

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2021

Joni Tuomisto

ÄLYKÄS LOKA-AUTON SÄILIÖN MITTAUSANTURI

Joni Tuomisto

ÄLYKÄS LOKA-AUTON SÄILIÖN MITTAUSANTURI

Esineiden internetin yleistyessä jokainen ala tulee parantamaan palveluitaan ja tapojaan suorittaa tehtäviään, eikä jätehuoltokaan ole poikkeus. Tämän opinnäytetyön tavoitteena oli parantaa jo olemassa olevaa ratkaisua, loka-autojen säiliöiden mittausta. Nykyinen ratkaisu on mekaaninen mittari, joka toimii säiliön sisäisen kellukkeen avulla. Tämän kellukkeen noustessa se kääntää akselilla säiliön ulkopuolella olevan mittarin nuolta ja osoittaa kohti arvioitua nesteiden määrää.

Tavoitteena oli kehittää ESP32-kehitysalustasta ja MinIMU-9 v5 -anturimoduulin kiihtyvyyssanturista herkkä kulma-anturi loka-auton säiliön kylkeen, missä mekaanisen mittarin viisari osoittaa tankin tilavuuden tilaa. Laite mittaa tarkasti mekaanisen mittarin viisarin liikkeen asteissa ja laskee kulman perusteella tankin sisäisen nesteiden määrän litroissa. Laskujen jälkeen laite on valmis lähettämään lasketun datan eteenpäin jatkokehityksessä kehitettävään puhelinsovellukseen Bluetooth-yhteyden välityksellä.

Laite kehitettiin Actmea Oy:lle yrityksen määriteltyjen vaatimusten mukaisesti. Opinnäytetyöhön kuului alustan valitseminen ja suunnittelu, laitteen ohjelmointi ja logiikan määrittely C++ -ohjelmointikieltä käyttäen sekä Bluetooth-yhteyden valmisteleminen, jonka välityksellä laskettu data saataisiin puhelimelle, joka puolestaan välittäisi datan Actmean AMCare tietokantaan.

Opinnäytetyön onnistunut toteutus mahdollistaa jatkokehityksen laitteeseen paritettavan puhelinsovelluksen parissa ja tämän jälkeen kenttätestauksien merkeissä. Näiden edistysten avulla voidaan optimoida lokasäiliöiden tyhjennysreitit sekä samalla pienentää CO₂-päästöjä lyhyempien ajoreittien ansiosta.

ASIASANAT:

ESP32, C++, Arduino IDE, IoT

Joni Tuomisto

INTELLIGENT SEPTIC TRUCK'S TANK GAUGE

As the Internet of Things becomes more widespread, each area of industry starts to improve their services and ways of performing their tasks. Waste management is no exception in this trend. The aim of this thesis was to improve an already existing waste management solution, the level indicator of a septic truck's tank. The current solution is a float gauge. When the float rises, it rotates the indicator outside of the tank to the estimated amount of liquid inside.

The objective of this thesis was to develop a sensitive tilt sensor using an ESP32 development board with a MinIMU-9 v5 sensor module's acceleration sensor which would be placed on the mechanical gauge's indicator which indicates the current volume of the tank by the side of the truck's tank. The device would precisely measure the indicators angle and movement with which it would calculate the volume of the liquid inside the tank in litres. After calculations, the device would be ready to transfer these values via Bluetooth to a mobile application developed in the future.

The device was developed for Actmea Ltd according to the requirements defined by the commissioning company. This thesis focuses on the selection and design of the platform, the programming of the device and defining the logic using C++ programming language and preparing the device for Bluetooth communication with the application with which the calculated data will be send to Actmea's AMCare database.

The implementation of the thesis was successful according to the requirements and the result allows for future development with the mobile application that would be paired to the device and after that the development can be continued to field testing. With these advancements we can monitor the status of the tanks, optimise the drive routes of the tank trucks for shorter routes and with that lower the CO2 emissions.

KEYWORDS:

ESP32, C++, Arduino IDE, IoT

SISÄLTÖ

KÄYTETYT LYHENTEET	7
1 JOHDANTO	8
2 TEKNOLOGIA	9
2.1 Internet of Things	9
2.2 Kehitysalustat	11
2.3 Anturit ja toimilaitteet	13
2.4 Bluetooth Low Energy	13
3 VAATIMUKSET	15
4 ARKKITEHTUURI	17
4.1 Kehitysalusta	17
4.2 Kiihtyvyyssanturi	18
4.3 PlatformIO	19
5 TOTEUTUS	21
5.1 Laskenta-algoritmit	21
5.2 Ohjelmisto	24
5.3 Mikrokontrolleri	26
5.4 Gyroskooppi ja AHRS	28
6 TESTAUS	31
6.1 Kulman tarkkuus	31
6.2 Tankkien tilavuus	33
7 YHTEENVETO	38
8 JATKOKEHITYS	40
LÄHTEET	42

KAAVAT

Kiihtyvyyssarvojen suodatus	21
X-akselin asteet	21
Y-akselin asteet	21
Säiliön tilavuus	23
Pinnan korkeus säiliön katosta	23
Säiliön säde	23
Nesteen määrä	23

KUVAT

Kuva 1. Esimerkki IoT-järjestelmän tiedon kulusta.	10
Kuva 2. ESP32-WROOM-32D-kehitysalusta	18
Kuva 3. MinIMU-9 v5 -anturimoduuli.	19
Kuva 4. Säiliön ulkomitat	22
Kuva 5. Säiliön sisämitat	23
Kuva 6. Ohjelman vuokaavio	25
Kuva 7. Laitteen kytkentäkaavio.	26
Kuva 8. Esimerkkikytkentä kytkentäalustalle.	27
Kuva 9. Säiliön mekaaninen mittari. Kuva: Sami Jalonen.	28
Kuva 10. LIS3MDL-gyroskoopin kalibrointi-arvot sarjamonitorissa.	29
Kuva 11. LIS3MDL-gyroskoopin kalibrointi visualisoitu Jupyter-ohjelman kautta	29
Kuva 12. LIS3MDL-anturin kompassimagneetin kalibrointi-arvoja sarjamonitorissa	30
Kuva 13. Anturin tarkkuuden testaus painovoimaa vastaan.	31
Kuva 14. Suorakulmainen tasakylkinen kolmio anturin tarkkuuden mittausta varten.	32
Kuva 15. Sarjamonitorin tulostus suorakulmion hypotenuusaa vasten.	33
Kuva 16. Sarjamonitorin tulostus suorakulmion kateettia vasten.	33
Kuva 17. Testitankki 1:n mittaustulokset.	34
Kuva 18. Testitankki 2:n mittaustulokset.	35
Kuva 19. Testitankki 3:n mittaustulokset.	36
Kuva 20. Esimerkki BLE-lähetys näytettynä sarjamonitorissa.	38
Kuva 21. Kuvankaappaus samasta lähetyksestä kuin kuvassa 17 nähtynä nRF-sovelluksessa.	39

TAULUKOT

Taulukko 1. Kehitysalustojen vertailua.	12
Taulukko 2. Bluetooth versioiden vertailua	14
Taulukko 3. Vaatimukset MoSCoW-menetelmällä	16

Taulukko 4. Testitankki 1:n tarkat mittausluvut.	34
Taulukko 5. Testitankki 2:n tarkat mittausluvut.	35
Taulukko 6. Testitankki 3:n tarkat mittausluvut.	36

KÄYTETYT LYHENTEET

8DPSK	Vaihesiirtokoodaus modulaatio menetelmä (8 Phase Differential Phase Shift Keying).
BLE	Langaton lähiverkkotekniikka (Bluetooth Low Energy).
CPU	Suoritin (Central Processing Unit).
DQPSK	Vaihesiirtokoodaukseen pohjautuva modulaatiomenetelmä (Differential Quaternary Phase Shift Keying).
EDR	Bluetoothin parannettu datanopeuslaajennus (Enhanced Data Rate).
ESP32	Espressif Systemsin valmistama kehitysalusta.
FHSS	Taajuushyppely (Frequency-Hopping Spread Spectrum).
FLASH	Uudelleenohjelmoitava puolijohdemuisti.
GFSK	Taajuussiirtokoodaukseen pohjautuva modulaatiomenetelmä (Gaussian Frequency-Shift Keying).
GPIO	Yleiskäyttöinen sisäänmeno tai ulostulo (General Purpose Input/Output).
LoRaWAN	Pitkän kantaman radioverkko (Long Range Wide Area Network).
IDE	Ohjelmointiympäristö (Integrated Development Environment).
IMU	Inertian mittaussyksikkö (Inertia Measurement Unit).
IoT	Esineiden internet (Internet of Things) – Laitteita, joita voi hallita tai seurata verkkoyhteyden avulla.
ISM	ISM-taajuusalue (Industrial, Scientific and Medical) – Radio-taajuudet tarkoitettu teolliseen, tieteelliseen ja lääketieteelliseen käyttöön.
ROM	Lukumuisti (Read-Only Memory).
RTC	Reaaliaikainen piiriin integroitu kello (Real Time Clock).
SCL	I ² C-väylän sarjamuotoinen kellolinja (Serial Clock Line).
SDA	I ² C-väylän sarjamuotoinen datalinja (Serial Data Line).
SRAM	Staatinnen hajasaantimuisti (Static Random-Access Memory).

1 JOHDANTO

Esineiden internetin yleistyessä jokainen ala tähtää parantamaan ja modernisoimaan palveluitaan ja tapojaan suoriutua tehtävistään, eikä jätehuoltokaan ole poikkeus. On jo olemassa lukuisia ratkaisuja tehostamassa jätehuoltoa, kuten jäteastioiden pinnanmittausanturit [1]. Tämänkin työn tavoitteena on parantaa olemassa olevaa jätehuoltoratkaisua.

Tämän opinnäytetyön tavoitteena on rakentaa älykäs kulma-anturi, joka kykenee mittaamaan loka-auton säiliön nesteen määrää vertaamalla mekaanisen pintakorkeusmittarin viisarin lähtö- ja lopetuskulmaa ja laskemalla tämän muutoksen perusteella säiliön sisäisen nesteen määrän litroissa. Viisari liikkuu, kun säiliön sisällä oleva kelluke nousee ja kääntää akselia, joka on kiinnitetty viisariin. Mittaus tapahtuu säiliötä täytettäessä. Laitteessa on valmius Bluetooth-yhteyteen tätä varten rakennettavaa puhelinsovellusta varten, josta data tulee liikkumaan tietokantaan. Sovelluksen kautta tullaan myös säätämään säiliökohtaiset asetukset laitteelle sekä näkemään pinnan muutos reaaliajassa.

Älykkään laitteen ansiosta tulevaisuudessa voidaan seurata säiliöiden pintaa ActMea Oy:n AMCare-tietokannan kautta ja voidaan minimoida turhat ajoreitit ja täten vähentää hiilidioksidipäästöjä ja vaadittavaa aikaa tyhjennysajoihin. Laite tulee olemaan osa suurempaa jätehuoltokokonaisuutta, jossa tullaan käyttämään hyväksi useampaa IoT-ratkaisua.

Ratkaisu toteutetaan ESP32-WROOM-32D-alustalle käyttäen MinIMU-9 v5 -anturimoduulin kiihtyvyyssanturia. Laitteeseen tulee fyysinen painike, jota painamalla laite aktivoituu ja käy läpi esiasetetut vaiheet läpi, joissa mittaus, laskeminen ja tiedon välitys tapahtuvat vuorollaan.

