

# **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

---

<b>Kapitel 1 Termliste</b>	<b>5</b>
<b>Kapitel 2 Kravspecifikation</b>	<b>7</b>
2.1 Indledning . . . . .	7
2.2 Use cases . . . . .	10
2.3 Aktører . . . . .	10
2.4 Fully dressed use cases . . . . .	11
2.5 Ikke-funktionelle krav . . . . .	14
2.6 Web-skitser . . . . .	15
2.7 E-mail beskeder . . . . .	17
<b>Kapitel 3 Baby Watch</b>	<b>19</b>
3.1 Systemarkitektur . . . . .	19
3.1.1 Domænemodel . . . . .	19
3.1.2 System-states . . . . .	20
3.1.3 BDD . . . . .	21
3.1.4 IBD . . . . .	21
3.2 Signalbeskrivelse . . . . .	22
3.2.1 Allokering af logiske funktionaliteter . . . . .	23
3.3 Grænseflader . . . . .	24
3.3.1 Grænseflade mellem Intelligent Lydmonitor og Controller . . . . .	24
3.3.2 Grænseflade mellem Vuggesystem og Controller . . . . .	25
3.3.3 Grænseflade mellem Controllers servere og ekstern internet enhed . . . . .	27
<b>Kapitel 4 Controller</b>	<b>29</b>
4.1 Systemarkitektur . . . . .	29
4.1.1 Softwarearkitektur . . . . .	30
4.1.2 Hardware arkitektur . . . . .	32
4.1.3 Signaltabel . . . . .	32
4.2 Design . . . . .	33
4.2.1 Softwaredesign . . . . .	33
4.2.2 Hardwaredesign . . . . .	38
4.3 Implementering . . . . .	44
4.3.1 Softwareimplementering . . . . .	44
4.3.2 Hardwareimplementering . . . . .	47
4.4 Modultest . . . . .	47
<b>Kapitel 5 Intelligent lydmonitor</b>	<b>49</b>
5.1 Forundersøgelse . . . . .	49
5.1.1 Metode . . . . .	49
5.1.2 Analysekode gennemgang . . . . .	50

---

5.1.3 Analyser . . . . .	52
5.1.4 Konklusion . . . . .	67
5.2 Systemarkitektur . . . . .	68
5.2.1 Hardware arkitektur . . . . .	69
5.2.2 Software arkitektur . . . . .	70
5.3 HW design og implementering . . . . .	77
5.3.1 Mikrofon preamp . . . . .	77
5.4 SW design og implementering . . . . .	82
5.4.1 Prefilter . . . . .	82
5.4.2 RecBuf . . . . .	82
5.4.3 Analyzer . . . . .	82
5.4.4 Categorizer . . . . .	82
5.4.5 Statistician . . . . .	82
5.4.6 TwoWireCom . . . . .	82
5.5 Modultest . . . . .	84
5.5.1 Modultest Prefilter . . . . .	84
5.5.2 Modultest RecBuf . . . . .	84
5.5.3 Modultest Analyzer . . . . .	84
5.5.4 Modultest Categorizer . . . . .	84
5.5.5 Modultest Statistician . . . . .	84
5.5.6 Modultest TwoWireCom . . . . .	84
<b>Kapitel 6 Vuggesystem</b>	<b>87</b>
6.1 Systemarkitektur . . . . .	87
6.1.1 Hardware arkitektur . . . . .	87
6.1.2 Grænsefladebeskrivelse . . . . .	90
6.1.3 Software arkitektur . . . . .	91
6.2 Design . . . . .	93
6.2.1 Hardware design . . . . .	93
6.2.2 Software design . . . . .	99
6.3 Implementering . . . . .	102
6.3.1 Hardware implementering . . . . .	102
6.3.2 Software implementering . . . . .	102
6.4 Modultest . . . . .	102
6.4.1 Hardware modultest . . . . .	102
6.4.2 Software modultest . . . . .	103
<b>Kapitel 7 Power Supply Unit</b>	<b>105</b>
7.0.3 DC-DC konverter (LM2596S) . . . . .	105
7.0.4 Spændingsregulatorer (LM317) . . . . .	106
7.0.5 Relæ styring . . . . .	106
<b>Kapitel 8 Integrationstest</b>	<b>107</b>
<b>Kapitel 9 Accepttestspezifikation</b>	<b>109</b>
<b>Kapitel 10 Bilag (CD-indhold)</b>	<b>115</b>

# 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.

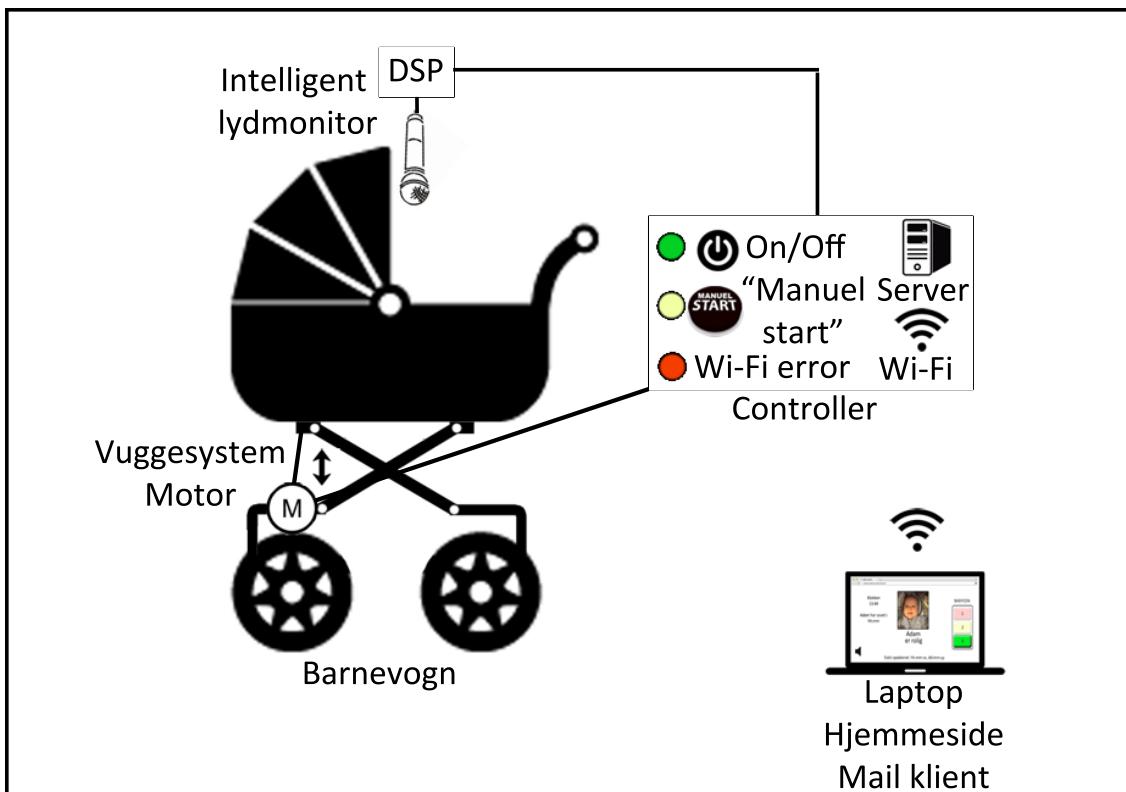
- **BABYCON2**

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.



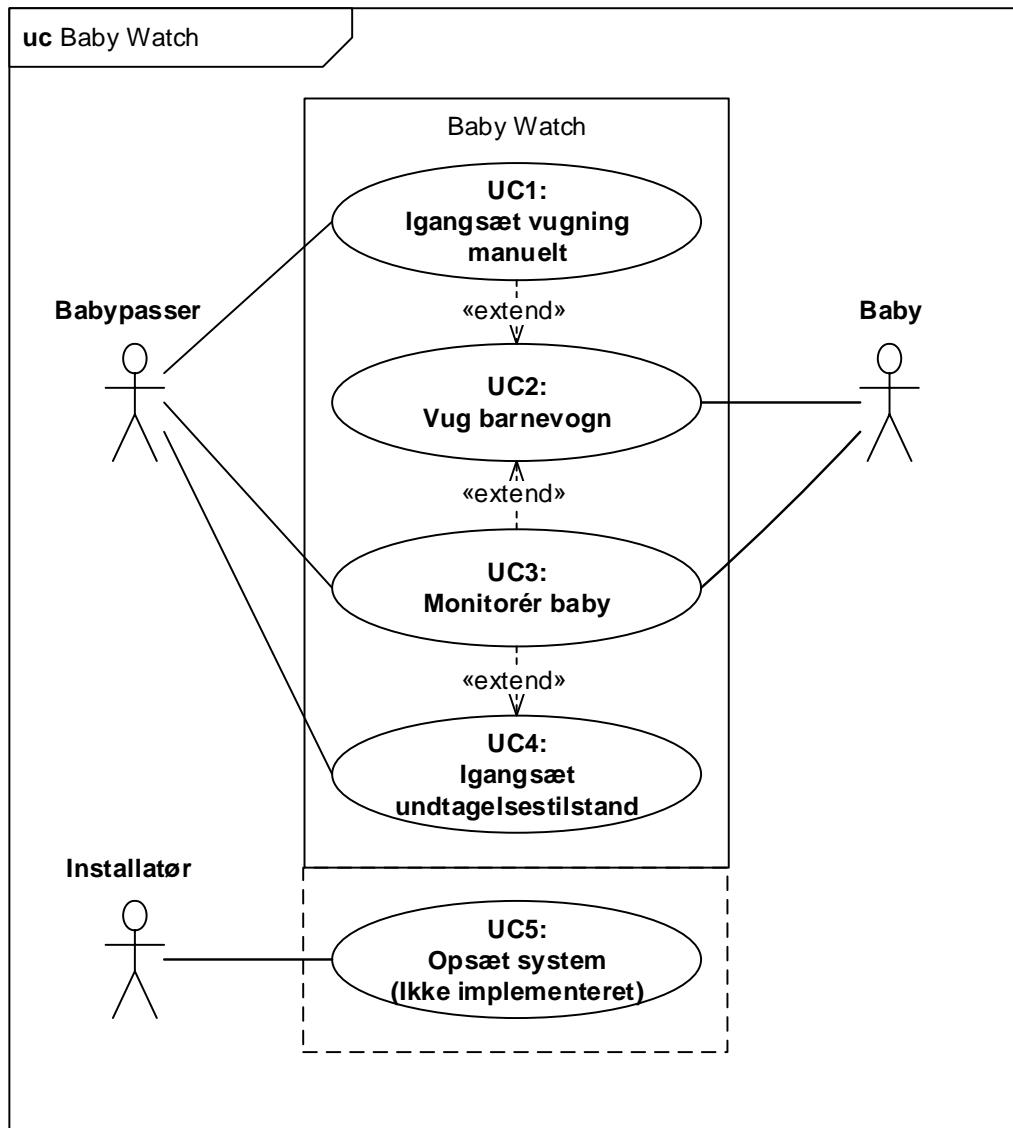
*Figur 2.2.* Controllerens knapper og lysdioder

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"> <li>1. Babypasser igangsætter manuel vugning ved tryk på "Manuel start"-knap på Baby Watch brugerpanel</li> <li>2. Systemet starter vugning jf. UC2</li> <li>3. Efter 2 minutter igangsætter systemet passiv monitorering af baby jf. UC3</li> <li>4. Efter 5 minutter igangsætter systemet automatisk monitorering af baby jf. UC3 [Und: 4.a Alarmerende baby detekteret indenfor 5 min]</li> </ol>
Undtagelser	<p>4.a Alarmerende baby detekteret indenfor 5 min</p> <ul style="list-style-type: none"> <li>• Systemet igangsætter undtagelsestilstand jf. UC4</li> </ul>

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

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

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

## 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.<sup>1</sup>
- 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:

<sup>1</sup>FIXME indsæt reference til studie om gråd volumen

- Vuggen skal kunne vippe planet, hvorpå babyen ligger, med op til  $10^\circ$  i hver retning fra dets vandrette udgangspunkt, med en fejlmargin på  $2^\circ$ .
- 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  $2^\circ$  når systemet lukkes ned.
- Vuggen skal have en begrænsning på vinkelfrekvensen ved  $80^\circ \text{ s}^{-1}$ , med en fejlmargin på 10 %.
- Vuggens vinkel acceleration skal være begrænset ved  $20^\circ \text{ s}^{-2}$ , 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.



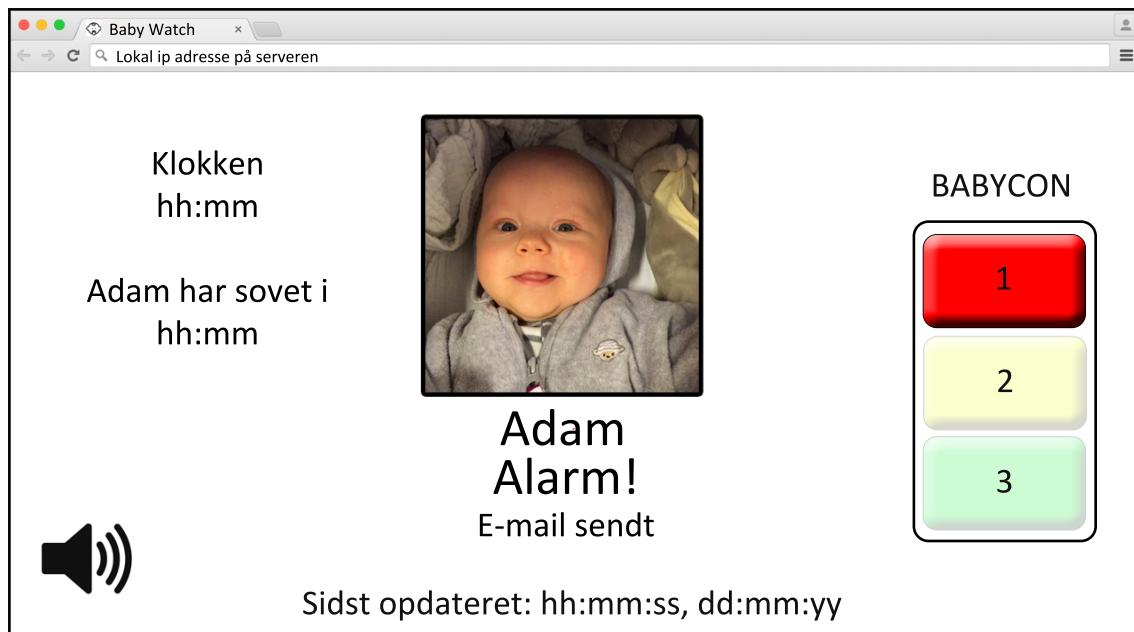
*Figur 2.4.* BABYCON3

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.

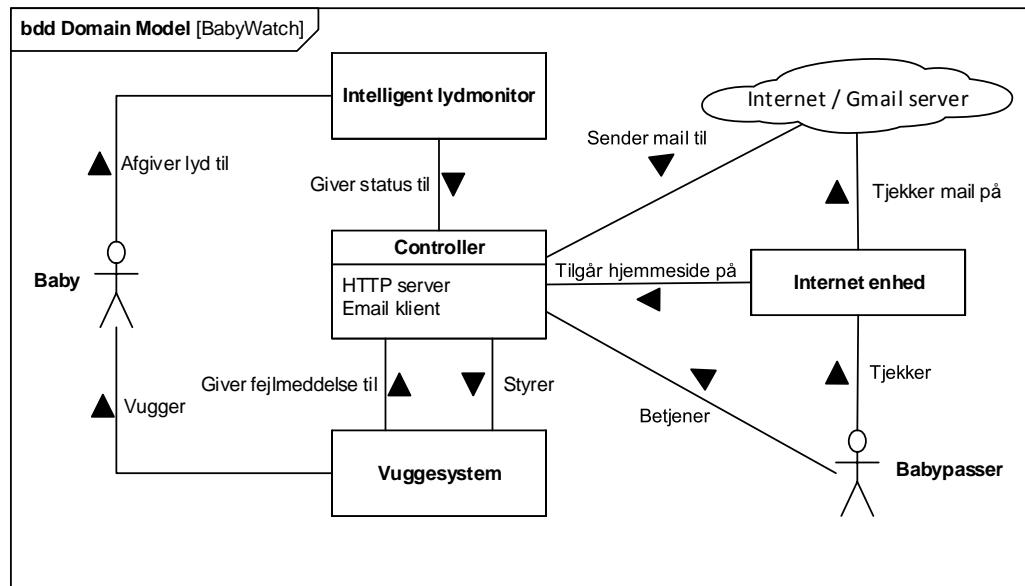


# Baby Watch 3

## 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.

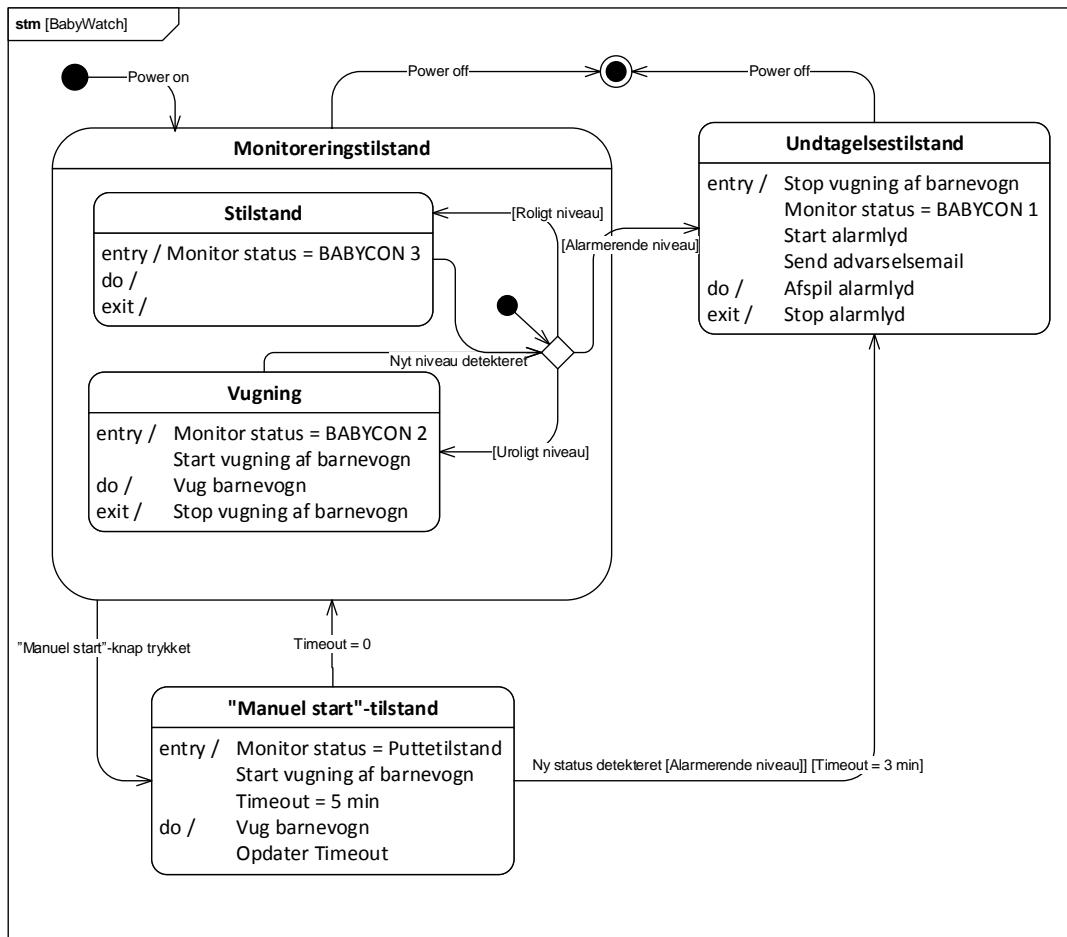
### 3.1.1 Domænemodel



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.

