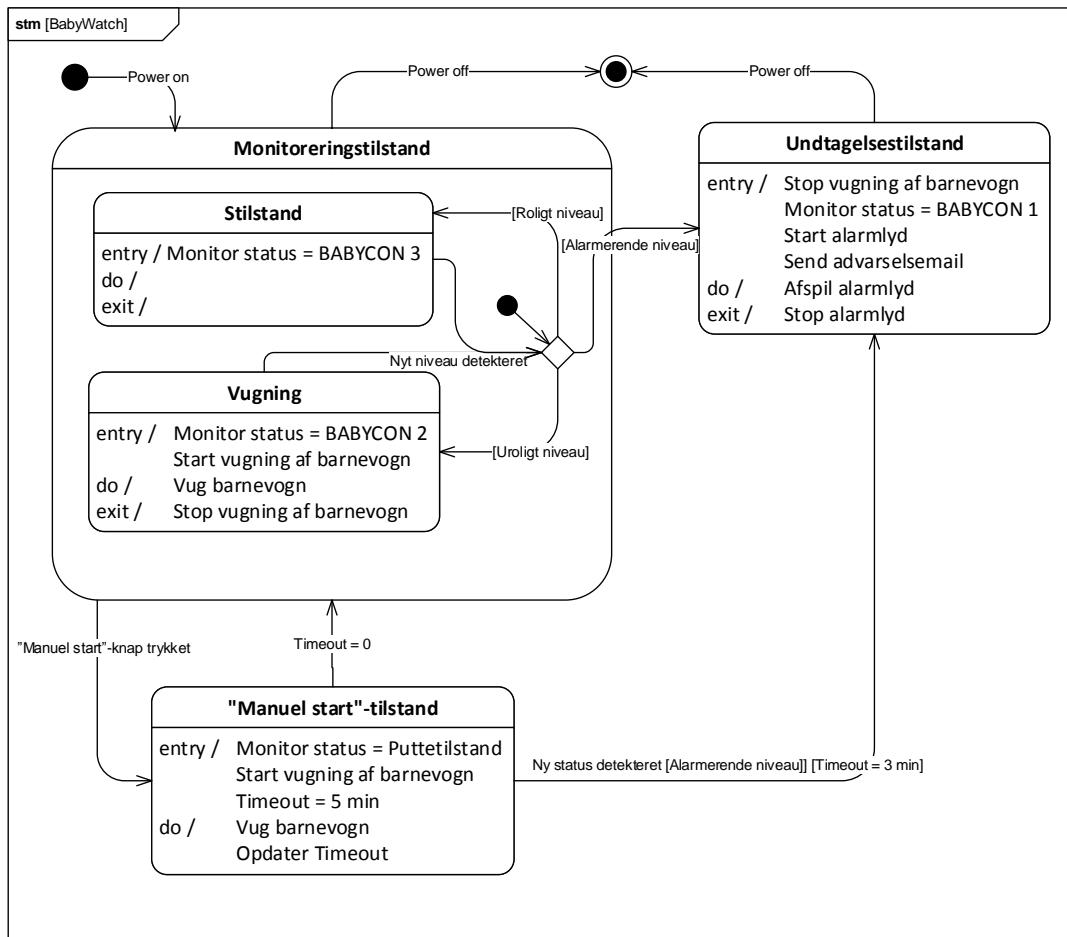
3.1.1 Domænemodel



Figur 3.1. Domænemodel for Baby Watch

Figur 3.1 viser domænemodellen for Baby Watch. Modellen beskriver systemets funktionalitet og indbyrdes sammenhæng. Multipliceter på 1 til 1 er udeladt i diagrammet. Babypasseren kan tjekke status på flere enheder (antal simultane enheder begrænses af HTTP serverens kapacitet). Systemet er som udgangspunkt kun designet til én baby og multipliciteten for denne er derfor også 1 til 1.

3.1.2 System-states



Figur 3.2. Overordnede states for Baby Watch

Systemets har tre overordnede states: Monitoringstilstand, ”Manuel start”-tilstand og Undtagelsestilstand.

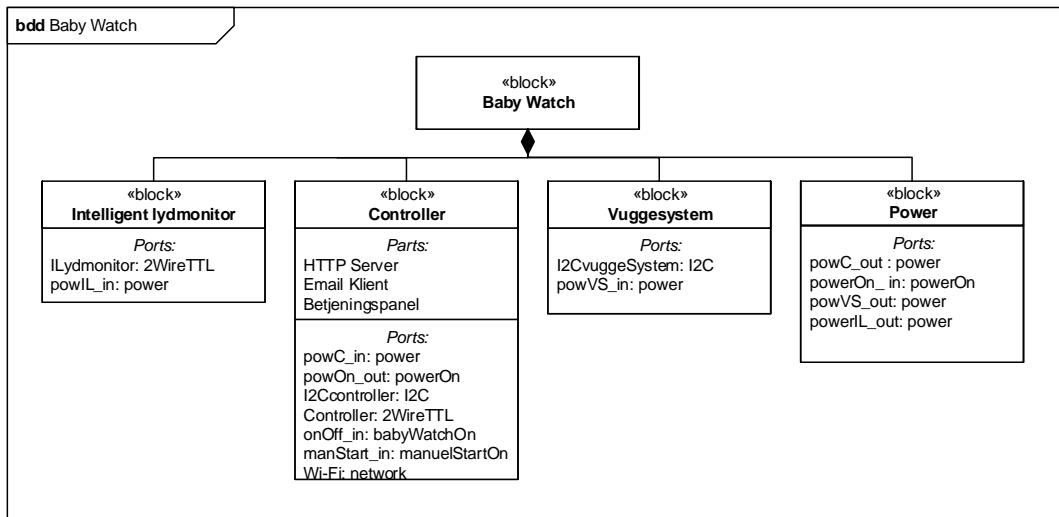
Monitoreringstilstand har to substates:

- **Stilstand**, der indikerer BABYCON3 niveau med rolig baby. Vuggesystemet er for denne tilstand inaktivt
- **Vugning**, der indikerer BABYCON2 niveau med urolig baby og hvori barnevognen vugger automatisk.

”Manuel start”-tilstand igangsættes manuelt af babypasseren. Her sættes barnevognen til at vugge i 5 minutter, hvorefter systemet går i Monitoringstilstand. Efter to minutter afbryder en måling af barneaktivitet svarende til BABYCON1 også ”Manuel start”-tilstand. Systemet går herefter i undtagelsestilstand.

Undtagelsestilstand indikerer BABYCON1 niveau med højeste alarmberedskab. Her stoppes barnevogens vuggefunktion, der sendes en advarselsemail til Babypasser og hjemmeside opdateres til BABYCON1, som igangsætter afspilning af en alarmlyd.

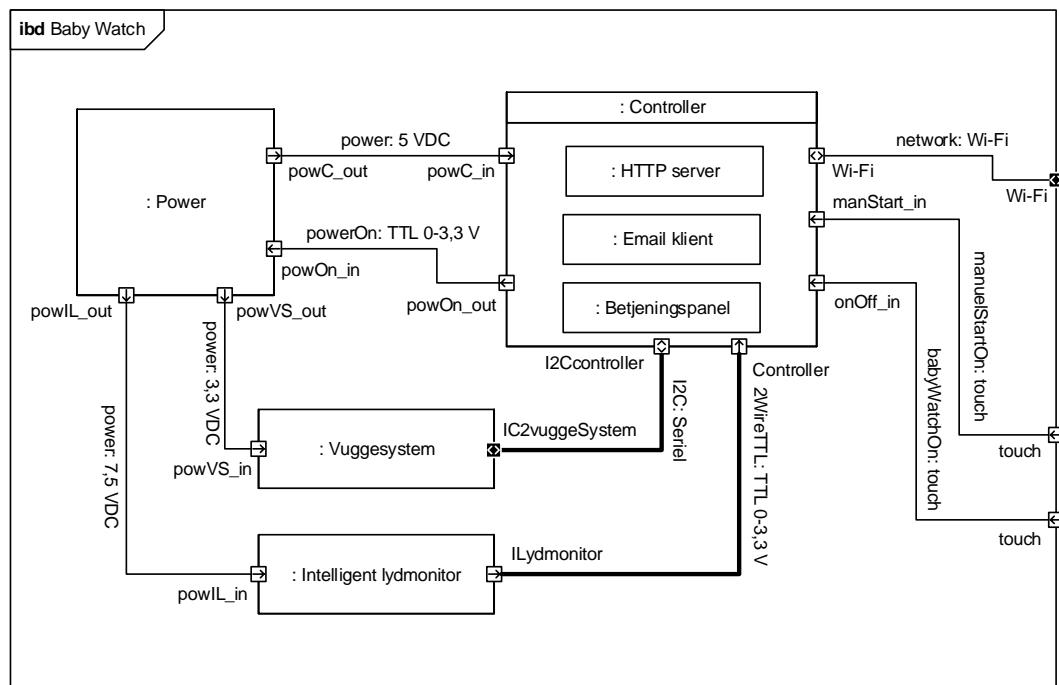
3.1.3 BDD



Figur 3.3. Overordnet BDD for Baby Watch

Figur 3.3 viser det overordnede BDD for Baby Watch systemet. Som figuren viser, består systemet af en intelligent lydmonitor, et vuggesystem, en power blok, samt en controller, der indeholder en HTTP server, en Email klient samt et betjeningspanel.

3.1.4 IBD



Figur 3.4. Overordnet IBD for Baby Watch

Figur 3.4 viser det overordnede IBD for Baby Watch systemet. Figuren viser de interne forbindelser for blokkene i figur 3.3. For yderlige specifikation af porte og signaler se signalbeskrivelsen 3.2.

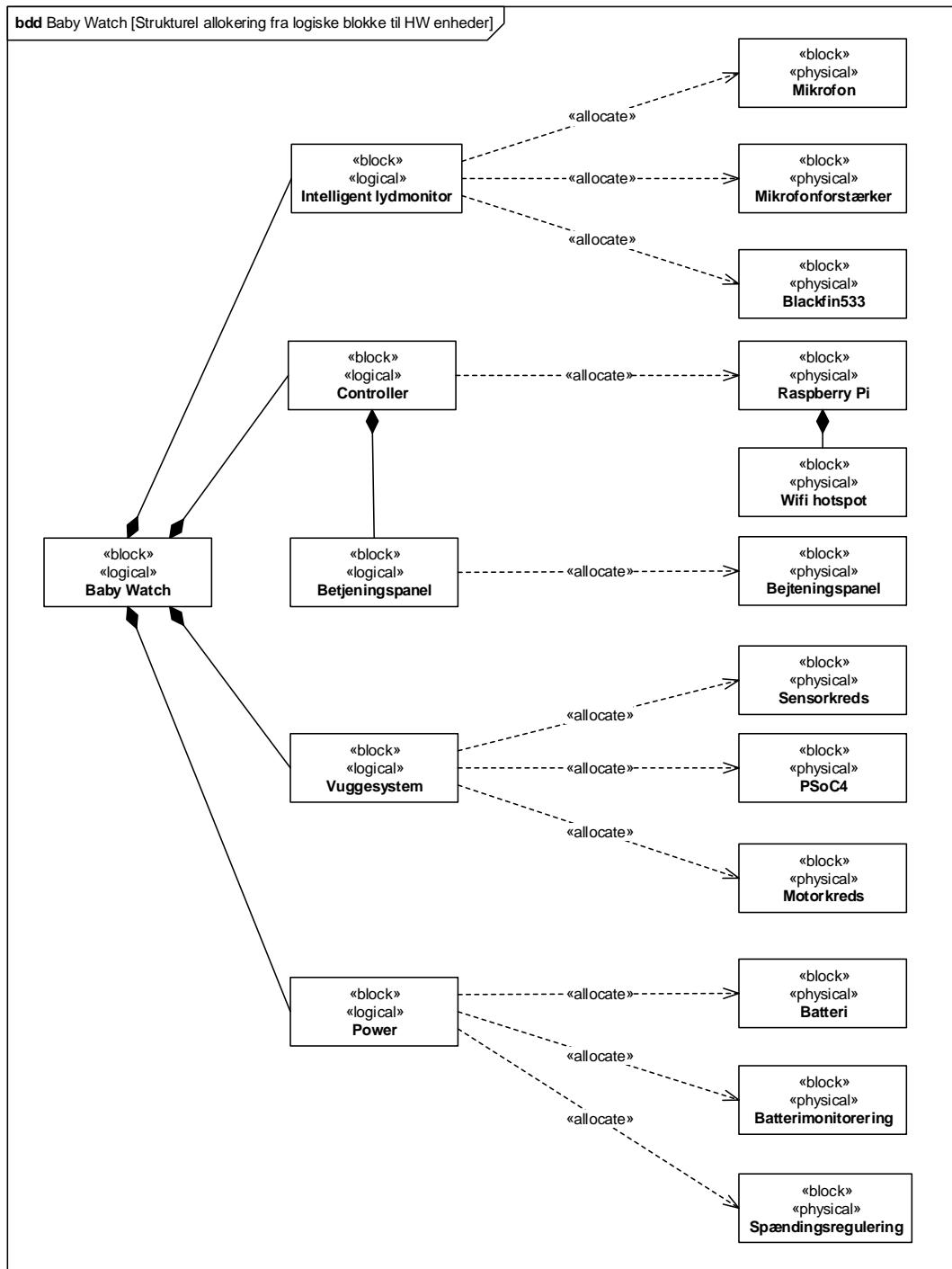
3.2 Signalbeskrivelse

Følgende viser signaltabellen for det overordnede system med forbindelser, portnavne og en beskrivende kommentar.

Tabel 3.1: Signaltabel for Baby Watch

Signal-navn	Type	Port 1	Port 2	Kommentar
manuelStartOn	touch	Babypasser (touch)	Controller (manStart_in)	Babypasser der laver et tryk på knappen
babyWatchOn	touch	Babypasser (touch)	Controller (onOff_in)	Babypasser der laver et tryk på knappen
power	5 VDC, 1200 mA	Power (powC_out)	Controller (powC_in)	Forsyning til Controller
power	3,3 VDC, 500 mA	Power (powVS_out)	Vuggesystem (powVS_in)	Forsyning til Vuggesystem
power	7,5 VDC, 1500 mA	Power (powIL_out)	Int. lydmonitor (powIL_in)	Forsyning til Intelligent Lydmonitor
powerOn	TTL 0-3,3 VDC	Controller (powOn_out)	Power (powOn_in)	Benyttes til at tænde og slukket for strømforsyningen
I2C	Seriell	Controller (I2CController)	Vuggesystem (I2Cvuggesystem)	Seriell kommunikation
2WireTTL	TTL 0-3,3 VDC	Int. lydmonitor (ILydmonitor)	Controller (Controller)	Dobbelt ledet TTL signal til identifikation af BABYCON niveau
network	Wi-Fi	Controller (Wi-Fi)	Baby Watch (Wi-Fi)	Lokal trådløs lan forbindelse

3.2.1 Allokering af logiske funktionaliteter



Figur 3.5. Allokéringsdiagram for Baby Watch

Figur 3.5 viser allokeringen af systemets logiske blokke til specifikke HW-enheder.

Således allokeres Intelligent Lydmonitor som en mikrofon, en mikrofonforstærker og et Blackfin533 til optagelse, analyse og kommunikation. Controller allokeres som en Raspberry Pi med tilhørende Wi-Fi modul. Controllerens HTTP server og Email klient er således

også allokeret på Raspberry Pi'en. Controllerens betjeningspanel allokeres som et fysisk betjeningspanel. Vuggesystem allokeres som en sensorkreds, en motorkreds og en PSoC4 til regulering og kommunikation med Controller. Power allokeres som et forsyningsbatteri samt en spændingsregulerende kreds og et kredsløb til at monitorere batteriet.

3.3 Grænseflader

I dette afsnit beskrives specifikke hardware grænseflader og protokoller for kommunikation mellem systemets moduler. Disse grænseflader er således modul-udviklernes kontrakt for fyldestgørelse af modulets funktionalitet.

3.3.1 Fysiske forbindelser

Der er valgt at benytte USB kabler til forbindelse mellem Controller og Intelligent Lydmonitor, samt til forbindelsen mellem Controller og Vuggesystem. USB kablerne der benyttes er standart AB kabler, med et USB A stik i den ene ende samt USB B stik i den anden ende. Controlleren skal have USB type B hun stik, mens den Intelligente Lydmonitor samt Vuggesystemet skal benytte USB A hun stik.

Strømmen til Controller, Intelligent Lydmonitor samt til Vuggesystemet leveres fysisk via et 2,5 mm DC stik.

3.3.2 Grænseflade mellem Intelligent Lydmonitor og Controller

Kommunikationen mellem Intelligent Lydmonitor og Controller foregår via 2 signal ledere, der forbindes til 2 GPIO'er på Controller. Den intelligente Lydmonitor opdaterer via denne forbindelse BABYCON niveauet.

Kommunikationsprotokol

Tabellerne 3.2 til 3.5 viser kommunikationsprotokollen for 2WireTTL forbindelsen mellem Intelligent Lydmonitor og Controller. De 2 GPIO porte på controlleren repræsenteres af hhv MSB og LSB, som mest og mindst betydende bit af 2WireTTL forbindelsen, som skal være aktiv-høj.

Tabel 3.2. BABYCON1

	2WireTTL_MSБ	2WireTTL_LSB
Bit værdi	0	1

Tabel 3.3. BABYCON2

	2WireTTL_MSБ	2WireTTL_LSB
Bit værdi	1	0

Tabel 3.4. BABYCON3

	2WireTTL_MSБ	2WireTTL_LSB
Bit værdi	1	1

Tabel 3.5. Fejl fra Intelligent Lydmonitor

	2WireTTL_MSB	2WireTTL_LSB
Bit værdi	0	0

2WireTTL forbindelsen har som ovenstående tabeller 3.2 til 3.5 viser 4 kombinationsmuligheder. 3 af dem til de 3 BABYCON niveauer, samt en til fejlindikation fra den Intelligentes Lydmonitors side.

3.3.3 Grænseflade mellem Vuggesystem og Controller

Kommunikationen mellem Vuggesystemet og Controllen foregår via I2C. Der benyttes 5 skifтерegistre til kommunikationen, disse registre beskrives i kommunikationsprotokollen herunder.

Kommunikationsprotokol

Forbindelserne ud og ind af dette delsystem er samlet i en I2C bus, som er beskrevet på næste side...

Tabel 3.6. Specifikation af I2C grænseflade

I2C Adresse: 0b1111000X (Write: 0xF0, Read: 0xF1)				
I2C Frekvens: 100kHz				
Reg#	Navn	Type	Beskrivelse	Startværdi
0x00	ID	R	Indeholder et id som kan benyttes til at identificere denne enhed, eller til at teste forbindelsen til denne.	0xFB
0x01	ON_OFF	(R/W)	Dette register benyttes til at tænde og slukke for systemet. Skrives et nul til dette register begynder systemet at lukke ned. Strømmen til systemet bør ikke afbrydes før SD_RDY i status registeret er skiftet til et. Hvis systemet er tændt indeholder registeret en værdi forskellig fra 0.	0xFF
0x02	Frekvens	R/W	Værdien i dette register styrer frekvensen hvormed der vugges. Område: 0 Hz = 2,550 Hz, 1 LSB = 10 mHz.	0x00
0x03	Vinkeludsving	R/W	Værdien i dette register kontrollerer størrelsen af vuggens udsving i grader. Område: +/- 12,75°, 1 LSB = 0,05°.	0x00
0x04	Status	R	Indeholder en bitsekvens som indikerer systemets status. Registeret indeholder følgende: [ERR STALL END_STP SD_RDY X X X X] et 1 tal indikerer en fejl	0b0000XXXX
ERR		Indikerer at der er opstået en fejl i systemet. Vuggesystemet er deaktivert indtil der er blevet genstartet. Kendes årsagen til fejlen indikeres denne i STALL og END_STP		
STALL		Indikerer at systemet har været ude af stand til at drive motoren, formentlig pga. for stor belastning		
END_STP		Indikerer at vuggen har ramt en af sine mekaniske yderpositioner.		
SD_RDY		Indikerer at systemet er klar til at få afbrudt strømmen		

3.3.4 Grænseflade mellem Controllers servere og ekstern internet enhed

Fysiske porte

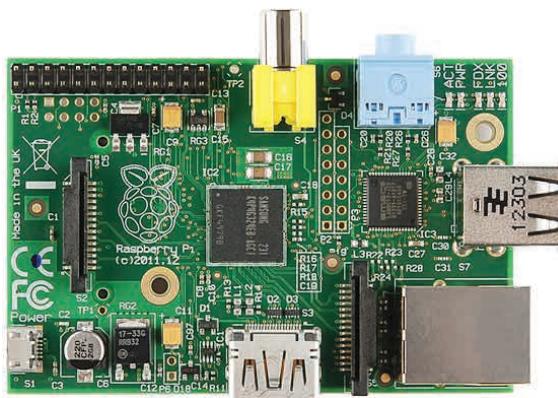
Raspberry Pi'en der er Controlleren hovedenhed, er forbundet til netværket via en trådløs Wi-Fi dongle. Netværket består af en router der er koblet op til internettet.

Kommunikationsprotokol

Standard Ethernet TCP/IP protokol

Controller 4

Controllerafsnittet dækker over dokumentation for den enhed der styrer hele Baby Watch. Controlleren er implementeret på en Raspberry Pi Model B, som er en lille linux platform med en GPIO port og ethernet funktioner. På Raspberryen skal Baby Watch hovedprogrammet køre, samt en hjemmeside server og en e-mail klient.



Figur 4.1. Raspberry Pi Model B

4.1 Systemarkitektur

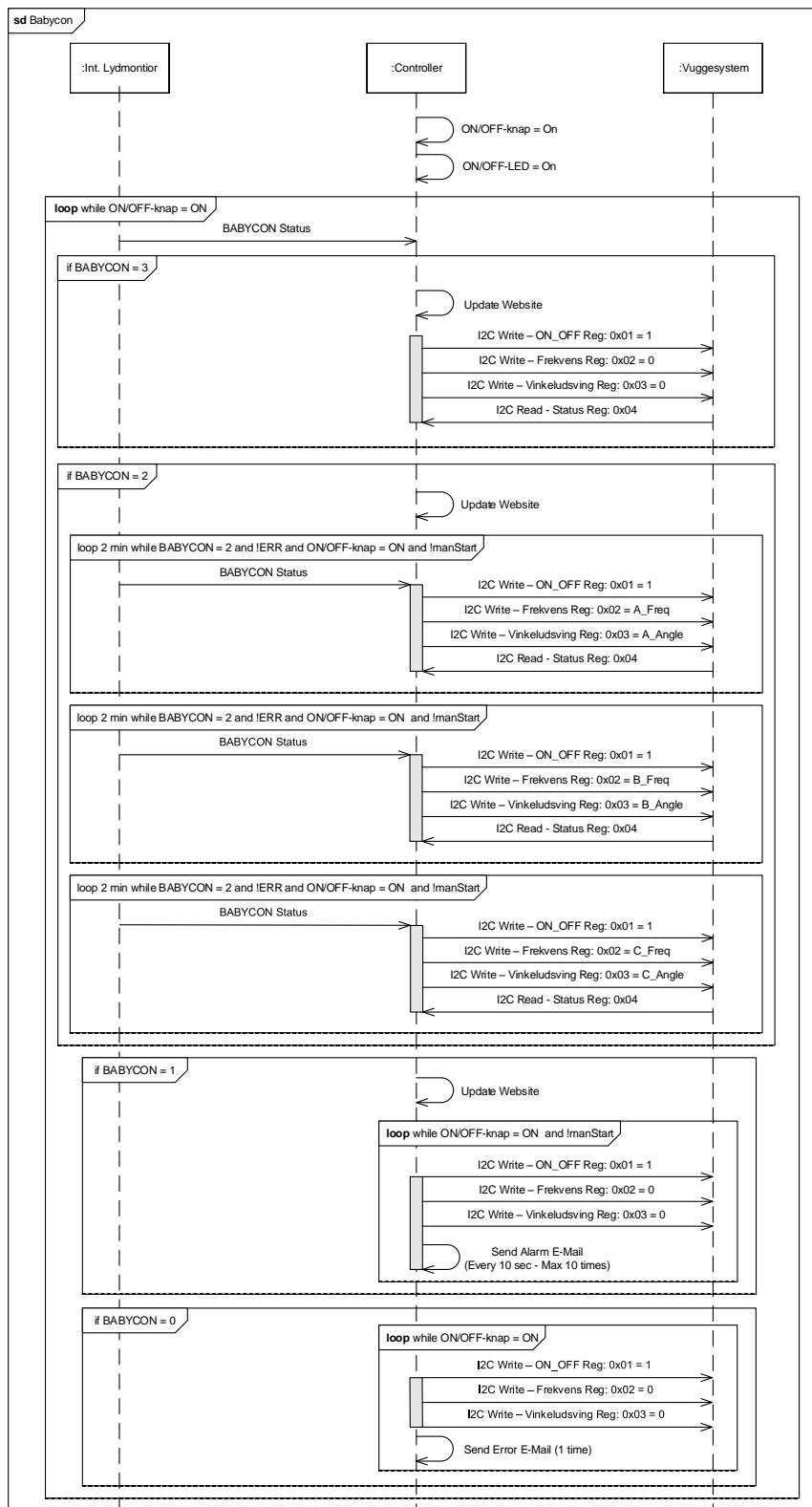
I dette afsnit beskrives systemarkitekturen for Controller.

Softwarearkitekturen er beskrevet i sekvensdiagrammer der giver et indblik i hvordan mainprogrammet interagerer med den intelligente lydmonitor og vuggesystemet.

Hardwarearkitekturen beskriver den fysiske grænseflade til den intelligente lydmonitor og vuggesystemet.

4.1.1 Softwarearkitektur

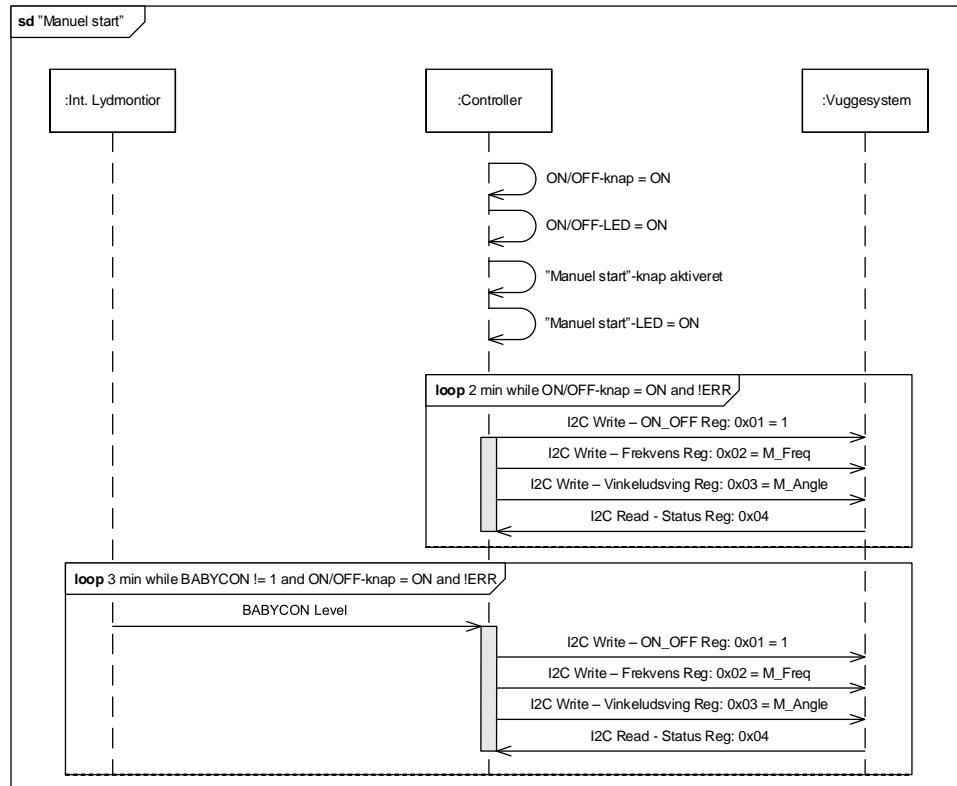
Sekvensdiagram (BABYCON-niveauer)



Figur 4.2. Sekvensdiagram for Babycon niveauer

Sekvensdiagrammet for BABYCON-niveauerne beskriver hvordan Controlleren reagere på BABYCON-niveauer fra Intelligent lydmonitor. Samt hvad der kommunikeres ud til Vuggesystemet. Det er tænkt som et stort loop der hele tiden kigger på variabler og handler på disse. BABYCON har så 3 niveauer og en fejtilstand 0, som den reagerer på. BABYCON2 har sine egne loops at switche imellem, disse kan dog afbrydes af fejl og slukning af systemet.

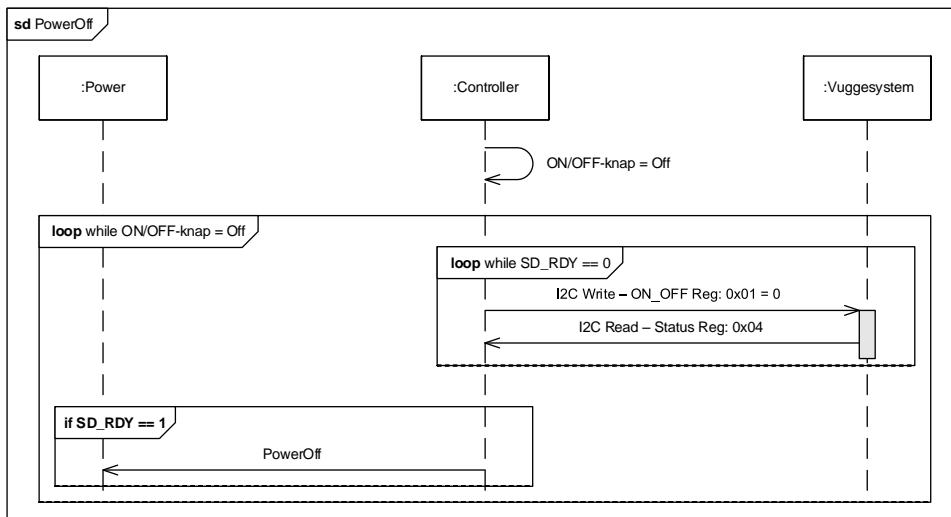
Sekvensdiagram ("Manuel start")



Figur 4.3. Sekvensdiagram for "Manuel Start"

"Manuel start" skal køre et loop i 5 min. fordelt på hhv. 2 min. "forced" loop hvor der vugges uden hensyntagen til BABYCON-niveau og i 3 min. hvor BABYCON1 kan afbryde vugningen. "Manuel start" kan selvfølgelig afbrydes af et Power OFF eller fejl i vuggesystemet.

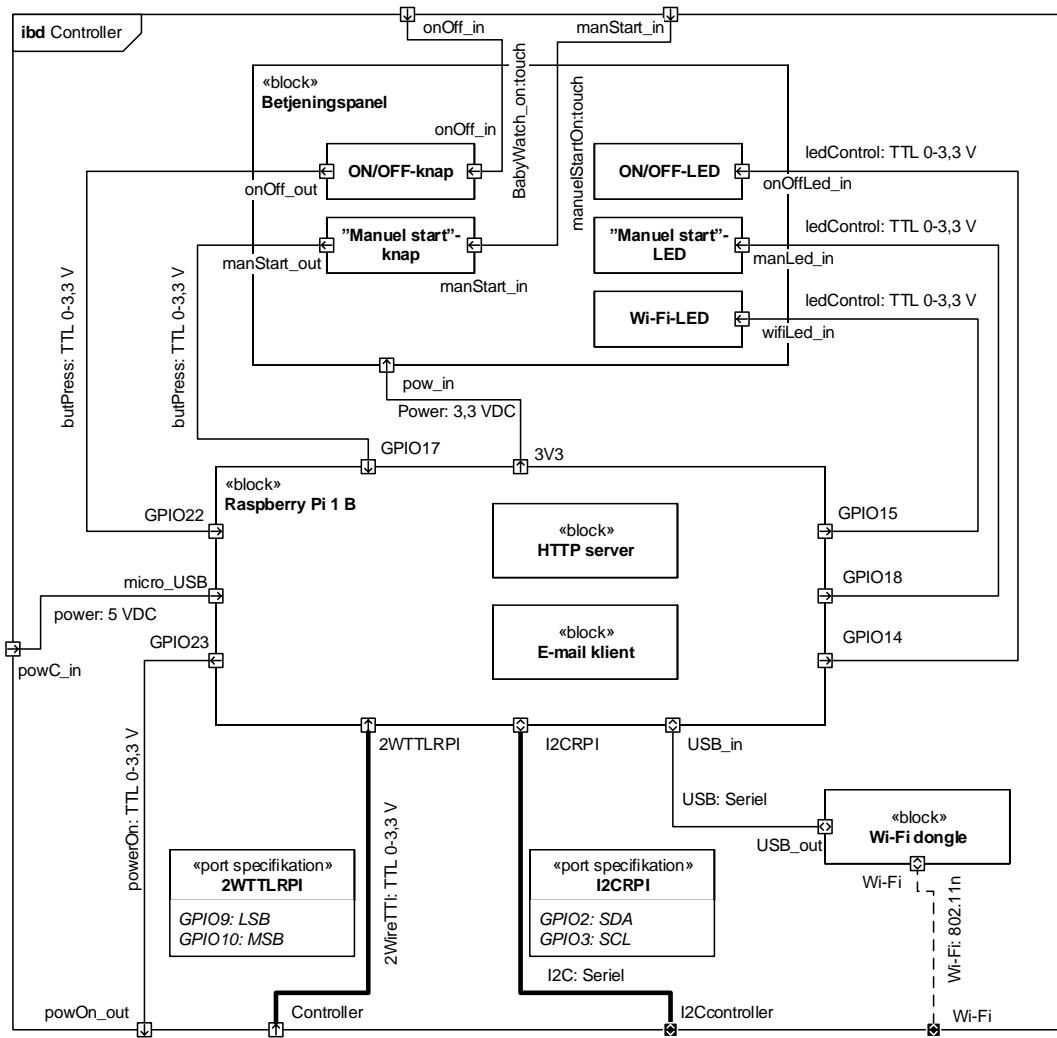
Sekvensdiagram (Power Off)



Figur 4.4. Sekvensdiagram for nedlukning

Power off skal starte et loop der kigger på om vuggesystemet er klar til at lukke ned. Dvs. der sendes et power off signal til Vuggesystemet og ventes på et "shutdown ready" fra samme. Når signalet er modtaget, kan der afbrydes for strømmen til Vuggesystemet og Int. lydmonitor.

4.1.2 Hardware arkitektur



Figur 4.5. IBD diagram for Controller

Figur 4.5 viser IBD'et for Controlleren. Signaltabellen ?? angiver specificerer de interne forbindelser. De 2 lidt bredere sorte forbindelser i IBD'et angiver en forbindelse der består af flere ledere, frem for de alm. forbindelser der udgør en enkelt ledet forbindelse.

Controlleren består af 2 dele

Raspberry Pi Linux baseret computer der styrer hele systemet.

Betjeningspanel Panel med 3 lysdioder og 2 trykknapper til brugerinteraktion.

4.1.3 Signaltabel

I signaltabellen beskrives hvilke signaler der sendes imellem blokkene i Controller.

Tabel 4.1: Signaltabel for Controller

Signal-navn	Type	Port 1	Port 2	Kommentar
manStartOn	touch	Betjeningspanel (manStart_in)	Babypasser (touch)	Babypasser der laver et tryk på knappen
babyWatchOn	touch	Betjeningspanel (onOff_in)	Babypasser (touch)	Babypasser der laver et tryk på knappen
power	5 VDC	RaspberryPi (micro_USB)	Controller (powC_in)	5 V forsyning til controller
power	3,3 VDC	RaspberryPi (3V3)	Betjeningspanel (pow_in)	3,3 V forsyning til Betjeningspanel
powerOn	TTL 0-3,3 VDC	RaspberryPi (GPIO23)	Controller (powOn_out)	Benyttes til at tænde og slukket for strømforsyningen
2WireTTL	TTL 0-3,3 VDC	RaspberryPi (2WTTLRPI) GPIO9: LSB GPIO10: MSB	Controller (I2CController)	Seriell kommunikation
I2C	Seriell	RaspberryPi (I2CRPI) GPIO0: SDA GPIO1: SCL	Controller (Controller)	BABYCON niveau
USB	Seriell	RaspberryPi (USB_in)	Wi-Fi dongle (USB_out)	Seriell kommunikation for Wi-Fi forbindelse
ledControl	TTL 0-3,3 VDC	RaspberryPi (GPIO14) RaspberryPi (GPIO15) RaspberryPi (GPIO18)	Betjeningspanel (onOffLed_in) Betjeningspanel (manLed_in) Betjeningspanel (wifiLed_in)	Logisk signal til at styre LED på Betjeningspanel
butPress	TTL 0-3,3 VDC	RaspberryPi (GPIO17) RaspberryPi (GPIO22)	Betjeningspanel (onOff_out) Betjeningspanel (manStart_out)	Logisk signal til kontrol af knapper på betjeningspanel

Figur 4.6 viser GPIO porten på Raspberry Pi'en. Der er desuden tilføjet enkelte portes specielle muligheder (SPI, I2C mm.), hvis de er tilstede.



