

Underskrifter

Dato for aflevering: 28/05-2015

Gruppemedlemmer:

Kristian Boye Jakobsen (KB) 201370738

Lukas Hedegaard Jensen (LH) 201370768

Poul Overgaard (PO) 201370794

Jeppe Hofni (JH) 201370801

Mick Kirkegaard (MK) 201370970

Felix Blix Everberg (FB) 201371008

Vejleder: Arne Justesen

Indholdsfortegnelse

Kapitel 1 Termliste	5
Kapitel 2 Kravspecifikation	7
2.1 Indledning	7
2.2 Use cases	9
2.3 Aktører	9
2.4 Fully dressed use cases	10
2.5 Ikke-funktionelle krav	13
2.6 Web-skitser	14
2.7 E-mail beskeder	16
Kapitel 3 Overordnet	17
3.1 Systemarkitektur	17
3.1.1 Domænemodel	17
3.1.2 System-states	18
3.1.3 BDD	19
3.1.4 IBD	19
3.2 Signalbeskrivelse	20
3.2.1 Allokering af logiske funktionaliteter	21
3.3 Grænseflader	22
3.3.1 Grænseflade mellem Intelligent Lydmonitor og Controller	22
3.3.2 Grænseflade mellem Vuggesystem og Controller	22
3.3.3 Grænseflade mellem Controllers servere og ekstern internet enhed . .	25
Kapitel 4 Controller	27
4.1 Systemarkitektur	27
4.1.1 Hardware arkitektur	27
4.1.2 Signal	28
4.2 Design	29
4.2.1 Softwaredesign	29
4.3 Implementering	38
4.4 Modultest	38
Kapitel 5 Intelligent lydmonitor	39
5.1 Forundersøgelse	39
5.1.1 Metode	39
5.1.2 Situationer	40
5.1.3 Analysekode	41
5.1.4 Analyser	44
5.1.5 Konklusion	48
5.2 Systemarkitektur	49

5.2.1	Hardware arkitektur	50
5.2.2	Software arkitektur	51
5.3	Design	57
5.3.1	Mikrofon preamp	57
5.4	Implementering	61
5.5	Modultest	61
Kapitel 6	Vuggesystem	63
6.1	Systemarkitektur	63
6.1.1	Hardware arkitektur	63
6.1.2	Grænsefladebeskrivelse	66
6.1.3	Software arkitektur	67
6.2	Design	69
6.2.1	Software design	70
6.3	Implementering	71
6.3.1	Motorstyringskreds	71
6.3.2	strømforbrug	75
6.4	Modultest	76
Kapitel 7	Integrationstest	77
Kapitel 8	Accepttestspezifikation	79
Kapitel 9	Bilag (CD-indhold)	85

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).

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

Manuel start Knap med funktion der manuelt starter vuggemekanismen

Monitorerings-tilstand En tilstand hvor der systemet kører efter hensigten

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

Putte-tilstand En tilstand hvor barnevognen vugger uden at blive styret af lydinput

SPL Sound Preasure Level (dB)

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

Vugge-indsving Skal forstås som at når barnevognen begynder en vugning så skal vuggesystemet give det en blid opstart

Wi-Fi-LED Diode der lyser når WiFi forbindelsen 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 putte-tilstand, hvor barnet lægges til at sove, en monitorerings-tilstand og en undtagelses-tilstand. I **putte-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** låses systemet. Barnevognen skal ikke vuges. 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 vuges. Systemet indsamler lyd og afventer en ændring i lydsignalet, som vil medføre en ændring i BABYCON niveau.

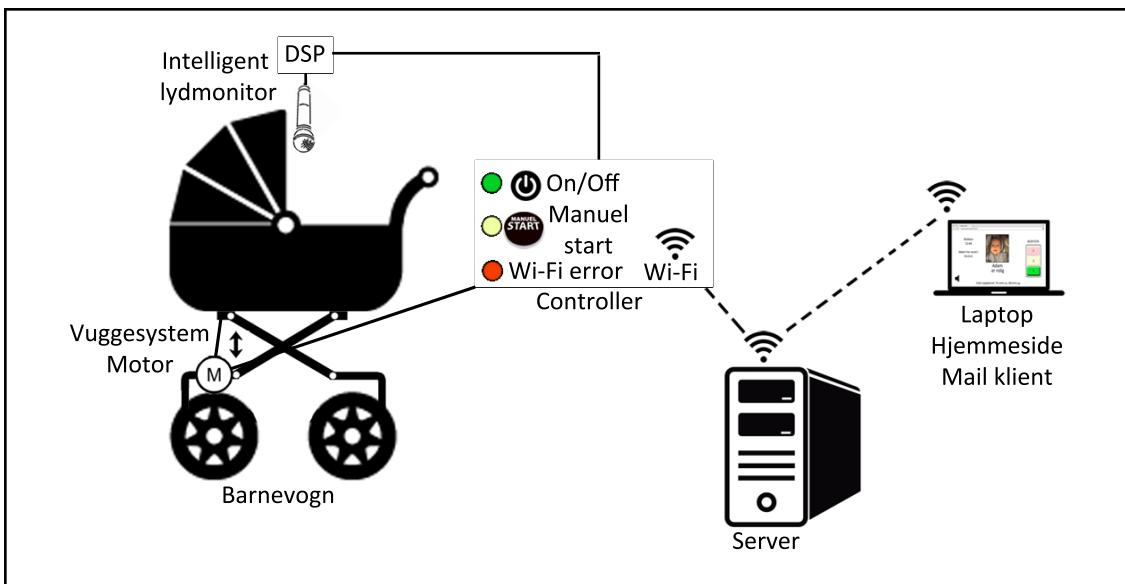
- **BABYCON2**

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 vugges 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.

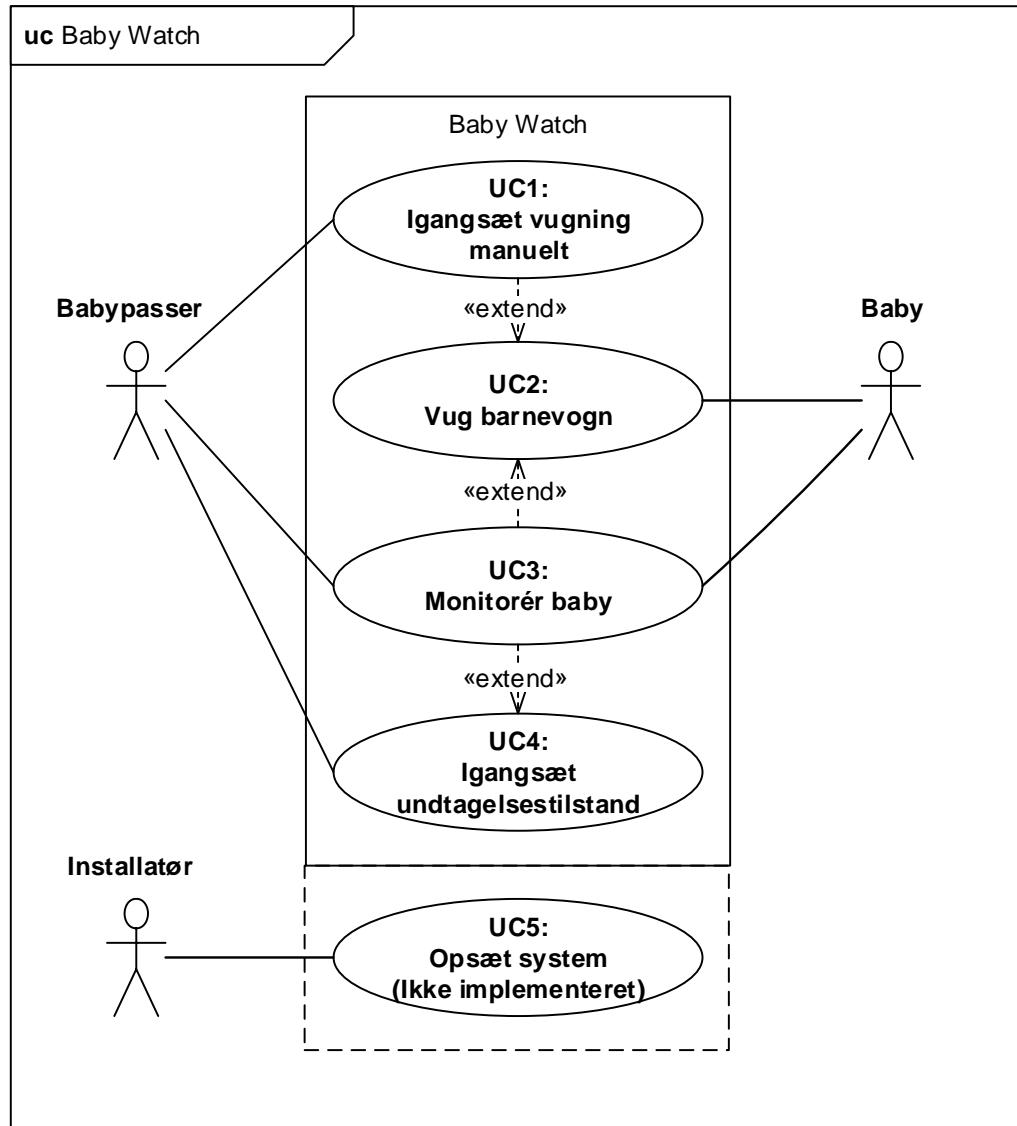


Figur 2.2. Forklaring af controllerens knapper og lysdioder

2.2 Use cases

I dette afsnit specificeres brugssituationer for systemet Baby Watch.

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



Figur 2.3. Usecase diagram for Baby Watch

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

2.3 Aktører

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

2.4 Fully dressed use cases

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

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

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

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

2.5 Ikke-funktionelle krav

Mikrofon

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

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

Vuggemekanisme

Nivellering:

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

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

¹FIXME indsæt reference til studie om gråd volumen

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

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

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

Baby status

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

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

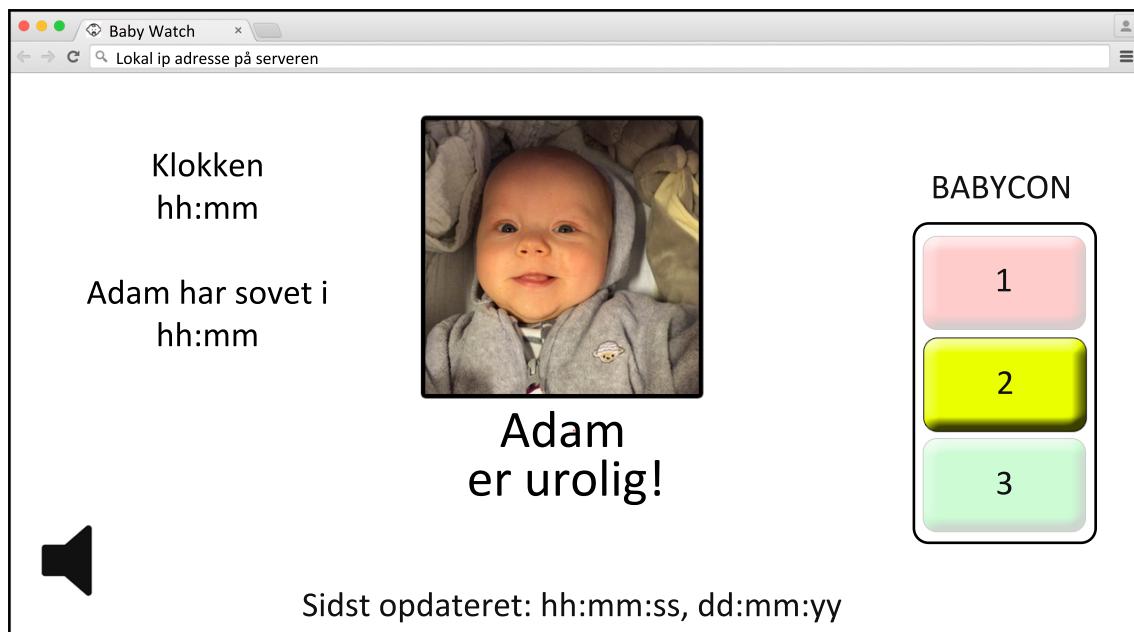
2.6 Web-skitser

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.



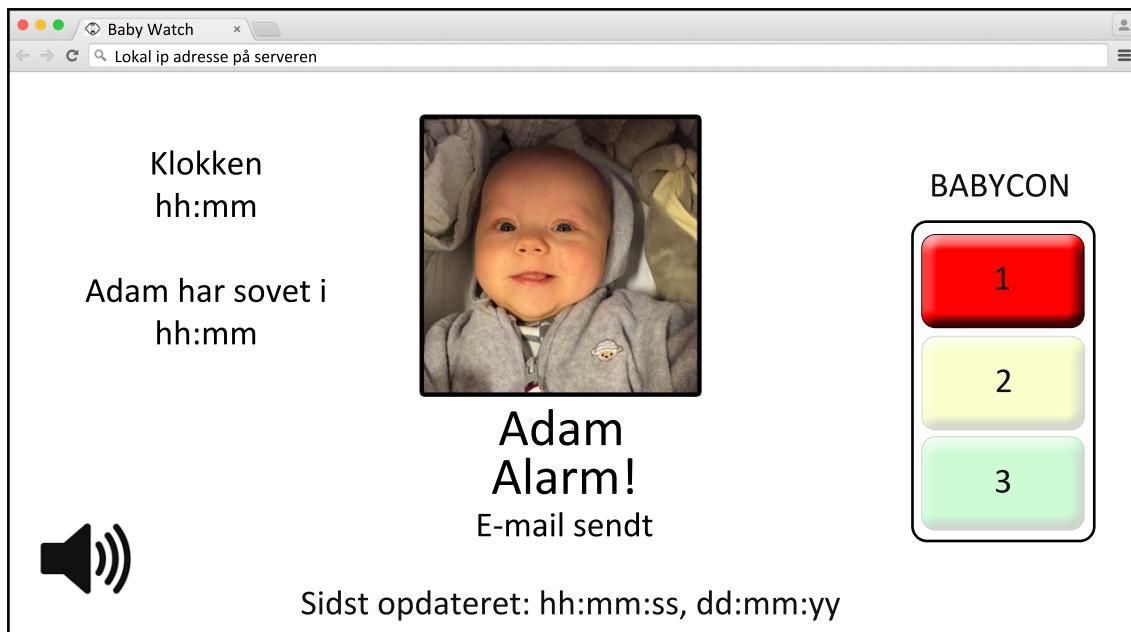
Figur 2.4. BABYCON3

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!

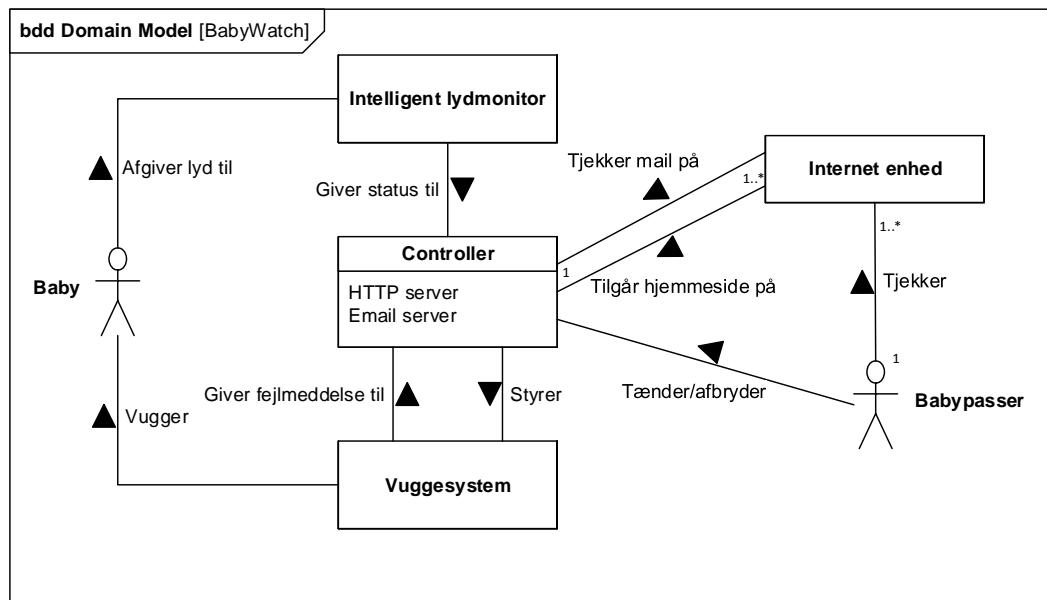
Tekst: Fejl i vuggesystem, afventer genstart.

Overordnet 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.

