



TALLINNA TEHNIKAÜLIKOOL  
INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

EE40LT

## **Patareitoitel WiFi andmesidega anduriplatvorm**

BAKALAUREUSETÖÖ

MEHHATROONIKA ÕPPEKAVA

Üliõpilane: Steven Marga

Üliõpilaskood: 120822MAHB

Juhendaja: Ervin Piigli

2017

## AUTORIDEKLARATSIOON

Deklareerin, et käesolev lõputöö on minu iseseisva töö tulemus.

Esitatud materjalide põhjal ei ole varem akadeemilist kraadi taotletud.

Lõputöös kasutatud kõik teiste autorite tööd ja seisukohad ning materjalid on varustatud vastavate viidetega.

Töö valmis Ervin Piigli juhendamisel

”.....“ ..... 201....a.

Töö autor: .....

/allkiri/

Töö vastab lõputööle esitatavatele nõuetele

”.....“ ..... 201....a.

Juhendaja: .....

/allkiri/

Lubatud kaitsmisele

”.....“ ..... 201....a.

..... õppekava lõputööde kaitsmiskomisjoni esimees: .....

/allkiri/

TTÜ inseneriteaduskond  
**BSC LÕPUTÖÖ ÜLESANNE**  
2016/2017 aasta 2. semester

Üliõpilane: Steven Marga 120822MAHB  
Õppekava: MAHB02/09  
Eriala: Mehhatroonika  
Juhendaja: MSc Ervin Piigli

**LÕPUTÖÖ TEEMA:**

(eesti keeles) Patareitoitel WiFi andmesidega anduriplatvorm

(inglise keeles) Battery powered sensor platform with WiFi communication

**Töös lahendatavad ülesanded ja nende täitmise ajakava:**

Nr	Ülesande kirjeldus	Täitmise tähtaeg
1.	Turul olemasolevate seadmete lühikirjeldus ja võrdlus. Töös disainitavale anduriplatvormile esitatavate nõuete väljatootmine.	03.04.2017
2.	Peamiste elektroonikakomponentide valik arvestades nõutavaid parameetreid ning majanduslikku otstarbekust. Prototüüpskeemil süsteemi programmeerimine ja testimine.	17.04.2017
3.	Elektroonikaskeemi lõplik väljatootamine ja disain. Elektroonikaskeemi baasil trükkplaadi disain.	24.04.2017
4.	Korpuse disain anduriplatvormile arvestades nõudeid seadme esteetilisele välimusele ning ohutusele.	08.05.2017
5.	Töös väljatootatud seadmele hinnangu andmine, ideed tulevikus süsteemi täiendamiseks. Majandusanalüüsi tegemine.	15.05.2017

**Lahendatavad insenertehnilised ja majanduslikud probleemid:**

Antud töös töötatakse välja patareitoitel anduriplatvorm, mille eesmärk on mõõta ümbritseva õhu temperatuuri ja suhtelist õhuniiskust ning edastada mõõtetulemused üle WiFi andmeside. Peamisteks lahendatavateks probleemideks on süsteemi energiatarbe minimeerimine ning konkurentsivõimelise toote omahinna ja omaduste suhte saavutamine.

**Töö keel:** eesti keel

Kaitsmistootlus esitada hiljemalt 15.05.2017

**Töö esitamise tähtaeg** 25.05.2017

**Üliõpilane** Steven Marga

/allkiri/.....

kuupäev: 08.05.2017

**Juhendaja** Ervin Piigli

/allkiri/.....

kuupäev: 08.05.2017

Konfidentsiaalsusnõuded ja muud ettevõttepoolsed tingimused formuleeritakse pöördel

# Sisukord

Autorideklaratsioon .....	2
Eessõna.....	7
Sissejuhatus .....	8
1. Turul olemasolevate seadmete võrdlus ning nõuded projekteeritavale seadmele .....	9
2. Juhtmevabade tehnoloogiate võrdlus.....	10
3. Võtmekomponentide valik anduriplatvormile .....	11
3.1 WiFi mooduli valik.....	11
3.2 Temperatuuri ja suhtelise õhuniiskuse anduri valik.....	12
4. Prototüüpskeemil süsteemi programmeerimine ja testimine .....	14
4.1 Testmoodulite valik .....	14
4.2 Süsteemi programmeerimine .....	15
4.2.3 Moodulite testimine ESP-IDF raamistikuga .....	16
4.2.4 Arduinoga programmeerimine.....	17
4.2.5 Energiatarbe minimeerimise algoritm .....	18
5. Elektroonikaskeemi ning trükkplaadi disain, töösükli arvutamine .....	20
5.1 Komponentide koostamine.....	20
5.2 Süsteemi pinge reguleerimine .....	21
5.2.1 Vooluallikas.....	22
5.3 Elektriskeemi koostamine .....	22
5.3.1 WiFi elektriskeem .....	22
5.3.2 Anduri elektriskeem .....	24
5.3.3 Boost-muunduri skeem.....	25
5.4 Trükkplaadi disain.....	28
5.5 Töösükli arvutus.....	30
6. Korpuse disain ning ohutus .....	32
6.1 Disain ning korpuse elemendid .....	32
6.1.1 Korpus .....	32
6.1.2 Korpuse kaas ning lisad.....	34
6.2 Seadme ohutus .....	35
7. Majandusanalüüs, seadme hinnang ning ideed süsteemi täiendamiseks .....	36

7.1 Majandusanalüüs.....	36
7.2 Hinnang seadmele.....	37
7.3 Ideed süsteemi täiendamiseks.....	37
Kokkuvõte.....	38
Summary.....	39
Kasutatud kirjandus.....	40
LISAD.....	45
Lisa 1 Lähtekood I <sup>2</sup> C seadme aadressi testimiseks.....	46
Lisa 2 Anduriplatvormi lõplik välja arendatud lähtekood.....	47
Lisa 3 Anduriplatvormi elektriskeem.....	50
Lisa 4 Korpuse kaas.....	51
Lisa 5 Distantpuks.....	52
Lisa 6 Patareipesa hoidja.....	53
Lisa 7 Korpus.....	54

## Selede ja tabelite loetelu

Tabel 1.1 - Erinevate anduriplatvormide võrdlus .....	9
Sele 3.1 – ESP-WROOM-32 WiFi moodul pealt (vasakul) ning alt (paremal) koos sisendite ja väljunditega.....	11
Tabel 3.1 – DHT11 ning DHT22 andurite kirjeldus .....	12
Sele 3.2 – Si7021 andur pealt poolt (vasakul) ning alt koos nähtavate I <sup>2</sup> C viikudega (paremal) .....	13
Sele 4.1 – ESP32 WiFi arendusmoodul (vasakul) ning Si7021 anduriplaat (paremal).....	14
Sele 4.2 – Anduri näitude saatmine Interneti.....	17
Sele 4.3 – Energiatarbe algoritmi plokkiagramm .....	19
Sele 5.1 – Kolme tüüpi patarei testimine pingelangule [25] .....	21
Sele 5.2 – Rööbiti ühendusega kahe AA patareipesa .....	22
Sele 5.3 – WiFi mooduli elektriskeem koos vajalike komponentidega .....	23
Sele 5.4 – Si7021 anduri skeem koos vajalike komponentidega .....	24
Sele 5.5 – Mitu I <sup>2</sup> C seadet ühel liinil.....	24
Sele 5.6 – Pingemuunduri skeem koos töötamiseks vajalike komponentidega .....	27
Sele 5.7 – Trükkplaadi 3D mudel pealt vaadatuna .....	28
Sele 5.8 – Trükkplaadi 3D mudel alt vaadatuna.....	29
Sele 5.9 – Trükkplaat koos mõõtudega millimeetrites .....	29
Tabel 5.1 – Erinevate protsesside voolutarbe kirjeldus .....	30
Sele 6.1 – Anduriplatvorm eest vaadatuna .....	32
Sele 6.2 – Korpuse sisemine osa .....	33
Sele 6.3 – Vasakult järjestatuna: korpuse, trükkplaat, distantspuksid, patareipesa hoidja, 2 M3x12 polti, patareipesa, korpuse kaas .....	34
Sele 6.4 – Korpuse kaas vasakul ning paremal kaas koos korpusega.....	34
Tabel 7.1 – Kasutatud komponentide hinnakiri .....	36

# EESSÕNA

Käesolev bakalaureusetöö teema on välja pakutud Tallinna Tehnikaülikooli poolt. Teema valikul said otsustavaks mitmed asjaolud. Tähtsaimaks võib-olla oli see, et asjade interneti puhul on tegu väga huvipakkuva valdkonnaga oma kiire arengu tõttu ning selle kiire arenguga kaasneb võimalus igaühel luua huvitavaid ning kasulikke süsteeme. Näiteks üks selline süsteem on n-õ tark kodu, kus erinevad elektroonikaseadmed ja valgustid on ühendatud ühtsesse võrku, mille kaudu saab nende tööd juhtida nutitelefoniga. Samuti on positiivne see, et sellega tegelemine ei vaja alustades liigselt ressursse.

Eraldi tänatakse töö juhendajat, Ervin Piiglit. Kuna töö autor ei olnud eelnevalt tegelenud elektriskeemide tegemise ning trükkplaatide loomisega, siis juhendaja näpunäited ning nõuanded olid nende koostamisel suureks abiks.

## SISSEJUHATUS

Elektronika hinna langemise ning lihtsama kättesaadavuse tõttu tuuakse turule kogu aeg uusi seadmeid, mille eesmärk on jälgida väliseid signaale ja seda informatsiooni edastada eri viisidel kasutajale. Tehnoloogia kiire arengu tõttu on olemas näiteks käekellad, mis ei näita ainult aega, vaid ka emaille ning teavitavad muutustest sotsiaalvõrgustikes. Kodudes on suur hulk andureid, mis jälgivad keskkonna temperatuuri või teavitavad kui keegi avab kodu ukse või ega lemmikloom toitmata ei ole. Sõltumata nende andurite või seadmete eesmärgist on kõik ühendatud mõne muu seadmega või omavahel üle raadioside ning kokku moodustavad asjade interneti ehk eesti keeles värgvõrgu.

Värgvõrk on Interneti kaudu seotud asjade võrk, kus need kasutaja ja üksteisega informatsiooni jagavad ja vahetavad ning koos teatud ülesandeid täidavad. Nendeks seadmeteks võivad olla nutitelefonid, autod, pesumasinad, hooned jt. Seega võimalike seadmete piir lõppeb inimeste enda kujutlusvõime piirimail. Esimest korda kasutati inglise keelset väljendit *Internet of Things* aastal 1999 [1], kuid esimene seade, mis Internetti ühendati, oli juba 1980ndatel [2] ning selleks oli Coca-Cola müügiautomaat. Nende seadmete arv tõuseb iga päevaselt ning ekspertide hinnangul on aastaks 2020 maailmas värgvõrgu seadmeid umbes 20,8 miljardit [3].

Käesolevas töös tehakse taustauuring ning projekteeritakse WiFi andmesidel anduriplatvormi esmane prototüüp. Süsteemi peamiseks komponendiks valiti universaalne mikrokontroller, mida on võimalik kasutada erinevate andurite ja lisaseadmetega. Täpsemalt keskenduti selles töös temperatuuri ja suhtelise õhuniiskuse mõõtmisele. Kuigi turul on selliseid seadmeid juba palju, siis kohati on nende hind liiga kallis või osade andurite Internetti ühendamiseks on siiski vaja lisaseadmeid. Kasutusmugavuse tõttu töötavad paljud neist seadmetest patareitoitel, et anduriplatvormi kasutus ei oleks piiratud vaid kohtadega, kus on ligipääs vooluvõrgule. Seega turu nõudluse tõttu disainitakse ka antud töös olev anduriplatvorm töötama patareitoitel.

Töö esimeses osas võrreldakse olemasolevaid seadmeid ning selgitatakse nende töötamise põhimõtteid. Seejärel kirjeldatakse peamised lähteülesandest tulenevad nõuded projekteeritavale seadmele. Kui süsteemi võtmekomponendid on valitud ning esialgne plaan koostatud, testitakse süsteemi tööpõhimõtteid prototüüp-plaadil kasutades olemasolevaid arendusmooduleid, mis põhinevad eelnevalt valitud komponentidel. Kuna süsteem koostatakse patareitoitel, siis on tähtsaks osaks energiatarbe miinimumi viimine. Töös kirjeldatakse sellega seotud võimalikke probleeme kui eesmärgiks on kasutada WiFi andmeside ning kuidas neid probleeme lahendada. Seadmele koostatakse elektriskeem ja selle baasil luuakse ka trükkplaat ning kogu elektroonikale projekteeritakse korpus. Töö viimases osas tehakse majandusanalüüs ning näidatakse, mis moodustab suurema osa kuludest. Samuti kirjutatakse lühidalt ohutusest, antakse hinnang kogu tööle ning tuuakse välja võimalikud täiendused süsteemi efektiivsemaks muutmiseks. Töö koostamisel kasutati järgnevaid programme: DipTrace, Solidworks 2016, DrawIO, Simplify3D, Arduino, Eclipse, Microsoft Paint.



# 1. TURUL OLEMASOLEVATE SEADMETE VÕRDLUS NING NÕUDED PROJEKTEERITAVALE SEADMELE

Töö esimeseks ülesandeks oli uurida seadmeid, mis juba saadaval on. Sarnaseid seadmeid on turul tegelikult palju erinevate tootjate poolt. Kuna antud töös koostatakse anduriplatvorm, kus anduriteks on temperatuuri ning suhtelise õhuniiskuse andur, siis erinevaid valmis tooteid võrreldi ka täpselt nende andurite põhjal. Vaadates erinevaid tooteid, jäi esmapilgul silma ainult kaks erinevust: nende välimus ja hind. Enamik seadmeid on tehtud ka patareitoitel töötama tänu nende mitmekülgsetele kasutusvõimalustele ning see ei ole ka enam eriline müügiargument. Andmaks ülevaade sellistest seadmetest on näiteks toodud (tabel 1.1) kolm turul eksisteerivat anduriplatvormi.

Tabel 1.1 - Erinevate anduriplatvormide võrdlus

Tootja ja mudel	Hind	Juhtmevaba tehnoloogia	Head küljed	Halvad küljed
Xiaomi Mi Smart Temperature and Humidity Sensor	10,7€ [4]	ZigBee	+ Väga odav + Väiksed mõõtmed + patareitoitel	- Vajab eraldi juurde Mi Smart WiFi kesksel kontrolleri, (28€) - Vajab töötamiseks 220 – 240V, turvalisuse risk
ConnectSense CS-TH Wireless Temperature and Humidity Sensor	134€ [5]	WiFi	+ Töötab nii 5V USB toitega kui ka patareitoitel	- Suured mõõtmed - Silmapaistmatu disain
Samsung SmartThings Moisture Sensor	27.9€ [6]	ZigBee	+ Väike patarei, + ilus disain, + väiksed mõõtmed	- Vajab juurde kesksel Samsungi Hub kontrolleri (€99, ~115€) - Hub vajab juurde vooluadapterit

Tooteid lähemalt analüüsid selgub, et kõik neist ei kasuta WiFi andmeside tehnoloogiat info edastamiseks vaid teist juhtmevaba andmeedastust nimega ZigBee. Nende kolme toote võrdlusest selgub, et kui tahta WiFi andmesidel töötavat anduriplatvormi, siis läheb esmakordse ostu puhul hind kõrgeks just lisaseadmete vajaduse tõttu. Ostes seadme, mis toetab koheselt WiFi andmeside, pole juurde vaja lisaseadmeid, kuid hind on selle võrra ka kallim.

**Nõuded** seega töös projekteeritavale anduriplatvormile on edastada juhtmevabalt temperatuuri ja suhtelise õhuniiskuse näite. Andmete edastus peab töötama üle WiFi andmeside ning seda ilma lisaseadmeid kasutamata. Samuti peab seade töötama patareitoitel ning sellest tulenevalt peab tegelema energiatarbe miinimumi viimisega.