2 TEKNOLOGIA

Tässä luvussa käydään läpi käytetyt teknologiat ja tärkeimmät käsitteet.

2.1 Internet of Things

Internet of Things (IoT), eli esineiden internetillä ei ole yhtä tiettyä selvää määritelmää. Sitä usein ajatellaan laitteina, joilla on verkkoyhteys ja ovat sulautettuna fyysisessä ympäristössä parantaen jo olemassa olevaa ratkaisua tai antaen mahdollisuuden uuteen ratkaisuun, joka ei ollut aikaisemmin mahdollinen. Nämä laitteet, tai esineet, keräävät informaatiota anturien avulla ympäristöstään tai antavat muiden järjestelmien luoda yhteyden niihin ja täten vaikuttavat ympäristöönsä erilaisten toimilaitteiden avulla [2].

Eri toimialoilla on useita eri käyttötapoja ja tilaisuuksia hyödyntää IoT ratkaisuja. IoT projektit tuovat lisäulottuvuuksia lisäämään niiden monimutkaisuutta verrattuna muihin pilvikeskeisiin sovelluksiin, mukaan lukien:

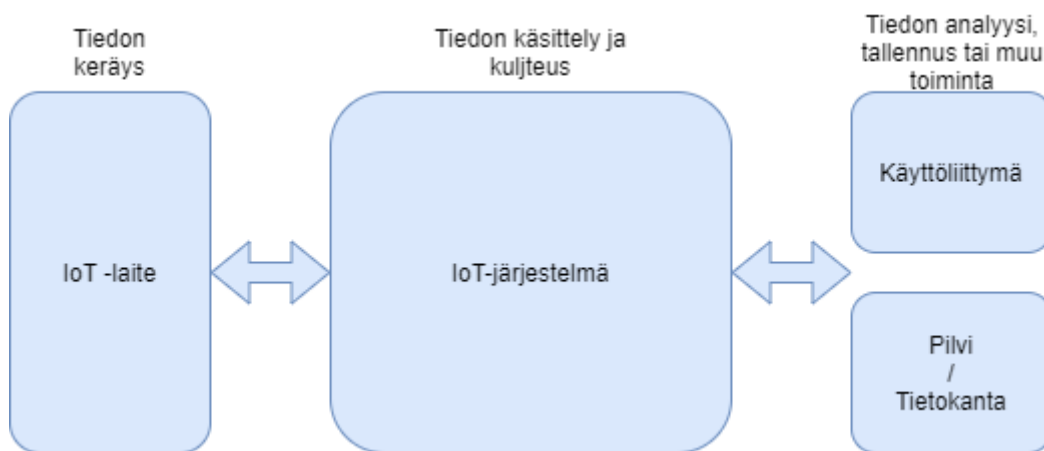
- Monipuolinen laitteisto
- Monipuoliset käyttöjärjestelmät ja ohjelmat laitteilla
- Erilaiset verkon yhdyskäytävän vaatimukset [2]

IoT on yleistynyt räjähdysmäisesti viimeisen kymmenen vuoden aikana koko maailmassa ja myös Suomessa. Nykyään tähän kategoriaan kuuluvia laitteita löytyy todella useasta kotitaloudestakin. Vuonna 2020 tehdystä kuluttajatutkimuksesta, johon vastasi 2000 ihmistä ympäri suomea jokaisesta ikäluokasta, 64 % vastanneista kertoi omistavansa yhden tai useamman internettiin kytketyn laitteen, poissulkien tietokoneet ja puhelimet. Esimerkiksi 48 % vastanneista kertoi omistavansa internetyhteydellä varustetun äly-tv, ääni-, media- tai videotoistolaitteen [3].

Yrityksetkin ovat alkaneet hyödyntää IoT-laitteita kiihtyvällä suosiolla. Tilastokeskuksen vuonna 2020 toteutetussa tutkimuksessa selvisi, että esineiden internet on käytössä keskimäärin 40 %:lla suomalaisista yrityksistä. Tutkimus toteutettiin yrityksillä, jotka työllistivät vähintään 10 ihmistä. Suurilla vähintään 100 ihmistä työllistävillä yrityksillä käyttö on 52 % yrityksistä ja pienimmistä 10–19 henkilöä työllistävillä käyttö on 34 %. Yleisin tapa hyödyntää esineiden internetiä on toimitilojen turvallisuuden valvonnassa, joka kattaa 31 % yrityksistä, joissa esineiden internetiä hyödynnetään [4].

IoT-järjestelmien toiminta

IoT-järjestelmät toimivat kommunikoiden eri laitteiden, järjestelmien tai ohjelmien kanssa erilaisten verkkojen välityksellä. Kommunikaatio joko molemminpuolinen tai vain yksisuuntainen, laite mittaa sensoreilla ympäristöä ja lähettää mitatut arvot tai arvojen perusteella toimitettujen laskujen tulosten arvot eteenpäin, tai vaikuttaa ympäristöönsä toimilaitteilla. Laite siirtää mitatut tulokset eteenpäin verkon välityksellä tai saa verkosta ohjeet toimia toimilaitteillaan. (Kuva 1.)



Kuva 1. Esimerkki IoT-järjestelmän tiedon kulusta.

Yhdistetyt "Smart Home" -ratkaisut ovat hyvä esimerkki IoT-järjestelmistä toiminnassa. Internet-yhdistetyt termostaatit, ovikellot, palovaroittimet ja turvahälyttimet luovat kattavan verkkokeskuksen, josta tieto jaetaan fyysisten laitteiden välillä ja käyttäjät voivat etänä hallita näitä "esineitä" tämän verkkokeskuksen välityksellä. Kuten säätää lämpötilaa tai avata lukittuja ovia puhelin sovelluksen tai nettisivun kautta.

Kotiratkaisut eivät ole ainoita IoT-järjestelmiä yleisessä käytössä. Niitä löytyy useissa laitteissa, teollisuuksissa ja ympäristöissä. Luokahuoneiden älykkäistä liitutauluista lääketieteellisiin laitteisiin, jotka havaitsevat merkkejä Parkinsonin taudista. IoT tekee maailmasta nopeasti älykkäämpää yhdistämällä fyysisen ja digitaalisen [5].

2.2 Kehitysalustat

Esineiden internetin kehityksen erittäin tärkeä vaihe on kehitysalustan avulla toteutettu testaus- ja prototyypivaihe, jonka aikana voidaan toteuttaa ja suunnitella keskeisiä toimintoja halutulle IoT-järjestelmälle. Kehitysalustoja on useita erilaisia useisiin eri tarkoituksiin. Jotkin alustat tulevat omilla integroiduilla moduuleillaan tarkoitettu tietyntylaisiin tehtäviin ja toimintoihin, mutta lähes mitä vain alustaa voi käyttää melkein mihin tahansa tehtävään, kunhan alustan teho ja muisti sallivat kyseisen toiminnon. Kehitysalustoja on tarjolla useita kappaleita valmiilla kokoonpanoilla eri tarkoituksiin, mutta asiakas voi myös tilata omiin tarkoituksiinsa halutun piirilevyn juuri heidän toivomillaan mitoilla sekä komponenteilla käyttäen pohjana jotakin olemassa olevaa kehitysalustaa, kuten ESP32.

Tässä opinnäytetyössä harkittiin muutamaa sopivaa vaihtoehtoa työn toteutukseen (Taulukko 1). Tärkeimpiä kriteerejä oikean alustan valitsemiseen olivat laitteen teho ja muisti sekä tähän projektiin, sekä jatkokehitystä ajatellen.

	ESP32-WROOM-32D [6]	Arduino Mega [7]	Arduino Uno [8]
Mikrokontrolleri	Xtensa® LX6	ATmega2560	ATmega328P
Flash muisti	4 MB	256 kB	32 kB
SRAM	520 kB	8 kB	2 kB
ROM	448 kB	4 kB	1 kB
Kellotaajuus	240 MHz	16 MHz	16 MHz
GPIO määrä	34	70	20
Pituus	55 mm	101.52 mm	68.6 mm
Leveys	25 mm	53.3 mm	53.4 mm
Wi-Fi	✓	x	x
Bluetooth	✓	x	x

Taulukko 1. Kehitysalustojen vertailua.

2.3 Anturit ja toimilaitteet

Anturit ja toimilaitteet ovat IoT-järjestelmien selkäranka, niiden avulla kehitysalustat voivat mitata ja seurata ympäristöään sekä vaikuttaa ympäristöönsä. Erilaisia ratkaisuja on useita ja löytyy digitaalisia sekä analogisia ratkaisuja. Tähän opinnäytetyöhön tarvitimme vain kiihtyvyyssanturia ja mahdollisia saatavilla olevia vaihtoehtoja oli vähän. Haluamaamme tarkoitukseen olisi riittänyt yksinkertainen kolmiakselinen kiihtyvyyssanturi, mutta nopeasti saatavilla oli vain IMU-anturimoduuli, joka sisälsi kiihtyvyyssanturin lisäksi myös gyroskoopin sekä magnetometrin. Kyseisiä moduuleja yleensä käytetään lentotietokoneissa osana inertiasuunnistusta. Moduulin täyttä potentiaalia pyrittiin aluksi käyttämään, mutta lopulta kävi ilmi, ettei tämä ollut hyödyllistä laitteelle tarkoitettuun käyttötarkoitukseen ja moduulin kiihtyvyyssanturi oli riittävä.

2.4 Bluetooth Low Energy

Laite kommunikoi puhelimelle Bluetooth-yhteyden välityksellä ja tämä pystyttiin toteuttamaan joko Bluetooth Classic- tai Bluetooth Low Energy-yhteyttä käyttäen. BLE on uudempi versio Bluetoothista, joka toimii huomattavasti pienemmällä kulutuksella, mutta harvemmalla kanavalla. Koska virrankulutuksen minimointi oli yksi tavoitteista, päädyttiin käyttämään BLE:tä, mutta vertailtiin myös sen suorituskykyä Bluetooth Classiciin. (Taulukko 2.)

	Bluetooth Low Energy [9]	Bluetooth Classic [9]
Taajuusalue	2.4 GHz ISM-taajuusalue	2.4GHz ISM-taajuusalue
Kanavat	40 kanavaa 2 MHz:n välityksellä	79 kanavaa 1 MHz:n välityksellä
Kanavan käyttö	FHSS	FHSS
Modulointi	GFSK	GFSK, $\pi/4$ DQPSK, 8DPSK
Virrankulutus	~0.01–0.5 kertainen vertausarvosta (riippuu käytöstapauksesta)	1 (vertausarvo)
Datanopeus	2 Mb/s 1 Mb/s 500 kb/s 125 kb/s	3 Mb/s 2 Mb/s 1 Mb/s
Max Tx Teho	Luokka 1: 100 mW (+20 dBm) Luokka 1.5: 10 mW (+10 dBm) Luokka 2: 2.5 mW (+4 dBm) Luokka 3: 1 mW (0 dBm)	Luokka 1: 100 mW (+20 dBm) Luokka 2: 2.5 mW (+4 dBm) Luokka 3: 1 mW (0 dBm)
Verkon topologiat	Point-to-Point (mukaan lukien piconet) Broadcast Mesh	Point-to-Point (mukaan lukien piconet)

Taulukko 2. Bluetooth versioiden vertailua

3 VAATIMUKSET

Tässä kappaleessa käydään läpi työlle asetetut vaatimukset. Vaatimusten avulla rajataan ja määritellään työ sekä ohjataan oikeisiin kompromisseihin.

