



TALLINNA TEHNICAÜLIKOOL  
INSENERITEADUSKOND  
Elektroenergeetika ja mehhatroonika instituut

## **EHITUSPLATSI ELEKTRITOITE AUTOMATISEERITUD JÄLGIMINE**

### **AUTOMATED MONITORING OF CONSTRUCTION SITE ELECTRICAL POWER SUPPLY**

BAKALAUREUSETÖÖ

Üliõpilane: Jan Robin Lauk  
/nimi/

Üliõpilaskood: 213713EAAB

Juhendaja: Anton Rassõlkin,  
kaasprofessor tenuuris  
/nimi, amet/

# 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.

"....." ..... 20.....

Autor: .....

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

"....." ..... 20.....

Juhendaja: .....

/ allkiri /

Kaitsmisele lubatud

"....." .....20... .

Kaitsmiskomisjoni esimees .....

/ nimi ja allkiri /

## **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina Jan Robin Lauk (autori nimi)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Ehitusplatsi elektritoite automatiseeritud jälgimine“, (lõputöö pealkiri)

mille juhendaja on Anton Rassõlkin, (juhendaja nimi)

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.

---

13.05.2024 (kuupäev)

<sup>1</sup> 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 loomingu 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.

# LÕPUTÖÖ LÜHIKOKKUVÕTE

Autor: Jan Robin Lauk	Lõputöö liik: Bakalaureusetöö
Töö pealkiri: Ehitusplatsi elektritoite automatiseeritud jälgimine	
Kuupäev: 03.05.2024	
Ülikool: Tallinna Tehnikaülikool	
Teaduskond: Inseneriteaduskond	
Instituut: Elektroenergeetika ja mehhatroonika instituut	
Töö juhendaja(d): kaasprofessor tenuuris Anton Rassõlkin, eriosade spetsialist-käidukorraldaja Lauri Eisler	
Sisu kirjeldus: Lõputöö eesmärgiks on prototüüpseadme(te) loomine ning selle potentsiaali uurimine, mis võimaldaks keerulistes ehitusplatsi keskkonnatingimustes automatiseeritult jälgida elektritoite olemasolu kriitilise tähelepanu alla kuuluvates asukohtades. Prototüüpseadme(te) loomiseks rakendatakse juhtmevaba võrgu kasutamist võimaldavat mikrokontrollerit ning lihtsustatud veebitarkvara testimiste visualiseerimiseks. Lõpptulemuseks esitatakse funktsioneeriv prototüüpseade ning reaalses keskkonnas tehtud katsetulemuste üldistatud analüüs süsteemi toimivuse kohta ehitusplatsi keskkonnatingimustes.	
Märksõnad: mikrokontroller, veebitarkvara, võrk, andmeedastus	

## THESIS SHORT SUMMARY

Author: Jan Robin Lauk	Thesis type: Bachelor's
Caption: Automated Monitoring of Construction Site Electrical Power Supply	
Date: 03.05.2024	
University: Tallinn University of Technology	
Faculty: Engineering faculty	
Institute: Institute of Electroenergetics and mechatronics	
Instructor(s): Tenured Associate Professor Anton Rassõlkin, special operations specialist-manager Lauri Eisler	
Description of contents: The goal of the thesis is to create a prototype device(s) and investigate its potential, enabling automated monitoring of the presence of electrical power in critical locations within challenging construction site environments. A microcontroller that supports wireless networking and simplified web-based software for visualizing test results will be applied to create the prototype device(s). The end result will include a functioning prototype device and a generalized analysis of test results obtained in real-world conditions to evaluate the system's performance in a construction site environment.	
Keywords: microcontroller, web software, grid, mesh, data transmission	

TalTech Elektroenergeetika ja mehhatroonika instituut

## LÕPUTÖÖ ÜLESANNE

**Üliõpilane:** Jan Robin Lauk.....213713EAAB.....(nimi, üliõpilaskood)  
Õppekava, peeriala: EAAB16/17...mehhatroonika.....(kood ja nimetus)  
**Juhendaja(d):** Kaasprofessor tenuuris, Anton Rassõlkin, 6203202 (amet, nimi, telefon)  
**Konsultant:** Lauri Eisler, eriosade spetsialist-käidukorraldaja (nimi, amet)  
Merko Ehitus Eesti AS, 5680 5355, lauri.eisler@merko.ee.(ettevõtte, telefon, e-post)

### Lõputöö teema:

(eesti keeles) Ehitusplatsi elektritoite automatiseeritud jälgimine

(inglise keeles) Automated Monitoring of Construction Site Electrical Power Supply

### Lõputöö põhieesmärgid:

1. Prototüüpseadme elektritoite automatiseeritud jälgimiseks loomine
2. Kasutajaliidese loomine
3. Prototüüpsüsteemi potentsiaali uurimine

### Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Prototüüpseadme elektritoite automatiseeritud jälgimiseks loomine	30.04
2.	Kasutajaliidese loomine	30.04
3.	Prototüüpsüsteemi katsetamine	06.05

**Töö keel:** eesti keel **Lõputöö esitamise tähtaeg:** 13 mai 2024 a

**Üliõpilane:** Jan Robin Lauk ..... ".....".....20.....a  
/allkiri/

**Juhendaja:** Anton Rassõlkin ..... ".....".....20.....a  
/allkiri/

**Konsultant:** Lauri Eisler ..... ".....".....20.....a  
/allkiri/

**Programmijuht:** Marek Tull ..... ".....".....20.....a  
/allkiri/

# SISUKORD

<b>LÕPUTÖÖ LÜHIKOKKUVÕTE .....</b>	<b>4</b>
<b>THESIS SHORT SUMMARY .....</b>	<b>5</b>
<b>EESSÕNA.....</b>	<b>8</b>
Lühendite ja tähiste loetelu .....	9
<b>SISSEJUHATUS.....</b>	<b>10</b>
<b>1. OLEMASOLEVAID LAHENDUSI .....</b>	<b>12</b>
<b>1.1 Elektritoite varustajate lahendused .....</b>	<b>12</b>
1.1.1 Faasimõõtur.....	12
1.1.2 Elektriarvestid .....	13
1.1.3 Seadmete seisukorra jälgimine.....	13
<b>1.2 Tarbija võimalused elektritoite jälgimiseks .....</b>	<b>14</b>
1.2.1 PowerTag moodulilaiend .....	14
1.2.2 Aqara SP-EUC01 nutipistik.....	15
<b>1.3 Olemasolevate lahenduste sobivusest .....</b>	<b>16</b>
<b>2. PLANEERITUD LAHENDUS .....</b>	<b>18</b>
<b>2.1 Arendatava süsteemi tööpõhimõte .....</b>	<b>18</b>
<b>2.2 Prototüüpseade .....</b>	<b>19</b>
<b>2.3 Kasutajaliides .....</b>	<b>20</b>
<b>3. LAHENDUSE REALISEERIMINE .....</b>	<b>22</b>
<b>3.1 Prototüüpseade .....</b>	<b>22</b>
3.1.1 Mikrokontrollerite programmeerimine.....	23
3.1.2 Seadme ehitus .....	28
<b>3.2 Kasutajaliides .....</b>	<b>29</b>
<b>4. KATSED REAALSES KESKKONNAS JA TULEMUSED .....</b>	<b>31</b>
<b>4.1 Katse 1 - Tallina Tehnikaülikooli NRG maja .....</b>	<b>31</b>
<b>4.2 Katse 2 - Arter kvartal .....</b>	<b>32</b>
4.2.1 Soojakust ehitusplatsile katse .....	32
4.2.2 Keldrikorruse katse.....	33
4.2.3 Kõrgeima hoone katse .....	34
4.2.4 Kombineeritud katse.....	35
<b>5. ARENDUSVÕIMALUSED .....</b>	<b>36</b>
<b>5.1 Võimalikud lahendused.....</b>	<b>36</b>
<b>5.2 Projekti arendus .....</b>	<b>36</b>
<b>6. KOKKUVÕTE .....</b>	<b>37</b>
<b>KASUTATUD KIRJANDUSE LOETELU .....</b>	<b>38</b>
<b>LISAD LISA 1 JUHTIV-SEADME KOOD .....</b>	<b>39</b>
<b>LISAD LISA 2 JUHTIV-SEADME KOOD.....</b>	<b>52</b>
<b>LISAD LISA 3 KOMPONENDI LM317T ANDMELEHT .....</b>	<b>60</b>
<b>LISAD LISA 4 EHITUSAEGSE ELEKTRIKILBI NÄIDIS.....</b>	<b>61</b>

## **EESSÕNA**

Käesolev bakalaureusetöö on valminud professor Anton Rassõlkini juhendamisel ning Merko Ehitus Eesti AS eriosade spetsialist-käidukorraldaja Lauri Eisleri kaasjuhendamisel.

Merko Ehitus Eesti AS loal viiakse läbi realses keskkonnas Arteri loomelinnaku ehitusplatsil katseid ning uurimisi töö käigus valmistatud prototüüpseadme(te)ga.

Töö sisuks on ehitusplatsi elektritoite olemasolu jälgimine mikrokontrollerite ning kasutajaliidese abil.

Töö võtmesõnadeks on: automatiseerimine, mikrokontroller, ehitus, bakalaureusetöö.



## Lühendite ja tähiste loetelu

GPS – globaalne positsioneerimissüsteem (inglise keeles Global Positioning System)

HTML – dokumendi paigutuse keel, hüpertexti märgistuskeel, veebilehitsejas (inglise keeles HyperText Markup Language)

ESP32 – Espressif Systems-i poolt toodetud mikrokontroller Wi-Fi ja Bluetooth ühilduvusvõimalustega

V – pinge, volt.

VDC – alalispinge (inglise keeles Voltage Direct Current)

VAC – vahelduvpinge (inglise keeles Voltage Alternating Current)

MAC – seadme füüsiline aadress (inglise keeles Media Access Control)

## SISSEJUHATUS

Tänapäeval seadmete kasutamise juures liigutakse üha enam automatiseerituse poole ning andmete kogumine süsteemide tööst on muutumas järjest olulisemaks süsteemide analüüsimiseks, parandamiseks, optimeerimiseks ning riketele kiirelt ja efektiivselt reageerimiseks.

Andmete kogumist elektriseadmete tööst ja seisukorrast kasutatakse valdavas enamuses ettevõtete poolt, kes pakuvad elektritoitega varustamist, kuid jälgimiseks kasutatavad tehnoloogiad on võrdlemisi kallid ning paigaldatud enamasti kõrgepinge- ning keskpingsesüsteemidesse, näiteks alajaamadesse. On olemas lahendusi lõpptarbijatele, kes soovivad jälgida enda madalpinge kohtsüsteemi tarbimist võrgust, kuid antud lahendused ei ole sobivad ehitusplatsi karmidesse keskkonnatingimustesse. Väljakutseteks ehitusplatsil on suured kogused tolmu ja niiskust, palju vibratsioone ning pidevad füüsilised kahjustused seadmetele, s.t. kukkumised, löömised, peale astumised jms.

Iga infrastruktuuri ja hoone ehituse juures on üks olulisemaid aspekte ehitusplatsi elektritoite varustus, alustades esmastest kaevetöödest ning lõpetades viimaste viimistlustega. Toite katkemise korral on suur tõenäosus, et tehtavad tööd tuleb peatada. See aga tähendab omakorda rahalisi kahjusi. Mõnes olukorras võivad rikneda materjalid, mis on vaja uuesti tellida, millega kaasnevad tarneajaga seonduvad kulud, teises olukorras tuleb tehtud töö teha uuesti või viia läbi parandustöid, mis tuleb kinni maksta, kergeimas olukorras tööliste raisatud tööaeg, mille käigus ei saadud tööd teha, kuid lõppkokkuvõttes, mida kauem on elektritoide katkenud, seda suurem on kumuleeruv rahaline kulu. Ehitusplatsi toite jälgimise automatiseerimine aitab drastiliselt vähendada toite taastamiseks kulunud aega ning seeläbi ka toitekatkestustest tulenenud nii otsest kui ka kaudset rahalist kahju.

Lõputöö hõlmab peamiselt praktilist tööd, mis on suunatud prototüübi loomisele ja sellega seotud lihtsustatud veebitarkvara arendamisele. See prototüüpseade ja sellega seonduv tarkvara on keskse tähtsusega planeeritud lahenduse potentsiaali hindamisel ning võimalike nõrkade kohtade väljaselgitamisel ja analüüsimisel.

Praktiline osa hõlmab mitmeid etappe, alates prototüübi kavandamisest ja ehitamisest kuni selle testimise ja optimeerimiseni. Esialgsete ideede ja kontseptsioonide põhjal luuakse prototüüp, mis on võimeline demonstreerima lahenduse põhifunktsionaalsust. Selle protsessi käigus kasutatakse tehnilisi oskusi ning meetodeid, sealhulgas programmeerimist ning mikrokontrolleril baseeruvat elektroonikat. Veebitarkvara arendamine toetab prototüüpseadme funktsionaalsust, toimides kasutajaliidesena füüsilisele seadmele.

Lõputöö eesmärk on valmis saada toimiv prototüüpseade, mis registreerib toitekatkestusi, kommunikeerib võrgustikus olevate teiste prototüüpseadmetega, kogub

kokku andmed ning visualiseerib need veebitarkvaras ning on vaadeldavad ehitusplatsi soojaku kontoriruumides.

Lõputöö tulemusena esitatakse näidistena prototüüpseadmeid ning nende ühilduvust veebitarkvaraga. Tuuakse välja lahenduse puudujäägid ning võimalikud arendused prototüübi parendamiseks ning optimeerimiseks.

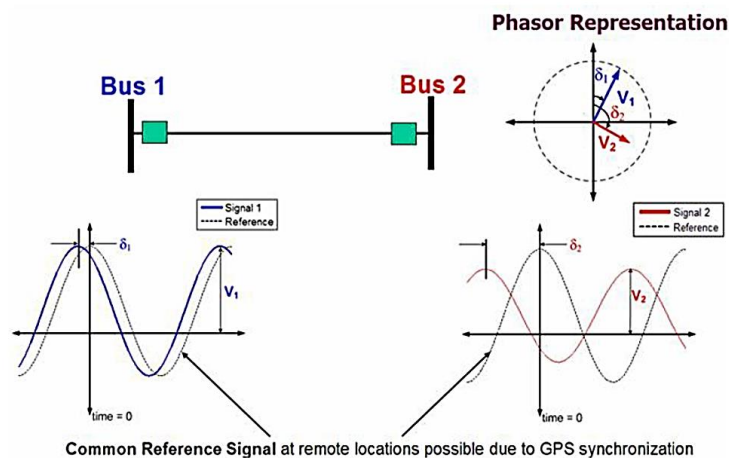
# 1. OLEMASOLEVAID LAHENDUSI

## 1.1 Elektritoite varustajate lahendused

Elektritoite varustajate kasutatud lahendused on mõeldud geograafiliselt suurele mastaabile, mis tähendab, et nende jälgimissüsteemid on loodud jälgimaks üle suure ala ning tulenevalt sellest on need seadmed kallid ning käesoleva töö jaoks üledimensioneeritud. Käesolevas peatükis uuritakse elektrivarustajate poolt paigaldatavate seadmetena faasimõõturit ja elektriarvestit ning analüüsitakse nende sobivust või mitesobivust lõputöö praktilise ülesande lahendamiseks. Valdavas enamuses on elektritoite varustajate seadmetel olemas üle võrgu andmete edastamise võimalus, kuid kasutavad selleks kas rangelt seadustega reguleeritud süsteeme või vajavad kallist infrastruktuuri.

### 1.1.1 Faasimõõtur

Faasimõõtur on seade, mis mõõdab elektriliinil spetsiifilises kohas vahelduvvoolu pinget ja/või voolu faasi [1]. Need on üldjuhul paigaldatud koos nutikate kontrollritega. Faasimõõturitelt tulevad andmed on samuti ühildatud GPS süsteemiga, mis annab kogutud andmetele juurde ka ajamärgise. See võimaldab üle suure geograafilise ala jälgida, võrrelda ning analüüsida erinevate faasimõõturite andmeid, mis annab operaatorile informatsiooni, kas vool, pinge ja sagedus on vastavuses tolerantsidega. Muus osas annab see varustajale võimaluse parendada modelleerimissüsteemi täpsust, ennustada ja märgata ebastabiilsusi toitevõrgus, saada informatsiooni sündmuse analüüsiks peale häire tekkimist, ebaefektiivsuste tuvastamiseks ning liinide ummistumise ennetamiseks ja kõrvaldamiseks.