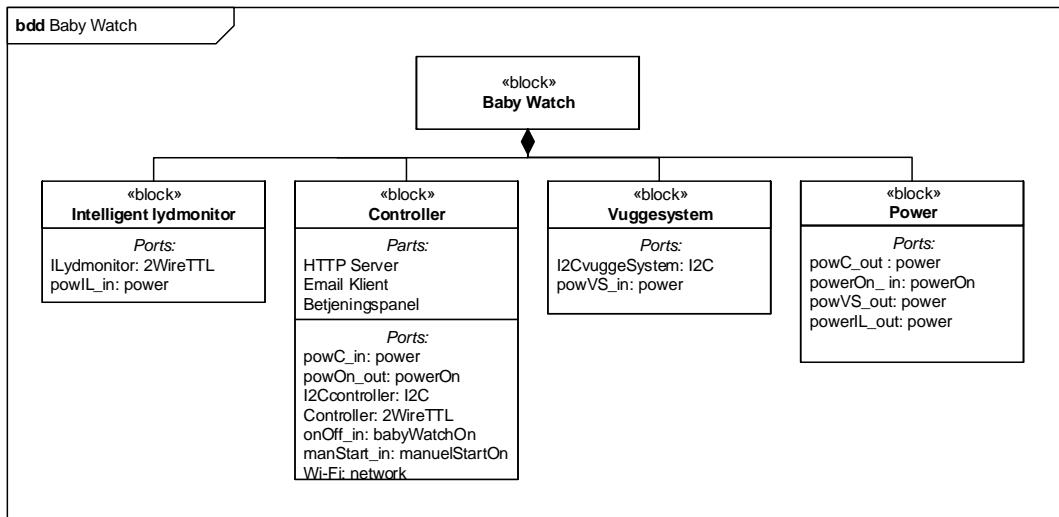
**Monitoringstilstand** har to substates:

- **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.

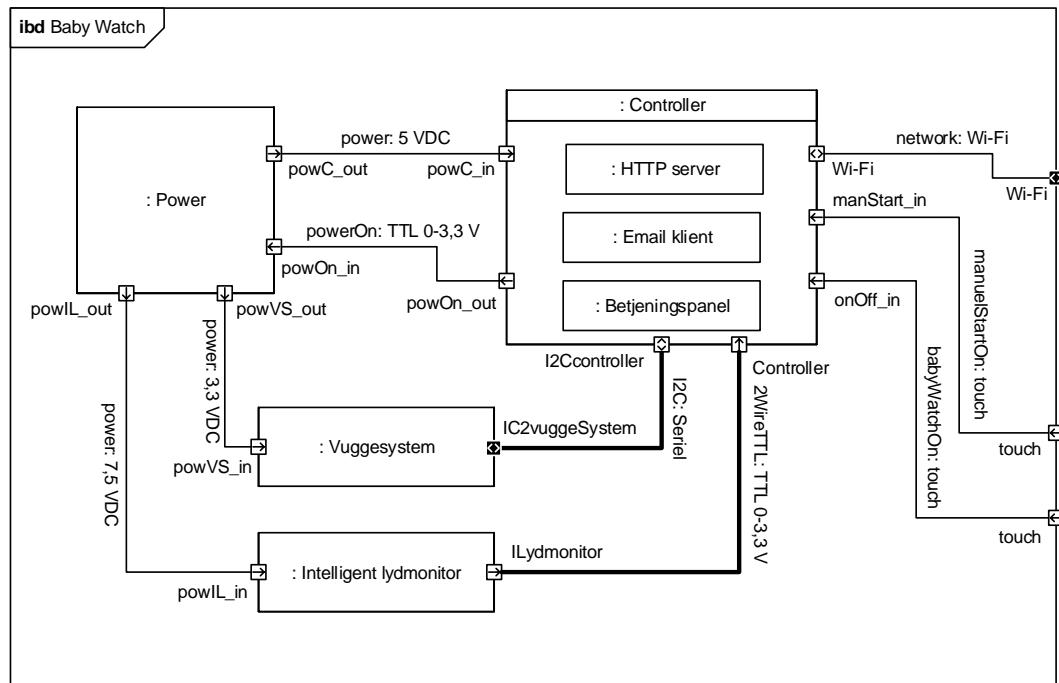
### 3.1.3 BDD



**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.

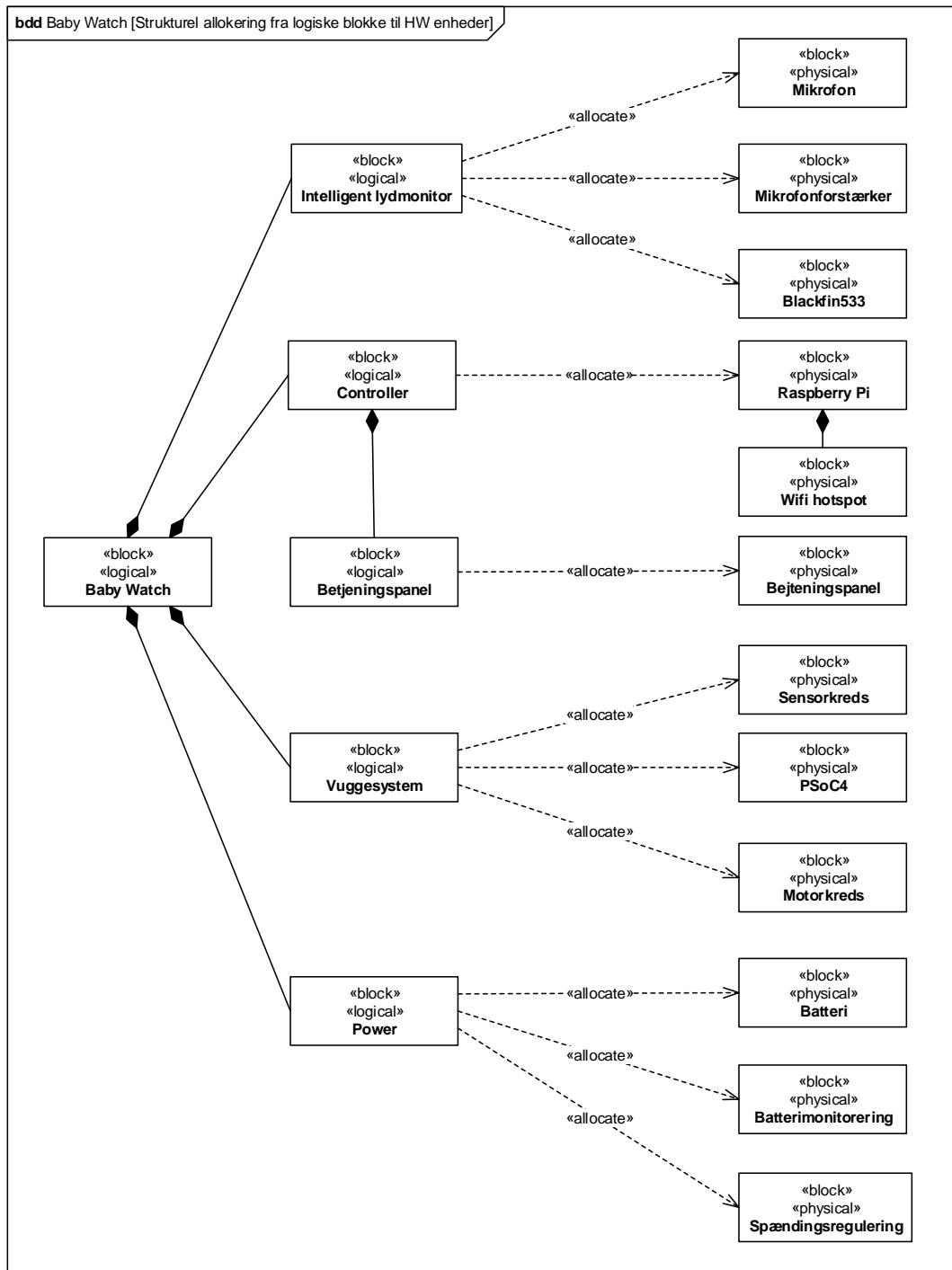
## 3.2 Signalbeskrivelse

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	Power (powC_out)	Controller (powC_in)	5 V forsyning til controller
power	3,3 VDC	Power (powVS_out)	Vuggesystem (powVS_in)	Forsyning til Vuggesystem
power	7,5 VDC	Power (powIL_out)	Int. lydmonitor (powIL_in)	Forsyning til Vuggesystem
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 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 niveauer.

##### 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*

	<b>2WireTTL _ MSB</b>	<b>2WireTTL _ LSB</b>
<b>Bit værdi</b>	0	1

*Tabel 3.3. BABYCON2*

	<b>2WireTTL _ MSB</b>	<b>2WireTTL _ LSB</b>
<b>Bit værdi</b>	1	0

*Tabel 3.4. BABYCON3*

	<b>2WireTTL _ MSB</b>	<b>2WireTTL _ LSB</b>
<b>Bit værdi</b>	1	1

*Tabel 3.5. Fejl fra Intelligent Lydmonitor*

	<b>2WireTTL _ MSB</b>	<b>2WireTTL _ LSB</b>
<b>Bit værdi</b>	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 Intelligent Lydmonitors side.

### 3.3.2 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

<b>I2C Adresse:</b> 0b1111000X (Write: 0xF0, Read: 0xF1)				
<b>I2C Frekvens:</b> 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: <b>[ERR STALL END_STP SD_RDY X X X X]</b> et 1 tal indikerer en fejl	0b0000XXXX
<b>ERR</b>		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		
<b>STALL</b>		Indikerer at systemet har været ude af stand til at drive motoren, formentlig pga. for stor belastning		
<b>END_STP</b>		Indikerer at vuggen har ramt en af sine mekaniske yderpositioner.		
<b>SD_RDY</b>		Indikerer at systemet er klar til at få afbrudt strømmen		

### 3.3.3 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

---

## 4.1 Systemarkitektur

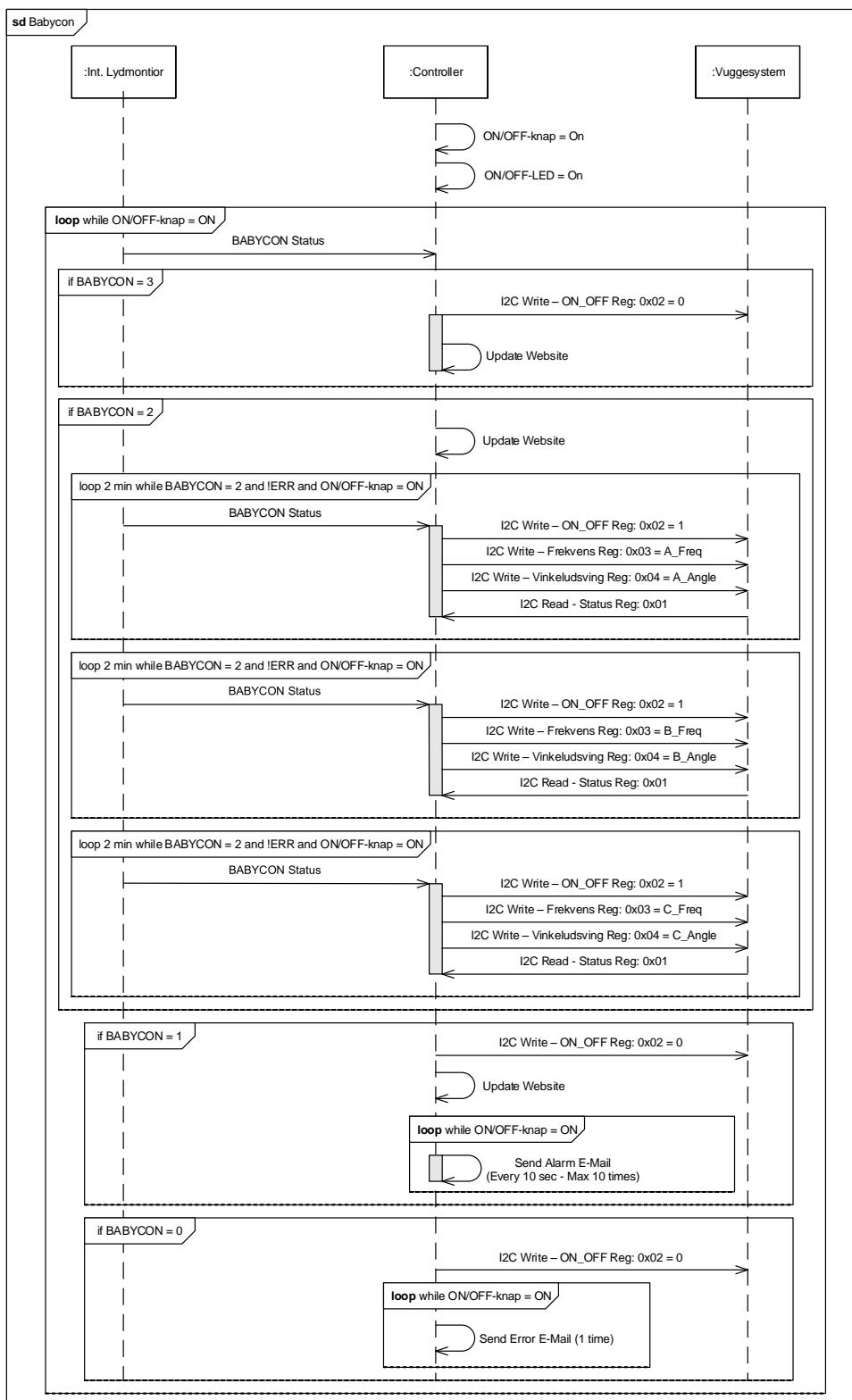
I dette afsnit beskrives systemarkitekturen for Controller.

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

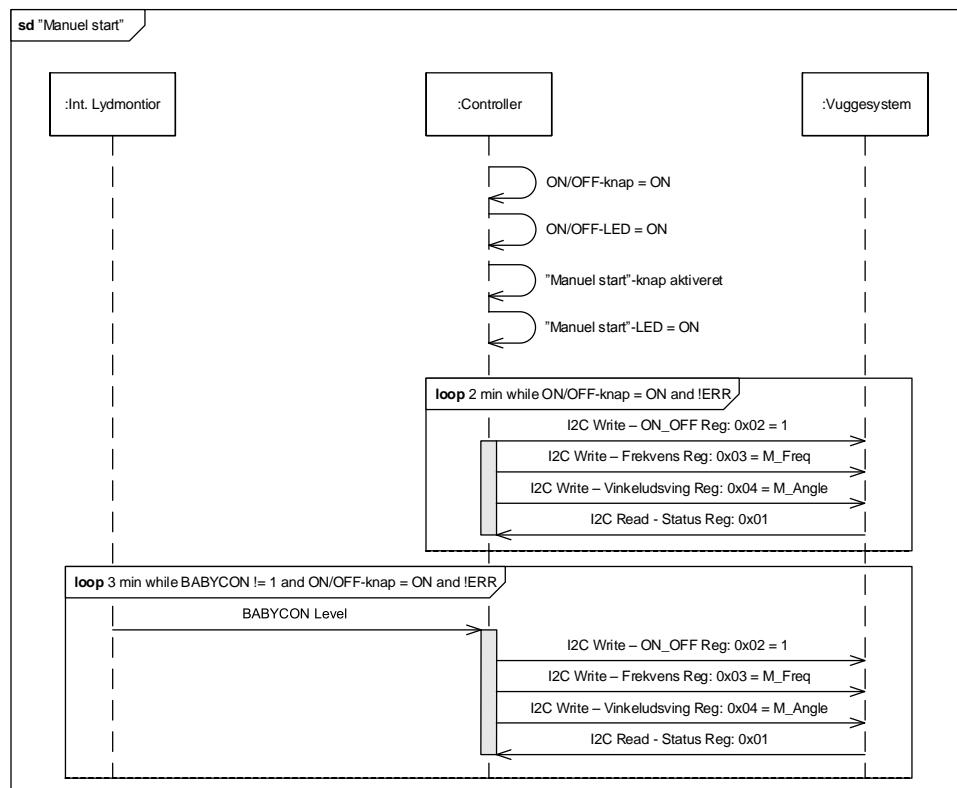
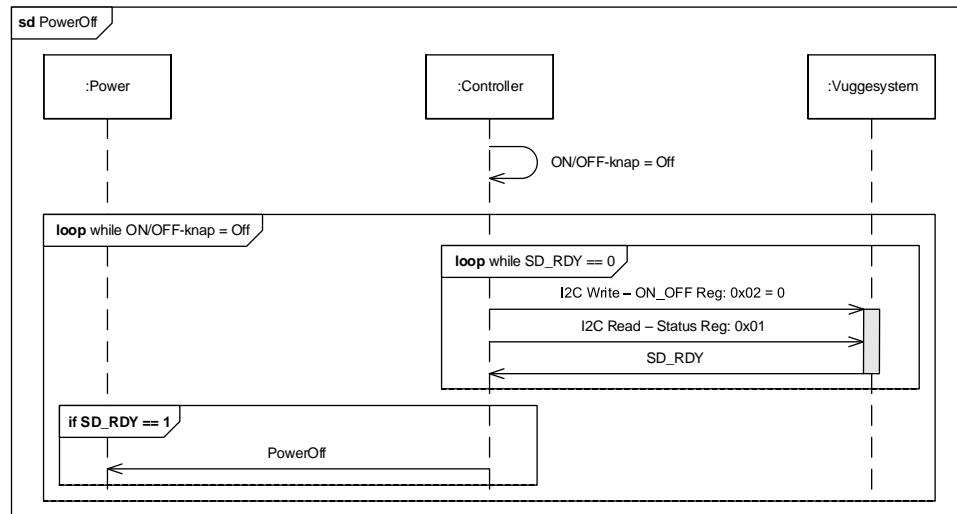
Hardwarearkitekturen beskriver den fysiske grænseflade til den intelligente lydmonitor og vuggesystemet.

### 4.1.1 Softwarearkitektur

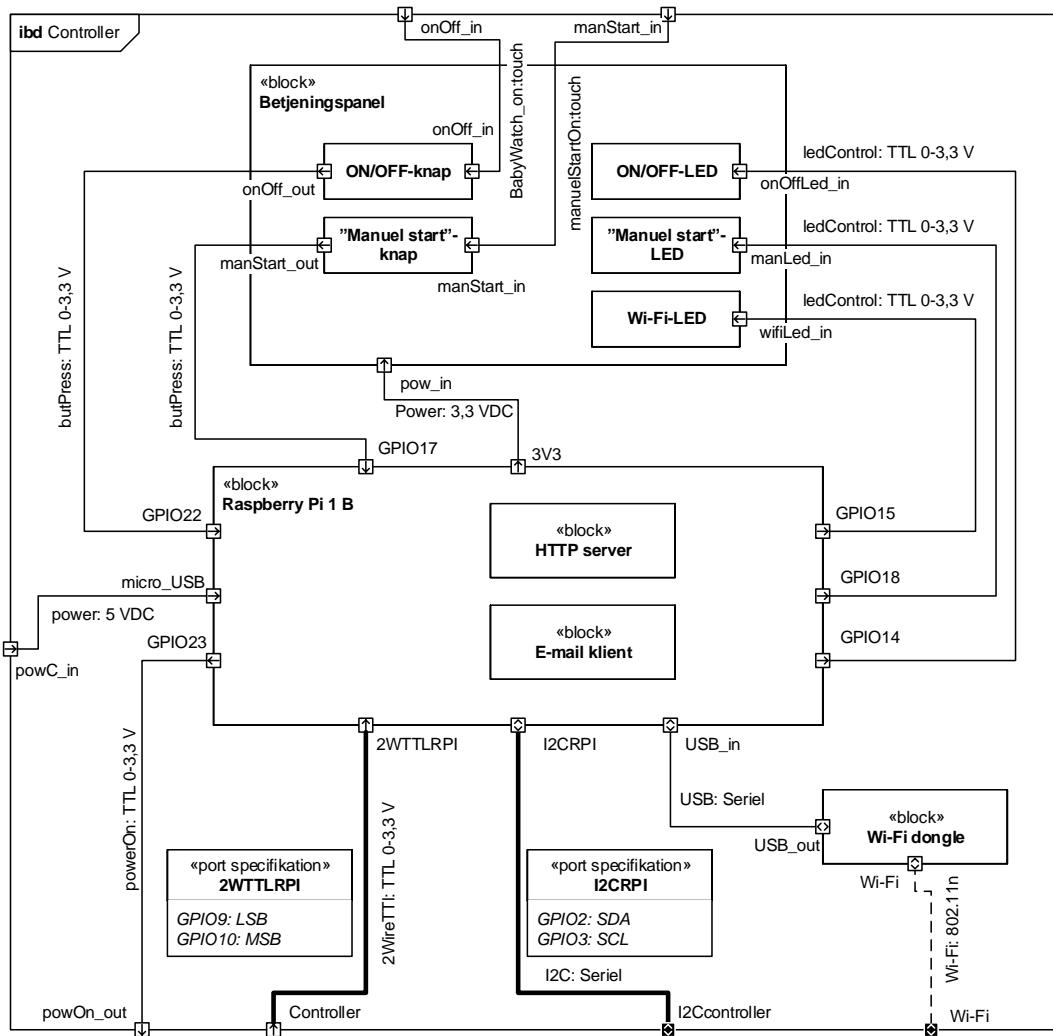
#### Sekvensdiagrammer



**Figur 4.1.** Sekvensdiagram for Babycon niveauer i main

**Figur 4.2.** Sekvensdiagram for ”Manuel Start” i main**Figur 4.3.** Sekvensdiagram for nedlukning i main

### 4.1.2 Hardware arkitektur



**Figur 4.4.** IBD diagram for Controller

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

Signal-navn	Type	Port 1	Port 2	Kommentar
manStartOn	touch	Betjeningspanel (manStart_in)	Babypasser (touch)	Babypasser der laver et tryk på knappen