Laite ei tule käyttämään LoRaWAN-yhteyttä, koska tällä tavoin käyttäjä ei tule saamaan dataa näkyviin itselleen. Bluetooth-yhteyden avulla data saadaan suoraan puhelimeen ja täten näkyviin käyttäjälle, sekä lähetettyä tietokantaan puhelimen oman verkkoyhteyden välityksellä.

Vaatimukset listattuna Taulukko 3 seuraavanlaisesti kategorioihin:

- Must – Täytyy olla
- Should – Pitäisi olla
- Could – Voisi olla
- Won't – Ei tule olemaan

Vaatus	Kuvaus	Prioriteetti
Säiliön pinnan mittaus	Laitteen tulee laskea säiliön nesteen määrä mekaanisen pintamittarin viisarin asteen perusteella	Must
Bluetooth yhteys	Laitteella tulee olla valmius lähettää lasketut arvot Bluetooth yhteyden välityksellä	Must
Mahdollisuus useaan säiliöasetukseen	Koodissa otettava huomioon mahdollisuus asettaa säiliökohtaiset parametrit, jotta laitetta voidaan käyttää erikokoisissa säiliöissä	Must
Paikannusmahdollisuus	Laitteeseen asetetaan GPS paikannus	Could
Fyysinen näyttö	Käyttäjä voi nähdä laitteesta suoraan kulman reaaliajassa sekä säiliön tilavuuden. Tämä toteutetaan puhelimen avulla, eikä pieni näyttö mittarissa olisi käytännöllinen.	Won't
LoRaWAN yhteys	Laitte kommunikoi LoRaWAN radion avulla. Kommunikaatio toteutetaan Bluetooth yhteyden välityksellä puhelinsovellukseen.	Won't

Taulukko 3. Vaatimukset MoSCoW-menetelmällä

4 ARKKITEHTUURI

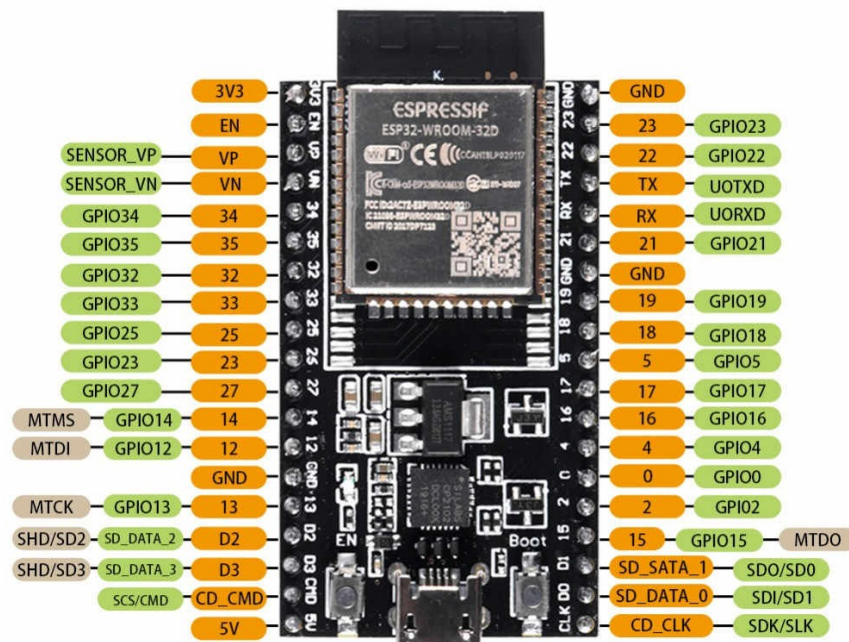
Tässä luvussa käydään läpi opinnäytetyön arkkitehtuurivalintoja suhteessa laitteelle asetettuihin vaatimuksiin. Valitaan kriittiset komponentit.

4.1 Kehitysalusta

Kehitysalusta valittiin laajan tuen, tehokkuuden sekä mahdollisen jatkokehityksen helpottamisen kannalta. Vaihtoehtoja oli erilaiset Arduino-alustat sekä ESP-pohjaiset alustat, joihin saataisiin tarvittavat kirjastot sekä moduulit. Jo alkuvaiheessa kallistuttiin ESP-alustoihin helpon jatkokehityksen kannalta, sillä varmistettaisiin, ettei tulisi muissa mahdollisissa kehitystarkoituksissa teho- tai muistipuutteita.

Tässä opinnäytetyössä käytettiin ESP32-WROOM-32D-mallia. Tämä kyseinen alusta valittiin ESP32:n tehokkaan prosessorin, riittävän FLASH-muistin sekä sisäänrakennetun Bluetooth-moduulin takia, jolla saataisiin yhteys älypuhelimeen ja sen kautta data näkyviin käyttäjälle ja välitettyä tietokantaan.

ESP32-WROOM-32D on Espressif Systems -yhtiön valmistama kehitysalusta, jossa on integroidut Wi-Fi- ja Bluetooth-komponentit. Alustassa on kaksiytiminen Xtensa® 32-bit LX6 -mikroprosessori, jonka kellotaajuutta voi säätää 80 MHz:n ja 240 MHz:n välillä. Mikroprosessorissa on 38 porttia, joista 34 on käyttäjän määriteltävissä, näitä portteja kutsutaan GPIO-porteiksi (Kuva 2) [10]. Näitä portteja voidaan käyttää vastaanottamaan tai lähettämään signaaleja. Alustan sisäiset muistit ovat 448 kB-ROM, 520 kB-SRAM sekä 16 kB RTC SRAM -muistia, josta 8 kB on nopeaa ja 8 kB hidasta RTC-muistia. Alustassa on myös integroitu 4 MB:n FLASH-muisti, johon tallennetaan ajettava ohjelma [6].



ESP32-WROOM-32D

Kuva 2. ESP32-WROOM-32D-kehitysalusta

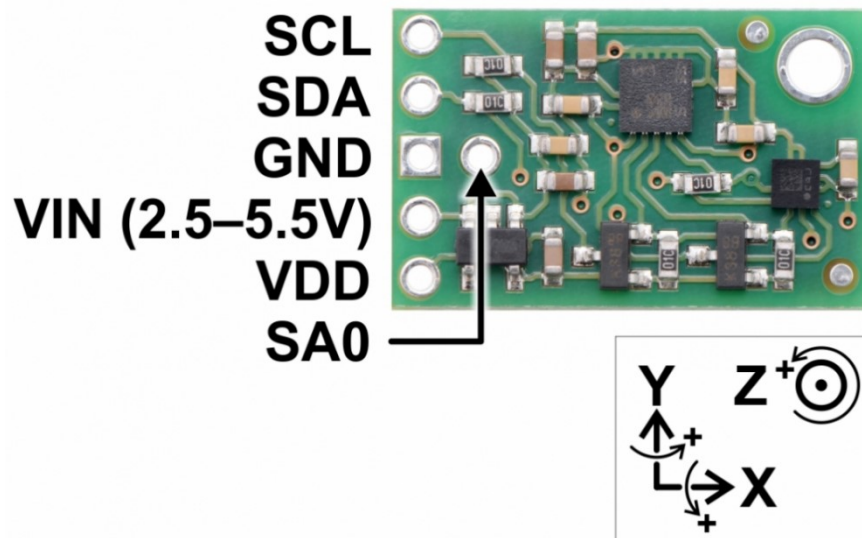
4.2 Kiihtyvyyssanturi

Vaadittavat anturilukemat saavutettaisiin käyttämällä Pololu-yhtiön valmistamaa MinIMU-9 v5 -anturimoduulia. Tämä moduuli sisältää kolme eri anturia: Gyroskoopin, magneettikompassin sekä kiihtyvyyssanturin, joista pelkästään viimeistä tarvitaan tässä työssä, joka selvisi pitkän IMU järjestelmätestailun jälkeen. Tämä käsitellään tarkemmin 5. kappaleessa. Anturit toimivat yhden I²C-yhteyden välityksellä, tämä myös helpotti testialustan suunnittelua.

Laitteen vaatimusten valossa jokin yksinkertaisempi anturialusta olisi parempi vaihtoehto, sillä tämän anturialustan ylimääräiset anturit tuovat lisähintaa laitteelle ja kustannusten minimoimiseksi olisi hyvä hankkia yksittäinen kiihtyvyyssanturi usean anturin IMU-anturimoduulin sijasta. Kyseiseen anturimoduuliin päädyttiin kuitenkin sen helpon saatavuuden takia. Testausvaiheessa selvisi kuitenkin anturin olevan erittäin tarkka ja täten

riittävän sopiva tarkoitukseensa. Jatkokehityksen aikana voi etsiä sopivamman, halvemmän kiihtyvyyssanturin tähän tarkoitukseen.

Pololun valmistama MinIMU-9 v5 -anturimoduuli koostuu kolmesta eri anturista, LIS3MDL-gyroskooppi ja kompassi sekä LSM6DS33-kiihtyvyyssanturi. Tässä opinnäytetyössä näistä käytettiin vain kiihtyvyyssanturia, koska alkuperäinen suunnitelma luoda AHRS-järjestelmä todentui liian monimutkaiseksi tähän tarkoitukseen. Anturimoduuli toimii I²C-yhteydellä ESP32-kehitysalustaan, joka toimii vain kahden portin yhteydellä, SDA- ja SCL-portteihin. (Kuva 3.) [11].



Kuva 3. MinIMU-9 v5 -anturimoduuli.

Tässä työssä käytettiin hyväksi X-akselia. Laite on siis asennettava siten, että anturi on oikein päin. Koodissa on myös valmius käyttää Y-akselia, muutaman muuttujan vaihdon jälkeen.

4.3 PlatformIO

PlatformIO on alustariippumaton laajennus Microsoftin Visual Studio Code IDE:en, sen avulla voidaan ohjelmoida useita erilaisia sulautetun järjestelmän alustoja, kuten tässä työssä ESP32 pohjaista alustaa [12].

PlatformIO toimii avoimen lähteen koodikirjastoilla, joita on tuhansia. Esimerkiksi kaikki Arduinon ja ESP:n kirjastot löytyvät suoraan kirjastohausta. Kirjastohaku on erittäin helpokäyttöinen ja tehokas työkalu. Sen avulla voi etsiä hakusanoilla tai suoraan kirjaston

nimellä tarvittavia koodikirjastoja omaan projektiin. PlatformIO:n avulla tuotiin Arduinon IDE:n ja MinIMU-9 v5 käyttämät koodikirjastot tähän projektiin sekä käytetyn alustan tiedot, jotta pystyttiin kääntämään käytetty koodi ESP32 alustalle käyttäen Visual Studio Code IDE:tä.

5 TOTEUTUS

Tässä luvussa käydään läpi työn toteutuksen eri vaiheet, vaadittavat laskut ja ohjelmiston logiikka.

5.1 Laskenta-algoritmit

Laitteen tärkein tarkoitus, eli säiliön tilavuuden laskeminen, vaatii tiettyjä laskukaavoja toimiakseen. Tässä alakappaleessa käydään läpi tarvittavat kaavat.