Joonis 1.1 Faasimõõturite edastatav informatsioon liini erinevates otstes [1]

### 1.1.2 Elektriarvestid

Erinevalt faasimõõturitest, loendavad elektriarvestid tarbitud elektrivõimsust mingi hetke minevikus ja oleviku vahel. On olemas ka seadmeid, mis mõõdavad hetketarbimist, aga spetsiifilist faasimõõturi sarnast faasimõõtmist arvestid ei tee. [2] Arvestite mõõtmistulemusi kasutatakse tarbimise eest arve esitamiseks ning neid saab paigaldada nn hierarhiliselt, sõltuvalt tarbija iseloomust, kas ja kuidas kinnisvaral on elektrivarustus teostatud ning kellelt raha nõutakse.

Kuigi arvestitel on samuti kaugloetavuse võimalus olemas, siis käesolevaks tööks arvestid ei sobi, kuna need ei edasta toite olemasolu kohta informatsiooni, vaid selle koguse kohta. Kui elektrit ei tarbita, ei tähenda see ilmingimata toite katkestust, mida antud ülesande püstituses on vaja jälgida.



Joonis 1.2 Elektriarvesti näidis [3]

### 1.1.3 Seadmete seisukorra jälgimine

Alajaamades asuvate seadmete, näiteks trafode, jälgimiseks kasutatakse erinevaid andureid, näiteks erinevaid temperatuuriandureid või termokaameraid. Neid andureid kasutatakse seadmete seisukorra hindamiseks ja kulumisest tulenevate parandustööde ennetuseks.

Andmevahetuseks kasutatakse peamiselt IEC 61850 sideprotokolli, mis põhineb TCP/IP võrgustruktuuril. Lisaks on vanemates alajaamades kasutusel IEC 60870-5-101 ja IEC 60870-5-103 protokollid, mis põhinevad suuresti järjestiksidel. Kohtterminalide omavaheline side toimub IEC 61850 sideprotokollis sisalduvate GOOSE (inglise keeles General Object Oriented Substation Event) sõnumitega. Andmesidevõrgu paremaks

haldamiseks on horisontaalside (kohtterminalide vahel) jaoks ette nähtud eraldiseisev virtuaalne kohtvõrk (VLAN ehk inglise keeles Virtual Local Area Network). [4]

Kuna varustajate seadmed on füüsiliselt fikseeritud asukohtades, on nende jälgimine lihtsustatud ning peale esmast paigaldust vajavad ideaalkorras vaid korralisi hooldusi. Käesolevas ülesandes aga liiguvad ehitusplatsi kilbid tihti ühest kohast teise, mis muudab eraldiseisvate anduritega süsteemi jälgimise komplitseerituks ning ebakindlaks. Sellest tulenevalt on vaja süsteemi, mis ühilduks suurema võrgustikuga, olenemata asukoha füüsilisest fikseeritusest.

## 1.2 Tarbija võimalused elektritoite jälgimiseks

Eraisikutel on tänapäeval erinevaid võimalusi endale soetada nutikaid seadmeid, mida enda kodus keskkonnas kasutada isikliku elektritarbimise jälgimiseks ning seadmete kaugelt sisse ja välja lülitamiseks. Siiani turul saadavalolevad seadmed, millel on võrguga ühildumise võimalus olemas, vajavad funktsiooni kasutamiseks kas spetsiaalse kaabli vedamist koduse ja/või töökoha ruuteri juurest või ei toeta ehitusplatsi töömaakilpide ülesehitust. Nendest seadmetest uuritakse käesolevas peatükis Schneider Electricu toodet PowerTag ning Aqara SP-EUC01 nutipistikut.

### 1.2.1 PowerTag moodulilaiend

Tegemist on käesoleva bakalaureusetöö parima paralleeliga. Schneider Electricu poolt loodud seade on ühildatav igasse kilpi, ei võta seal palju ruumi ning on võimeline jälgima igat kaitset läbivat võimsust. Lisaks sellele edastab seade informatsiooni nutiseadmesse ning muudab jälgitavuse väga lihtsaks. [5] [6]

Ühtlasi on seadmega kaasnevas tarkvaras võimalik uurida ja jälgida lisaks elektritarbimise maksumust ning kaugjuhitavalt tarbijaid lülitada sisse ja välja.



Joonis 1.3 Schneider Electricu toode PowerTag moodulilaiend [6]

Ühtlasi vajab PowerTag moodulilaiend toimimiseks kilpi sisestatud PowerTag Link [7] seadet, mille külge ühendub elektrikilpi veetud andmeedastuse kaabel.

Mõlema komponendi, nii PowerTag moodulilaiendi kui ka Link seadme hinnad jäävad 150 euro juurde. Tegemist on võrdlemisi kuluka süsteemiga, mille paigaldamine oleks tarbekas vaid ohutus keskkonnas.



Joonis 1.4 Schneider Electricu toode Acti9 PowerTag Link [7]

Kuigi seade toetab kilbiseseselt erinevate PowerTag seadmetevahelist juhtmevaba võrguühendust, on kilpi paigaldatud PowerTag Link andmeedastuse funktsiooni toimimiseks vajalik siiski elektrikilpi vedada internetikaabel. Internetikaabel, näiteks CAT-6, on 8-sooneline peenike ja habras kiudkaabel, mis ei talu palju väänamist, väntsutamist ega pealeastumist. Ühtlasi muudavad ka väikseimad sisselõiked kaabli kasutamatuks. Mainitud situatsioone esineb ehitusplatsi keskkonnas rohkem kui piisavalt, mille tõttu ei oleks tarbekas vedada internetikaablit või kulutada ajalist ja ka rahalist ressursi selle kaitsmiseks.

### 1.2.2 Aqara SP-EUC01 nutipistik

Aqara nutipistik on kasulik seade, mis võimaldab ühendada elektriseadmeid seinakontakti ning samal ajal jälgida nende elektritarbimist ja juhtida nende sisse- ja väljalülitamist [8]. See on osa laiemast Aqara nutiseadmete ökosüsteemist, mis kasutab Zigbee protokollit, võimaldades ühildumist teiste Aqara toodete ja paljude teiste Zigbee-võrgus töötavate seadmetega. See teeb Aqara nutipistikust mitmekülgse ja mugava valiku koduautomaatika süsteemides.

**Funktsioonideks** on Aqara nutipistikul elektrienergia ning tarbimise mõõtmine ühendatud seadmete kaudu. See on kasulik energiakulu jälgimiseks ja aitab tuvastada, kui palju elektrit erinevad seadmed kasutavad. Nutipistik võimaldab kaugjuhtida ühendatud seadmete sisse- ja väljalülitamist, mis on mugav näiteks ajastamise või nutikodu automatiseerimise korral.

Piiranguteks ja probleemideks on Aqara nutipistikul ühefaasiline disain ning seda saab ühendada tavalistesse ühefaasilistesse pistikupesadesse. See piirab kasutamist kohtades, kus kasutatakse kolmefaasilist elektrisüsteemi, näiteks tööstushoonetes või

ehitusplatsidel. Seadme maksimaalne koormus on 16 amprit. See võib olla piisav koduseks kasutamiseks, kuid võib osutuda piiravaks suurema tarbimisega seadmete või tööstuslike rakenduste puhul. Ehitusplatsidel ja suurema koormusega keskkondades võib olla tavaline, et seadmed ületavad selle piiri.



Joonis 1.5 Aqara SP-EUC01 nutipistik [9]

### 1.3 Olemasolevate lahenduste sobivusest

Käesolevas peatükis mainitud seadmed ei sobi ehitusplatsi elektritoite jälgimiseks erinevatel põhjustel. Selleks on välja toodud igale seadmele nende juurde käivad põhjused.

**Faasimõõturid** on mõeldud ajaliselt sünkroniseerima kogutud andmeid geograafiliselt riigisuuruses mastaabis. Andmete sünkroonimist ajas ei ole aga ehitusplatsil vaja ning see muudab suure osa faasimõõturi tööst ebavajalikuks. Lisaks on faasimõõturid ja nende paigaldus väga kallis teenus, ulatudes alates 23000 eurost kuni 280000 euroni, sõltuvalt süsteemist. [10]

**Elektriarvestid** tänapäeval, millel on andmeedastusfunktsioon, vajavad kilpi andmesidekaablit, mida ehitusaegses elektrivõrgustikus ei eksisteeri ning paigaldamine ei oleks mõttekas selle kaabli enda ning selle kaitsmiseks kuluvate rahaliste ressursside tõttu. Lisaks on elektriarvestid kallid, olles oma hinnaga ligikaudu 150 euro juures.

**PowerTag ja PowerTag Link** vajavad sarnaselt elektriarvestitele kilpi andmesidekaabli vedamist. Lisaks vajab süsteem toimimiseks mõlemat komponenti, millede hinnad jäävad, sarnaselt elektriarvestile, 150 euro kanti, mis teeb ühe kilbi süsteemi maksumuseks ligikaudu 300 eurot.

**Aqara nutipistik** mõõdab andmeid vaid ühest ühefaasilisest pistikupesast, mis ei ole piisav kogu ehitusplatsi elektritoite jälgimiseks. Ehitusaegsed kilbid on kõik kolmefaasilise sisendiga ning selle jälgimiseks on vaja seda jälgida juba sisendis, mitte suvalist ühefaasilist pistikupesa kilbi küljes. Lisaks puudub neil ka andmeedastuse



kaugus, et mõõdetud andmeid ehitusplatsi soojakus analüüsida. Tulenevalt ka pistiku ppaigaldusviisist, on seda väga lihtne lahti ühendada ning ära varastada, mille tõttu ei ole mõttekas neid ehitusplatsile tuua.

## **2. PLANEERITUD LAHENDUS**

Käesolevas peatükis kirjeldatakse lahenduse tööpõhimõtet, prototüüpseadme planeeritud skemaatikat ning veebitarkvaraks kasutatavaid lahendusi.

### **2.1 Arendatava süsteemi tööpõhimõte**

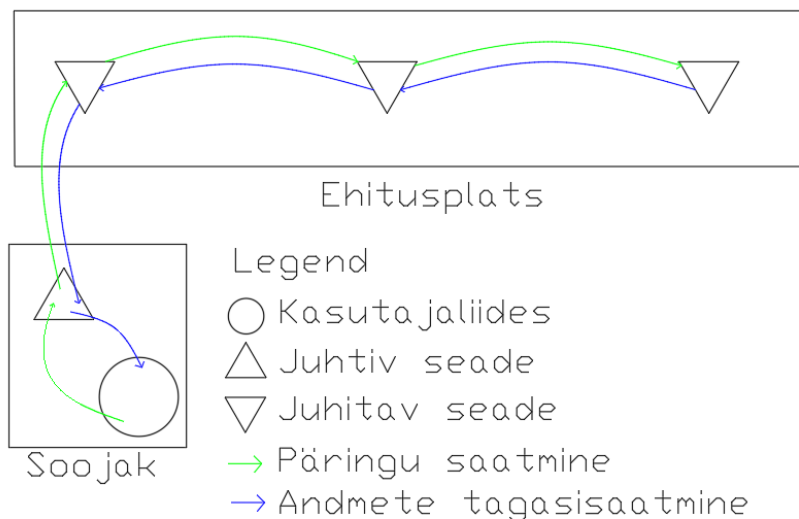
Planeeritud lahenduse tööpõhimõte on luua mikrokontrolleritel baseeruv juhtmevaba võrku kasutatav seadmete võrgustik.

Kuna ehitusplatsi keskkond on ohtlik erinevatele tundlikele kaablitele nagu võrgukaabel, tuleb lahenduseks kasutada juhtmevaba süsteemi. Selle jaoks tuleb kasutada mitut seadet, mis toimiksid signaali kordajatena, et signaal jõuaks sinna, kuhu vaja.

Selleks tuleks ehitusplatsi soojakusse üles seada juhtiv-seade, kus puuduvad ehitusplatsi keskkonnaohud ning mis ühilduks soojaku internetivõrku. Päringu saatmisel peaks juhtiv-seade päringu edastama lähimasse juhitav-seadmesse, mida see tuvastab, ning see omakorda selle poolt tuvastatud juhitav-seadmetesse, jne. Päring peaks endas sisaldama ehitusplatsile üles seatud seadmete tuvastusandmeid, et nendele juurde kirjutada kilbi toite olemasolu seisukord. Kui seade saab päringu, kontrollitakse seadme sisendist, kas elektrikilbi kaitsme väljundist tulev signaal on aktiivne või mitte. See info komplekteeritakse seadme tuvastusandmetega ning kirjutatakse logisse. Jõudes ehitusplatsil viimase kilbini, s.t logis olevatel kilbi tuvastusandmetele on kõigile juurde kirjutatud seisukorra andmed. Peale seda tuleb logi sama võrgustikku pidi tagasi saata, et see jõuaks juhtiv-seadmesse.

Juhtiv-seade saab päringu veebitarkvaralt, mis annab päringu korral kaasa ka tuvastusandmete logi, mis kirjutatakse valmis süsteemi üles seades. Kui päringu tagasijõudmisel juhitav-seadmetelt on logi kirjutatud täis, kuvatakse logisse kirjutatud andmed veebitarkvara abil kasutajale, kes saab saadud infoga vajadusel võtta ette vastavad tegevused. Veebitarkvara on ühtlasi ka kasutajaliideseks kasutajale.

Lisaks tuleks seadmed seadistada nii, et kilbi kaitsme rakendumise intsidendi korral, peaks vastav seade saatma mööda võrgustikku selle kohta signaali ning ootama juhtiv-seadmelt vastust saadud signaali kohta.



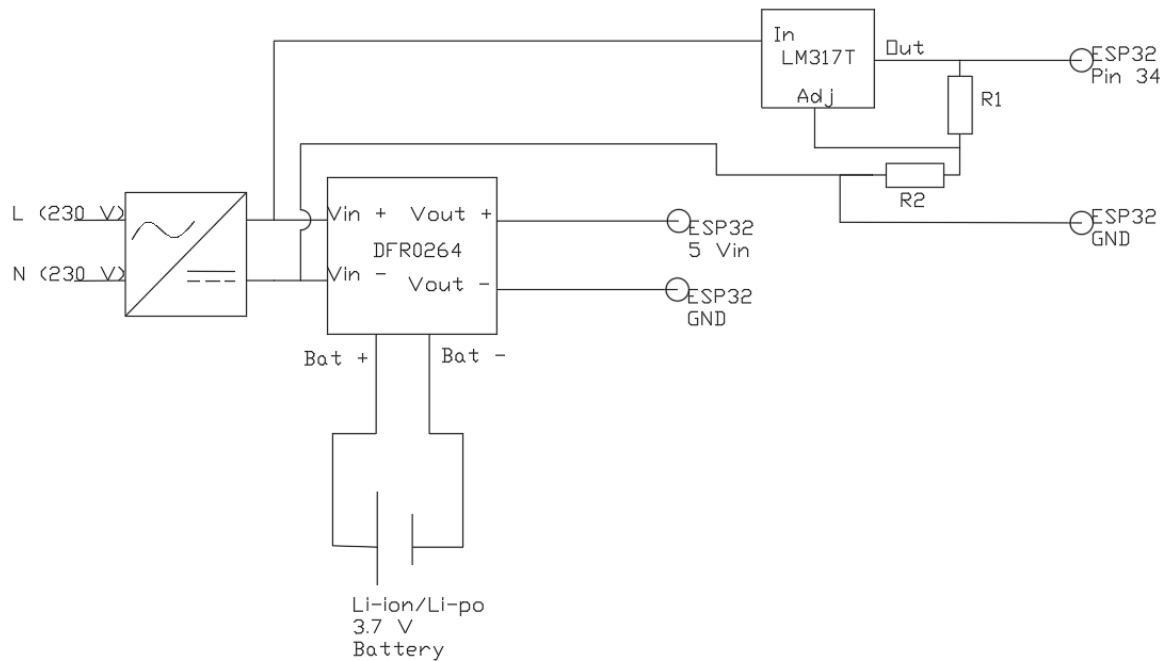
Joonis 2.1 Planeeritud lahenduse lihtsustatud funktsionaalskeem

## 2.2 Prototüüpseade

Prototüüpseadme põhikomponendiks andmete käitlemiseks ning juhtmevabalt edastamiseks on plaanitud kasutada mikrokontrollerit ESP32 [11], millel on juba külgeehitatud juhtmevaba võrgu kasutamise võimalus. Mikrokontrolleri toitepinge on 5 VDC.

