Indhold

[Projektformulering 4](#_Toc401586453)

[Kravspecifikation 5](#_Toc401586454)

[1.1 Aktører 5](#_Toc401586455)

[1.2 Aktørbeskrivelse 5](#_Toc401586456)

[1.2.1 Bruger 5](#_Toc401586457)

[1.2.2 MIDI-modtager 5](#_Toc401586458)

[1.2.3 Højtalersystem 5](#_Toc401586459)

[1.3 Termliste 6](#_Toc401586460)

[1.3.1 Body 6](#_Toc401586461)

[1.3.2 Rock 6](#_Toc401586462)

[1.3.3 Lydpakke: 6](#_Toc401586463)

[1.3.4 MIDI-parameter 6](#_Toc401586464)

[1.3.5 Lydmodul 6](#_Toc401586465)

[1.3.6 Mapping scheme 6](#_Toc401586466)

[1.3.7 Sensorkonfiguration 6](#_Toc401586467)

[1.3.8 Buffer 6](#_Toc401586468)

[1.3.9 Preset 6](#_Toc401586469)

[1.4 Use cases 7](#_Toc401586470)

[1.4.1 Forbind Body og Rock 7](#_Toc401586471)

[1.4.2 Installér lydpakker 8](#_Toc401586472)

[1.4.3 Konfigurer sensorer 9](#_Toc401586473)

[1.4.4 Konfigurer preset 11](#_Toc401586474)

[1.4.5 Vælg preset 12](#_Toc401586475)

[1.4.6 Indsamle sensordata 13](#_Toc401586476)

[1.4.7 Generér MIDI 14](#_Toc401586477)

[1.4.8 Afspil lyd 15](#_Toc401586478)

[Systemarkitektur Hardware 16](#_Toc401586479)

[Overordnet arkitektur 16](#_Toc401586480)

[Domæne model BodyRock3000 17](#_Toc401586481)

[System tegning 17](#_Toc401586482)

[BDD diagram BodyRock3000 17](#_Toc401586483)

[Rock 18](#_Toc401586484)

[Body 18](#_Toc401586485)

[IBD Generisk sensorenhed 20](#_Toc401586486)

[Allokering for systemet 21](#_Toc401586487)

[Systemarkitektur Software 22](#_Toc401586488)

[Software BDD Rock 22](#_Toc401586489)

[Software IBD Rock 23](#_Toc401586490)

[Software BDD Body 24](#_Toc401586491)

[Klassediagram Sensor-to-Midi 24](#_Toc401586492)

[Klassediagram GUI 25](#_Toc401586493)

[Klassediagram Bluetooth-Rock 25](#_Toc401586494)

[Klassediagram Bluetooth-body 26](#_Toc401586495)

[Klassediagram GUI 27](#_Toc401586496)

# Projektformulering



Opgaven i dette projekt er at udvikle et intuitivt musikinstrument, baseret på kropslige bevægelser, der kan implementeres på diverse synthesizere, drummachines og DAW's (Digital Audio Workstation).

Instrumentet skal kunne afspille lydsamples, samt generere MIDI-tone- og CC-signaler, på baggrund af data fra accelerometer-, gyroskop-, proksimitets- og taktile trykmålinger.

Dette medfører at det færdige produkt skal kunne opfange:

* Accelerationer
* Tilt
* Afstande
* Taktile tryk

Og på baggrund af disse:

* Afspille samples
* Generere MIDI-signaler

Systemet består af to dele, som kommunikerer trådløst:

1. En bærbar enhed med sensorer til aflæsning af kropslige bevægelser og positioner
2. En stationær enhed til processering af aflæst rådata fra sensorer, afspilning af samples fra downloadede lydpakker, generering af MIDI-signaler og styring af systemindstillinger

Visionen bag projektet er at skabe et nyt udtryks-medie for musikere, foruden at inkludere hidtil ulærte musiktalenter i den kreative og musikalske glæde, systemet vil medføre.

Systemets fleksibilitet og muligheder for udvidelse, samt den lette integration med eksisterende synthesizer- og sample-systemer, gør det til en kærkommen udvidelse af repertoiret for midler til musikalsk udtryk.

