



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Julius Yli-Suomu

SÄHKÖPOTKULAUDAN MOOTTORIN OHJAIN

Tekniikka

2025

TIIVISTELMÄ

Tekijä	Julius Yli-Suomu
Opinnäytetyön nimi	Sähköpotkulaudan moottorin ohjain
Vuosi	2025
Kieli	suomi
Sivumäärä	45
Ohjaaja	Jani Ahvonen

Tämän opinnäytetyön tavoitteena oli ymmärtää harjattoman sähkömoottorin toiminta ja ohjausjärjestelmä sekä valmistaa sellainen ohjain sähköpotkulautaa varten. Se on hyvin monimutkainen toteuttaa, joten ohjain rakennettiin valmiiksi suunnitellusta VESC-pohjaisesta Cheap FOCer -piiristä.

Työssä käsiteltiin miten sähkömoottorin staattorin ja roottorin magneettikentät saavat aikaan liikkeen ja miten tehdään kommutointi, jolla saadaan aikaan pyörivä liike käyttäen kolmivaiheista moottoria. Työssä käsiteltiin myös, miksi ohjain tarvitsee tiedon moottorin asennosta, ja miten tieto saadaan anturien kanssa tai ilman sekä käytiin läpi BLDC-, siniaalto- ja FOC-ohjaustekniikat.

Ohjain valmistettiin tilaamalla piirilevyt ja komponentit ja juottamalla ne käsin kokoon. Juottaminen oli haasteellista erityisesti DRV8302- ja STM32-piireissä, joissa on paljon pieniä jalkoja, ilmeni runsaasti ongelmia piirilevyjen saamisessa toimimaan. Lopulta saatiin yksi ohjain toimimaan halutulla tavalla.

Työssä kävin läpi, miten ohjaimen asetukset säädetään VESC toolilla ja testasin miten anturiton ja anturillinen sekä BLDC- ja FOC-ohjaustekniikka eroavat käytännössä. Anturittomana liikkeellelähdössä ja pienessä vauhdissa jarruttaessa voi tulla hieman nykimistä, mikä poistuu käyttämällä hallantureita. FOC-ohjaustekniikkaa käyttäen moottorista tuli hiljaisempi ja siihen tuli lisää vääntöä verrattuna BLDC-ohjaustekniikkaan.

ABSTRACT

Author	Julius Yli-Suomu
Title	Motor Controller for an Electric Scooter
Year	2025
Language	Finnish
Pages	45
Name of Supervisor	Jani Ahvonen

The goal of this thesis was to understand the operation and control system of a brushless electric motor, and to make such a controller for an electric scooter. It is very complex to implement, so the controller was built from a pre-designed VESC-based circuit named Cheap FOCer.

Thesis discussed how magnetic fields of the stator and rotor of an electric motor create motion, and how commutation is performed to create rotational motion on a three-phase motor. The thesis discusses why the controller needs information about the motor's position, and how the information is obtained sensorlessly or with sensors, and reviewed BLDC, sine wave and FOC control techniques.

The controller was made by ordering circuit boards and components, and soldering them together by hand. Soldering was challenging, especially with the DRV8302 and STM32 chips, and there were many problems getting circuit boards to work. In the end, one controller worked as it should.

The controller was adjusted using the VESC tool and it was also tested how the sensorless and sensed, and the BLDC and FOC control techniques differ in practice. With sensorless, there can be a little jerking when starting to move or brake at low speeds, which does not happen when using hall sensors.

As result, the motor became quieter, and obtained more torque using the FOC control technique compared to the BLDC technique.

SISÄLLYS

TIIVISTELMÄ.....	2
ABSTRACT.....	3
1 JOHDANTO.....	7
1.1 Valmistettavan piirin valinta.....	7
1.2 Tekoälyn käyttö tässä opinnäytetyössä.....	8
2 HARJATTOMAN MOOTTORIN TOIMINTAPERIAATE.....	9
2.1 Kommutointi.....	10
2.2 Moottorin asentotiedot.....	11
2.3 Ohjaustekniikat.....	12
3 OHJAIMEN TOIMINTA.....	14
3.1 Lohkokaavio.....	14
3.2 Vuokaavio.....	15
4 OHJAIMEN SIMULOINTI.....	16
4.1 Virransyötön simulointi.....	16
4.2 Virran mittauksen simulointi.....	20
5 OHJAIMEN VALMISTUS.....	22
5.1 Piirilevy.....	22
5.2 Komponenttien juottaminen.....	23
5.3 Ohjelmointi.....	29
6 TESTAUS.....	32
6.1 Kytkenä sähköpotkulautaan.....	32
6.2 Asetuksien säätö.....	36
7 JOHTOPÄÄTÖKSET.....	43
LÄHTEET.....	44

KUVAT

Kuva 1. Harjattoman moottorin rakenne [4].....	9
Kuva 2. Puolisiltakytkentä 3-vaihemoottorille [5, s. 9].....	10
Kuva 3. Moottorin vaiheisiin syötettävä aaltomuoto [7, s. 7].....	12
Kuva 4. Moottorinohjaimen lohkokaavio.....	14
Kuva 5. Vuokaavio moottorinohjaimen ohjelmistosta.....	15
Kuva 6. Moottorin virransyöttö KiCadissä.....	16
Kuva 7. Jännitelähteet simulaatiota varten.....	17
Kuva 8. Vaihe 1 ja sen ohjaus simulaatiossa.....	18
Kuva 9. Vaiheiden jännitteet simulaatiossa.....	18
Kuva 10. Jännitejakajat 3 vaiheelle.....	19
Kuva 11. Vaiheen ja jännitejakajan jännite simulaatiossa.....	19
Kuva 12. Virransyöttö shunttivastuksien kanssa.....	20
Kuva 13. Moottorin vaiheen virta ja shunttivastuksen jännite simulaatiossa.....	21
Kuva 14. Piirilevy edestä ilman komponentteja.....	22
Kuva 15. Piirilevy takaa ilman komponentteja.....	22
Kuva 16. Ensimmäinen piirilevy suurin osa pintaliitoskomponenteista juotettuna.....	23
Kuva 17. Piirilevy DRV8302- ja STM32-piirit juotettuna.....	24
Kuva 18. Piirilevy takaa kiteen ja osan pintaliitoskondensaattoreista juottamisen jälkeen.....	24
Kuva 19. Piirilevy edestä pintaliitosvastuksia, -kondensaattoreita, ledit ja kela juotettuna.....	25
Kuva 20. Piirilevy takaa, kun osa pintaliitosvastuksista ja -diodit on juotettu.....	25
Kuva 21. Lisää pintaliitosvastuksia juotettuna.....	25
Kuva 22. Piirilevy takaa, kun lisää pintaliitosvastuksia ja 3,3v regulaattori on juotettu.....	26
Kuva 23. Läpijuotettavia kondensaattoreita ja liittimiä juotettuna.....	26

Kuva 24. Piirilevy MOSFETit, ohjelmointiliitin ja läpijuotettavat kondensaattorit juotettuna.....	27
Kuva 25. Läpijuotettava kondensaattori pintaliitoskondensaattorin paikalla.	27
Kuva 26. 220nF kondensaattorit juotettuna.....	28
Kuva 27. Toinen piirilevy kasattuna testauskuntoon.....	29
Kuva 28. Kolmas piirilevy testauskunnossa.....	29
Kuva 29. Nucleo kytkettynä piirilevyyn ohjelmointia varten.....	30
Kuva 30. Tiedosto conf_general.h muokattuna tälle piirille sopivaksi.....	31
Kuva 31. Moottorin vaiheet ja kaasukahva kytkettynä ohjaimeen.....	33
Kuva 32. Piirilevyn sisääntulojännitteen positiivinen puoli vahvistettuna.....	35
Kuva 33. Piirilevyn sisääntulojännitteen maa vahvistettuna.....	35
Kuva 34. ADC valittuna kaasukahvalle vesc toolissa.....	36
Kuva 35. Kaasun ohjaustyyppi valittuna.....	36
Kuva 36. Kaasukahvan mapping-asetukset säädettyinä.....	37
Kuva 37. Virran rajoituksen oletusasetukset.....	38
Kuva 38. BLDC-asetukset ja moottorin tunnistus.....	39
Kuva 39. FOC-asetukset moottorin tunnistuksen jälkeen.....	41

LYHENTEET

BEMF	Moottorin pyöriessä vaiheisiin indusoituva jännite, back electromotive force
FOC	Harjattoman moottorin ohjaustekniikka, Field oriented control.
MOSFET	Transistorityyppi, metal-oxide-semiconductor field-effect transistor
VEESC	Avoimen lähdekoodin BLDC-moottorinohjain, Vedder's Electronic Speed Controller

1 JOHDANTO

Sähköpotkulaudoissa käytetään nykyään yleensä harjatonta napamoottoria, joka sijaitsee pyörän keskellä. Sillä saadaan parempi hyötysuhde ja se vaatii vähemmän huoltoa kuin harjalliset tai voimansiirtoa käyttävät moottorit, sillä harjat ja voimansiirto lisäävät kitkaa ja ovat kuluvia osia. Huono puoli on se, että harjattoman moottorin ohjaus on monimutkaisempi toteuttaa. [1.]

Tämän opinnäytetyön tavoitteena on ymmärtää miten sähköpotkulaudan harjaton moottori ja sen ohjausjärjestelmä toimii ja valmistaa tällainen ohjausjärjestelmä. Piirin suunnittelu alusta alkaen on työlästä, joten päätettiin, että otetaan valmiiksi suunniteltu piiri ja rakennetaan sellainen.

1.1 Valmistettavan piirin valinta

Valitsin pohjaksi VESCiin perustuvan piirin, sillä siihen on saatavilla valmis ohjelmisto jossa on monipuoliset säätömahdollisuudet. VESC on Benjamin Vedderin suunnittelema ohjain, joka on lyhenne sanoista Vedder's Electronic Speed Controller. Tarkoituksena oli tehdä ohjain, jonka voin asentaa sähköpotkulautaan, jossa on 36 V nimellispännitteellinen akku eli akku, jossa on 10 3,6 V li-ion-akkukennoa sarjaan kytkettynä. Halusin myös vähän paremmat säätömahdollisuudet kuin kiinalaisissa ohjaimissa, joita olen aiemmin käyttänyt.