Kulma lasketaan painovoiman avulla, laskemalla jokaisen akselin kiihtyvyyssvektorien välinen kulma voidaan päätellä mikä suunta on alaspäin ja mikä laitteen kulma on tähän suuntaan verrattuna. Kulman laskeminen aloitetaan lukemalla ja suodattamalla kiihtyvyyssanturin kaikkien kolmen akselin raa'at kiihtyvyyssarvot. Suodattamalla arvot **Error! Reference source not found.** olevalla kaavalla, niistä saadaan luotettavimmat eikä ilmene epätarkkaa kohinaa luetuissa arvoissa. Kaavan k-arvo määrittää suodattimen tarkkuuden, mutta mitä tarkempi arvo on, niin sitä hitaammaksi laskenta muuttuu. Optimi luku saatiin säätämällä sitä pikkuhiljaa tarkemmaksi, kunnes saatiin mahdollisimman tarkka suodatin samalla kun vielä pidettiin toiminta nopeus tarpeeksi nopeana. Arvoksi päättyi 0.969.

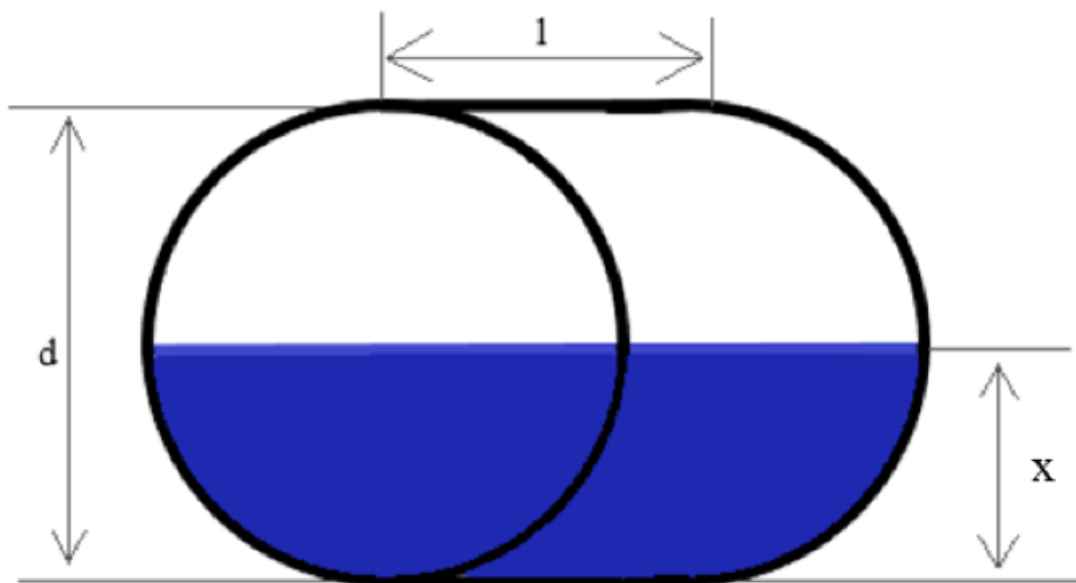
$$xFiltered = k * xFiltered + (1.0 - k) \cdot accX \quad (1)$$

Seuraavaksi suodatetuista kiihtyvyyssarvoista alla lasketaan X- ja Y-akselit asteissa. Z-akselia ei lasketa, koska laitteen käyttökulman ansiosta sitä ei tulla tarvitsemaan.

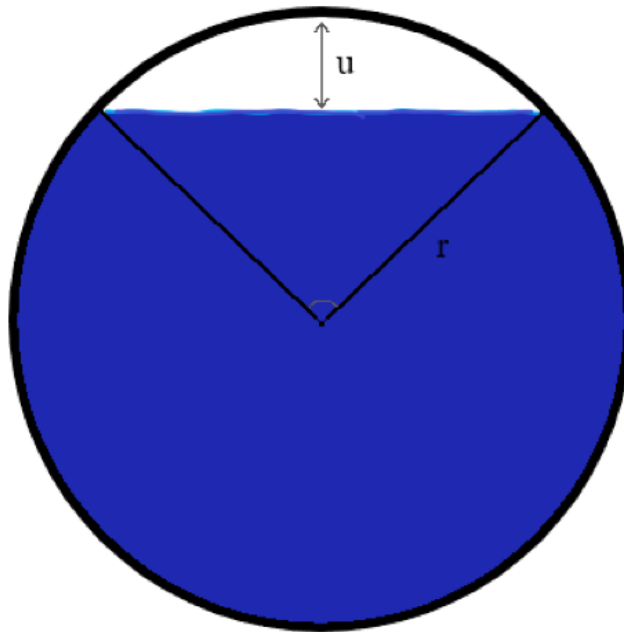
$$angleX = \arctan(yFiltered/zFiltered) \cdot 57.3 + 180 \quad (2)$$

$$\text{angle}Y = \arctan\left(\frac{-x_{Filtered}}{\sqrt{y_{Filtered}^2 + z_{Filtered}^2}}\right) \cdot 57.3 + 180 \quad (3)$$

Tilavuuden laskemiseksi laite tarvitsee aluksi käyttäjältä säiliön mitat (Kuva 4). Pituus (l) ja halkaisija (d) metreissä sekä mekaanisen mittarin viisarin alku- ja loppukulmat asteissa. Laitteen mittaaman kulman perusteella lasketaan säiliön sisäisen nesteen pinnan etäisyys säiliön katosta metreinä (u) (Kuva 5). Mekaaninen mittari toimii siten, että kun säiliön sisällä oleva kelluke nousee pinnan mukana, se kääntää akselia, joka puolestaan kääntää ulkona olevaa viisaria osoittaen arvioitua tilavuutta. Tietämällä tämän viisarin sektorin kulma voidaan laskea sen suhde pinnan korkeuteen ja täten laskea säiliön tilavuus.



Kuva 4. Säiliön ulkomitat



Kuva 5. Säiliön sisämitat

Aluksi alla lasketaan säiliön tilavuus kokonaisuudessaan, jota käytetään laskemaan täytetty tilavuus. Tätä varten määritellään myös pinnan korkeus vähentämällä mitattu pinnan etäisyys säiliön katosta (u) säiliön halkaisijasta (d)

$$V_1 = \pi r^2 \cdot l \quad (4)$$

$$x = d - u \quad (5)$$

$$r = d/2 \quad (6)$$

Viimeiseksi alla lasketaan itse nesteen tilavuus käyttäen edellälaskettuja arvoja.

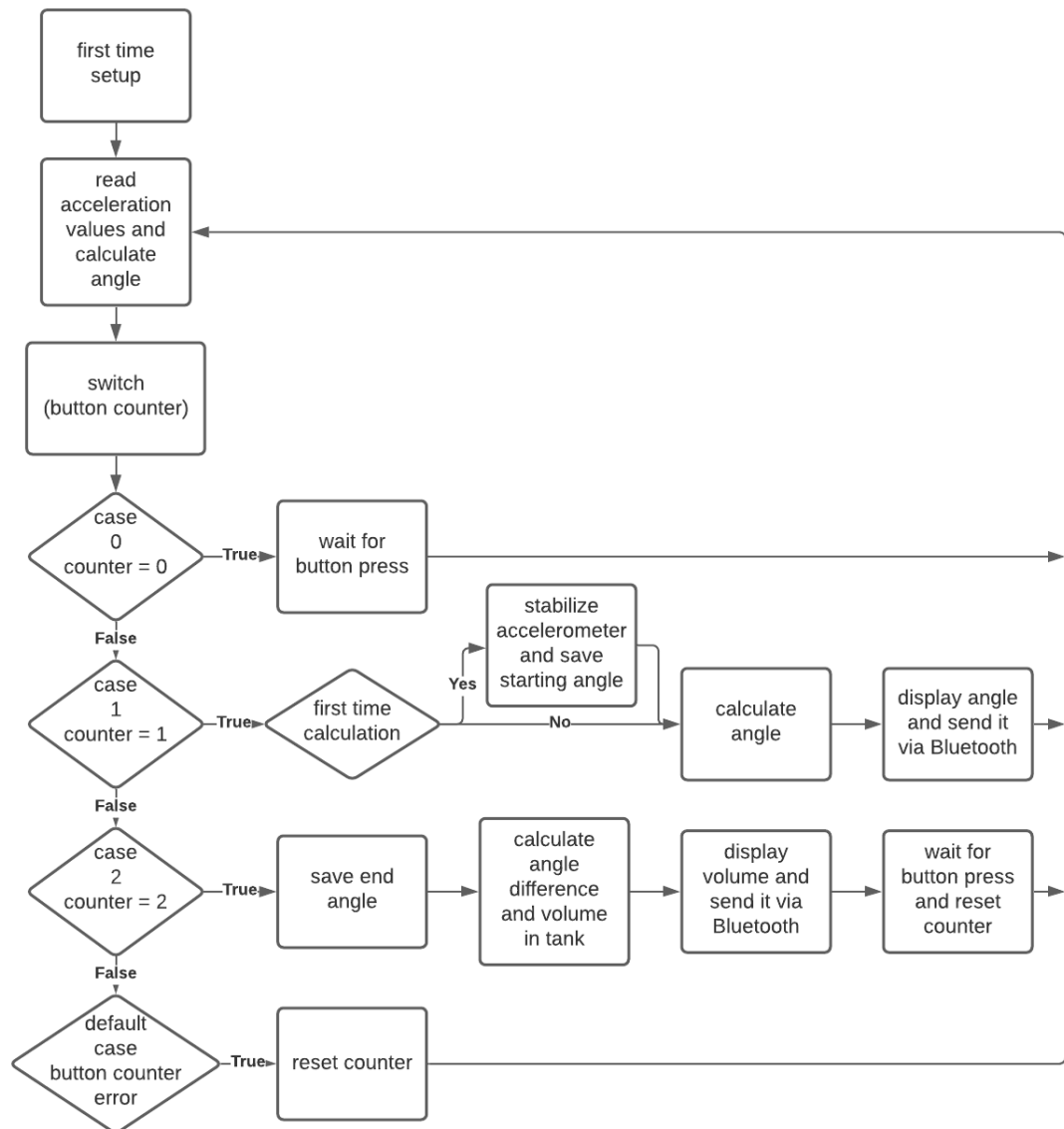
$$\begin{aligned} \text{part 1} &= V_1 / (\pi r^2) \\ \text{part 2} &= r^2 \cdot \arccos((r - x)/r) \\ \text{part 3} &= (r - x) \cdot \sqrt{r^2 - (r - x)^2} \\ V_2 &= \text{part 1} \cdot (\text{part 2} - \text{part 3}) \end{aligned} \quad (7)$$

5.2 Ohjelmisto

Laitteen toiminnan puolesta projektin tärkein osa-alue oli kehitysalustan ohjelmointi. Laitteen ohjelmointi tapahtui Visual Code -ohjelmointiympäristössä, johon oli asennettu PlatformIO IDE, tämä laajennus antoi mahdollisuuden käyttää Arduino IDE:n toimintoja ja koodikirjastoja. Täten ESP32:n ohjelmointi tapahtui kuin Arduinoa ohjelmoisi, kun saatiin kaikki tähän kuuluvat koodikirjastot mukaan. Ohjelmointi tapahtui käyttäen C++ ohjelmointikieltä.

Laitteen oma logiikka toteutettiin käyttäen switch...case -rakennetta, joka riippui ulkoisesta painikkeesta ja käyttäjän siihen syöttämistä painalluksista. Kun laitteen käynnistää ensimmäisen kerran niin se suorittaa ensiasennuksen, jonka aikana laite asettaa vaadittavat oletusarvot ja aloittaa mainostamaan itseään Bluetoothin välityksellä. Tämän jälkeen laite jää lepotilaan odottamaan käyttäjää samalla kun laite lukee kiihtyvyyssarvoja sensorista ja laskee näistä kulman, jonka laite tulostaa aktiivisesti näytölleen.