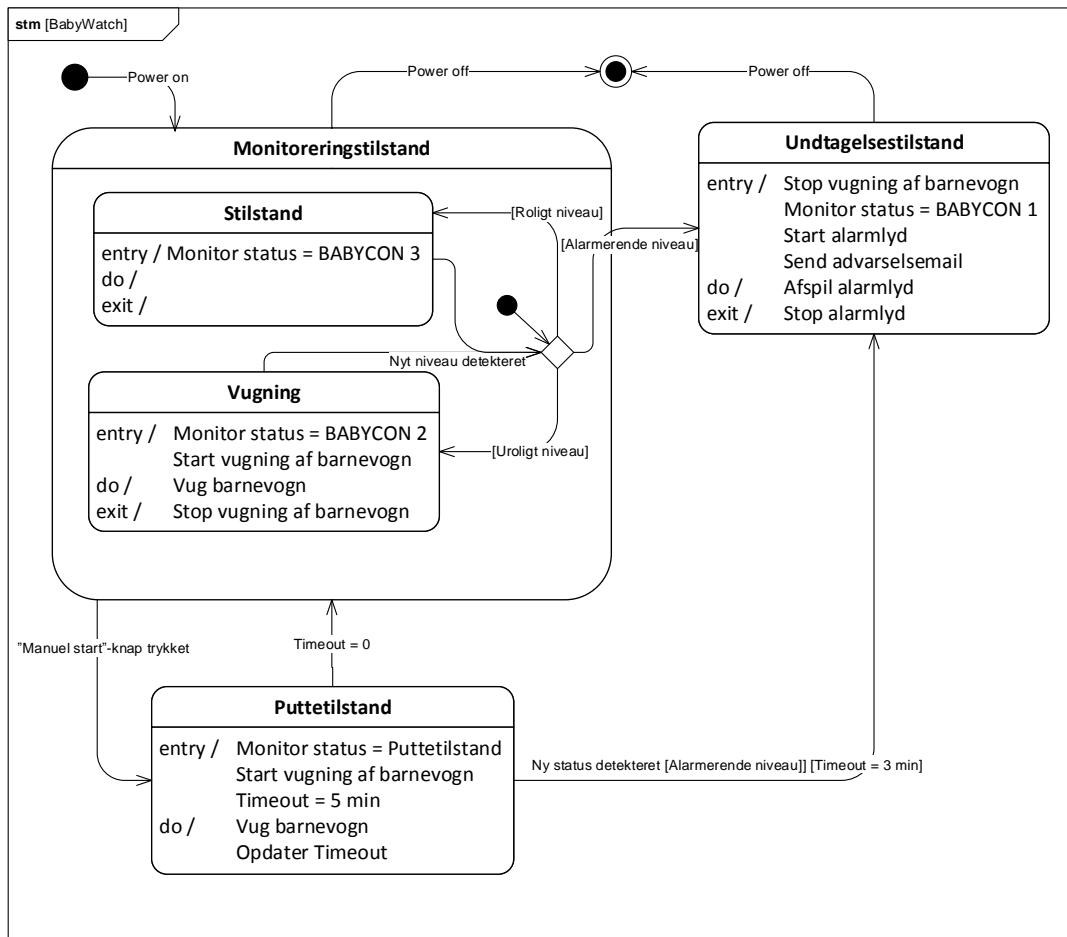
3.1.1 Domænemodel



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 systemet

Systemets har tre overordnede states: Monitoringstilstand, Puttetilstand og Undtagelsestilstand.

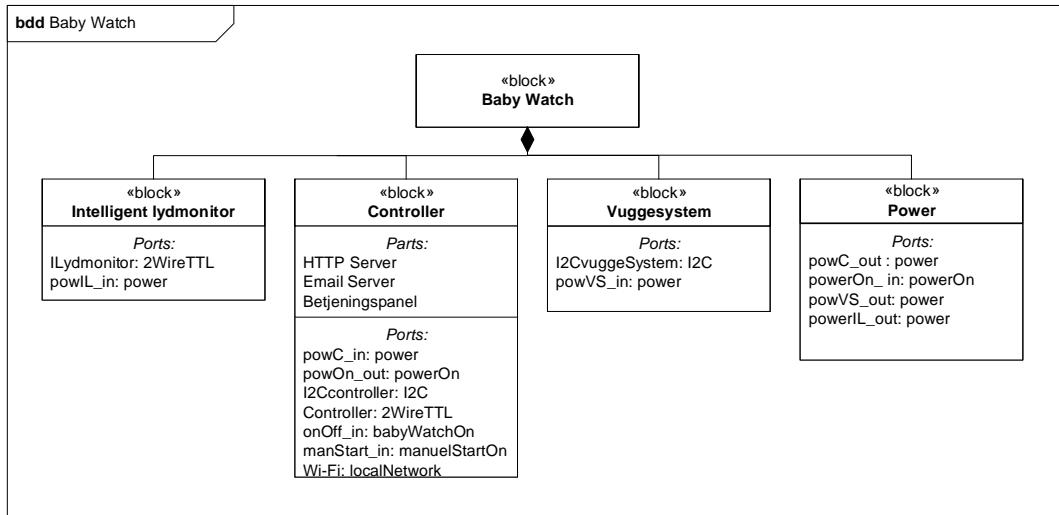
Monitoringstilstand har to substates:

- **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.

Puttetilstand 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 puttetilstand. 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.

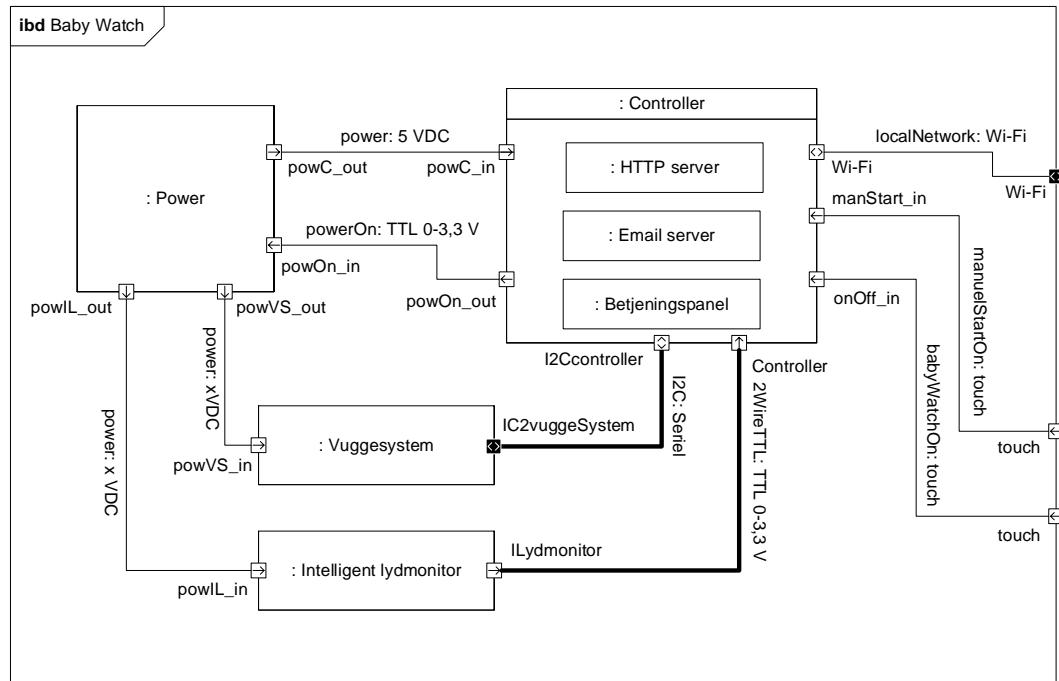
3.1.3 BDD



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 server samt et betjeningspanel.

3.1.4 IBD



Figur 3.4. Overordnet IBD for Baby Watch

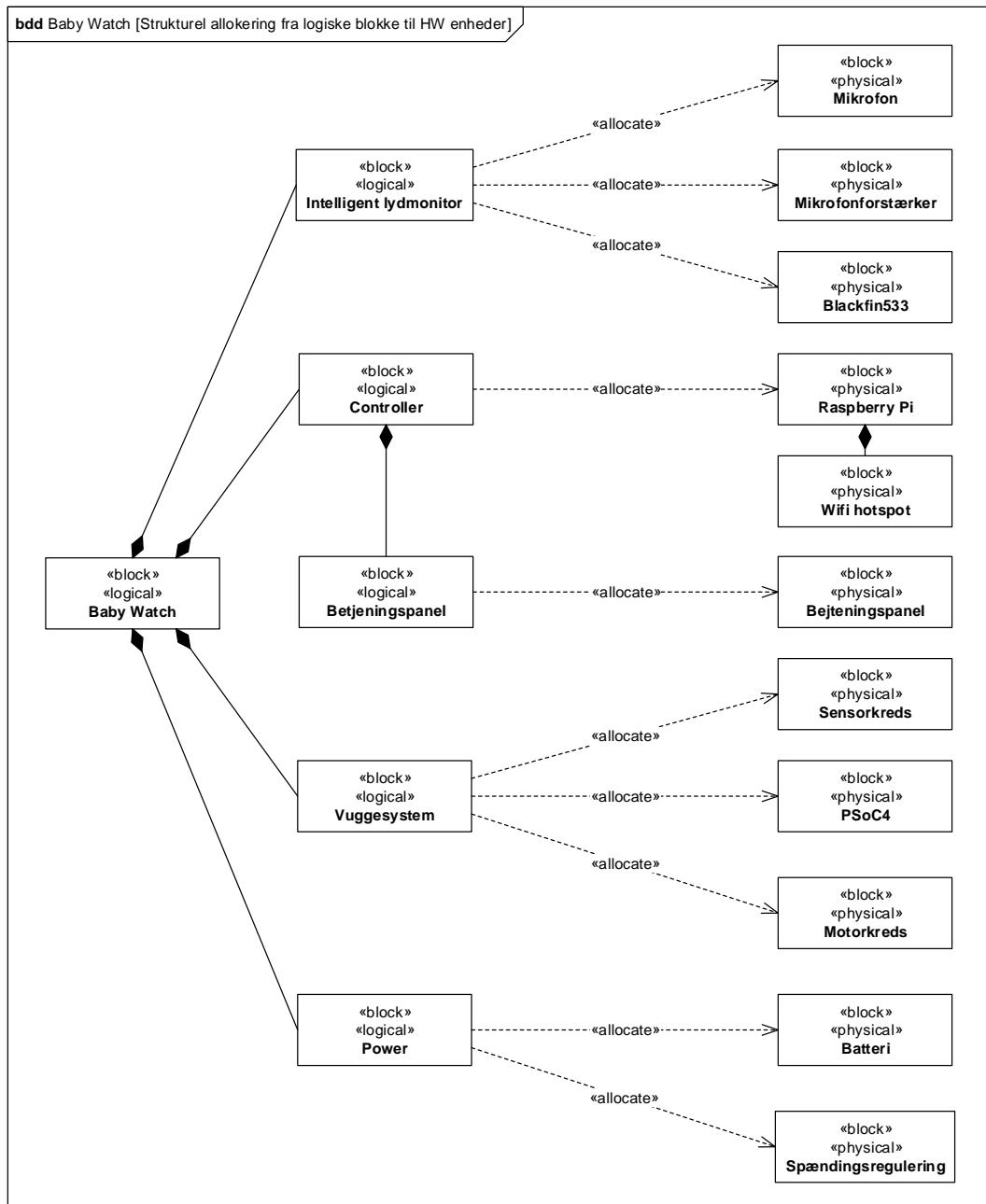
Figur 3.4 viser det overordnede IBD for Baby Watch systemet. Figuren viser de interne forbindelser for blokkene i figur 3.3. For yderlige specifikation af porte og signaler se signalbeskrivelsen 3.2.

3.2 Signalbeskrivelse

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

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	x VDC	Power (powVS_out)	Vuggesystem (powVS_in)	x V forsyning til Vuggesystem
power	x VDC	Power (powIL_out)	Int. lydmonitor (powIL_in)	x V 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
localNetwork	Wi-Fi	Controller (Wi-Fi)	Baby Watch (Wi-Fi)	Lokal trådløs lan forbindelse

3.2.1 Allokering af logiske funktionaliteter



Figur 3.5. Allokeringsdiagram 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- og Email- server er således også allokteret 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 spændingsregulerende kreds.

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 niveauet.

Kommunikationsprotokol

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

Tabel 3.2. BABYCON1

	2WireTTL_MSБ	2WireTTL_LSB
Bit værdi	0	1
	2WireTTL_MSБ	2WireTTL_LSB

Tabel 3.3. BABYCON2

	2WireTTL_MSБ	2WireTTL_LSB
Bit værdi	1	0
	2WireTTL_MSБ	2WireTTL_LSB

Tabel 3.4. BABYCON3

	2WireTTL_MSБ	2WireTTL_LSB
Bit værdi	1	1
	2WireTTL_MSБ	2WireTTL_LSB

Tabel 3.5. Fejl fra Intelligent Lydmonitor

	2WireTTL_MSБ	2WireTTL_LSB
Bit værdi	0	0
	2WireTTL_MSБ	2WireTTL_LSB

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 Intelligente Lydmonitors side.

3.3.2 Grænseflade mellem Vuggesystem og Controller

Kommunikationen mellem Vuggesystemet og Controllen foregår via I2C. Der benyttes 5 skifteredistre 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 her:

Tabel 3.6: Specifikation af I2C grænseflade

I2C Adresse: 0b1111000X (Write: 0xF0, Read: 0xF1)				
I2C Frekvens: 100kHz				
Reg#	Navn	Type	Beskrivelse	Startværdi

fortsættes på næste side ->

Tabel 3.6 -> fortsat fra forrige side

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	Status	R	Indeholder en bitsekvens som indikerer systemets status. Registeret indeholder følgende: [ERR STALL END_STP SD_RDY X X X X]	0b0000XXXX
	ERR		Indikerer at der er opstået en fejl i systemet. Kendes årsagen til fejlen indikeres denne i STALL og END_STP	
	STALL		Indikerer at systemet har været ude af stand til at drive motoren, formegentlig pga for stor belastning	
	END_STP		Indikerer at vuggen har ramt en af sine mekaniske yderpositioner, og vuggesystemet er deaktivert indtil der er blevet genstartet.	
	SD_RDY		Indikerer at systemet er klar til at få afbrudt strømmen	
0x02	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
0x03	Frekvens	R/W	Værdien i dette register styrer frekvensen hvormed der vugges. Område: 0 Hz = 2,550 Hz, 1 LSB = 10 mHz.	0x00
0x04	Vinkeludsving	R/W	Værdien i dette register kontrollerer størrelsen af vuggens udsving i grader. Område: +/- 12,75°, 1 LSB = 0,05°.	0x00

3.3.3 Grænseflade mellem Controllers servere og ekstern internet enhed

Fysiske porte

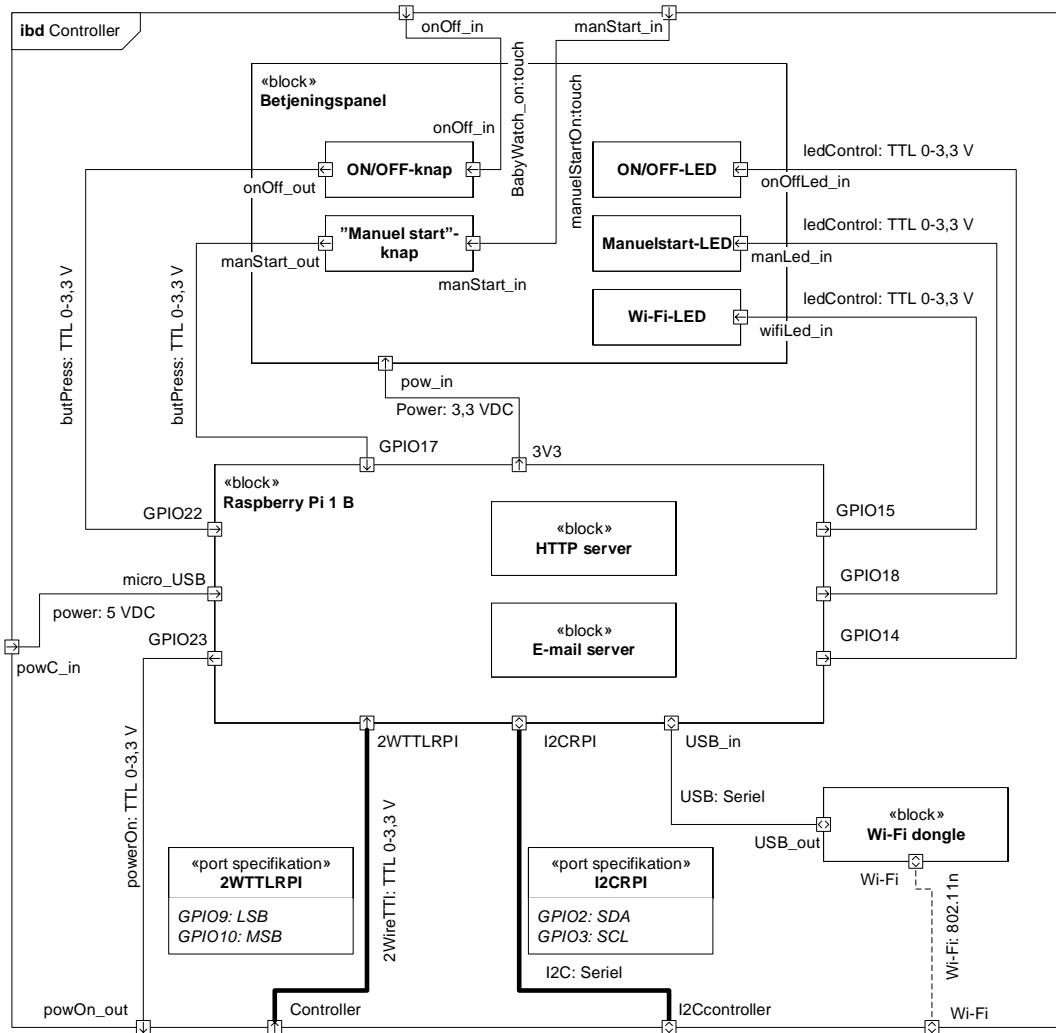
Kommunikationsprotokol

Controller 4

4.1 Systemarkitektur

I dette afsnit beskrives systemarkitekturen for Controller.

