

# Loppuraportti

## Project #04

## Sähkökeskuksen kuormanohjausyksikkö



Date: 31.8.2018

Ilpo Pere  
Riikka Nousiainen  
Henri Tuovinen  
Anton Klami  
Leo Andersson

# Information page

## Students

Ilpo Pere

Riikka Nousiainen

Henri Tuovinen

Anton Klami

Leo Andersson

## Project manager

Henri Tuovinen

## Sponsoring Company

Ensto Oy

## Starting date

4.6.2018

## Submitted date

31.8.2018

## Tiivistelmä

Projektin tarkoitus oli tuottaa yritykselle prototyyppi kuormanohjausyksiköstä liitettäväksi kotitalouden sähkökaappiin. Yksikkö mittaa ja visualisoi kulutusdataa, sekä mahdollistaa yksittäisten kuormien pudotuksen. Ohjaus tapahtuu automaattisesti, mutta tarvittaessa se on mahdollista myös etänä pilven yli.

Yksikkö palvelee kuluttajaa, sähköasentajaa ja operaattoria. Kuluttaja hyötyy datan visualisoinnista ja voi tiedon varassa muuttaa kulutuskäyttäytymistään. Sähköasentajalle yksikkö antaa tietoa kulutushistoriasta, jota voidaan käyttää hyödyksi uusien kytkentöjen tekemiseen. Operaattorille yksikkö mahdollistaa kuormien pois kytkennän etänä, jolloin yksikkö toimii työkaluna osana laajempaan verkon hallintaa. Automaattinen kuormanohjaus estää epätoivottuja kulutuspiikkejä ja tasoittaa kulutusta sulkemalla ja avaamalla kuormia.

Laitteisto koostuu ohjattavista releistä, virtamittareista, mikrokontrollerista, piirilevystä, sekä virtalähteistä kontrollerille ja releille. Piirilevy tuotettiin ensin itse, mutta lopullinen projektissa käytettävä piirilevy päätettiin tilata valmistajalta. Tähän piirilevyyn tehtiin myös lopulliset kytkennät.

Projektissa ehdimme toteuttaa kaikki halutut toiminnallisuudet ajallaan. Ainut lisättävä ominaisuus, joka jäi toteuttamatta oli mobiilisovellus, joka olisi auttanut kuluttajia käyttämään ja tutustumaan laitteeseen helpommin. Mobiilisovelluksen sijaan etäohjaus toimii yksinkertaisella nettisivulla.

Automaattisen ohjauksen kanssa kokeilimme monenlaisia algoritmeja. Päädyimme lopulta ohjausjärjestelmään, jossa pilvipalveluun syötetään kullekin kuormalle oma prioriteettinsa ja tunnin aikana käytettävä maksimienergia. Syötetyn maksimituntienergian avulla ohjelma laskee maksimitehon, jäljellä olevan ajan ja energian perusteella. Lisäksi jos hetkellinen teho on alle maksimitehon, kertyy energiaa talteen, joka mahdollistaa korkeamman maksimitehon lopputunnille. Täten tunnin alussa käytettävä teho on matalin, ja tunnin edetessä käytettävä teho myös kasvaa, ellei teho ole jatkuvasti rajapinnassa.

Maksimitehon avulla laite rajoittaa virran kulutusta sulkemalla kuormia prioriteettijärjestyksessä tehon ylittäessä sallitun maksimitehon. Ohjelma osaa myös palauttaa suljettuja kuormia takaisin kulutuksen laskiessa tarpeeksi matalaksi. Tähän ohjelma käyttää tallentamaansa kuorman viimeistä tehoa ennen sen sulkemista ja vertaa sitä nykyiseen kulutukseen.

Lisäksi ohjelma suojelee pääsulaketta pudottamalla kuormia ennen pääsulakkeen laukeamista. Esimerkkitilanne sulakkeen laukeamisesta voi olla oikosulku, taikka lopputunnin korkea arvoinen sallittu hetkellinen teho. Tässä huomasimme ongelman ohjelmassa sulakkeiden suojelemisessa viimeisellä viikolla laitetta esiteltäessä. Ohjelma sulkee kuorman tarpeeksi aikaisin, mutta jos kuorman sulkemisen syynä oli oikosulku ja täten epänormaalin korkea teho, ohjelma ei palauta kuormaa. Tässä tilanteessa ainut tapa palauttaa suljettu kuorma on resetoida laite manuaalisesti.

## Abstract

The goal of the project is to produce a prototype load controller which can be connected to a household distribution board. The unit measures and visualizes energy consumption data and enables switching off of individual loads. Controlling happens automatically, but it is also possible to control the device manually through a cloud service.

The device serves consumers, electricians and the operator. The consumer benefits from data visualization and can alter his consumption behavior accordingly. The unit gives consumption data to the electrician, which can help with new installations. The device enables the operator to remotely switch off loads, in which case the device works as a tool for larger scale grid control. Automatic load control prevents undesirable consumption spikes and unnecessary fuse or circuit breaker trips.

The hardware is composed of relays, current sensors, microcontroller, circuit board and the power sources for the microcontroller and relays. The necessary couplings to couple the microcontroller to the distribution board are complete. The first version of the circuit board was designed and assembled by us, however, we decided to order our final design of the circuit board from a pcb manufacturer. The final couplings are also made to this circuit board.

During the project we were able to complete all of the desired functionalities in time. Only additional feature that we didn't have time to complete was a mobile app which would have helped the consumer in using and getting to know the device. Instead of an app the remote control works using a rudimentary website.

We tried multiple algorithms for the automatic control. We ended up with a control system where the user inputs different priorities for each load and the maximum energy allowed for each hour.

With the maximum hourly energy the program calculates maximum power, based on the remaining time and energy. In addition if the power being used is below the threshold, extra energy will be stored for later use. Therefore the available power is lowest at the beginning of the hour, while it increases as time passes and you approach the end of the hour.

When the maximum power is exceeded, the device limits the current usage by closing loads starting with the one with the lowest priority. The system can also return closed loads when the momentary consumption has lowered enough. For this the device saves the last consumption value before closing a load and compares it to the current live consumption.

In addition the device protects main fuse by dropping out loads before the fuse trips. Causes for the tripping of a fuse may be for example a short circuit or too high allowed maximum power at the end of the hour. Here we found out a possible problem with the protection of the fuses at the last week while presenting the device. The device does close the load in time before the fuse trips, but the value saved for returning the load is extremely high, if the cause for shutting the load was a short circuit. Therefore the device never returns this kind of load and the only way to reopen it, is to reset the device manually.

# Sisällysluettelo

1. Johdanto.....	6
2. Tavoite.....	6
3. Laitteisto.....	7
3.1. Piirilevy.....	7
3.2. Sipy.....	11
3.3. Sähkökeskus.....	13
3.4. Sensorit.....	18
3.5. Releet.....	19
4. Ohjelmisto.....	20
4.1. Mittaus.....	20
4.2. Ohjaus.....	23
4.3. Pilvi.....	24
5. Projektitoiminta.....	26
5.1. Tavoitteiden saavuttaminen.....	26
5.2. Aikataulu.....	26
5.3. Riskit.....	27
5.4. Projektin toteuma.....	29
6. Yhteenveto ja johtopäätökset.....	30

# 1. Johdanto

Projektin tarkoituksena oli tuottaa yritykselle prototyyppi sähkökeskuksen kuormanohjausyksiköstä kotitalouskäyttöön. Yksikkö kerää reaaliaikaista kulutusdataa pilveen, sekä mahdollistaa kuormien ohjauksen määrättyjen parametrien mukaisesti tai etänä. Valmis prototyyppi palvelee kuluttajaa, sähköasentajaa, sekä operaattoria.

Kuluttajalle prototyypin on tarkoitus visualisoida sähkönkulutusta, sekä kotiverkon rajoitteita. Asiakas pystyy näiden tietojen pohjalta tekemään päätöksen esimerkiksi siitä, koska sähköauton lataus on järkevää ja antaa mahdollisesti suosituksia siitä, mitä kuormia on suositeltavaa sulkea latauksen ajaksi. Yksikkö valmistelisi kotitaloutta myös tehopohjaiseen sähkönsiirtohinnoitteluun, jolloin kuormien ohjaus toteutuisi tunnin maksimitehon mukaisesti.

Sähköasentaja pystyy asentamaan yksikön kotitalouden sähkökeskukseen ja määrittelemään rajoitteita ja parametreja sähkönkulutukselle. Sähköasentaja pystyy myös tallennetun datan avulla arvioimaan mihin uudet laitteet, kuten sähköauton latausasema tai aurinkopaneeli kannattaa kytkeä.

Yksikkö antaa operaattorille mahdollisuuden ohjata kuormia etänä, jolloin yksikkö toimii osana laajempaa kuormanohjausjärjestelmää. Operaattori pystyy kytkemään pois kuormia, tasoittaakseen yllättäviä tehopiikkejä.

# 2. Tavoite

Tavoitteenamme oli toimittaa yritykselle toimiva prototyyppi, pohjaksi heidän tuotekehitykselleen.