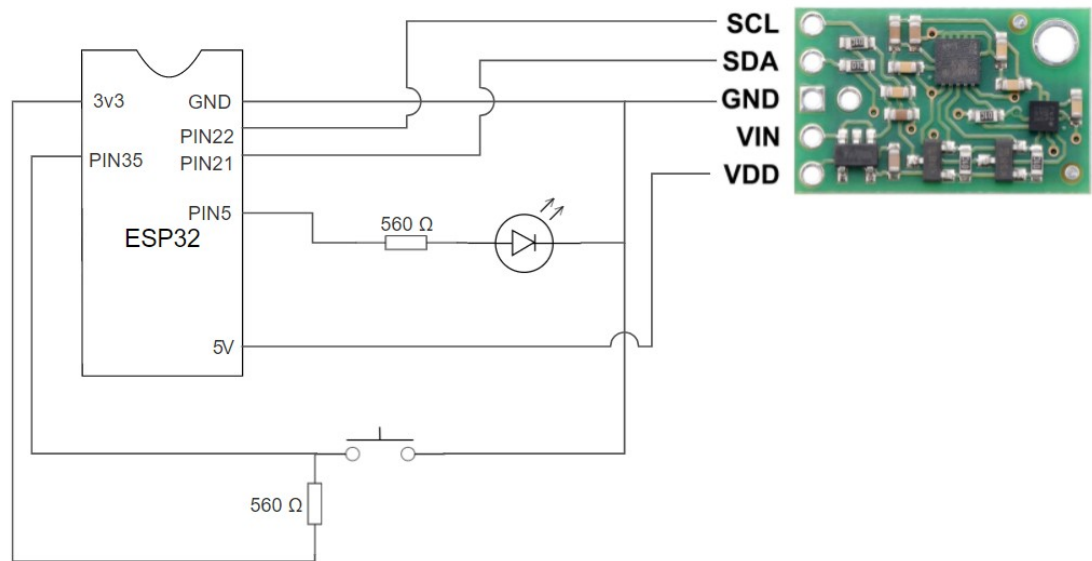
Ensimmäisen painalluksen jälkeen laite kalibroi itsensä asetettuun aloituskulmaan sekä aloittaa lähettämään aktiivisesti laskettuja arvoja myös Bluetooth-yhteyden välityksellä paritettuun Bluetooth-laitteeseen. Tämä tila pidetään päällä niin kauan kuin säiliötä täytetään. Kun säiliön täyttö lopetetaan, käyttäjä painaa nappulaa uudelleen. Seuraavassa vaiheessa laite tallentaa lopetuskulmansa ja laskee näiden kahden kulman perusteella säiliön litramäärän ja kuinka paljon se muuttui täytön aikana. Lasketut arvot välitetään myös Bluetooth-yhteyden välityksellä. Tämän jälkeen nappulan painallus nolaa laitteen takaisin ensimmäiseen lepotilaan. (Kuva 6.)



Kuva 6. Ohjelman vuokaavio

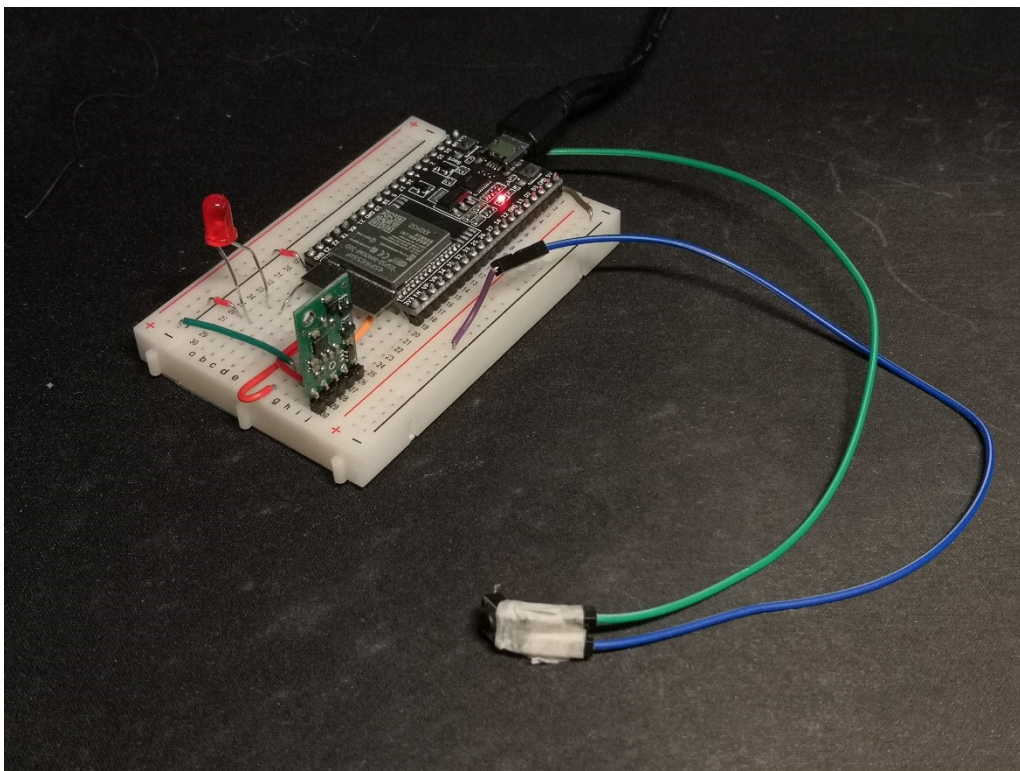
5.3 Mikrokontrolleri

Itse mikrokontrollerin toteutus onnistui vaikeuksitta arkkitehtuuri ja komponenttivalintojen ansiosta. Mikrokontrolleri ei tarvinnut kuin kiihtyvyyssanturin, ledin ja painikkeen toimiakseen. Laite kytkettiin kytkentäalustaan kytkentäkaavion mukaisesti (Kuva 7).



Kuva 7. Laitteen kytkentäkaavio.

Esimerkkikytkennässä laitteen painike jätettiin johdon varaan, jottei sen käyttö vaikuttanut testauksessa tallennettuihin arvoihin. Anturimoduuli oli testausympäristössä asennettu pystyasentoon, mutta ohjelmoinnissa on otettu huomioon, jos lopputuotteessa anturi asennetaan vaakatasoon. (Kuva 8).



Kuva 8. Esimerkkikytkentä kytchentäalustalle.

Laite on hyvä asentaa 90°:n kulmaan, latausportti ylöspäin, jos mittari kääntyy vastapäivään kuten esimerkkimittari (Kuva 9). Näin saadaan varmimmat tulokset sekä poistetaan riski, että laite kääntyisi pisteen yli, jossa mitatut asteet aloittavat uuden kierroksen.



Kuva 9. Säiliön mekaaninen mittari. Kuva: Sami Jalonen.

5.4 Gyroskooppi ja AHRS

Kulma-anturin toteutus aloitettiin käyttämällä jokaista anturia MinIMU-anturialustassa yhteistyössä, täten luoden AHRS (Attitude and Heading Reference System) -järjestelmän. Gyroskoopin tarkka toiminta vaati jokaisella käynnistyskerralla sen kalibrointia, joka onnistuisi parhaiten kääntelemällä anturia edestakaisin muutaman sekunnin ajan. Kompassin kalibrointi taas vaati laitteen kääntelyä muutaman sekunnin ajan, mutta tämä olisi tarpeellinen vain ensimmäisellä käynnistyskerralla. Kalibroinninkin kanssa laitteella oli vaikeuksia pysyä tarkkana yli 60°:n käännöksen jälkeen.

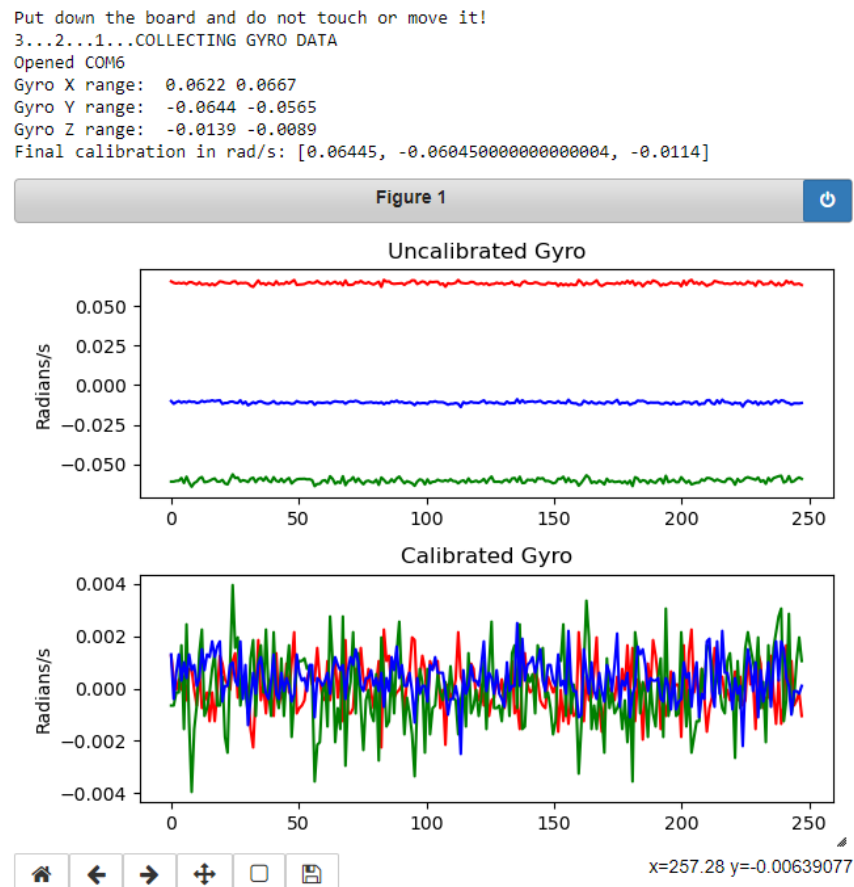
Gyroskoopin kalibroimiseen käytin Adafruitin tarjoamaa kirjastoa ja ohjeita tälle kyseiselle LIS3MDL-anturille [13].

Gyroskoopin kalibrointi tapahtuu aina laitteen käynnistyessä, tällöin laitteen tulisi olla mahdollisimman paikallaan useamman sekunnin ajan kun gyroskooppi lukee lepoluvut siinä asennossa johon se on asetettu. (Kuva 10.)

```
Final zero rate offset in radians/s:
0.0650, -0.0577, -0.0101
```

Kuva 10. LIS3MDL-gyroskoopin kalibrointi-
tiarvot sarjamonitorissa.

Kun ohjelma tietää minkälaiset kiihtyvyyssarvot (rad/s) laitteella on ollessaan paikallaan, se kalibroi nämä arvot vastaamaan mahdollisimman lähelle nollatilaa. Kalibroinnin jälkeen Radians/s skaala on pienentynyt huomattavasti pienempään skaalaan, joka saa tuloksen vaikuttamaan enemmän kohisevalta, mutta huomattavasti tarkemmalta (Kuva 11).