## 2. JUHTMEVABADE TEHNOLOOGIATE VÕRDLUS

ZigBee tehnoloogia edastab infot raadiolainetega sagedustel 868 MHz, 900 MHz ja 2.4 GHz ning on piiratud WPAN võrgu kasutusega. Võrgu leviala jääb 20-50 meetri vahemikku. Võrku iseloomustab see, et kõik seadmed selles võrgus on omavahel ühenduses erinevalt näiteks WiFi võrgust, kus kõik seadmed ühenduvad kesksesse võrku. Antud tehnoloogia andmeedastuse maksimaalseks kiiruseks on ainult 250 kbps. Parim selle tehnoloogia omadus on väga väike energiatarve, mis muudab selle sobilikuks seadmetele, mis töötavad patareitoitel. [7]

WiFi tehnoloogia kasutab andmete edastamiseks raadiolainete sagedust 2.4 GHz ja 5.8 GHz. Nagu eelnevalt mainitud, siis WiFi seadmed ühenduvad kõik kesksesse võrku ja suudavad kasutada PAN ja WLAN võrke 30-100 meetri ulatuses. WiFi tehnoloogial on mitmeid erinevaid andmeedastus-protokolle, mis erinevad omadustelt põhiliselt just kiiruse poolest ning enamlevinud seadmed suudavad andmeid edastada kiirusega 11-54 mbps. Kuna WiFi ei ole loodud esialgselt töötamiseks seadmetega, mis peaksid töötama patareitoitel, siis WiFi energiatarve on umbes neli korda suurem kui ZigBee tehnoloogia oma. [8]

Arvestades WiFi ja ZigBee erinevusi, siis on selgelt näha, et suur eelis on just ZigBee tehnoloogial. Arenduses on ka WiFi madala energiatarbega protokollid [9], mis oleks otseselt suunatud just sensoritele ning seadmetele, mis töötavad patareitoitel, kuid hetkel neid veel ei eksisteeri. Arvestades antud erinevusi ja esitatud nõudeid, on ka näha, mis muutis käesoleva töö keeruliseks ning miks enamik seadmeid turul kasutab just ZigBee juhtmevaba tehnoloogiat.

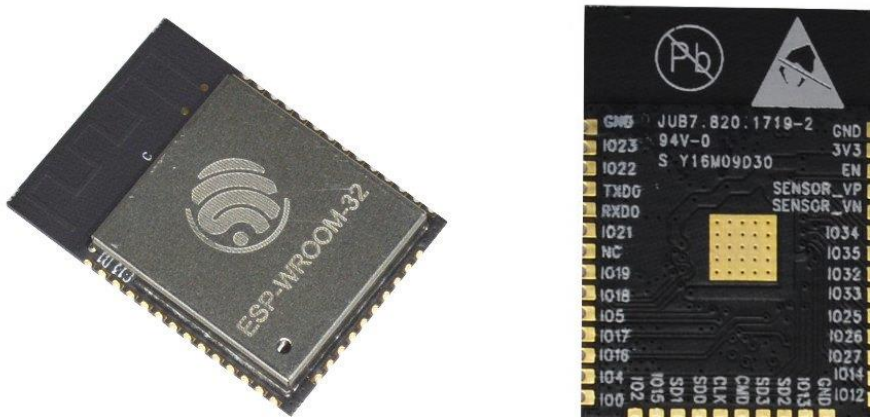
### 3. VÕTMEKOMPONENTIDE VALIK ANDURIPLATVORMILE

Kogu süsteemi kaheks peamiseks komponendiks on WiFi moodul ning temperatuuri ja suhtelise õhuniiskuse andur. Et kogu süsteemi mitte liialt keerukaks ajada, otsustati uurida WiFi mooduleid, mis sisaldavad ka mikrokontrollerit. Mikrokontroller koos WiFi-mooduliga oli piisav, sest koostatud anduriplatvormil ei olnud keerukaid ning palju ressursinõudlikke ülesandeid. Uurides erinevate tootjate poolt pakutavaid lahendusi oli näha, et keskmine hind sellistel moodulitel jääb paarikümne euro vahemikku. Siit selgus ka see, miks on WiFi ühendust kasutavate seadmete hind niivõrd kallis. Hobielektronikute seas on tuntud tootja nimega Espressif, kes toodab mikrokontrolleriga WiFi mooduleid ning seda märgatavalt odavamama lõpphinnaga kui teised turul olevad tootjad.

#### 3.1 WiFi mooduli valik

Täpsemalt hakati uurima Espressif tootenimistus kõige hiljutisemat WiFi moodulit nimega ESP-WROOM-32 [10].

Moodul töötab pingevahemikus 2.2-3.6 V. Lisaks WiFi andmeside erinevatele protokollidele on võimalus kasutada ka Bluetooth versioon 4.2 ühendust. Lisaseadmete ühendamiseks on moodulil palju erinevaid sisendeid ja väljundeid (sele 3.1), mida ka käesolevas töös kasutatakse suhtluseks anduriga. ESP-WROOM-32 mooduli hind jääb 4-8€ vahemikku [11], mis tähendab, et võrreldes teiste saadaval olevate moodulitega on seade mitmeid kordi odavam.



Sele 3.1 – ESP-WROOM-32 WiFi moodul pealt (vasakul) ning alt (paremal) koos sisendite ja väljunditega

## 3.2 Temperatuuri ja suhtelise õhuniiskuse anduri valik

Teiseks põhikomponendiks on temperatuuri ja suhtelise õhuniiskuse andur. Neid on saadaval sadu erinevaid, kui mitte rohkem. Leidub versioone, kus temperatuuri ja suhtelise õhuniiskuse andur on kokku pandud ühte integraalskeemi, aga variante, kus need on eraldi. Mugavuse ja otstarbekuse kohapealt otsiti esialgu andureid, kus on mõlemad ühe kiibi peal koos. Esmalt otsiti andurit n-õ populaarsuse järgi ehk millised on kõige enamkasutatavad. Enamkasutatavate andurite eeliseks on see, et nende kohta leidub palju rohkem info. See tähendab, et keegi on eelnevalt juba ühendanud seda mingi platvormiga ning kõik võimalused ja probleemid on kasutajate poolt ära kirjeldatud. Samuti on populaarsuse suureks eeliseks see, et keegi on kirjutanud sellele andurile ka mõnes programmeerimiskeeles teegi (*library*). Teek on kollektsioon funktsioone ja muid komponente, mis on mõeldud korduvkasutuseks programmides. Teegid sisaldavad samuti juhiseid, kuidas seadmed oma vahel suhtlevad, antud juhul anduri suhtlus mikrokontrolleriga. Kuna juhised sisaldavad bittide muutmist registrites õiges järjekorras ja õigel ajal, siis selliste teekide kirjutamine on küllaltki aeganõudev ning seepärast on võimaluse korral mõistlik kasutada seadmeid, millel on teegid valmis kirjutatud. Otsingu tulemusena jäid silma kaks andurit: DHT11 [12] ning DHT22 [13].

Tabel 3.1 – DHT11 ning DHT22 andurite kirjeldus

Anduri ID	Mõõtmisvahemik	Mõõtmistäpsus	Pinge- vahemik	Andmevahetus	Hind
DHT11	Suhteline õhuniiskus 20-80% RH, temperatuur 0-50 °C	Õhuniiskus 5% RH, temperatuur $\pm 2$ °C	3-5 V	Ühe digitaalväljundiga	5€ [12]
DHT22	Suhteline õhuniiskus 0-100% RH, temperatuur -40-80 °C	Õhuniiskus 2-5% RH, temperatuur $\pm 0.5$ °C	3-5 V	Ühe digitaalväljundiga	10€ [13]

Võrreldes esitatud andureid (tabel 3.1) selgub, et mõlema anduri hind on WiFi mooduliga võrreldes kallis. Andur DHT11 ei olnud antud töös sobiv oma mõõtmistäpsuse tõttu, sest näidu kõikumine  $\pm 2$  °C võrra on liiga suur. Kuna andurid on tehtud kasutajasõbralikus pakendis, et neid oleks lihtsam ühendada süsteemi, siis asuti uurima DHT22 andmelehte [14], et aru saada milline on tegelikult kiip, mis temperatuuri ning suhtelist õhuniiskust mõõdab. Sealt selgus, et suhtelist õhuniiskust mõõdetakse polümeer-kondensaatoriga ning temperatuuri anduriga DS18B20. Nimetatud temperatuuriandur suhtleb üle 1-Wire tehnoloogia, mis antud juhul valitud WiFi moodulile ei sobinud. Samuti oli selle temperatuurianduri ühiku hind 4€, mis on võrreldes WiFi mooduliga siiski liiga kallis.

Edasi otsustati uurida andureid otse elektroonika hulгимүүgi veebilehtedelt. Et leida nendest sadadest erinevatest anduritest välja sobiv, määrati otsingu parameetrid järgnevalt:

1. Temperatuuri- ja niiskusandur ühel integraalskeemil
2. Andmeside mikrokontrolleriga käib üle I<sup>2</sup>C ühenduse (suhtlustehnoloogia, mida WiFi-moodul toetab lisaseadmete ühendamiseks)
3. Andur peab töötama ESP32-ga sama pinge pealt, et ei oleks vaja lisaks pingeregulaatorit (rohkem komponente tähendab suuremat kulu ning keerukamat süsteemi)

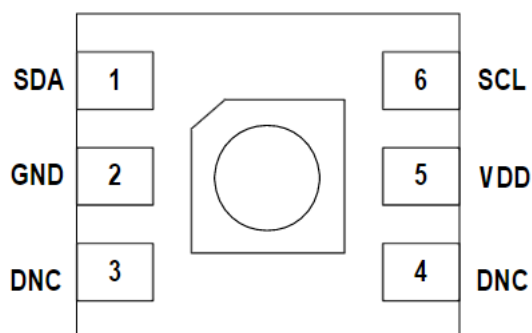
Kuna nende otsingu parameetrite alusel tuli vasteid siiski veel liiga palju, siis järjestati vasted edasi hinna ja populaarsuse järgi.

Valituks osutus Silicon Labs Si7021-A20 andur [15], kus ühe kiibi peal on nii temperatuuri kui ka suhtelise õhuniiskuse mõõtmise.

Anduri tehnilised parameetrid on järgnevad:

- Pingevahemik töötamiseks 1.9 kuni 3.6 V
- Suhtelise õhuniiskuse mõõtepiirkond 0-80% RH ja täpsus  $\pm 3\%$  RH
- Temperatuuri mõõtepiirkond -10 kuni 85°C ja täpsus  $\pm 0.4^\circ\text{C}$
- Tehases eelnevalt kalibreeritud
- Madal volutarve – 150  $\mu\text{A}$  aktiivses olekus ning 60 nA oote-olekus
- Suhtlus käib üle I<sup>2</sup>C tehnoloogia

Vaadates tehnilisi parameetreid, siis oli näha, et andur on sobiv koostatavale anduriplatvormile ning WiFi moodul toetab ka I<sup>2</sup>C suhtluse tehnoloogiat (sele 3.2). Määravaks sai siinkohal veel see, et toote ühikuhind oli aktsepteeritav (hind 3-4€) ning sellele andurile on ka valmis kirjutatud teegid.



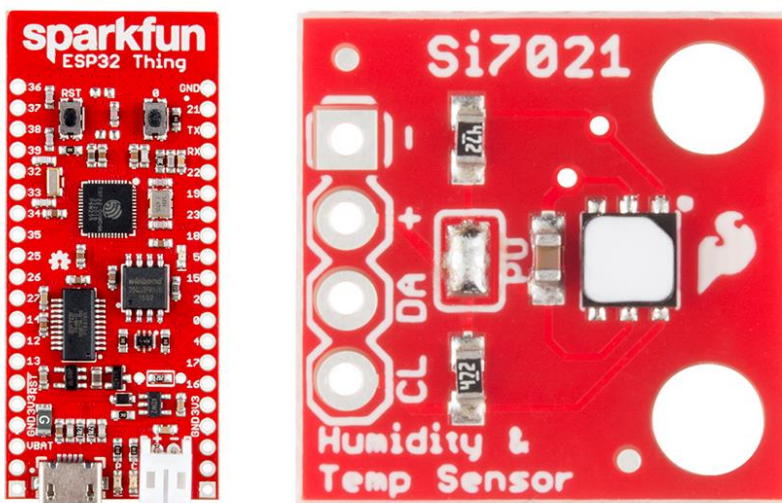
Sele 3.2 – Si7021 andur pealt poolt (vasakul) ning alt koos nähtavate I<sup>2</sup>C viikudega (paremal)

## 4. PROTOTÜÜPSKEEMIL SÜSTEEMI PROGRAMMEERIMINE JA TESTIMINE

Saamaks teada, kas valitud komponentidega on võimalik koostada anduriplatvorm nii nagu plaanis, oli esmalt vaja komponente ning nende tööd testida. Kuna nii andur kui WiFi moodul olid niivõrd populaarsed, telliti elektroonikaseadmete tootja SparkFun kodulehelt trükkplaadid (*breakout board*), kus on vastav moodul või andur peal ning sisendid ja väljundid on lihtsustatud kujul ligipääsetavaks tehtud. Lihtsustatud kuju tähendab, et trükkplaadi on võimalik lihtsalt ühendada teiste süsteemidega ning seda ainult moodulile piikriba klemmide jootmisega. Need on tehtud seetõttu selliselt, et enamik kiibid tulevad erinevates elektroonika pakendites ning mõõtudes, mis muudab prototüüpide koostamise ning testimise aeganõudvaks ning ebamugavaks.

### 4.1 Testmoodulite valik

WiFi moodul, millega testimist alustati on tootjalt SparkFun ning tootenimega ESP32 Thing [16]. Tegu on arendusmooduliga, mis tähendab et trükkplaadil on välja toodud kõik integraalskeemi kiibi viigud (sele 4.1), lisatud vajalikud nupud ning elektroonikakomponendid süsteemi töötamiseks. Samuti on lisatud arendusmoodulile USB ühendus, mis tähendab, et seda on võimalik lihtsalt ühendada arvutiga kirjutatud koodi peale laadimiseks. Selle võimaldamiseks, on trükkplaadile lisatud USB ühenduse konverter, mis muudab WiFi mooduli mikrokontrollerilt tuleva jadasiin (*serial*) signaali universaalseks jadasiin-signaaliks (USB).



Sele 4.1 – ESP32 WiFi arendusmoodul (vasakul) ning Si7021 anduriplaat (paremal)

Temperatuuri ja õhuniiskuse anduri testimiseks kasutati samuti moodulit [17], mis koosneb trükkplaadist, mille peal on eelnevalt välja valitud Si7021 andur ning vajalikud komponendid anduri töötamiseks. Anduri kiibi viigud on samuti välja toodud sellisel kujul, et moodulit oleks lihtne ühendada süsteemi. Kuna anduri ühendus mikrokontrolleriga käib üle I<sup>2</sup>C side, siis väljatoodud viikude arv on ainult neli: SCL ja SDA (I<sup>2</sup>C suhtlusviigud), pluss ning miinus (toiteklemmid).

## 4.2 Süsteemi programmeerimine

ESP32 moodulit on võimalik programmeerida vähemalt kahel erineval viisil ja kasutades C või C++ programmeerimiskeelt. Tootja poolt oli saadaval ESP-IDF (*Espressif's IoT Development Framework*) arenduse raamistik [18]. See on ESP32 ametlik arenduse raamistik, mis annab kasutajale võimaluse programmeerida kõike, milleks moodul on võimeline. Teine viis seadme programmeerimiseks on kasutada Arduino programmeerimiskeskonda [19]. See võimalus on kasutajate kogukonna poolt tehtud, mis tähendab, et ametlik arenduse raamistik on ümber kirjutatud nii, et see oleks kasutatav Arduino keskkonnas. Seda on tehtud sel põhjusel, et kasutajatel oleks võimalikult lihtne oma süsteeme luua. Lihtsus seisneb selles, et kui ametlikku raamistikku kasutada, siis on vaja mikrokontrollerile kordades rohkem instruksioone ette kirjutada kui Arduino keskkonnas, sest Arduinole ümberkirjutatud raamistikus on paljud instruksioonid kirjutatud kokku ühtseteks funktsioonideks. See võimaldab teha sama ülesannet mitmete koodiridade asemel vaid ühe reaga, mis sisaldab endas mingit kindlat funktsiooni.

WiFi mooduli juures väga tähtsaks valiku elemendiks osutus see, et mikrokontrolleril on erinevad töörežiimid. Kuna koostatud töö üheks peamiseks ülesandeks oli energiatarbe minimeerimine, siis võimalus kasutada erinevaid töörežiime oli väga oluline. Tootja poolt välja toodud erinevad töörežiimid [20] on järgnevad:

- Aktiivne režiim (*Active mode*) – raadiosignaalide saatmine ja vastuvõtmine töötab, voolutarve 80-260 mA
- Modemi une režiim (*Modem-sleep mode*) – protsessor töötab kuid modem magab, voolutarve 3-20 mA
- Kerge une režiim (*Light-sleep mode*) – voolutarve 0.8 mA
- Sügava une režiim (*Deep-sleep mode*) – väga madala voolutarbe protsessor (*ULP-coprocessor*) töötab, samuti ka reaalaja taimer ja reaalaja mälu, voolutarve 20 µA - 0.15 mA
- Talveune režiim (*Hibernation mode*) – ainult reaalaja taimer töötab, voolutarve 5 µA

Nagu nimistust näha, siis on mikrokontrolleril palju erinevaid töörežiime. Võrreldes aktiivset ning modemi une režiimi selgus, et voolutarbe erinevus võib olla üle 10 korra. Kuna eesmärk oli andurilt

iga mõne sekundi järel küsida uut infot, siis oli tähtis, et protsessor töötaks kogu aja ning suudaks andurilt tulemaid andmeid vajaduse korral analüüsida. See tähendab, et saadaval olevatest režiimidest oli sobilik modemi une režiim ning voolutarbe koha pealt oli see samuti mõistlik.

