

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Arvutisüsteemide instituut

Jörgen Vedom 123584 IASB

**MITME KIIRENDUSANDURIGA  
KUKKUMISE TUVASTAMISE SÜSTEEMI  
ARENDAUS**

Bakalaureusetöö

Juhendaja: Mairo Leier  
Doktorikraad

Tallinn 2017

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Jürgen Vedom

16.05.2017

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks on luua prototüüp, mida kasutatakse kukkumise tuvastamise algoritmi välja töötamisel.

Töös on kolm põhiülesannet, millest esimene on prototüübi riistvara arendus, mis sisaldab mikrokontrolleri ja sensori valikut. Teine ülesanne on mikrokontrolleri seadistamine töötamiseks mitme kiirendusanduriga. Kolmandaks on kiirendusandurite seadistamine vastavalt algoritmi poolt sisendandmetele esitatavatele nõuetele, nende kalibreerimine ja testimine.

Töös on esitatud kõigi kasutatud seadete ja tarkvara kirjeldused ning kasutamise põhjendused. Lisaks on selgitatud kuidas toimub suhtlus mikrokontrolleri ja sensori vahel.

Töö tulemuseks on prototüüp, mida saab kasutada kukkumise tuvastamise algoritmi välja töötlemisel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 51 leheküljel, 5 peatükki, 31 joonist, 22 tabelit.

## **Abstract**

### **Development of multi-accelerometer fall detection system**

The aim of this thesis is to create a prototype on which one could develop a fall detection algorithm.

There are three main tasks. The first task is to develop a hardware for the prototype which includes selecting the microcontroller, and selecting the protocol for hardware communication. Second task is to configure the microcontroller to work with multiple accelerometers. Third task is to configure the accelerometers according to the input data requirements of the algorithm. This includes the range selection for the accelerometers and the calibration of them.

This thesis presents descriptions about all of the used equipments and software including how the communication between the microcontroller and the sensors works.

Tests will be conducted for proof of concept. The tests will reveal if the device works and if it has any flaws. As a result we will have a prototype that can be used to develop a fall detection algorithm.

The thesis is in Estonian and contains 51 pages of text, 5 chapters, 31 figures, 22 tables.

## Lühendite ja mõistete sõnastik

ABP	<i>Alternating bit protocol</i> , vahelduvbitiga protokoll
AHB	<i>Advanced High-performance Bus</i> , kõrgjõudlusega siin
API	<i>Application Programming Interface</i> , rakendusliides
ARM	<i>Advanced RISC machine</i> , kärbitud käsustikuga arvutiarhitektuur
ASCII	<i>American Standard Code for Information Interchange</i> , Ameerika Informatsioonivahetuse Standardkood
ASI	TTÜ Arvutisüsteemide instituut
CAN	<i>Controller area network</i> , kontrollervõrk
CPU	<i>Central processing unit</i> , keskprotsessor
DMA	<i>Direct Memory Access</i> , otsemällupöörduskanal
DPI	<i>Dots per inch</i> , punkti tolli kohta
GND	<i>Ground</i> , maandus
GPIO	<i>General purpose input/output</i> , üldotstarbeline sisend/väljund
GPS	<i>Global positioning system</i> , globaalne positsioneerimissüsteem
HAL	<i>Hardware abstraction layer</i> , riistvara abstrakt kiht
HID	<i>Human interface device</i> , inimliidese seade
I2C	<i>Inter-integrated circuit</i> , kahesuunaline kahesoonealine järjestiksiin
IDE	<i>Integrated development environment</i> , integreeritud programmeerimiskeskond
LED	<i>Light emitting diode</i> , valgusdiod
LSB	<i>Least significant bit</i> , vähima kaaluga bitt
MSB	<i>Most significant bit</i> , suurima kaaluga bitt
PS	<i>Protocol select</i> , protokollivalik
RXD	<i>Receive data signal</i> , vastuvõtmissignaali
SAI	<i>Serial audio interface</i> , järjestik heliliides
SCL	<i>Clock line</i> , takti liin
SDA	<i>Data line</i> , andme liin
SDIO	<i>Secure digital input/output card</i> , SD sisend-väljundkaart
SPI	<i>Serial Peripheral interface</i> , järjestikliidese väliseade
SRAM	<i>Static random access memory</i> , staatiline muutmälu
TXD	<i>Transmit data signal</i> , saatmissignaali
UART	<i>Universal asynchronous receiver/transmitter</i> , universaalne asünkroontransiiver
USART	<i>Universal synchronous receiver/transmitter</i> , universaalne sünkroontransiiver
USB	<i>Universal serial bus</i> , universaalne järjestiksiin
WIFI	<i>Wireless fidelity</i> , raadiokohtvõrk

## Sisukord

1 Sissejuhatus .....	11
2 Riistvara.....	13
2.1 Süsteemi nõuded.....	13
2.2 Arendusplaat STM32 F446RE .....	15
2.3 Kiirendusandur Bosch BNO055 .....	16
2.3.1 Kiirendusandur Bosch BNO055 register .....	16
2.4 Andmeside protokollid .....	17
2.4.1 UART protokoll.....	17
2.4.2 RS-485 protokoll .....	18
2.5 Arendusplaadi ja sensori vahelised ühendused .....	19
2.6 Ülevaade sarnastest lahendustest.....	21
2.6.1 Wearable sensors for Reliable fall detection .....	21
2.6.2 Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information .....	21
2.6.3 Philips GoSafe .....	22
2.6.4 Shimmer sensing .....	23
3 Tarkvara.....	24
3.1 STM32CubeMX programm .....	24
3.1.1 STM32Cube HAL teek.....	25
3.2 Programmi töö .....	26
3.3 Arendusplaadi initsialiseerimine .....	27
3.4 Sensori initsialiseerimine.....	28
3.5 Andmete päring sensorilt.....	29
3.5.1 Andmete lugemine sensorilt .....	29
3.5.2 Sensori registrisse kirjutamine .....	31
3.6 Häirete kontroll.....	33
3.7 Kiirendusanduri kalibreerimine .....	33
4 Testimine ja katsed .....	35
4.1 Paigaloleku katse .....	36

4.2 Hüppamise katse.....	38
4.3 Istumise ja tõusmise katse .....	41
4.4 Erinevad liikumised.....	43
4.5 Kukkumise ja pörutuse katse.....	46
5 Kokkuvõte .....	49
Kasutatud kirjandus .....	50
Lisa 1 – Kood .....	52

## Jooniste loetelu

Joonis 1. Süsteemi ülesehitus .....	13
Joonis 2. Arendusplaat STM32F446RE .....	15
Joonis 3. Kiirendusandur BNO055.....	16
Joonis 4. UART andmeformaad .....	17
Joonis 5. TXD ja RXD .....	18
Joonis 6. RS-485 ühendus .....	18
Joonis 7. Arendusplaat sensoritega.....	19
Joonis 8. Kantav sensor kukkumise tuvastamiseks .....	21
Joonis 9. Kehahoiakut arvestav kukkumise tuvastus.....	22
Joonis 10. Philips GoSafe.....	22
Joonis 11. Consensys Development Kit .....	23
Joonis 12. STM32CubeMX projekti seaded.....	25
Joonis 13. Programmi töö.....	26
Joonis 14. Arendusplaadi initsialiseerimine .....	27
Joonis 15. Sensori initsialiseerimine .....	28
Joonis 16. Häirete kontroll .....	33
Joonis 17. Arendusplaadi kalibreerimise nupp B1 ja LED LD2 .....	34
Joonis 18. RealTerm andmete kuvamine.....	35
Joonis 19. Paigalolek .....	36
Joonis 20. Kiikumine.....	37
Joonis 21. Hüppamine murul.....	38
Joonis 22. Hüppamine puidul .....	39
Joonis 23. Hüppamine puitaluselt murule .....	40
Joonis 24. Istumine .....	41
Joonis 25. Tõusmine .....	42
Joonis 26. Kõndimine .....	43
Joonis 27. Trepil kõndimine .....	44
Joonis 28. Jooksmine.....	45
Joonis 29. Esimene murul kukkumine.....	46



Joonis 30. Teine murul kukkumine .....	47
Joonis 31. Põrutus vastu seina .....	48

## Tabelite loetelu

Tabel 1. Bosch BNO055 UART seaded.....	18
Tabel 2. Arendusplaadi ja sensorite ühendused .....	20
Tabel 3. UART seaded .....	24
Tabel 4. MSB ja LSB andmetöötlus.....	27
Tabel 5. Lugemise päring.....	29
Tabel 6. Registrist õnnestunud lugemise vastus.....	30
Tabel 7. Registrist lugemise veateade .....	30
Tabel 8. Kirjutamine.....	31
Tabel 9. Kirjutamise vastus .....	32
Tabel 10. Paigaloleku suurim amplituud.....	36
Tabel 11. Kiikumise suurim amplituud .....	37
Tabel 12. Murul hüppamise suurim amplituud .....	38
Tabel 13. Puidul hüppamise suurim amplituud.....	39
Tabel 14. Hüppamine puitaluselt murule suurim amplituud.....	40
Tabel 15. Istumise suurim amplituud .....	41
Tabel 16. Tõusmise suurim amplituud .....	41
Tabel 17. Kõndimise suurim amplituud .....	43
Tabel 18. Trepil kõndimise suurim amplituud .....	44
Tabel 19. Jooksmise suurim amplituud .....	45
Tabel 20. Esimese murul kukkumise amplituud .....	46
Tabel 21. Teise murul kukkumise amplituud.....	47
Tabel 22. Vastu seina põrutuse amplituud.....	48

## 1 Sissejuhatus

Käesoleva lõputöö teemaks on kukkumise tuvastamise süsteemi arendus. Töö põhieesmärgiks on luua prototüüp, mille abil saab välja töötada kukkumise tuvastamise algoritmi. Prototüüpi kasutatakse uuringute läbiviimisel liikuvatel alustel, milleks võivad olla meresõidukid ja merel olevad kalafarmid. Merel puututakse kokku ettearvamatute olukordadega, mis võivad olla eluohtlikud. Näiteks libedal põrandal võib kukkuda ning end vigastada. Kukkumist mitte märgates võib inimene jääda abita ning halvimal juhul paiskuda lainetega üle parda. Mida kauem kulub aega, et märgata inimese puudumist, seda suurem on oht tema elule ja leidmine on keerulisem. Seadme eesmärgiks on tuvastada kukkumine ning sellest koheselt märku anda.

Sellest tulenevalt kasutatakse seadet kinnitatuna inimese küljes, kus üle keha on paigutatud mitu erinevat sensorit. Väritingimustes tuleb arvestada keskkonnamõjudega. Merel paiknevat alust mõjutavad peale lainete ka teised olud, näiteks ilmastik. Seade peab olema vastupidav, et tagada töökindlus ja andmeside sellistes tingimustes.

Sensorite ühendamiseks põhiplaadiga kasutatakse riidesse integreeritud juhtmeid. Sensorid asuvad teineteisest kuni mõnekümne sentimeetri kaugusel ning on ühendatud põhiplaadiga, mille ülesanne on teostada signaalitöötlust ja kommunikatsiooni. Juhtmed tagavad kiire ja kindla ühenduse põhiplaadiga. Mida pikem on ühendus, seda rohkem esineb häireid, seega tuleb protokollis valikus arvestada veakindlusega.

Seoses sellega, et süsteemi eesmärgiks on hoiatada inimelu ohtu sattumise korral, tehakse otsuseid reaalsajas. Olud merelistes tingimustes muutuvad kiiresti ja seetõttu on kohene reageerimine oluline. Süsteem peab olema võimeline andmeid kiiresti töötleva. Mida sagedamini toimub andmete töötlus ja kukkumise tuvastus, seda kiiremini saab juhtunust teavitada. Kukkumine võib toimuda nii põrandale kui ka vette. Kukkumisel võib seade saada kahjustada, seega tuleb silmas pidada, kuidas seadet kaitsta purunemise eest. Kuna konkreetset prototüüpi kasutatakse testide läbiviimisel, siis katsete tulemused annavad hea tagasiside, mida seadme töökindluse suurendamisel silmas pidada.

Arendusplaati peab saama kiirelt ümber seadistada. Seega tuleb sensorite ja arendusplaadi vahelisel suhtlusel pidada silmas, et programm pakuks seadistamisvõimalusi algoritmi töötlejale.

Järgnevalt on välja toodud lõputöö käigus lahendatavad ülesanded:

- Prototüübi riistvara arendus
- Mikrokontrolleri seadistamine töötamaks mitme kiirendusanduriga
- Kiirendusandurite seadistamine vastavalt algoritmi poolt sisendandmetele esitatavatele nõuetele

Esimeseks ülesandeks tuleb valida riistvara, milleks on mikrokontroller ja sensorid. Seejärel peab valima protokollid, mis sobib riistvaraga suhtlemiseks. Arendusplaat tuleb initsialiseerida ning panna suhtlema sensoriga.

Teiseks püstitatud ülesandeks on mikrokontrolleri seadistamine töötamaks mitme kiirendusanduriga. Seadistamise käigus võib esineda probleeme. Näiteks, kuidas sünkroniseerida andmeid, kuidas tagada mõlema anduri ühendused, ja muud ettenägematud takistused.

