TALLINNA TEHNIKAÜLIKOOL
INSENERITEADUSKOND
Mehaanika ja tööstustehnika instituut

TALLINNA TEHNIKAÜLIKOOL

ENERGIAMONITOORINGU LAHENDUSE PROJEKTEERIMINE JA ARENDAMINE TÖÖSTUSLIKULE ASJADE INTERNETI ANDMEHÕIVESÜSTEEMILE

ENERGY MONITORING SOLUTION DESIGN AND DEVELOPMENT FOR AN INDUSTRIAL INTERNET OF THINGS DATA ACQUISITION SYSTEM

BAKALAUREUSETÖÖ

Üliõpilane: Ingmar Sell

Üliõpilaskood EARB206273EARB

Juhendaja: Heiko Pikner, teadur

Tallinn 2024

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.
Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud.
Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

"30" detsember 2024

Autor: / allkirjastatud digitaalselt /

Töö vastab bakalaureusetöö esitatud nõuetele

"30" detsember 2024

Juhendaja: / allkirjastatud digitaalselt /

Kaitsmisele lubatud

".....".....20… .

Kaitsmiskomisjoni esimees .....
/ nimi ja allkiri /

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina Ingmar Sell

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose Energiamonitooringu lahenduse projekteerimine ja arendamine tööstuslikule asjade interneti andmehõivesüsteemile,

mille juhendaja on Heiko Pikner,

1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

---

30. detsember 2024

[1] *Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.*

TalTech Mehaanika ja tööstustehnika instituut

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Ingmar Sell, 206273EARB

Õppekava, peaeriala: EARB16/17 – Tootearendus ja robootika, 1 - robootika

Juhendaja(d): Heiko Pikner, teadur, +372 539 05 108

Lõputöö teema:

(eesti keeles) *Energiamonitooringu lahenduse projekteerimine ja arendamine tööstuslikule asjade interneti andmehõivesüsteemile*

(inglise keeles) Energy monitoring solution design and development for an industrial internet of things data acquisition system

Lõputöö põhieesmärgid:

1. Loodud süsteemi integreerimine ja testimine
2. Andmesideprotokolli ja energiamonitooringu lahenduse projekteerimine
3. Andmesideprotokolli ja energiamonitooringu lahenduse arendus
4. Süsteemi testimine ja korrigeerimine
5. Kasutajaliidese loomine

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Lõputöö teema ja eesmärkide seadmine	30.09.2024
2.	Loodava lahenduse projekteerimine	31.10.2024
3.	Loodava lahenduse implementeerimine	30.11.2024
4.	Lõputöö kirjutamine	31.12.2024

Töö keel: eesti keel **Lõputöö esitamise tähtaeg:** "31" detsember 2024.a

Üliõpilane: Ingmar Sell / allkirjastatud digitaalselt / "30" detsember 2024.a

Juhendaja: Heiko Pikner / allkirjastatud digitaalselt / "30" detsember 2024.a

Programmijuht: Raivo Sell / allkirjastatud digitaalselt / "30" detsember 2024.a

SISUKORD

EESSÕNA 6
Lühendite ja tähiste loetelu 7
1. SISSEJUHATUS 8
2. PÕHIOSA 10
2.1 Olemasoleva süsteemi kirjeldus 10
2.1.1 Dimusa versioon 2 tausta tutvustus 10
2.1.2 Probleemid ja nõuded uuele süsteemile 12
2.2 Andmesideprotokolli projekteerimine 14
2.2.1 Andmeside protokollide analüüs 14
2.2.2 Üldine andmeside disain 17
2.2.3 CAN-protokolli andmeside disain 20
2.2.4 MQTT-protokolli andmeside disain 24
2.3 Peaplaadi tarkvaraarendus 26
2.4 Energiamonitooringu anduriplaadi arendus 29
2.4.1 Riistvara arendus 29
2.4.2 Tarkvaraarendus ja integratsioon 31
2.5 Süsteemi testimine ja korrigeerimine 34
2.6 Kasutajaliidese loomine 37
KOKKUVÕTE 40
SUMMARY 42
KASUTATUD KIRJANDUSE LOETELU 44
LISAD 47

EESSÕNA

Käesoleva bakalaureusetöö teema, "Energiamonitooringu lahenduse projekteerimine ja arendamine tööstuslikule asjade interneti andmehõivesüsteemile", kujunes välja Tallinna Tehnikaülikooli Mehaanika ja tööstustehnika instituudis läbiviidava Dimusa projekti raames. Töö eesmärk oli luua modulaarne ja laiendatav energiamonitooringulahendus, mis võimaldab reaalajas koguda ning analüüsida tootmisprotsesside andmeid, tagades seeläbi võimalused protsesside optimeerimiseks ja ressursikasutuse parandamiseks.

Põhilised algandmed koguti Tallinna Tehnikaülikooli Mehaanika ja tööstustehnika instituudi laboritingimustes. Suur osa tööst toetus Dimusa projekti raames saadud sisenditele ning praktilisele tagasisidele, mis aitas lahendust kujundada reaalseid tööstusvajadusi silmas pidades. Loodud lahendus hõlmab nii riistvaralisi kui ka tarkvaralisi komponente, sealhulgas kontrollerite alavõrku (CAN, ingl k Controller Area Network) ja sõnumijärjekorra telemeetria transporti (MQTT, ingl k Message Queuing Telemetry Transport), STM- ja ESP-mikrokontrollerite kasutamist, InfluxDB andmebaasi ning Grafana visualiseerimistööriistu. Tulemuseks valmis skaleeritav ja paindlik prototüüp, mida saab rakendada mitmesugustes tööstuslikes keskkondades.

Soovin avaldada tänu juhendajale Heiko Piknerile, kelle pädev ja suunav roll oli hindamatu kogu arendusprotsessi vältel. Samuti tänan Dimusa projekti meeskonda, eriti Kristo Karjust, Tõnis Raamets ja Riho Uusjärv, kelle kaasamõtlemine ja asjakohane tagasiside aitasid lahenduse kvaliteeti tõsta.

Loodan, et käesolev töö toetab nii Dimusa projekti arengut kui ka pakub uusi võimalusi energiamonitooringu rakendamiseks ja tööstusprotsesside juhtimise täiustamiseks ka laiemalt.

Võtmesõnad: energiamonitooring, tööstuslik IoT, CAN, MQTT, bakalaureusetöö

Lühendite ja tähiste loetelu

IoT – asjade internet (ingl k Internet of Things), süsteem, kus omavahel ühendatud seadmed ja andurid suhtlevad üle interneti, võimaldades automatiseerimist ja andmete reaalajas kogumist ja analüüsi. IoT-rakendused on levinud näiteks koduautomaatikas, tervishoius ja tööstuses.

CAN – kontrollerite alavõrk (ingl k Controller Area Network), sõidukites ja tööstussüsteemides laialdaselt kasutatav andmesideprotokoll, mis võimaldab mikroprotsessoritel ja seadmetel omavahel efektiivselt andmeid vahetada.

MQTT – sõnumipõhine telemeetriaprotokoll (ingl k Message Queuing Telemetry Transport), kerge ja laialdaselt kasutatav protokoll, mis on mõeldud vähese ressursiga seadmete jaoks andmete edastamiseks üle TCP/IP-võrgu, näiteks IoT-süsteemides.

ESP – mikrokontrolleriperekond, mille on välja töötanud Espressif Systems. ESP-seadmeid (nt ESP8266, ESP32) kasutatakse sageli IoT-rakendustes tänu nende sisseehitatud Wi-Fi ja Bluetooth-ühendusele ning energiasäästlikule disainile.

STM – STMicroelectronicsi mikrokontrolleriperekond (nt STM32), mida iseloomustab lai arhitektuurivalik (ARM Cortex-M südamikud), kõrge töökindlus ja laialdane kasutus tööstuslikes ja sisseehitatud süsteemides.

SSL – turvaline soklite kiht (ingl k Secure Sockets Layer), andmesideprotokoll, mida kasutatakse andmete krüpteerimiseks ja turvalise suhtluse tagamiseks arvutivõrkudes. SSL on tänapäeval asendatud uuema protokolli TLS (Transport Layer Security) poolt, kuid nimetus SSL on endiselt laialdaselt kasutusel.

CERN – Euroopa Tuumauuringute Organisatsioon (ingl k European Organization for Nuclear Research), rahvusvaheline uurimiskeskus, mis tegeleb elementaarosakeste füüsika uurimisega. Tuntud ka kui maailma suurima osakestekiirendi LHC (Large Hadron Collider) asukoht.

DIP – lülitiplokk (ingl k Dual In-line Package switch), elektrooniline komponent, mis koosneb väikesemõõtmelistest käsitsi seadistatavatest lülititest. Kasutatakse peamiselt seadmete konfigureerimiseks ja seadete valimiseks.

1. SISSEJUHATUS

Tänapäeva tööstussektor seisab silmitsi järjest keerukamate väljakutsetega: tuleb parandada tootmise efektiivsust, vähendada kulusid, suurendada paindlikkust ja tagada stabiilsed toimimisviisid. Oluliseks vahendiks nende eesmärkide saavutamisel on reaalajaline andmehõive ja analüüs, mis võimaldab koguda teavet nii energiakasutuse kohta kui ka laiemalt tootmisprotsesside toimimisest [1]. Lisaks energiatarbimise vähendamisele ja ressursikasutuse optimeerimisele võimaldab andmete kogumine analüüsida masina aktiivsuse aegu, tootmistsüklite pikkusi, seadmete kasutuskoormust ja seisakute põhjuseid [2]. Üha kasvav andmemaht loob omakorda eeldusi keerukamate analüüsimeetodite, sealhulgas tehisintellektil põhinevate mudelite rakendamiseks. Viimased suudavad reaalajas mustreid tuvastada, anomaaliaid avastada ning pakkuda mudelipõhiseid soovitusi tulemuslikumaks tootmise juhtimiseks.

Hoolimata sellest, et energiamonitooringu ja andmehõive lahendusi on turul juba olemas, ei pruugi nende paindlikkus, mastaapsus ja laienemisvõimekus vastata tänapäeva tööstuskeskkondade nõuetele. Tavapärased kommerts- ja vabavaralised lahendused on sageli loodud kindlate stsenaariumite jaoks, mistõttu nende rakendamine keerukamates oludes osutub vaevaliseks. Käesoleva lõputöö eesmärk ongi luua modulaarne ja laiendatav tööstuslik IoT-lahendus, mis toetab nii energiamõõtmist kui ka laiemat tootmisprotsesside analüüsi, luues aluse andmete reaalajaliseks töötlemiseks ning nende edasiseks kasutamiseks, sealhulgas tehisintellekti rakendamiseks.

Lõputöö teema tuleneb Tallinna Tehnikaülikooli juhtimisel arendatavast Dimusa projektist, mille eesmärk on luua paindlik ja skaleeritav tööstuslik andmehõiveplatvorm. Kuigi Dimusa 2 versioon tõi süsteemi arendamisse olulisi uuendusi, jäid alles kitsaskohad, nagu piiratud laiendatavus, raskused eri tüüpi andurite integreerimisel ning andmete piiratud kasutusvõimalused. Käesolev töö keskendub Dimusa versioon 3 edasiarendamisele, eesmärgiga tõsta platvormi tehnilist taset nii, et see suudaks pakkuda terviklikku andmehõivelahendust – alustades täpsest energiatarbe mõõtmisest ja lõpetades laiema tootmistegevuse analüüsiga.

Väljatöötatav lahendus integreeritakse Euroopa Tuumauuringute Organisatsiooni (CERN) projektiga seotud metalli 3D-printeri jälgimiskeskkonnaga, kus on oluline reaalajas ülevaade energiatarbest, protsessiaegadest ning muudest kriitilistest parameetritest [3]. Andmete salvestamiseks kasutatakse InfluxDB andmebaasi, mis on optimeeritud ajareaga andmete kiireks töötlemiseks. Visualiseerimise eesmärgil

rakendatakse Grafanat, mis pakub mugavat viisi andmete ülevaatlikuks ja paindlikuks esitamiseks. Andmeedastuseks kasutatakse CAN- ja MQTT-protokolle, mis tagavad tõhusa ja paindliku suhtluse eri süsteemikomponentide vahel. Vajadusel kohandatakse vabavaralisi vahendeid, nagu OpenEnergyMonitor, et vastata konkreetsetele projektiinõuetele. Riistvaraliselt tuginetakse STM- ja ESP-mikrokontrolleritele, mis sobivad suure hulga andurite haldamiseks ja keerukamate arvutuste teostamiseks.

Lõputöö teemavalik on aktuaalne nii tehnoloogilisest, majanduslikust kui ka tulevikku suunatud vaatenurgast. Täpsed andmed ja reaalajaline analüüs võimaldavad ettevõtetel teha teadlikumaid otsuseid, ressursse säästa, protsesse optimeerida ning sellega kulusid vähendada. Lisaks luuakse eeldused tehisintellektil põhinevate lahenduste kasutamiseks, mis võivad oluliselt parandada otsustamise kiirust ja täpsust. Dimusa 3 edendamine tugevdab projekti ökosüsteemi ja laiendab selle rakendusalasid, soodustades uute funktsioonide, anduritüüpide ning tööstuslike rakenduste lisandumist.

Lõputöös rakendatakse mitmekülgseid meetodeid: riistvaraarenduses kujundatakse anduriplaadid ja laaditakse neile vajalikke funktsioone täitvad programmid; tarkvaraarenduses valitakse ja kohandatakse andmesideprotokolle, kasutatakse vabavaralisi lahendusi ning rakendatakse konteineripõhist juurutust (Docker) lihtsustamaks hooldust ja laiendamist. SSL-sertifikaatide haldamiseks kasutatakse Certboti. Töö käigus hinnatakse arenduste tulemusi nii kohalikus keskkonnas kui ka pilvepõhiselt.

Lõputöö on jagatud kolmeks peamiseks osaks:

1. Esimeses peatükis antakse ülevaade tööstuslikust IoT-st, energiamonitooringust, tootmistegevuse andmete analüüsi vajadusest ja seni kasutusel olnud lahendustest.
2. Teises peatükis käsitletakse loodud süsteemi arendust, sealhulgas riistvara, andmeside, tarkvaraarhitektuuri ja kasutajaliidese loomist.
3. Kolmandas peatükis keskendutakse süsteemi testimisele, tulemuste analüüsile ja võimalikule edasisele arengule.
4. Kokkuvõttes esitatakse järeldused ja kaardistatakse edasised arengusuunad.

Töö tulemusena valmib energiamonitooringu ja tootmistegevuse andmeanalüüsi lahendus, mis täiendab Dimusa ökosüsteemi ning loob eeldused sügavamaks andmeanalüüsiks, pakkudes tööstusettevõtetele paindlikku, laiendatavat ja tulevikukindlat andmehõive- ning visualiseerimiskeskkonda.