### 4.2.3 Moodulite testimine ESP-IDF raamistikuga

Esmalt asuti testima ning programmeerima kasutades ESP-IDF raamistikku. Selleks laaditi vajalikud failid Internetist ning valmistati ette programmeerimiskeskond juhendi järgi [18]. Põhjus miks valiti raskem viis programmeerimiseks oli see, et uurides Arduinole ümber kirjutatud lähtekoodi, selgus, et nimetatud töörežiimidest on hetkel Arduinol võimalik kasutada ainult sügava une režiimi. Kuna see ei ole eelistatud valik, siis tuli kasutada ESP-IDF raamistikku.

Testimist alustati mooduli ühendamisega olemasolevasse WiFi võrku. Selle eesmärk oli teha kindlaks, kas WiFi-moodul on töökorras ning töötab nagu peaks. Selleks kasutati tootja poolt kirjutatud lähtekoodi [21], mis oli näitena raamistiku failidega kaasa pandud. Testi tulemusena selgus, et WiFi moodul on töökorras ning suudab ühenduda olemasolevasse WiFi võrku. Edasi alustati andurimooduli testimist. Et anduriga esialgu ühendust saaks on vaja määrata selle aadress. Kuna andmevahetus anduri ja mikrokontrolleri vahel käib üle I<sup>2</sup>C suhtluse, siis määratakse igale seadmele oma 7-bitine baas alama (*slave*) aadress. See aadress on tootja poolt andmelehele [15] kirja pandud ning Si7021 anduri puhul on selleks 0x40. Kuna I<sup>2</sup>C tehnoloogiaga saab ülemale (*master*) ühendada palju alamaid, siis esitatakse ka aadress kuueteistkümnendsüsteemis mugavuse ja loetavuse pärast.

Esialgseks anduri ühenduvuse testimise baasiks kasutati Internetist saadud lähtekoodi [22] ning muudeti seda vastavalt vajadusele, et saada töötav lähtekood (lisa 1).

Lähtekoodist on näha, kuidas esmalt kaasatakse erinevad teegid kontrolleri suhtlemiseks. Need sisaldavad erinevaid instruktsioone, mis on kirjutatud ühtseteks funktsioonideks. Pildil oleva lähtekoodi sisus on ka näha, kuidas esialgu valmistatakse kontrolleri I<sup>2</sup>C suhtluseks ette n-ö suhtluskanali avamisega ja seejärel on algoritm, mis käib kõik aadressid läbi ning kui mõnelt aadressilt leitakse seade, siis salvestatakse selle asukoht. Kõik see on kirjutatud üheks funktsiooniks, kaasaarvatud suhtluskanali avamine, sest seda tuleb teha iga kord kui I<sup>2</sup>C baasil andmevahetust kasutatakse. Lähtekoodi lõpus on selle sama funktsiooni kutsumine, mis alustab saadud instruktsioonide käivitamist. Testides lähtekoodi WiFi mooduli peal, oli näha, et andmelehel [15] kirjutatud anduri aadress vastab sellega, mis kasutatud kood tulemuseks andis.

Kuna moodul ja andur töötasid, siis järgmisena otsustati testida kontrolleri töörežiimide funktsionaalsust. Eelnevalt töörežiime kirjeldades sai otsustatud, et andmete järgi sobivaimaks on modemi une režiim. Selle testimiseks kasutati lähtekoodi [23], mille tootja oli näitena kaasanud.

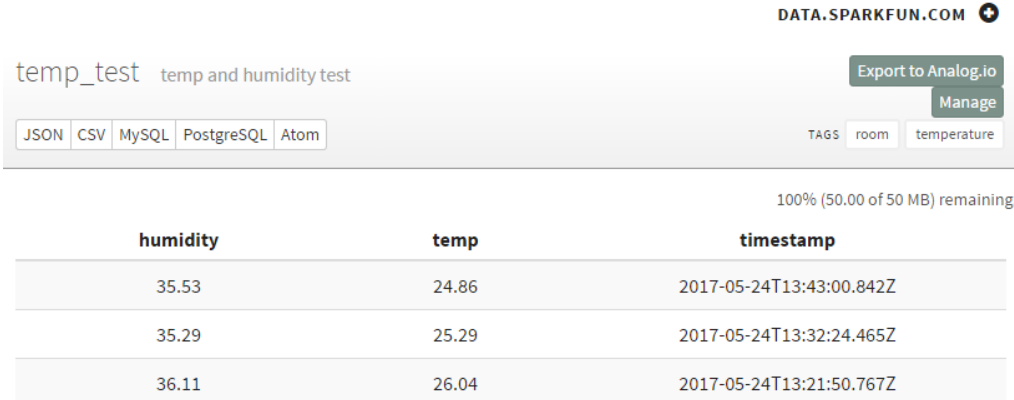


Testimine kukkus teadmata põhjusel läbi, sest mikrokontroller ei sisenenud une režiimi ning edasi tuli uurida tootja kodulehelt ning ametlikest foorumitest, kus seisneb viga. Selgus, et kuigi funktsionaalsus oli olemas ning kasutajate jaoks väljakirjutatud, siis tegelikult sel hetkel mikrokontroller seda ei võimaldanud ning kasutatav oli ainult sügava une režiim [24].

## 4.2.4 Arduinoga programmeerimine

Vajaliku töörežiimi puudumise tõttu ei olnud enam otsest vajadust kasutada keerukat ning aeganõudvat ESP-IDF raamistikku ning otsustati jätkata Arduino programmeerimiskeskkonnas kasutades ümberkirjutatud raamistikku [19]. Eelnevaid testide tulemusi arvestades asuti esimesena testimise sügava une režiimi ning selle korrektset töötamist. Kui eelnevalt räägiti Arduino keskkonnas programmeerimise lihtsusest, siis modemi une režiimi testides väljendus see selgelt: võrreldes kümnete ridade koodiga ESP-IDF raamistikus, on võimalik Arduino keskkonnas kasutada seda funktsionaalsust vaid kahe koodireaga. Test osutus läbituks ning WiFi moodul suutis siseneda sügava une režiimi määratud ajaks.

Edasi testiti kõiki kolme komponenti koos, et näha kas süsteem suudab töötada ka tervikuna nii nagu eesmärk oli. Selle jaoks kirjutati Internetist saadud näidete baasil kood, mis paneb WiFi mooduli ühenduma olemasolevasse WiFi võrku, mis on omakorda ühendatud Internetiga. Olles Internetiga ühendunud, saadetakse veebiaadressile andurilt saadud temperatuuri ja suhtelise õhuniiskuse näit. Veebiaadress, mida kasutati, on osa ettevõtte SparkFun poolt pakutavast tasuta teenusest. Teenus on põhiliselt loodud värkvõrgu seadmete testimiseks ning vajadusel ka nende seadmete poolt saadetava info hoiustamiseks. Selleks oli vaja registreerida enda nimele aadress kuhu andmed saadetakse ning seejärel genereeritakse võtmed, mida info edastamiseks kasutatakse. Seadmete poolt saadatud infot näeb aadressilt, mis eelnevalt registreeriti. Testimaks süsteemi tervikuna kirjutati programmi ka sügava une režiimi kasutamine, milleks antud juhul oli 10 minutit. Test loeti sooritatuks kuna andmete saatmine teenusesse koos sügava une režiimiga töötas planeeritult.



The screenshot shows the SparkFun Data API interface for a device named 'temp\_test'. The interface includes a header with the device name and a description 'temp and humidity test'. There are buttons for 'Export to Analog.io' and 'Manage'. Below the header, there are tabs for different data formats: JSON, CSV, MySQL, PostgreSQL, and Atom. The 'TAGS' section shows 'room' and 'temperature'. A progress indicator shows '100% (50.00 of 50 MB) remaining'. The main content is a table with three columns: 'humidity', 'temp', and 'timestamp'. The table contains three rows of data.

humidity	temp	timestamp
35.53	24.86	2017-05-24T13:43:00.842Z
35.29	25.29	2017-05-24T13:32:24.465Z
36.11	26.04	2017-05-24T13:21:50.767Z

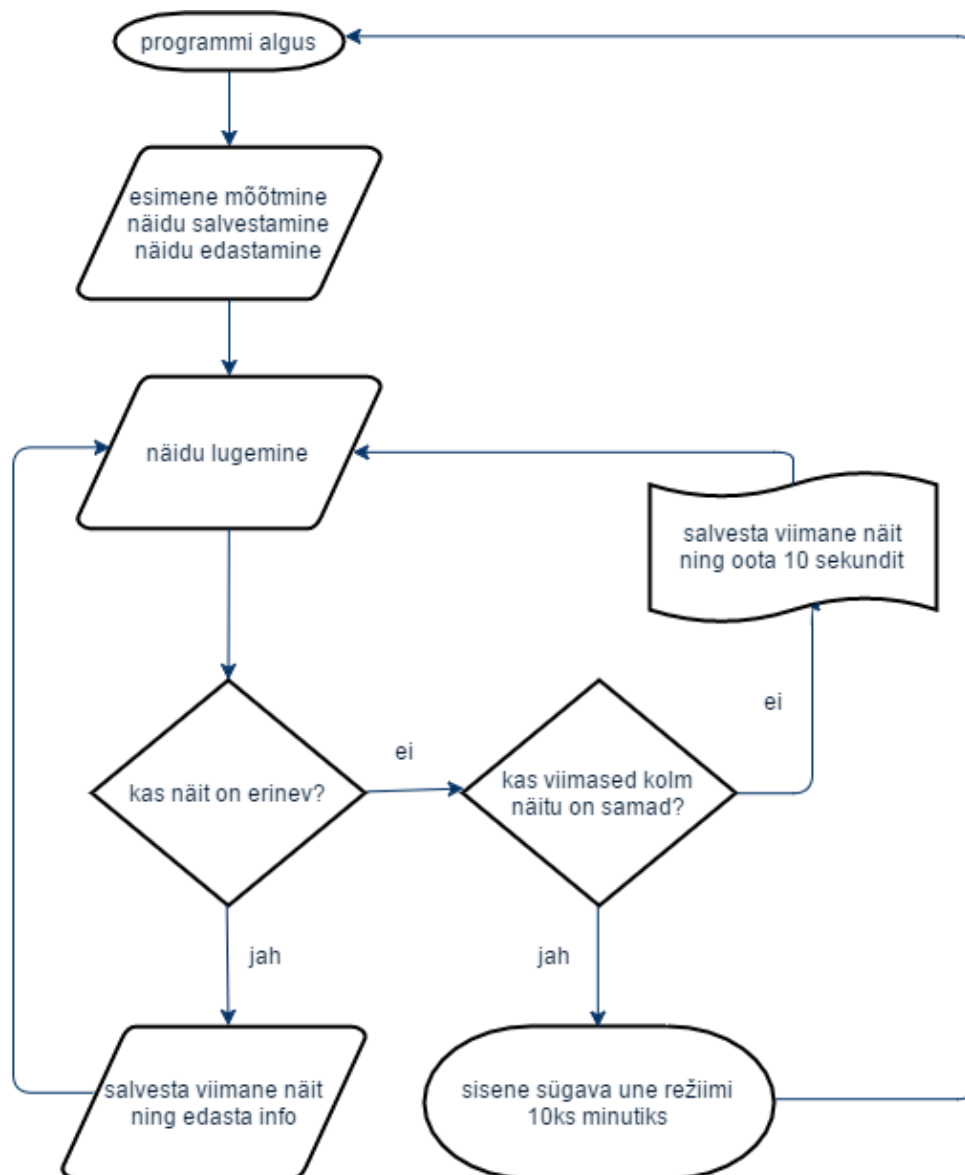
Sele 4.2 – Anduri näitude saatmine Interneti

## 4.2.5 Energiatarbe minimeerimise algoritm

Kuna reaalselt temperatuur ning suhteline õhuniiskus siseruumides ei muutu kogu aeg aktiivselt, siis otsustati, et samuti pole vajalik edastada kogu aeg andurilt saadud näite. Kui vaadata mikrokontrolleri aktiivse töörežiimi voolutarve vahemikku, mis oli määratud 80 – 260 mA, siis andmelehel [20] on täpsustavalt kirjeldatud millised protsessid kui palju voolu kasutavad:

- 160-260 mA kui edastatakse WiFi signaali tugevusega 13-21 dBm
- 120 mA kui edastatakse signaali tugevusega 0 dBm
- 80-90 mA kui ainult kuulatakse sissetulevaid WiFi signaale ning midagi ei edastata

Kasutades seda infot mõeldi välja süsteem, kus andurilt tulevaid näite loetakse lühikese aja tagant, kuid neid edastatakse ainult juhul, kui saadud näit on eelnevast näidust 0.5°C või 5% RH võrra erinev. Kirjeldatud süsteem tagab muutuse korral anduri valmisoleku info saatmiseks ning ei kuluta tulutult energiat sama informatsiooni edastamiseks. Sellise süsteemi eesmärgiks oli vähendada veel energiatarvet niivõrd palju kui võimalik. Selleks loodi algoritm (sele 4.2), mis võrdleb iga uut näitu eelmise näiduga. Kui näit on eelmisest näidust mingi ettemääratud arvu võrra erinev, siis see edastatakse ning salvestatakse järgmiseks kasutuseks. Kui näit ei ole erinev, siis näit lihtsalt salvestatakse. Pärast seda pannakse mikrokontroller ootele ja kogu protsessi korratakse 10 sekundi pärast uuesti. Süsteem pannakse ootele samuti energiatarbe mõistlikkuse seisukohast, kuna vaadates anduri Si7021 andmelehte [15] on seal välja toodud, et tüüpiline andurilt lugemise protsess kasutab 3.5 mA voolu, kuid hoides andurit ooterežiimis on voolutarve ainult 0.06  $\mu$ A. Ühe n-ö standard sessiooni ajal teostatakse neli andurilt näidu lugemist. Kui kolme järjestikuse lugemise korral ei ole kordagi edastatud üle WiFi andmeside infot, siis kogu süsteem siseneb sügava une režiimi 10ks minutiks ning pärast seda korratakse kogu protsessi uuesti. Algoritm koos lähtekoodiga SparkFun teenusesse info edastamisega on esitatud lisan 2. Lähtekood on koostatud SparkFun'i poolt väljastatud koodi baasil, kuhu eelnevalt oli sisse kirjutatud funktsionaalsus SparkFun'i teenusesse info saatmiseks.



Sele 4.3 – Energiatarbe algoritmi plokkiagramm

Testimiseks kasutatud lähtekoodide baasil kirjutati valmis töötav kood Arduino programmeerimiskeskonnas, mis sisaldab eelnevalt kirjeldatud funktsionaalsust. Koodi põhilisteks elementideks on andurilt näidu lugemine, selle edastamine üle WiFi andmeside ning väljatöötatud algoritm, mis vähendab kogu süsteemi energiatarvet. Valmis kirjutatud koodi (lisa 2) testiti ning test loeti sooritatuks.