...fortsat fra forrige side

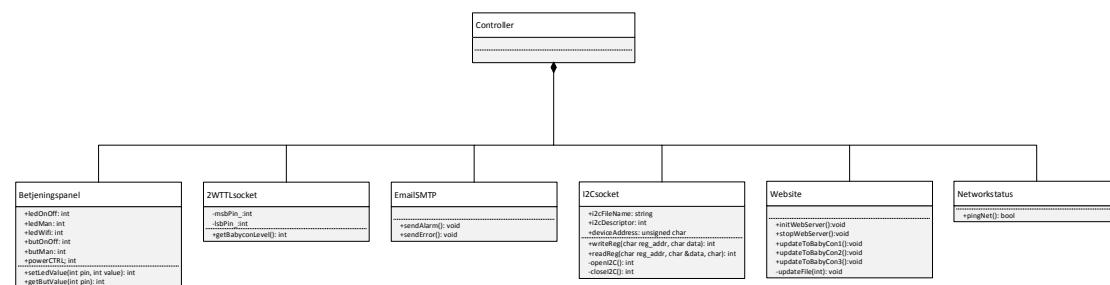
Signal-navn	Type	Port 1	Port 2	Kommentar
babyWatchOn	touch	Betjeningspanel (onOff_in)	Babypasser (touch)	Babypasser der laver et tryk på knappen
power	5 VDC	RaspberryPi (micro_USB)	Controller (powC_in)	5 V forsyning til controller
power	3,3 VDC	RaspberryPi (3V3)	Betjeningspanel (pow_in)	3,3 V forsyning 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)	Seriel kommunikation
I2C	Seriel	RaspberryPi (I2CRPI) GPIO2: SDA GPIO3: SCL	Controller (Controller)	BABYCON niveau
USB	Seriel	RaspberryPi (USB_in)	Wi-Fi dongle (USB_out)	Seriel 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

## 4.2 Design

### 4.2.1 Softwaredesign

I dette afsnit beskrives hvordan Controllerens software er designet med klassediagrammer

## Klassebeskrivelser



**Figur 4.5.** Oversigt over klassediagrammerne for Controller

### Betjeningspanel

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

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

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

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

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

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

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

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

### EmailSMTP

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

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

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

### GPIOsocket

**Ansvar:** At oprette et GPIO filarkiv

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

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

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

### I2Csocket

**Ansvar:** At kommunikere over I2C

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

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

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

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

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

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

<b>Attribut</b>	<code>unsigned char deviceAddress</code>
<b>Beskrivelse</b>	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

<b>Funktion</b>	<code>bool pingNet()</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	True ved forbindelse ellers false
<b>Beskrivelse</b>	Pinger googles 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

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

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

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

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

### Website

**Ansvar:** At styrer HTTP serveren med Baby Watch websiden

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

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

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

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

<b>Funktion</b>	<code>void updateToBabyCon3()</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Opdatere Baby Watch HTTP-webserveren til BabyCon 1 niveau

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

### 4.2.2 Hardwaredesign

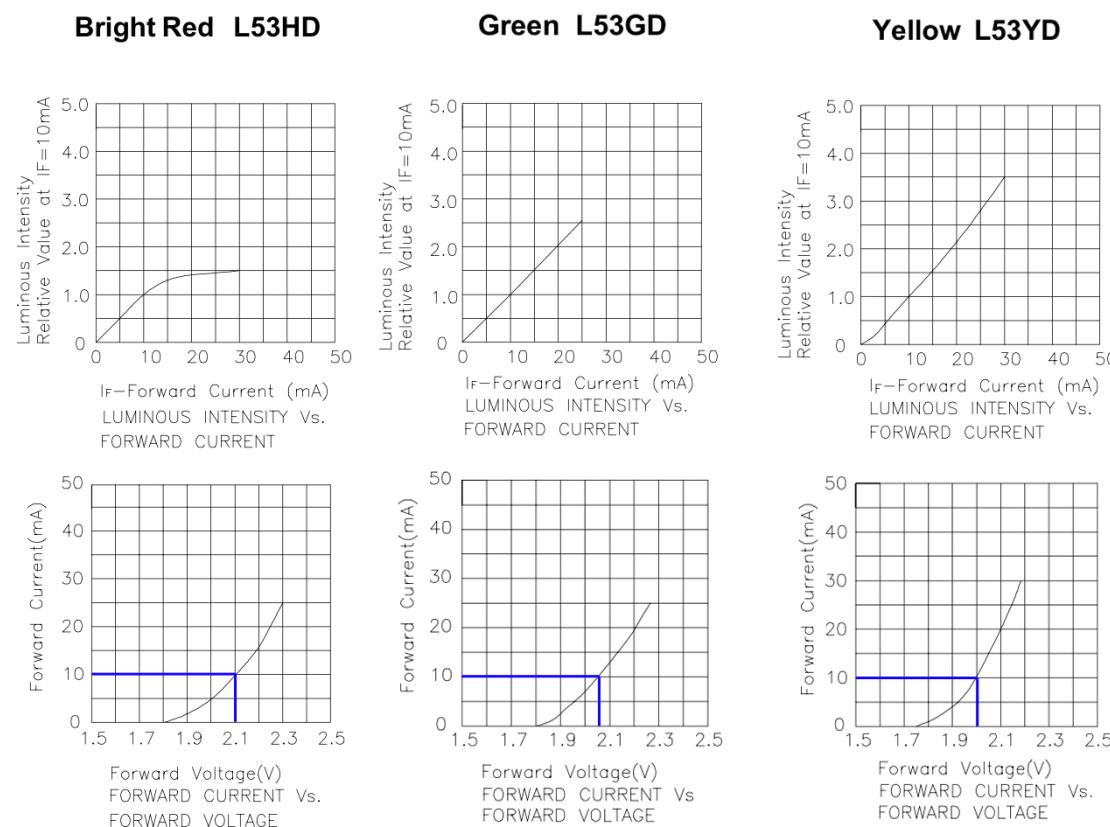
Controlleren består af et betjeningspanel samt en Raspberry Pi der kontrollerer betjeningspanelet og styrer forbindelserne til hhv. den Intelligente Lydmonitor og Vuggesystemet.

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 betjeningspanelet 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 4.6.** Udsnit af datablad for KINGBRIGHT L53 HD, GD og YD

Øverst på figur 4.6 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.

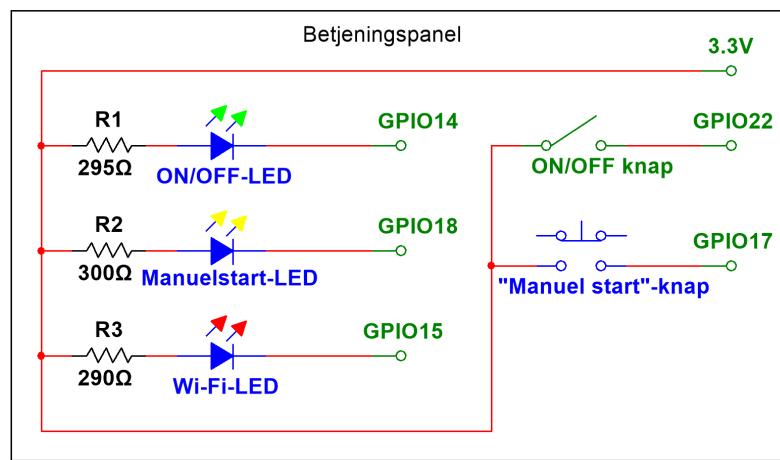
$$R_{L53HD(RED)} = \frac{5V - 2,1V}{10mA} = \frac{2,9V}{10mA} = 290\Omega$$

$$R_{L53GD(GREEN)} = \frac{5V - 2,05V}{10mA} = \frac{2,95V}{10mA} = 295\Omega$$

$$R_{L53YD(YELLOW)} = \frac{5V - 2V}{10mA} = \frac{3V}{10mA} = 300\Omega$$

**Figur 4.7.** Formodstandsberegninger for de 3 dioder

Ud fra modstandsberegningerne i figur 4.7 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 4.8.** Kredsløbsdiagram for Betjeningspanel

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

...fortsat fra forrige side

GPIO	Opsætning	Kommentar
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

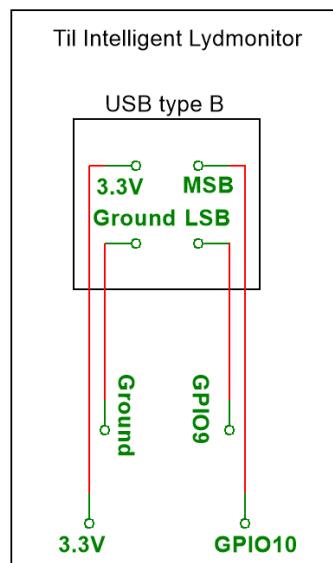
I de to følgende afsnit vil kommunikationen med hhv Intelligent Lydmonitor og Vuggesystem blive beskrevet.

### Kommunikation med Intelligent Lydmonitor

Kommunikationen med den Intelligent 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.

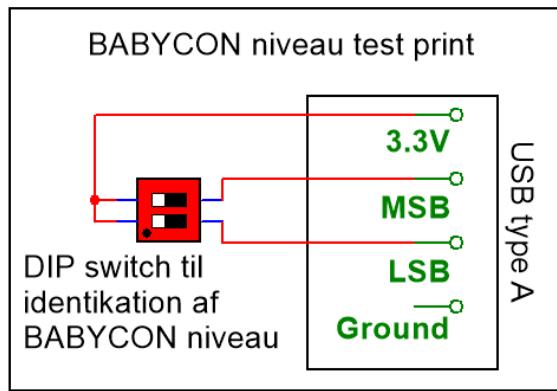
GPIO9 og GPIO10 benyttes til denne to ledet forbindelse til Intelligent Lydmonitor, hvor GPIO9 er LSB delen og GPIO10 er MSB delen.

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 4.9. 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 4.10.



**Figur 4.9.** USB type B for kommunikation med Intelligent Lydmonitor

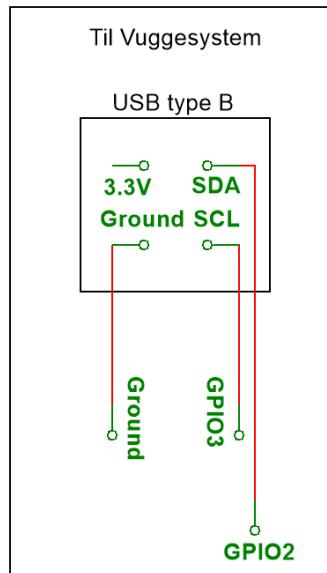
Der designes desuden et testprint se figur 4.10



**Figur 4.10.** Testprint med dipswitch til at angive BABYCON niveau

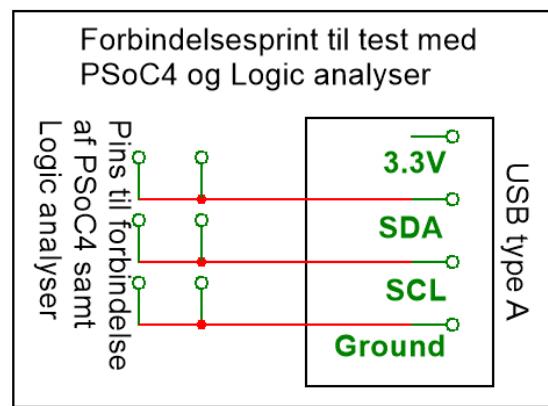
### Kommunikation med Vuggesystem

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 4.11.** USB type B for kommunikation med Vuggesystem

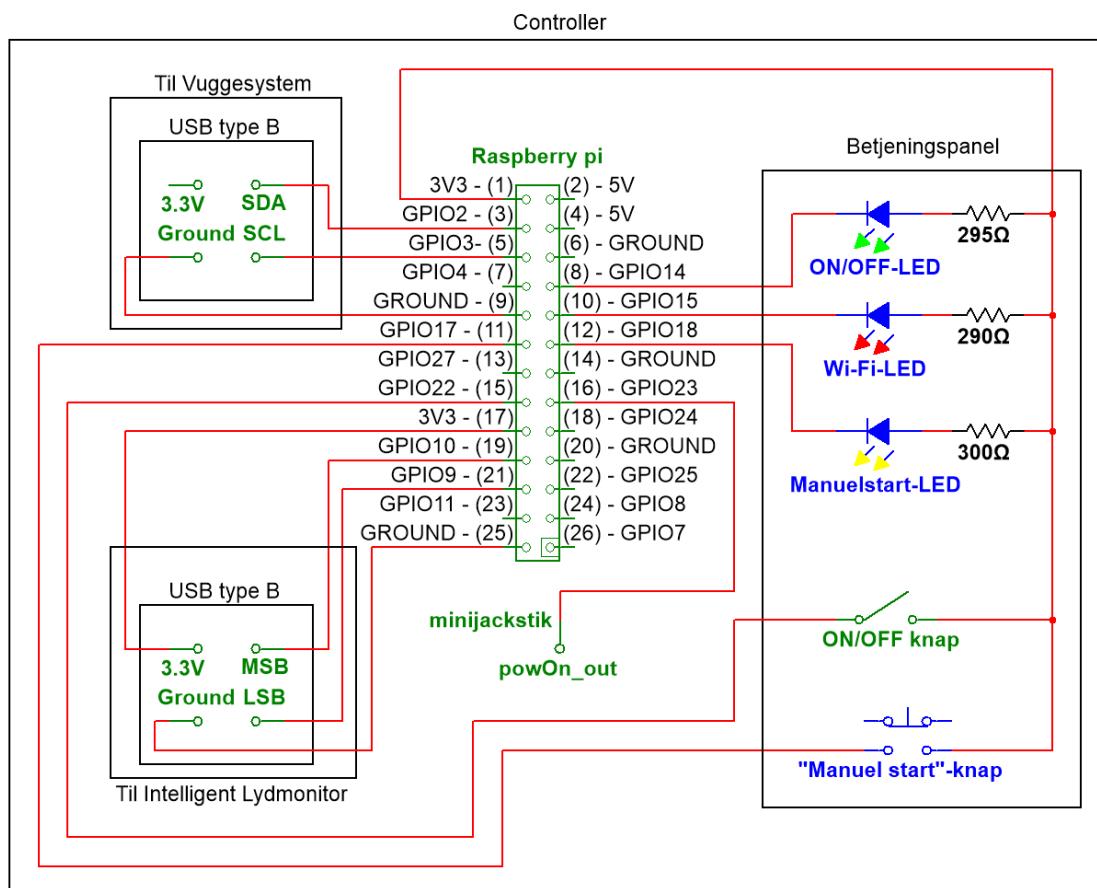
Der designes et testprint således at controlleren kan testes med et simpelt PSoC4 program og datakommunikationen kan verificeres vha. en Logic Analyser.



**Figur 4.12.** Testprint til I2C forbindelsen, med pins til PSoC4 samt Logic Analyser

### Samlet HW-design af Controller

Figur 4.13 viser samlet kredsløbsdiagram for Controlleren. Betjeningspanelets knapper og dioder, USB stik til kommunikation med Intelligent Lydmonitor og Vuggesystem samt Raspberry Pi's GPIO socket.

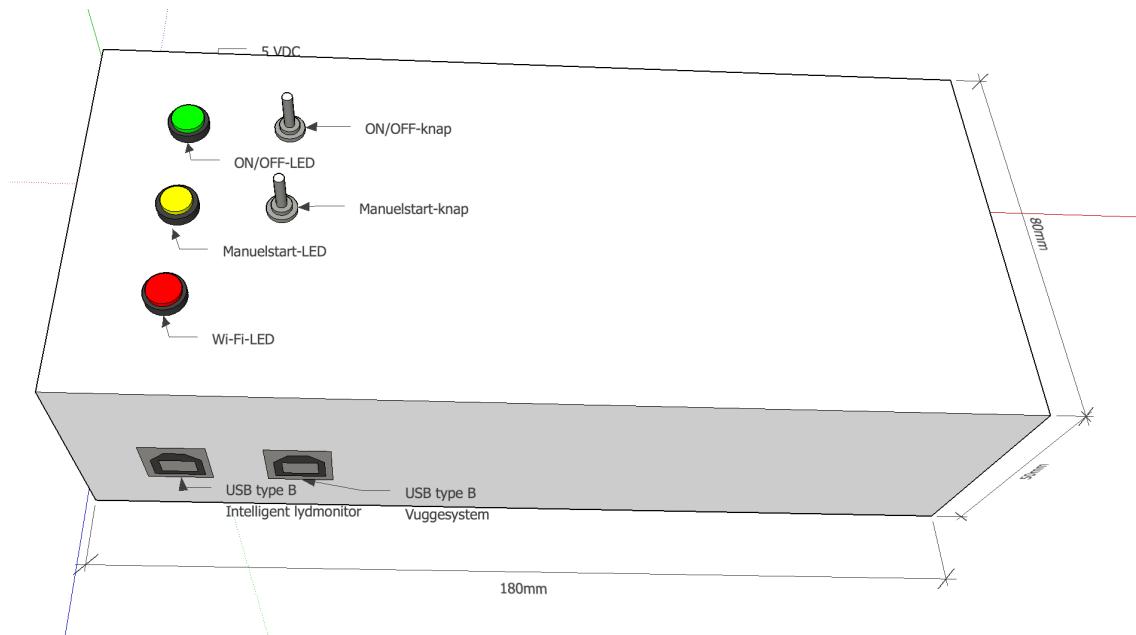


**Figur 4.13.** Samlet kredsløbsdiagram for Controller

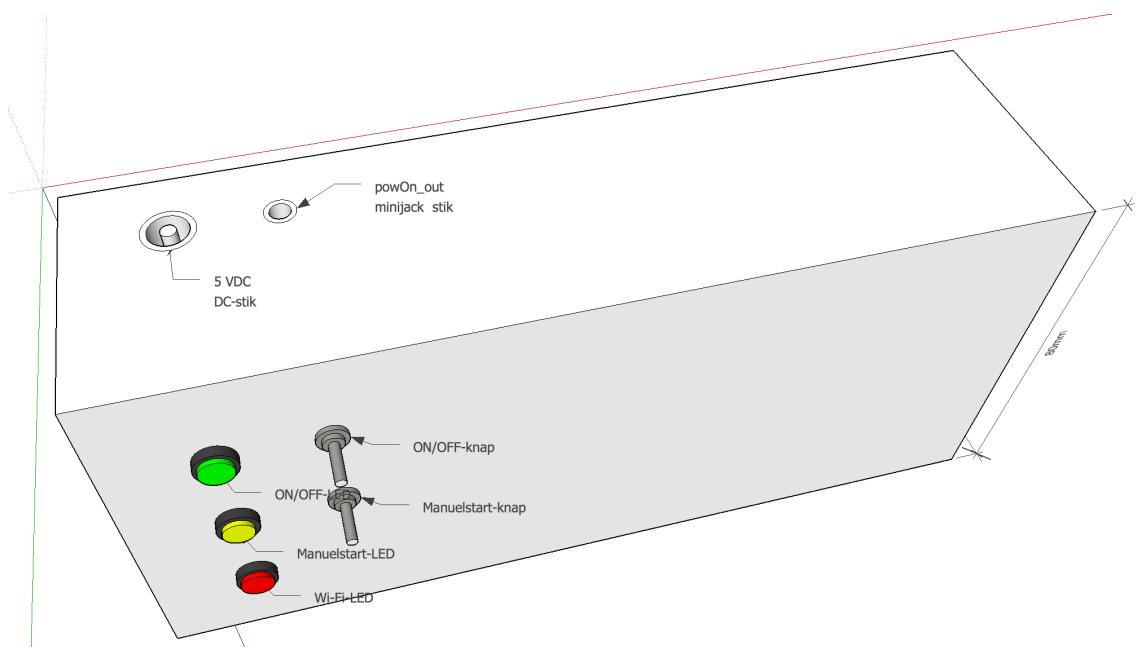
### Fysisk design af Controller

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 4.14 og 4.15 viser kassen, med tilhørende LEDs, kontakter, stik samt mål på selve kassen.



**Figur 4.14.** Arbejdstegning - Betjeningspanel og USB stik



**Figur 4.15.** 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 kunne 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 krydskompile 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"<sup>1</sup> og en "Official Raspbian (armhf) cross compiling toolchain"<sup>2</sup> 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 biblioteket<sup>3</sup> 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 I2Csocet klassen er taget fra Hertaville bloggen<sup>4</sup> der skulle bruges en

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

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

<sup>3</sup><http://wiringpi.com/>

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

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 ...

