|  |  |
| --- | --- |
| tallinna tehnikaülikool | |
| Infotehnoloogia teaduskond | |
| Arvutisüsteemide instituut | |
|  | |
| Jörgen Vedom 123584 IASB | |
| Mitme Kiirendusanduriga kukkumise tuvastamise süsteemi arendus | |
| Bakalaureusetöö | |
| Juhendaja: | Mairo Leier |
|  | Doktorikraad |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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: Jörgen Vedom

16.04.2017

Annotatsioon

Lõputöö on kirjutatud keeles ning sisaldab teksti leheküljel, peatükki, joonist, tabelit.

Abstract

The thesis is in and contains pages of text, chapters, figures, tables.

Lühendite ja mõistete sõnastik

|  |  |
| --- | --- |
| ATI | TTÜ Arvutitehnika instituut |
| DPI | *Dots perinch*, punkti tolli kohta |
| SAI |  |
| UART  USART  I2C  LED  USB  CAN  CEC  SDIO  NA  CRC |  |
| IDE  API | integreeritud programmeerimiskeskkond  Application Programming Interface |

Sisukord

[2.1 Süsteemi nõuded 11](#_Toc483086162)

[2.2 Arendusplaat STM32 F446RE: 13](#_Toc483086163)

[2.3 Kiirendusandur Bosch BNO055 14](#_Toc483086164)

[2.3.1 Kiirendusandur Bosch BNO055 Register 14](#_Toc483086165)

[2.4 Protokollid 15](#_Toc483086166)

[2.4.1 Protokoll UART 15](#_Toc483086167)

[2.4.2 Protokoll RS-485 16](#_Toc483086168)

[2.5 Ülevaade sarnastest lahendustest 17](#_Toc483086169)

[2.5.1 Wearable sensors for Reliable fall detection 17](#_Toc483086170)

[2.5.2 Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information 17](#_Toc483086171)

[2.5.3 Philips GoSafe 17](#_Toc483086172)

[2.5.4 Shimmer sensing 18](#_Toc483086173)

[2.6 Ühendused 18](#_Toc483086174)

[3.1 STM32CubeMX 19](#_Toc483086175)

[3.1.1 STM32Cube HAL teek 20](#_Toc483086176)

[3.2 Programmi töö 21](#_Toc483086177)

[3.3 Arendusplaadi initsialiseerimine: 22](#_Toc483086178)

[3.4 Sensori Initsialiseerimine: 23](#_Toc483086179)

[3.5 Andmete päring 24](#_Toc483086180)

[3.5.1 Lugemine 24](#_Toc483086181)

[3.5.2 Kirjutamine 26](#_Toc483086182)

[3.6 Häirete kontroll: 28](#_Toc483086183)

[3.7 Kiirendusanduri kalibreerimine: 28](#_Toc483086184)

Jooniste loetelu

[Joonis 1 - Arendusplaat sensoritega 11](#_Toc483086185)

[Joonis 2 – Arendusplaat STM32F446RE 13](#_Toc483086186)

[Joonis 3 - Kiirendusandur BNO055 14](#_Toc483086187)

[Joonis 4 - TXD ja RXD 16](#_Toc483086188)

[Joonis 5 - RS-485 ühendus 16](#_Toc483086189)

[Joonis 6 - STM32CubeMX projekti seaded 20](#_Toc483086190)

[Joonis 7 - Programmi töö 21](#_Toc483086191)

[Joonis 8 - Arendusplaadi initsialiseerimine 22](#_Toc483086192)

[Joonis 9 - Sensori initsialiseerimine 23](#_Toc483086193)

[Joonis 10 - Häirete kontroll 28](#_Toc483086194)

[Joonis 11 - B1 ja LD2 29](#_Toc483086195)

[Joonis 12 - Realterm Port seaded 30](#_Toc483086196)

[Joonis 13 - Realterm Displei seaded 31](#_Toc483086197)

[Joonis 14 - Paigal olek 32](#_Toc483086198)

Tabelite loetelu

[Tabel 1 - UART andmeformaat 15](#_Toc483086199)

[Tabel 2 - Arendusplaadi ja sensorite ühendused 18](#_Toc483086200)

[Tabel 3 - UART seaded 19](#_Toc483086201)

[Tabel 4 - MSB ja LSB andme töötlus 22](#_Toc483086202)

[Tabel 5 - Lugemise päring 24](#_Toc483086203)

[Tabel 6 - Registrist õnnestunud lugemise vastus 25](#_Toc483086204)

[Tabel 7 - Registrist lugemise veateade 25](#_Toc483086205)

[Tabel 8 - Kirjutamine 26](#_Toc483086206)

# Sissejuhatus

Käesoleva lõputöö teemaks on kukkumise tuvastamise süsteemi arendus. Töö põhieesmärgiks on luua platvorm, mille abil saab välja töötada kukkumise tuvastamiseks algoritmi. Prototüüpi kasutatakse uuringute läbiviimisel liikuvatel alustel, milleks võivad olla meresõidukid ja merel olevad kalafarmid. Sellest tulenevalt kasutatakse seadet kinnitatuna inimese küljes, kus üle keha on paigutatud mitu erinevat sensorit. Välitingimustes tuleb arvestada keskkonna mõjudega. Merel paiknevat alust mõjutavad peale lainete ka teised olud näiteks ilmastik. Seade peab olema vastupidav, et tagada andmeside nendes tingimustes.

Nagu eelnevalt mainitud, on sensorid paigaldatud inimese külge. Sensorite ühendamiseks kasutatakse juhtmeid. Sensorid asuvad teineteisest mõnekümne sentimeetri kaugusel ning on ühendatud arendusplaadiga. Juhtmed tagavad kiire ja püsiva ühenduse, kuid seejuures tuleb arvestada, et juhtmed võivad liikumisel venida ning ühendused võivad puruneda. Selle ära hoidmiseks tuleb juhtmetele jätta ruumi. Mida pikem on ühendus, seda rohkem esineb häireid, seega tuleb protokolli valikus arvestada veakindlusega.

Seoses sellega, et süsteem vastutab inimelu eest, on tegemist reaalajasüsteemiga. Kukkumine toimub kiiresti ning seetõttu tuleb andmeid pärida tihti. Süsteem peab selle juures olema võimeline andmeid kiiresti töötlema. Mida sagedamini toimub kukkumise tuvastus, seda kiiremini saab juhtunust teavitada. Kukkumine võib toimuda põrandale kui ka vette, kukkumisel võib seade saada kahjustada, seega tuleb mõelda, kuidas seadet kaitsta. Arendusplaadi kui ka sensori valikul tuleb seda kõike silmas pidada.