## 5. ELEKTRONIKASKEEMI NING TRÜKKPLAADI DISAIN, TÖÖTSÜKLI ARVUTAMINE

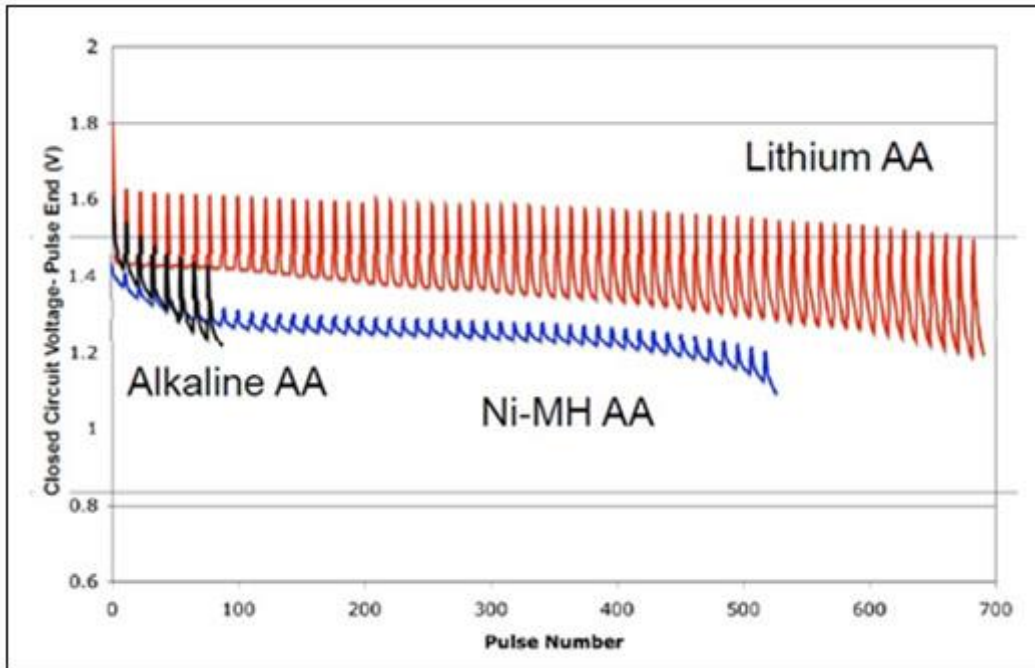
### 5.1 Komponentide koostamine

Anduriplatvormi koostamiseks tuli valmis teha elektroonikaskeem kasutades valitud komponente ja mooduleid. Süsteemi põhilised elemendid ning eesmärk jäid samaks, st. tuli koostada elektriskeem, kuidas WiFi moodul saaks anduriga suhelda ning selle näitu edastada. Elektroonikaskeem koostati arvutiprogrammiga DipTrace. Tegu on programmiga, mida kasutatakse elektroonikaskeemide ning trükkplaatide disainiks. Programmi teegid sisaldavad enimlevinud komponente ja kiipe, st. andureid, mikrokontrollereid jms. Kuna olemasolevate komponentide nimistus ei leidunud vajalikke kiipe ega mooduleid – WiFi moodul ning Si7021 andur – siis tuli need koostada ise.

Komponentide koostamiseks sisaldab DipTrace eraldi moodulit, kus on võimalik erinevate mallide järgi koostada oma komponent või siis see täiesti ise teha. WiFi mooduli eripärase kuju tõttu ei olnud võimalik malle kasutada (moodulil on viigud kolmes küljes tavapärase kahe või nelja asemel) ning kõik tuli manuaalselt teha. Selle tegemiseks tuli kasutada WiFi mooduli tootja poolt koostatud andmelehte [20], kus on näidatud vajalikud mõõdud komponentide disainiks. Samuti tuli valmis teha jalajälg (*footprint*), mis oleks piisavate mõõtudega, et komponendi viigud saaks sinna külge joota. Jalajälg peab olema koostatud väga täpselt vastavalt tootja poolt antud juhiste. Vastasel juhul võib tekkida olukord, kus trükkplaadile on jalajälg peale kantud, kuid see ei vasta mõõtudele ning komponenti on võimatu sinna peale joota. Sama oli vaja teha ka anduri Si7021 jaoks. Kuna anduri andmelehel [15] oli tootja kõik vajaliku info välja toonud, siis protseduur oli sama ning koostati jalajälg ja komponent.

## 5.2 Süsteemi pinge reguleerimine

Kuna anduriplatvorm disainiti töötama patareitoitel, siis oli vaja moodust, mis suudaks anduriplatvormi pinge hoida kogu aeg ühtlasena. Pinge on vajalik hoida ühtlasena, et tagada süsteemi korrektne töö. Sõltuvalt sellest, kuidas patareisid koormatakse, väheneb nende väljastatav pinge märgatavalt. Sele 5.1 näitab kolme erinevat tüüpi patareisid ning nende pingelangu (Y-teljel) koormates patareisid 1.3 W suuruste pulssidega mingi arv kordi (X-teljel).



Sele 5.1 – Kolme tüüpi patarei testimine pingelangule [25]

Töökindlas süsteemis peab olema pinge kontrollitud ning seetõttu on vaja kontrollsüsteemi, mis seda teeks. Selle jaoks kasutatakse patareitoitel töötavates lahendustes boost-muundurit (boost-converter). Tegu on alalisvoolu pingemuunduriga, mille väljundpinge on võrdne või kõrgem kui sellele rakendatav sisendpinge. Kasutades sellist lahendust, on võimalik garanteerida, et patareitoitel süsteem töötab alati vajalikus pingevahemikus. Antud töös kasutati Texas Instrumentsi poolt loodud boost-muunduri kiipi TPS61021A [26]. Selle olulisemad tehnilised parameetrid on:

- Sisendpinge vahemikus 0.5 kuni 4.4 V
- Väljundpinge vahemikus 1.8 kuni 4.0 V
- Väljundpinge 3.3 V juures on väljundvooluks 1.5 A ning seda süsteemi kasuteguriga 91% [26]

Kuna DipTrace programmis ei olnud ka seda komponentide nimekirjas, siis koostati pingemuunduri andmelehe abiga komponent ja jalajälg nagu eelneval kahel juhul.

## 5.2.1 Vooluallikas

Üldiselt kasutatakse süsteemides patareide puhul jadamisi ühendust. See on lihtne viis saada kõrgem pingeline ilma pingemuundureid kasutamata. Kuna projekteeritavas süsteemis on pingemuundur olemas ning selle miinimum sisendpinge on 0.5 V, siis pole vajadust lisada patareid jadamisi, kuna ühe AA patarei väljundpinge on 1.5 V. Lisades patareid omavahel rööbiti jääb süsteemi pingeline võrdseks ühe patarei pingega, kuid süsteemi toiteallika mahtuvus on kahe üksiku patarei mahtuvuse summa. See tähendab, et süsteem on võimeline kauem töötama patareitoitel, mis on selliste süsteemide puhul tähtis. Sellisel ühendusel tuleb silmas pidada, et mõlemad patareid peavad olema sama mahtuvuse ning pingega ja mitte eelnevalt kasutatud. Erinevat tüüpi patareide rööbiti ühenduse kasutamine võib kahjustada patareid ning on potentsiaalselt ohtlik. Selel 5.2 on näidatud rööbiti ühenduse patareipesa, mida ka projekteeritava anduriplatvormil kasutatakse.



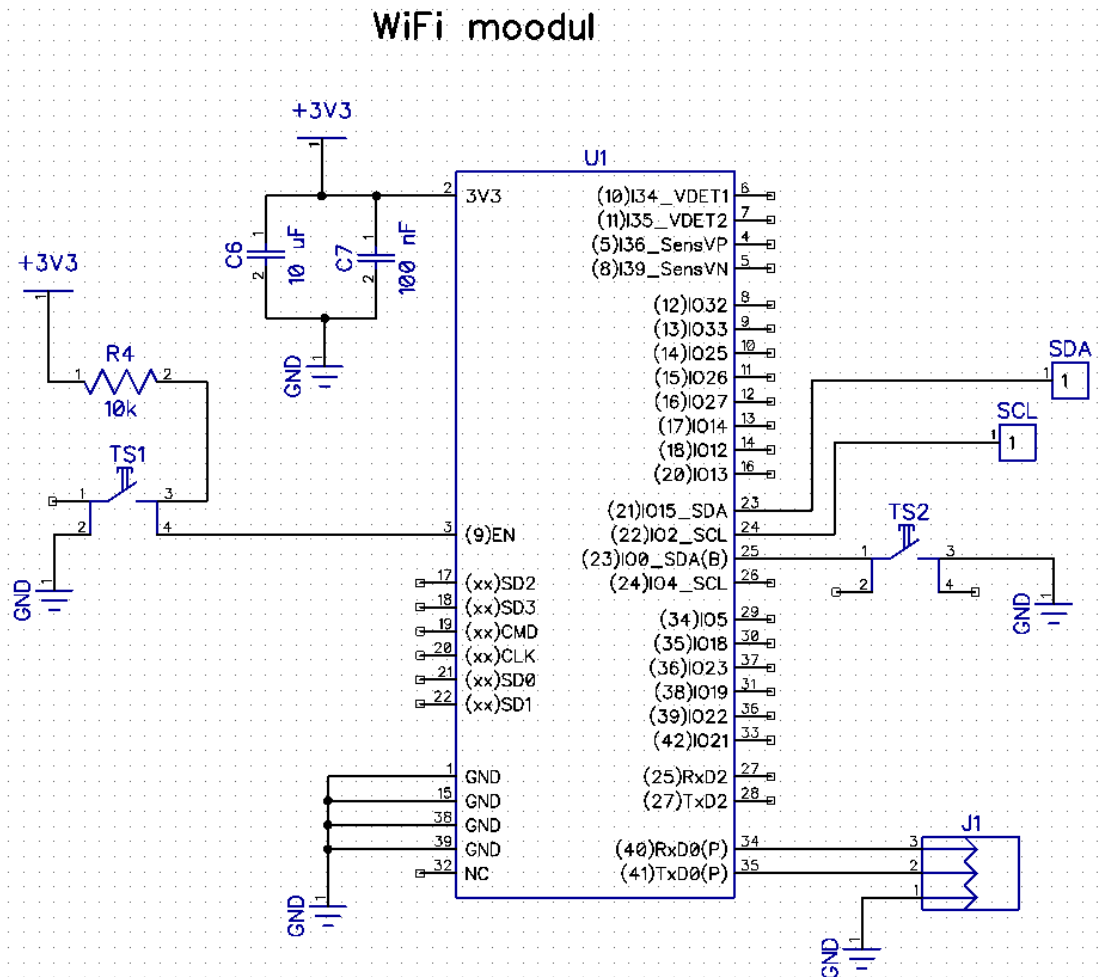
Sele 5.2 – Rööbiti ühendusega kahe AA patareipesa

## 5.3 Elektriskeemi koostamine

### 5.3.1 WiFi elektriskeem

Anduriplatvormil kasutatavate elementide töötamiseks oli samuti vaja lisada erinevaid elektroonikakomponente. WiFi mooduli toiteviikudele tuli lisada toidet siluvad kondensaatorid. Samuti tuli lisada kaks vajutatavat nuppu millest ühele tuli lisada ka takisti, et tavaolekus oleks sisend hoiatud kõrgel nivool. Sisend tuleb hoida kõrge seetõttu, et sisendi väärtus oleks kogu aeg määratud. Kuna antud juhul on tegu mikrokontrolleri EN sisendiga, siis kõrge väärtus tähendab, et kontrolleri töötab ning sisendi madalaks tõmmates lülitatakse moodul välja. Selle jaoks on ka lisatud nupp, mis töötab taaskäivituse nupuna. Teine nupp on lisatud sisendile IO0, kuid ilma kõrgeks määrava takistita, sest see on moodulis juba eelnevalt lisatud. Mõlemad nupud on vajalikud WiFi mooduli

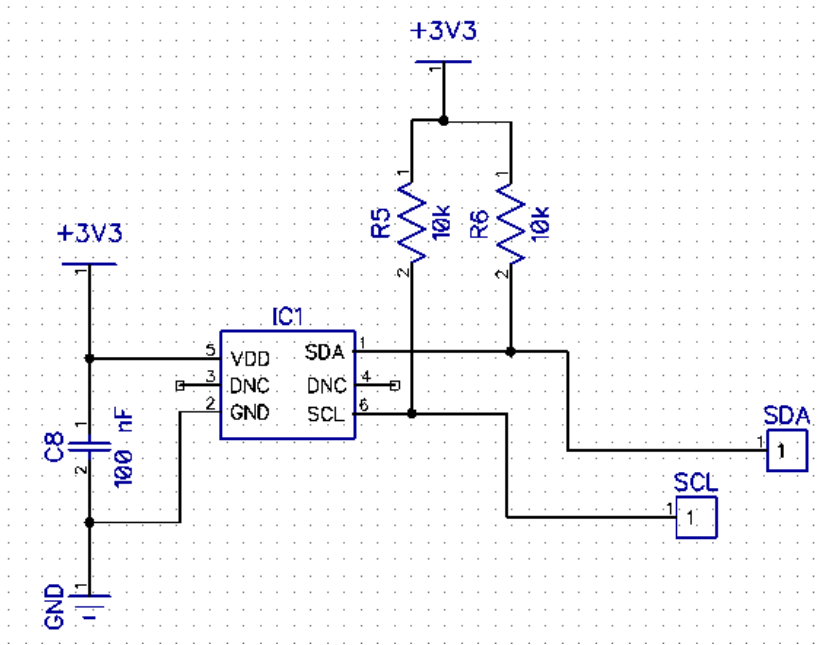
programmeerimiseks. Veel on lisatud kolme piikriba klemmi ühendus, mida kasutatakse mooduli programmeerimiseks. Kolmest klemmist kaks on ühendatud mikrokontrolleri Tx ja Rx viikudega, mille üle käib suhtlus programmeerimiseks. Antud lahendus koostati seetõttu, et nii majanduslikult kui ka funktsionaalsuselt oleks ebaotstarbekas lisada anduriplatvormile USB ühendus. USB ühenduse läbi programmeerimiseks tuleks lisada anduriplatvormile signaali konverter süsteem, mis oleks antud olukorras kulukas ning ebavajalik. Seega on süsteem disainitud nii, et seda programmeeritakse eraldi USB signaali konverter-seadmega. WiFi mooduli elektriskeem on näidatud seel 5.3.



Sele 5.3 – WiFi mooduli elektriskeem koos vajalike komponentidega

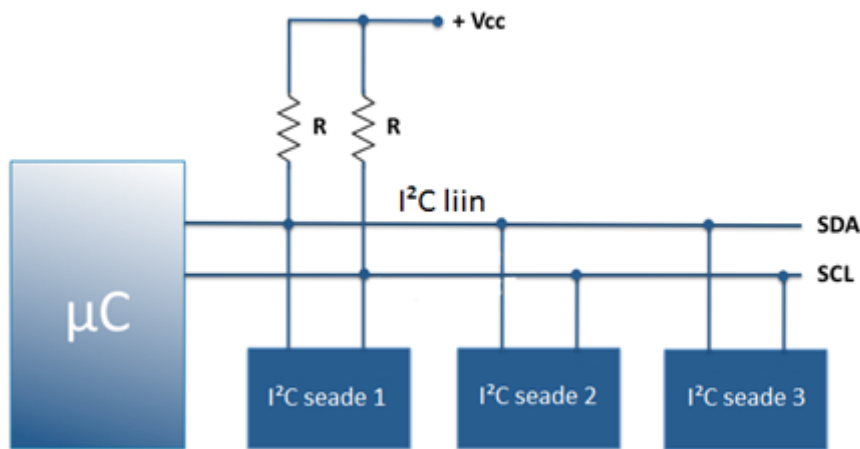
### 5.3.2 Anduri elektriskeem

Temperatuuri ja suhtelise õhuniiskuse andurile oli vaja lisada samuti lisaks komponente (sele 5.4). Toitesisendile lisati toidet siluva eesmärgiga kondenssaator ning kahele suhtlus viigule – SCL ja SDA – lisati signaali kõrgena hoidvad takistid.



Sele 5.4 – Si7021 anduri skeem koos vajalike komponentidega

Kogu anduriplatvorm on disainitud nii, et vajadusel on võimalik andur asendada mõne teise I<sup>2</sup>C anduriga või siis neid juurde lisata ilma süsteemi kogu disaini drastiliselt muutmata. See on võimalik seetõttu, et I<sup>2</sup>C tehnoloogia kasutab alati andmevahetuseks kahte liini ning nendele liinidele võib ühendada mitu erinevat seadet (sele 5.5).



Sele 5.5 – Mitu I<sup>2</sup>C seadet ühel liinil



### 5.3.3 Boost-muunduri skeem

**Väljundpinge** määramine pingemuunduri süsteemides vajab täpset komponentide valimist tootja poolt väljastatud juhendi [26] järgi. Kuna pingemuunduri väljund- ning sisendpinge vahemik on lai, siis väljundparameetrid saavutatakse erinevate komponentide täpse ning väljaarvutatud valikuga. Kuna väljundpinge määratakse välise takistist pingejaguriga, siis esmalt tuleb arvutada selle takisti väärtus valemiga 5.1.

$$R3 = \left( \frac{V_{OUT}}{V_{REF}} - 1 \right) * R2 \quad (5.1)$$

Kus:

- $V_{OUT}$  on reguleeritud väljundpinge
- $V_{REF}$  on sisemine referentspinge FB viigul

$R2$  väärtuse soovib tootja hoida väiksema kui  $400\text{k}\Omega$ , et saavutada parim täpsus ning selleks peab läbi takisti minev vool olema vähemalt 100 korda suurem kui FB viigu lekkevool ( $20\text{nA}$ ). Valides  $R2$  takisti väärtuseks  $100\text{k}\Omega$ , saadi  $R3$  väärtuseks:

$$R3 = \left( \frac{3.3}{0.795} - 1 \right) * 100000 = 315094\Omega$$

**Induktiivpool** on pingemuunduris üks tähtsamaid elemente. Selle elemendi valiku määravad kolm parameetrit: induktiivpooli induktiivsus, küllastusvool ja alalisvoolu takistus. Valitud pingemuundur on loodud töötama induktiivsusega vahemikus  $0.33\ \mu\text{H}$  ja  $1.0\ \mu\text{H}$ . Selle elemendi valikuks tuleb arvutada induktiivpooli maksimaalne hetkeline voolutarve antud skeemis. Suurima võimaliku voolu arvutamiseks kasutati miinimum sisendpinget, maksimum väljundpinget ning suurimat voolutarvet antud skeemile.