Ehitusplatside ehitusaegsete elektrikilpide toitepinge on 230 VAC. See tähendab, et mikrokontrolleri jaoks on vaja pinget alaldada ning teisendada 5 VDC peale. Selleks on plaanitud kasutada telefonilaadijat (väljundiga 5 VDC) ja pingestabilisaatorit koos takistitega, et saada väljundpingeks  $\sim 3$  VDC, millest viimast on vaja mikrokontrolleri viikude jaoks. Sellega saadakse lahendatud ESP32 toitevajadused ning pingeteisendused mikrokontrolleri viikude kasutamiseks. Kuna toide on plaanitud võtta peale elektrikilbi kaitseseadet, siis see tähendab mikrokontrolleri jaoks toitekatkestust, mille lahendamiseks oleks vahele vaja panna taaslaetav akuelement, mis võtaks toite tagamise ülesande üle, kui mõõdetav kaitse peaks rakenduma.

Kontrollimaks toite katkestust, tuleb luua ühendusskeem, et mikrokontrolleri toide läheks ümber akuelemendi peale ning signaali klemmi pealt kaoks toide ära, mille baasil saab registreerida toitekatkestuse ning edastada signaal.



Joonis 2.2 Planeeritud lahendus prototüüpseadme toiteks ning katkestuse kontrolliks

Joonisel 2.2 on kujutatud alaldit võrgupinge 230 VAC teisendamiseks 5 VDC, sekundaarne DC toiteallikas, mis on planeeritud tagama toite mikrokontrollerile, mis ühenduvad laadimismoodulisse DFR0264 [12]. Pinge reguleerimiseks on plaanitud kasutada LM317T komponenti koos 1 k $\Omega$  ja 1,5 k $\Omega$  takistitega, reguleerimaks pinget ligikaudu 3 VDC peale, mis on sobilik mikrokontrolleri universaalse sisend-väljund (GPIO, inglise keeles General Purpose Input/Output) viigule.

Takistite valikul võeti suvaliselt takisti R1 väärtuseks 1 k $\Omega$ . Takisti R2 leidmiseks kasutati LM317T väljundpinge arvutamiseks valemit:

$$U_{\text{out}} = 1,25 * \left(1 + \frac{R_2}{R_1}\right) \quad (2.2.1)$$

## 2.3 Kasutajaliides

Veebitarkvara peaks olemuselt olema kasutajaliides, kus on võimalik sisestada süsteemi paigaldatud seadmed ning kuvada nende seisukord ning pidada selle kohta logi.

Veebitarkvara on planeeritud kirjutada HTML-ina. Sellisel juhul ei ole vaja muretseda eraldi veebiserveri tasude eest ülesande lahendamise mõttes, kuid saame siiski edastada skripte ning kujundada kasutajaliidest.

Kasutajaliides peaks olema võimeline võtma kasutaja poolt vastu sisestatud seadmete tuvastusandmed ning need logima mälusse. Ühtlasi peaks olema võimalik edastada juhtiv-seadmele päringut koos seadmete tuvastusandmetega.

Disaini poolest tuleb kasutajaliides disainida lihtsasti arusaadavalt. Samuti on kasutajal mugavam liidest kasutada, kui toimub visuaalne tagasiside tehtud toimingute kohta. See tähendab, et päringute saatmisel tuleb kuvada informatsiooni, et päringu teostamine on pooleli ning mis on saadud informatsioon. Ühtlasi peab funktsioonide kasutamine olema lihtne ning sisaldama dokumentatsiooni, kuidas mis töötab.

## Seadmete kontroll

Sisesta kilbi IP:

Lisa seade

Kontrolli staatust

Joonis 2.3 Kasutajaliidese planeeritud lihtsustatud näidis

## 3. LAHENDUSE REALISEERIMINE

### 3.1 Prototüüpseade

Prototüüpsüsteemi loomiseks kasutatakse mitut mikrokontrollerit ESP32-DevKitC-32E [11]. Süsteem sisaldab endas juhtiv-seadet, mis tegeleb päringu edastamisega juhitav-seadmetele ning neilt saadud andmete kuvamisega kasutajaliidesesse. Juhitav-seadmete ülesanne on vastavalt saadud päringu sisule kas saata enda andmeid, saata edasi teise juhitav-seadme andmeid või edastada päringut teistele juhitav-seadmetele. ESP32 on mitmekülgne ja väga võimekas süsteem-kiibilahendus (SoC, inglise keeles system-on-chip), mille on välja töötanud Espressif Systems ja mis on peamiselt mõeldud asjade interneti (IoT, inglise keeles Internet of Things) rakenduste jaoks. Selle tuumaks on topelttuumaga Xtensa protsessor, mille taktsagedus ulatub kuni 240 MHz-ni, pakkudes märkimisväärset arvutusvõimsust erinevateks ülesanneteks. Üks ESP32 silmapaistvatest omadustest on selle sisseehitatud ühendusvõimalused, mis hõlmavad nii Wi-Fi kui ka Bluetoothi, võimaldades sellel juhtmevabalt ühenduda võrkudega ja teiste seadmetega. Lisaks on ESP32-l olemas andmeedastusprotokoll ESP-NOW, mis võimaldab ESP mikrokontrolleritel juhtmevabalt üksteisega suhelda ilma Wi-Fi võrgu olemasoluta. See võimalus on kasulik tegeledes juhtmevaba andmeedastusega kohtades, kus puuduvad IP-põhised andmeedastusvõimalused, nagu näiteks ehitusplats.

ESP32 pakub laia valikut komponente, sealhulgas mitmeid universaalseid sisend-väljund (GPIO) viike, analoog-digitaal muundureid (ADC, inglise keeles analog to digital converter), digitaal-analoog muundureid (DAC, inglise keeles digital to analog converter), pulseeriva laiuse modulatsiooni (PWM, inglise keeles pulse width modulation) kontrollereid ja suhtlusliideseid nagu I2C, SPI ja UART. See lai seadmete valik võimaldab ESP32-l suhelda erinevate andurite, ajamite ja muude riistvarakomponentidega, muutes selle suurepäraseks valikuks prototüüpimiseks ja manussüsteemide ehitamiseks.

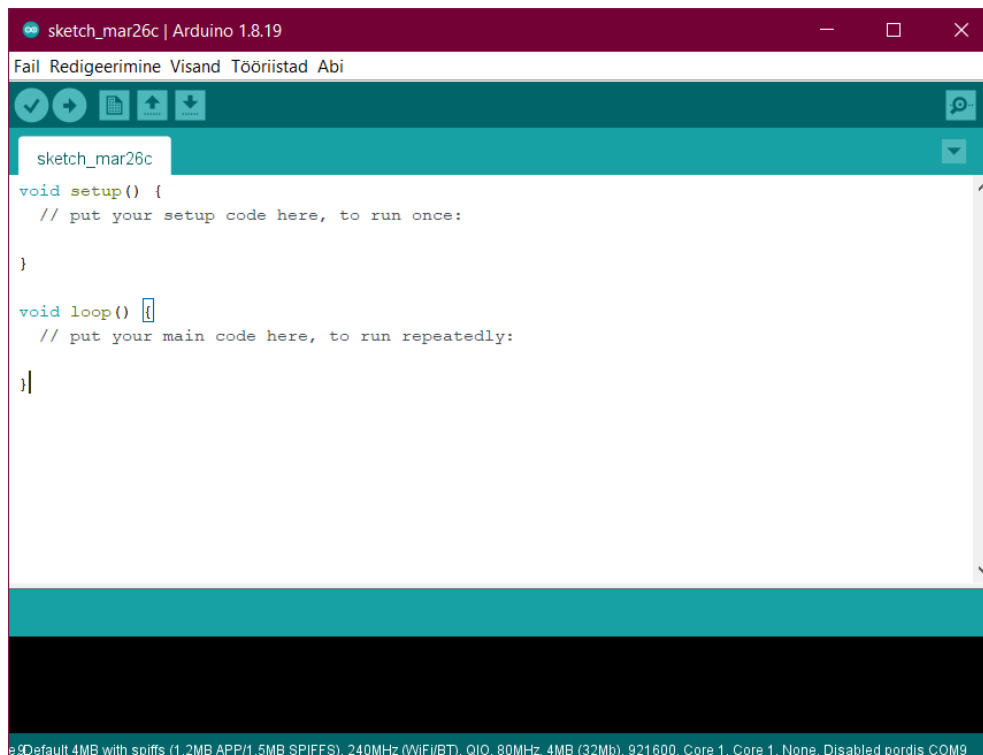
Tarkvarapoolselt pakub ESP32 ulatuslikku arendustuge. See ühildub Arduino integreeritud arenduskeskkonnaga (IDE, inglise keeles integrated development environment), Espressifi enda ESP-IDF (IoT Development Framework) ning PlatformIO-ga, pakkudes arendajatele, erinevat tausta ja eelistusi silmas pidades, paindlikkust. Need arenduskeskkonnad toetavad mitmeid teke ja raamistikke, muutes keerukate rakenduste loomise lihtsamaks, mis hõlmavad võrguvõimekusi, nagu HTTP serverid, MQTT kliendid ja WebSocket-i põhised suhtlust.

Turvalisus on ESP32 puhul samuti oluline aspekt, milleks on näiteks turvaline käivitus, mälu krüptimine ja riistvarapõhised krüptograafia kiirendid. Need võimalused aitavad

tagada turvalise andmeedastuse ja salvestuse, mis on paljude IoT rakenduste puhul kriitilise tähtsusega.

### 3.1.1 Mikrokontrollerite programmeerimine

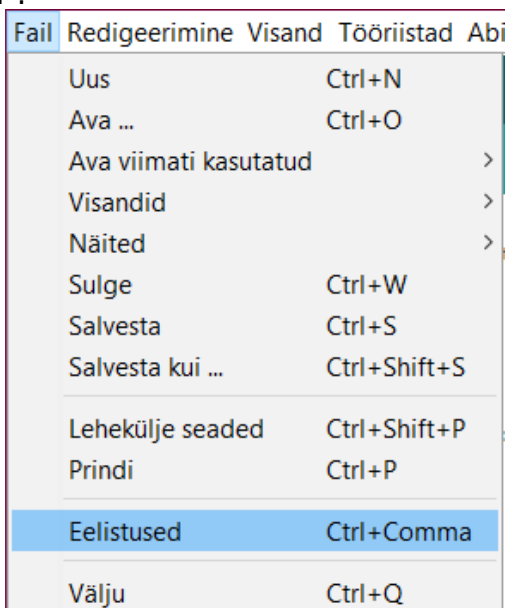
Kontrollereid programmeeritakse Arduino IDE 1.8.19 programmeerimistarkvaras [13].



Joonis 3.1 Arduino IDE 1.8.19 programmeerimistarkvara

**Tarkvara ülesseadmiseks** tuli enne alustamist läbi viia mõningad seadistused. Allpool on toodud programmeerimistarkvara ülesseadmiseks vajalikud etapid.

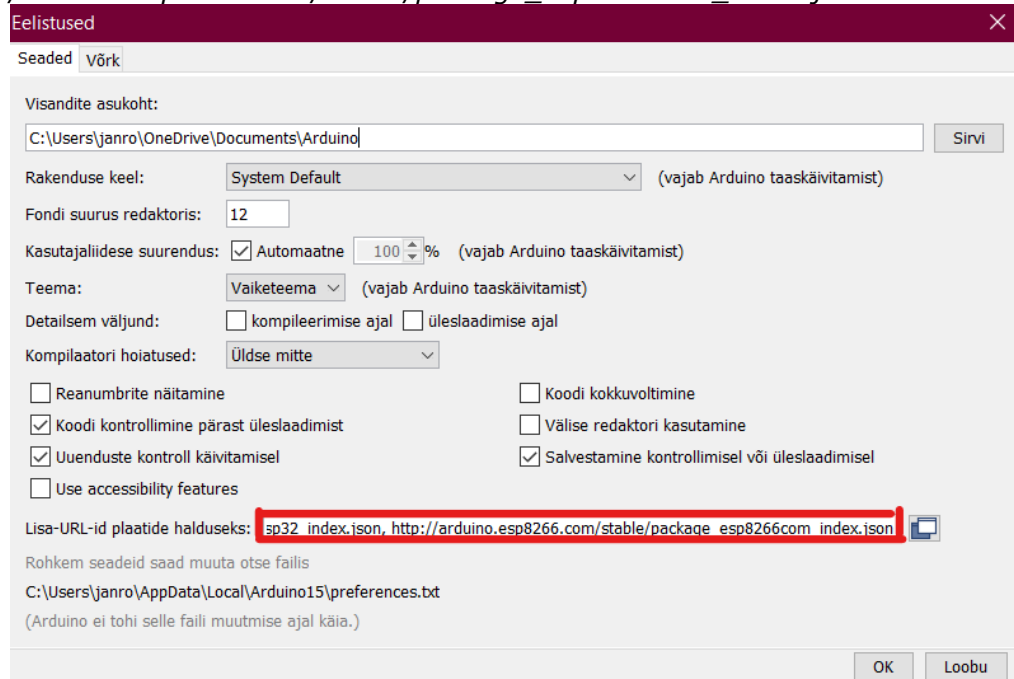
**ESP32 teekide lisamiseks** tuli tarkvaras teha järgnevad seadistused. Menüüst tuli valida "Fail -> Eelistused".



Joonis 3.2 IDE-s eelistuste valimine

Peale hüpikakna avanemist, tuli joonisel 3.3 märgitud välja sisestada järgmine tekst:

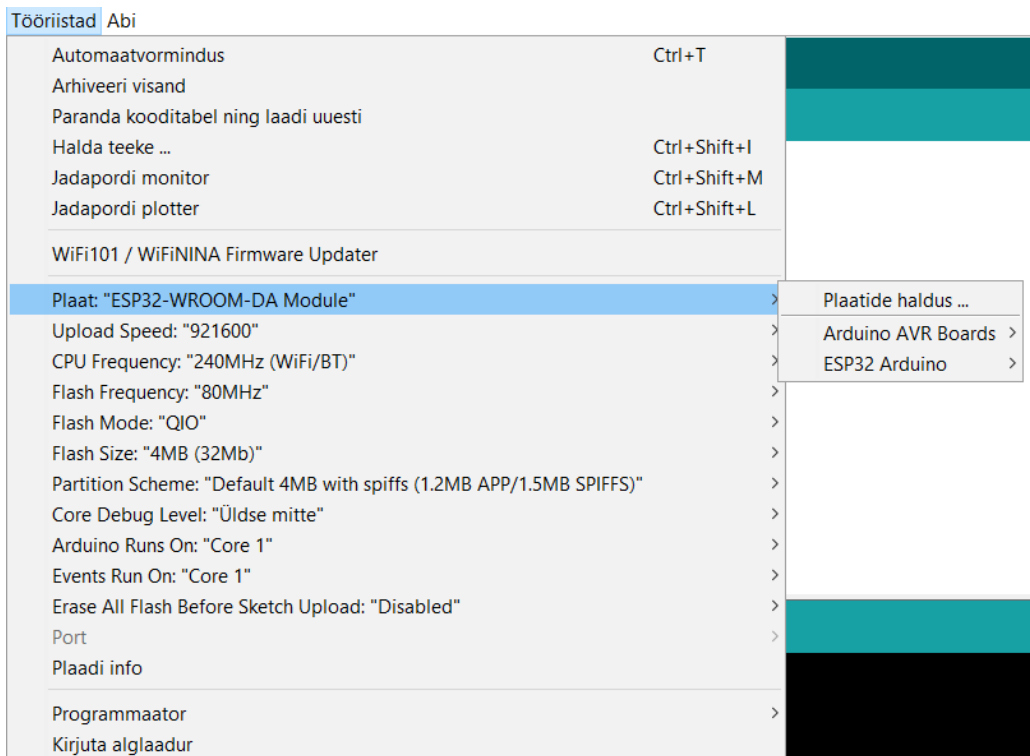
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json),  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



Joonis 3.3 Eelistuste hüpikaken, märgitud tekstiväljaga

Peale teksti sisestamist, tuli vajutada „OK“.

