|  |  |
| --- | --- |
| 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 Ülevaade sarnastest lahendustest 11](#_Toc482564593)

[2.2 Süsteemi nõuded 12](#_Toc482564594)

[2.3 Ühendused 13](#_Toc482564595)

[2.4 Tehnilised andmed 14](#_Toc482564596)

[2.5 Protokollid 19](#_Toc482564597)

[3.1 Arendusplaadi ühendamine arvutiga 24](#_Toc482564598)

[3.2 Arendusplaadi seadistamine 24](#_Toc482564599)

[STM32CubeMX seadistamine: 24](#_Toc482564600)

[4.1 Programmi töö 27](#_Toc482564601)

Jooniste loetelu

[Joonis 1 - Arendusplaadi PIN skeem 13](#_Toc482564611)

[Joonis 2 - Blokk diagramm 16](#_Toc482564612)

[Joonis 3 - UART 19](#_Toc482564613)

[Joonis 4 - RS485 22](#_Toc482564614)

[Joonis 5 - STM32CubeMX Projekti vaade 25](#_Toc482564615)

[Joonis 6 - Programmi töö 28](#_Toc482564616)

[Joonis 7 - Arendusplaadi initsialiseerimine 29](#_Toc482564617)

[Joonis 8 - Sensori initsialiseerimine 30](#_Toc482564618)

[Joonis 9 - Andmete päring 31](#_Toc482564619)

[Joonis 10 - Häirete kontroll 32](#_Toc482564620)

[Joonis 11 - Realterm Port seaded 34](#_Toc482564621)

[Joonis 12 - Realterm Displei seaded 34](#_Toc482564622)

[Joonis 13 - Paigal olek 35](#_Toc482564623)

[Joonis 14 - Liikumine 35](#_Toc482564624)

[Joonis 15- Tugevam põrutus 36](#_Toc482564625)

[Joonis 16 - Arendusplaadi pilt 39](#_Toc482564626)

[Joonis 17- Led tuluke 40](#_Toc482564627)

Tabelite loetelu

[Tabel 1 – Sensor 1 13](#_Toc482566402)

[Tabel 2 - Sensor 2 14](#_Toc482566403)

[Tabel 3 - Registrisse kirjutamine 20](#_Toc482566404)

[Tabel 4 - Kinnituse vastus 20](#_Toc482566405)

[Tabel 5 - Registrist lugemine 21](#_Toc482566406)

[Tabel 6 - Registrist õnnestunud lugemise vastus 21](#_Toc482566407)

[Tabel 7 - Registrist nurjunud lugemise vastus 21](#_Toc482566408)

[Tabel 8 - Saadetud baidid 22](#_Toc482566409)

[Tabel 9 - Vastuvõetud baidid 23](#_Toc482566410)

[Tabel 10 - Registeri leht 0 40](#_Toc482566411)

[Tabel 11 - Registri leht 1 43](#_Toc482566412)

# 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 kinnitatult inimese küljes, kus üle keha on paigutatud mitu erinevat sensorit. Välitingimustes tuleb arvestada keskkonna mõjudega. Merel paiknevat sõidukit 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 ning püsiva ühenduse, kuid seejuures tuleb arvestada, et nad võivad liikumisel venida ning ühendused võivad puruneda. Selle ära hoidmiseks tuleb neile anda ruumi. Mida pikem on ühendus, seda rohkem esineb häireid, seega tuleb protokolli valikus arvestada veakindlusega.

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

Järgnevalt on välja toodud lõputöö käigus lahendatavad ülesanded:

* Prototüübi riistvara arendus
* Mikrokontrolleri seadistamine töötamaks mitme kiirendusanduriga
* Kiirendusandurite seadistamine vastavalt algoritmi poolt sisendandmetele esitatavatele nõetele

# Riistvara

Süsteemi komponendid:

Arendusplaat STM32 -F446RE [1]

Kiirendusandur Bosch BNO055 [2]

## Süsteemi nõuded

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

Arendusplaate on mitmeid, kuulsamad neist on Arduino [3], Raspberry PI [4] ning BeagleBone [5]. Arduino arendusplaate on mitmeid, üks võimekamaid nendest on Arduino STAR – OTTO [6], mis on töötatud välja koos STMicroelectronic-siga. Paraku antud plaat langeb välja, selletõttu et sisend-väljund porte on vähe. Raspberry PI võimekam plaat – Raspberry PI 3 Model B [7] langeb välja sama põhjusega. BeagleBoardi võimekam plaat – BeagleBoard-X15 [8] jääb samuti välja, sellepärast et tegemist on arvutiga, mille peal töötab Linux operatsiooni süsteem [9], kusjuures sama takistus on ka Raspberry PI ja Arduinol. Probleem seisneb selles, et operatsioonisüsteem vajab resurssi töötamiseks. Isegi kui nendel arendusplaatidel oleks vajalikud liidesed olemas, võib tekkida olukord, kus operatsioonisüsteem piirab jõu kasutuse enda jaoks ning kukkumisetuvastamis süsteem kannataks selle tulemusel. STMelectronics pakub arendusplaati STM32F446 [1], mis on antud töö jaoks sobilik.

Kiirendusanduri valikul tuleb tähele panna, et arendusplaat toetaks andmevahetuseks samu liideseid. Andur peab mõõtmete poolest olema väike ning kerge, sest on võimalus, et see paigutatakse riide külge. Lisaks peale kiirendusanduri võiks sensoril olla mõningaid muid andureid, näiteks temperatuuri andur. 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 paraku langeb välja, sest suhtluseks kasutab I2C või SPI porti. Võttes arvesse juhtmete kaugused ei taga need protokollid vea kindlat andmeside. Bosch pakub mitmeid kiirendusandureid sealhulgas kiirendusandurit Bosch BNO055 [2], mis sobib antud töö jaoks.

Kiirendusandur Bosch BNO055 [2] andmevahetuseks on toetatud HID-I2C, I2C ning UART. Arendusplaat STM32 F446RE [1] pakub suhtluseks I2C, UART, SPI, SAI, CAN, GPIO, SDIO. Pidades silmas seda, et juhtme pikkused võivad olla mõnikümmend sentimeetrit või rohkemgi, siis tuleb võtta kasutusele mingi teine protokoll näiteks RS-485. RS-485 protokoll põhineb UART protokollil, seega oleks võimalik kasutada seda protokolli.

## Arendusplaat STM32 F446RE:

Tähtsamad tehnilised andmed [1]:

Protsessor: 180 MHz Cortex-M4 [11]

Mälu: 512 kB Välk mälu, 128 KB SRAM

Toide: 1.7 – 3.6 V

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

Arendusplaadiks sai valitud STM32 Nucleo F446RE. Valikul tuli arvestada, et tegemist on reaalajasüsteemiga. Antud töö raames tähendab, et protsessor peab olema suuteline tihedalt andmeid vastu võtma, neid töötlema ning olema suuteline algoritmiga toime tulema. Plaat peab võimaldama ühendada mitmeid sisend väljund seadmeid. Kaks kiirendusandurit, Bluetooth mooduli ning mõne muu vaja mineva seadme jaoks, näiteks LED tuled. Mõõtmetelt on seade samuti kompaktne: 82.50 mm x 70.00 mm. STMicroelectronics pakub enda mikrokontrolleritele tarkvara STM32Cube [12], mis lihtsustab arendaja tööd. Tarkvaraga on võimalik genereerida initsialiseerimis kood. Arendaja määrab pordid, protokollid ja konfiguratsiooni seaded ning STM32Cube programm genereerib esialgse koodi.