4.1.1 Hardware arkitektur



Figur 4.1. 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.

4.1.2 Signal

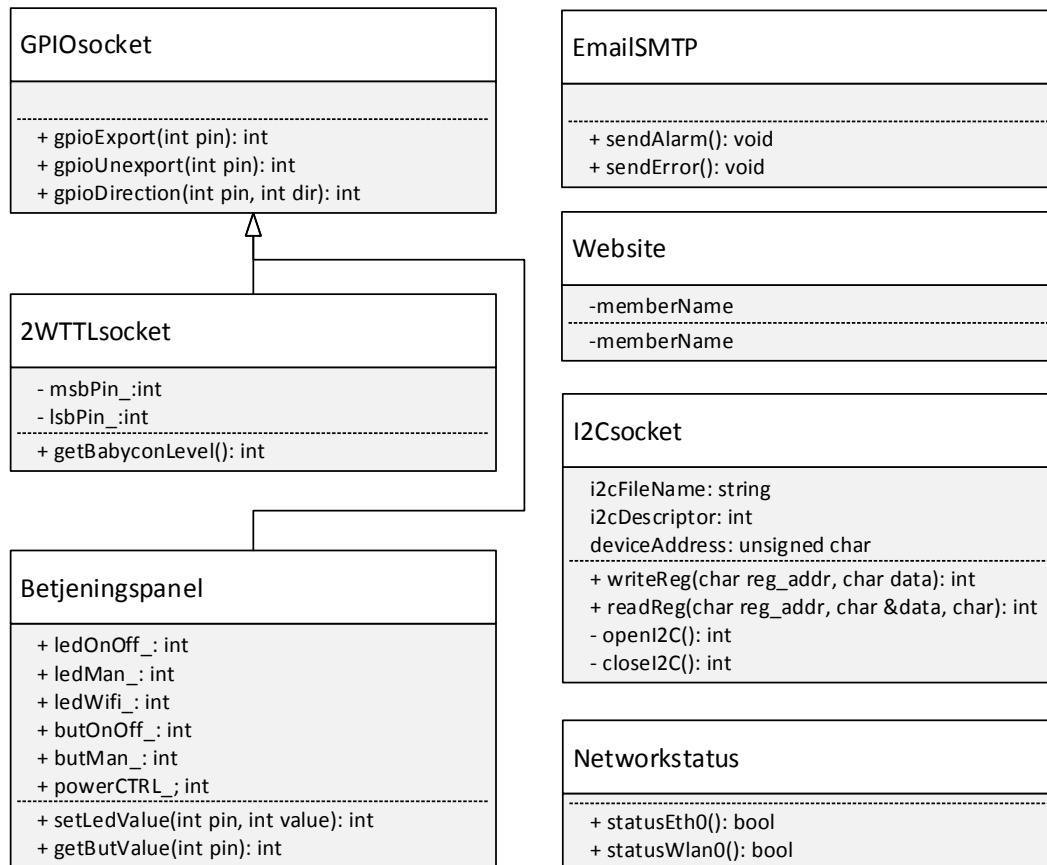
Signal-navn	Type	Port 1	Port 2	Kommentar
manStartOn	touch	Betjeningspanel (manStart_in)	Babypasser (touch)	Babypasser der laver et tryk på knappen
babyWatchOn	touch	Betjeningspanel (onOff_in)	Babypasser (touch)	Babypasser der laver et tryk på knappen
power	5 VDC	RasberryPi (micro_USB)	Controller (powC_in)	5 V forsyning til controller
power	53,3 VDC	RasberryPi (3V3)	Betjeningspanel (pow_in)	3,3 V forsyning til Betjeningspanel
powerOn	TTL 0-3,3 VDC	RasberryPi (GPIO23)	Controller (powOn_out)	Benyttes til at tænde og slukket for strømforsyningen
2WireTTL	TTL 0-3,3 VDC	RasberryPi (2WTTLRPI) GPIO9: LSB GPIO10: MSB	Controller (I2CController)	Seriell kommunikation
I2C	Seriell	RasberryPi (I2CRPI) GPIO2: SDA GPIO3: SCL	Controller (Controller)	BABYCON niveau
USB	Seriell	RasberryPi (USB_in)	Wi-Fi dongle (USB_out)	Seriell kommunikation for Wi-Fi forbindelse
ledControl	TTL 0-3,3 VDC	RaspberryPi (GPIO14) RaspberryPi (GPIO15) RaspberryPi (GPIO18)	Betjeningspanel (onOffLed_in) Betjeningspanel (manLed_in) Betjeningspanel (wifiLed_in)	Logisk signal til at styre LED på Betjeningspanel
butPress	TTL 0-3,3 VDC	RaspberryPi (GPIO17) RaspberryPi (GPIO22)	Betjeningspanel (onOff_out) Betjeningspanel (manStart_out)	Logisk signal til kontrol af knapper på betjeningspanel

4.2 Design

4.2.1 Softwaredesign

I dette afsnit beskrives hvordan Controllerens software er designet med klassediagrammer

Klassebeskrivelser



Figur 4.2. Oversigt over klassediagrammerne for Controller

Betjeningspanel

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

```
int setLedValue( int pin , int value )
```

Parametre: Modtager et GPIO pin-nummer og en værdi 0 for ON og 1 for OFF

Returværdi: 0 ved succes. Minus værdi ved fejl

Beskrivelse: Bruges til at tænde og slukke for LED dioderne på controller

```
int getButValue( int pin ) const
```

Parametre: Modtager et GPIO pin-nummer

Returværdi: Status på knaptryk

Beskrivelse: Bruges til at læse fra trykknapper på controller

```
int ledOnOff_
```

Beskrivelse: Attribut til at holde GPIO nummer for On/Off LED

```
int ledMan_
```

Beskrivelse: Attribut til at holde GPIO nummer for Manuelstart LED

```
int ledWifi_
```

Beskrivelse: Attribut til at holde GPIO nummer for Netværksstatus LED

```
int butOnOff_
```

Beskrivelse: Attribut til at holde GPIO nummer for On/Off knappen

```
int butMan_
```

Beskrivelse: Attribut til at holde GPIO nummer for "Manuelstart"-knap

EmailSMTP

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

```
void sendError()
```

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Sender en Email med en fejlmeddeelse til Babypasser

```
void sendAlarm()
```

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Sender en Email med en alarm besked til Babypasser

GPIOsocket

Ansvar: At oprette et GPIO filarkiv

```
int gpioExport( int pin )
```

Parametre: GPIO pin nummer

Returværdi: 0 ved succes. Minus værdi ved fejl

Beskrivelse: Opretter et fil arkiv for det modtagne pin-nummer

```
int gpioUnexport( int pin )
```

Parametre: GPIO pin nummer

Returværdi: 0 ved succes. Minus værdi ved fejl

Beskrivelse: Fjerner filarkivet for det modtagne pin-nummer

```
int gpioDirection( int pin , int dir )
```

Parametre: GPIO pin nummer og pin retning

Returværdi: 0 ved succes. Minus værdi ved fejl

Beskrivelse: Sætter retningen for GPIO pin, INPUT(0) eller OUTPUT(1)

I2Csocket

Ansvar: At kommunikere over I2C

```
int writeReg( unsigned char reg_addr , unsigned char data )
```

Parametre: Register adresse og data til skrivning

Returværdi: 0 ved succes. Minus værdi ved fejl

Beskrivelse: Skriver data til et register på en given enhed

```
int readReg( unsigned char reg_addr , unsigned char &data )
```

Parametre: Register adresse og en data adresse til at gemme læst data i

Returværdi: 0 ved succes. Minus værdi ved fejl

Beskrivelse: Læser data fra et register på en given enhed

```
int openI2C()
```

Parametre: Ingen

Returværdi: 0 ved succes. Minus værdi ved fejl

Beskrivelse: Åbner I2C forbindelsen til en enhed i constructor

```
int closeI2C()
```

Parametre: Ingen

Returværdi: 0 ved succes. Minus værdi ved fejl

Beskrivelse: Lukker I2C forbindelsen til en enhed i destructor

```
std::string i2cFileName
```

Beskrivelse: Enhedsnavn på Raspberry Pi ”/dev/i2c-0” eller ”/dev/i2c-1”

```
int i2cDescriptor
```

Beskrivelse: Fil descriptor til åbning af I2C forbindelsen

```
unsigned char deviceAddress
```

Beskrivelse: Attribut til at holde Enheds adressen på den enhed der skal kommunikeres med

Networkstatus

Ansvar: At aflæse om der er forbindelse til et netværk

`bool statusEth0()`

Parametre: Ingen

Returværdi: True ved forbindelse ellers false

Beskrivelse: Læser på ”operstate” for Eth0 forbindelsen på Raspberry Pi og returnere om den er oppe eller nede

`bool statusWlan0()`

Parametre: Ingen

Returværdi: True ved forbindelse ellers false

Beskrivelse: Læser på ”operstate” for Wlan0 forbindelsen på Raspberry Pi og returnere om den er oppe eller nede

TWTLLsocket

Ansvar: At aflæse status fra Lydmonitoreringen

`int getPinValue(int pin) const`

Parametre: GPIO pin nummer

Returværdi: Pin værdi

Beskrivelse: Bruges til læsning af MSB og LSB bit fra den Intelligente Lydmonitor

`int getBabyconLevel(void)`

Parametre: Ingen

Returværdi: BABYCON niveau 0, 1, 2 eller 3

Beskrivelse: Udlæsning af BABYCON niveau

`int msbPin_`

Beskrivelse: Attribut til at holde GPIO nummer for MSB-bit

`int lsbPin_`

Beskrivelse: Attribut til at holde GPIO nummer for LSB-bit

Website

Ansvar: Til opdatering af hjemmeside

Hardwaredesign

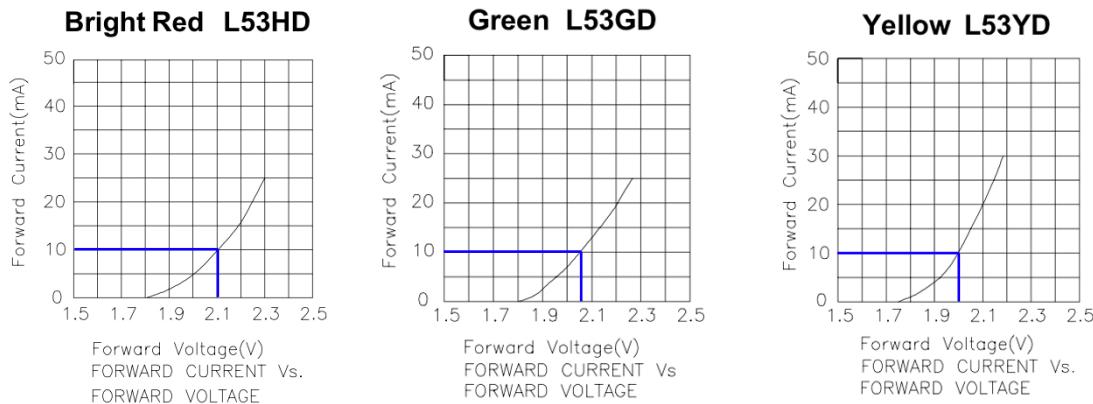
Controlleren består af et betjeningspanel samt en raspberry pi der kontrollerer betjeningspanelet og styrer forbindelserne til hhv. den Intelligente Lydmonitor og Vuggesystemet.

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.

Lysdioder

Der 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 4.3. Udsnit af datablad for KINGBRIGHT L53 HD, GD og YD

Ud fra figur 4.3 ses strømmen som funktion af spaendingen 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

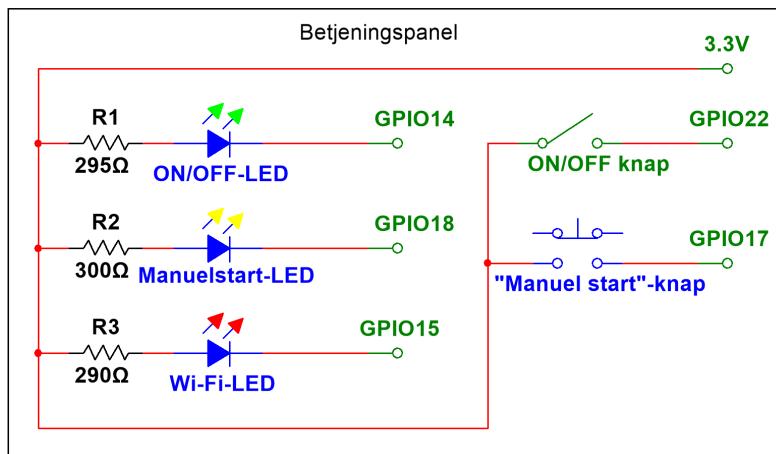
$$R_{L53HD(RED)} = \frac{5V - 2,1V}{10mA} = \frac{2,9V}{10mA} = 290\Omega$$

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

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

Figur 4.4. For modstandsberinger for de 3 dioder

Ud fra modstandsberingerne i figur 4.4 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 4.5. Kredsløbsdiagram for Controller

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

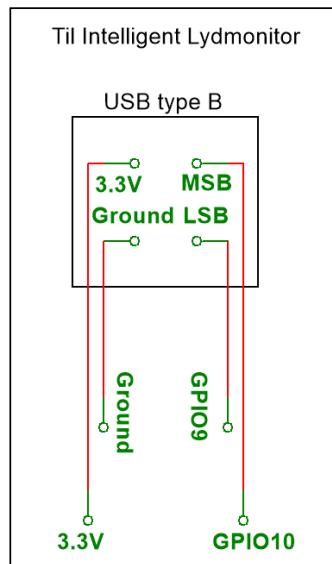
GPIO	Opsætning	Kommentar
GPIO14	Indgang sættes aktiv-lav	Styrer ON/OFF-LED
GPIO18	Indgang sættes aktiv-lav	Styrer Manuelstart-LED
GPIO15	Indgang sættes aktiv-lav	Styrer Wi-Fi-LED
GPIO22	Indgang sættes aktiv-høj, med intern pull-down modstand	Tænd/sluk signal for Baby Watch
GPIO17	Indgang sættes aktiv-høj,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

Kommunikation til/fra Intelligent Lydmonitor og Vuggesystem

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.

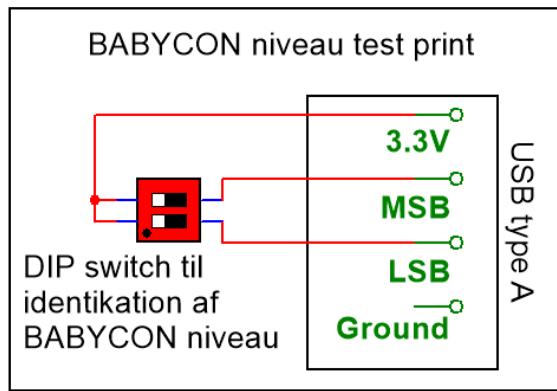
GPIO9 og GPIO10 benyttes til denne to ledet forbindelse til Intelligent Lydmonitor, hvor GPIO9 er LSB delen og GPIO10 er MSB delen.