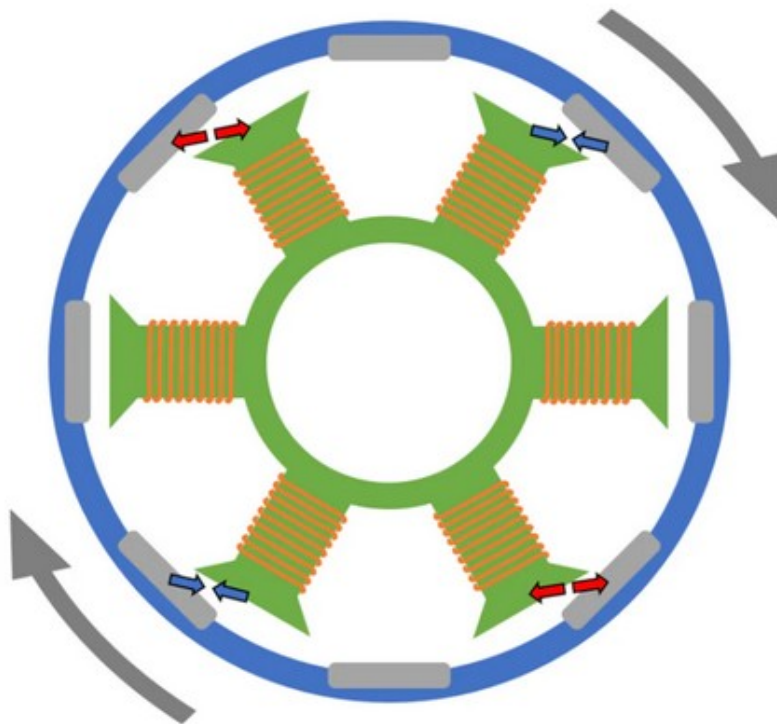
Lopulta päädyin valitsemaan piirin nimeltä Cheap FOCer, jonka nimi tulee siitä, että se on suunniteltu olemaan suhteellisen halpa, ja siinä on tuki FOC-ohjaustekniikalle. Ratkaisuuni vaikutti myös se, että komponentit on valittu siten, että ne eivät olisi kovin vaikea juottaa käsin, Cheap FOCerissa on vähemmän komponentteja kuin muissa VESC-pohjaisissa ohjaimissa. [2.]

1.2 Tekoälyn käyttö tässä opinnäytetyössä

Tässä opinnäytetyössä käytin Google Geminiä apuna tiivistelmän teossa siten, että kirjoitin tiivistelmän omin sanoin ottaen Google Geminin tekemästä tiivistelmästä esimerkkiä.

2 HARJATTOMAN MOOTTORIN TOIMINTAPERIAATE

Sähkömoottori koostuu staattorista ja roottorista. Staattorissa on käämit, ja kun niihin syötetään sähkövirtaa, syntyy magneettikenttä, joka hylkii tai vetää puoleensa roottorissa olevia kestopagneetteja riippuen magneettikentän napaisuudesta [3]. Yleensä harjattomissa moottoreissa käämit ovat 3:ssa eri vaiheessa ja käämeissä kulkevan virran suuntaa muuttamalla magneettikentän napaisuus muuttuu. Kuvassa 1 on piirros harjattoman moottorin rakenteesta [4]. Kuvassa ulompi ympyrä on roottori, ja sisempi on staattori. Siniset ja punaiset nuolet kuvaavat magneettien yhteenvetävää ja hylkivää voimaa.



Kuva 1. Harjattoman moottorin rakenne [4].

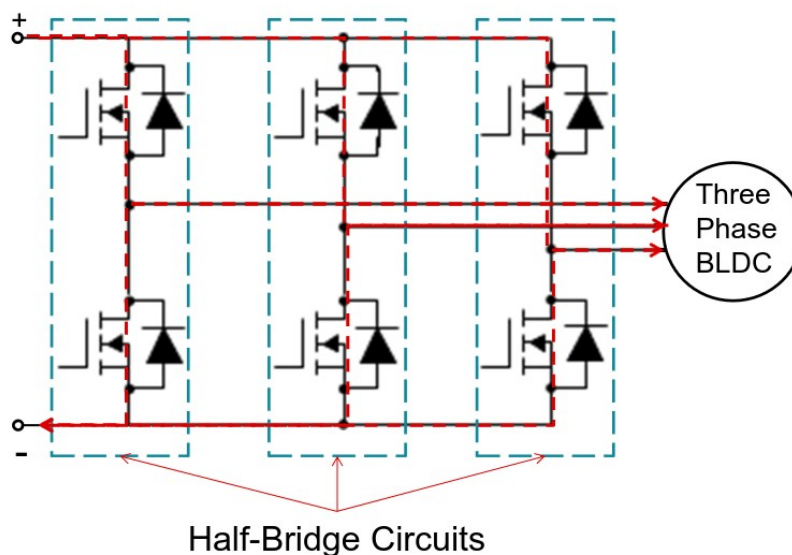
Kuva esittää moottoria, jossa on 8 magneettista napaa. Napojen määrä voi vaihdella riippuen moottorista. Napojen määrä on aina parillinen, koska joka pohjoisnavalle pitää olla myös etelänapa. Moottorin elektronisen kierrosluvun ja mekaanisen kierrosluvun suhde riippuu napojen määrästä.

Jakamalla elektroninen kierrosluku napaparien määrällä saadaan mekaaninen kierrosluku [5, s. 7]. Tämä tarkoittaa, että kuvan 1 moottorissa elektroninen kierrosluku on 4 kertaa isompi kuin mekaaninen kierrosluku.

2.1 Kommutointi

Moottorin kommutointi tarkoittaa sitä, miten virtaa syötetään eri vaiheisiin, että saadaan aikaan pyörivä liike [6, s. 13]. Perinteisissä harjallisissa moottoreissa tämä on toteutettu mekaanisella kommutaattorilla, joka pitää huolen siitä, että sähkövirta on aina oikean suuntainen moottorin pyörittämiseen, jolloin moottorille tarvitsee vain syöttää tasavirtaa. Harjattomissa moottoreissa se on toteutettu elektronisesti, moottorin ohjain syöttää virtaa moottorin vaiheisiin saadakseen aikaan magneettikentän. Pystyäkseen pyörittää moottoria tasaisesti erilaisilla kuormilla ja nopeuksilla, ohjain tarvitsee tiedon moottorin asennosta. Tästä lisää seuraavassa kappaleessa.

Yleensä ohjaimessa virran kulkua eri vaiheissa ohjataan kanavatransistoreista tehdyllä puolisolitytkennällä (half-bridge circuit), jollainen on erikseen joka vaiheelle. Yleensä nämä käytetyt transistorit ovat MOSFET-tyyppisiä. Kuvassa 2 on esimerkki tällaisesta kytkennästä 3-vaihemoottorille käyttäen 6 MOSFET-kanavatransistoria. [5, s. 9.]



Kuva 2. Puolisiltakytkentä 3-vaihemoottorille [5, s. 9].

Puolisiltakytkentää käytetään siten, että moottorin vaihe johdetaan syöttöjännitteeseen tai maahan. Kanavatransistoreiden ohjauksessa voidaan myös käyttää PWM-signaalia moottorin tehon säätämiseen.

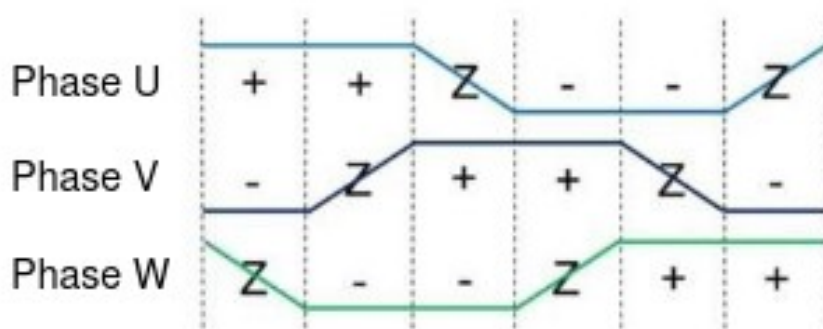
2.2 Moottorin asentotiedot

Moottorin asentotiedon selvittämiseen on monta eri tapaa, joita on anturillisia ja anturittomia. Anturina voidaan käyttää hall-antureita tai enkooderia. Anturittomasti asentotieto saadaan käyttämällä moottorin johtamattomaan vaiheeseen indusoituvaa jännitettä, josta käytetään nimitystä BEMF. Se voidaan myös laskea jos tiedetään moottorin resistanssi, induktanssi, virta ja jännite [7, s. 9].

Hall-anturit ovat magneettikenttää tunnistavia antureita. Kolmivaiheisessa moottorissa käytetään kolmea hall-anturia jotka ovat moottorissa 120 elektronista astetta erillään toisistaan [8, s. 395]. Hall-antureita on montaa eri tyyppiä. Unipolaarinen hall-anturi reagoi vain toiseen magneetin napaan, jonka ollessa lähellä sitä, se antaa ulos jännitteen, muulloin ei. Bipolaarinen hall-anturi antaa ulos jännitteen kun yksi magneetin napa on sen lähellä, ja se jää tähän tilaan kunnes magneetin toinen napa tulee sen lähelle. Omnipolaarinen taas antaa ulos jännitteen kumman tahansa magneetin navan ollessa sitä lähellä. Lineaarinen hall-anturi antaa jännitteen joka on sitä isompi, mitä lähempänä yksi magneetin napa on, ja sitä pienempi, mitä lähempänä toinen napa on. [9.] Sähköpotkulautojen moottoreissa käytetään unipolaarisia tai bipolaarisia hall-antureita, ja lineaarisia käytetään niiden kaasukahvoissa tai -vivoissa.

2.3 Ohjaustekniikat

Yksinkertaisin tekniikka BLDC-ohjaustekniikka, joka tuottaa moottorin vaiheisiin jännitteen, jonka aaltomuoto muistuttaa puolisuunnikasta. Tällä ohjaustavalla on 6 eri tilaa, joissa aina yksi vaihe on kytketty syöttöjännitteeseen, toinen maahan ja kolmas ei ole kytketty mihinkään. Jos käytetään anturiton ohjausta, kytkemättömältä vaiheelta mitataan BEMF, josta saadaan selville, milloin täytyy vaihtaa seuraavaan tilaan. Kuvassa 3 näytetään, millainen aaltomuoto moottorille syötetään. [7, s. 7.]



Kuva 3. Moottorin vaiheisiin syötettävä aaltomuoto [7, s. 7].

Kuvassa + tarkoittaa sitä, että kyseinen vaihe on kytketty syöttöjännitteeseen ja - että se on kytketty maahan. Z tarkoittaa, että sitä ei ole kytketty.