## Kiirendusandur Bosch BNO055

Tähtsamad tehnilised andmed [2]:

Andurid: Kiirendusandur, güroskoop, geomagnetiline sensor

Toide: 2.4 – 3.6 V

Liidesed: HID-I2C, I2C, UART

Kiirendus vahemik: ± 2g - ± 16 g

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 sagedasti andmeid, milleks sensor sobib, sest 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 adafruit-i [14].

### Kiirendusandur Bosch BNO055 Register

Register on jagatud kaheks loogiliseks leheks. Leht 1 omab endas sensori põhist konfiguratsiooni seadeid. Leht 0 omab kõiki muid konfiguratsiooni parameetreid ning väljund andmeid [15].

## Protokollid

Arendusplaadil ja kiirendusanduril on ühiseks liidesteks I2C ning 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 keeruline, sest RS-485 põhineb UART protokollil.

### Protokoll UART

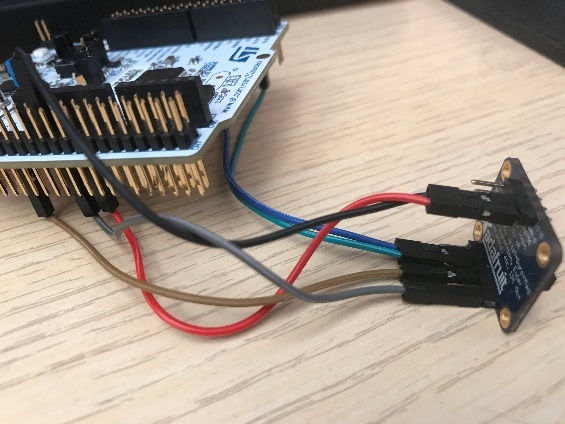
Laialdaselt kasutatud protokoll, mida kasutatakse andmete saatmiseks. UART protokoll on asünkroonne, mis tähendab, et puudub taktsignaal. Andmete saatmisel on sõnumil algus ja lõpp bitt. Oluline, et mõlemad seadmed töötaksid samal boodikiirusel. UART protokolli andmevahetuse kaugus on ~2m [16]. Boodikiirus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

Et saata 1 Bait andmeid, tuleb saata kokku 10 bitti. 1 bitt on algus bitt, 8 bitti andmeid ning 1 stopp bitt.

Tabel 1 - 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 1 , kus roheline juhe tähistab TXD signaali, ning sinine juhe RXD signaali.

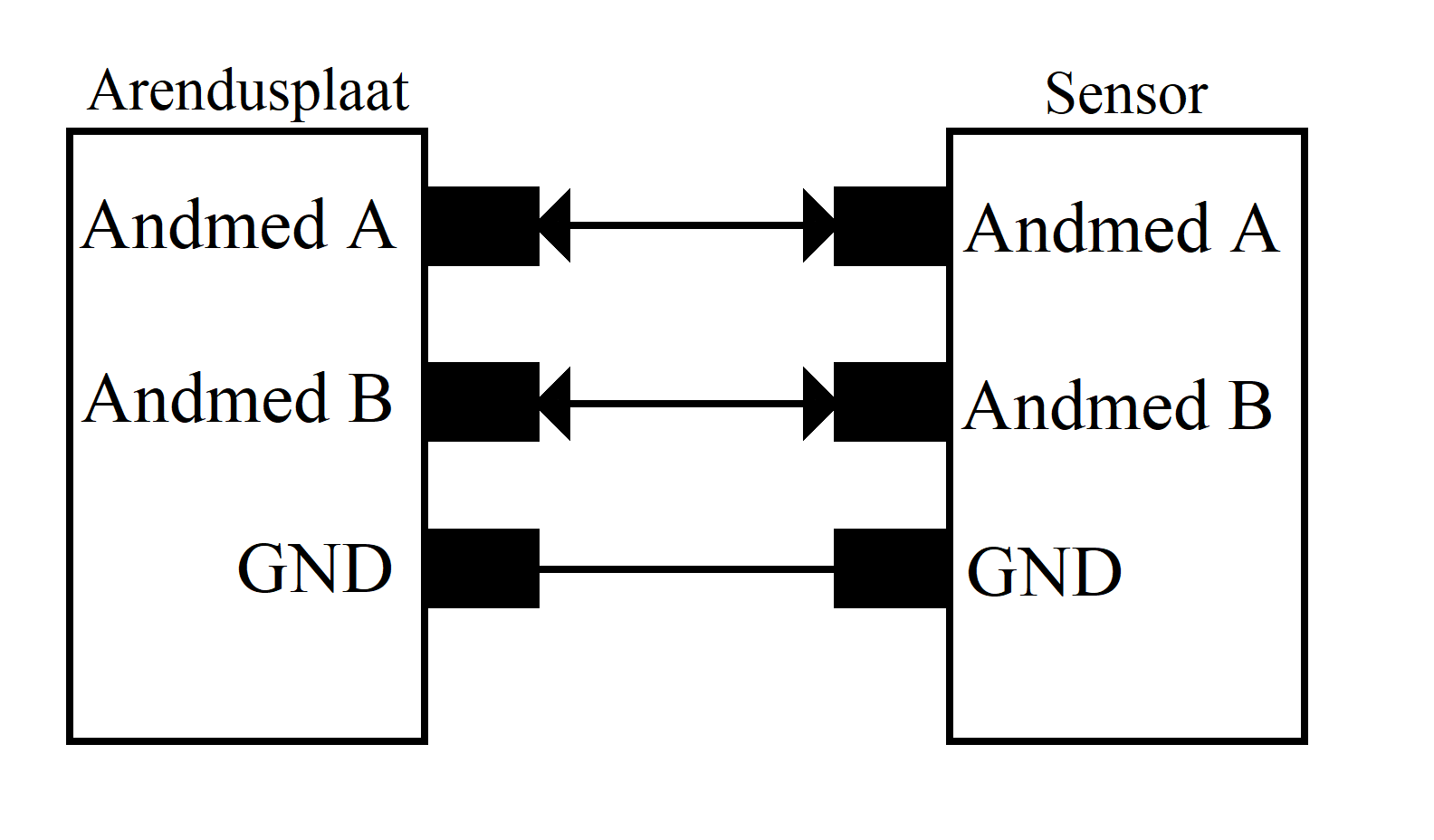


Joonis 1 - TXD ja RXD

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, mis UART, kuid erineb liinidraiveri poolest. Liinidraiver konverteerib ühesuunalise UARRT signaali kahesuunaliseks. See annab kaks andmeliini. Kahesuunalise signaali eeliseks on see, et süsteemil on väiksem müra ning lubab pikemaid ühendusi. RS-485 protokolli andmevahetuse kaugus on ~1200 m. RS-485 ühendus on näidatud joonisel 2.