### 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 dobbeltjek 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 pyhton-templetten og dens funktionaliteter

#### 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 tiende 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** 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" >
![BabyCon]({{ url_for('static', filename=BabyConState) }})

```

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**

```
initWebServer()
stopWebServer()
```

updateToBabyCon1()

updateToBabyCon2()

updateToBabyCon3()

updateFile(int)

#### 4.3.2 Hardwareimplementering

### 4.4 Modultest



# Intelligent lydmonitor

5

## 5.1 Forundersøgelse

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

### Situationer

Der udføres analyser på lyd-optagelser af følgende situationer:

- Højlydt babygråd
- Moderat babygråd
- Fuglefløj
- Trafik
- Katte

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).

### Tonal Power Ratio

Tonal Power Ratio benyttes til beregne tonaliteten i optagelsen, og er altså et udtryk for forholdet mellem 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, sætte den ønskede vindues-længde og shift, samt at benytte hanning-vinduesfunktionen for at minimere ripple i den senere diskrete fourier transformation. Slutteligt 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 hørnæst 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.

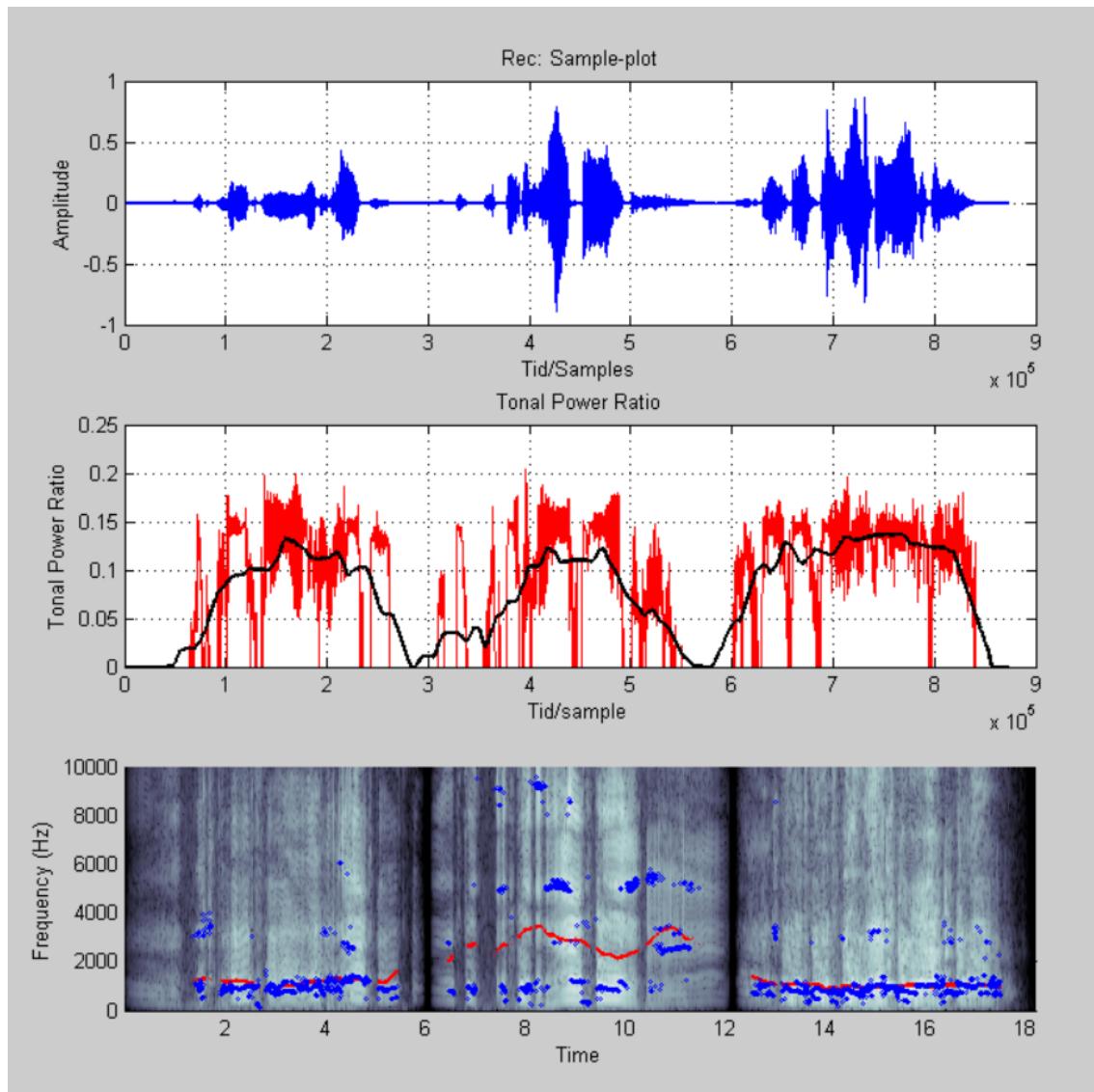
Resultatet af langtids-analysen er:

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 0.13 (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).

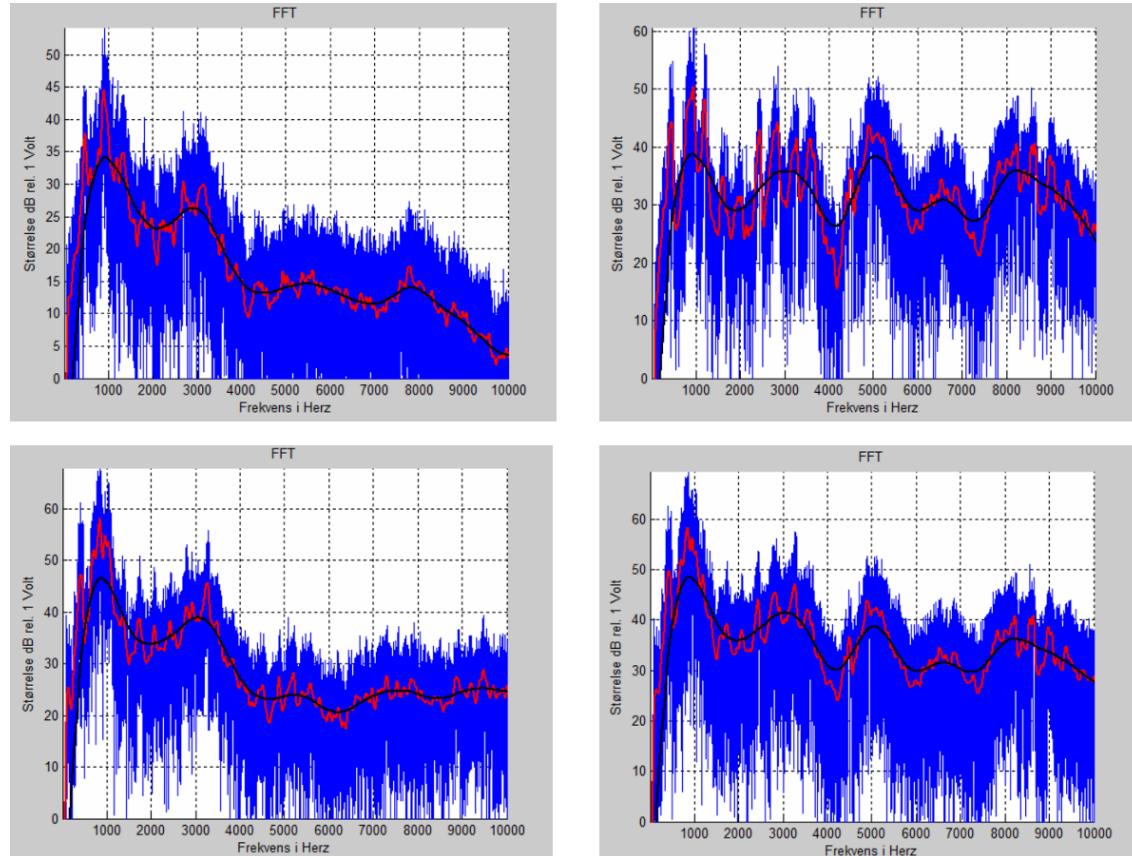


**Figur 5.1.** Korttids-analyse af højlydt babygråd optagelser

### 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 med magnitudeforskel på laveste og højeste frekvens (på sort, midlet kurve) på omrent 25 dB.

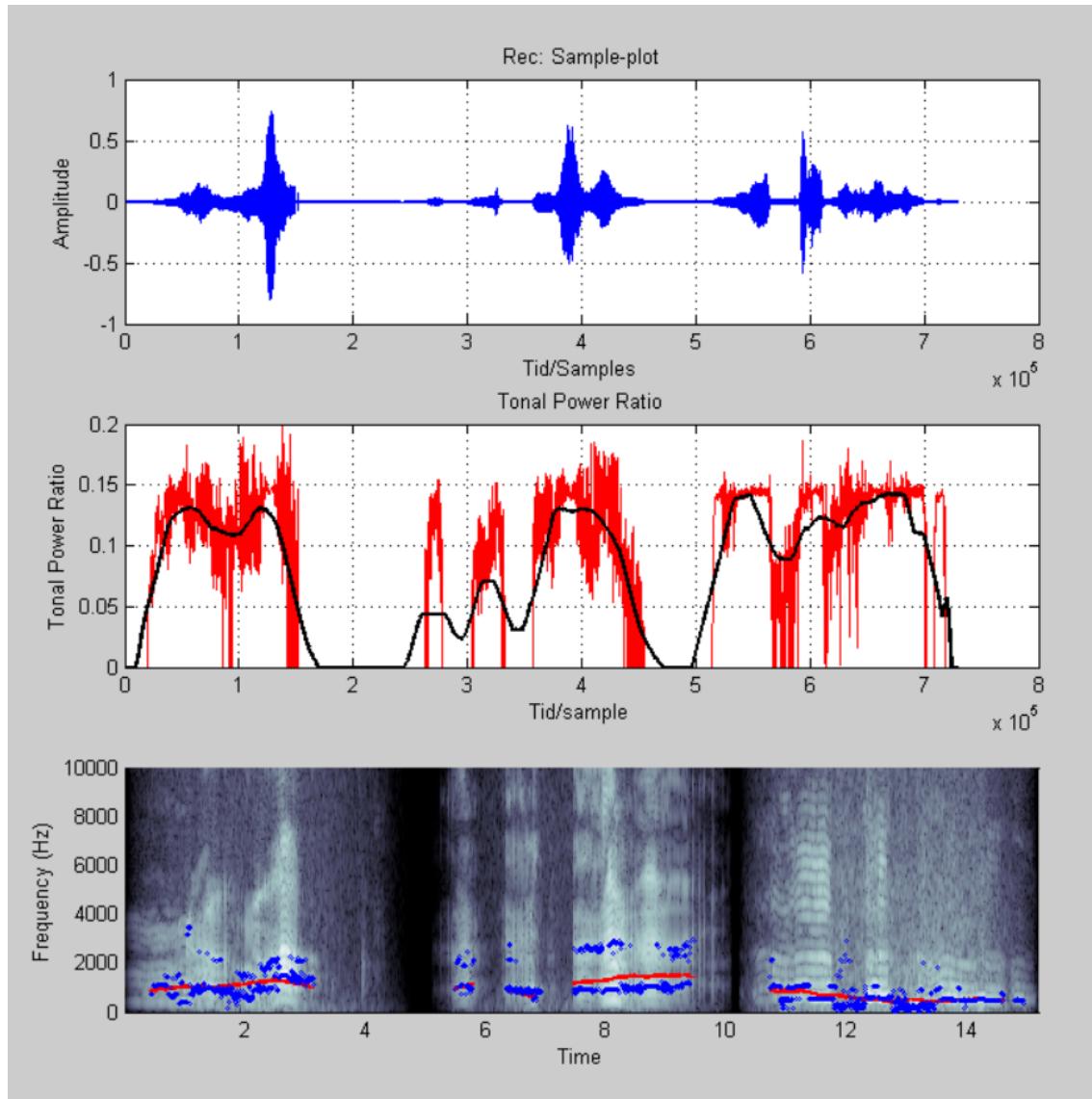
De tre optagelser samlet (nederste højre graf) har en TPR = 0.336.



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

*Moderat babygråd***Korttids-analyse**

Karakteristisk ved optagelsen af moderat babygråd er, at store dele af TPR ligger over en værdi på 0.13 (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 dominante frekvensindhold mellem 900 Hz og 1500 Hz (røde streger).

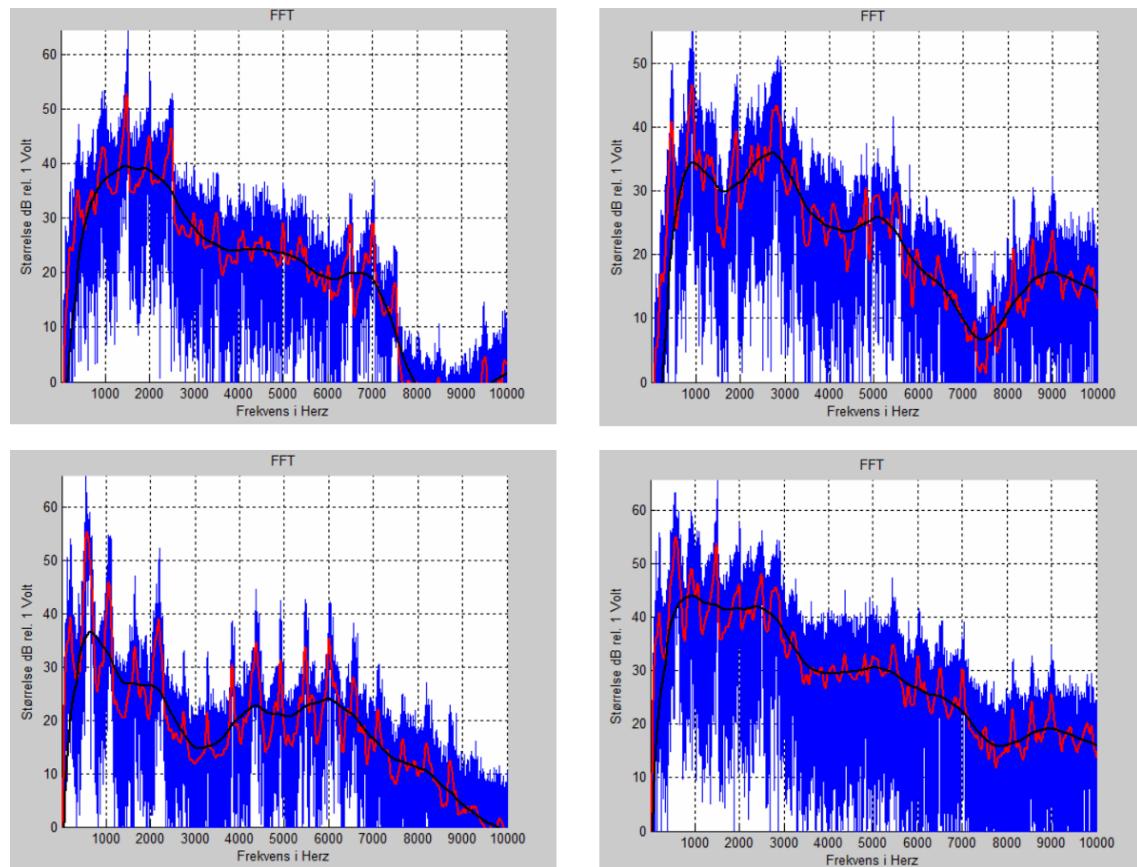


*Figur 5.3.* Korttids-analyse af moderat babygråd optagelser

### 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 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.

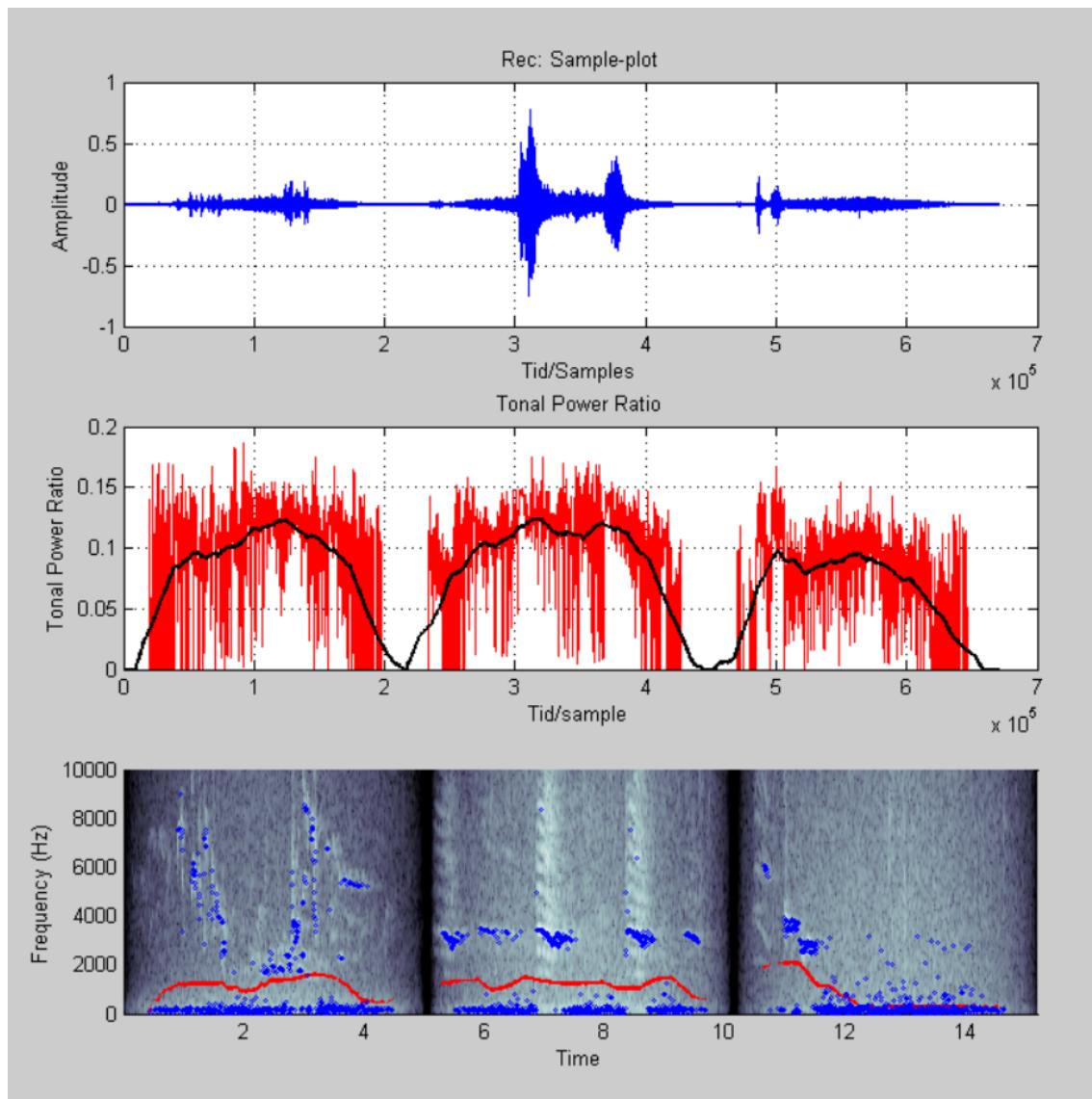


**Figur 5.4.** Langtids-analyser af moderat babygråd optagelser. Nederste venstre graf er analyse af de tre optagelser samlet

### ***Fuglefløjte***

#### **Korttids-analyse**

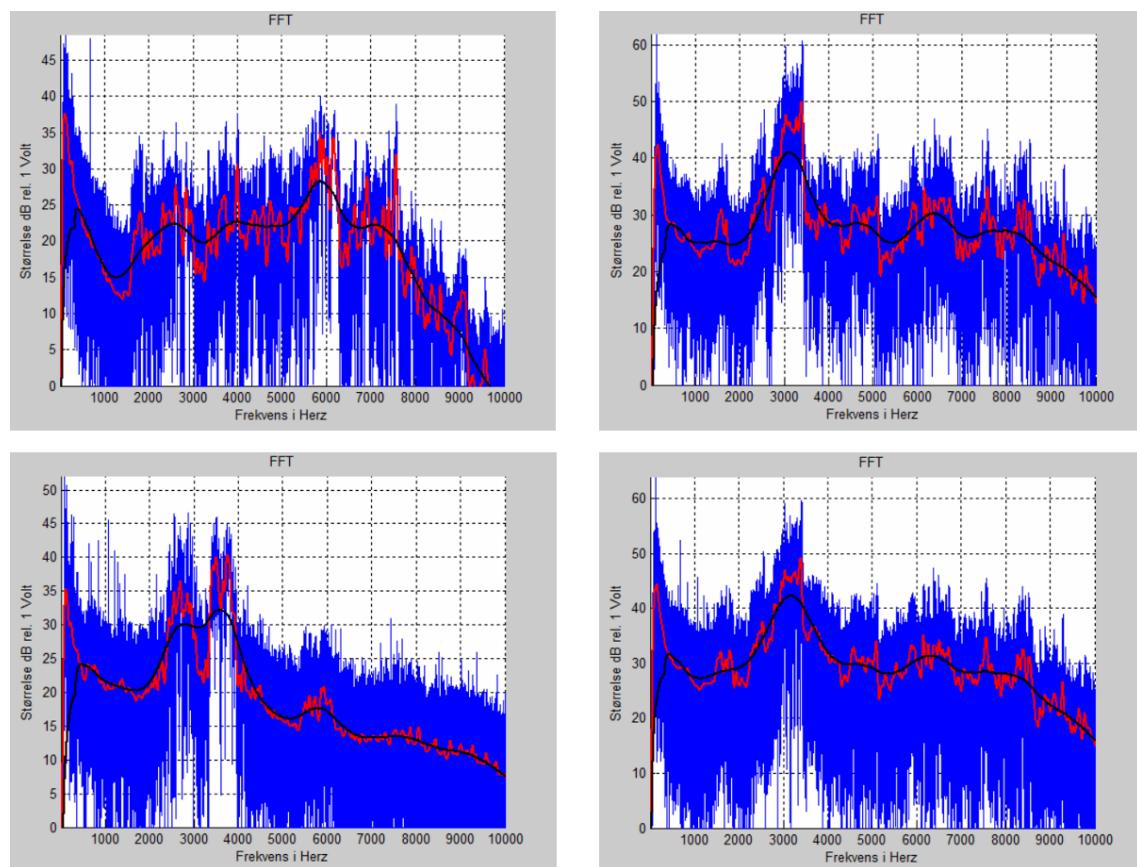
Karakteristik for fuglefløjte og omgivelsesstøj fra natur er, at TPR ligger på en værdi over 0.11, men under 0.13 (plot 2). På spektrogrammet (plot 3) ses det at de dominerende frekvenser ligger meget spredt; det meste af tiden dominerer vind og vejr (ca. 100 Hz), og andre gange dominerer fuglefløjte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



**Figur 5.5.** Korttids-analyse af fuglefløjte optagelser

### Langtids-analyse

For FFT'en af en længere optagelse af fuglefløjts og naturomgivelser, er vind og vejr 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, har en lang indsvingning 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 magnitudeforskel 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.

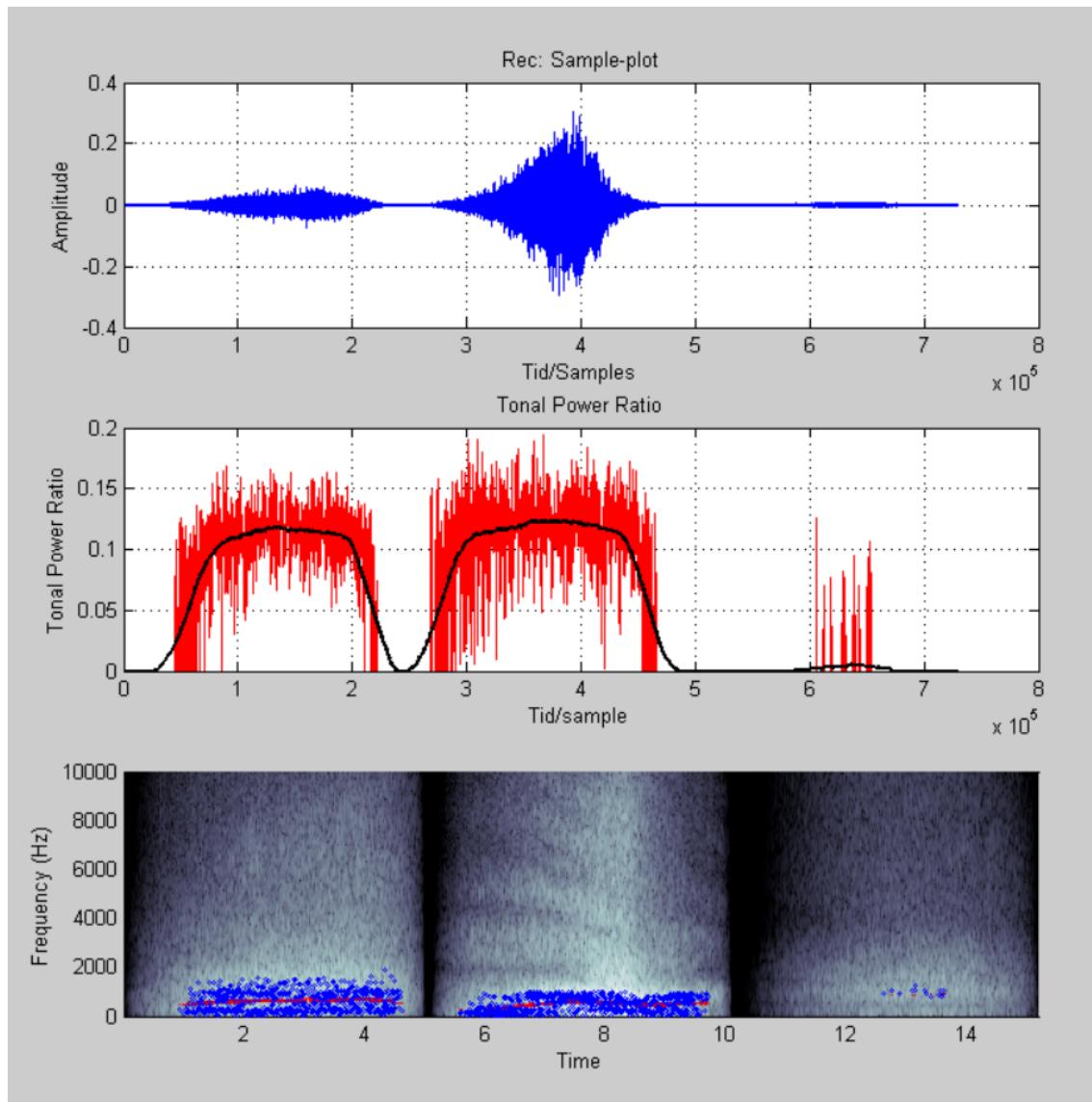


**Figur 5.6.** Langtids-analyser af fuglefløjts optagelser. Nederste venstre graf er analyse af de tre optagelser samlet

### Trafikstøj

#### Korttids-analyse

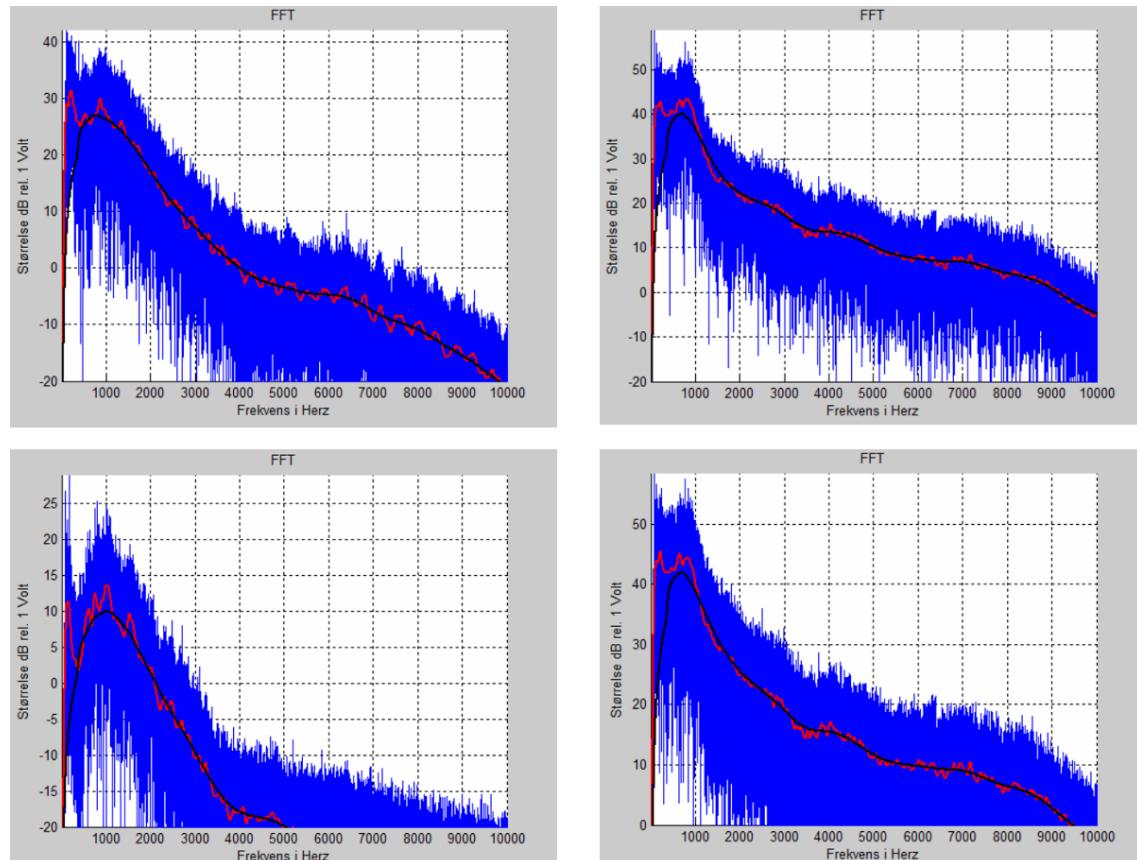
Karakteristisk for trafikstøj er, at TPR ligger over 0.10, men under 0.13 (plot 2). Af spektrogrammet (plot 3) fremgår det at frekvenserne ligger spredt mellem 0 og 1500 Hz med en middel-værdi på ca 700 Hz.



*Figur 5.7.* Korttids-analyse af trafikstøjs-optagelser

### Langtids-analyse

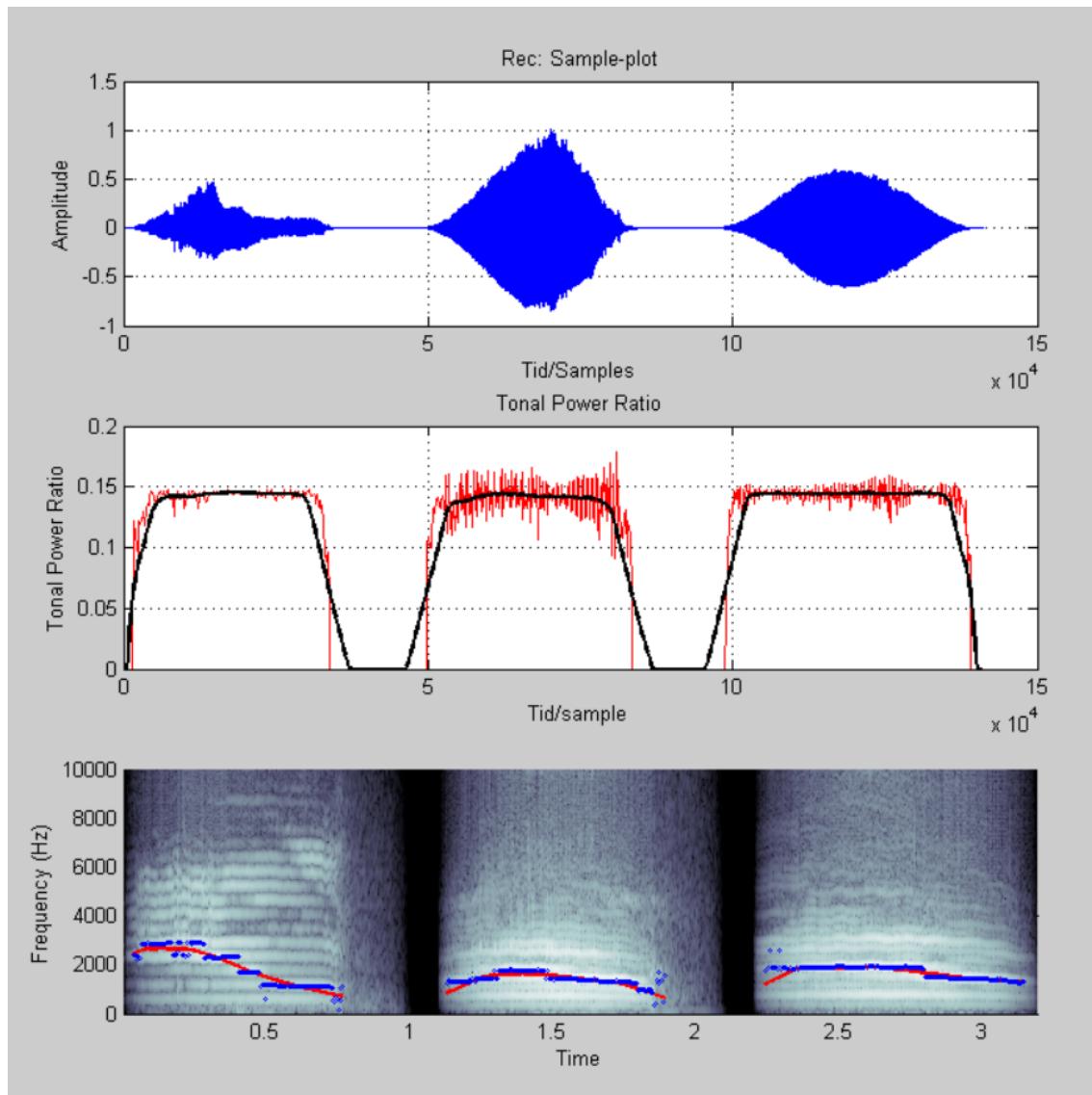
For trafikstøjens frekvensspektrum, er toneindholdet under 1 kHz klart dominerende. Magnituden for frekvenser derover falder jævnt som funktion af frekvensen med en karakteristik, der kunne mindre om lyserød støj med et fald på omkring 40 dB/dek. De tre optagelser samlet (nederste højre graf) har TPR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



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

*Kat***Korttids-analyse**

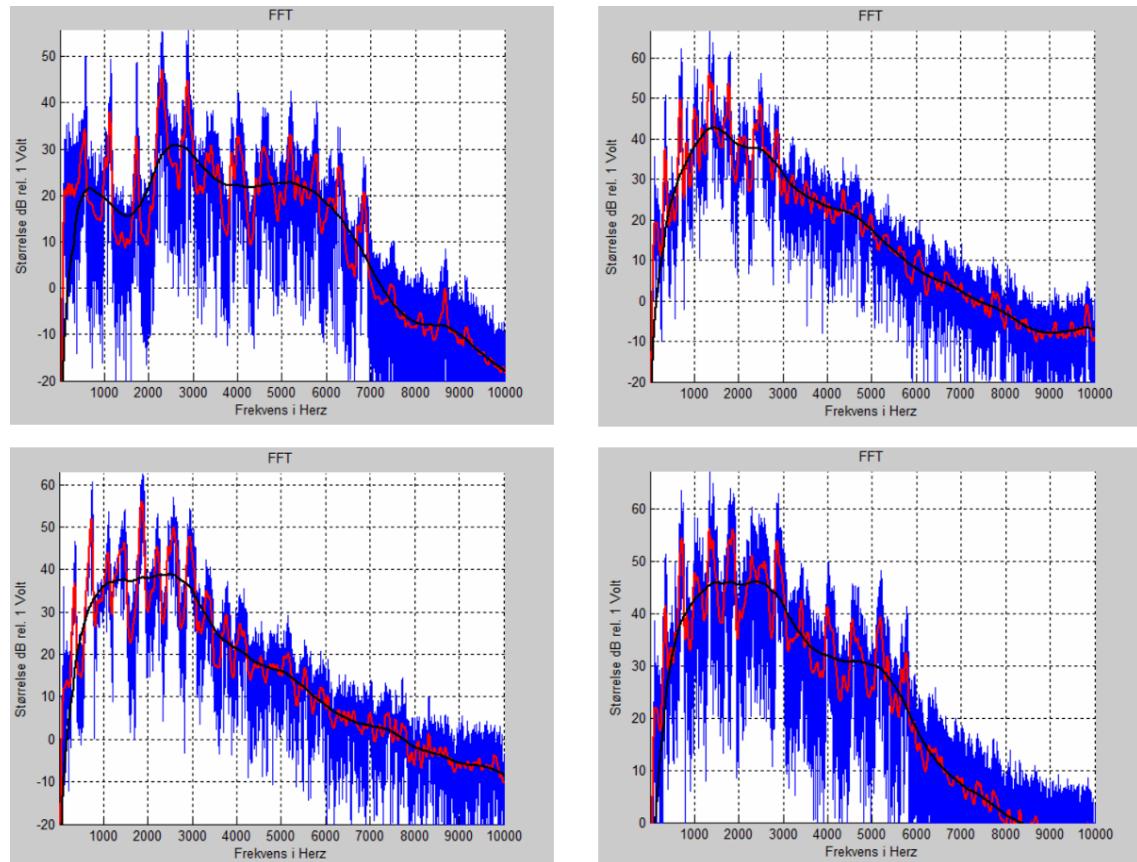
Korttids analysen af katteoptagelserne resulterer i en TPR omkring 0.15 (plot 2). Af spektrogrammet (plot 3) fremgår det at frekvenserne ligger fordelt i bånd på mellem 300 og 500 Hz. Dominerende toner ligger mellem 1,5 og 3 kHz.



*Figur 5.9.* Korttids-analyse af katte-optagelser

### Langtids-analyse

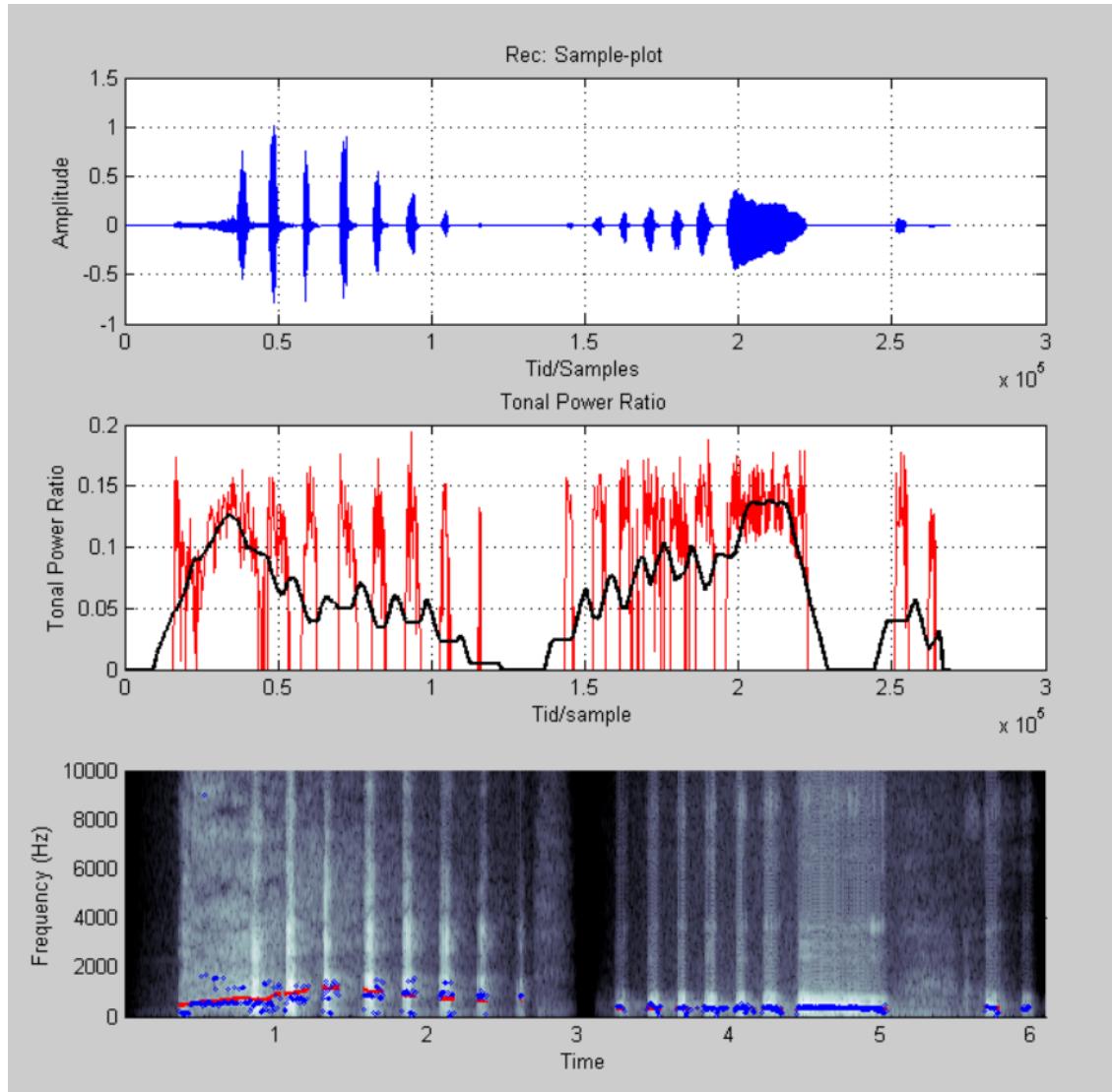
Katte-optagelsen producerer et frekvensspektrum med tydelig grundtone og harmoniske (rød kurve). Midler der, findes peaks omkring 1000 Hz og 2,5 kHz. Generelt falder magnituden jo højere frekvensen bliver. For optagelse 1, falder den drastisk efter 5,5 kHz, men for optagelse 2 og 3 falder den fra 2, 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 venstre graf er analyse af de tre optagelser samlet

*Latter***Korttids-analyse**

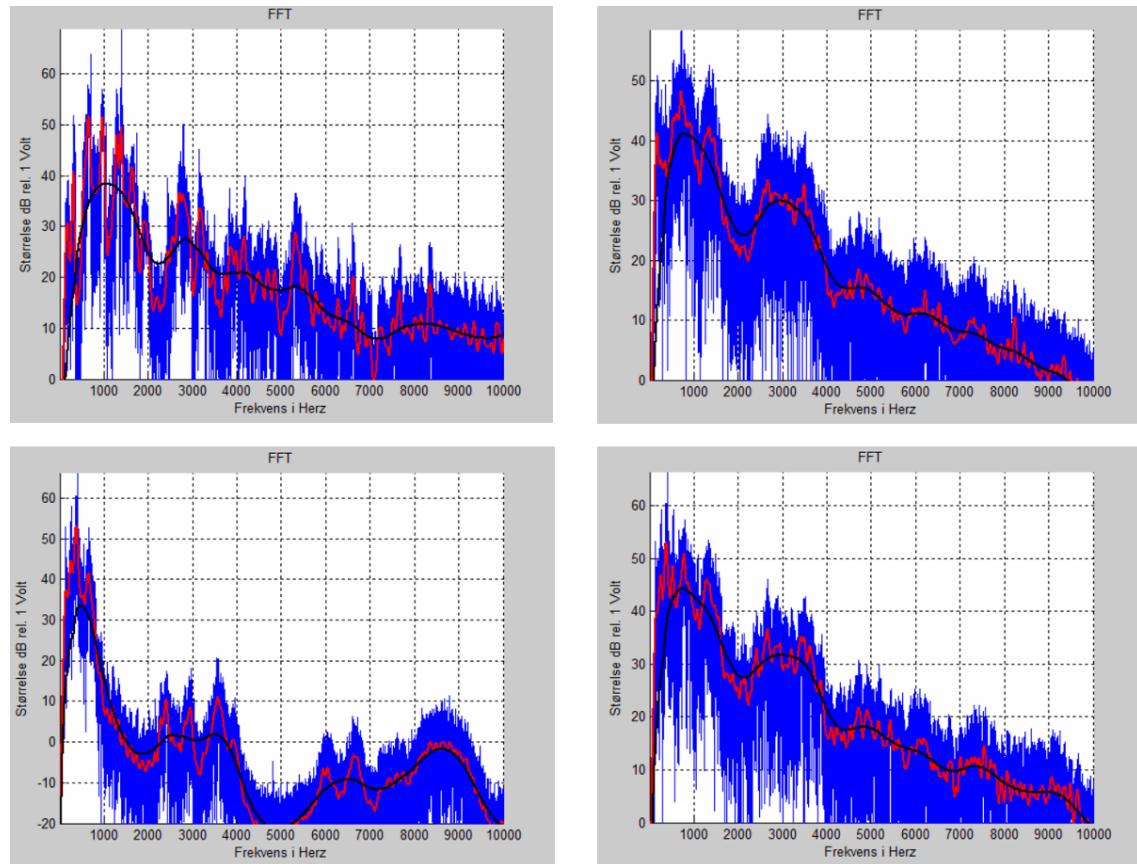
Latter forekommer stødvis med dominerende tone mellem 300 og 1200 Hz. Der produceres et tonalt indhold med TPR omkring 0.15 for latter-stødene (figur 2, rød kurve). Midler TPR, ligger den mellem 0.12 og 0.04 for en enkelt optagelse.