# **Figur 1** Use case-diagram over BodyRock3000 Kravspecifikation

## Aktører

På figuren til højre ses use case-diagrammet med alle aktører og deres forhold til systemet BodyRock3000

## Aktørbeskrivelse

### Bruger

|  |  |
| --- | --- |
| **Aktørnavn:** | **Bruger** |
| **Type:** | Primær |
| **Beskrivelse:** | Ønsker at benytte systemet, og indstille diverse konfigurationer for systemet ved hjælp af Rock-enheden. Herudover kan Bruger installere lydfiler |

### MIDI-modtager

|  |  |
| --- | --- |
| **Aktørnavn:** | **MIDI-modtager** |
| **Type:** | Sekundær |
| **Beskrivelse:** | Ekstern MIDI-enhed som overholder MIDI-standarden[[1]](#footnote-1) |

### Højtalersystem

|  |  |
| --- | --- |
| **Aktørnavn:** | **Højtalersystem** |
| **Type:** | Sekundær |
| **Beskrivelse:** | Det eksterne højtalersystem modtager lydsignal fra audio-outputtet, og afspiller den ønskede lyd |

## Termliste

### Body

Body er den bærbare enhed for systemet BodyRock3000. Den består af en controller-enhed, som kan sende sensordata til den stationære enhed Rock. På controller-enheden kan der tilsluttes et vilkårligt antal sensorer, der kan foretage accelerometer-, gyroskop-, proksimitets- og taktile trykmålinger. Da det er et krav at anvende PSoC4-boardet i projektet, vil denne blive brugt som Body-enhed.

### Rock

Rock er den stationære enhed for systemet BodyRock3000. Denne enhed modtager sensordata fra Body-enheden via trådløs kommunikation. Sensordataene konverteres herefter til MIDI, som fastlægger lydoutputtet. Det er ligeledes et krav at anvende DevKit8000 i projektet, derfor vil DevKittet blive brugt som Rock-enhed.

### BodyRock3000

BodyRock3000 er en samlet betegnelse for systemet. Denne bruges bl.a. som initiator ved Use Case 6, og indikerer at Use Casen startes af interne processer op systemet.

### Lydpakke:

Bestemmer hvilke lyde der skal syntetiseres, på baggrund af de modtagne MIDI-signaler

### MIDI-parameter

Der er generelt tre typer MIDI-parametre:

* **Note**: Angiver hvilken tone, der skal afspilles i intervallet [0-127]
* **Control Change (CC):** Angiver hvilken parameter, der ønskes ændret i intervallet [0-127]
* **Program Change**: Angiver hvilket preset, der ønskes i intervallet [0-127]

### Lydmodul

Et digitalt datamodul, der agerer som bindeled mellem de valgte lydpakker og MIDI-parameter-inputs, således at den korrekte lyd afspilles analogt.

### Mapping scheme

En agenda for hvorledes et givet sensor-input mappes til en given MIDI-parameter

### Sensorkonfiguration

En sensorkonfiguration indeholder:

* ID for en specifik sensor-enhed
* MIDI-parameter
* Mapping scheme
* Lydpakke

### Buffer

Et midlertidigt hukommelseslager mellem to moduler i BodyRock3000.

### Preset

Et sæt af sensorkonfigurationer, der simultant[[2]](#footnote-2) kan indlæses på systemet

## Use cases

For alle use cases gælder det, at i enhver undermenu, findes der altid mulighed for at annullere og vende tilbage til hovedmenuen.

### Forbind Body og Rock

|  |  |
| --- | --- |
| **Navn** | **Forbind Body og Rock** |
| **Use Case ID** | 1 |
| **Samtidige forekomster** | 1 (inklusiv denne) |
| **Primær aktør** | Bruger |
| **Sekundær aktør** | Ingen |
| **Initiator** | Bruger |
| **Prækondition** | Bruger har adgang til Body og Rock |
| **Formål** | Body og Rock er forbundne og kan anvendes |
| **Postkondition** | Body og Rock er klar til brug |

|  |
| --- |
| **Hovedscenarie** |
| 1. Bruger tænder Body og Rock 2. Systemet viser hovedmenuen på Rocks display 3. Bruger vælger ”Forbind enhed” på Rocks touch-displayet 4. Rock søger efter Body 5. Bruger vælger Body på touch-displayet   [Undtagelse 5.a: Rock kunne ikke finde en Body-enhed]   1. Systemet forsøger at oprette forbindelse mellem Body og Rock 2. Systemet giver besked om godkendt forbindelse på displayet   [Undtagelse 7.a: Enhederne kunne ikke forbindes]   1. Systemet returnerer til hovedmenu, og UC1 afsluttes |

|  |
| --- |
| **Undtagelser** |
| [Undtagelse 5.a: Rock kunne ikke finde en Body-enhed]   1. 1. Systemet giver besked om at der ikke er nogen Body-enheder   2. Bruger slukker enhederne, og der fortsættes fra punkt 1 i hovedscenariet (ved gentagne fejl kontaktes kundeservice)  [Undtagelse 7.a: Enhederne kunne ikke forbindes]   1. 1. Systemet giver besked om ikke oprettet forbindelse på displayet 2. 2. Bruger fejlfinder på enhederne, og der fortsættes fra punkt 2 i hovedscenariet |