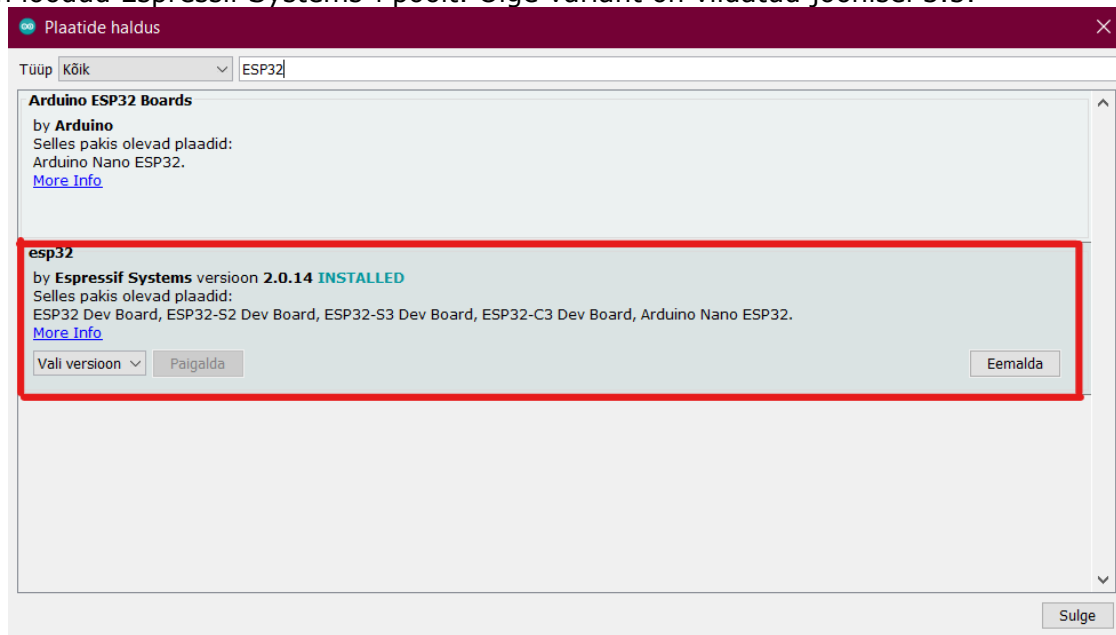
**Õige plaadi valimiseks** tuli suunduda „Tööriistad -> Plaat -> Plaatide haldus ...“.



Joonis 3.4 Näidis, jõudmaks plaatide haldusesse

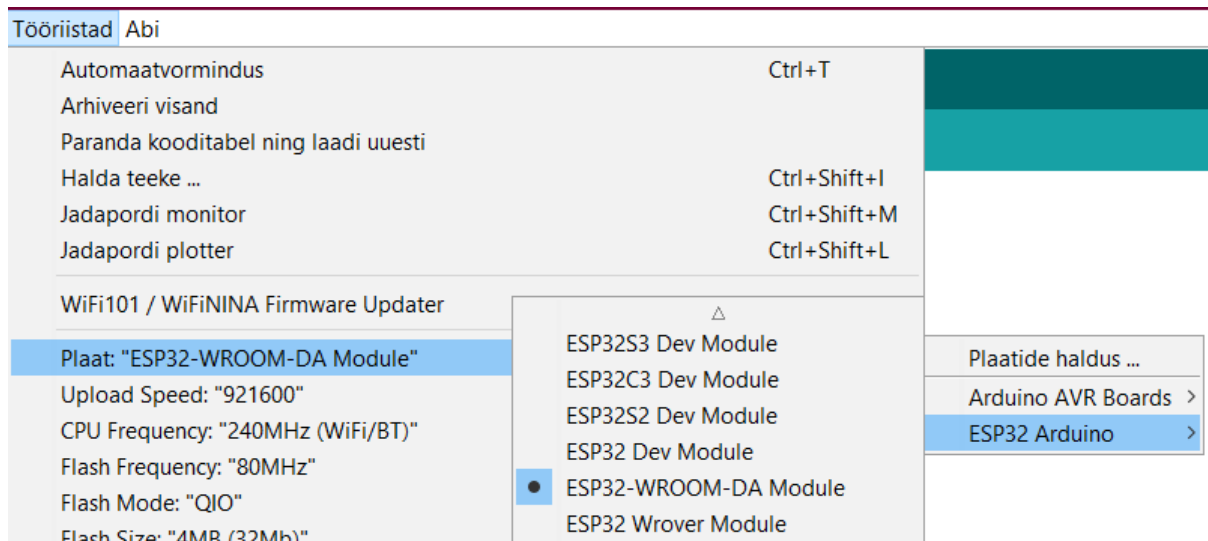


Plaatide halduse hüpikaknas tuli otsingusse sisestada „ESP32” ning valida variant, mis on loodud Espressif Systems-i poolt. Õige variant on viidatud joonisel 3.5.



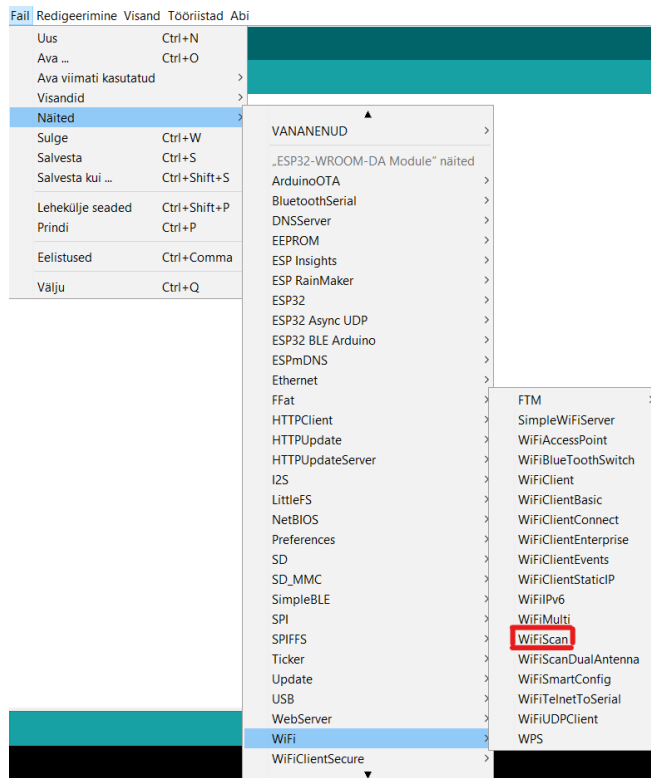
Joonis 3.5 Õige valik plaatide halduse hüpikaknas

Kui õige pakk oli alla laetud, tuli valida nimekirjast õige plaat. Selleks mindi "Tööriistad -> Plaat -> ESP32 Arduino -> ESP32-WROOM-DA Module".



Joonis 3.6 Nimekirjast õige plaadi valik

**Programmeerimistarkvara töökorra kontrolliks** tuli testida kontrolleriit tarkvarasisesega näitekoodiga. Selleni pääseb „Fail -> Näited -> (Valitud plaadi näited) WiFi -> WiFiScan”.



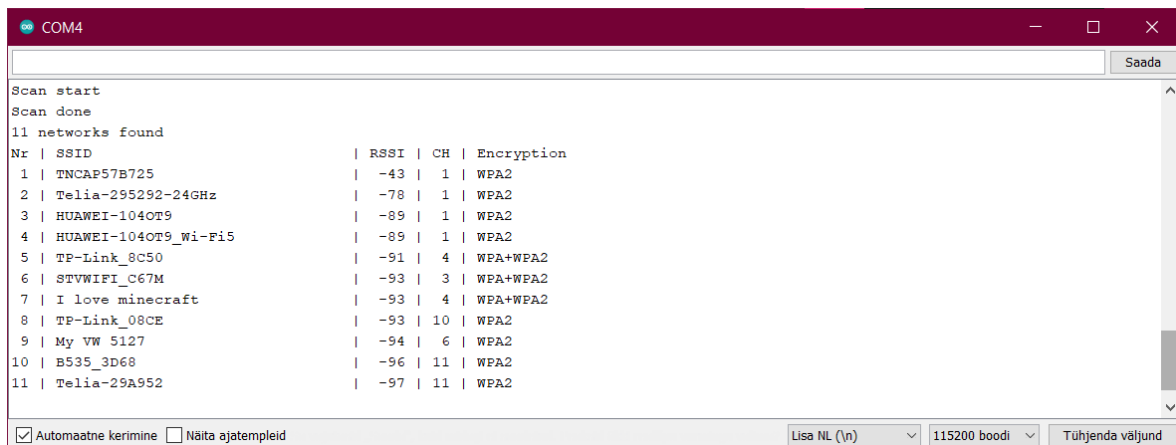
Joonis 3.7 Wi-Fi ühenduse testimise näitekoodi avamine

Koodi avamisel tuli see laadida plaadile, mis oli ühendatud sülearvutisse USB A ning USB micro B kaabliga. Üleslaadimise ajal tuli jälgida tarkvara all servas olevat infotahvlit, kui ilmus sõnum „Connecting...“, tuli plaadil all hoida „Boot“ nuppu, kuni algasid ilmuma uued read.



Joonis 3.8 ESP32 plaat, punane – boot nupp, roheline – EN nupp, teibil MAC-aadress

Peale koodi üleslaadimist plaadile, tuli avada jadaport, et näha plaadi poolt leitavaid läheduses olevaid Wi-Fi võrke. Testi tulemus on näha joonisel 3.9.



Joonis 3.9 Wi-Fi näidiskoodi tulemus jadapordis

Saadaval on ka inglise keelne juhend mõneti teistsuguse plaadiga Random Nerd Tutorials veebileheküljel [14].

**Kontrollerite valmisseadmiseks programmeerimiseks** tuli tuvastada iga kontrolleri MAC-aadress. Selleks laeti igasse kontrollerisse koodilõik:

```
#include <WiFi.h>
```

```
void setup(){
  Serial.begin(115200);
  Serial.println();
  Serial.print("ESP Board MAC Address: ");
  Serial.println(WiFi.macAddress());
}
```

```
void loop(){
```

```
}
```

Peale jadapordi avamist tuli plaadil vajutada nuppu "EN" (vt. joonis 3.8). Selle tulemusel saadi ekraanilt lugeda plaadi MAC-aadress ning see üles täheldada. Edasiseks tööks kirjutati see teibile ning kinnitati vastavale plaadile [15]. Jadapordis printitud MAC aadress on viidatud joonisel 3.9.

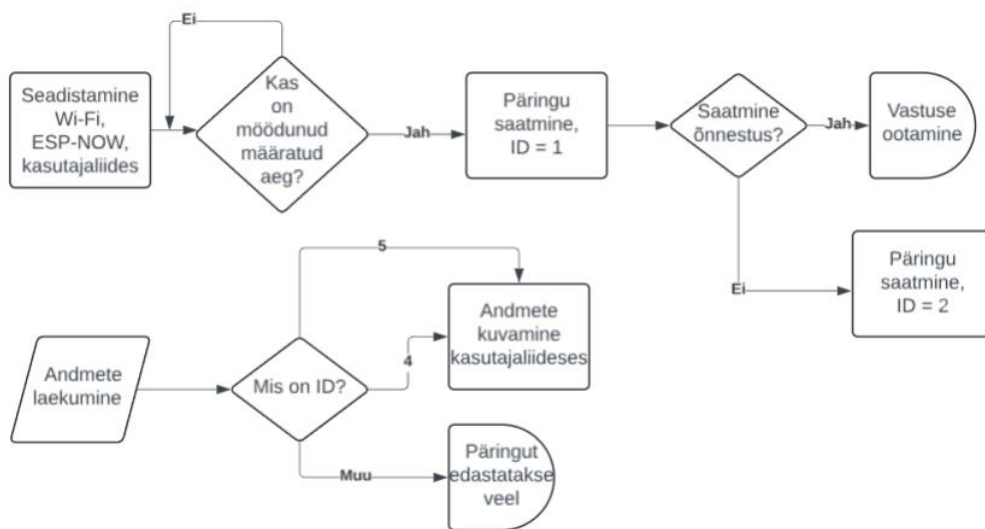


Joonis 3.9 Jadapordi väljund MAC-aadressi leidmise koodiga

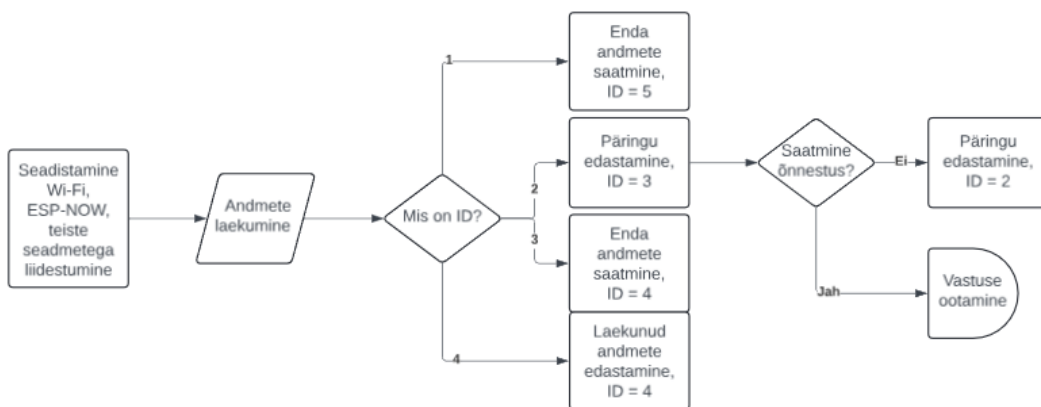
**Mikrokontrollerite programmeerimiseks** mõeldi välja andmeedastuse süsteem, mis põhineb päringu ID-del. Kuna kontrolleritevaheline andmeedastus on üles ehitatud

põhimõttel, et iga edastus saadetakse kõigile seotud seadmetele, mis on andmeedastuse ulatuses, tuleb teha iga andmepaketi saatmise korral kontroll, kas saadetud pakett jõudis kohale, või mitte. See funktsioon on ESP32 ESP-NOW protokollis ehitatud ning hõlpsasti rakendatav. Selle abil on võimalik teha otsuseid, kas uue päringu saatmine on vajalik ning mis peaks olema uue päringu sisu.

Joonistel 3.11 ning 3.12 on toodud vastavalt juhtiv-seadme ning juhitav-seadme põhimõttelised plokkskeemid. Kontrollerite koodid on toodud lisades L1 ja L2, juhtiv-seadme ning juhitav-seadme koodid vastavalt.



Joonis 3.10 Juhtiv-seadme koodi põhimõtteline plokkskeem [16]



Joonis 3.11 Juhitav-seadme koodi põhimõtteline plokkskeem [16]

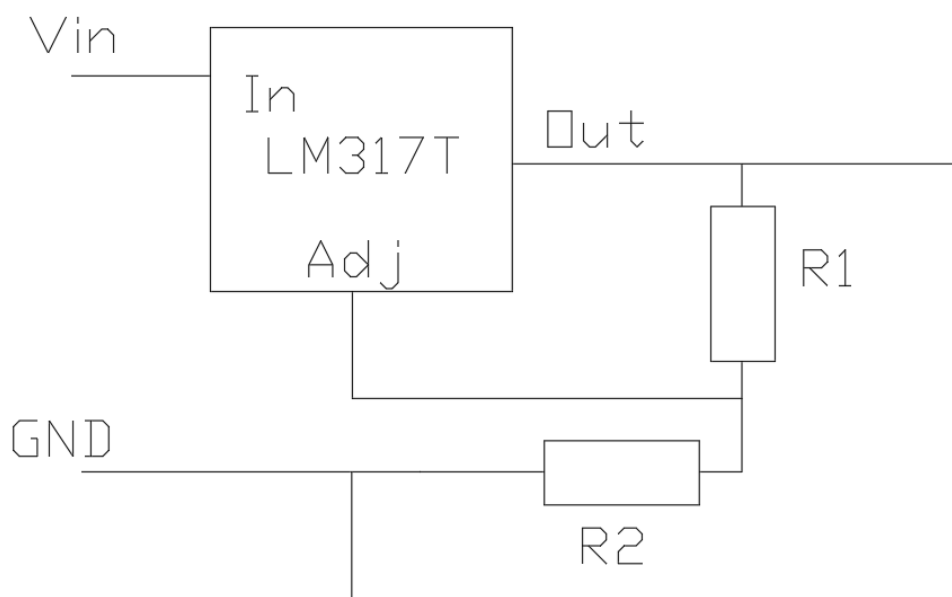
### 3.1.2 Seadme ehitus

Seadme ehituseks oli kõigepealt vaja tegeleda toitelahendusega. Kuna standardses töömaakilbis on pinge 230 VAC, aga kontroller töötab toitepingega 5 VDC ning GPIO

kasutamiseks on vajalik 3,3 VDC, tuli pinget alaldada ning alandada. Selle tegemiseks jaotati töö kahte etappi.