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 4.6



Figur 4.6. USB type B for kommunikation med Intelligent Lydmonitor

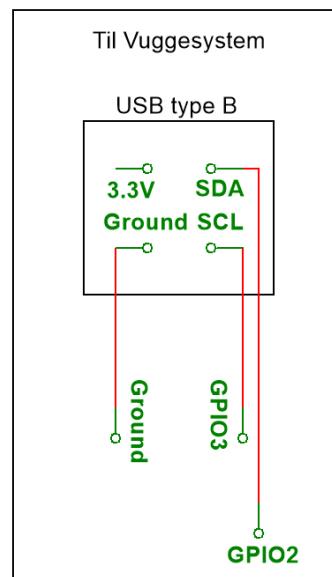
Der designes desuden et testprint se figur 4.7



Figur 4.7. Testprint med dipswitch til at angive BABYCON niveau

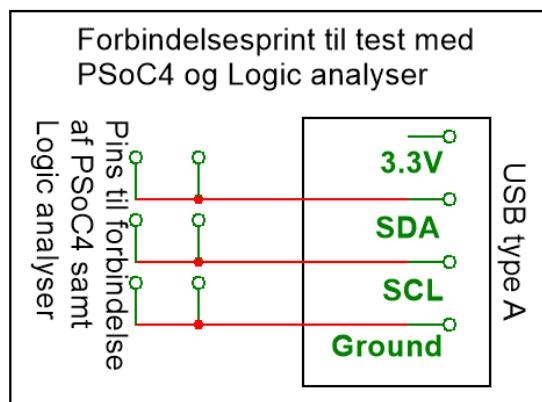
Kommunikation med Vuggesystem

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 4.8. USB type B for kommunikation med Vuggesystem

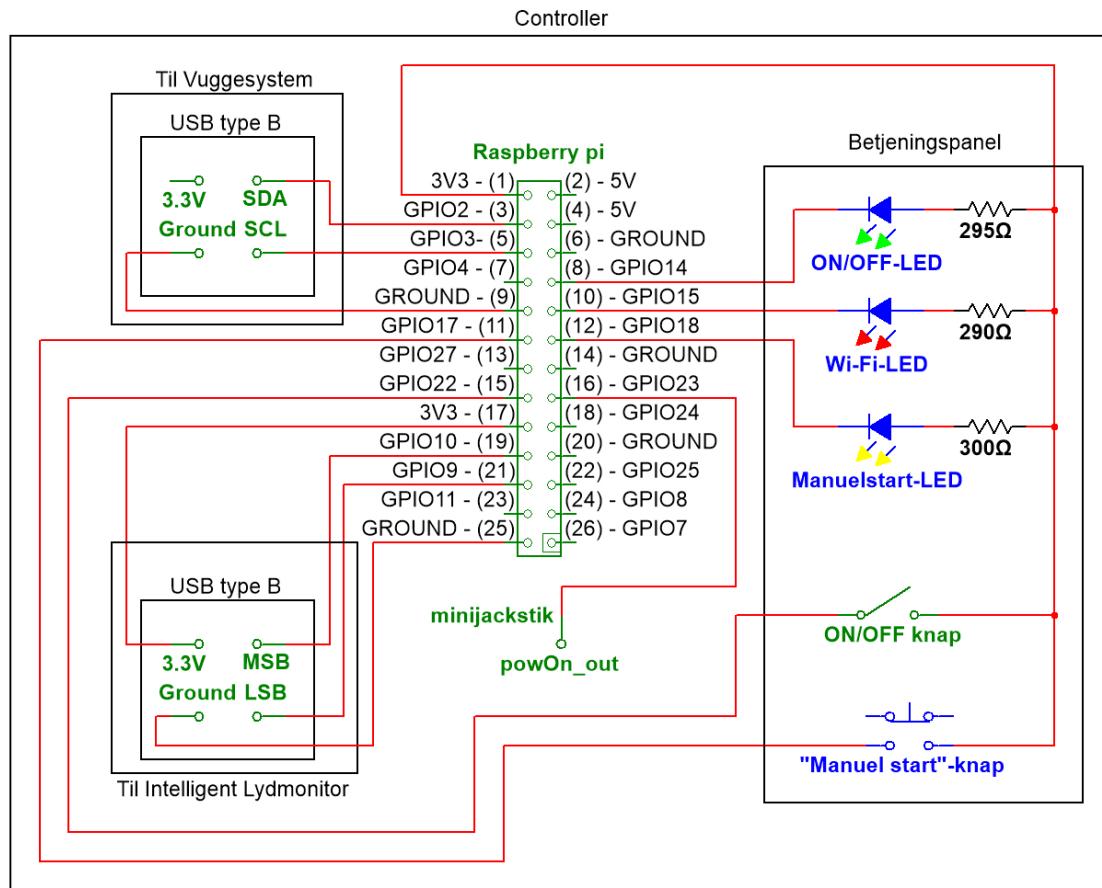
Der designes et testprint således at controlleren kan testes med et simpelt PSoC4 program og datakommunikationen kan verificeres vha. en Logic Analyser.



Figur 4.9. Testprint til I2C forbindelsen, med pins til PSoC4 samt Logic Analyser

Samlet HW-design af Controller

4.10

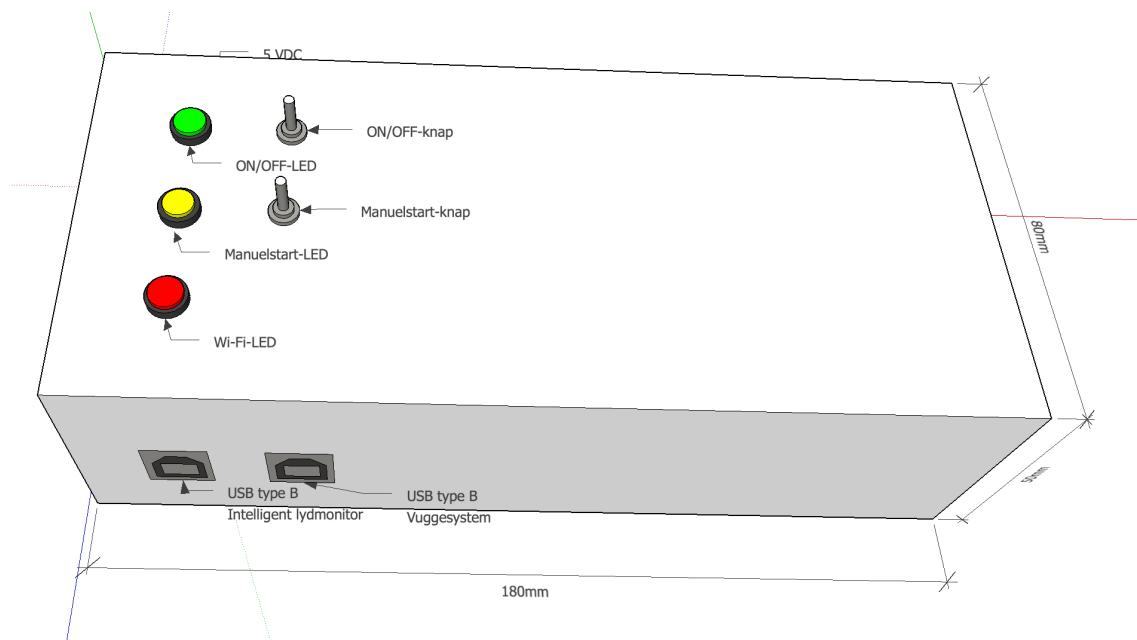


Figur 4.10. Samlet kredsløbsdiagram for Controller

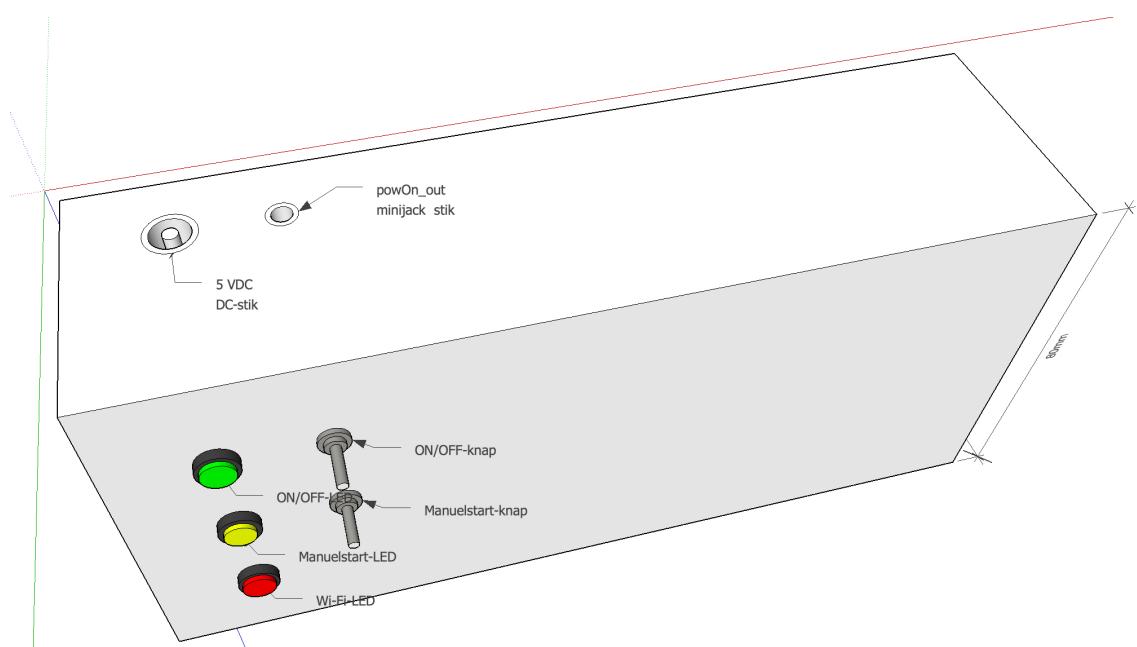
Fysisk design af Controller

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 4.11 og 4.12 viser kassen, med tilhørende LEDs, kontakter, stik samt mål på selve kassen.



Figur 4.11. Arbejdstegning - Betjeningspanel og USB stik



Figur 4.12. Arbejdstegning - Betjeningspanel og forsyningsstik

4.3 Implementering

4.4 Modultest

Intelligent lydmonitor

5

5.1 Forundersøgelse

I dette afsnit undersøges og analyseres udvalgte lyde fra babygråd og typisk omgivelsesstøj såsom billarm og fuglefløjte.

5.1.1 Metode

Der findes optagelser af de udvalgte lyde i formatet .wav. Lydene analyseres i Matlab med hensyn til:

- Generelt frekvensindhold
- Dominant tone
- Grad af tonalitet

Disse parametre analyseres med redskaberne Short-Time DFT, en max-funktion, Tonal Power Spectrum, 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 Spectrum

Tonal Power Spectrum benyttes til beregne tonaliteten i optagelsen, og er altså et udtryk for forholdet mellem dominante toner og det samlede energiindhold.

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

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

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

Smoothing

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

5.1.2 Situationer

Der udføres analyser på lyd-optagelser af følgende situationer:

- Højlydt babygråd
- Moderat babygråd
- Fugleflojt
- Trafik

5.1.3 Analysekode

```
%*****  

% Project: E4PRJ4 BABY WATCH  

% Authors: Lukas Hedegaard og Kristian Boye  

% Task: Forundersøgelse af fire udvalgte lydfiler,  

% Hoejlydt babygraad, Moderat babygraad, Fuglefloejt og Trafik  

%*****  

%**** EXTRACT SAMPLES ****  

[rec, f_s] = wavread('Hoejlydt_babygraad');  

%[rec, f_s] = wavread('Moderat_babygraad');  

%[rec, f_s] = wavread('Fuglefloejt');  

%[rec, f_s] = wavread('Trafik');  

x = rec(:,1)'; %extract left channel'  

%**** Set sample window ****  

frame_sec = 0; %Set frame size from recording length  

%0 for full recording  

if frame_sec == 0  

    N = length(x);  

    shift_N = 1;  

else  

    N = frame_sec*f_s;  

    shift_sec = 1.1; %Set time shift  

    shift_N = shift_sec*f_s;  

end  

x_win = x(shift_N:N-1+shift_N); %extract framed sample  

x_win = x_win .* hanning(length(x_win))'; %window with hamming  

%**** Bandpass prefiltrening ****  

n = 4; % Order  

bs_low_cutoff = 100; % Cut off freq (Hz)  

bs_low_cutoff_rad = bs_low_cutoff/(0.5*f_s); % Cut off freq (rad/sample)  

bs_high_cutoff = 10000;  

bs_high_cutoff_rad = bs_high_cutoff/(0.5*f_s);  

[b,a] = butter(n, [bs_low_cutoff_rad bs_high_cutoff_rad]); % Butter BP  

x_win = filter(b,a,x_win);  

%**** Play sample ****  

sound(x_win, f_s)  

%**** Plot sample ****
```

```

offset = 0;
x = x_win + offset;      %use offset to fit both sampleplot and TPR

figure(1)
subplot(3,1,1)
plot(x)
xlabel('Tid/Samples')
ylabel('Amplitude')
title('Rec: Sample-plot')
grid on

***** FAST FOURIER TRANSFORM *****
NFFT = 2^nextpow2(N);    %Find next power of 2
X = fft(x_win,NFFT);

***** Find Maximal FFT frequency *****
X_max = max(abs(X));

***** SHORT-TIME DFT (Spectrogram) *****
segmentlen = 256;
nOverlap = 60;
NFFT = 1280;

figure(1)
subplot(3,1,3)
spectrogram(x_win, segmentlen, nOverlap, NFFT, f_s, 'yaxis');
specMinFreq = 0;
specMaxFreq = 5000;
axis([-inf inf specMinFreq specMaxFreq])
colormap bone
%colorbar

%Repeat and save data for use in tonal analysis functions:
[s,f,t,p] = spectrogram(x_win, segmentlen, nOverlap, NFFT, f_s,'yaxis');

***** TONAL POWER RATIO *****
TPR = FeatureSpectralTonalPowerRatio(abs(s), f_s, 5*10^-2)';
TPR = TPR*5;    % upscaling

TPR_smooth = smooth(TPR(:,0.1,'moving'));% Smooth data

***** Plot Tonal Power Ratio *****
xAxis = 1:(length(x_win)/length(TPR)):length(x_win);

```

```

figure(1)
subplot(3,1,2)
plot(xAxis,TPR,'r','linewidth',1)
hold on
plot(xAxis,TPR_smooth,'k','linewidth',2)
hold off
xlabel('Tid/sample')
ylabel('Tonal Power Ratio')
title('Tonal Power Ratio')
grid on

***** FREQ WITH MAX POWER *****
[q,nd] = max(10*log10(p)); %q is used for x-axis in later plot

maxPwr_smooth = smooth(f(nd),0.05,'moving'); % Smooth data

***** FREQUENCY ANALYSIS PLOTS *****
***** Discard values beneath Tonal Power Ratio Threshold *****
cutThresh = 0;
maxPwr_smooth_cut = maxPwr_smooth;
maxPwr_cut = f(nd);

for i = 1 : length(maxPwr_smooth)
    %determine spectral energy at freq
    if ((TPR_smooth(i) < cutThresh) || (TPR(i) == 0))
        maxPwr_smooth_cut(i) = NaN(1);
        maxPwr_cut(i) = NaN(1);
    end
end

***** Plot frequency analysis *****
figure(1)
subplot(3,1,3)

hold on
plot3(t,maxPwr_smooth_cut,q,'r','linewidth',2) %maxPwr_smooth plot
scatter(t,maxPwr_cut,5,'b')
view(2)
hold off