Boost-muunduris induktiivpooli alalisvoolu suurus arvutatakse valemiga 5.2.

$$I_{L(DC)} = \frac{V_{OUT} * I_{OUT}}{V_{IN} * \eta} \quad (5.2)$$

Kus:

$V_{OUT}$  on väljundpinge pingemuundurist

$I_{OUT}$  on väljundvool pingemuundurist

$V_{IN}$  on sisendpinge pingemuundurisse

$\eta$  on pingemuunduri kasutegur, tootja poolt soovitatud arvutuslik väärtus on enamasti 90%

Asendades muutujad vastavate väärtustega on vooluks:

$$I_{L(DC)} = \frac{3.6 * 0.5}{1.5 * 0.9} = 1.25A$$

Et leida induktiivpooli voolu kõikumist on esmalt vaja arvutada töötsükkel valemiga 5.3.

$$D = 1 - \frac{V_{IN} * \eta}{V_{OUT}} \quad (5.3)$$

Kus:

$V_{OUT}$  on väljundpinge pingemuundurist

$V_{IN}$  on sisendpinge pingemuundurisse

$\eta$  on pingemuunduri kasutegur, tootja poolt soovitatud arvutuslik väärtus on enamasti 90%

Töösükliks on:

$$D = 1 - \frac{1.5 * 0.9}{3.3} = 59\%$$

Induktiivpooli voolu kõikumiseks kasutatakse valemit 5.4.

$$\Delta I_{L(P-P)} = \frac{V_{IN} * D}{L * f_{SW}} \quad (5.4)$$

Kus:

$D$  on töötsükkel, mille arvutamiseks kasutati valemit 5.3

$L$  on induktiivpooli induktiivsus

$f_{SW}$  on lülituste sagedus

$V_{IN}$  on sisendpinge pingemuundurisse

Seega valemit 5.4 kasutades selgub, et voolu kõikumine on:

$$\Delta I_{L(P-P)} = \frac{3 * 0.59}{4.7 * 10^{-6} * 2 * 10^6} = 0.19A \quad (5.4)$$

Teades kõiki eelnevaid väärtusi on võimalik arvutada induktiivpoolil tekkiv maksimum vool valemiga 5.5.

$$I_{L(P)} = I_{L(DC)} + \frac{\Delta I_{L(P-P)}}{2} \quad (5.5)$$

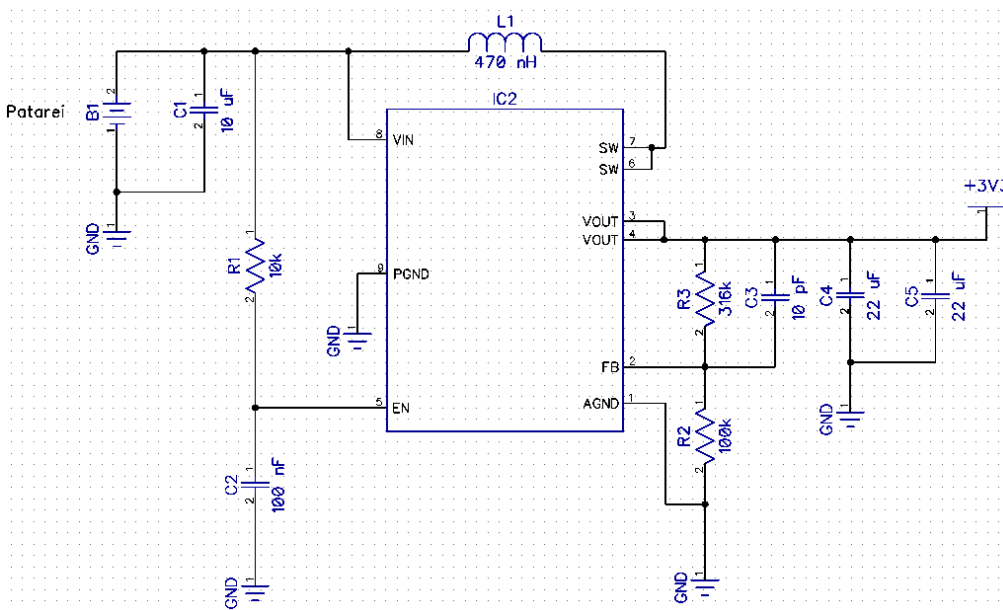
Mille tulemuseks on:

$$I_{L(P)} = 1.25 + \frac{0.19}{2} = 1.35A$$

Pingemuunduri andmelehel [26] soovitab tootja kasutada induktiivpooli, mille reaalne maksimum vool poolil on vähem kui 40% tootja poolt lubatud maksimum voolust komponendil. Samuti on andmelehel väljatoodud tootja poolt soovitatud induktiivpoolid, mida kasutada pingemuunduri skeemis. Nende põhjal osutus kõige sobivamaks tootja Toko induktiivpool DFE252012P-R47M. Komponenti induktiivsus on 0.47  $\mu$ H ning küllastusvoolu taluvus 6.6 A ning see vastab nõuetele.

**Kondensaatorite** eesmärk pingemuunduri väljundis on hoida vooluring stabiilsena. Siin valiti komponendid nii nagu tootja andmelehel oli soovitatud. Väljundisse ühendati kaks 22  $\mu$ F keraamilist X5R kondensaatorit. Et suurendada pingemuunduri tagasisideringi stabiilsust soovitab tootja lisada väljund viigu ning FB viigu vahele 10 pF keraamilise kondensaatori ning seda ka tehti. Pingemuunduri sisendisse ühendati 10  $\mu$ F keraamiline kondensaator toiteliini silumiseks. Kui tavaliselt kasutatakse lisaks elektrolüüt-kondensaatoreid suuremate pingekõikumiste vähendamiseks ning toiteliini silumiseks, siis antud juhul pole see vajalik, kuna vool tuleb patareide pealt ning need on piisavalt stabiilsed.

Pingemuunduri EN viigule tuli lisada ka komponendid mille eesmärk oli määrata selle viigu nivoo kõrgeks. Selleks kasutati 10 k $\Omega$  takistit ning 100 nF kondensaatorit. Pinge muunduri skeem on esitatud seel 5.6



Sele 5.6 – Pingemuunduri skeem koos töötamiseks vajalike komponentidega

## 5.4 Trükkplaadi disain

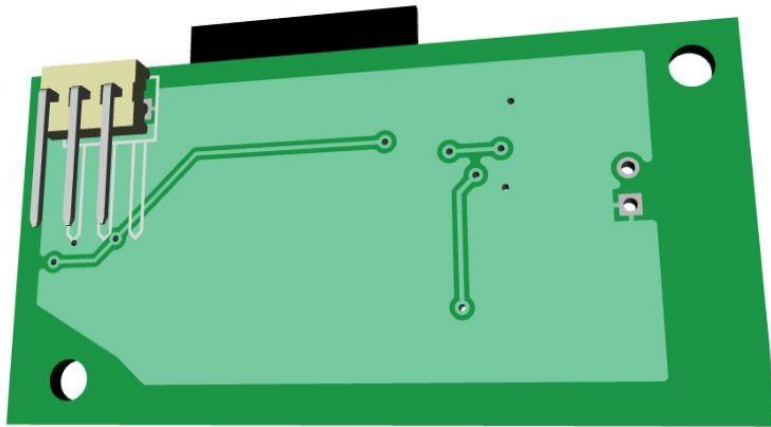
Kuna anduriplatvorm on disainitud töötama kahe AA patarei pealt, siis see määras ära kogu süsteemi miinimum mõõtmed. Kui esialgu oli ruumi kokkuhoidu silmas pidades plaan disainida trükkplaat nii, et komponendid asetsevad mõlemal poolel, siis lõpplahendusena otsustati, et komponendid asetatakse kõik ühele poole. Seda seetõttu, et elemente on trükkplaadil pingemuunduri tõttu palju ning paremaks lahenduseks otsustati pingemuunduri skeem hoida teistest komponentidest eemal. Trükkplaadi keskmisse osasse on paigutatud WiFi moodul ning plaadi paremasse nurka andur. Trükkplaadi disainil arvestati ka võimalike temperatuuri muutustega, mis võivad toimuda kõrgema voolu tõttu ning seetõttu on andur paigutatud toitesüsteemi elementidest eemale, et tagada võimalikult täpne näit jälgitava keskkonna kohta. WiFi mooduli antenni osa ulatub trükkplaadi enda mõõtmetest välja sihilikult. Selle eesmärgiks oli plaadi mõõtmete vähendamine ühes teljes ning paigutades WiFi moodul esitatud viisil, on võimalik vähendada signaali takistamist patareide poolt. Samuti aitab selline komponentide paigutus hoida trükkplaadi disaini universaalse, st. on võimalik lisada andureid või vahetada olemasolev andur.

Kõikide radade laius on välja arvatud vastavalt läbivale voolule kasutades vastavat kalkulaatorit Internetis [27], et ei tekiks olukorda, kus süsteemi mingi osa on aladimensioneeritud. Trükkplaadi kinnitamiseks korpuse külge on disainitud kahte nurka ka kinnitusavad. Kogu disaini juures on arvestatud Saksamaa trükkplaadi tootja Multi Leiterplatten GmbH disaini reegleid [28], et disainitud trükkplaati (sele 5.9) oleks võimalik valmistada tootja poolt. Kuigi Multi Leiterplatten GmbH lubab miinimum raja laiuseks 0,125 mm, siis trükkplaadil on miinimum raja laiuseks valitud 0,3 mm. See on valitud põhjusel, et väiksemad raja laiused tõstavad trükkplaadi hinda (vaja täpsemaid masinaid). Samuti on ka tootjaid, kes ei suuda sellise täpsusega trükkplaatide teha. Valides laiemad rajad plaadile on tõenäolisem, et erinevate tootjate juures ei teki trükkplaatide valmistamisega probleeme.

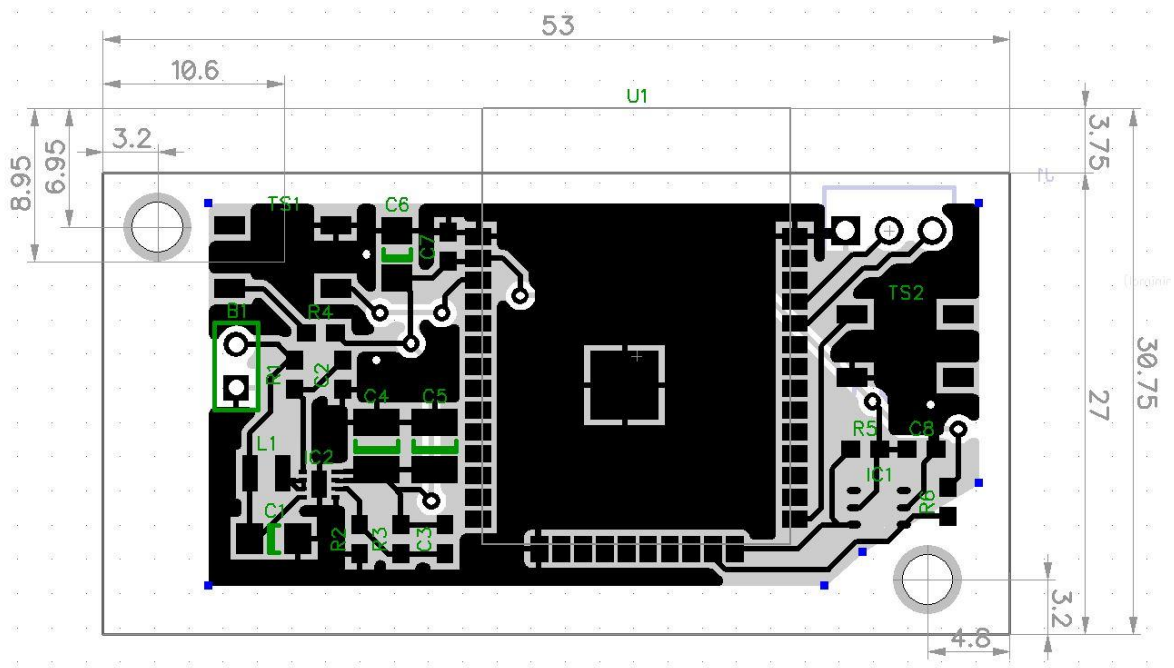
Selel 5.7 on näidatud ka trükkplaadi 3D mudel pealiselt poolt ning selel 5.8 alumiselt poolelt. ESP-WROOM-32 WiFi mooduli 3D mudel on võetud GrabCAD 3D mudelite keskkonnast [29]. Teiste trükkplaadil olevate komponentide mudelid on DipTrace programmist.



Sele 5.7 – Trükkplaadi 3D mudel pealt vaadatuna



Sele 5.8 – Trükkplaadi 3D mudel alt vaadatuna



Sele 5.9 – Trükkplaat koos mõõtudega millimeetrites

## 5.5 Töötükli arvutus

Töötükliina arvestati aega, kui kaua suudab anduriplatvorm töötada kahe rööbiti patarei toitel patareide täislaetud olekust kuni vajaliku laengu kaotamise piirini. Kuna reaalseid teste läbi ei viidud, siis on kõik arvutused tehtud andmelehtedelt saadud andmete alusel ning esitatud tulemused on teoreetilised.

Kuna temperatuuri ja suhtelise õhuniiskuse anduri erinevate operatsioonide voolutarve vahe on väike (4 mA töötükkel ning 3.8  $\mu$ A ooterežiim) [15] arvestades operatsioonidele kuluvat aega, siis lihtsuse mõttes arvestati maksimum väärtustega. Kuna ühe mõõtmise tegemiseks kuluv aeg on millisekundites, siis ümardati see samuti lihtsuse mõttes üheks sekundiks. Tegelikuses algoritm kasutab neid töötükkeid ning reaalne voolutarve on arvatust väiksem. WiFi moodulil on kolm voolutarbe väärtust, mis on arvutuste jaoks olulised. Nendeks on:

- Andmete saatmise protsess, maksimum 260 mA
- Andmete vastuvõtmine üle WiFi kommunikatsiooni ja protsessori kasutus, maksimum 90 mA
- Sügava une režiim, 0.15 mA

Üheks sessiooniks arvestati olukorda, kus andur tegi mõõtmise ning edastas selle näidu. Mõõtmist korraldi kolm korda ning iga mõõtmise vahel oodati 10 sekundit. Rohkem andmete edastamist ei toimunud ning süsteem sisenes sügava une režiimi. Kuna WiFi mooduli andmelehel protsessiaegade kohta midagi kirjas ei olnud [20], siis eelnevalt süsteemi testimise alusel võeti kogemuse najal käivitumise ning esimese näidu saatmise ajaks umbes 15 sekundit. Ühe patarei mahtuvuseks võeti 3000 mAh ehk rööbiti ühendades on süsteemi kogu mahtuvus 6000 mAh. Tabelis 5.1 on arvatud välja teoreetiline voolutarve ühe sessiooni kohta.

Tabel 5.1 – Erinevate protsesside voolutarbe kirjeldus

	Aeg (s)	Voolutarve (mA)	Voolutarve arvestades protsessi aega (mA)	Kordade arv	Voolutarve arvestades protsessi aega ning kordusi (mA)
Käivitus ning saatmine	15	260	1,083	1	1,083
Anduriga mõõtmine	1	4	0,001	3	0,003
Ootamine	10	90	0,250	2	0,500
Sügava une režiim	600	0,15	0,025	1	0,025
<b>Kokku</b>	<b>638</b>	-	-	-	<b>1,612</b>

Jagades aku mahtuvuse ühe sessiooni voolutarbega saadi tulemuseks:

$$\text{Sessioonide arv} = \frac{6000}{1,612} = 3723 \text{ kordust}$$

Kuna üks sessioon võtab aega 638 sekundit, siis arvestades see ümber tundidesse ning korrutades maksimaalse sessioonide arvuga, saadi tulemuseks ümardatult 660 tundi, mis on võrdne peaaegu 28 päevaga. See tähendab, et tehes iga 10 minuti tagant mõõtmisi ja andmete edastus ning ülejäänud aja hoides anduriplatvorm sügava une režiimis, on anduriplatvorm võimeline töötama 52 päeva kahe 3000 mAh mahtuvusega patarei toitel. Suurendades sügava une režiimi 20 minutile või 30 minutile töötaks anduriplatvorm vastavalt 58 päeva või peaaegu 76 päeva.

