|  |  |
| --- | --- |
| tallinna tehnikaülikool | |
| Infotehnoloogia teaduskond | |
| Arvutitehnika 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 |
|  |  |
|  |  |
|  |  |

Sisukord

[2.1 Tehnilised andmed 10](#_Toc481531585)

[2.2 Arendusplaadi seadistamine 13](#_Toc481531586)

[2.2.1 STM32CubeMX seadistamine 14](#_Toc481531587)

[3.1 BNO055 kiirendus anduri seadistamine 17](#_Toc481531588)

[4.1 Programmi funktsioonide kirjeldus 18](#_Toc481531589)

Jooniste loetelu

**No table of figures entries found.**

Tabelite loetelu

**No table of figures entries found.**

# Sissejuhatus

Töö eesmärgiks oli arendada platvorm kukkumise tuvastamiseks, mida saaks kasutada eksperimentide läbiviimiseks, andmete kogumiseks ning nende andmete põhjal välja töötata erinevaid algoritme.

Järgnevalt on välja toodud lõputöö käigus lahendatavad ülesanded:

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

Mitme Kiirendusanduriga kukkumise tuvastamise süsteemi arendus

Kirjutan lõputöö eesmärgist, kus seda kasutama hakatakse, mis on nõuded. Millised on taolised tehtud tööd. Mis programme kasutasin, et ülesanne lahendada. Protokollide näide ning miks sai valitud USART. Esinenud probleemid ning kuidas probleemid lahedati.

# Arendusplaadi kirjeldus

Platformiks sai valitud STM32 Nucleo F446RE. STM32 Nucleo F446RE on paindlik platform mille peal prototüüpe arendada. Arendusplaadi valimisel oli oluline, et PIN-e oleks piisavalt, et saaks ühendada rohkem kui ühe sisend väljund seadme.