Kuna tegemist on platvormi arendamisega, millele abil töötatakse välja algoritm, peab töö olema paindlik. Arendusplaati peab saama kiirelt ümber seadistada. Seega tuleb sensorite ja arendusplaadi vahelisel suhtlusel pidada silmas seda, et programm pakuks seadistamise võimalusi algoritmi töötlejale.

Käesolev töö sisaldab kolme põhiülesannet, millest esimene on prototüübi riistvara arendus. Teiseks mikrokontrolleri seadistamine töötamaks mitme kiirendusanduriga. Kolmandaks kiirendus andurite seadistamine vastavalt algoritmi poolt sisendandmetele esitatavatele nõuetele.

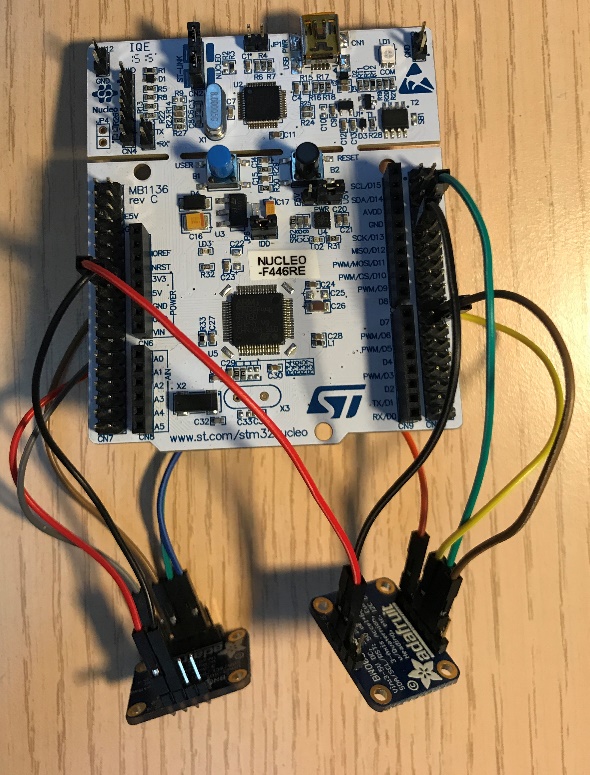
# Riistvara

Süsteemi komponendid:

Arendusplaat STM32 -F446RE [1]

Kiirendusandur Bosch BNO055 [2]

Süsteemi ülesehitus on näidatud Joonisel 1, joonisel on kujutatud arendusplaati koos sensoritega.



Joonis - Arendusplaat sensoritega

## Süsteemi nõuded

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

Arendusplaate on mitmeid, millest kuulsamad tootjad on Arduino [3], Raspberry Pi [3] ning BeagleBoard [4]. Arduino tootlevalikus leidub palju arendusplaate, millest üks võimekamaid on Arduino STAR-OTTO [5], mis on välja töötatud koos STMicroelectronics’iga [6]. Nimetatud seade ei leia antud töö raames kasutust, sest sisend-väljund porte pole piisavalt. Raspberry PI võimekaim plaat – Raspberry PI 3 Model B [7] – ei sobi samuti eeltoodud põhjuse tõttu. BeagleBoard’i – BeagleBoard-X15 [8] – jääb valikust välja, sest tegemist on arvutiga, mille peal töötab Linux operatsiooni süsteem [9]. Sama põhjus on takistuseks ka Raspberry PI’l ja Arduino’l. Probleem seisneb selles, et operatsioonisüsteem vajab ressurssi töötamiseks. Isegi kui arendusplaatidel oleks vajalikud liidesed olemas, võib tekkida olukord, kus operatsioonisüsteem piirab jõu kasutuse enda jaoks ning kukkumise tuvastamissüsteem kannatab selle tulemusel. STMicroelectronics pakub arendusplaati STM32F446 [1], mis on antud töö jaoks sobilik.

Kiirendusanduri valikul tuleb veenduda, et arendusplaat toetaks andmevahetuseks samu liideseid. Andur peab mõõtmete poolest olema väike ja kerge, sest on võimalus, et see paigutatakse riide külge. Lisaks peale kiirendusanduri võiks sensoril olla mõningaid muid andureid, näiteks temperatuuriandur. Temperatuuri kiirel muutumisel, võib see tähendada, et inimene kukkus vette. Tähtis on, et andur mõõdaks X, Y, Z telgi. STMelectronics pakub IIS328DQ [10] madala tarnega kiirendusandurit, mis ei ole sobilik, sest suhtluseks kasutab I2C või SPI porti. Võttes arvesse juhtmete kaugusi ei taga need protokollid vea kindlat andmeside. Bosch pakub mitmeid kiirendusandureid sealhulgas kiirendusandurit Bosch BNO055 [2], mis on sobilik antud töö jaoks.

Kiirendusandur Bosch BNO055 [2] andmevahetuseks on toetatud HID-I2C, I2C ning UART protokollid. Arendusplaat STM32 F446RE [1] pakub suhtluseks I2C, UART, SPI, SAI, CAN, GPIO, SDIO liideseid. Arvestades, et juhtme pikkused võivad olla mõnikümmend sentimeetrit või rohkemgi, tuleb kasutada vea kindlat protokolli näiteks RS-485. RS-485 protokoll põhineb UART protokollil, mida toetab nii sensor kui ka arendusplaat, seega oleks võimalik arendusplaadi ja sensori suhtluseks kasutada seda protokolli.

## Arendusplaat STM32 F446RE

Tähtsamad tehnilised andmed [1]:

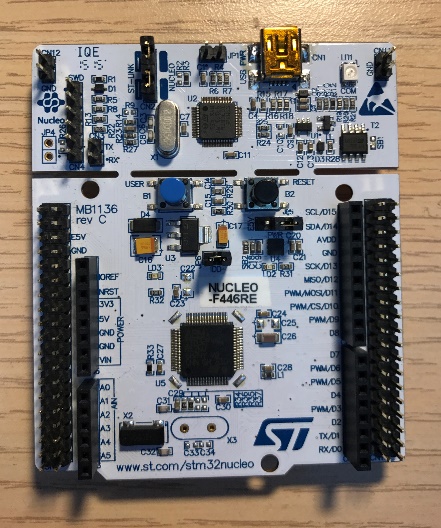
Protsessor: 180 MHz Cortex-M4 [11]

Mälu: 512 kB Välkmälu, 128 KB SRAM

Toide: 1.7 – 3.6 V

Liidesed: I2C, UART, SPI, SAI, CAN, GPIO, SDIO

Arendusplaat on kujutatud joonisel 2.