Joonis 2 - RS-485 ühendus

## Ülevaade sarnastest lahendustest

Kukkumisetuvastus seadmeid on mitmeid, kuid sellist toodet, mis vastab nõuetele puudub.

### Wearable sensors for Reliable fall detection

Tegu on Ad hoc tüüpi võrguga. Mõte seisneb selles, et on baas seade, mille külge teised seadmed ühilduvad. Sensorid pannakse keha külge ning baas seadmele saadetakse sensorite informatsioon. Andmete töötlus toimub baas seadmes. Lahendusel on mõned puudused. Kuna andmete töötlus toimub baas jaamas, tähendab see, et kui on palju sensoreid, siis andmete töötlus aeglustab kogu protsessi. Andmed kuhjuvad, võib juhtuda, et selleks hetkeks kui inimene kukub, toimub parajasti mingi teise sensori andmete töötlus. Samuti võib esineda ka andmete kadu. Sensoriks kasutatakse MICA2DOT 2 sensor [17], mis kasutab operatsiooni süsteemiks 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 kukkumised, mis olid tegelikult näiteks istumised ja tõusmised. Kasutatakse mitut sensorit, mille abil saab tuvastada keha hoiakuid: Seismine, istumine, venitamine ning lamamine. Selleks ühendatakse üks sensor rinna külge ning teine jala külge. Lahendus on hea, sest tõe poolest on nii tagatud rohkem tõepäraseid kukkumisi. Laeva peal lahendus ei toimiks – pole arvestatud kõikumistega, võimalik, et selle alusel oleks võimalik lahendus leida. [20]

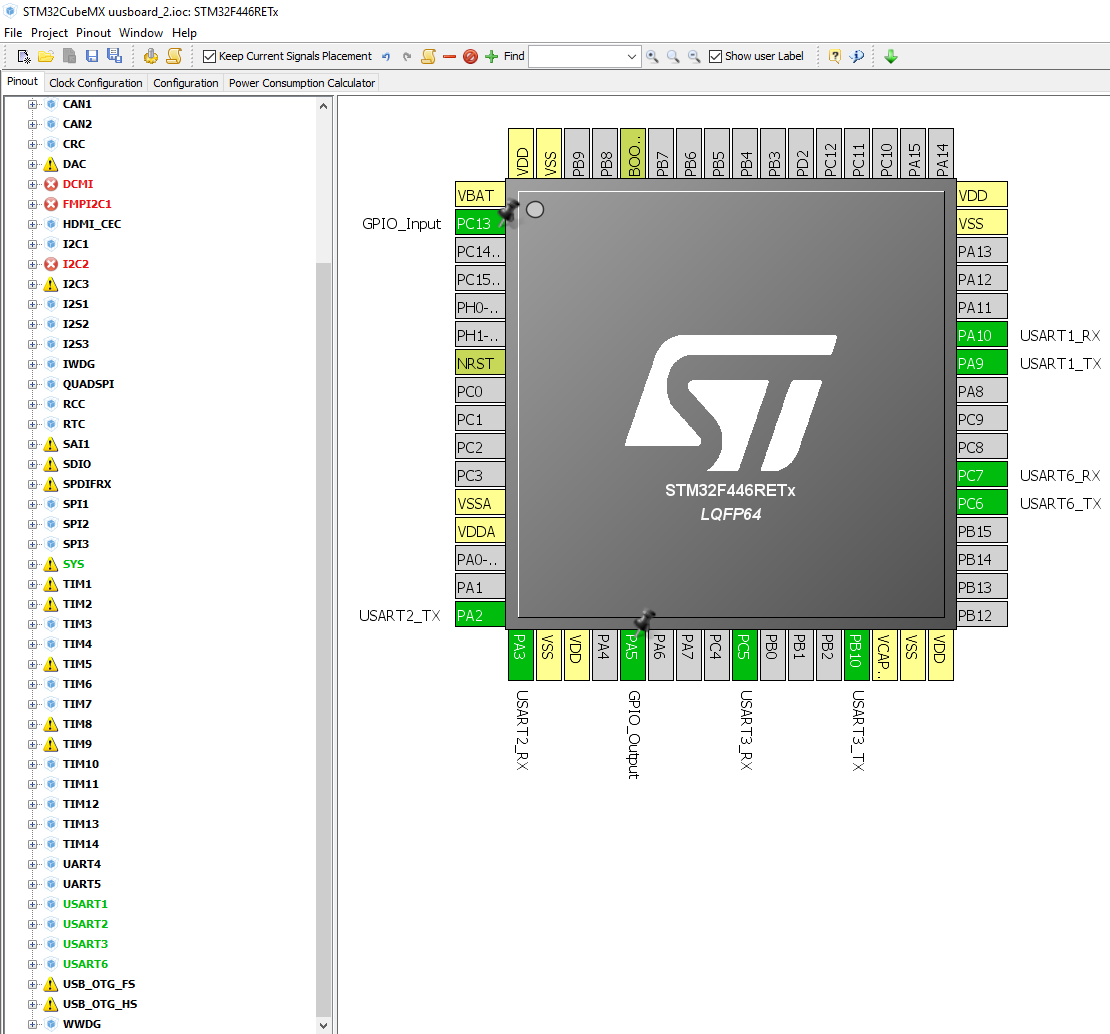
### Philips GoSafe

Tegemist on seadmega, mis on suunatud nii sise- kui välistingimustesse. Asukoha tuvastamiseks on kasutusel Wifi, GPS, helialarm ning lisaks salvestatakse aegajalt salvestab asukohta kasutades 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 reageerimis keskusele, kus uuritakse, mis inimesel täpsemalt juhtus ning seejärel tugiisik võtab 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. Firmalt on võimalik tellida tellimustöö. Antud probleemile oleks 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



STM32F446-RE arendusplaadil on 76 PIN-i. Kasutusel on 4 UART liidest. 2 UART liidest kasutatakse sensoritega suhtlemiseks. 2 Andmete kuvamiseks kui ka *debugimiseks*.

USART1 - Sensor 1 on ühendatud järgmistesse PIN-idesse, mida näitab tabel 2.

Tabel 2 – Sensor 1

|  |  |
| --- | --- |
| Sensori Port | Arendusplaadi Port |
| PS0 | GND |
| PS1 | +3V3 |
| Vin | +3V3 |
| GND | GND |
| SDA | PA10 |
| SCL | PA9 |

USART3 - Sensor 2 on ühendatud järgmistesse PIN-idesse, mida näitab tabel 3.