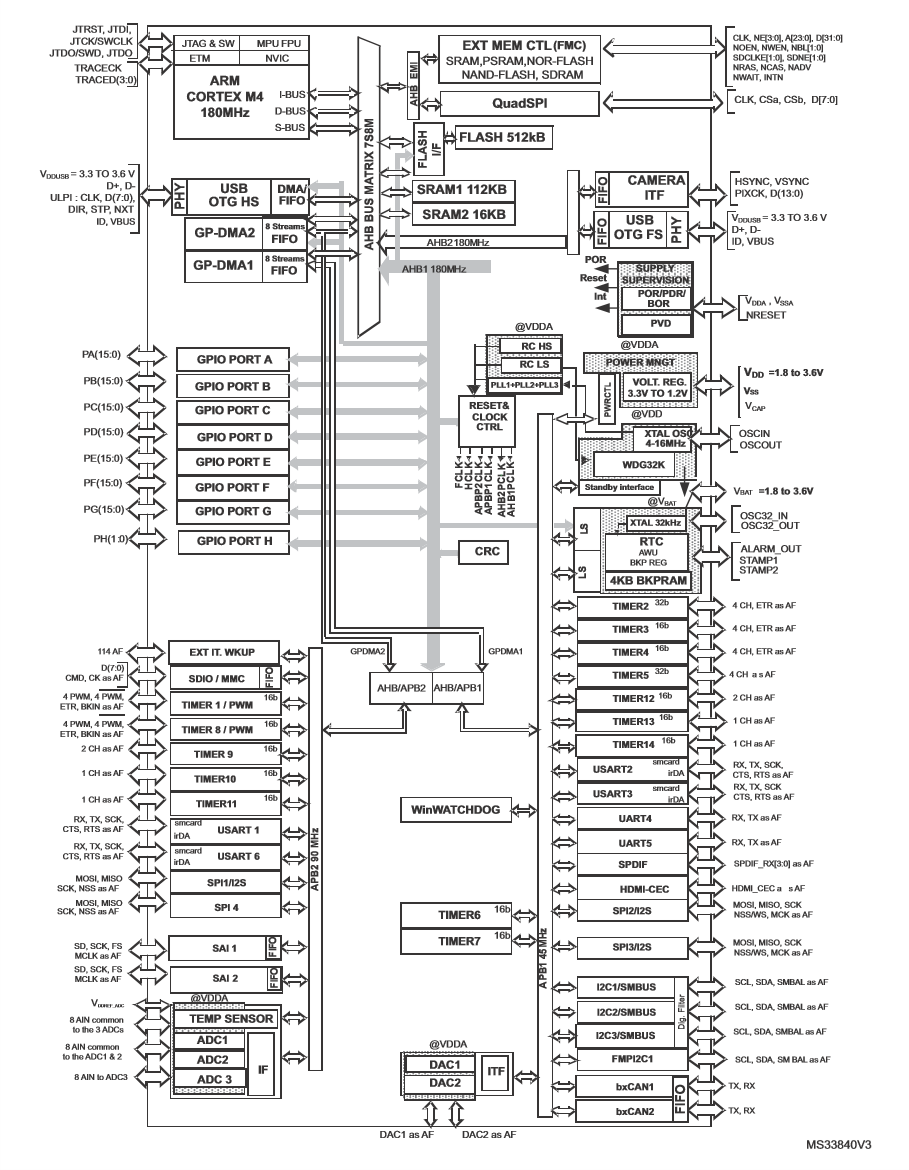
## Tehnilised andmed

Kuhu kirjutada kus ma võtsin need :

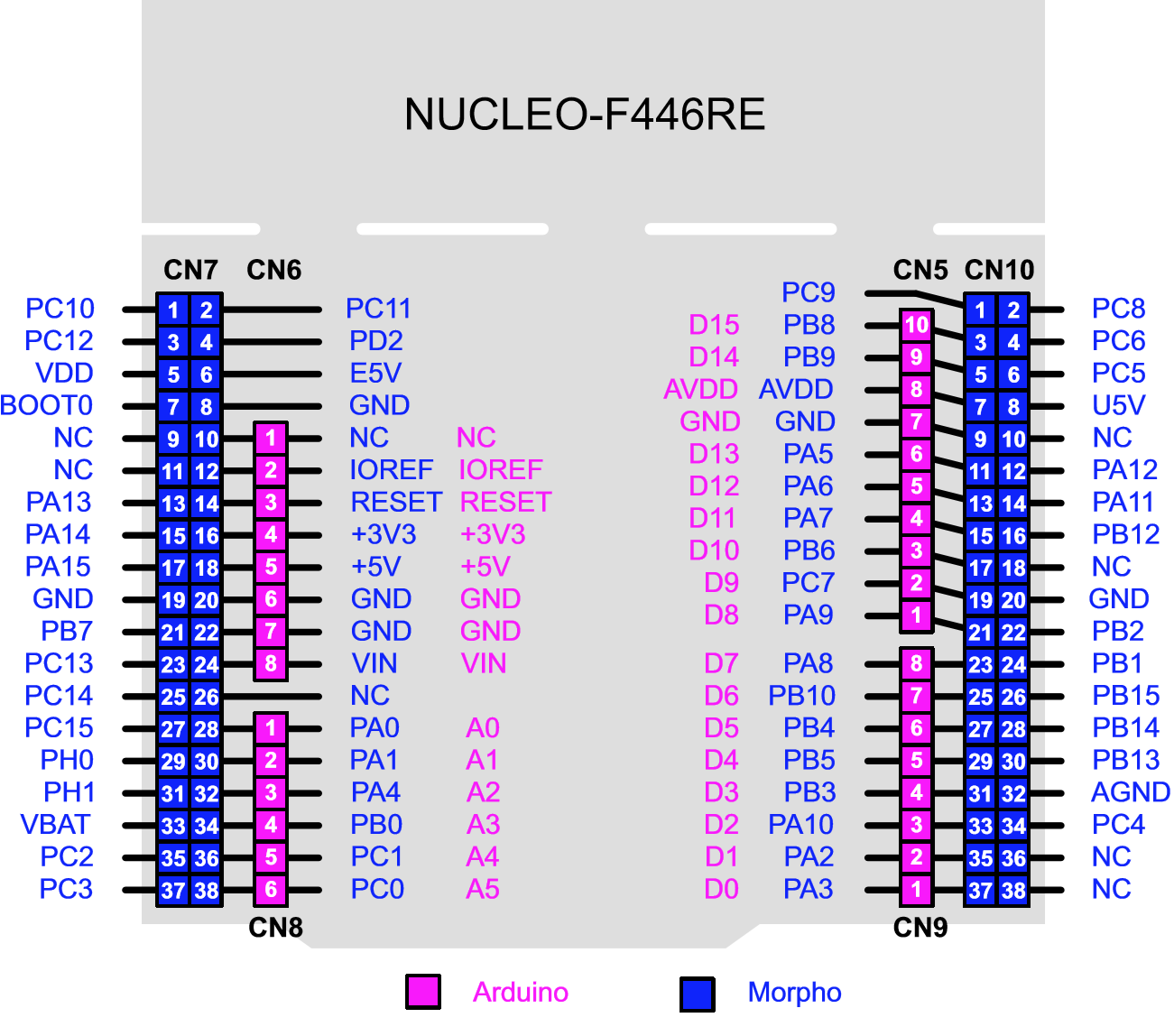
(„http://www.st.com/content/ccc/resource/technical/document/datasheet/65/cb/75/50/53/d6/48/24/DM00141306.pdf/files/DM00141306.pdf/jcr:content/translations/en.DM00141306.pdf”)