2. PÕHIOSA

2.1 Olemasoleva süsteemi kirjeldus

2.1.1 Dimusa versioon 2 tausta tutvustus

Dimusa on Tallinna Tehnikaülikooli juhtimisel arendatav tööstuslik andmehõiveplatvorm, mis vastab kasvavale vajadusele liidestada tööstusprotsessid tänapäevaste andmete kogumise, edastamise, analüüsi ja visualiseerimise tehnoloogiatega. Varasem projektietapp, Dimusa 1, andis ülevaate sellest, milliseid tehnilisi väljakutseid tuleb lahendada, et luua paindlik, kulutõhus ja kasutajamugav lahendus tööstuslike andmetega toimetulekuks. Nii leiti, et senised levinud tööstuslikud andmehõivemehhanismid baseerusid sageli vananenud või liigselt jäikadel tehnoloogiatel, mis muutsid süsteemide laiendamise, uuendamise ja kohandamise kasutaja vajadustega ebamugavaks ning aeganõudvaks.

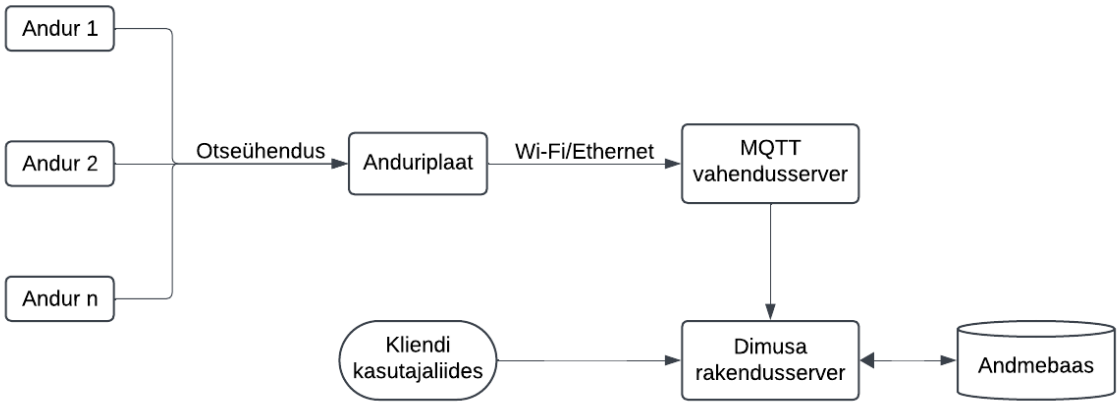
Sellele taustale tuginedes arendati välja Dimusa 2, mis juba projekteerimisel keskendus moodsa IoT-filosoofia rakendamisele. IoT (Internet of Things) on lähenemine, kus erinevad seadmed – antud kontekstis tööstuslikud andurid, kontrollerid ja aktuaatorid – on interneti või muude võrguühenduste kaudu omavahel seotud. See loob eelduse märksa suuremale paindlikkusele, skaleeritavusele ning võimaldab kiiremini rakendada uusi funktsionaalsusi. Dimusa 2 eesmärk oli tuua tööstusesse tavakasutuses levinud IoT-lahenduste paindlikkus ja lihtne laiendatavus, mis seni iseloomustas peamiselt koduautomaatika või nutiseadmete valdkonda.

Süsteemi arhitektuuri keskmeks on ESP-seeria mikrokontrolleril põhinevad andurimoodulid, mille abil saab hõlpsalt ühendada erinevaid andureid vastavalt konkreetse tööstusprotsessi nõuetele. Olgu selleks temperatuuri, rõhu, vibratsiooni, voolutugevuse või mõne muu füüsikalise suuruse mõõtmine – modulaarne arhitektuur (vt joonis 1) võimaldab paindlikult reageerida erinevatele situatsioonidele, ilma et oleks vaja kogu süsteemi algosadeks lahti võtta. Lisaks antakse süsteemile valik sidemeetodite vahel: Ethernet ja Wi-Fi, mis võimaldab rakendada Dimusa lahendust nii traditsioonilise tööstusvõrgu kui ka juhtmevabade lahendustega keskkondades, kus kaabeldus on problemaatiline.

Mõõdetud andmete edastamiseks kasutatakse tavapärast IoT-maailmas laialt levinud MQTT andmeedastusprotokolli, mis on kergekaaluline, tõhus ja hästi skaleeritav. MQTT

abil saab andurimoodul andmeid edastada Dimusa rakenduserverisse, kus need salvestatakse andmebaasi. Selline arhitektuur pakub mitmeid eeliseid:

- Hõlpsam laiendatavus: Uue anduri lisamisel ei ole vaja põhjalikke süsteemimuudatusi, piisab mooduli konfigureerimisest.
- Tsentraliseeritud andmehaldus: Kõik andmed koonduvad ühte kesksesse andmebaasi, mis lihtsustab nii nende analüüsi, hooldust kui ka edasiarendust.
- Kasutajamugav visualiseerimine: Andmete jaoks on loodud veebirakendus, mis võimaldab neid mugavalt sirvida, analüüsida ja kohandada kliendi vajadustest lähtuvalt. Niimoodi saab kasutaja luua unikaalseid vaateid erinevate mõõteandmete põhjal, pannes kokku just talle olulise info.



Joonis 1. Dimusa 2 arhitektuur

Praktikas rakendatud Dimusa 2. versioon on pakkunud väärtuslikku kogemust ja tagasisidet. Reaalsetest tööstuspaigaldustest saadud kogemus on näidanud, et kuigi Dimusa 2 on juba märkimisväärne edusamm võrreldes varasemate lahendustega, leidub endiselt kohti, mida saaks optimeerida. Kasutajad on tõstatanud küsimusi süsteemi veelgi parema skaleeritavuse, seadistusprotsessi lihtsustamise, turvalisuse, efektiivsema andmetöötluse ja mugavama kasutajaliidese osas. Selline otsene tagasiside võimaldab projekti meeskonnal teadlikult suunata järgmisi arendus- ja projekteerimistegevusi.

Käesoleva töö raames keskendutaksegi Dimusa järgmisele loogilisele sammule: Dimusa versioon 3 arendamisele. Võttes arvesse nii seniseid õppetunde Dimusa 2 rakendustest kui ka kasutajatelt saadud soovitusi, on eesmärgiks luua veelgi paindlikum, usaldusväärsem ja tulevikukindlam lahendus. Järgmistes peatükkides käsitletaksegi üksikasjalikult nende probleemide kaardistamist, uue süsteemi nõuete püstitamist ning

lahenduste väljatöötamist, mis viiksid Dimusa 3 platvormi tööstuslikus kontekstis uuele, veelgi konkurentsivõimelisemale tasemele.

2.1.2 Probleemid ja nõuded uuele süsteemile

Varasem kogemus Dimusa 2 kasutamisel ning otsene klienditagasiside on näidanud, et hoolimata platvormi võimekusest andmete kogumisel, edastamisel ja analüüsimisel, esineb mitmeid kitsaskohti, mis piiravad süsteemi paindlikkust, laiendatavust ja kasutusmugavust. Klientide esitatud tagasiside viitab nii tehnilistele kui ka organisatsioonilistele väljakutsetele: andureid on keerukas lisada, tarkvaraversioonide haldus on aeganõudev, süsteemi töökindlus vajab parandamist ning klient eeldab suuremat iseseisvust lahenduse seadistamisel.

Tuvastatud probleemid:

1. Energiatarbe mõõtmise võimaluse puudumine
2. Piiratud paindlikkus ja moodulite lisamise keerukus
3. Keeruline tarkvaraversioonide haldamine
4. Vajadus eri seadistuste korral kohandatud tarkvara järele
5. Ebapiisav töökindlus
6. ESP-seeria mikrokontrolleri piirangud keerukamate andurite kasutamisel
7. Konfigureerimise keerukus kliendi poolt

Nõuded uuele süsteemile:

1. Lahendus peab olema modulaarne, et süsteemi saaks dünaamiliselt laiendada ja kohandada
2. Erinevate rakenduste korral ei tohi tekkida vajadust kohandatud tarkvara järele. Sama tüüpi andurid peavad kasutama sama tarkvaralahendust
3. Süsteemi seadistamine peab olema lihtne ja kliendipoolsel kasutajal kergesti teostatav, ilma tehnilise toe kaasamiseta
4. Andmeside lahendus peab toetama dünaamilist ja töökindlat ülesehitust, vältides käsitsi seadistamist uute andurite lisamisel
5. Süsteem peab olema hõlpsalt laiendatav ja nõudma minimaalset lisakonfigureerimist uute funktsionaalsuste rakendamisel
6. Anduriplaat peab olema võimekas, toetades ka keerukamaid andureid
7. Kõik andurid peavad edastama andmeid MQTT vahenduserverisse, luues ühtse ja standardse andmevoo

Neid nõudeid järgides on Dimusa 3 eesmärk olla varasemast paindlikum, hõlpsamini hooldatav ja stabiilsem andmehõiveplatvorm, mis vastab paremini tänapäevaste tööstusettevõtete vajadustele. Järgmised peatükid keskenduvad üksikasjalikumalt nende nõuete rakendamisele andmesideprotokolli projekteerimisel, riistvara- ja tarkvaralahenduste väljatöötamisel ning kasutajaliidese loomisel.

2.2 Andmesideprotokolli projekteerimine

2.2.1 Andmeside protokollide analüüs

Enne uue lahenduse kavandamist tuleb hoolikalt läbi mõelda, milliseid andmesideprotokolle süsteem kasutab. Sobiva protokolli valik mõjutab otseselt nii süsteemi laiendatavust, töökindlust, andmeside jõudlust kui ka paindlikkust. Eelnevalt püstitatud nõuded annavad selged suunised lahenduse tehnilise ülesehituse osas: süsteem peab olema modulaarne, hõlpsalt konfigureeritav, toetama keerukaid andureid ning tagama ühtse andmevoo läbi MQTT vahenduserveri [4].

Nende nõuete täitmiseks otsustas autor esmalt läbi viia andmesideprotokollide analüüsi. Selleks defineeriti valikukriteeriumid, mille alusel saab võrrelda erinevaid andmesideprotokolle ning teha teadliku otsuse, milline protokoll pakub parimat sobivust kavandatavale süsteemile. Analüüs hõlmab nii juhtmega kui ka juhtmevabu lahendusi, hinnates nende andmeside kiirust, latentsust, skaleeritavust, töökindlust, integreerimise keerukust, voolutarvet, kasutusvaldkondi ja modulaarsust. See lähenemine aitab valida protokolli, mis loob tugevad eeldused Dimusa 3 edukaks rakendamiseks ning toetab tulevikus süsteemi jätkuvat arengut.

Esmalt määratles autor parameetrid, mille põhjal hinnata erinevaid andmesideprotokolle. Neid parameetreid kasutatakse valikukriteeriumitena, et tagada protokolli sobivus eelnevalt püstitatud nõuete ja kavandatud arhitektuuriliste põhimõtetega. Võrdluseks võeti järgmised tunnused:

1. Tüüp
2. Andmeside kiirus
3. Latentsus
4. Skaleeritavus
5. Töökindlus
6. Integreerimise keerukus
7. Voolutarve
8. Peamised kasutusalad
9. Modulaarsus

Järgnevalt tuuakse välja erinevate andmesidetehnoloogiate analüüsi tulemuste kokkuvõte (vt lisa 3):

1. CAN on juhtmega jadasiin, mis võimaldab vähese latentsusega ja usaldusväärset reaalajas andmeedastust. Uuringud on näidanud, et see tehnoloogia sobib eriti hästi tööstuslikes rakendustes, kus on oluline kiire andmeedastus ning süsteemi paindlikkus sensorite lisamisel või eemaldamisel [5, 6].
2. MQTT kasutab väljastamis-tellimise (publish-subscribe) mudelit, mis tagab paindliku ja kergesti skaleeritava andmeedastuse üle IP-võrkude. Teadusallikad osutavad, et see protokoll on äärmiselt tõhus hajutatud ja kaugjuhtimisega süsteemides, kus on vajalik lihtne integreerimine ning laiendusvõimalused [7].
3. Modbus RTU on traditsiooniline juhtmega jadaliidese protokoll, mida kasutatakse tööstuskeskkondades endiselt laialdaselt selle lihtsuse ja töökindluse tõttu. Kuigi Modbus RTU paindlikkus ja mastaapsus on piiratud, sobib see rakendustesse, kus nõutakse üksnes põhilisi andmeedastusfunktsioone [8].
4. Zigbee on traadita "mesh"-tüüpi võrguprotokoll, mis toetab suurt hulka seadmeid ning võimaldab hajutatud keskkondades andureid hõlpsasti lisada. Seda soovitatakse eriti rakendustes, kus on oluline madal energiatarve ja ulatuslik laiendatavus. Samas tuleb arvestada, et Zigbee andmeedastuskiirus ja leviulatus jäävad madalamaks kui juhtmega süsteemidel [9, 10].
5. HTTP/REST põhineb päring-vastus mudelil ning on laialdaselt kasutusel pilveteenuste ja veebirakenduste integreerimisel. Uuringud näitavad, et kuigi HTTP/REST on hästi skaleeritav ja laialt toetatud, võivad selle latentsus ja energiatarve olla suuremad kui mitmetel teistel reaalaja protokollidel, eriti otseselt seadmetes kasutatavate rakenduste korral [11].

Läbiviidud analüüsi tulemusel selgus, et soovitud arhitektuuri ja nõuete täitmiseks on otstarbekas rakendada protokollide kombineeritud lähenemist. Autor otsustas kasutada võrgutasandil MQTT-protokolli ning anduriplaatide ja põhiplaadi vaheliseks suhtluseks CAN-siini.

MQTT-protokoll sobib süsteemi jaoks seetõttu, et see toetab avaldamise ja tellimise mudelit, mille abil saab hõlpsalt korraldada andurite eristamist ning tagada sujuv laiendatavus. MQTT võimaldab edastada andmeid perioodiliselt keskserverisse, kasutades seejuures teemastruktuuri, mille abil saab andureid omavahel selgelt piiritleda. See protokoll on hästi kohandatav erinevate kasutusjuhtudega, toetab suurel hulgal seadmeid ning võimaldab tulevikus süsteemi vajadusel laiendada.

Anduriplaatide ja põhiplaadi vaheliseks andmeedastuseks valiti CAN-siin, mis on eriti sobiv töökindlate tööstuslike rakenduste tarbeks. CAN-siin võimaldab moodulite dünaamilist lisamist ning eemaldamist, hoides seejuures andmevoo hästi hallatavana. CAN kasutab iga edastatud sõnumi eristamiseks ainulaadseid tunnuseid, võimaldades seeläbi eristada erinevaid andureid ja tagada andmevahetuse prioriteedist lähtuva korrastatuse. Lisaks vastutab CAN-siin ka sõnumite edastusvoo juhtimise eest, mis vähendab seadistamis- ja hoolduskoormust ning parandab süsteemi üldist töökindlust.