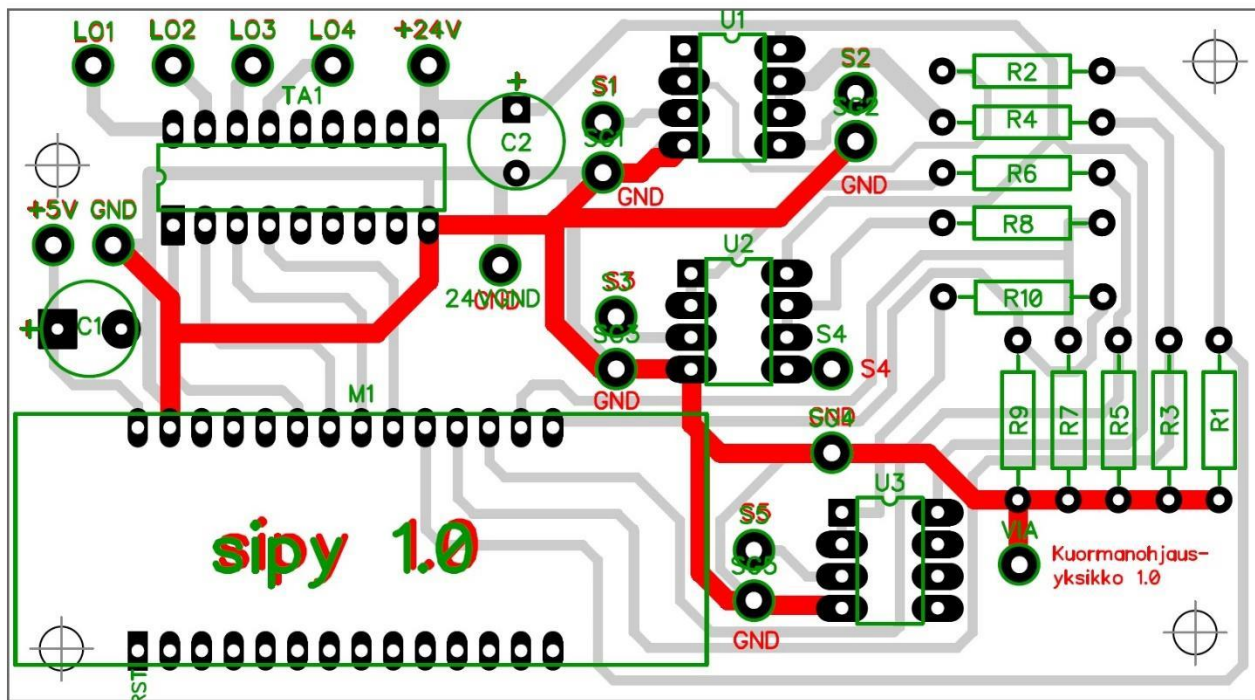
- Toiminnallisuudet
  - Mittaa ja rajoittaa kuormia, jolloin vältetään epätoivottuja kulutustilanteita.
  - Lähettää reaaliaikaista kulutusdataa pilveen ja kuormia voidaan rajoittaa myös etänä.
  - Sähkönkulutuksen ja rajoitteiden visualisointi kuluttajalle.
- Käyttäjät: Kuluttaja, sähköasentaja ja operaattori
- Lopputulos on toimiva prototyyppi laitteesta ja vaatimusmäärittely

## 3. Laitteisto

### 3.1. Piirilevy

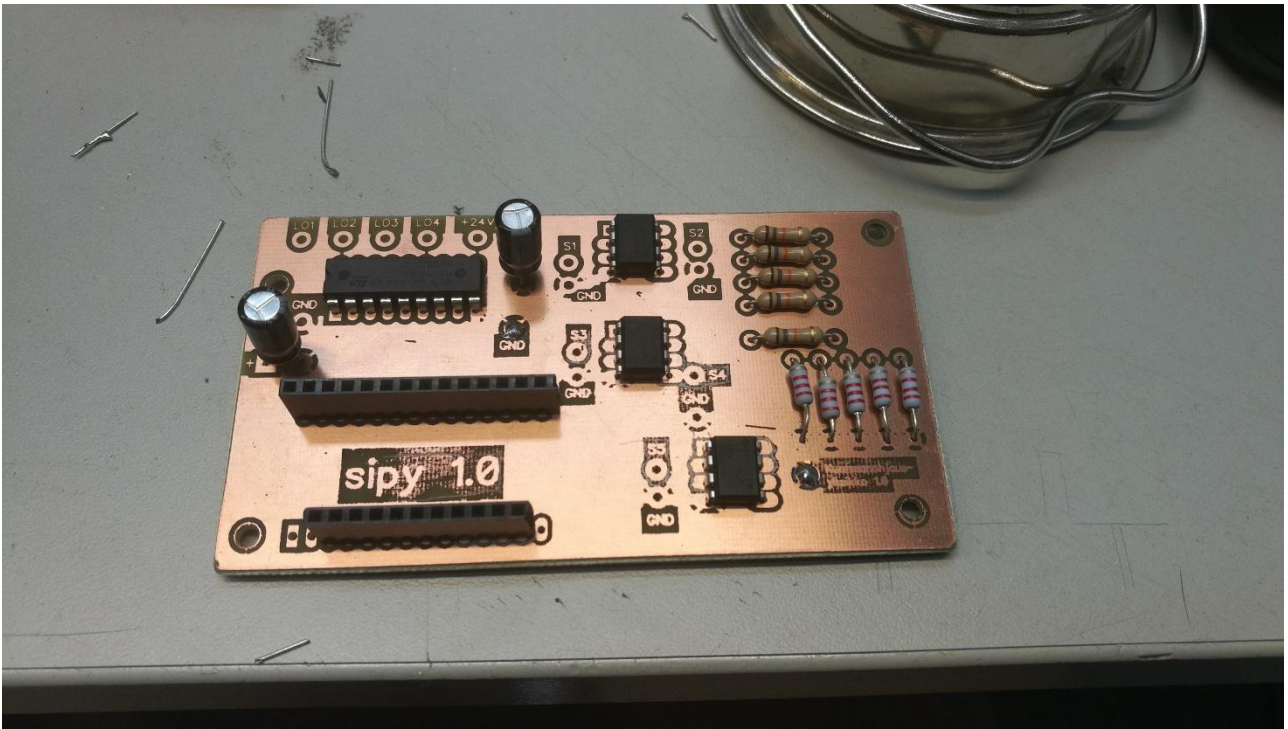
Suunnittelimme SiPylle tarpeellisen piirilevyn, katso kuva 1. Piirilevyn avulla SiPy voi ohjata releitä ja saada oikean suuruisia jännitteen arvoja. Piirilevyllä on transistorijoukko releiden ohjaamiseen. Piirilevyllä olevat operaatiovahvistimet seuraavat sensorien jännitteitä, jonka jälkeen jännitteet jaetaan vastusten avulla SiPylle sopiviksi. Valmistimme piirilevyn suunnitelmien mukaisesti pajalla, katso kuva 2. Piirilevyn valmistuksessa piirilevyn syöpymisen kanssa oli ongelmia, kun piirilevy syöpyi toiselta puolelta huomattavasti nopeammin kuin toiselta. Kun piirilevy oli saatu valmiiksi, testasimme sen toimintaa jännitelähteillä, emmekä löytäneet mitään siihen liittyviä ongelmia, katso kuva 3. Piirilevystä on suunniteltu myös toinen versio, katso kuva 5.

SiPyn A/D-muuntimen tarkoitukseemme sopimattomuuden vuoksi päätimme käyttää erillistä A/D-muunninta projektissamme. Tämä vaati uuden piirilevyn suunnittelemisen, katso kuva 4. Uusi piirilevy on kytkennältään muilta osin identtinen ensimmäisen version kanssa, mutta A/D-muuntimen takia levyllä on lisätty myös useita kondensaattoreita suodattamaan häiriöitä. Piirilevyä on muutenkin yritetty suunnitella mahdollisimman häiriövapaaksi, käyttäen mm. suuria maapintoja, joita ei tosin näy kuvassa selkeyden vuoksi. Piirilevyjä tilattiin Kiinasta, ja niiden laatu on erinomainen. Valitsimme 8-kanavaisen, 12-bittisen A/D-muuntimen 5V referenssijännitteellä, joka kommunikoi mikroprosessorin kanssa SPI-väylän kautta.