* Protsessor
  + Cortex-M4 [[1]](#endnote-1)
* Mälu
  + 512 kB Välk mälu
  + 128 KB SRAM
  + väline mälukontroller kuni 16 bitise siiniga
* Taktsagedus, lähtestus, varu haldus
  + 1.7 V kuni 3.6 V applikatsiooni ja sisend väljund seadmed.
  + POR, PDR, PVD ja BOR ?????????
  + 4 kuni 26 MHz kristallostsillaator
  + Sisemine 16 MHz tehase häälestatud RC ??
  + 32 kHZ ostsillaator RTC jaoks koos kalibreerimisega???
  + Sisemine 32 kHz RC koos kalibreerimisega
* Madal tarne
  + Uinak, Stop, Ootel režiim
  + Vbat supply for RTC, 20x32 bit backup registers + optional 4 KB backup SRAM
* 3x12/bit, 2.4 MSPS ADC: up to 24 channels and 7.2 MSPS in triple interleaved mode
* 2x12-bit D/A converters
* General puropse DMA: 16-stream DMA controller with FIFOs and burst support
* Up to 17 timers: 2x watchdog, 1x SysTick timer and up to twelve 16-bit and two 32-bit timers up to 18+ MHz, each with up to 4 IC/OC/PWM or puse counter
* Debug mode
  + SWD & JTAB interfaces
  + Cortex-M4 Trace macrocell
* Up to 114 I/O ports with interrupt capability
  + Up to 111 fast I/Os up to 90 MHz
  + Up to 112 5 V-tolerant I/Os
* Up to 20 communication interfaces
  + SPDIF-Rx
  + Up to 4 x (SMBus/PMBus)
  + Up to 4 USARTs/2 UARTs (11.25 Mbit/s, ISO7816 interface, LIN, IrDA, modem control)
  + Up to 4 SPIs (45 Mbits/s), 3 with muxed I2S for audio class accuracy via internal audio PLL or external clock
  + 2 x SAI (serial audio interface)
  + 2 × CAN (2.0B Active)
  + SDIO interface
  + Consumer electronics control (CEC) I/F
* Advanced connectivity
  + USB 2.0 full-speed device/host/OTG controller with on-chip PHY
  + USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
  + Dedicated USB power rail enabling on-chip PHYs operation throughout the entire MCU power supply range
* 8- to 14-bit parallel camera interface up to 54 Mbytes/s
* CRC calculation unit
* RTC: subsecond accuracy, hardware calendar
* 96-bit unique ID

**Blokk diagramm:**

****

**PIN-id:**

****

Tehnilised andmed. PIN-ide nimetus. Kasutatud PIN-id. Pinged. LED-id. Nuppude kirjeldused. Joonised.

## Arendusplaadi seadistamine

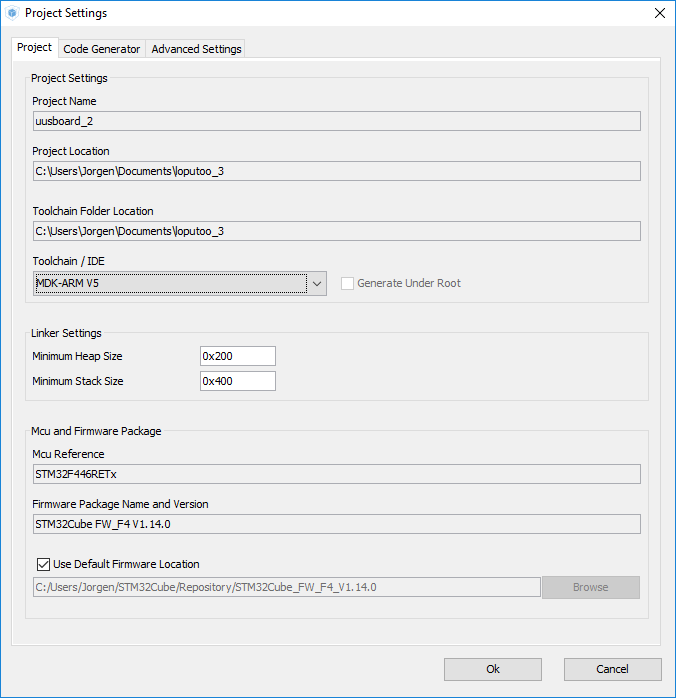
STMicroelectronics on loonud aplikatsiooni STM32CubeMX. Tegu on rakendusega, mis aitab arendajal genereerida initsialiseerimis koodi kasutades graafilist liidest.