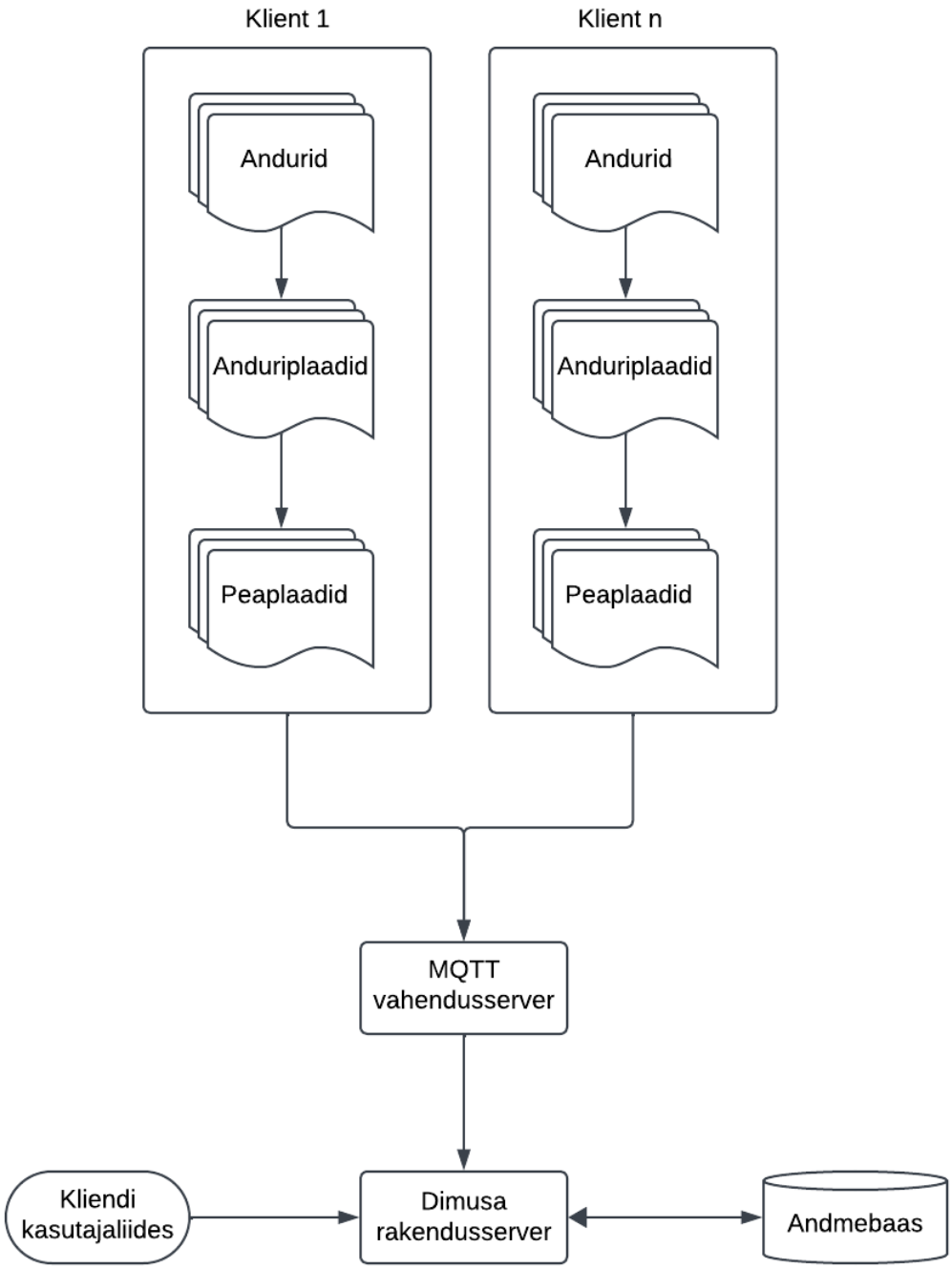
Selline protokollide kombinatsioon vastab esitatud nõuetele: tagatakse dünaamiline laienemine, selge andurite eristamine, andmete usaldusväärne edastus ning paindlik ja jätkusuutlik arhitektuur. Järgmistes peatükkides käsitletakse üksikasjalikumalt, kuidas neid valitud protokolle Dimusa 3 lahenduses rakendada.

2.2.2 Üldine andmeside disain

Selleks, et täita varasemalt sõnastatud nõuded ning luua tulevikukindel, paindlik ja laiendatav lahendus, kavandas autor süsteemile mitmekihilise andmesidetehnoloogia. Lõppkasutajale üleantav lahendus koosneb kolmest põhikihist: anduritest, anduriplaatidest ja peaplaatidest. Iga kiht moodustab sõltumatu tasandi, kuhu saab dünaamiliselt komponente lisada või neid eemaldada, ilma et see eeldaks keerukat riist- või tarkvaralist ümberseadistust. Niisugune ülesehitus tagab, et süsteemi koosseisu on võimalik lihtsalt kohandada vastavalt konkreetsetele vajadustele – lisades uusi andureid, täiendavaid anduriplaate või peaplaate (vt joonis 2).

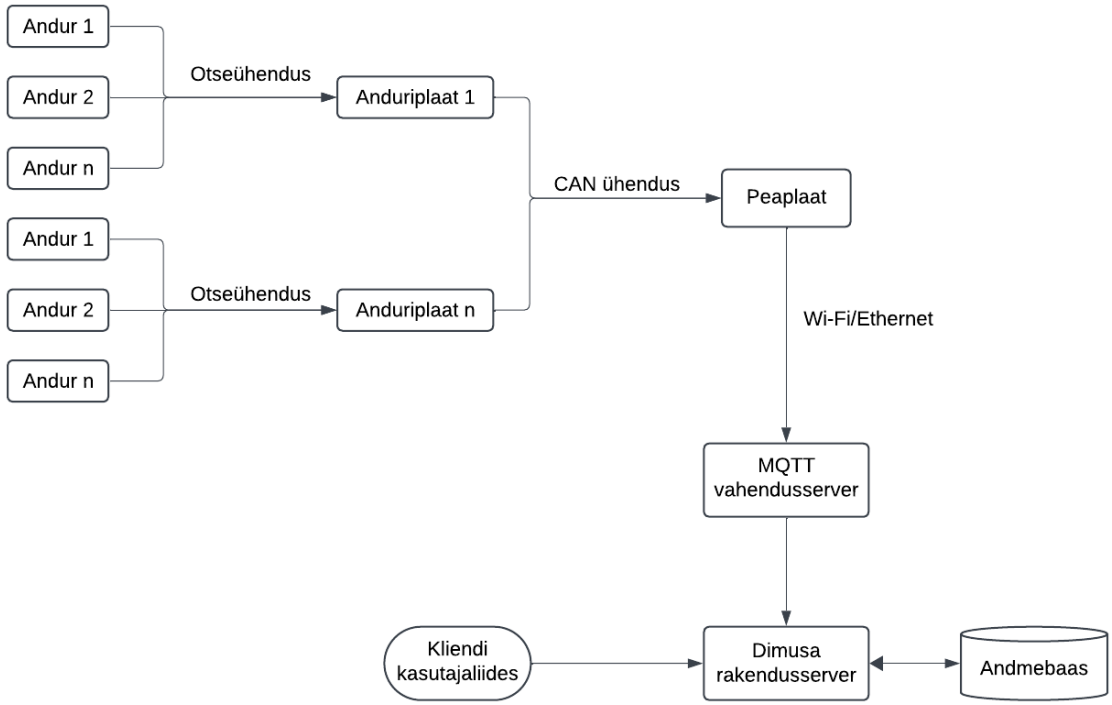
Andurikiht sisaldab erinevaid mõõtekomponente, mis võivad ulatuda lihtsatest temperatuuri- või rõhuanduritest kuni keerukamate energiatarvet jälgivate seadmeteni. Iga andur ühendatakse otse anduriplaadiga, mis vastutab mõõteandmete eeltöötluse ning nende edastamise eest järgmisse kihti. Tänu sellele saab andurikihti hõlpsasti laiendada, lisades uusi andureid ilma süsteemi üldist toimimist häirimata.

Anduriplaatide kiht on vaheaste andurite ja peaplaadi vahel, kasutades omavaheliseks suhtluseks CAN-siini. See pakub töökindlat, madala viivitusega ja hästi laiendatavat andmeedastusviisi. CAN-siini kaudu saab juurde lisada uusi anduriplaate, ilma et see nõuaks põhjalikku ümberseadistamist.



Joonis 2. Dimusa 3 kihiline arhitektuur

Peaplaat on hierarhia tipus ning vahendab anduritelt anduriplaatide kaudu saabuvaid andmeid keskserverisse. Siin kasutatakse andmete edastamiseks MQTT-protokolli, mis võimaldab rakendada avaldamise ja tellimise mudelit. See lähenemine muudab süsteemi laiendamise lihtsaks ja läbipaistvaks, kuna uute andurite lisamisel ei ole tarvis muuta kogu ülesehitust. Peaplaat vastutab andmepakettide koondamise, töötlemise ja nende keskserverisse suunamise eest.



Joonis 3. Dimusa 3 andmeside ülesehitus

Selle kolmeastmelise struktuuri tulemusel saab süsteemi paindlikult kohandada ning vajaduse korral laiendada, lisades või eemaldades andureid ja anduriplaate (vt joonis 3). Samuti on võimalik kliendi erinevatel masinatel kasutada mitut peaplaati, nii et iga masin on varustatud oma peaplaadiga. Iga peaplaat edastab anduridelt kogutud teabe MQTT vahendusserverisse, tagades andmete ühtlase jaotuse ning hoides süsteemi kergekaalulisena ja lihtsalt hallatavna. Niisugune ülesehitus vastab nii laiendatavuse, paindlikkuse kui ka töökindluse nõuetele, luues tugeva aluse Dimusa 3 andmesidestruktuurile. Varasemad uuringud on näidanud, et MQTT vahendusserveritel põhinevad süsteemid toetavad väga hästi hajutatud ja reaalajas toimivaid lahendusi, suurendades tõhusust ja andmete kättesaadavust [12].

2.2.3 CAN-protokolli andmeside disain

CAN-protokolli kasutatakse anduriplaatide ja peaplaadi vaheliseks andmeedastuseks. Autori eesmärk oli luua andmesidestruktuur, mis vastaks eelnevalt sätestatud nõuetele, tagades nii paindliku, laiendatava kui ka töökindla lahenduse.

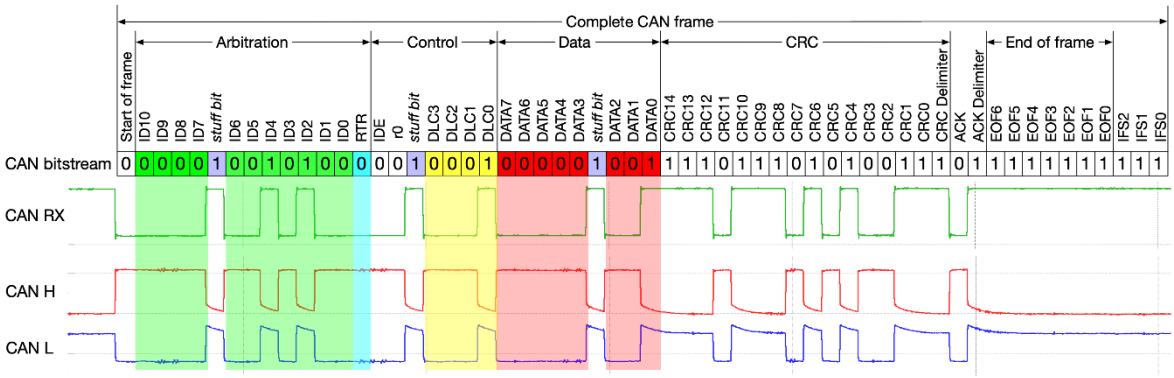
Käesoleva kihi puhul on olulised järgmised nõuded:

1. Süsteem peab olema modulaarne, võimaldades lisada või eemaldada mooduleid ilma keeruka ümberseadistuseta
2. Süsteem ei tohi nõuda erinevate juurutuste korral kohandatud tarkvara
3. Andmeside peab toimima dünaamiliselt ja töökindlalt
4. Süsteem peab olema hõlpsalt laiendatav
5. Anduriplaat peab toetama ka keerukaid andureid
6. Sama tüüpi anduriplaatidel tuleb kasutada sama tarkvara

Nendest nõuetest lähtuvalt tuletati järgmised andmesidetasandi tingimused:

1. CAN-sõnumis peab olema tuvastatav anduriplaadi tüüp
2. CAN-sõnumis peab olema tuvastatav konkreetne anduriplaadi instants
3. CAN-sõnum peab sisaldama anduri andmeid

CAN-sõnum koosneb 8-bitisest identifikaatoriväljast ja 8-baidisest andmeosast (vt joonis 4) [5]. Autor jagas identifikaatoriosa kaheks lõiguks: 6 bitti anduriplaadi tüübi määramiseks ja 5 bitti konkreetse anduriplaadi eksemplari eristamiseks.