**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 venstre 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
Fuglefløjt	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 fuglefløjt. Katte-optagelse 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. Fuglefløjt 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. Udover 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
Fugleflojt	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 analysemetodik.

#### **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 analyse-metode 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 tilstrækligt 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.

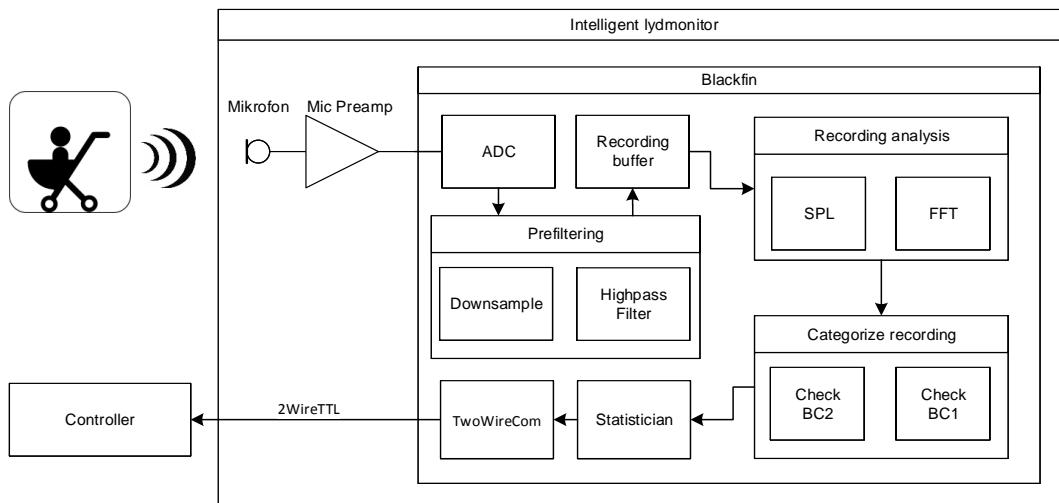
## 5.2 Systemarkitektur

I dette afsnit beskrives systemarkitekturen for den intelligente lydmonitor.

### Overordnet virkemåde

Overordnet skal den intelligente lydmonitor fungere som følger:

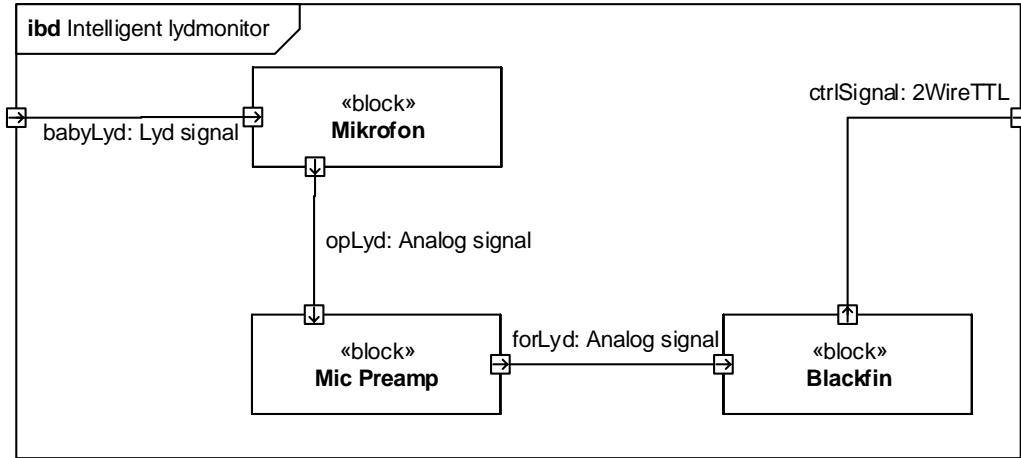
- Babyens gråd detekteres med en mikrofon og forstærkes med en mikrofonforstærker inden det analoge signal konverteres til diskrete samples med Blackfin's ADC
- Den diskrete sample-sekvens forfiltreres. Sample-sekvensen lavpasfiltreres og ned-samples for herefter at blive højpasfiltreret. Det forfiltrerede 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:

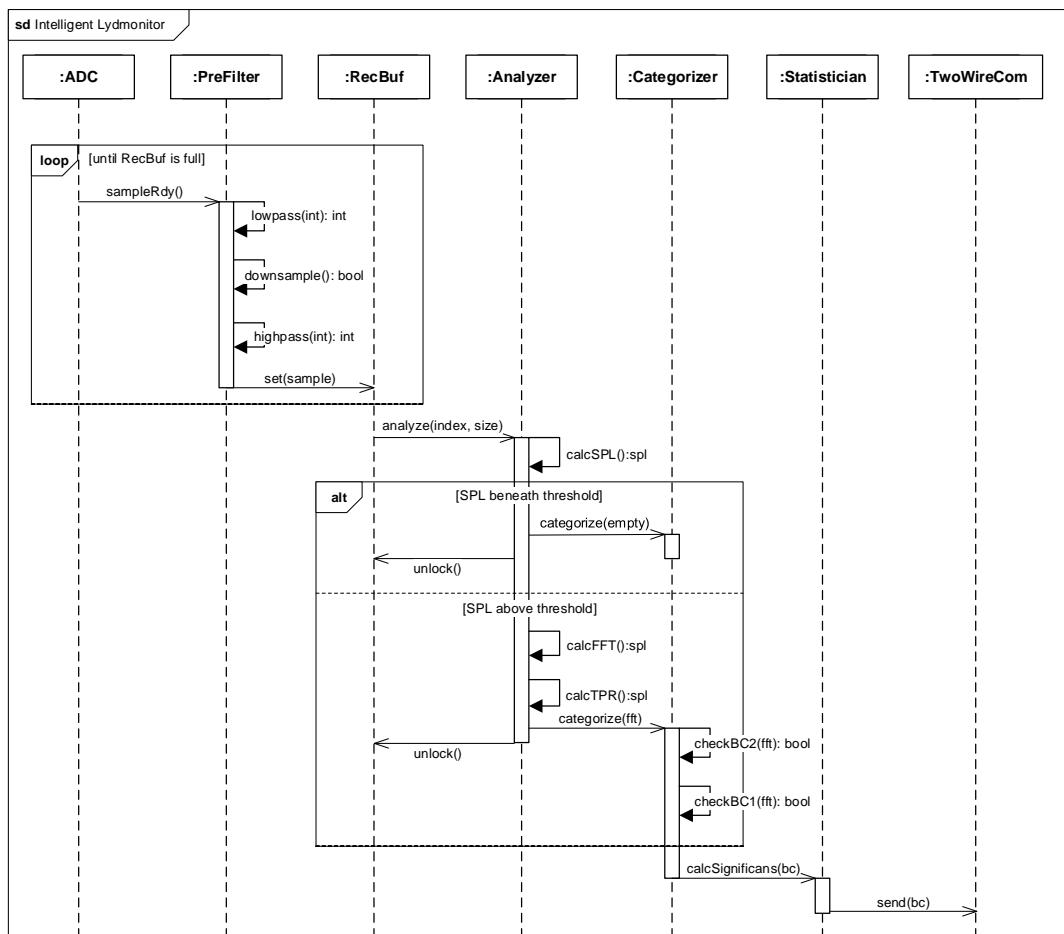
- **Mikrofon** optager signalet babyLyd, som er lyden Baby producerer.
- **Mic Preamp** modtager og forstærker signalet opLyd fra Mikrofon, hvilket er den lyd Mikrofon har optaget.
- **Blackfin** modtager forstærket lyd, forLyd, fra Mic Preamp. Blackfin analyserer lyden og kategoriserer denne, inden 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 funktionsbeskrivelser.

#### Sekvensdiagram

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



**Figur 5.15.** SD for Intelligent Lydmonitor

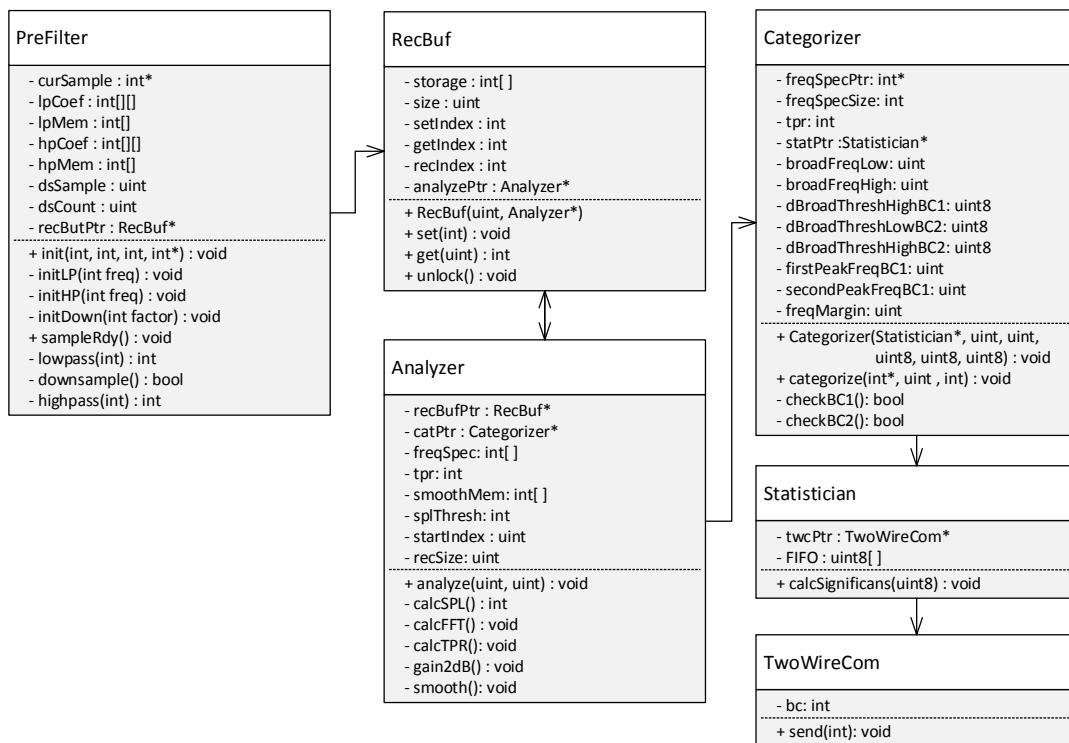
5.15 viser interaktionen mellem Intelligent Lydmonitors subklasser.

Klassen **PreFilter** står for forfiltreringen af sample-sekvensen. PreFilter har tre funktioner, *lowpass()* som lavpasfiltrerer sample-sekvensen, *downsample()* som nedsampler sekvensen og *highpass()* som højpasfiltrerer den nedsampled 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 5.24 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 herved 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.

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

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

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

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

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

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

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

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

*RecBuf*

**Ansvar:** For at kunne opsamle data, mens der bliver behandlet andet data, er det valgt at benytte en cirkulær buffer med mulighed for midlertidig locking af bufferområder. Det er altså en buffer der muliggør påfyldning ét sted mens aflæsning foregår fra et låst segment.

<b>Funktion</b>	<code>RecBuf(uint size, Analyzer* analyzePtr)</code>
<b>Parametre</b>	uint size, størrelsen af cirkulær buffer; analyzePtr, pointer til Analyzer
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Constructor sætter størrelsen på cirkulær buffer, samt initierer set-, get- og rec-Index

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

<b>Funktion</b>	<code>int get(uint index)</code>
<b>Parametre</b>	uint index, plads i RecBuf
<b>Returværdi</b>	int, returnerer sampleværdi
<b>Beskrivelse</b>	Returnerer værdi på given index plads

<b>Funktion</b>	<code>void unlock(void)</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Frigiver låst område i RecBuf

*Analyzer*

**Ansvar:** Klassen Analyzer indeholder analyse-funktionerne.

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

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

<b>Funktion</b>	<code>void calcFFT(void)</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	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

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

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

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

### *Categorizer*

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

<b>Funktion</b>	<code>Categorizer(Statistician* statPtr,           uint broadFreqLow, uint broadFreqHigh,           uint dBroadThreshHighBC1,           uint dBroadThreshLowBC2, uint dBroadThreshHighBC2,           uint firstPeakFreqBC1, uint secondPeakFreqBC1,           uint freqMargin)</code>
<b>Parametre</b>	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
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Constructor sætter relevante kategoriseringparametere

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

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

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

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

*Statistician*

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

<b>Funktion</b>	<code>void calcSignificans(uint bc)</code>
<b>Parametre</b>	uint bc, BABYCON-niveau
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	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. 3.3.1

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

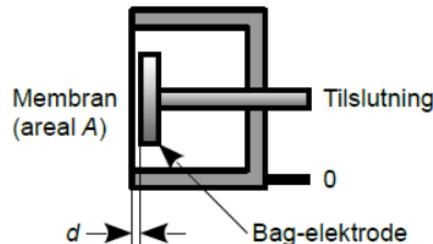