### Installér lydpakker

|  |  |
| --- | --- |
| **Navn** | **Installér lydpakker** |
| **Use Case ID** | 2 |
| **Samtidige forekomster** | 1 (inklusiv denne) |
| **Primær aktør** | Bruger |
| **Sekundær aktør** | Ingen |
| **Initiator** | Bruger |
| **Prækondition** | Der er fysisk forbundet en USB-hukommelsesenhed til Rock |
| **Formål** | Der installeres en lydpakke |
| **Postkondition** | En lydpakke er installeret |

|  |
| --- |
| **Hovedscenarie** |
| 1. Bruger vælger ”Installér lydpakke” i Rocks hovedmenu 2. Bruger vælger ”Importer lydpakke” i menu   [Undtagelse 2.a: Bruger vælger ”Slet lydpakke”]   1. System opretter forbindelse til USB-hukommelsesenhed   [Undtagelse 3.a: Der kan ikke oprettes forbindelse]   1. Bruger vælger lydpakke fra liste over kildens lydpakker 2. System installerer lydpakke, og UC2 afsluttes |

|  |
| --- |
| **Undtagelser** |
| [Undtagelse 2.a: Bruger vælger ”Slet lydpakke”]  1. Bruger vælger lydpakke fra liste over systemets lydpakker  2. Bruger bekræfter sletning af lydpakke  3. System sletter lydpakke, og UC2 afsluttes  [Undtagelse 3.a: Der kan ikke oprettes forbindelse]   1. 1. Der vises en fejlmeddelelse, og UC2 afsluttes |

### Konfigurer sensorer

|  |  |
| --- | --- |
| **Navn** | **Konfigurer sensorer** |
| **Use Case ID** | 3 |
| **Samtidige forekomster** | 1 (inklusiv denne) |
| **Primær aktør** | Bruger |
| **Sekundær aktør** | Ingen |
| **Initiator** | Bruger |
| **Prækondition** | Rock er tændt, og mindst én lydpakke er installeret |
| **Formål** | Der oprettes, nedlægges eller ændres i en sensorkonfiguration |
| **Postkondition** | En sensorkonfiguration er oprettet, nedlagt eller ændret |

|  |
| --- |
| **Hovedscenarie** |
| 1. Bruger vælger ”Konfigurer sensor” i Rocks hovedmenu 2. Bruger vælger ”Opret ny” i menuen   [Undtagelse 2.a: Bruger vælger ”Rediger eksisterende”]  [Undtagelse 2.b: Bruger vælger ”Slet”]   1. Bruger indtaster konfigurationens navn   [Undtagelse 3.a: Det indtastede navn er optaget]   1. Bruger vælger sensor på liste over sensorer 2. Bruger vælger MIDI-parameter på liste over MIDI-parametre 3. Bruger vælger mapping scheme på liste over mapping schemes 4. Bruger vælger lydpakke på liste over installerede lydpakker 5. Bruger vælger at tilføje sensor til et eller flere presets på liste over eksisterende presets 6. Bruger præsenteres for sine opsætningsvalg, og UC3 afsluttes |

|  |
| --- |
| **Undtagelser** |
| [Undtagelse 2.a: Bruger vælger ”Rediger eksisterende”]  1. Bruger vælger sensorkonfiguration fra liste over sensorkonfigurationer  2. Der fortsættes fra punkt 4 i hovedscenariet, hvor alle menuer herefter er udfyldt med  data fra den valgte opsætning  [Undtagelse 2.b: Bruger vælger ”Slet”]  1. Bruger vælger sensorkonfiguration fra liste over sensorkonfigurationer  2. Bruger bekræfter valg   1. 3. Sensorkonfiguration slettes, og UC3 afsluttes   [Undtagelse 3.a: Det indtastede navn er optaget]  1. Systemet viser fejlmeddelelse og der fortsættes fra hovedscenariets punkt 3 |