Viimase ülesandena peab seadistama kiirendusandurid vastavalt algoritmi nõuetele, milleks on kiirendusandmete vahemiku valik ja kalibreerimine. Seadme töökindluse tagamiseks tuleb katseid läbi viia erinevates oludes.

## 2 Riistvara

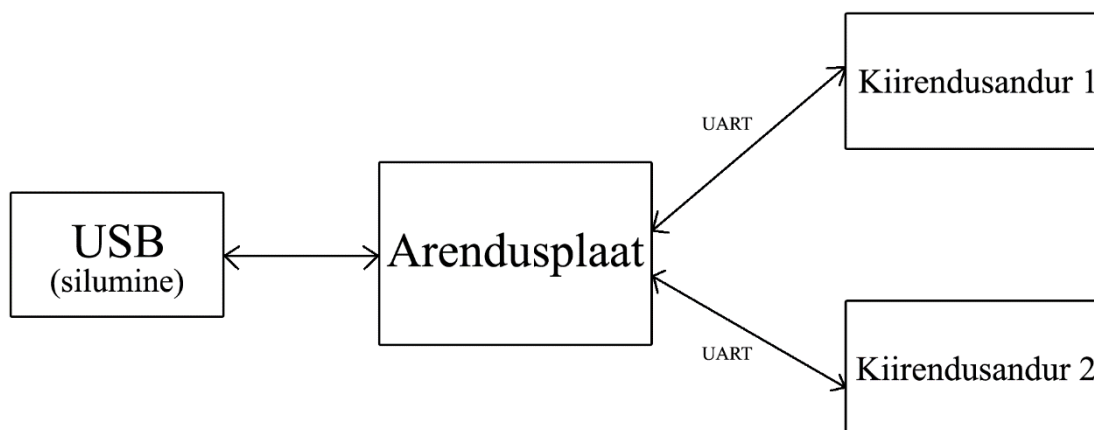
Prototüübi jaoks on vaja mikrokontrollerit ning kiirendusandureid. Tuleb leida viis kuidas ühendada arendusplaat kiirendusanduritega. Seejärel tuleb valida protokoll, mida kasutatakse andmevahetuseks.

Süsteemi komponendid:

Arendusplaat STM32 -F446RE [1]

Kiirendusandur Bosch BNO055 [2]

Süsteemi ülesehitus on näidatud joonisel 1. Arendusplaadi külge on ühendatud kaks kiirendussensorit. Juhtmed on kujutatud joonisel nooltena, mis tähistavad andmevahetust.



Joonis 1. Süsteemi ülesehitus

### 2.1 Süsteemi nõuded

Kuna tegemist on reaajasüsteemiga, siis on oluline, et andurite ja arendusplaadi vahel toimuks andmevahetus kiirelt ning häirevabalt. Kiirendusanduri ja arendusplaadi valikul tuleb veenduda, et valitud protokoll oleks mõlema poolt toetatud. Arendusplaat peab toetama vähemalt nelja porti, mille kaudu oleks võimalik suhelda kiirendusanduritega. Seade peab olema kerge, vastupidav ning mõõtmetelt väike.

Mikrokontrollerite tootjaid on mitmeid, näiteks Silicon Labs [3], Texas Instruments [4] ja STMicroelectronics [5]. Silicon Labs tootevalikust leidub palju arendusplaat, millest sobilikum on Giant Gecko [6]. Nimetatud seade ei leia töö raames kasutust, sest protsessor pole piisavalt võimekas. Antud seadme protsessori taktsagedusest (48 MHz) võib jääda väheseks, sest vaja on töödelda mitmelt kiirendusandurilt saadud andmeid. Texas Instruments'i võimekaim plaat – TM4C129XNCZAD iseenesest on tegemist hea mikrokontrolleriga [7]. Protsessori taktsageduseks on 120 MHz, võimaldab mitmeid liideseid ja palju viike, kuid initsialiseerimiskoodi tuleb arendajal ise kirjutada, mis on ajakulukas. STMicroelectronic'i pakutud STM32F446RE on töö jaoks sobilik. Tootja pakub mitmeid programme, millega lihtsustatakse arendaja tööd.

Kiirendusanduri valikul tuleb veenduda, et arendusplaat toetaks andmevahetuseks samu liideseid. Andur peab mõõtmete poolest olema väike ja kaalult kerge, sest on võimalus, et see paigutatakse riide külge. Lisaks peale kiirendusanduri võiks sensoril olla mõningaid muid andureid, näiteks temperatuuriandur. Tähtis on, et andur mõõdaks kolme telge. STMicroelectronics pakub IIS328DQ madala tarnega kiirendusandurit, mis ei ole sobilik, sest suhtluseks kasutab I2C(*Inter-integrated circuit*) või SPI porti [8]. Võttes arvesse juhtmete kauguseid ei taga need protokollid veakindlat andmeside. Bosch pakub mitmeid kiirendusandureid, sealhulgas kiirendusandurit Bosch BNO055, mis on sobilik antud töö jaoks, sest toetab UART (*Universal asynchronous receiver/transmitter*) protokollid [2].

Kiirendusandur Bosch BNO055 andmevahetuseks on toetatud HID-I2C(*Human interface device*), I2C ning UART protokollid [2]. Arendusplaat STM32 F446RE toetab UART liidest [1]. Arvestades, et tegemist on prototüübiga võib tekkida vajadus veelgi pikemate juhtmete kasutamiseks, mida UART ei pruugi toetada. RS-485 protokoll, mis põhineb UART protokollil, on veel kindlam ning võimaldab ühel siinil ühendada kuni 32 seadet.

## 2.2 Arendusplaat STM32 F446RE

Tähtsamad tehnilised andmed [1]:

Protsessor: 180 MHz Cortex-M4 [9]

Mälu: 512 kB Välkmälu, 128 KB SRAM (*Static random access memory*)

Toide: 1.7 – 3.6 V

Liidesed: I2C, UART, SPI (*Serial peripheral interface*), SAI (*Serial audio interface*), CAN (*Controller area network*), GPIO (*General purpose input/output*), SDIO (*Secure digital input/output card*)

Arendusplaat on kujutatud joonisel 2.



Joonis 2. Arendusplaat STM32F446RE

Arendusplaadiks sai valitud STM32 Nucleo F446RE [1]. Valikul tuli arvestada, et tegemist on reaajasüsteemiga. Antud töö raames tähendab see seda, et protsessor peab suutma pidevalt andmeid vastu võtta, neid töödelda ning olema võimeline algoritmiga toime tulema. Plaat peab toetama mitut sisend- ja väljundseadet. Kaks kiirendusandurit, Bluetooth moodulit ning ühte lisaseadet, milleks võib olla näiteks LED (*Light emitting diode*) tuli. Mõõtmetelt on seade kompaktne: 82.50 mm x 70.00 mm.

STMicroelectronics pakub enda mikrokontrolleritele tarkvara STM32Cube, mis lihtsustab arendaja tööd [10]. Tarkvaraga on võimalik genereerida initsialiseerimiskood. Arendaja määrab pordid, protokollid ja konfiguratsiooniseaded ning STM32Cube programm genereerib esialgse koodi vastava seadistusega.

## 2.3 Kiirendusandur Bosch BNO055

Tähtsamad tehnilised andmed [2]:

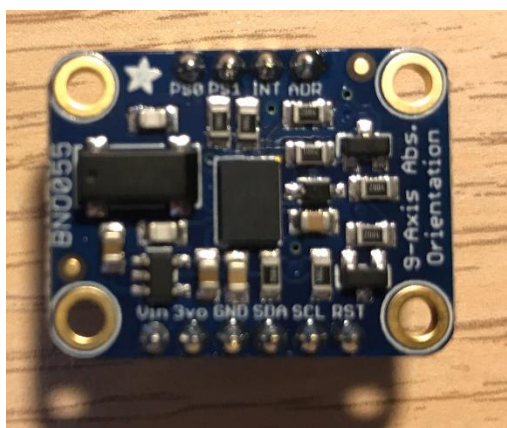
Andurid: Kiirendusandur, güroskoop, geomagnetiline sensor

Toide: 2.4 – 3.6 V

Liidesed: HID-I2C, I2C, UART

Kiirendusvahemik:  $\pm 2 \text{ g} - \pm 16 \text{ g}$

Kiirendusandur on kujutatud joonisel 3.



Joonis 3. Kiirendusandur BNO055

Sensoriks sai valitud Bosch BNO055. Andmeedastuseks kasutab sensor HID-I2C, I2C ja UART liidest. Sensor koosneb kolmest sisseehitatud andurist (kiirendusandur, güroskoop ja geomagnetiline andur). Seade on võimeline töötama *fusion* režiimis, kasutades kõiki andureid andmete arvutamiseks, mida juhib protsessor on ARM Cortex-M0 (*Advanced RISK machine*) [11]. Sensor on mõõtmetelt väike: 5.2 mm x 3.8 mm. Samuti leidub sensori kohta palju materjale Internetis. Näiteks on tehtud sensoriga aplikatsioon orientatsiooni kuvamiseks, kasutades selleks arduino't [12].

### 2.3.1 Kiirendusandur Bosch BNO055 register

Register on jagatud kaheks loogiliseks leheks. Leht 1 omab endas sensoripõhiseid konfiguratsiooni seadeid. Leht 0 omab kõiki muid konfiguratsiooni parameetreid ning väljundandmeid [13].

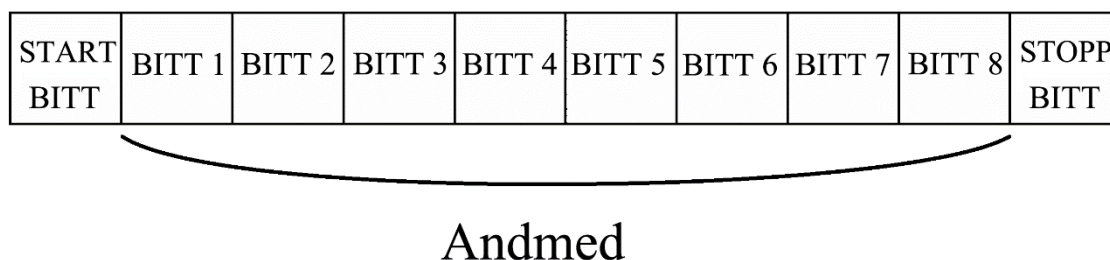


## 2.4 Andmeside protokollid

Arendusplaadil ja kiirendusanduril on ühisteks liidesteks I2C ja UART. Seoses sellega, et juhtmete pikkused võivad ulatuda mõnekümne sentimeetri pikkuseks, tuleb tagada häireteta andmeside. Protokolliks sai valitud UART ning RS-485. Esialgne arendus toimub UART protokolliga. Üleminek UART protokollilt RS-485 peale ei ole keeruline, sest RS-485 põhineb UART protokollil. RS-485 eeliseks on tema veakindlus distantsidel ning võimalus ühendada ühel siinil kuni 32 seadet.

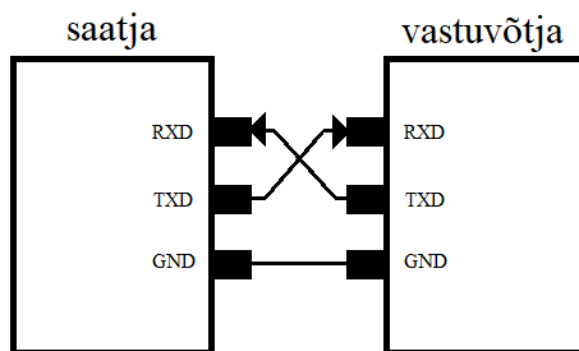
### 2.4.1 UART protokoll

UART on laialdaselt kasutatud protokoll, mida kasutatakse andmete saatmiseks. UART protokoll on asünkroonne, mis tähendab, et puudub taktsignaali. Andmete saatmisel on sõnumil algus- ja lõpubit. Oluline on, et mõlemad seadmed töötaksid samal boodikiirusel. UART protokolliga andmevahetuse maksimaalseks kauguseks on ~2m. Boodikiirus määrab kui kiirelt toimub andmevahetus [14]. Arendusplaadi ja andurite vahel on boodikiiruseks 115200 boodi. UART'i andmeformaati on näidatud joonisel 4. Ühe andme baidi saamiseks, tuleb saata kokku 10 bitti. 1 bitt on algusbitt, 8 bitti andmeid ning 1 stoppbitt.



Joonis 4. UART andmeformaati

Andmete saatmine toimub läbi TXD (*Transmit data signal*) signaali, mis on ühendatud vastuvõtja RXD (*Receive data signal*) signaali külge. Andmete vastuvõtmine käib läbi RXD signaali, mis on ühendatud saatja TXD signaali külge. Ühendusi kujutab joonis 5 ning sensori UART seadeid kujutab tabel 1.