```

5.1.4 Analyser

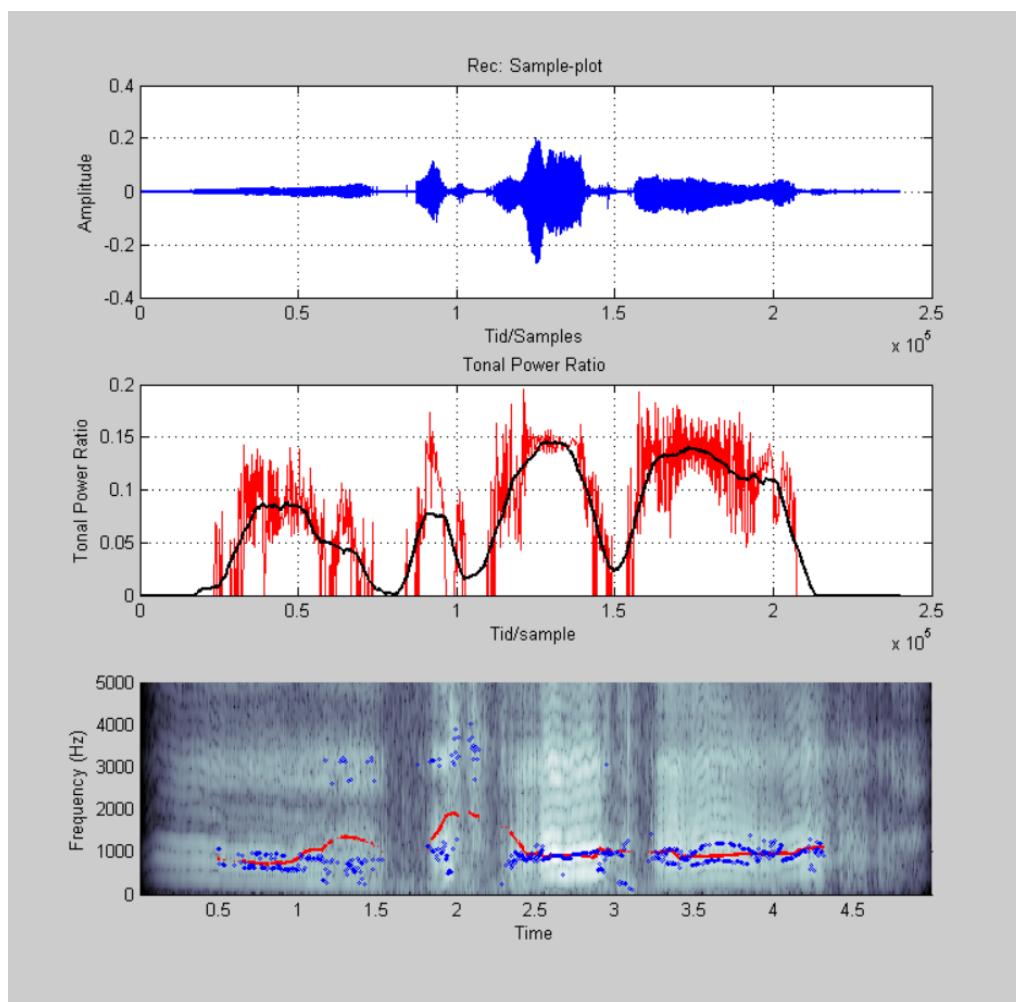
Resultatet af analysekoden er 3-subplots:

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

Højlydt babygråd

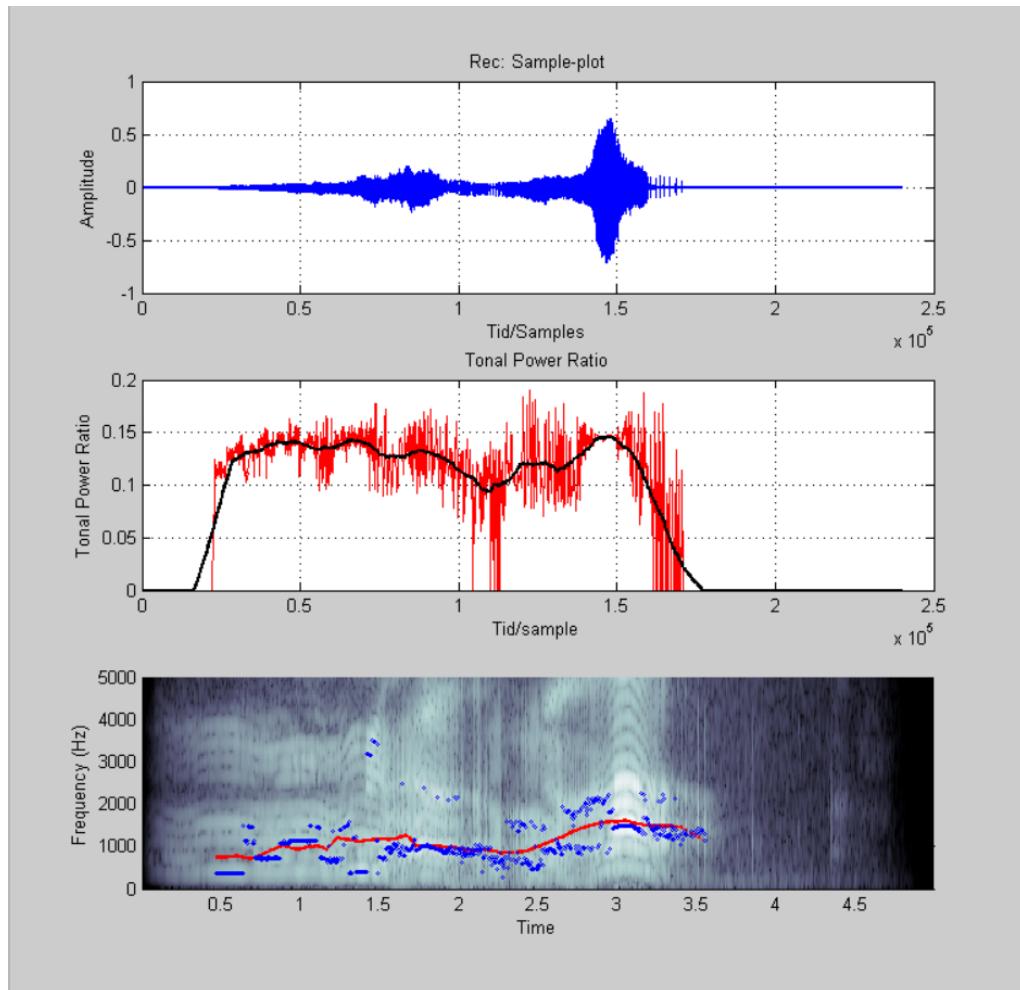
Karakteristisk ved højlydt babygråd er, at dele af Tonal Power Ratio (TPR) ligger over en værdi på 0.13 (se subplot 2). Herudover ses det på subplot 3 at de dominerende frekvenser typisk ligger mellem 700 og 3000 Hz (blå prikker). Midt disse værdier, ligger det dominerende frekvensindhold mellem 900 Hz og 2100 Hz (røde streger).



Figur 5.1. Analyseresultat af optagelsen Højlydt babygråd

Moderat babygråd

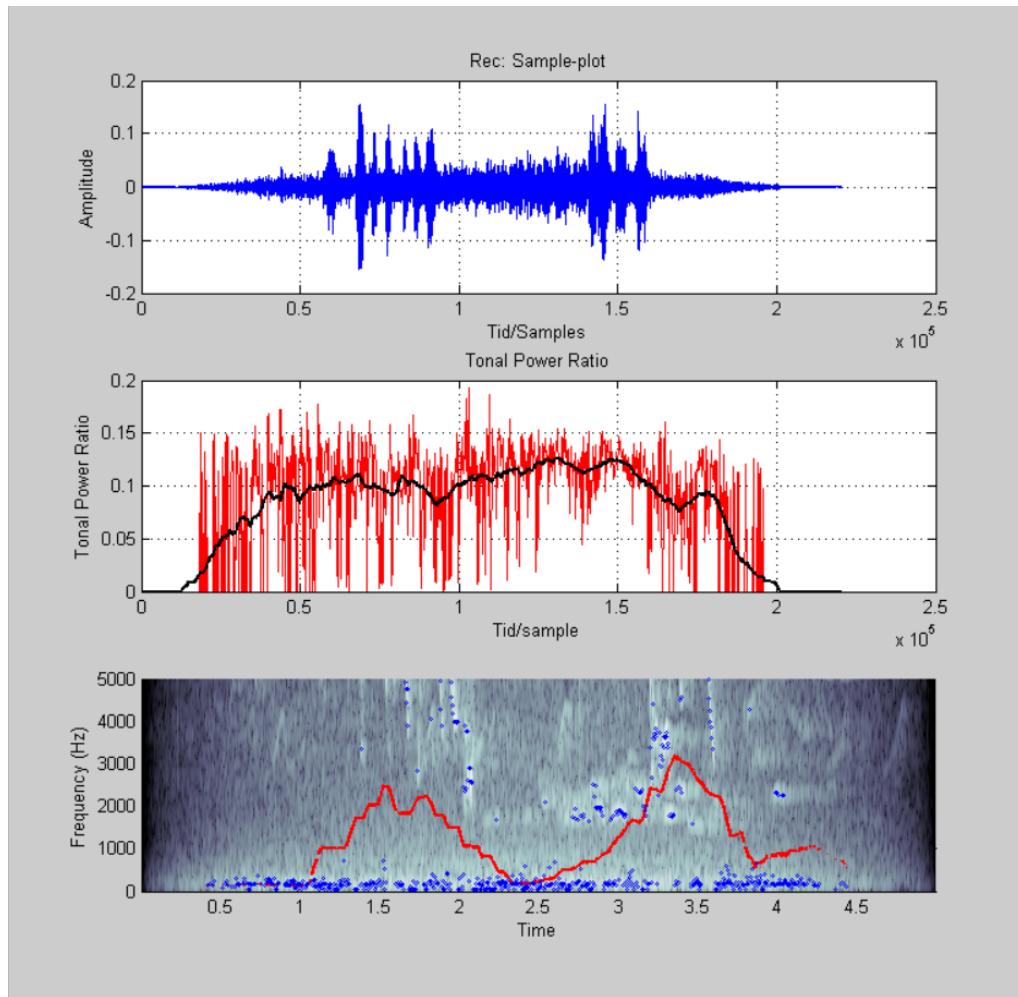
Karakteristisk ved optagelsen af moderat babygråd er, at store dele af TPR ligger over en værdi på 0.13 (subplot 2). Af subplot 3 fremgår det at de dominerende frekvenser typisk ligger mellem 500 og 3500 Hz (blå prikker). Midt mellem disse værdier, ligger det dominerende frekvensindhold mellem 900 Hz og 1700 Hz (røde streger).



Figur 5.2. Analyseresultat af optagelsen Moderat babygråd

Fuglefløjt

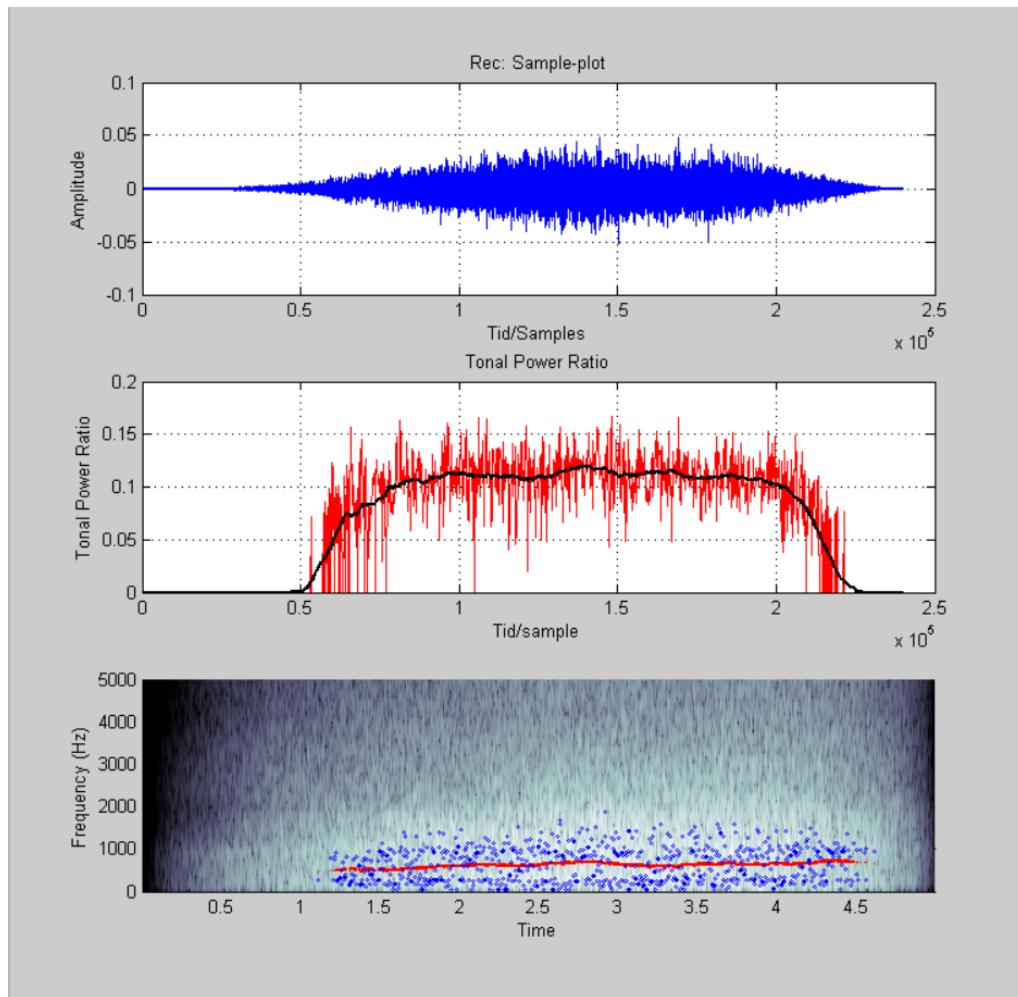
Karakteristik for fuglefløjt og omgivelsesstøj fra natur er, at TPR ligger på en værdi over 0.11, men under 0.13 (subplot 2). På spektrogrammet (subplot3) ses det at de dominerende frekvenser ligger meget spredt: det meste af tiden dominerer vind og vejr (ca. 100 Hz), og andre gange dominerer fuglefløjt med frekvenser fra 1800 Hz til 5000 Hz. Den midlere dominerende tone ligger mellem 35 Hz og 2500 Hz.



Figur 5.3. Analyseresultat af optagelsen Fuglefløjt

Trafikstøj

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



Figur 5.4. Analyseresultat af optagelsen Trafikstøj babygråd

Sammenligning af maximal TPR

Matlabs funktion til at finde maksimum-værdien for en array er benyttet til at finde den højeste værdi af TPR for de forskellige situationer. Resultaterne kan ses i tabellen herunder.

Situation	Maximal TPR
Højlydt babygråd	0.144
Moderat babygråd	0.146
Fuglefløjt	0.127
Trafikstøj	0.120

Babygråd producerer altså en højere TPR end typisk omgivelsesstøj, hvad enten det drejer sig om moderat eller højlyd gråd.

Sammenligning af dominerende frekvensindhold

Minimum- og maksimumfrekvens for analysen af dominerende toner er vist i tabellen herunder:

Situation	Min frekvens	Max frekvens
Højlydt babygråd	770 Hz	1932 Hz
Moderat babygråd	676 Hz	1613 Hz
Fuglefløjt	69 Hz	3120 Hz
Trafikstøj	38 Hz	710 Hz

Fuglefløjt har største båndbredde, med udfald mellem 69 Hz og 3120 Hz. Trafikstøjens dominerende frekvenser bevæger sig ikke over 710 Hz. Babygråd har sit dominante frekvensindhold mellem 600 og 2000 Hz. Trafikstøj har således lavere frekvensindhold end babygråd. Frekvensindholdet for fuglefløjt har en større båndbredde end babygråd og har sit indhold både over- og under båndbredden for babygråd.

5.1.5 Konklusion

Ved hjælp af *Tonal Power Ratio* er det altså muligt at kende forskel på babygråden og almindelige støjsignaler som trafikstøj og fuglefløft/natur.

Af *frekvensanalysen* ses det, at højlydt babygråd vil have dominant-toneindhold der ligger omrent 300 Hz højere end det for moderat babygråd. Forekommer dominant toneindhold (midlet) over 2500 Hz, antages tonen ikke at komme fra babyen.

Forskellen i frekvensindhold mellem højlydt og moderat gråd er dog ikke signifikant, og en *dB-måling* af grådens lydstyrke kunne evt være nødvendig for at skelne mellem højlydt- og moderat babygråd.

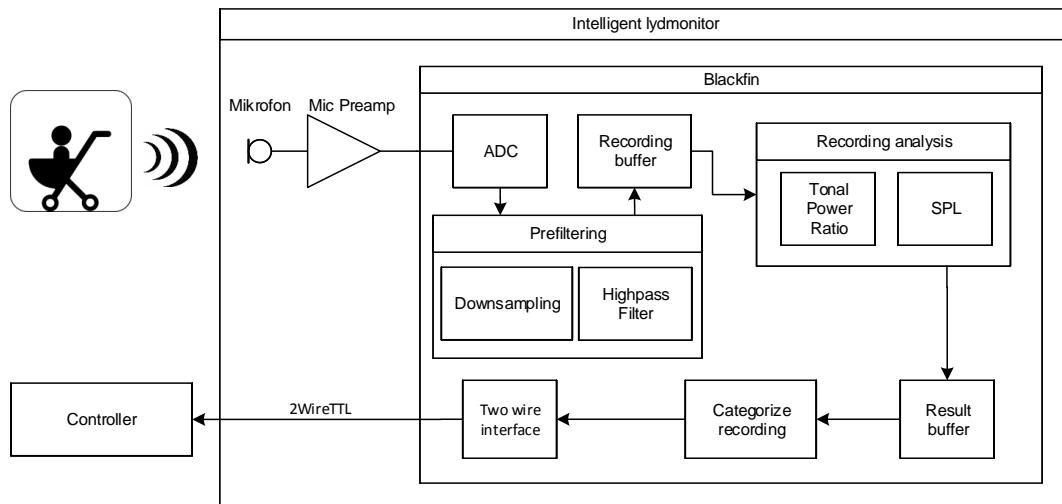
5.2 Systemarkitektur

I dette afsnit beskrives systemarkitekturen for den intelligente lydmonitor.