### Konfigurer preset

|  |  |
| --- | --- |
| **Navn** | **Konfigurer preset** |
| **Use Case ID** | 4 |
| **Samtidige forekomster** | 1 (inklusiv denne) |
| **Primær aktør** | Bruger |
| **Sekundær aktør** | Ingen |
| **Initiator** | Bruger |
| **Prækondition** | Der findes mindst én sensorkonfiguration i systemet |
| **Formål** | At oprette, redigere eller slette et preset |
| **Postkondition** | Der er foretaget en ændring i et preset |

|  |
| --- |
| **Hovedscenarie** |
| 1. Bruger vælger ”Konfigurer preset” i Rocks hovedmenu 2. Bruger vælger ”Opret nyt” i menuen   [Undtagelse 2.a: Bruger vælger ”Rediger eksisterende”]  [Undtagelse 2.b: Bruger vælger ”Slet”]   1. Bruger indtaster ønsket navn på preset 2. Bruger markerer de ønskede konfigurationer på liste over sensorkonfiguration, og tilføjer disse til det nuværende preset 3. Bruger trykker på ”Fortsæt” 4. Systemet præsenterer de valgte indstillinger på displayet, og anmoder om bekræftelse 5. Bruger accepterer ændringer, og UC4 afsluttes   [Undtagelse 7.a: Bruger vælger ”Fortryd”] |

|  |
| --- |
| **Undtagelser** |
| [Undtagelse 2.a: Bruger vælger ”Rediger eksisterende”]   * + - * 1. Bruger vælger preset fra liste over eksisterende presets         2. Der fortsættes fra punkt 4 i hovedscenariet, hvor alle menuer herefter er udfyldt med data fra det valgte preset   [Undtagelse 2.b: Bruger vælger ”Slet”]   * + - * 1. Bruger vælger preset fra liste over eksisterende presets         2. Bruger bekræfter valg   [Undtagelse 2.b.2.a: Bruger vælger ”Fortryd”]  [Undtagelse 2.b.2.a: Bruger vælger ”Fortryd”]   * + - * 1. Der fortsættes fra punkt 2.b.1 i Undtagelser   [Undtagelse 7.a: Bruger vælger ”Fortryd”]   * + - * 1. Der fortsættes fra punkt 4 i hovedscenariet |

### Vælg preset

|  |  |
| --- | --- |
| **Navn** | **Vælg preset** |
| **Use Case ID** | 5 |
| **Samtidige forekomster** | 1 (inklusiv denne) |
| **Primær aktør** | Bruger |
| **Sekundær aktør** | Ingen |
| **Initiator** | Bruger |
| **Prækondition** | Mindst ét preset er konfigureret |
| **Formål** | Bruger vælger et preset |
| **Postkondition** | Der er skiftet til et nyt preset |

|  |
| --- |
| **Hovedscenarie** |
| * 1. Bruger vælger preset på Body   2. Body sender et signal til Rock med det valgte preset   [Undtagelse 2.a. Afsendelse mislykkedes]   * 1. Rock opdaterer det aktive sæt af sensorkonfigurationer på baggrund af det valgte preset, og UC5 afsluttes |

|  |
| --- |
| **Undtagelser** |
| [Undtagelse 2.a: Afsendelse mislykkedes]  1. Body afgiver fejlmeddelelse  2. Bruger afvikler UC1 ”Forbind Body og Rock”, og der fortsættes fra punkt 1 i hovedscenariet. |

### Indsamle sensordata

|  |  |
| --- | --- |
| **Navn** | **Indsaml sensordata** |
| **Use Case ID** | 6 |
| **Samtidige forekomster** | 1..\* (inklusiv denne) |
| **Primær aktør** | BodyRock3000 |
| **Sekundær aktør** | Ingen |
| **Initiator** | BodyRock3000 |
| **Prækondition** | Body og Rock er forbundet, og der er forbundet mindst én sensor |
| **Formål** | At sende rådata fra Body til Rock |
| **Postkondition** | Systemet har indsamlet og gemt sensordata i buffer på Rock |

|  |
| --- |
| **Hovedscenarie** |
| 1. Sensor genererer rådata 2. Body sender rådata trådløst til Rock   [Undtagelse 2.a: Afsendelse mislykkedes]   1. Rock indlæser data i buffer 2. Rock er klar til at modtage nye data og UC6 afsluttes |

|  |
| --- |
| **Undtagelser** |
| [Undtagelse 2.a: Afsendelse mislykkedes]   1. 1. Body afgiver fejlmeddelelse |