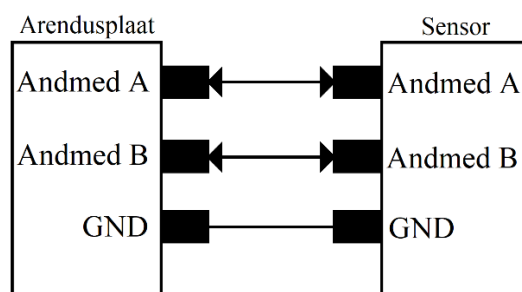
Joonis 5. TXD ja RXD

Tabel 1. Bosch BNO055 UART seaded

Boodikiirus	115200 bitti/s
Sõna pikkus	8 Bitti (sealhulgas paarsus)
Stoppbitt	1
Maksimaalne pikkus lugemiseks ja kirjutamiseks	128 baiti

#### 2.4.2 RS-485 protokoll

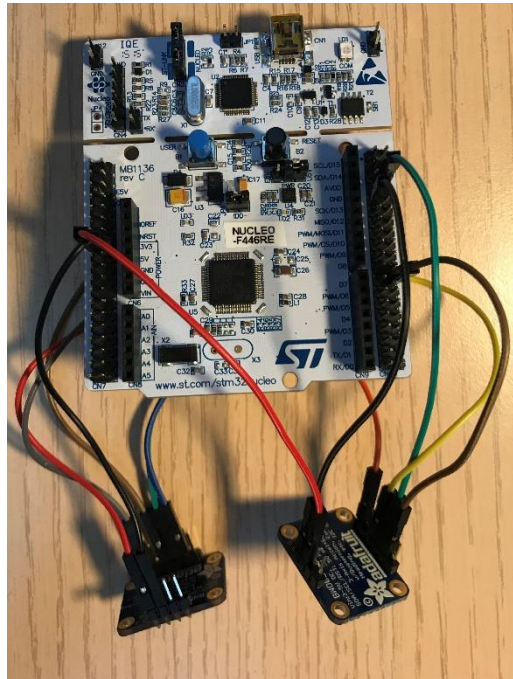
RS-485 protokoll kasutab sama loogikat nagu UART, kuid erineb liinidraiveri poolest. Liinidraiver konverteerib ühesuunalise UART signaali kahesuunaliseks, mis annab kaks andmeliini. Kahesuunalise signaali eeliseks on see, et süsteemil on parem mürataluvus ning lubab seejuures pikemaid ühendusi. RS-485 protokoll maksimaalne andmevahetuse kaugus on ~1200 m. Protokoll võimaldab ühel siinil ühendada kuni 32 seadet [14]. RS-485 ühendus on näidatud joonisel 6.



Joonis 6. RS-485 ühendus

## 2.5 Arendusplaadi ja sensori vahelised ühendused

STM32F446RE arendusplaadil on 76 viiku. Arendusplaadi ja sensori vahel on ühendus tagatud kaablitega. Kasutusel on kokku 4 UART liidest. 2 UART liidest (USART1 ja USART3) suhtlevad sensoritega. Ülejäänud 2 (USART2 ja USART6) on andmete kuvamiseks ning silumiseks. Arendusplaat koos sensoritega on kujutatud joonisel 7. USART1, USART3, USART2, USART6 ühendusi kajastab tabel 2.



Joonis 7. Arendusplaat sensoritega

Tabel 2. Arendusplaadi ja sensorite ühendused

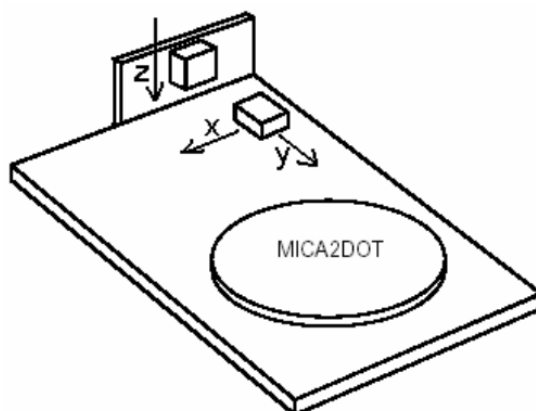
Arendusplaat	Kiirendusandur 1 USART1	Kiirendusandur 2 USART3	USART2 (Silumine)	USART6 (Kiirendusandur)
GND	PS0	PS0	-	-
+3V3	PS1	PS1	-	-
+3V3	Vin	Vin	-	-
GND	GND	GND	-	-
PA10	SDA	-	-	-
PA9	SCL	-	-	-
PC5	-	SDA	-	-
PB10	-	SCL	-	-
PA2	-	-	TX	-
PA3	-	-	RX	-
PC7	-	-	-	RX
PC6	-	-	-	TX

## 2.6 Ülevaade sarnastest lahendustest

Kukkumise tuvastusseadmeid on mitmeid, kuid sellist toodet, mis vastab nõuetele ei leidu. Järgnevalt on toodud ülevaade sarnastest lahendustest.

### 2.6.1 Wearable sensors for Reliable fall detection

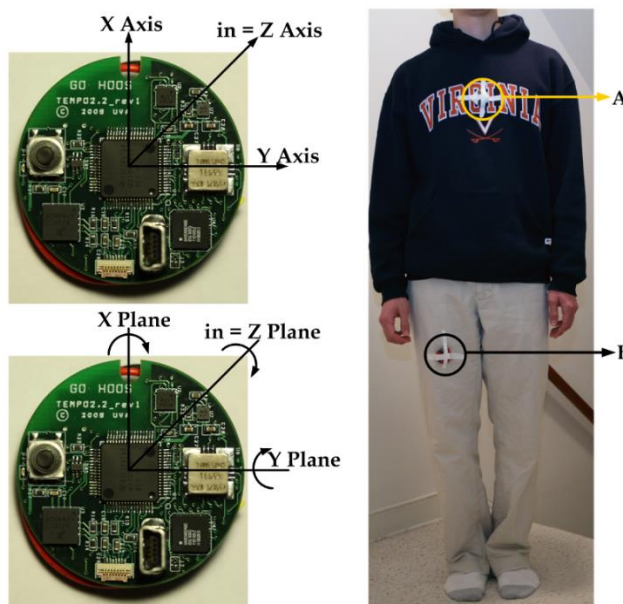
Tegu on *Ad-hoc* tüüpi võrguga, mis tähendab, et on üks baasseade, mille külge on ühendatud teised seadmed. Sensorid pannakse keha külge ning baasseadmele saadetakse andmed. Andmete töötlus toimub baasseadmes. Lahendusel on mõned puudused. Kuna andmetöötlus toimub baasjaamas, tähendab see seda, et kui on palju sensoreid, siis andmetöötlus aeglustab kogu protsessi. Andmed kuhjuvad ning võib juhtuda, et selleks hetkeks kui inimene kukub, toimub parajasti mingi teise sensori andmetöötlus. Lisaks võib esineda ka andmete kadu. Sensoriks kasutatakse MICA2DOT 2 sensori [15], mis kasutab operatsioonisüsteemiks TinyOS 1.0 [16] [17]. Seadet näitab joonis 8.



Joonis 8. Kantav sensor kukkumise tuvastamiseks [17]

### 2.6.2 Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information

Kasutatakse nii kiirendusandurit kui ka güroskoopi. See võimaldab tuvastada istumised ning tõusmised, mis on andmete poolest justkui kukkumised. Kasutatakse mitut sensorit, mille abil saab tuvastada kehahoiakuid: seismine, istumine, venitamine ning lamamine. Selleks ühendatakse üks sensor rinna külge ning teine jala külge. Lahendus on hea, sest nii on tagatud rohkem tõepäraseid kukkumisi. Võimalik, et eeltoodud lahendust edasi arendades leiaks seade kasutust ka laeva peal [18]. Seadet kajastab joonis 9.



Joonis 9. Kehahoiakut arvestav kukkumise tuvastus [18]

### 2.6.3 Philips GoSafe

Tegemist on seadmega, mis on välja töötatud nii sise- kui ka välistingimustes. Asukoha tuvastamiseks on kasutusel WIFI (*Wireless fidelity*), GPS (*Global positioning system*), helialarm ning lisaks salvestatakse aegajalt asukohta, kasutades selleks GPS'i. Tootel on mõned puudused. Tootel on igakuine maks ning see jaotub pakettidesse. Kallimad paketid pakuvad paremaid lahendusi. Kui tegu on mitme inimesega, osutub see kulukaks. Hädakõne ei suunata hädaabisse, vaid reageerimiskeskusele, kus uuritakse, mis inimesel täpsemalt juhtus ning seejärel võtab tugiisik ühendust hädaabiga [19]. Seade on näidatud joonisel 10.



Joonis 10. Philips GoSafe [19]

#### 2.6.4 Shimmer sensing

Firma, mis keskendub kantavatele sensoritele. Sensorid on andmete kogumiseks ja analüüsiks. Platvormi kasutatakse prototüübi arenduseks. Kogutakse andmeid ning andmete põhjal töötatakse välja lahendus, mida on võimalik tellida tellimustööna. Antud probleemile saaks neilt lahenduse, kuid see oleks kulukas ja aeganõudev. Juba ainuüksi andmekaabel maksab 39 €. Sobiva arenduskomplekti (Consensys Development Kit) saaks neilt 499 € [20]. Seadmes on altimeeter, kiirendusandur, güroskoop ning magnetometer. Kõik arenduskomplektid on varustatud kõige vajalikuga (rihmad, juhtmed, elektroodid, jne.) [21]. Seade on sobilik testimiseks, seda saab kasutada referentsandmete kogumiseks. Seadet kujutab joonis 11.



Joonis 11. Consensys Development Kit [21]

### 3 Tarkvara

Programmikood on kirjutatud C keeles. C keel on laialdaselt kasutatud süsteemne programmeerimiskeel. Keel loodi aastatel 1977 – 1979 paralleelselt operatsioonisüsteemiga UNIX. Esialgu kasutati keelt UNIX süsteemi jaoks, kuid tänapäeval on see kujunenud üheks peamiseks arvutitööstuses kasutatavaks programmeerimiskeeleks. C keele eelisteks on tema kiirus ja vähene mälu kasutus [22]. Programmeerimiskeskkonnaks valisin Keil uVision5, sest keskkond on toetatud STMicroelectronics'i poolt.

Kood on kergelt muudetav ja mõistetav. Kiirendusanduri erinevaid osi on võimalik seadistada. Koodi kirjutamisel oli arvestatud, et algoritmi töötlejal oleks vajalikud funktsioonid olemas või nende puudumisel oskaks olemasolevate vahenditega neid luua. Registrileht on defineeritud päises. Kiirendusandurit on võimalik kalibreerida. Lisaks on koodis ka funktsioone, mis aitavad silumisel: registrist lugemine ning kirjutamine.

#### 3.1 STM32CubeMX programm

STM32CubeMX on programm, mis lihtsustab arendaja tööd. Tegemist on tarkvaraga, mis võimaldab genereerida initsialiseerimiskoodi C keeles. Koodi genereerimiseks tuleb läbi graafilise liidese valida projekti jaoks seaded. Lisaks initsialiseerimiskoodile antakse kaasa STM32Cube HAL (*Hardware abstraction layer*) teek [23].

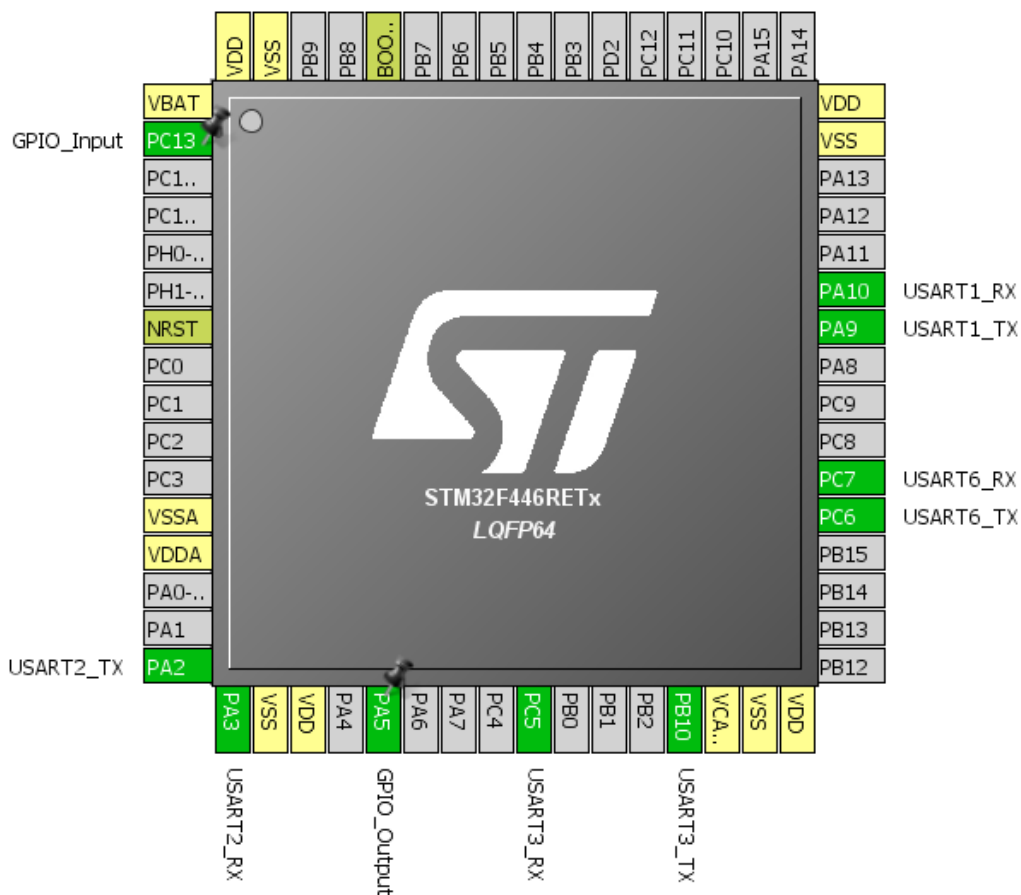
STM32CubeMX programmis on olulised järgmised projekti seaded. IDE (*Integrated development environment*) valik – MDK-ARM V5. Aktiveerida tuleb USART1, USART2, USART3, USART6 ning neile määrata globaalne katkestus, täpsemad seadeid näitab tabel 3. Koodi genereerimise alammenüüst tuleb vaadata, et kasutajakood jääks alles iga kord kui programm genereerib uue koodi. Joonis 12 näitab STM32 projekti seadistusi.