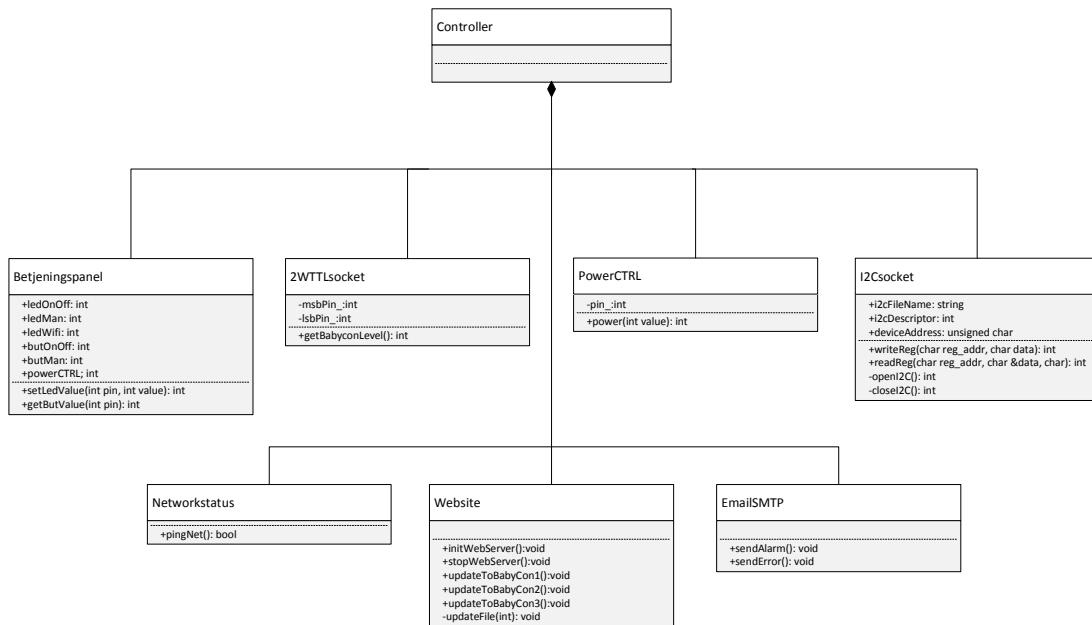
Figur 4.6. Raspberry Pi model Bs GPIO port

4.2 Design

4.2.1 Softwaredesign

I dette afsnit beskrives hvordan Controllerens software er designet med klassediagrammer

Klassebeskrivelser



Figur 4.7. Oversigt over klassediagrammerne for Controller

Betjeningspanel

Ansvar: At styre forbindelserne til det hardwarenære på den fysiske controller boks

Funktion	<code>int setLedValue(int pin , int value)</code>
Parametre	Modtager et GPIO pin-nummer og en værdi 0 for ON og 1 for OFF
Returværdi	0 ved succes. Minus værdi ved fejl
Beskrivelse	Bruges til at tænde og slukke for LED dioderne på controller

Funktion	<code>int getButValue(int pin) const</code>
Parametre	Modtager et GPIO pin-nummer
Returværdi	Status på knaptryk
Beskrivelse	Bruges til at læse fra trykknapper på controller

Attribut	<code>int ledOnOff_</code>
Beskrivelse	Attribut til at holde GPIO nummer for On/Off LED

Attribut	<code>int ledMan_</code>
Beskrivelse	Attribut til at holde GPIO nummer for Manuelstart LED

Attribut	<code>int ledWifi_</code>
Beskrivelse	Attribut til at holde GPIO nummer for Netværksstatus LED

Attribut	<code>int butOnOff_</code>
Beskrivelse	Attribut til at holde GPIO nummer for On/Off knappen

Attribut	<code>int butMan_</code>
Beskrivelse	Attribut til at holde GPIO nummer for "Manuelstart"-knap

EmailSMTP

Ansvar: At sende Alarm og Fejl emails til Babypasser direkte fra main programmet

Funktion	<code>void sendError()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Sender en Email med en fejlmeldelse til Babypasser

Funktion	<code>void sendAlarm()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Sender en Email med en alarm besked til Babypasser

GPIOsocket

Ansvar: At oprette et GPIO filarkiv

Funktion	<code>int gpioExport(int pin)</code>
Parametre	GPIO pin nummer
Returværdi	0 ved succes. Minus værdi ved fejl
Beskrivelse	Opretter et fil arkiv for det modtagne pin-nummer

Funktion	<code>int gpioUnexport(int pin)</code>
Parametre	GPIO pin nummer
Returværdi	0 ved succes. Minus værdi ved fejl
Beskrivelse	Fjerner filarkivet for det modtagne pin-nummer

Funktion	<code>int gpioDirection(int pin , int dir)</code>
Parametre	GPIO pin nummer og pin retning
Returværdi	0 ved succes. Minus værdi ved fejl
Beskrivelse	Sætter retningen for GPIO pin, INPUT(0) eller OUTPUT(1)

I2Csocket

Ansvar: At kommunikere over I2C

Funktion	<code>int writeReg(unsigned char reg_addr , unsigned char data)</code>
Parametre	Register adresse og data til skrivning
Returværdi	0 ved succes. Minus værdi ved fejl
Beskrivelse	Skriver data til et register på en given enhed

Funktion	<code>int readReg(unsigned char reg_addr , unsigned char &data)</code>
Parametre	Register adresse og en data adresse til at gemme læst data i
Returværdi	0 ved succes. Minus værdi ved fejl
Beskrivelse	Læser data fra et register på en given enhed

Funktion	<code>int openI2C()</code>
Parametre	Ingen
Returværdi	0 ved succes. Minus værdi ved fejl
Beskrivelse	Åbner I2C forbindelsen til en enhed i constructor

Funktion	<code>int closeI2C()</code>
Parametre	Ingen
Returværdi	0 ved succes. Minus værdi ved fejl
Beskrivelse	Lukker I2C forbindelsen til en enhed i destructor

Attribut	<code>std::string i2cFileName</code>
Beskrivelse	Enhedsnavn på Raspberry Pi ”/dev/i2c-0” eller ”/dev/i2c-1”

Attribut	<code>int i2cDescriptor</code>
Beskrivelse	Fil descriptor til åbning af I2C forbindelsen

Attribut	<code>unsigned char deviceAddress</code>
Beskrivelse	Attribut til at holde Enheds adressen på den enhed der skal kommunikeres med

Networkstatus

Ansvar: At aflæse om der er forbindelse til netværket og internettet

Funktion	<code>bool pingNet()</code>
Parametre	Ingen
Returværdi	True ved forbindelse ellers false
Beskrivelse	Pinger google's offentlige DNS server, og hvis den fejler pinger den en anden offentlig DNS server for at dobbelttjekke

TWTTLsocket

Ansvar: At aflæse status fra Intelligent Lydmonitor

Funktion	<code>int getPinValue(int pin) const</code>
Parametre	GPIO pin nummer
Returværdi	Pin værdi
Beskrivelse	Bruges til læsning af MSB og LSB bit fra den Intelligent Lydmonitor

Funktion	<code>int getBabyconLevel(void)</code>
Parametre	Ingen
Returværdi	BABYCON niveau 0, 1, 2 eller 3
Beskrivelse	Udlæsning af BABYCON niveau

Attribut	<code>int msbPin_</code>
Beskrivelse	Attribut til at holde GPIO nummer for MSB-bit

Attribut	<code>int lsbPin_</code>
Beskrivelse	Attribut til at holde GPIO nummer for LSB-bit

Website

Ansvar: At styrer HTTP serveren med Baby Watch websiden

Funktion	<code>void initWebServer()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Initierer Baby Watch HTTP-webserveren til BabyCon 3 niveau

Funktion	<code>void stopWebServer()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Lukker Baby Watch HTTP-webserveren

Funktion	<code>void updateToBabyCon1()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Opdatere Baby Watch HTTP-webserveren til BabyCon 1 niveau

Funktion	<code>void updateToBabyCon2()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Opdatere Baby Watch HTTP-webserveren til BabyCon 2 niveau

Funktion	<code>void updateToBabyCon3()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Opdatere Baby Watch HTTP-webserveren til BabyCon 3 niveau

Funktion	<code>void updateFile(int BCon)</code>
Parametre	int Bcon angiver en værdi fra 1 til 3 som svarer til en BabyCon værdi
Returværdi	Ingen
Beskrivelse	Opdatere Baby Watch filer efter anvisninger fra BCon værdi

4.2.2 Hardwaredesign

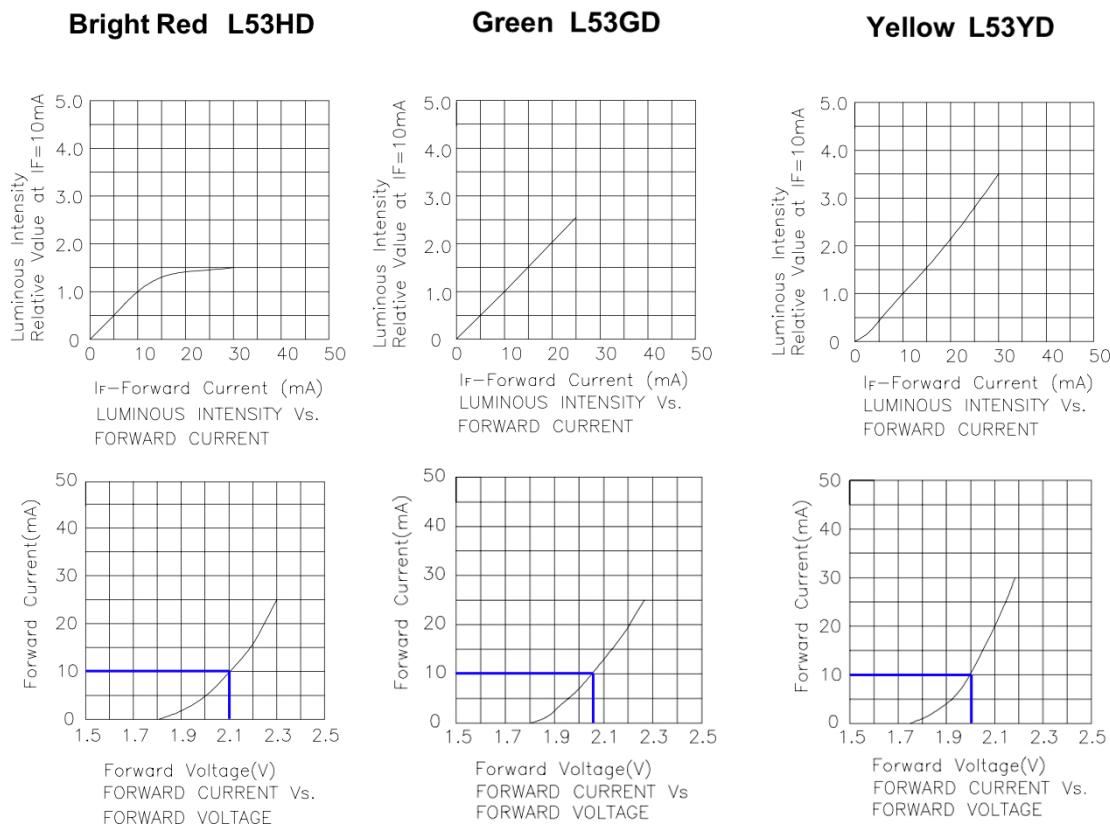
Controlleren består af et betjeningspanel samt en Raspberry Pi der kontrollerer betjeningspanelet og styrer forbindelserne til hhv. den Intelligente Lydmonitor og Vuggesystemet.

Betjeningspanelet består som beskrevet af 3 lysdioder. Én grøn, én gul og én rød der hhv. indikerer at Baby Watch er tændt/slukket, at Manuel start er aktiveret/deaktiveret samt Wi-Fi status. Ydermere består betjeninsapanelet af 2 knapper, én til at tænde/slukke for Baby Watch systemet og én til at aktivere Manuel start.

Betjeningspanel

Til lysdioderne benyttes 5mm dioder fra komponentrummet:

- Grøn 5mm LED: KINGBRIGHT L-53 GD
- Gul 5mm LED: KINGBRIGHT L-53 YD
- Rød 5mm LED: KINGBRIGHT L-53 HD



Figur 4.8. Udsnit af datablad for KINGBRIGHT L53 HD, GD og YD

Øverst på figur 4.8 ses information om lysintensitet som funktion af strømmen og ens for alle tre dioder er at "Relative Value at $IF = 10 mA$ ". Strømmen vælges derfor til 10 mA for at der herved er samme lysintensitet i de tre dioder. Nederst ses strømmen som funktion af spændingen over dioderne. Den indtegnete blå linje på hver af 3 kurver angiver spændingsfaldet over hver diode når strømmen er sat til 10 mA. Ud fra aflæsning på kurverne beregnes for modstandene for dioderne.

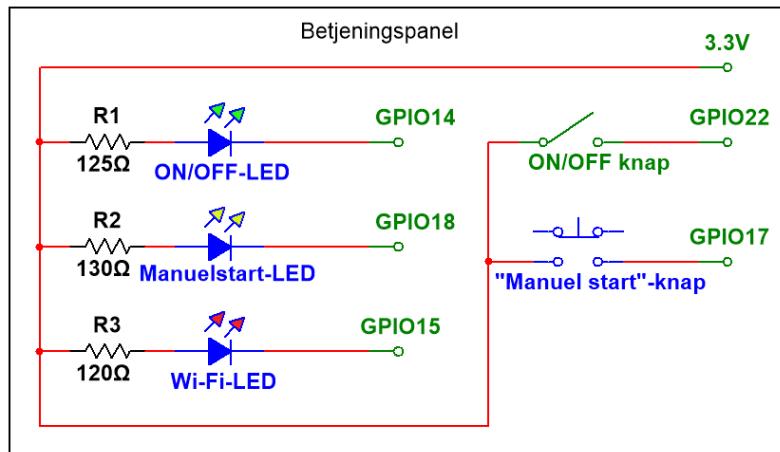
$$R_{L53HD(RED)} = \frac{3,3V - 2,1V}{10mA} = \frac{1,2V}{10mA} = 120\Omega$$

$$R_{L53GD(GREEN)} = \frac{3,3V - 2,05V}{10mA} = \frac{1,25V}{10mA} = 125\Omega$$

$$R_{L53YD(YELLOW)} = \frac{3,3V - 2V}{10mA} = \frac{1,3V}{10mA} = 130\Omega$$

Figur 4.9. Formodstandsberegninger for de 3 dioder

Ud fra modstandsberegningerne i figur 4.9 er kredsløbsdiagrammet, se figur ??, opbygget. De to knapper for ON/OFF samt Manuel start er medtaget sammen med deres GPIO porte på Raspberry Pi model b.



Figur 4.10. Kredsløbsdiagram for Betjeningspanel

Raspberry Pi'en forsyner LED dioderne med strøm. Der er taget hensyn til at Raspberryens sink current har et maksimum på 16 mA pr output pin samt at 3,3 V udgangen kan levere 50 mA. Dioderne kræver samlet 30 mA i følge designet og det stemmer fint overens med hvad Raspberryen kan levere.

Tabellen herunder angiver opsætningen af GPIOerne på Raspberry Pi'en for at ovenstående diagram kan fungere.

Tabel 4.2: Oversigt over GPIO pins og deres funktioner

GPIO	Opsætning	Kommentar
GPIO14	Indgang sættes aktiv-lav	Styrer ON/OFF-LED
GPIO18	Indgang sættes aktiv-lav	Styrer Manuelstart-LED
GPIO15	Indgang sættes aktiv-lav	Styrer Wi-Fi-LED
GPIO22	Indgang sættes aktiv-høj, med intern pull-down modstand	Tænd/sluk signal for Baby Watch
GPIO17	Indgang sættes aktiv-høj, med intern pull-down modstand	Aktivering af "Manuel-start"

Kommunikation til/fra Intelligent Lydmonitor og Vuggesystem

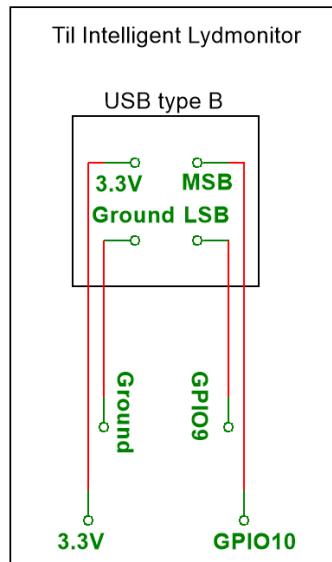
I de to følgende afsnit vil kommunikationen med hhv Intelligent Lydmonitor og Vuggesystem blive beskrevet.

Kommunikation med Intelligent Lydmonitor

Kommunikationen med den Intelligente Lydmonitor foregår som beskrevet i systemarkitekturen med 2 signalledere (2WireTTL). Herved opnås der fire kombinationsmuligheder, en for hvert BABYCON niveau samt en til indikation af fejl.

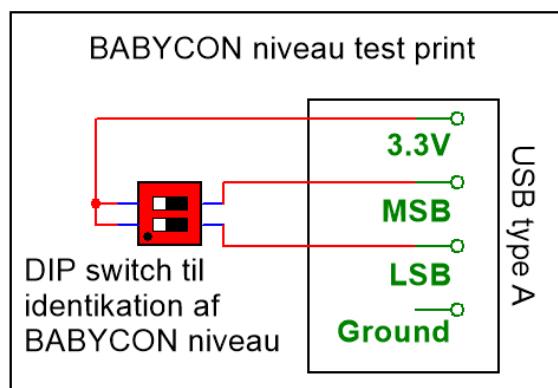
GPIO9 og GPIO10 benyttes til denne to ledet forbindelse til Intelligent Lydmonitor, hvor GPIO9 er LSB delen og GPIO10 er MSB delen.

Til den fysiske forbindelse designes der efter at benytte et USB kabel. Controllerens del af forbindelsen er et USB hun type B stik. Se figur 4.11. Figuren viser at 3,3 V er en del af USB stikket, det er kun af testhensyn, for at testprintet vil fungere med dipswitchen, se figur 4.12.



Figur 4.11. USB type B for kommunikation med Intelligent Lydmonitor

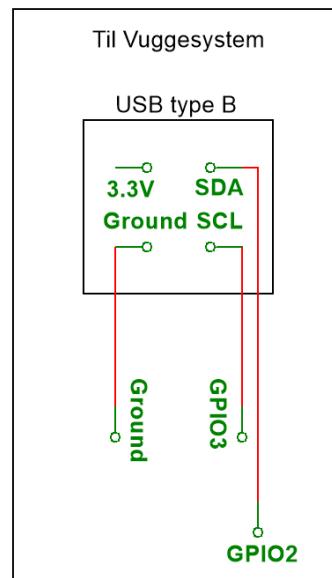
Der designes desuden et testprint se figur 4.12



Figur 4.12. Testprint med dipswitch til at angive BABYCON niveau

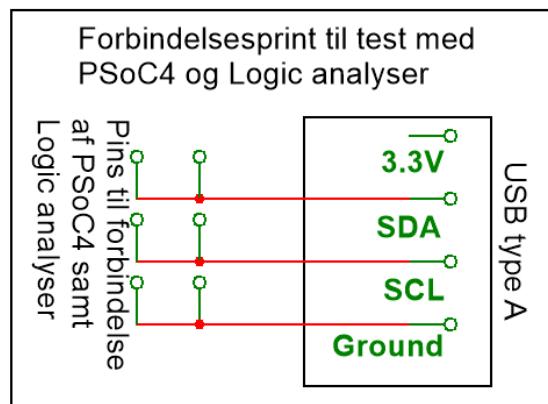
Kommunikation med Vuggesystem

I2C forbindelsen består som beskrevet af en clock (SCL) og en datalinje (SDA) samt en fælles ground forbindelse. Ligeledes med med kommunikationen til den Intelligente Lydmonitor benyttes et USB type B stik.



Figur 4.13. USB type B for kommunikation med Vuggesystem

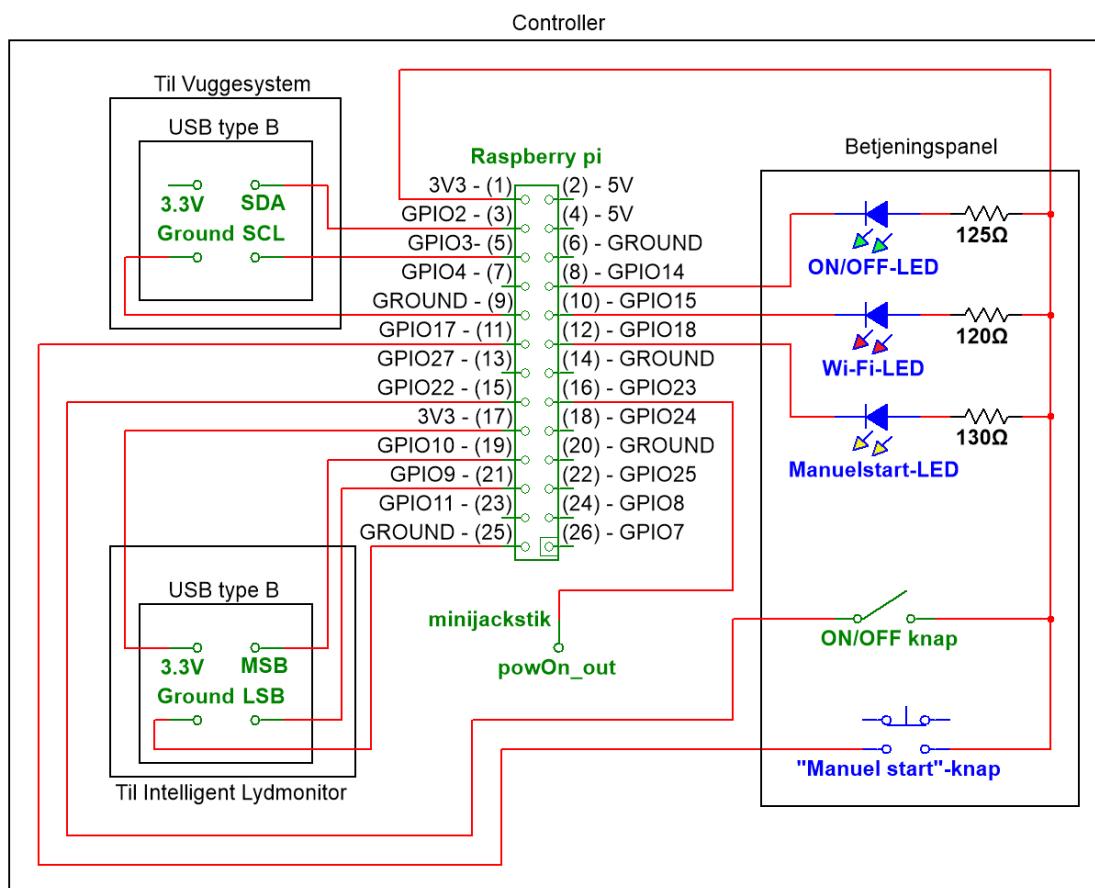
Der designes et testprint således at controlleren kan testes med et simpelt PSoC4 program og datakommunikationen kan verificeres vha. en Logic Analyser.



Figur 4.14. Testprint til I2C forbindelsen, med pins til PSoC4 samt Logic Analyser

Samlet HW-design af Controller

Figur 4.15 viser samlet kredsløbsdiagram for Controlleren. Betjeningspanelets knapper og dioder, USB stik til kommunikation med Intelligent Lydmonitor og Vuggesystem samt Raspberry Pi'ens GPIO socket.

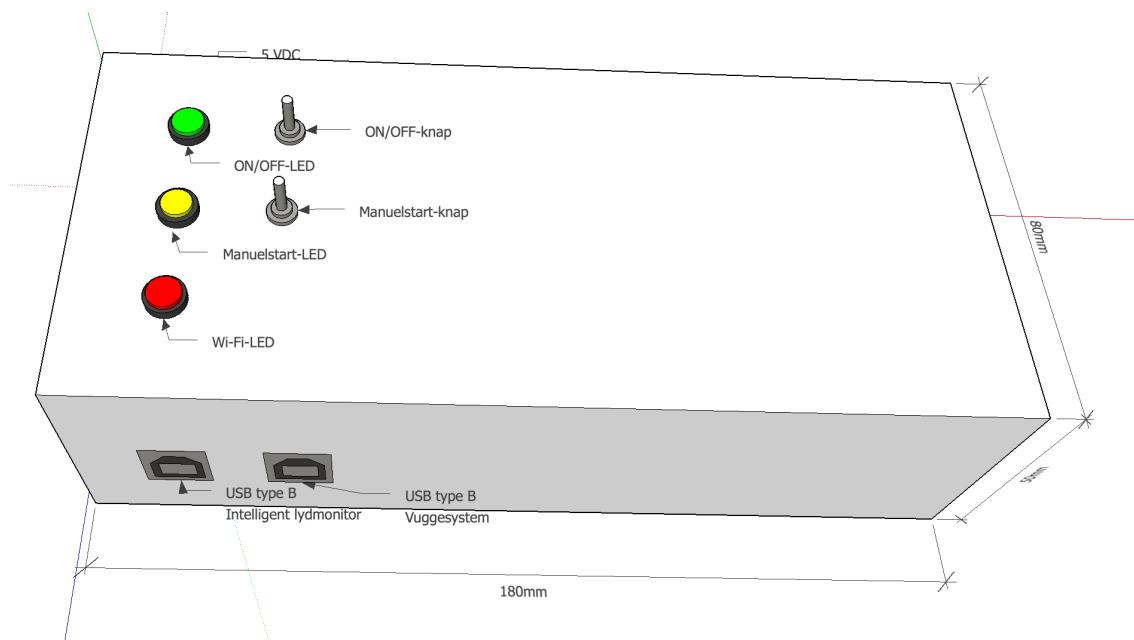


Figur 4.15. Samlet kredsløbsdiagram for Controller

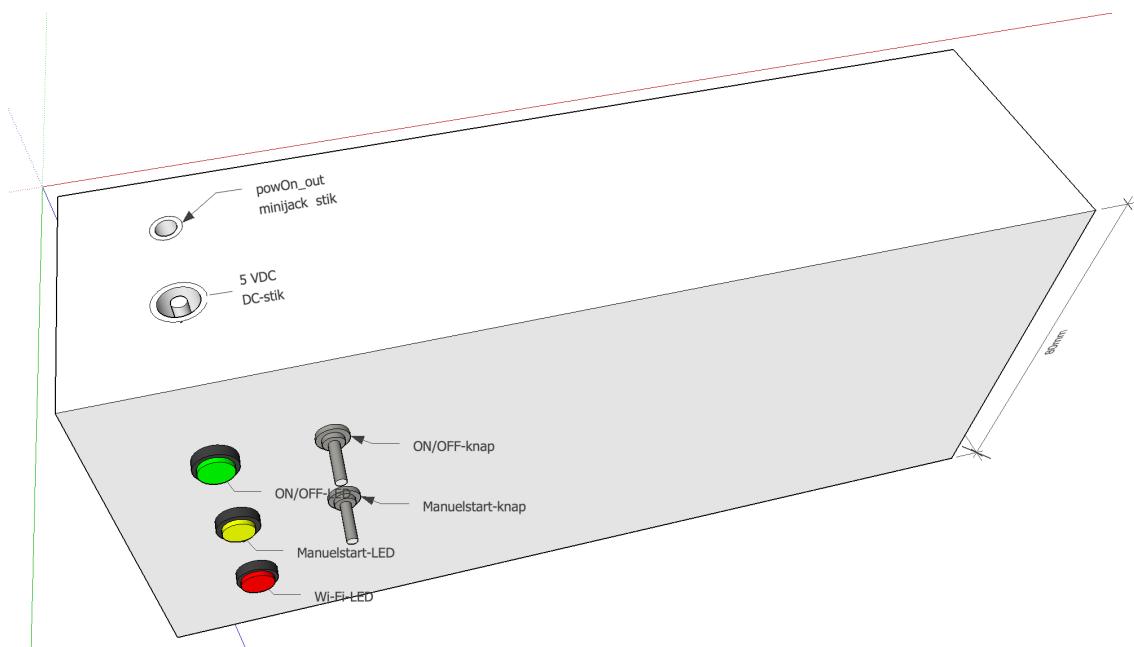
Fysisk design af Controller

Der designes en kasse på 18x8x5 cm hvori Raspberry Pi og Betjeningspanelets kredsløb skjules i. Babypasseren skal herved kun ha adgang til de tre LED dioder, de to kontakter, de 2 USB type B stik til kommunikation med hhv. Intelligent Lydmonitor og Vuggesystem, 5 VDC forsyningen af controlleren samt et minijackstik som er det signal der står for at styre forsyningen.

Figurene 4.16 og 4.17 viser kassen, med tilhørende LEDs, kontakter, stik samt mål på selve kassen. Kassens materiale forventes at være sort Forex.



Figur 4.16. Arbejdstegning - Betjeningspanel og USB stik



Figur 4.17. Arbejdstegning - Betjeningspanel og forsyningsstik

4.3 Implementering

4.3.1 Softwareimplementering

Cross Compiler

For at gøre udviklingsprocessen hurtigere og for at kunne debugge koden blev der tidligt i forløbet lavet et ønske om at kunne krydskompilerer til Raspberry Pi (Raspbian OS). Det har ikke været uden en del problemer, da der i starten gerne ville kodes fra en Windows

platform. Dette er muligt og blev også realiseret, men da der blev introduceret eksterne biblioteker til brug i Raspberry Pi, blev det meget svært at få kompileret og inkluderet disse.

Derfor blev der valgt en løsning om at bruge Eclipse til Linux. Og da Raspberry Pi kører på en Linux platform, gjorde det tingene meget lettere.

For at kunne krydskompilere er det nemmeste at bruge en speciel version af Eclipse, som er modificeret til formålet, efter lidt undersøgelser blev der fundet frem til "Eclipse Kepler Release for C/C++ Developers"¹ og en "Official Raspbian (armhf) cross compiling toolchain"² med kompilere til Raspberry Pi (Raspbian OS). Eclipse skal lige sættes op, så der bruges `arm-linux-gnueabihf-` som kompiler prefix i projektet.

I "Remote System Explorer" i Eclipse laves en ny SSH forbindelse til Raspberry Pi'ens IP, så der fra programmet kan oprettes direkte forbindelse til Raspberry Pi'en. Derefter i Debug Configurations sættes "Connection" til den oprettet enhed og stien på Raspberry Pi'en hvor man ønsker at køre sit program fra. I dette tilfælde `/home/pi/Debug/`

GPIO

På Controllerboksen sidder der 2 knapper og 3 LED-dioder, disse skal kunne tilgås fra programmet vha. GPIO benene på Raspberry Pi'en. Der blev hurtigt fundet frem til WiringPi biblioteket³ til formålet. Dette er et bibliotek der gør det meget nemt at håndtere GPIO benene på Raspberry Pi'en.

Der blev dog valgt at lave 3 GPIOsocket funktioner der tager sig af at skrive og læse fra det fil arkiv der oprettes når gpio funktionen bliver kaldt. Funktionerne består en `Export`, en `Unexport` samt en `Direction` funktion som hhv. opretter og nedlægger et filbibliotek til det ønskede ben på Raspberry Pi'en og sætte retningen på benet (input/output).

I2C

I2C forbindelsen til Vuggesystemet er der også brugt interne funktioner fra WiringPi biblioteket. Der laves et system kald til et script der opretter en I2C forbindelse. Scriptet består af kun 1 linje `gpio load i2c`. På Raspberry Pi Model B, oprettes der en filnode `/dev/i2c-1` som der benyttes i `I2Csocet` klassen.

Inspirationen til I2Csocket klassen er taget fra Hertaville bloggen⁴ der skulle bruges en åben kode, som kunne redigeres om nødvendigt, derfor er klassen blevet oprettet i stedet for at bruge WiringPiI2C.h biblioteket.

Mail SMTP

Mail SMTP er 2 shell scripts der kaldes med et systemkald. Der er ikke noget fejlcheck på

...

¹<https://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/keplerr>

²<https://github.com/raspberrypi/tools>

³<http://wiringpi.com/>

⁴<http://hertaville.com/2013/04/01/interfacing-an-i2c-gpio-expander-mcp23017-to-the-raspberry-pi-using-c/>

Netværksstatus

Baby Watch kræver at der er noget synligt, der indikerer at der er en fejl på netværket, nærmere bestemt internet forbindelsen, da det ikke er muligt at sende en fejl e-mail når internet forbindelsen er afbrudt. Det bliver indikeret af en rød LED på forsiden af Controllerboksen.

For at kontrollere internet forbindelsen er der oprettet en tråd i programkoden der med et interval pinger googles DNS server for svar og returnere **true** eller **false** alt efter om der er forbindelse eller ej. For at gøre det ekstra sikkert er der indført en ekstra ping adresse hvis googles DNS ikke svarer, dvs. der er et dobbelttjek på internet forbindelsen.

Der blev undersøgt lidt på hvordan man pinger inde fra et C++ program. Løsningen blev at lavet et system kald til den indbyggede ping funktion i Linux. Når der laves et system kald til ping med kommandoen `system("ping -w 2 ip-adresse")` returneres 0 ved svar og et positivt tal forskellig fra 0 ved intet svar. På den måde er det let kontrollere internet forbindelsen. -w flaget der bliver brugt at en deadline for svar i sekunder, dette er brugt for at undgå at programmet står i unødig lang tid og prøver at få svar.

HTTP Server

Baby Watch webserveren er implementeret med en microframework Flask webserver udviklet af Armin Ronacher(REFERENCE). Servertypen hører under en fri BSD license(REFERENCE) og er igennem sin simple og lette struktur et ideel valg til en webserver integreret på et embedded-system med begrænset CPU-kraft.

En Flask webserver består af en python-template der styrer en eller flere html-templates. Igennem python-templetten kan diverse billeder og tekst-strenge styres og opdateres løbende på html siden. Webserveren sørger selv for at sende de nødvendige data til en given bruger(browser) som tilgår websiden.

Beskrivelse af python-templetten og dens funktionaliteter

En flask server python templetten er opbygget på samme måde som andre højniveau sprog. Først kaldes de nødvendige biblioteker ved hjælp af `import`. I dette tilfælde bruges "flask", "render template", "logging" og "datetime"(Den næst sidste er til at skjule flask serverudskrifter og den sidste er til opdateringstidspunktet).

Herefter oprettes et Flask server objekt kaldet app vha. af kommandoen: `app = Flask(name)` Objektet fyldes med funktionaliter til blandt andet at vise opdateringstidspunktet og til at ændre tekst og billeder på hjemmesiden. Objektet returner til sidst sin templateData ved hjælp af en pointer.

Eksempel på hvordan tekst gemmes i en `string` og efterfølgende gemmes i et flask objekts template data:

```
msg = open ('/root/Flask/static/tilstand.txt', 'r')
babyStateString = msg.read()
templateData = (
    'BabyConState' : babyStateString
)
```

Dette flask objekts data kan nu kaldes fra html-templetten ved at kalde 'BabyConState'. Templetten afsluttes med et kald til flask server objektet der kører det som et main objekt.

Beskrivelse af html-templetten og dens funktionaliteter

En `html` side består i grove træk af en header og en body. Headeren indeholder definitioner og funktioner som styrer de bagvedlæggende handlinger på hjemmesiden. Body-delen kalder disse definitioner og funktioner til at lave det grafiske interface på hjemmesiden.

`BabyWatch.html`s header:

Er implementeret med en overordnet html-funktion som opdaterer hjemmesiden hvert tredje sekund. Herefter defineres fem styles, der henholdsvis styrer toppen, bunden, siderne og et centreret billede på hjemmesiden.

Eksempel på toppen:

```
#header  
background-color:lightblue;  
color:white;  
text-align:center;  
padding:5px;
```

Denne style laver et lyseblå felt med hvid skrift. Skriften centres og der paddes med en margin på 5 pixels rundt om feltet. De andre styles er udarbejdet på samme vis.

Efter styles bliver der defineret to javascript funktioner, en til at hente klokkeslættet `function startTime()` og en til at opdatere hjemmesiden med det pågældende klokkeslæt `function checkTime(i)`. Implementeringen på disse kan ses i bilag (REFERENCE)

`BabyWatch.html`s body:

Body'en består af syv divisioner, en til højreside billede, en til toppen, en til bunden, en til klokkeslættet i venstreside, en til klokkeslættet af seneste opdatering af hjemmesiden, en til centerbilledet og en til den tilhørende tekst.

Eksempel på html-kald af højreside billede:

```
<div id="rightsection" >  
</img>  
</div>
```

Øverst er et kald til rightsection style. `src` laver et specielt kald som flaskserver-templetten kan genkende udfra filnavnet `BabyConState` hvilket indsætter babycon billedet. `Width="100"` definerer størrelsen på billedet vidde. `ALT = "BabyCon"` giver et alias til divisionen som kan bruges til debugging af hjemmesiden.

Eksempel på et html-kald af et java-script:

```
<div id ="leftsection">  
<h2>
```

```
<div id ="txt">
```

På samme måde som ved eksemplet laves der et kald til leftsection style. Herefter sættes en skriftstørrelse med `<h2>`. Tilsidst kaldes javascript-funktionen der opdaterer med det pågældende tidspunkt vha. "txt".

Implementeringen af webserver funktionerne på Controller

Tanken bag implementeringen er at ved henholdsvis oprettelsen og nedlæggelsen af et Website objekt skal `initWebServer()` og `stopWebServer()` kaldes så starten og terminering af flaskserveren følger Website objektets livscyklus.

De tre opdaterings funktioner; `updateToBabyCon1()`, `updateToBabyCon2()` og `updateToBabyCon3()` er implementeret således at de hver især sender en parameter fra 1-3 til `updateFile(int)`.
+`updateFile(int)+` er skrevet sådan at den ved hjælp af en switch case tjekker på input parameteren og herefter opdaterer funktionen de valgte filer ved hjælp af FILE pointere. Disse FILE pointerer bruges til at indlæse filer som angiver Babycon tilstanden og herefter overskrive filerne som flask-serveren anvender til hjemmesiden.

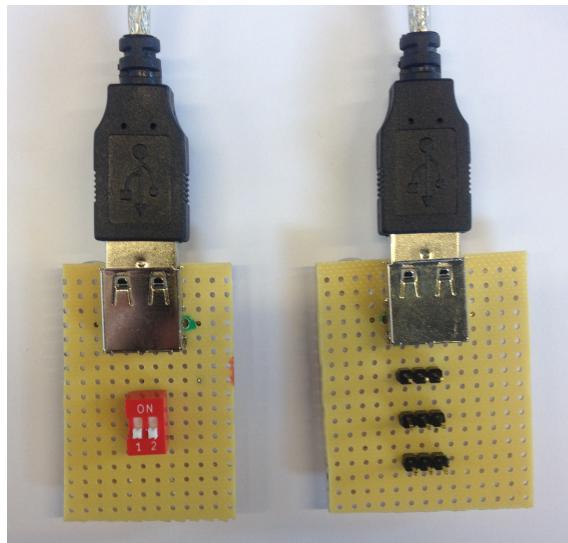
4.3.2 Hardwareimplementering

Implementeringen af Controlleren er udført ved at udarbejde en Forex kasse ud fra arbejdstegningerne, se afsnit 4.2.2. Herefter er de nødvendige hulle boret og LED dioderne isat. Dioderne med tilhørende modstande er monteret på veroboard. Raspberry Pi'en er fast monteret. Alle forbindelser er herefter trukket. Det endelige resultat kan ses på figur 4.18



Figur 4.18. Færdig implementeret Controller

De to testprint der er brugt i forbindelse med debug samt modultest kan ses på figur 4.19



Figur 4.19. De to test prints. Til venstre: BABYCON niveauer (Intelligent Lydmonitor). Til højre: I2C med PSoC testprogram (Vuggesystem)

4.4 Modultest

Modultesten af Controller er opstillet som en modultest hvor Controlleren er forbundet til et testprint der simulerer BABYCON-niveauer og en PSoC4 med et testprogram til at kommunikere vha. I2C, hvor statusværdien kan ændres manuelt. I kildekoden er der implementeret en MODULTEST variabel, der kan sættes til true eller false, denne bliver refereret til løbende i modultest beskrivelsen.

”ON/OFF”-knap

Formål

At teste om ”ON/OFF”-LEDen lyser og programmet starter sit ON-loop når knappen er ON. Samt kontrollere om ”ON/OFF”-LEDen slukker når knappen er OFF og programmet forlader sit ON-loop.

Fremgangsmåde

1. Tilkoble strøm til Controlleren med ”ON/OFF” knappen i OFF position
2. Start programmet i MODULTEST-tilstand på Raspberry Pi’en
3. Sæt ”ON/OFF” knappen i ON position
4. Iagttag ”ON/OFF”-LEDen samt terminal udskrift
5. Sæt ”ON/OFF” knappen i OFF position
6. Iagttag ”ON/OFF”-LEDen samt terminal udskrift

Forventet resultat

Det forventes at LEDen tænder når ”ON/OFF” knappen bringes i ON position. Det forventes at terminal udskriften viser at programmet reagerer med udskrifterne [INSIDE ON LOOP] når knappen er i ON position og [OUTSIDE ON LOOP] når knappen er i OFF position.

Resultat

LEDen tænder og slukker som forventet og terminaludskriften passer med forventningerne. Se terminaludskriften på figuren herunder 4.20

```
root@raspberrypi:/home/pi/Debug# ./E4PRJ4
gpio: Warning (not an error): File not present: /dev/i2c-0
Opening I2C Device
[OUTSIDE ON LOOP]
[INSIDE ON LOOP]
[OUTSIDE ON LOOP]
```

Figur 4.20. Terminaludskrift - Modultest:"ON/OFF"-knap

Testen er godkendt

"Manuel start"-knap

Formål

At sikre "Manuel start" knappen aktiverer ISR-rutinen, her sættes en variabel som får en funktion til at køre.