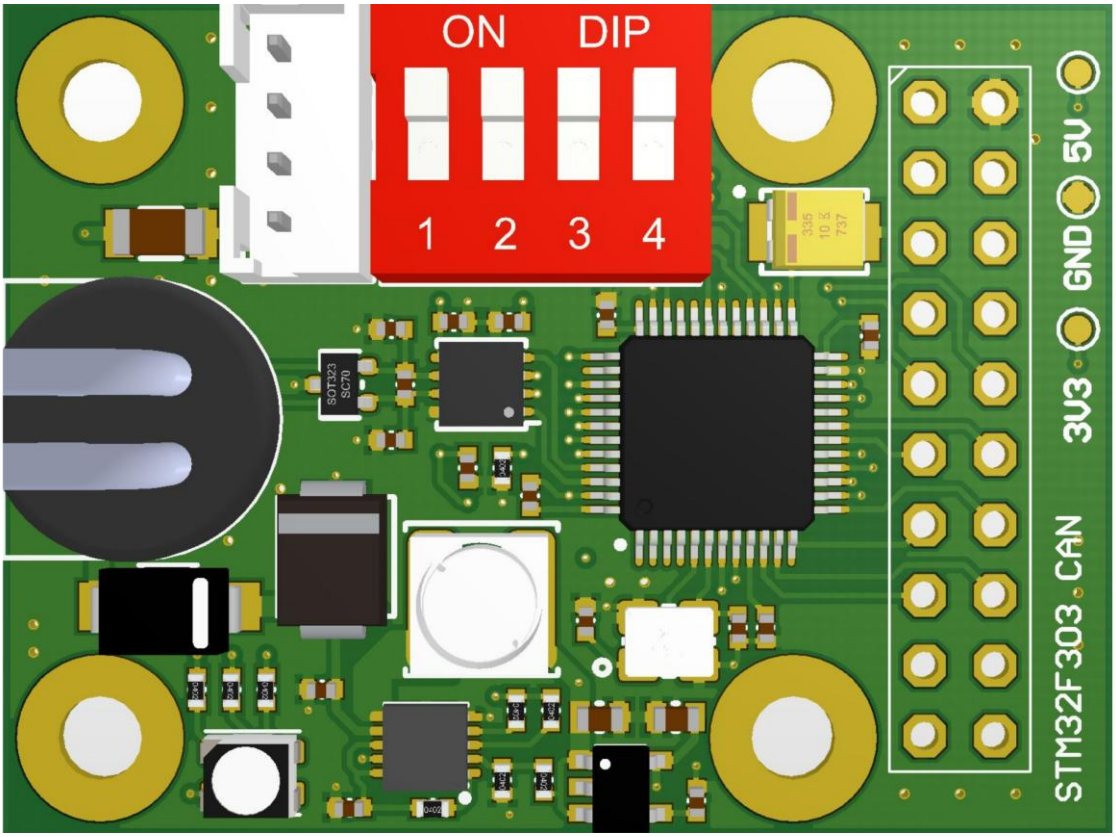


Joonis 4. CAN-siini andmeraami visualisatsioon [13]

6-bitine lõik on mõeldud anduriplaadi tüübi kodeerimiseks. Iga anduriplaadi tüüp saab projekti andmebaasis unikaalse koodi. Niimoodi saab CAN-sõnumi vastuvõtja igast

sõnumist tuvastada, millise anduriplaadi tüübiga on tegu. See võimaldab hõlpsalt lisada süsteemi uusi anduriplaatide tüüpe, ilma et oleks vaja tarkvara spetsiaalselt kohandada.

5-bitine lõik on ette nähtud konkreetse anduriplaadi eristamiseks. Anduriplaatide riistvaraline kujundus võimaldab lõppkasutajal seadistada 5 DIP-lülitit, millest igaüks võib olla kahes asendis (vt joonis 5). Nii saame 2^5 = 32 unikaalset sama tüüpi anduriplaati ühe CAN-siini kohta. See tähendab, et vajadusel on võimalik sama tüüpi anduriplaate liita süsteemi mitmekümne kaupa, ilma eraldi tarkvaralist seadistamist nõudmata.



Joonis 5. DIP-lülitid anduriplaadi instantsi eristamiseks [14]

Järgnevalt esitatakse inseneriarvutused koos vastavate valemite ja arvutuskäikudega, mis aitavad hinnata CAN-protokolli andmesidestruktuuri vastavust projekti vajadustele.

Anduriplaadi tüübi eristamiseks ettenähtud bittide arv:

$$n_{tüüp} = 6 \tag{2.1}$$

Võimalike tüüpide koguarv on kahekordne 6. astmes:

$$N_{tüüp} = 2^{n_{tüüp}} = 2^6 = 64 \tag{2.2}$$

Seega on maksimaalselt võimalik kasutada kuni 64 erinevat anduriplaadi tüüpi.

Anduriplaadi konkreetse eksemplari eristamiseks ettenähtud bittide arv:

$$N_{instants} = 2^{n_{instants}} = 2^5 = 32 \tag{2.3}$$

Igal anduriplaadi tüübil saab seega olla kuni 32 eksemplari.

Maksimaalne teoreetiline anduriplaatide koguarv ühe CAN-siini ulatuses on eri tüüpi anduriplaatide koguarvu ja nende instantside koguarvu korrutis:

$$N_{max} = N_{tüüp} \times N_{instants} = 64 \times 32 = 2048 \tag{2.4}$$

Seega on arhitektuurselt võimalik hallata kuni 2048 anduriplaati ühe CAN-siini kaudu (teoreetiline väärtus, mida praktikas piiravad füüsikalised ja elektroonilised omadused). Iga CAN-sõnum sisaldab 8 baiti (64 bitti) andmeosa. Oletame, et üks mõõtetulemus on 16 bitti (2 baiti):

$$n_{mõõte} = \frac{8 \text{ baiti}}{2 \text{ baiti/tulemus}} = 4 \text{ mõõtetulemust/sõnum} \tag{2.5}$$

Seega mahub ühte CAN-sõnumisse kuni neli 16-bitist mõõtetulemust. Vajadusel saab suuremaid andmehulkasid jagada mitme sõnumi vahel.

Oletame, et CAN-siini bitikiirus on:

$$v_{CAN} = 125 \text{ kbit/s} \tag{2.6}$$

Ühe sõnumi pikkuse hinnang sõltub täpsest kaadri struktuurist, kuid eeldades, et üks sõnum koos üldandmetega jääb mõnesaja biti piiridesse, on võimalik ligikaudselt leida sekundis edastatavate sõnumite arv:

$$N_{\frac{sõnum}{s}} \approx \frac{125000 \text{ bitti/s}}{200 \text{ bitti/sõnum}} = 625 \text{ sõnumit/s} \qquad (2.7)$$

Tegu on konservatiivse hinnanguga, mis näitab, et CAN-siin suudab edastada sadu sõnumeid sekundis, mis on enamasti piisav perioodiliste tööstuslike mõõteandmete vahendamiseks. Kiiruse tõstmine (näiteks 250 kbit/s või 500 kbit/s) suurendaks veelgi süsteemi läbilaskevõimet.

Esitatud inseneriarvutused näitavad, et valitud CAN-põhine andmesidestruktuur on paindlik, laiendatav ja suudab toetada mitut tüüpi ning suurt hulka sama tüüpi anduriplaate. Andmeploki maht on piisav enamike tööstusandmete edastamiseks, ning andmekiirus ja läbilaskevõime vastavad tüüpilistele tööstuslikele vajadustele. Niimoodi kinnitatakse, et CAN-protokolli rakendamine valitud konfiguratsioonis vastab süsteemi pikaajalistele eesmärkidele.

2.2.4 MQTT-protokolli andmeside disain

Peaplaadi ja rakendusserveri vaheliseks andmeedastuseks otsustati kasutada MQTT-protokolli, mille eelmises versioonis saadud kogemus osutus usaldusväärseks. Kuna varasemate kasutusjuhtude põhjal ei ilmnenud olulisi tõrkeid, leidis autor, et senise protokollivaliku säilitamine on asjakohane. Samal ajal otsustati andmeside struktuuri täiustada, et tõsta süsteemi töökindlust, paindlikkust ja laiendatavust.

MQTT-protokolli toimimiseks on mitmeid keskseid elemente [7]:

1. Teemad (hierarhiline struktuur andmete liigitamiseks)
2. Teenuse kvaliteeditase (QoS), mis määrab sõnumite kohaletoimetamise kindluse
3. Sõnumid (andmeüksused, mida vahendusserver edastab)
4. Sõnumitüübid (kinnitab, millisel viisil andmeid edastatakse)
5. MQTT vahendusserver (korraldab sõnumite edastust avaldajatelt tellijateni)

Erinevalt CAN-põhisest lahendusest, kus anduriplaatide ja peaplaadi sõnumites kasutati sihtotstarbelist identifikaatorit, rakendab MQTT-protokoll teemade hierarhiat. Teemade struktuuri saab kujundada paindlikult, lisades alamteemasid vastavalt süsteemi vajadustele. Iga anduri ja süsteemi komponendi jaoks saab määrata selge aadressi, mis võimaldab vahendusserveril ja tellijatel sõnumid hõlpsasti liiklusse suunata, filtreerida ning hiljem analüüsida.

Eelnevatest nõuetest tulenesid järgmised vajadused andmesidekujunduse osas:

1. MQTT-sõnum peab võimaldama tuvastada konkreetset klienti (st millisele ettevõttele või asukohale antud mõõteandmed kuuluvad)
2. MQTT-sõnum peab võimaldama tuvastada Dimusa lahenduse konkreetset instantsi, kui samal kliendil on korraga mitmeid sarnaseid süsteeme
3. MQTT-sõnum peab võimaldama tuvastada anduriplaadi tüüpi, sest erinevad anduriplaadid võivad edastada erinevat liiki andmeid
4. MQTT-sõnum peab võimaldama tuvastada anduriplaadi instantsi ehk konkreetset eksemplari, kui sama tüüpi plaate on ühe süsteemi koosseisus mitu
5. MQTT-sõnum peab sisaldama mõõteandmeid, mille alusel saab edaspidi teha analüüse, koostada aruandeid või rakendada automaatika funktsioone

Selleks, et täita neid tingimusi, otsustas autor rakendada teemade hierarhiat, milles igal astmel eristatakse vastava taseme tunnust. Niimoodi on kogu vajaminev tuvastusteave teemastruktuuris ning MQTT-sõnum ise piirdub ainult mõõteandmete esitamisega. Autor otsustas kasutada järgmist teemade vormi:

/Kliendi_nimi/Dimusa_instantsi_nimi/Anduriplaadi_tüüp/Anduriplaadi_identifikaator

Selles vormis määrab „Kliendi_nimi" konkreetse kasutaja või ettevõtte; „Dimusa_instantsi_nimi" osutab konkreetsele masina- või seadmerühmale, kus Dimusa süsteem tegutseb; „Anduriplaadi_tüüp" viitab andurite liigile; „Anduriplaadi_identifikaator" aga konkreetsele eksemplarile.

Näiteks, kui kliendi nimi on „TalTech", Dimusa instantsi nimetus „3D_Printer", anduriplaadi tüüp on koodiga „6" (näiteks vooluandur) ja selle konkreetsel anduriplaadil on identifikaator „0100", siis on teema vorm:

/taltech/3d_printer/6/0100

Niisugune ülesehitus võimaldab MQTT vahendusserveril suunata andmed vajalikesse kanalitesse, kus tellijad (nt andmeanalüütika tööriistad, rakendusserverid või kasutajaliidesed) saavad valida, milliste teemadega nad soovivad liituda. Kuna teemastruktuur on hierarhiline, saab liituda nii konkreetse anduri kui ka terve masinakomplekti või klienditaseme teemaga. See lihtsustab süsteemi haldamist, laiendamist ja andmete analüüsimist. Lisaks ei vaja ühe uue anduriplaadi lisamine või eemaldamine ega uue kliendi või seadmekomplekti juurutamine eraldi tarkvaralisi muudatusi – piisab teemade sobivast seadistamisest.

Selline lähenemine tõstab märkimisväärselt Dimusa 3 lahenduse paindlikkust, muutes MQTT-protokolli abil toimiva andmeedastuse lihtsamini hallatavaks, paremini laiendatavaks ning selgemini struktureerituks. Kogu identiteedihaldus toimub teemade tasandil, samas kui sõnumid keskenduvad puhtalt mõõteandmete edastamisele. Nii saavutatakse selge, arusaadav ja tulevikukindel MQTT-andmesidestruktuur, mis toetab Dimusa 3 laiemat arhitektuurilist visiooni.

2.3 Peaplaadi tarkvaraarendus

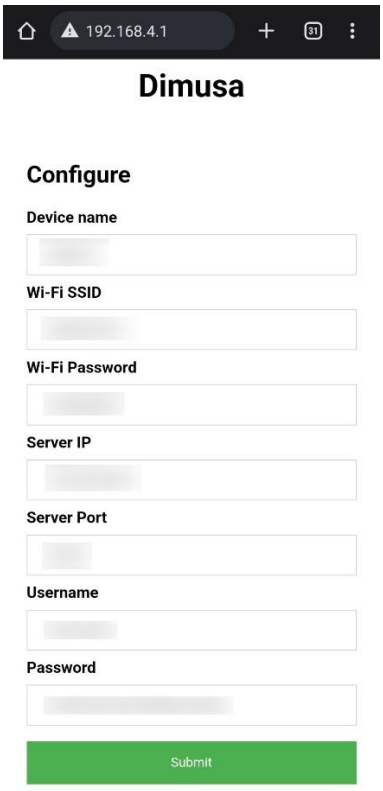
Peaplaadi tarkvaralahenduse aluseks valiti Dimusa eelmise versiooni mikrokontrolleri põhine lahendus. Kuna riistvaraplatvorm jäi samaks, oli võimalik jätkata olemasoleva koodibaasi kaasajastamist ning selle kohandamist uue projekti vajadustele. Tarkvaras kasutati ITT Groupi loodud ITTIoT-teeki, mis hõlbustas mitme olulise funktsiooni, nagu plaadi konfigureerimise ja MQTT-põhise andmeedastuse, elluviimist [15, 16].

Esimene probleem tekkis ITTIoT-teegi ühilduvusega. Kuna teek ei olnud pikka aega hooldatud, ei töötanud see enam tõrgeteta koos uuemate ESP32 teegiversioonidega. Mitmed vanemad funktsioonid olid kasutusest kõrvaldatud või asendatud, mille tulemusel ei õnnestunud olemasolevat koodibaasi kompileerida. Autor parandas teegi koodi, vahetades välja vananenud kutsungid ning kohandades struktuuri nüüdisaegsete nõuetega sobivaks. Nii muudeti ITTIoT-teek taasrakendatavaks, mis võimaldas jätkata arendust juba olemasoleval alusel, ilma täielikult uut raamistiku otsimata.

Teine märkimisväärne probleem puudutas peaplaadi konfiguratsiooniprotsessi. Varasemas lahenduses pidi lõppkasutaja või hooldustehnik sisenema konfiguratsiooni-režiimi ning ühendama oma seadme (nt nutitelefon) peaplaadi loodud traadita võrguga. Seejärel edastati vajalikud seaded spetsiaalse Androidi rakenduse abil. Paraku ei olnud vana rakendus enam kooskõlas uuemate Androidi versioonidega, samas puudus see ka ametlikest rakenduste poodidest. See piiras märkimisväärselt konfiguratsioonivõimalusi, luues sõltuvuse konkreetsetest seadmetest ning raskendades süsteemi laiemat kasutuselevõttu.

Probleemi lahendamiseks kujundas autor ümber peaplaadi seadistuslahenduse. Nüüdsest serveerib peaplaat ise lihtsat veebilehte, millele saab ligi mis tahes seadmest, millel on veebilehitseja ja traadita võrgu tugi. Kasutaja sisestab vajalikud seaded sellel veebiliidesel (vt joonis 6 ja joonis 7). Andmete edastamiseks valiti lihtne HTTP POST päring, kus parameetrid esitatakse JSON (ingl k JavaScript Object Notation) vormingus. Lisaks lisati veebiliidesele võimalus kuvada kogu konfiguratsiooni JSON-kujul, võimaldades kasutajatel seadistusi hõlpsasti talletada või teisele sarnasele seadmele üle kanda.