Joonis – Arendusplaat STM32F446RE

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

STMicroelectronics pakub enda mikrokontrolleritele tarkvara STM32Cube [12], mis lihtsustab arendaja tööd. Tarkvaraga on võimalik genereerida initsialiseerimiskood. Arendaja määrab pordid, protokollid ja konfiguratsiooni seaded ning STM32Cube programm genereerib esialgse koodi vastava seadistusega.

## Kiirendusandur Bosch BNO055

Tähtsamad tehnilised andmed [2]:

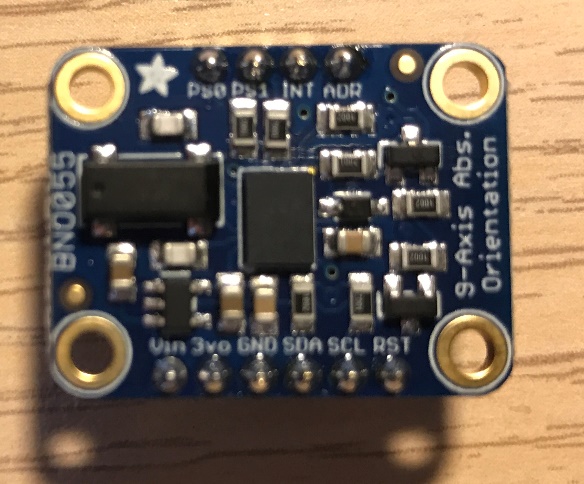
Andurid: Kiirendusandur, güroskoop, geomagnetiline sensor

Toide: 2.4 – 3.6 V

Liidesed: HID-I2C, I2C, UART

Kiirendusvahemik: ± 2g - ± 16 g

Kiirendusandur on kujutatud joonisel 3.



Joonis - Kiirendusandur BNO055

Sensoriks sai valitud Bosch BNO055. Andmeedastuseks kasutab sensor HID-I2C, I2C ja UART liidest. Kuna tegemist on reaalajatarkvara süsteemiga, siis on oluline, et sensor oleks võimeline saatma pidevalt andmeid, milleks sensor sobib. Sensori protsessoriks on ARM Cortex-M0 [13]. Sensor koosneb kolmest andurist (kiirendusandur, güroskoop ja geomagnetiline andur). Sensor on mõõtmetelt väike: 5.2 mm x 3.8 mm. Samuti leidub sensori kohta palju materjale internetis. Näiteks on tehtud sensoriga orientatsiooni aplikatsioon kasutades selleks ardunio’t [14].

### Kiirendusandur Bosch BNO055 Register

Register on jagatud kaheks loogiliseks leheks. Leht 1 omab endas sensoripõhist konfiguratsiooni seadeid. Leht 0 omab kõiki muid konfiguratsiooni parameetreid ning väljundandmeid [15].

## Protokollid

Arendusplaadil ja kiirendusanduril on ühiseks liidesteks I2C ja UART. Seoses sellega, et juhtmete pikkuseks võib olla mõnikümmend sentimeetrit, 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.

### Protokoll UART

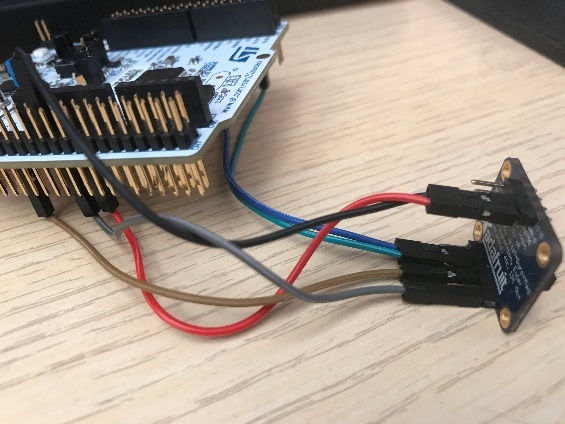
Laialdaselt kasutatud protokoll, mida kasutatakse andmete saatmiseks. UART protokoll on asünkroonne, mis tähendab, et puudub taktsignaal. Andmete saatmisel on sõnumil algus- ja lõpubitt. Oluline on, et mõlemad seadmed töötaksid samal boodikiirusel. UART protokolli andmevahetuse maksimaalseks kauguseks on ~2m. Boodikiirus määrab kui kiirelt toimub andmevahetus. Arendusplaadi ja andurite vahel on boodikiiruseks 115200 boodi. Mis on 15200 / 10 = 1520 bitti sekundis [16]. UART’i andme formaat on näidatud tabelis 1.

Ühe andme baidi saamiseks, tuleb saata kokku 10 bitti. 1 bitt on algusbitt, 8 bitti andmeid ning 1 stoppbitt.

Tabel - UART andmeformaat

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.Bitt | 2.Bitt | 3.Bitt | 4.Bitt | 5.Bitt | 6.Bitt | 7.Bitt | 8.Bitt | 9.Bitt | 10.Bitt |
| A  L  G  U  S | 1  A  N  D  M  E  D | 2  A  N  D  M  E  D | 3  A  N  D  M  E  D | 4  A  N  D  M  E  D | 5  A  N  D  M  E  D | 6  A  N  D  M  E  D | 7  A  N  D  M  E  D | 8  A  N  D  M  E  D | S  T  O  P |

Andmete saatmine toimub läbi TXD signaali, mis on ühendatud vastuvõtja RXD signaali külge. Andmete vastuvõtmine käib läbi RXD signaali, mis on ühendatud saatja TXD signaali külge. Ühendusi kujutab joonis 4, kus roheline juhe tähistab TXD signaali, ning sinine juhe RXD signaali.