Alaldamiseks kasutati lihtsat, odavat ning kättesaadavat telefonilaadijas leiduvat plaati, mille eesmärk on võrgupinget alaldada ning ühtlasi konverteerida see 230 V pealt 5 VDC peale. Plaatide ühendusskeemid erinevad laadijati, aga tööpõhimõtte ning komponendid on suures osas sarnased. Kõigepealt kasutatakse trafosid/solenoidide pinget alandamiseks, seejärel dioode pinget alaldamiseks ning kondensaatoreid pinget silumiseks. Saadud väljundiks on 5 VDC.

Pinget korrigeerimiseks mikrokontrollerile kasutati pingestabilisaatori komponenti LM317T. Toote andmeleht on toodud lisas L3. Komponenti andmelehest tulenevalt on komponendi sisendpingeks 3-40 VDC ning reguleeritavaks väljundpingeks 1,7-37 VDC, lubades kuni 1,5 A voolu [17]. Tegemist on kolme jalaga elektroonikakomponendiga, mille väljundpinget on võimalik reguleerida kahe takistiga. Takistite väärtuste arvutamiseks kasutatakse valemit 2.2.1, eelduseks võetakse takisti R1 väärtuseks 1 k $\Omega$ . Väljundpingeks peaks olema  $\sim$ 3 VDC. Valemi kasutamisel leiti sobivuse ja komponentide saadavuse optimaalsuse skaalal takisti R2 väärtuseks 1,5 k $\Omega$ . Joonisel 3.13 on toodud LM317T ning takistite R1 ja R2 ühendusskeem.

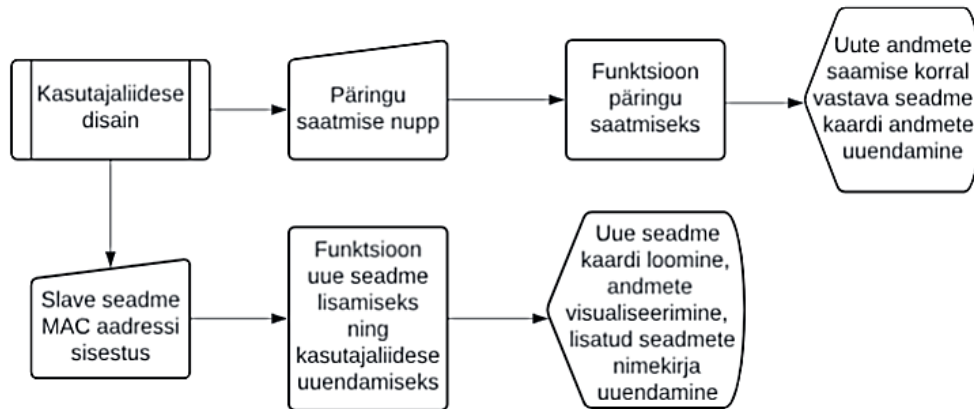


Joonis 3.12 LM317T ning takistite R1 ja R2 ühendusskeem

## 3.2 Kasutajaliides

Kasutajaliidese disain, andmete visualiseerimine ning sisu on kirjutatud sisse juhtiv-seadme koodi, milleks tuli lisada veebiserveriga seonduvad teegid. Selline lahendus võimaldab kasutajaliidest uuendada ja täita infoga dünaamiliselt, sõltuvalt juhtiv-seadmesse laekuvast infost.

Joonisel 3.14 on toodud juhtiv-seadme koodist kasutajaliidese osa põhimõtteline plokkskeem ning lisas L1 (juhtiv-seadme koodis) on kasutajaliidest mõjutav kood leitav funktsioonis "String generateDynamicHTML()".



Joonis 3.13 Kasutajaliidese põhimõtteline plokkskeem [16]

## **4. KATSED REAALSES KESKKONNAS JA TULEMUSED**

Käesolevas peatükis viiakse prototüüpsüsteemiga läbi katsed Tallinna Tehnikaülikooli energeetika maja teisel korrusel ning reaalsetes ehitusjärgus hoonetes Tallinna Arteri kvartali kõrgeimas hoones, keldrikorruse parklas, soojaku platsilt ehitusplatsi vahemaal ning lisaks ka kombineerituna ühelt korruselt teisele ja seinale, testimaks prototüüpseadmete andmeedastuse ulatust ning leidmaks potentsiaalseid kitsaskohti.

### **4.1 Katse 1 - Tallina Tehnikaülikooli NRG maja**

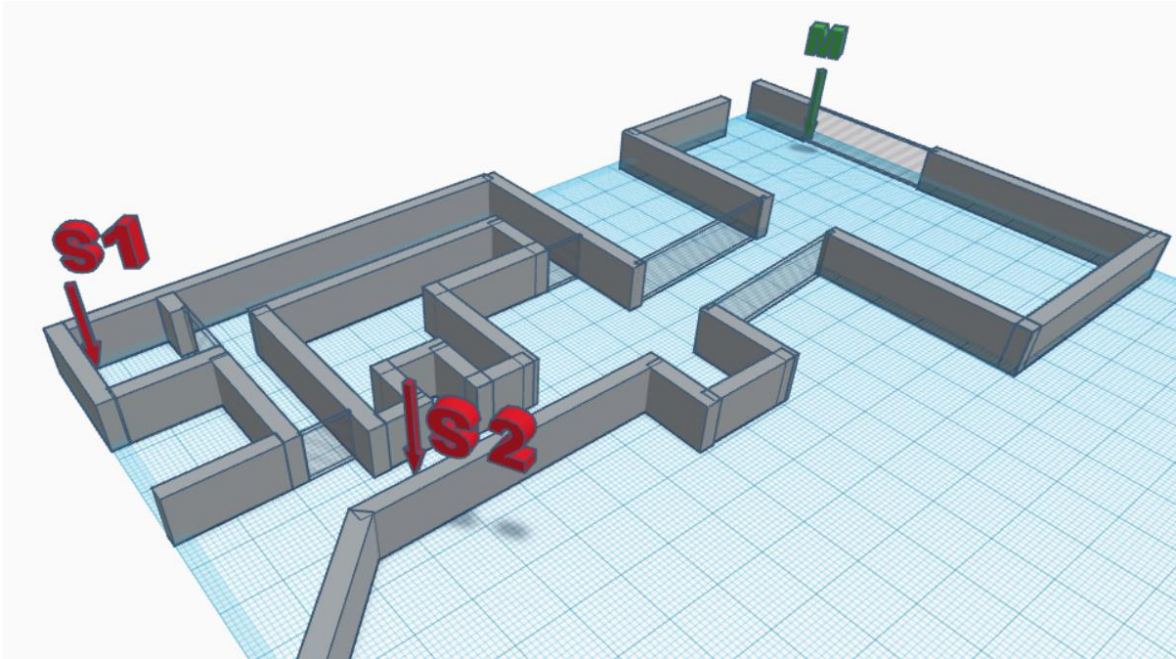
Käesoleva katse eesmärgiks oli tuvastada valmistatud süsteemi andmeedastuse funktsiooni juhul, kui mõni juhitav-seade peaks olema juhtiv-seadme andmeedastusulatusest väljaspool, kasutades selleks juhitav-seadmesse sissekodeeritud andmeedastuse pikenduse funktsiooni.

Katse läbiviimiseks asetati esimene juhitav-seade (S1) energeetika maja teisel korrusel asuvasse laboriruumi laua peale, ligikaudu 1,5 meetri kõrgusele põrandapinnast. Toidet võeti võrgust telefonilaadija adapteriga. Teise juhitav-seadme (S2) jaoks saadi toidet akupangast (5 VDC) ning hoiti käes ligikaudu samal kõrgusel energeetika maja teise korruse koridoris. Juhtiv-seade(M) oli ühendatud sülearvutisse jadapordi jälgimiseks ja toite andmiseks ning asus Tehnikaülikooli raamatukogu teise korruse ukse ees, ligikaudu 1 meetri madalamal, kui juhitav-seadmed, tulenevalt raamatukogu teise korruse põranda kõrguse suhtest energeetika hoone teise korruse põranda kõrgusest. Esmalt tehti juhtiv-seadme abil kindlaks kaugeim punkt esimese juhitav-seadme suhtes, et teada, kuhu paigutada katse käiguks teine juhitav-seade. Seejärel mindi juhtiv-seadmega antud punktist maksimaalsele kaugusele teisest juhitav-seadmest, milleks jäi raamatukogu ukseesine. Selles punktis tehti kindlaks, et esimese juhitav-seadmega ei oleks otseühendust, ühendades selleks toitest lahti teine juhitav-seade. Olles teinud kindlaks otseühenduse puudumise, ühendati mõlemad juhitav-seadmed toitesse ning liidestati läbi kasutajaliidese.

Tulemuseks oli edukas andmeedastuse ulatuse pikendus ning kasutajaliidese oli võimalik päringu ID baasil tuvastada, et esimese juhitav-seadme info tõepoolest edasi "põrgatatakse" teise juhitav-seadme poolt.

Joonisel 4.1 on toodud Tallinna Tehnikaülikooli energeetika hoone ning raamatukogu teise korruse ligikaudne 3D mudel ning viidatud katses kasutatud seadmete asukohti. 3D mudelis olevad hoone seinad on ligikaudu 0,2 meetri paksused betoonseinad ning uste kohtadel kujutatud klaaside korral on tegemist pakettklaasidega.

Juhtiv-seadmete S1 ning S2 sirgjooneline vahekaugus on ligikaudu 20 meetrit ning juhitav-seadme S2 ning juhtiv-seadme M vahekaugus on sirgjooneliselt ligikaudu 65 meetrit.



Joonis 4.1 Tehnikaülikooli katse seadmete ligikaudsed asukohad 3D mudelis [18]

## 4.2 Katse 2 - Arter kvartal

Arteris sooritatud katsete eesmärgiks oli tuvastada seadmete reaalne andmeedastuse kaugus, olles paigutatud ehitusaegsete kilpide metallkesta sisse. Eelduseks on metallkestast tulenev drastiline andmeedastuse kauguse vähenemine. Kasutatavaid seadmeid ei kasutata töös kirjeldatud komponentidega alaldiga ega pingeregulaatoriga tulenevalt katse olemusest seadmeid katseti ringi tõsta, kuigi seadmete planeeritud lõppolemus on kilbisine ning paigaldatud DIN-liistule. Katses kasutatava ehitusaegse elektrikilbi näidis on toodud lisan L4.

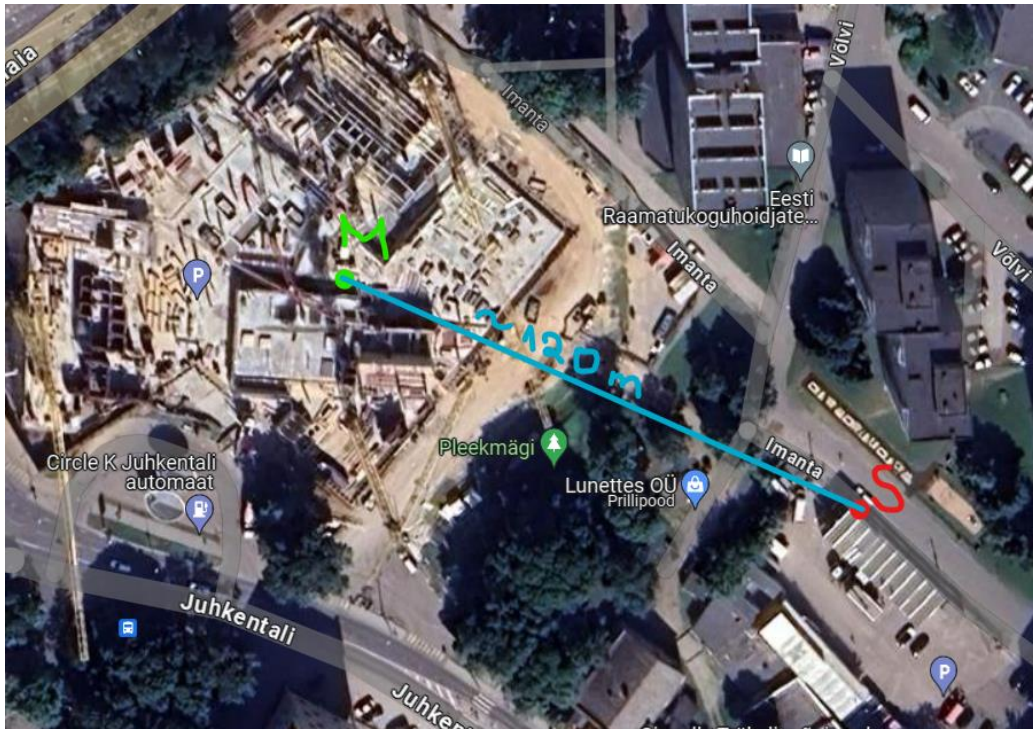
Katsete visualiseerimiseks kasutatakse Merko Ehitus Eesti AS poolt kasutatavat tarkvara programmi Dalux [19], kus on kujutatud ehitusplatsi 3D mudel.

### 4.2.1 Soojakust ehitusplatsile katse

Tulenevalt katse tegemise ajast nädalavahetusel, puudus ligipääs soojaku siseruumidesse. Selle tõttu asetati üks juhitav-seade (S) soojakutele toidet tagavasse kilpi soojakute platsil. Toide juhitav-seadmele võeti akupangast ning asetati läbi kilbiukse kilpi sisse. Juhtiv-seade (M) ühendati sülearvutisse jadapordi jälgimiseks ning toite andmiseks. Seejärel mindi juhtiv-seadmega juhitav-seadmest maksimaalsele kaugusele, kuni kaotati signaal, mis tuvastati kasutajaliidese abil. Katse mõjutavaks faktoriks oli võrdlemisi füüsiliste takistusteta keskkond.

Juhtiv-seade kaotas juhitav-seadmega andmeside sirgjooneliselt ligikaudu 120 meetri kauguselt. Katses kasutatud seadmete ligikaudsed asukohad on toodud joonisel 4.2.





Joonis 4.2 Soojakust ehitusplatsile katse tulemus, roheline – juhtiv-seade, punane – juhitav-seade [20]

#### 4.2.2 Keldrikorruse katse

Katse sooritamiseks asetati üks juhitav-seade (S) -1 korrusel keldris suvaliselt valitud ehitusaegsesse kilpi ning korraldati distantse katset sarnaselt eelmisele punktile. Juhtiv-seadmega käidi läbi kogu keldrikorrus ning täheldati üles signaali puudumise tsoonid. Käesolevas katses on eelduse kohaselt tulemusi enim mõjutavaks faktoriks betoonseinad.

Signaal saadi kätte peaaegu igal pool keldrikorruse parklas, välja arvatud, kui signaali teekonnale jäi mõne teise hoone liftihalli betoonseinad või parkla sissesõidutee betoonseinad.

Juhtiv-ja juhitav-seadme vahelise ühenduse katkemise tsoonide tulemused on toodud punasega joonisel 4.3.



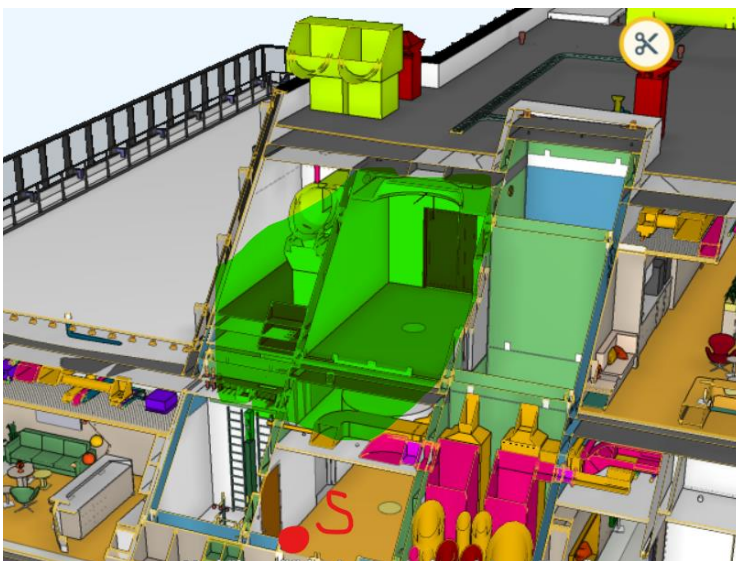
Joonis 4.3 Keldrikorruse katse visualisatsioon 3D mudelis

### 4.2.3 Kõrgeima hoone katse

Katses kasutatavate seadmete ülesseadmine toimus samaselt eelnevatele katsetele. Juhitav-seade (S) asetati ehitusaegsesse kilpi hoone 27. korrusel liftihallis ning juhtiv-seadme ja sülearvutiga mindi hoone 28. korrusele. Eelduse kohaselt on katset enim mõjutavaks faktoriks betoonist vahelagi. Juhtiv-seadmega kõnniti ringi 28. korrusel ning täheldati üles signaali olemasolu tsoonid.