Tabel 3 - Sensor 2

|  |  |
| --- | --- |
| Sensori Port | Arendus plaadi Port |
| PS0 | GND |
| PS1 | +3V3 |
| Vin | +3V3 |
| GND | GND |
| SDA | PC5 |
| SCL | PB10 |

USART2 ning USART6 on kasutusel *debugimiseks*.

# 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].

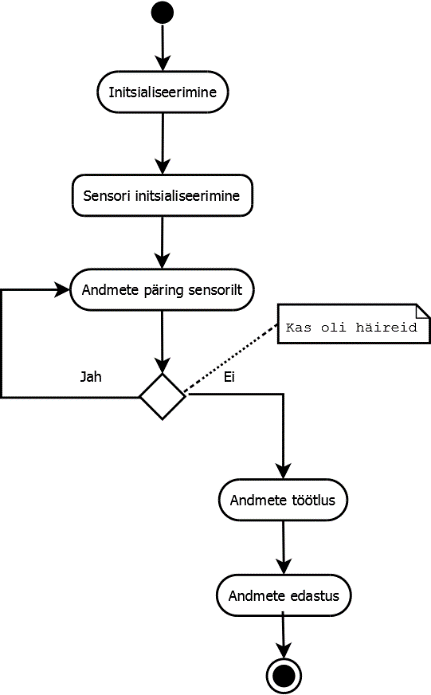
## Programmi töö

Esialgu initsialiseeritakse kõik sisemised seadistused arendusplaadil, seejärel saadetakse konfiguratsiooni seaded sensorile. Peale initsialiseerimist toimub andmete päring sensorilt. 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 hoopis vea teate, see võib juhtuda andmete pärimise alguses või lõpus. Sellest hoolimata vigaste andmete korral, ei tohi neid töödelda.

Kui andmete päring õnnestus, toimub andmete töötlus. Et saada loetavad andmeid, tuleb panna 2 baiti kokku. Täisarv koosneb kahest baidist, kus esimene bait tähistab vähima kaaluga baiti ning teine bait suurimat. Programmi tööd kirjeldab joonis 6.

Näiteks: MSB – 1111 1110 (OxFE) . LSB – 0000 0110 (0x06), siis töödeldud kujul on esitatud arvu -506.



Joonis 8 - Programmi töö

Arendusplaadi initsialiseerimine:

Arendusplaadi initsialiseerimine on näidatud joonisel .

HAL teegi initsialiseerimine [[1]](#endnote-1)

Peab olema esimene funktsioon, mida käivitatakse. HAL draiver on disainitud, et pakkuda rikkaliku API-si, et suhelda aplikatsiooni kõrgemate kihtidega. Iga draiver koosneb funktsioonide hulgast, mis katavad enamus väliseadme funktsioone.

Näide:

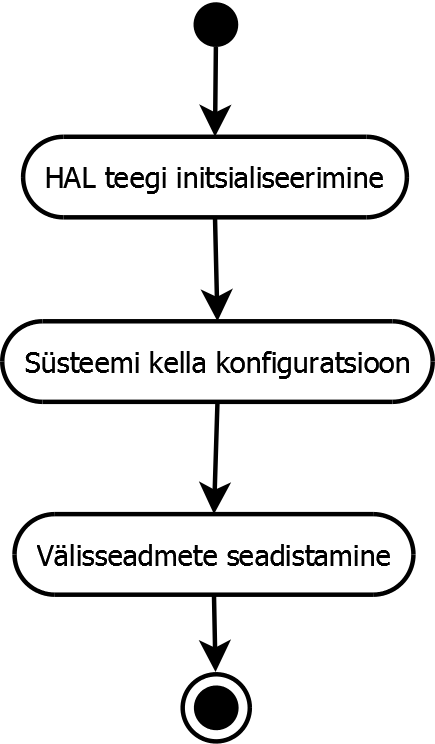
HAL\_UART\_Abort (UART\_HandleTypeDef \*huart) – Tühistab käimasolevad edastused.

Süsteemi kella konfiguratsioon:

Initsialiseerib sisemise väljund pinge. Initsialiseerib CPU, AHB ja ABP siini kellad. Süsteemi kell käsitleb arvutisüsteemis kõiki sünkroniseerimisist.

Välisseadmete seadistamine:

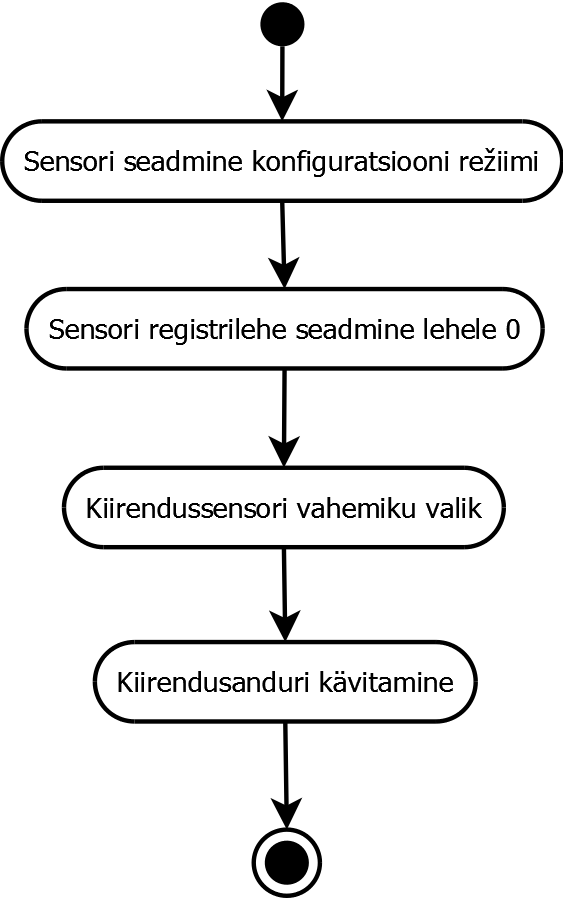
Initsialiseerib kõik konfigureeritud väljundid: GPIO, UART, USART.



Joonis 9 - Arendusplaadi initsialiseerimine

Sensori Initsialiseerimine:

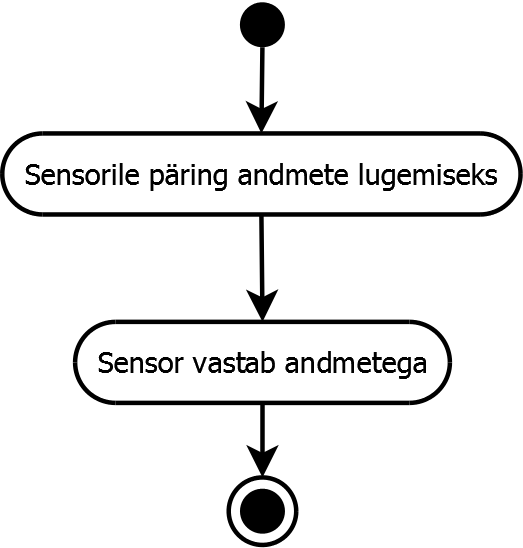
Sensori käivitamisel saadetakse sensorile konfiguratsiooni seaded, mis seadetega sensor tööle hakkab. Esmalt lülitakse sensor konfiguratsiooni režiimi, seejärel valitakse registrileht 0. Sensorile saadetakse kiirendusanduri vahemik 8G ning seejärel käivitatakse sensor kiirendusanduri režiimis. Sensori initsialiseerimis kirjeldab joonis 8.