## 6. KORPUSE DISAIN NING OHUTUS

### 6.1 Disain ning korpuse elemendid

Anduriplatvormi põhilisteks disaini nõueteks oli esteetiline välimus ning otstarbekas ruumi kasutus. Minimaalsed mõõtmed määras korpusel ära toiteallikas ehk antud juhul kaks AA patareid. Seda ka seetõttu, et eelnevalt disainiti trükkplaat nii, et element, mis anduriplatvormil oleks kõige suuremate mõõtmetega, oleks toiteallikas. Anduriplatvormi korpuse materjali paksuseks on disainitud 3 mm ning korpuse välismõõdud on 45 x 70 x 40 mm. Korpus on disainitud nii, et kõiki selle elemente oleks võimalik teha 3D printimise meetodil ning materjaliks on valitud PLA plastik.

#### 6.1.1 Korpus

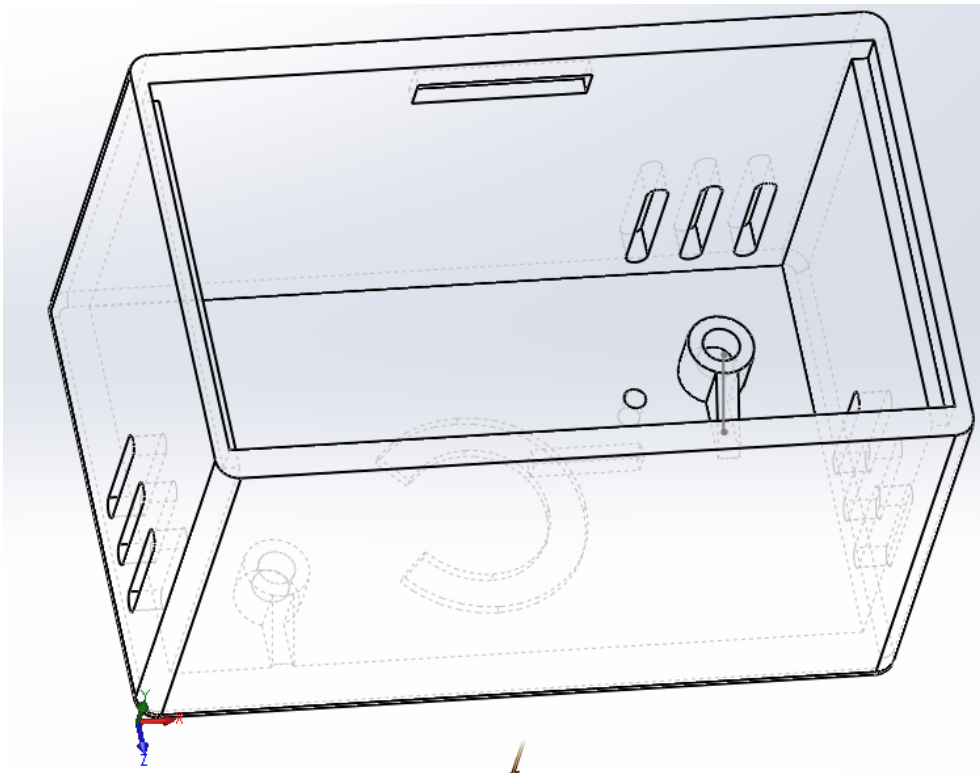
Anduriplatvormi esinduslik osa ehk korpuse välimise osa disainil tuli arvestada mitmete faktoritega. Külgedele ning pealmisele osale tehti õhutusavad, et süsteemi sisemine temperatuur sarnaneks võimalikult palju välise keskkonna temperatuuriga. Anduriplatvormi eest vaadatuna (sele 6.1) on vasakpoolses osas õhuavasid kaks – kõrval ja peal. Seda seetõttu, et kuna pingemuunduri skeem paigutati trükkplaadil eraldi ning potentsiaalselt võib sealt eralduda soojust, siis soojuste ära juhtimiseks on tehtud õhuavad.



Sele 6.1 – Anduriplatvorm eest vaadatuna



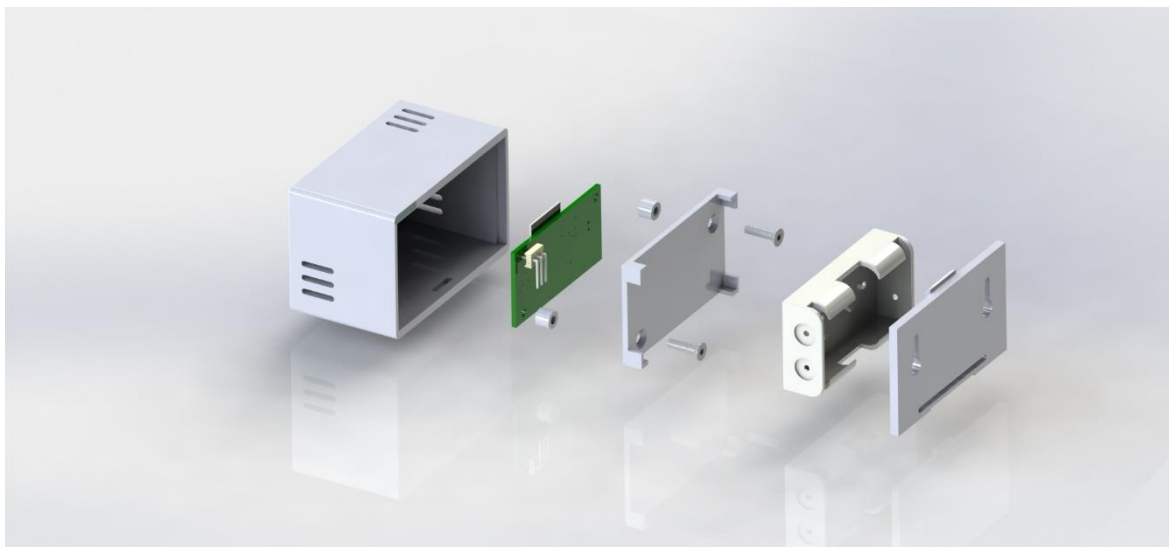
Samuti on tehtud ka ühed õhuavad paremale poole, kus asub andur, et korpusest väline õhutemperatuur oleks võrdne korpuses sees olevaga sellel kohal, kus toimuvad mõõtmised. Sisemisele osale (sele 6.2) on lisatud trükkplaadi kinnitamiseks kaks kruviaukudega kõrgemat silindrikujulist osa, mille eesmärgiks on tagada trükkplaadi mahtumine korpusesse nii, et selle elemendid ei oleks kontaktis korpuse esiküljega. Korpuse esimesele küljele (sele 6.1) on lisatud väikese läbimõõduga ava, mille eesmärgiks on tagada ligipääs trükkplaadil olevale RESET funktsionaalsusega nupule. Kuna nupp ei oleks nii hästi ligipääsetav korpuse tagumisest osast toiteallika tõttu, siis lähtuti kasutusmugavusest ning paigutati see esimesele küljele. Vältimaks igavat ega mitte midagi ütlevat disaini, lisati esiküljele ka täht C, mis paigutades RESET nupu ava lähedale jätab vaatajale mulje nagu esiküljel oleks kirjas temperatuurühik °C.



Sele 6.2 – Korpuse sisemine osa

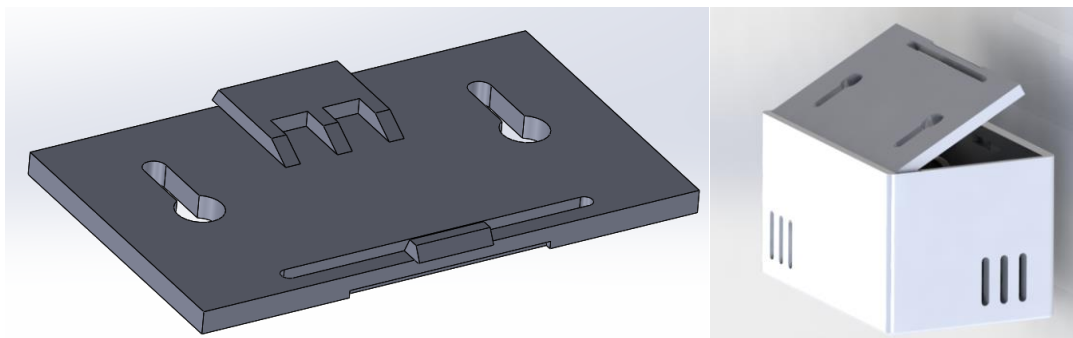
## 6.1.2 Korpuse kaas ning lisad

Lisaelementideks on korpuse sisse projekteeritud plastikust distantspuksid, mis eraldavad trükkplaati ning patareipesa hoidjat. Patareipesale on tehtud lihtne hoidja, mis koosneb ristkülikukujulisest plaadist ning millel on plaadist kõrgemad nurgad, kuhu vahele patareipesa fikseeritakse. Trükkplaat, distantspuksid ning patareipesa hoidja kinnituvad kahe M3x12 mõõtu poldiga korpuse külge. Patareipesa ei ole ise koostatud ning on võetud GrabCADi 3D mudelite keskkonnast [30]. Pesa mudel vastab olemasolevale tootele [31], mida antud töös ka kasutati. Samuti on poldi mudel võetud GrabCADi keskkonnast [32]. Kõik korpuse osad on näidatud seel 6.3.



Sele 6.3 – Vasakult järjestatuna: korpus, trükkplaat, distantspuksid, patareipesa hoidja, 2 M3x12 polti, patareipesa, korpuse kaas

**Korpuse kaas** (sele 6.4) on disainitud töötama plastiku elastsuse põhimõttel. Korpuse kahele siseküljele on tehtud pesad, kuhu kaas õrna jõuga vahele surutakse. Selle jaoks on kaane ühele küljele disainitud ava, mis laseb ühel korpuse avasse fikseeritaval osal liikuda plastiku elastsuse tõttu. Kaane korpusest välja ulatumise vältimiseks tehti korpuse sisse kaane jaoks ka eraldi rajad. Samuti lihtsustab see kaane kinnitamist. Seadme seina peale paigaldamiseks on tehtud ka kaks ava kruvide jaoks.



Sele 6.4 – Korpuse kaas vasakul ning paremal kaas koos korpusega

## 6.2 Seadme ohutus

Seadme korpus on disainitud nii, et kui isegi korpuse kaas avada, siis ilma patareipesa eemaldamata ning polte välja keeramata, ei pääse ligi elektroonikakomponentidele. See välistab elektrivooluga seotud ohud kui seadet kasutatakse ettenähtud viisil. Kogu süsteemi disainil valiti ohutuse eesmärgil kasutatavateks patareideks liitiumelement patareid, mis kasutavad liitiumit anoodina ning ei ole laetavad. Sellised liitiumpatareid on parimad asendamaks kõige tavalisemaid leeliselement patareid. Nende toitepinge on sama, kuid kaotavad laengut riulil hoiustades aeglasemalt ning eluiga on pikem seadmetes, mis kasutavad suuremat voolu. Samuti ei ole liitiumelement patareidel probleemi lekkimisega nagu seda on leeliselement patareidel. Arvestades eelnevalt mainitud ohutuse tagamiseks tehtud samme, siis saab öelda, et seadme ohutuse hinnang on hea.

## 7. MAJANDUSANALÜÜS, SEADME HINNANG NING IDEED SÜSTEEMI TÄIENDAMISEKS

### 7.1 Majandusanalüüs

Majandusanalüüs koostati kahes osas. Esmalt tuuakse välja kõikide komponentide hind ning kogused (tabel 7.1). Seejärel kirjeldatakse elementide valmistamisega seotud kulusid ning lõpuks lisatakse anduriplatvormi koostamise kulu. Komponentide hinnad on arvestatud selle järgi, kui palju oleks kulu ühe seadme valmistamisel ning ei arvestata massproduksioonist tulenevaid soodustusi.

Tabel 7.1 – Kasutatud komponentide hinnakiri

Komponent	Tootenimi	Tükkide arv (tk)	Hind (EUR)
WiFi moodul	ESP32-WROOM-32	1	4,85 [11]
Andur	Si7021	1	3,17 [33]
Pingemuundur	TPS61021A	1	1,44 [34]
Kondensaator 100 nF	GRM033R60J104KE19D	3	0,11 [35]
Kondensaator 10 uF	GRT31CR60J106KE01L	2	1,04 [36]
Kondensaator 22 uF	GRT32ER60J226KE01L	2	1,66 [37]
Kondensaator 10 pF	GRM1885C1H100JA01J	1	0,1 [38]
Induktiivpool 470 nH	DFE252012P-R47M	1	0,49 [39]
Nupuga lüliti	SKQGAKE010	2	0,3 [40]
Takisti 316 kΩ	CRCW0603316KFKEA	1	0,1 [41]
Takisti 100 kΩ	CRCW0603100KFKEA	1	0,1 [42]
Takisti 10 kΩ	CRCW060310K0FKEA	4	0,4 [43]
Piikriba klemmid	Molex 22-28-8040	1	0,3 [44]
Kinnituspolt	M3x12	2	0,24 [45]
Patareipesa	BC2AA	1	1 [31]
<b>Kokku</b>			15,3

Nagu ka trükkplaadi koostamise peatükis mainiti, siis trükkplaat on disainitud sedasi, et kulud oleksid võimalikult madalad. Kuna trükkplaat on disainitud vastama tootja Multi Leiterplatten GmbH disaini reeglitele [28], siis trükkplaadi valmistamise hinnapakkumine on samuti sealtpärast. Kui tellida vaid üks trükkplaat, siis on selle hinnaks 39,7€ ning sinna lisandub veel saatmiskulu. Tellides 10 tükki, kujuneb ühe plaadi hinnaks 4.04€ pluss saatmiskulu. Nagu näha, on masstootmisel alati hinna suhtes eelised, kuna suurem osa hinnast moodustab seadmete ettevalmistamine ning see tuleb kinni maksta tootjale ka ühe trükkplaadi valmistamise pealt. Korpuse detailide 3D printimise hinnaks FDM meetodil ning kasutades PLA plastikut, tuleb 20€ kuni 40€. Seda sõltuvalt teenusepakkujast. Kasutades 3D printimise tarkvara Simplify3D arvutati välja materjali kulu, mis jääb korpuse plastikust elementide printimisel kokku alla 2€. Sõltuvalt 3D printerist tuleb siia juurde ka masina kasutamise

kulu, mis jääb alla 1€ antud korpuse printimisel (enamasti koosneb vaid elektri hinnast, sest 3D printeritel füüsiline kulumine on väga väike). Seega on näha mõlema teenuse puhul, nii trükkplaadi kui ka korpuse valmistamisel, et suurem osa hinnast moodustab teenus ise mitte materjali kulu. Viimaseks lisandub tööjõu hind, et anduriplatvorm füüsiliselt valmis teha. See tähendab komponentide trükkplaadile jootmist, mikrokontrollerile tarkvara laadimist ning kõikide elementide korpusesse fikseerimist.

Arvestades kokku komponentide ning tootmisprotsesside hind kasutades ainult ettevõtete poolt pakutavaid teenuseid, tuleb ühe anduriplatvormi hind ilma tööjõu hinda arvestamata umbes 90€. Võrreldes antud töös koostatud seadme hinda näiteks turul oleva ConnectSense seadme hinnaga [5], siis on näha, et isegi ainult ühe anduriplatvormi koostamisel on hind odavam kui massproduksioonis oleva toote hind.

## 7.2 Hinnang seadmele

Töös kirjeldatud anduriplatvorm on disainitud põhiliselt majanduslikust aspektist, kuna lõppkasutajale pole määravaks millest seade koosneb, vaid kui palju see maksab ja kui lihtne seda kasutada on. Samuti pole seadme kvaliteedi kohapealt hinnas järeleandmisi tehtud ning süsteem on disainitud olema kvaliteetne võimalikult madala hinna juures. Seadme omahinda oleks võimalik langetada asendades mõned passiivsed elektroonikakomponendid, kuid nende kasutamine disainitud süsteemis vajaks esialgu ka füüsilist testimist.

## 7.3 Ideed süsteemi täiendamiseks

Üks põhilisi eesmärke antud töös oli minimeerida seadme energiatarvet ning seda oleks võimalik tegelikult veel madalamaks tuua. Kuna esialgu planeeritud modemi une töörežiim WiFi moodulil ei töötanud, siis on võimalik kasutada mikrokontrolleris ära madala energiatarbe protsessorit (ULP). Hetkel on tarkvara koostatud selliselt, et tehakse mõõtmised ning seejärel sisenetakse sügava une režiimi ehk suurem energia säästmise käib ainult selles töörežiimis. Kuna ULP protsessor on ainuke, mis töötab sügava une režiimis, siis oleks võimalik seda programmeerida nii, et andur teostab kogu aeg mõõtmisi ning võrdlusi, kuid WiFi moodul äratatakse üles ainult siis, kui on toimunud reaalne muudatus ning info edastatakse. Käesolevas töös seda ei tehtud, sest selle programmeerimine oleks palju keerulisem ja aeganõudvam.