Joonis 6. Peaplaadi konfiguratsiooni veebiliidese tutvustuse vaade



Joonis 7. Peaplaadi konfiguratsiooni veebiliidese konfiguratsiooni sisestuse vaade

Kõik uuendused, mille autor ITTIoT-teegis ja konfiguratsioonis tegi, kooskõlastati ka teegi algsete loojatega. Uuendatud teek avaldati teegi ametlikus koodirepositooriumis, et ka teised arendajad saaksid sellestsamast täiendusest kasu ning teek oleks tulevikus hõlpsamini hooldatav.

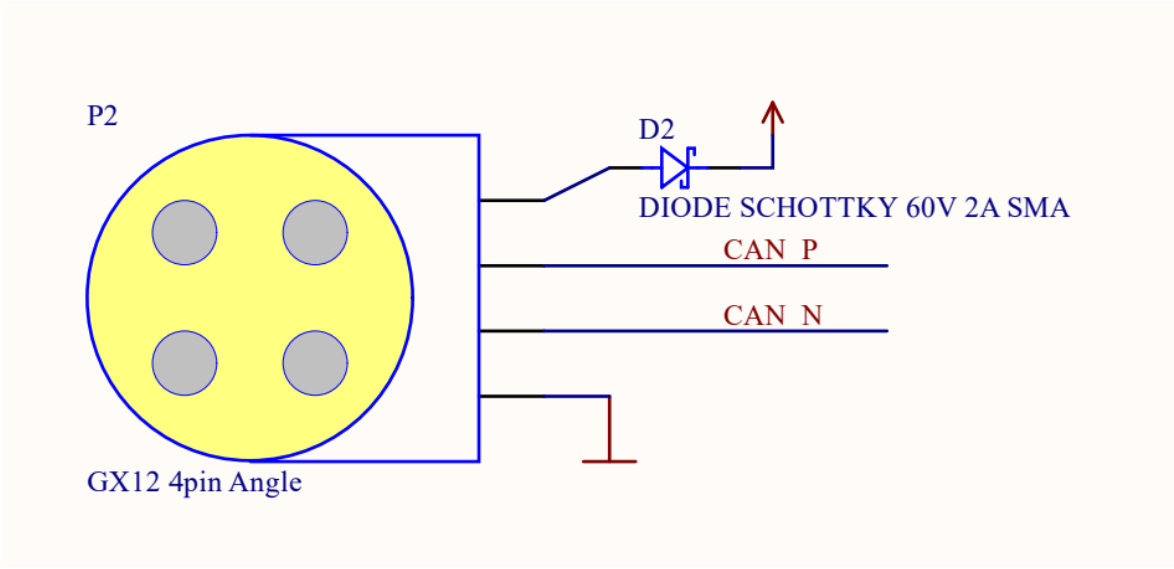
Kokkuvõttes võimaldasid tehtud parandused ja uuendused nii teegi ühilduvuse taastamist kui ka paremat kasutusmugavust ning suuremat sõltumatust spetsiifilistest rakendustest või seadmetest. See tõstis oluliselt nii hooldatavuse kui ka platvormi paindlikkuse taset ning lõi eeldused Dimusa 3 terviklikumaks, kliendisõbralikumaks ja efektiivsemaks rakendamiseks.

2.4 Energiamonitooringu anduriplaadi arendus

2.4.1 Riistvara arendus

Energiamonitooringu anduriplaadi väljatöötamine tõusis esile seniste kasutajate tagasisidest, kus sooviti täpsemat teavet tööstusprotsesside energiatarbimise kohta. See tingis uue Dimusa versiooni arenduses lahenduse, mis võimaldaks mõõta voolutarvet. Projekti meeskond töötas välja anduriplaadi elektroonikalahenduse, mille keskmes on STM32-seeria mikrokontroller. Selline valik annab piisava arvutusvõimsuse ja paindlikkuse erinevate andurite teenindamiseks ning andmete edastamiseks ülejäänud süsteemi.

Anduriplaadi liidestamiseks kogu süsteemiga kasutatakse nelja viigulist ümmargust lukustuvat pistikut GX12 (vt joonis 8). Kaks viiku on pühendatud toite edastamiseks (+5 V ja maandus), ülejäänud kaks viiku aga CAN-siini jaoks. CAN-i liinid jaotati vastavalt CAN-siini kõrgema potentsiaali liiniks (CAN_HIGH) ja madalama potentsiaali liiniks (CAN_LOW). Selline lahendus vastab Dimusa 3 arhitektuurilisele visioonile, kus CAN-siin võimaldab paindlikult lisada või eemaldada anduriplaate, tagades süsteemi laiendatavuse ja töökindluse.

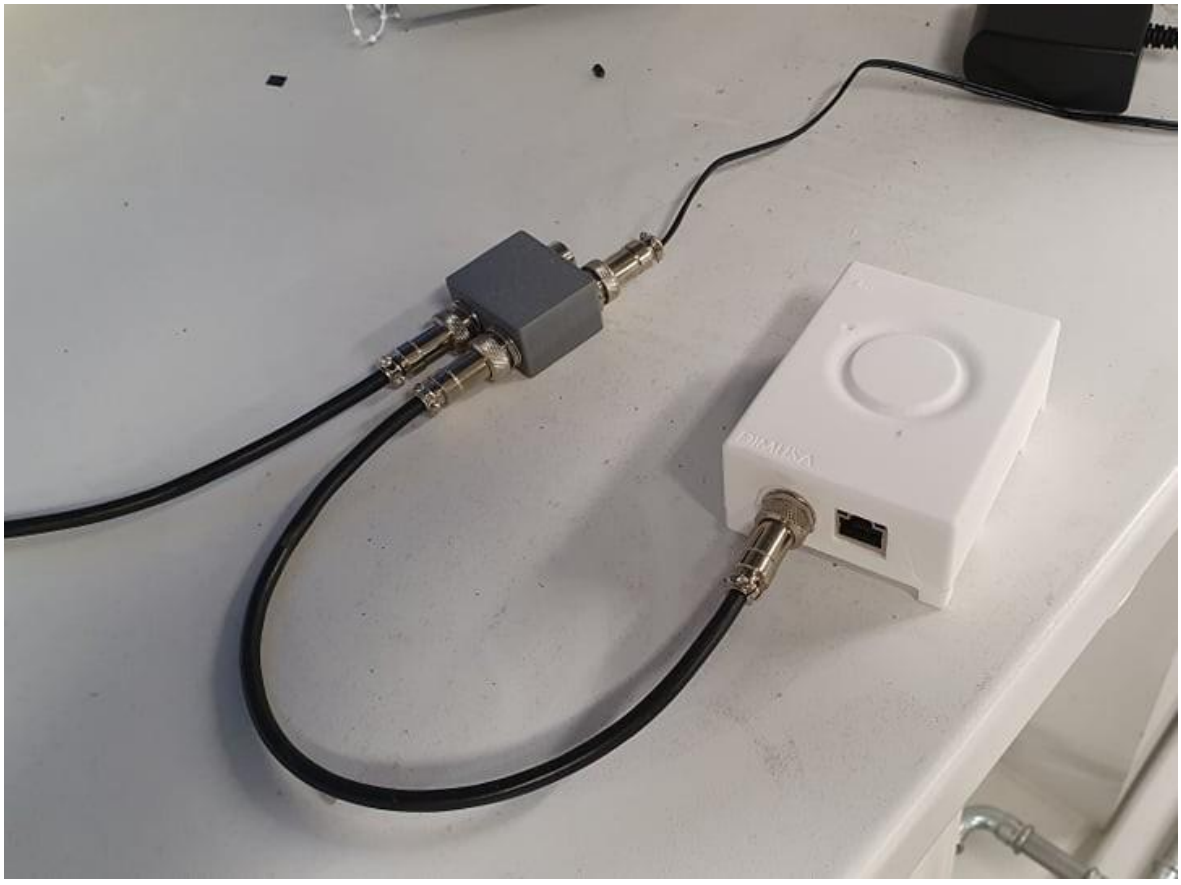


Joonis 8. GX12 pistiku viikude jaotus [14]

Energiatarbe mõõtmiseks on plaadil ette nähtud eraldi sisendid vooluandurite ühendamiseks. Andurite liidestamiseks kasutatakse 3,5 mm ümmargusi pistikuid, mis on selle anduritüübi puhul standartne lahendus. Lisaks vooluanduritele saab

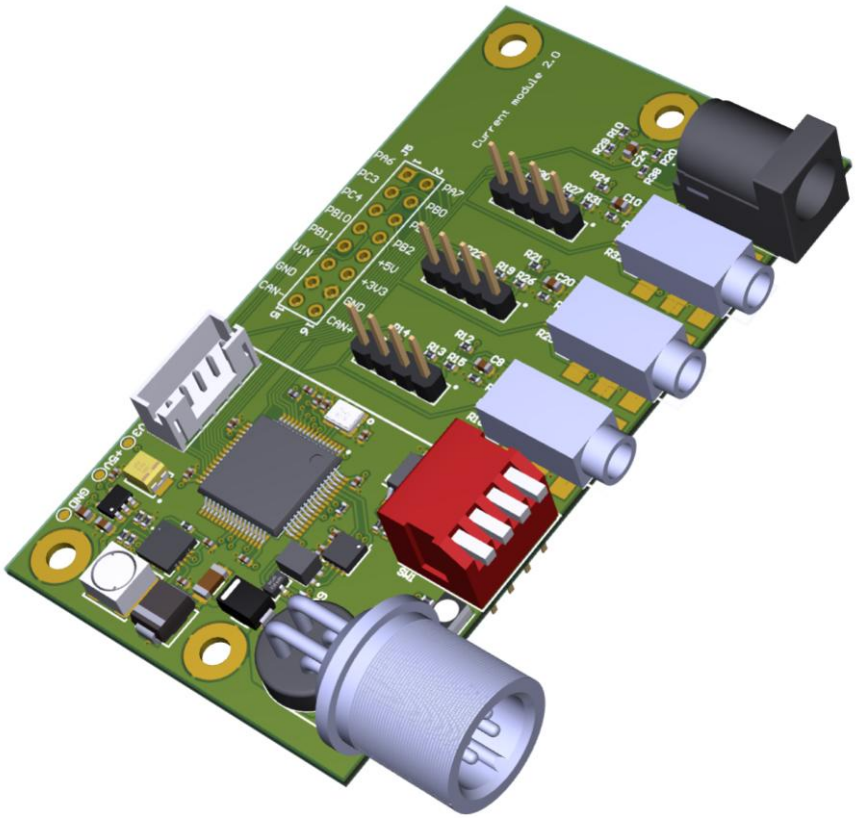
anduriplaadi kaudu mõõta ka pinget. Anduriplaadil paiknevad DIP-lülitid võimaldavad seadistada plaadi unikaalse tunnuse. Nii on võimalik sama tüüpi anduriplaate samas süsteemis selgelt eristada, kui neid on vaja korraga kasutada suuremal hulgal.

Autoril tuli riistvara arendusprotsessis lahendada anduriplaatide omavaheline ühendamine (vt joonis 10). Selleks valmistati vajalikud juhtmed ja toiteseade. Kuna CAN-siinile võib lisada mitu anduriplaati, tekkis vajadus jaoturi järele, mis lubab ühte ühenduspunkti laiendada mitmeks väljundiks. Jaotur ehitati prototüübi vormis: autor disainis ja 3D-printeriga valmistas korpuse, mis koosneb kahest osast. Korpuse sisse paigaldati neli pistikut, mis ühendati omavahel juhtmetega. Nii sündis lihtne lahendus, mida saab vajaduse korral hõlpsasti ümber seadistada või kohandada (vt joonis 9).



Joonis 9. Jaotur ühendatud Dimusa peaplaadiga

Pärast juhtmete, jaoturi ja toiteallika valmimist viis autor läbi esialgse toimivuskontrolli. Testi eesmärk oli veenduda, et nii peaplaat kui ka anduriplaat saavad toite ja lülituvad korralikult sisse. Testprogramm, mis oli eelnevalt plaadile laaditud, vilgutas valgusdioodi, andes märku, et toide ja põhifunktsioonid toimivad. Kuigi andmesidet selles etapis ei kontrollitud, tõestas eduka testi tulemus, et riistvara on töökorras ja valmis edasiseks tarkvaralise osa rakendamiseks.



Joonis 10. Energiamonitooringu anduriplaat [17]

Loodud riistvaralahendus loob eeldused Dimusa 3 süsteemi energiamonitooringu funktsionaalsuse edukaks rakendamiseks. Tänu paindlikule CAN-põhisele arhitektuurile ning lihtsale laiendamisvõimalusele saab samasse süsteemi lisada vajadusel mitu anduriplaati, võimaldades täpsemat, detailsemat ja koondatavat teavet energiakasutusest erinevate tööstusprotsesside lõikes.

2.4.2 Tarkvaraarendus ja integratsioon

Käesoleval peatükis analüüsitakse kahte vabavaralist projekti, mis näivad autorile olevat sobivad lähtekohad energiamonitooringu tarkvara arendamiseks käesoleva projekti raames. Nende valikul arvestati, et lahendus peab toetama kolmefaasilist mõõtmist, võimaldama voolu ja pinget mõõta kuni 40 A ulatuses ning arvutada nii aktiiv-, passiiv- kui ka reaktiivvõimsust ning võimsustegurit. Samuti oli oluline, et tarkvara ühilduks STM32 mikrokontrolleriga, mida uus anduriplaat kasutab.

OpenEnergyMonitor on tuntud vabavaraline projekt, mis keskendub põhjalikule energiatarbe mõõtmisele ja analüüsile. Projekt pakub laialdaselt dokumenteeritud lahendusi kolmefaasilise voolumõõtmise, pingeandmete kogumise ning aktiiv- ja reaktiivvõimsuse arvutamiseks [18]. Samuti toetab OpenEnergyMonitor mitmesuguseid arvutuskäike, mis võimaldavad tuvastada võimsustegurit, mis on tööstuslike energiamõõtmiste juures sageli oluline parameeter [19]. Tegemist on küpse ökosüsteemiga, kus leidub näidisprojekte, tööriistu ja juhendeid STM32 integratsiooniks. Lai dokumentatsioon ning aktiivne kasutajaskond annavad võimaluse lahendada tekkivaid küsimusi ja kohandada tarkvara vastavalt konkreetsetele projektivajadustele. Kõik see loob eelduse sujuvaks üleminekuks olemasolevale baasfunktsionaalsusele ning võimaldab keskenduda eelkõige projekti-spetsiifiliste lisade väljatöötamisele.