### Generér MIDI

|  |  |
| --- | --- |
| **Navn** | **Generér MIDI** |
| **Use Case ID** | 7 |
| **Samtidige forekomster** | 1..\* (inklusiv denne) |
| **Primær aktør** | Ingen |
| **Sekundær aktør** | MIDI-Modtager |
| **Initiator** | Ingen |
| **Prækondition** | Mindst ét preset er konfigureret for den aktuelle sensor |
| **Formål** | At konvertere rådata til MIDI-signaler som sendes til hhv. MIDI-afspiller og MIDI-udgang |
| **Postkondition** | MIDI-signaler, baseret på rådata, er sendt til hhv. MIDI-afspiller og MIDI-udgang |

|  |
| --- |
| **Hovedscenarie** |
| 1. Konverteringsmodul læser data fra buffer   [Undtagelse 1.a: Aflæsning af buffer mislykkedes]   1. Konverteringsmodul[[3]](#footnote-3) omdanner rådata til MIDI-meddelelser 2. Konverteringsmodul henter lydpakken ift. sensorkonfiguration for den pågældende sensor 3. Signalet fra konverteringsmodul sendes til hhv. MIDI-afspiller og MIDI-udgang, og UC7 afsluttes |

|  |
| --- |
| **Undtagelser** |
| [Undtagelse 1.a: Aflæsning af buffer mislykkedes]   1. 1. Rock afgiver en fejlmeddelelse 2. 2. Der fortsættes fra punkt 1 i hovedscenariet. |

### Afspil lyd

|  |  |
| --- | --- |
| **Navn** | **Afspil lyd** |
| **Use Case ID** | 8 |
| **Samtidige forekomster** | 1 (inklusiv denne) |
| **Primær aktør** | Ingen |
| **Sekundær aktør** | Højtalersystem |
| **Initiator** | Ingen |
| **Prækondition** | Mindst én lydpakke er installeret, mindst ét preset et konfigureret, og MIDI-signal modtages fra konverteringsmodul |
| **Formål** | At afspille lyd genereret af BodyRock3000 |
| **Postkondition** | Et lydsignal, der repræsenterer valgt preset, lydpakke og MIDI-signal, er genereret og sendt til højtalersystem |

|  |
| --- |
| **Hovedscenarie** |
| 1. Lydmodul aflæser valgt lydpakke   [Undtagelse 1.a: Aflæsning af lydpakke mislykkedes]   1. Lydmodulet aflæser valgte preset   [Undtagelse 2.a: Aflæsning af preset mislykkedes]   1. Lydmodulet modtager MIDI-signaler   [Undtagelse 3.a: Modtagelse af MIDI-signaler mislykkedes]   1. Lydfil der skal manipuleres vælges ud fra MIDI-signal 2. Valgt lydfil manipuleres i overensstemmelse med indkommende MIDI-signal (LFO, pitch, volumen, tænd, sluk, eller lignende) 3. Lydsignal udsendes analogt på en port til højtalersystem, via en DAC, og UC8 afsluttes |

|  |
| --- |
| **Undtagelser** |
| [Undtagelse 1.a: Aflæsning af lydpakke mislykkedes]   1. 1. Rock afgiver fejlmeddelelse   2. Bruger trykker på ”Forsøg igen”, og der fortsættes fra punkt 1 i hovedscenariet.  [Undtagelse 1.a.2.a: Bruger trykker på ”Afbryd”]  1. Der returneres til Hovedmenu, og UC8 afsluttes  [Undtagelse 1.a.2.a: Bruger trykker på ”Afbryd”]  1. Der returneres til Hovedmenu, og UC8 afsluttes  [Undtagelse 2.a: Aflæsning af preset mislykkedes]   1. 1. Rock afgiver fejlmeddelelse   2. Bruger trykker på ”Forsøg igen”, og der fortsættes fra punkt 2 i hovedscenariet.  [Undtagelse 2.a.2.a: Bruger trykker på ”Afbryd”]  1. Der returneres til Hovedmenu, og UC8 afsluttes  [Undtagelse 2.a.2.a: Bruger trykker på ”Afbryd”]  1. Der returneres til Hovedmenu, og UC8 afsluttes  [Undtagelse 3.a: Modtagelse af MIDI-signal mislykkedes]  1. Der fortsættes fra punkt 3 i hovedscenariet |

## Ikke funktionelle krav

### Generelt

* 1. Den trådløse forbindelse mellem Body og Rock skal have en rækkevidde af minimum 10 meter.

### Body

* 1. Skal ved aktiv brug kunne holde strøm i min. 2 timer.
  2. Skal forsynes af et 9V batteri?
  3. Skal have en power-indikator i form af en diode, der lyser grønt når enheden er tændt.
  4. Skal have en fejl-indikator i form af en diode, der lyser rødt ved fejl.[[4]](#footnote-4)
  5. Skal kunne styre op til 16 sensorer på én gang.

### Rock

* 1. Skal have en strømforsyning, der tilsluttes et 220V 50 Hz el-net.
  2. Skal have en power indikator i form af en diode, der lyser grønt når enheden er tændt.
  3. Skal have en fejlindikator i form af en diode, der lyser rødt ved fejl.
  4. Skal have en brugergrænsefladen der er en GUI baseret på Linux Kubuntu.
  5. Skal have et MIDI output af typen ”5-pin DIN hun”
  6. Skal have et lydoutput af typen ”stereo mini-jack hun”.