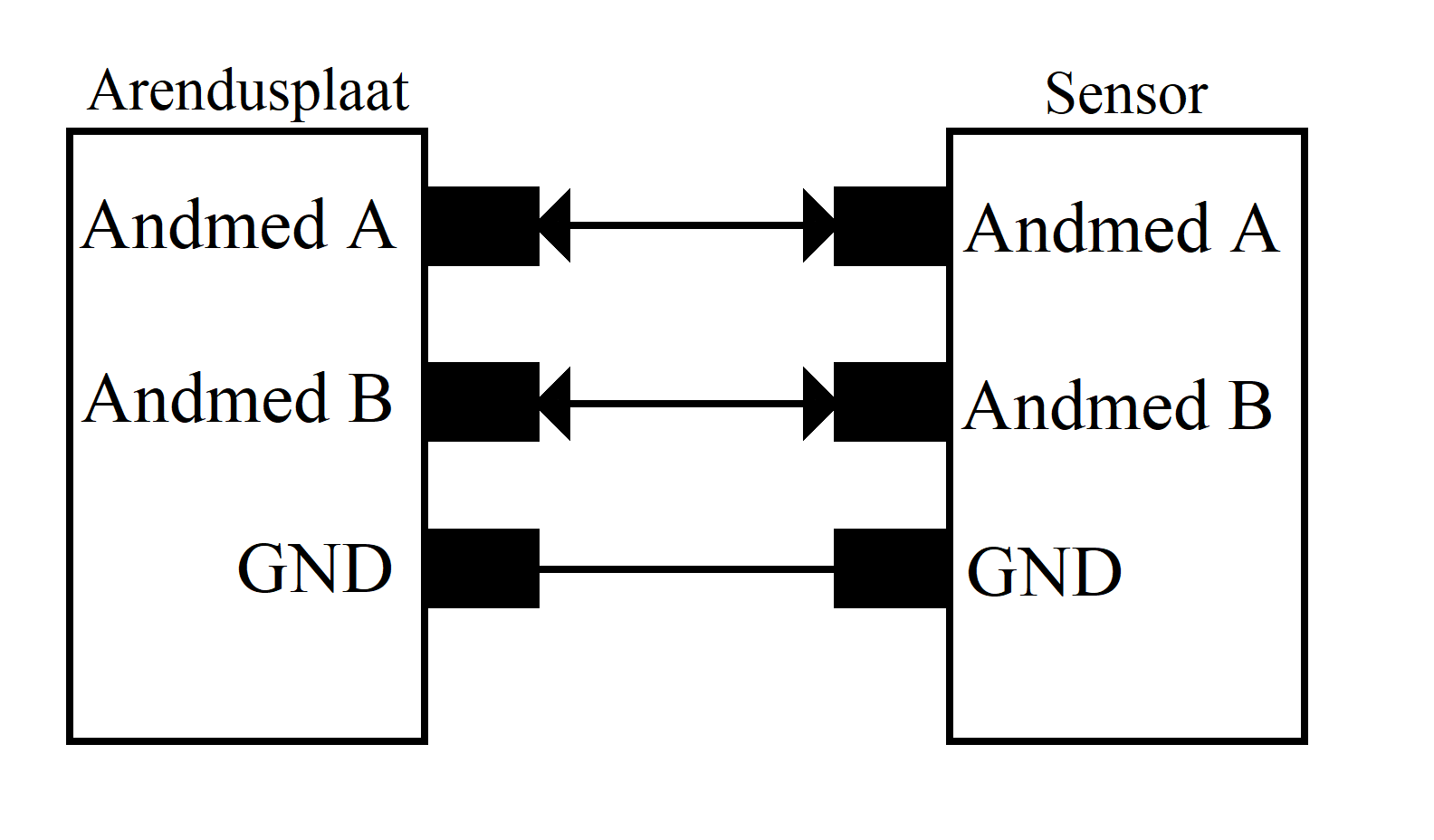


Joonis - TXD ja RXD

Bosch BNO055 toetab UART protokolli järgmiste seadetega: 115200 boodikiirus, 8N1 (8 andme bitti, 0 paarsus bitti, 1 stopp bitt). Maksimaalne pikkus lugemiseks ja kirjutamiseks 128 baiti.

### Protokoll RS-485

RS-485 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 müra taluvus ning lubab seejuures pikemaid ühendusi. RS-485 protokolli andmevahetuse kaugus on ~1200 m [16]. RS-485 ühendus on näidatud joonisel 5.



Joonis - RS-485 ühendus

## Ülevaade sarnastest lahendustest

Kukkumise tuvastusseadmeid on mitmeid, kuid sellist toodet, mis vastab nõuetele ei leidu.

### 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 baas seadmes. 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 [17], mis kasutab operatsioonisüsteemiks TinyOS 1.0 [18]. [19]

### 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. Laeva peal lahendus ei toimiks – pole arvestatud kõikumistega. Võimalik, et eeltoodud lahendust edasi arendades leiaks seade kasutust ka laeva peal. [20]

### Philips GoSafe

Tegemist on seadmega, mis on välja töötatud nii sise- kui ka välistingimustesse. Asukoha tuvastamiseks on kasutusel Wifi, GPS, 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. [21]

### Shimmer sensing

Firma, mis keskendub kantavatele sensoritele. Sensorid on andmete kogumiseks ning 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 ning aeganõudev. Juba ainuüksi andmekaabel maksab 39 €. Sobiva arenduskomplekti (Consensys IMU Development Kit [22]) saaks neilt 499 €. [23]

## Ühendused

STM32F446RE arendusplaadil on 76 PIN’i. Arendusplaadi ning 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 *debugimiseks*.

USART1 ning USART3 ühendusi kajastab tabel 2.

Tabel - Arendusplaadi ja sensorite ühendused

|  |  |  |  |
| --- | --- | --- | --- |
| Arendusplaat | Sensor 1 port  USART1 | Arendusplaat | Sensor 2 port  USART3 |
| GND | PS0 | GND | PS0 |
| +3V3 | PS1 | +3V3 | PS1 |
| +3V3 | Vin | +3V3 | Vin |
| GND | GND | GND | GND |
| PA10 | SDA | PC5 | SDA |
| PA9 | SCL | PB10 | SCL |

# Tarkvara

Programmikood on kirjutatud C keeles. C keel on laialdaselt kasutatud süsteemne programmeerimis keel. Keel loodi aastatel 1977 – 1979 paralleelselt operatsiooni süsteemi UNIX’iga. Esialgu kasutati keelt UNIX süsteemi jaoks, kuid tänapäeval on see kujunenud üheks peamiseks arvutitööstuses kasutatavaks programmeerimis keeleks. C keele eelisteks on tema kiirus ning vähene mälukasutus [24]. Programmeerimiskeskkonnaks vailisin Keil uVision5, mis on toetatud ka STM32CubeMxis.