IoTaWatt on samuti vabavaraline lahendus, kuid keskendub peamiselt aktiivvõimsuse jälgimisele, jättes tahaplaanile reaktiivvõimsuse, võimsusteguri ning kolmanda faasi mõõtmisega seotud keerukamad arvutused [20]. Kuigi IoTaWatt pakub teatud kasutusjuhtude jaoks piisavaid funktsioone, eeldab kolmanda faasi integreerimine ning reaktiivvõimsuse arvutamine juba märksa suuremat ümbertöötamist [21]. Dokumentatsiooni baas on IoTaWatti puhul kitsam ning projekt ei ole võrreldes OpenEnergyMonitoriga nii paindlikult kohandatav. Seetõttu võib IoTaWatti kohandamine mitmefaasiliseks ning laiaulatuslikke parameetreid mõõtvaks lahenduseks osutuda oluliselt ressursimahukamaks.

Kokkuvõttes pakub OpenEnergyMonitor laiemat funktsionaalsust, paremat dokumenteeritust ning otsesemat tuge kolmefaasilise ja reaktiivse energiatarbe arvutamiseks. Kuna Dimusa 3 arenduses on oluline arendada süsteemi, mis suudab täita komplekssеid mõõte- ja arvutusnõudeid ilma liigse lisaarenduse vajaduseta, on OpenEnergyMonitor kõige otstarbekam lähtekoht. Valides just selle projekti, on võimalik säilitada kõrge töökindlus ja paindlikkus, keskendudes peamiselt kohandustele, mis on vajalikud konkreetse riistvara (STM32 mikrokontroller ja vastav anduriplaat) ning Dimusa süsteemi üldarhitektuuriga sidumiseks.

Pärast sobiva tarkvaralahenduse valimist asus autor OpenEnergyMonitori lähtekoodi Dimusa energiamonitooringu anduriplaadi tarbeks kohandama. Vajalik lähtekood hangiti avalikust GitHubi hoidlast, kus leidus eri andurikonfiguratsioonide jaoks mõeldud tarkvarakomponente. Autor valis alguspunktiks „EMONSTM_3CT" tarkvarapaki, mis toetas kolme eraldiseisva vooluanduri ning ühe pingeanduri kasutamist. Kuigi algne lahendus oli mõeldud võimsamale STM32F4-seeria kontrollerile, otsustas autor sobitada

selle Dimusa anduriplaadil kasutatava tagasihoidlikuma STM32F0-seeria kontrolleriga. Analüüsides STM32F0 mikrokontrolleri mälu- ja sisend-väljundivõimalusi, järeldati, et minimaalse kohandamisega on siirdamine teostatav.

Esimeses etapis muudetigi programm selliselt, et see ühilduks uue riistvaraplatvormiga. Tööprotsess oli iteratiivne: autor viis koodi sisse vajalikud muudatused, laadis programmi mikrokontrollerisse ning kontrollis, kas see käivitub ja suudab mõõtefunktsioone täita. Pärast järjepidevaid kohandusi alustas tarkvara STM32F0-kontrolleril toimimist sellisel määral, et programmi edasine täiendamine oli võimalik.

Järgmiseks lisati tarkvarasse CAN-põhine andmeedastus, mis on Dimusa 3 arhitektuuri keskne suhtluskanal anduriplaatide ja peaplaadi vahel. Autor integreeris standardse STM32 CAN-teegi, rakendades eelnevalt määratletud CAN-sõnumite struktuuri ja tuvastamismehhanisme. See tagas, et andmed, mida anduriplaat töötleb, saavad CAN-siini kaudu edasi liikuda peaplaadini vastavalt projekti arhitektuurilistele põhimõtetele.

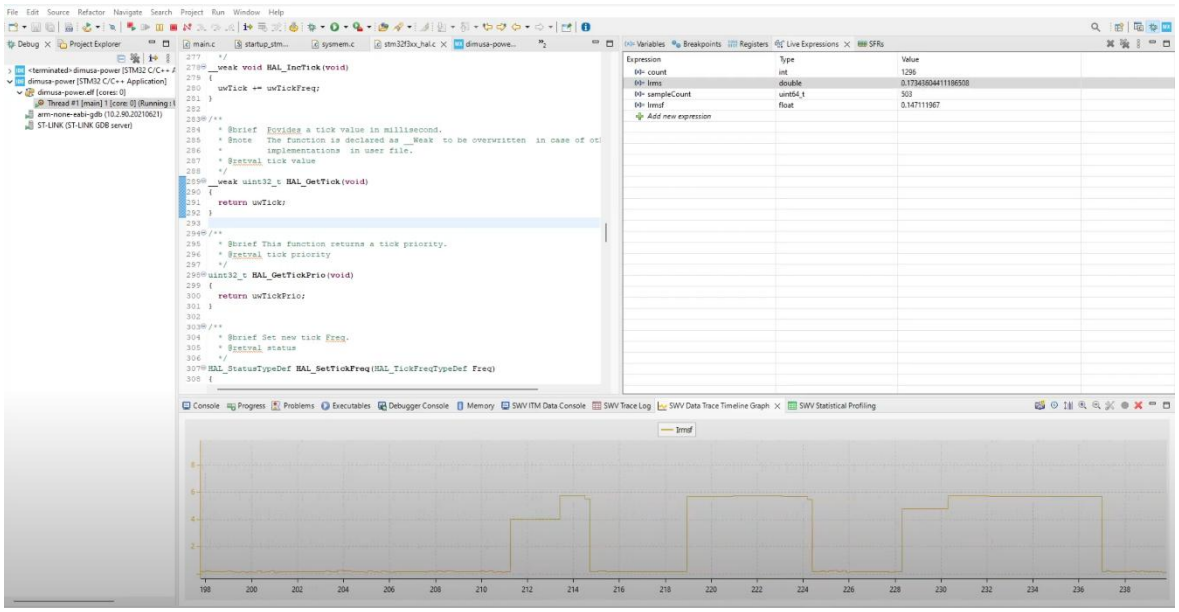
Viimaks seoti mõõtmistulemused CAN-andmeedastusega. Autor lisas tarkvarasse loogika, mis võimaldab igal mõõteperioodil arvutatud tulemused esitada sobivas CAN-sõnumi formaadis. Nii on tagatud, et voolu, pinge ning erinevate võimsuse parameetrite arvutused ei jää üksnes lokaalseks arvutusresultaadiks, vaid neid saab potentsiaalselt ka kaugemas andmeanalüüsi- või visualiseerimiskeskkonnas kasutada.

Kokkuvõttes kujunes integratsiooniprotsessist välja tarkvaralahendus, mis suudab kohandatud STM32F0-seeria mikrokontrolleri baasil töötada koos OpenEnergyMonitori komponentidega ning edastada arvutatud energiatarbe näitajad CAN-siini kaudu edasi Dimusa 3 arhitektuuri.

2.5 Süsteemi testimine ja korrigeerimine

Pärast üksikute süsteemikomponentide väljatöötamist ja integreerimist alustas autor loodud lahenduse toimivuse kontrollimist. Testimise eesmärk oli veenduda, et süsteem vastab eelnevalt püstitatud nõuetele ning on võimeline edastama mõõteandmeid CAN-siini ja MQTT-protokolli vahendusel.

Esimeses etapis keskenduti anduriplaadi põhifunktsionaalsuse kontrollimisele, kasutades selleks vaid ühte vooluandurit. Autor ühendas vooluanduri tavalise koduse veekeetja vooluliini külge, varustas anduriplaadi toitega ning monitooris CAN-siinil edastatavaid sõnumeid USB-CAN adapteri abil. Anduriplaat hakkas viivitamata saatma mõõteandmeid, mis kinnitas CAN põhise sidelahenduse korrektset toimimist (vt joonis 11). Järgmiseks lülitati veekeetja sisse, et kontrollida mõõtmistäpsust. Võrdluseks kasutati ampermeetrit. Mõlema seadme mõõtetulemused langesid piisavalt hästi kokku, et autor võis esimese testi edukaks lugeda.



Joonis 11. Tarkvara testimine

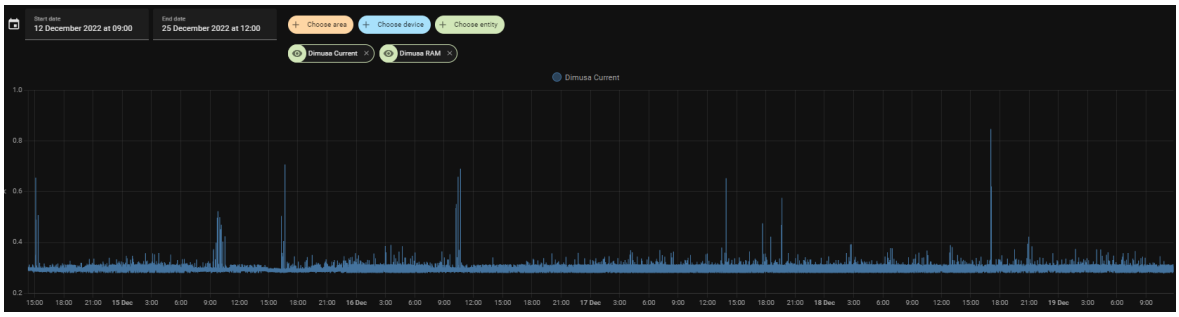
Järgmise sammuna püüti rakendada kogu esialgu kavandatud funktsionaalsust, mis hõlmas kolme vooluanduri, pingeanduri ning vajalike arvutuskäikude (aktiiv-, reaktiivvõimsus, võimsustegur) samaaegset toetust. Selle käigus ilmnes probleem: lisafunktsionaalsuste lisamisel kasvas programmimaht sedavõrd, et see ei mahtunud enam kasutatava STM32F0-seeria mikrokontrolleri mällu. Autor püüdis tarkvara optimeerida, kuid ilma põhitingimusi muutmata ei leidunud rahuldavat lahendust.

Selles olukorras otsustas autor esitada projekti meeskonnale nõude luua anduriplaadist uus versioon, mis vastaks paremini OpenEnergyMonitori referentsdisaini nõuetele ning võimaldaks kasutada võimsamat STM32F4-seeria mikrokontrollerit. Pärast uue anduriplaadi valmimist kordas autor eelmist testi. Seekord mahtus kogu programm edukalt mikrokontrolleri mällu ning anduriplaat suutis CAN-siinil edastada kõik vajalikud näitajad, mis tagas soovitud funktsionaalsuse.

Seejärel ühendati kõik süsteemi elemendid terviklahenduseks ning paigaldati need Tallinna Tehnikaülikooli metalli 3D-printerile (vt joonis 12). Autor kasutas varem loodud konfiguratsiooniliidest, et seadistada võrguühenduse ning MQTT-serveri andmed. Tulemuste kontrollimiseks avati Linuxi keskkonnas MQTT-klienttarkvara, millega oli võimalik reaalajas jälgida vahendusserverisse jõudvaid andmeid (vt joonis 13).

Joonis 12. Dimusa 3 süsteem paigaldatud metalli 3D-printerile

Peale seadistamist ja andmete mõne ajalist monitoorimist ilmnes, et kõik vajalikud mõõteparameetrid jõuavad MQTT-vahendusserverisse. Autor võrdles neid andmeid koostöös 3D-printeri haldajaga. Kontrollkatsed näitasid, et mõõteandmed vastasid reaalsele olukorrale ja eelnevalt seatud nõuetele.



Joonis 13. MQTT vahendusserverisse jõudnud mõõteandmed.

Antud testimisetapis saadud tulemused näitavad, et teostatud lahendus suudab reaalsetes tingimustes edastada mõõteandmeid ning täita projekti eesmärke. Samas ei välistata edaspidist täiustamist või täpsemat kalibreerimist. Käesoleval hetkel võib siiski järeldada, et süsteem on saavutanud toimiva lähtekoha projekti edasiseks arenguks.

2.6 Kasutajaliidese loomine

Projekti lõppeesmärk oli luua terviklik andmehõive- ja visualiseerimislahendus, mis võimaldaks lõppkasutajatel – antud juhul CERN-i projektis osalevatel inseneridel ja tehnikutel – mugavalt jälgida, analüüsida ja hallata mõõteandmeid. Senistes peatükkides käsitletud riist- ja tarkvaralised arendused seadsid aluse toimivale andmekogumissüsteemile, kuid kasutatavus ja väärtus lõppkasutajale sõltuvad olulisel määral sellest, kui selgelt ja arusaadavalt on andmed esitatud. Seetõttu keskendus projekti viimane etapp liidese loomisele, mis võimaldaks andmeid mugavalt visualiseerida, pärida ja jagada.

Kasutajaliidese loomisel tuli arvestada mitmete nõuetega. Esiteks pidi liides pakkuma juurdepääsu andmetele, mis pärinevad erinevatest anduriplaatidest ja mõõtepunktidest. Kuna kogutavad andmed on ajareaga suurused (näiteks vool, pinge, temperatuur või voolutarve teatud ajavahemikus), oli vaja lahendust, mis võimaldaks kiiresti ja efektiivselt salvestada, pärida ja analüüsida ajaloolisi mõõteandmeid. Teiseks pidi liides olema turvaline, skaleeritav ning kergesti hallatav, sest projekti nõudmised võivad aja jooksul muutuda ja laieneda. Kolmandaks pidi visualiseerimine olema kasutajasõbralik, võimaldades hõlpsat ülevaadet süsteemi seisundist ning võimaldades vajadusel ka detailsemaid vaateid.