Joonis 10 - Sensori initsialiseerimine

Andmete päring:

Andmete saamiseks tuleb sensorile saata andmete lugemiseks päring, mis on näidatud tabel 5-es. Sensor saadab selle peale vastuse. Vastus võib endas sisaldada kas õigeid andmeid või viga. Õiged andmed on kirjeldatud tabelis 6 ning vigane vastus on kirjeldatud tabelis 7. Andmete päringut kujutab joonis 9.

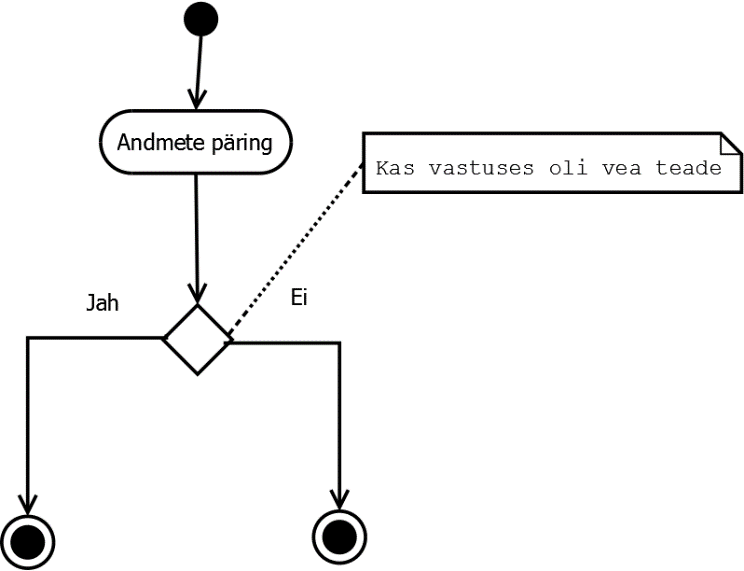


Joonis 11 - Andmete päring

Häirete kontroll:

Peale andmete päringut kontrollitakse, kas andmed olid õiged. Õnnestunud lugemise vastus on kirjeldatud tabelis 6. Nurjunud lugemise vastus on kirjeldatud tabelis 7.

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 12 - Häirete kontroll

Andmete edastus:

Juhul kui andmed on õiged, siis saadetakse andmed USB porti. Andmeid saab lugeda kasutades programmi RealTerm[[2]](#endnote-2).

Kiirendusanduri kalibreerimine:

Kiirendusandureid on võimalik kalibreerida, selleks tuleb hoida all sinist nuppu viieks sekundiks, mis on näidatud joonisel 13 B1.

Seejärel on kahesekundiline auk. Juhul kui selle kahesekundi jooksul vajutada sinist nuppu seadistatakse sensor kahte. Juhul kui nuppu ei vajutata, seadistatakse sensor üks.

Tulukest LD2 ning nuppu B1 kasutatakse seadistamisel joonis 13 ning joonis 14.

Esialgu seadistatakse sensori X telge. Tuluke LD2 hakkab vilkuma sagedusega 1000 ms. Sensor tuleb paigutada X telje suhtes tasasele pinnale ning seejärel tuleb vajutada sinist nuppu B1.

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.

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.

Järgnevalt saadetakse sensorile andmed kalibreerimis andmed ning sensor on seadistatud.

Testimine:

Andmete kuvamiseks on kaks võimalust. Üheks on ühendada arendusplaat USB kaabliga arvuti külge ning kasutada programmi RealTerm, et jälgida andmeid.

Realterm seaded:

Realterm seaded on näidatud joonisel 11 ja joonisel 12.

# Seadistamine ja testimine

Arvuti ja arendusplaat ühendatakse arvutiga läbi mikro-USB juhtme. Läbi juhtme toimub koodi peale laadimine, *debuggimine*. STMicroeletronics on loonud abistava programmi STM32CUBEMX. Läbi graafilise liidese määratakse initsialiseerimis seaded, mille järel rakendus genereerib initsialiseerimis koodi.

## STM32CubeMX seadistamine:

STM32CubeMX programmis olulised projekti seaded. IDE valik – MDK-ARM V5. Aktiveerida tuleb USART1, USART2, USAR3, USART6 ning neile määrata globaalne katkestus. Koodi genereerimise alam menüüst tuleb vaadata, et kasutaja kood jääks alles igakord kui programm genereerib uue koodi.

#### USART1 Seadistused

Vajalikud seaded, et seadistada USART1, USART2, USART3 ja USART6 . USART1 ja USART2 kaudu käib suhtlus arendusplaadi ja sensorite vahel. USART3 ja USART6 kasutatakse andmete saatmiseks ning *debuggimiseks*.

**Seaded:**

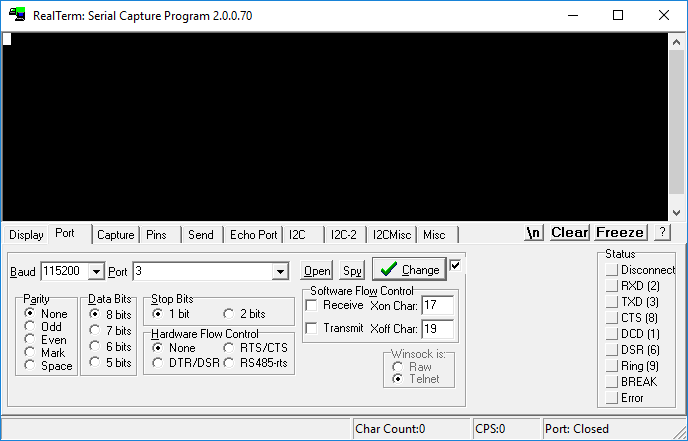
Boodikiirus– 115200 Bitti/s

Sõna pikkus - 8 Bitti (sealhulgas paarsus)