Katse tulemusena avastati signaali olemasolu vaid 28. korruse liftihallis.

Juhtiv-ja juhitud-seadme vahelise ühenduse olemasolu tsoonide tulemused on toodud rohelisega joonisel 4.4.



Joonis 4.4 Kõrgeima hoone katse visualisatsioon 3D mudelis

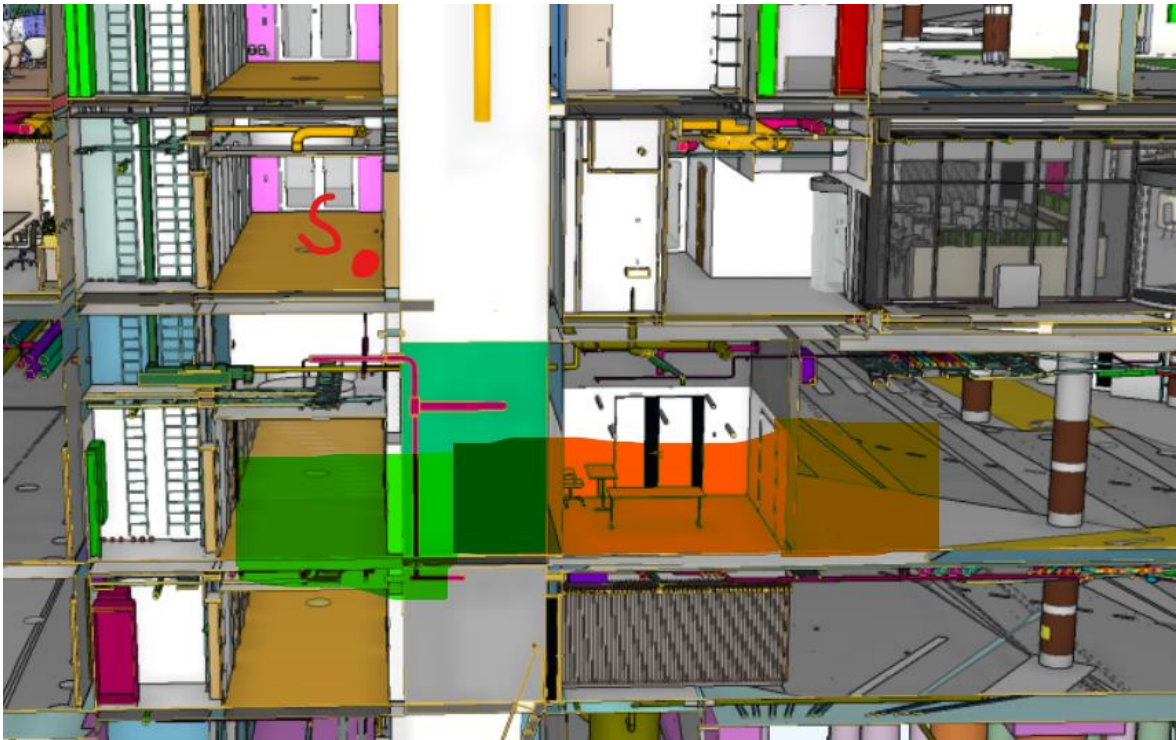
#### 4.2.4 Kombineeritud katse

Antud katse eesmärk on tuvastada seadmete andmeedastuse ulatus, kui signaalil on vaja levida läbi korruste vahelae ning jõuda ka nurga taha.

Selleks katseks valiti kvartalis asukohaks kõrgeima hoone esimese korruse liftihall ning keldrikorrusel katses kasutatud kilp. Juhitav-seade (S) asetati akupangaga esimese korruse liftihallis olevasse kilpi ning juhtiv-seadmega mindi keldrikorrusele. Andmete edastust jälgiti taaskord läbi kasutajaliidese.

Kontrollides signaali olemasolu keldrikorruse kilbi juures, tuli välja selle puudumine. Küll aga saadi signaal kätte keldrikorrusel just vahetult väljaspool liftihalli.

Saadud katse tulemused on kujutatud joonisel 4.5.



Joonis 4.5 Kombineeritud katse visualisatsioon 3D mudelis

## **5. ARENDUSVÕIMALUSED**

Tulenevalt katsetulemustest Tallinna Tehnikaülikooli energeetika majas (NRG) ning Tallinna Arteri kvartalis, saab öelda, et andmeedastuse kaugus on suuresti sõltuv keskkonnast ning edastatavate andmete teekonnale jäävatest takistustest. Suurimateks mõjuriteks on seinad, laed, aknad, uksed ning nende paksused ja materjal. Katsetes selgus, et signaali levik kilbi metallkestast välja ei olnud suureks mõjuriks. Tuli ka välja, et korruste vahelaed mõjutavad signaali levikut tunduvalt rohkem kui vaheseinad, mis tuleneb tõenäoliselt nende paksusest ning signaali võimalusest minna ümber takistuse.

### **5.1 Võimalikud lahendused**

Signaali teekonnale ettejäävate takistuste juures muutusi teha ei saa, küll aga on ESP32-l olemas eraldi viik antenni ühendamiseks signaali tugevuse suurendamiseks. Kilpidele traditsioonilise antenni ühendamine ei ole mõttekas tulenevalt ehitusplatsi keskkonnatingimustest; on suur tõenäosus nende purunemiseks. Küll aga on võimalik disainida selline antenn, mis ei mõjutaks kilbi välist geomeetriat, oleks vastupidav ning võimendaks signaali tugevust, et saata signaal paremini läbi paksude betoonseinte ja lagede.

### **5.2 Projekti arendus**

Prototüüpseadmele on veel vaja juurde disainida kaitsev kest, mille abil saaks seadet kinnitada ka kilbisisele DIN-liistule, ning vastupidavad ühendused erinevate komponentide vahel.

Kuna prototüüpseade on valdavalt kokku ehitatud käepärastest komponentidest, tuleks projekti komponente siluda ning kombineerida need ühele trükkplaadile hoidmaks kokku mõõtmelteil ning juhtmete vajadusest. Selle tulemusel on ka lihtsam disainida moodulile kesta, mis on 3D-prinditav.

ESP32-l on veel mitmeid kasutamata viike, mille tõttu on tulevikus võimalik seadmele lisada ka mooduleid kilbi voolutarbe mõõtmiseks ning arvutuslikult kuvada ka kilbi poolt kasutatav võimsus, mis on ehitusplatsil väga kasulik informatsioon käidukorralduseks, probleemide ennetamiseks ning diagnoosimiseks.

## 6. KOKKUVÕTE

Käesolev bakalaureusetöö käsitleb ehitusplatsi elektritoite olemasolu automatiseeritud jälgimist. Töö eesmärgiks on prototüüpseadme loomine, mis võimaldaks reaajas elektritoite olemasolu jälgimist ehitusplatsi erinevates kriitilistes punktides. Sellise jälgimise vajadus tuleneb ehitusplatsi keskkonnatingimuste eripäradest, kus elektrikatkestused võivad põhjustada tõsiseid ajalisi viivitusi ning rahalisi kahjusid.

Prototüüpsüsteem on loodud mikrokontrollerite ESP32 baasil, mis võimaldavad juhtmevaba andmeedastust ka Wi-Fi võrgu puudumisel. Mikrokontroller saab toidet ehitusaegsest töömaakilbist läbi alaldi, mis muundab 230 VAC pinget 5 VDC-ks. Seadmetel on ka varutoide akuelemendi näol, et tagada andmete säilimine ning edastamine elektrikatkestuse korral. Jälgimiseks ning andmete kuvamiseks kasutatakse ka kasutajaliidest, mis võimaldab veebipõhisel tarkvaral visualiseerida andmeid ning saata päringuid eemalt.

Töö praktiline osa hõlmab prototüüpseadmete ja-süsteemi ehitust, tarkvara arendamist ja katsetamist realses keskkonnas. Töö käigus kogutud andmete analüüs kinnitab planeeritud lahenduse tõhusust ning sobivust ehitusplatsi keskkonnatingimustes. Samuti on lõputöös kirjeldatud potentsiaalseid tulevikusuundumusi ja parendusvõimalusi.

Bakalaureusetöö tulemuseks on demonstreeritav prototüüpsüsteem koos kasutajaliidesega, mis suudab reaajas jälgida elektritoidet ehitusplatsi erinevates asukohtades, potentsiaalselt olles võimeline vähendama elektrikatkestustest tulenevaid ajalisi viivitusi ning rahalisi kahjusid. Töö toob esile ka süsteemi võimalikud nõrkused ning pakub välja lahendused nende parendamiseks. Töö panus seisneb praktikas kasutatava lahenduse loomises, mis aitab tõsta ehitusplatsi elektrivarustuse jälgimise ja juhtimise tõhusust ning ohutust.

## KASUTATUD KIRJANDUSE LOETELU

- [1] B. Shively, „Enerdynamics,” 24 02 2024. [Võrgumaterjal]. Available: [https://www.enerdynamics.com/Energy-Currents\\_Blog/What-Is-a-Phasor-Measurement-Unit-and-How-Does-it-Make-the-Grid-More-Reliable.aspx](https://www.enerdynamics.com/Energy-Currents_Blog/What-Is-a-Phasor-Measurement-Unit-and-How-Does-it-Make-the-Grid-More-Reliable.aspx). [Kasutatud 24 02 2024].
- [2] Reallin, „Reallin,” 24 02 2024. [Võrgumaterjal]. Available: <https://ee.reallinmeter.com/news/how-does-an-electric-meter-measure-electricity-51494918.html>.
- [3] WEG Eesti, „Tooted,” WEG eesti, [Võrgumaterjal]. Available: <https://www.weg.ee/elektriarvesti-din-3f1t3-230-400v-10-100-a-ilma-taatluseta-110889>. [Kasutatud 07 05 2024].
- [4] M. Melder, „Protsessiini digitaliseerimise ja minimeerimise võimalused Eesti kõrgepingealajaamade andmehõivesüsteemides,” Tallinna Tehnikaülikool, Tallinn, 2015.
- [5] Schneider Electric Eesti, „Äripäev,” 22 09 2022. [Võrgumaterjal]. Available: <https://www.ehitusuudised.ee/sisuturundus/2022/09/27/nutikas-vidin-mille-abil-saab-kodutarbija-oma-elektrikulud-kontrolli-alla>. [Kasutatud 24 02 2024].
- [6] Schneider, „Product Info,” 01 02 2024. [Võrgumaterjal]. Available: [https://www.productinfo.schneider-electric.com/wiser\\_eu/power-tag-e\\_device-user-guide\\_wiser\\_se/English/PowerTagE-%20Device%20User%20Guide.pdf](https://www.productinfo.schneider-electric.com/wiser_eu/power-tag-e_device-user-guide_wiser_se/English/PowerTagE-%20Device%20User%20Guide.pdf).
- [7] Schneider Electric, „Tooted,” Schneider Electric, [Võrgumaterjal]. Available: <https://www.se.com/ee/et/product/A9XMWD20/acti9-power-tag-link-wireless-to-modbus-tcp-ip-concentrator/>. [Kasutatud 24 04 2024].
- [8] Arvutitark OÜ, „Arvutitark,” 2024 02 2024. [Võrgumaterjal]. Available: <https://arvutitark.ee/nutiseadmed/nutikodu/aqara-smart-plug-1118719>.
- [9] Aqara, „Product,” Aqara, [Võrgumaterjal]. Available: <https://www.aqara.com/en/product/smart-plug-eu/>. [Kasutatud 03 05 2024].
- [10] Ieso, „Estimated Quantitative Cost Range Analysis for Installation of Phasor Measurement Units (PMUs) in Ontario’s Power System,” Iaso, Ontario, 2021.
- [11] Espressif, „Mouser Electronics,” 26 03 2024. [Võrgumaterjal]. Available: <https://www.mouser.ee/ProductDetail/356-ESP32-DEVKITC32E>.
- [12] DFRobot, „DFR0264,” DFRobot, [Võrgumaterjal]. Available: [https://wiki.dfrobot.com/Solar\\_Charger\\_SKU\\_DFR0264](https://wiki.dfrobot.com/Solar_Charger_SKU_DFR0264). [Kasutatud 30 04 2024].
- [13] Arduino, „Arduino,” 26 03 2024. [Võrgumaterjal]. Available: <https://www.arduino.cc/en/software>.
- [14] Tundmatu, „Random Nerd Tutorials,” 29 09 2023. [Võrgumaterjal]. Available: <https://randomnerdtutorials.com/getting-started-with-esp32/>.
- [15] Tundmatu, „Random Nerd Tutorials,” 26 03 2024. [Võrgumaterjal]. Available: <https://randomnerdtutorials.com/esp-now-many-to-one-esp32/>.
- [16] J. R. Lauk, „Lucid,” Lucid, [Võrgumaterjal]. Available: lucid.app. [Kasutatud 01 05 2024].
- [17] UMW, „AllDdatasheet,” 26 03 2024. [Võrgumaterjal]. Available: <https://html.alldatasheet.com/html-pdf/1785203/UMW/LM317T/129/2/LM317T.html>.
- [18] J. R. Lauk, „Tinkercad,” Autodesk, [Võrgumaterjal]. Available: tinkercad.com. [Kasutatud 02 05 2024].
- [19] Dalux, „Home,” Dalux, [Võrgumaterjal]. Available: <https://www.dalux.com/et/>. [Kasutatud 03 05 2024].
- [20] Google Inc., „Maps,” Google Inc, [Võrgumaterjal]. Available: <https://www.google.com/maps/@59.4293882,24.7603333,213m/data=!3m1!1e3?entry=ttu>. [Kasutatud 04 05 2024].