Tabel 3. UART seaded

USART1, USART2, USART3, USART6 seadistused	
Boodikiirus	115200 bitti/s



Sõna pikkus	8 Bitti (sealhulgas paarsus)
Paarsus	Puudub
Stopp bitte	1
Globaalne katkestus	Aktiveeritud



Joonis 12. STM32CubeMX projekti seaded

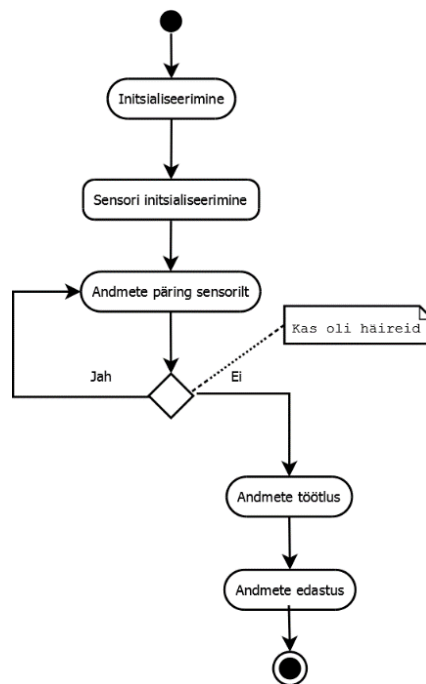
### 3.1.1 STM32Cube HAL teek

STM32Cube HAL teek on STMicroelectronics'i initsiatiivil arendatud draiver. Draiver pakub kasutamiseks API'se (*Application programming interface*), mis lihtsustavad kasutajat rakenduse loomisel. Iga draiver koosneb funktsioonide hulgast, mis katavad tavapärasel vajadused. Näiteks saab teegiga initsialiseerida ning seadistada UART liidest, hallata andmevahetust, töödelda katkestusi või DMA'si (*Direct Memory Access*) ning hallata vigu [24].

## 3.2 Programmi töö

Esmalt initsialiseeritakse kõik sisemised seadistused arendusplaadil, mis saadakse STM32CubeMX programmist. Seejärel saadetakse sensorile seadistused. Peale sensori initsialiseerimist hakkab arendusplaat pärima sensorilt kiirendusandmeid. Juhul kui andmed olid vigased, küsitakse andmed sensorilt uuesti.

Andmete kontroll toimub järgnevalt. X, Y, Z koordinaatide jaoks tuleb kokku küsida 6 andmebaiti. On võimalik, et sensor saadab andmete asemel veateate. Viga võib tekkida, mis tahes hetkel 6 baidi pärimisel. Kui üks bait kuuest on vigane, ei saa andmeid töödelda. Täpsema info, kuidas toimub andmete pärimine, saab lugeda peatükis 3.5. Programmi tööd kirjeldab joonis 13.



Joonis 13. Programmi töö

Juhul, kui andmete päring õnnestus, toimub andmete töötlus. Loetavate andmete saamiseks tuleb kokku panna 2 baiti. Täisarv koosneb kahest baidist, kus esimene bait tähistab vähima kaaluga baiti ning teine bait suurimat. Tabel 4 on toodud näide MNB (*Most significant bit*) ja LSB (*Less significant bit*) andmetöötlustest.

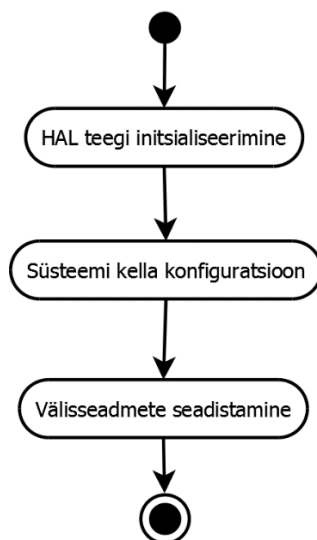
Tabel 4. MSB ja LSB andmetöötlus

Andme tüüp	MSB	LSB	Töödeldud kujul
Andmete ühiku suurus	8 bitti	8 bitti	16 bitti
Binaar	1111 1101	1110 0010	1111 1101 1110 0010
Kuueteistkümnendsüsteem	0xFD	0xE2	0xFDE2
Detsimal	253	226	-770

### 3.3 Arendusplaadi initsialiseerimine

Arendusplaadi initsialiseerimisel käivitatakse esmalt HAL teek. Seejärel seadistatakse süsteemitakt. Initsialiseeritakse sisemine väljundpinge, CPU (*Central processing unit*)-, AHB (*Advanced High-performance Bus*)- ja ABP (*Alternating bit protocol*) siini taktid.

Seejärel toimub väliseadmete seadistus. Käivitatakse kõik konfigureeritud väljundid: GPIO, UART, USART. Arendusplaadi initsialiseerimine on näidatud joonisel 14.

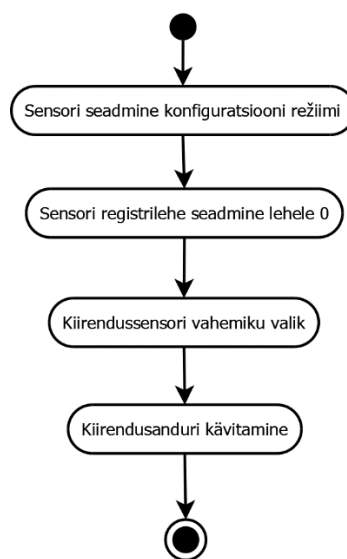


Joonis 14. Arendusplaadi initsialiseerimine

### 3.4 Sensori initsialiseerimine

Sensori käivitamisel saadetakse sensorile konfiguratsiooniseaded, milliste seadetega sensor tööle hakkab. Juhul, kui tekib vajadus muuta seadistust jooksvalt, tuleb selleks sensor viia uuesti konfiguratsiooni režiimi.

Sensori initsialiseerimiseks lülitakse sensor konfiguratsiooni režiimi. Sensorile saadetakse kiirendusanduri vahemik 8G, temperatuuriühikuks celsiuse kraadid ning seejärel käivitatakse sensor kiirendusanduri režiimis. Sensori initsialiseerimist kirjeldab joonis 15.



Joonis 15. Sensori initsialiseerimine

### 3.5 Andmete päring sensorilt

Andmete vastuvõtmiseks sensorilt tuleb saata päring üle UART liidese. Nii lugemisel kui ka kirjutamisel on algusbait sama (0xAA). Teine bait määrab, kas tegemist on lugemise või kirjutamisega. Juhul, kui on tegemist kirjutamisega, on teise baidi sisuks 0x00. Kui tegu on lugemisega, on sisuks 0x01.

#### 3.5.1 Andmete lugemine sensorilt

Teine bait määrab, kas tegemist on lugemise või kirjutamisega. Lugemisel on teise baidi sisuks 0x01. Lugemiseks võib küsida maksimaalselt 128 baiti andmeid, sest ühel registrilehel on kokku 128 registriaadressi. Näiteks, kui on vaja lugeda kiirendusanduri X, Y, Z koordinaate, siis saab seda teha ühe päringuga või mitme päringuga.

Üheks päringuks tuleb näidata neljanda baidiga, et soovitakse lugeda 6 baiti. Kolmas bait, mis näitab registriaadressi, peab sellisel juhul viitama registri X-koordinaadi LSB registrile. Alternatiivina võib teha 6 päringut, küsides igat baiti eraldi, mis on aeglasem ning andmed võivad muutuda järgmise baidi küsimisel. Mõistlikum on seda teha ühe päringuga, sellisel juhul on vigaste andmete tõenäosus väiksem. Lugemise päring on kirjeldatud tabelis 5.

Tabel 5. Lugemise päring

Bait 1	Bait 2	Bait 3	Bait 4
Algusbait	Lugemine	Registriaadress	Pikkus
0xAA	0x01	<..>	<..>

### 3.5.1.1 Õnnestunud lugemise vastus sensorilt

Lugemise kinnituseks saadab sensor vastuse. Õnnestunud lugemisel saadetakse vähemalt 3 baiti. Kui tegemist on õnnestunud lugemisega, algab esimene bait 0xAA'ga.

Õnnestunud lugemist kujutab tabel 6.

Tabel 6. Registrist õnnestunud lugemise vastus

Bait 1	Bait 2	Bait 3	...	Bait (n + 2)
Algusbait	Pikkus	Andmed 1	...	Andmed n
0xBB	<..>	<..>	...	<..>

### 3.5.1.2 Nurjunud lugemise vastus sensorilt

Juhul kui lugemine nurjus algab esimene bait 0xEE'ga. Järgnev bait annab mõista, mis tüüpi veaga on tegu. Registrist lugemise veateadet on kujutatud tabelis 7.

Tabel 7. Registrist lugemise veateade

Bait 1	Bait 2
Algusbait	Staatus
0xEE	0x02: Lugemine nurjus 0x04: Vale Registriaadress 0x05: Registri kirjutamine väljalülitud 0x06: Vale algusbait 0x07: Siini ülekoormuse viga 0x08: Maksimaalse pikkuse viga 0x09: Minimaalse pikkuse viga 0x0A: Vastuvõetud märgi taimaut

### 3.5.2 Sensori registrisse kirjutamine

Nagu eelnevalt mainitud, määrab teine bait, et tegu on kirjutamisega. Kirjutamisel on teise baidi sisuks 0x00. Kirjutada võib maksimaalselt 128 baiti andmeid, sest ühel registrilehel on 128 registriaadressi. Süsteemiseadete muutmiseks tuleb kirjutamisel veenduda, et sensor on viidud konfiguratsiooniseadete režiimi.

Kui on vaja kirjutada kiirendusanduri kalibreerimiseks X, Y, Z koordinaatide nihe, saab seda teha sarnaselt lugemisega. Kas ühe päringuga või mitme päringuga.

Samamoodi nagu lugemisel tuleb kirjutamisel näidata neljanda baidiga, et soovitakse kirjutada 6 baiti. Kolmas bait peab viitama registri X koordinaadi LSB nihke registriaadressile. Mõistlikum on seda teha ühe päringuga, sellisel juhul on vigaste andmete tõenäosus väiksem. Kirjutamise päring on näidatud tabelis 8.

Tabel 8. Kirjutamine

Bait 1	Bait 2	Bait 3	Bait 4	Bait 5	...	Bait (n + 4)
Algusbait	Kirjutamine	Registriaadress	Pikkus	Andmed 1	...	Andmed n
0xAA	0x00	<..>	<..>	<..>	...	<..>

### 3.5.2.1 Sensori registrisse kirjutamise vastus

Erinevalt lugemisest on kirjutamisel kinnituseks vaid üks vastus. Õnnestumise või nurjumise korral saadetakse 2 baiti, millest esimene on nagu nurjunud lugemise korral (0xEE). Eristamiseks peab vaatama teise baidi sisu.

Kirjutamise õnnestumist kajastab staatus 0x01, teistel puhkudel on tegemist veateadetega. Kirjutamise vastus on kujutatud tabelis 9.

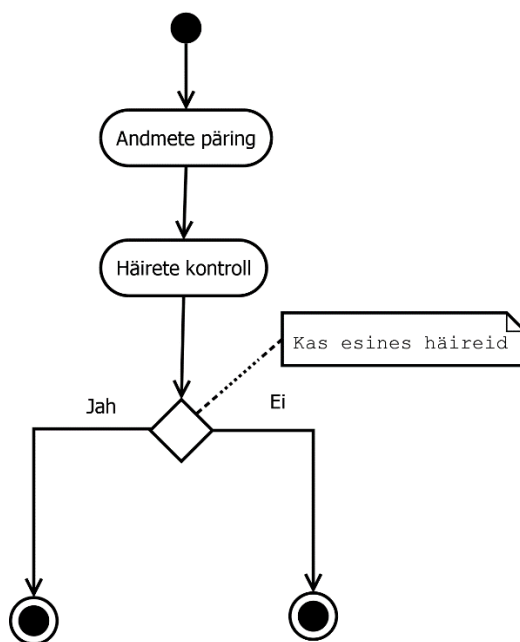
Tabel 9. Kirjutamise vastus

Bait 1	Bait 2
Vastuse algusbait	Staatus
0xEE	0x01: Kirjutamine õnnestus  0x03: Kirjutamine nurjus  0x04: Vale registriaadress  0x05: Registri kirjutamine väljalülitud  0x06: Vale algusbait  0x07: Siini ülekoormuse viga  0x08: Maksimaalse pikkuse viga  0x09: Minimaalse pikkuse viga  0x0A: Vastuvõetud märgi taimaut



### 3.6 Häirete kontroll

Peale andmete päringut kontrollitakse, kas andmed olid õiged. Juhul, kui algusbait on 0xBB, on tegemist vigadeta andmetega. Juhul, kui algusbait on 0xEE, on tegemist nurjunud lugemisega. Vigase lugemise korral andmeid ei töödelda. Häirete kontrolli kujutab joonis 16.