Toinen ohjaustekniikka antaa siniaallon muotoista virtaa moottorin vaiheisiin. Jännitteen voimakkuutta muutetaan moottorin asennon mukaan, moottorin väännöstä saadaan tasaisempi ja se on hiljaisempi. Se on monimutkaisempi toteuttaa, varsinkin jos halutaan tehdä se anturittomasti. Yksikään vaihe ei ole kytkemättämässä tilassa, jotta moottorin BEMF voitaisiin mitata. Silloin moottorin asento voidaan arvioida käyttäen käämien resistanssia, induktanssia, virtaa ja jännitettä. [7, s. 7.]

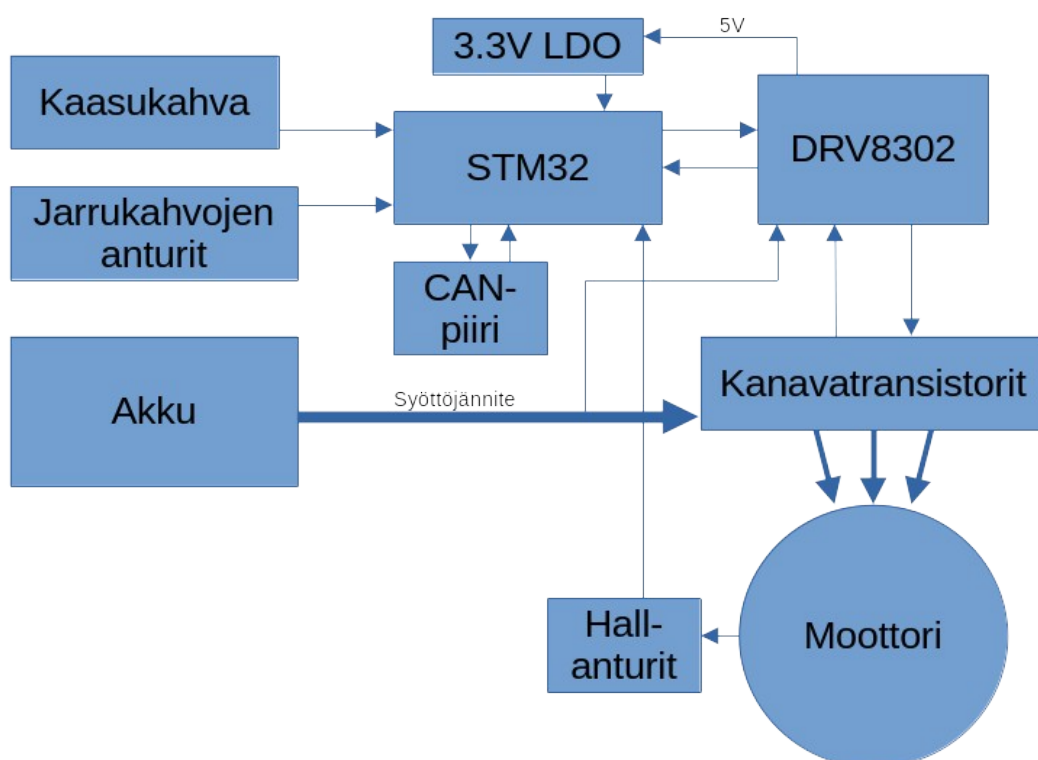
Kolmas ohjaustekniikka on FOC (Field-oriented Control), jossa moottoriin kulkevaa virtaa ohjataan siten, että saadaan sen suuntainen magneettikenttä,

jolla saadaan maksimi vääntö. Käytännössä tälläkin ohjaustekniikalla tulee siniaaltoa muistuttava aaltomuoto pyörittäessä moottoria. Tämä on kaikista monimutkaisin toteuttaa, ja se vaatii ohjaimelta eniten laskentatehoa. Tätä ohjaustekniikkaa voidaan myös käyttää silloin, jos moottorin halutaan pitää joku tietty asento, esimerkiksi roboteissa. [7, s. 7.]

3 OHJAIMEN TOIMINTA

3.1 Lohkokaavio

Kuvassa 4 on lohkokaavio moottorinohjaimen laitteistosta yksinkertaistettuna.

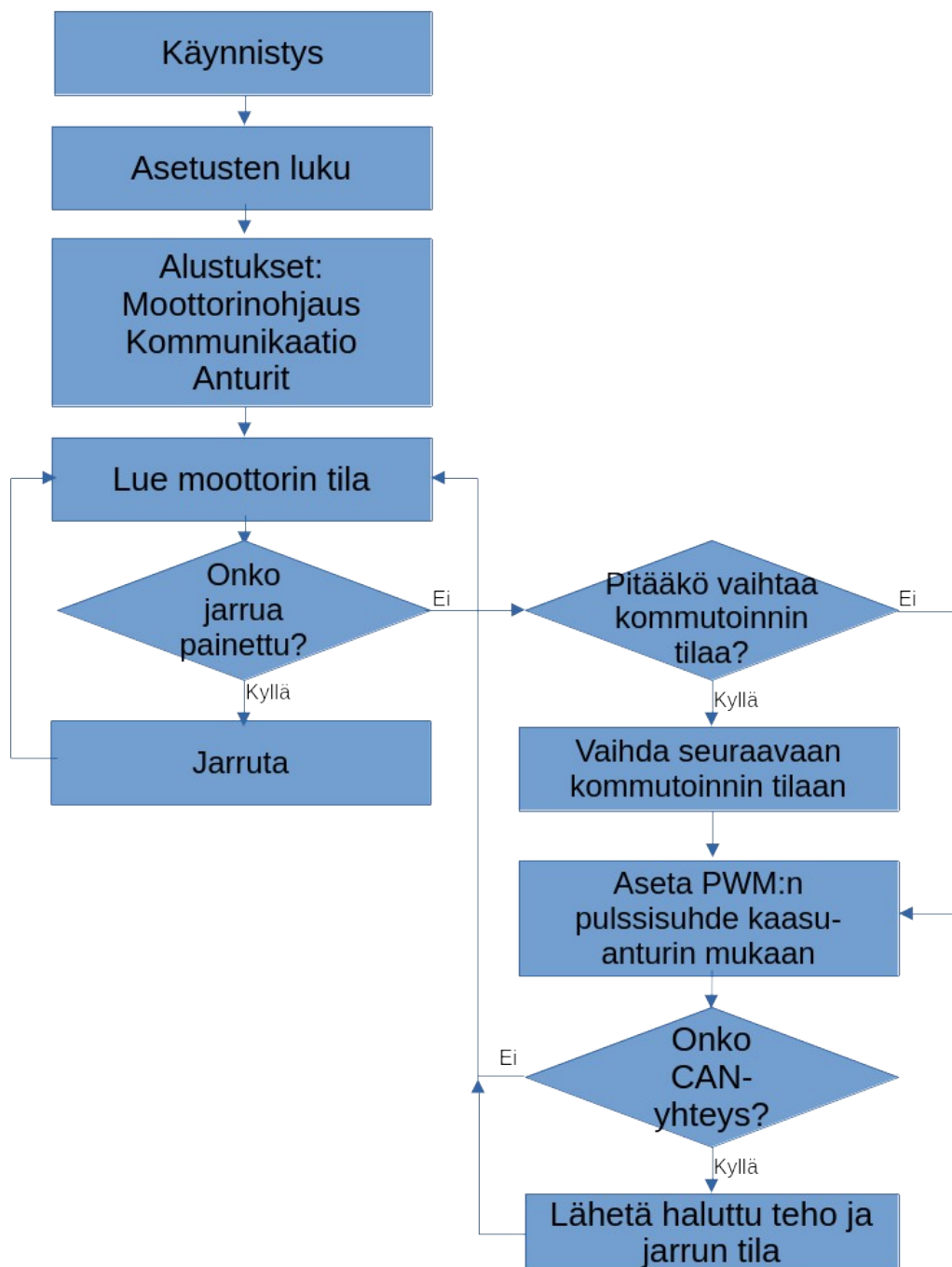


Kuva 4. Moottorinohjaimen lohkokaavio.

Kuvan 4 paksummat nuolet kuvaavat sitä, mistä kulkee moottorille syötettävä virta, eli suurin osa akulta otettavasta virrasta.

3.2 Vuokaavio

Kuvassa 5 on yksinkertaistettu vuokaavio ohjaimen toiminnasta.

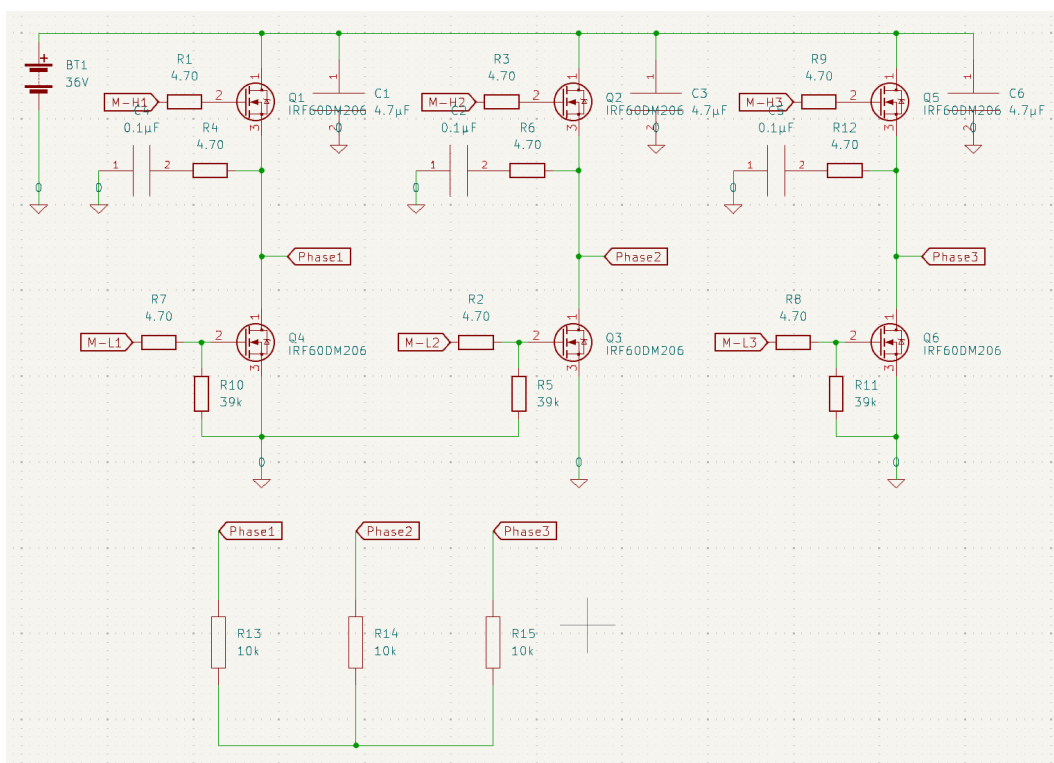


Kuva 5. Vuokaavio moottorinohjaimen ohjelmistosta.