Fremgangsmåde

1. Start programmet i MODULTEST-tilstand og sæt "ON/OFF" knappen i ON position
2. Aktiver "Manuel start"-knappen ved at trykke på den
3. Iagttag "Manuel start"-LED'en og terminaludskriften
4. Afvent at LED'en slukker og iagttag terminaludskriften

Forventet resultat

Det forventes at "Manuel start"-LED'en tænder umiddelbart når "Manuel start"-knappen aktiveres. Terminal udskriften reagerer ved [MANUELSTART ISR] når knappen aktiveres, [MANUELSTART RUNNING] funktionen kører og [MANUELSTART DONE] når funktionen er kørt. Herefter slukkes "Manuel start"-LED'en

Resultat

"Manuel start"-LED'en tænder med det samme når "Manuel start"-knappen aktiveres, som forventet. Og LED'en slukker når funktionen er kørt. Figur 4.21 viser terminal udskriften. [MANUELSTART ISR] skrives flere gange, det skyldes kontaktprel og er håndteret ved at der sættes en global variabel første gang knappen aktiveres, denne variabel nulstilles af funktionen selv.

```
root@raspberrypi:/home/pi/Debug# ./E4PRJ4
gpio: Warning (not an error): File not present: /dev/i2c-0
Opening I2C Device
[OUTSIDE ON LOOP]
[INSIDE ON LOOP]
[MANUELSTART ISR]
[MANUELSTART ISR]
[MANUELSTART ISR]
[MANUELSTART ISR]
[MANUELSTART ISR]
[MANUELSTART RUNNING]
[MANUELSTART DONE]
[OUTSIDE ON LOOP]
```

Figur 4.21. Terminaludskrift - Modultest:”Manuel start”-knap

Testen er godkendt

Internet status - Wi-Fi LED

Formål

At teste systemet med og uden internet tilkoblet

Fremgangsmåde

1. Start programmet i MODULTEST-tilstand og sæt ”ON/OFF” knappen i ON position
2. Fjern internet forbindelsen ved at afbryde routeren fra netværket
3. Iagttag Wi-Fi LED'en og terminaludskrift
4. Tilføj internet forbindelsen ved at tilkoble routeren fra netværket
5. Iagttag Wi-Fi LED'en og terminaludskrift

Forventet resultat

Det forventes at Wi-Fi LEDen lyser efter maks. 10 sekunder og slukker igen maks. 20 sekunder efter routeren er tilkoblet netværket. Terminalen viser tilsvarende ping udskrifter som viser status for internetforbindelsen.

Resultat

Wi-Fi-LEDen lyser som forventet ved afkobling af internettet. Og slukker igen ved re-tilkobling.

Testen er godkendt

BABYCON niveau fra testprint

Formål

At teste systems reaktion på de 3 BABYCON niveauer samt ved error/ikke tilkoblet Intelligent Lydmonitor.

Fremgangsmåde

1. Sæt testprintet i BABYCON3
2. Start programmet i MODULTEST-tilstand og sæt "ON/OFF" knappen i ON position
3. Iagttag terminaludskriften
4. Sæt testprintet i BABYCON2
5. Iagttag terminaludskriften
6. Sæt testprintet i BABYCON1
7. Iagttag terminaludskriften
8. Sæt testprintet i BABYCON0
9. Sæt ON/OFF-knap i OFF position
10. Sæt ON/OFF-knap i ON position
11. Iagttag terminaludskriften

Forventet resultat

Terminal udskriften skal vise de respektive BABYCON niveauer, samt andre MODULTEST udskrifter, der fortæller om hvilke LOOP man befinner sig i, og hvilke e-mails der sendes

Resultat

Resultatet kan ses på følgende terminal skærmbillede figur ???. Det ses at BABYCON niveauerne skifter som forventet.

```
root@raspberrypi:/home/pi/Debug# ./E4PRJ4
gpio: Warning (not an error): File not present: /dev/i2c-0
Opening I2C Device
[OUTSIDE ON LOOP]
[INSIDE ON LOOP]
BABYCON #3
BABYCON #3
BABYCON #3
BABYCON #3
BABYCON #3
BABYCON #3
BABYCON #2
BABYCON #1
Sender alarm e-mail... [0]
...alarm e-mail sendt
BABYCON #0 (ERROR)
Sender fejl e-mail...
...fejl e-mail sendt
[OUTSIDE ON LOOP]
[INSIDE ON LOOP]
BABYCON #0 (ERROR)
Sender fejl e-mail...
...fejl e-mail sendt
[OUTSIDE ON LOOP]
```

Figur 4.22. Terminaludskrift - Modultest: BABYCON niveauer fra testprint

Testen er godkendt

I2C kommunikation med PSoC testprogram

Formål

Formålet med denne modultest er at teste I2C kommunikationen med vuggesystemet vha. et testprogram på PSoCen, der udskriver i et terminalvindue via UART.

Fremgangsmåde

1. Start TeraTerminal, med PSoCen forbundet med testprogrammet på
2. Test UART ved tryk på reset på PSoC - se terminal udskrift ”Baby Watch - USB UART”
3. Sæt testprintet i BABYCON3
4. Sæt ON/OFF-knap i ON
5. Iagttag terminaludskriften
6. Sæt testprintet i BABYCON2
7. Iagttag terminaludskriften
8. Sæt testprintet i BABYCON1
9. Sæt ON/OFF-knap i OFF

Forventet resultat

TeraTerminalen udskrifterne skal udskrive værdier der passer med testværdierne.

1. ON: 0xF0
2. OFF: 0x0F
3. BABYCON1 - Frekvens: 0x00 Vinkel: 0x00
4. BABYCON2A - Frekvens: 0x32 Vinkel: 0xC8
5. BABYCON2B - Frekvens: 0x64 Vinkel: 0x78
6. BABYCON2C - Frekvens: 0xC8 Vinkel: 0x50
7. BABYCON3 - Frekvens: 0x00 Vinkel: 0x00

Resultat

TeraTerminalen udskriver som forventet de afsendte værdier i figur 4.25. På figuren er indsats BABYCON niveauer for forståelsens skyld.

```

Baby Watch - USB Uart!
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x04 - Status: 0x7f } BABYCON#3

Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x32
Reg 0x03 - Vinkel: 0xc8
Reg 0x04 - Status: 0x7f } BABYCON#2A

Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x32
Reg 0x03 - Vinkel: 0xc8
Reg 0x04 - Status: 0x7f } BABYCON#2B

Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x64
Reg 0x03 - Vinkel: 0x78
Reg 0x04 - Status: 0x7f } BABYCON#2C

Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0xc8
Reg 0x03 - Vinkel: 0x50
Reg 0x04 - Status: 0x7f } BABYCON#1

Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0xc8
Reg 0x03 - Vinkel: 0x50
Reg 0x04 - Status: 0x7f } OFF

```

Figur 4.23. TeraTerminaludskrift - Modultest: I2C kommunikation med PSoC testprogram

Testen er godkendt

Fejlhåndtering

Formål

Formålet med denne modultest er at teste om Controller reagerer korrekt på en error fra Vuggesystemet. Testprogrammet er programmeret med en statusværdi på 0b10000000 (0x80).

Fremgangsmåde

1. Start TeraTerminal, med PSoCen forbundet med testprogrammet på
2. Test UART ved tryk på reset på PSoC - se terminal udskrift "Baby Watch - USB UART!"
3. Sæt testprintet i BABYCON3
4. Sæt ON/OFF-knap i ON

5. Iagttag terminaludskriften

Forventet resultat

TeraTerminalen udskrifterne skal udskrive den første BABYCON3 skrivning fra TeraTerminalen, som slutter med at aflæse status.

Resultat

På figur 4.24 ses at den første BABYCON3 sekvens modtages, og i statusregisteret står 0x10 som forventet, dette handles der på i Controller og den går i fejl-tilstand og stopper vugningen.

```
Baby Watch - USB Uart!
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x04 - Status: 0x80

Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
Reg 0x01 - ON/OFF: 0xf0
Reg 0x02 - Frekvens: 0x0
Reg 0x03 - Vinkel: 0x0
```

Figur 4.24. TeraTerminaludskrift - Modultest: Fejlhåndtering

Testen er godkendt

I2C kommunikation med PSoC testprogram

Formål

Formålet med denne modultest er at teste I2C kommunikationen med vuggesystemet vha. et testprogram på PSoC, der udskriver i et terminalvindue via UART.

Fremgangsmåde

1. Start TeraTerminal, med PSoC forbundet med testprogrammet på
2. Test UART ved tryk på reset på PSoC - se terminal udskrift "Baby Watch - USB UART!"
3. Sæt testprintet i BABYCON3
4. Sæt ON/OFF-knap i ON

5. Iagttag terminaludskriften
6. Sæt testprintet i BABYCON2
7. Iagttag terminaludskriften
8. Sæt testprintet i BABYCON1
9. Sæt ON/OFF-knap i OFF

Forventet resultat

TeraTerminalen udskrifterne skal udskrive værdier der passer med testværdierne.

1. ON: 0xF0
2. OFF: 0x0F
3. BABYCON1 - Frekvens: 0x00 Vinkel: 0x00
4. BABYCON2A - Frekvens: 0x32 Vinkel: 0xC8
5. BABYCON2B - Frekvens: 0x64 Vinkel: 0x78
6. BABYCON2C - Frekvens: 0xC8 Vinkel: 0x50
7. BABYCON3 - Frekvens: 0x00 Vinkel: 0x00

Resultat

TeraTerminalen udskriver som forventet de afsendte værdier i figur 4.25. På figuren er indsat BABYCON niveauer for forståelsens skyld.

Baby Watch - USB Uart!		
Reg 0x01 - ON/OFF: 0xf0		
Reg 0x02 - Frekvens: 0x0		
Reg 0x03 - Vinkel: 0x0		
Reg 0x04 - Status: 0x7f		
		}
		BABYCON#3
Reg 0x01 - ON/OFF: 0xf0		
Reg 0x02 - Frekvens: 0x32		
Reg 0x03 - Vinkel: 0xc8		
Reg 0x04 - Status: 0x7f		
		}
		BABYCON#2A
Reg 0x01 - ON/OFF: 0xf0		
Reg 0x02 - Frekvens: 0x32		
Reg 0x03 - Vinkel: 0xc8		
Reg 0x04 - Status: 0x7f		
		}
		BABYCON#2B
Reg 0x01 - ON/OFF: 0xf0		
Reg 0x02 - Frekvens: 0x64		
Reg 0x03 - Vinkel: 0x78		
Reg 0x04 - Status: 0x7f		
		}
		BABYCON#2C
Reg 0x01 - ON/OFF: 0xf0		
Reg 0x02 - Frekvens: 0xc8		
Reg 0x03 - Vinkel: 0x50		
Reg 0x04 - Status: 0x7f		
		}
		BABYCON#1
Reg 0x01 - ON/OFF: 0xf0		
Reg 0x02 - Frekvens: 0x0		
Reg 0x03 - Vinkel: 0x0		
Reg 0x01 - ON/OFF: 0xf		
Reg 0x02 - Frekvens: 0x0		
Reg 0x03 - Vinkel: 0x0		
Reg 0x04 - Status: 0x7f		
		}
		OFF

Figur 4.25. TeraTerminaludskrift - Modultest: I2C kommunikation med PSoC testprogram

Testen er godkendt

Powercontrol

Formål

At teste funktionaliteten af power on og off signal fra Controller til PSU.

Fremgangsmåde

- 1.
 2. Sæt ON/OFF-knap til ON

Forventet resultat

Resultat

Testen er godkendt

Intelligent lydmonitor

5

5.1 Forundersøgelse

I dette afsnit undersøges og analyseres udvalgte lyde fra babygråd og typisk omgivelsesstøj såsom trafikstøj og fuglefløjt.

Situationer

Der udføres analyser på lyd-optagelser af følgende situationer:

- Højlydt babygråd
- Moderat babygråd
- Fuglefløjt og omgivelsesstøj fra natur
- Trafik
- Katte
- Latter

Disse analyseres således som beskrevet i efterfølgende afsnit.

5.1.1 Metode

Der analyseres tre optagelser af hver af de udvalgte lyde i formatet .wav. Lydene analyseres i både lange (5 sek) og korte (5 msec) optagelsessegmenter i Matlab med hensyn til:

- Frekvensindhold
- Dominant tone
- Grad af tonalitet (andel af signalets effekt, der tilskrives dominante toner frem for generel støj)

Disse parametre analyseres med redskaberne Short-Time DFT, en max-funktion, Tonal Power Ratio, samt en smoothing-funktion.

Short-Time DFT

FFT af kortere optagelsessegmenter. Produktet af dette, kan plottes som et spektrogram, der således viser optagelsens frekvensindhold som variation af tiden.

Max

Matlabs funktion "max", finder den højeste værdi i matrixen. I denne analyse, vil den benyttes på FFT'en af de enkelte optagelsessegmenter (Short-Time DFT'ens output). For yderlige beskrivelse af max-funktionen se [Mathworks, 1994-2015]

Tonal Power Ratio

Tonal Power Ratio benyttes til beregne tonaliteten i optagelsen, og er altså et udtryk for

forholdet mellem energiindholdet i dominante toner og det samlede energiindhold. Hvid støj vil således have en TPR = 0 og en ren sinus-tone vil have en TPR = 1. TPR beregnes som følger:

$$v_{Tpr} = \frac{E_T(n)}{\sum_{i=0}^{K/2-1} |X(k,n)|^2}$$

Hvor nævneren er det totale spektrale energiindhold og tælleren, E_T , er det tonale indhold. Det tonale indhold beregnes ved at tage FFT'en af de enkelte optagelsessegmenter (Short-Time DFT'ens output) og summere alle de bins, der:

- Er lokalt maximum: $|X(k-1, n)|^2 \leq |X(k, n)|^2 \leq |X(k+1, n)|^2$
- Ligger over en forudbestemt grænseværdi G_T .

Resultatet af vil ligge mellem $0 \leq v_{Tpr} \leq 1$

Smoothing

Til at udjævne analysesignalene benyttes matlabs "smooth"funktion. Default filteret, moving average, benyttes.

5.1.2 Analysekode gennemgang

I dette afsnit findes en kort gennemgang i de benyttede funktion i matlab. Syntax for plots er udeladt, og der henvises til bilag med komplet analysekode for denne.

Klargøring af optagelserne. Dette gøres ved at loade de ønskede filer, ekstrahere den ene kanal, for at få lyden i mono hvilket gør den lettere at analysere, sætte den ønskede længde af hanning-vinduet og shift, samt at benytte hanning-vinduesfunktionen for at minimere ripple i den senere diskrete fourier transformation. Sluttligt konkatteneres de tre optagelser med lidt pause imellem dem, og vi får vores endelige analyse materiale i variablen "x". Bemærk at samplefrekvensen for de tre optagelser regnes ens, samt at kun ekstraktionen af den ene optagelse er noteret mellem de to "..." i koden herunder:

```
%**** EXTRACT SAMPLES ****%
[rec1, f_s] = wavread('Hoejlydt_babygraad1');
[rec2, f_s] = wavread('Hoejlydt_babygraad2');
[rec3, f_s] = wavread('Hoejlydt_babygraad3');

...
r1 = rec1(:,1)'; %extract left channel'

frame_sec = 5.0; %Set frame size
shift_sec = 0.1; %Set time shift

frame_N = frame_sec*f_s;
shift_N = shift_sec*f_s;

r1_win = r1(shift_N:frame_N-1+shift_N); %Extract framed sample
```

```
r1_win = r1_win .* hanning(length(r1_win))'; %Window with hanning
...
x = [r1_win, rSpace, r2_win, rSpace, r3_win];
```

Efterfølgende båndpas filtreres optagelsen men 100 Hz som nedre knækfrekvens og 10 kHz som øvre knækfrekvens. Se bilag med komplet kode for syntax.

Korttids-analyse. For korttids-analysen, opdeles optagelsen i korte slices, der hver især behandles med dft og efterfølgende analyseres med hensyn til dominant tone i dft-spektret via matlabs "max"-funktion samt generelt toneindhold via tonal power ratio funktionen.

```
%**** SHORT-TIME DFT (Spectrogram) ****%
segmentlen = 256;
nOverlap = 60;
NFFT = 1280;
[s,f,t,p] = spectrogram(x, segmentlen, nOverlap, NFFT, f_s,'yaxis');
...
[q,nd] = max(10*log10(p));
maxPwr_smooth = smooth(f(nd),0.1,'moving'); % Smooth data
...
TPR = TonalPowerRatio(abs(s), f_s, 5*10^-2);
TPR_smooth = smooth(TPR(:,0.05,'moving'));% Smooth data
```

Langtids-analyse. For langtids-analysen, behandles hele optagelsen med en dft. Dft-spektret midles hernæst med et moving average. Midlingen foretages på spektret i dB og gentages med flere længder (parameter 2). Sluttelige findes den globale TPR for hele optagelsen.

```
%**** FAST FOURIER TRANSFORM ****%
NFFT = 2^nextpow2(N); %Find next power of 2
X = fft(x,NFFT);
...
X_abs = abs(X);
X_abs_log = 20*log10(2*abs(X(1:NFFT/2+1)));
X_smooth_log1 = smooth(X_abs_log,0.002,'moving');
X_smooth_log1 = smooth(X_smooth_log1,0.002,'moving');
X_smooth_log2 = smooth(X_abs_log,0.03,'moving');
X_smooth_log2 = smooth(X_smooth_log2,0.03,'moving');
...
%TPR for global DFT
sig = X_abs;
fSum = sum(sig); %sum all freq bins
[afPeaks] = findpeaks(sig); %find peaks
G_T = 5*10^-2; %threshold
k_peak = find(afPeaks > G_T); %find peak above thresh
```

```
TPR_global = sum(afPeaks(k_peak))/fSum
```

5.1.3 Analyser

Resultatet af korttids-analysen er 3-plots:

1. Sampleplot af det givne lydsignal (blå)
2. Tonal Power Ratio (rød) og midlet TPR (sort)
3. Spectrogram (colormap: bone), dominerende tone (blå prikker), midlet dominerende tone (rød streg)

Bemærk for plot 3, at data for dominerende frekvens er fjernet ved TPR på 0.

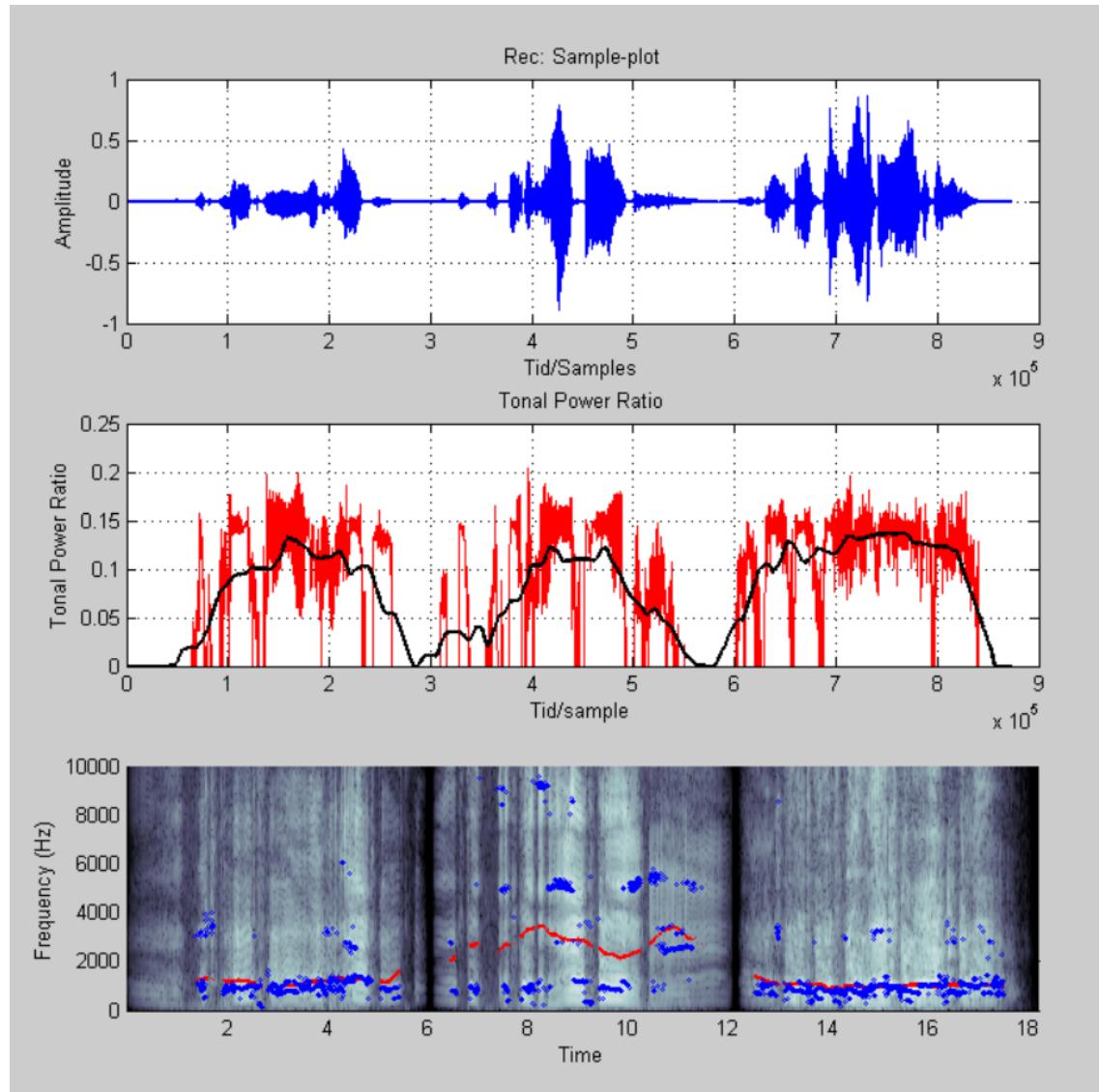
Resultatet af langtids-analysen er:

1. DFT-spektrum: Originalt (blå), moderat filtreret (rød), kraftigt filtreret (sort)
2. Global TPR-værdi

Højlydt babygråd

Korttids-analyse

Karakteristisk ved højlydt babygråd er, at dele af Tonal Power Ratio (TPR) når over en værdi på 0.12 (se plot 2). Herudover ses det på plot 3 at de dominerende frekvenser (blå prikker) typisk ligger omkring 900, 3100 og 5000 Hz. Midtvejs disse værdier over længere tid, ligger det dominerende frekvensindhold mellem 900 Hz og 3500 Hz (røde streger).

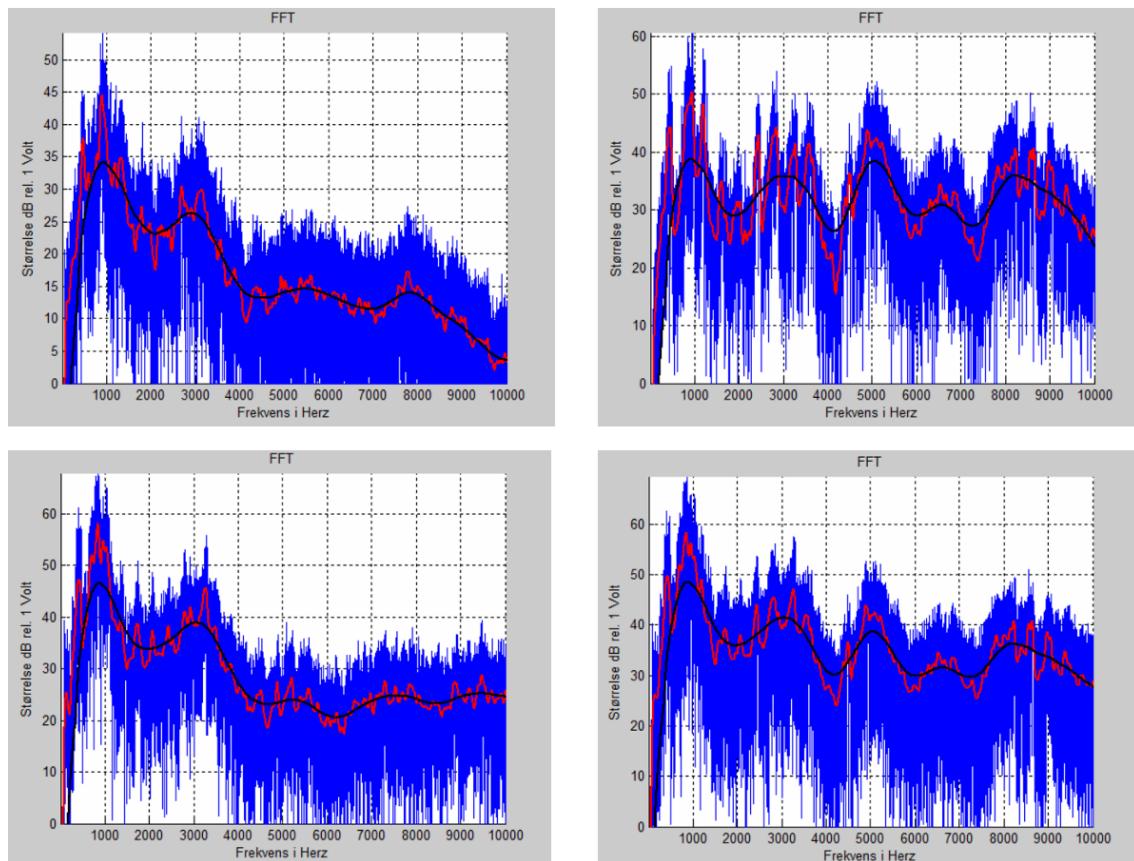


Figur 5.1. Korttids-analyse af højlydt babygråd optagelser

Langtids-analyse

Karakteristisk ved langtidsanalysen for højlydt babygråd er, at den dominerende tone (sort streg) ligger omkring 900 Hz. Næste dominante frekvens ligger mellem 3 kHz og 3,1 kHz. Endvidere ses at magnituden af frekvenserne mellem 1 kHz og 6 kHz er relativt ligeligt distribuerede, i forhold til andre analyserede signaler, med magnitudeforskæl på laveste og højeste frekvens (på sort, midlet kurve) på omtrent 25 dB.

De tre optagelser samlet (nederste højre graf) har en TPR = 0.336.

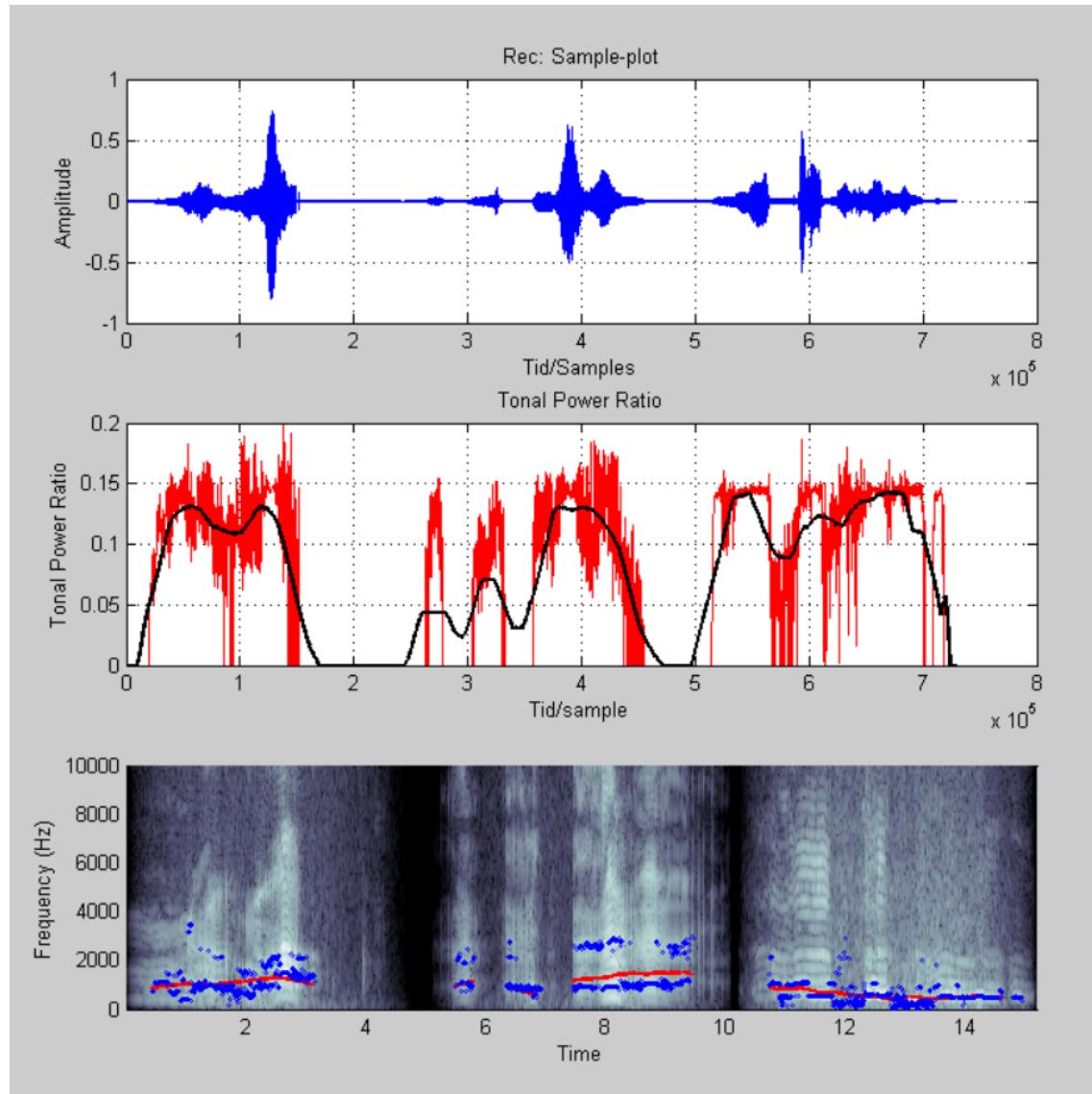


Figur 5.2. Langtids-analyser af højlydt babygråd optagelser. Nederste højre graf er analyse af de tre optagelser samlet

Moderat babygråd

Korttids-analyse

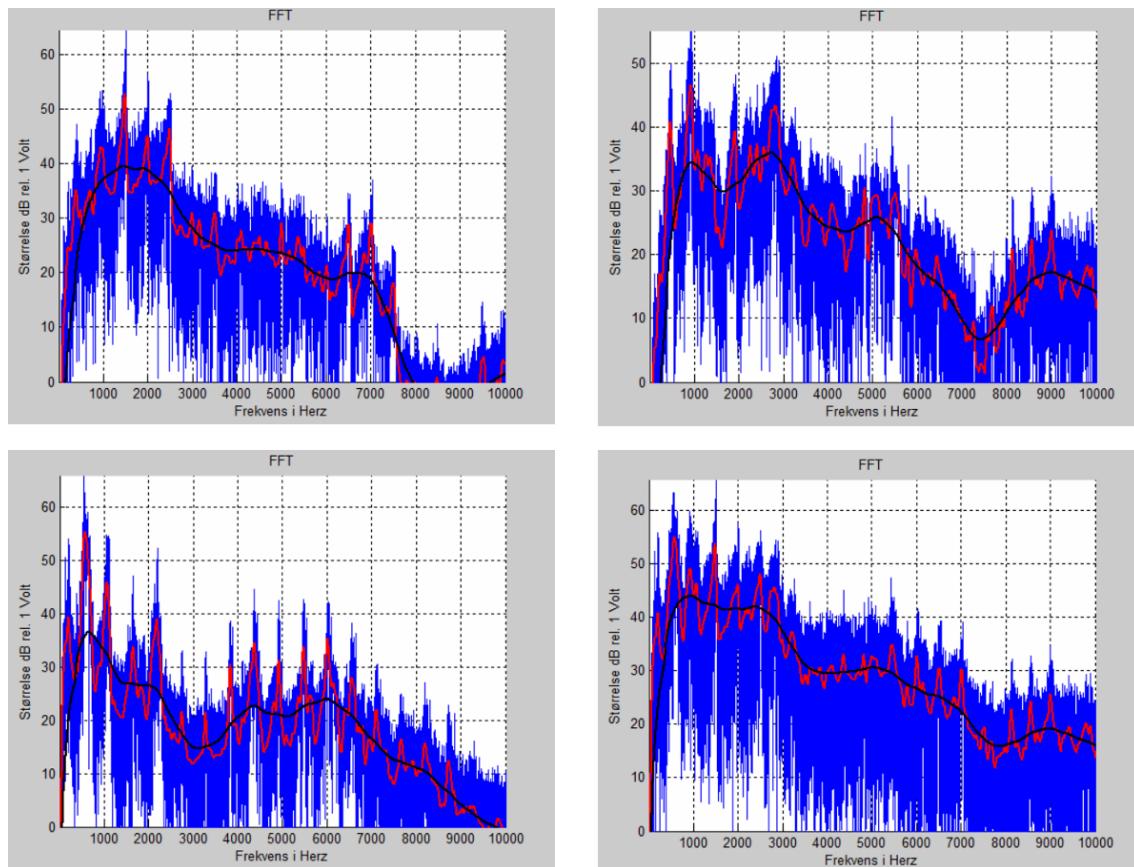
Karakteristisk ved optagelsen af moderat babygråd er, at en forholdsvis stor del af TPR ligger over en værdi på 0.12 (plot 2). Af plot 3 fremgår det at de dominerende frekvenser typisk ligger mellem 500 og 3500 Hz (blå prikker). Midt mellem disse værdier, ligger det dominerende frekvensindhold mellem 900 Hz og 1500 Hz (røde streger).



Figur 5.3. Korttids-analyse af moderat babygråd optagelser

Langtids-analyse

Ved langtidsanalysen for højlydt babygråd ligger den dominerende dybe tone (sort streg) på hhv. 1,5 kHz (optagelse 1, øverst venstre), 1 kHz (optagelse 2, øverst højre) og 600 Hz (optagelse 3, nederst venstre). Næste dominante frekvens ligger omkring 2,5 kHz for optagelse 2 og 3. For optagelse 1, findes intet tydeligt peak. Der er altså ikke tydelig korrelation for placering af frekvenspeaks. Magnituden af frekvenserne mellem 1 kHz og 6 kHz er relativt ligeligt distribuerede, i forhold til andre analyserede signaler, med magnitudeforskel på laveste og højeste frekvens (sort, midlet kurve) på omtrent 20 dB. De tre optagelser samlet (nederste højre graf) har TPR = 0.374.

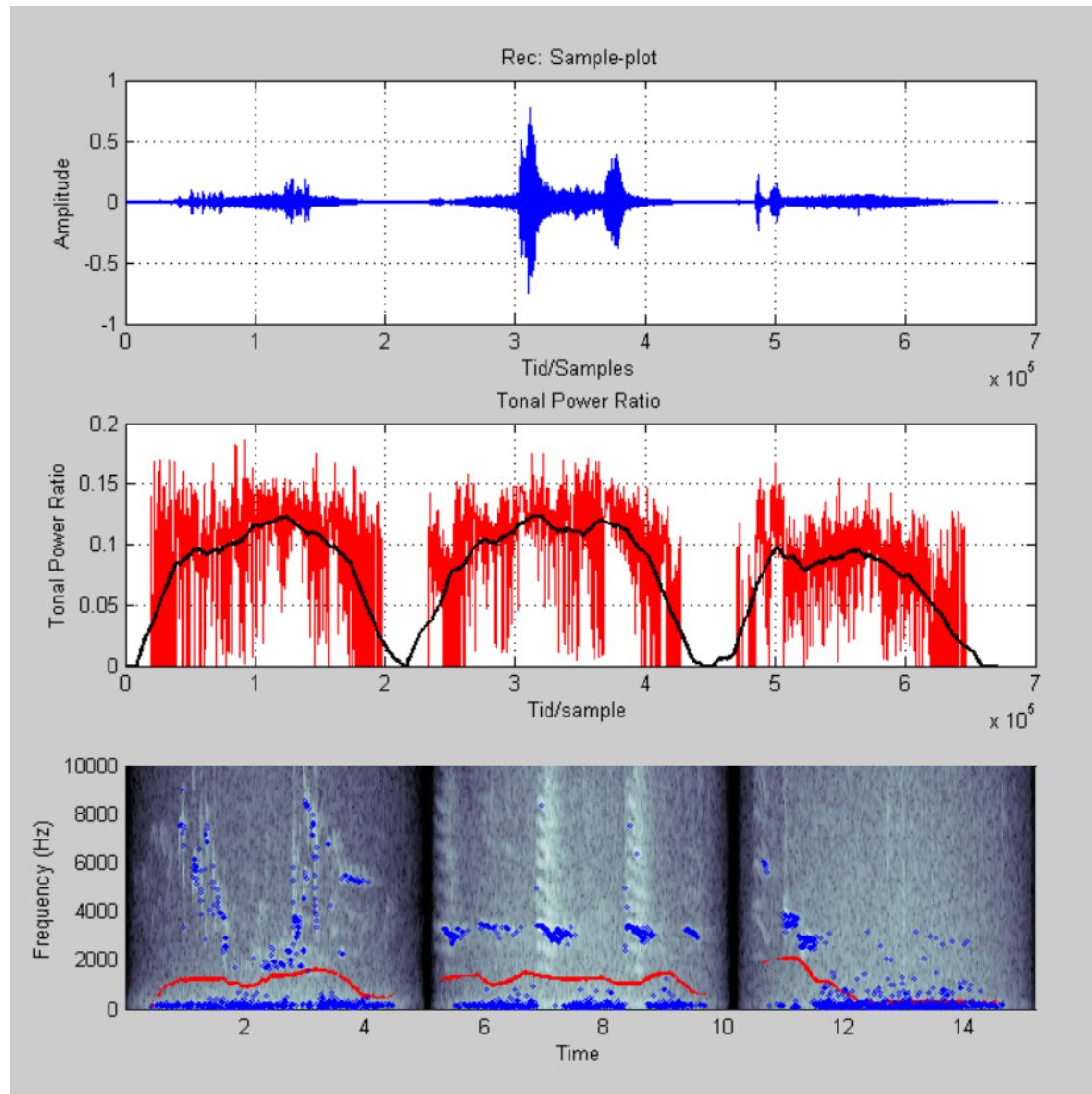


Figur 5.4. Langtids-analyser af moderat babygråd optagelser. Nederste højre graf er analyse af de tre optagelser samlet

Fuglefløjt og omgivelsesstøj fra natur

Korttids-analyse

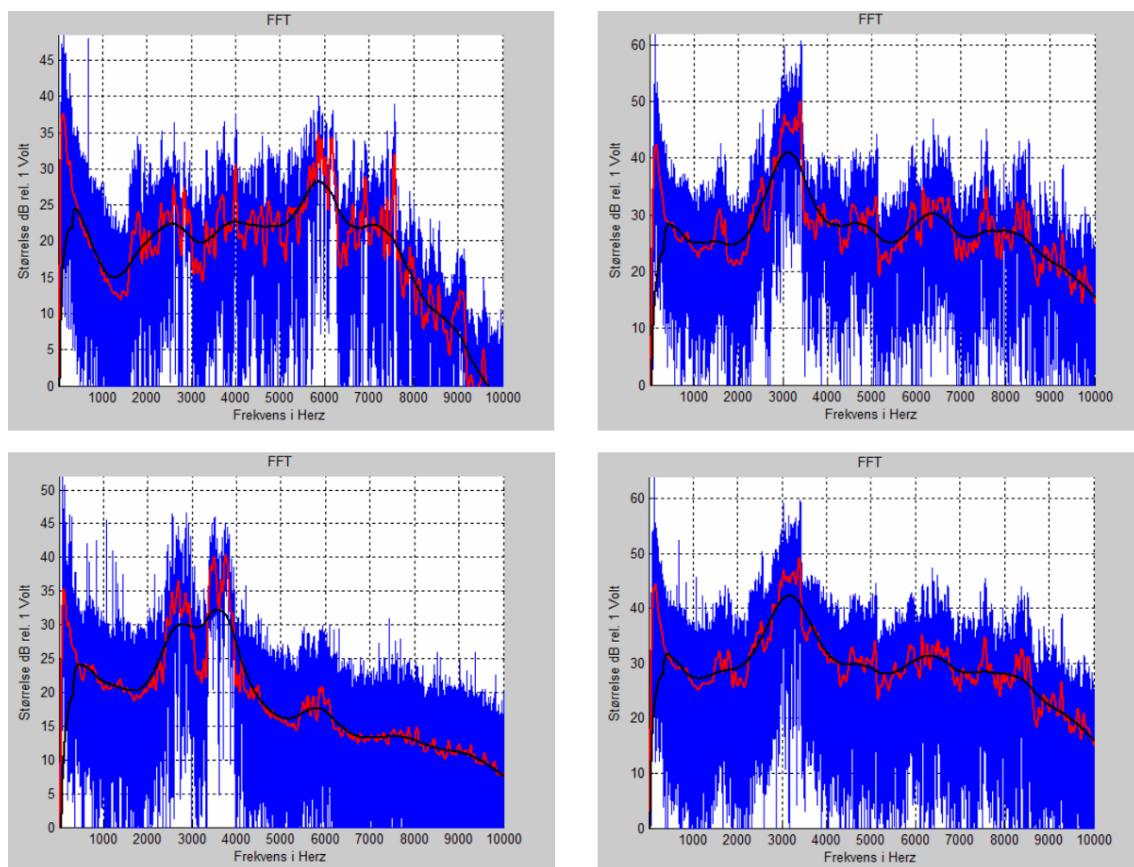
Karakteristik for fuglefløjt og omgivelsesstøj fra natur er, at TPR ligger på en værdi over 0.11, men under 0.13 (plot 2). På spektrogrammet (plot3), de blå prikker, ses det at de dominerende frekvenser ligger meget spredt; det meste af tiden dominerer omgivelsesstøj fra natur (ca. 100 Hz), og andre gange dominerer fuglefløjt med frekvenser fra 1,8 kHz til 6 kHz. Den midlede dominerende tone, røde prikker plot 3, ligger mellem 35 Hz og 2500 Hz.