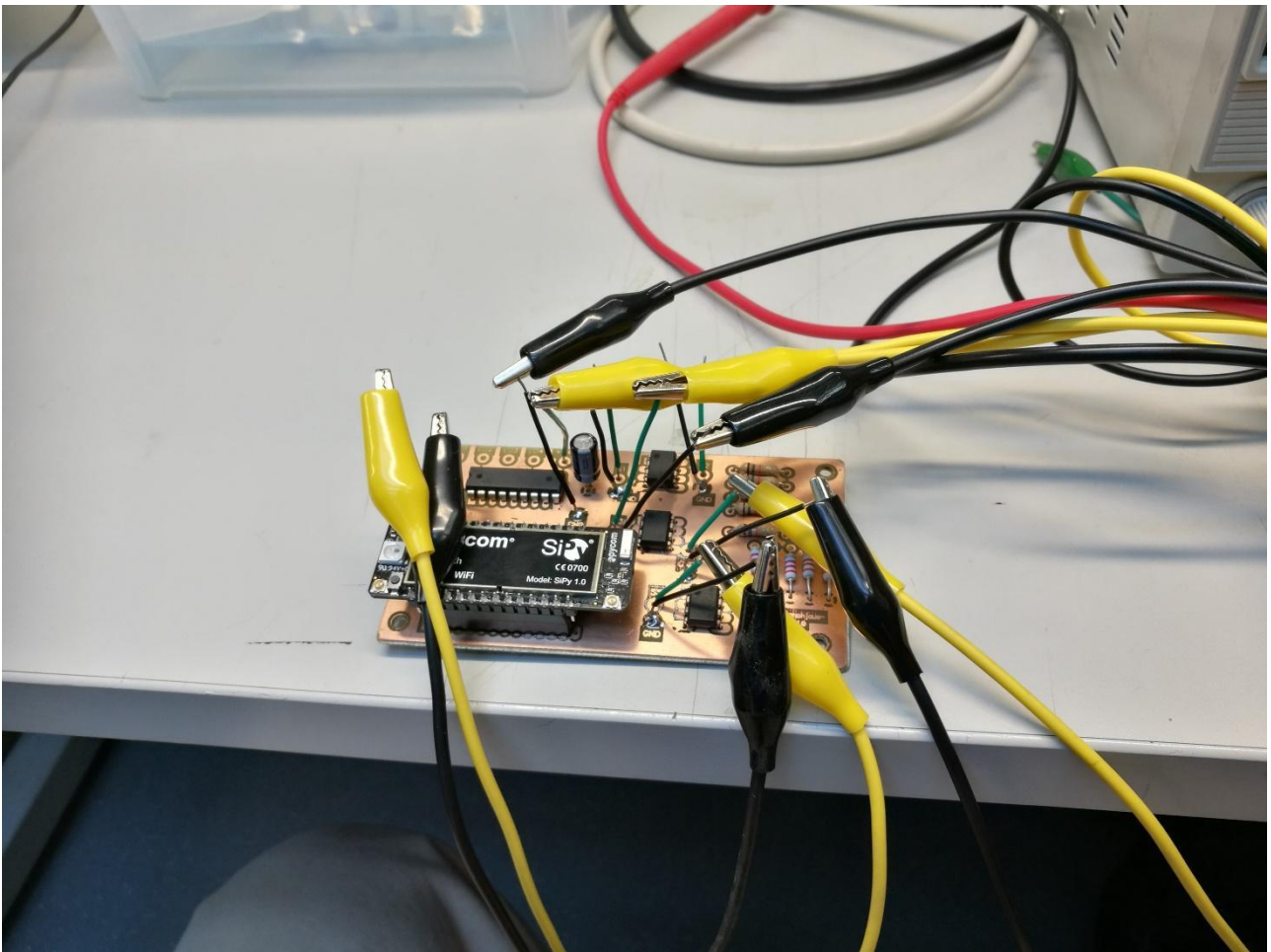


kuva 1. Piirilevyn suunnitelma



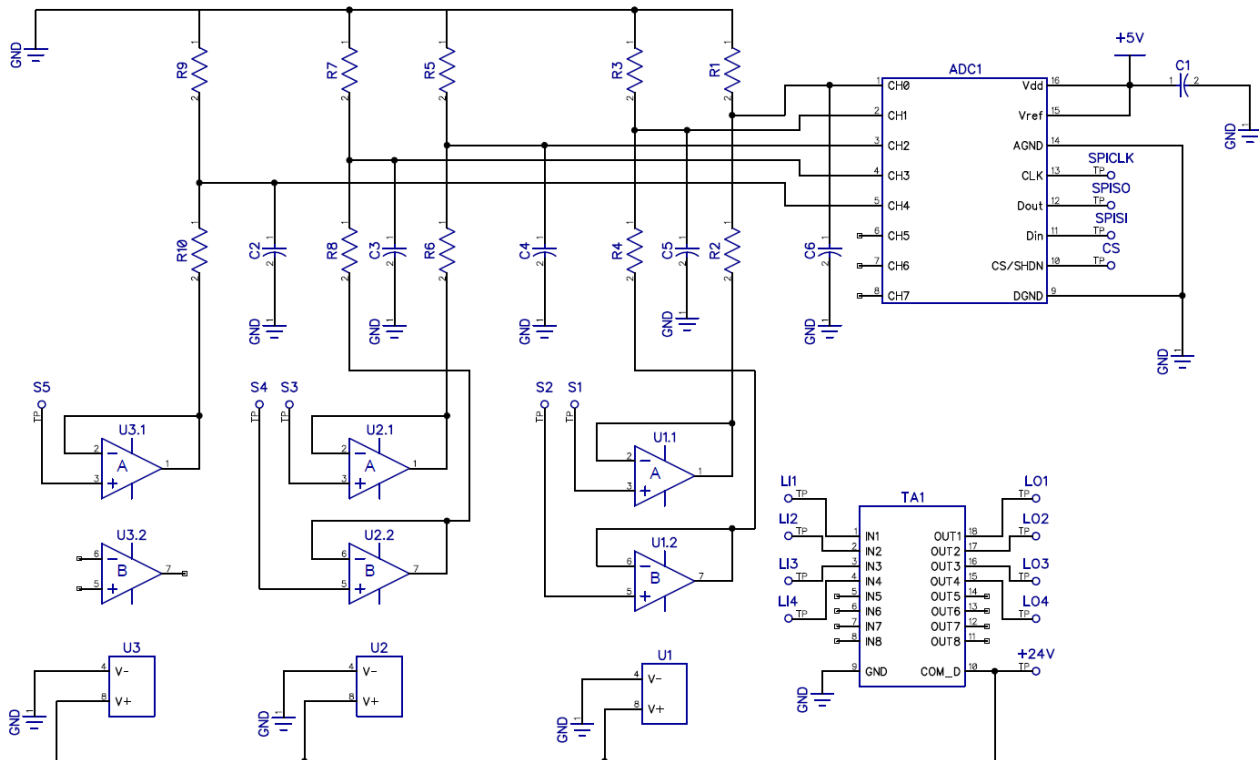


kuva 2. Koottu piirilevy

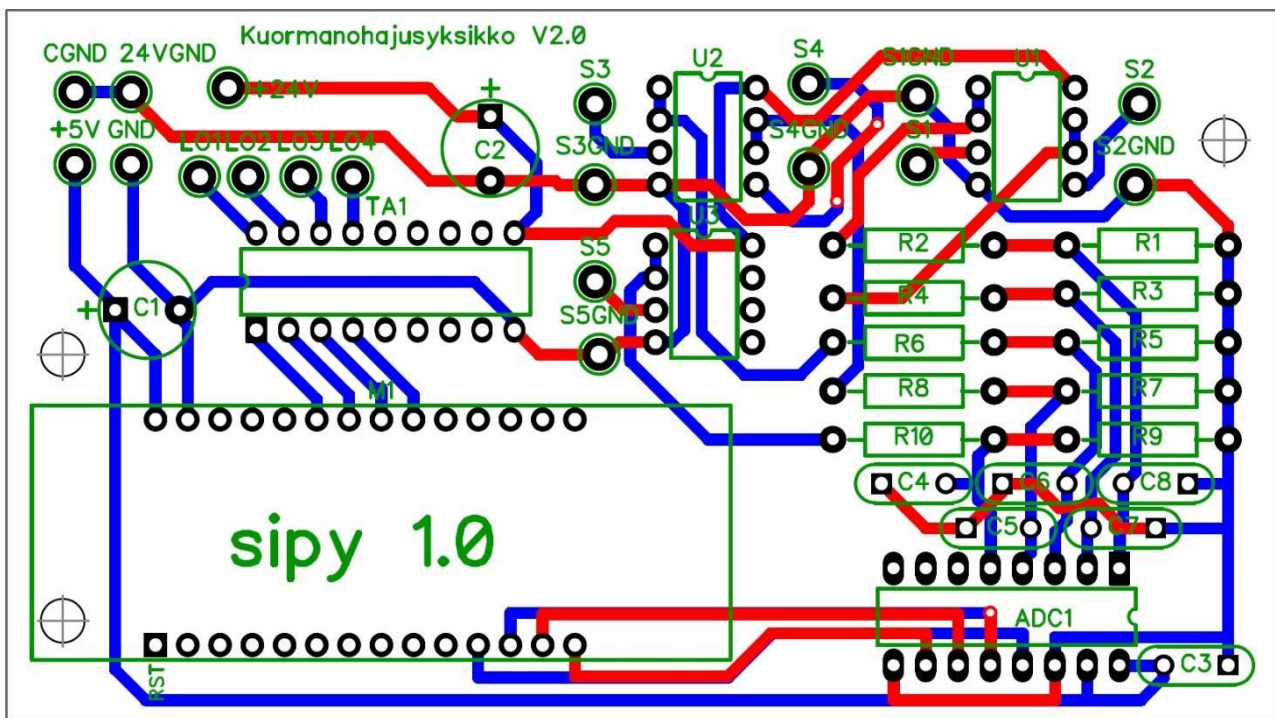


kuva 3. Piirilevyn testausta

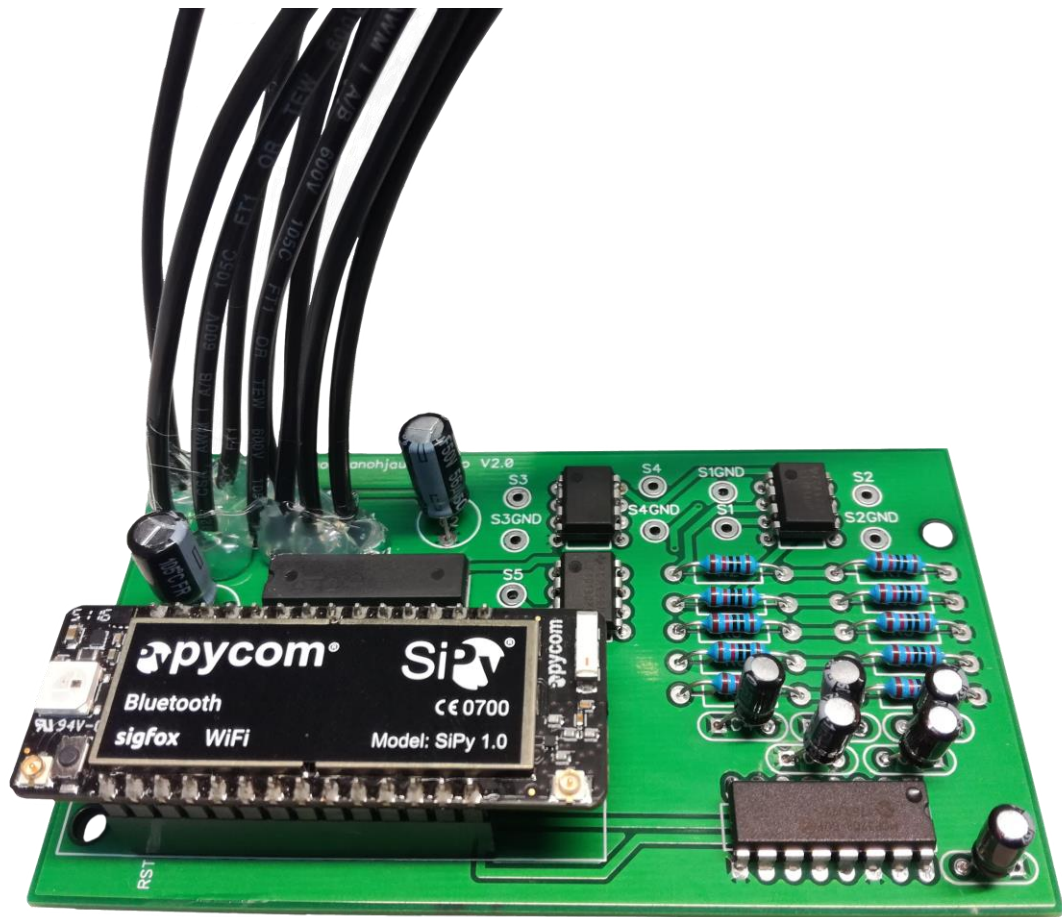




Kuva 4. Piirilevyn piirikaavio, ei sisällä sipyä



kuva 5. Piirilevyn toinen versio



kuva 6. Piirilevy lähes valmiina

### **3.2. SiPy**

Valitsimme mikrokontrolleriksi Pycomin valmistaman SiPyn, koska siinä on sisäänrakennettuna wifi, bluetooth, sekä A/D-muunnin. Lisäksi kontrollerissa on myös sisäänrakennettu SigFox, mutta tätä yhteyttä emme hyödyntäneet projektissa. Tarkoituksena oli aluksi tilata vastaava kontrolleri, WiPy, ilman kyseistä ominaisuutta. Niitä ei kuitenkaan tilaushetkellä ollut saatavilla tavarantoimittajalta ja hintaero kontrollereiden välillä on mitätön.