# Systemarkitektur Hardware

## Overordnet arkitektur

Dette afsnit beskriver systemarkitekturen for ”BodyRock3000”. Yderligere beskrivelse kan ses i projektformuleringen og kravspecifikationen.

Formålet er at identificere:

* De overordnede komponenter, samt at bestemme deres grænseflader
* De eksterne komponenter, samt at beskrive dem
* Arbejdsopgaver for projektets design- og implementeringsfase

### Domæne model BodyRock3000



På figuren herover ses en overordnet domæne model med konteksten for systemet ”BodyRock3000”.

Notér at Body og Rock har hver sit bluetooth-modul, der benyttes som interface mellem deres indbyrdes kommunikation.

## Body HW Arkitektur

### BDD Body



**Body** består af Controller, Spændingsforsyning, Bluetooth-modul og Preset knapper.

**Controller** er hjernen i Body. Denne står for indsamling, pakning og videre afsendelse af data fra sensorBus til Bluetooth-modul. Den afkoder og videresender desuden information fra Preset knapper.

**Spændingsforsyning** leverer spænding til de resterende komponenter i Body.

**Bluetooth-modul** videresender data fra Controller til Rocks Bluetooth-modul.

**Preset knapper** bruges af brugeren til vælge preset.

### Allokeringsdiagram Body



**Controller** implementeres på en **PSoC4**

**Spændingsforsyningen** implementeres som hhv. en **Reguleringskreds** og et **Batteri**

**Bluetooth-modul** implementeres med komponenten **HC-05**

**Preset knapper** implementeres med en **3x4 knapmatrix**

### IBD Body

Herunder ses et internt block diagram for Body. Dette illustrerer interne forfindelser i Body:



**Batteri** generer den strøm til systemet.

**Reguleringskreds** bruges til at transformere den indgående batterispænding til 3,3V, som de resterende Body enheder samt eksterne sensorer kører på.

**PSoC4** er systemets kontrol-enhed. Den benyttes til at indsamle, bearbejde og pakke data fra sensorbus og 3x4 knapmatrix, og sender disse til bluetooth-modulet RN42. Den kontrollerer desuden systemets statusLEDer.

**StatusLEDer** består af består af hhv. en grøn- og en rød LED, der indikerer hhv. power status (hvorvidt enheden er tændt) og error status (hvorvidt der opstår fejl på enheden).

**3x4 knapmatrix** benyttes af brugeren til at skifte mellem forskellige systempresets.

**HC-05** Forbinder Body og Rock trådløst via bluetooth. Den er forbundet til PSoC4 med en RX og en TX forbindelse

**Sensorbus** er en I2C bus, der består af hhv. GND, VDD, SDA og SCL forbindelserne.[[5]](#footnote-5)

## Sensor IBD (Generisk)

Alle sensor-enheder, kobles på sensorbus som følger:



Sensorenhederne er alle koblet parrallelt op med en strømforsyning (GND og VCC) samt I2C forbindelserne SCL (serial clock) og SDA (serial data). SDA forbindelsen er non-atomic, da både master (Body) og slave (den pågældende sensorenhed) kan tage styring over linjen. Sensorenhedernes interfaces er lavet således, at de kan kobles i forlængelse af hinanden.

## Rock HW Arkitektur

### BDD Rock



**Rock** består af Functionalities, Bluetooth-modul og Touchscreen.

**Functionalities** er en abstraktion for alle Rocks logiske funktionaliteter. Se ”BDD Functionalities”[[6]](#footnote-6) for yderligere beskrivelse.

**Bluetooth-modul** modtager data fra Bodys Bluetooth-modul.

**Touchscreen** benyttes af brugen til at operere Rocks GUI og ændre indstillinger for Rock.

### Allokeringsdiagram Rock



**Functionalities** og **Touchscreen** implementeres på et DevKit8000.

**Bluetooth-modul** implementeres med komponenten **RN42**

## Grænsefladebeskrivelse

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Navn** | **Beskrivelse** | **Signaler** | **Område** | **Kommentar** |
| sensorbus | I2C [[7]](#footnote-7) bus forbundet med RJ11-stik | GND | 0 V |  |
| VCC | 3,3 V |  |
| SCL | 0-3,3 V | I2C serial clock signal.  Følger 3,3V CMOS standard[[8]](#footnote-8) |
| SDA | 0-3,3 V | I2C serial data signal Følger 3,3V CMOS standard |
| BluetoothCon | Forbindelse fra Controller til RN42-modul | RX? |  |  |
| TX? |  |  |
| MIDIOut | 5 pin MIDI output |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| soundOut | Analog lyd output |  |  |  |
|  |  |  |

# Systemarkitektur Software

I følgende afsnit beskrives softwarearkitekturen for systemet ”BodyRock3000”. Softwarearkitekturen er opbygget på baggrund af projektformuleringen og kravspecifikationen.