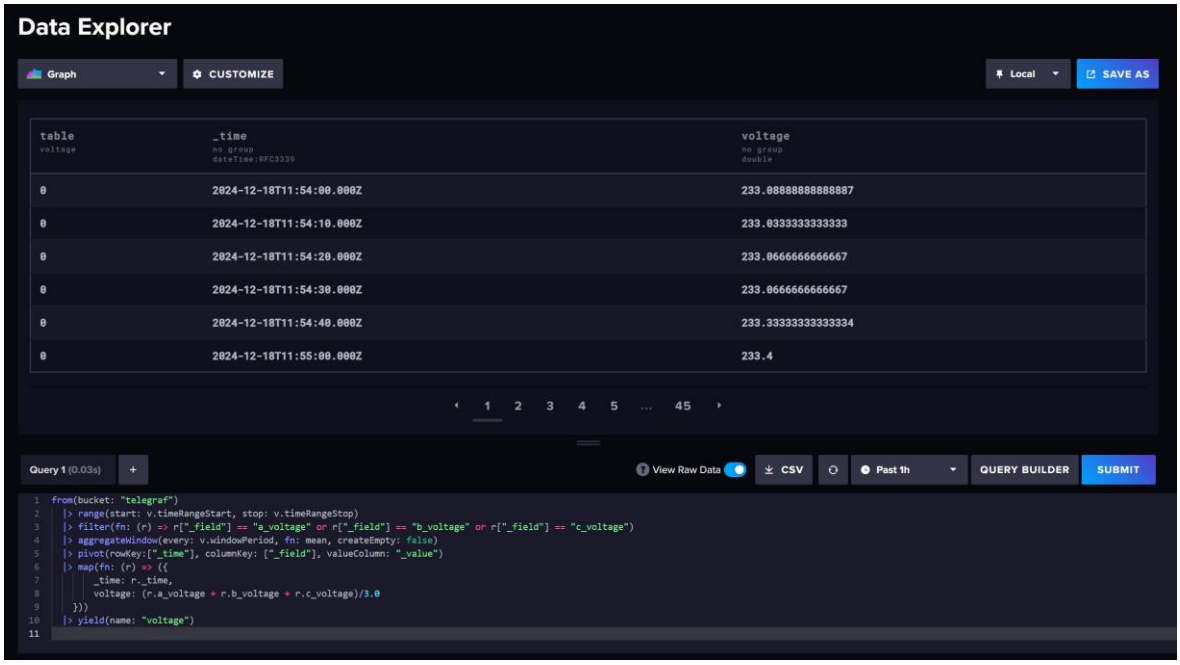
Nende eesmärkide saavutamiseks valis autor välja järgmised komponendid:

1. Andmebaas: InfluxDB, mis on spetsialiseerunud ajareaga andmete kiirele säilitamisele ja päringute töötlemisele. InfluxDB eelis seisneb ka selles, et see toetab hõlpsasti erinevaid visualiseerimis- ja analüüsitööriistu, mis on energiamõõtmise projektide puhul sageli vajalikud [22].
2. MQTT andmevahendus: Telegraf, mis on laialt levinud andmeedastusagent. See võimaldab kuulata kindlaid MQTT teemasid ja edastada saadud andmeid otse InfluxDB-sse. Niimoodi vähendatakse vahendusrakenduste hulka ja lihtsustatakse andmevoogu [23].
3. Visualiseerimise tööriist: Grafana, mis on laialdaselt kasutatav graafilise esituse platvorm. Grafana võimaldab luua töölaudu, mis kuvavad andmeid graafikute, tabelite, mõõdikute ja teiste visuaalsete elementidena. Seda kõike saab kohandada vastavalt konkreetsetele vajadustele, võimaldades lõppkasutajal

valitud ajavahemikke suurendada, päringuid kohandada ning andmetesse süvitsi minna [24].

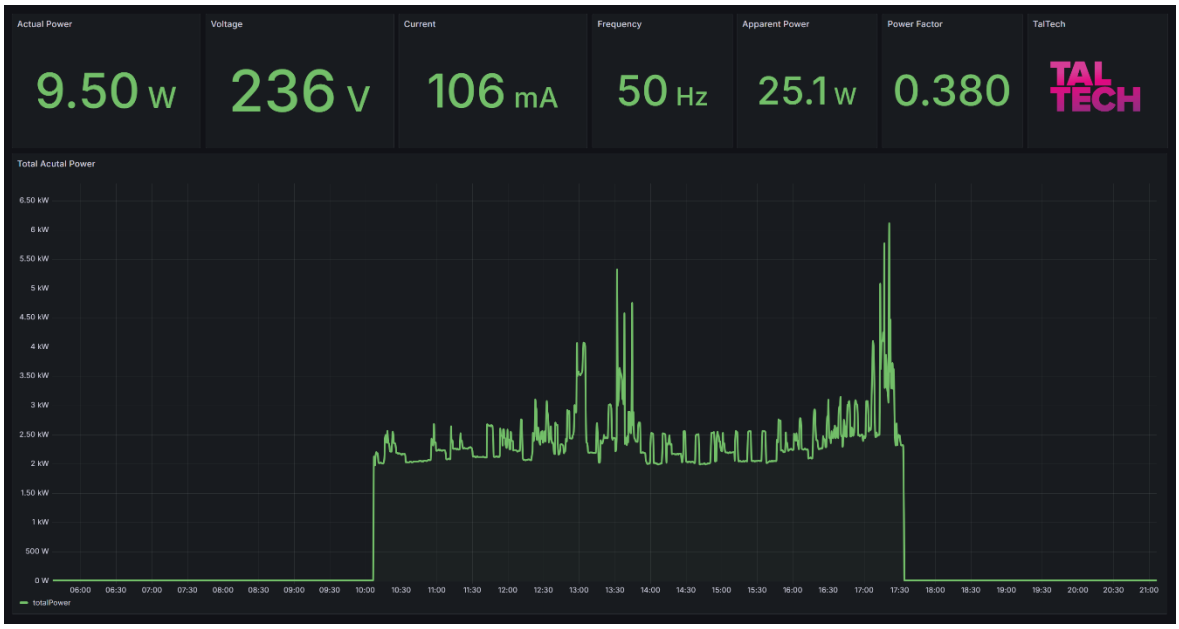
4. Turvalisus ja sertifikaadid: Certbot tagab, et juurdepääs loodud veebiliidesele oleks krüpteeritud ja turvaline. SSL-sertifikaatide kasutuselevõtt on oluline, kui andmeid esitatakse väliskasutajatele või kui soovitakse tagada andmete konfidentsiaalsus [25].
5. Konteinerite haldus: Docker ning Docker Compose kasutamine võimaldab hõlpsasti komplekteerida kõik eelnimetatud komponendid ühtseks lahenduseks. Konteinerpõhine juurutus muudab süsteemi skaleeritavaks, kergesti hallatavaks ja vajadusel teise keskkonda üle kantavaks [26].

Esialgne juurutus tehti Google Cloud Platformi keskkonnas, kus autor sai kiiresti testida ja kohandada lahendust. Pärast konfiguratsioonide seadistamist kontrolliti, kas andmed salvestuvad InfluxDB andmebaasi oodatud viisil (vt joonis 14) [27].



Joonis 14. InfluxDB andmebaasi kasutajaliides

Kui andmete salvestamine oli kinnitatud, asuti Grafana keskkonnas looma visuaalset töölauda. Autor kasutas Grafana intuitiivset veebipõhist redaktorit, et valida sobilikud graafikud, mõõdikud ja filtrid. Visualiseerimist kohandati nii, et see pakuks vajadusel nii üldist ülevaadet süsteemi toimimisest kui ka detailsust huvi pakkuvates valdkondades (vt joonis 15).



Joonis 15. Andmete töölaud Grafana keskonnas

Pärast esialgse lahenduse testimist GCP keskkonnas tuli see üle viia Tallinna Tehnikaülikooli Autonoomsete Sõidukite Labori serveriarvutisse. Seda tehti kulude ja haldusmugavuse kaalutlustel: ülikool eelistab, et süsteemid töötaksid nende enda riistvaral, et vältida püsikulude tekkimist. Docker Compose'i ja konteineriseeritud lahenduse kasutamine võimaldas lahenduse sujuva kolimise teise keskkonda minimaalse ümberkonfigureerimisega.

Kui süsteem oli uues asukohas edukalt käivitatud, lõi autor vajalikud kasutajakontod ja andis neile ligipääsu liidesele. Kasutajate esmane tagasiside oli positiivne: töölaud pakkus selget ülevaadet mõõteandmetest. Mõned detailid vajasid siiski kohandamist, et paremini vastata lõppkasutajate ootustele. Autor tegi vastavad muudatused, näiteks kohandas mõnda mõõdikut, lisas filtreerimisvõimalusi või täiustas visuaalset märgistust. Pärast paranduste rakendamist kinnitas projekti juhtkond, et loodud kasutajaliides on piisavalt informatiivne, lihtne kasutada ja hõlpsasti laiendatav.

Kokkuvõttes kujunes kasutajaliidese loomine edukaks lõpplahenduse osaks, mis võimaldas Dimusa süsteemil täita oma põhieesmärki: pakkuda terviklikku andmehõive-, analüüsi- ja visualiseerimislahendust nii praeguste kui ka tulevaste tööstuslike rakenduste tarbeks.

KOKKUVÕTE

Käesoleva töö eesmärk oli luua modulaarne ja laiendatav tööstuslik IoT-lahendus, mis võimaldab reaalajas koguda, analüüsida ja visualiseerida nii energiakasutuse kui ka laiemate tootmisprotsesside andmeid. Seejuures pidi lahendus vastama keerukate tööstuskeskkondade nõudlikele tingimustele, pakkudes suuremat paindlikkust, skaleeritavust ja integratsioonivõimalusi võrreldes seni kasutatud kommerts- või vabavaraliste lahendustega. Töö vajadus tulenes Dimusa 2 versiooni rakendamisel ilmnenud kitsaskohtadest – piiratud laiendatavusest, keerukast andurite integreerimisest ning andmete piiratud kasutusvõimalustest. Nende probleemide ületamine kujunes Dimusa 3 arenduse põhiülesandeks.

Töö käigus projekteeritud süsteem keskendus andmesidelahenduse parandamisele, kombineerides CAN- ja MQTT-protokolle. CAN-siini kasutamine tagas stabiilse ja madala viivitusega andmeedastuse anduriplaatide ning peaplaadi vahel, lihtsustades moodulite juurde lisamist või eemaldamist. MQTT võimaldas efektiivset andmeedastust kekserverisse, pakkudes paindlikku ja tulevikukindlat lahendust suure hulga andurite ja anduritüüpide haldamiseks. Riistvaraarenduse käigus juurutati energiamonitooringu anduriplaat, mis koos STM- ja ESP-mikrokontrolleritega võimaldas täpselt mõõta energiatarvet, arvutada nii aktiiv- kui ka reaktiivvõimsust ning tuletada võimsustegurit. Sellega loodi alus tootmistegevuse laiemale analüüsile, sealhulgas masina aktiivsuse aegade, tootmistsüklite kestuste, seisakute põhjuste ja kasutuskoormuse hindamisele.

Lahendus integreeriti CERN-iga seotud metalli 3D-printeri jälgimiskeskkonnaga, kus reaalajas andmete kättesaadavus on kriitilise tähtsusega. Andmete salvestamiseks kasutati InfluxDB andmebaasi, mis võimaldas ajareaga andmete kiiret töötlemist. Visualiseerimiseks rakendati Grafanat, mis andis paindliku ja ülevaatliku pildi energiatarbest ning tootmisprotsesside võtmenäitajatest. Kasutajasõbralik konfiguratsiooniliides tagas süsteemi hõlpsa kohandatavuse erinevate tööstuskeskkondade vajadustele. Need sammud lõid eeldused ka tehisintellektil põhinevate analüüsimeetodite tulevaseks rakendamiseks, aidates tuvastada mustreid, avastada anomaaliaid ja pakkuda mudelipõhiseid soovitusi tulemuslikumaks tootmise juhtimiseks.

Testimine näitas, et loodud süsteem suudab usaldusväärselt koguda, edastada ja analüüsida andmeid, pakkudes seeläbi tootmiskeskkondadele reaalajas infot nii energiakasutuse kui ka tootmistegevuste kohta. Kuigi testide käigus ilmnes vajadus täiendavate optimeerimiste järele (nt võimsamate andurite parem tugi,

analüütikafunktsioonide laiendamine), kujutab lõputöö tulemus endast tugevat alust edasiseks arendustegevuseks. Valminud lahendus täiendab Dimusa ökosüsteemi, muutes selle sobivaks suuremas mahus andmete kogumiseks, analüüsimiseks ja visualiseerimiseks ning sillutades teed uute funktsionaalsuste arenduseks ja laienemiseks.

Kokkuvõttes saavutas töö püstitatud eesmärgid: loodi paindlik, laiendatav ja tulevikukindel energiamonitooringu ning tootmisanalüüsi lahendus, mis aitab tööstusettevõtetel optimeerida protsesse, vähendada kulusid ning ühtlasi valmistuda veelgi keerukamate analüüsivõimaluste rakendamiseks.

SUMMARY

The goal of this thesis was to develop a modular and scalable industrial IoT solution capable of collecting, analyzing, and visualizing real-time data related not only to energy consumption but also to broader aspects of the production process. The solution was required to meet the demanding conditions of complex industrial environments, offering greater flexibility, scalability, and integration possibilities compared to previously used commercial or open-source platforms. This need arose from the limitations encountered during the implementation of Dimusa 2, which included restricted expandability, complicated sensor integration, and limited data utilization. Overcoming these challenges became the primary objective in developing Dimusa 3.

During the project, the system was designed to improve data communication by combining CAN and MQTT protocols. The use of a CAN bus ensured stable, low-latency data transmission between sensor boards and the main board, simplifying the addition or removal of modules. MQTT enabled efficient data transfer to a central server, providing a flexible and future-proof solution for managing large numbers of diverse sensors. On the hardware side, an energy monitoring sensor board was deployed and integrated with STM and ESP microcontrollers. This setup allowed for precise measurements of energy consumption, the calculation of both active and reactive power, and the derivation of the power factor. These capabilities laid the groundwork for more comprehensive production activity analysis, including machine uptime, cycle duration, downtime reasons, and load assessments.

The solution was integrated into an environment monitoring a CERN project affiliated metal 3D printer, where real-time data accessibility is critical. Data storage utilized an InfluxDB time-series database for rapid data processing, while visualization was handled by Grafana, providing a flexible and insightful view of energy usage and key production metrics. A user-friendly configuration interface ensured that the system could be easily adapted to various industrial contexts. These steps also paved the way for future AI-based analytical methods to detect patterns, identify anomalies, and provide model-based recommendations for more effective production management.

Testing demonstrated that the developed system can reliably collect, transmit, and analyze data, thus offering real-time insights into both energy usage and manufacturing activities. Although testing revealed a need for further optimizations—such as better support for more powerful sensors and expanded analytical functions—the results provide a robust foundation for continued development. The solution enhances the

Dimusa ecosystem, making it suitable for larger-scale data collection, analysis, and visualization and clearing the path for new functionalities and broader scalability.

In conclusion, the project achieved its objectives: a flexible, extensible, and future-ready solution for energy monitoring and production analysis was created. This solution helps industrial enterprises optimize processes, reduce costs, and prepare for the implementation of even more complex analytical capabilities in the future.

KASUTATUD KIRJANDUSE LOETELU

[1] K. Yasumoto, H. Yamaguchi, and H. Shigeno, “Survey of real-time processing technologies of IoT data streams,” Journal of Information Processing, vol. 24, no. 2, pp. 195–204, 2016, doi: 10.2197/ipsjjip.24.195.

[2] A. Bueno, M. G. Filho, and A. G. Frank, "Smart production planning and control in the Industry 4.0 context: A systematic literature review," Computers & Industrial Engineering, vol. 149, Article 106774, 2020, doi: 10.1016/j.cie.2020.106774.

[3] J. Petrone, "TalTech to Take Part in CERN-Led Accelerator Innovation Project," Research in Estonia, Jun. 1, 2021. [Online]. Saadaval: https://researchinestonia.eu/2021/06/01/taltech-to-take-part-in-cern-led-accelerator-innovation-project/. Vaadatud: Dec. 29, 2024.

[4] R. Cavallini-Rodriguez and P. Portillo-Mendoza, "Design and Implementation of a Monitoring and Alert System Applied to the Residential Electrical Network Using NodeMCU," International Journal of Emerging Trends in Engineering and Technology, vol. 72, no. 6, pp. 125–130, 2024, doi: 10.14445/22315381/IJETT-V72I6P125.