4 OHJAIMEN SIMULOINTI

4.1 Virransyötön simulointi

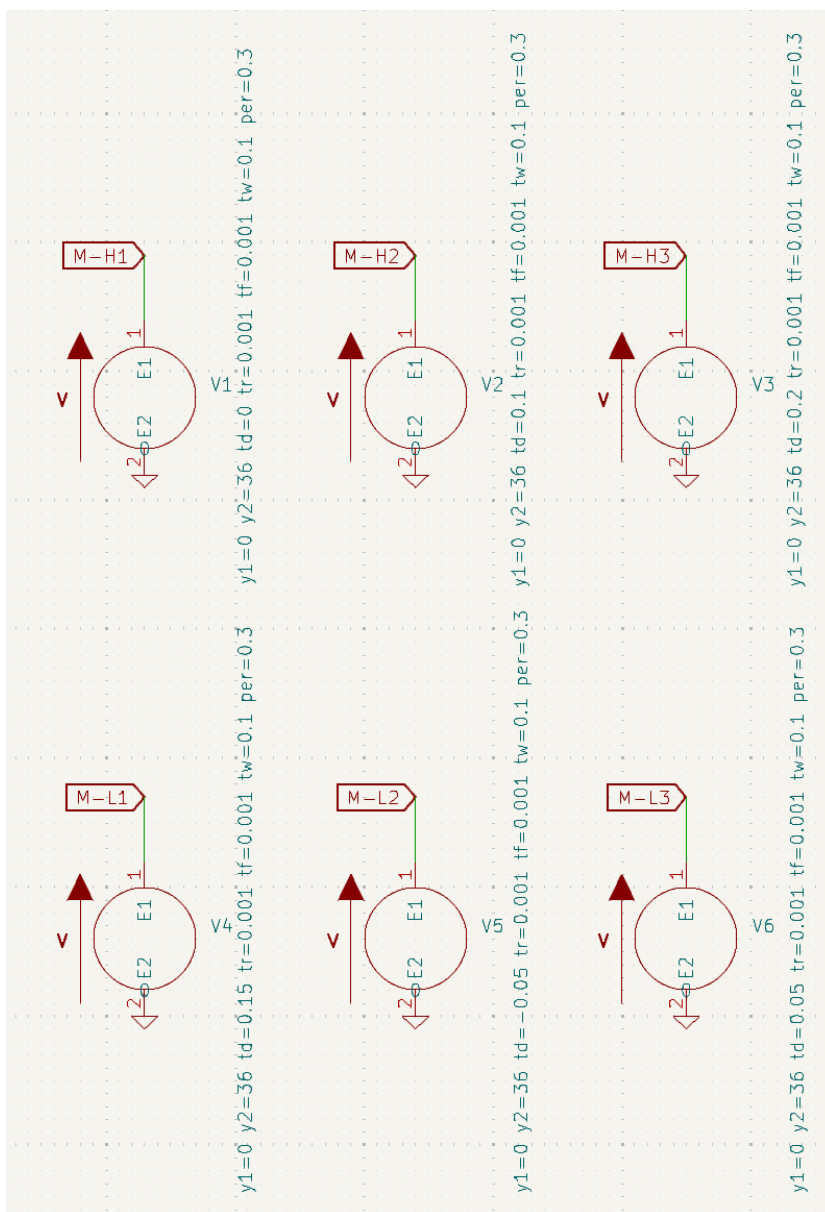
Simuloidakseni ohjaimen toimintaa piirsin vastaavan kytkennän kuin ohjaimen alkuperäisessä piirikaaviossa käyttäen KiCadia. Alkuperäinen piirikaavio on saatavilla ohjaimen GitHub-sivulla [2]. Kuvassa 6 on moottorin virransyötön simulointiin käytetty kytkentä.



Kuva 6. Moottorin virransyötö KiCadissä.

Simulaatiossa moottorin paikalle on kytketty 3 vastusta. Kanavatransistorien ohjaukseen simulaatiossa käytettiin 6 jännitelähdettä, joista jokaiseen kanavatransistoriin kytkettiin 1. Puolisuunnikkaan muotoisen ohjaustekniikan simulointiin määritettiin ne antamaan 0,1 sekunnin pituisia pulsseja 0,3 sekunnin jaksoajalla. Tällöin 0,1 sekuntia vastaa 120 astetta. Jos vaihe kytketään syöttöjännitteeseen 0 sekunnin kohdalla, vaiheen maahan

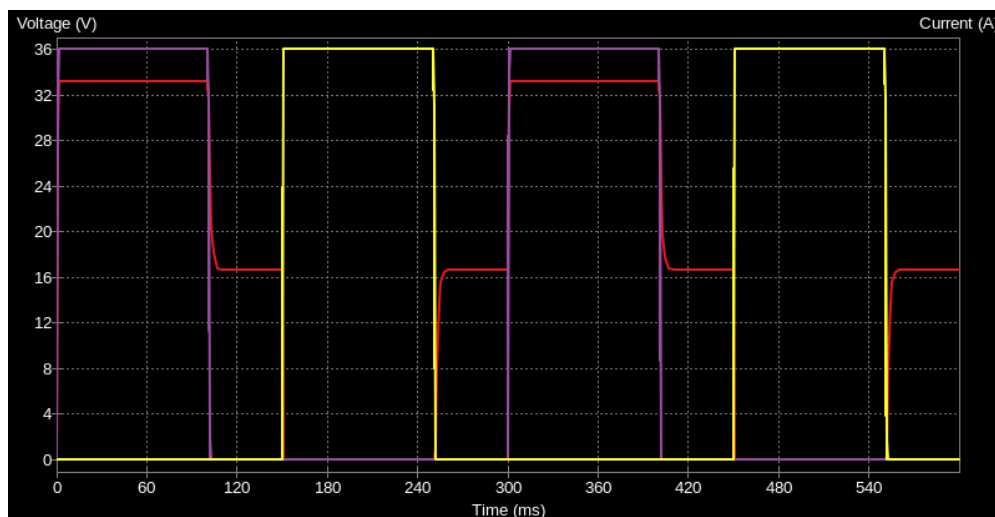
kytkävän transistorin täytyy alkaa johtaa 0,15 sekuntia myöhemmin. Seuraava vaihe taas kytketään syöttöjännitteeseen 0,1 sekuntia myöhemmin, ja toistamalla nämä vaiheelle, saadaan oikein ajoitetut pulssit. Kuvassa 7 on niiden jännitelähteiden kytkentä.



Kuva 7. Jännitelähteet simulaatiota varten.

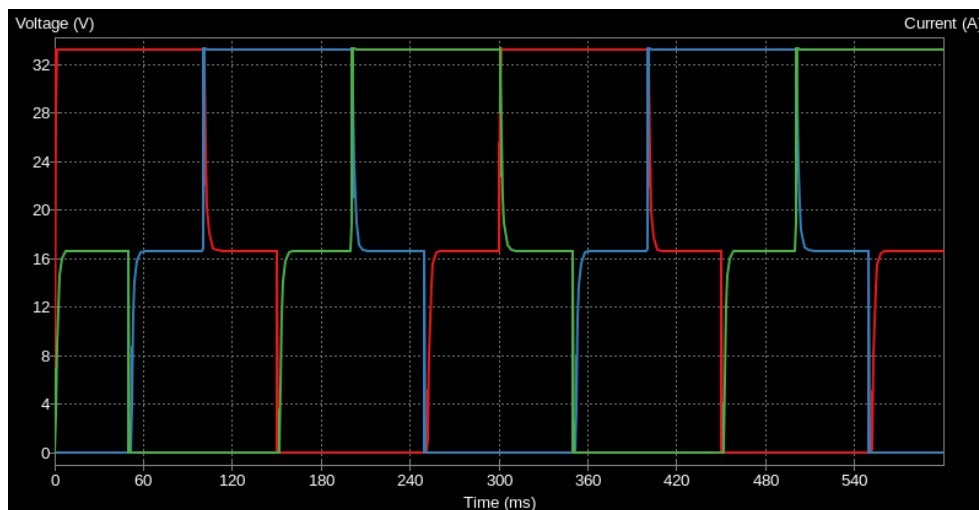
Kun simulaatiosta valitsi näkyville 1 vaiheen transistorien ohjauksen ja vaiheeseen tulevan jännitteen, tästä saatiin kuva 8, jossa näkyy kuinka

vaiheen jännite nousee ylös kun ylemmälle transistorille syötetään jännite, ja laskee alas kun alemmalle transistorille syötetään jännite.



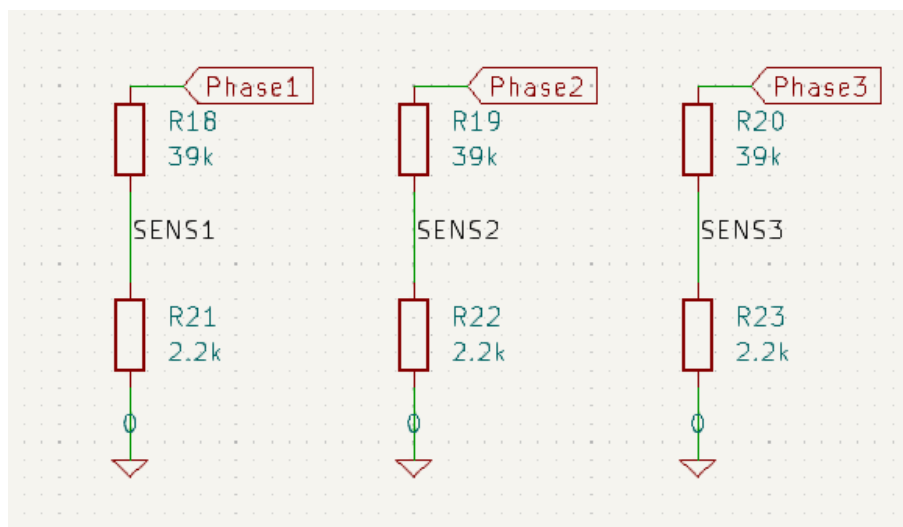
Kuva 8. Vaihe 1 ja sen ohjaus simulaatiossa.

Kun valitsi näkyville pelkästään kaikkien vaiheiden jännitteet, siitä saatiin kuva 9.