Kuten jo edellisessä luvussa mainittiin, jouduimme korvaamaan SiPyn A/D-muuntimen erillisellä muuntimella. A/D-muuntimen lisäksi toinen haittapuoli SiPyssä on pieni käyttäjäjyhteisö, eikä se perustu avoimeen laitteistoon esimerkiksi Arduino Unon tapaan. Tämä johti monien ongelmien ratkaisemiseen ”trial and error” -tyylillä.

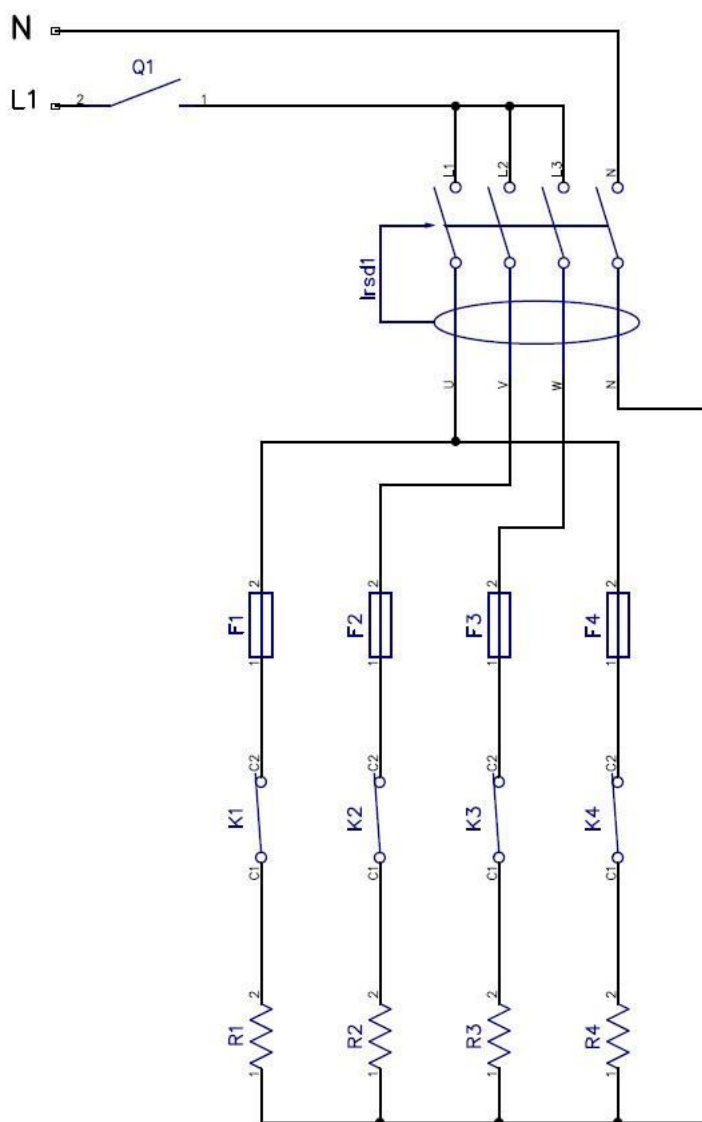
Kontrolleria ohjelmoitiin Pythoniin perustuvalla ja mikrokontrollereille optimoidulla MicroPythonilla. Lisäksi valmistajalla oli myös omia MikroPython-kirjastoja laitteilleen, joita käytimme ohjelmassa. Käytimme ohjelmoidessa editorina Atomia, sillä siihen oli saatavilla liitännäinen Pycomin laitteistolle (Pymakr). Liitännäinen mahdollisti ohjelman ajamisen SiPyssä Atomin avulla.

### **3.3. Sähkökeskus**

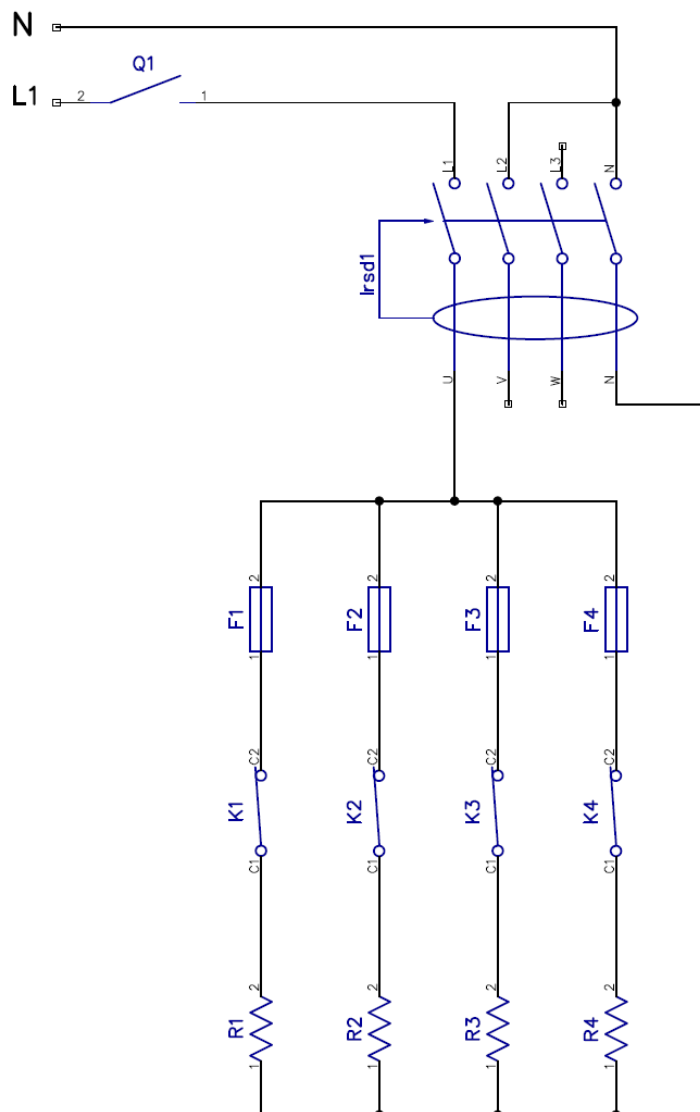
Keskuksen kaikki ohjattavat pistorasiat kytkettiin samaan vaiheeseen, katso kuva 7. Sähkökeskukseen on kytketty virtalähteet, releet ja pistorasiat, katso kuva 9. Sähkökeskuksen telinettä varten on myös tehty pohjakiinnike, joka estää keskuksen pohjan liikkumisen telineessä, katso kuva 11.

Alun perin keskus kytkettiin siten, että kaikki kolme vaihetta yhdistettiin pääkytkimessä, johon kytketyn syöttöjohdon ympärille laitettiin virtamittari. Tämän jälkeen vaihejohdin kytkettiin vikavirtasuojakytkimiin. Vikavirtasuojakytkimet olivat jo valmiiksi liitetty sulakkeisiin, joista neljä johtoa kytkettiin releisiin ja kaksi johtoa kytkettiin suoraan pistorasioihin, koska halusimme lisätä testausympäristöömme ohjaamatonta kuormaa. Releistä johdot kytkettiin pistorasioihin ja näiden johtojen ympärille laitettiin virtamittarit. Tämän lisäksi virtalähteet kytkettiin vikavirtasuojamattomaan sulakkeeseen.

Testattuamme keskusta huomasimme, että vikavirtasuojakytkimien testipainikkeet eivät toimineet. Tämän takia meidän piti tutkia testipainikkeen kytkentää, ja kävi ilmi, että se on kytketty ensimmäisen ja toisen vaiheen välille, eli kytkentää täytyi muuttaa. Lopullisessa kytkennässä vaiheita ei kytketty yhteen pääkytkimessä, ja vikavirtasuojakytkimestä sulakkeisiin on kytketty vain ensimmäinen vaihe. Tämän lisäksi vikavirtasuojakytkimen toisen vaiheen liittimeen on kytketty nollajohdin, jotta testipainike toimisi oikein yksivaiheisesti kytkettynä.