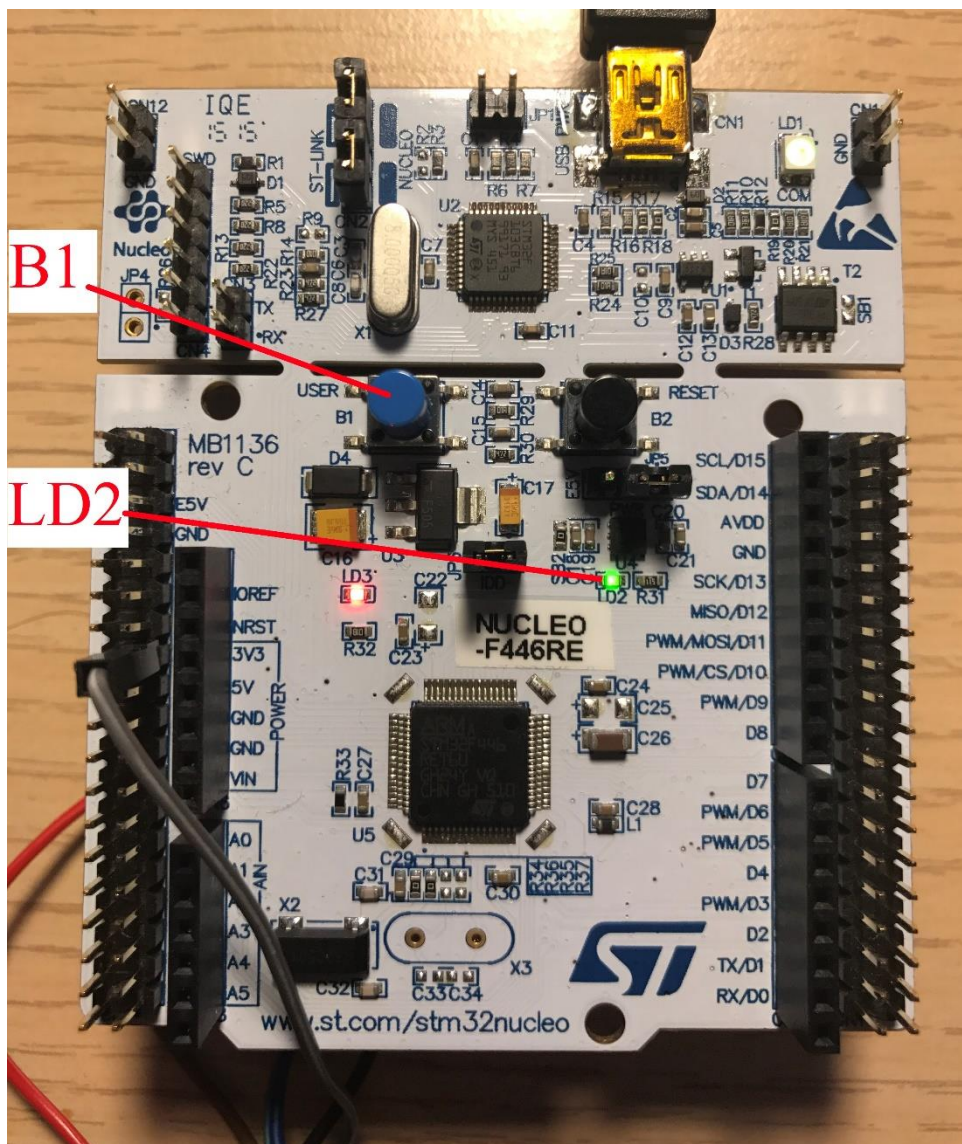
Joonis 16. Häirete kontroll

### 3.7 Kiirendusanduri kalibreerimine

Kiirendusanduri kalibreerimine on jaotatud kuueks etapiks:

- 1) Sinist nuppu (B1), mis on näidatud joonisel 17, tuleb hoida all viis sekundit.
- 2) Seejärel on kahesekundiline paus.
  - a) Juhul, kui selle kahe sekundi jooksul vajutada sinist nuppu, seadistatakse teist sensorit.
  - b) Kui nuppu ei vajutata, seadistatakse esimest sensorit.
- 3) Esmalt seadistatakse sensori X-telge. LED LD2, mis on näidatud joonisel 17, hakkab vilkuma intervalliga 1000 ms. Sensor tuleb paigutada X-telje suhtes tasasele pinnale ning seejärel tuleb vajutada sinist nuppu B1.

- 4) Järgmisena seadistatakse sensori Y-telge. LED LD2 hakkab vilkuma intervalliga 400ms. Sensor tuleb paigutada Y-telje suhtes tasasele pinnale ning seejärel tuleb vajutada sinist nuppu B1.
- 5) Viimasena seadistatakse sensori Z-telge. LED LD2 hakkab vilkuma intervalliga 100 ms. Sensor tuleb paigutada Z-telje suhtes tasasele pinnale ning seejärel tuleb vajutada sinist nuppu B1.
- 6) Järgnevalt saadetakse sensorile kalibreerimisandmed, mille tulemusel saab sensor seadistatud.



Joonis 17. Arendusplaadi kalibreerimise nupp B1 ja LED LD2

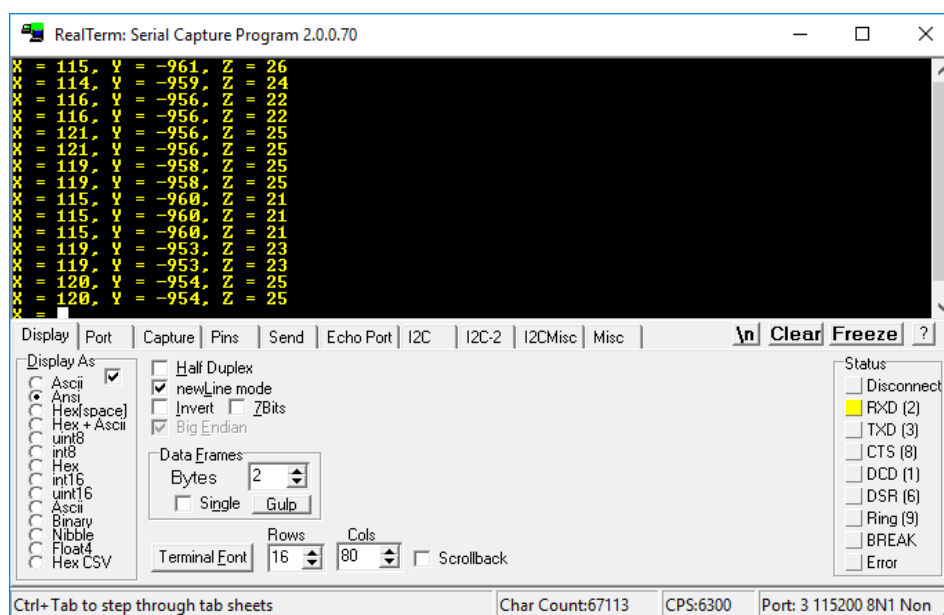
## 4 Testimine ja katsed

Süsteem saadab mõõteandmed edasiseks analüüsiks läbi USART 2 pordi, mida saab monitoorida programmiga RealTerm [25].

Selleks tuleb seadistada RealTerm järgmiste seadetega:

- Boodikiirus – 115200
- Port – USB (*Universal serial bus*)
- Andmete kuvamise formaat – ASCII (*American standard code for information interchange*)
- Uue rea režiim - aktiveeritud.

Andmed on kuvatud joonisel 18.



Joonis 18. RealTerm andmete kuvamine

Testide läbiviimiseks kasutasin programmi STM Studio [26]. Programm pakub võimalust monitoorida ja visualiseerida andmeid. Joonistel on punaselt tähistatud X-telg, roheliselt Y-telg ning siniselt Z-telg. Tabelites esitatud amplituudid on mG-des, mis on  $\frac{1}{1000} G$ , 1 G on  $9.8 \frac{m}{s^2}$ . Kokku viisin läbi kolmteist testi enda peal.

Katsete eesmärgiks oli veenduda, et seade töötab ning ei esine tõrkeid. Samuti tuli kontrollida, et kiirendusanduri töövahemik oleks sobilik kukkumise tuvastamiseks.

Testid jagunevad:

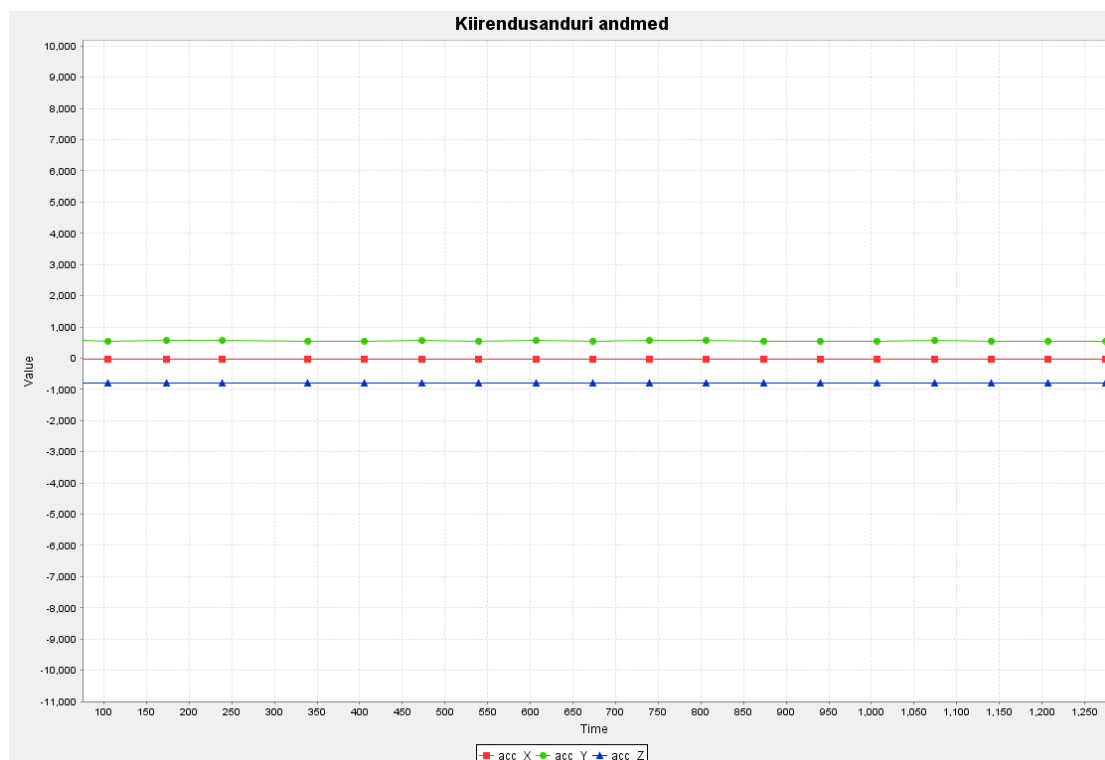
- 1) Paigalolek – 2 katset
- 2) Hüppamine – 3 katset
- 3) Istumine ja tõusmine – 2 katset
- 4) Erinevad liikumised – 3 katset
- 5) Kukkumine ja põrutus – 4 katset

## 4.1 Paigaloleku katse

Paigal olles viisin läbi kaks katset, üks seistes ning teine kiikudes, mis imiteerib laineid. Katse, kus seisin paigal, on kujutatud joonisel 19. Jooniselt on näha, et X-, Y-, Z-telg on paigal. Suurim amplituud on kajastatud tabelis 10.