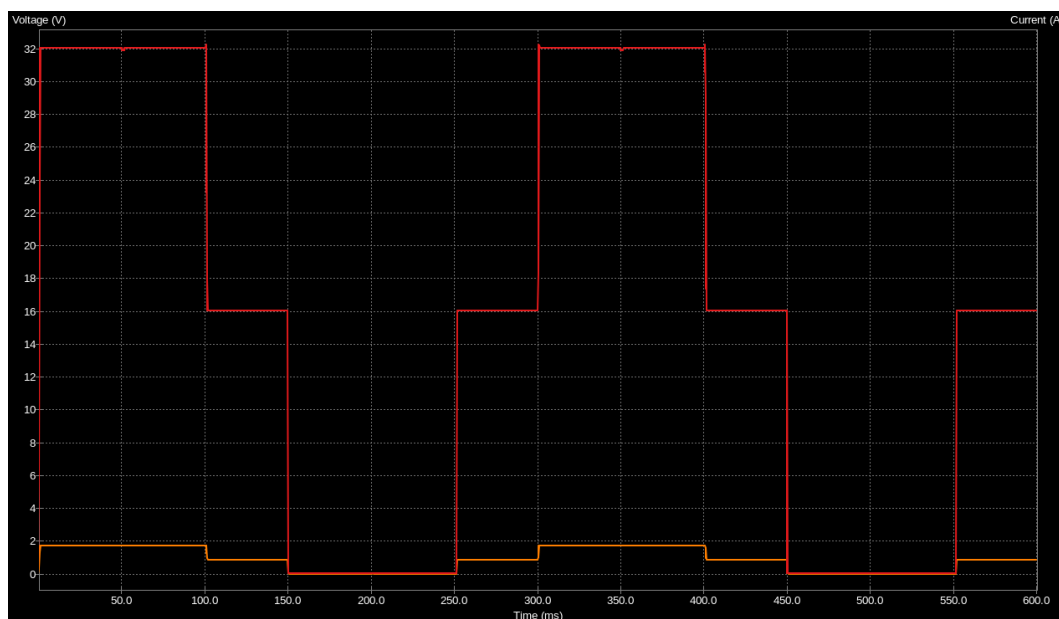
Kuva 9. Vaiheiden jännitteet simulaatiossa.

Kytkentään lisättiin myös joka vaiheeseen jännitejakaja, jossa on 39 ja 2,2 kilo-ohmin vastukset. Tämä kytkentä näkyy kuvassa 10.



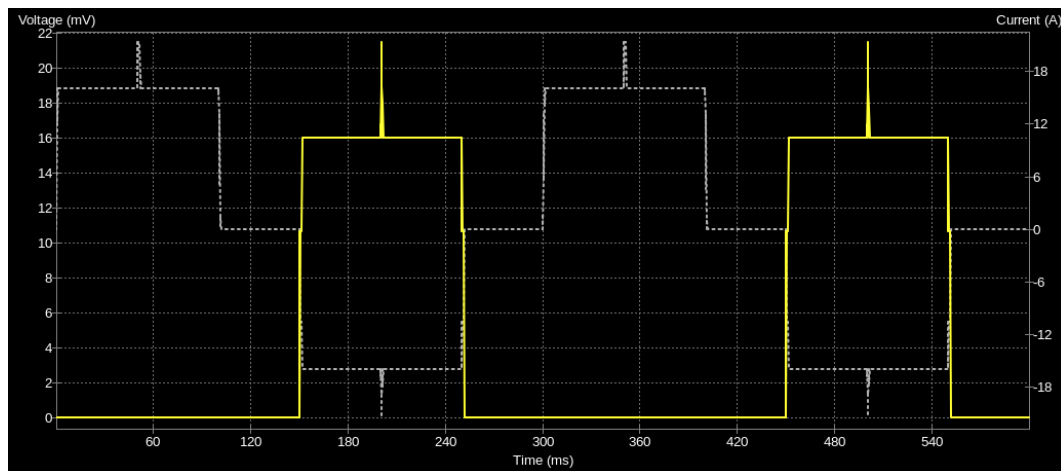
Kuva 10. Jännitejakajat 3 vaiheelle.

Simuloimalla tämän jälkeen kytkentä ja laittamalla näkyviin vaiheen ja siihen kytketyn jännitejakajan jännitteen näkyville, saatiin kuva 11.



Kuva 11. Vaiheen ja jännitejakajan jännite simulaatiossa.

Kuvasta 11 näkyy, miten jännitejakajan jännite muuttuu vaiheen jännitteen mukaan.



Kuva 13. Moottorin vaiheen virta ja shunttivastuksen jännite simulaatiossa.

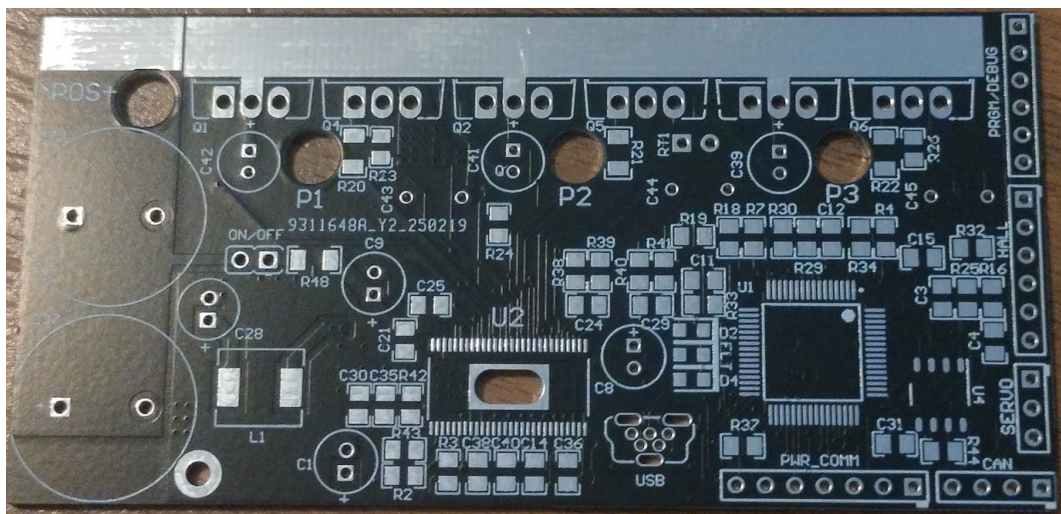
Kuvasta 13 näkyy, että kun vaihe johdetaan maahan, shunttivastuksen yli oleva jännite (keltainen) muuttuu suhteessa vaiheen virran suuruuteen.

5 OHJAIMEN VALMISTUS

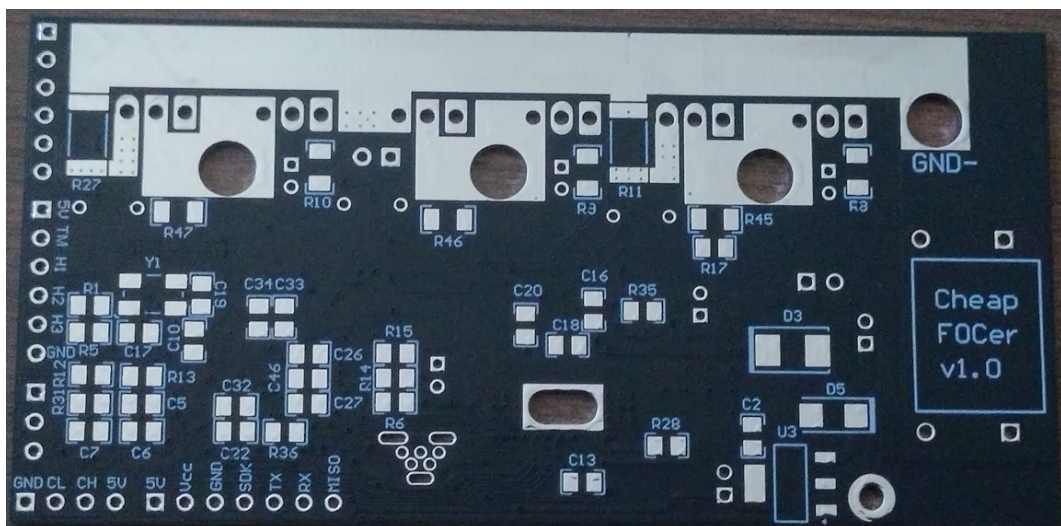
5.1 Piirilevy

Valmistus aloitettiin tilaamalla piirilevyjä JLCPCB:ltä. Minimimäärä tilaukseen oli 5 kappaletta, joten otin sen. Tilasin komponentteja useasta eri paikasta, sillä kaikkia komponentteja ei ollut saatavilla samasta paikasta. Päätin tilata komponentteja päätin tilata sellaisen määrän, että niillä voi kasata vähintään 3 piirilevyä. Osassa minimimäärä oli kuitenkin niin suuri, että niitä jää yli vaikka kasaisi kaikki 5 piirilevyä.

Parin viikon kuluttua tilauksesta sain valmiit piirilevyt. Kuvissa 14 ja 15 on piirilevyt ennen komponenttien juottamista.



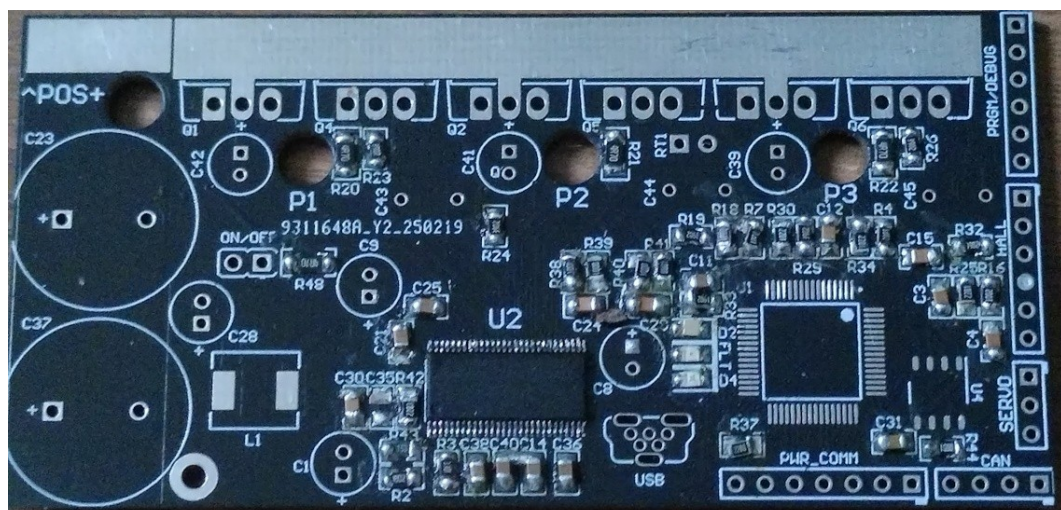
Kuva 14. Piirilevy edestä ilman komponentteja.