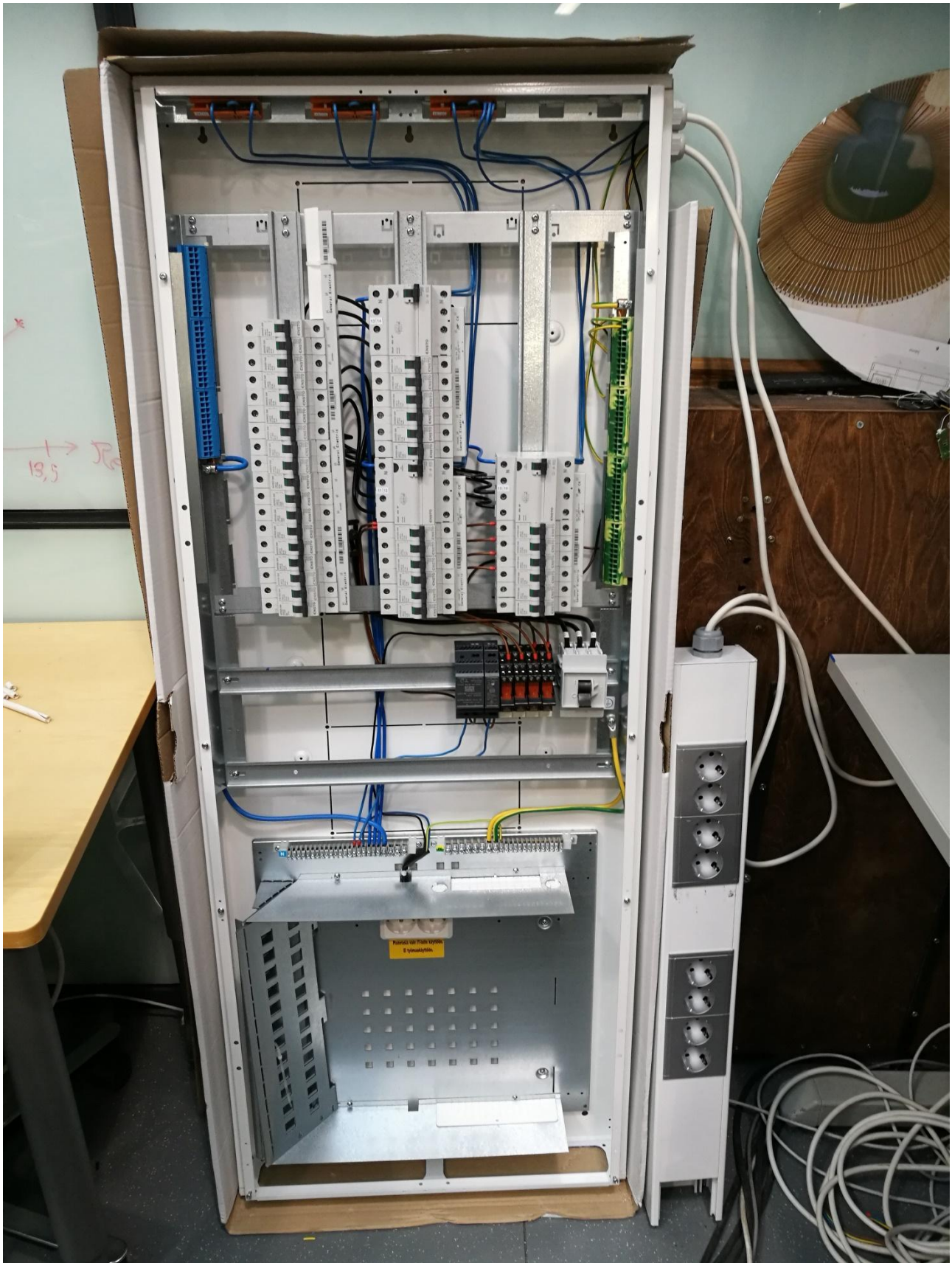


kuva 7. Keskuksen kytkentäkaavio



kuva 8. Keskuksen kytkentäkaavion parannettu versio



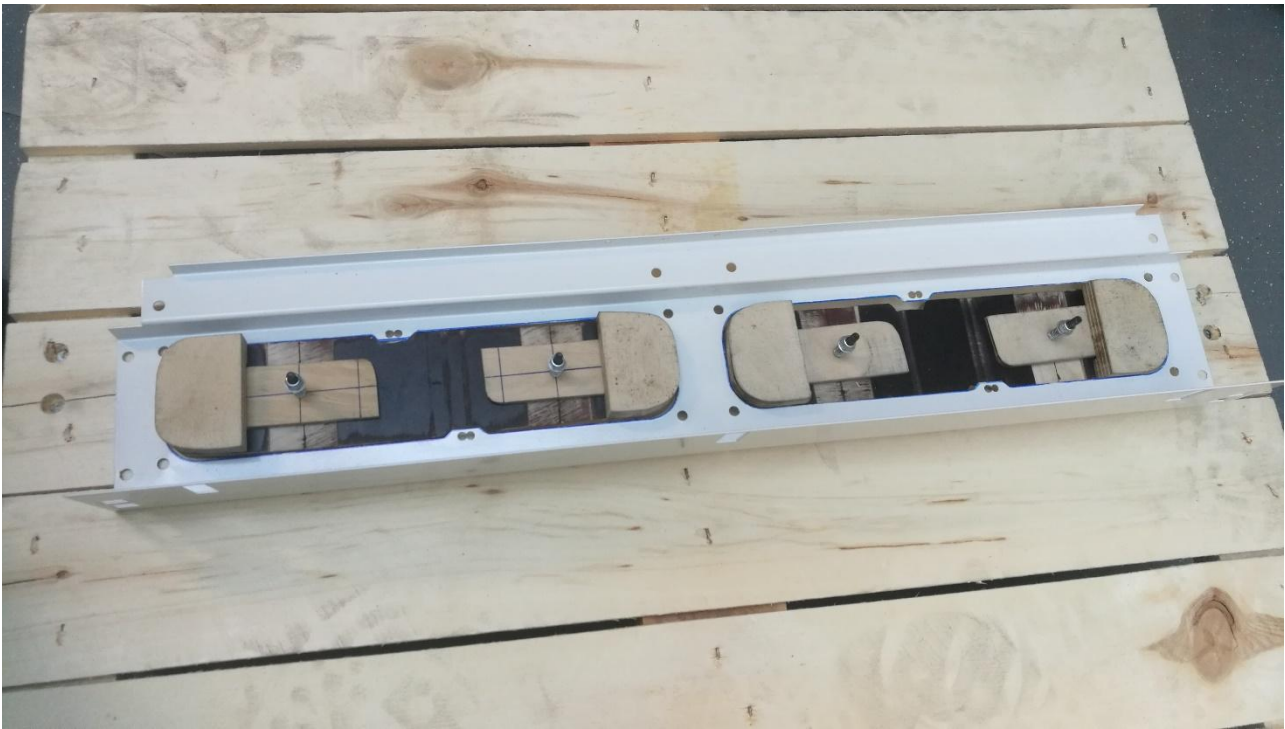


kuva 9. Sähkökeskus kytkentöjen jälkeen



kuva 10. Sähkökeskus projektin valmistuttua





kuva 11. Sähkökeskuksen pohjakiinnike

### **3.4. Sensorit**

Piirilevyyn on kytketty ohjausta varten viisi virtamittaria. Virtamittarit on kiinnitetty sähkökeskuksen DIN-kiskoon nippusiteillä ja niiden syöttöjohdot ovat häiriösuojattuja. Käyttöjännitteensä virtamittarit saavat passiivisesti mitattavasta johdosta. Virtamittarit mittaavat rms-virtaa ja antavat ulos lineaarisesti tätä virtaa vastaavaa jännitettä välillä 0-10V. Koska virtamittarit eivät käytä ulkoista virtalähdettä, ne pystyvät antamaan vain hyvin vähän virtaa, jonka vuoksi piirilevyllä on oltava hyvin suuri tuloimpedanssi.

Tehon mittauksessa pelkkä virta ei tietenkään riitä, vaan tarvitaan myös jännitteen suuruus. Tehon saisi laskettua virran ja jännitteen aaltomuodoista, mutta tällöin meidän olisi pitänyt valita toiset virtamittarit ja hankkia jännitemittari. Tämän lisäksi meidän olisi ollut järkevää hankkia energiamittarimikropiiri, joka laskisi energian virrasta ja jännitteestä. Virran ja jännitteen erikseen mittaamisessa hyvänä puolena on se, että saisimme erotettua pätö- ja loistehon, mikä on tavoiteltavaa, koska kuluttaja ei maksa loistehosta. Pohdittuamme asiaa päätimme kuitenkin kysyä kuinka tarkkoja mittauksia Ensto haluaa ja kävi ilmi, että on riittävää olettaa jännitteen suuruus vakioksi ja laskea teho sen avulla. Tämä yksinkertaisti mittausprosessia ja salli meidän käyttää jo aiemmin löytämiämme sensoreita.



kuva 12. Syöttöjohdon virtamittari

### 3.5. Releet

Keskukseen on asennettu neljä relettä, jotka ovat kytketty nc-asentoon. Tämän takia normaalitilassa kuormat saavat sähköä ja mikrokontrollerin ohjauksesta kuormat kytkeytyvät pois päältä. Releet käyttävät samaa 24V virtalähdettä kuin piirilevyllä olevat operaatiovahvistimet ja transistorit.



kuva 13. Ohjaukseen käytetyt releet

## 4. Ohjelmisto

### 4.1. Mittaus

Ohjelmiston ensimmäinen toteutettu ominaisuus oli mittaus. Tätä ennen piti kuitenkin toteuttaa ohjelmaan perusfunktioita, kuten kuormien tietojen avaus paikallisesta tiedostosta ja määrittellä kuormille ja päävaiheille oliopohjainen rakenne. Lopullisessa prototyypissä kuormien ja vaiheiden tiedot ladataan erillisestä tietokannasta palvelimen kautta.

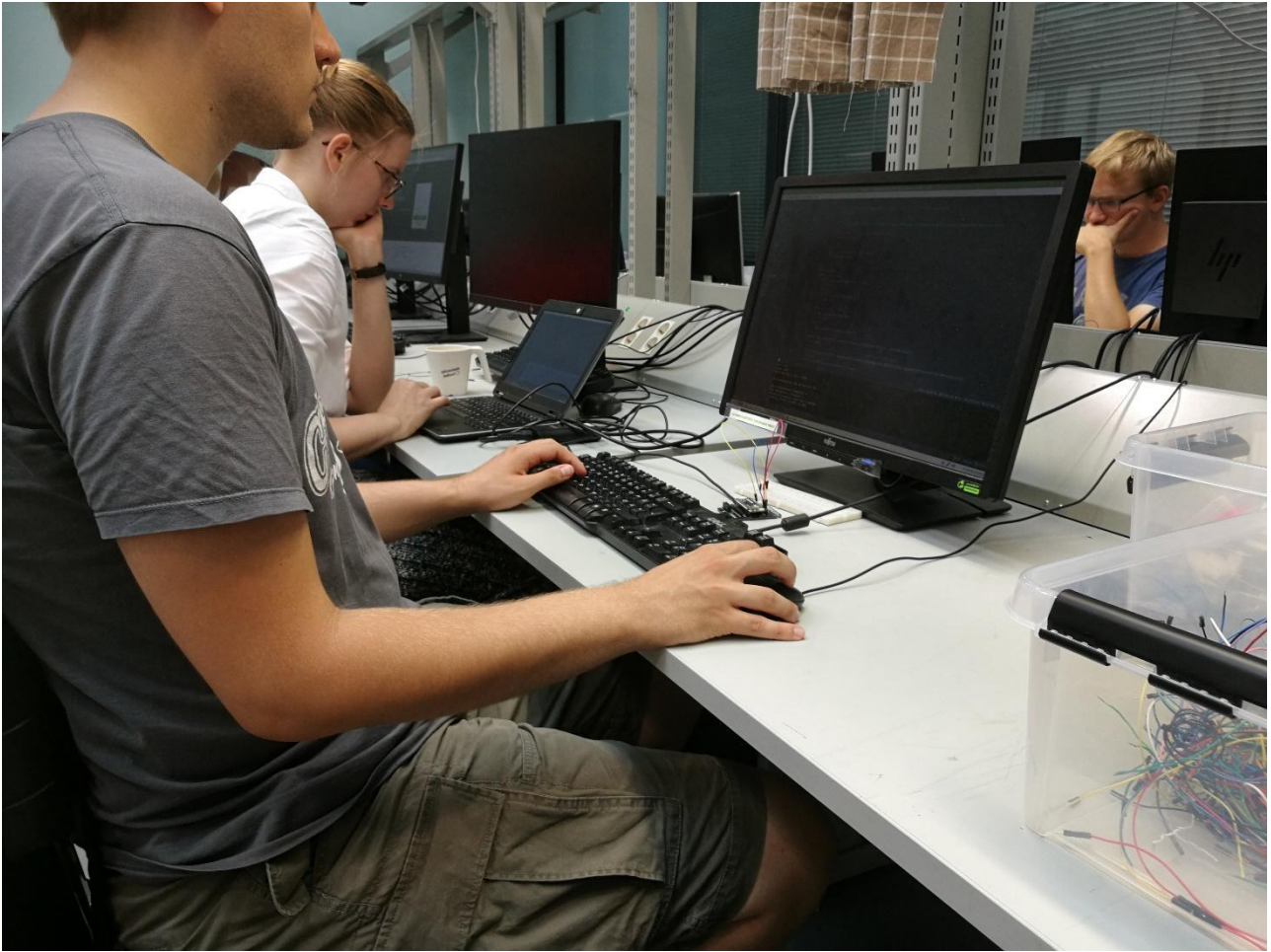
Mittaus toteutetaan virtasilmukalla, joka mittaa itsensä läpi kulkevaa virtaa. Virtasilmukka syöttää ulos jännitettä välillä 0-10V, joka riippuu virrasta lineaarisesti. Piirilevyssä vastuksilla toteutetulla jännitteen jaolla skaalattiin jännitettä pienemmäksi SiPylle.