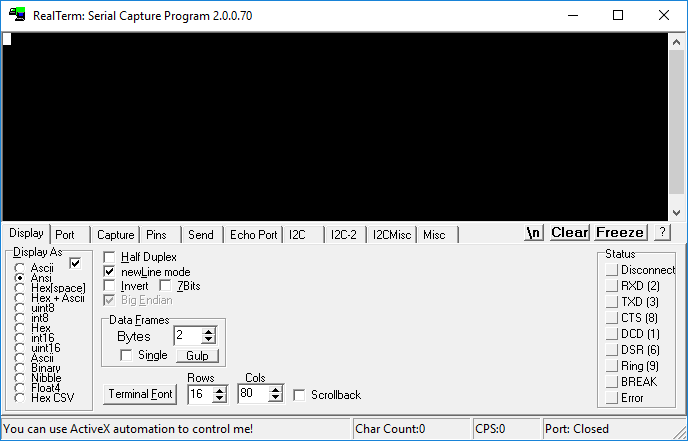
Paarsus – Puudub

Stopp bitte – 1

Globaal katkestus aktiveerida

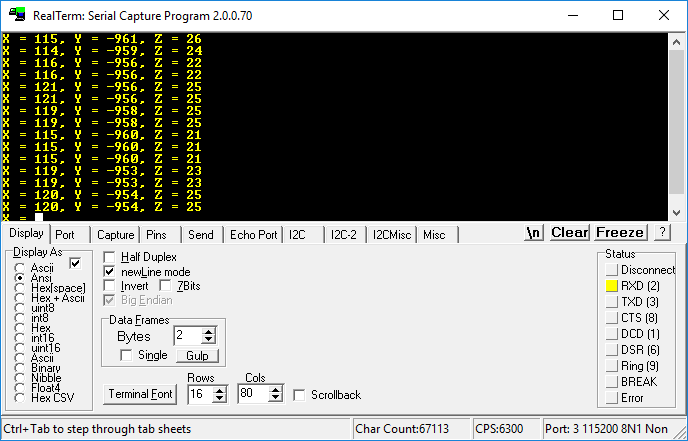


Joonis 13 - Realterm Port seaded



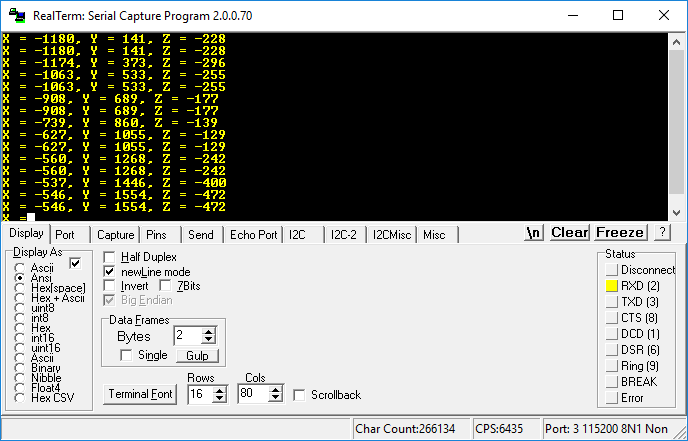
Joonis 14 - Realterm Displei seaded

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



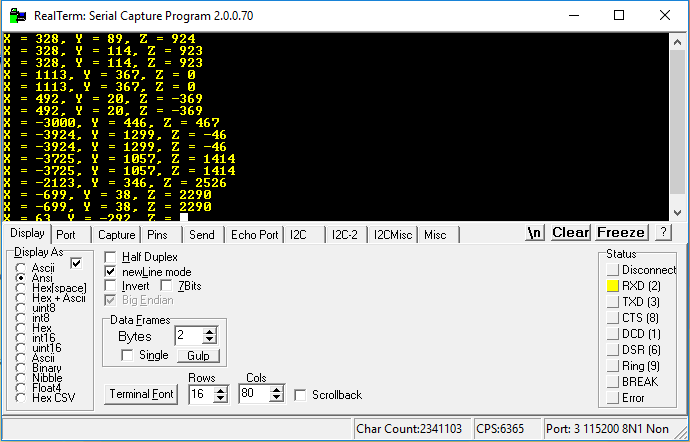
Joonis 15 - Paigal olek

Kui toimub liikumine näiteks laev kõigub, siis andmed peaksid jääma -2000 ja 2000 vahele. Nagu näidatud joonisel 14.



Joonis 16 - Liikumine

Kui toimub tugevam kokkupõrge peaksid andmed olema -2000 ja 8000 vahemikus. Nagu näitab joonis 15.



Joonis 17- Tugevam põrutus

**Näide kuidas lugeda temperatuuri:**

Temperatuuri andmed asuvad registris 34, mis ise asub 0 registri lehel. Kokku tuleb saata selleks saata 4 baiti, millest 1 bait - algus bait, 2 bait – lugemis bait, 3 – loetava registri aadress, 4 – pikkus (kui palju andmeid saata). Kui sensor saab õiged andmed kätte, siis vastab ta 3 baidiga, millest 1 bait – vastuse päise, 2 – bait pikkus (kui palju andmeid saadetakse), 3 – andmed. Saadetud baitide näidised asuvad tabelis 8 ning vastuvõetud baidid asuvad tabelis 9.

Tabel 9 - Saadetud baidid

|  |  |  |  |
| --- | --- | --- | --- |
| Bait 1 | Bait 2 | Bait 3 | Bait 4 |
| 0xAA | 0x01 | 0x34 | 0x01 |

Tabel 10 - Vastuvõetud baidid

|  |  |  |
| --- | --- | --- |
| Bait 1 | Bait 2 | Bait 3 |
| 0xBB | 0x01 | 0x20 |

Tabel 4 - Registrisse 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 | <..> | <..> | <..> | … | <..> |

Tabel 5 - Kinnituse 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 |

Tabel 6 - Registrist lugemine

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

Tabel 7 - Registrist õnnestunud lugemise vastus

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

Tabel 8 - Registrist nurjunud lugemise vastus

|  |  |
| --- | --- |
| Bait 1 | Bait 2 |
| Vastuse 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 |