Tabel 10. Paigaloleku suurim amplituud

X telg	Y telg	Z telg
0	0	0

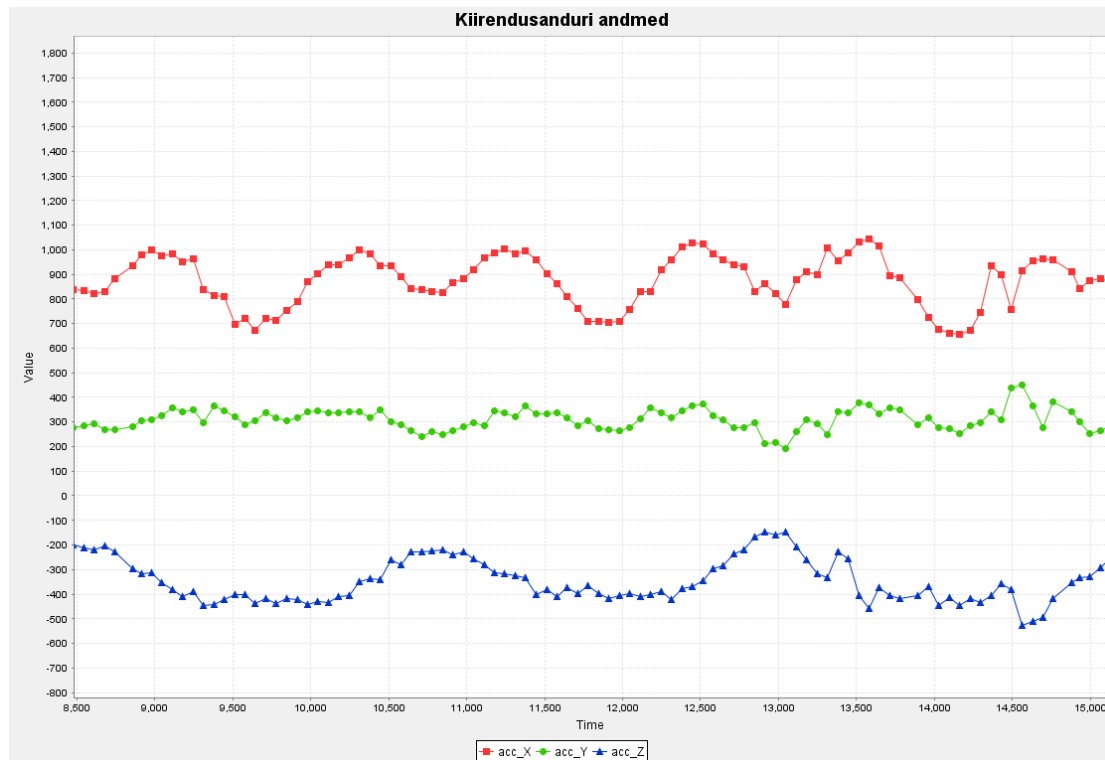


Joonis 19. Paigalolek

Kiikumiskatsel, mis on näidatud joonisel 20, on näha, et telgede väärtused muutuvad sujuvalt. Suurim amplituud on näidatud tabelis 11.

Tabel 11. Kiikumise suurim amplituud

X telg	Y telg	Z telg
230	160	220



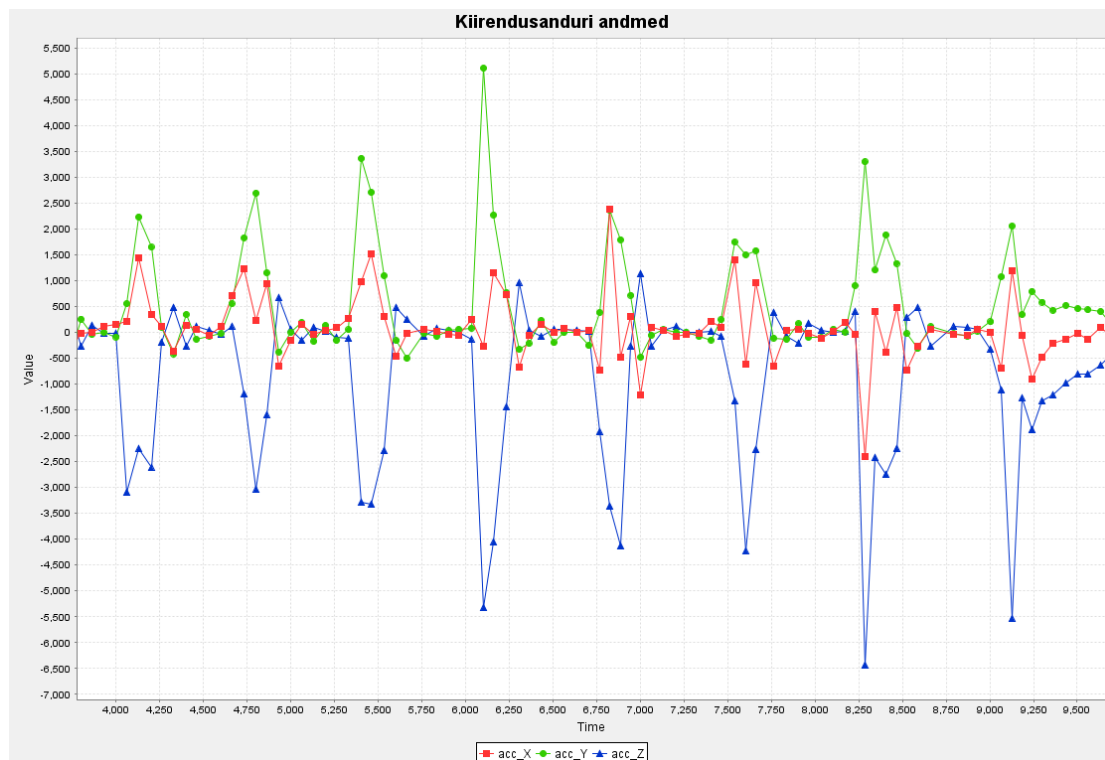
Joonis 20. Kiikumine

## 4.2 Hüppamise katse

Hüppamise jaoks viisin läbi kolm testi, millest esimene katse oli hoovõtuta hüppamine murul. Sooritasin murul kaheksa hüpet, mis on kujutatud joonisel 21. Joonisel on näha kaheksat amplituudi muutust, milleks olid hüpped. Suurimaid amplituude kajastab tabel 12.

Tabel 12. Murul hüppamise suurim amplituud

X telg	Y telg	Z telg
2400	5100	6500

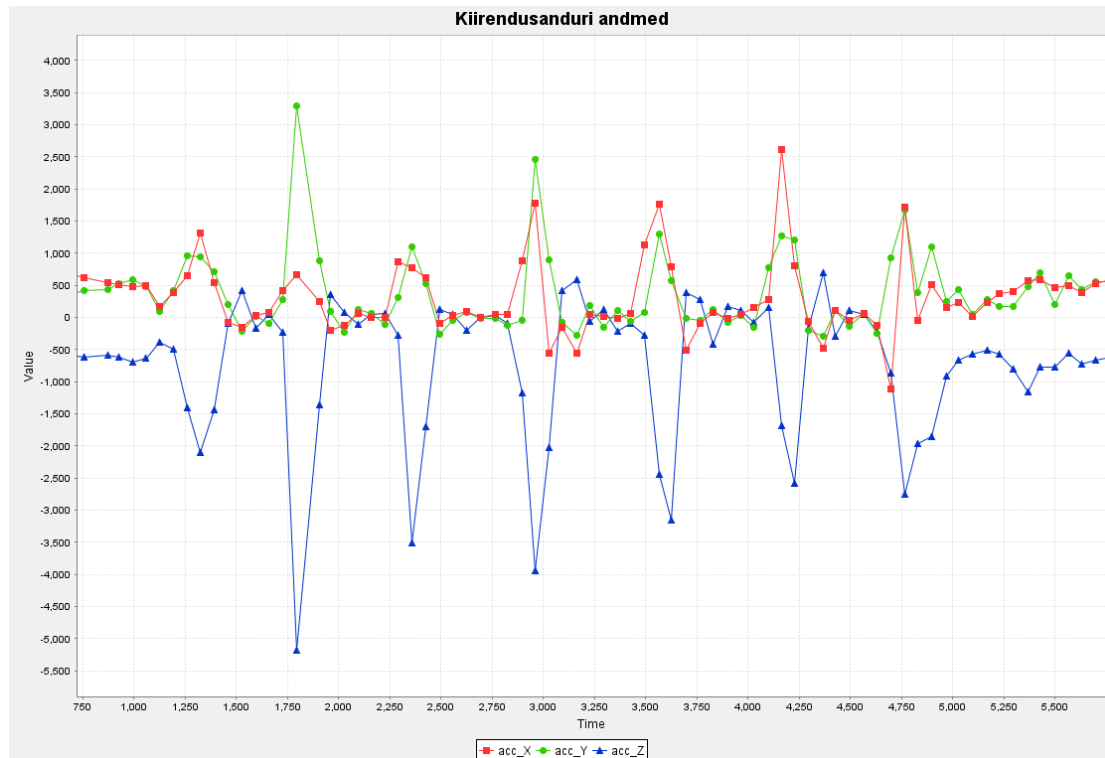


Joonis 21. Hüppamine murul

Teine katse on joonisel 22. Jooniselt nähtub seitse hüpet, mis on sooritatud puitpinnal. Võib öelda, et hüppamine puidul ei erine hüppamisest murul. Suurimaid amplituude on kajastatud tabelis 13.

Tabel 13. Puidul hüppamise suurim amplituud

X telg	Y telg	Z telg
1900	3400	5200

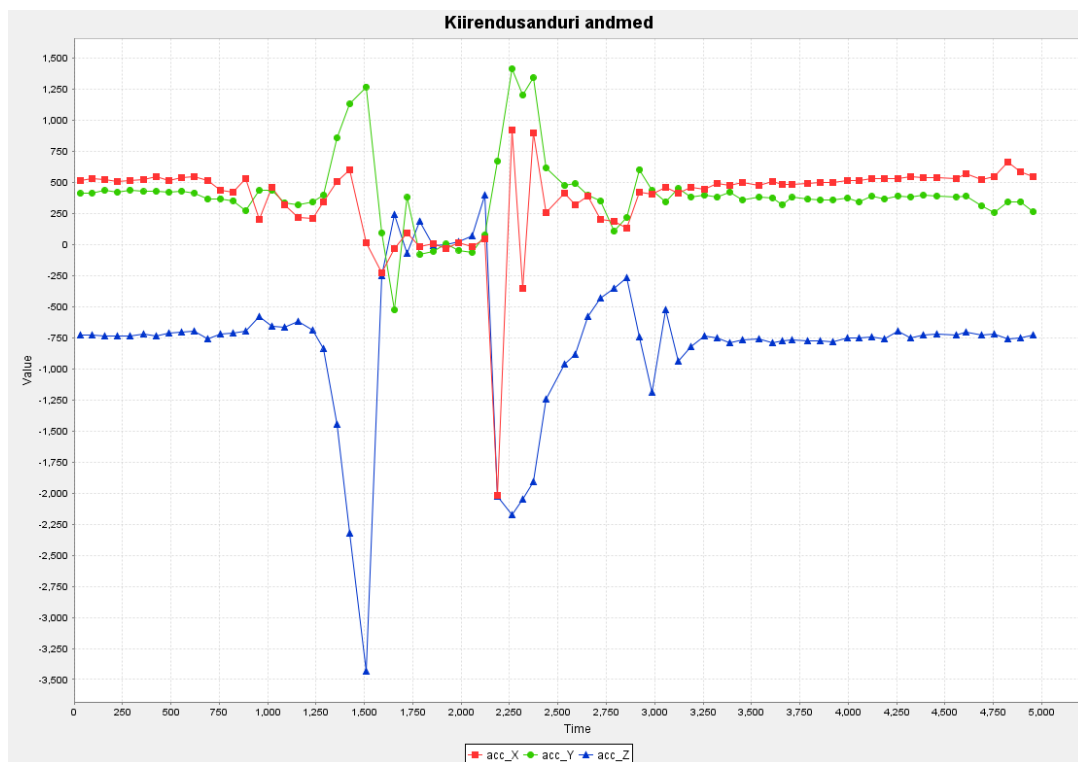


Joonis 22. Hüppamine puidul

Viimast katset, mis toimus puitaluselt hüppamisel, on näidatud joonisel 23. Suurimad amplituudid on kajastatud tabelis 14. Jooniselt on näha, et ajaliselt oli hüpe võrreldes teiste hüpetega kõige pikem. Esimene tõus joonisel on hüpe õhku ja teine tõus maandumine.