Kuva 11. LIS3MDL-gyroskoopin kalibrointi visualisoitu Jupyter-ohjelman kautta

Anturin kompassimagneetti piti kalibroida erikseen käyttämällä anturia varten luodun koodikirjaston kalibrointiohjelman. Ajamalla tämän ohjelman samalla kun kääntelee anturia, ohjelma tallentaa magneetin suurimmat ja pienimmät arvot jokaiselta akselilta. (Kuva 12.)

```
min: { -3529, -4720, -6231} max: { +3712, +1844, +266}  
min: { -3529, -4720, -6231} max: { +3712, +1844, +266}  
min: { -3529, -4720, -6231} max: { +3712, +1844, +266}  
min: { -3529, -4720, -6231} max: { +3712, +1844, +266}
```

Kuva 12. LIS3MDL-anturin kompassimagneetin kalibrointiarvoja sarjamonitorissa

Kun nämä arvot lisätään koodiin kalibrointifunktioon, niin kompassi kompensoi lukemansa arvot näiden parametrien avulla ja saadaan tulokset mahdollisimman tarkaksi.

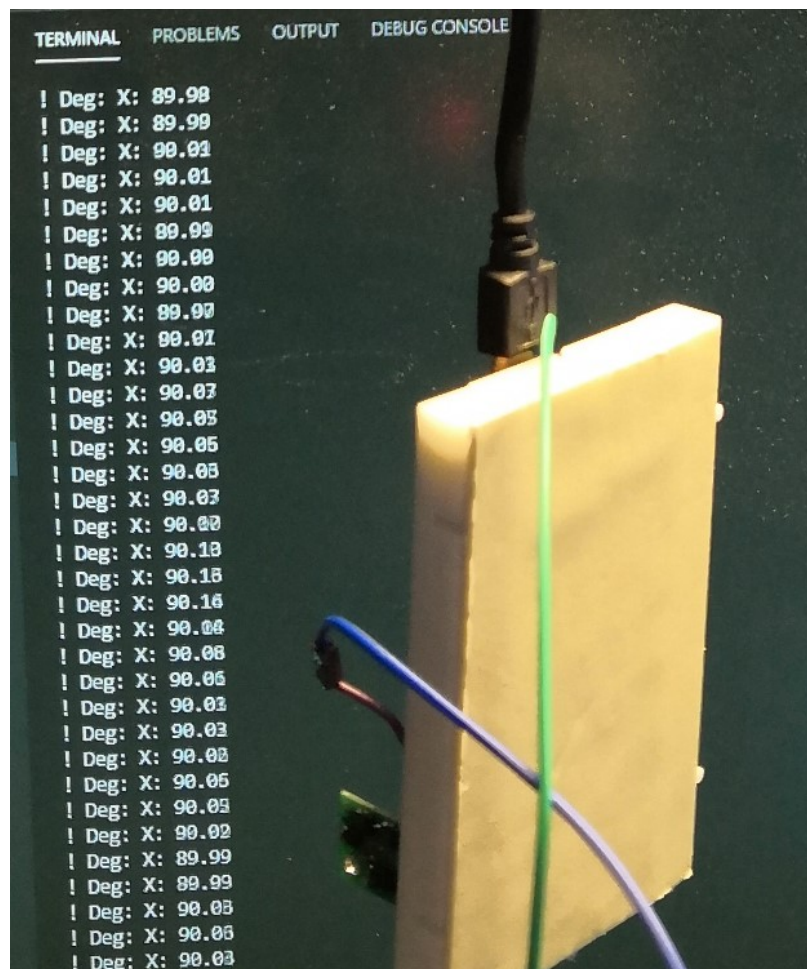
Näiden kalibrointien jälkeen laite olisi vasta toimintakykyinen, mutta ei silloinkaan tarpeeksi tarkka suurien käännösten jälkeen. Lopulta kun kävi ilmi, että AHRS-ratkaisu olisi liian monimutkainen laitteen tarpeisiin ja aiheuttaisi turhaa ja ongelma riskialtista säätöä käyttäjälle laitteen käyttöönoton yhteydessä. Näiden säätöjenkin onnistuessa oli todennäköistä, etteivät mittaustulokset olisi niin tarkkoja mitä vaatimuksissa esitettiin. Täten lähestymistapaa muutettiin käyttämään vain moduulin kiihtyvyyssanturia.

6 TESTAUS

Tässä luvussa käydään läpi laitteen testaus ja miten testaustulokset vastaavat työn vaatimuksia.

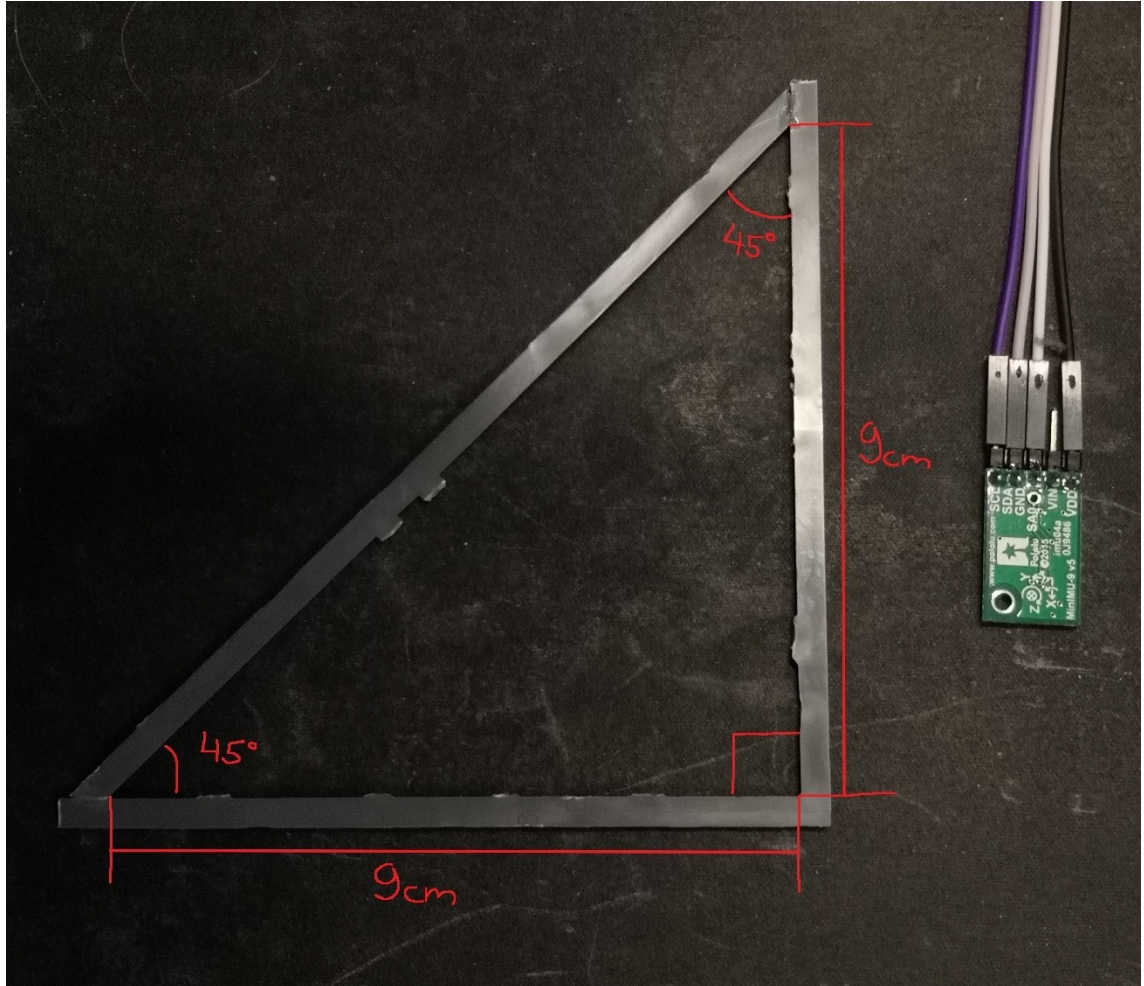
6.1 Kulman tarkkuus

Kiihtyvyyssanturin kulmalaskennan tarkkuus testattiin ensin ripustamalla laite kokonaisuudessaan johdon varaan. Täten voidaan olettaa painovoiman asettavan laitteen siten, että saisimme kulman tulokseksi 90° . Testauksessa näkyy, että arvot pysyvät 89.98° ja 90.16° välillä (Kuva 13). Tässä testissä oli havaittavissa hieman epätarkkuutta, joka saattaa johtua hieman jäykästä johdosta, tai ettei laite ollut vielä täysin asettunut paikalleen ripustamisen jälkeen.



Kuva 13. Anturin tarkkuuden testaus painovoimaa vastaan.

Seuraavaksi käytin muovista suorakulmaista tasakylkistä kolmiota määrittämään anturin tarkkuuden muissakin asennoissa. Koska tämä kolmio on suorakulmainen ja tasakylkinen kolmio, sen kulmat ovat 90° , 45° ja 45° kulmia. (Kuva 14.)



Kuva 14. Suorakulmainen tasakylkinen kolmio anturin tarkkuuden mittausta varten.

Anturi asetettuna kolmion hypotenuusalle nähdään, että anturin lukemat olivat $45,07^\circ$ ja $45,12^\circ$ välillä (Kuva 16). Arvojen pieni kohina oli oletettavissa ja pysyy alhaisena. Kolmion pystykateetille asetettuna havaittiin, että lukemat pysyivät $90,10^\circ$ ja $90,17^\circ$ välillä (Kuva 15). Näillä mittaustuloksilla käy selväksi, että anturin kulmatarkkuus on hyvin alle vaaditun 1° :n tarkkuuden. Tarkkuus on noin $\pm 0,10^\circ$.

```
! Deg: X: 45.07
! Deg: X: 45.09
! Deg: X: 45.08
! Deg: X: 45.08
! Deg: X: 45.08
! Deg: X: 45.08
! Deg: X: 45.11
! Deg: X: 45.12
! Deg: X: 45.10
! Deg: X: 45.09
! Deg: X: 45.08
```

Kuva 16. Sarjamonitorin tulostus suorakulmion kateettia vasten.

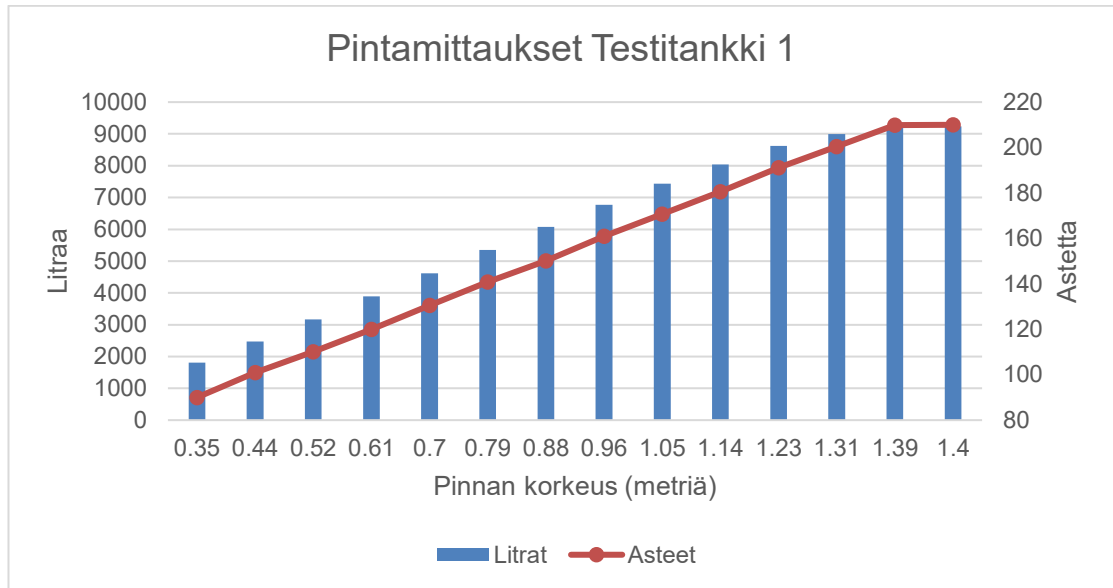
```
! Deg: X: 90.11
! Deg: X: 90.14
! Deg: X: 90.15
! Deg: X: 90.17
! Deg: X: 90.13
! Deg: X: 90.11
! Deg: X: 90.10
! Deg: X: 90.10
! Deg: X: 90.13
! Deg: X: 90.11
! Deg: X: 90.10
```

Kuva 15. Sarjamonitorin tulostus suorakulmion hypotenuusaa vasten.