Kuva 15. Piirilevy takaa ilman komponentteja.

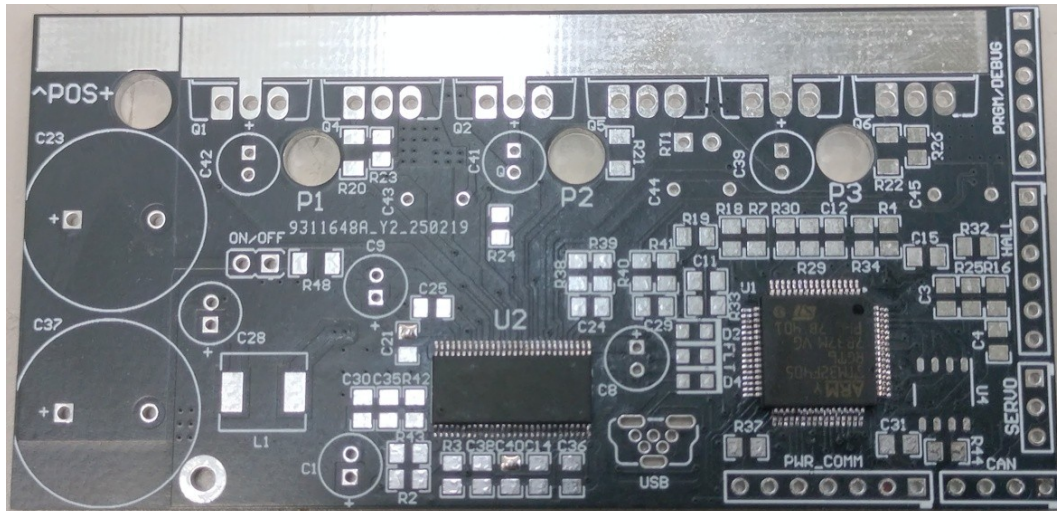
5.2 Komponenttien juottaminen

Kun suurin osa komponenteista oli saapunut, päätin kokeilla, miten kasaus onnistuu omilla juotosvälineillä, jotka eivät ole kovin hyviä tällaisten pienten komponenttien juottamiseen. Pintaliitosvastusten ja -kondensaattorien juottaminen alkoi sujua pienen harjoittelun jälkeen. Erityisen haastavaksi osoittautui DRV8302-piiri, jossa on paljon pieniä jalkoja. Sen alapuoli pitää myös juottaa kiinni, piirilevyssä on reikä sen takana helpottamassa tätä. Juottamisessa kävi siten, että joitakin jalkoja vääntyi. Kuvassa 16 on ensimmäinen piirilevy sen jälkeen kun suurin osa pintaliitoskomponenteista oli juotettu.



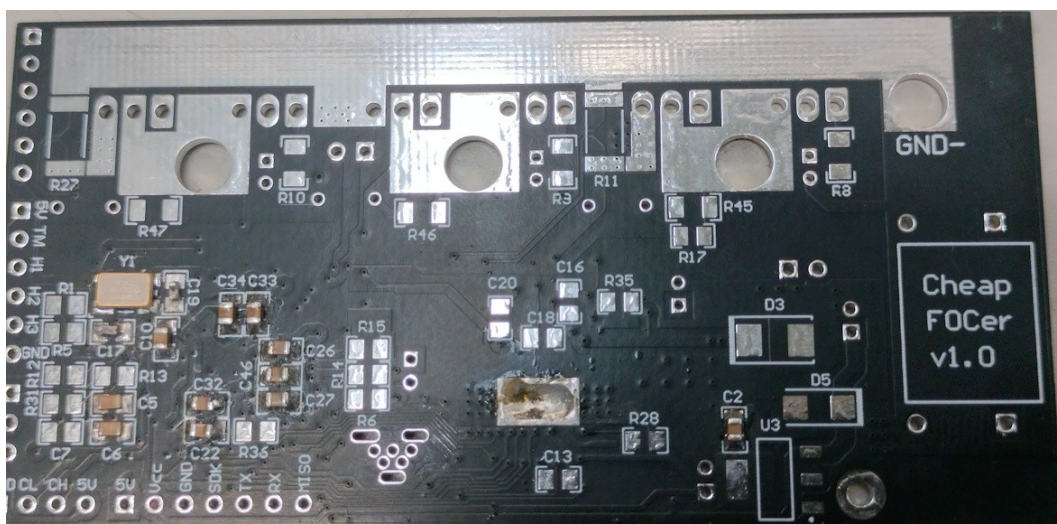
Kuva 16. Ensimmäinen piirilevy suurin osa pintaliitoskomponenteista juotettuna.

Seuraavaksi menin opetus- ja tutkimuslaboratorio Technobotnialle juottamaan piirilevyjä. Siellä juottaminen onnistui paremmin käyttäen pienempikärkistä kolvia ja ohuempaa juotoslankaa. Tällä kertaa päätin juottaa DRV8302 ja STM32-piirit ensimmäisenä, kuvassa 17 on piirilevy juottamisen jälkeen.

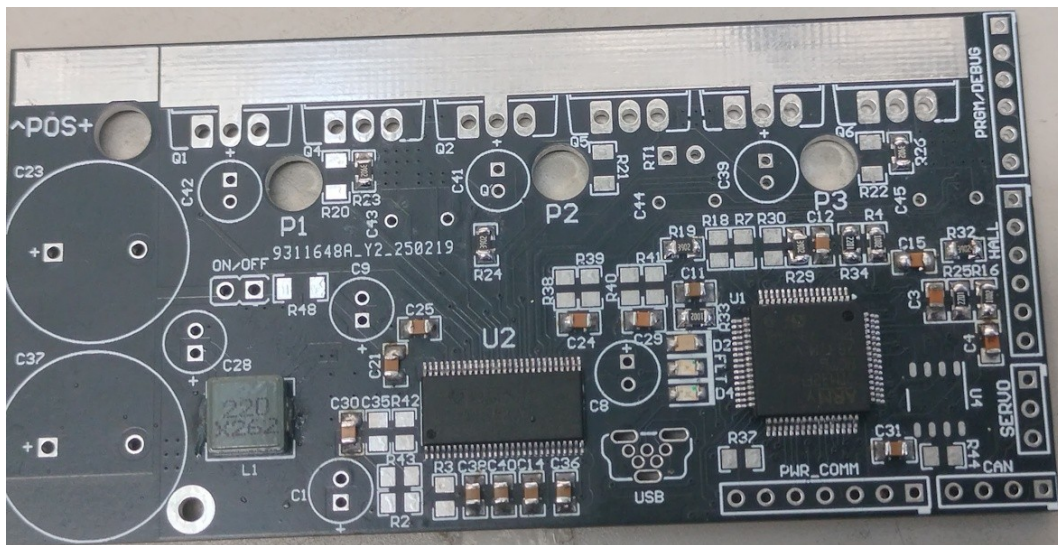


Kuva 17. Piirilevy DRV8302- ja STM32-piirit juotettuna.

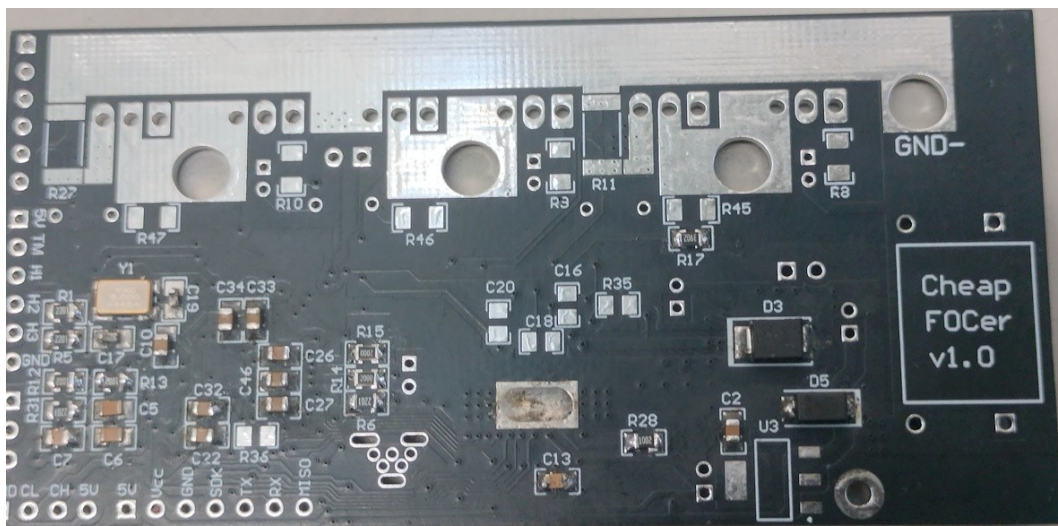
Sen jälkeen juotin kiteen, pintaliitoskondensattoreita, -vastuksia, diodeja, ledit ja kelan piirilevylle. Kuvissa 18-22 näkyy juottamisen eri vaiheita.



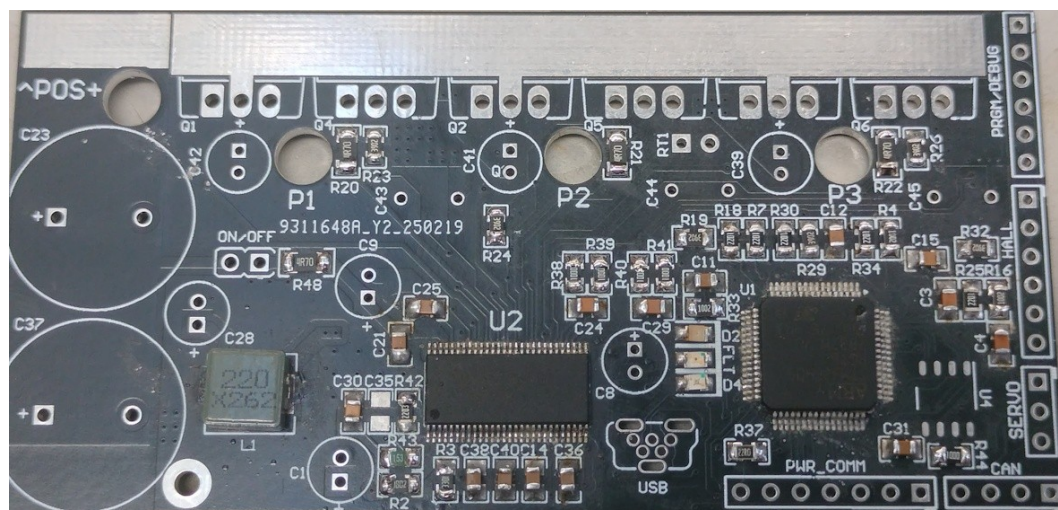
Kuva 18. Piirilevy takaa kiteen ja osan pintaliitoskondensattoreista juottamisen jälkeen.