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

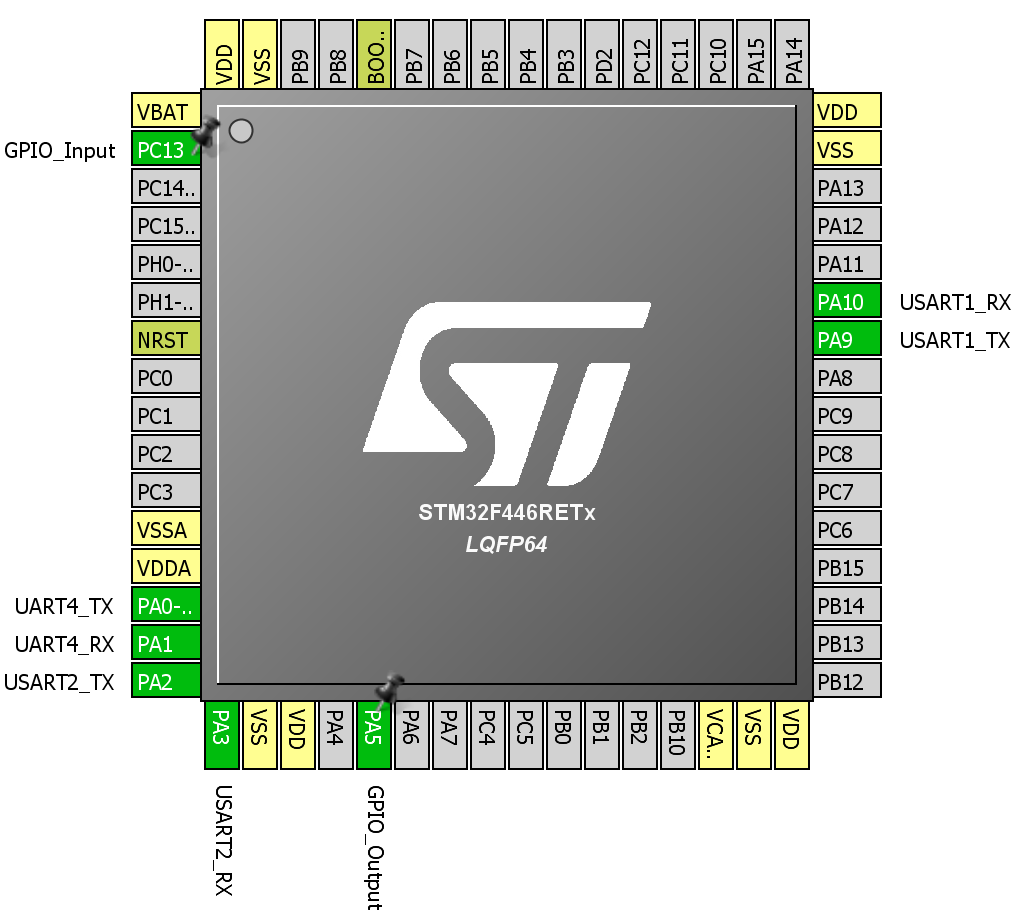
## STM32CubeMX

STM32CubeMX on programm, mis lihtsustab arendaja tööd. Tegemist on tarkvaraga, mis võimaldab genereerida initsialiseerimis koodi C keeles. Koodi genereerimiseks tuleb läbi graafilise liidese valida projekti jaoks seaded. Lisaks initsialiseerimis koodile antakse kaasa STM32Cube HAL teek. [25]

STM32CubeMX programmis olulised projekti seaded. IDE valik – MDK-ARM V5. Aktiveerida tuleb USART1, USART2, USART3, USART6 ning neile määrata globaalne katkestus, täpsemad seadeid näitab tabel 3. Koodi genereerimise alam menüüst tuleb vaadata, et kasutajakood jääks alles igakord kui programm genereerib uue koodi. Joonis 6 näitab STM32 Projekti seadistusi.

Tabel - 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 - STM32CubeMX projekti seaded

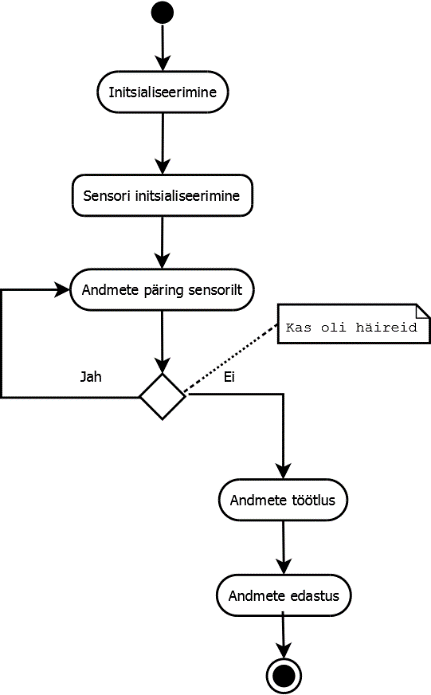
### STM32Cube HAL teek

STM32Cube HAL teek on STMicroeletronics’i initsiatiivil arendatud draiver. Draiver pakub kasutamiseks API’se, mis lihtsustavad kasutajat rakenduse loomisel. Iga draiver koosneb funktsioonide hulgast, mis katavad tavapärased vajadused. Näiteks saab teegiga initsialiseerida ning seadistada UART liidest, hallata andmevahetust, töödelda katkestusi või DMA’si ning hallata vigu. [26]

## Programmi töö

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

Andmete kontroll toimub järgnevalt. Kokku X, Y, Z koordinaatide jaoks tuleb küsida 6 andme baiti. On võimalik, et sensor saadab andmete asemel vea teate. Viga võib toimuda, mis tahes hetkel 6 baidi pärimisel. Kui üks bait kuuest on vigane, ei saa andmeid töödelda. Täpsema info kuidas toimub andmete pärimise kohta saab peatükis 3.5 Andmete päring. Programmi tööd kirjeldab joonis 7.



Joonis - Programmi töö

Juhul kui andmete päring õnnestus, toimub andmete töötlus. Loetavate andmete, saamiseks tuleb kokku panna 2 baiti. Täisarv koosneb kahest baidist, kuueteistkümnest bittist, kus esimene bait tähistab vähima kaaluga baiti ning teine bait suurimat. Tabel 4 on toodud näide MNB ja LSB andme töötlusest.

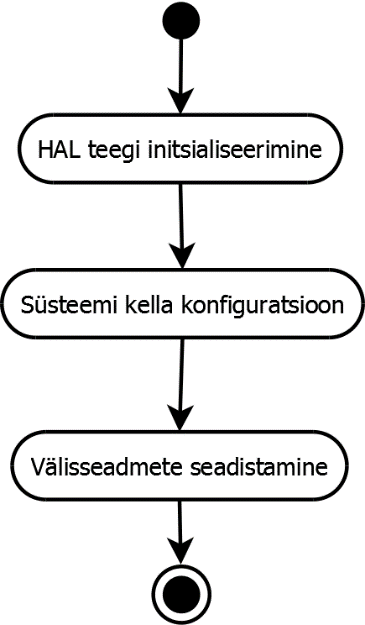
Tabel - MSB ja LSB andme töötlus

|  |  |  |
| --- | --- | --- |
| MSB | LSB | Töödeldud kujul |
| 8 baiti | 8 baiti | 16 baiti |
| 1111 1101 | 1110 0010 | 1111 1101 1110 0010 |
| 0xFD | 0xE2 | 0xFDE2 |
| 253 | 226 | -770 |

## Arendusplaadi initsialiseerimine:

Arendusplaadi initsialiseerimisel käivitatakse esmalt HAL teek. Seejärel seadistatakse süsteemitakt. Initsialiseeritakse sisemine väljund pinge, CPU-, AHB- ja ABP siini taktid. Süsteemitakt käsitleb arvutisüsteemis kõiki sünkroniseerimisist.