Figur 5.5. Korttids-analyse af fuglefløjt og omgivelsesstøj fra natur optagelser

Langtids-analyse

For FFT'en af en længere optagelse af fuglefløjts og omgivelsesstøj fra natur, er omgivelsesstøj fra natur en markant bidrager til frekvensspektret, som det ses af det store frekvensindhold under 1 kHz. Bemærk her at filtreringen, der resulterer i den sorte kurve, det tager lidt tid inden filtreringen træder i kraft, og at den derfor ikke er troværdig for frekvenser under ca. 400 Hz. Næste frekvenspeak ses for optagelse 2 og 3 (og til dels for optagelse 1) omkring 3 kHz. Dette frekvenspeak kunne interferere med detekteringen af højlydt babygråd, der har et peak samme sted. Magnituden af frekvenserne mellem 1 kHz og 6 kHz er ligeligt distribuerede med magnitideforskæl på laveste og højeste frekvens (sort, midlet kurve) på omtrent 20 dB. Magnitudeforskellen på endepunkter 1 og 6 kHz, ligger dog under 10 dB. De tre optagelser samlet (nederste højre graf) har TPR = 0.291.

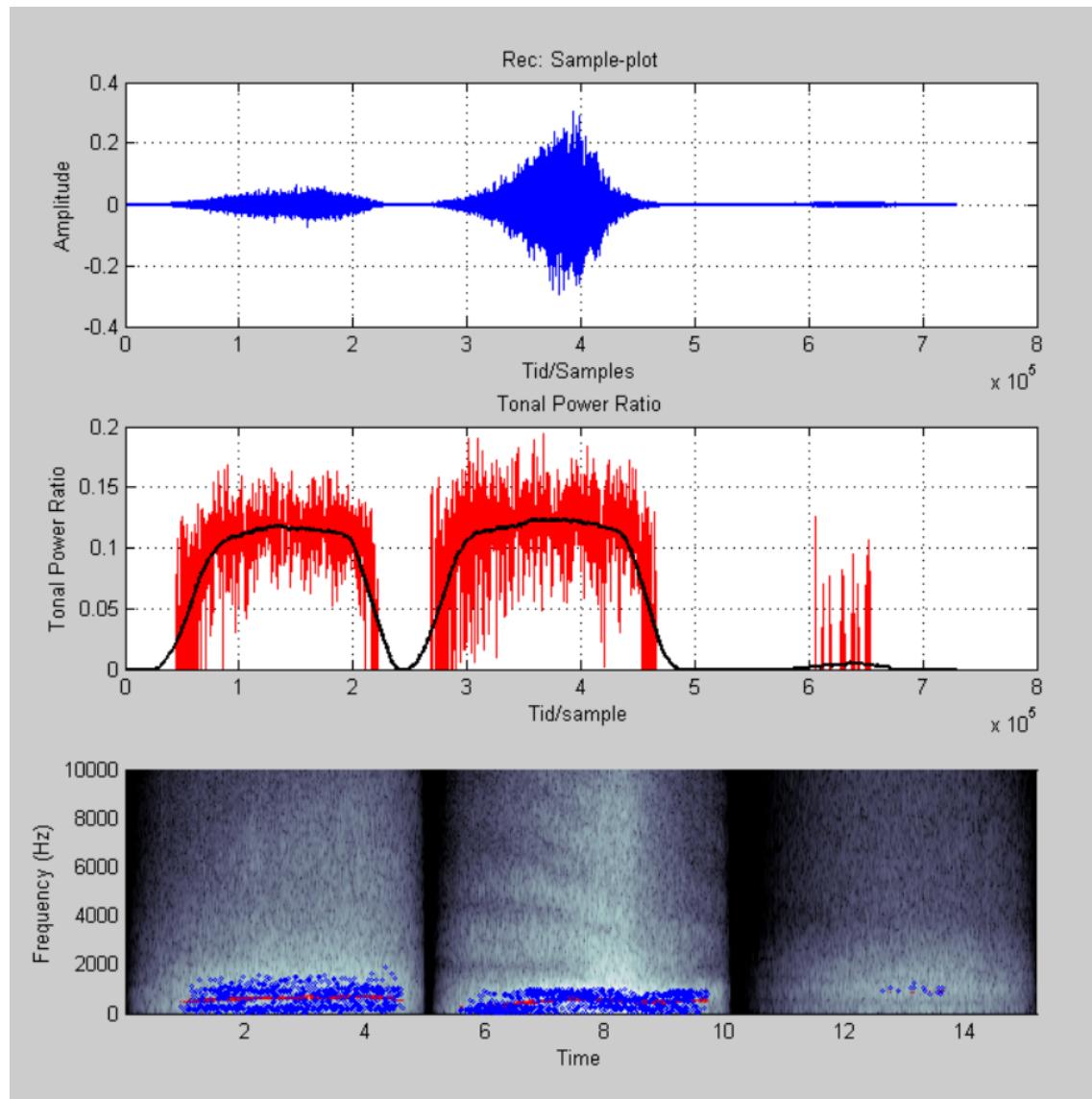


Figur 5.6. Langtids-analyser af fuglefløjts og omgivelsesstøj fra natur optagelser. Nederste højre graf er analyse af de tre optagelser samlet

Trafikstøj

Korttids-analyse

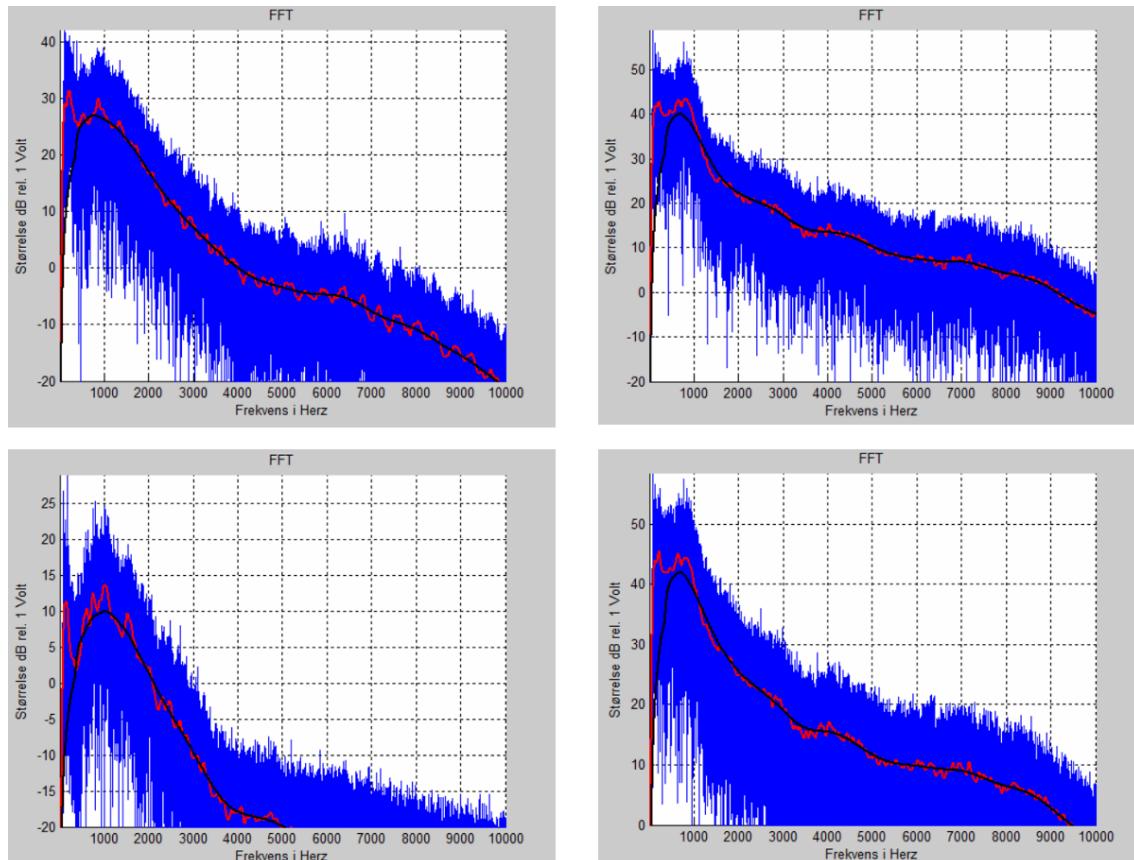
Karakteristisk for trafikstøj er, at TPR ligger over 0.10, men under 0.13 (plot 2). Af spektrogrammet (plot 3), de blå prikker, fremgår det at frekvenserne ligger spredt mellem 0 og 1500 Hz med en middel-værdi, røde prikker, på ca 700 Hz.



Figur 5.7. Korttids-analyse af trafikstøjs-optagelser

Langtids-analyse

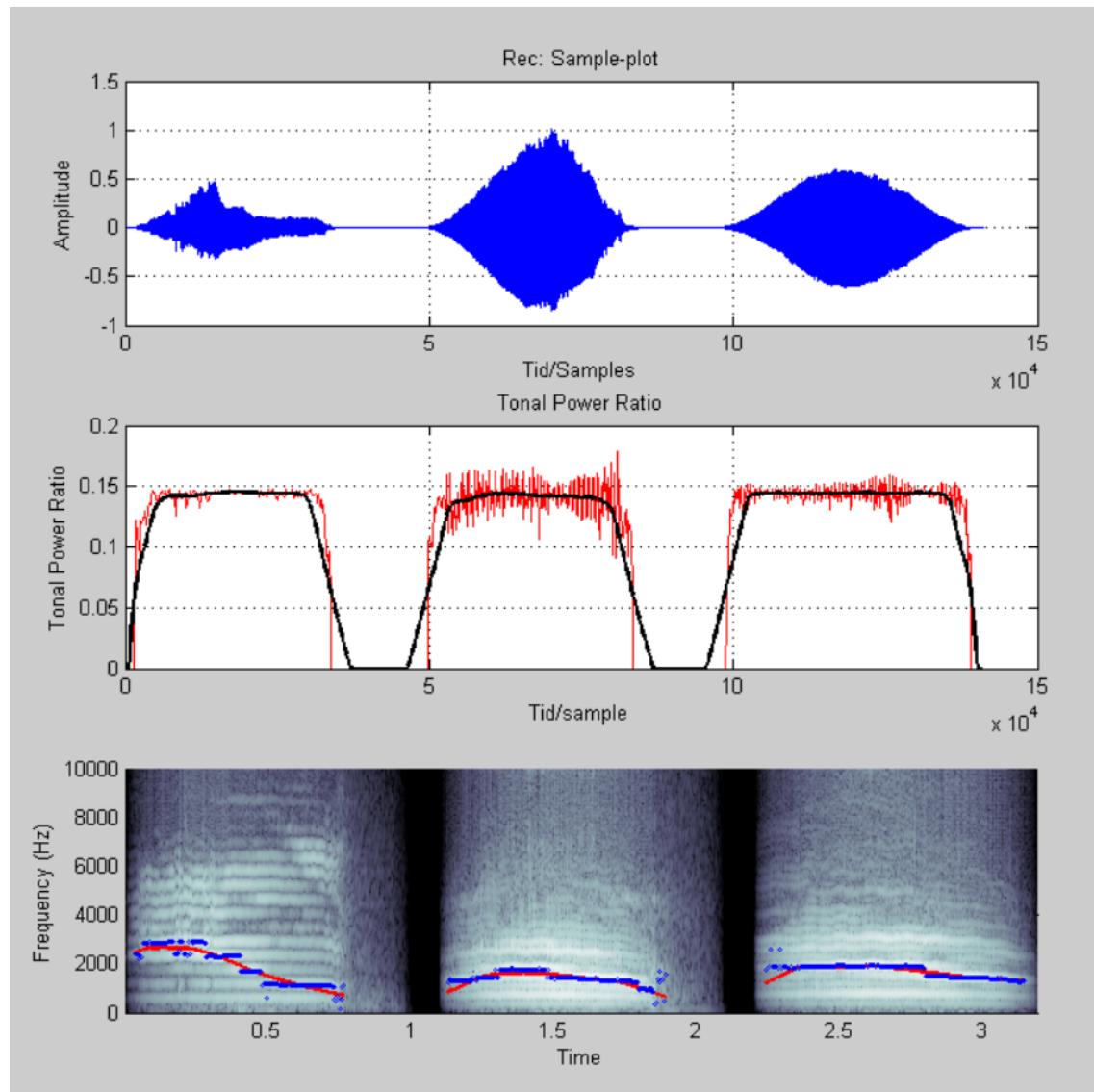
For trafikstøjens frekvensspektrum, er toneindholdet under 1 kHz klart dominerende. Magnituden for frekvenser derover falder jævnt som funktion af frekvensen med en karakteristik, der kunne minde om lyserød støj med et fald på omkring 40 dB/dek. De tre optagelser samlet (nederste højre graf) har TPR = 0.269.



Figur 5.8. Langtids-analyser af trafikstøjs-optagelser. Nederste højre graf er analyse af de tre optagelser samlet

Kat**Korttids-analyse**

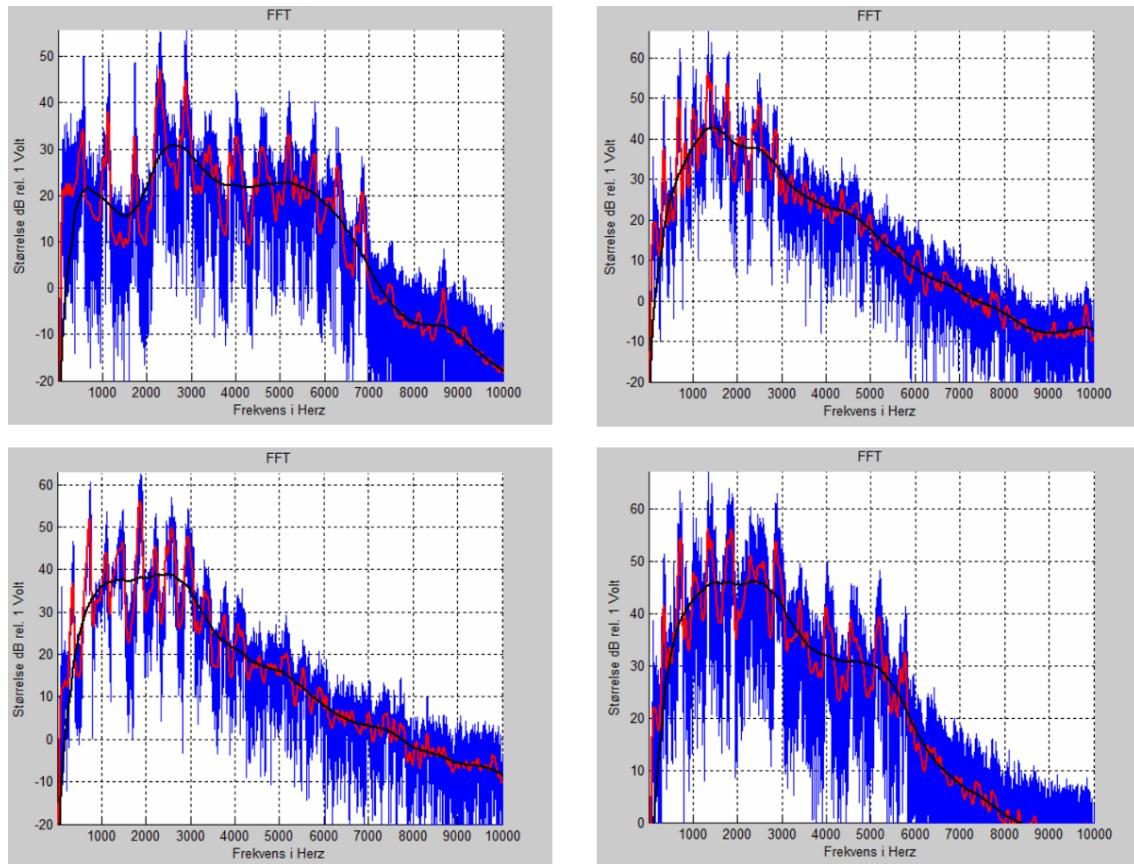
Korttids analysen af katteoptagelserne resulterer i en TPR omkring 0.15 (plot 2). Af spektrogrammet (plot 3) fremgår det at frekvenserne ligger fordelt i bånd, båndene har en afstand på mellem 300 og 500 Hz. Dominerende toner ligger mellem 1,5 og 3 kHz.



Figur 5.9. Korttids-analyse af katte-optagelser

Langtids-analyse

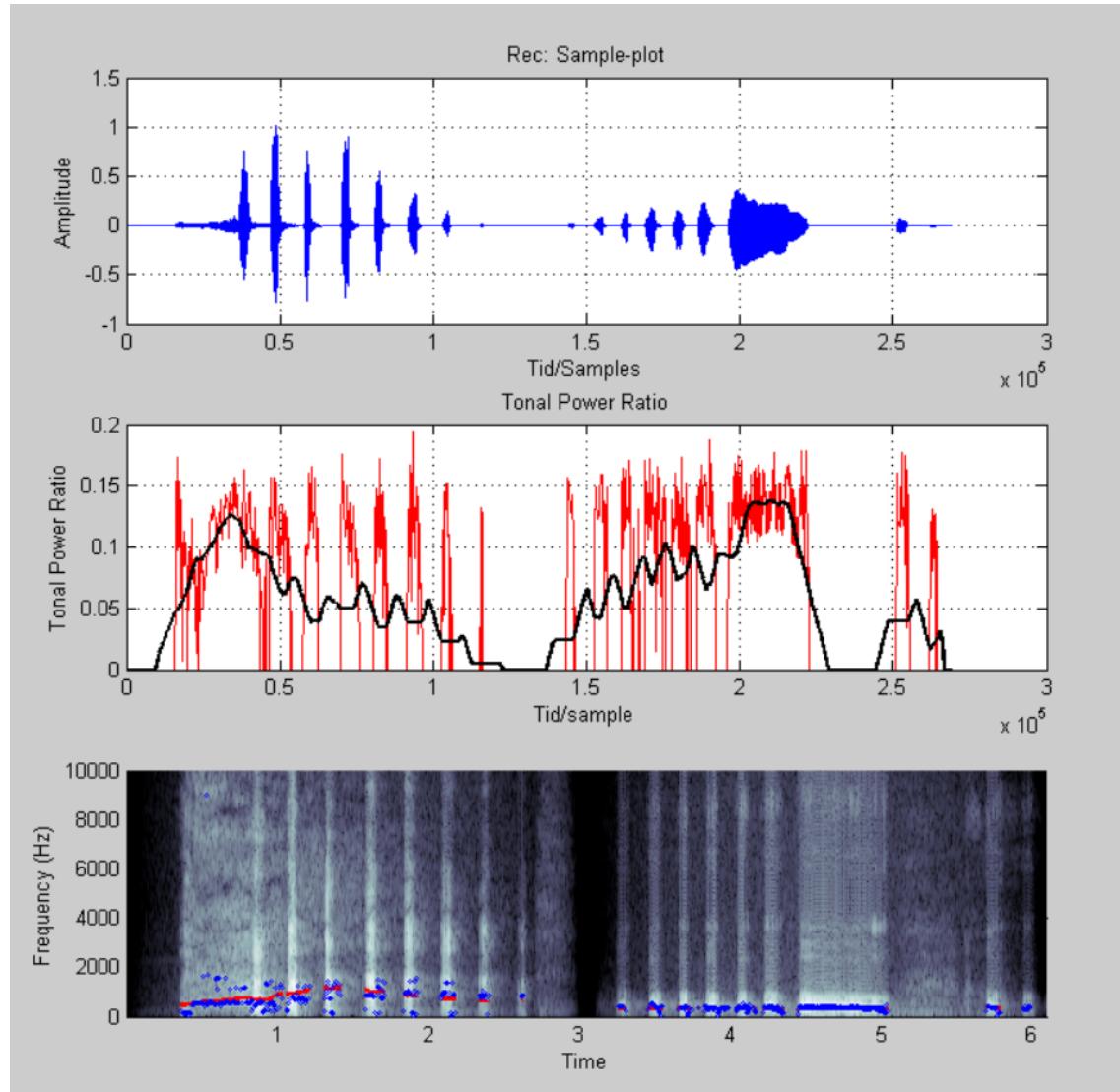
Katte-optagelsen producerer et frekvensspektrum med tydelig grundtone og harmoniske (rød kurve). Midtler der, findes peaks omkring 1000 Hz og 2,5 kHz. Generelt falder magnituden jo højere frekvensen bliver. For optagelse 1, falder den drastisk efter 5,5 kHz, men for optagelse 2 og 3 falder den fra 2, kHz. For optagelse 2 og 3 er der omkring 30 dB fald mellem 1kHz og 6kHz. De tre optagelser samlet (nederste højre graf) har TPR = 0.295.



Figur 5.10. Langtids-analyser af katte-optagelser. Nederste højre graf er analyse af de tre optagelser samlet

Latter**Korttids-analyse**

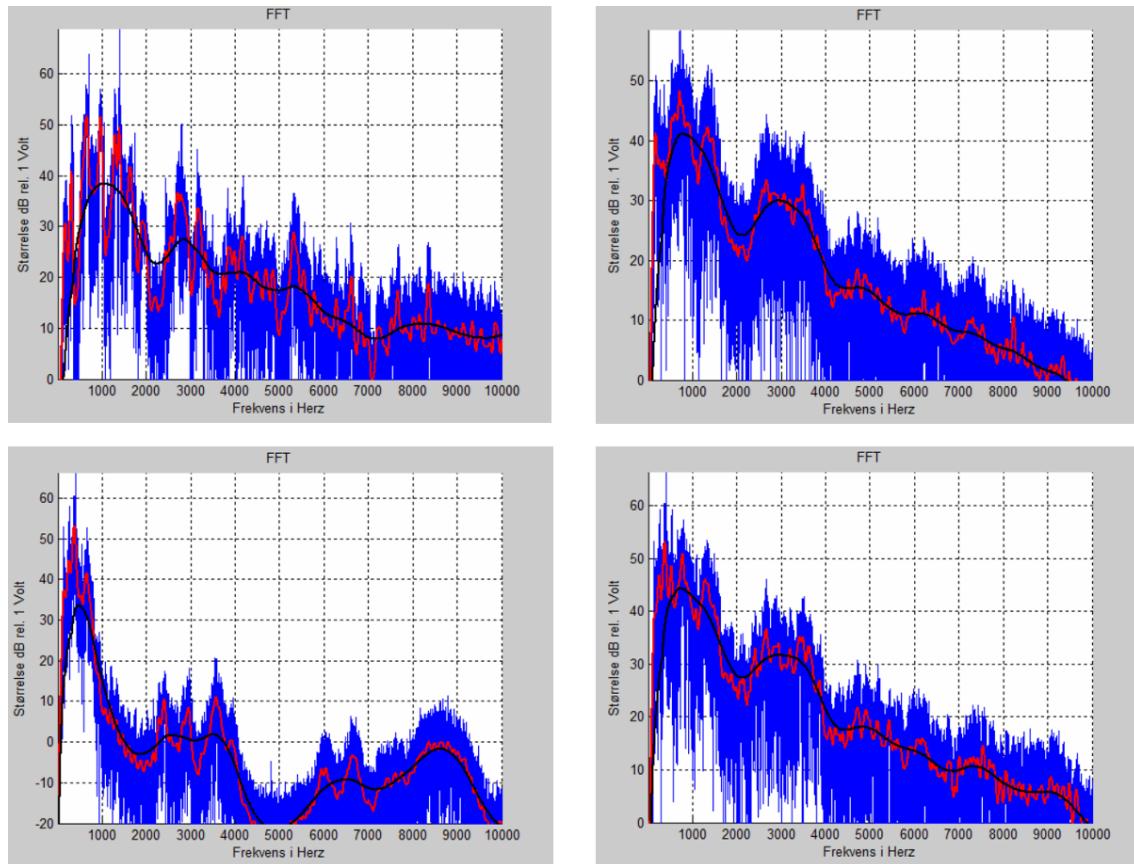
Latter forekommer stødvis med dominerende tone mellem 300 og 1200 Hz. Der produceres et tonalt indhold med TPR omkring 0.15 for latter-stødene (figur 2, rød kurve). Midler TPR, ligger den mellem 0.12 og 0.04 for en enkelt optagelse.



Figur 5.11. Korttids-analyse af latter-optagelser

Langtids-analyse

Latter genererer et spektralt indhold med første peak omkring 900 Hz og andet peak mellem 2,6 og 2,9 kHz. Disse peaks ligger meget tæt på de frekvenspeaks, der genereres af højlydt babygråd. Det generelle magnitude-fald er mellem 30 og 50 dB for de analyserede optagelser. De tre optagelser samlet (nederste højre graf) har TPR = 0.253.



Figur 5.12. Langtids-analyser af latter-optagelser. Nederste højre graf er analyse af de tre optagelser samlet

Korttids-analyse delkonklusion

Nøgle-værdier for kort-tidsanalysen opsummeret i tabellen herunder. Maximal TPR-værdi er angivet for den midlede værdi (sorte kurve i figur 2). For minimum- og maksimum-frekvenser er værdier for den midlede kurve (rød kurve, figur 3) ligeledes noterede.

Situation	Max TPR	Min frekvens	Max frekvens
Højlydt babygråd	0.144	770 Hz	1932 Hz
Moderat babygråd	0.146	676 Hz	1613 Hz
Fugleflojt	0.127	169 Hz	2220 Hz
Trafikstøj	0.123	138 Hz	710 Hz
Kat	0.148	720 Hz	2452 Hz
Latter	0.138	350 Hz	1250 Hz

Korttids-analyse: Tonal Power Ratio

Babygråd producerer en højere TPR end omgivelsesstøj som trafik og fugleflojt. Katteoptagelse producerer imidlertid en maxTPR på højde med babygråd, og latter kommer også deropad. Der er ikke tydelig forskel i TPR på højlydt og moderat babygråd, og det er således ikke et korttids analyseværktøj, der kan bruges til at skelne mellem disse. Ej helle kan det bruges til med sikkerhed at konkludere hvorvidt der er tale om en baby.

Korttids-analyse: Dominerende frekvensindhold

Dominerende frekvenser for babygråd ligger typisk mellem 700 og 1900 Hz, med lidt højere båndbredde for Højlydt babygråd end for moderat. Fugleflojt og katte optagelserne har lyseste dominerende frekvens på op til 2,3 kHz-området for den midlede kurve. Trafikstøjens dominerende frekvenser bevæger sig ikke over 710 Hz. Latter når ligeledes ikke op på lyseste dominerende frekvens for babygråd. Med isolerede optagelser er det altså umiddelbart muligt at identificere både højlydt og moderat babygråd, hvis maksimale udsving analyseres over længere tid (for denne forundersøgelse 15 sek).

Langtids-analyse delkonklusion

Nøgle-værdier for kort-tidsanalysen opsummeret i tabellen herunder. Uover global værdi for Tonal Power Ratio, er de visuelt analyserede frekvenspeaks opsummerede. Ligeledes er det gennemsnitlige magnitudefald mellem analysefrekvenserne 1 kHz og 6 kHz noteret.

Situation	TPR	Frekvenspeak 1	Frekvenspeak 2	Magnitudefald (1 kHz til 6 kHz)
Højlydt babygråd	0.336	900 Hz	3050 Hz	-25 dB
Moderat babygråd	0.374	1000±400 Hz	N/A	-20 dB
Fuglefjøjt og omgivelsesstøj fra natur	0.291	N/A Hz	3 kHz	-10 dB
Trafikstøj	0.269	>1 kHz	N/A	-40 dB
Kat	0.295	1000±300 Hz	2,5 kHz	-30 dB
Latter	0.253	900 Hz	2,8±0,2 kHz	-30±10 dB

Langtids-analyse: Tonal Power Ratio

Begge typer babygråd producerer højere TPR end de resterende optagelser. Nærmeste støjoptagelse (Kat) ligger 12% lavere. Dette vurderes som tilstrækkelig til adskille almindelig baggrundsstøj fra babygråd. Der er 10% forskel i TPR mellem højlydt og moderat babygråd. Dette antages tilstrækkeligt til at kende forskel.

Langtids-analyse: Dominerende frekvensindhold

Højlydt babygråd har tydelige frekvenspeaks, der ligger ret ensartet prøveoptagelserne imellem. Moderat babygråd har ikke tydelige frekvenspeaks. Selvom både kat og latter har lavt frekvenspeak, liggende i samme område som højlydt babygråd, ligger deres andet frekvenspeak lavere end for højlydt babygråd. Latter kan i nogle tilfælde også nå et frekvenspeak i samme område som højlydt babygråd. Langtidsanalyse af dominerende frekvensindhold er en valid måde at identificere højlydt babygråd på. Visse latter-optagelser, vil imidlertid også fanges af denne analysemетодik.

Langtids-analyse: Magnitudefald fra 1 kHz til 6 kHz

Både højlydt og moderat babygråd har magnitudefald i omegnen af 20 dB. Højlydt babygråd har magnitudefald under på 25 dB, men den nedre grænse er ikke ensartet, som det ses af 5.4 analysen øverst højre. Moderat babygråds fald er imidlertid mere ensformigt, og ligger hver gang mellem -12 dB og -20 dB. Støjsignalernes magnitudefald ligger samtidig mellem 10 og 20 dB fra det for babygråd. Magnitudefaldet mellem 1 kHz og 6 kHz

5.1.4 Konklusion

For korttids-analysen giver maxværdien af tonal power ratio ikke et ensformigt billede af hvorvidt der er tale om en baby. Min og max frekvenserne for det dominerende toneindhold er en indikator, men denne metodik vurderes følsom overfor uforudset støj.

For langtids-analysen er TPR en plausibel metode for detektion af babyens tilstand. Højlydt babygråd vil da have TPR i omegnen 0.336 og moderat babygråd vil have TPR i nærheden af 0.374. Implementeres denne metodik, bliver kalibrering dog nødvendig. Analyse af det dominerende frekvensindhold giver et udmaerket billede af hvorvidt der er højlydt babygråd. Dog giver visse tilfælde af latter lignende frekvenspeaks, og endnu en analysemetode vil være nødvendig for at frasortere denne støjtype. Moderat babygråd viser ikke på samme måde tydelige frekvenspeaks. Analyse af magnitudefaldet fra 1 kHz til 6 kHz giver et billede af hvorvidt der er tale om babygråd.

Analyse af magnitudefaldet vil sammen med detektering af frekvenspeaks være tilstrækkeligt for detekteringen og kategoriseringen af babygråd. Til forfining af detektion kunne tonal power ratio benyttes. Til videre forfining kunne en dB-måling af grådens lydstyrke (i tids-domænet) endvidere benyttes til at fjerne støj fra det fjerne.

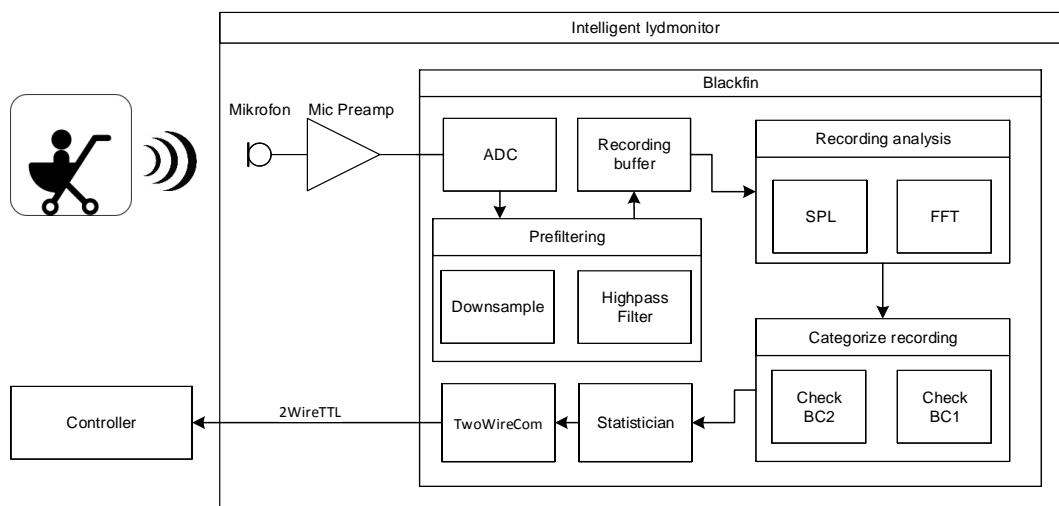
5.2 Systemarkitektur

I dette afsnit beskrives systemarkitekturen for den intelligente lydmonitor.

Overordnet virkemåde

Overordnet skal den intelligente lydmonitor fungere som følger:

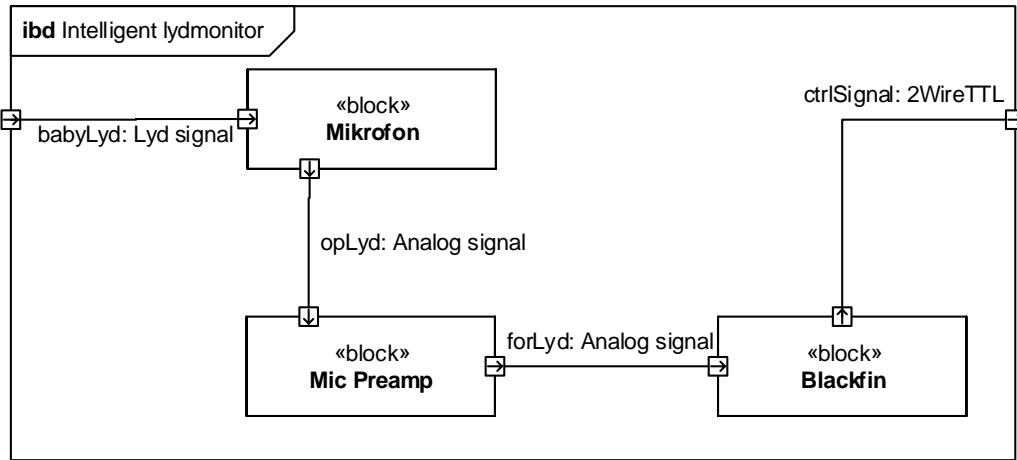
- Babyens gråd detekteres med en mikrofon og forstærkes med en mikrofonforstærker inden det analoge signal konverteres til diskrete samples med Blackfin's ADC
- Den diskrete sample-sekvens forfiltreres. Sample-sekvensen lavpasfiltreres og ned-samples for herefter at blive højpasfiltreret. Det forfiltrerede sample gemmes i bufferen, der skal holde 5 sekunders optagelse.
- Den filtrerede sample-sekvens i bufferen analyseres først for Sound Pressure Level (SPL) for at bedømme hvorvidt der er lyd nok til at den efterfølgende FFT kan betale sig.
- Et evt. resultat af FFT'en kategoriseres som én af tre BABYCON-states, som beskrevet i 2.1
- Den fundne BABYCON state holdes op mod en statistik over tidligere states for at sikre mod evt. fejl-state.
- Det endelige kategoriseringsresultat sendes til Controller via TwoWireCom



Figur 5.13. Overordnet virkemåde for Intelligent Lydmonitor

5.2.1 Hardware arkitektur

I dette afsnit beskrives hardware arkitekturen for Intelligent Lydmonitor. Den er skitseret IBD'et herunder:



Figur 5.14. IBD for Intelligent Lydmonitor

Som det ses af Figur 5.14 består **Intelligent Lydmonitor** af tre dele:

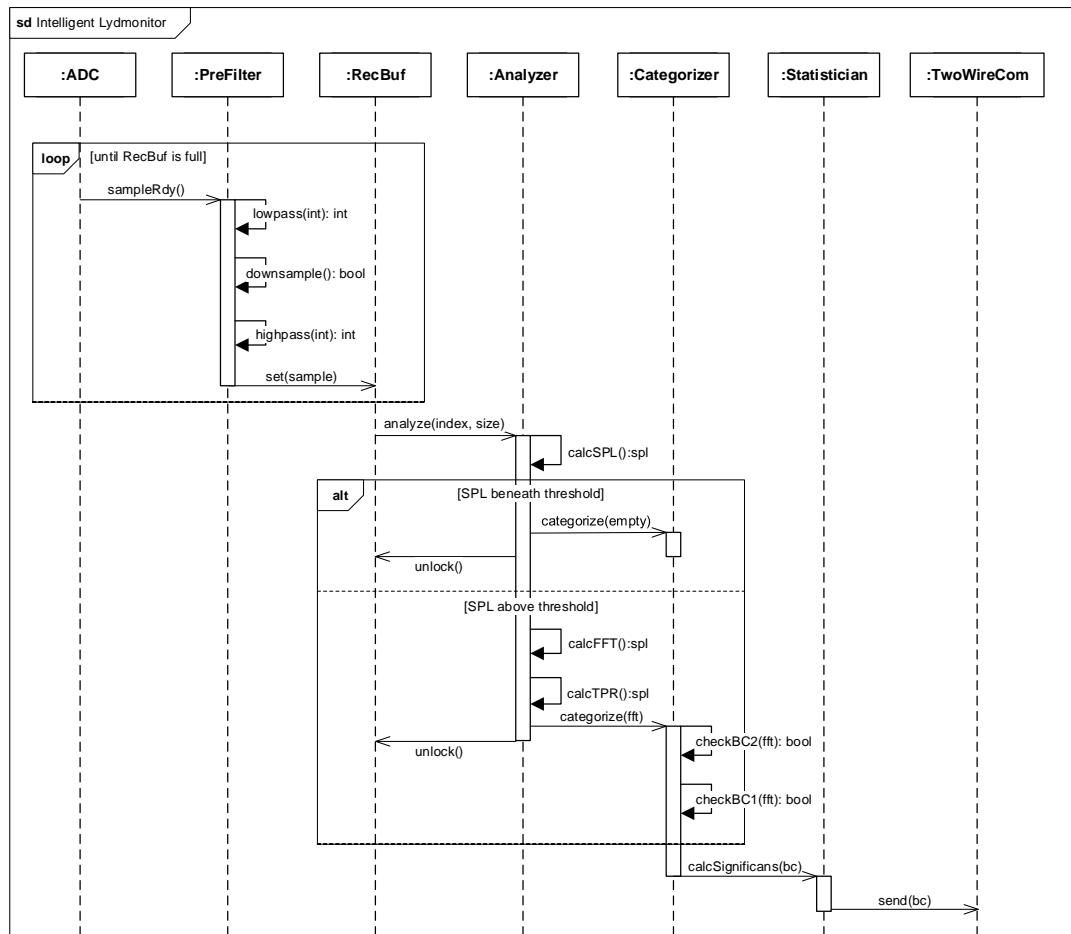
- **Mikrofon** optager signalet babyLyd, som er lyden Baby producerer.
- **Mic Preamplifier** modtager og forstærker signalet opLyd fra Mikrofon, hvilket er den lyd Mikrofon har optaget.
- **Blackfin** modtager forstærket lyd, forLyd, fra Mic Preamplifier. Blackfin analyserer lyden og kategoriserer denne, inden den via to signalledere sender kategoriseringen til Controller.