Overordnet virkemåde

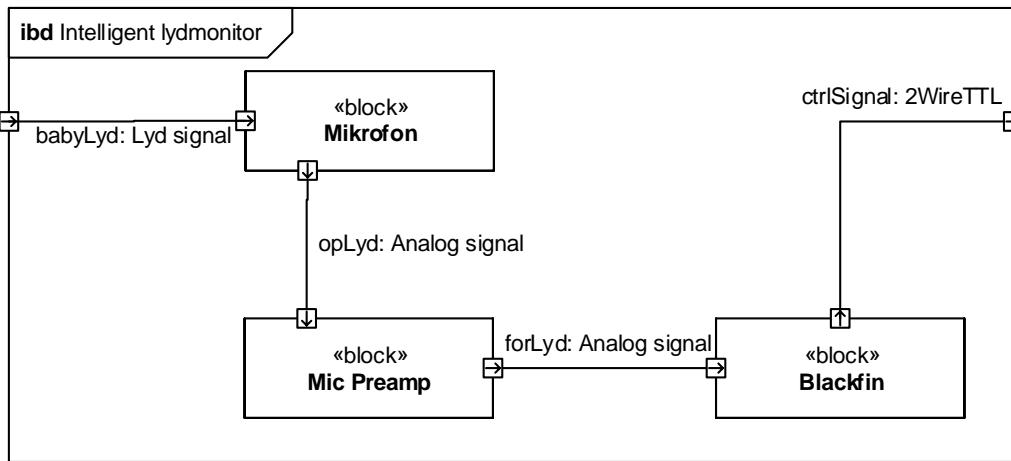
Overordnet skal den intelligente lydmonitor fungere som følger:

- 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 lavfiltreres og nedsamples før herefter at blive højfiltreret.
- Den filtrerede sample-sekvens i bufferen analyseres først for Tonal Power Ratio før herefter at blive analyseret for Sound Pressure Level (SPL). Resultatet heraf gemmes i endnu en buffer med analyseresultater.
- Disse analyseresultater kategoriseres som tre BABYCON-states, som beskrevet i 2.1
- Kategoriseringsresultatet sendes til Controller via Two wire interface



Figur 5.5. Overordnet virkemåde for Intelligent Lydmonitor

5.2.1 Hardware arkitektur



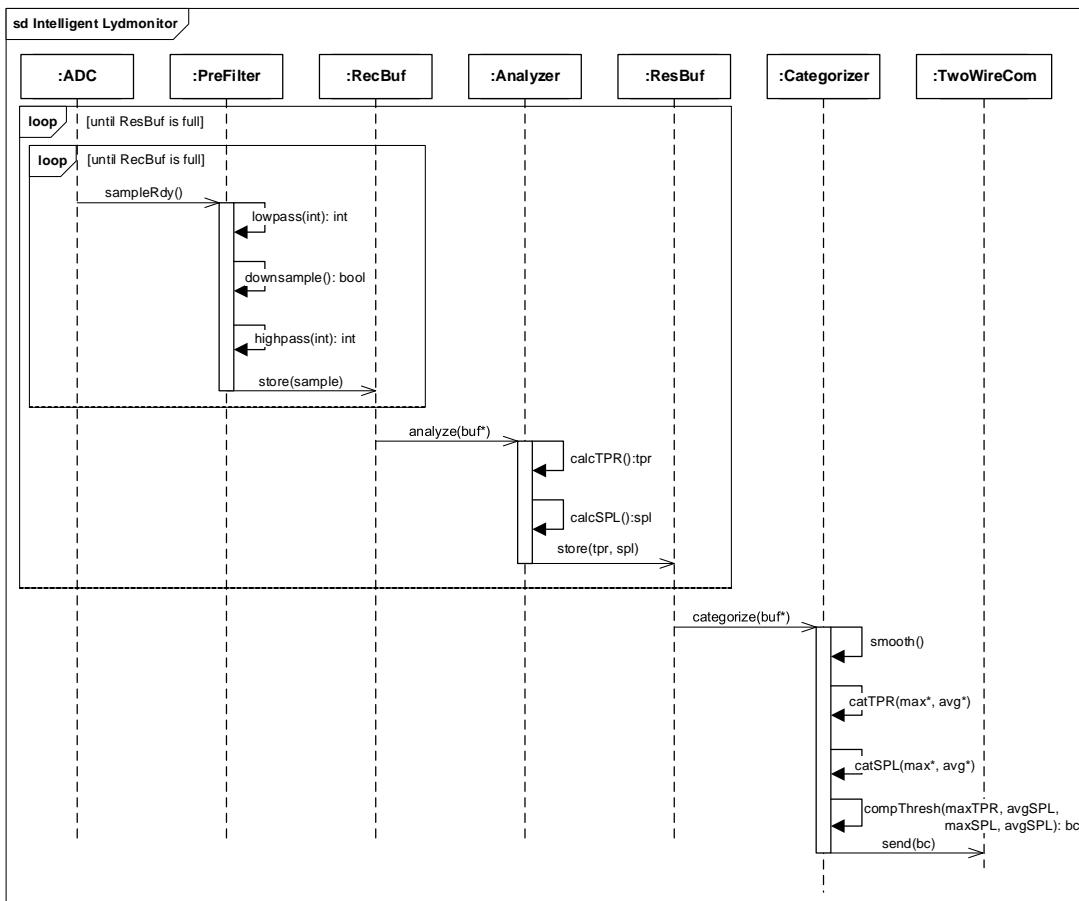
Figur 5.6. IBD for Intelligent Lydmonitor

Intelligent Lydmonitor består 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

Sekvensdiagram

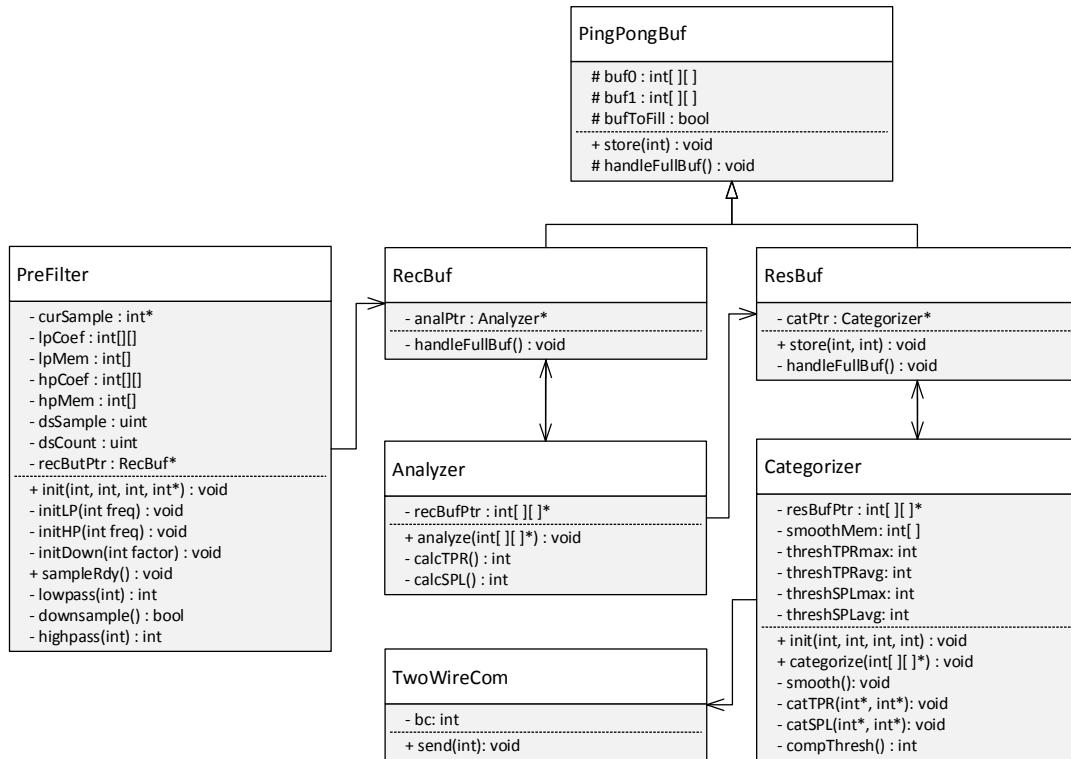


Figur 5.7. SD for Intelligent Lydmonitor

Som det ses i sekvensdiagrammet, er der opstillet nogle klasser for Intelligent Lydmonitor. Diagrammet viser hvordan- og i hvilken rækkefølge de forskellige klasser interagerer med hinanden.

PreFilter er den klasse som står for forfiltreringen af sample-sekvensen. PreFilter har tre funktioner, `lowpass()` som lavpasfiltrerer sample-sekvensen. `downsample()` som nedsamplet sekvensen. Herefter kaldes funktionen `highpass()` som højpasfiltrerer den nedsamplede sekvens. PreFilter kalder herefter RecBuf's funktion `store()` for at gemme samplen. **RecBuf** er en ping pong buffer hvor de diskrete lydsamples gemmes. Det er altså en buffer der indeholder to separate buffere, således at data fra den ene kan behandles mens data fyldes i den anden. Når RecBuf's ene buffer er fyldt kaldes Analyzers' funktion `bufRdy()`. **Analyzer** analyserer nu på data i RecBuf. Når Analyzer er færdig med at behandle data, gemmes analyseresultatet i **RecBuf** med funktionen `store()`. Når RecBuf er fyldt sendes besked til klassen **Categorizer** som herefter kategoriserer resultatet indenfor de tre BABYCON niveauer (1,2 og 3). Til slut sendes BABYCON niveauet til Controller via TwoWireCom.

Klassediagram



Figur 5.8. UML for Intelligent Lydmonitor

Klasser og deres tilhørende funktioner samt parametre er specifiseret i ovenstående UML for Intelligent Lydmonitor.

Funktionsbeskrivelse

PreFilter

PreFilter klassen indeholder funktionaliteten til forfiltrering og nedsampling af den diskrete sampling-sekvens.

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

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

Funktion	void initLP(int freqLP)
Beskrivelse	Initialiserer LP-filter.
Parametre	int lpFreq knækfrekvens for LP-filter
Returværdi	Ingen

Funktion	void initHP(int freqHP)
Beskrivelse	Initialiserer HP-filter.
Parametre	int hpFreq knækfrekvens for HP-filter
Returværdi	Ingen

Funktion	void initDown(int factor)
Beskrivelse	Initialiserer downsampling
Parametre	int factor faktor for nedSampling
Returværdi	Ingen

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

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

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

PingPongBuf

For at kunne opsamle data, mens at der bliver behandlet andet data, er det valgt at benytte pingpong-buffering. Dermed er det unødvendigt at kopiere bufferen når denne er fyldt. I stedet sendes til den klasse, som skal behandle data, en pointer til den fyldte buffer. Mens data behandles, kan nyt data fyldes i den anden buffer.

Funktion	void store(int sample)
Beskrivelse	Gemmer diskret sample i PingPongBuf. Hvis bufToFill er sat til 0 påfyldes data i attributten buf0. Ellers fyldes data i buf1.
Parametre	int sample: Diskret værdi som repræsenterer det momentane spændingsniveau på Blackfin 533 ADC indgang
Returværdi	Ingen

Funktion	void handleFullBuf(void) = 0
Beskrivelse	Virtuel funktion
Parametre	Ingen
Returværdi	Ingen

RecBuf

Afledt klasse af PingPongBuf.

Funktion	void handleFullBuf(void)
Beskrivelse	Kalder klassen Analyzers funktion bufRdy(int[][]*) med en pointer til den buffer som er fyldt. Herudover toggles bufToFill
Parametre	Ingen
Returværdi	Ingen

ResBuf

Afledt klasse af PingPongBuf.

Funktion	void handleFullbuf(void)
Beskrivelse	Kalder klassen Categorizers funktion bufRdy(int[][]*) med en pointer til den buffer som er fyldt. Herudover toggles bufToFill
Parametre	Ingen
Returværdi	Ingen

Analyzer

Klassen Analyzer indeholder analyse-funktionerne for beregning af Tonal Power Ratio og Sound Pressure level

Funktion	void analyze(int[][]* recBufPtr)
Beskrivelse	Gemmer pointeren til RecBuf kalder funktionerne calcTPR() og calcSPL()
Parametre	int[][]* recBufPtr, pointer til RecBuf.
Returværdi	Ingen

Funktion	int calcTPR(void)
Beskrivelse	Beregner Tonal Power Ratio
Parametre	Ingen
Returværdi	int, værdi af TPR

Funktion	int calcSPL(void)
Beskrivelse	Beregner Sound Pressure Level
Parametre	Ingen
Returværdi	int, værdi af SPL

Categorizer

Klassen Categorizer står for BABYCON kategorisering af resultaterne fra ResBuf

Funktion	void init(int threshTPRmax, int threshTPRavg, int threshSPLmax, int threshSPLavg)
Beskrivelse	Sætter threshholds for TPR og SPL
Parametre	int threshTPRmax, int threshTPRavg, int threshSPLmax, int threshSPLavg
Returværdi	Ingen

Funktion	void categorize(int[][]* resBufPtr)
Beskrivelse	Gemmer pointeren til ResBuf, kalder funktionerne smooth(), catTPR(),catSPL() og compThresh()
Parametre	int[][]* resBufPtr, pointer til ResBuf
Returværdi	Ingen

Funktion	void smooth(void)
Beskrivelse	Udglatter data-sekvensen gemt i ResBuf
Parametre	Ingen
Returværdi	Ingen

Funktion	void catTPR(int* maxTPR, int* avgTPR)
Beskrivelse	Finder maksimum og average Tonal Power Ratio værdi
Parametre	int* maxTPR pointer til maksimum TPR int* avgTPR pointer til average TPR
Returværdi	Ingen

Funktion	void catSPL(int* maxSPL, int* avgSPL)
Beskrivelse	Finder maksimum og average Sound Pressure Level værdi
Parametre	int* maxSPL pointer til maksimum SPL int* avgSPL pointer til average SPL
Returværdi	Ingen

Funktion	int compThresh(int maxTPR, int avgTPR, int maxSPL, avgSPL)
Beskrivelse	Sammenligner modtagne værdier med foruddefinerede thresholds, for at bestemme BABYCON-niveau
Parametre	int maxTPR, int avgTPR, int maxSPL, avgSPL
Returværdi	int, BABYCON-niveau mellem 1-3

TwoWireCom

Klassen TwoWireCom står for at sende BABYCON-niveau til Controller jf. 3.3.1

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

5.3 Design

5.3.1 Mikrofon preamp

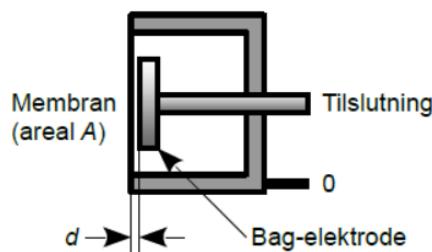
Mikrofon

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

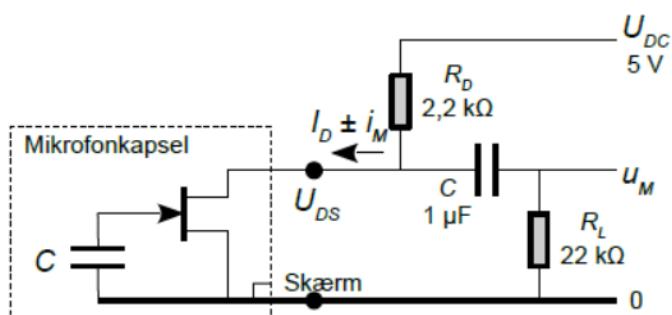
Se datablad (INDSÆT REFERENCE TIL Pro-MCE-100-datasheet) for komplet specifikation.

En elektret mikrofon er en videreudvikling af kondensatormikrofonen, hvor bagelektroden har en "indefrosset" ladning i materialet. Kapaciteten i mikrofonen (ca. 10 pF) ændres ved trykvariationer idet afstanden mellem membranen og bagelektroden varieres.