Seejärel toimub väliseadmete seadistus. Käivitatakse kõik konfigureeritud väljundid: GPIO, UART, USART. Arendusplaadi initsialiseerimine on näidatud joonisel 8.

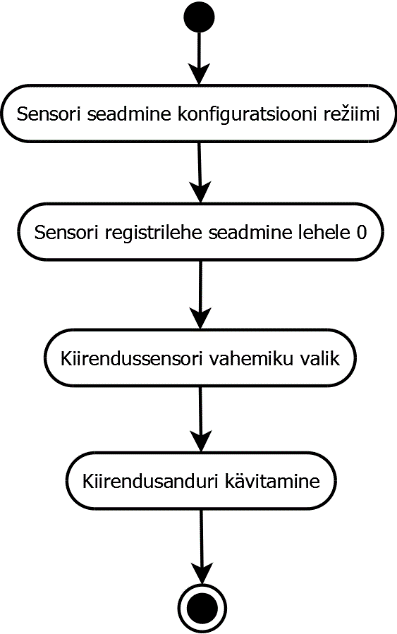


Joonis - Arendusplaadi initsialiseerimine

## Sensori Initsialiseerimine:

Sensori käivitamisel saadetakse sensorile konfiguratsiooni seaded, mis seadetega sensor tööle hakkab. Juhul kui tekkib vajadus muuta seadistust jooksvalt, tuleb selleks sensor viia uuesti konfiguratsiooni režiimi.

Sensori initsialiseerimiseks lülitakse sensor konfiguratsiooni režiimi. Sensorile saadetakse kiirendusanduri vahemik 8G, temperatuuri ühikuks kraadid ning seejärel käivitatakse sensor kiirendusanduri režiimis. Sensori initsialiseerimis kirjeldab joonis 9.



Joonis - Sensori initsialiseerimine

## Andmete päring

Andmete saamiseks või saatmiseks tuleb sensorile saata andmete jaoks päring. Nii lugemisel kui ka kirjutamisel on algus bait sama. Teine bait määrab, kas tegemist on lugemise või kirjutamisega. Juhul kui on tegemist kirjutamisega on teise baidi sisuks 0x00 teisel juhul 0x01.

### Lugemine

Teine bait määrab, kas tegemist on lugemise või kirjutamisega. Lugemisel on teise baidi sisuks 0x01. Lugemiseks võib küsida maksimaalselt 128 baiti andmeid.

Näiteks, kui on vaja lugeda kiirendusanduri X, Y, Z koordinaate. Saab seda teha ühe päringuga või mitme päringuga.

Üheks päringuks tuleb näidata neljanda baidiga, et soovitakse lugeda 6 baiti. Ning kolmas bait, mis näitab registri aadressi, peab viitama registri X koordinaadi LSB registrile. Alternatiivina võib teha 6 päringut, küsides igat baiti eraldi.

Mõistlikum on seda teha ühe päringuga, sellisel juhul on vigaste andmete tõenäosus väiksem. Lugemise päring on kirjeldatud tabel 5.

Tabel - Lugemise päring

|  |  |  |  |
| --- | --- | --- | --- |
| Bait 1 | Bait 2 | Bait 3 | Bait 4 |
| Algus Bait | Lugemine | Registri aadress | Pikkus |
| 0xAA | 0x01 | <..> | <..> |

#### Õnnestunud lugemine

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 - Registrist õnnestunud lugemise vastus

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bait 1 | Bait 2 | Bait 3 | … | Bait (n + 2) |
| Algus Bait | Pikkus | Andmed 1 | … | Andmed n |
| 0xBB | <..> | <..> | … | <..> |

#### Nurjunud lugemine

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

Tabel - Registrist lugemise veateade

|  |  |
| --- | --- |
| Bait 1 | Bait 2 |
| Algus bait | Staatus |
| 0xEE | 0x02: Lugemine nurjus  0x04: Vale Registri aadress  0x05: Registri kirjutamine väljalülitud  0x06: Vale algus bait  0x07: Siini ülekoormuse viga  0x08: Maksimaalse pikkuse viga  0x09: Minimaalse pikkuse viga  0x0A:Vastuvõetud märgi taimaut |

### Kirjutamine

Nagu eelnevalt mainitud, määrab teine bait, et tegu on kirjutamisega. Kirjutamisel on teise baidi sisuks 0x00. Kirjutada võib maksimaalselt 128 baiti andmeid. Kirjutamisel tuleb veenduda, et süsteemi seadete muutmiseks, tuleb eelnevalt viia sensor konfiguratsiooni režiimi.

Kui on vaja kirjutada kiirendusanduri kalibreerimiseks X, Y, Z koordinaate 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 registri aadressile.

Mõistlikum on seda teha ühe päringuga, sellisel juhul on vigaste andmete tõenäosus väiksem. Kirjutamise päring on näidatud tabelis 8.

Tabel - Kirjutamine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bait 1 | Bait 2 | Bait 3 | Bait 4 | Bait 5 | … | Bait (n + 4) |
| Algus Bait | Kirjutamine | Registri aadress | Pikkus | Andmed 1 | … | Andmed n |
| 0xAA | 0x00 | <..> | <..> | <..> | … | <..> |

#### 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 2. baidi sisu.

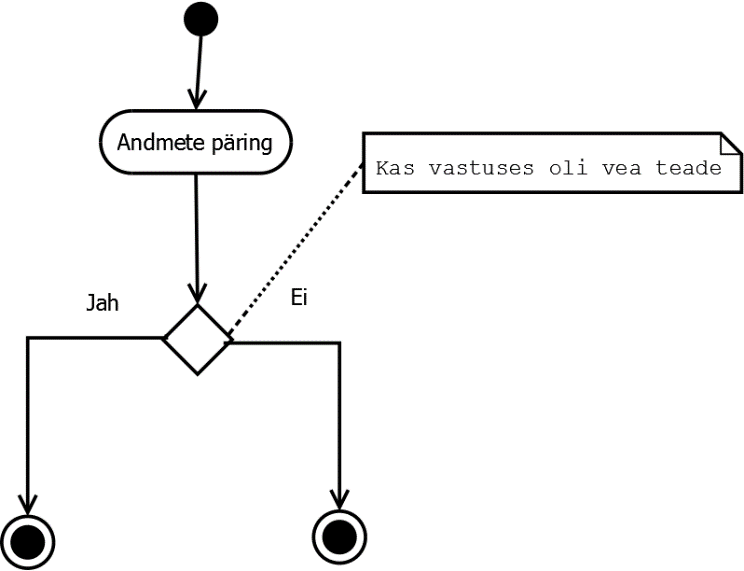
Kirjutamise õnnestumist kajastab staatus 0x01, teistel puhkudel on tegemist vea teadetega. Kirjutamise vastus on kujutatud tabelis 7.