[5] S. Corrigan, Introduction to the Controller Area Network (CAN), Application Report SLOA101B, Texas Instruments, Aug. 2002, revised May 2016. [Online]. Saadaval: https://www.ti.com/lit/pdf/sloa101. Vaadatud: Dec. 29, 2024.

[6] W. Voss, A Comprehensible Guide to Controller Area Network, Copperhill Technologies Corporation, 2008. [Online]. Saadaval: https://books.google.ee/books?id=PU6ppO3XbUwC.

[7] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks," in 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08), Bangalore, India, 2008, pp. 791–798, doi: 10.1109/COMSWA.2008.4554519.

[8] Modicon, Inc., Modbus Protocol Reference Guide, 1979. [Online]. Saadaval: https://www.modbus.org/docs/PI_MBUS_300.pdf. Vaadatud: Dec. 29, 2024.

[9] Silicon Labs, UG103.2: Zigbee Fundamentals, Silicon Labs, 2024. [Online]. Saadaval: https://www.silabs.com/documents/public/user-guides/ug103-02-fundamentals-zigbee.pdf. Vaadatud: Dec. 29, 2024.

[10] F. Tramarin, A. K. Mok, and S. Han, "Real-Time and Reliable Industrial Control Over Wireless LANs: Algorithms, Protocols, and Future Directions," Proceedings of the IEEE, vol. 107, no. 6, pp. 1027–1052, Jun. 2019, doi: 10.1109/JPROC.2019.2913450.

[11] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, Doctoral Dissertation, University of California, Irvine, 2000. [Online]. Saadaval: https://www.epai-ict.ch/nexus/repository/course-material/m133/fielding_dissertation.pdf. Vaadatud: Dec. 29, 2024.

[12] S. Bansode, S. Kawde, H. Shinde, and M. N. Namewar, "Industrial IoT Based Health Monitoring System of Motor and Transformer," Journal of Wireless Networking and Communications, vol. 10, no. 3, pp. 13–19, 2024. [Online]. Saadaval: https://journalspub.com/wp-content/uploads/2024/06/13-19-Industrial-IoT-Based-Health-Monitoring-System-of-Motor-and-Transformer.pdf. Vaadatud: Dec. 29, 2024.

[13] K. Tindell, "CAN bus and CAN security," Ken Tindell Blog, Canis Automotive Labs, Jan. 2020. [Online]. Saadaval: https://kentindell.github.io/2020/01/03/canframe_py_tool/. Vaadatud: Dec. 29, 2024. Vaadatud: Dec. 29, 2024.

[14] H. Pikner, Universal Expansion Module V1.0, unpublished internal document, June 2023.

[15] ITTIoT: ITT Group, ITTIoT Library. Saadaval: https://registry.platformio.org/libraries/robolabor/ITTIoT. Vaadatud: Dec. 29, 2024.

[16] RoboticLab, "ITTIoT raamistik," [Online]. Saadaval: https://www.roboticlab.eu/homelab/et/iot/examples/setup/ittiot. Vaadatud: Dec. 29, 2024.

[17] H. Pikner, Current Expansion Module V2.0, unpublished internal document, March 2024.

[18] OpenEnergyMonitor, "OpenEnergyMonitor ametlik koduleht ja dokumentatsioon." [Online]. Saadaval: https://openenergymonitor.org. Vaadatud: Dec. 29, 2024.

[19] OpenEnergyMonitor, "OpenEnergyMonitor GitHubi repositoorium." [Online]. Saadaval: https://github.com/openenergymonitor. Vaadatud: Dec. 29, 2024.

[20] IoTaWatt, "IoTaWatt ametlik koduleht ja dokumentatsioon." [Online]. Saadaval: https://iotawatt.com. Vaadatud: Dec. 29, 2024.

[21] IoTaWatt, "IoTaWatt GitHubi repositoorium." [Online]. Saadaval: https://github.com/boblemaire/IoTaWatt. Vaadatud: Dec. 29, 2024.

[22] InfluxDB Documentation, "Introduction to InfluxDB." [Online]. Saadaval: https://docs.influxdata.com/influxdb/latest/. Vaadatud: Dec. 29, 2024.

[23] Telegraf Documentation, "Telegraf as a MQTT Consumer." [Online]. Saadaval: https://docs.influxdata.com/telegraf/latest/. Vaadatud: Dec. 29, 2024.

[24] Grafana Documentation, "Introduction to Grafana." [Online]. Saadaval: https://grafana.com/docs/grafana/latest/. Vaadatud: Dec. 29, 2024.

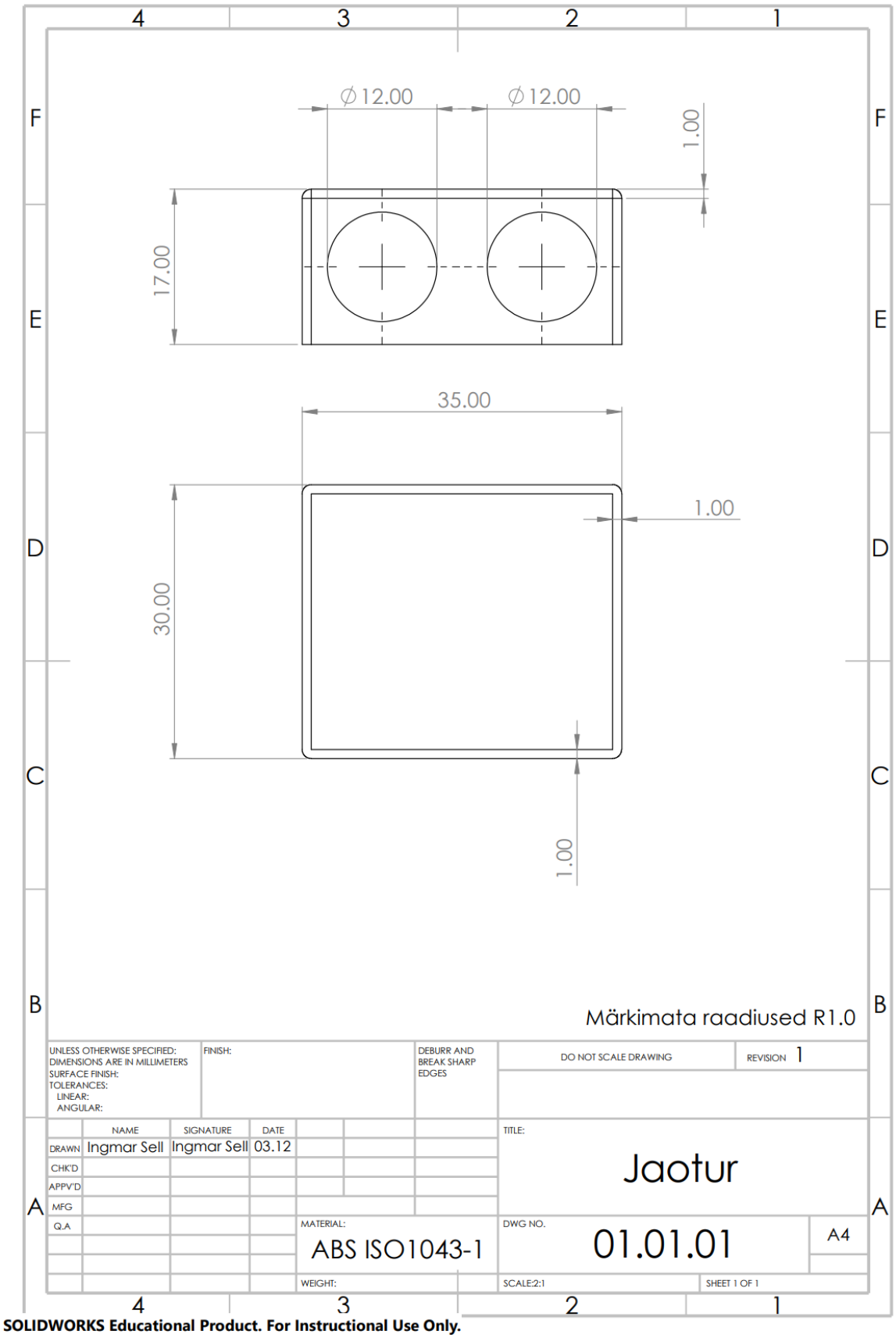
[25] Certbot Documentation, "Certbot User Guide." [Online]. Saadaval: https://certbot.eff.org/. Vaadatud: Dec. 29, 2024.

[26] Docker Documentation, "Overview of Docker Compose." [Online]. Saadaval: https://docs.docker.com/compose/. Vaadatud: Dec. 29, 2024.

[27] D. Denkovski, T. Dimova, and I. Kalendar, "An Innovative Approach of API Automation Testing Implemented on Cloud Environments Using Container Management Services," in Proceedings of the ICAIIT 2024, Halle, Germany, 2024. [Online]. Saadaval: https://opendata.uni-halle.de/bitstream/1981185920/117656/1/4_1_ICAIIT_2024_Part_4_paper_10.pdf. Vaadatud: Dec. 29, 2024.

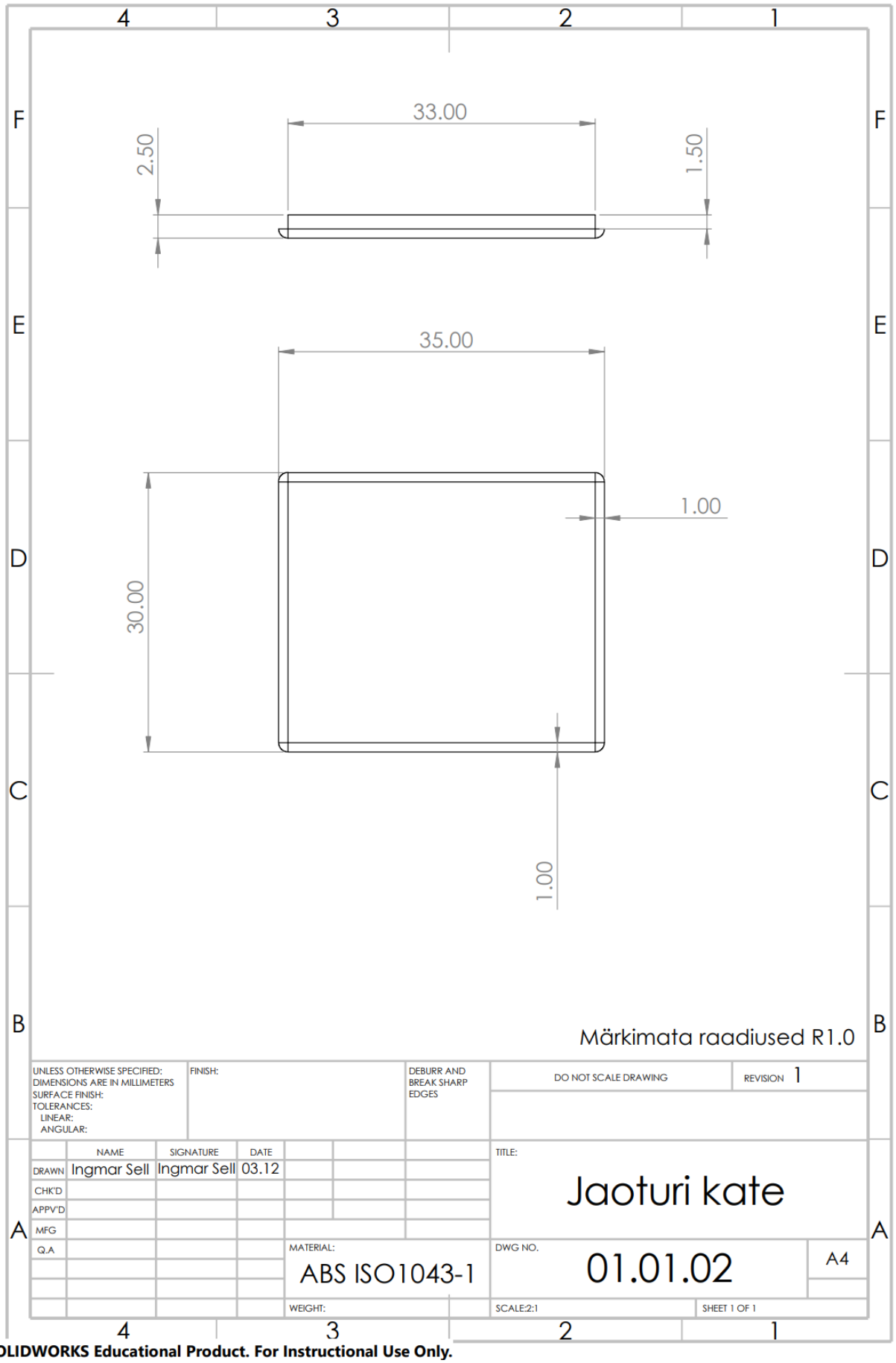
LISAD

Lisa 1 Jaoturi detailijoonis



SOLIDWORKS Educational Product. For Instructional Use Only.

Lisa 2 Jaoturi katte detailijoonis



Lisa 3 Andmesideprotokollide analüüs

Protokoll	**Tüüp**	**Andmeside kiirus**	**Latentsus**	**Skaleeritavus**	**Töökindlus**	**Integreerimise keerukus**	**Energitarve**	**Modulaarsus**	**Peamised kasutusvaldkonnad**
CAN	Juhtmega, jadasiin	Kuni 1 Mb/s (klassikaline CAN)	Madal (reaalajas)	Piiratud (~64 seadet)	Kõrge (tsüklilise korduskontrolli toel)	Keskmine	Madal	Kõrge (lihtne sõlmi lisada/ eemaldada)	Autotööstus, tööstussüsteemid
MQTT	Avaldamis- ja tellimispõhine	Sõltub võrgukiirusest	Keskmine	Kõrge	Keskmine (teenusekvaliteedi tasemed)	Kõrge	Keskmine	Keskmine (dünaamiline klientide tugi)	IoT, telemeetria, kaugjälgimine
Modbus RTU	Juhtmega, jadaside	Kuni 115,2 kb/s	Madal	Piiratud (~247 seadet)	Keskmine	Madal	Madal	Madal (staatiline pearakenduse ja alluvate mudel)	Traditsioonilised tööstuslikud juhtsüsteemid
Zigbee	Traadita mitmepunktiline ühendus	250 kb/s	Keskmine	Kõrge (~65 000 seadet)	Keskmine	Kõrge	Väga madal	Kõrge (lihtne võrgu laiendamine)	Traadita sensorvõrgud, IoT
HTTP/ REST	Päringu-vastuse mudel	Sõltub võrgukiirusest	Kõrge	Kõrge	Madal (olekuta)	Kõrge	Kõrge	Madal (nõuab käsitsi lõpp-punktide seadistust)	Pilveteenustega integreerimine, veebipõhised rakendusliidesed