## KOKKUVÕTE

Bakalaureusetöö teema pakuti välja Tallinna Tehnikaülikooli poolt. Ülikooli poolt oli töö juhendajaks Tallinna Tehnikaülikooli Mehhanotehnika eriala doktorant Ervin Piigli.

Käesoleva bakalaureusetöö teemaks oli valmistada patareitoitel anduriplatvorm, mis edastab infot üle WiFi andmeside. Eesmärgiks oli projekteerida anduriplatvormi võimalikult universaalne prototüüp, mille disain võimaldaks erinevaid andureid kasutada, kui trükkplaati natukene muuta. Töös koostatud anduriplatvorm kasutab temperatuuri ning suhtelise õhuniiskuse andurit. Töö esimeses pooles võrreldakse turul olemasolevaid temperatuuri ning õhuniiskust mõõtvaid WiFi seadmeid ning seda omaduste ja tehnoloogia poolest. Samuti kirjeldatakse lühidalt juhtmevabasid andmeedastus tehnoloogiaid ja nende plusse ning miinuseid. Järgmises faasis asuti valima põhilisi komponente millest süsteem koosneb. WiFi andmeside lisati süsteemile mooduli kujul, mis sisaldab endas ka mikrokontrollerit kogu süsteemi juhtimiseks. Valikul oli tähtsamateks argumentideks hind ning selle kasutamise lihtsus. Samadest omadustest lähtuti ka süsteemi teise peamise komponendi, temperatuuri ja suhtelise õhuniiskuse anduri valikul, kus valiku määras hind ning lihtne kasutus olemasolevate programmeerimiskeelte teekide tõttu.

Süsteemi põhimõtete testimiseks kasutati nende komponentide baasil olevaid arendusmooduleid firmalt SparkFun. Kuigi esialgu oli energiatarbe vähendamiseks plaanis kasutada kindlat töörežiimi mikrokontrolleril, siis alles testimise käigus selgus, et reaalselt pole seda funktsionaalsust veel mikrokontrollerisse sisse kirjutatud. Selle asendusena kasutati teisi töörežiime ning kirjutati valmis eraldi algoritm, mille eesmärk oli energiatarve vähendamine rakendades ainult hädavajalikke süsteemi protsesse teatud ajahetkedel.

Edasi koostati kogu süsteemi komponentidest elektriskeem ning seejärel selle baasil ka trükkplaat. Trükkplaadi disainil oli tähtis silmas pidada anduriplatvormi universaalsust ning samas hoida plaadi pindala väike, et omakorda hoida selle tootmiskulud madalamad. Kogu elektroonikale disainiti ümber korpus, mis on samal ajal nii esteetilise välimuse kui ka efektiivse disainiga.

Töö viimases osas tehti majandusanalüüs ning näidati, kuidas seadme omahind kujuneb. Samuti kirjeldati milline on hinnavahe teatud protsessidel masstootmise ja üksiku seadme tegemisel. Lühidalt kirjutati ka ohutusest ja ideedest süsteemi täiendamiseks ning anti hinnang loodud seadmele.

Tehtud töö võib lugeda eesmärkide põhjal sooritatuks, kuna suudeti projekteerida anduriplatvormi esmane prototüüp, mis hiljem ka valmistatakse. Anduriplatvormi disain võimaldab lihtsa vaevaga trükkplaati muutes asendada olemasolev andur või lisada seadmeid juurde. Samuti on projekteeritud prototüüp hinnalt odavam kui olemasolevad WiFi andmeside kasutavad seadmed. Tootes projekteeritud anduriplatvormi massiliselt on võimalik hinda veel langetada. Samuti täideti osalt ka energiatarbe minimeerimise eesmärk ning töös antud soovitude najal oleks energiatarvet võimalik veel enamgi vähendada.

## SUMMARY

The topic of the Bachelor's thesis was proposed by Tallinn University of Technology. Thesis was tutored by Ervin Piigli, who is a Ph.D student in Department of Mechanical and Industrial Engineering in Tallinn University of Technology.

The purpose of this thesis was to create a battery powered sensor platform prototype, which broadcasts data over WiFi communication. The goal was to engineer a universal sensor platform with a design which would allow to install different devices or sensors to this platform with little effort. Sensor platform engineered in this thesis uses sensor to measure temperature and relative humidity. At the first stage of the thesis, author compares similar measuring devices which already exist on the market. Comparison is based on their technology used and technical specification. There is also a superficial description of some existing wireless communication technologies and their advantages and disadvantages. In the next phase the key components of the system were chosen. WiFi communication was added as a module, which includes also a microcontroller for driving the whole platform. The main arguments for picking the WiFi module were the low price and support from existing users, since the module is widely used. The same approach was used for picking the second key component, temperature and relative humidity sensor, as it was chosen because of its low price and already existing libraries in different programming languages.

For testing the principles of the system, author used developer modules from SparkFun which were based on the two key components. To minimize the whole platform's energy consumption there was a plan to use a certain low energy working state, but while testing the platform's functionality it turned out that this low energy state functionality wasn't yet written in the microcontroller. Instead of that, other working states were used combined with a specially written algorithm. Its main goal was to minimize energy consumption by using only necessary parts of the system at any given time.

After that a circuit diagram was made and based on that a layout for a printed circuit board (PCB) was created. While creating the PCB layout it was important to keep the overall system design universal and area of the board as small as possible, since the size of the board is in direct correlation with the board's price. Also an enclosure was designed for the electronics while keeping in mind the esthetical aspect and effective design.

In the last part of the thesis there was an economic analysis made to show what kind of expenses has to be made to create this sensor platform and what does these expenses consist of. Author also writes superficially about the safety of this device, proposes new ideas to advance the efficiency of the system and gives an overall assessment of this thesis.

Overall, this thesis can be considered a success. Required prototype of the sensor platform was engineered while keeping in mind its universality and which will be built afterwards. The goal to minimize energy consumption was mostly successful and with the use of ideas given by the author the efficiency level of the system can be raised even more.

## KASUTATUD KIRJANDUS

- [1] K. Ashton, „That 'Internet of Things' Thing,“ 22 juuni 2009. [Võrgumaterjal]. Available: <http://www.rfidjournal.com/articles/view?4986>. [Kasutatud 20 mai 2017].
- [2] The Carnegie Mellon University, „Coke Machine,“ [Võrgumaterjal]. Available: [https://www.cs.cmu.edu/~coke/history\\_long.txt](https://www.cs.cmu.edu/~coke/history_long.txt). [Kasutatud 19 mai 2017].
- [3] Gartner, „Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015,“ 10 november 2015. [Võrgumaterjal]. Available: <http://www.gartner.com/newsroom/id/3165317>. [Kasutatud 19 mai 2017].
- [4] NIS Store, „Xiaomi Mi Smart Home Temperature / Humidity Sensor,“ 2017. [Võrgumaterjal]. Available: <https://nis-store.com/sockets-and-sensors/xiaomi-mi-temperature-humidity-sensor/>. [Kasutatud 10 aprill 2017].
- [5] Amazon, „ConnectSense CS-TH Wireless Temperature and Humidity Sensor,“ 2017. [Võrgumaterjal]. Available: [https://www.amazon.com/dp/B00ISKS8HG/ref=cm\\_sw\\_su\\_dp](https://www.amazon.com/dp/B00ISKS8HG/ref=cm_sw_su_dp). [Kasutatud 10 aprill 2017].
- [6] SAMSUNG , „SmartThings Moisture Sensor,“ 2017. [Võrgumaterjal]. Available: <http://www.samsung.com/uk/smartthings/sensors-plug-f-wtr-uk-v2/>. [Kasutatud 10 aprill 2017].
- [7] B. Ray, „The ZigBee Vs WiFi Battle For M2M Communication,“ 2 november 2015. [Võrgumaterjal]. Available: <https://www.link-labs.com/blog/zigbee-vs-wifi-802-11ah>. [Kasutatud 11 aprill 2017].
- [8] EngineersGarage, „Zigbee v/s Wi-Fi,“ [Võrgumaterjal]. Available: <https://www.engineersgarage.com/contribution/zigbee-vs-wi-fi>. [Kasutatud 11 aprill 2017].
- [9] J. Kastrenakes, „There's a new type of Wi-Fi, and it's designed to connect your smart home,“ 4 jaanuar 2016. [Võrgumaterjal]. Available: <https://www.theverge.com/2016/1/4/10691400/new-wifi-halow-standard-announced-iot-ces-2016>. [Kasutatud 11 aprill 2017].
- [10] Espressif, „ESP-WROOM-32 Overview,“ [Võrgumaterjal]. Available: <https://espressif.com/en/products/hardware/esp-wroom-32/overview>. [Kasutatud 21 märts 2017].



- [11] Watterott, „Espressif ESP32 ESP-WROOM-32 WiFi/WLAN+Bluetooth Module,“ [Võrgumaterjal]. Available: <http://www.watterott.com/en/ESP-WROOM-32-Module>. [Kasutatud 21 märts 2017].
- [12] Adafruit, „DHT11 basic temperature-humidity sensor + extras,“ [Võrgumaterjal]. Available: <https://www.adafruit.com/product/386>. [Kasutatud 21 märts 2017].
- [13] Adafruit, „DHT22 temperature-humidity sensor + extras,“ [Võrgumaterjal]. Available: <https://www.adafruit.com/product/385>. [Kasutatud 21 märts 2017].
- [14] Aosong (Guangzhou) Electronics Co., Ltd, „Digital-output relative humidity & temperature sensor/module AM2303,“ [Võrgumaterjal]. Available: <https://cdn-shop.adafruit.com/datasheets/DHT22.pdf>. [Kasutatud 21 märts 2017].
- [15] Silicon Labs, „Si7021-A20 datasheet,“ august 2016. [Võrgumaterjal]. Available: <http://www.mouser.com/ds/2/368/Si7021-A20-707879.pdf>. [Kasutatud 2 aprill 2017].
- [16] SparkFun, „ESP32 Thing hookup guide,“ [Võrgumaterjal]. Available: <https://learn.sparkfun.com/tutorials/esp32-thing-hookup-guide>. [Kasutatud 20 märts 2017].
- [17] SparkFun, „Si7021 Humidity and Temperature Sensor Hookup Guide,“ [Võrgumaterjal]. Available: <https://learn.sparkfun.com/tutorials/si7021-humidity-and-temperature-sensor-hookup-guide>. [Kasutatud 6 aprill 2017].
- [18] Espressif, „ESP-IDF Programming Guide,“ 2016. [Võrgumaterjal]. Available: <https://esp-idf.readthedocs.io/en/latest/index.html>. [Kasutatud 18 aprill 2017].
- [19] „Arduino core for the ESP32,“ [Võrgumaterjal]. Available: <https://github.com/espressif/arduino-esp32>. [Kasutatud 18 aprill 2017].
- [20] Espressif, „ESP-WROOM-32 Datasheet,“ aprill 2017. [Võrgumaterjal]. Available: [https://espressif.com/sites/default/files/documentation/esp\\_wroom\\_32\\_datasheet\\_en.pdf](https://espressif.com/sites/default/files/documentation/esp_wroom_32_datasheet_en.pdf). [Kasutatud 26 märts 2017].
- [21] Espressif, „WPA2 Enterprise Example,“ [Võrgumaterjal]. Available: [https://github.com/espressif/esp-idf/tree/master/examples/wifi/wpa2\\_enterprise](https://github.com/espressif/esp-idf/tree/master/examples/wifi/wpa2_enterprise). [Kasutatud 16 aprill 2017].
- [22] N. Kolban, 25 jaanuar 2017. [Võrgumaterjal]. Available: <https://github.com/nkolban/esp32-snippets/blob/master/i2c/scanner/i2cscanner.c>. [Kasutatud 17 aprill 2017].
- [23] Espressif, „Wifi Power Save Example,“ [Võrgumaterjal]. Available: [https://github.com/espressif/esp-idf/blob/master/examples/wifi/power\\_save/main/power\\_save.c](https://github.com/espressif/esp-idf/blob/master/examples/wifi/power_save/main/power_save.c). [Kasutatud 18 aprill 2017].

- [24] Espressif, „Modem sleep,“ 21 detsember 2016. [Võrgumaterjal]. Available: <https://www.esp32.com/viewtopic.php?t=773#p3272>. [Kasutatud 17 aprill 2017].
- [25] Battery University, „Advantages of Primary Batteries,“ [Võrgumaterjal]. Available: [http://batteryuniversity.com/learn/article/primary\\_batteries](http://batteryuniversity.com/learn/article/primary_batteries). [Kasutatud 8 mai 2017].
- [26] Texas Instruments, „TPS61021A 3-A Boost Converter with 0.5-V Ultra Low Input Voltage,“ juuni 2016. [Võrgumaterjal]. Available: <http://www.ti.com/lit/ds/symlink/tps61021a.pdf>. [Kasutatud 24 aprill 2017].
- [27] N. d. Smith, „ANSI IPC-2221A PCB Trace Width Calculator,“ 13 juuli 2010. [Võrgumaterjal]. Available: <http://www.desmith.net/NMdS/Electronics/TraceWidth.html>. [Kasutatud 10 mai 2017].
- [28] Multi Leiterplatten GmbH, „Calculator,“ [Võrgumaterjal]. Available: <https://portal.multi-circuit-boards.eu/Customer/>. [Kasutatud 10 mai 2017].
- [29] M. H. YANIK, „ESP-WROOM-32 3D Step and Iges Models,“ 22 jaanuar 2017. [Võrgumaterjal]. Available: <https://grabcad.com/library/esp-wroom-32-3d-step-and-iges-models-1>. [Kasutatud 21 mai 2017].
- [30] J. S. M. Martínez, „AA Battery Holder,“ 30 august 2013. [Võrgumaterjal]. Available: <https://grabcad.com/library/aa-battery-holder-step-file-1>. [Kasutatud 15 mai 2017].
- [31] Fasttech, „2\*AA Parallel Batteries Holder Case Box with Leads,“ [Võrgumaterjal]. Available: <https://www.fasttech.com/product/1289504-2aa-parallel-batteries-holder-case-box-with-leads>. [Kasutatud 12 mai 2017].
- [32] A. Alexandrov, „Винт DIN 7991,“ 20 juuli 2015. [Võrgumaterjal]. Available: <https://grabcad.com/library/din-7991-1>. [Kasutatud 19 mai 2017].
- [33] Farnell, „SILICON LABS SI7021-A20-GM Temperature Sensor IC,“ [Võrgumaterjal]. Available: <http://ee.farnell.com/silicon-labs/si7021-a20-gm/temp-humidity-sensor-0-4deg-c/dp/2473670>. [Kasutatud 18 mai 2017].
- [34] Mouser, „Texas Instruments TPS61021ADSGR,“ [Võrgumaterjal]. Available: <http://www.mouser.ee/ProductDetail/Texas-Instruments/TPS61021ADSGR/?qs=%2fha2pyFaduhv%2f6PmvIssVT6xdzoapPKAXWxu5VksxM7p6QEutuf2nQ%3d%3d>. [Kasutatud 15 mai 2017].
- [35] Farnell, „MURATA GRM033R60J104KE19D,“ [Võrgumaterjal]. Available: <http://ee.farnell.com/murata/grm033r60j104ke19d/cap-mlcc-x5r-100nf-6-3v-0201/dp/1775994>. [Kasutatud 15 mai 2017].