#### 5.3.1 Mikrofon preamp

##### Mikrofon

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

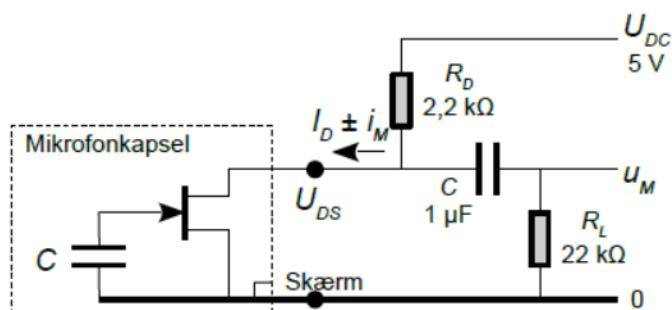
- Frequency range: 50 Hz to 10 kHz
- Sensitivity: 5,6 mV / Pa / 1 kHz
- Output impedance: 2 kOhm
- Power Supply: 1.5 to 10 V dc, 5 mA

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

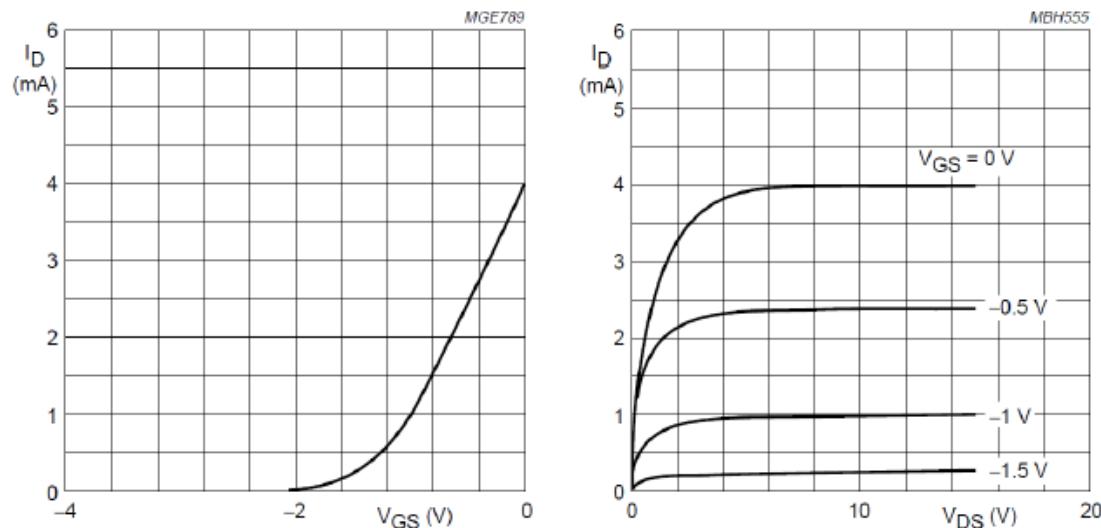
Dog har kablingen mellem mikrofonen og forstærkeren en betydelig kapacitet, så mikrofonpakken er implementeret med en indbygget JFET som buffer, som det ses af Figur 5.18



**Figur 5.18.** Elektret forsnyningskredsløb. Kilde: Analogteknik

Forsyningen til mikrofonen trækkes fra Blackfin, og er således  $U_{cc} = 5$  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 1 V over  $R_d$  og dermed 4 V over den i mikrofonkapslen indbyggede JFET. I dette område af  $U_{DS}$  vil transistoren have en strømbegrænsende virkning og kan derfor bruges som en tilnærmedesvis lineær strømkilde.



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

På Figur 5.19 ses karakteristikken for JFET. I 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)

Spændingen i dette punkt,  $u_M$ , er givet ved parallelværdien mellem de to modstande og  $i_M$ .

$$u_M = (R_D || R_L) \cdot i_M$$

### PreAmp

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

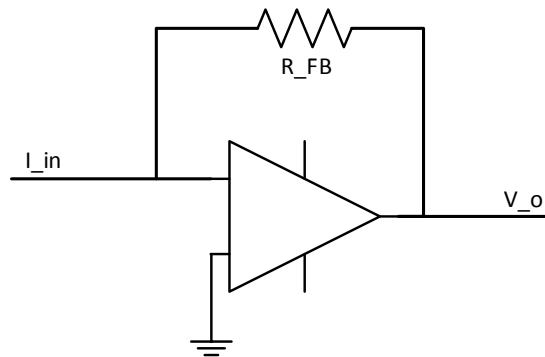
Blackfin ADC'en tager et line-level input på +/- 1,65 V. Det ønskes at udnytte det maksimale dynamiske område uden at lade signalet klippe. Det regnes med at mikrofonen ikke udsættes for mere end 2 Pa ved almindelig brug. Den maksimale strøm-amplitude bliver derfor:

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

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.



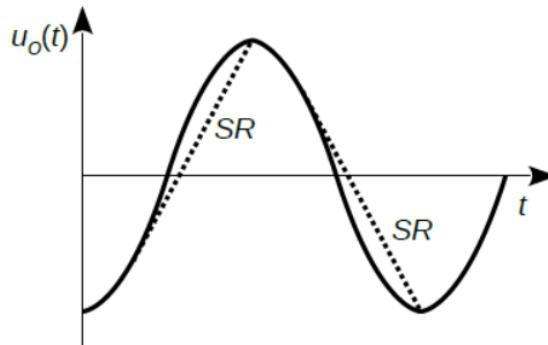
**Figur 5.20.** Transimpedansforstærker:  $V_o = R_{FB} \cdot I_{in}$

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. 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  $u_M < 20\text{mV}$ .

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

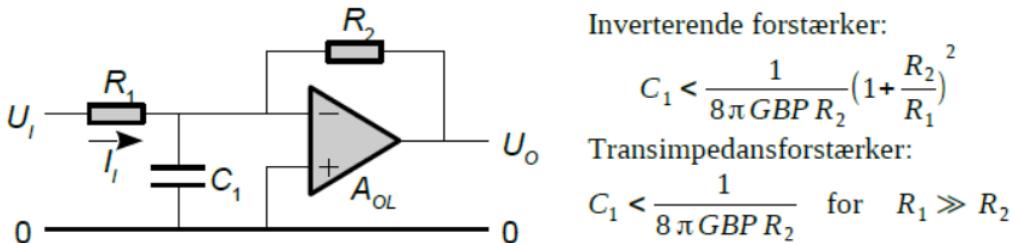
Dette krav er altså opfyldt.

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.

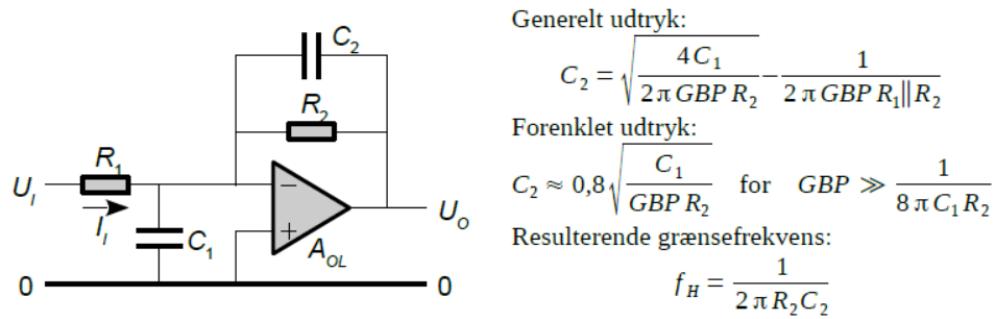


**Figur 5.22.** OpAmp med kapacitiv belastning på indgangen. Kilde: Analogteknik

På figur 5.22 ses TIA'en som de næste udregninger tager udgangspunkt i. Mikrofonen er kobles med ca. 1m coax kabel med kapaciteten 100pF/m. Dette giver altså en  $C_1$  på 100 pF. Der regnes med en typisk GBP på 9MHz. Der testes for nødvendighed af kondensator i tilbagekoblingen:

$$C_1 < \frac{1}{8\pi \cdot GBP \cdot R_2} \implies 100\text{pF} < \frac{1}{8\pi \cdot 9\text{MHz} \cdot 28,9k\Omega} \implies 100 \cdot 10^{-12} < 153 \cdot 10^{-15}$$

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:

$$C_2 \approx 0,8 \sqrt{\frac{C_1}{GBP \cdot R_2}} \implies C_2 = 0,8 \sqrt{\frac{100pF}{9MHz \cdot 28,9k\Omega}} = 15,7pF$$

Der benyttes en 18 pF kondensator. Den *resulterende grænsefrekvens* bliver således:

$$f_H = \frac{1}{2\pi \cdot R_2 \cdot C_2} \implies f_H = \frac{1}{2\pi \cdot 28,9k\Omega \cdot 18pF} = 305,95kHz$$

Dette begrænser altså ikke signalets båndbredde.

## 5.4 SW design og implementering

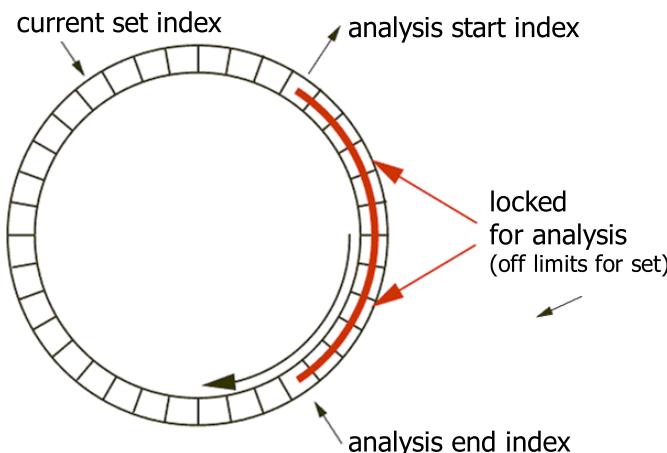
Design og implementering af SW-moduler er beskrevet i dette afsnit.

### 5.4.1 Prefilter

### 5.4.2 RecBuf

I dette afsnit beskrives principper og design for RecBuf i Intelligent Lydmonitor. Samples modtaget fra prefILTERET gemmes i RecBuf. Når der ligger en samplesekvens svarende til 5 sekunders samples i RecBuf, skal denne kaldes Analyzers analyze() med startindexet for denne 5 sekunders optagelse. Idet den kaldes, skal samplesekvensen låses og dermed skærmes mod overskrivning. Imidlertid fortsættes samplingen og et nyt start-index sættes. Når analysen afsluttes notificeres RecBuf og låser sekvensen op. Det opstår således et hastigheds-krav til analyze og et størrelseskrav til RecBuf, idet analyze skal afslutte inden RecBuf når den låste sekvens igen.

Herunder ses en princip-skitse for en cirkulær buffer med låsning af segmenter.



**Figur 5.24.** Princip for cirkulær buffer med segment-locking

Dette afsnit er endnu ikke færdigt

### 5.4.3 Analyzer

### 5.4.4 Categorizer

### 5.4.5 Statistician

### 5.4.6 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

### 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

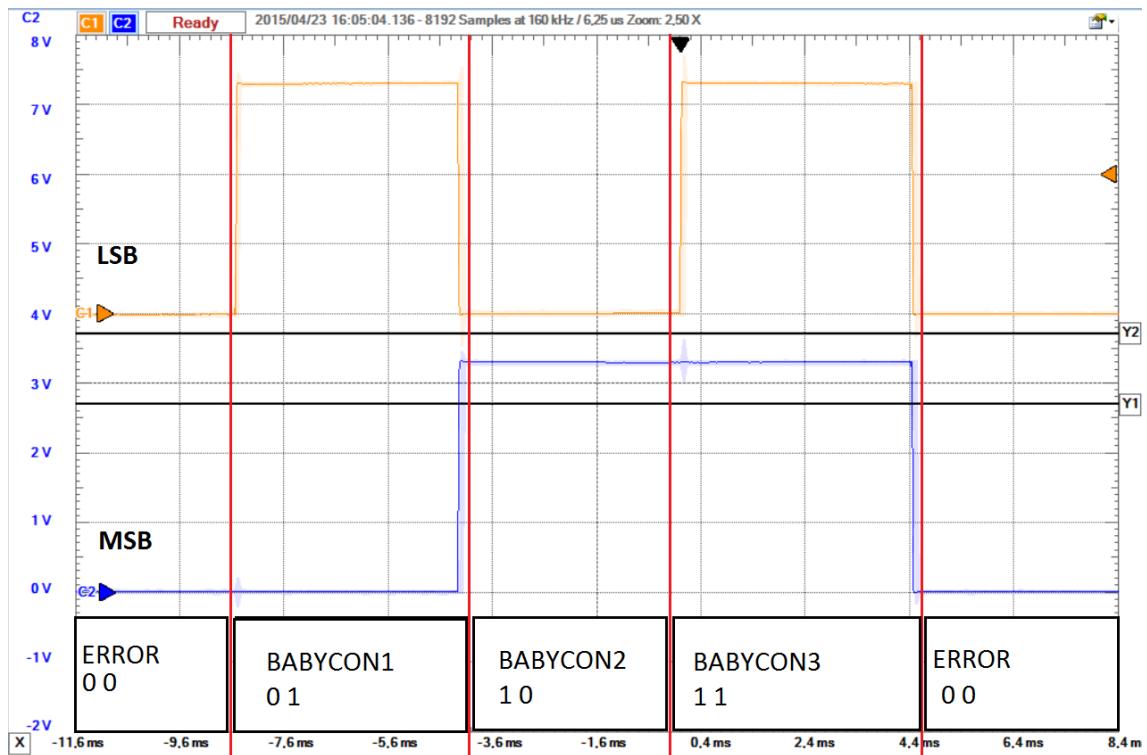
1. Projektet med TwoWireCom testen eksekveres
2. Oscilloskoperne optages og billedet gemmes

#### Forventet resultat

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

#### Resultat

Som det ses af Figur 5.25 forekommer alle fire kombinationer.



Figur 5.25. Oscilloskop billede af test

Testen er godkendt



# Vuggesystem 6

## 6.1 Systemarkitektur

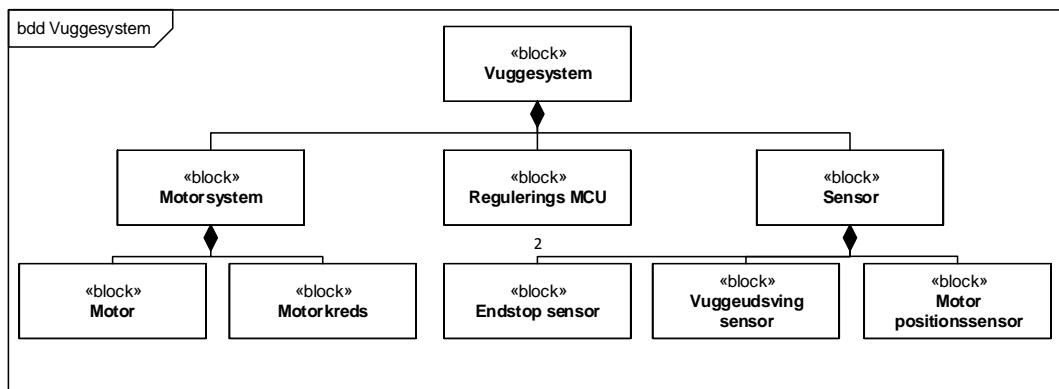
I dette afsnit beskrives systemarkitekturen for vuggesystemet.

### 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

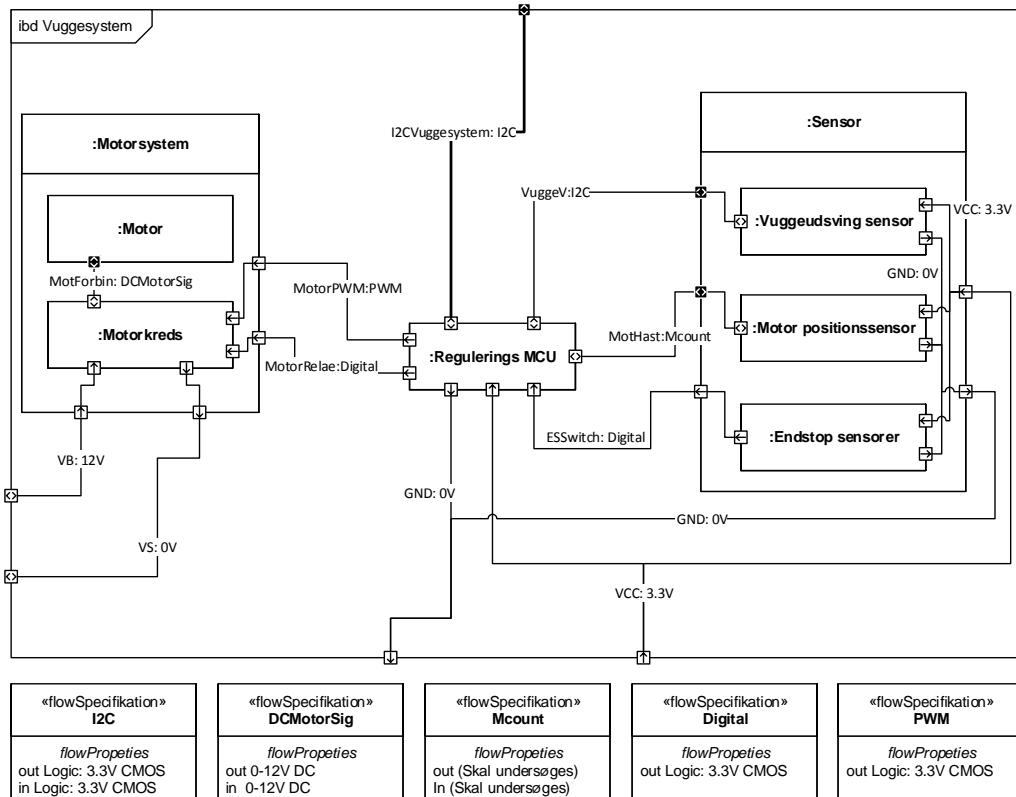
Vuggesystemet består af tre dele

**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.

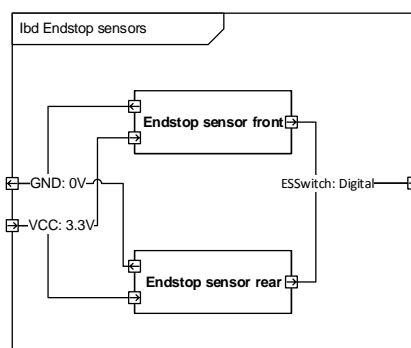
**Regulerings MCU block:** Styringsenheden for vuggesystemet. Denne sørger for reguleringen af vuggesystemet samt kommunikationen til og fra controlleren.

**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.

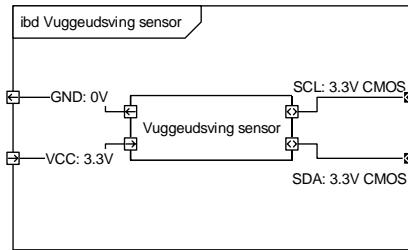


**Figur 6.2.** IBD for Vuggesystem



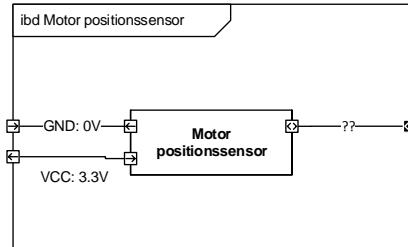