5.2.2 Software arkitektur

I dette afsnit beskrives software-arkitekturen for Intelligent Lydmonitor. Det vil tage udgangspunkt i 5.13 og består af et sekvensdiagram, et klassediagram og en efterfølgende funktionsbeskrivelse.

Sekvensdiagram

Herunder ses sekvensdiagrammet for en komplet lyd-analyse i Intelligent Lydmonitor.



Figur 5.15. SD for Intelligent Lydmonitor

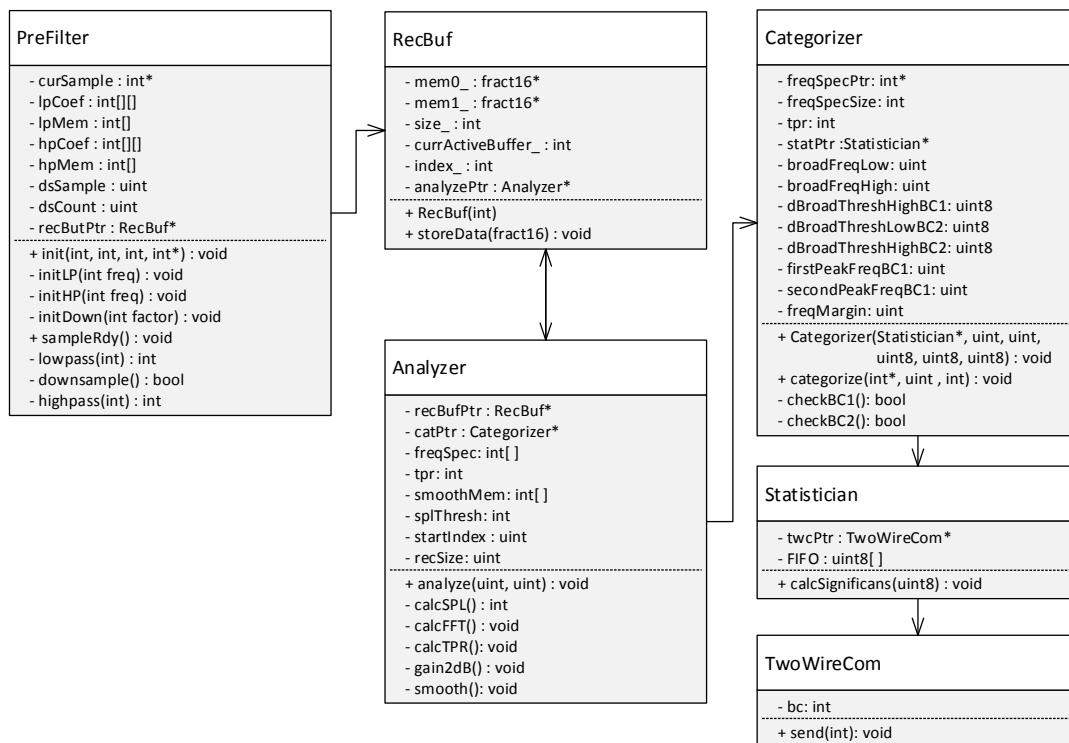
5.15 viser interaktionen mellem Intelligent Lydmonitors subklasser.

Klassen **PreFilter** står for forfiltreringen af sample-sekvensen. PreFilter har tre funktioner, `lowpass()` som lavpasfiltrerer sample-sekvensen, `downsample()` som nedsamplet sekvensen og `highpass()` som højpasfiltrerer den nedsamplede sekvens af hensyn til lavfrekvent støj fra vind og vejr. PreFilter kalder efter sit gennemløb RecBuf's funktion `set()` for at gemme samplen. **RecBuf** er cirkulær buffer med mulighed for midlertidig locking af bufferområder. Det er altså en buffer der muliggør påfyldning af data ét sted mens aflæsning foregår fra et låst segment. Se ?? for yderligere beskrivelse. Når RecBuf har indsamlet nye samples

svarende til 5 sekunders optagelse kaldes *analyze()*. **Analyzer** analyserer nu data i RecBuf via funktionerne *calcSPL()* og dernæst *calcFFT*, hvis SPL er over threshold. Resultatet sendes til **Categorizer** via funktionen *categorize*, der kategoriserer resultatet indenfor de tre BABYCON niveauer (1,2 og 3). Funktionen *calcSignificans* i **Statistician** kaldes hørnæst og det nyligt fundne BABYCON-niveau sammenlignes med tidligere niveauer for at finde sænke risiko for en evt. fejlmelding. Til slut sendes det fundne BABYCON-niveau til Controller via **TwoWireCom**.

Klassediagram

I UML-klassediagrammet herunder ses en oversigt over system-klassernes attributter og funktioner samt deres indbyrdes relationer.



Figur 5.16. UML for Intelligent Lydmonitor

Funktionsbeskrivelser findes i det efterfølgende afsnit.

Funktionsbeskrivelse

PreFilter

Ansvar: Klassen indeholder funktionaliteten til forfiltrering og nedSampling af den diskrete sampling-sekvens.

Funktion	<code>void init(int lpFreq, int hpFreq, int factor, int *curSample)</code>
Parametre	int lpFreq knækfrekvens for LP-filter, int hpFreq knækfrekvens for HP-filter, int factor nedsamplingsfaktor, int *curSample pointer til placering af nyeste sample
Returværdi	Ingen
Beskrivelse	Initialiserer filtre og downsampling, ved at kalde deres init-funktioner

Funktion	<code>void initLP(int freqLP)</code>
Parametre	int lpFreq knækfrekvens for LP-filter
Returværdi	Ingen
Beskrivelse	Initialiserer LP-filter

Funktion	<code>void initHP(int freqHP)</code>
Parametre	int hpFreq knækfrekvens for HP-filter
Returværdi	Ingen
Beskrivelse	Initialiserer HP-filter

Funktion	<code>void initDown(int factor)</code>
Parametre	int factor faktor for nedsampling
Returværdi	Ingen
Beskrivelse	Initialiserer downsampling

Funktion	<code>void sampleRdy(void)</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Kaldes ved erhvervelse af nyt sample. Kalder funktionerne lowpass(), highpass() og downsample()

Funktion	<code>int highpass(int sample)</code>
Parametre	Int sample, sample til HP-filtrering
Returværdi	int, værdi efter filtrering
Beskrivelse	Højpasfiltrerer sample

Funktion	<code>int lowpass(int sample)</code>
Parametre	Int sample, sample til LP-filtrering
Returværdi	int, værdi efter filtrering
Beskrivelse	Lavpasfiltrerer sample

Funktion	<code>bool downsample(void)</code>
Parametre	Ingen
Returværdi	bool, 1 hvis det givne sample skal beholdes og 0 hvis det skal frasorteres
Beskrivelse	Frasorterer samples jf. valgt nedsampling

RecBuf

Ansvar: For at kunne opsamle data, mens der bliver behandlet andet data, er det valgt at benytte multiple buffering. Der fyldes i en buffer til denne er fuld, dette data sættes til behandling mens den næste buffer fyldes osv.

Funktion	<code>RecBuf(int size)</code>
Parametre	int size, størrelsen a buffer
Returværdi	Ingen
Beskrivelse	Constructor allokerer og initialiserer buffer, mem0 og mem1 med størrelsen size

Funktion	<code>void storeData(fract16 sample)</code>
Parametre	fract16 sample: Diskret værdi som repræsenterer det momentane spændingsniveau på Blackfin 533 ADC indgang efter filtrering
Returværdi	Ingen
Beskrivelse	Gemmer diskret sample i RecBuf med pladsen index_

Analyzer

Ansvar: Klassen Analyzer indeholder analyse-funktionerne.

Funktion	<code>void analyze(uint startIndex, uint size)</code>
Parametre	uint startIndex, startindex for datatilgang i RecBuf uint size, størrelse på datasekvens
Returværdi	Ingen
Beskrivelse	Sætter startIndex og recSize og kalder funktionerne calcSPL() og hvis SPL er over splThresh kaldes calcFFT(). Slutteligt kaldes categorize() med analyseresultatet

Funktion	<code>int calcSPL(void)</code>
Parametre	Ingen
Returværdi	int, værdi af SPL
Beskrivelse	Beregner Sound Pressure Level af samplesekvensen fra RecBuf. Denne tilgås via RecBuf's get()

Funktion	<code>void calcFFT(void)</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Beregner frekvensspektret via en fast fourier transformation af samplesekvensen fra RecBuf. Denne tilgås via RecBuf's get(). Herefter kaldes gain2dB() og smooth() for at klargøre data til efterfølgende kategorisering. Frekvensspektret gemmes i freqSpec

Funktion	<code>void gain2dB(void)</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Konverterer frekvensspektret freqSpec fra gg til dB

Funktion	<code>void smooth(void)</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Udglatter frekvensspektret vha. et rekursivt midlingsfilter

Funktion	<code>void calcTPR(void)</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Beregner TPR (Tonal power ratio) for frekvensspektret freqSpec

Categorizer

Ansvar: Klassen Categorizer står for BABYCON kategorisering af resultaterne fra Analyzer

Funktion	<code>Categorizer(Statistician* statPtr, uint broadFreqLow, uint broadFreqHigh, uint dBroadThreshHighBC1, uint dBroadThreshLowBC2, uint dBroadThreshHighBC2, uint firstPeakFreqBC1, uint secondPeakFreqBC1, uint freqMargin)</code>
Parametre	Statistician* statPtr, pointer til klassen Statistician uint broadFreqLow, lavest tilladelige værdi for frekvenshældning for BABYCON2 uint broadFreqHigh, højst tilladelige værdi for frekvenshældning for BABYCON2 uint dBroadThreshHighBC1, høje threshhold for frekvenshældning ved BABYCON1 uint dBroadThreshLowBC2, lave threshhold for frekvenshældning ved BABYCON2 uint dBroadThreshHighBC2, høje threshhold for frekvenshældning ved BABYCON2 uint firstPeakFreqBC1, analysefrekvensen for frekvensspektrets første peak uint secondPeakFreqBC1, analysefrekvensen for frekvensspektrets anden peak uint freqMargin, margin til bestemmelse af frekvenspeaks
Returværdi	Ingen
Beskrivelse	Constructor sætter relevante kategoriseringsparametere

Funktion	<code>void categorize(int* freqSpecPtr, uint freqSpecSize, int tpr)</code>
Parametre	int* freqSpecPtr, pointer til frekvensspektrum uint freqSpecSize, størrelse af freqSpec int tpr, tonal power ratio
Returværdi	Ingen
Beskrivelse	Kategoriserer BABYCON-niveau på baggrund af frekvensspektrum og TPR. Kalder Statisticians calcSignificans() med BABYCON-niveau

Funktion	<code>void categorize(void)</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Hvis categorize kaldes uden parametre kaldes calcSignificans() med BABYCON3

Funktion	<code>bool checkBC1(void)</code>
Parametre	Ingen
Returværdi	bool, true hvis BABYCON1
Beskrivelse	Returnerer true hvis BABYCON1

Funktion	<code>bool checkBC2(void)</code>
Parametre	Ingen
Returværdi	bool, true hvis BABYCON2
Beskrivelse	Returnerer true hvis BABYCON2

Statistician

Ansvar: Klassen Statistician står for at beregne mest signifikante BABYCON status. Giver besked til TwoWireCom.

Funktion	<code>void calcSignificans(uint bc)</code>
Parametre	uint bc, BABYCON-niveau
Returværdi	Ingen
Beskrivelse	Indsætter den modtagne BABYCON i FIFO-kø og beregner mest signifikante BABYCON status, sender aktiv besked til TwoWireCom

TwoWireCom

Ansvar: Klassen TwoWireCom står for at sende BABYCON-niveau til Controller jf. 3.3.2

Funktion	<code>void send(int bc)</code>
Parametre	int bc: Nuværende BABYCON-niveau
Returværdi	Ingen
Beskrivelse	Sender BABYCON-niveau til Controller

Design og implementering

I det følgende afsnit beskrives design og implementering af Intelligent Lydmonitors moduler.

5.3 HW design og implementering

Design og implementering af HW-moduler er beskrevet i dette afsnit.

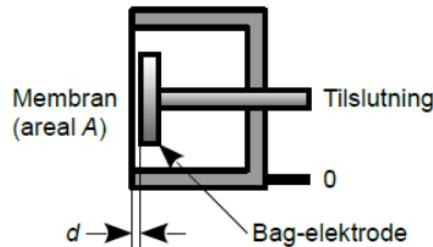
5.3.1 Mikrofon preamp

Mikrofon

Som mikrofon er valgt en MCE-100 elektret mikrofon. Et udpluk af specifikationer ses herunder:

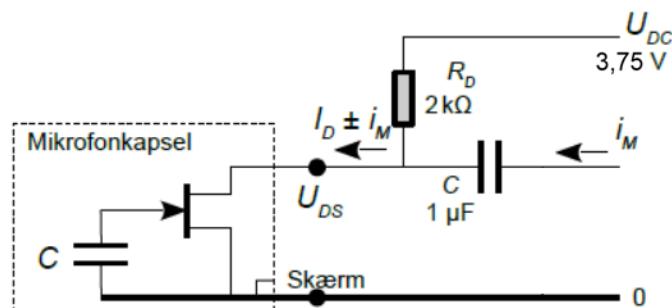
- Frequency range: 50 Hz to 10 kHz
- Sensitivity: 5,6 mV / Pa / 1 kHz
- Output impedans: 2 kOhm
- Power Supply: 1.5 to 10 V dc, 5 mA

På Figur 5.17 ses en tegning af en elektret mikrofon som er en videreudvikling af kondensatormikrofonen, hvor bagelektroden har en "indfrosset" ladning i materialet. Kapaciteten i mikrofonen (ca. 10 pF) ændres ved trykvariationer idet afstanden mellem membranen og bagelektroden varieres.



Figur 5.17. Tegning af elektrektrostatisk mikrofon. Kilde: Analogteknik

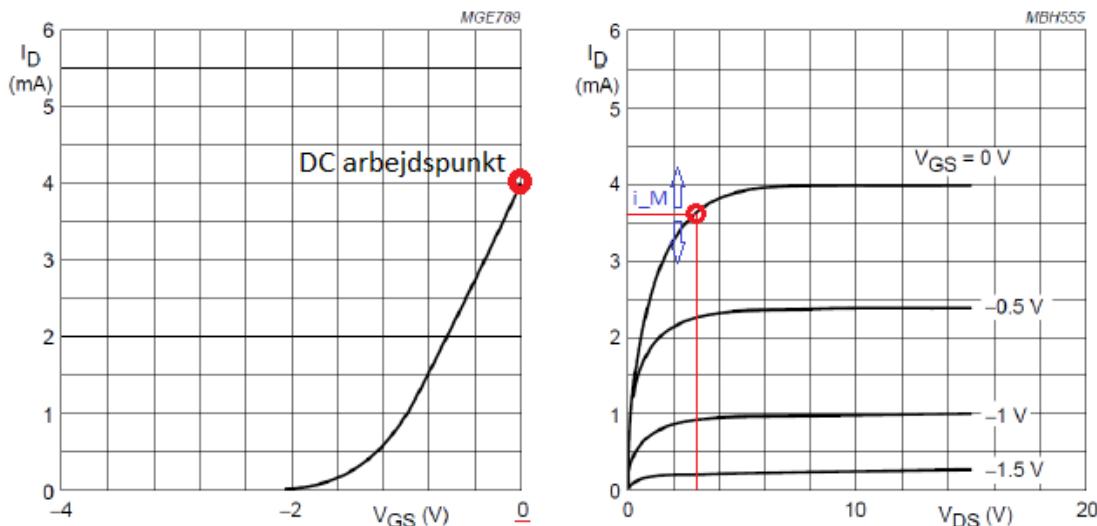
Dog har kablingen mellem mikrofonen og forstærkeren en betydelig kapacitet, så mikrofonpakken er implementeret med en indbygget JFET som buffer, som det ses af Figur 5.18



Figur 5.18. Elektret forsyningskredsløb

Forsyningen til mikrofonen bliver det halve af Blackfin-kittets 7,5 V, således at det efterfølgende forstærker-kredsløb kan implementeres med +/- 3,75 V forsyning. Således $U_{DC} = 3,75$ V. Vi ved fra databladet for mikrofonen, at den trækker 5mA og at modstanden $R_d = 2$ kOhm (opgivet som "Output Impedance").

Dette giver et spændingsfald på 0.8 V over R_d og dermed 2,95 V over den i mikrofonkapslen indbyggede JFET. I dette område af U_{DS} vil transistoren have en strømbegrænsnende virkning og kan derfor bruges som en tilnærmelsesvis lineær strømkilde. Herunder ses for illustration af principippet en JFET med DC arbejdspunktet indtegnet.



Figur 5.19. Karakteristik for JFET'en BF245A. Kilde: Analogteknik

Uden lydtryk vil $V_{GS} = 0$ V, hvilket resulterer i en $I_D = 4$ mA (markeret på venstre graf i figur 5.19). Dette DC-arbejdspunkt er for vores mikrofon 5 mA. V_{DS} er ca. 3 V og det ses på højre graf at strømmen i arbejdspunktet, I_D , vil ligge lidt lavere. Småsignal strømmen, i_M , vil således lade strømmen variere omkring dette punkt.

For figur 5.18, punktet på højre side af kondensatoren C, vil DC-strømmen være sorteret fra og den af mikrofonen modulerede AC-strøm, i_M , vil være at finde. Af databladet ved vi, at mikrofonens sensitivitet er $S = 5,6$ mV/Pa, og i_M vi således være givet ved:

$$i_M = \frac{S}{R_D} \cdot p$$

Hvor p er lydniveau (Pa)

PreAmp

Det er PreAmpens opgave at omdanne modulationsstrømmen i_M til en line level spænding for Blackfin's ADC

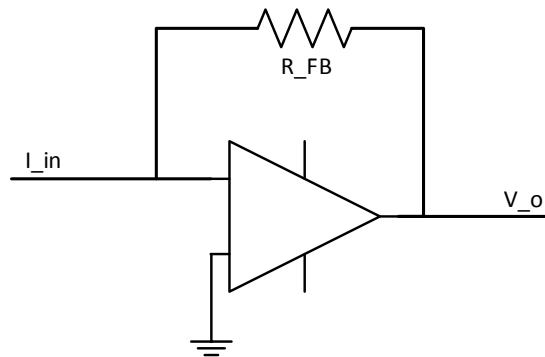
Blackfin ADC'en tager et line-level input på +/- 1,65 V. Det ønskes at udnytte det maksimale dynamiske område uden at lade signalet klippe. Det regnes med at mikrofonen ikke udsættes for mere end 2 Pa ved almindelig brug. Den maksimale strøm-amplitude bliver derfor:

$$i_M = \frac{5,6mV/Pa}{2k\Omega} \cdot 2Pa = 5,6\mu A$$

Dette signal skal forstærkes med en TIA op til det ønskede line level på 1,65 V. Den ønskede forstærkning, G, bliver således:

$$G = \frac{1,65V}{5,6\mu A} = 2,89 \cdot 10^5 \frac{V}{A}$$

Der benyttes en transimpedansforstærker til at realisere denne forstærkning.



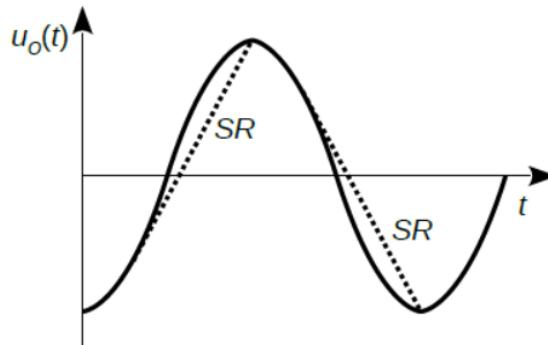
Figur 5.20. Transimpedansforstærker: $V_o = R_{FB} \cdot I_{in}$

Den ønskede transimpedansforstærkning er givet direkte ved værdien af feedbackmodstanden, R_{FB} .

$$R_{FB} = \frac{V_o}{I_{in}} = G = 289k\Omega$$

Valg af OpAmp

Da kredsen skal bruges til lydbehandling, er forstærkerens *Slew Rate* specifikation vigtig. Operationsforstærkerens interne kondensator, C_c , udgør en begrænsning for hvor hurtigt udgangen kan flytte sig, og for lydbehandling skal denne være så høj som muligt.



Figur 5.21. Illustration af OpAmp parameteren "Slew Rate". Kilde: Analogteknik

En forstærkers slew rate er givet ved:

$$SR = \left[\frac{du}{dt} \right]_{MAX} \Rightarrow SR = 2 \cdot \pi \cdot f \cdot U_M$$

Hvor f er højeste arbejdsfrekvens og U_M er udgangsspændingens amplitude. Slew Rate er således en værdi for hvor hurtigt operationsforstærkeren kan flytte udgangen et antal volt (målt i MV/s). For vores applikation har vi en $U_M = 1.65$ V, og som båndbredde, f , der vælges en konservativ værdi på 20 kHz, i det tilfælde, at kredsen senere bruges med en bedre mikrofon. Dette giver følgende SR:

$$SR = 2 \cdot \pi \cdot 20\text{kHz} \cdot 1,65\text{V} = 0,2\text{MV/s}$$

Ved denne SR påkræves dog et meget kraftigt indgangssignal, der også vil resultere i høj forvrængning. For at sikre en forvrængning på under 1% bør $u_M < 20\text{mV}$.

$$u_M = (2k\Omega || 22k\Omega) \cdot 5,6A = 10,3\text{mV}$$

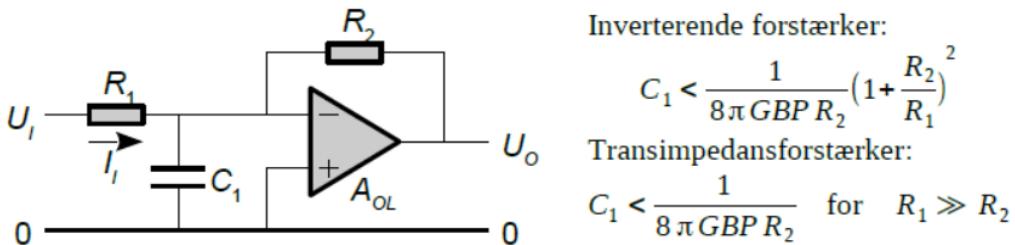
Dette krav er altså opfyldt.

Ved en grænse på 20mV vil 20 % af differentialtrinnets udstyringsmulighed på $\pm I_E$ udnyttes. Den påkrævede SR vil derfor være 5 gange den hidtil beregnede, altså 1MV/s.

Dette betyder altså at vi kan nøjes med at bruge en billig operationsforstærker uden større krav til SR, såsom en OpAmp fra den i lydbehandling almindeligt anvendte TL071-serie ville give ($SR = 13\text{MV/s}$).

Stabilitet

Der er en betydelig kapacitet på forstærkerens indgang idet mikrofonen er koblet med et coax-kabel.

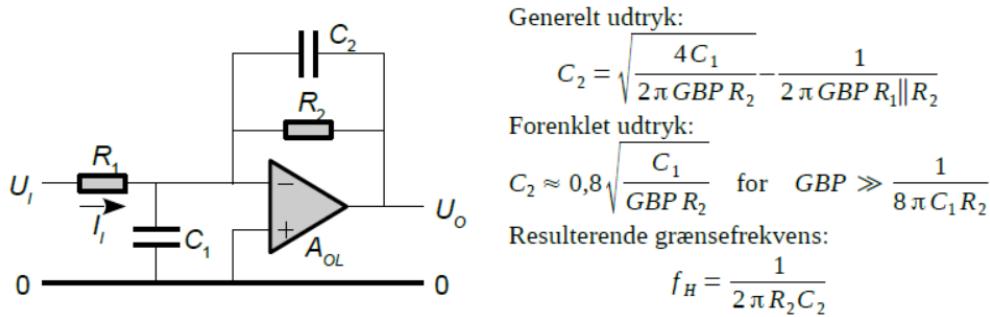


Figur 5.22. OpAmp med kapacitiv belastning på indgangen. Kilde: Analogteknik

På figur 5.22 ses TIA'en som de næste udregninger tager udgangspunkt i. Mikrofonen er kobles med ca. 1m coax kabel med kapaciteten 101pF/m. Desuden foregår der en HF filtrering med et RC filter og en 100nF kondensator i parallel mellem signal og stel. Se afsnittet med særlige EMC-hensyn INDSÆT REF. Dette giver altså en C_1 på 100 pF. Der regnes med en typisk GBP på 9MHz. Der testes for nødvendighed af kondensator i tilbagekobling:

$$C_1 < \frac{1}{8\pi \cdot GBP \cdot R_2} \implies 101nF < \frac{1}{8\pi \cdot 9\text{MHz} \cdot 28,9k\Omega} \implies 101 \cdot 10^{-9} < 153 \cdot 10^{-15}$$

Det er altså nødvendigt at sætte en kondensator i tilbagekoblingen, som illustreret i 5.23.



Figur 5.23. OpAmp med kapacitiv belastning på indgangen og stabiliserende kondensator i tilbagekoblingen. Kilde: Analogteknik

Feedback kondensatoren beregnes som følger:

$$C_2 \approx 0,8 \sqrt{\frac{C_1}{GBP \cdot R_2}} \implies C_2 = 0,8 \sqrt{\frac{101nF}{9MHz \cdot 28,9k\Omega}} = 496pF$$

Der benyttes en 500 pF kondensator. Den *resulterende grænsefrekvens* bliver således:

$$f_H = \frac{1}{2\pi R_2 C_2} \implies f_H = \frac{1}{2\pi \cdot 28,9k\Omega \cdot 18pF} = 11,01kHz$$

Dette begrænser altså ikke netop ikke signalets båndbredde.

5.3.2 ADC forfiltrering

For at tilpasse signalet til ADC'en i Blackfin533 bedst muligt, laves der en filtrering af signalet inden ADC'en. Ordenen af dette filter beregnes ud fra ADC'ens SNR (Signal To Noise Ratio). Blackfin's ADC er en AD1836, hvilken kan køre med henholdsvis 16-/18-/20-/24-Bit data. Herudover understøtter den en samplerate på 96KHz. Alle specifikationer for AD1836 kan ses i **INDSÆT REFERENCE TIL DATASHEET AD1836**.

Til beregning af AD1836's SNR benyttes følgende formel:

$$SNR_{ADC} = 6.02 \cdot N + 1.78dB$$

Hvor N er ADC'ens antal bits.

I dette projekt arbejdes med 16-Bit og AD1836's SNR bliver heraf:

$$SNR_{AD1836} = 6.02 \cdot 16 + 1.78dB = 98.1dB$$

Der omregnes nu fra dB til gain:

$$\begin{aligned} Gain &= 10^{\frac{SNR_{ADC}}{20}} \\ Gain &= 10^{\frac{98.1}{20}} = 80352gg \end{aligned}$$

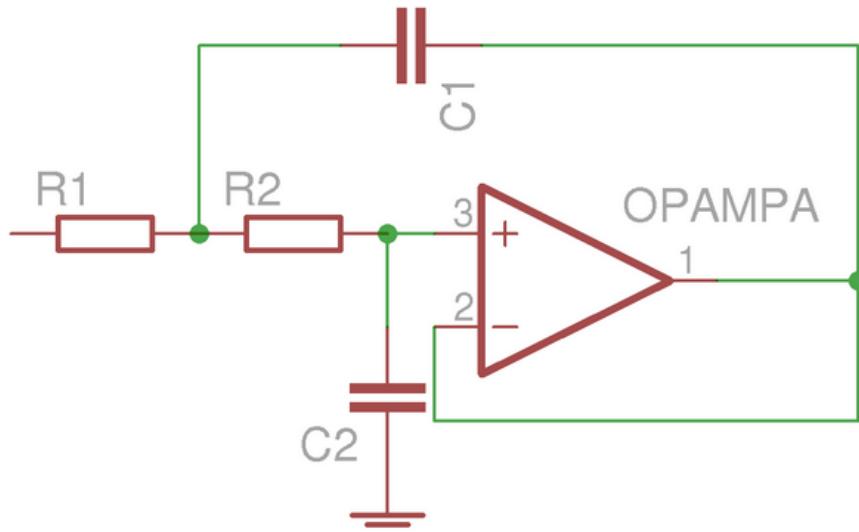
Når gain er fundet, er det herefter muligt at beregne ordnen på filtret. Dette gøres ud fra dæmpningsformlen.

$$\text{Damping} = \left(\frac{f_s}{f_c} \right)^M \geq \text{Gain}$$

$$\text{Damping} = \left(\frac{96\text{kHz}}{7\text{kHz}} \right)^M \geq 80352 \rightarrow M = 5.9$$

Hvor M er ordnen af filtret, fc er knækfrekvensen og fs/2 er den halve samplingsfrekvens.

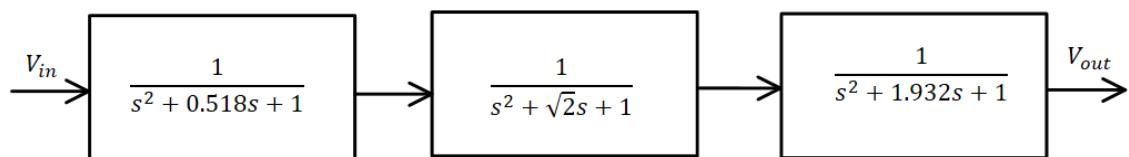
Som det ses heraf skal der implementeres et 6. ordens lavpasfilter for den bedste tilpasning. Dette er en høj orden, men det kan lade sig gøre at implementere analogt, så længe man tager højde for de forskellige poler. Dette gøres ved at kaskadekoble tre 2. ordens butterworth filtre af samme type som på Figur 5.24



Figur 5.24. Standard anden ordens filter

Der følges en design procedure hvor filtret designes med modstandsværdier på 1 ohm, samt en knækfrekvens på 1 rad/s. Når filtret er designet med disse værdier skalleres modstande og kondensatorer så de passer den ønskede knækfrekvens på 7kHz.

Blokdiagrammet for 6. ordens filter er vist på Figur 5.25



Figur 5.25. Blokdiagram for 6. ordens LP-filter

Hver blok er på standardformen:

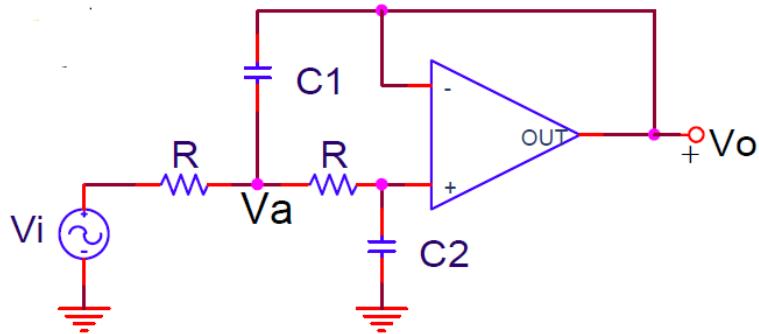
$$\frac{1}{a \cdot s^2 + b \cdot s + k}$$

Polynomiet for 6. ordens filter ses herunder:

$$(s^2 + 0.518s + 1)(s^2 + \sqrt{2}s + 1)(s^2 + 1.932s + 1)$$

Proceduren for udregning af polynomiet kan findes i **INDSÆT REFERENCE TIL BILAG HER**. Heri findes tabellen hvor polynomier op til 6. orden kan slås op.

Der foretages en kredsløbsanalyse for at finde overføringsfunktionen for et anden ordens lavpasfilter:



Figur 5.26. Anden ordens filter med knudepunkter

Følgende kredsløbsligninger kan opstilles ud fra Figur 5.26

$$\frac{V_a - V_i}{R} + (V_a - V_o) \cdot sC_1 + \frac{(V_a - V_o)}{R} = 0$$

$$2 + RC_1 V_a - (1 + RC_1 s) V_o = V_i$$

$$-V_a + (1 + RC_2 s) V_o = 0$$

$$V_o s C_2 + \frac{(V_o - V_a)}{R} = 0$$

Ud fra disse kredsløbsligninger kan V_o isoleres:

$$V_o = \frac{V_i}{R^2 C_1 C_2 s^2 + 2 R C_2 s + 1}$$

Herefter findes overføringsfunktionen:

$$H(s) = \frac{V_o}{V_i} = \frac{\frac{1}{R^2 C_1 C_2}}{s^2 + \frac{2}{R C_1} s + \frac{1}{R^2 C_1 C_2}}$$

I det vi har sat modstandsværdien R til 1 ohm fås følgende:

$$H(s) = \frac{\frac{1}{C_1 C_2}}{s^2 + \frac{2}{C_1} s + \frac{1}{C_1 C_2}}$$

C1 og C2 kan nu udledes:

$$b = \frac{2}{C_1} \text{ og } k = \frac{1}{C_1 C_2}$$

Ud fra dette kan C1 og C2 beregnes for hver filterblok.

Blok 1

$$0.518 = \frac{2}{C_1} \Rightarrow C_1 = 3.86F$$

$$1 = \frac{1}{C_1 C_2} \Rightarrow C_2 = 0.259F$$

Blok 2

$$C_1 = \sqrt{2}F$$

$$C_2 = 0.707$$

Blok 3

$$C_1 = 1.035F$$

$$C_2 = 0.966F$$

Nu er 6. ordens filter med modstandsværdierne 1 ohm og knækfrekvensen 1 rad/s færdig designet. Vi skal nu skallere værdierne af modstande og kondensator med to skalleringsfaktorer km og kf.

Det ønskes at modstandsværdien skal være på 1kohm og km bliver heraf:

$$k_m = 1000$$

Frekvens skalleringsfaktoren kf findes herunder:

$$k_f = \frac{\omega_c}{1 \frac{\text{rad}}{\text{s}}} = 2 \cdot \pi \cdot 7000$$

De nye modstands- og kondensatorværdier findes ved følgende formler:

Ny modstandsværdi

$$R' = R \cdot k_m$$

Ny kondensatorværdi

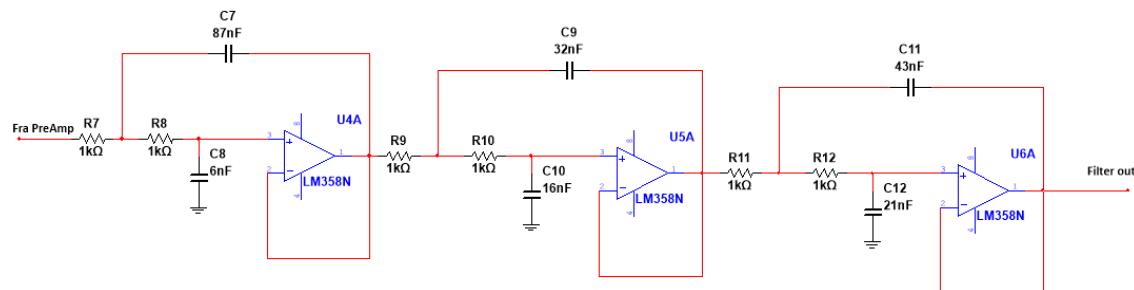
$$C' = \frac{C}{k_m \cdot k_f}$$

De nye værdier kan ses i Tabel 5.1

Tabel 5.1. Nye værdier

Værdi	Blok 1	Blok 2	Blok 2
Modstand R1	1 kohm	1 kohm	1 kohm
Modstand R2	1 kohm	1 kohm	1 kohm
Kondensator C1	87 nF	32 nF Hz	43 nF
Kondensator C2	6 nF	16 nF Hz	21 nF

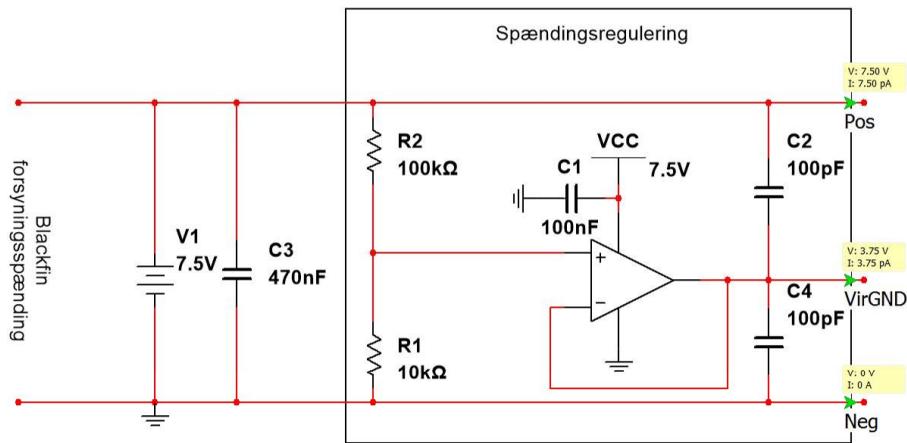
Filtret er opbygget i multisim og kan ses på Figur 5.27



Figur 5.27. 6. ordens kaskadekoblet filter

5.3.3 Spændingsregulering

Fra systemets overordnede strømdistribution modtages en 7,5 V DC-forsyning der kan levere op til 1,5 A. I laboratoriet er Blackfin533-kittets strømforsyning målt til 1,23 A, og de 250 mA regnes for tilstrækkeligt til kredsløbene med mikrofon preamp og lavpasfiltrering før ADC. Blackfin533 får sit strømudtag direkte fra den modtagne DC-forsyning. Dog indsættes en 470 nF kondensator til udglatning af signalet ved strømfluktuation. De integrerede OpAmps i preamp og lavpasfilter, forsyneres ligeledes med 0 og 7,5V, men der skal genereres et arbejdspunkt på 3,75 V, der skal tjene som virtuel ground. Ud fra et signalbehandlingsmæssigt synspunkt forsynes disse således med +/- 3,75 V. Det virtuelle nulpunkt genereres ved en spændingsdeling og en spændingsfølger, så en belastning af det virtuelle nulpunkt ikke ændrer spændingen. En simulering af kredsløbet for denne opstilling ses herunder:



Figur 5.28. Spændingsregulerings kredsløb.

5.4 SW design og implementering

Design og implementering af SW-moduler er beskrevet i dette afsnit.

5.4.1 Generelt

Datatyper

For på letteste og mest effektive vis at implementere filter og fft funktioner på systemet, uden at skulle programmeres inline assembly, gøres brug af native fixed-point typen **fract**, som defineret i Kapitel 4 af *Extensions to support embedded processors* ISO/IEC Technical Rapport 18037. Denne type har altså radix punktet i en fast position med et antal fractionelle bits over nul.

Herunder ses en tabel over fract type, repræsentation og range:

Type	Repræsentation	Range
short fract	s1.15	[-1.0,1.0)
fract	s1.15	[-1.0,1.0)
long fract	s1.31	[-1.0,1.0)

Typerne **short fract** og **fract** er altså 16 bit typer med et signed bit på venstre side af radix punktet og 15 brøk-bits på højre side. Dens range er større end -1.0 og skarpt mindre end 1. **long fract** har 31 brøk-bits og har således højere opløsning.

5.4.2 Prefilter

Prefilter står for forfiltreringen af signalet modtaget fra Blackfin-kittets ADC inden det gemmes i RecBuf.