## LISAD

## Lisa 1 Juhtiv-seadme kood

```
/*
 * Nimetus: Ehitusaegse elektritoite automatiseeritud jälgimine - Juhtiv-seade
 * Autor: Jan Robin Lauk
 * Kuupäev: 30.04.2024
 */

/*
 * Päringu ID tuvastuslegend:
 * 1 - Esmane päringu ID
 * 2 - Juhtiv-seadmeni mittejõudmise korral edastatav ID
 * 3 - Juhtiv-seade ei ulatanud pingimiseks teise juhitav-seadmeni
 * 4 - Ulatusest kaugemalt tulnud juhitav-seadme andmed
 * 5 - Ulatuses oleva juhitav-seadme andmed
 * 6 - Määratud ID juhitav-seadmele, milleni esialgu ei ulatatud, määrab kasutajaliideses staatuse
 "Offline"
 */

// Vajalikud teegid
#include <esp_now.h>
#include <WiFi.h>
#include "ESPAsyncWebServer.h"
#include <Arduino_JSON.h>

#define MAX_PEERS 10 // Maksimaalne juhitav-seadmete kogus ja/või projektiks kasutatavate
juhitav-seadmete kogus

// Andmete kuvamiseks vajaminev 2.4GHz Wi-Fi võrgu nimi ja parool
const char* ssid = "TNCAP57B725"; //AndroidAP62A9 TNCAP57B725
const char* password = "96EC9345D2"; //utlepalun 96EC9345D2
int32_t channel = 0; //ESP-NOW andmeedastuseks 0, kasutab sama kanalit, mis Wi-Fi

// Taimer automaatseks andmeedastuse päringuks
static unsigned long lastQueryTime = millis();
static const unsigned long QUERY_INTERVAL_MS = 60000;

// Massiiv juhitav-seadmete MAC aadresside hoidmiseks ning nende kogus
uint8_t peerMACs[MAX_PEERS][6] = { {0, 0, 0, 0, 0, 0} };
int currentPeerCount = 0;

// Päringus edastatavate andmete algväärtus
int queryID = 0;
```

```

int masterID = 0;
float value = 0.00;

// Saadetava päringu struktuur
typedef struct queryData {
    int queryID;
    int masterID;
    float value;
} queryData;

queryData query;

// Saabuvate andmete struktuur
struct incData{
    int responseID;
    int slaveID;
    float analogValue;
};

incData receivedData;

// Juhitav-seadmete andmete struktuur
struct SlaveData{
    int responseID = 0;
    int slaveID = 0;
    float analogValue = 0;
    int currentPeerCount = 0;
    bool checkForDelivery = false;
};

SlaveData slaveData[MAX_PEERS];

// Võrdluseks vajaminev struktuur
struct CompData{
    int responseID = 0;
    int slaveID = 0;
    float analogValue = 0;
    int currentPeerCount = 0;
};

CompData compData[MAX_PEERS];

esp_now_peer_info_t peerInfo; // Juhitav-seadmetega sidumiseks vajaliku muutuja loomine

```



```

JSONVar board; // JSON muutuja

// Asünkroonse veebiserveri tegurid
AsyncWebServer server(80);
AsyncEventSource events("/events");

// Päringu saatmine
void sendQuery(int queryID){
    delay(100);
    Serial.print("Query ID sent: ");
    Serial.println(queryID);
    query.queryID = queryID;
    esp_err_t result = esp_now_send(0, (uint8_t *) &query, sizeof(queryData));
    if (result == ESP_OK) {
        Serial.println("Sent with success");
    }
    else {
        Serial.println("Error sending the data");
    }
}

// Tagasikutse andmete saatmisel
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
    uint8_t tempMAC[6];
    int index = 0;
    char macStr[18];
    // Määratakse indeks sõltuvalt asukohale juhitud-seadmete massiivis
    memcpy(tempMAC, mac_addr, sizeof(tempMAC));
    for (int i = 0; i < currentPeerCount; i++) {
        if (memcmp(peerMACs[i], tempMAC, sizeof(tempMAC)) == 0) {
            index = i;
            Serial.printf("Data sent to board from slaveData at index %d\n", index);
            break;
        }
    }
    Serial.print("Packet to: ");
    // Visualiseerime päringu saaja MAC aadressi
    snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
        mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr[5]);
    Serial.print(macStr);
    Serial.print(" send status:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
}

```

```

if (slaveData[index].checkForDelivery) {
    Serial.printf("Check for delivery is already true for board at index %d\n", index);
}
// Päringu saatmise ebaõnnestumisel saadame uue päringu ning määrame spetsiifilise ID,
uuendame kasutajaliidest
if(status != ESP_NOW_SEND_SUCCESS && !slaveData[index].checkForDelivery){
    slaveData[index].responseID = 6;
    Serial.printf("Query sent fail, setting board on index %d responseID to 6 before sending query
ID 2\n", index);
    sendReadingsToWeb(index);

    queryID = 2;
    slaveData[index].checkForDelivery = true;
    Serial.printf("Check for delivery for board at index %d: ", index);
    Serial.println(slaveData[index].checkForDelivery);
    sendQuery(queryID);
}
}

// Kasutajaliidese uuendamine
void sendReadingsToWeb(int slaveIndex) {
    int i = slaveIndex;
    Serial.printf("Sending data to web from slavedata at index %d\n", i);
    if (slaveData[i].slaveID != compData[i].slaveID || slaveData[i].responseID !=
compData[i].responseID || slaveData[i].analogValue != compData[i].analogValue) {
        compData[i].slaveID = slaveData[i].slaveID;
        compData[i].responseID = slaveData[i].responseID;
        compData[i].analogValue = slaveData[i].analogValue;
        compData[i].currentPeerCount = slaveData[i].currentPeerCount;
        Serial.println("New data sent to web: ");
        Serial.print("Slave ID: ");
        Serial.println(compData[i].slaveID);
        Serial.print("Response ID: ");
        Serial.println(compData[i].responseID);
        Serial.print("Voltage: ");
        Serial.println(compData[i].analogValue);
        board["id"] = compData[i].slaveID;
        board["voltage"] = compData[i].analogValue;
        board["responseID"] = compData[i].responseID;
        board["peerCount"] = compData[i].currentPeerCount;
        String jsonString = JSON.stringify(board);
        events.send(jsonString.c_str(), "new_readings", millis());
        delay(100);
    }
}

```

```

}
else {
    Serial.printf("No new data to be sent to web from slave ID: %d\n", compData[i].slaveID);
}
}

// Tagasikutse andmete sissetulekul
void onDataReceived(const uint8_t * mac, const uint8_t *incomingData, int len) {

    // Kopeerime saabuvad andmed ümber saabuvate andmete struktuuri
    memcpy(&receivedData, incomingData, sizeof(receivedData));

    // Juhitav-seadme ID põhise indeksi määramine
    int index = receivedData.slaveID - 1; // Eeldusel, et indekseerimine algab 1-ga
    Serial.printf("Response ID received: %d\n", receivedData.responseID);
    // Vastuse ID sobilikkuse kontroll ning kasutajaliidese uuendamine
    if (receivedData.responseID == 5 || receivedData.responseID == 4){
        if (index >= 0 && index < MAX_PEERS) {
            slaveData[index].responseID = receivedData.responseID;
            Serial.print("Board responded with query ID: ");
            Serial.println(slaveData[index].responseID);
            slaveData[index].slaveID = receivedData.slaveID;
            Serial.print("Received board ID: ");
            Serial.println(slaveData[index].slaveID);
            slaveData[index].analogValue = receivedData.analogValue;
            Serial.print("Voltage: ");
            Serial.println(slaveData[index].analogValue);
            sendReadingsToWeb(index);
        } else {
            Serial.println("Invalid slave ID!");
        }
    } else if (receivedData.responseID == 2){
        Serial.println("Pinging query is being forwarded!");
    } else {
        Serial.println("Slaves are trying to reach further nodes");
    }
}

// MAC aadressi baidi formaati muutmine
void macStringToBytes(const String& mac, uint8_t* macArray) {
    sscanf(mac.c_str(), "%02x:%02x:%02x:%02x:%02x:%02x",
        &macArray[0], &macArray[1], &macArray[2],
        &macArray[3], &macArray[4], &macArray[5]);
}

```

```

char macStr[18];
Serial.print("MAC of peer being added: ");
snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
         macArray[0], macArray[1], macArray[2], macArray[3], macArray[4], macArray[5]);
Serial.print(macStr);
}

```

*// Kasutajaliidese uuendus uue juhitav-seadme lisamisel algandmetega*

```

void notifyPeerAdded(int peerIndex) {
    int i = peerIndex;
    if (i >= 0) {
        Serial.println("Peer added to web");
        compData[peerIndex].currentPeerCount = slaveData[peerIndex].currentPeerCount;
        board["id"] = compData[i].slaveID;
        board["voltage"] = compData[i].analogValue;
        board["responseID"] = compData[i].responseID;
        board["peerCount"] = compData[i].currentPeerCount;
        String jsonString = JSON.stringify(board);
        events.send(jsonString.c_str(), "peer_event", millis());
        delay(100);
    }
    else {
        Serial.printf("Invalid slave index: %d\n", i);
    }
}

```

*// Juhitav-seadmete sidumine kasutajaliidese sisestatud MAC aadressiga*

```

void addPeerByMAC(const String& mac) {
    if (currentPeerCount >= MAX_PEERS) {
        Serial.println("Cannot add more peers, maximum limit reached.");
        return;
    }
    peerInfo.ifidx = WIFI_IF_AP;
    peerInfo.channel = channel;
    peerInfo.encrypt = false;
    Serial.print("peerInfo ifidx: ");
    Serial.println(peerInfo.ifidx);
    Serial.print("peerInfo channel: ");
    Serial.println(peerInfo.channel);
    Serial.print("peerInfo encrypt: ");
    Serial.println(peerInfo.encrypt);
}

```

```

// Olemasolu kontroll, massiivi lisamine ning kasutajaliidese uuendamise
uint8_t newMAC[6];
macStringToBytes(mac, newMAC);
memcpy(peerInfo.peer_addr, newMAC, 6);
if (esp_now_is_peer_exist(newMAC)) {
    Serial.println("Peer already added");
} else if (esp_now_add_peer(&peerInfo) == ESP_OK) {
    Serial.println("Successfully added new peer");
    slaveData[currentPeerCount].currentPeerCount = currentPeerCount + 1;
    currentPeerCount++;
    Serial.print("Current amount of added peers: ");
    Serial.println(currentPeerCount);
    memcpy(peerMACs[currentPeerCount - 1], newMAC, 6);
    notifyPeerAdded(currentPeerCount - 1);
} else {
    Serial.println("Failed to add new peer");
}
}

// Dünaamilise kasutajaliidese genereerimine
String generateDynamicHTML() {
    String html = "";
    // Kasutajaliidese disain
    html += "<!DOCTYPE HTML><html><head><title>ESP-NOW Dashboard</title>";
    html += "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">";
    html += "<link rel=\"stylesheet\" href=\"https://use.fontawesome.com/releases/v5.7.2/css/all.css\" ";
    html += "integrity=\"sha384-fnmOCqbTIWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr\" ";
    html += "crossorigin=\"anonymous\">";
    html += "<link rel=\"icon\" href=\"data:;\">";
    html += "<style>";
    html += "html { font-family: Arial; text-align: center; }";
    html += "body { margin: 0; }";
    html += ".topnav { background-color: #2f4468; color: white; font-size: 1.7rem; }";
    html += ".content { padding: 20px; }";
    html += ".cards { max-width: 700px; margin: 0 auto; display: grid; grid-gap: 2rem; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); }";
    html += ".card { background-color: white; box-shadow: 2px 2px 12px 1px rgba(140,140,140,.5); }";
    html += ".reading { font-size: 2.8rem; }";
    html += ".packet { color: #bebebe; }";
    html += ".card.voltage { color: #fd7e14; }";

```

```

// Juhitav-seadmete nimekirja keskele joondamine
html += "#peerListContainer { text-align: center; }";
html += "#peerList { display: inline-block; list-style-type: none; padding: 0; margin: 0; }";
// Punkt-nimekirja eemaldamine

// Pealkiri
html += "</style></head><body>";
html += "<div class='topnav'><h3>ESP-NOW DASHBOARD</h3></div>";

// Päringu saatmise nupp
html += "<div class='content'>";
html += "<div style='margin-top: 20px;'>";
html += "<button onclick='sendQuery()'>Send Query to Slaves</button>";
html += "</div>";

// Juhitav-seadme MAC aadressi sisestamise riba
html += "<div style='margin-top: 20px;'>";
html += "<form onsubmit='addPeer(); return false;'>";
html += "<input type='text' id='macInput' placeholder='Enter MAC address (XX:XX:XX:XX:XX:XX)'>";
html += "<button type='submit'>Add Peer</button>";
html += "</form>";
html += "</div>";

// Lisatud MAC aadresside nimekirja
html += "<div id='peerListContainer' style='margin-bottom: 20px;'>";
html += "<h4>Connected Peers</h4>";
html += "<ul id='peerList'></ul>";
html += "</div>";

// Dünaamiline klass lisatud juhita-seadmete kaartide loomiseks
html += "<div class='cards' id='dynamicContent'></div>";

html += "</div>";

// Dünaamilise sisu kood, funktsioonid
html += "<script>";
html += "var currentBoardCount = 0;";

// Päringu saatmise funktsioon
html += "function sendQuery() {";
html += "  fetch('/send_query', { method: 'GET' });";

```

```

html += " .then(response => response.text());";
html += " .then(data => console.log('Query sent:', data));";
html += " .catch(error => console.error('Error sending query:', error));";
html += " }";

// Juhitav-seadme lisamise funktsioon
html += "function addPeer() {";
html += "  var mac = document.getElementById('macInput').value.trim(); // Sisestatud
MAC address
html += "  if (mac.match(/^[0-9A-F]{2}(:[0-9A-F]{2}){5}$/i)) {"; // Kontrolli formaati
html += "    var peerList = document.getElementById('peerList'); // Juhitav-seadmete
nimekiri
html += "    var peerExists = false; // Kontroll-lipp olemasolu kontrolliks
html += "    for (var i = 0; i < peerList.children.length; i++) {"; // Nimekirja läbikäimine
html += "      var listItem = peerList.children[i].innerText; // Nimekirja objekti tekst
html += "      if (listItem.includes(mac)) {"; // MAC aadressi olemasolu kontroll
html += "        peerExists = true;";
html += "        break;";
html += "      };
html += "    };
html += "    if (!peerExists) {"; // MAC puudub nimekirjast
html += "      currentBoardCount++;";
html += "      var newPeerItem = document.createElement('li'); // Uue nimekirja objekti
loomine
html += "        newPeerItem.innerText = 'Board ' + currentBoardCount + ': ' + mac; //
Objekti tekst
html += "        peerList.appendChild(newPeerItem); // Listi lisamine
html += "        fetch('/add_peer?mac=' + encodeURIComponent(mac), { method: 'GET' });";
html += "          .then(response => response.text());";
html += "          .then(data => console.log('Added peer:', data));";
html += "          .catch(error => console.error('Error adding peer:', error));";
html += "    } else {"; // MAC olemasolul
html += "      console.log('Peer with MAC ' + mac + ' already exists.');"
html += "    };
html += "  } else {"; // Vale formaat
html += "    alert('Invalid MAC address format.');"
html += "  };
html += "};";

html += "var events = new EventSource('/events');"

// Sündmuse kuulaja uue juhitaav-seadme lisamisel, loob uue kaardi
html += "events.addEventListener('peer_event', function(e) {";

```

```

html += " var newPeerData = JSON.parse(e.data);";
html += " updateDynamicContent(newPeerData);";
html += "});";

// Sündmuse kuulaja uute andmete korral uuendatakse vastavat kaarti
html += "events.addListener('new_readings', function(e) {";
html += " var reading = JSON.parse(e.data);";
html += " updateDynamicContent(reading);";
html += "});";

// Dünaamilise sisu uuendamine
html += "function updateDynamicContent(reading) {";
html += " var dynamicContent = document.getElementById('dynamicContent');";
html += " var existingCard = document.getElementById('card_' + reading.peerCount);";

// Kaardi olemasolu kontroll
html += " if (existingCard) {";
html += "     var voltageElement = existingCard.querySelector('.voltage-value');";
html += "     var boardIdElement = existingCard.querySelector('.board-id');";
html += "     var responseElement = existingCard.querySelector('.response-id');";
html += "     var statusElement = existingCard.querySelector('.status');";

// Kaardi olemasolul uuendatakse sisu
html += "     voltageElement.innerHTML = reading.voltage.toFixed(2);";
html += "     boardIdElement.innerHTML = reading.id; // Insert the board ID
html += "     responseElement.innerHTML = reading.responseID;";

// Staatusu uuendamine sõltuvalt vastuse ID-st
html += "     if (reading.responseID < 6) {";
html += "         statusElement.innerHTML = 'Online';";
html += "         statusElement.style.color = 'green';";
html += "     } else if (reading.responseID == 6) {";
html += "         statusElement.innerHTML = 'Offline';";
html += "         statusElement.style.color = 'red';";
html += "     }";
html += " } else {";

// Uue kaardi genereerimise vajadusel populeerida baasandmetega
html += "     var newCard = document.createElement('div');";
html += "     newCard.className = 'card voltage';";
html += "     newCard.id = 'card_' + reading.peerCount;";

// Saadud andmete kuvamine

```



```

    html += "      newCard.innerHTML = `

#### Board #${reading.peerCount} - Voltage</h4>` +"; html += " `


```

```

Serial.println(WiFi.localIP());
Serial.print("Wi-Fi Channel: ");
Serial.println(WiFi.channel());

WiFi.mode(WIFI_AP_STA);

// ESP-NOW
if (esp_now_init() != ESP_OK) {
  Serial.println("Error initializing ESP-NOW");
  return;
}
else {
  Serial.println("ESP-NOW initialized");
}

// Tagasikutsete registreerimine
esp_now_register_send_cb(OnDataSent);
esp_now_register_rcv_cb(onDataReceived);

// Kasutajaliidese päringu saatmise initsialiseerimine
server.on("/send_query", HTTP_GET, [](AsyncWebServerRequest *request) {
  sendQueryWithReset();
  request->send(200, "text/plain", "Query sent");
});

// Kasutajaliidese uue juhitav-seadme lisamine
server.on("/add_peer", HTTP_GET, [](AsyncWebServerRequest *request) {
  if (request->hasParam("mac")) {
    String mac = request->getParam("mac")->value();
    addPeerByMAC(mac);
    request->send(200, "text/plain", "Added peer with MAC: " + mac);
  } else {
    request->send(400, "text/plain", "MAC parameter missing");
  }
});

// Kasutajaliidese loomine
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
  request->send(200, "text/html", generateDynamicHTML());
});

// Kasutajaliidese ühenduvuse kontroll
events.onConnect([](AsyncEventSourceClient *client){