Figur 5.9. Tegning af elektrekrostatisk mikrofon. Kilde: Analogteknik

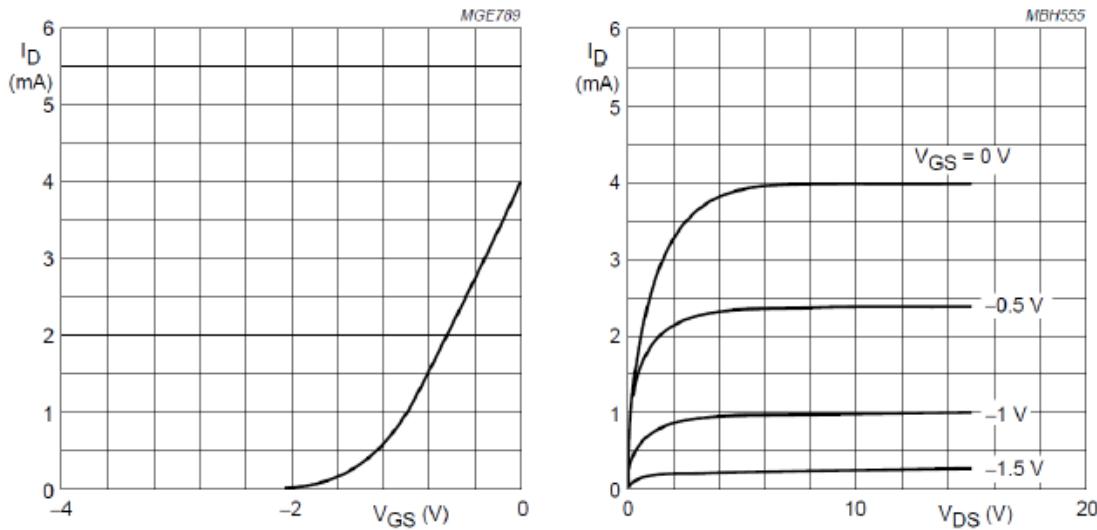
Dog har kablingen mellem mikrofonen og forstærkeren en betydelig kapacitet, så mikrofonpakken er implementeret med en indbygget JFET som buffer, som det ses af figuren herunder:



Figur 5.10. Elektret forsyningskredsløb. Kilde: Analogteknik

Forsyningen til mikrofonen trækkes fra Blackfin 533, 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 5.11. Karakteristik for JFET'en BF245A. Kilde: Analogteknik

I punktet på højre side af kondensatoren C, vil strømmen 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 $S = 5,6 \text{ mV/Pa}$, og i_M 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

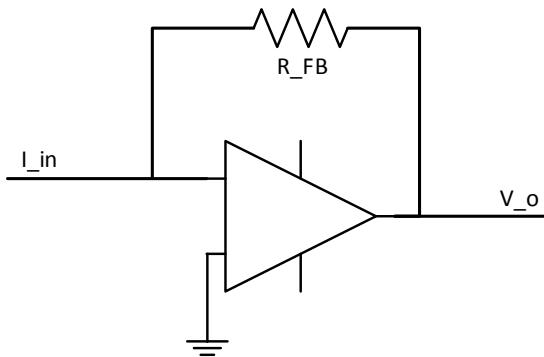
Blackfin ADC'en tager et line-level input på +/- 1,65 V. Det ønskes at udnytte det maksimale dynamiske område uden at lade signalet klippe. Det regnes med at mikrofonen ikke udsættes for mere end 2 Pa ved almindelig brug. Den maksimale strøm-amplitude bliver derfor:

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

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.



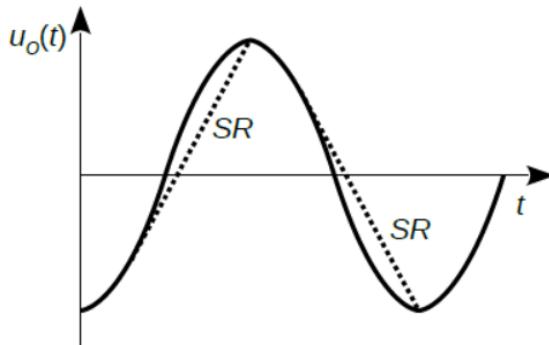
Figur 5.12. Transimpedansforstærker: $V_o = R_{FB} \cdot I_{in}$

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 5.13. 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:

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

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 $u_M < 20\text{mV}$.

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

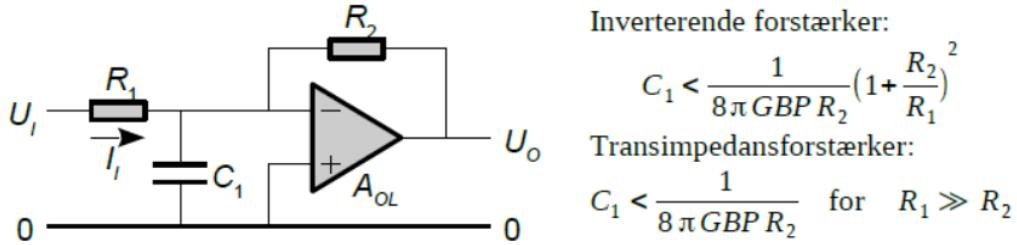
Dette krav er altså opfyldt.

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 OpAms 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.

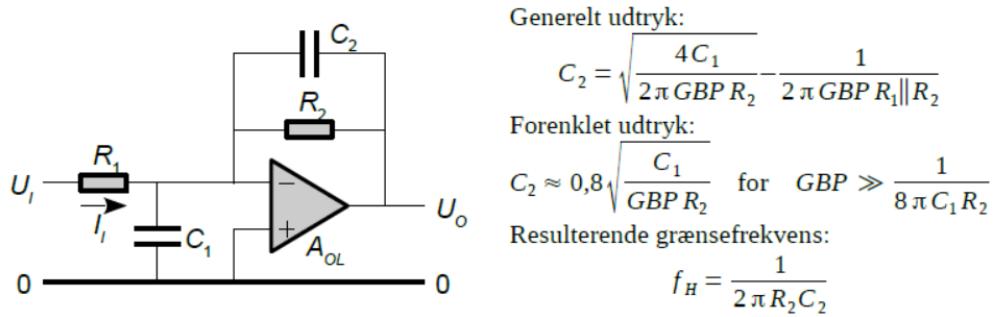


Figur 5.14. OpAmp med kapacitiv belastning på indgangen. Kilde: Analogteknik

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 tilbagekobling:

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

Det er altså nødvendigt at sætte en kondensator i tilbagekoblingen, som illustreret i 5.15.



Figur 5.15. 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 Implementering

5.5 Modultest

Vuggesystem 6

6.1 Systemarkitektur

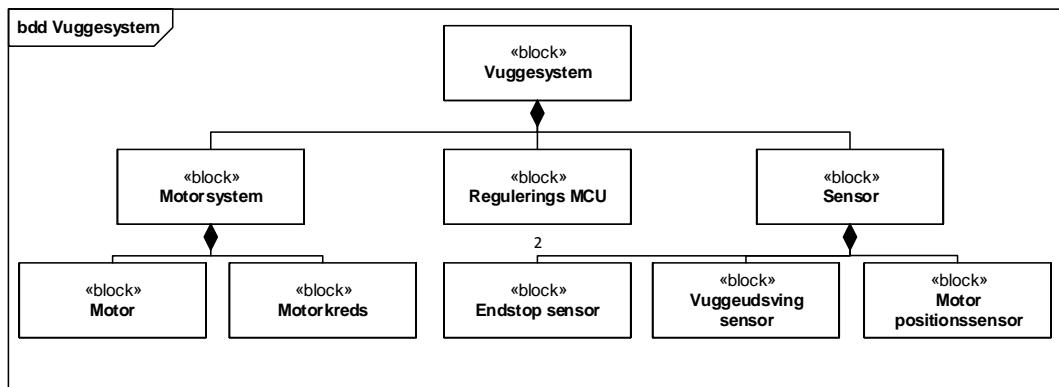
I dette afsnit beskrives systemarkitekturen for vuggesystemet.

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 vinkeludsvinget overholdes
- Kommunikation mellem Regulerings MCU og Controlleren foregår via I2C protokollen

6.1.1 Hardware arkitektur



Figur 6.1. BDD for Vuggesystem

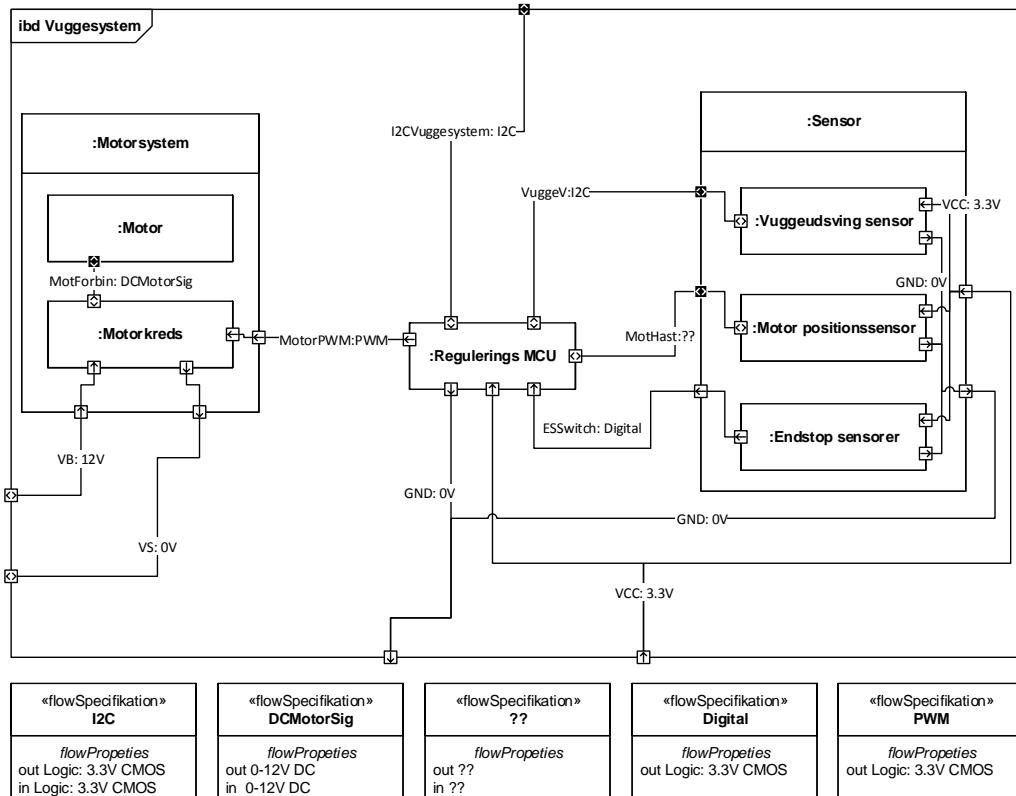
Vuggesystemet består af tre dele

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.

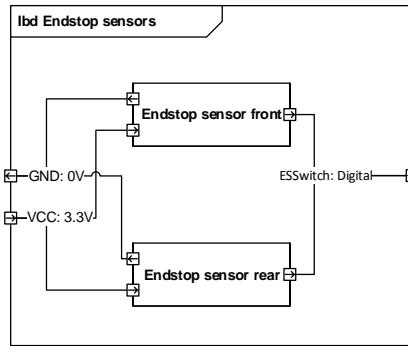
Regulerings MCU block: Styringsenheden for vuggesystemet. Denne sørger for reguleringen af vuggesystemet samt kommunikationen til og fra controlleren.

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

Følgende beskriver vuggesystemets kobling og grænseflade.

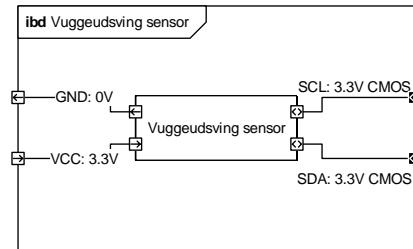


Figur 6.2. IBD for Vuggesystem



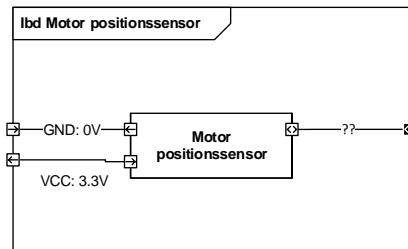
Figur 6.3. IBD for Endstop sensorer

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 frekvenssignal relativ til motorens nuværende hastighed.

6.1.2 Grænsefladebeskrivelse

Herunder findes en beskrivelse af de interne forbindelser.

Signalbeskrivelse

Tabel 6.1: Signalbeskrivelse

Signal	Type	Kommentar
VB	12V DC	Dette signal kommer direkte fra systemets batteri.
VS	0V DC	Dette er batteriforsyningens retur
VCC	3.3V DC	Dette er forsyningen til PSoC og andet logik.
GND	0V stel	Stelforbindelse til PSoC og andet logik
MotorPWM	PWM	PWM som styrer motor spændingen ved VB samt retningen for motoren. Fra 0-50%duty-cycle genererer en negativ motorspænding, fra 50-100 %duty-cycle genererer en positiv motorspænding. PWM signalet 0-3.3V CMOS med f = [20kHz]
VuggeV	I2C	Angiver vinklen af vuggen relativt til tyngdefeltet.
MotHast	??	Angiver hastigheden af motoren
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.

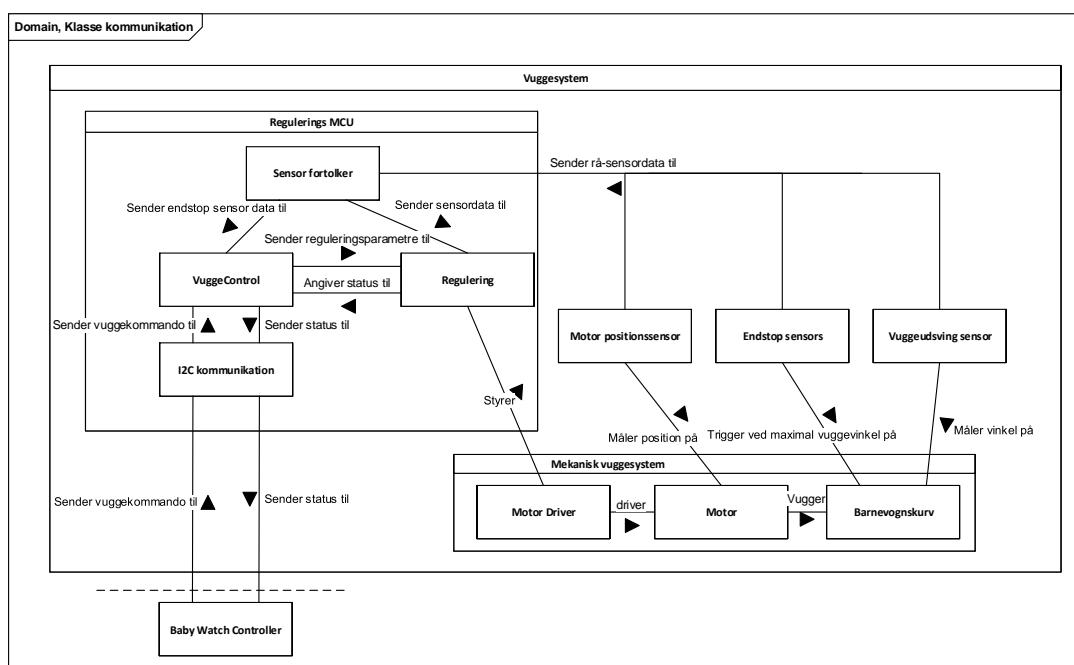
Softwarearkitekturen for Vuggesystemet består af:

- 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

Det er valgt at lave en applikationsmodel for modulerne/modulgrupperne internt frem for direkte Use Case baseret. Applikationsmodellen indeholder således funktionaliteter for op til flere Use Cases, men 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.