Pienemmäksi skaalattu jännite syötettiin SiPyn GPIO(general purpose input output)-pinniin, joka on kytketty SiPyn A/D(analog-digital)-muuntimeen. A/D-muunnin muutti analogisen jännitteen digitaalseksi tiedoksi tietyllä tarkkuudella.

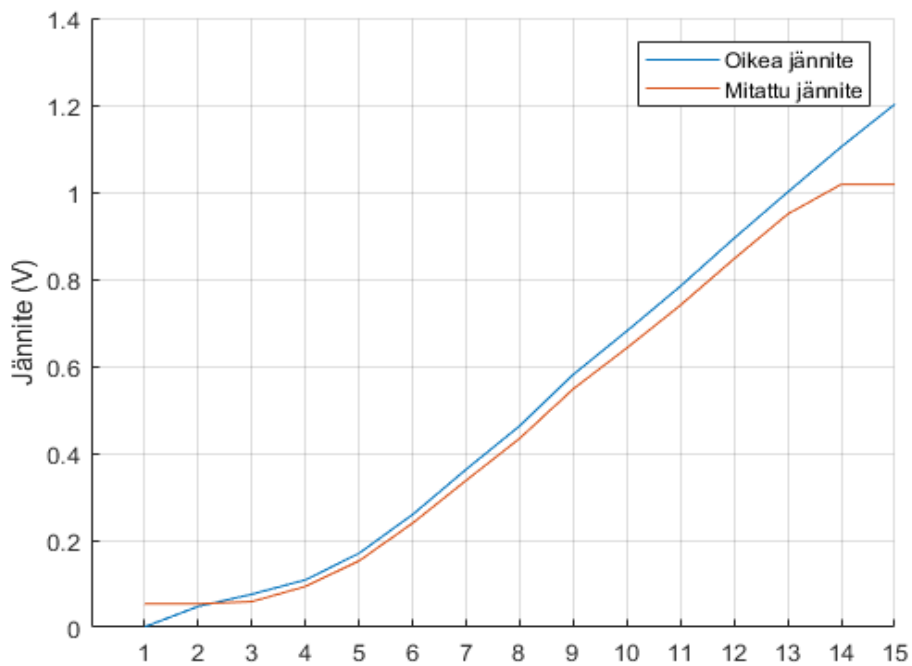
SiPyn A/D-muuntimen tarkkuus on  $2^{12}$  (4096) yksikköä ja se toimii välillä 0-1,1 voltia. SiPyllä on omat funktiot syötteen jännitteeksi muuttamiselle, jonka jälkeen oikealla kertoimella jännite voidaan muuttaa virtasilmukan mittaamaksi virraksi. Emme kuitenkaan päätyneet käyttämään SiPyn omaa A/D-muunninta, koska huomasimme sen olevan liian epätarkka. Tämä selvisi valmistajan vastauksesta käyttäjäyhteisön kysymyksiin muuntimen ongelmista, ja vahvistui omissa mittauksissa. katso kuva 15, kuva 16, kuva 17.

Lopullisessa prototyypissä A/D-muunnoksen toteuttaa erillinen A/D-muunnin, MCP3208, joka on piirilevyllä. Mittarit ovat kytketty operaatiovahvistimien kautta A/D-muuntimen sisäänsyöttöpinneihin, jotka jokainen vastaavat yhtä A/D-muuntimen kanavaa. Jännite on jälleen skaalattu pienemmäksi A/D-muuntimelle, tosin eri suhteessa johtuen suuremmasta referenssijännitteestä. SiPy pystyy SPI-protokollan avulla pyytämään eri kanavien mittausrvoja digitaalisessa muodossa.

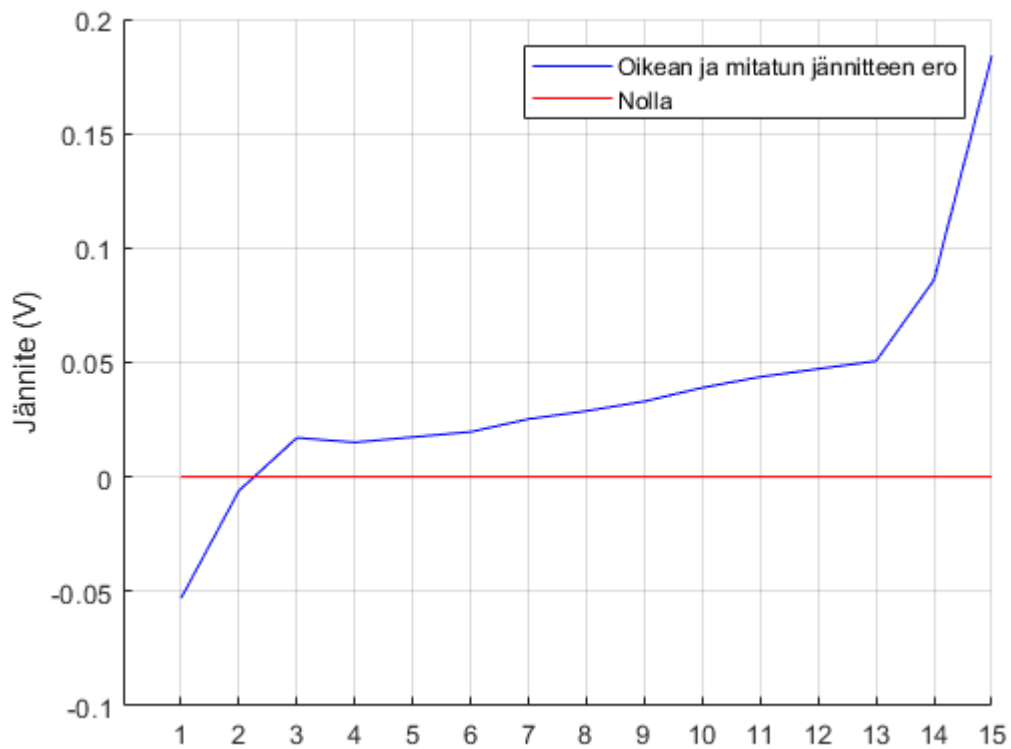
Mittauksia tehdään noin puolen sekunnin välein, tarkka arvo annetaan funktioille ajastimien avulla. Mitattuja virtoja tallennetaan hetkellisesti listoihin myöhempää käyttöä varten, jotta niiden keskiarvo voidaan lähettää pilveen. Lisäksi lasketaan tämänhetkinen tehonkulutus, jonka avulla voidaan laskea myös kulunut energia, kun otetaan huomioon mittausten väliset ajat.



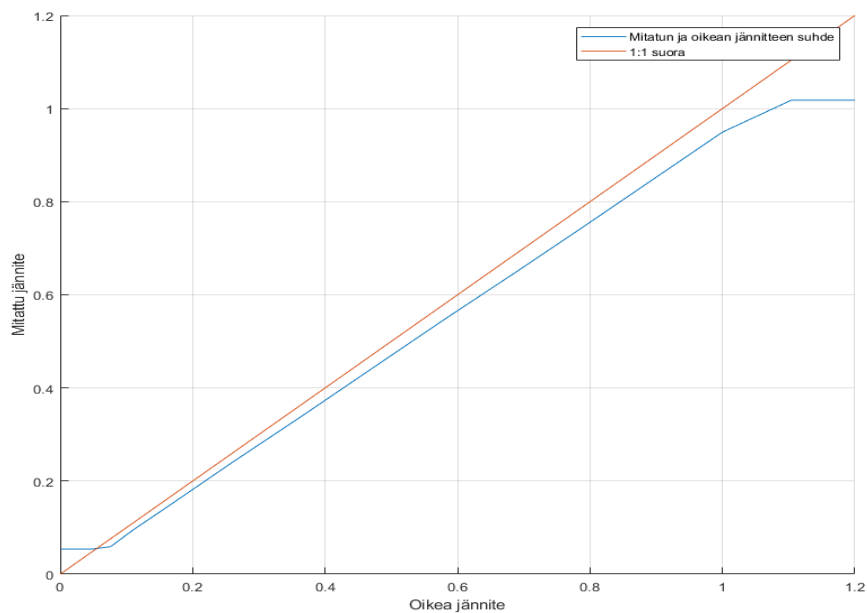
kuva 14. SiPyn ohjelmointia



kuva 15. SiPyn sisäisen A/D-muuntimen mittaaman jännitteen ja syötetyn jännitteen kuvaajat



kuva 16. SiPyn sisäisen A/D-muuntimen mittaaman jännitteen ja syötetyn jännitteenvälinen ero



kuva 17. SiPyn sisäisen A/D-muuntimen mittaaman jännitteen ja syötetyn jännitteen suhde

## **4.2. Ohjaus**

Ohjelmiston toinen ominaisuus on järjestelmän releiden ohjaus itsenäisesti. Releiden tarkoitus on olla kuormien ja verkon välissä, jossa niiden avulla virta voidaan katkaista tarvittaessa.