Tabel 7 - Kirjutamise vastus

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

## Häirete kontroll:

Peale andmete päringut kontrollitakse, kas andmed olid õiged. Juhul kui algus bait on 0xBB on tegemist vigadeta andmetega. Juhul kui algus bait on 0xEE on tegemist nurjunud lugemisega. Vigase lugemise korral andmeid ei töödelda.

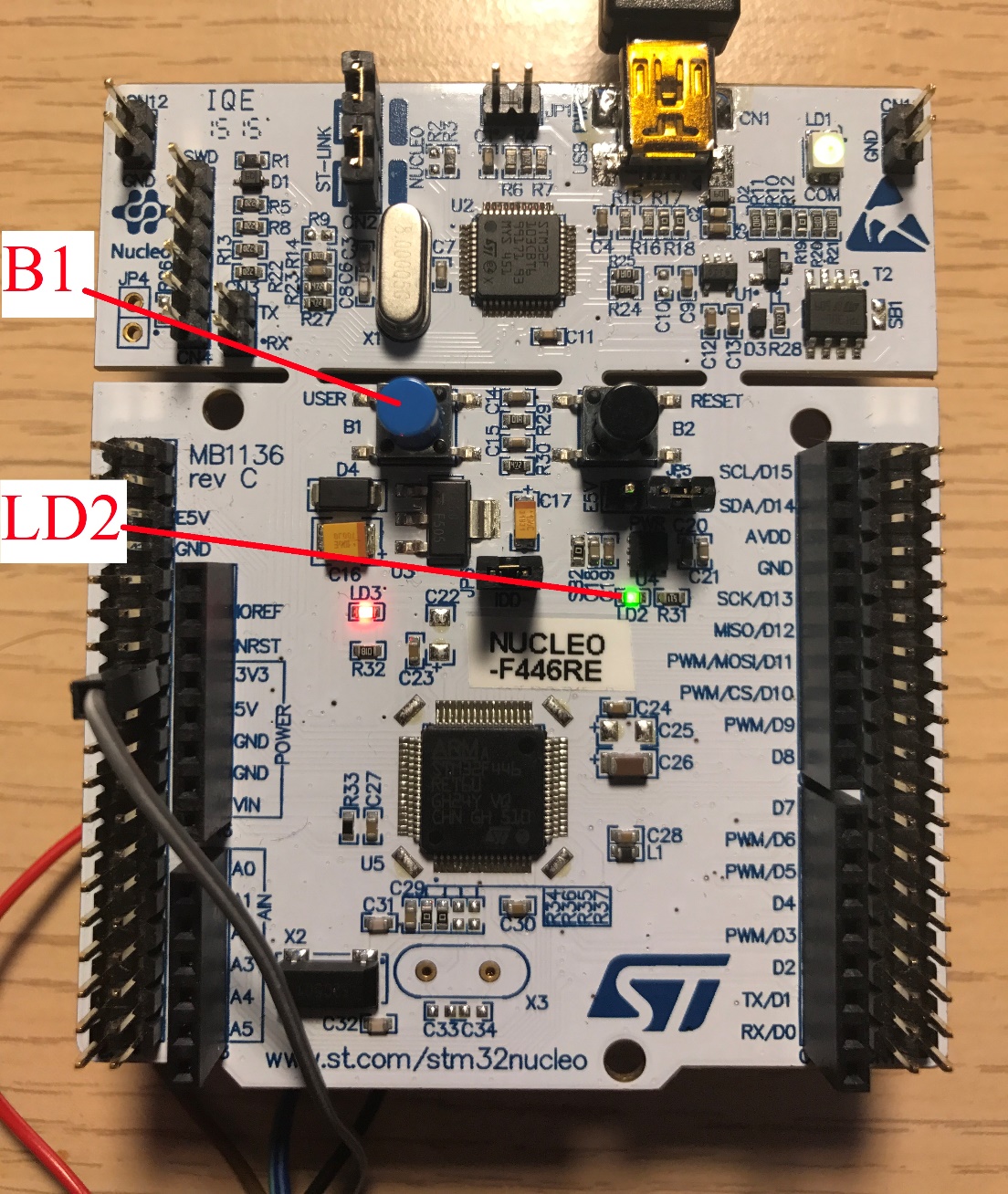


Joonis - Häirete kontroll

## Kiirendusanduri kalibreerimine:

Kiirendusanduri kalibreerimine on jaotatud 6’te etappi:

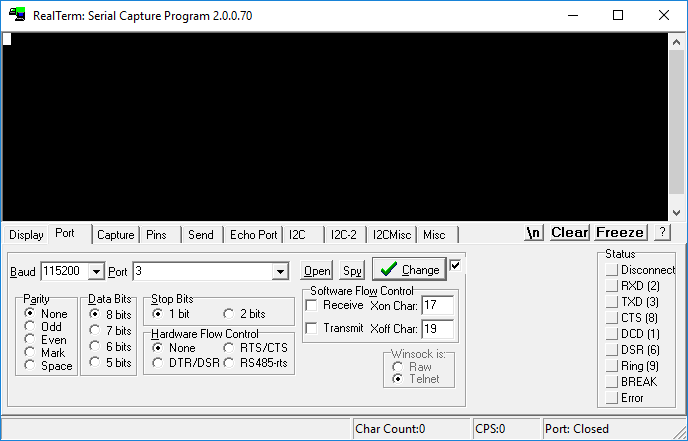
1. Sinist nuppu, mis on näidatud joonisel 11, tuleb hoida all viis sekundit.
2. Seejärel on kahesekundiline paus.
   1. Juhul kui selle kahesekundi jooksul vajutada sinist nuppu, seadistatakse teist sensori.
   2. Kui nuppu ei vajutata, seadistatakse esimest sensori.
3. Esmalt seadistatakse sensori X telge. Tuluke LD2, mis on näidatud joonisel 11, hakkab vilkuma sagedusega 1000 ms. Sensor tuleb paigutada X telje suhtes tasasele pinnale ning seejärel tuleb vajutada sinist nuppu B1
4. Järgmisena seadistatakse sensori Y telge. Tuluke LD2 hakkab vilkuma sagedusega 400ms. Sensor tuleb paigutada Y telje suhtes tasasele pinnale ning seejärel tuleb vajutada sinist nuppu B1.
5. Viimasena seadistatakse sensori Z telge. Tuluke LD2 hakkab vilkuma sagedusega 100 ms. Sensor tuleb paigutada Z telje suhtes tasasele pinnale ning seejärel tuleb vajutada sinist nuppu B1.
6. Järgnevalt saadetakse sensorile kalibreerimisandmed mille tulemusel on sensor seadistatud.