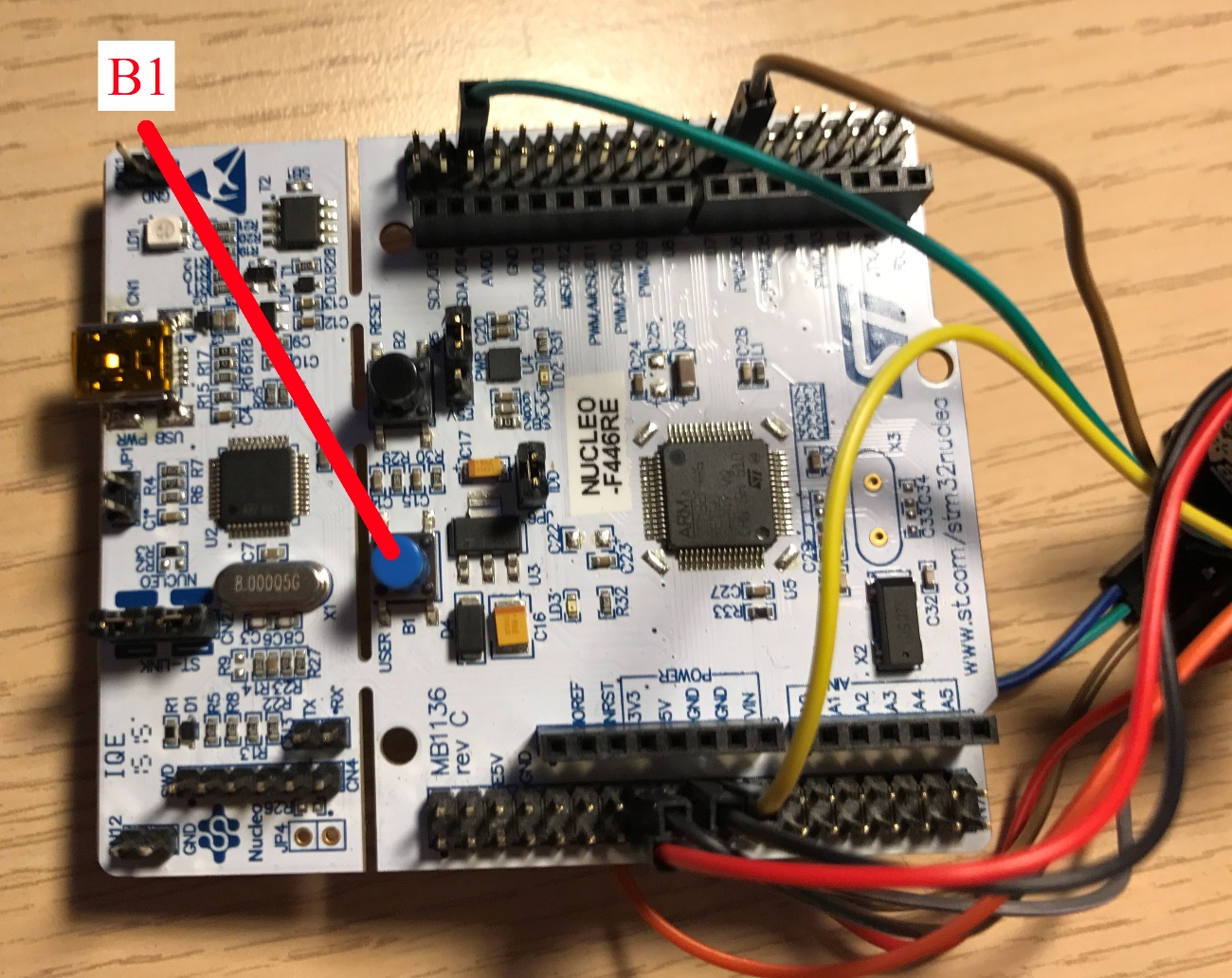
# Kokkuvõte

Kasutatud kirjandus

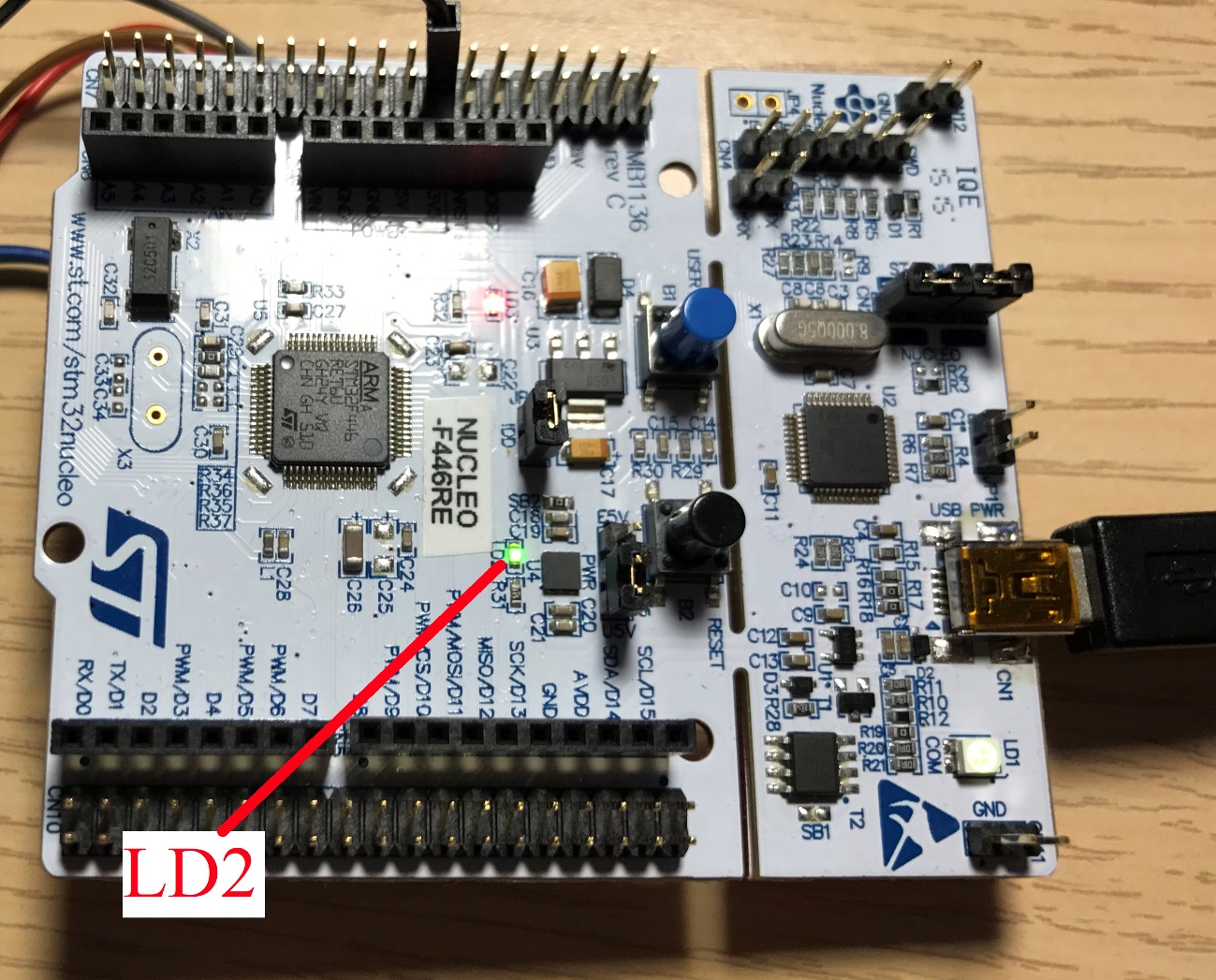
# Bibliography

|  |  |
| --- | --- |
| [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.arduino.cc/. |
| [4] | [Võrgumaterjal]. Available: https://www.raspberrypi.org/. |
| [5] | [Võrgumaterjal]. Available: http://beagleboard.org/bone. |
| [6] | [Võrgumaterjal]. Available: http://www.arduino.org/products/boards/arduino-star-otto. |
| [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] | L. S. Sterling, The Art of Agent-Oriented Modeling, London: The MIT Press, 2009. |
| [26] | „asdf,“ [Võrgumaterjal]. Available: http://www.st.com/en/microcontrollers/stm32f446.html?querycriteria=productId=LN1875. |
| [27] | [Võrgumaterjal]. |
| [28] | [Võrgumaterjal]. Available: http://www.ti.com/tool/tmdslcdk6748#descriptionArea. |
| [29] | [Võrgumaterjal]. Available: https://www.shimmersensing.com/products/shimmer3-development-kit#download-tab. |