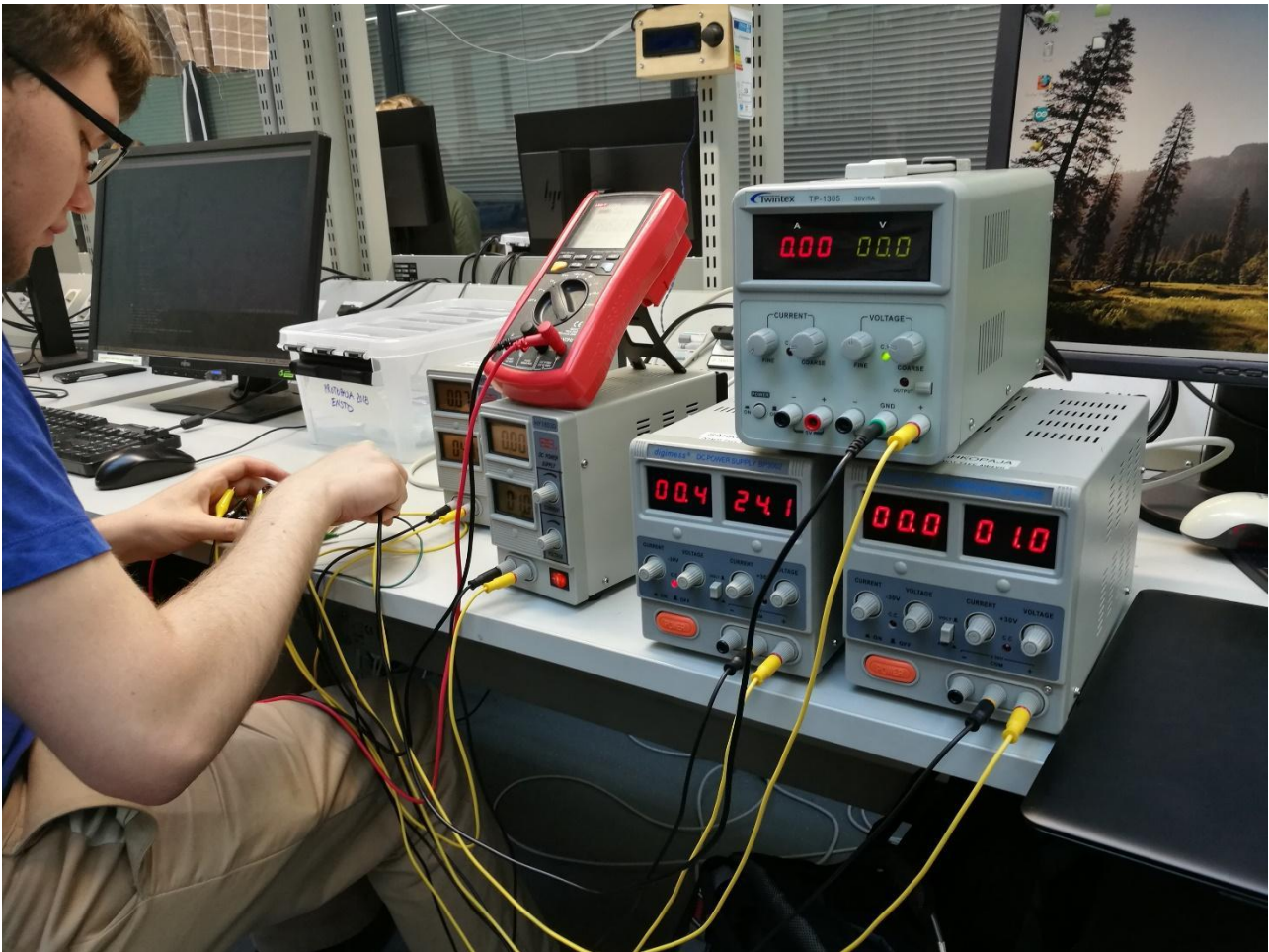
Tällaisia tilanteita ovat päävaiheen virran kohoaminen liian lähelle päävaiheen sulakkeen maksimivirtaa tai tuntitehon (eli tunnin aikana kulutetun energian) kohoamista liian lähelle maksimiarvoaan.

Jokaiselle kuormaoliolle on määritetty oma rele-pinni. Tämä rele-pinni toimii ON/OFF-periaatteella. Kun se ei ole päällä, releessä ei kulje virtaa ja silmukka on kiinni. Kun siinä on jännite, releeseen kulkee virtaa, joka aiheuttaa sähkömagneettisesti releen aukeamisen ja kuorman irtoamisen verkosta.

Mittauksien yhteydessä päävaiheiden virtoja verrataan maksimiarvoihin ja nykyistä kulutusta tunnin maksimikulutukseen. Mikäli kuormia pitää pudottaa, ohjelma käy kuormat läpi prioriteettijärjestyksessä, joka on määritetty valmiiksi. Pienimmän prioriteetin kuormat pudotetaan ensimmäisenä.

Samoin mittauksien yhteydessä verrataan nykyisen kulutuksen ja suljetun kuorman edellisen kulutuksen (ennen sulkemista) summaa maksimiarvoihin. Mikäli kuorma voidaan kytkeä takaisin ilman, että maksimiavot ylittyvät, tehdään niin.





kuva 18. Ohjauksen testausta jännitelähteiden avulla

### 4.3. Pilvi

Ohjelma käyttää uRequests-kirjastoa pilvipalvelun tietokantojen kanssa kommunikointiin. Tietokannat mittauksille ovat InfluxDB-tietokantoja ja niillä on API endpointit write ja query. Write-endpointiin pystyy oikeanlaisella käskyllä lähettämään dataa tietokantaan ja query-endpointista pystyy hakemaan dataa.

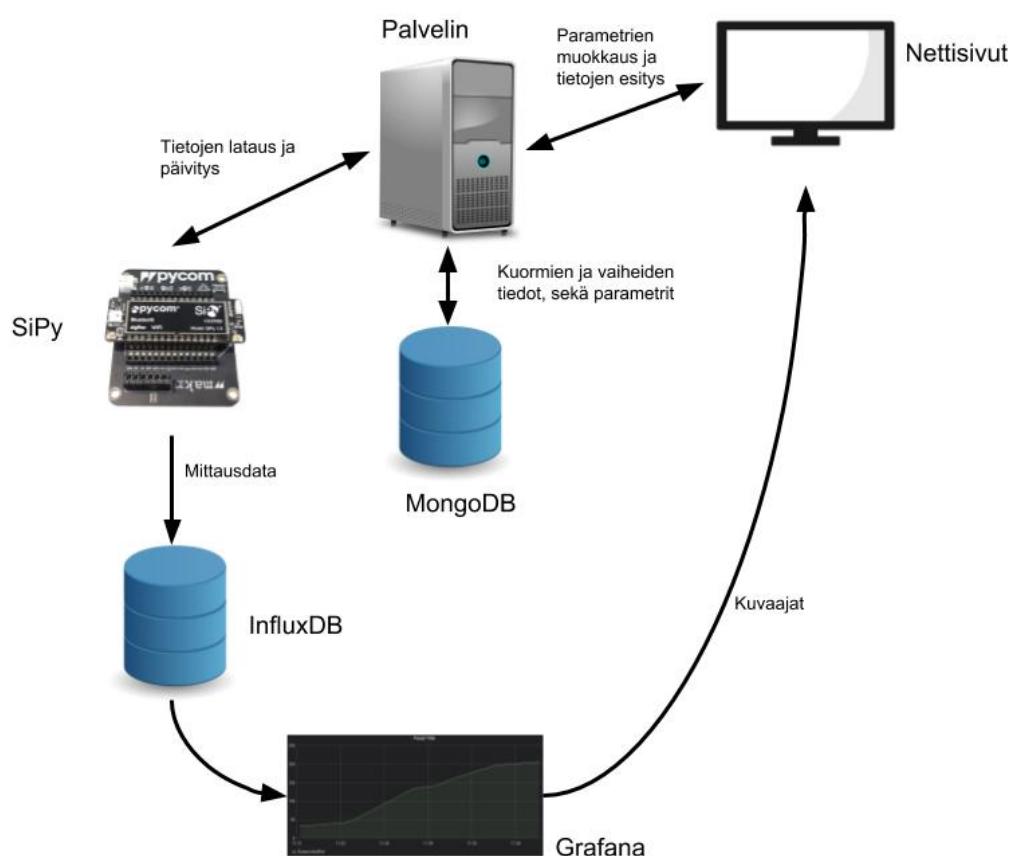
Ohjelma lähettää pilveen tietoa kuormista ja päävaiheista, kuten nimen, hetkellisen tehon ja tunnin energiankulutuksen. InfluxDB-tietokannat perustuvat aikasarjoihin, eli datan lisäksi ne myös tallentavat ajan jolloin data on lähetetty. Näin data on helppo visualisoida pilvipalvelun työkaluilla (Grafana), katso Kuva 19. Ohjelma lähettää tietoa pilveen vain 10 sekunnin välein, jonka aikana se laskee kuormille ja päävaiheille mittauksen keskiarvot lähetettäväksi.

SiPy lataa tiedot kuormista ja vaiheista käynnistyessään palvelimelta, mikä on yhteydessä tietokantaan (MongoDB). Erillistä tietokantaa tarvittiin, sillä InfluxDB soveltuu vain aikasarjoitetun datan tallentamiseen. Tietokantaan tallentuvat myös parametrit, joita on mahdollista manipuloida komentoriviltä tai käyttöliittymässä. Näitä parametrejä ovat kuormien prioriteetit ja ohjausarvot, sekä vaiheen maksimaalinen tuntiteho. Sivusto ja palvelin pyörivät tuotantopalvelimella (Heroku). Palvelinta ei ole suojattu. Ohjelmointirajapinta toteutettiin REST API:n mukaisesti. Kuormat, vaiheet ja manipuloitavat parametrit ovat resursseja, joille on



yksilöity URL-osoite. Resursseja voidaan hakea HTTP GET-pyyynnöllä. Muita käytettäviä metodeja ovat, PUT ja POST, joilla pystyy muokkaamaan parametreja, sekä lisäämään kuormia ja vaiheita tietokantaan.