Tabel 14. Hüppamine puitaluselt murule suurim amplituud

X telg	Y telg	Z telg
750	1100	-2750



Joonis 23. Hüppamine puitaluselt murule

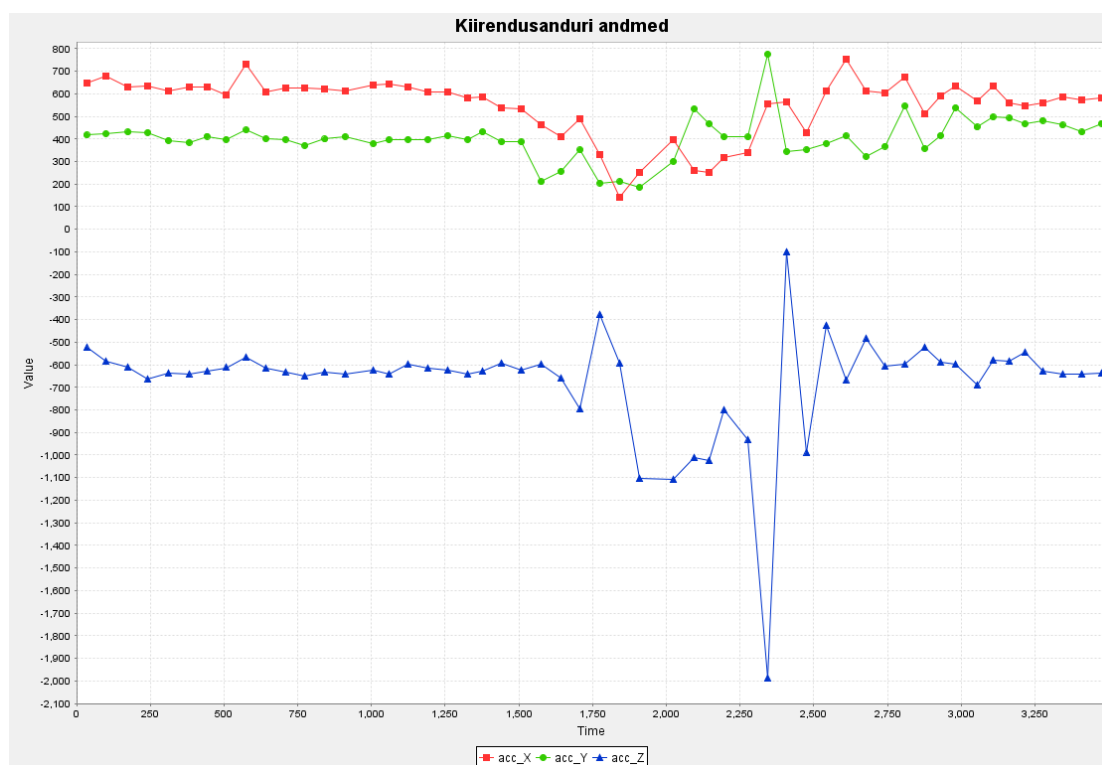


### 4.3 Istumise ja tõusmise katse

Sooritasin istumiseks ja tõusmiseks kaks katset, millest esimene oli istumine. Katse sooritamisel istusin toolile, mis oli valmistatud puidust. Istumist kujutab joonis 24 ning suurimad amplituudid on toodud tabelis 15.

Tabel 15. Istumise suurim amplituud

X telg	Y telg	Z telg
550	400	1500

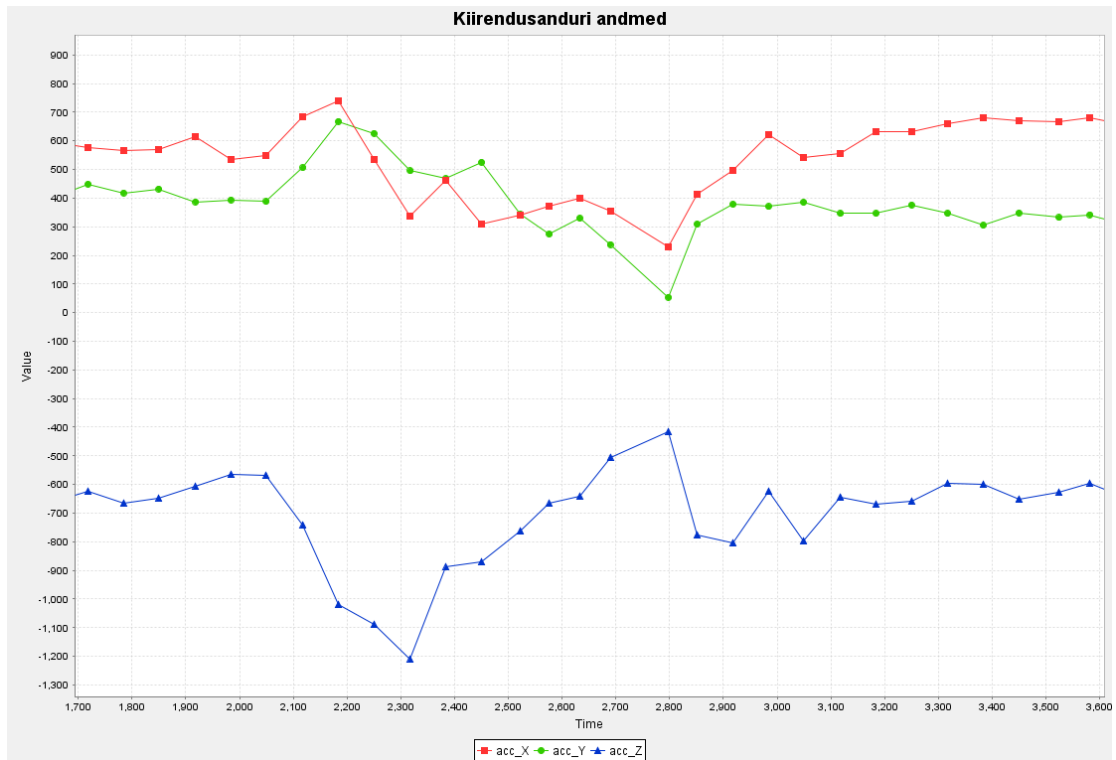


Joonis 24. Istumine

Tõusmiskatsel istusin toolil ning tõusin püsti, mis on näidatud joonisel 25. Suurimad amplituudi muutused on näidatud tabelis 16. On näha, et tõusmisel ei esine nii suurt amplituudi muutust kui istumisel.

Tabel 16. Tõusmise suurim amplituud

X telg	Y telg	Z telg
400	380	600



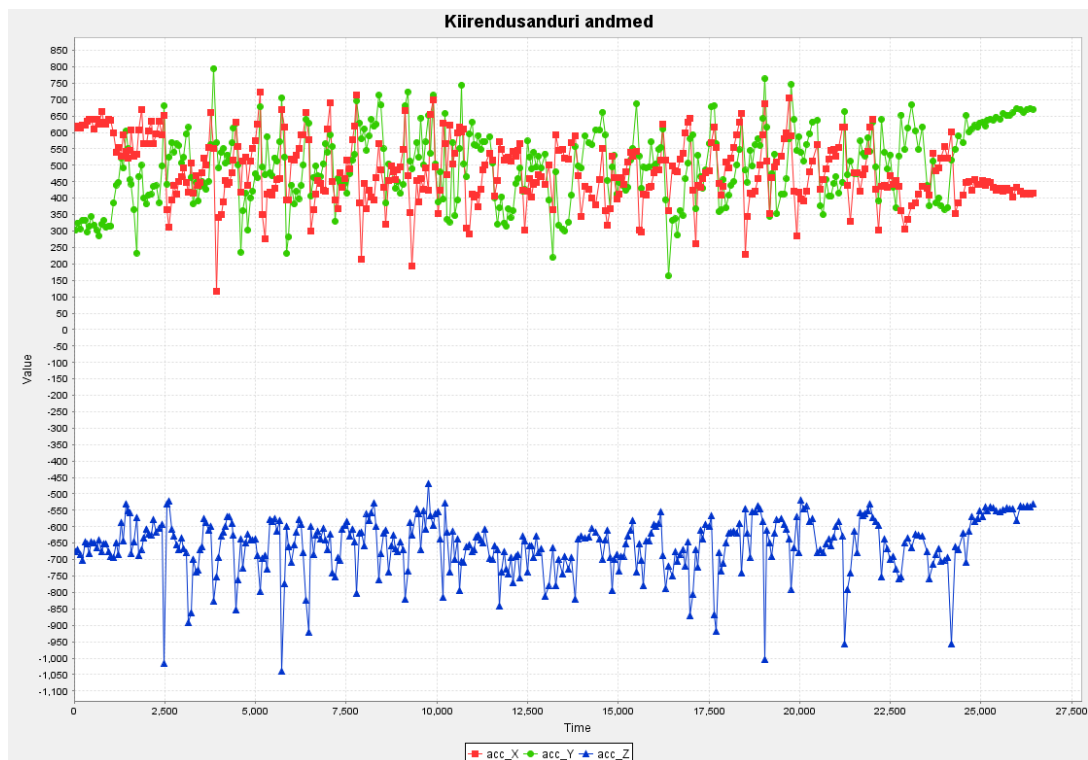
Joonis 25. Tõusmine

## 4.4 Erinevad liikumised

Liikumiste jaoks sooritasin kolm katset. Millest esimene oli kõndimine. Kõndimiskatse sooritasin tasasel pinnal toas, katset näitab joonis 26 ning amplituudid on toodud tabelis 17. Kõndimisel on näha, et amplituudi muutus pole suur.

Tabel 17. Kõndimise suurim amplituud

X telg	Y telg	Z telg
500	500	350

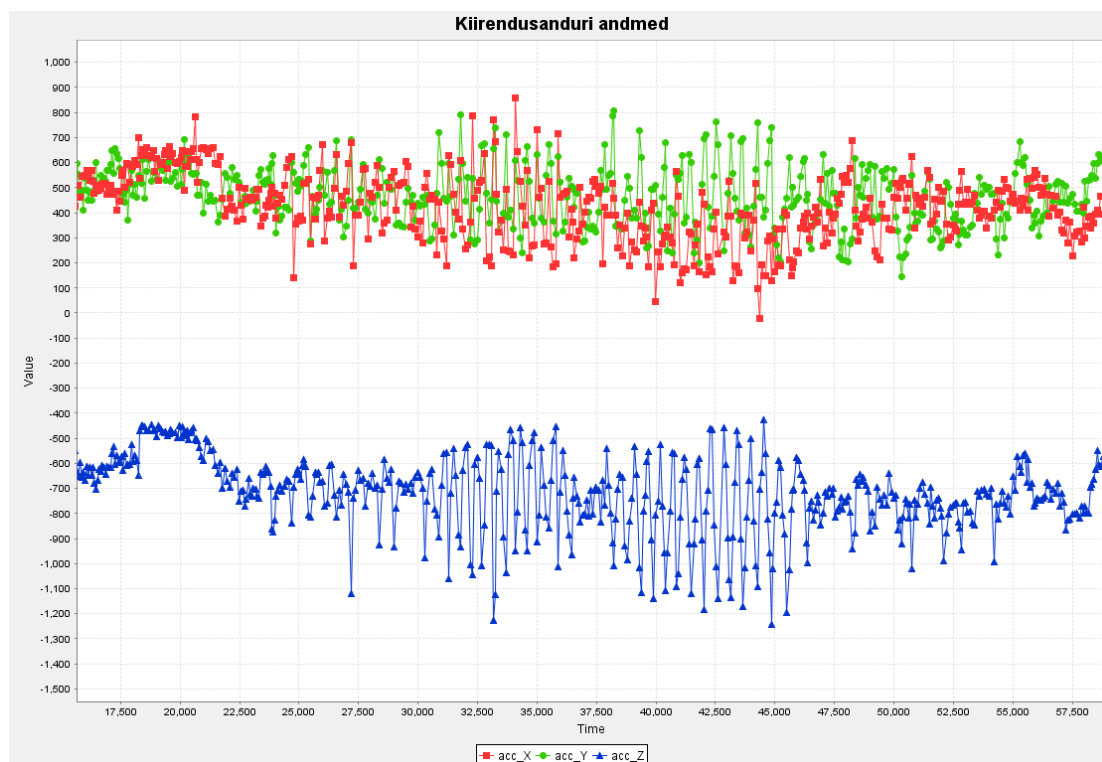


Joonis 26. Kõndimine

Teiseks katseks kõndisin puidust trepil üles ja alla, mida näitab joonis 27. Suurimad amplituudid on toodud tabelis 18.

Tabel 18. Trepil kõndimise suurim amplituud

X telg	Y telg	Z telg
400	500	700

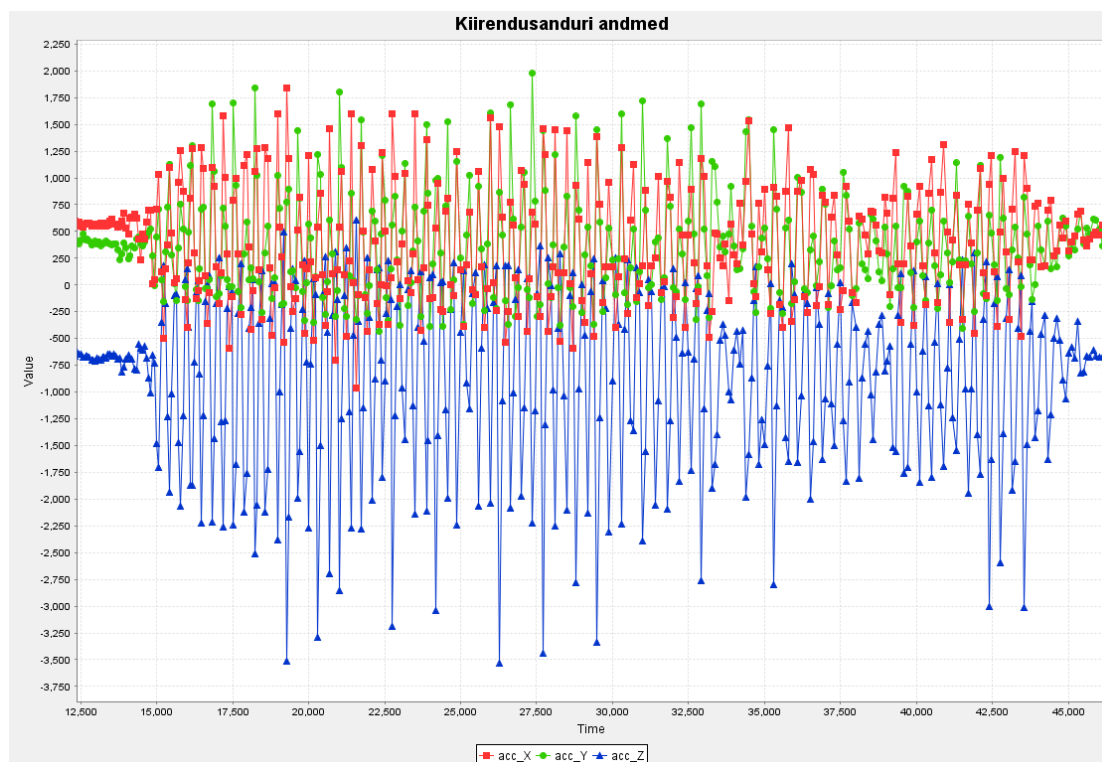


Joonis 27. Trepil kõndimine

Viimaseks katseks valisin jookmise. Jookmiskatse viisin läbi murul. Amplituudi kajastab tabel 19 ning joonis 28 kirjeldab katset.