6.2 Tankkien tilavuus

Laitteen päätoiminta, eli tankkien sisäisen nesteen mittaus, testattiin asettamalla laitteelle kuvitteellisen tankin parametrit. Parametreihin kuuluu säiliön pituus, halkaisija sekä säiliön mekaanisen mittarin viisarin aloitus- ja lopetuskulma. Säiliöiden mekaaninen mittari ei liiku ennen kuin nesteen pinta on jo neljänneksen halkaisijan korkeudesta ja tämä otetaan laskuissa huomioon. Eli laite ei aloita laskuja olettaen, että säiliö on tyhjä. Esimerkiksi testitankissa 1 pinnan korkeuden pitää ylittää $0,35$ metrin korkeus, ennen laskujen aloitusta koska säiliön halkaisija on $1,4$ metriä ($1,4 * 0,25 = 0,35$). Kuva 17, Kuva 18 ja Kuva 19 nähdään erikokoisten testitankkien ja mittariastetaulukoiden mittaustulokset. Näiden alla on Taulukko 4, Taulukko 5 ja Taulukko 6, joissa nähdään tarkat mittaustulokset kyseisiin kuviin. Mittausarvot kirjattiin ylös noin 10° :n välein. Viimeiset kaksi mittaustulosta olivat alle asteen ennen täyttymistä ja täysin täytetyn säiliön arvoja.

Testitankki 1: 6,0 m pitkä, 1,4 m halkaisija, aloituskulma ja lopetuskulmat 90° ja 210°. Täysi tankillinen olisi 9 236,28 litraa.

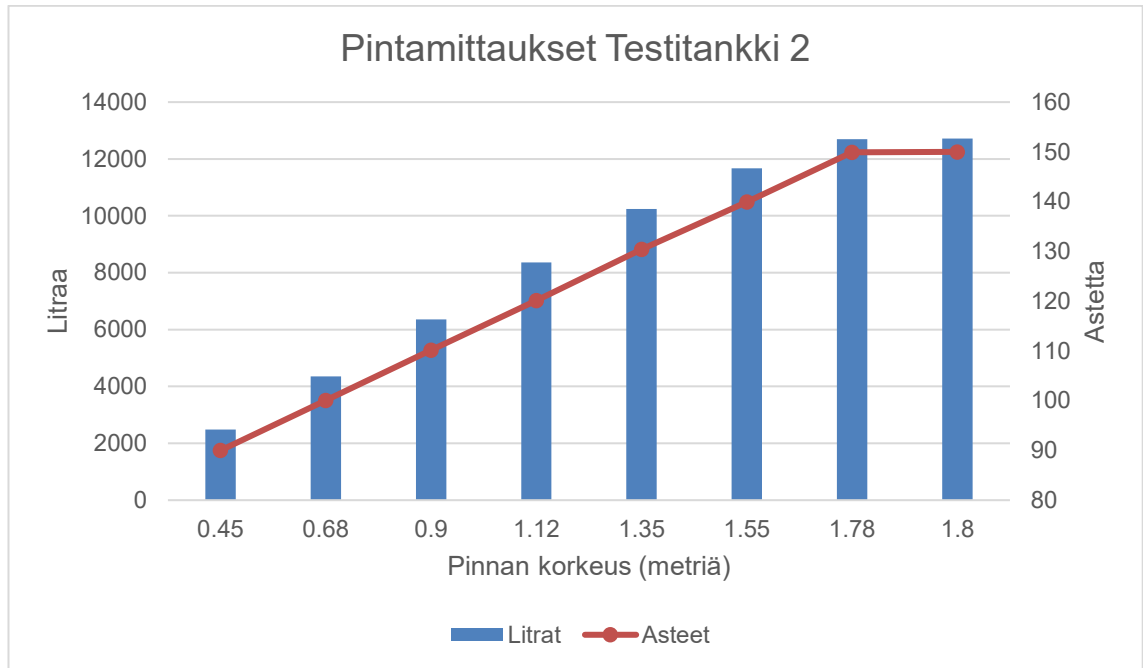


Kuva 17. Testitankki 1:n mittaustulokset.

Taulukko 4. Testitankki 1:n tarkat mittaustulokset.

Tankki 1	Pinta	Litrat	Asteet
6.0 m pitkä 1.4 m halkaisija 90°–210°	0.35	1 805.70	90.00
	0.44	2 469.86	100.96
	0.52	3 163.60	110.08
	0.61	3 889.23	120.03
	0.70	4 618.14	130.50
	0.79	5 355.39	140.84
	0.88	6 072.68	150.18
	0.96	6 774.21	160.99
	1.05	7 430.58	170.75
	1.14	8 040.70	180.53
	1.23	8 619.36	191.15
	1.31	8 997.96	200.50
	1.39	9 228.22	209.89
1.40	9 236.28	210.00	

Testitankki 2: 5,0 m pitkä, 1,8 m halkaisija, aloituskulma ja lopetuskulma 90° ja 150°.
Täysi tankillinen 12 723,45 litraa.

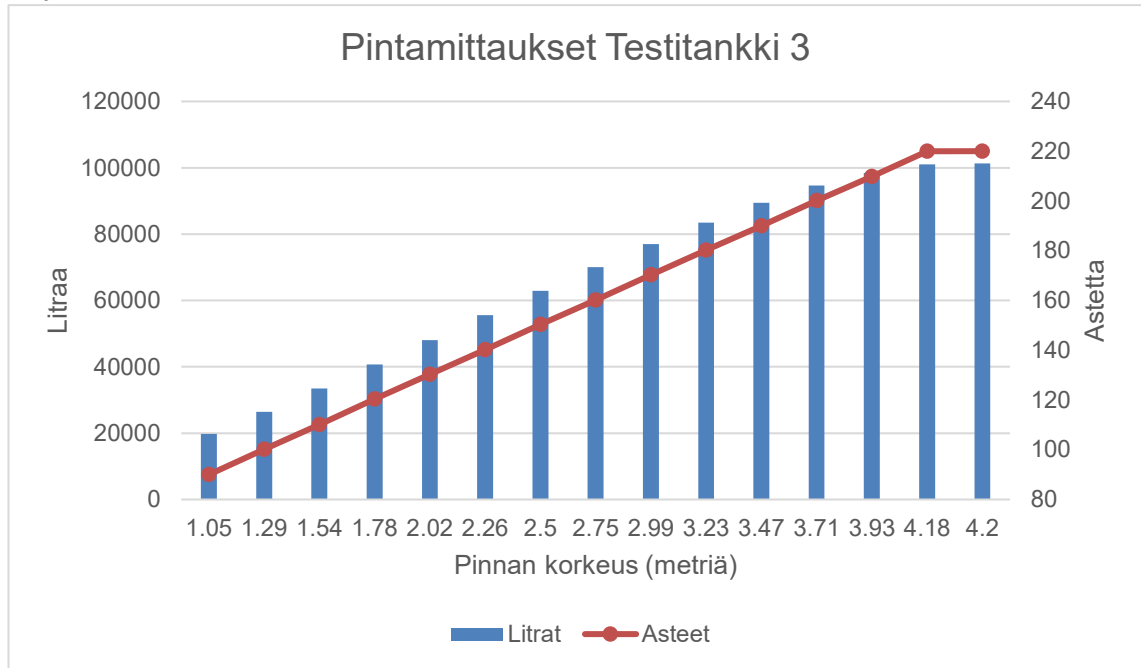


Kuva 18. Testitankki 2:n mittaustulokset.

Taulukko 5. Testitankki 2:n tarkat mittaustulokset.

Tankki 2	Pinta	Litrat	Asteet
5.0 m pitkä	0.45	2 489.45	90.00
1.8 m halkaisija	0.68	4 358.02	100.04
90°–150°	0.90	6 361.73	110.17
	1.12	8 365.43	120.13
	1.35	10 236.00	130.40
	1.55	11 671.85	139.95
	1.78	12 694.37	149.90
	1.80	12 723.45	150.00

Testitankki 3: 7,3 m pitkä, 4,2 m halkaisija, aloituskulma ja lopetuskulma 90° ja 220°.
Täysi tankillinen 101 137,29 litraa.



Kuva 19. Testitankki 3:n mittaustulokset.

Taulukko 6. Testitankki 3:n tarkat mittausluvut.

Tankki 3	Pinta	Litrat	Asteet
7.3 m pitkä	1.05	19 772.45	90.00
4.2 m halkaisija	1.21	26 420.96	100.16
90°–220°	1.54	33 457.07	110.08
	1.78	40 704.65	120.44
	2.02	48 085.80	130.28
	2.26	55 530.64	140.18
	2.50	62 878.45	150.35
	2.75	70 058.04	160.08
	2.99	76 960.06	170.30
	3.23	83 486.96	180.26
	3.47	89 435.88	190.05
	3.71	94 638.24	200.21
	3.93	98 438.36	209.87
	4.18	101 063.25	219.94
	4.20	101 337.29	220.00

Tankkitestien laskuista nähdään, että laite kykenee laskemaan alle litran tarkkuudella säiliön tilavuus laitteen asennon perusteella, kun sille asetetaan oikeat säiliön parametrit. Kuvista myös näkee, asteiden nousu on lineaarinen, mutta litramäärän nousu kiihtyy puoleen täyttöasteeseen saakka ja laskee taas, kunnes viimein säiliö täyttyy. Tämä johtuu säiliön pyöreästä muodosta ja kuinka keskivaiheessa pinnan nousu merkitsee enemmän litroja kuin esimerkiksi yhtä suuri nousu pinnan lähentyessä säiliön kattoa.

Laskujen logiikan puolesta säiliön koolla ei ole merkitystä, vaan laite kykenee laskemaan minkä tahansa kokoisen säiliön tilavuuden, kunhan säiliö on horisontaalisesti asetettu lieriösäiliö ja säiliö käyttää mekaanista pinnanmittausanturia, jossa on asteikolla kääntyvä viisari.

7 YHTEENVETO

Opinnäytetyön tavoitteena oli kehittää anturi, joka kykenee laskemaan säiliön tilavuuden mekaanisen pintamittarin asennon perusteella ja laitteen tulisi olla valmis lähettämään laskettu tieto eteenpäin Bluetooth-yhteyden välityksellä. Työn toteutus tullaan arvioimaan osana lopullista kokonaisuutta, jossa laite tulee toimimaan yhteistyössä sitä varten luotavan puhelinsovelluksen kanssa.