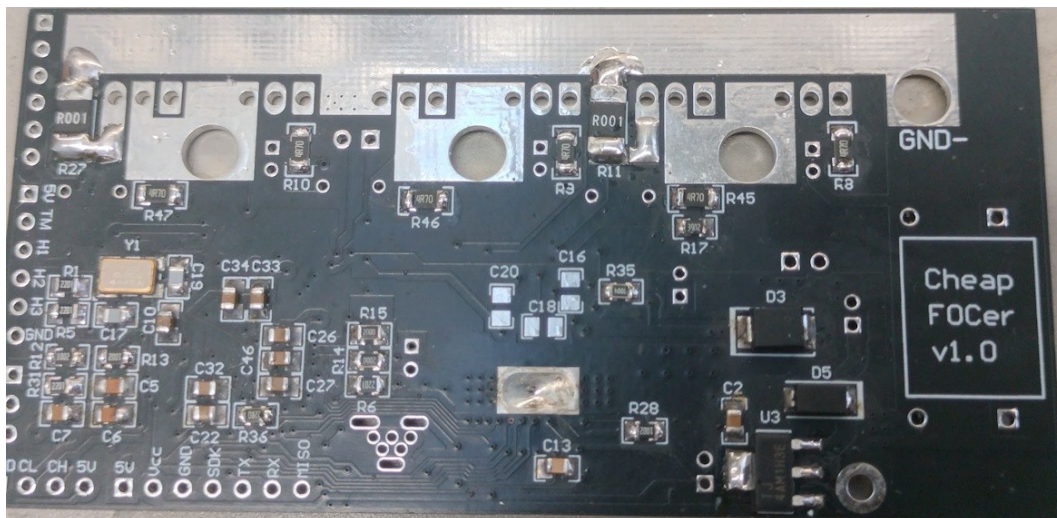
Kuva 19. Piirilevy edestä pintaliitosvastuksia, -kondensaattoreita, ledit ja kela juotettuna.



Kuva 20. Piirilevy takaa, kun osa pintaliitosvastuksista ja -diodit on juotettu.

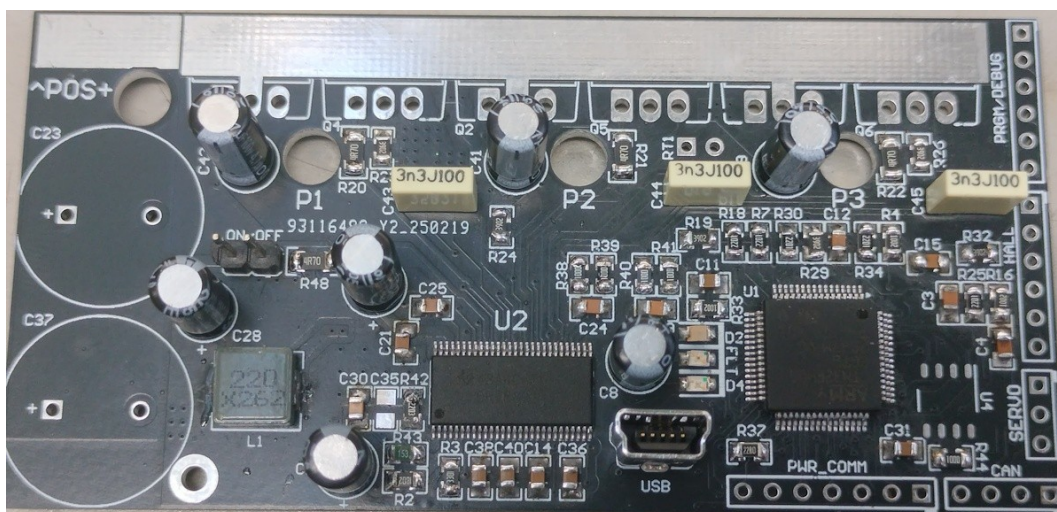


Kuva 21. Lisää pintaliitosvastuksia juotettuna.



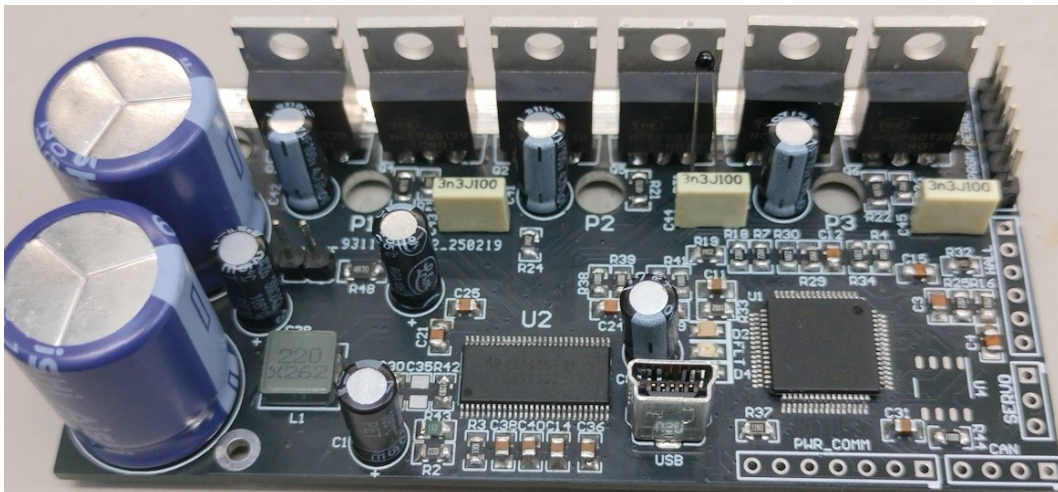
Kuva 22. Piirilevy takaa, kun lisää pintaliitosvastuksia ja 3,3v regulaattori on juotettu.

Seuraavaksi aloin juottaa läpijuotettavia kondenssaattoreita, liittimen on-off kytkimelle ja miniUSB-liittimen, jotka näkyvät kuvassa 23.



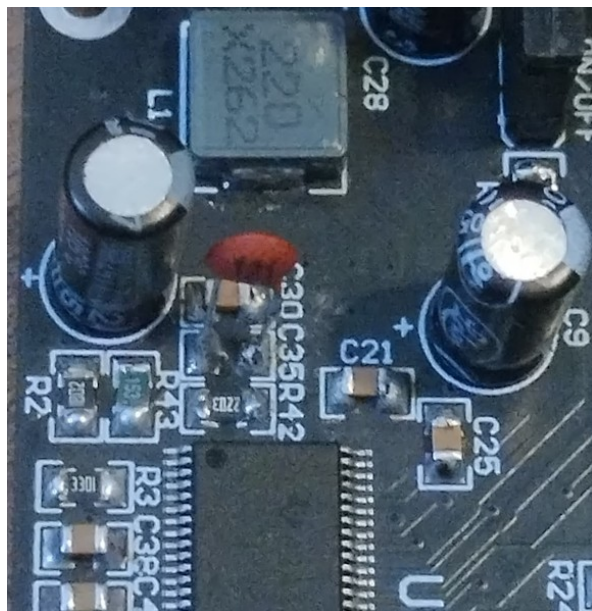
Kuva 23. Läpijuotettavia kondenssaattoreita ja liittimiä juotettuna.

Sitten juotin MOSFETit, loput 2 läpijuotettavaa kondenssaattoria ja ohjelmointiliittimen. Tässä vaiheessa jätin vielä loput liittimet piirilevyn reunalla pois, etteivät ne ole tiellä, jos siihen juottaa CAN-piirin, jota minulla ei vielä ollut. Kuvassa 24 on piirilevy tässä vaiheessa.



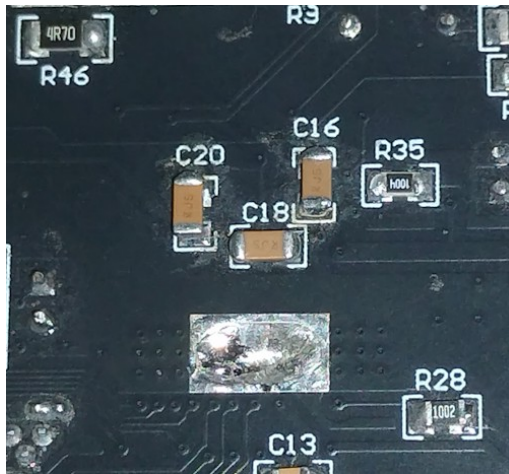
Kuva 24. Piirilevy MOSFETit, ohjelmointiliitin ja läpijuotettavat kondensaattorit juotettuna.

Tämän jälkeen päätin hakea StarEleciltä läpijuotettavan 120pF kondensaattorin pintaliitoskondensaattorin tilalle, ja 1206-kokoisia pintaliitoskondensaattoreita 0805-kokoisten tilalle, että pääsen kokeilemaan piiriä ilman oikeanlaisten komponenttien odottamista. Kuvassa 25 näkyy 120pF läpijuotettava kondensaattori juotettuna pintaliitoskondensaattorin paikalle.



Kuva 25. Läpijuotettava kondensaattori pintaliitoskondensaattorin paikalla.