Tabel 19. Jookmise suurim amplituud

X telg	Y telg	Z telg
1200	1500	2750



Joonis 28. Jooksmine

## 4.5 Kukkumise ja pörutuse katse

Kukkumiseks ja pörutuseks sooritasin kokku kolm katset, millest kaks olid kukkumist ning üks pörotus vastu maja. Esimeseks katseks sooritasin püsti seistes kukkumise murule. Kukkumise andmed on toodud joonisel 29 ning amplituud tabelis 20. Nagu näha on kukkumise amplituud joonisel kõige ilmekam.

Tabel 20. Esimese murul kukkumise amplituud

X telg	Y telg	Z telg
7500	1750	1750

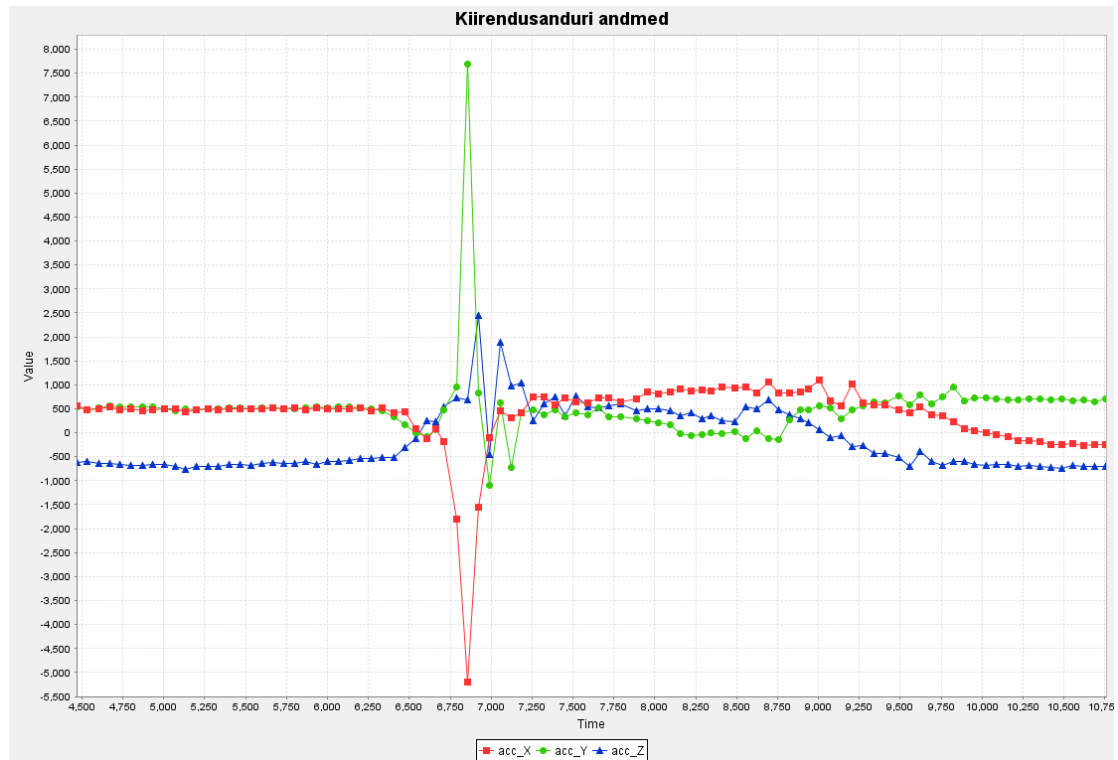


Joonis 29. Esimene murul kukkumine

Teiseks kukkumiseks sooritasin sama katse, kuid kukkumine oli tugevam. Kukkumist kirjeldab joonis 30 ning amplituudi näitab tabel 21.

Tabel 21. Teise murul kukkumise amplituud

X telg	Y telg	Z telg
5500	8800	3000

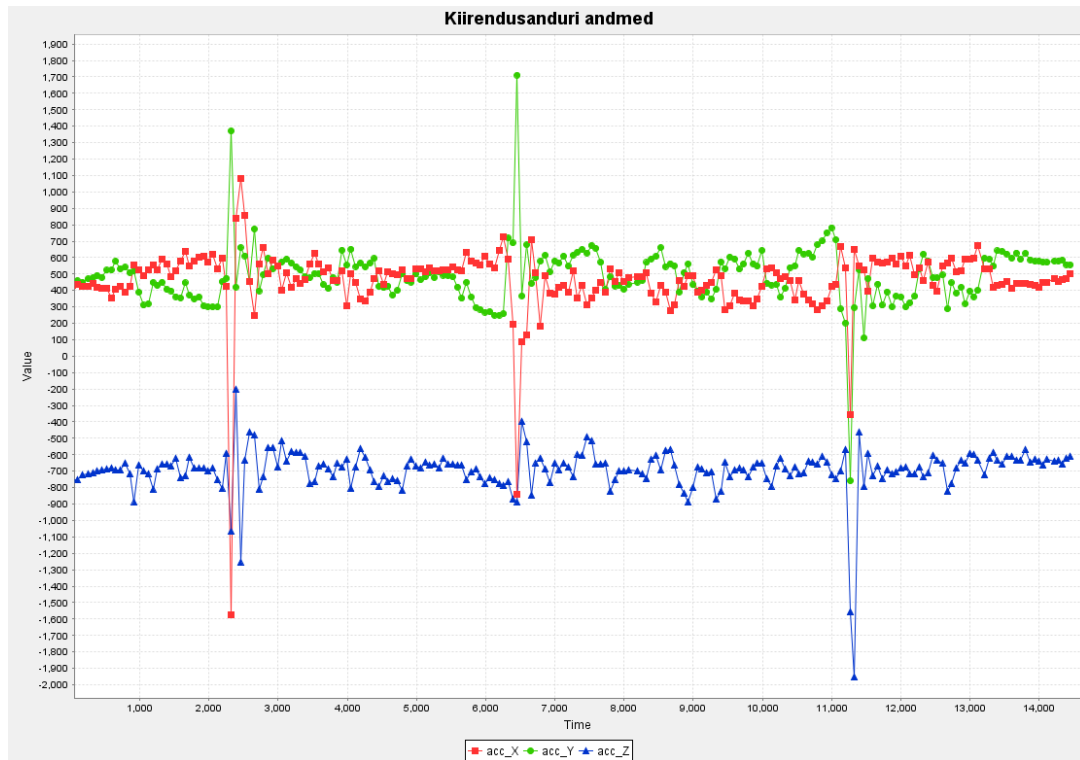


Joonis 30. Teine murul kukkumine

Viimaseks katseks oli põrutus vastu seina, mille andmed on joonisel 31 ning amplituud tabelis 22.

Tabel 22. Vastu seina põrutuse amplituud

X telg	Y telg	Z telg
2700	1200	1300



Joonis 31. Põrutus vastu seina



## 5 Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli arendada kukkumise tuvastamise süsteem, mille abil saaks välja töötada kukkumise tuvastamise algoritmi.

Prototüübi riistvara arenduseks valiti välja STMicroeletronic'i mikrokontroller STM32F446RE ning kiirendusandurid Bosch BNO055. Suhtluseks arendusplaadi ja sensorite vahel kasutati UART protokollit. Veakindlaks andmevahetuseks soovitatakse asendada UART RS-485'ga.

Mikrokontroller on seadistatud töötamiseks kahe kiirendusanduriga, kuid vajadusel saab ühendada kuni neli sensorit.

Prototüüp on seadistatud vastavalt algoritmi nõuetele ning vajadusel saab kiirendusandureid kalibreerida.

Töö käigus valmis seade, mille abil on võimalik töötada välja kukkumise tuvastamise algoritmi. Samuti saab kiirendusandureid kalibreerida.

## Kasutatud kirjandus

- [1] STM32 -F446RE mikrokontroller [WWW] <http://www.st.com/en/microcontrollers/stm32f446.html?querycriteria=productId=LN1875> (16.05.2017)
- [2] Bosch BNO055 sensor [WWW] [https://www.bosch-sensortec.com/bst/products/all\\_products/bno055](https://www.bosch-sensortec.com/bst/products/all_products/bno055) (16.05.2017)
- [3] Silicon Labs koduleht [WWW] <http://www.silabs.com> (16.05.2017)
- [4] Texas Instruments koduleht [WWW] <https://www.ti.com> (16.05.2017)
- [5] STMicroelectronics koduleht [WWW] [http://www.st.com/content/st\\_com/en.html](http://www.st.com/content/st_com/en.html) (16.05.2017)
- [6] Giant Gecko mikrokontroller [WWW] <http://www.silabs.com/products/mcu/32-bit/efm32-giant-gecko> (16.05.2017)
- [7] Texas instruments mikrokontroller TM4C129XNCZAD [WWW] <http://www.ti.com/product/TM4C129XNCZAD> (16.05.2017)
- [8] STMicroelectronics IIS328DQ sensor [WWW] <http://www.st.com/en/mems-and-sensors/iis328dq.html> (16.05.2017)
- [9] Cortex-M4 Protsessor [WWW] <https://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php> (16.05.2017)
- [10] STM32Cube tarkvara [WWW] <http://www.st.com/en/embedded-software/stm32cube-embedded-software.html?querycriteria=productId=LN1897> (16.05.2017)
- [11] Cortex M0 Protsessor [WWW] <https://www.arm.com/products/processors/cortex-m/cortex-m0.php> (16.05.2017)
- [12] Adafruit BNO055 absoluut positsiooni näide [WWW] <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview> (16.05.2017)
- [13] Bosch BNO055 kiirendusanduri manuaal [WWW] [https://ae-bst.resource.bosch.com/media/\\_tech/media/datasheets/BST\\_BNO055\\_DS000\\_14.pdf](https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST_BNO055_DS000_14.pdf) (16.05.2017)

- [14] UART protokoll [WWW]  
[http://www.ftdichip.com/Support/Documents/TechnicalNotes/TN\\_111%20What%20is%20UART.pdf](http://www.ftdichip.com/Support/Documents/TechnicalNotes/TN_111%20What%20is%20UART.pdf) (16.05.2017)
- [15] MICA2DOT [WWW]  
<https://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2dot.pdf> (16.05.2017)
- [16] TinyOS Github [WWW] <https://github.com/tinyos/tinyos-main> (16.05.2017)
- [17] Wearable Sensors for Reliable Fall Detection [WWW]  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1617246> (16.05.2017)
- [18] Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information [WWW]  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5226903> (16.05.2017)
- [19] Philips GoSafe [WWW] <https://www.lifeline.philips.com/safety-solutions/gosafe.html> (16.05.2017)
- [20] Shimmer Sensing Consensy IMU Development Kits [WWW]  
<https://www.shimmersensing.com/products/shimmer3-development-kit#download-tab> (16.05.2017)
- [21] Shimmer sensing tooted [WWW] <https://www.shimmersensing.com/products/> (16.05.2017)
- [22] C programmeerimiskeel [WWW] <https://www.bell-labs.com/usr/dmr/www/chist.pdf> (16.05.2017)
- [23] STMCubeMX [WWW] <http://www.st.com/en/development-tools/stm32cubemx.html> (16.05.2017)
- [24] HAL teek [WWW]  
[http://www.st.com/content/ccc/resource/technical/document/user\\_manual/2f/71/ba/b8/75/54/47/cf/DM00105879.pdf/files/DM00105879.pdf/jcr:content/translations/en.DM00105879.pdf](http://www.st.com/content/ccc/resource/technical/document/user_manual/2f/71/ba/b8/75/54/47/cf/DM00105879.pdf/files/DM00105879.pdf/jcr:content/translations/en.DM00105879.pdf) (16.05.2017)
- [25] RealTerm [WWW] <https://realterm.sourceforge.io/> (16.05.2017)
- [26] STM-STUDIO-STM32 [WWW] <http://www.st.com/en/development-tools/stm-studio-stm32.html> (16.05.2017)

## **Lisa 1 – Kood**

Kood asub GitLabis aadressil [https://gitlab.pld.ttu.ee/thesis/2017\\_vedom\\_stm32](https://gitlab.pld.ttu.ee/thesis/2017_vedom_stm32).