Body Control er styresystemet på Body-delen af BodyRock3000. Systemet står for at samle sensordata fra de fire[[1]](#footnote-1) forskellige sensorer via I2C, konverterer disse data og videresende dem vha. en UART forbundet til et Bluetooth modul.

### Klassediagram Body



Figur 1 UML af Body Control

#### Main

Main er Bodys control klasse. Main står for at initerer I2C, UART og de tilkoblede sensorer samt at kalde funktioner der kan læse sensorer, standardisere sensordata og sende sensor- og presetdata via SerialUnit. På nuværende tidspunkt skal initieringsfunktionerne til sensorene stå i Main klassen for at funktionerne kan implementeres.

#### Sensor

Sensor står for at klargøre I2C, læse sensordata via I2C, gemme sensordata i en midlertidig buffer, konvertere sensordata til et generisk interval fra 0-127. Sensor sender også de generiske sensordata til SerialUnit. Sensor indeholder også I2C fejlhåndtering.

#### SerialUnit

Står for at klargøre sensordata til afsendelse via UART og videresende sensordata’en over uart.

#### Knap

Ikke implementeret da denne klasse omhandler preset.

#### Funktioner

|  |  |
| --- | --- |
| Funktioner \* | Beskrivelse |
| initSensors() | Kalder de forskellige initieringsfunktioner for hver sensor. |
| initADXL345() | Initialiserer accelerometeret. |
| initMPU6050() | Initialiserer gyroskopet. |
| initUART() | Starter UART komponenten. |
| setupI2C() | Starter I2C komponenten. Returnerer master status og clearer status flaget. |
| readAllSensors() | Kalder funktionen readI2C for hvert sensorregister der er, at læse fra. |
| readI2C(int numOfReg) | Modtager det register nummer der skal læses fra og følger herefter I2C protokollen for læsning og gemmer det læste data. |
| convSensData() | Samler sensoraksernes most-significant og least-significant byte til en.  Konverterer herefter til MIDI-skalaen 0-127.  Kalder funktionen setdataArray(int, int, int, int). |
| setdataArray(int, int, int, int) | Fylder array med sensor ID og data fra convSensData() |
| sendDataArray() | Sender et array med data for hver sensor. |
| handleI2CError() | Funktion som kaldes ved fejl i I2C forbindelsen. Genererer en stop condition. |
| setSensArray(int, int, int) | Fylder det multidimensionelle array. |

\*Alle preset funktioner er ikke skrevet da man har valgt ikke at implementere preset I den iteration.

Uddybende funktionsbeskrivelser kan ses Projekt Dokumentationen(REFERENCE)

1. Denne iteration samler kun data fra en sensor. [↑](#footnote-ref-1)