**Figur 6.3.** IBD for Endstop sensorer

**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 barnevognskurvens plan i forhold til jordens tyngdefelt.



*Figur 6.5.* IBD for Motor positionssensor

**Motor positionssensor** giver et frekvensignal 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.

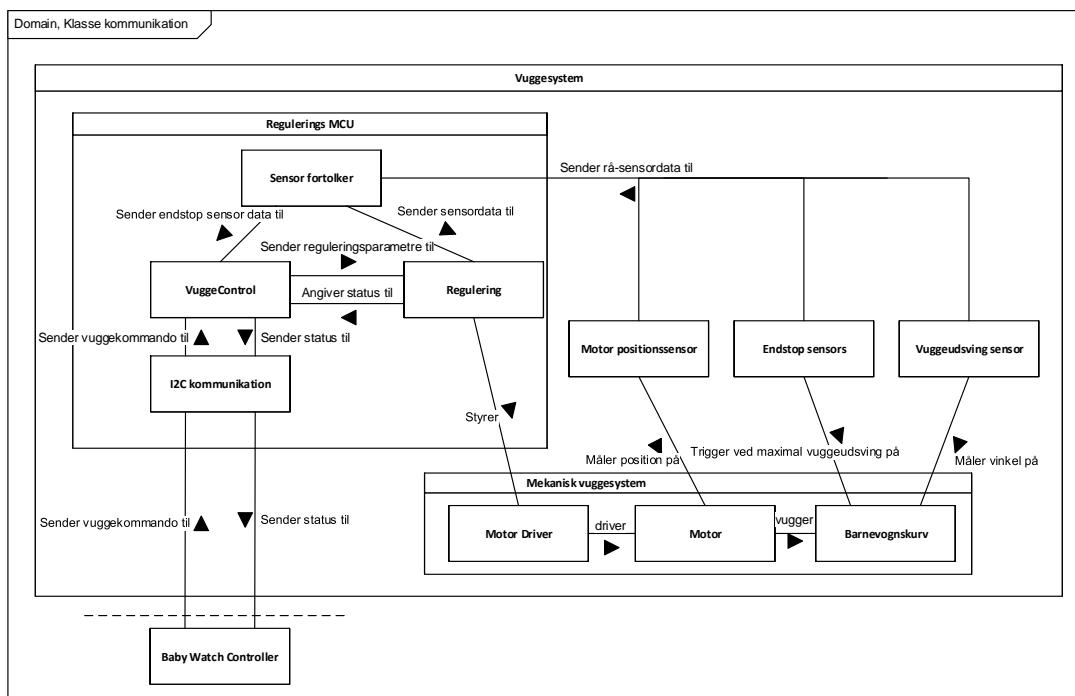
Softwarearkitekturen for Vuggesystemet består af:

- 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.

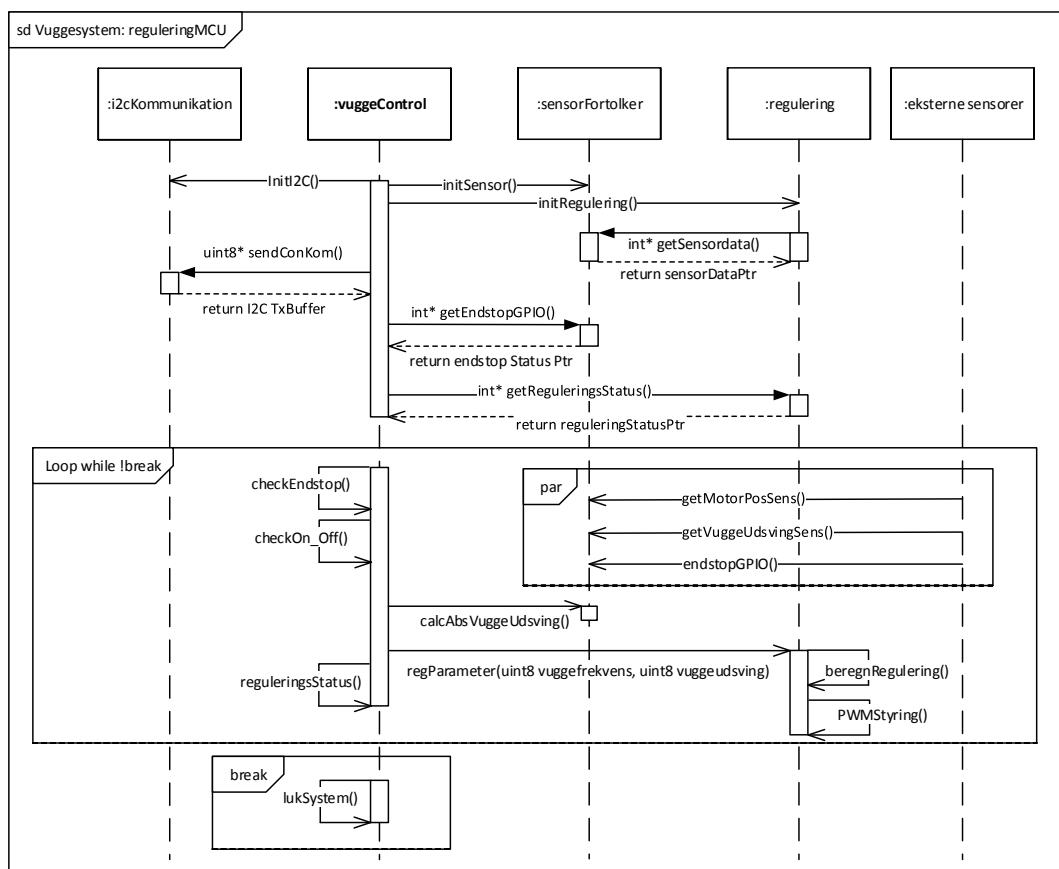
## Klasseidentifikation

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

Beskrivelse af system:

Vuggesystemet 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. Vuggesystemet 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.

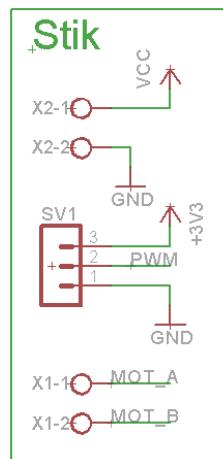
- I2C kommunikation - klassenavn: **i2cKommunikation** står for I2C interfacet ud til Baby Watchs Contoller.
- Motorregulering - klassenavn: **regulering** står for beregningerne til reguleringen af motoren der trækker vugningen af barnevognenskurv
- Sensor Fortolker - klassenavn: **sensorFortolker**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

## 6.2 Design

### 6.2.1 Hardware design

#### Motorstyringskreds

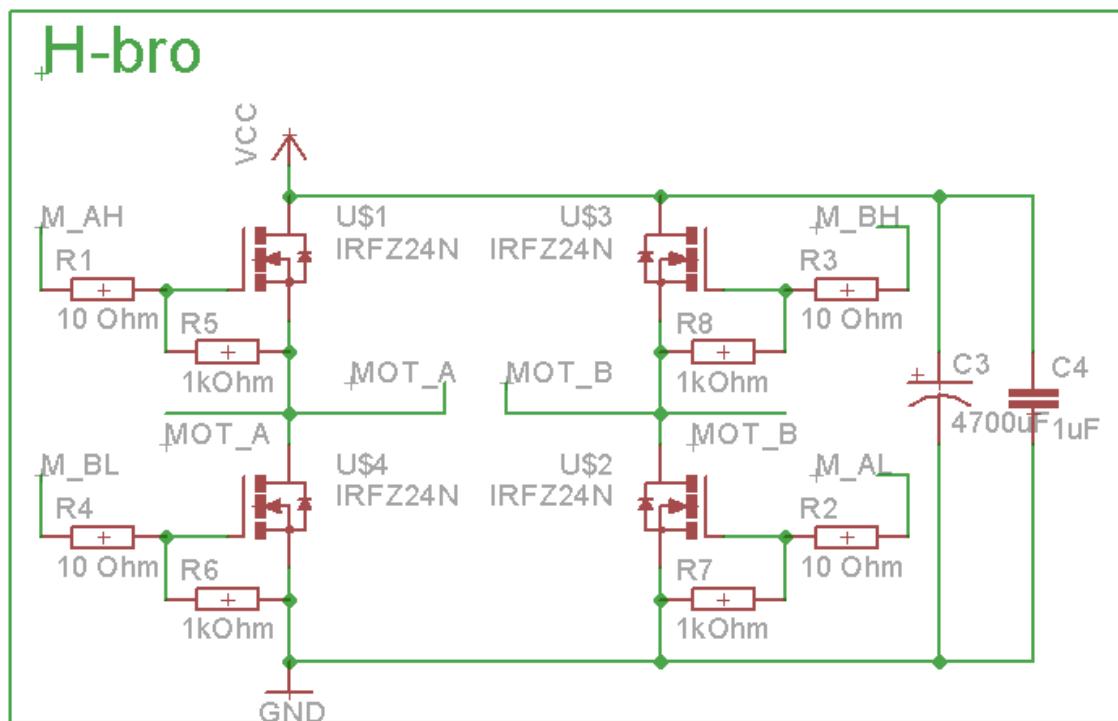
I denne sektion beskrives det kredsløb der benyttes til at styre ankerspændingen for vuggesystemets motor. Kredsløbet modtager et PWM signal fra controlleren som styrer ankerspændingen. Udenfor styresignalet og motor terminalerne, er kredsløbet koblet til systemets 12V forsyning(VCC i diagram) hvorfra motoren forsynes, og systemets 3.3V forsyning som leverer strøm til de logiske kredsløb.



*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

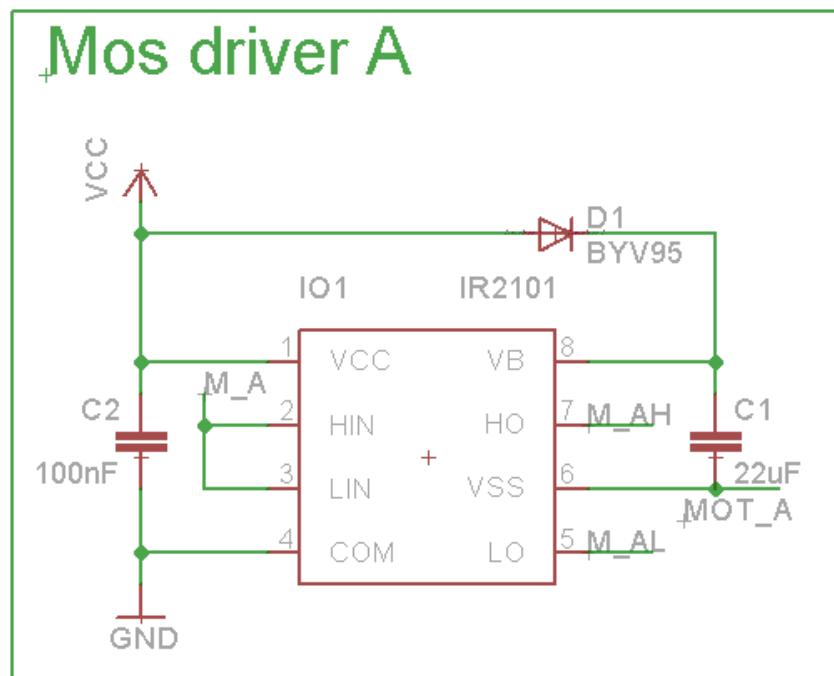
De fire N-channel er koblet med en lille modstand foran gaten, og en pull-down modstand fra gaten til source. Den lille modstand sikrer at der ikke opstår or meget ring som resultat af seriekobling af ledningens induktans og gatens kapacitet, og pull-down modstanden sikrer at gaten bliver trukket lav 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 (LAV ORDENTLIG REF).

### MOSFET driver

H-broens MOSFETs bliver drevet ved hjælp af 2 dobbelt sidede 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.

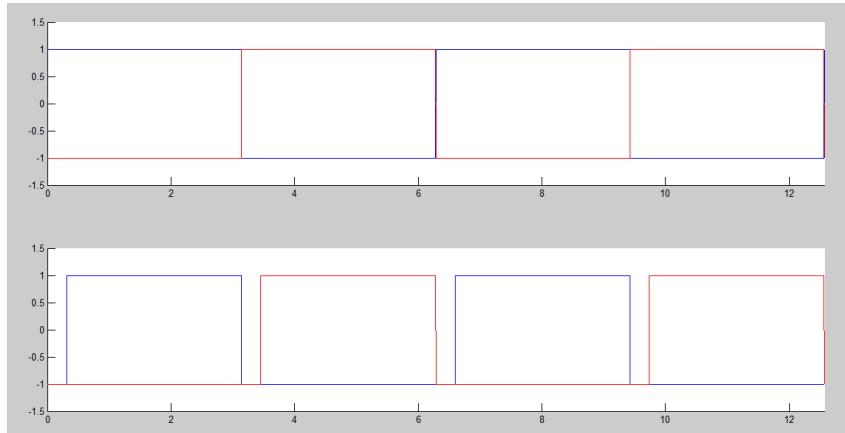
Driverens input er et digitalt input, med en fast overgang fra lav til høj mellem 0.8 og 3 V, og kan således styres med 3.3V logik uafhængigt af hvilken forsyningsspænding driveren er koblet med.

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 tænd 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.

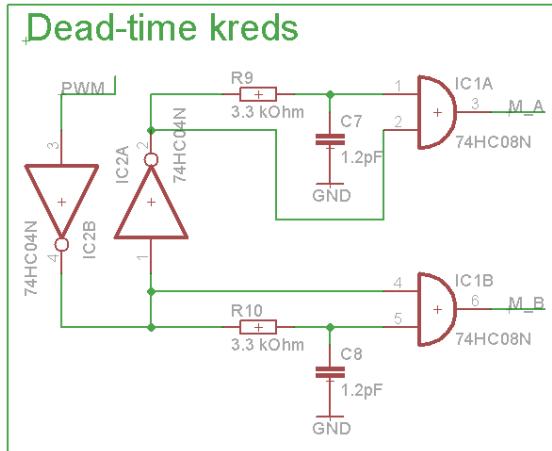
Da MOSFET transistorne tager en hvis tid om at slukke, skal der introduceres en dødtid i styresignalene, således at signalerne alle går kortvarigt til 0 før de skifter. Dette sikrer at to transistorer fra hver deres side af broen ikke er tændt samtidig, hvilket ville resultere i en kortslutning fra VCC til GND gennem de to transistorer.



**Figur 6.11.** Deadband eksempel

Dødbåndet skabes ved at introducere en forsinkelse på den stigende kant af både signalet og det inverterede signal, som det ses på figuren herover. Forsinkelsens størrelse skal være tilpasset, således at den har samme længde som den tid det tager transistorene i broen at slukke helt. På denne måde sikres det at transistorene ikke er tændt samtidig, men også at der spildes mindst muligt af PWM perioden på skift. På ovenstående eksempel ses det også at duty cyclen ikke ændres, idet de to signaler stadig er tændt i en lige stor del af perioden.

Denne forsinkelse på den stigende kant introduceres vha et RC led og en and gate, som vist nedenfor.



**Figur 6.12.** Deadband kredsløb

Den første inverter fungerer som en buffer for at sikre at indgangssignalet møder en høj impedans, mens den anden inverter skaber et ikke inverteret udtag til PWM signalet. Modstanden er dimensioneret således at der højst trækkes en milliampere fra inverteren, og kondensatoren kan så dimensioneres til at give en passende tidsforsinkelse. 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  $5 * \tau = 18.5$  ns, hvilket med den valgte modstand på 3.3kOhm giver en kondensator værdi på 1.12 pF hvilket rundes op til 1.2pF.

Til sidst er de ubrugte logiske elementer er koblet som følger:

### 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.

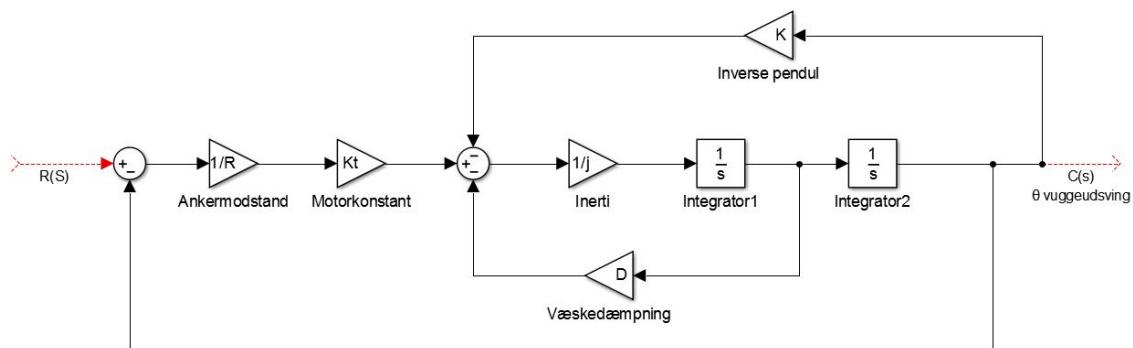
### Endstopsensor

### Vuggeudsving sensor

### Motor positions sensor

### Mekanisk vuggesystem

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



*Figur 6.13.* 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 akslen der vugges over i systemet
- **Det Inverse pendul** er et ulineært element der stammer fra at barnevognskurvvens masse ligger højere end akslen der vugges over. Dette element er ustabilt af natur og kræver påvirkning for at holdes stabilt. Dette er især gældende forstørre

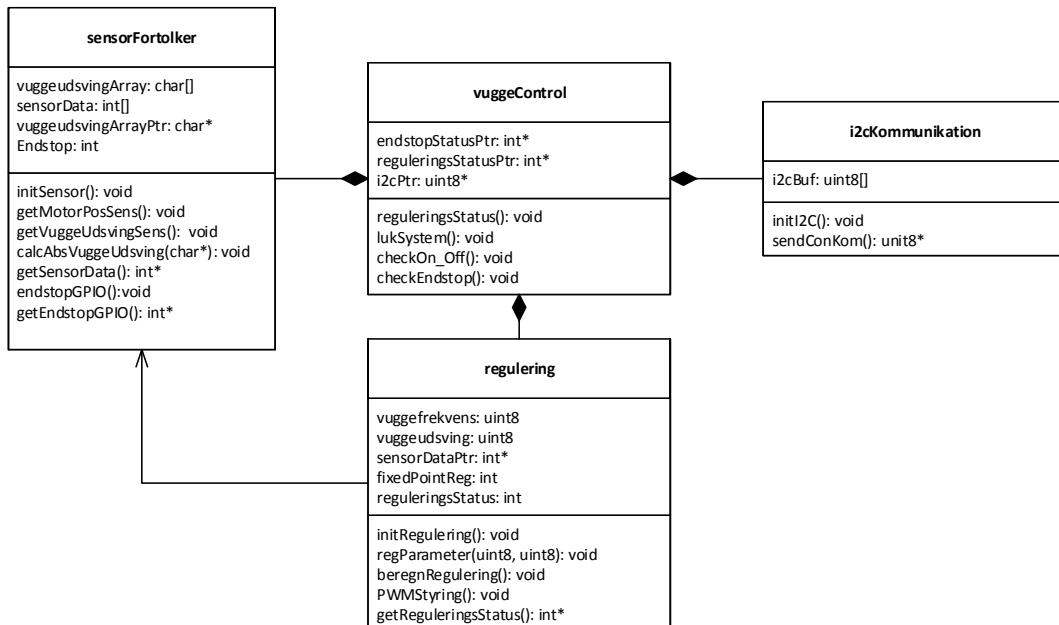
udfaldsvinkler, men da niveauet der skal vugges over maximalt udgør  $\pm 10$  grader i forhold til tyngdefeltet kan dette element gøres tilnærmelsesvist lineært

- Fra højre mod venstre skal punktet efter **Interator2** 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 6.14.** UML klassediagram for Regulerings MCU

#### Funktionsbeskrivelser

##### vuggeControl

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

<b>Funktion</b>	<code>void reguleringsStatus()</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	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>

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