### STM32CubeMX seadistamine

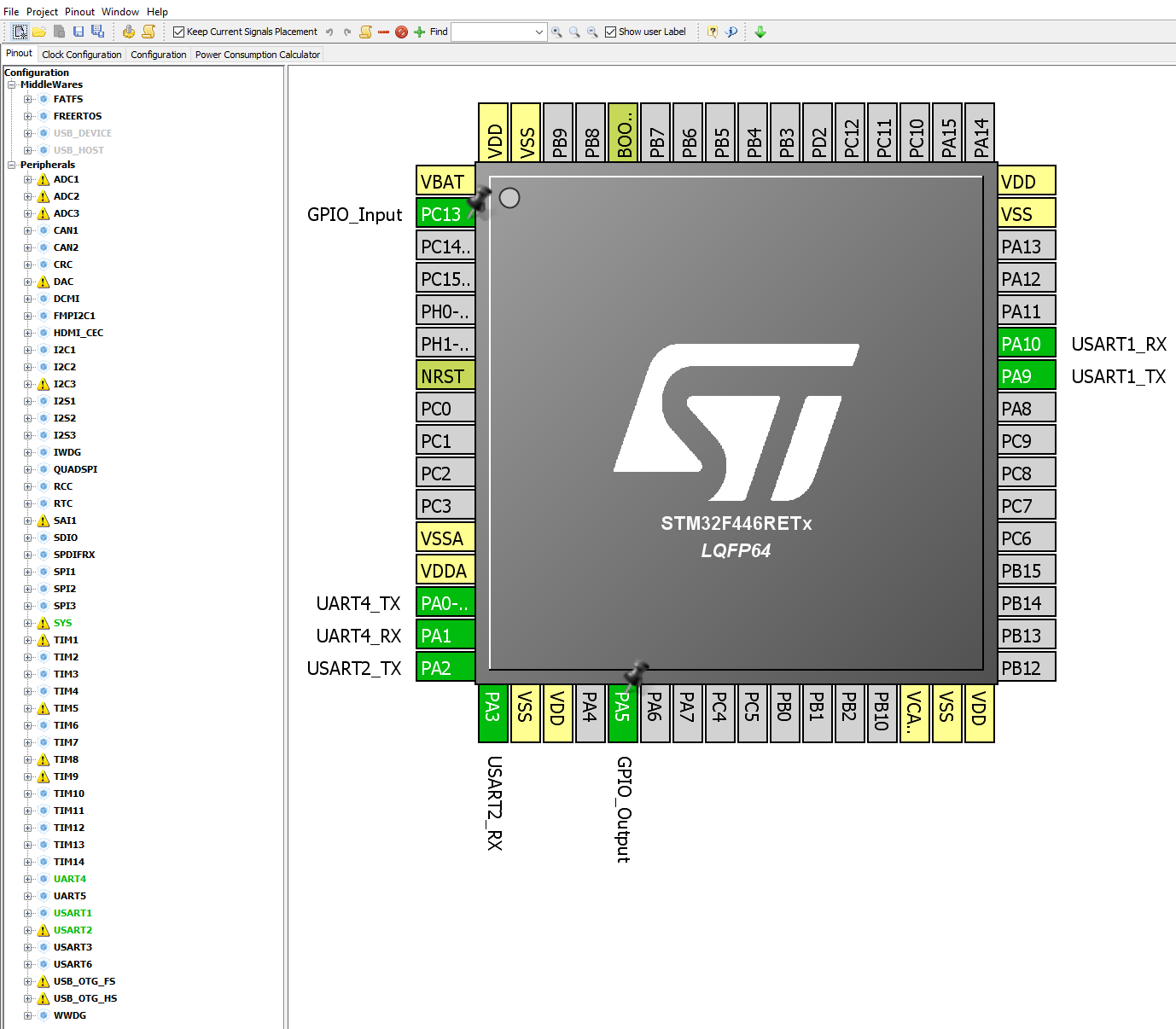
Platformi arendusel vajalikud seaded: vaja muuta õigeks.