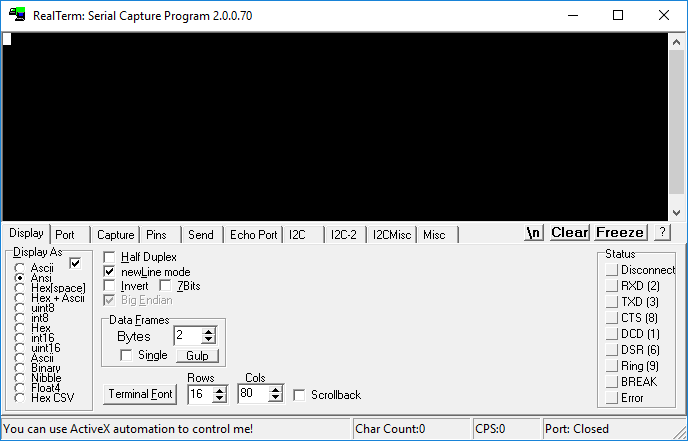
Joonis - B1 ja LD2

# Seadistamine ja testimine

Arvuti ja arendusplaat ühendatakse arvutiga läbi mikro-USB juhtme. Läbi juhtme toimub koodi peale laadimine ning *debuggimine*. *Debuggimiseks* on Keil programmeerimiskeskkonnas mitmeid.

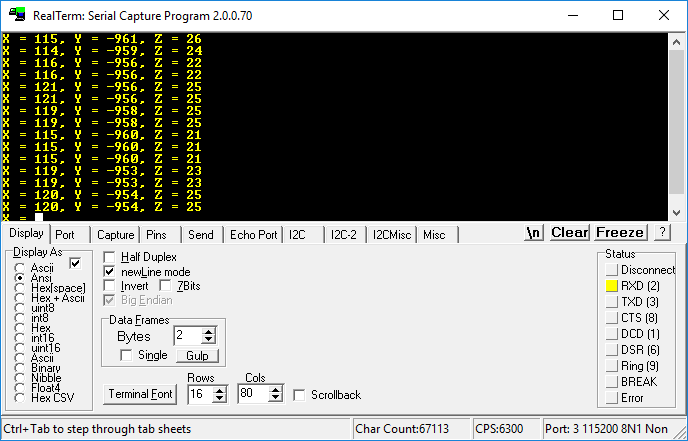


Joonis - Realterm Port seaded



Joonis - Realterm Displei seaded

Normaal tingimustes peaksid andmed jääma -981 ja 981 vahemiku. Nagu näidatud joonisel 13.



Joonis - Paigal olek

# Kokkuvõte

Kasutatud kirjandus

# Bibliography

|  |  |
| --- | --- |
| [1] | „STM32 - F446RE arendusplaat,“ [Võrgumaterjal]. Available: http://www.st.com/en/microcontrollers/stm32f446.html?querycriteria=productId=LN1875. |
| [2] | [Võrgumaterjal]. Available: https://www.bosch-sensortec.com/bst/products/all\_products/bno055. |
| [3] | [Võrgumaterjal]. Available: https://www.raspberrypi.org/. |
| [4] | [Võrgumaterjal]. Available: http://beagleboard.org/bone. |
| [5] | [Võrgumaterjal]. Available: http://www.arduino.org/products/boards/arduino-star-otto. |
| [6] | [Võrgumaterjal]. Available: http://www.st.com/content/st\_com/en.html. |
| [7] | [Võrgumaterjal]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. |
| [8] | [Võrgumaterjal]. Available: https://beagleboard.org/x15. |
| [9] | [Võrgumaterjal]. Available: https://www.linux.org/. |
| [10] | [Võrgumaterjal]. Available: IIS328DQ. |
| [11] | [Võrgumaterjal]. Available: https://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php. |
| [12] | [Võrgumaterjal]. Available: http://www.st.com/en/embedded-software/stm32cube-embedded-software.html?querycriteria=productId=LN1897. |
| [13] | [Võrgumaterjal]. Available: https://www.arm.com/products/processors/cortex-m/cortex-m0.php. |
| [14] | [Võrgumaterjal]. Available: https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview. |
| [15] | [Võrgumaterjal]. Available: https://ae-bst.resource.bosch.com/media/\_tech/media/datasheets/BST\_BNO055\_DS000\_14.pdf. |
| [16] | [Võrgumaterjal]. Available: http://www.ftdichip.com/Support/Documents/TechnicalNotes/TN\_111%20What%20is%20UART.pdf. |
| [17] | [Võrgumaterjal]. Available: https://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2dot.pdf. |
| [18] | [Võrgumaterjal]. Available: https://github.com/tinyos/tinyos-main. |
| [19] | [Võrgumaterjal]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1617246. |
| [20] | [Võrgumaterjal]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5226903. |
| [21] | [Võrgumaterjal]. Available: https://www.lifeline.philips.com/safety-solutions/gosafe.html. |
| [22] | [Võrgumaterjal]. Available: https://www.shimmersensing.com/products/shimmer3-development-kit#download-tab. |
| [23] | [Võrgumaterjal]. Available: https://www.shimmersensing.com/products/. |
| [24] | [Võrgumaterjal]. Available: https://www.bell-labs.com/usr/dmr/www/chist.pdf. |
| [25] | [Võrgumaterjal]. Available: http://www.st.com/en/development-tools/stm32cubemx.html. |
| [26] | [Võrgumaterjal]. |
| [27] | L. S. Sterling, The Art of Agent-Oriented Modeling, London: The MIT Press, 2009. |
| [28] | „asdf,“ [Võrgumaterjal]. Available: http://www.st.com/en/microcontrollers/stm32f446.html?querycriteria=productId=LN1875. |
| [29] | [Võrgumaterjal]. |
| [30] | [Võrgumaterjal]. Available: http://www.ti.com/tool/tmdslcdk6748#descriptionArea. |
| [31] | [Võrgumaterjal]. Available: https://www.arduino.cc/. |
| [32] | [Võrgumaterjal]. Available: https://www.shimmersensing.com/products/shimmer3-development-kit#download-tab. |
| [33] | [Võrgumaterjal]. Available: https://realterm.sourceforge.io/. |

Lisa 1 – Registri leht