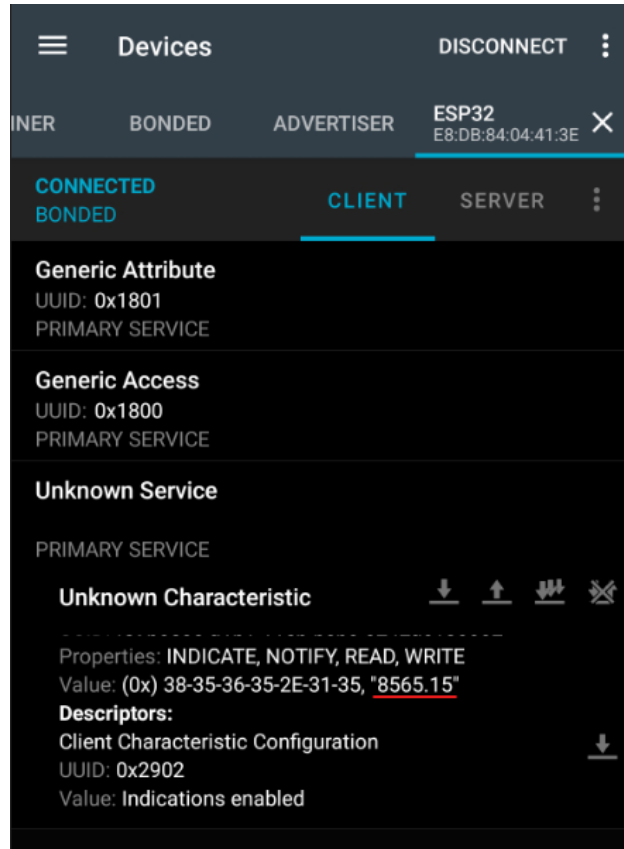
Säiliön mittaus on toteutettu ESP32-alustalle asennetun IMU-moduulin kiihtyvyyssanturin avulla ja tämä toteutus onnistui erinomaisesti. Laitteen mittaamat arvot vastaavat esiasetetun säiliön parametrien mukaisia tuloksia (Kuva 20). Samat arvot näkyvät vastaanotettuna puhelimesta Bluetooth-yhteyden välityksellä (Kuva 21). BLE-lähetysten näkemiseen käytettiin nRF Connect -sovellusta [14]. Laitteen tarkkuus oli toinen erittäin tärkeä tavoite, ja laite suoriutuu tästäkin moitteettomasti. Kulman tarkkuus on jopa $\pm 0,10^\circ$, eli alle 1° :n kuten vaatimuksissa esitettiin, ja täten saadaan laskettua erittäin tarkka litramäärä lokaalisti ennen datan liikkumista eteenpäin. Tarkkuus on erinomainen, vaikka sen kohinaa olisi voinut vieläkin hioa paremmaksi, mutta siinä tapauksessa laitteen reaktiivisuus ja nopeus olisivat kärsineet, sillä kohinanpoisto algoritmi hidastaa mitattujen arvojen ulostuloa sitä enemmän, mitä tarkemmaksi suodatin asetetaan. Vaatimukseen verrattuna nykyinen pieni kohina tuloksissa ($\pm 0,05^\circ$) hyväksytään sillä se on riittävän pieni tähän tarkoitukseen eikä vaikuta laitteen tarkkuuteen liikaa ja täten pitää myös ohjelmassa nopean reaktionopeuden.

```
End angle X:121.47
X axis change: 57.50 degrees
Tank Ullage: 0.65 meters
Sent value :8565.15
Liquid in tank: 8565.15 litres
Resetting device...
```

Kuva 20. Esimerkki BLE-lähetys näytettynä sarjamonitorissa.

Kulman laskeminen onnistui hyvin kiihtyvyyssanturin avulla. Mittaus onnistuu X- tai Y-akseleilla, joten laite on asennettava asentoon, jossa anturi pystyy lukemaan toisen näistä akseleista kääntyessään. Tämä akseli on tällä hetkellä valittava koodissa erikseen. Esimerkkikytkennässä anturi on asennettu 90° :n kulmaan piirilevystä juuri tätä varten.

Tässä tilanteessa laite tulisi asentaa pohja edellä viisariin, joka on pystysuunnassa säiliön kyljessä. Täten laite osaa laskea X-akselin muutokset ja sen perusteella säiliön tilavuuden.



Kuva 21. Kuvankaappaus samasta lähetyksestä kuin Kuva 20 nähtynä nRF-sovelluksessa.

Tilavuuden laskeminen onnistui erinomaisesti ja toimii millä tahansa parametreillä. Säiliö voi olla joko 10-litrainen tai 100 000-litrainen. Kunhan pinnanmittausmekanismi toimii samoin ja säiliö on horisontaalinen lieriö, niin laite kykenee päättelemään siitä säiliön sisäisen nesteen. Säiliön koko vaikuttaa laitteen litratarkkuuteen. Pienemmässä säiliössä sama pinnan muutos kuin suuremmassa säiliössä tarkoittaa pienempää litraeroa, joten litramäärän laskemisen tarkkuus heikkenee mitä suurempaa säiliötä mitataan samanlaisella mekaanisella pintamittarilla. Tulosten absoluuttisen tarkkuuden varmistamiseksi säiliön olisi hyvä olla mahdollisimman tasaisella maastolla mittaushetkellä, sillä tällä hetkellä koodi ei ota huomioon ajoneuvon omaa kulman muutosta.

8 JATKOKEHITYS

Työ laitteen parissa voi jatkua puhelinsovelluksen kehityksen myötä. Puhelimen kautta voitaisiin asettaa helposti uudet säiliön parametrit sekä seurata reaaliajassa säiliön täyttöä. Tärkeintä olisi, että tämä data saataisiin siirrettyä tietokantaan. Tällä hetkellä laitteen lähettämät arvot ovat jo luettavissa Bluetooth-laitteen toimesta, kunhan siihen luo yhteyden. Laite mainostaa itseään jatkuvasti odottaen linkitystä. Laitteen puolesta pitäisi parantaa BLE-lähetysten lukemista, tällä hetkellä laite on lukuvalmis, muttei tee luetuilla viesteillä mitään. Laitteelle tulisi lisätä lähdekoodiin toiminnot luetuille arvoille, kuten parametrien asetus eri kokoisten säiliöiden laskuja varten. Ohjelma tarvitsisi myös toiminnon edetä laitteen vaiheissa ilman fyysistä painiketta, kuten puhelinsovelluksesta lähetettyjen komentojen toimesta.

Jatkokehitystä voidaan myös suunnitella alustanvaihdon parissa, jossa laitteeseen tuodaan GPS-paikannusmahdollisuus sekä erillinen akku ja tämän lataus suoraan ajoneuvon sähköjärjestelmästä. Täten saataisiin myös jatkuva tieto itse laitteelta sen sijainnista. Tämän voisi myös toteuttaa puhelinsovellukseen, jossa voitaisiin käyttää puhelimen omaa paikanninmahdollisuuksia.

Laitteen toiminnan puolesta voisi ottaa käyttöön FLASH-muistin, johon luotaisiin alue, jonne tallennettaisiin laitteisiin säiliökohtaiset asetukset ehkä useammalle säiliölle. Laitteelle voisi myös luoda puhelinsovelluksen kautta hallittavan tilan, jossa voisi mitata mittarin lähtö- ja lopetuskulmat helpommin ja tallentaa ne laitteen muistiin suoraan ilman, että käyttäjän pitää itse arvioida kulman suuruus. Tähän arvioimiseen voi käyttää jo olemassa olevaa reaaliaikaista kulman tulostusta ja sen avulla asettaa mittarin kulmaetäisyyden tarkasti.

Tällä hetkellä laite laskee kulman ottamatta huomioon ajoneuvon omaa lepokulmaa mahdollisen liikkeen jälkeen, joten laitteen tarkkuus ei välttämättä ole niin tarkka jokaisessa olosuhteessa kuin vaatimuksissa esitettiin. Tämä voidaan korjata jatkokehityksen aikana muokkaamalla koodia siten, että laite laskee pinnan korkeuden aina asteiden muutoksen perusteella vertaamalla edelliseen mittauskulmaan eikä nykyisen ratkaisun mukaan, jossa sama kulma tarkoittaa aina samaa pinnankorkeutta joka tilanteessa. Kyseinen ratkaisu on jo olemassa, mutta toteutus vaatisi lisätyöskentelyä. Pitämällä muistissa aina edelliset lukemat ja aloituslukemat ja lisäämällä uudet laskennat edellisiin olisi mahdollinen ratkaisu ongelmaan. Toinen vaihtoehto olisi lisätä tietokantaan uusi kohta

säiliöiden mittarin asennoille ja lähettää mitatut arvot aina tietokantaan ja lukea arvot sieltä laskuja varten ilman, että tarvitsee käyttää laitteen omaa FLASH-muistia.

LÄHTEET

- [1] Mun Oulu 2021. Oululainen Hauru kehitteli älykkään jätehuoltoratkaisun. <https://www.munoulu.fi/artikkeli/-/id/oululainen-hauru-kehitteli-alykkaan-jatehuoltoratkaisun>. Viitattu 2.5.2021.
- [2] Google 2021. Technical overview of internet of things <https://cloud.google.com/architecture/iot-overview>. Viitattu 10.5.2021.
- [3] Traficom 2020. Viestintäpalvelujen kuluttajatutkimus 2020. <https://www.trafficom.fi/fi/search?limit=20&offset=0&query=kuluttajatutkimus&toggle=Viestint%C3%A4palvelujen%20kuluttajatutkimus%202020&toggle=Viestint%C3%A4palvelujen%20kuluttajatutkimus%202018>. Viitattu 12.5.2021.
- [4] Tilastokeskus 2020. Tietotekniikan käyttö yrityksissä 2020. https://www.stat.fi/til/ict/2020/ict_2020_12-03_fi.pdf. Viitattu 13.5.2021.
- [5] Built In 2021. IoT: Internet of Things. <https://builtin.com/internet-things>. Viitattu 11.06.2021.
- [6] Espressif Systems 2021. ESP32-WROOM-32D & ESP32-WROOM-32U Datasheet Version 2.1. https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf. Viitattu 20.4.2021.
- [7] Arduino 2021b. Arduino Mega Rev3. <https://store.arduino.cc/arduino-mega-2560-rev3>. Viitattu 5.5.2021.
- [8] Arduino 2021a. Arduino Uno Rev3. <https://store.arduino.cc/arduino-uno-rev3>. Viitattu 5.5.2021.
- [9] Bluetooth 2021. Bluetooth Radio Versions <https://www.bluetooth.com/learn-about-bluetooth/radio-versions/>. Viitattu 10.5.2021.
- [10] Amazon 2020. AITRIP 3pcs for ESP32-DevKitC core Board ESP32 Development Board ESP32-WROOM-32D Compatible with Arduino IDE. <https://www.amazon.com/AITRIP-ESP32-DevKitC-Development-ESP32-WROOM-32D-Compatible/dp/B08HMJ1X6W>. Viitattu 21.5.2021.
- [11] Pololu 2021. MinIMU-9 v5. <https://www.pololu.com/product/2738>. Viitattu 26.4.2021.
- [12] PlatformIO 2021. <https://docs.platformio.org/en/latest/what-is-platformio.html>. Viitattu 4.5.2021.
- [13] Adafruit 2020. Gyro Calibration with Jupyter. <https://learn.adafruit.com/adafruit-sensorlab-gyroscope-calibration/gyro-calibration-with-jupyter>. Viitattu 23.3.2021.
- [14] Nordic Semiconductor ASA 2021. nRF Connect for Mobile. <https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=en&gl=US>. Viitattu 2.5.2021.