Lisa 1 – Registri leht



Joonis 18 - Arendusplaadi pilt



Joonis 19- Led tuluke

Tabel 11 - Registeri leht 0

|  |  |
| --- | --- |
| **Registri aadress** | **Registri nimi** |
| 7F-6B | Reserved |
| 6A | MAG\_RADIUS\_MSB |
| 69 | MAG\_RADIUS\_LSB |
| 68 | ACC\_RADIUS\_MSB |
| 67 | ACC\_RADIUS\_LSB |
| 66 | GYR\_OFFSET\_Z\_MSB |
| 65 | GYR\_OFFSET\_Z\_LSB |
| 64 | GYR\_OFFSET\_Y\_MSB |
| 63 | GYT\_OFFSET\_Y\_LSB |
| 62 | GYR\_OFFSET\_ X\_MSB |
| 61 | GYR\_OFFSET\_X\_LSB |
| 60 | MAG\_OFFSET\_Z\_MSB |
| 5F | MAG\_OFFSET\_Z\_LSB |
| 5E | MAG\_OFFSET\_Y\_MSB |
| 5D | MAG\_OFFSET\_Y\_LSB |
| 5C | MAG\_OFFSET\_X\_MSB |
| 5B | MAG\_OFFSET\_X\_LSB |
| 5A | ACC\_OFFSET\_Z\_MSB |
| 59 | ACC\_OFFSET\_Z\_LSB |
| 58 | ACC\_OFFSET\_Y\_MSB |
| 57 | ACC\_OFFSET\_Y\_LSB |
| 56 | ACC\_OFFSET\_X\_MSB |
| 55 | ACC\_OFFSET\_X\_LSB |
| 43 – 54 | Reserved |
| 42 | AXIS\_MAP\_SIGN |
| 41 | AXIS\_MAP\_CONFIG |
| 40 | TEMP\_SOURCE |
| 3F | SYS\_TRIGGER |
| 3E | PWR\_MODE |
| 3D | OPR\_MODE |
| 3C | Reserved |
| 3B | UNIT\_SEL |
| 3A | SYS\_ERR |

|  |  |
| --- | --- |
| **Registri aadress** | **Registri nimi** |
| 39 | SYS\_STATUS |
| 38 | SYS\_CLK\_STATUS |
| 37 | INT\_STA |
| 36 | ST\_RESULT |
| 35 | CALIB\_STAT |
| 34 | TEMP |
| 33 | GRV\_Data\_Z\_MSB |
| 32 | GRV\_Data\_Z\_LSB |
| 31 | GRV\_Data\_Y\_MSB |
| 30 | GRV\_Data\_Y\_LSB |
| 2F | GRV\_Data\_X\_MSB |
| 2E | GRV\_Data\_X\_LSB |
| 2D | LIA\_Data\_Z\_MBS |
| 2C | LIA\_Data\_Z\_LSB |
| 2B | LIA\_Data\_Y\_MBS |
| 2A | LIA\_Data\_Y\_LSB |
| 29 | LIA\_Data\_X\_MBS |
| 28 | LIA\_Data\_X\_LSB |
| 27 | QUA\_Data\_Z\_MSB |
| 26 | QUA\_Data\_Z\_LSB |
| 25 | QUA\_Data\_Y\_MSB |
| 24 | QUA\_Data\_Y\_LSB |
| 23 | QUA\_Data\_X\_MSB |
| 22 | QUA\_Data\_X\_LSB |
| 21 | QUA\_Data\_W\_MSB |
| 20 | QUA\_Data\_W\_LSB |
| 1F | EUL\_Pitch\_MSB |
| 1E | EUL\_Pitch\_LSB |
| 1D | EUL\_Roll\_MSB |
| 1C | EUL\_Roll\_LSB |
| 1B | EUL\_Heading\_MSB |
| 1A | EUL\_Heading\_LSB |
| 19 | GYR\_DATA\_Z\_MSB |
| 18 | GYR\_DATA\_Z\_LSB |
| 17 | GYR\_DATA\_Y\_MSB |
| 16 | GYR\_DATA\_Y\_LSB |
| 15 | GYR\_DATA\_X\_MSB |
| 14 | GYR\_DATA\_X\_LSB |
| 13 | MAG\_DATA\_Z\_MSB |
| 12 | MAG\_DATA\_Z\_LSB |
| 11 | MAG\_DATA\_Y\_MSB |
| 10 | MAG\_DATA\_Y\_LSB |
| F | MAG\_DATA\_X\_MSB |
| E | MAG\_DATA\_X\_LSB |
| D | ACC\_DATA\_Z\_MSB |

|  |  |
| --- | --- |
| **Registri aadress** | **Registri nimi** |
| C | ACC\_DATA\_Z\_LSB |
| B | ACC\_DATA\_Y\_MSB |
| A | ACC\_DATA\_Y\_LSB |
| 9 | ACC\_DATA\_X\_MSB |
| 8 | ACC\_DATA\_X\_LSB |
| 7 | Page ID |
| 6 | BL\_Rev\_ID |
| 5 | SW\_REV\_ID\_MSB |
| 4 | SW\_REV\_ID\_LSB |
| 3 | GYR\_ID |
| 2 | MAG\_ID |
| 1 | ACC\_ID |
| 0 | CHIP\_ID |

Tabel 12 - Registri leht 1

|  |  |
| --- | --- |
| **Registri aadress** | **Registri nimi** |
| 7F – 60 | Reserved |
| 5F – 50 | UNIQUE\_ID |
| 4F – 20 | Reserved |
| 1F | GYR\_AM\_SET |
| 1E | GYR\_AM\_THRES |
| 1D | GYR\_DUR\_Z |
| 1C | GYR\_HR\_Z\_SET |
| 1B | GYR\_DUR\_Y |
| 1A | GYR\_HR\_Y\_SET |
| 19 | GYR\_DUR\_X |
| 18 | GYR\_HR\_X\_SET |
| 17 | GYR\_INT\_SETING |
| 16 | ACC\_NM\_SET |
| 15 | ACC\_NM\_THRE |
| 14 | ACC\_HG\_THRES |
| 13 | ACC\_HG\_DURATION |
| 12 | ACC\_INT\_Settings |
| 11 | ACC\_AM\_THRES |
| 10 | INT\_EN |
| F | INT\_MSK |
| E | Reserved |
| D | GYR\_Sleep\_Config |
| C | ACC\_Sleep\_Config |
| B | GYR\_Config\_1 |
| A | GYR\_Config\_0 |
| 9 | MAG\_Config |
| 8 | ACC\_Config |
| 7 | Page ID |
| 6 - 0 | Reserved |

1. 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 [↑](#endnote-ref-1)
2. https://realterm.sourceforge.io/ [↑](#endnote-ref-2)