Formålet er at skabe en ramme for softwaren-udviklingen, hvorfra delopgaver kan uddelegeres til udviklere.

Softwarearkitekturen består af:

* Identifikation af problemer, klasser og metoder med udgangspunkt i en detaljeret domænemodel for system og det software-moduler.
* Oprettelse af skelet for videre implementering

## Detaljeret domænemodel for software-moduler

Herunder ses en domænemodel for kommunikationen mellem systemets interne SW-moduler.



## Klasse-identifikation

På baggrund af domænemodellen for softwaremoduler, identificeres følgende klasser for hhv. Body og Rock:

**Body** generer data ud fra brugerens påvirkning af sensorer og tryk på knapmatrix. Disse data pakkes i Body’s Serial Unit og videresendes via Bluetooth-modul.

* **Main** er Body’s central for indsamling af sensordata og håndtering af knap-tryk.
* **SerialUnit** står for at pakke de fra Body genererede data og adaptere dem til Bluetooth-moduls interface
* **Bluetooth-modul** er driveren til det fysiske Bluetooth-modul HC-05, der muliggør afsendelse af datapakker fra Body til Rock.

**Rock** afkoder data fra Body og generer MIDI og lyd på baggrund af de indstillede værdier i dataStorage. Sidstnævnte indstilles fra GUI.

* **GUI** er systemets bruger-interface, der benyttes til bruger-indstillinger for systemet.
* **Controller** er Rocks styrings-hub. Den konfigurerer dataStorage på baggrund af brugeren interaktion med GUI, og konfigurer Receiver (dataAdapter), MIDI-module (MIDI-mapper) og Sound-module (Linux Sampler) ved modtagelse af besked om preset-skift fra Receiver.
* **DataStorage** opbevarer data på systemet. Denne består af Preset-bank, Sensorkonfiguration og Sample Bibliotek.
  + **Preset-bank** indeholder systemets samlede presetindstillinger. Den består af op til 9 Presets, der hver kan tilknyttes op til 16 Sensorkonfigurationer
  + **Sensorkonfigurationer** indeholder systemets brugerindstillinger for hver sensor-enhed, der kan tilkobles.
  + **Samplebibliotek** indeholder de lyd-samples, som Linux Sampler kan afspille.
* **Receiver** står for modtagelse og udpakning af data fra Body via Bluetooth, samt bestemmelse af datatype og heraf modtager på videre afsendelse. Receiver består af følgende sub-moduler:
  + **Bluetooth-modul** er driveren til det fysiske Bluetooth-modul RN-42, der modtagelse afsendelse af datapakker fra Body til Rock.
  + **SerialUnit** står for at udpakke de fra Body modtagne data.
  + **dataAdapter** står for genkendelse af typen for de udpakkede data (hhv. sensor-data eller preset-valg) samt adaptering og afsendelse af disse til det relevante modul i systemet. Adaptering kan i fremtidige systemudgaver indeholde implementering af velocity-kurve for sensorData. Presetvalg og systeminfo videresendes til Controller og sensordata videresendes til MIDI-module.
* **MIDI-module** står for at generere MIDI på baggrund af modtagne sensordata. Disse outputtes på en fysisk MIDI-port og internt til Sound-module. MIDI-module består af følgende sub-moduler:
  + **MIDI-mapper** indeholder policy for hvordan der på baggrund af sensordata fra en specifik sensor genereres MIDI. Konkrete indstillinger for den givne sensor findes i submodulet Sensorkonfiguration i dataStorage.
  + **MIDI-generator** er driveren, der indeholder MIDI core-funktionaliteten. Denne videresender den generede MIDI til hhv. Sound-module og MIDI Output.
  + **MIDI Output** er driveren for Rocks fysiske MIDI-port.
* **Sound-module** står for generering af lyd på baggrund af modtagne MIDI-signaler
  + **Linux Sampler** afspiller lyd-samples fra Sample Bibliotek på baggrund af modtaget MIDI og givne indstillinger i Sensorkonfiguration.
  + **Sound Output** er Rocks lydkort.

## Tråde i Rock

For på bedst mulig vis at håndtere DevKit8000’s resurser oprettes en lav- og en høj-prioritets. Disse kaldes hhv. Slow Lane og Fast Lane.