Denne forfiltrering består af

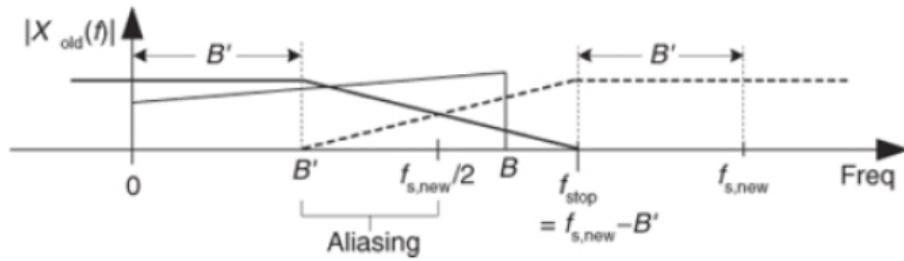
- Decimation bestående af lavpasfiltrering og downsampling
- Højpasfiltrering til fra filtrering af lavfrekvent støj såsom vind.

Decimation

Blackfin kittets AD1836 kan sample enten med 48kHz eller 96 kHz. Idet mikrofonens båndbredde er 10kHz, tillader shannons samplings sætning os at opnå en beregningsmæssig besparelse ved at decimere fra 48kHz til et 24kHz signal (min 20kHz er nødvendig). Vores range of interest bliver 0-10 kHz.

Lavpasfiltrering

Det ønskes at lave et FIR lavpasfiltr, der dæmper ripple fra det spektrale spejlings-replika omkring den nye sample-frekvens med ca. 60dB. I figurudsnittet herunder ses en illustration af lavpasfilterets frekvensrespons relativt til båndbredden B' :



Figur 5.29. Lavpasfilterets frekvensrespons relativt til båndbredden B' . Kilde: Lyons, fig. 10.2

Vi kan beregne den nødvendige filterorden ved Lyons Eq.5-49:

$$N_{FIR} \approx \frac{\text{Atten}}{22 \cdot (f_{stop} - f_{pass})}$$

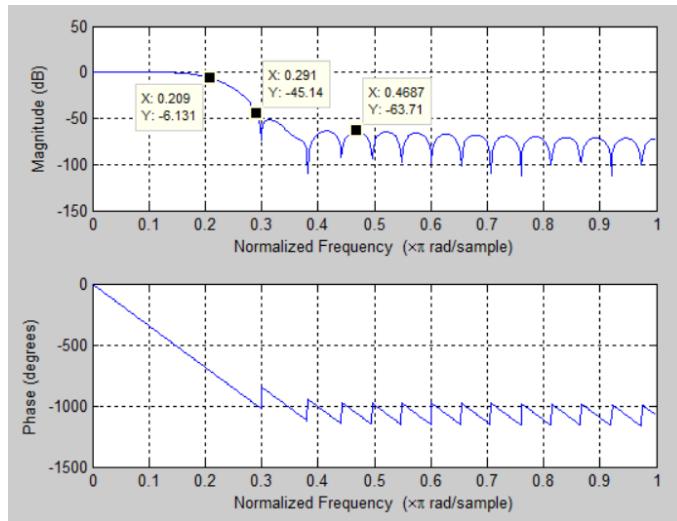
Hvor Atten er den ønskede dæmpning i dB og f_{stop} og f_{pass} er frekvenser normaliseret til fs . I vores tilfælde bliver $N_{FIR} = 33$. Dette er relativt småt, og en two-stage decimation er ikke nødvendig for at sænke den totale orden.

Filteret beregnes i matlab med funktionen `fir1()` som følger:

```
%**** LOW PASS ****
atten = 60; %ønsket dæmpning i dB
f_pass = bw/fs %freq normalized to fs
f_stop = (fs_new - bw)/fs %freq normalized to fs
N_fir = ceil(atten/(22*(f_stop-f_pass)))

b = fir1(N_fir, f_pass);
figure(2), freqz(b)
%*****
```

Denne filterorden når dog ikke helt ønskede dæmpning ved f_{stop} og en orden 5 lægges oveni. Følgende filter-karakteristik opnås:



Figur 5.30. Frekvensrespons for designet lavpasfilter

Markeret på figuren er f_{pass} , f_{stop} og et punkt midt i stopbåndet. Det ses at der opnås ca. 45 dB dæmpning ved f_{stop} , men at den generelle dæmpning i pasbåndet er ca. 63 dB. Dette regnes for tilstrækkeligt

Downsampling

Nedsamplingen sker ved en faktor 2 og er således ligetil: Et evt. polyphase filter er unødvendigt

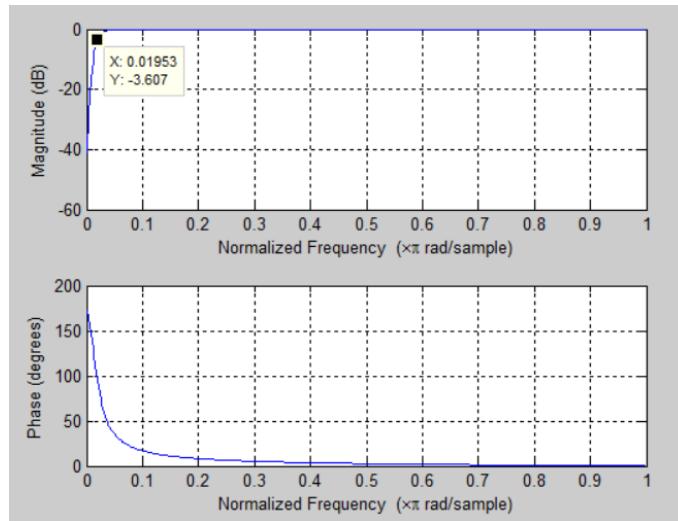
Højpasfiltrering

Af forundersøgelsen konkluderes det, at frekvenser under 500 Hz ikke skulle benyttes til bestemmelse af babygråd. Der designes således et 2. orden IIR butterworth højpasfilter til frafiltrering af disse. Filtret designes med matlabs `butter()` funktion som følger:

```
%***** HIGH PASS *****
order = 2;
f_hp = 500;
f_cutoff = f_hp/fs_new;

[b,a] = butter(order, f_cutoff,'high')
figure(3),freqz(b,a)
%*****
```

Følgende filterkarakteristik opnås:



Figur 5.31. Frekvensrespons for designet højpasfilter

Markeret på figuren er f_{cutoff} .

Implementering på blackfin

Datatype

Filterkoefficienterne beregnes i matlab i formatet `float`. Ved implementering på blackfin omkonverteres disse værdier til typen `fract16` med funktionen `float_to_fr16`.

5.4.3 RecBuf

I dette afsnit beskrives principper og design for RecBuf i Intelligent Lydmonitor. Samples modtaget fra prefiltret gemmes i et ud af to identiske buffere som RecBuf indeholder. Når det første array er fyldt kaldes Analyzers `analyze()` med værdien 0 eller 1 hvilken repræsenterer hvilken buffer Analyzer skal analysere data fra. Indholdet af den buffer som Analyzer arbejder på skal altså være færdig analyseret inden den andet buffer er fyldt. Der opstår således et hastigheds-krav til analyze og et størrelseskrav til bufferne i Recbuf.

```
class RecBuf {
friend class Analyzer;
public:
RecBuf(int size_);
void storeData(fract16 sample_);
virtual ~RecBuf();

private:
Analyzer* analyzePtr;
int currActiveBuffer_;
int index_;
int size_;
fract16 *mem0_;
fract16 *mem1_;
```

```
};
```

Herover ses et udsnit fra headeren for kassen RecBuf. Det er her værd at lægge mærke til at klassen benytter **friend class Analyzer** for at Analyzer har mulighed for at tilgå privatdata i RecBuf. Af private attributter ses det at der er en pointer til klassen Analyzer, som skal bruges til at kalde funktionen **analyze()** ved fyldt buffer.

```
RecBuf::RecBuf(int size)
{
    currActiveBuffer_ = 0;
    index_ = 0;
    size_ = (size>0 && size<MAXSIZE ? size : MAXSIZE);

    //Creating array mem0 and mem1
    mem0_ = new fract16[size_] ();
    mem1_ = new fract16[size_] ();
}
```

Herover ses constructoren for RecBuf. Først initieres de to private attributter **currActiveBuffer_** og **index_** til 0. RecBuf modtager en int kaldet **size**. Det tjekkes at size er større end 0 og mindre end en define kaldet **MAXSIZE**, passer betingelsen sættes den private attribut **size_** lig med size og ellers sættes **size_** lig med **MAXSIZE**. Herefter oprettes der dynamisk to buffere af typen fract16 med størrelsen **size_** som initieres til 0.

```
RecBuf::~RecBuf()
{
    delete [] mem0_;
    delete [] mem1_;
}
```

Herover ses destructoren for RecBuf. Denne sørger for at delete det dynamisk allokerede hukommelse til **mem0_** og **mem1_** for at undgå memory-leak.

Klassen RecBuf indeholder kun en funktion kaldet **storeData()** denne funktion er implementeret som en switch case der tjekker på variablen **currActiveBuffer_**. Case 0 er vist herunder. Hvis den private variable **index_** er lig med **size_** betyder det at **mem0_** er fuld. I dette tilfælde sættes **index_** lig med 0 og **currActiveBuffer_** opdateres til 1 så den repræsenterer bufferen **mem1_**. Herefter sættes den første plads i **mem1_** med den værdien **sample**. Slutteligt kaldes **analyze** med et 0, da **mem0_** nu er fuld og klar til at blive analyseret. Hvis ikke **index_** er lig med **size_** sættes **mem0_[index_]** med **sample**. Slutteligt inkrementeres **index_**.

```
switch(currActiveBuffer_)
{
    case 0:
```

```

if(index_ == size_)
{
index_ = 0;
currActiveBuffer_ = 1;
mem1_[index_] = sample;
analyzePtr->analyze(0);
}
else
{
mem0_[index_] = sample;
}
index_++;
break;

```

Case 1 er vist herunder og er næsten magen til case 0. Forskellen på case 1 og case 0 er følgende: currActiveBuffer_ sættes til 0 for at repræsentere mem0_ i stedet for mem1_. analyze kaldes med 1 i stedet for 0.

```

case 1:
if(index_ == size_)
{
index_ = 0;
currActiveBuffer_ = 0;
mem0_[index_] = sample;
analyzePtr->analyze(1);
}
else
{
mem1_[index_] = sample;
}
index_++;
break;
}

```

5.4.4 Analyzer

SPL

I dette afsnit beskrives design og implementering af Analyzers funktion `calcSPL()`.

Princip

For at ikke at benytte unødig processorkraft på beregne FFT af en støj-signal, beregnes først et udtryk for hvor kraftigt signal, der modtages. Dette beregnes som dBV, hvor et signal med amplituden 1 V vil resultere i en SPL måling på 0 dBV. Den maksimale amplitude for input-signalet er 1.65 V. Målt i dBV er dette:

$$SPL_{max} = 20 \cdot \log_{10} \left(\frac{1.65V}{1V} \right) = 4,35 \text{ dBV}$$

Ligeledes vil et signal med amplituden 100 mV resultere i følgende SPL:

$$SPL_{100mV} = 20 \cdot \log_{10} \left(\frac{0.1V}{1V} \right) = -20 \text{ dBV}$$

Idet ADC'en kører i med en opløsning på 16 bit, hvoraf det første er sign, vil dette resultere i følgende værdier som fract16 for maximal amplitude og amplitude på 100 mV:

$$\begin{aligned} floatval_{1,65V} &= 1 - 2^{-15} = 0.9999694824 \\ fractVal_{100mV} &= \frac{2^{15}-1}{1.65V} \cdot 100mV \cdot 2^{-16} = 0.06060421105 \end{aligned}$$

Implementering

Det målte SPL-niveau skal kun tænktes som grov-filtrering, implementeres ganske simpelt ved at finde optagelsens højeste værdi. Analyzer-modulet initieres således med en forudbestemt SPL threshhold, der benyttes til at vurdere hvorvidt optagelsen indeholder signal nok til at en FFT kan betale sig.

5.4.5 Categorizer

5.4.6 Statistician

5.4.7 TwoWireCom

TwoWireCom er klassen som står for at sende BABYCON-niveauet til Controller via to GPIO pins.

Klassen er implementeret ved hjælp af funktioner fra T_GPIO_BANKS. Inkluderingen af de nødvendige biblioteker kan ses i kodestykket herunder.

```
#include "adi_initialize.h"
#include <sys\exception.h>
#include <cdefBF533.h>
#include "sysreg.h"
#include "ccblkfn.h"
```

Som det ses i kodestykket herunder, er klassen TwoWireCom simpel med kun en funktion *send(int bc)* som modtager et BABYCON-niveau i form af en int. Der benyttes to GPIO pins PF4 og PF6, disse skal repræsentere henholdsvis MSB og LSB jf. 3.2 . Yderlige specifikationer kan findes i databladet **INDSÆT REFERENCE** side 84.

```
class TwoWireCom {
public:
    TwoWireCom();
    virtual ~TwoWireCom();
    void send(int);
};
```

send(int bc) implementeres som en switch case, der switcher på de tre BABYCON-niveauer 1, 2 og 3. Som default sendes 0. Dette er koden for error. Det kan også være at funktionen er kaldt med en værdi der ikke er 1, 2 eller 3. pFIO_FLAG's benyttes til at sætte GPIO-pin højt eller lavt, ved at sætte flaget lig med den GPIO pin man ønsker høj eller lav.

```
void TwoWireCom::send(int bc)
{
switch(bc){
case 1: //babycon_level = 1, must set GPIO OUTPUT_PIN_MSB = 0, OUTPUT_PIN_LSB = 1.
*pFIO_FLAG_S = OUTPUT_PIN_LSB; // Pointer to BANKS' GPIO write to set register (Set high)
*pFIO_FLAG_C = OUTPUT_PIN_MSB; //Pointer to BANKS' GPIO Write to Clear register (set low)
break;

case 2:
//babycon_level = 2, must set GPIO OUTPUT_PIN_MSB = 1, OUTPUT_PIN_LSB = 0.
*pFIO_FLAG_S = OUTPUT_PIN_MSB;
*pFIO_FLAG_C = OUTPUT_PIN_LSB;
break;

case 3:
//babycon_level = 3, must set GPIO OUTPUT_PIN_MSB = 1, OUTPUT_PIN_LSB = 1.
*pFIO_FLAG_S = OUTPUT_PIN_LSB;
*pFIO_FLAG_S = OUTPUT_PIN_MSB;
break;

default:
//babycon_level != 1-3, must set GPIO OUTPUT_PIN_MSB = 0, OUTPUT_PIN_LSB = 0 for error
*pFIO_FLAG_C = OUTPUT_PIN_MSB;
*pFIO_FLAG_C = OUTPUT_PIN_LSB;
break;
}
```

5.5 Modultest

5.5.1 Modultest Prefilter

5.5.2 Modultest RecBuf

Formål

Formålet er at teste om klassen RecBuf er i stand til at håndtere dens to buffere og at det rigtige data forekommer.

Opstilling til test af RecBuf test

- Blackfin 533 er sluttet til 230VAC
- Blackfin 533 er tilsluttet PC via COM-port
- CCES er kørende på PC

Testprocedure

1. Projektet med RecBuf testen eksekveres
2. Kommandoprompten tjekkes for udskrift

Forventet resultat

Det forventede resultat af første udskrift kan ses i Tabel 5.2:

Tabel 5.2. Forventede værdier

Buffer	index 0	index 1	index 2
mem0_	0	1	2
mem1_	3	4	5

Det forventede resultat af anden udskrift kan ses i Tabel 5.3:

Tabel 5.3. Forventede værdier

Buffer	index 0	index 1	index 2
mem0_	10	11	12
mem1_	13	14	15

Resultat

På Figur 5.32 ses den først udskrift, det ses ligeledes at det stemmer overens med de forventede værdier.

```
Content of mem0_ is
Index 0 is: 0
Index 1 is: 1
Index 2 is: 2

Content of mem1_ is
Index 0 is: 3
Index 1 is: 4
Index 2 is: 5
```

Figur 5.32. Første udskrift i kommandoprompten

På Figur 5.33 ses den anden udskrift, det ses ligeledes at det stemmer overens med de forventede værdier.

```
Content of mem0_ is  
Index 0 is: 10  
Index 1 is: 11  
Index 2 is: 12  
Content of mem1_ is  
Index 0 is: 13  
Index 1 is: 14  
Index 2 is: 15
```

Figur 5.33. Første udskrift i kommandoprompten

Testen er godkendt

5.5.3 Modultest Analyzer

5.5.4 Modultest Categorizer

5.5.5 Modultest Statistician

5.5.6 Modultest TwoWireCom

Formål

Formålet er at teste om klassen TwoWireCom sætter GPIO's PF4 og PF6 korrekt jf. Tabel 3.2

Opstilling til test af TwoWireCom test

- Blackfin 533 er sluttet til 230VAC
- Blackfin 533 er tilsattet PC via COM-port
- CCES er kørende på PC
- Analog Discovery er tilsluttet PC via COM-port
- Waveform er kørende på PC
- Analog Discovery oscilloskop 1 er sat til Blackfin533's GPIO-pin PF4
- Analog Discovery oscilloskop 2 er sat til Blackfin533's GPIO-pin PF6

Testprocedure

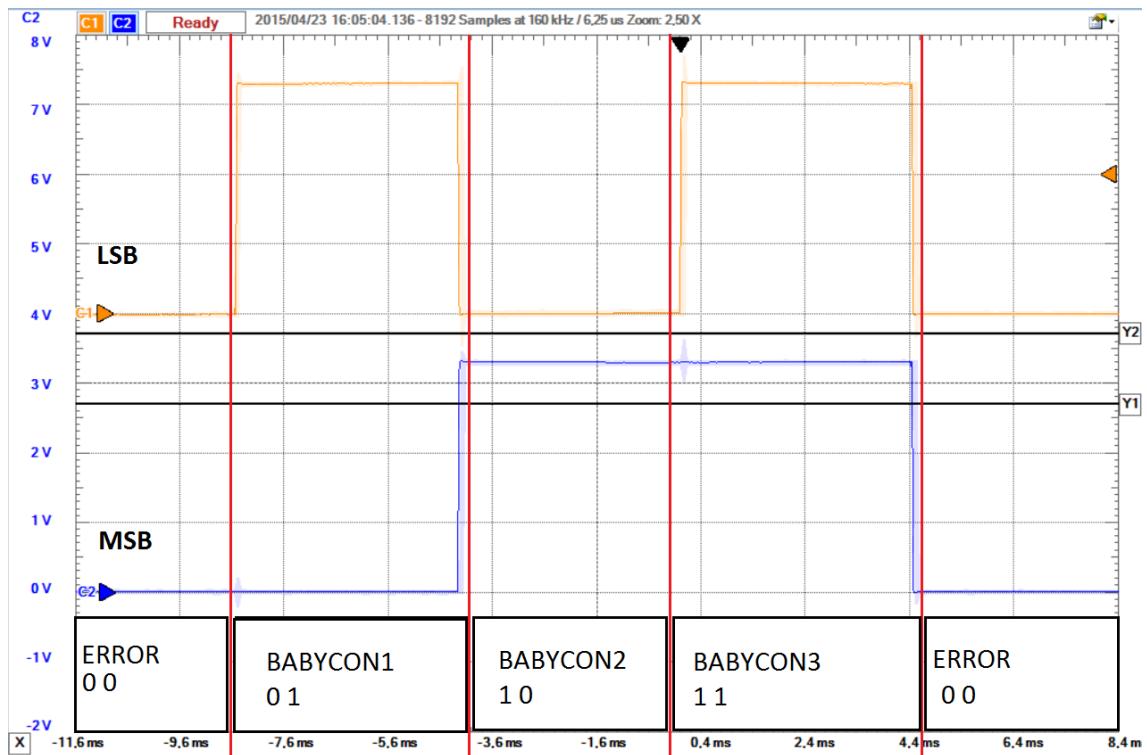
1. Projektet med TwoWireCom testen eksekveres
2. Osciloskoperne optages og billedet gemmes

Forventet resultat

Det forventes at alle fire kombinationer for de to GPIO-pins er registreret.

Resultat

Som det ses af Figur 5.34 forekommer alle fire kombinationer.



Figur 5.34. Oscilloskop billede af test

Testen er godkendt

Vuggesystem 6

6.1 Systemarkitektur

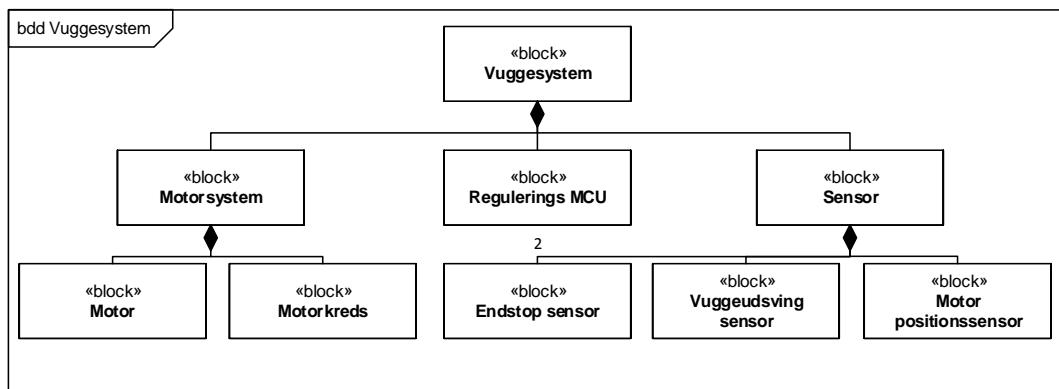
I dette afsnit beskrives systemarkitekturen for vuggesystemet.

Overordnet virkemåde

Vuggesystemet fungerer overordnet som følger:

- Vuggesystemets Regulerings MCU står for at vugge barnevognens kurv med en bestemt vuggefrekvens og et bestemt vinkeludsving. Dette sker på baggrund af værdier for disse modtaget fra Controller
- Regulerings MCU'en regulerer vuggesystemet således at vugningen altid foregår ud fra en vandret akse samt at de modtagne værdier for vuggefrekvensen og vinkeludsvingen overholdes
- Kommunikation mellem Regulerings MCU og Controlleren foregår via I2C protokollen

6.1.1 Hardware arkitektur



Figur 6.1. BDD for Vuggesystem

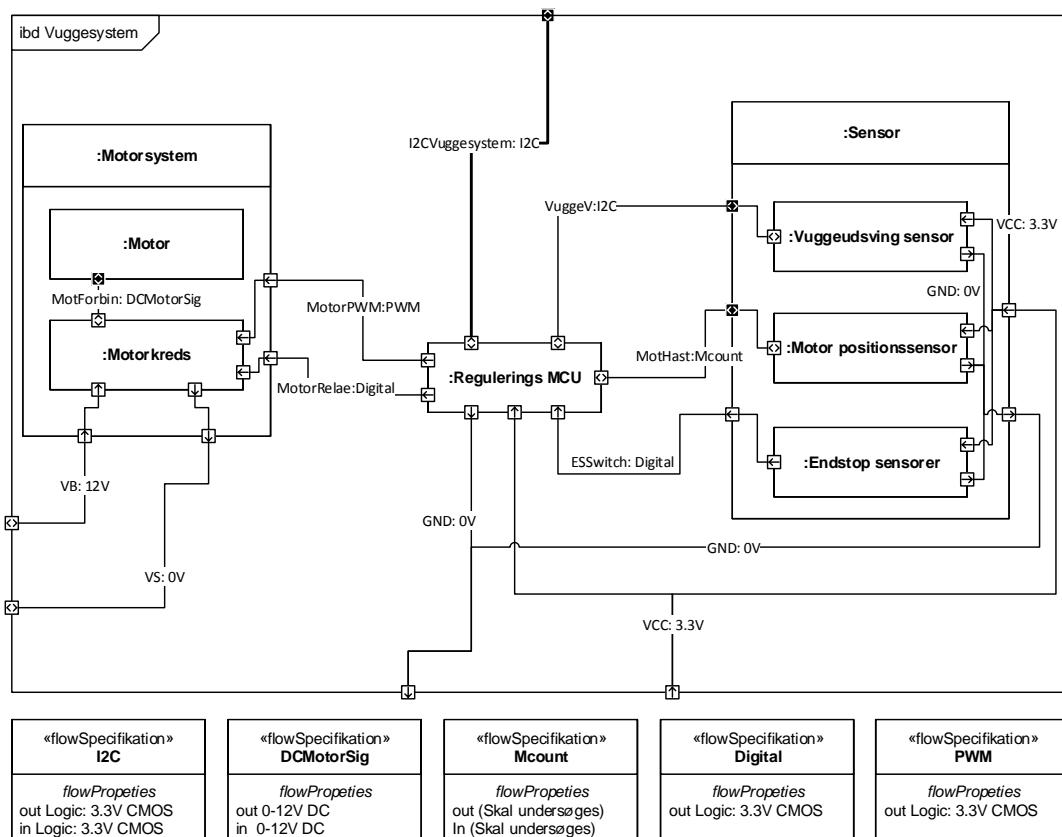
Vuggesystemet består af tre dele

Motorsystem block: Består af en motor til at drive vuggebevægelsen samt en motorkreds til strømstyringen af motoren. Motorkredsen sørger for at motorens ankerspænding styres vha dutycycle på et PWM-signal og retningen på motoren styres af et logisk signal.

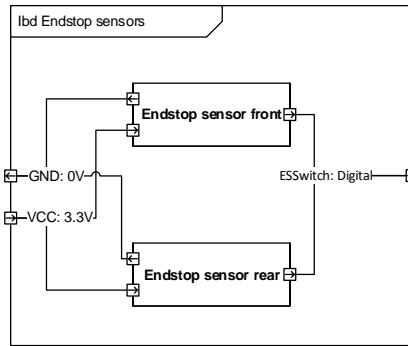
Regulerings MCU block: Styringsenheden for vuggesystemet. Denne sørger for reguleringen af vuggesystemet samt kommunikationen til og fra controlleren.

Sensor block: Består af fire sensorer; to Endstop sensorer som måler om barnevognens kurv har nået den mekaniske vuggegrænse, Vuggeudsving sensor måler kurvens absolutte vinkel i forhold til tyngdefelt og Motor positionssensoren måler motorens position.

Følgende beskriver vuggesystemets kobling og grænseflade.

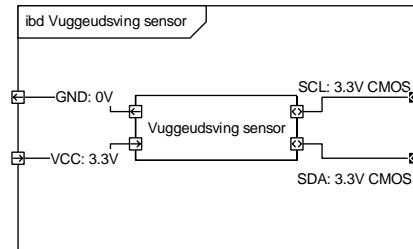


Figur 6.2. IBD for Vuggesystem



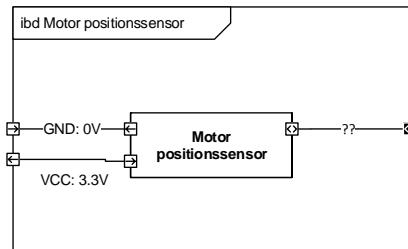
Figur 6.3. IBD for Endstop sensorer

Endstop sensorer består af to ens sensorer, hhv. **Endstop sensor front**, placeret til at detektere hvis barnevognskurven når den mekaniske vuggegrænse ved fremad vuggeretning, og **Endstop sensor rear**, placeret til at detekterer hvis barnevognskurven når den mekaniske vuggegrænse ved bagud vuggeretning. Sensorerne giver det samme signal uanset om detekteringen sker i front eller bag.



Figur 6.4. IBD for Vuggeudsving sensor

Vuggeudsving sensor er placeret så den mäter barnevognskurvvens plan i forhold til jordens tyngdefelt.



Figur 6.5. IBD for Motor positionssensor

Motor positionssensor giver et frekvenssignal relativ til motorens nuværende hastighed.

6.1.2 Grænsefladebeskrivelse

Herunder findes en beskrivelse af de interne forbindelser.

Signalbeskrivelse

Tabel 6.1: Signalbeskrivelse

Signal	Type	Kommentar
VB	12V DC	Dette signal kommer direkte fra systemets batteri
VS	0V DC	Dette er batteriforsyningens retur
VCC	3.3V DC	Dette er forsyningen til PSoC og andet logik
GND	0V stel	Stelforbindelse til PSoC og andet logik
MotorPWM	PWM	PWM som styrer motor spændingen ved VB samt retningen for motoren. Fra 0-50%duty-cycle genererer en negativ motorspænding, fra 50-100 %duty-cycle genererer en positiv motorspænding. PWM signalet 0-3.3V CMOS med f = [20kHz]
MotorRelae	Digital	Kontakt som holder motoren tændt, HIGH = Tændt LOW = slukket
VuggeV	I2C	Angiver vinklen af vuggen relativt til tyngdefeltet
MotHast	Mcount	Angiver hastigheden af motoren(Undersøges færdig)
ESSwitch	Digital	Endstop status, ES-for OR ES-bag, HIGH = ikke ramt LOW = ramt
I2CVuggesystem	I2C	Styresignal til vuggesystemet
MotForbind	DCMotorSig	Motor tilkobling

6.1.3 Software arkitektur

I følgende afsnit beskrives softwarearkitekturen for Vuggesystem delen af Baby Watch. Softwarearkitekturen er udarbejdet på baggrund af projektformuleringen og kravspecifikationen.

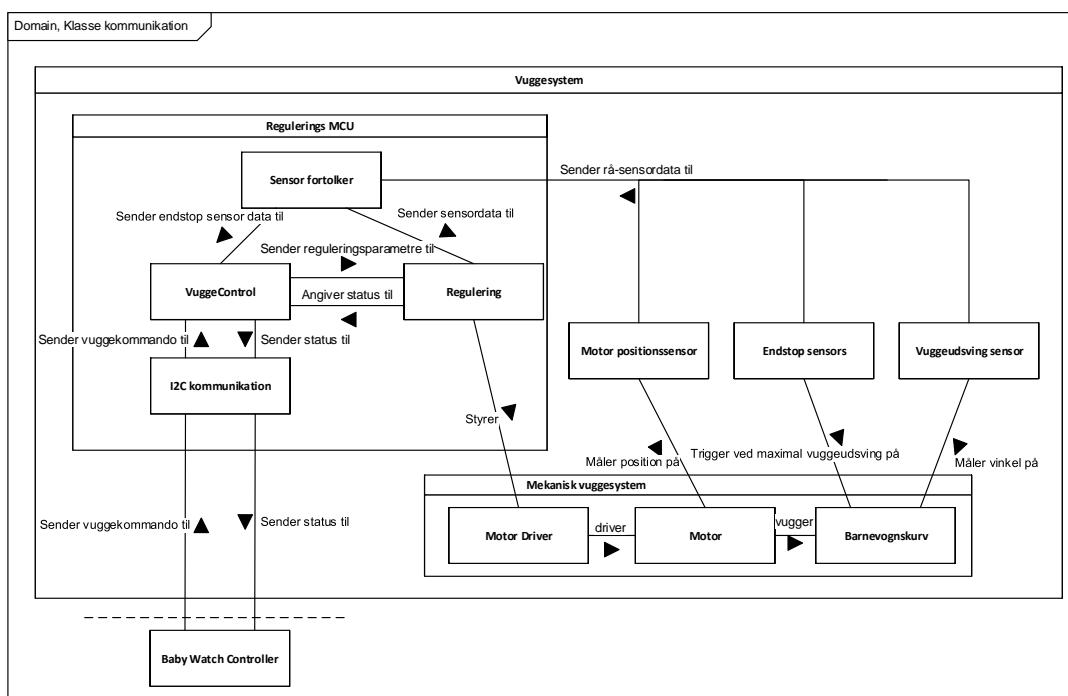
Softwarearkitekturen for Vuggesystemet består af:

- Identifikation af problemer, klasser og metoder med udgangspunkt i en domænemodel for system og dets softwaremoduler
- Oprettelse af skelet for videre implementering af Vuggesystemets SW vha. et sekvensdiagram til klasseidentifikation og programflow

Applikationsmodellen indeholder funktionaliteter for op til flere Use Cases, men er kun afgrænset til Vuggesystemet.

Applikationsmodel

Domænemodel



Figur 6.6. Domain kommunikations model for Vuggesystem

Figur 6.6 viser en domænemodel for kommunikationen mellem systemets interne SW-moduler. Blokken Baby Watch Controller tilhører ikke Vuggesystemet, men er sat på for at identificere udadgående grænseflader derfor er denne markeret bag en stiplet linje.

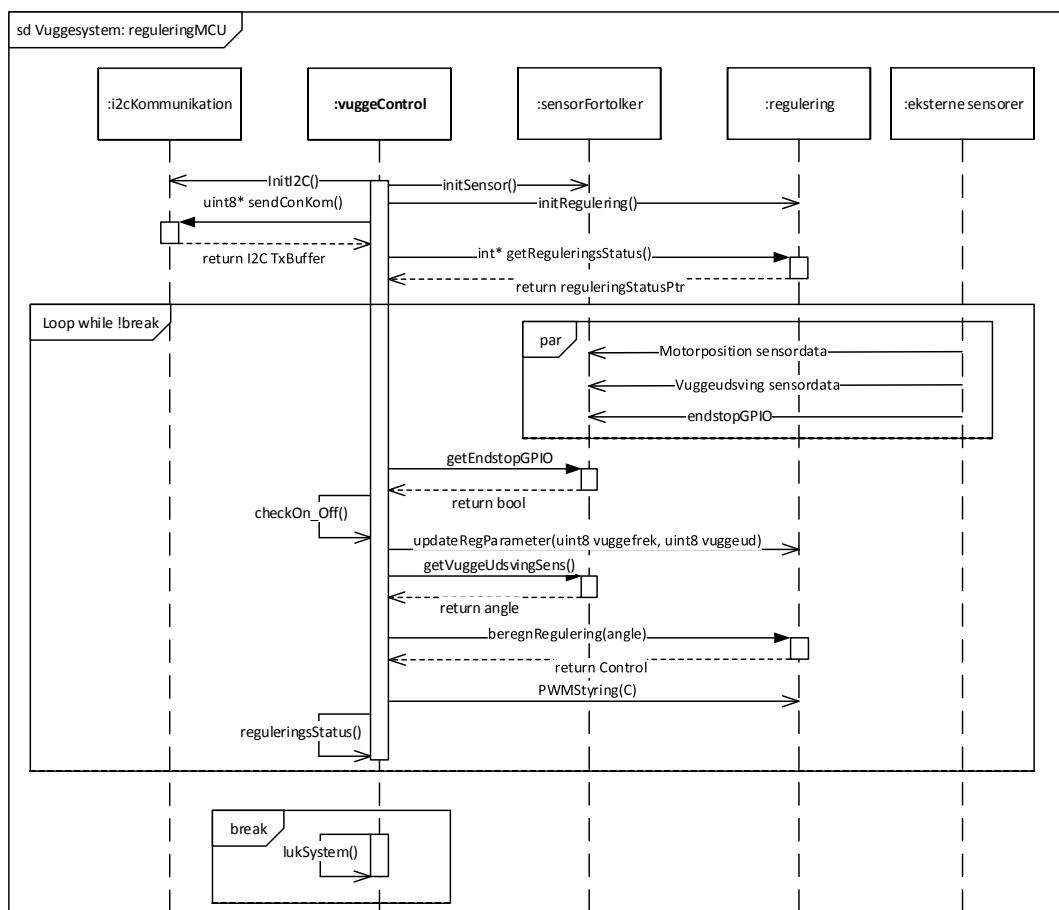
Klasseidentifikation

På baggrund af domænemodellen for softwaremoduler og følgende beskrivelse, udarbejdes et sekvensdiagram der identifierer programflow og klasser for **Regulerings MCU** blokken.

Beskrivelse af system:

Vugesystemet sørger for at vugge barnevognskurven med en bestemt vuggefrekvens og vuggeudsning angivet af Baby Watch Controller. Denne vugning reguleres ud fra sensorinput der angiver; den absolute vinkel for vuggeudsvinget i forhold til tyngdefeltet, motorens position og om den mekaniske grænse for vuggeudsvinget er nået. Vugesystemet skal også give besked til den overordnede Baby Watch Controller om eventuelle fejl i denne regulering.

Sekvensdiagram



Figur 6.7. Sekvensdiagram for Regulerings MCU

Blokkene i **Regulerings MCU** fra domænemodellen ændres til følgende klassenavne:

- Regulerings MCU - klassenavn: **vuggeControl** er Vuggesystems control-klasse.
- I2C kommunikation - klassenavn: **i2cKommunikation** står for I2C interfacet ud til Baby Watchs Contoller.
- Motorregulering - klassenavn: **regulering** står for beregningerne til reguleringen af motoren der trækker vugningen af barnevognenskurv
- Sensor Fortolker - klassenavn: **sensorFortolker** står for at behandle sensorinput fra Motor positionssensor, Endstop sensors og Vuggeudsving sensor

Motorposition sensordata, Vuggeudsving sensordata og endstopGPIO skal forståes som interruptbaserede funktioner. Disse opdaterer automatisk sensordataene fra de tre forskellige sensorer.

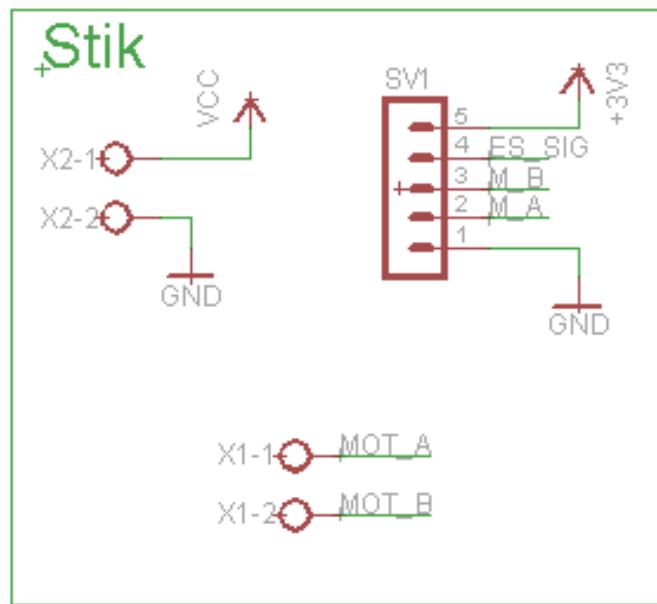
6.2 Design

Design afsnittet for vuggesystemet beskriver først hardware designet efterfulgt af software designet.

6.2.1 Hardware design

Motorstyringskreds

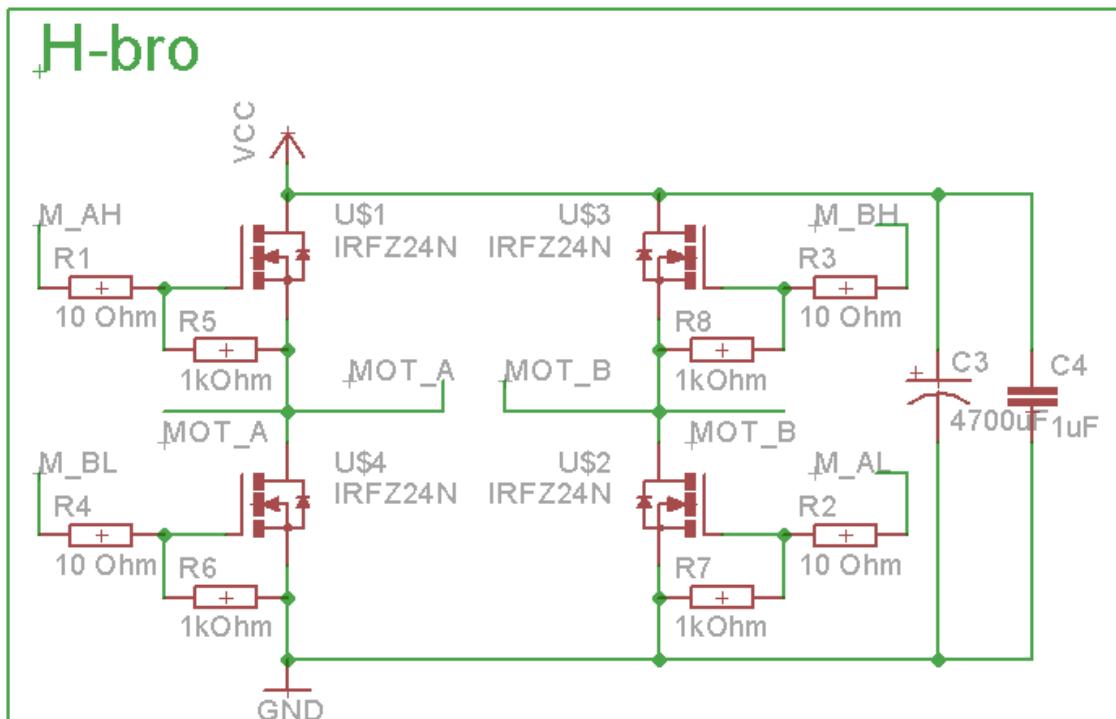
I denne sektion beskrives det kredsløb der benyttes til at styre ankerspændingen for vuggesystemets motor. Kredsløbet modtager et PWM signal fra controlleren som styrer ankerspændingen. Kredsen er forsynet med bateriets 12V (VCC) og 3.3 volt fra PSoC. Kredsen styres med et PWM signal, og det inverterede signal, kaldt M_A og M_B. Den sidste forbindelse, ES_SIG, er outputtet fra endstop kredsløbet.