```

```

if(client->lastId()){
    Serial.printf("Client reconnected! Last message ID that it got is: %u\n", client->lastId());
}
client->send("hello!", NULL, millis(), 10000);
});

// Kasutajaliidese initsialiseerimine
server.addHandler(&events);
server.begin();
}

void loop() {
    // Korduv automaatne päringu saatmine
    if (currentPeerCount != 0) {
        if ((millis() - lastQueryTime) > QUERY_INTERVAL_MS) {
            lastQueryTime = millis();
            for (int i = 0; i < currentPeerCount; i++) {
                slaveData[i].checkForDelivery = false;
            }
            sendQuery(1);
        }
    }
    else {
        if ((millis() - lastQueryTime) > QUERY_INTERVAL_MS) {
            Serial.println("No peers added, no point to ping!");
            lastQueryTime = millis();
        }
    }
}

// Automaatne kasutajaliidese ühenduse kontroll
static unsigned long lastEventTime = millis();
static const unsigned long EVENT_INTERVAL_MS = 60000;
if ((millis() - lastEventTime) > EVENT_INTERVAL_MS) {
    events.send("ping",NULL,millis());
    lastEventTime = millis();
}
}

```

## LISAD

## Lisa 2 Juhitav-seadme kood

```
/*
 * Nimetus: Ehitusaegse elektritoite automatiseeritud jälgimine - Juhitav-seade
 * Autor: Jan Robin Lauk
 * Kuupäev: 30.04.2024
 */

/*
 * Päringu ID tuvastuslegend:
 * 1 - Esmane päringu ID
 * 2 - Juhitav-seadmeni mittejõudmise korral edastatav ID
 * 3 - Juhtiv-seade ei saanud kedagi kätte, edastame päringu
 * 4 - Ulatuses kaugemalt tulnud juhitav-seadme andmed
 * 5 - Päringu 1 saamisel enda andmete tagasisaatmine
 */

// Vajalikud teegid
#include <esp_now.h>
#include <esp_wifi.h>
#include <WiFi.h>

#define DEVICES 6 // Võrgustiku kogu seadmete arv

const int pin = 34; // Analoogviik pinge lugemiseks

const int port = 8888; // Kommunikeerimise port

int ID; // Seadme isiklik ID
int forwardID; // Päringu edastamise ID
bool pingingQuery = false; // Edastuse kontrolli muutuja
int32_t channel = 2; // ESP-NOW kanal, peab vastama Juhtiv-seadme Wi-Fi kanaliga

// Massiivide puhastuse intervall
static unsigned long lastClearTime = millis();
static const unsigned long CLEAR_INTERVAL_MS = 5000;

// ID massiiv seadmetele, millele on toimunud edastus
int forwardedPeerIDs[DEVICES] = {0};

// Struktuur sissetulevatele andmetele
typedef struct incQueryData {
    int queryID;
```

```

    int masterID;
    float value;
} incQueryData;

incQueryData queryData;

// Struktuur seadme enda andmetele
struct myData{
    int responseID;
    int slaveID;
    float analogValue;
}myData;

// Ajutine struktuur kontrolliks
struct tempData{
    int queryID;
    int boardID;
    float value;
};

tempData temp;

// Kõik süsteemis olevate seadmete MAC aadressid
const uint8_t macAddresses[][6] = {
    {0xA0, 0xA3, 0xB3, 0x2C, 0xF8, 0xD8}, // Juhtiv-seadme MAC
    {0xA0, 0xA3, 0xB3, 0x2C, 0xF1, 0xD4}, // Juhitav- seadme MAC 1
    {0xA0, 0xA3, 0xB3, 0x2D, 0x1E, 0xC8}, // Juhitav- seadme MAC 2
    {0xA0, 0xA3, 0xB3, 0x2D, 0x34, 0xB4}, // Juhitav- seadme MAC 3
    {0xA0, 0xA3, 0xB3, 0x2D, 0x2A, 0xA0}, // Juhitav- seadme MAC 4
    {0xA0, 0xA3, 0xB3, 0x2C, 0xF1, 0xF0} // Juhitav- seadme MAC 5
};

// Massiivide lähtestamine
void resetArrays() {
    Serial.println("Resetting arrays.");
    pingingQuery = false;
    for (int i = 0; i < DEVICES; ++i) {
        forwardedPeerIDs[i] = 0;
    }
}

// Edastuse kontroll
bool isForwarded(int slaveID) {

```

```

for (int i = 0; i < DEVICES; i++) {
    if (forwardedPeerIDs[i] == slaveID) {
        return true;
    }
}
return false;
}

// Edasi saadetud
void markAsForwarded(int slaveID) {
    if (slaveID != ID) {
        Serial.printf("Board %d data has been forwarded and stored in array\n", slaveID);
    }
    else {
        Serial.println("Own data has been forwarded and stored in array");
    }
    forwardedPeerIDs[slaveID - 1] = slaveID;
}

// Tagasikutse andmete saamisel
void onDataReceived(const uint8_t * mac, const uint8_t *incomingData, int len) {
    Serial.println("Data received, manipulating...");
    memcpy(&queryData, incomingData, sizeof(incQueryData));
    Serial.print("Query ID received: ");
    Serial.println(queryData.queryID);
    // Sissetulnud päring ID 1 Juhtiv-seadmelt
    // Enda andmete saatmine ID-ga 5, vajadusel massiivide lähtestamine
    if (queryData.queryID == 1){
        if ((millis()-lastClearTime) > CLEAR_INTERVAL_MS) {
            lastClearTime = millis();
            resetArrays();
        }

        if (!isForwarded(ID)) {
            markAsForwarded(ID);
            myData.responseID = 5;
            sendResponse();
        } else {
            int timeLeft = (CLEAR_INTERVAL_MS - (millis()-lastClearTime))/1000;
            Serial.printf("Data has already been sent, wait %d seconds to reacquire\n", timeLeft);
        }
    }
}

// Päringu ID 2, edastamine ID-ga 3

```

```

else if (queryData.queryID == 2){
  if (!pingingQuery) {
    forwardID = 3;
    forwardQuery(forwardID);
  } else {
    Serial.println("Already forwarded query ID 3 to further nodes!");
  }
}
// Päringu ID 3, enda andmete saatmine ID-ga 4
else if (queryData.queryID == 3){
  if ((millis()-lastClearTime) > CLEAR_INTERVAL_MS) {
    lastClearTime = millis();
    resetArrays();
  }
  if (!isForwarded(ID)) {
    markAsForwarded(ID);
    myData.responseID = 4;
    sendResponse();
  } else {
    int timeLeft = (CLEAR_INTERVAL_MS - (millis()-lastClearTime))/1000;
    Serial.printf("Data has already been sent, wait %d seconds to reacquire\n", timeLeft);
  }
}
// Päringu ID 4, saadud andmete edastamine sama ID-ga
else if (queryData.queryID == 4) {
  if (!isForwarded(queryData.masterID)) {
    markAsForwarded(queryData.masterID);
    forwardID = 4;
    temp.queryID = forwardID;
    temp.boardID = queryData.masterID;
    temp.value = queryData.value;
    forwardQuery(forwardID);
  }
  else {
    Serial.print("Already forwarded this Slave ID's data: ");
    Serial.println(queryData.masterID);
  }
}
}

// Tagasikutse andmete saatmisel
void onDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
  char macStr[18];

```

```

Serial.print("Packet to: ");
// Saadetud MAC aadressi kuvamine
snprintf(macStr, sizeof(macStr), "%02x:%02x:%02x:%02x:%02x:%02x",
         mac_addr[0], mac_addr[1], mac_addr[2], mac_addr[3], mac_addr[4], mac_addr[5]);
Serial.print(macStr);
Serial.print(" send status:\t");
Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
// Andmete saatmise ebaõnnestumisel päringu ID 2 edastamine
if(status != ESP_NOW_SEND_SUCCESS && pingingQuery == false && queryData.queryID ==
3){
    forwardID = 2;
    pingingQuery = true;
    Serial.print("Slave can't reach a node, forwarding Query ID 2: ");
    Serial.println(pingingQuery);
    forwardQuery(forwardID);
}
}

// Ülesseadmine
void setup() {
    Serial.begin(115200);

    // Wi-Fi
    WiFi.mode(WIFI_STA);
    // Wi-Fi kanali muutmise
    WiFi.printDiag(Serial);
    esp_wifi_set_promiscuous(true);
    esp_wifi_set_channel(channel, WIFI_SECOND_CHAN_NONE);
    esp_wifi_set_promiscuous(false);
    WiFi.printDiag(Serial);

    // ESP-NOW
    if (esp_now_init() != ESP_OK) {
        Serial.println("Error initializing ESP-NOW");
        return;
    }
    else {
        Serial.println("ESP-NOW initialized");
    }
}

// Tagasikutse funktsioonide registreerimine
if(esp_now_register_send_cb(onDataSent) == ESP_OK) {
    Serial.println("Send callback registered successfully");
}

```



```

}
else {
    Serial.println("Send callback registering failed");
}
if (esp_now_register_recv_cb(onDataReceived) == ESP_OK) {
    Serial.println("Receive callback registered successfully");
}
else {
    Serial.println("Receive callback registering failed");
}

// Teiste seadmete lisamine võrgustikku
addPeers();
}

void loop() {

}

// Teiste seadmete lisamine
void addPeers() {
    esp_now_peer_info_t peerInfo;
    peerInfo.ifidx = WIFI_IF_STA;
    peerInfo.encrypt = false;
    peerInfo.channel = channel;
    Serial.print("peerInfo ifidx: ");
    Serial.println(peerInfo.ifidx);
    Serial.print("peerInfo channel: ");
    Serial.println(peerInfo.channel);
    Serial.print("peerInfo encrypt: ");
    Serial.println(peerInfo.encrypt);
    uint8_t currentMac[6];
    WiFi.macAddress(currentMac);

    Serial.print("Current MAC address: ");
    for (int i = 0; i < 6; i++) {
        if (i > 0) Serial.print(":");
        Serial.print(String(currentMac[i], HEX));
    }
    Serial.println();

    // Teadaolevate MAC aadresside läbi itereerimine
    for (int i = 0; i < sizeof(macAddresses) / sizeof(macAddresses[0]); i++) {

```

```

// Kontroll, et ei oleks tegemist käesoleva seadmega, muul juhul siduda
if (memcmp(macAddresses[i], currentMac, 6) != 0) {
    memcpy(peerInfo.peer_addr, macAddresses[i], 6);

    if (esp_now_add_peer(&peerInfo) == ESP_OK) {
        if (i == 0) {
            Serial.print("Main device added: ");
        }
        else {
            Serial.print("Peer added: ");
        }
        for (int j = 0; j < 6; ++j) {
            if (j > 0) Serial.print(":");
            Serial.print(String(macAddresses[i][j], HEX));
        }
        Serial.println();
    } else {
        Serial.println("Failed to add peer.");
    }
}
else {
    // Seadme ID määramine
    ID = i;
}
Serial.print("This board's ID: ");
Serial.println(ID);
}

// Vastuse saatmine vajalike andmetega
void sendResponse() {
    myData.slaveID = ID; // Adjust as needed
    float tegur = 1241;
    float voltage = analogRead(pin)/tegur;
    myData.analogValue = voltage;
    Serial.println("Sending response:");
    Serial.print("Response query ID: ");
    Serial.println(myData.responseID);
    Serial.print("Board ID: ");
    Serial.println(myData.slaveID);
    Serial.print("Voltage: ");
    Serial.println(myData.analogValue);
    esp_err_t result = esp_now_send(0, (uint8_t *) &myData, sizeof(myData));
}

```

```

if (result == ESP_OK) {
    Serial.println("Sent with success");
}
else {
    Serial.println("Error sending the data");
    Serial.print("esp_now_send result: ");
    Serial.println(result);
}
delay(2000);
}

// Päringu edastamine vajalike andmetega
void forwardQuery(int forwardID) {
    // Päringu edastamine
    if (queryData.queryID == 2 || queryData.queryID == 3) {
        queryData.queryID = forwardID;
        Serial.print("Forwarding query with ID: ");
        Serial.println(queryData.queryID);
        esp_err_t result = esp_now_send(0, (uint8_t *) &queryData, sizeof(incQueryData));
    }
    // Saadud andmete edastamine
    else if (queryData.queryID == 4){
        esp_err_t result = esp_now_send(0, (uint8_t *) &temp, sizeof(tempData));
    }
    else {
        Serial.println("Data has already been forwarded once!");
    }
}
}

```



**Absolute Maximum Ratings**

Symbol	Parameter		Value	Unit
V <sub>VO</sub>	Input-Output Voltage Differential		40	V
T <sub>LEAD</sub>	Lead Temperature		230	°C
P <sub>D</sub>	Power Dissipation	TO-220	Internally limited	W
		TO-252	2	
		SOT-223	1	
T <sub>J</sub>	Operating Junction Temperature Range		0~125	°C
T <sub>STG</sub>	Storage Temperature Range		-55~125	
ΔV <sub>VO</sub> /ΔT	Temperature Coefficient of Output Voltage		±0.02	%/°C

**ELECTRICAL CHARACTERISTICS**

(V<sub>O</sub>=V<sub>I</sub>-5V<sub>D</sub>±0.5A)      \*T<sub>J</sub>=125°C, I<sub>MAX</sub>=1.5A, P<sub>D</sub>MAX=20W (unless otherwise specified)

Parameter	Symbol	Test conditions	MIN	TYP	MAX	UNIT
Line Regulation(note1)	R <sub>line</sub>	I <sub>A</sub> =25 °C 3V ≤ V <sub>I</sub> -V <sub>O</sub> ≤ 40V		0.01	0.04	%V
		3V ≤ V <sub>I</sub> -V <sub>O</sub> ≤ 40V		0.02	0.07	
Load Regulation(note1)	R <sub>load</sub>	I <sub>A</sub> =25 °C, 10mA ≤ I <sub>O</sub> ≤ I <sub>MAX</sub> V <sub>O</sub> < 5V V <sub>O</sub> ≥ 5V		18 0.4	25 0.5	mV
		10mA ≤ I <sub>O</sub> ≤ I <sub>MAX</sub> V <sub>O</sub> < 5V V <sub>O</sub> ≥ 5 V		40 0.8	70 1.5	
Adjustable Pin Current	I <sub>ADJ</sub>	-		46	100	μA
Adjustable Pin Current Change	ΔI <sub>ADJ</sub>	3V ≤ V <sub>I</sub> -V <sub>O</sub> ≤ 40 V 10mA ≤ I <sub>O</sub> ≤ I <sub>MAX</sub> , P <sub>D</sub> ≤ P <sub>MAX</sub>		2.0	5	μA
Reference Voltage	V <sub>REF</sub>	3V ≤ V <sub>I</sub> -V <sub>O</sub> ≤ 40V 10mA ≤ I <sub>O</sub> ≤ I <sub>MAX</sub> , P <sub>D</sub> ≤ P <sub>MAX</sub>	1.20	1.25	1.30	V
Temperature Stability	ST	-		0.7		%V <sub>O</sub>
Minimum Load Current to Maintain Regulation	I <sub>L(MIN)</sub>	V <sub>I</sub> -V <sub>O</sub> = 4V		3.5	12	mA
Maximum Output Current	I <sub>O(MAX)</sub>	V <sub>I</sub> -V <sub>O</sub> ≤ V <sub>I</sub> -V <sub>O</sub> (P <sub>D</sub> MAX) V <sub>I</sub> -V <sub>O</sub> ≤ 40V, P <sub>D</sub> ≤ P <sub>MAX</sub> T <sub>J</sub> = 25 °C	1.0	2.2 0.3		A
RMS Noise,% of V <sub>O</sub> UT	e <sub>N</sub>	I <sub>A</sub> =25 °C, 10Hz ≤ f ≤ 10KHz		0.003	0.01	%V <sub>O</sub>
Ripple Rejection	RR	100-10KHz without C <sub>ADJ</sub> C <sub>ADJ</sub> = (1-10)μF	66	60 75		dB
Long-Term Stability,T <sub>J</sub> =110KΩ	ST	T <sub>J</sub> = 25 °C for end point measurements, 1 0 0 0 HR		0.3	1	%
Thermal Resistance Junction to case	θ <sub>JC</sub>	-		5		°C/W

**Notes:**

1. Load and line regulation are specified at constant junction temperature. Change in V<sub>O</sub> due to heating effects must

be taken into account separately. Pulse testing with low duty is used.(P<sub>MAX</sub>=20W)

Joonis 0.1 Komponendi LM317T andmeleht [17]