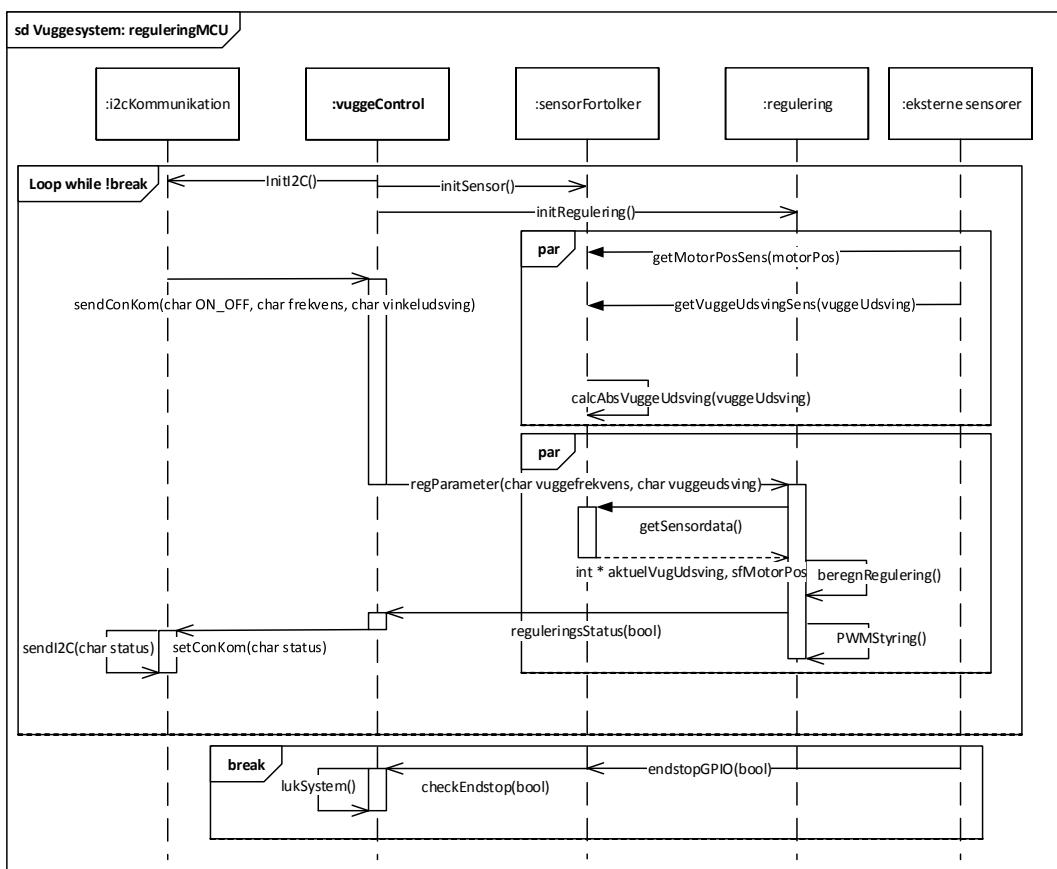
Klasseidentifikation

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 Watchs 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

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

- Regulerings MCU - klassenavn: **vuggeControl** er Vuggesystems control-klasse.

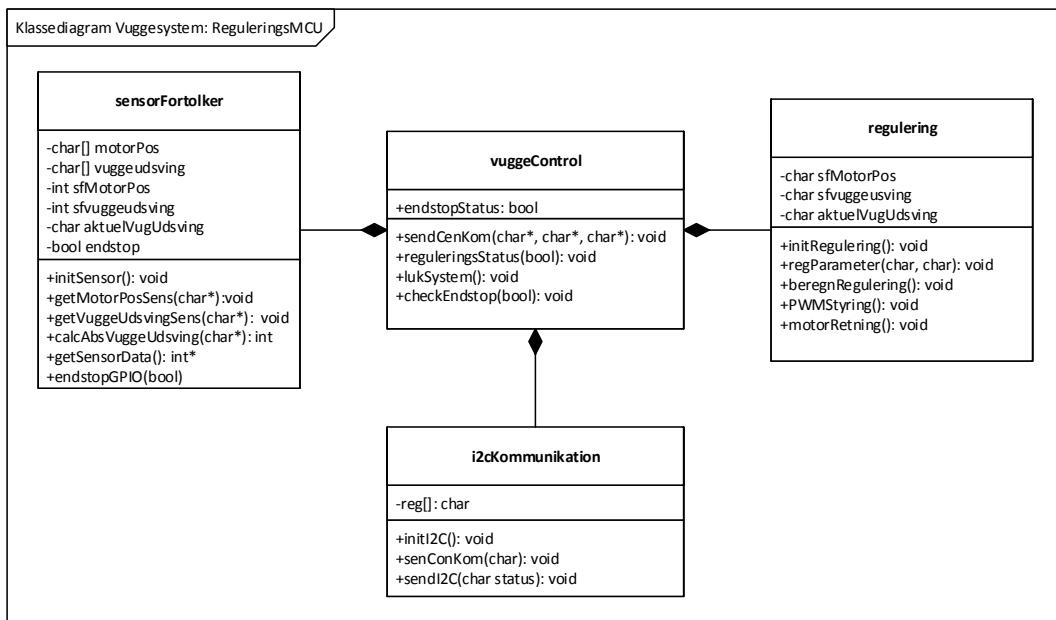
- I2C kommunikation - klassenavn: **i2cKommunikation** står for I2C interfacet ud til Baby Watchs Contoller.
- Motorregulering - klassenavn: **regulering** står for beregningerne til reguleringen af motoren der trækker vugningen af barnevognenskurv
- Sensor Fortolker - klassenavn: **sensorFortolker**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

6.2 Design

6.2.1 Software design

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

Klassediagram



Figur 6.8. Klassediagram for Regulerings MCU

Funktionsbeskrivelser

vuggeControl

Funktion	IKKE FÆRDIG
Beskrivelse	Gemmer diskret sample i RecordingBuffer
Parametre	int Sample: Diskret værdi som repræsenterer det momentane spændingsniveau på Blackfin 533 ADC indgang
Returværdi	Ingen

i2cKommunikation

Funktion	void sendConKom(char* reg[2], char* reg[3], char* reg[4])
Beskrivelse	Sender tre char pointere der peger på registrene reg[2], reg[3] og reg[4] til vuggeControl klassen
Parametre	char * reg[2]: peger på ????, char * reg[3]: peger på ??? og char * reg[4]: peger på ????
Returværdi	Ingen

reguelering

Funktion	IKKE FÆRDIG
Beskrivelse	Gemmer diskret sample i RecordingBuffer
Parametre	int Sample: Diskret værdi som repræsenterer det momentane spændingsniveau på Blackfin 533 ADC indgang
Returværdi	Ingen

sensorFortolker

Funktion	char getMotorPosSens(void)
Beskrivelse	Returnerer den nyeste motorposition fra motorpositionssensoren.
Parametre	Ingen
Returværdi	char: Den nyeste position angivet i værdi fra 0-255

Funktion	char getVuggeUdsving(void)
Beskrivelse	Returnerer den nyeste måling fra Vuggeudsringssensoren.
Parametre	Ingen
Returværdi	char: Den nyeste måling fra Vuggeudsringssensoren angivet i ??? accelerometer og gyroskop ???

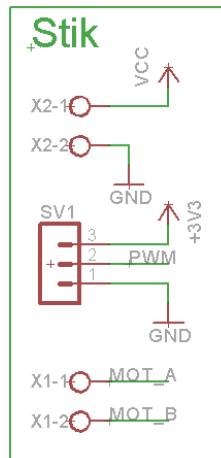
Funktion	int calcAbsVuggeUdsving(char udsving)
Beskrivelse	Udfra acceletermeter- og gyroskopmålingerne beregnes og returneres den absolutte vinkel for barnevognskurven.
Parametre	char: Udsving er målinger fra Vuggeudsringssensoren
Returværdi	int: Den absolutte vinkel for barnevognskurven

Funktion	void checkEndstop(bool)
Beskrivelse	Tjekker endstop sensorernes værdi.
Parametre	Ingen
Returværdi	wav

6.3 Implementering

6.3.1 Motorstyringskreds

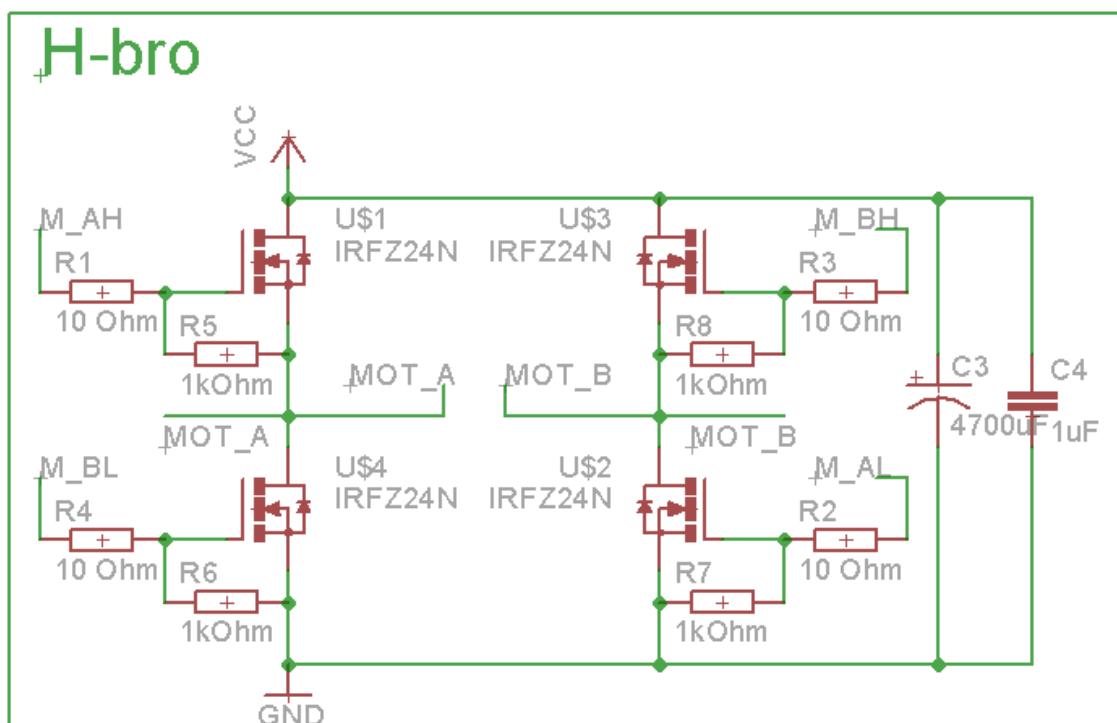
I denne sektion beskrives det kredsløb der benyttes til at styre ankerspændingen for vugge-systemets motor. Kredsløbet modtager et PWM signal fra controlleren som styrer ankerspændingen. Udenfor styresignalet og motor terminalerne, er kredsløbet koblet til systemets 12V forsyning(VCC i diagram) hvorfra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



Figur 6.9. Forbindelser til motorkreds

H-bro

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



Figur 6.10. 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.

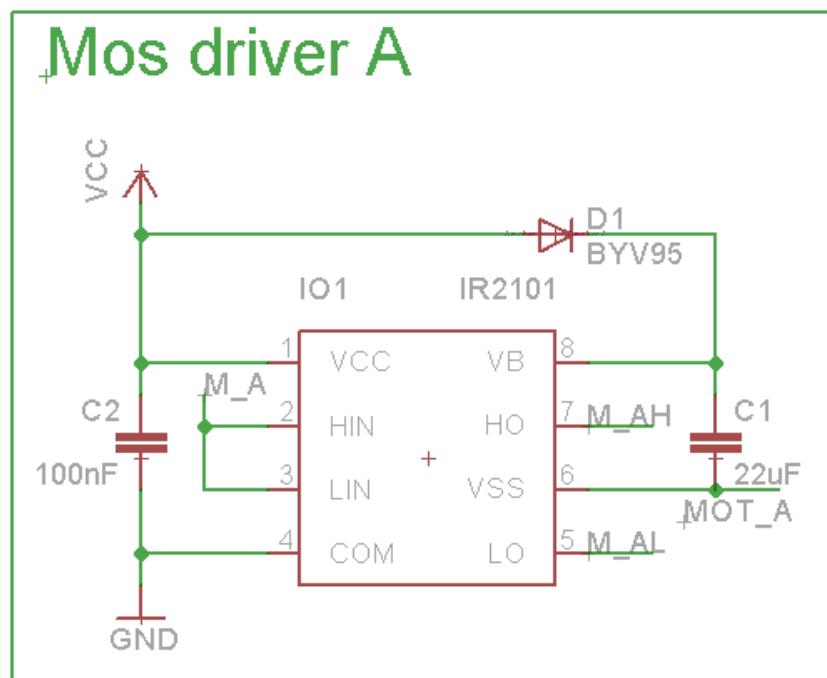
Signalerne M_AH, M_AL, M_BH, og M_BL udgør således H-broens styresignaler til

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 6.11. 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.

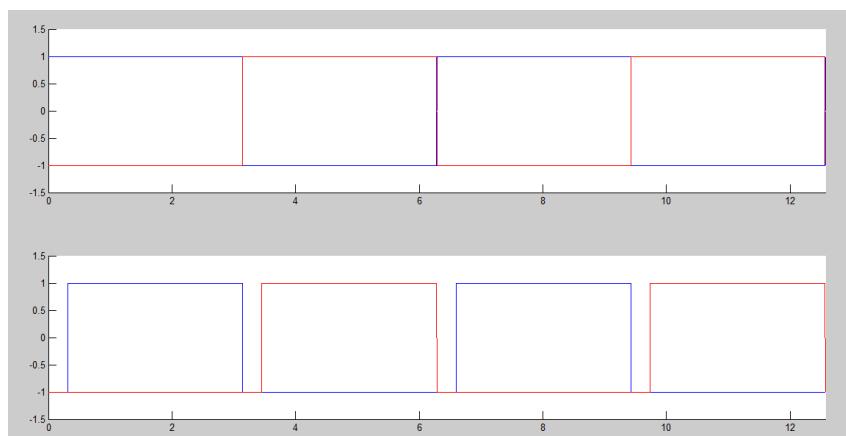
Driverens input er et digitalt input, med en fast overgang fra lav til høj mellem 0.8 og 3 V, og kan således styres med 3.3V logik uafhængigt af hvilken forsyningsspænding driveren er koblet med.

Kondensatoren C2 hjælper til at afkoble driveren fra forsyningen.

Styresignal

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

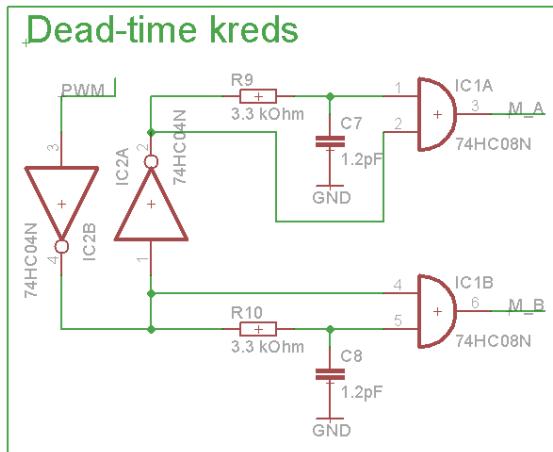
Da MOSFET transistorne tager en hvis tid om at slukke, skal der introduceres en dødtid i styresignalerne, således at signalerne alle går kortvarigt til 0 før de skifter. Dette sikrer at to transistorer fra hver deres side af broen ikke er tændt samtidig, hvilket ville resultere i en kortslutning fra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



Figur 6.12. 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 transistorne i broen at slukke helt. På denne måde sikres det at transistorne 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 6.13. Deadband kredsløb

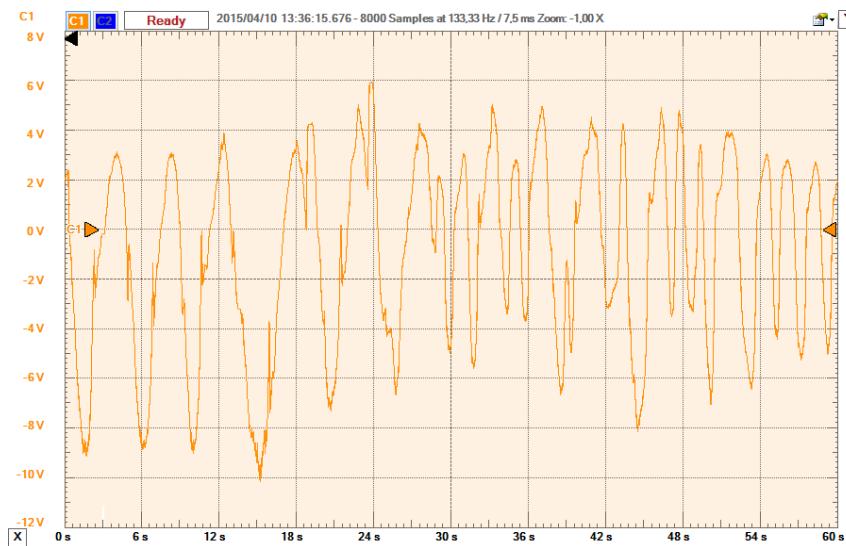
Den første inverter fungerer som en buffer for at sikre at indgangssignalet møder en høj impedans, mens den anden inverter skaber et ikke inverteret udtag til PWM signalet. Modstanden er dimensioneret således at der højst trækkes en milliampere fra inverteren, og kondensatoren kan så dimensioneres til 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:

6.3.2 strømforbrug

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

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



Figur 6.14. Test af strømforbrug for motor

6.4 Modultest

Integrationstest 7

Accepttestspezifikation

8

Use case 1

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

Use case 2

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

Use case 3

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

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

...fortsat fra forrige side

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

Use case 4

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

...fortsat fra forrige side

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

Ikke-funktionelle krav

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

...fortsat fra forrige side

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

Bilag (CD-indhold)

9