Figur 6.8. Forbindelser til motorkreds

H-bro

Motoren styres med en H-bro implementeret med 4 N-ch power MOSFETs, som vist herunder:



Figur 6.9. N-ch enh. Power MOSFET H-bro

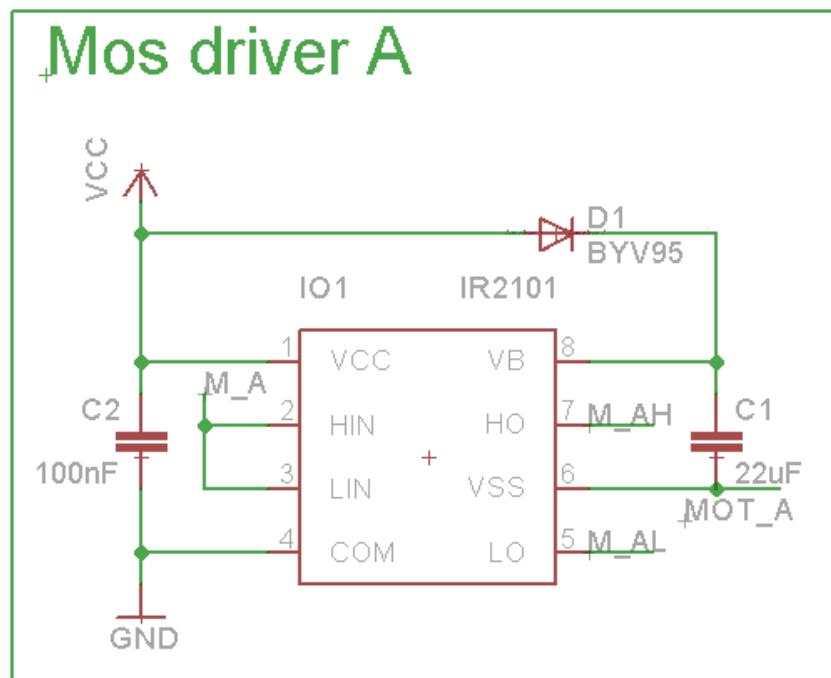
De fire N-channel er koblet med en lille modstand foran gaten, og en pull-down modstand fra gaten til source. Den lille modstand sikrer at der ikke opstår or meget ring som resultat af seriekobling af ledningens induktans og gatens kapacitet, og pull-down modstanden sikrer at gaten bliver trukket lav hvis MOS driveren skulle fejle, således at transistoren afbryder.

Signalerne M_AH, M_AL, M_BH, og M_BL udgør således H-broens styresignaler til hhv. den høje og lave gate i A og B siden af broen. Terminalerne MOT_A og MOT_B går til motorens terminaler. Hvis A siden af broen er tændt påtrykkes en spænding med MOT_A som den positive terminal, og omvendt for B siden.

Kondensatorerne C3 og C4 er placeret så tæt som muligt på H-broens positive og negative forsyning, og hjælper til at afkoble motoren, således at der ikke introduceres for store forstyrrelser på forsyningen. Dimensionering af disse er beskrevet nærmere i EMC afsnittet om vuggesystemet.

MOSFET driver

H-broens MOSFETs bliver drevet ved hjælp af 2 dobbelt siddede MOSFET drivere, som er koblet som vist herunder:



Figur 6.10. Dual MOSFET driver, A siden

Driveren kan drive en MOSFET der sidder til ground, og en MOSFET der sidder fra VCC, og de to drivere er koblet ens til hhv. A siden og B siden. signalet til den lave MOSFET, og det interne logik drives fra VCC, mens signalet til den høje MOSFET drives fra en bootstrap kreds bestående af C1 og D1. Kredsen fungerer ved at C1 oplades til en diode spænding under VCC når den lave MOSFET fra den anden side af broen er tændt. Når den lave MOSFET slukkes vil potentialet ved VB således ligge ca. 11,3 V over potentialet ved source benet af den høje MOSFET, og der er således den nødvendige spænding for at trække gaten i mætning.

Driverens input er et digitalt input, med en fast overgang fra lav til høj mellem 0.8 og 3 V, og kan således styres med 3.3V logik uafhængigt af hvilken forsyningsspænding driveren er koblet med.

Kondensatoren C2 hjælper til at afkoble driveren fra forsyningen.

Styresignal

Broen styres, som nævn ovenfor, fra et PWM signal. Dette gøres ved at A sidens MOSFETs modtager PWM signalet direkte, mens B siden modtager det inverterede signal. Benyttes halvtreds procent duty cycle vil de to sidder af broen være tændt i lige lang tid, og middelspændingen over motoren vil således blive 0V. Hæves duty cyclen til over 50 procent opnås en gradvis større middelspænding med MOT_A som den positive terminal, og omvændt ved duty cycles under halvtreds procent.

Da MOSFET transistorne tager en hvis tid om at slukke, skal der introduceres en dødtid i styresignalerne, således at signalerne alle går kortvarigt til 0 før de skifter. Dette sikrer at

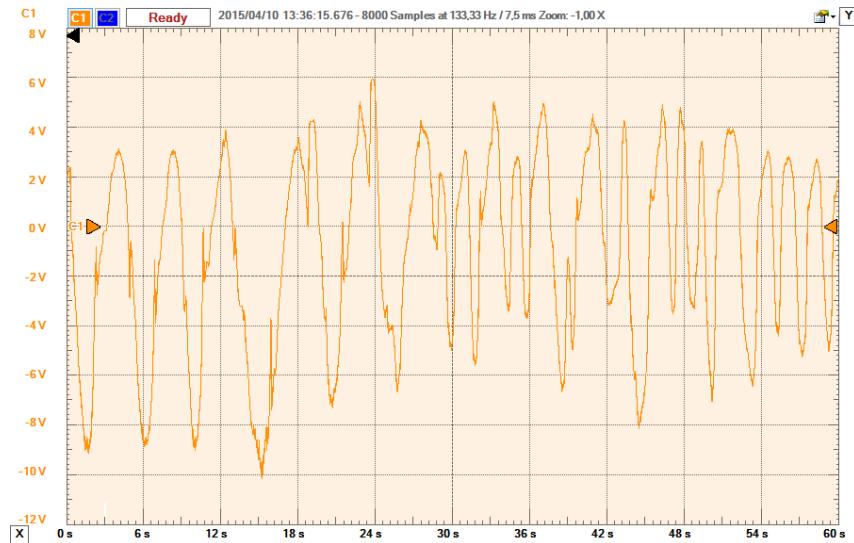
to transistorer fra hver deres side af broen ikke er tændt samtidig, hvilket ville resultere i en kortslutning fra VCC til GND gennem de to transistorer.

Dødbåndet i pwm signalet genereres i PSoC'ens PWM komponent, og er sat til det mindst mulige tid på 1% af periodetiden, eller 500ns. Da MOSFET driveren kortslutter gate signalet til source benet når transistoren skal afbrydes, kan slukketiden estimeres ved at anskue transistoren og dens formodstand, som et RC led der skal aflades for at slukke transistoren. Gate kapaciteten oplyses i databladet som 370pF, og formodstanden er på 10 Ohm, og man får således en tidskonstant på $3.7 \cdot 10^{-9}$. Transistoren anskues som afbrudt efter $5 \cdot \tau = 18.5$ ns, hvilket er vel under det genererede deadband.

strømforbrug

Det dominerende strømforbrug i systemet stammer fra vuggesystemets motor. Denne strøm trækkes direkte fra systemets batteri, og har derfor ikke indflydelse på dimensioneringen af systemets reguleringskreds. Tilgengæld er den vigtig for at kunne vurdere hvor stort et batteri der skal til for at drive systemet en given tid.

Et tidligt skøn over motorens strømforbrug er gjort ved at sætte et manuelt styret PWM signal på motor kredsen, og så vugge barnevognen med vægt i, så godt som muligt ved manuel regulering. Herunder ses strømmålingen for 60 sekunders test. Målingen er gjort med en /10 probe på spændingsfaldet over en 0.1 Ohm's modstand, og den målte spænding svarer således direkte til strømmen.



Figur 6.11. Test af strømforbrug for motor

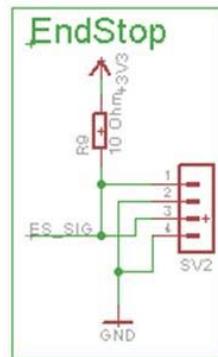
Ved måling i Digilent Waveforms (se figur 6.12 svarer ovenstående til en rms belastning ved batteriet på 3.9 Ampere, med peaks på op til 10 Ampere. Det er værd her at bemærke at disse peaks er med de langsomme fluktuationer der kommer som resultat af den varierende belastning ved vugning. Dette er ikke at forvæksle med den højfrekvente fluktuation der opstår som resultat af PWM-styringen, da denne, så vidt muligt, ikke eksponeres til batteriet.

Add	X	↑	↓
Ch	Name	Value	
C1	Maximum	5.948 V	
C1	Minimum	-10.126 V	
C1	DC RMS	3.922 V	

Figur 6.12. Databehandling fra strømmåling

Endstopsensor

Endstopsensor kredsløbet er designet som et pull-up kredsløb hvor to mekaniske switches kan parallelopkobles. Designet er vist på figuren nedenfor.



Figur 6.13. Endstop sensor schematic

Kredsløbet opererer med 3.3V og ved aktivering af en af endstop sensorene trækkes signalet ned til 0V (GND).

Vuggeudsving sensor

Vuggekurvens vinkel i forhold til jordens tyngdefelt måles vha af en mpu6050 integreret accelerometor og gyroskop, på et breakout board som vist på figur 6.14. Denne IC er valgt fordi den kan levele accelerometer- og gyroskopdata igennem et I2C interface, med en separat interrupt linje, som indikerer når der er nye data parat. Ved at bruge både accelerometer og gyroskop opnås et bedre estimat af sensorens orientering idet gyroskopet leverer pålidelige data omkring ændringer i orientering over en kort tidsskala hvilket

kan afhjælpe støjen fra vibrationer mm. som viser sig som støj i accelerometerdataen. Accellerometeret, som benytter en absolut reference (tyngdefeltet), kan samtidig afhjælpe den drift der sker i gyroskopet, idet fejl akkumuleres i gyroskopet. For en detaljeret beskrivelse af hvordan de to datakilder splejses se afsnit 6.3.2 på side 138.



Figur 6.14. MPU6050 breakout board

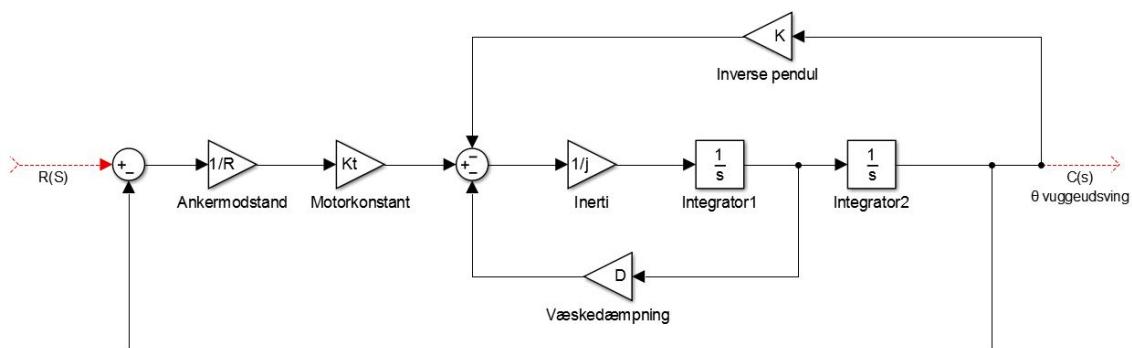
Motor positions sensor

Motor positions sensoren er ikke med i denne iteration af projektet.

Mekanisk vuggesystem

Det mekaniske vuggesystem er designet ud fra følgende skitse:

Nedenståede blokdiagram er en skitsering af den valgte teoretiske model for det mekaniske vuggesystem



Figur 6.15. Skitse af teoretiske model for mekanisk vuggesystem

Beskrivelses af modellen:

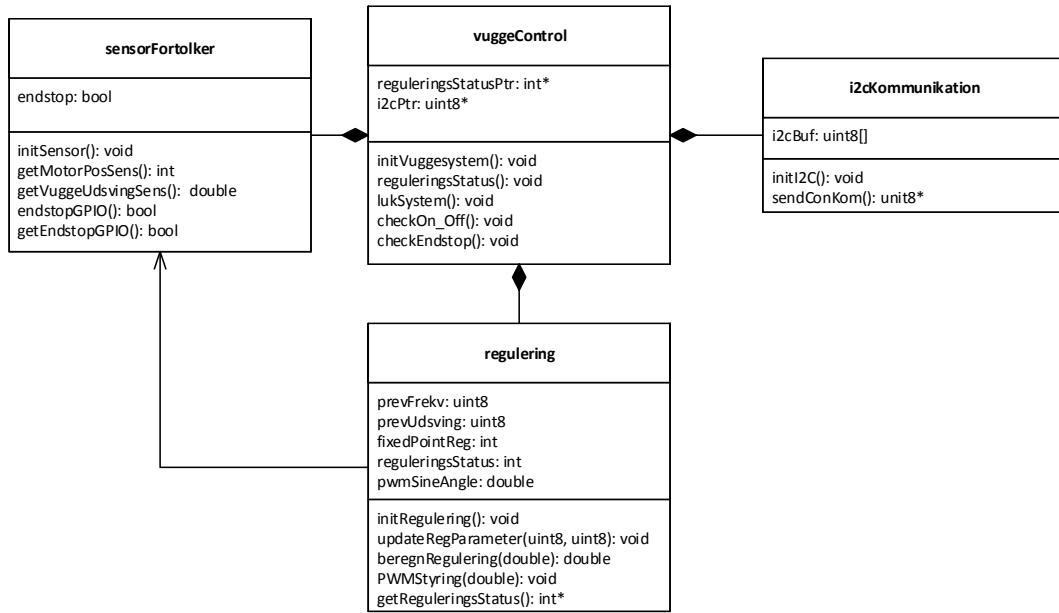
- **R(s)** skal ses som ankerspændingen der tilføres systemet
- **Ankermodstand** og **Motorkonstant** udgør DC-motoren
- **Inerti** er vuggesystemets mekaniske rotationsmoment
- **Væskedæmpning** er friktionen der stammer fra aksen der vugges over i systemet

- Det **Inverse pendul** er et ulineært element der stammer fra at barnevognskurvens masse ligger højere end aksen der vugges over. Dette element er ustabilt af natur og kræver påvirkning for at holdes stabilt. Dette er især gældende for større udfaldsvinkler, men da niveauet der skal vugges over maximalt udgør ± 10 grader i forhold til tyngdefeltet kan dette element gøres tilnærmelsesvist lineært
- Fra højre mod venstre skal punktet efter **Iterator2** ses som hastigheden af barnevognens rotation og punktet efter **Integrator1** skal ses som accelerationen af barnevognens rotation
- **C(s)** er vuggeudsvinget/vippevinklen for barnevognskurven i forhold til tyngdefeltet

6.2.2 Software design

Softwaredesignet for Vuggesystemet udarbejdes vha. et klassediagram med tilhørende funktionsbeskrivelser lavet på baggrund af applikationsmodellen fra Vuggesystemets software systemarkitektur jf. afsnit 6.1.3

Klassediagram



Figur 6.16. UML klassediagram for Regulerings MCU

Funktionsbeskrivelser

vuggeControl

Ansvar: Main klasse, styrer de andre klasser i reguleringsMCU'en.

Funktion	<code>void initVuggesystem()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Kalder funktionerne <code>void initI2C()</code> , <code>void initSensor()</code> og <code>void initRegulering()</code> samt initiere i2cPtr vha. <code>sendConKom()</code> og <code>reguleringsStatusPtr</code> vha. <code>getReguleringsStatus()</code>

Funktion	<code>void reguleringsStatus()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Tjekker reguleringsklassens driftstatus og i tilfælde af fejl videresendes denne fejl til i2cKommunikationsklassen, med fejlindikator værdierne 0b1000000 for error og 0b11000000 for stall. Herefter kaldes <code>lukSystem()</code>

Funktion	<code>void lukSystem()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Lukker strømforsyningen til det mekaniske vuggesystem og videresender fejlmeddelse; 0b1001000 til i2cKommunikationsklassen. Kaldes i tilfælde hvor det mekaniske vuggesystem skal lukkes

Funktion	<code>void checkEndstop()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Tjekker om endstop værdi i sensorFortolkerklassen er TRUE ved at kalde funktionen <code>getEndstopGPIO()</code> . Ved TRUE kalder funktionen <code>lukSystem()</code> og videresender fejlmeddelse; 0b10100000 til i2cKommunikationsklassen.

i2cKommunikation

Ansvar: Står for I2C kommunikationen til mellem vuggesystemet og Baby Watchs Controller.

Funktion	<code>void initI2C()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Initiere I2C-kommunikationen til operationel funktion

Funktion	<code>uint8 sendConKom()</code>
Parametre	Ingen
Returværdi	Returner pointer til i2cKommunikationsklassens I2CBuf
Beskrivelse	Returner pointer fra i2cKommunikationsklassens I2CBuf til vuggeControllklassen

regulering

Ansvar: Står for regulering og styring af det mekaniske vuggesystem ud fra beregner baseret på sensorværdier.

Funktion	<code>void initRegulering()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Initerer reguleringen til operationel status samt starter reguleringen

Funktion	<code>void updateRegParameter(uint8 vuggefrek, uint8 vuggeud)</code>
Parametre	uint8 vuggefrekvens indeholder en binærværdi fra 0-255 der udfra en skala svare til en vuggefrekvens jf. Kommunikationsprotokol 3.3.2, uint8 vuggeudsning indeholder en binærværdi fra 0-255 der ud fra en skala svare til et vuggeudsning jf. Kommunikationsprotokol 3.3.2
Returværdi	Ingen
Beskrivelse	Gemmer og validere ny vuggefrekvens og vuggeudsning til at regulerer udfra.

Funktion	<code>double beregnRegulering(double currentAngle)</code>
Parametre	double currentAngle
Returværdi	double
Beskrivelse	Beregner reguleringen for vuggesystem via angivet værdier vuggefrekvens, vuggeudsning og parameter currentAngle. Returnere en double med en værdi ud fra et 16.16 fixed-point nummer som svare til et output fra en overføringsfunktion "C(t)"

Funktion	<code>void PWMStyring(double C)</code>
Parametre	double C
Returværdi	Ingen
Beskrivelse	Styrer uadgående PWM udfra fixedPointReg værdi C (En output værdi fra overføringsfunktionen for vuggesystemet "C(t)")

Funktion	<code>int* getReguleringsStatus()</code>
Parametre	Ingen
Returværdi	Int*
Beskrivelse	Returnere pointer fra int reguleringsStatus

sensorFortolker

Ansvar: Står for grænsefladen ud til Vuggeudsning sensor, Endstop sensors og Motor positionssensoren. Udfra Vuggeudsning sensorens parametre beregnes desuden den absolutte vinkel af barnevognskurven i forhold til tyngdefeltet.

Funktion	<code>void initSensor()</code>
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	Initiere alt sensor-kommunikation til operationel funktion

Funktion	<code>int getMotorPosSens()</code>
Parametre	Ingen
Returværdi	int
Beskrivelse	Returner sensor data fra Motorpositions sensoren angivet i værdi fra -256 til 255

Funktion	double getVuggeUdsvingSens()
Parametre	Ingen
Returværdi	double
Beskrivelse	Returnere en fix16.16 double med den nyeste vinkel beregnet ud fra gyroskop- og accellerometersensor måling

Funktion	void endstopGPIO()
Parametre	Ingen
Returværdi	Ingen
Beskrivelse	ISR funktion som sætter flaget int endstop til 1

Funktion	bool getEndstopGPIO()
Parametre	Ingen
Returværdi	int*
Beskrivelse	Returner bool fra int endstop til klassen vuggeControl

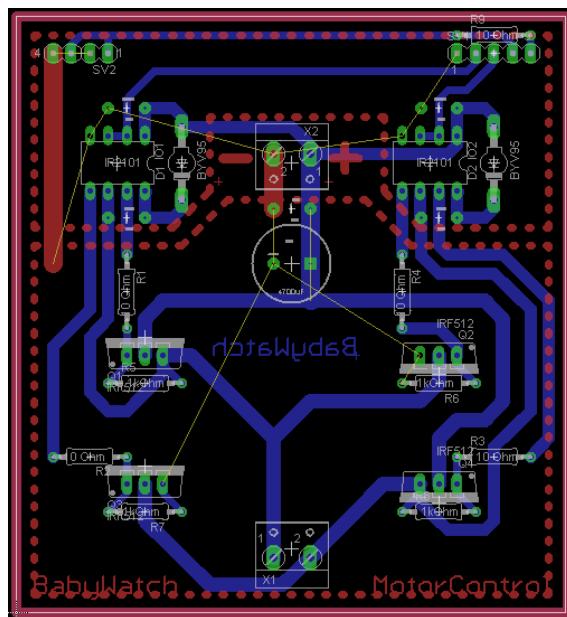
6.3 Implementering

Implementeringsafsnittet for vuggesystemet beskriver først hardware implementeringen og efterfølgende software implementeringen.

6.3.1 Hardware implementering

Motorstyringskreds og endstop kreds

Motorstyringskredsløbet er det kredsløb i hele systemet som trækker mest effekt, og har derfor været underlagt en række særlige betragtninger for at sikre at kredsløbet ikke forstyrer resten af systemet. Den fysiske kreds er implementeret på et print sammen med endstop kredsløbet. Layoutet er som følger:



Figur 6.17. Print layoutet for motorstyringskredsen og endstop kontakterne

De to kredsløb er placeret på samme print udelukkende for at spare plads. Dette er vurderet problemfrit udfra den betragting at endstop kredsen er stort set ufølsom overfor støj.

En mere detaljeret beskrivelse af designbetragtningerne for endstop kredsløbet kan findes i EMC rapporten. De to mest centrale ting er afkoblingskondensatorerne (markeret med gult i figur 6.17) og brugen af groundplan (måret med stiplet rødt omrids).

Kredsløbets eksterne forbindelser er lavet med Harwin pins til signalforbindelser (sv1 og sv2 i diagrammet), og skrueterminaler til effektforbindelserne (X1 og X2)

Endstopsensor

Endstop sensorene er blevet realiseret ved hjælp af to end stop switches. En endstop switch fungere som en alm. switch der ved aktivering skaber forbindelse mellem to punkter. Figuren under viser en tilsvarende udgave af en end stop switch som dem brugt i projektet.



Figur 6.18. Generisk end stop switch

Disse end stop switches er placeret således at når barnevognskurven på Baby Watch systemet går ud i en yder vippeposition så aktiveres en end stop switch.



Figur 6.19. Monteret endstop sensor

Endstop sensorene er herefter forbundet til endstop kredsløbet som anvist i Vuggesystem Design afsnit 6.2.1 på side 122.

Implementeringen af endstop sensor kredsløbet kan ses i det foregående Motorstyringskreds afsnit på figur 6.17 på forrige side

Vuggeudsving sensor

Vuggeudsvingssensoren er implementeret som beskrevet i Vuggesystem Design afsnittet 6.2.1 på side 122 ved hjælp af en MPU6050 chip med indbygget accelerometer, gyroskop og I2C bus interface. Den er placeret på undersiden af barnevognskurven ud for aksen der vippes henover. Nedenfor kan ses et billede af implementeringen.



Figur 6.20. Monteret Vuggeudsving sensor

Chippen er delvist skjult bag en beskyttelse. Den gule streg på billedet markerer aksen barnevognskurven og derved chippen vipper henover. De parsnoede ledning er for at beskytte I2C kommunikationen mod støj.

Motor positionssensor

Motor positionssensoren bliver ikke implementeret i denne iteration af projektet.

Mekanisk vuggesystem

Det mekaniske vuggesystem er implementeret ud fra skitsen i Vuggesystem Design Mekanisk Vuggesystem afsnit 6.2.1 på side 123.

Motor:

Motoren er placeret under barnevognskurvens tyngde center med en påsat gearing på 1:16 samt en kæde der giver en yderligere gearing på 1:8. Dette betyder at ved en hel vugning af barnevognskurven (fra yder position til yder position) roterer motoren 8 gange.



Figur 6.21. Motor til det mekaniske vuggesystem med vist gearing



Figur 6.22. Tandhjul påsat motoren til det mekaniske vuggesystem

Vuggeudsving: Som det ses på de tre nedenstående figurer så opfylder vuggesystemet kravspecifikationens ikke-funktionelle krav afsnit 2.5 om at barnevognen skal kunne vugge med mindst +/- 10 grader. Det implementerede vuggesystem kan ideelt vippe ud til +/- 24,5 grader. Dette gør at Baby Watch systemet vil være i stand til at stå på en skrånende flade på op til 14,5 grader. Dette bliver ikke afprøvet og tested i denne iteration af projektet.



Figur 6.23. Baby Watch barnevogn i vandret position



Figur 6.24. Baby Watch barnevogn i højre yderposition



Figur 6.25. Baby Watch barnevogn i venstre yderposition

Stabilisering af barnevogn: Barnevogne har som standard fjedrende led. Disse led er blevet stabiliseret med spændebånd for at gøre barnevognens overføringsfunktion mere lineær.



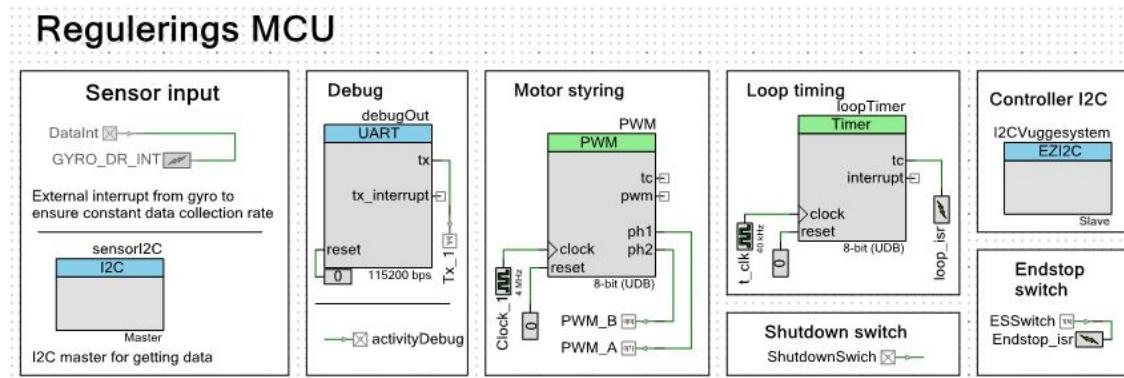
Figur 6.26. Billede af stabiliseret barnevognsled

6.3.2 Software implementering

Regulerings MCU

Regulerings MCU'en er implementeret på et Cypress CY8CKIT-042 PSoC4 Pioneer Kit [Cypress, 2014]. Denne platform består af en ARM Cortex-M0 CPU forbundet med en simpel FPGA der alt sammen kan programmes og styres med Cypress' egen udviklings suite, PSoC Creator 3 [Cypress, 2015]. Dette giver en fleksibel micro controller platform med mulighed for at bruge flere specialiseret embedded hardware FPGA blokke der kan styrer diverse interfaces, GPIO'er, Timer, interrupts mm.

På nedenstående Top Design for fra PSoC Creator kan hardware setup'et for Regulerings MCU'en ses. Hver indrammende boks markerer en specialiseret funktionalitet håndteret af en eller flere hardware blokke.



Figur 6.27. Top Design for Vuggesystems Regulerings MCU

Følgende vil gennemgå funktionaliteten for hver boks fra start fra venstre mod højre:

Sensor input:

Denne boks består af en I2C master blok der sørger for et I2C-interface mellem Regulerings MCU'en og Vuggeudsving sensoren. Dette bruges i aflæsningen af sensoren. I dette setup er overføringshastigheden sat til Normal-Mode dvs. med en bitrate på 100 kHz.

Debug:

Debug boksen sørger for et UART interface mellem PSoC4 micro controlleren og en computerterminal samt at sætte en GPIO-pin høj. Begge dele er til debugging af Regulerings MCU'en.

Motor styling:

Motor styling består af en PWM blok med to PWM outputs der går til henholdsvis Mos driver A og Mos driver B i motorstyringskredsen, se Vuggesystem Design Mosfet driver afsnit 6.2.1 på side 119. Outputtene er inverterede i forhold til hinanden og via PWMblokken er der lavet et deadband mellem de to PWMsignaler. For yderligere beskrivelse af PWM-styresignalet kan afsnit 6.2.1 på side 120 i Vuggesystem Design Styresignal ses.

Loop timing:

Denne funktionalitet bruges til at styre loop timingen for main funktionen i Regulerings MCU'en, også kaldet **vuggeControl**. Boksen indeholder tre hardware blokke, en Clock, en Timer og et Interrupt. Disse sørger tilsammen for at loop timingen for **vuggeControl** bliver 200 gange i sekundet.

Shutdown switch:

Denne GPIO-pin sættes høj (3.3V) hvis forbindelsen til motoren skal lukkes. Dette er ikke implementeret i denne iteration af projektet.

Controller I₂C:

Denne boks består af et I₂C-Slave interface der sørger for kommunikationen mellem Regulerings MCU'en og Baby Watch Controller. Denne I₂C bus forbindelse kører ligeledes med Normal-Mode.

Endstop switch:

Endstop switch boksen består af en pulled-up GPIO-pin forbundet med et Interrupt. Denne funktionalitet bruges til Endstopsensorne, afsnit 6.3 på side 113 i Vuggesystem Systemarkitektur Hardware arkitektur. Ved aktivering af en Endstop sensor trækkes signalet lav og interruptet aktives.

Beskrivelse af fysiske output fra PSoC4 Pioneer Kitte:

Følgende tabel beskriver sammenhængen mellem vuggesystemets Signalbeskrivelse under Vuggesystem System Arkitektur Grænsefladebeskrivelse 6.1 på side 114 og de tildelte fysiske porte i PSoC Creator 3.

Navn (Signalbeskrivelse)	Navn (PSoC Creator 3)	Port (PSoC)
MotorPWM	PWM A	P2[0]
MotorPWM	PWM B	P2[1]
MotorRelae	ShutdownSwitch	P3[4]
VuggeV	sensorI2C:scl	P0[4]
VuggeV	sensorI2C:sda	P0[5]
MotHast	Ikke implementeret	-
ESSwitch	ESSwitch	P1[5]
I2CVuggesystem	I2CVuggesystem:scl	P4[0]
I2CVuggesystem	I2CVuggesystem:sda	P4[1]

Til debugging er der tilføjet to ekstra porte. En til at styrer en GPIO-pin og en til et UART Tx output. De er beskrevet på forrige side under hardware boksen *Debug*

Navn (PSoC Creator 3)	Port (PSoC)
activityDebug	P3[6]
Tx 1	P1[4]

Portene er tildelt således at der tages hensyn til den interne fysiske kobling i PSoC4 chippen. Dette gøres for at reducerer støjen fra opadliggende porte så signaler såsom I2C ikke bliver påvirket af PWM signalerne eller andre I2C signaler, og GPIO interrupts ikke aktiveres tilfældigt. Nedenstående figur viser den interne kobling i PSoC4 chippen.

Beskrivelse af C kode filerne i Regulerings MCU:

Koden er implementeret som beskrevet i applikations modellen fra Vuggesystem Systemarkitektur afsnit 6.1.3 på side 116 og klassediagrammet fra Vuggesystem Design Software design 6.16 på side 125.

vuggeControl (main):

Tanken bag vuggesystemets main-funktion `VuggeControl`, er at den under opstart skal initier alle hardware blokkene og deres funktionaliteter igennem initierings kald til de andre filer i Regulerings MCU. Derefter skal den køre i loop som beskrevet i sekvensdiagrammet i afsnit 6.1.3 på side 116. Loopet kører med en frekvens på 200Hz ved hjælp af *Loop timing* hardware boksen beskrevet i ovenstående afsnit 6.3.2 på modstående side.

`VuggeControl`filen indeholder få hjælpefunktioner som hver især sørger for tjekke systemkritiske informationer, debugging eller nødstop af motoren(Hvilket ikke er fuldt implementeret i denne iteration af projektet).

Af systemkritiske informationer kan nævnes; et eksternt interrupt om at en Endstop Sensor er aktiveret og at systemet derfor skal gå i nødstop, at Baby Watch Controller vil slukke vuggesystemet og at reguleringens status skal tjekkes for fejl i reguleringen.

Nødstop-funktionen der lukker vuggesystemet er egentligt tiltænkt at skulle slukke et relæ der styrer strømtilførselen til motoren men som nævnt tidligere er denne funktionalitet ikke implementeret i denne udgave af projektet. Funktionen sørger istedet for at neutraliserer PWM-signalet der styrer motoren så motoren derigennem ikke får strøm.

Til at styrer main-loopets 200Hz hastighed bruges et timer-interrupt der sætter et flag højt hver gang et nyt gennemløb af loopet skal køres. I main-loopet kaldes der først fornævnte systemkritiske tjek efterfulgt af en opdatering af de nyeste vuggeudsving- og vuggefrekvens-værdier fra Baby Watch Controller, herefter hentes og beregnes barnevognskurvvens vinkel relativ til tyngdefeltet. Denne vinkel videresendes til reguleringen som beregner et output som lægges ind til motorstyringens PWM blok. Loopet slutter med at resette flaget så det er klart til et nyt gennemløb. Koden kan ses på [Bilags-CD, 2015] under `vuggeControl.c`

i2cKommunikation

`i2cKommunikation` filen sørger for initierer den fornævnte Controller I2C hardware der indeholdte et I2C Slave interface og derefter bruge denne til at opdaterer et I2C register med informationer som skal kommunikeres mellem de to moduler jf. Grænseflade mellem Vuggesystem og Controller i den overordnede Baby Watch Systemarkitektur afsnit 3.3.3.

Som sikkerhed i kommunikationen er der sat begrænsning på at register 0x04 der indeholder vuggesystemet status ikke kan ændres udfra dvs. fra Baby Watch Controller. Kode ses på: [Bilags-CD, 2015] under `i2cKommunikation.c` og `i2cKommunikation.h`

sensorFortolker

`sensorFortolker` står for alt kommunikation mellem sensorer (Endstop Sensor og Vuggeudsving sensor) og Regulerings MCU samt beregninger af data fra disse der giver barnevognskurvens nuværende vinkel. Dette gøres, som nævnt under Regulerings MCUs hardware moduler, vha. hjælp af et I2C Master interface og et GPIO styret interrupt.

Sensor konfiguration:

Da den valgte sensorkreds (MPU6050) er i stand til at køre på mange forskellige måder kræves der en indledende opsætning af sensoren før den kan benyttes. Dette gøres ved at skrive til en række registre på enheden gennem I2C bussen (REF I2C protokol). Der skrives til følgende registre:

Register	værdi	kommentar
0x19	0b00000100	sætter dataraten til 200Hz
0x1A	0b00000101	sætter knækfrekvensen for sensorens digitale lavpasfilter til 10 Hz
0x1A	0b00000101	sætter knækfrekvensen for sensorens digitale lavpasfilter til 10 Hz
0x1B	0b00011000	gyropskopet indstilles til +/- 2000 grader i sekundet
0x1C	0b00001000	accelerometeret indstilles til +/- 4g
0x1B	0b00011000	gyropskopet indstilles til +/- 2000 grader i sekundet
0x1C	0b00001000	accelerometeret indstilles til +/- 4g
0x6B	0b00000001	chippens clk sættes til at være x gyroskopets oscilator, da denne er mere præcis

Dataindsamling:

MPU6050 chippen genererer en puls på dataInt benet, hvorefter der læses fra følgende I2C registre:

Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3A	58	INT_STATUS	R	-	MOT_INT	-	FIFO_OFLOW_INT	I2C_MST_INT	-	-	DATA_ROY_INT
3B	59	ACCEL_XOUT_H	R					ACCEL_XOUT[15:8]			
3C	60	ACCEL_XOUT_L	R					ACCEL_XOUT[7:0]			
3D	61	ACCEL_YOUT_H	R					ACCEL_YOUT[15:8]			
3E	62	ACCEL_YOUT_L	R					ACCEL_YOUT[7:0]			
3F	63	ACCEL_ZOUT_H	R					ACCEL_ZOUT[15:8]			
40	64	ACCEL_ZOUT_L	R					ACCEL_ZOUT[7:0]			
41	65	TEMP_OUT_H	R					TEMP_OUT[15:8]			
42	66	TEMP_OUT_L	R					TEMP_OUT[7:0]			
43	67	GYRO_XOUT_H	R					GYRO_XOUT[15:8]			
44	68	GYRO_XOUT_L	R					GYRO_XOUT[7:0]			
45	69	GYRO_YOUT_H	R					GYRO_YOUT[15:8]			
46	70	GYRO_YOUT_L	R					GYRO_YOUT[7:0]			
47	71	GYRO_ZOUT_H	R					GYRO_ZOUT[15:8]			
48	72	GYRO_ZOUT_L	R					GYRO_ZOUT[7:0]			

Figur 6.28. udsnit af i2c register oversigt fra datablad

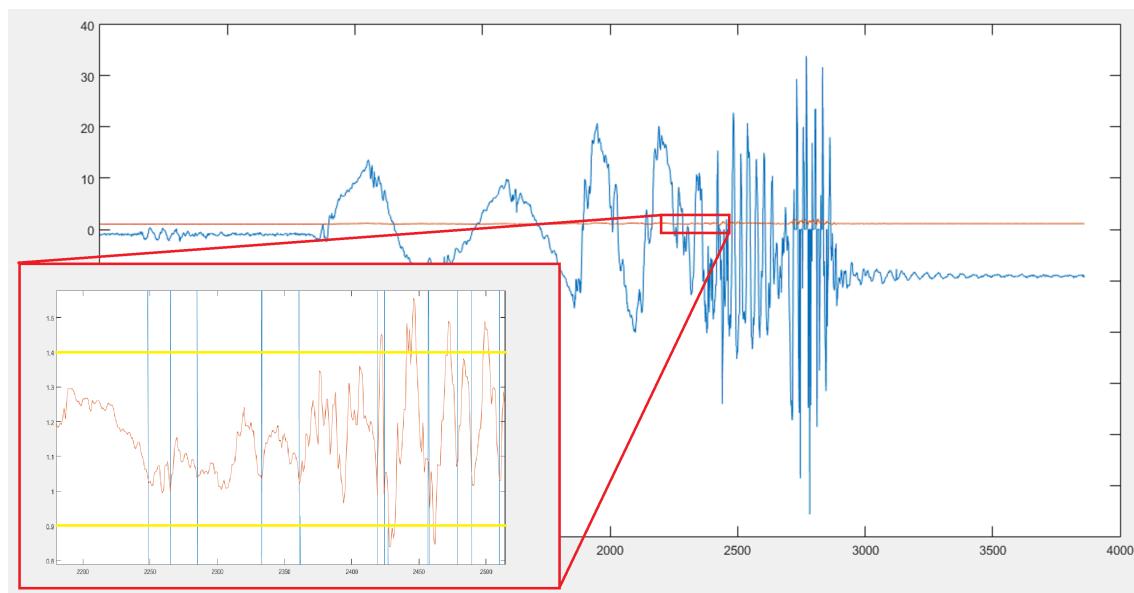
Heresfter bitshiftes de høje databytes (_H) 8 bits til vänstre og adderes med de tilhørende lave databytes.

textbfDatabehandling:

Før de indsamlede data fra de to sensorer kan samles udregnes først den estimerede vinkel for hver af de to sensorer hver for sig. Da Accellerometerets data kun kan bruges så længe sensoren ikke bliver udsat for nogen accelleration, benyttes accelerometer målingen kun så længe accellerationsvektoren har en længde næsten lig 1g (tyngdevektoren). For at kunne udregne dette uden at belaste processoren for meget benyttes følgende antagelse:

$$\sqrt{x^2 + y^2} \sim |x| + |y|$$

Da denne antagelse er ret upræcis, sættes grænserne lidt bredere end egentlig ønsket, og accellerometer dataene vurderes gældende hvis estimatet af accellerationsvektoren er i intervallet [0.9;1.4]. Dette interval er vurderet på baggrund af følgende måling: (fig 6.29)



Figur 6.29. plot af acc vector længde estimat ved alm vug og rytelser

Her ses et plot af accellerometer vinkelen (blå) og den estimerede længde af accellerationsvektoren (rød) først ved almindelig vugning, og derefter hvor barnevognen rystes, begyndende ved ca 2400 samples. De valgte grænser er markeret med gul.