- [36] Farnell, „MURATA GRT31CR60J106KE01L SMD Multilayer Ceramic Capacitor,“ [Võrgumaterjal]. Available: <http://ee.farnell.com/murata/grt31cr60j106ke01l/cap-mlcc-auto-x5r-10uf-6-3v-1206/dp/2672207>. [Kasutatud 15 mai 2017].
- [37] Farnell, „MURATA GRT32ER60J226KE01L SMD Multilayer Ceramic Capacitor,“ [Võrgumaterjal]. Available: <http://ee.farnell.com/murata/grt32er60j226ke01l/cap-mlcc-auto-x5r-22uf-6-3v-1210/dp/2672223>. [Kasutatud 15 mai 2017].
- [38] Mouser, „Murata Electronics GRM1885C1H100JA01J,“ [Võrgumaterjal]. Available: <http://www.mouser.ee/ProductDetail/Murata-Electronics/GRM1885C1H100JA01J/?qs=sGAEpiMZZMs0AnBnWHyRQApGL1VC6wChM2dyMcKCEjc%3d>. [Kasutatud 15 mai 2017].
- [39] Mouser, „Murata Electronics DFE252012P-R47M=P2,“ [Võrgumaterjal]. Available: <http://www.mouser.ee/ProductDetail/Murata-TOKO/DFE252012P-R47M=P2/?qs=KuGPmAKtFKXQwFQiwRwig%3D%3D>. [Kasutatud 15 mai 2017].
- [40] Mouser, „ALPS SKQGAKE010,“ [Võrgumaterjal]. Available: <http://www.mouser.ee/ProductDetail/ALPS/SKQGAKE010/?qs=sGAEpiMZZMmsgGjVA3toVBA1UHfTHbV9%252bZVm6oBay0Ts%3d>. [Kasutatud 15 mai 2017].
- [41] Mouser, „Vishay CRCW0603316KFKEA resistor,“ [Võrgumaterjal]. Available: <http://www.mouser.ee/ProductDetail/Vishay/CRCW0603316KFKEA/?qs=sGAEpiMZZMvdGkrng054tz1PN3csf7M2Rq0mfCPYsMg%3d>. [Kasutatud 15 mai 2017].
- [42] Mouser, „Vishay CRCW0603100KFKEA resistor,“ [Võrgumaterjal]. Available: <http://www.mouser.ee/ProductDetail/Vishay/CRCW0603100KFKEA/?qs=sGAEpiMZZMvdGkrng054txtDEX%2fRoWGKBsoPeP3%2fIro%3d>. [Kasutatud 15 mai 2017].
- [43] Mouser, „Vishay CRCW060310K0FKEA resistor,“ [Võrgumaterjal]. Available: <http://www.mouser.ee/ProductDetail/Vishay/CRCW060310K0FKEA/?qs=sGAEpiMZZMvdGkrng054t%252b8cnVAnHPZkqchqPaQRdrE%3d>. [Kasutatud 15 mai 2017].
- [44] Mouser, „Molex 22-28-8040,“ [Võrgumaterjal]. Available: <http://www.mouser.ee/ProductDetail/Molex/22-28-8040/?qs=sGAEpiMZZMs%252bGHln7q6pm%252bS0pk2Wo0Xx4nQNAUqdgA8%3d>. [Kasutatud 15 mai 2017].
- [45] Dutyhook, „M3x12 Sisekuuskant polt, peitpeaga, DIN 7991, happekindel,“ [Võrgumaterjal]. Available: <http://www.dutyhook.eu/bolt/flat-head/m3x12-din7991-a4.html>. [Kasutatud 18 mai 2017].

- [46] Farnell, „MURATA GRM033R60J104KE19D SMD Multilayer Ceramic Capacitor,“ [Võrgumaterjal]. Available: <http://ee.farnell.com/murata/grm033r60j104ke19d/cap-mlcc-x5r-100nf-6-3v-0201/dp/1775994>. [Kasutatud 14 mai 2017].

**LISAD**

## Lisa 1 Lähtekood I<sup>2</sup>C seadme aadressi testimiseks

```
#include <driver/i2c.h>
#include <esp_log.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <stdio.h>
#include "sdkconfig.h"

#define SDA_PIN 21
#define SCL_PIN 22

static char tag[] = "i2cscanner";

void task_i2cscanner(void *ignore) {
    ESP_LOGD(tag, ">> i2cScanner");
    i2c_config_t conf;
    conf.mode = I2C_MODE_MASTER;
    conf.sda_io_num = SDA_PIN;
    conf.scl_io_num = SCL_PIN;
    conf.sda_pullup_en = GPIO_PULLUP_ENABLE;
    conf.scl_pullup_en = GPIO_PULLUP_ENABLE;
    conf.master.clk_speed = 100000;
    i2c_param_config(I2C_NUM_0, &conf);
    i2c_driver_install(I2C_NUM_0, I2C_MODE_MASTER, 0, 0, 0);
    int i;
    esp_err_t espRc;
    printf(" 0 1 2 3 4 5 6 7 8 9 a b c d e f\n");
    printf("00:      ");
    for (i=3; i< 0x78; i++) {
        i2c_cmd_handle_t cmd = i2c_cmd_link_create();
        i2c_master_start(cmd);
        i2c_master_write_byte(cmd, (i << 1) | I2C_MASTER_WRITE, 1 /* expect ack */);
        i2c_master_stop(cmd);
        espRc = i2c_master_cmd_begin(I2C_NUM_0, cmd, 10/portTICK_PERIOD_MS);
        if (i%16 == 0) {
            printf("\n%.2x:", i);
        }
        if (espRc == 0) {
            printf(" %.2x", i);
        } else {
            printf(" --");
        }
        //ESP_LOGD(tag, "i=%d, rc=%d (0x%x)", i, espRc, espRc);
        i2c_cmd_link_delete(cmd);
    }
    printf("\n");
    vTaskDelete(NULL);
}
[code]
void app_main()
{
    xTaskCreate(task_i2cscanner, "task_i2cscanner", 1024 * 2, (void*) 0, 10, NULL);
}
}
```

## Lisa 2 Anduriplatvormi lõplik välja arendatud lähtekood

```
#include "SparkFun_Si7021_Breakout_Library.h"
#include <Wire.h>
#include <WiFi.h>
#include "esp_deep_sleep.h"

const char* ssid = "nimi"; // WiFi-võrgu nimi
const char* password = "parool"; // WiFi-võrgu parool

const char* host = "data.sparkfun.com";
const char* streamId = "38zJ30x9YbUoj0bOYY0X";
const char* privateKey = "oM5ZjvI4G6UxE4y2ZZ47";

float humidity = 0;
float tempf = 0;
float temp = 0;
Weather sensor;

float last_temp = 0;
float last_rh = 0;
int firstRun = 0;

void setup()
{
  int wifiCounter = 0;
  Serial.begin(115200);
  sensor.begin();
  delay(10);

  // Wifi võrku ühendumine

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  // WiFi ühenduse kontrollimine
  while (WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

int value = 0;

void loop()
{
  delay(1000);
  efficientSending(); // Energiatarbe minimeerimise
  printInfo();
  ++value;
}
```

```

Serial.print("connecting to ");
Serial.println(host);

// TCP kliendiga ühendumine
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
  Serial.println("connection failed");
  return;
}

// Veebiaadressi koostamine andmete saatmiseks
String url = "/input/";
url += streamId;
url += "?private_key=";
url += privateKey;
url += "&humidity=";
url += float(humidity);
url += "&temp=";
url += float(temp);

Serial.print("Requesting URL: ");
Serial.println(url);

// Serverisse päringu saatmine
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
  "Host: " + host + "\r\n" +
  "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println(">>> Client Timeout!");
    client.stop();
    return;
  }
}

// Serverilt tulnud vastus kuvatakse jadapordi ekraanil
while(client.available()) {
  String line = client.readStringUntil('\r');
  Serial.print(line);
}

Serial.println();
Serial.println("closing connection");
}

void getWeather()
{
  // Suhtelise õhuniiskuse mõõtmine
  humidity = sensor.getRH();

  // Temperatuuri mõõtmine Fahrenheitt ning Celsius skaalal
  tempf = sensor.getTempF();
  temp = sensor.getTemp();
}
//-----
void printInfo()
{
  //Jadapordi ekraanile kuvatakse mõõdetud näidud

```



```

Serial.print("Temp:");
Serial.print(tempf);
Serial.print("F, ");

Serial.print("Temp:");
Serial.print(temp);
Serial.print("C, ");

Serial.print("Humidity:");
Serial.print(humidity);
Serial.println("%");
}
void efficientSending()
{
// Avatakse jadapordi ja I2C ühendus
Serial.begin(115200);
sensor.begin();
delay(10);
getWeather();

// Lubatud temperatuuri ja õhuniiskuse vahemike määramine
float temp_max = tempf + (tempf * 0,01);
float temp_min = tempf - (tempf * 0,01);
float rh_max = humidity + (humidity * 0,05);
float rh_min = humidity - (humidity * 0,05);

int session = 0;
int var = 0;

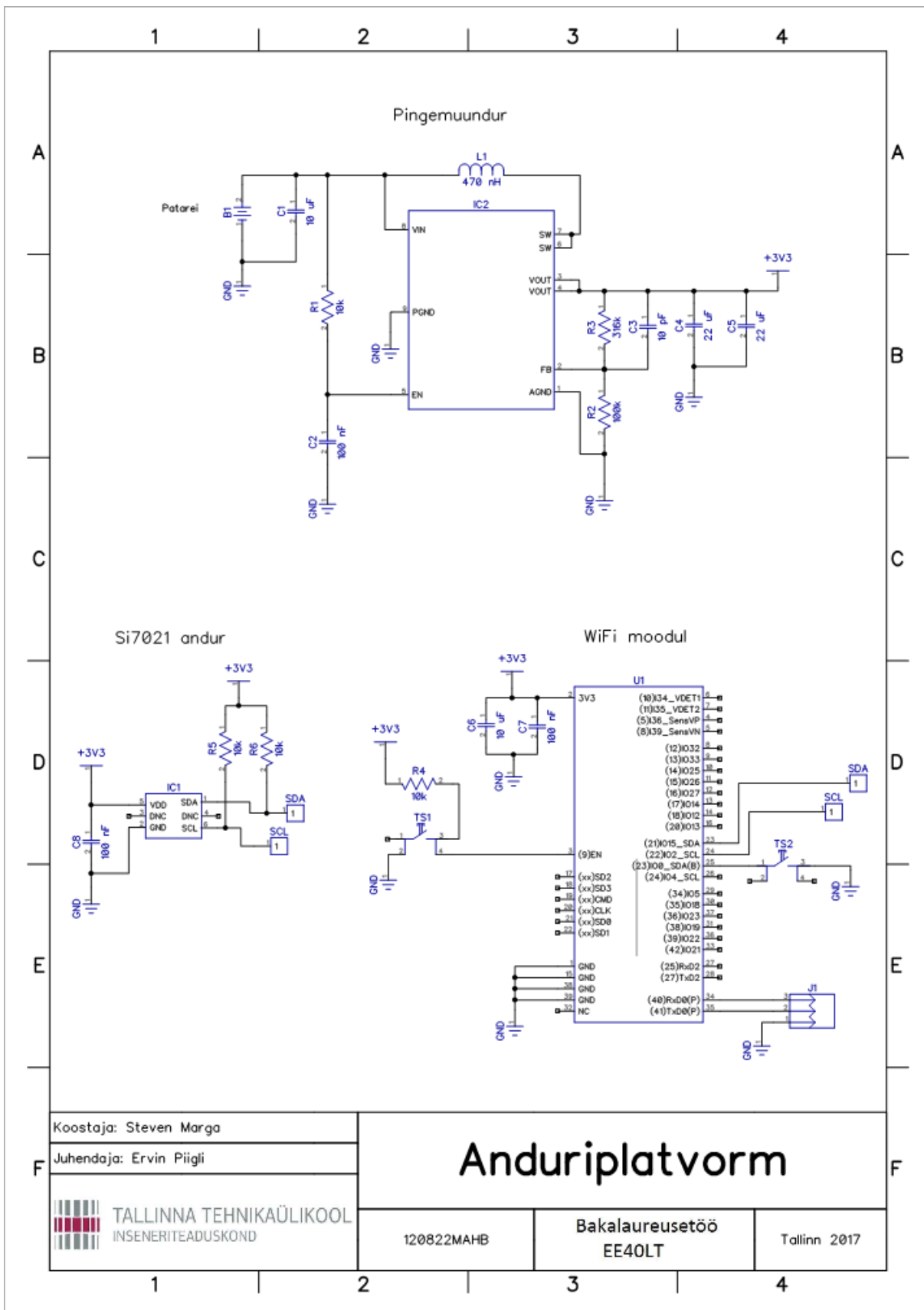
if(((last_temp > temp_min) && (last_temp < temp_max)) && ((last_rh > rh_min) || (last_rh <
rh_max)) && (firstRun != 0)) { // Näitude võrdlemine määratud vahemikega
while(session < 3) {
delay(10000);
sensor.begin();
delay(10);
getWeather();

if((tempf > temp_min && tempf < temp_max) && (humidity > rh_min || humidity < rh_max)) { //
Näitude võrdlemine vahemikega uuesti
session++;
var++;
}
else{
session = 3;
}
if((session == var) && (session == 3)){
Serial.println("Sisenetakse une režiimi");
esp_deep_sleep(600000000); // Unerežiimi sisenemine, väärtus mikrosekundites
}
}
}

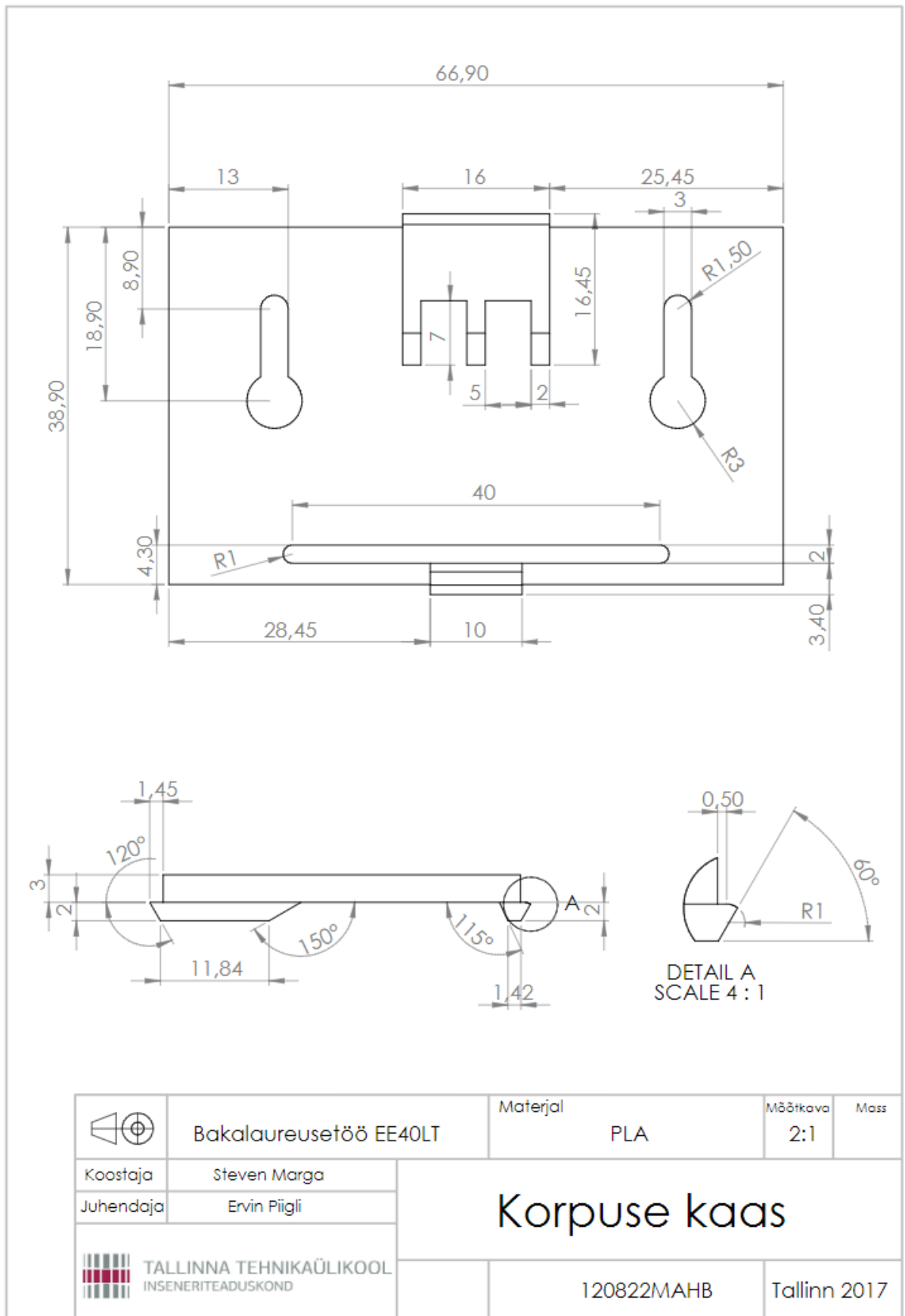
last_temp = tempf;
last_rh = humidity;
firstRun++;
}

```

# Lisa 3 Anduriplatvormi elektriskeem





## Lisa 4 Korpuse kaas



## Lisa 5 Distantspuks

The drawing shows a circular cross-section of a ring with an outer diameter of  $\Phi 6 \nabla 5$  and an inner diameter of  $\Phi 3 \nabla 5$ . The ring is mounted on a square base. The drawing includes a top view and a side view, with dashed lines indicating the hidden parts of the object.

	Bakalaureusetöö EE40LT	Materjal	PLA	Mõõtkava	10:1	Mass		
Koostaja	Steven Marga	<h1>Distantspuks</h1>						
Juhendaja	Ervin Piigli							
	TALLINNA TEHNIKAÜLIKOOL INSENERITEADUSKOND		120822MAHB		Tallinn 2017			

## Lisa 6 Patareipesa hoidja