Yksinkertainen käyttöliittymä toteutettiin Javascriptin React kirjastolla, mihin on liitetty myös Grafanan visualisointi kuormien ja vaiheen kulutuksesta. Käyttöliittymä mahdollistaa kuvaajan tarkastelun, kuormien ohjausparametrien muuttamisen, sekä maksimituntitehon määrittämisen. Sovelluksen front-end toteutettiin Bootstrapin kirjastolla.



kuva 19. Ohjelmiston ja tietokantojen väliset riippuvuudet.

## **5. Projektitoiminta**

### **5.1. Tavoitteiden saavuttaminen**

Ehdimme luoda laitteeseen kaikki haluamamme toiminnot. Alla yhteenveto toteutuneista toiminnallisuuksista:

- Kulutusdatan tallennus pilveen
- Datan visualisointi
- Automaatti- ja etäohjaus
- Käyttöliittymä etäohjaukselle

Tosin, jos aikaa olisi ollut enemmän olisimme nettisivun lisäksi tehneet laitteelle mobiilisovelluksen. Mobiilisovellus olisi tehnyt laitteen käyttämisestä yksinkertaisempaa ja helpompaa etenkin kuluttajalle. Lopullinen prototyyppi palvelee lähinnä operaattoria ja asentajaa. Toisaalta käyttöliittymän tarkoituksena olikin vain demonstroida ohjelmointirajapintaa.

Olemme itse tyytyväisiä lopputulokseen. Ensto oli myös erittäin tyytyväinen lopputulokseen ja piti rakennettua laitteistoa myös hyvänä kokeilupohjana omalle tutkimukselleen ja kehitystyölleen. Heidän mukaansa pelkkä valmiin projektin näkeminen herättää uusia ideoita.

### **5.2. Aikataulu**

Projektisuunnitelmassa olimme antaneet runsaasti lisää aikaa jokaiselle kohdalle. Täten itse projekti eteni suunnitelman aikataulun mukaisesti ja loppupuoella laitteen testaukseen jäi runsaasti toivottua aikaa.

Jokaisen viikon alussa pidetyt kokoukset myös auttoivat pysymään kartalla missä vaiheessa projekti etenee. Kokouksissa kävimme läpi eri ryhmien tehtävien edistymistä ja suunnittelimme tulevia viikkoja ja analysoimme projektin edistymistä.

Projektin alussa tehdyssä suunnitelmassa tehtyjä kaavioita, kuten kriittistä polkua tai Gantin karttaa emme tulleet tarkkailleeksi projektin edetessä. Koimme kuitenkin kaavioiden auttaneen projektin aikataulun ja etenemisen hahmottamista etenkin alkuvaiheessa.

Koska projektin alussa olimme ahkeria, ehdimme pitää suunnitelmassa olleen viikon loman lisäksi muutaman vapaapäivän nopean edistymisen kunniaksi.

### **5.3. Riskit**

Riskeihin olimme valmistautuneet luomalla listan eri riskeistä projektisuunnitelmaan projektin alussa. Tämä auttoi meitä valmistautumaan vastoinkäymisiin ja myöskin välttämään niitä. Esimerkiksi virtamittareita tilasimme viiden kappaleen sijaan aluksi vain yhden siltä varalta, jos muuttaisimme laitteen toimintaa, taikka laitteistoa. Budjettia jäikin lopulta runsaasti jäljelle.

Pieniä vastoinkäymisiä toki tuli vastaan, mutta ei mitään niin suurta, joka olisi hidastanut projektin etenemistä enempää kuin muutaman päivän. Suurin vastoinkäyminen oli SiPyn sisäänrakennetun A/D-muuntimen epätarkkuus, jota tarpeeksi tutkittuamme päädyimme ostamaan erillisen, ulkoisen A/D-muuntimen. Tämä johti piirilevyn uudelleen suunnittelemiseen, mutta onneksi olimmekin tehneet vain väliaikaisen piirilevyn pajalla.

Riski	Vakavuus (1-10)	Todennäköisyys (1-10)	Vaikutus (1-100)	Huomioonottaminen
Ajan loppuminen	8	5	40	Sovimme laitteen prioriteeteista yrityksen vastuuhenkilön kanssa. Jos aika loppuu kesken, voimme jättää toteuttamatta osan laitteen ominaisuuksista tärkeysjärjestyksessä.
Budjetin loppuminen	5	2	10	Tilaamme laitteen komponentit heti alkuvuikoilla, jotta projektin kustannuksissa ei tule yllätyksiä myöhemmin kesken projektin. Jos projektin budjetti on loppumassa kesken, otamme asian puheenaiheeksi yrityksen vastuuhenkilön kanssa ja sovimme järjestelyistä.
Osien viivästyminen	2	4	8	Tilaamme osat ajoissa. Koodia pystyy kirjoittamaan jossain määrin ennen osien saapumista.
Sähkökaapin tuhoutuminen	9	1	9	Työturvallisuuden huomioonottaminen, tarkka työskentely, pyydetään epäselvässä tilanteessa apua assistentilta.
Muun laitteiston rikkoutuminen	3	2	6	Työturvallisuuden huomioonottaminen, tarkka työskentely, pyydetään epäselvässä tilanteessa apua assistentilta.
Yhteensopivuusongelmat	4	3	12	Selvitetään miten pilvipalvelu toimii ja mitä sen käyttöön vaaditaan. Testataan käytännössä mahdollisimman alkuvaiheessa.
Lopputulos ei yritykselle toivottu	9	1	9	Projektisuunnitelman hyväksyttäminen yrityksellä ja jatkuva kommunikointi yrityksen kanssa.
Henkilöstöongelmat	5	2	10	Pysytään terveenä, sovitaan mahdollisista poissaoloista, jokainen ryhmän jäsen on vastuussa omasta ajankäytöstään.

## 5.4. Projektin toteuma

Projekti pystytettiin toteuttamaan annetulla budjetilla. Toteuma ylitti kuitenkin suunnitelman kahdellasadalla eurolla, johtuen varaosien, lisävirtamittarin ja muiden komponenttien hankinnasta. Tällainen arvion ylitys oli kuitenkin odotettavaa. Alla projektin toteuma.

Tavara	Määrä	Hinta	Yhteensä
Virtalähde sipyyyn	1	28,71€	28,71€
Releet	4	14,32€	57,28€
Virtalähde releille	1	22,37€	22,37€
Sipy	2	35,36€	70,72€
Sipyn laajennuslauta	1	20,55€	20,55€
Asennustarvikkeet	1	20,00€	20,00€
Virtamittarit 20A	4	48,85€	195,40€
Pistorasia	3	9,90€	29,70€
Operaatiovahvistimia 7kpl	1	14,91€	14,91€
Vedonpoistaja	8	1,27€	10,16€
Vedonpoistajan kiinnikke	10	0,39€	3,87€
Mikrokontrollerin kotelo	1	34,96€	34,96€
A/D muunnin	2	3,06€	6,12€
Piikkirima	10	0,27€	2,74€
Piirilevyt	10	1,73€	17,30€
Virtamittari 50A	1	72,71€	72,71€
<b>Yhteensä</b>			607,50€

## **6. Yhteenveto ja johtopäätökset**

Kaikki päätavoitteet saavutettiin aikataulussa, eikä budjettia ylitetty. Projektin lopputulokseen olivat tyytyväisiä koko ryhmä, sekä yrityksen edustajat. Prototyyppiä tehdessä, sekä sen valmistuttua, heräsi paljon uusia ideoita jatkokehittelyä varten.

Esimerkiksi prototyyppi vaatisi kehittelyä kuluttajamarkkinoita varten. Koska kuluttajaa laskutetaan vain päätötehosta, tulisi loisteho eliminoida mittauksissa. Tähän annoimme jo parannusehdotuksen ylempänä, missä virran lisäksi mitataan myös jännitettä.

Enston edustajien mielestä suunnittelemamme algoritmi oli hyvä ja sai miettimään sen ongelmatilanteita. Esimerkiksi kun kulutus on epätasaista, onko releiden nopea päälle ja pois kytkeminen järkevä ratkaisu? Valmiissa tuotteessa myös priorisoinnin tulisi muuttua ajan kuluessa, jotta kaikki kuormat olisivat tasaisesti päällä. Tämä ominaisuus oli tarkoitus sisällyttää ohjelmaan, mutta jätettiin tekemättä aikarajoitteiden vuoksi.

Edustajien mielestä myös sensori valintamme oli hyvä, sillä se säästää laskentatehoa. Prototyyppiä tullaan käyttämään valmiilla, sekä Enston omilla sovelluksilla tuotekehityksessä. Miettiessämme kuormien ohjauksen optimointia edustajat totesivat, että se vaatisi todennäköisesti kuormilta älyä.

Projekti vaati jokaiselta ryhmän jäseneltä uusia taitoja tai syventymistä jo opittuun. Projektin toteuttaminen vaati mm. tutustumista ohjelmointirajapintoihin, mistä kenelläkään ryhmän jäsenistä ei ollut kokemusta. Lisäksi prototyypin kehittäminen opetti kaikille projektityötaitoja, ja haastoi jokaisen ryhmäläisen ongelmanratkaisutaitoja, mm. laitteiston dokumentaation ollessa puutteellista. Näistä tilanteista selvittiin tarkalla ongelman määrittelyllä.

Kukaan ryhmän jäsenistä ei ole aikaisemmin toteuttanut käytännön sovellusta yrityksen kanssa yhteistyössä. Yritysyhteistyö vaatii projektin työstämistä myös asiakkaan näkökulmasta, mikä on varmasti työelämän kannalta tärkeä taito.



kuva 20. Ryhmä projektin valmistuttua



## ***Liitteet***

Gantin kartta (Excel)

## ***Linkit***

<https://github.com/riikkano/protopaja2018>

<http://protopaja.aalto.fi/protopaja-2018/ensto-2018/>

## **Lähteet**

<https://docs.pycom.io>