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

### i2cKommunikation

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

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

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

### regulering

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

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

<b>Funktion</b>	<code>void regParameter(uint8 vuggefrekvens, uint8 vuggeudsning)</code>
<b>Parametre</b>	uint8 vuggefrekvens indeholder en binærværdi fra 0-255 der udfra en skala svare til en vuggefrekvens jf. Kommunikationsprotokol 3.3.1, uint8 vuggeudsning indeholder en binærværdi fra 0-255 der ud fra en skala svare til et vuggeudsning jf. Kommunikationsprotokol 3.3.1
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Gemmer og validere ny vuggefrekvens og vuggeudsning til at regulerer udfra.

<b>Funktion</b>	<code>void beregnRegulering()</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Beregner reguleringen for vuggesystem via angivet værdier vuggefrekvens, vuggeudsning og sensordata samt overføringsfunktionen for vuggesystemet og opdaterer fixedPointReg med en værdi X? ud fra et 16.16 fixed-point real number tabel opslag(IRT undervisning først)

<b>Funktion</b>	<code>void PWMStyring()</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Styrer uadgående PWM udfra fixedPointReg værdi X? (IRT uv)

<b>Funktion</b>	<code>int* getReguleringsStatus()</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Int*
<b>Beskrivelse</b>	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.

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

<b>Funktion</b>	<code>void getMotorPosSens()</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Opdaterer sensorData arrays plads 0 med den nyeste motor position angivet i værdi fra 0-255

<b>Funktion</b>	<code>void getVuggeUdsvingSens()</code>
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Opdaterer vuggeudsningArray plads 0-5 med de nyeste gyroskop- og accellerometersensor målinger angivet i værdier fra 0-255

<b>Funktion</b>	void calcAbsVuggeUdsving()
<b>Parametre</b>	Ingen
<b>Returværdi</b>	Ingen
<b>Beskrivelse</b>	Udfra accellerometer- og gyroskopmålingerne i vuggeudsvingArray beregnes og opdateres int sensorData[] plads 1 med den absolutte vinkel for barnevognskurven angivet i værdi X?

<b>Funktion</b>	int* getSensorData()
<b>Parametre</b>	Ingen
<b>Returværdi</b>	int*
<b>Beskrivelse</b>	Sender pointer fra sensorData[] til klassen regulering

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

<b>Funktion</b>	int* getEndstopGPIO()
<b>Parametre</b>	Ingen
<b>Returværdi</b>	int*
<b>Beskrivelse</b>	Sender pointer fra int endstop til klassen vuggeControl

## 6.3 Implementering

### 6.3.1 Hardware implementering

Motorstyringskreds

Endstopsensor

Vuggeudsving sensor

Motor positionssensor

Mekanisk vuggesystem

### 6.3.2 Software implementering

Regulerings MCU

Vuggeudsving sensor

## 6.4 Modultest

### 6.4.1 Hardware modultest

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

**Motorkreds****Formål**

// Tekst

**Overordnet opstilling af x test**

- Første punkt er test miljøet
- Efterfølgende punkter er opstillingen beskrevet trinvist
- Osv..

// Figur 1 - med evt. opstilling

**Testbeskrivelse**

- Hvad skal ske først
- Hvad skal der ske efterfølgende
- Osv, osv..

**Forventet resultat**

// Tekst

**Resultat**

// Tekst

// Figur 2 - Billede af resultat (Oscilloskop, screenshot, e.l)

*Testen er godkendt/ikke-godkendt*

**Endstopsensor****Vuggeudsving sensor****Motor positions sensor****Mekanisk vuggesystem****6.4.2 Software modultest****Regulerings MCU****Vuggeudsving sensor**



# 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 / XX mA (Raspberry Pi + Betjeningspanel)
- **Vuggesystem:** Kræver 3,3 V / XX mA (PSoC + Motorstyringskreds)
- **Intelligent Lydmonitor:** Kræver 7,5 V / XX 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.

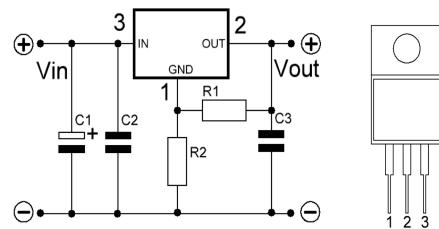


**Figur 7.1.** Fabriksbillede af LM2596S - DC-DC konverter

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

5 V til Controller

3,3 V til Intelligent Lydmonitor

#### 7.0.5 Relæ styring

# **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

<b>UC2: Vug barnevogn</b>				
	<b>Test</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt/ Kommentar</b>
<b>1</b>	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		
<b>2</b>	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		
<b>3</b>	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.

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

...fortsat fra forrige side

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

## Use case 4

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

...fortsat fra forrige side

	<b>Test</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt/ Kommentar</b>
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

	<b>Test</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt/ Kommentar</b>
<b>Mikrofon:</b>				
1	Mikrofonens parametre undersøges gennem producentens dokumentation.	De anførte mikrofon parametre er i overensstemmelse med kravene	Skriv her	Skriv her
<b>Vuggesystem:</b>				
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

	<b>Test</b>	<b>Forventet Resultat</b>	<b>Resultat</b>	<b>Godkendt / Kommentar</b>
<b>4</b>	FIXME: Test af begrænsning i vinkelfrekvens, og vinkelacceleration er svær at udtænke. TODO: snak med Carl	Skriv her	Skriv her	Skriv her
	<b>Baby status</b>			
<b>1</b>	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

---