Hvis målingen vurderes gyldig udregnes vinklen ved brug af arcus tangens for Z- og Y- aksens data, hvorved man får en vinkel, som vist herunder:

$$\theta_{acc} = atan2(a_z, a_y)$$

$$\text{hvor : } \text{atan2}(y, x) = \begin{cases} \arctan \frac{y}{x} & x > 0 \\ \arctan \frac{y}{x} + \pi & y \geq 0, x < 0 \\ \arctan \frac{y}{x} - \pi & y \geq 0, x < 0 \\ +\frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

Da denne beregning skal køre på psoc'en er der benyttet en fixed point implementering af cordic algoritmens atan2 funktion. For en mere detaljeret beskrivelse af Jim Shima's atan2 algoritme se dspGuru's artikkel [dspGuru fp atan2, 1999].

Gyroskopets data omregnes til en vinkel ved at integrere den målte vinkelfrekvens over hver sample periode som vist her:

$$\theta_{gyro} = \theta_{est[t-1]} + Gyro_x * T_s$$

$\theta_{est[t-1]}$ er den estimerede vinkel fra sidste sample periode. Grunden til at benytte den splejede måling frem for at gyroskop- og accellerometer data holdes separat, er at gyroskopets fejl ellers vil akkumulere, hvorved fordelen ved splejsning ville gå tabt.

Når de to vinkler er udregnes tages et vægtet gennemsnit af de to. Da gyroskopet giver de nærmeste data, og accellerometeret hovedsageligt bruges at at eliminere drift vægtes sensor dataene 9:1, og man får følgende formel:

$$\theta_{est} = 0.9 \cdot \theta_{gyro} + 0.1 \cdot \theta_{acc}$$

Hermed er den estimerede vinkel af barnevognen udregnet.

SensorFortolker koden ses på: [Bilags-CD, 2015] under `sensorFortolker.c` og `sensorFortolker.h`

regulerings

Denne fil sørger for at beregne regulerings outputtet, initierer PWM blokken og styrer denne PWM blok. Reguleringen sørger også for at lave fejltjek på på de vuggeudsvings- og vuggefrekvensværdier fra Baby Watch Controller før de bruges. Dette gøres for at øge datasikkerheden. Reguleringen bruger disse værdier sammen med den nuværende vinkel angivet i et fixed16.16 dataformat til at beregne et output til motorstyringens PWM.

MERE

Kode ses på: [Bilags-CD, 2015] under `regulering.c` og `regulering.h`

6.4 Modultest

Modultest for vuggesystemet beskriver først modultests af hardware modulerne efterfulgt af tests af softwaremodulerne.

6.4.1 Hardware modultest

Følgende afsnit beskriver tests for de individuelle hardwaremoduler i Vuggesystem-delen af Baby Watch.

Motorkreds

Formål

Formålet med denne modultest er at måle om motorkredsen kan styre en 12V motor der trækker op til 10A ved hjælp af et PWM kontrolleret input. For at testen kan godkendes skal motoren "vippe"en barnevognskurv over en horisontal akse jf. afsnit 2.5 i Ikke-funktionelle krav fra den ene yder position til den anden tre gange i sekvens. Der foretages to udgaver af testen en uden mervægt og en med en mervægt på 10kg placeret i barnevognskurven.

Overordnet opstilling af Motorkreds test

- Testen forgår i et rum med et horisontal nivelleret gulv
- Motoren er spændt på Baby Watch Vuggesystemet jf. afsnit Mekanik vuggesystem i Implementering for Vuggesystemet
- Motorens plus og minus indgange kobles til henholdsvis MOT A og MOT B jf. afsnit 6.2.1 i Hardware Design for Vuggesystemet
- Et PSOC4 Pioneer Kit (REFERENCE) programmeres med testprogrammet VuggeSinus jf. BILAG?? og de to PWM output kobles til henholdsvis Mos Driver A indgang M A og Mos Driver B indgang M B på motorkredsen jf. afsnit 6.2.1 i Hardware Design for Vuggesystemet
- I første udgave af testen placerets der ingen ekstra vægt i barnevognskurven. I anden udgave placeres en 10kg vægt på barnevognskurvens centertyngdepunkt.

Figur billede af opstillingen

Testbeskrivelse

- Opsæt systemet som beskrevet ovenfor
- Programmer PSoC4 Pioneer Kittet med VuggeSinus programmet
- Testen dokumenteres med video og billeder
- Testen gentages med en mervægt på 10kg i barnevognskurven

Forventet resultat

Det forventes at motorkredsløbet er i stand til at "vippe"barnevognen som beskrevet i Formålet tre gange i sekvens fra yder position til yder position både med og uden mervægt.

Resultat

// Tekst

// Figur 2 - Billede af resultat og reference til BILAG med video.

Testen er godkendt/ikke-godkendt

Endstopsensor

Formål

Formålet med testen er at se om Vuggesystemet melder om fejl ved aktivering af en af Endstopsensorerne.

Overordnet opstilling af Enstopsensor test

- Testen foregår i et rum med et horisontalt nivelleret gulv
- Baby Watch barnevognen monteres med Endstopsensorne jf. 6.3.1 i Implementeringen for Vuggesystemet
- Baby Watch barnevognskurven sættes i et vandret niveau
- PSoC4 Pioneer Kittet er tilsluttet en computer
- Computeren har startet et terminal program som har oprettet forbindelse til den givne com port PSoC4 Pioneer Kittet sidder på
- PSoC4 Pioneer Kittet programmeres med VuggeControl programmet

// Figur med billede af opstilling

Testbeskrivelse

- Opsæt systemet som beskrevet ovenfor
- Baby Watch barnevognskurven sættes ud i en yder position så Endstopsensoren aktiveres
- Der kontrollers på udskrift fra terminal programmet på computeren om Endstop-soren aktiverer en advarsel om aktivering. Tekst: "Endstop aktiveret"
- Testen gentages men denne gang med aktivering af Endstop-sensoren på den modsatte yder position

Forventet resultat

Det forventes at der ved aktivering ved begge yder positioner skrive "Endstop aktiveret" i terminal vinduet på computeren

Resultat

// Tekst

// Figur med billede af terminal udskriften samt et billede af den vippede barnevogn

Testen er godkendt/ikke-godkendt

Mekanisk vuggesystem

Formål

Formålet med denne modultest er at observere om det mekaniske vuggesystem er i stand til at "vippe" en barnevognskurv over en horizontal akse jf. afsnit 2.5 i Ikke-funktionelle krav fra den ene yder position til den anden tre gange i sekvens.

Overordnet opstilling af Mekanisk vuggesystem test

- Testen forgår i et rum med et horisontal nivelleret gulv
- Motoren er spændt på Baby Watch Vuggesystemet jf. afsnit Mekanik vuggesystem i Implementering for Vuggesystemet
- Motorens plus og minus indgange kobles til henholdsvis MOT A og MOT B jf. afsnit 6.2.1 i Hardware Design for Vuggesystemet
- Et PSOC4 Pioneer Kit (REFERENCE) programmeres med testprogrammet VuggeSinus jf. BILAG?? og de to PWM output kobles til henholdsvis Mos Driver A indgang M A og Mos Driver B indgang M B på motorkredsen jf. afsnit 6.2.1 i Hardware Design for Vuggesystemet

Figur billede af opstillingen

Testbeskrivelse

- Opsæt systemet som beskrevet ovenfor
- Programmer PSoC4 Pioneer Kittet med VuggeSinus programmet
- Testen dokumenteres med video og billeder

Forventet resultat

Det forventes at det mekaniske vuggesystem er i stand til at "vippe" barnevognen som beskrevet i Formålet tre gange i sekvens fra yder position til yder position.

Resultat

// Tekst

// Figur 2 - Billede af resultat og reference til BILAG med video.

Testen er godkendt/ikke-godkendt

6.4.2 Software modultest

Dette afsnit beskriver tests for de individuelle hardwaremoduler i Vuggesystem-delen af Baby Watch.

Vuggeudsving sensor

Formål

Formålet er at teste om Vuggeudsvingssensoren påmonteret en barnevognskurv der "vipper" over en horisontal akse jf. afsnit 2.5 i Ikke-funktionelle krav er i stand til at detektere vinklen i forhold til tyngdefeltet. For at testen kan godkendes forventes en præcision på mindst +/- 2 grader.

Overordnet opstilling af Vuggeudsving sensor test

- Testen foregår i et rum med et horisontalt nivelleret gulv
- Baby Watch barnevognen monteres med Vuggeudsvingssensoren jf. 6.3.1 i Implementeringen for Vuggesystemet
- Baby Watch barnevogns kurven sættes i et vandret niveau
- PSoC4 Pioneer Kittet er tilsluttet en computer
- Computeren har startet et terminal program som har oprettet forbindelse til den givne com port PSoC4 Pioneer Kittet sidder på
- PSoC4 Pioneer Kittet programmeres med vinkelmåler test programmet. (BILAG)

// Figur 1 - med evt. opstilling

Testbeskrivelse

- Opsæt systemet som beskrevet ovenfor
- Baby Watch barnevognskurven vippes med håndkraft fra yderposition til yderposition og stoppes i vandret position

- På terminalprogrammet på computeren udskrives data som angiver vinklen i fix16.16 format
- Disse data liggende ind i et data array i MatLab (REFERENCE) og plottes med en vertikal akse der angiver vinklen og en horisontal akse der angiver tid i sek

Forventet resultat

Der forventes en kontinuert kurve der begynder i 0 graders vinkel som efterfølgende skifter karakter til en trekantkurve der går fra -24.7 grader til 24.8 grader vinkel og til sidst stopper på 0 grader vinkel igen. Yderpositions vinklerne (-24,7 og 24.8 grader vinkel) baseres ud fra beskrivelsen af det mekaniske vugge implementationen afsnit 6.3.1.

Resultat

// Tekst

// Figur 2 - Billede af MatLab plot

Testen er godkendt/ikke-godkendt

Vuggesystemet

Formål

Formålet er at teste hele vuggesystemet som reguleringsmodul for Baby Watch systemet. Testen udføres i seks udgaver alle med en 10 kg mervægt i barnevognskurven. Testen dokumenteres med video.

*Testudgaver**

- Vuggesystemet regulerer barnevognskurven i en 0 graders vinkel-amplitude med en vippe-frekvens på 0 Hz
- Vuggesystemet regulerer barnevognskurven i sekvens af først 10 graders vinkel-amplitude med en vippe-frekvens på 0.5 Hz efterfulgt af 6 graders vinkel-amplitude med en vippe-frekvens på 1 Hz og til sidst af 4 graders vinkel-amplitude med en vippe-frekvens på 2 Hz
- Vuggesystemet regulerer barnevognskurven i sekvens af først 6 graders vinkel-amplitude med en vippe-frekvens på 1 Hz efterfulgt af 10 graders vinkel-amplitude med en vippe-frekvens på 0.5 Hz og til sidst af 4 graders vinkel-amplitude med en vippe-frekvens på 2 Hz
- Vuggesystemet regulerer barnevognskurven i 8 graders vinkel amplitude med en vippe frekvens på 0.75 Hz
- Barnevognskurven presses ned i en yderpositon så en endstop sensor aktives
- Barnevognskurven presses ned i modsatte yderpositon så den anden endstop sensor aktives

*Alle vinkler er i forhold til tyngdefeltet.

Overordnet opstilling af Vuggesystem modultest

- Testen foregår i et rum med et horisontalt nivelleret gulv
- Vuggesystemet er opsat jf. implementeringen af Vuggesystem afsnit 6.3
- Et videooptagelsesapparat er opsat og klar til optagelse

// Figur med billede af testopstillingen

Testbeskrivelse

- Testen opsættes om beskrevet ovenfor
- Hvert punkt i *Testudgaver* under Formål udføres og dokumenteres med video

Forventet resultat

Der forventes at:

- Barnevognskurven holdes i en vandret på position i forhold til tyngdefeltet
- Barnevognskurven vugges i en sekvens af først 10 graders vinkel-amplitude med en vippe-frekvens på 0.5 Hz efterfulgt af 6 graders vinkel-amplitude med en vippe-frekvens på 1 Hz og til sidst af 4 graders vinkel-amplitude med en vippe-frekvens på 2 Hz
- Barnevognskurven vugges i en sekvens af først 6 graders vinkel-amplitude med en vippe-frekvens på 1 Hz efterfulgt af 10 graders vinkel-amplitude med en vippe-frekvens på 0.5 Hz og til sidst af 4 graders vinkel-amplitude med en vippe-frekvens på 2 Hz
- Barnevognskurven vugges i en 8 graders vinkel amplitude med en vippe frekvens på 0.75 Hz
- Når barnevognskurven presses ned i en yderpositon aktiveres en endstop sensor og vuggesystemet stopper med alt vugning
- Når barnevognskurven presses ned i den modsatte yderpositon aktiveres en endstop sensor og vuggesystemet stopper med alt vugning

*Alle vinkler er i forhold til tyngdefeltet.

Resultat

// Resultatet beskrives og der henvis til optagelserne

Testen er godkendt/ikke-godkendt

Power Supply Unit

7

Dette kapitel omhandler strømdistributionen for Baby Watch. Som beskrevet kapitlet Overordnet så styrer Controlleren hvorvidt Vuggesystemet og den Intelligente Lydmonitor skal forsynes med strøm. Der benyttes et relæ til at styre dette. Forsyningssbatteriet skal reguleres ned til 3 forskellige spændinger:

- **Controller:** Kræver 5 V / 1200 mA (Raspberry Pi + Betjeningspanel)
- **Vuggesystem:** Kræver 3,3 V / 500 mA (PSoC + Motorstyringskreds)
- **Intelligent Lydmonitor:** Kræver 7,5 V / 1500 mA (BlackFin 533 kit + Mikrofonkreds)

Ved nedkonverteringen til 7,5 V benyttes en DC-DC konverter (LM2596S). Fra denne spænding reguleres der til hhv. 5 V og 3,3 V vha. spændingsregulatorer (LM317).

7.0.3 DC-DC konverter (LM2596S)

DC-DC konverteren (LM2596S), se figur 7.1, kan tage spændingsinput på 5 V til 35 V og outputte 1,25 V til 30 V. Maks strømoutput er 4 A og overstiger effekten 15 W, så skal den køles. I dette tilfælde skal DC-DC konverteren konvertere 12 V DC til 7,5 V DC. Ved 7,5 V og 15 W, så er der 2 A, da der skal benyttes maksimum 2,5 A, ca. 19 W, derfor er der i implementeringen påsat køleplade på begge sider af den regulator der skal køles.

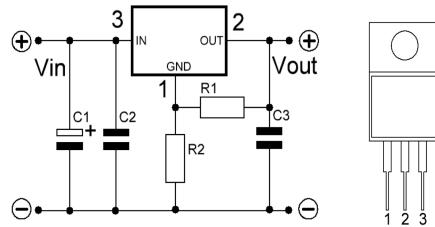


Figur 7.1. Fabriksbillede af LM2596S - DC-DC konverter

Som figur 7.1 er der 3 indstilige potentiometre samt 4 lodde øer. De 4 lodde øer er VIN-, VIN+, VOUT- og VOUT+. VIN- og VOUT- er ikke isoleret fra hinanden og kobles til samme stelplan. De 3 potentiometre bruges til at indstille udgangsspænding, maksimum udgangsstrøm og det sidste bruges til at styre en LED som lyser når strømmen når over det indstillede niveau.

7.0.4 Spændingsregulatorer (LM317)

Spændingsregulatoren LM317 er variabel justerbar mellem 1,2 V til 33 V og kan levere op til 3 A. Lm317 benyttes derfor til at regulere fra 7,5 V til hhv. 5 V og 3,3 V.



Figur 7.2. Schematic for LM317

Den variable udgangsspænding beregnes via formlen 7.1. I_Q er strømmen der løber fra ben 1, den er opgivet til at være $<100\mu A$ og typisk $50\mu A$, som er den værdi der vælges at benytte i formlen. Spændingsforskellen mellem udgangen (ben 2) og referencen (ben 1) er de $1,25 V$ der også optræder i formlen for V_{out}

$$V_{out} = 1,25V \cdot \left(1 + \frac{R2}{R1}\right) + I_Q \cdot R2 \quad (7.1)$$

Foruden den afsætte effekt til det objekt regulatoren skal forsyne, så løber der en strøm gennem R1 og R2, dette er et effekttab som skal regnes med i den samlede effekt afsættelse der afgør hvorvidt regulatoren skal køles. Strømen gennem R1 og R2 kaldes I_G og beregnes vha. ohms lov:

$$I_G = \frac{1,25V}{240\Omega} + 50\mu A = 5,26mA \quad (7.2)$$

Den samlede effekt beregnes ved formel 7.3

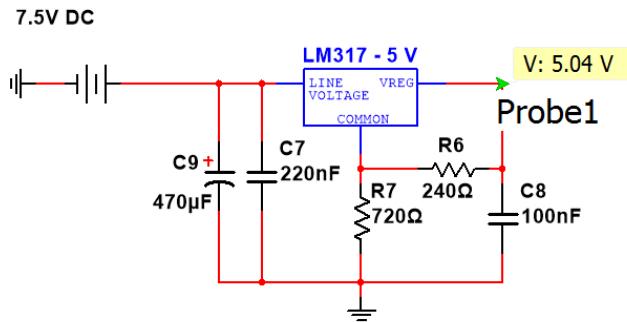
$$P = (V_{in} - V_{out}) \cdot I_L + V_{in} \cdot I_G \quad (7.3)$$

5 V til Controller

5 V skal benyttes til at forsynes Raspberry Pi'en. Ligning 7.4 viser beregningen af R2 når ønsket udgangsspænding er 5 V.

$$V_{out} = 1,25V \cdot \left(1 + \frac{720\Omega}{240\Omega}\right) + 50\mu A \cdot 720\Omega = 5V \quad (7.4)$$

R2 er beregnet til 720Ω og tilhørende simulering ses på figur 7.3. Simuleringen viser som forventet at udgangsspændingen er 5V.



Figur 7.3. Simulering af LM317 justeret til 5 V

Raspberry Pi'en forbruger maksimalt 1200mA og typisk 700mA i den ønskede opstilling. Der regnes med den maksimale værdi i forhold til at analyse på worst case tilfældet. Strømmen til Raspberry Pi'en skrives som I_L

$$P_{5V} = (7,5V - 5V) \cdot 1200mA + 7,5V \cdot 5,26mA = 3,39W \quad (7.5)$$

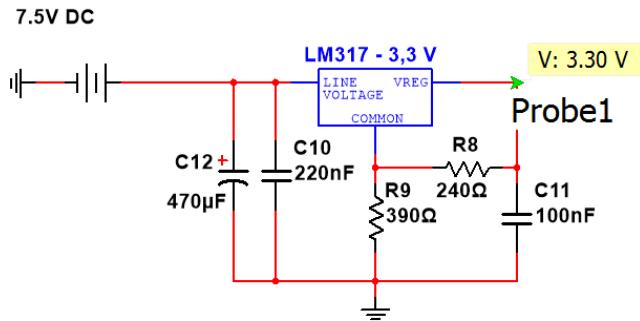
Ud fra formel 7.5 så er den maksimalt afsatte effekt i regulatoren 3,39W hvilket kræver køleplade og aktiv køling f.eks i form af en blæser, da effektafsættelsen er større en 1W som er grænsen for hvornår regulatoren kræver køling. Hvis der regnes med en $I_L = 700mA$ så bliver effektafsættelsen reduceret til 2,14W.

3,3 V til Vuggesystem

3,3 V til Vuggesystemet skal bruges til at forsyne PSoC'en, motorstyringskreds samt sensorer. Strømforbruget er sat til et maksimum på 500mA.

Ligning 7.6 viser at R2 er fundet til 390Ω når udgangsspændingen skal være 3,3 V. Figur ?? viser simuleringen og verificerer beregningen.

$$V_{out} = 1,25V \cdot \left(1 + \frac{390\Omega}{240\Omega} \right) + 50uA \cdot 390\Omega = 3,3V \quad (7.6)$$



Figur 7.4. Simulering af LM317 justeret til 3,3 V

Der foretages også effektberegning på denne regulator. Effektafsættelsen er mindre end ved 5 V reguleringen, men stadig over 1 W, så der skal også monteres køleplade på denne regulator.

$$P_{5V} = (7,5V - 3,3V) \cdot 500mA + 7,5V \cdot 5,26mA = 2,5W \quad (7.7)$$

7.0.5 Relæ styring

Forsyningen til den Intelligent Lydmonitor og til Vuggesystemet skal kunne afbrydes af Controlleren, når ON/OFF-knappen på betjeningspanelet sættes i OFF position. Vuggesystemet giver signal via I2C at forsyningen kan afbrydes og herefter sender Controlleren via signalet PowerOn et signal til PSU'en at 3,3 V og 7,5 V forsyningen skal afbrydes. Afbrydelsen foregår vha. et relæ. Relæet der benyttes er et Finder 40.52S som er godkendt til en middel belastning på 8 A hvilket dækker behovet.

DC coil data - 0.5 W sensitive (types 40.31/51/52/61)

Nominal voltage U_N V	Coil code 7.005	Operating range U_{min}^* V	U _{max} V	Resistance R Ω	Rated coil consumption I at U_N mA
5		3.7	7.5	50	100

Figur 7.5. Udsnit af datablad for Finder 40.52S

Figur 7.5 viser et udsnit af databladet for Finder 40.52S. Figuren viser at 5 V relæet fungerer med en spænding over spolen på 3,7 V til 7,5 V samt at relæet kræver 100 mA ved 5 V for at trække.

Raspberry Pi'ens GPIO23 skal sende powerOn signalet jf. Systemarkitekturen. Raspberry-en kan hverken forsyne relæet med 5 V eller 100 mA, derfor benyttes en transistor. BC517 kan løse opgaven da denne kan tåle 500 mA i kollektor strøm. Strømforstærkningen max h_{fe} er 30000, så i principippet burde en basestrøm på 3 uA være nok til de 100 mA. Der er dog valgt at basestrømmen skal være 0,1 mA for ikke at gå helt til grænsen for hvad der er minimum for at opnå de 100 mA.

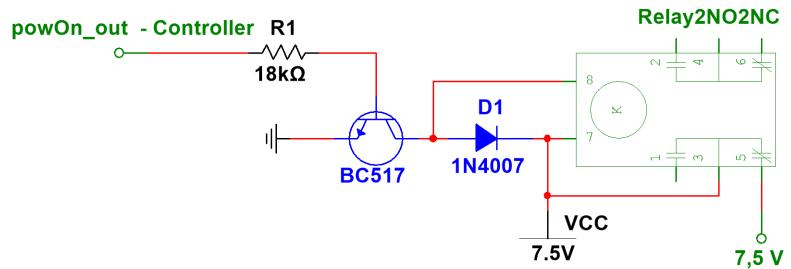
Raspberry Pi'en afgiver 3,3 V og BC517 har et base emitter spændingsfald på 1,4 V da det er en Darlington.

$$R_{base} = \frac{3,3V - 1,4V}{0,1mA} = \frac{1,9V}{0,1mA} = 19k\Omega \quad (7.8)$$

Ligning 7.8 viser udregningen af base modstanden. Modstandens værdi vælges til 18Ω da dette er en E24 standart.

Da relæet kan operere ved 7,5 V, benyttes denne spænding til at trække relæet og spændingen leveres af DC-DC konverteren. Der er en spændingsforskæl mellem kollektor og emitter på BC517 når denne er i mætning på 1 V ($V_{CEsat} = 1V$). Det efterlader 6,5 V til relæet når det skal klikke.

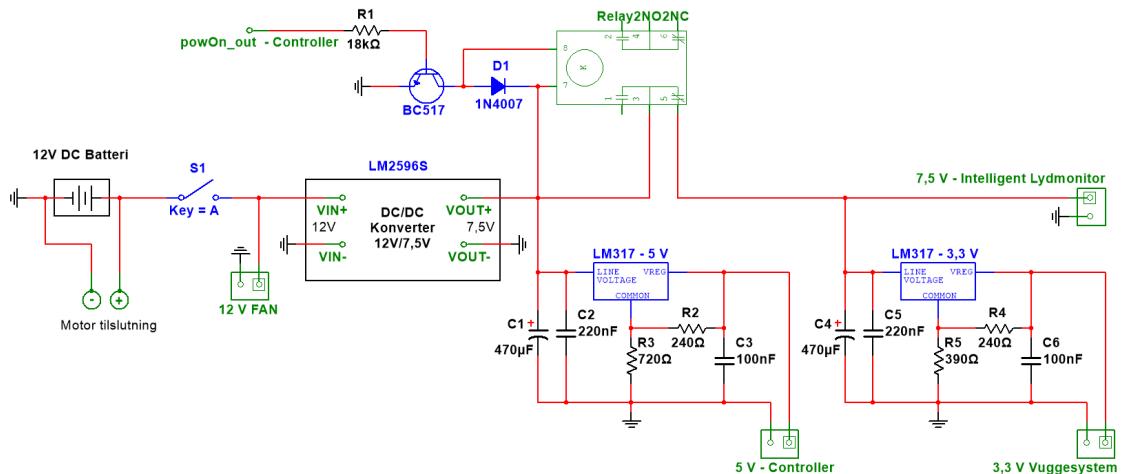
Samlet giver det anledning til nedenstående kredsløbsdiagram for relæstyringen se figur 7.6



Figur 7.6. Kredsløbsdiagram for relæstyring

Dioden (1N4007) i figur 7.6 er indsatt for at beskytte transistoren mod tilbageløbende strøm når relæet klikker.

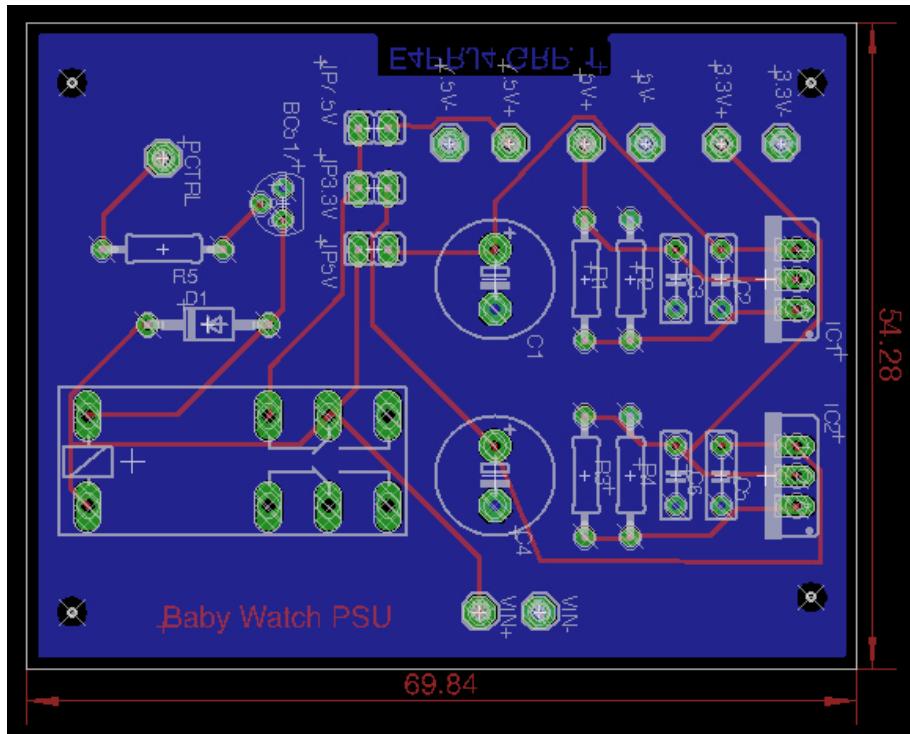
7.0.6 Samlet PSU print



Figur 7.7. Samlet kredsløbsdiagram for PSU'en

Figur 7.7 viser det samlede kredsløbsdiagram for PSU'en med tilhørende powOn_out pin som Raspberry Pi'ens GPIO23 skal forbindes til. Der er desuden indtegnet en hovedafbryder til PSU'en samt 12 V forbindelse til den blæser der skal køle spændingsregulatorerne.

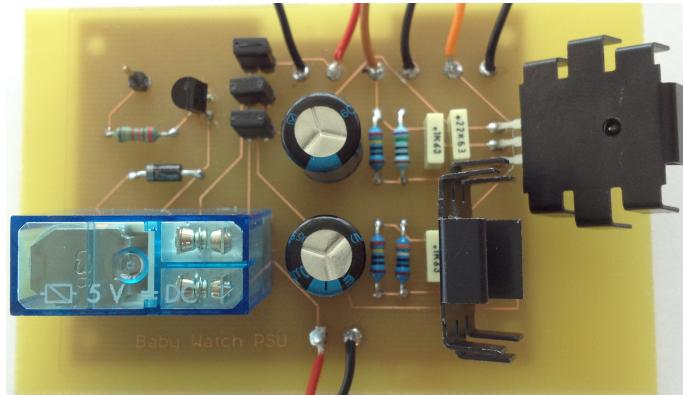
Figur 7.8 viser PCB printudlægget for PSU'en. Det er tegnet i programmet EAGLE, hvorefter de eksporterede Geber filer blev sendt til værkstedet som har produceret printet.



Figur 7.8. PCB layout

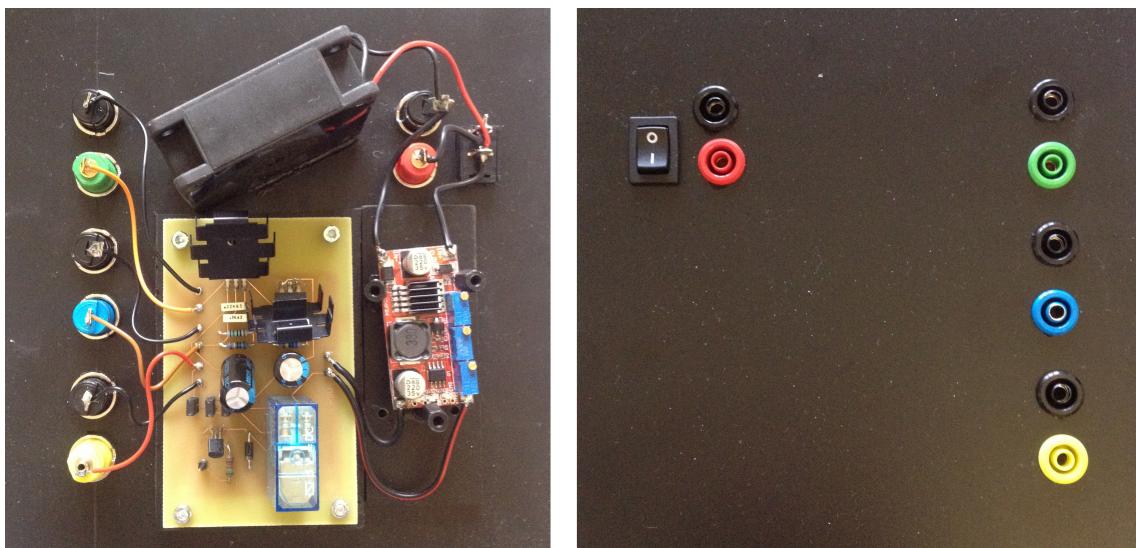
7.0.7 Implementering af PSU

Det udarbejde print er monteret med tilhørende komponenter. Komponenterne er monteret på oversiden, dvs. at de ikke nødvendigvis er helt nede ved printets overflade, da der skulle være plads nok til en loddekolbe. Det færdig monterede print kan ses på figur 7.9.



Figur 7.9. PSU print

Ovenstående print er sammen med DC-DC konverteren monteret på en Forex plade med banan bøsninger til ind og udgange af PSU'en. Der er isat en hovedafbryder i form af en enkeltafbryder. Blæseren er monteret således at begge spændingsregulatorer og DC-DC konverteren bliver kølet. Begge print er monteret med 1 cm luft til Forex pladen for at sikre god udluftning på begge sider af begge print. Det endelige resultat kan ses på figur 7.10.



Figur 7.10. PSU monteret på Forex.

7.0.8 Modultest

Integrationstest 8

Accepttestspezifikation

9

Use case 1

UC1: Igangsæt vugning manuelt				
	Test	Forventet Resultat	Resultat	Godkendt/ Kommentar
1	Tænd system. Tryk på "Manuel start"-knap	Visuel: Vugning starter på niveau jf. ikke-funktionelle krav i 5 min og stopper vugning		
2	Tænd system. Tryk på "Manuel start"-knap. Påvirk systemet med alarmerende lyd efter 1 min herefter med 1 min interval tre gange	Visuel: Vugning stopper efter påvirkning med alarmerende lyd efter 2. minut, alarmerende e-mail modtages og hjemmeside er opdateret til BABYCON1		
3	Tænd system. Tryk på "Manuel start"-knap. Påvirk system med lyden urolig efter 5 min.	Visuel: Vugning starter på niveau jf. ikke-funktionelle krav i 5 min og fortsætter vugning under monitorerings-tilstand		

Use case 2

UC2: Vug barnevogn				
	Test	Forventet Resultat	Resultat	Godkendt/ Kommentar
1	Tænd system. Påvirk systemet med lyden urolig i 3 min.	Visuel: Barnevognen vugger med en frekvens jf. ikke-funktionelle krav i 3 minutter og stopper derefter og hjemmesiden opdateres til BABYCON3		
2	Tænd system. Påvirk systemet med lyden urolig i 1 min. herefter med lyden alarmerende i 1 min.	Visuel: Barnevognen vugger med en frekvens jf. ikke-funktionelle krav i 1 minutter og går derefter i undtagelses-tilstand. Alarmerende e-mail modtages og hjemmesiden opdateres til BABYCON1		
3	Tænd system. Tryk på "Manuel start"-knap, vent 30 sekunder og overbelast barnevogn jf. ikke-funktionelle krav	Visuel: Overbelastnings e-mail modtages. Systemet går i undtagelses-tilstand		

Use case 3

Use case 3 testes i en sekvens fra start til slut.

UC3: Monitorér baby				
	Test	Forventet Resultat	Resultat	Godkendt/ Kommentar
1	Systemet tændes. Ingen lyd afspilles	Visuel: Barnevognen vugges ikke. Hjemmeside er opdateret til niveau BABYCON3		

...fortsat fra forrige side

	Test	Forventet Resultat	Resultat	Godkendt / Kommentar
2	Der afspilles lyden urolig	Visuel: Barnevognen vugges jævnfør de ikke funktionelle krav. Hjemmeside er opdateret til niveau BABYCON2		
3	Der afspilles lyde der afviger meget fra babylyde	Visuel: Barnevognen vugger ikke. Hjemmeside er opdateret til niveau BABYCON3		
4	Der afspilles lyden alarmerende	Visuel: Barnevognen vugger ikke. Hjemmeside er opdateret til niveau BABYCON1. Alarmerende e-mail modtages		
5	Netværksforbindelse afbrydes	Visuel: Wi-Fi-LED lyser		

Use case 4

UC4: Igangsæt undtagelsestilstand				
	Test	Forventet Resultat	Resultat	Godkendt / Kommentar
1	Tænd system. Påvirke system med alarmerende lyd	Visuel: Vugning er stoppet. Alarmerende e-mail modtages hvert 20. sekund. Hjemmeside er opdateret til BABYCON1		
2	Tænd system. Tryk på "Manuel start"-knap. Netværksforbindelse afbrydes	Visuel: Vugning er stoppet. Wi-Fi-LED lyser		
3	Tænd system. Påvirke system med urolig lyd. Afvent 30 sekunder. Netværksforbindelse afbrydes	Visuel: Vugning er stoppet. Wi-Fi-LED lyser		

...fortsat fra forrige side

	Test	Forventet Resultat	Resultat	Godkendt/ Kommentar
4	Tænd system, bring system i undtagelses-tilstand med alarmrende lyd. Genstart system. Afspil urolig lyd.	Visuel: Barnevogn vugger jf. UC2. Hjemmeside opdateret til BABYCON2.		

Ikke-funktionelle krav

	Test	Forventet Resultat	Resultat	Godkendt/ Kommentar
Mikrofon:				
1	Mikrofonens parametre undersøges gennem producentens dokumentation.	De anførte mikrofon parametre er i overensstemmelse med kravene	Skriv her	Skriv her
Vuggesystem:				
1	Vuggens vinkel måles med et digitalt vaterpas når vuggen er i højeste vugge niveau (aktiveret med manuel vuggestart), og det samlede udsving udregnes ud fra den største og mindste målte vinkel.	Det målte vinkel udsving ligger mellem 36° og 44°.	Skriv her	Skriv her
2	Vuggen indstilles til at vippe med hhv. 0.5, 1, 1.5, og 2 Hz, og der tælles hvor mange udsving vuggen gør over en periode på 30 sekunder.	Det samlede antal udsving for de 4 optællinger ligger inden for 10 % af hhv. 15, 30, 45, og 60.	Skriv her	Skriv her
3	Vuggen slukkes mens vuggen er nær ved fuldt udsving, og vinklen af vuggen måles når vuggen er faldet til ro.	Den målte vinkel skal være mellem -5° og 5°.	Skriv her	Skriv her

...fortsat fra forrige side

	Test	Forventet Resultat	Resultat	Godkendt / Kommentar
4	FIXME: Test af begrænsning i vinkelfrekvens, og vinkelacceleration er svær at udtænke. TODO: snak med Carl	Skriv her	Skriv her	Skriv her
	Baby status			
1	Systemet konfigureres til at skrive til en log ved opdatering af baby status, og får lov at køre i 1 min. Mens systemet kører overvåges hjemmesiden, og opdateringstidspunkterne noteres. Til sidst sammenholdes loggen med de noterede tider, og forsinkelsen beregnes som differencen fra opdatering af babystatus, til opdatering af hjemmesiden.	Den længste pause mellem to opdateringer af baby status, og den største forsinkelse til opdatering af hjemmesiden er under 5 s.	Skriv her	Skriv her

Bilag (CD-indhold)

10

Litteratur

- Bilags-CD (2015). ‘CD med bilag tilhørende E4GRP1’.
- Cypress (2014). ‘CY8CKIT-042 PSoC 4 Pioneer Kit’.
- Cypress (2015). ‘Cypress PSoC Creator 3’.
- dspGuru fp atan2 (1999). ‘DSP Trick: Fixed-Point Atan2 With Self Normalization’.
- Mathworks (1994-2015). ‘max’.