### Slow Lane

Denne tråd håndterer de software-moduler i systemet, hvis afvikling ikke er kritisk for den musikalske oplevelse (?) ved benyttelse af BodyRock3000.

#### SW-moduler i Slow Lane

#### Sekvensdiagram for indstillinger i main

### Fast Lane

Denne tråd håndterer de software-moduler i systemet, hvis afvikling er kritisk for den musikalske oplevelse (?) ved benyttelse af BodyRock3000.

#### SW-moduler i Fast Lane

#### Sekvensdiagram for modtagelse og konvertering af sensorData

## Klassediagrammer

### Klassediagram Body

### Klassediagram Slow Lane

### Klassediagram Fast Lane

### Software BDD Rock



### Software IBD Rock



#### Grænsefladebeskrivelse

**Body\_Rock\_UART:**

Full duplex UART. Fra Body til Rock sendes sensor data og preset valg eller svar på kommandoer fra Rock.

Fra Rock til Body sendes kommandoer til at styre scanning efter sensorer el. lign.

Alle data er pakket efter følgende protokol:

StartByte – KontrolByte – SampleIndexByte – DataByte0 … DataByteN – StopByte

**sensToMidi**

De behandlede sensordata fra Body kan tilgås af midigeneratoren gennem get metoder i TrådløsKom klassen. De returnerede data er lineariseret og skaleret til en 7 bit fortegnsløs værdi.

**PresetVal**

TrådløsKom klassen notificerer kontrolleren når der modtages en kommando om skift af preset.

**MidiOut**

MIDI beskeder på 1 – 3 bytes.

### Software BDD Body



### Klassediagram Sensor-to-Midi



### Klassediagram GUI

#### Funktions beskrivelser

### Klassediagram Bluetooth-Rock



#### Funktions beskrivelser

|  |  |
| --- | --- |
| **Funktion** | receiveUnpack(): struct |
| **Parametre** | Ingen |
| **Returværdi** | Struct med sensordata, sample-indeks og sensor-ID |
| **Beskrivelse** | Funktionen aflæser data fra Rx-bufferen og indlæser dem i sensorData-structen. Efter start-byte er læst, indlæses data i følgende rækkefølge:   * Sample-indeks * Sensor-ID * x-data * [y-data] * [z-data]   Ved slut-byte stoppes der med at indlæses data i struct’en, og evt. resterende dimensioner (y- og z-data) sættes til 0. |

|  |  |
| --- | --- |
| **Funktion** | Reconnect(): uint8\_t |
| **Parametre** | Ingen |
| **Returværdi** | 1 eller 0 |
| **Beskrivelse** | Funktionen forsøger at genoprette Bluetooth-forbindelsen.  Der returneres 1 hvis der er oprettet forbindelse, og 0 hvis ikke. |

### Klassediagram Bluetooth-body



#### Funktionsbeskrivelser

|  |  |
| --- | --- |
| **Funktion** | packSend( uint16, uint16, uint16, uint8, uint8 ): void |
| **Parametre** | x-, og evt. y- og z-værdier for rå sensordata, samt et index for samples og et sensor-ID |
| **Returværdi** | Ingen |
| **Beskrivelse** | Funktionen pakker og sender de medsendte attributer serielt over Bluetooth. Data sendes i følgende rækkefølge:   * Start-byte * Sample-indeks * Sensor-ID * x-data * [y-data] * [z-data] * Slut-byte |

|  |  |
| --- | --- |
| **Funktion** | Reconnect(): uint8 |
| **Parametre** | Ingen |
| **Returværdi** | 1 eller 0 |
| **Beskrivelse** | Funktionen forsøger at genoprette Bluetooth-forbindelsen.  Der returneres 1 hvis der er oprettet forbindelse, og 0 hvis ikke. |

### Klassediagram GUI



1. http://www.midi.org/techspecs/midispec.php [↑](#footnote-ref-1)
2. Teknisk set ikke simultant, men virtuelt simultant, da mennesket vil opfatte det sådan [↑](#footnote-ref-2)
3. Konverteringsmodulet indeholder én til flere konverteringstråde, afhængig af antal sensorer i systemet [↑](#footnote-ref-3)
4. Fejl defineres som XXXXX [↑](#footnote-ref-4)
5. For beskrivelse af I2C, se <http://www.nxp.com/documents/user_manual/UM10204.pdf> [↑](#footnote-ref-5)
6. Indsæt reference [↑](#footnote-ref-6)
7. For en detaljeret beskrivelse af I2C se <http://www.nxp.com/documents/user_manual/UM10204.pdf> [↑](#footnote-ref-7)
8. INDSÆT REFERENCE [↑](#footnote-ref-8)