**Projekti seaded:**

Oluline on valida Toolchain / IDE: MDK-ARM V5

****

**Arendusplaadi seaded:**

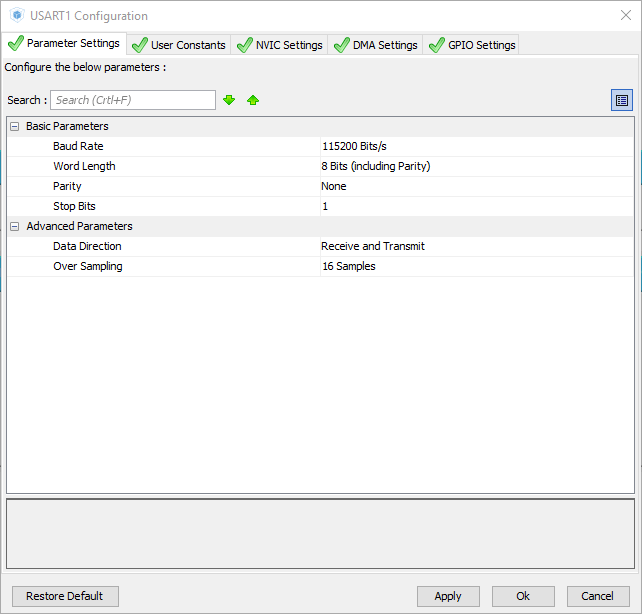


#### USART1 Seadistus

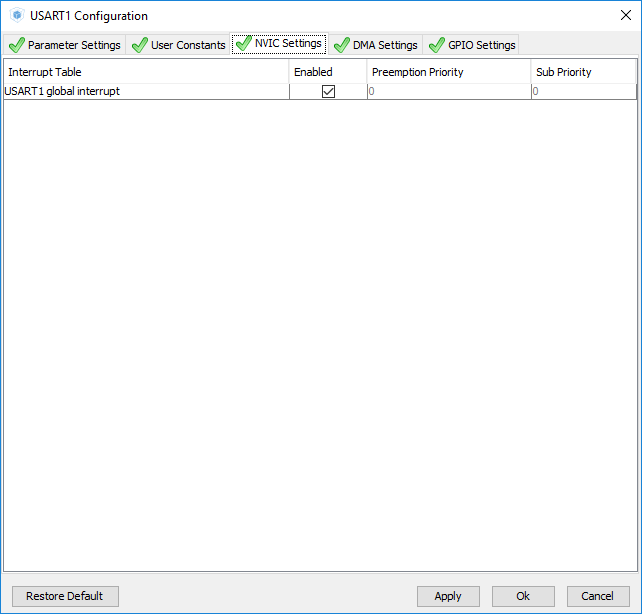
Vajalikud seaded, et seadistada USART1. USART1 kaudu toimub arendusplaadi ning sensori vahel andmete saatmine.

BNO055 sensori nõuded (vaja ülevaadata BNO055 manuaal) :

* Baud rate – 115200 Bits/s
* Word length – 8 Bits (including Parity)



**Global interrupt:**



Kuidas arvutiga ühendada. Mis programme on vaja, et seda arendada (Java, ST-LINK, Keil, Keil STM32 arendusplaadi library, STM32Cube). Baud kiirus, Interrupt.

### Arendusplaadi ühendamine arvutiga

Vajaminevad driverid ja programmid:

* Driverid
  + ST-LINK/V2-1
* Programmid
  + Java
  + MDK-ARM V5
  + STM32CubeMX

**Tüüpiline konfiguratsioon:**

* USB cable type A to mini-B
* Arvuti Windows XP, 7, 8

**Developer Toolchain:**

* IAR EWARM
* Keil MDK-ARM
* GCC-based IDE

# BNO055 Kiirendusandur

Ülevaade kiirendusandrusit, tehnilised andmed jne. Registrite ülevaade. Kuidas sensorile infot saata, kuidas sensor info vastuvõtab ning tagastab (UART kirjeldus). Erinevate töörežiimide kirjeldus. Kuidas Monitoorida (Real Term)

## BNO055 kiirendus anduri seadistamine

Mis PIN-id millega ühendada, ning miks. (nt et PIN1 ja PIN0-st sõltub, mis protokolli kasutada). Mis on initsialiseerimis seaded, mis info saadetakse.

# Programm

Üleüldine kirjeldus, koos skeemiga. Mida tehakse vigaste andmetega, mida tehakse vigade korral. Aja arvestus.

## Programmi funktsioonide kirjeldus

Selgitan igafunktsiooni ülesehitust ning miks seda kasutatakse.

# Kokkuvõte

Kasutatud kirjandus

**There are no sources in the current document.**

Lisa 1 –

1. https://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php [↑](#endnote-ref-1)