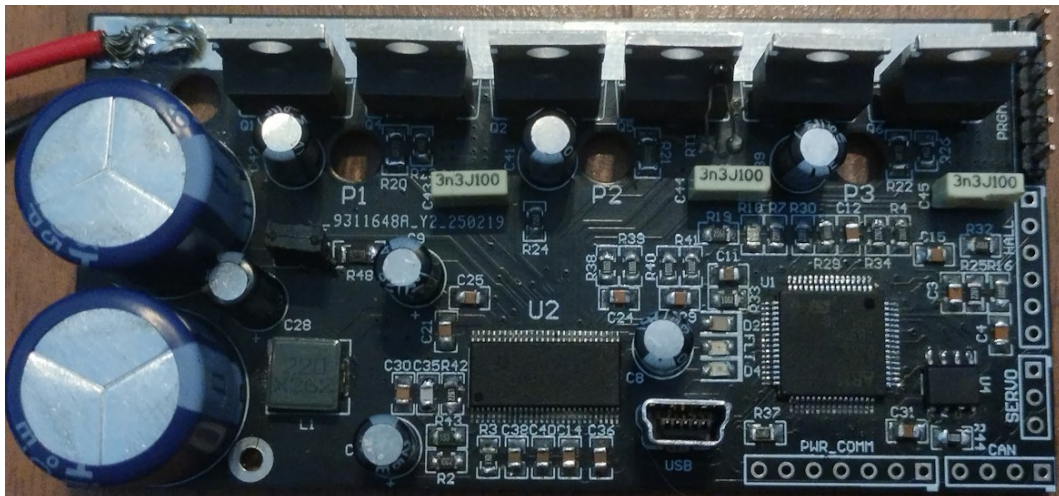
Kuvassa 26 näkyy kolme 220nF 1206 kondensaattoria juotettuna 0805 paikalle.



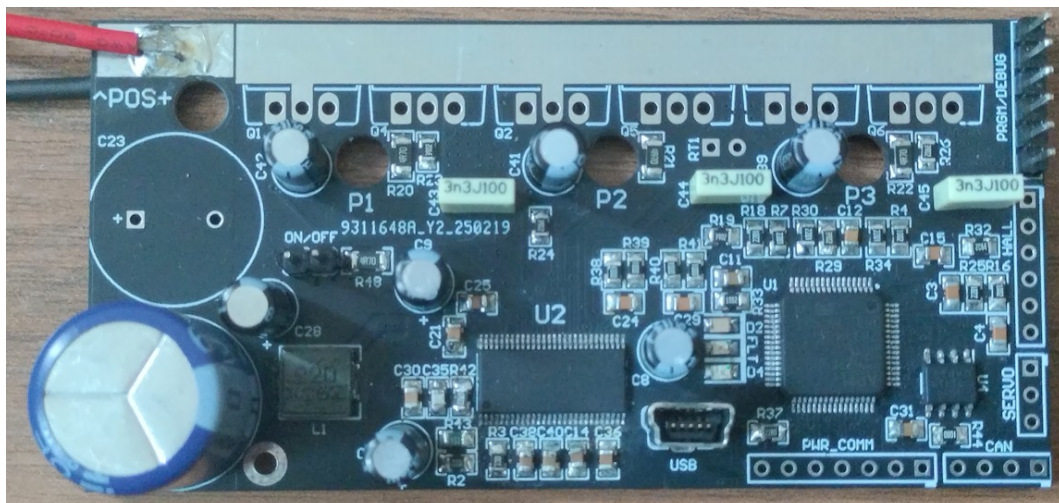
Kuva 26. 220nF kondensaattorit juotettuna.

Kondensaattorit mahtuivat paikalleen, vaikka niiden paikat piirilevyllä on tarkoitettu pienemmälle 0805 koolle. Tämän jälkeen juotin piirille virtakaapelit ja testasin kytkemistä 12v muuntajaan. Ilmeni ongelma, että ledi, joka näyttää, onko virta päällä, ei pysynyt koko ajan päällä. Osan ajasta se vilkkui tai ei palanut ollenkaan. Selvisi, että DRV8302-piirissä oli huono juotos, ja vika korjaantui juottamalla se uusiksi. Seuraavaksi kokeilin ohjelmoida tämän kasatun piirin, siitä lisää luvussa 5.3.

Myöhemmin kasasin toisen ja kolmannen piirilevyn testauskuntoon. Kolmannen piirilevyn kohdalla päätin kokeilla sitä, että MOSFETeja ja toista 680 μ F kondensaattoria ei ole vielä juotettu. Tällöin ei tietenkään voi vielä kokeilla, pyörittääkö se moottoria, mutta voin kokeilla ohjelmoida ja yhdistää piirilevyn tietokoneeseen. Sen jälkeen piirilevyt olivat kuvan 27 ja 28 mukaiset.



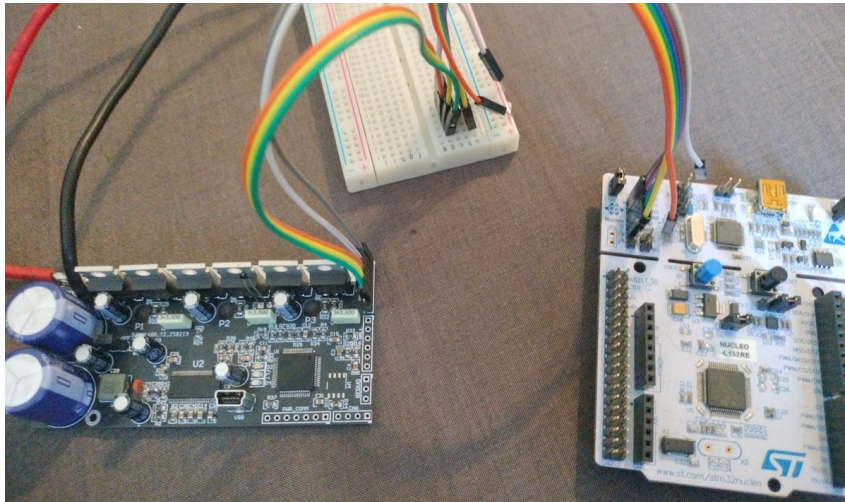
Kuva 27. Toinen piirilevy kasattuna testauskuntoon.



Kuva 28. Kolmas piirilevy testauskunnossa.

5.3 Ohjelmointi

Ohjelmoin piiriin käyttäen nucleo-L152RE-piiriä, jossa on ST-link samassa piirilevyssä. Sitä voi käyttää muiden piirilevyjen, joissa on STM32 ohjelmointiin. Nucleosta otetaan irti 2 jumpperia ohjelmointiliittimen vierestä, ja kytketään johdot sen ohjelmointiliittimestä ohjelmoitavan piirin ohjelmointiliittimeen. Kuvassa 29 on nucleo kytkettynä piirilevyyn.



Kuva 29. Nucleo kytkettynä piirilevyyn ohjelmointia varten.

Kääntääkseen piirille olevan koodin tarvitaan sulautetulle arm-prosessorille sopiva gcc-kääntäjä. Käännetyn ohjelman lähettämiseen piirille on käytössä OpenOCD. Ensiksi latsin piirin käynnistyslataajan lähdekoodin komennolla `git clone https://github.com/vedderb/bldc-bootloader.git`. Sen kääntäminen ja lähetys piirille onnistui komennolla `make upload`.

Sitten vuorossa oli itse moottorinohjaimen koodin kääntö ja lähetys. Sen latsin komennolla `git clone https://github.com/vedderb/bldc.git`, ja vaihdoin versioon 3.48 komennolla `git checkout 3.48`, koska tällä piirillä on kerrottu olevan ongelmia uudemmilla versioilla. Tämän jälkeen piti käydä muokkaamassa tiedostoa nimeltä `conf_general.h`, josta vaihdoin piirille oikeat `#define HW_SOURCE` ja `HW_HEADER` -rivit käyttöön, jotka ovat nimellä `hw_410.c` ja `hw_410.h`, ja poistin käytöstä rivit, joissa on `hw_60`. Tämän jälkeen rivit tiedostossa olivat kuvan 30 mukaiset.

```
//#define HW_SOURCE "hw_48.c"
//#define HW_HEADER "hw_48.h"

//#define HW_SOURCE "hw_49.c"
//#define HW_HEADER "hw_49.h"

#define HW_SOURCE "hw_410.c" // Also for 4.11 and 4.12
#define HW_HEADER "hw_410.h" // Also for 4.11 and 4.12

//#define HW_SOURCE "hw_60.c"
//#define HW_HEADER "hw_60.h"

//#define HW_SOURCE "hw_r2.c"
//#define HW_HEADER "hw_r2.h"
```

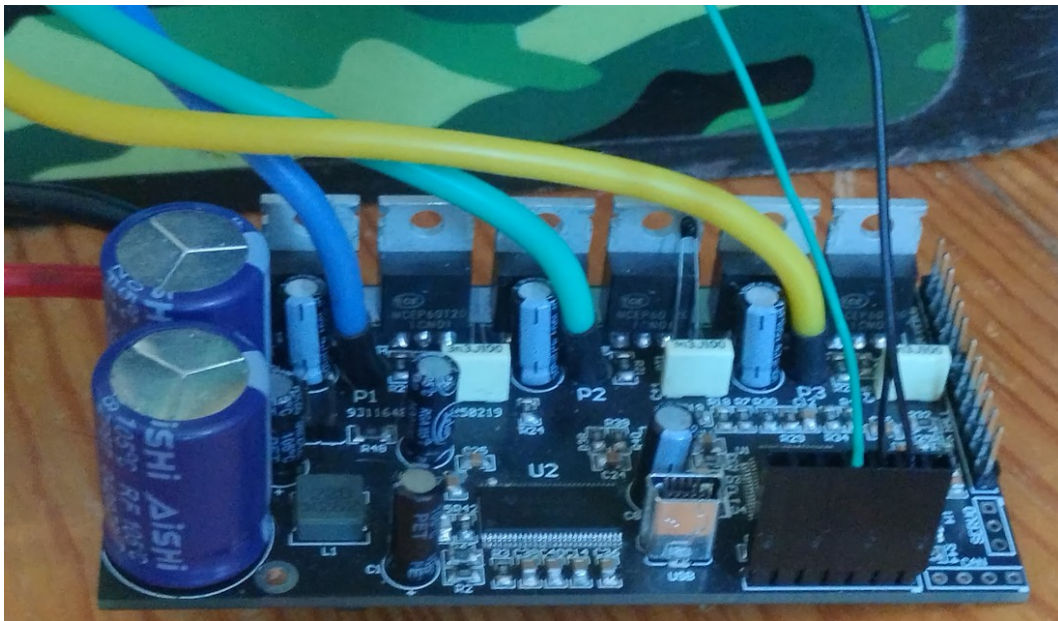
Kuva 30. Tiedosto `conf_general.h` muokattuna tälle piirille sopivaksi.

Sitten koodi voitiin kääntää ja lähettää piirille komennolla `make upload`. Sen onnistuttua kokeilin yhdistää piirin USB:llä tietokoneeseen ja yhdistää piiriin VESC toolilla. Se on ohjelma, jolla voidaan säätää tämän moottorinohjaimen asetuksia, mutta se ei tunnistanut piiriä. Epäilin tämän johtuvan siitä, että piirilevyllä ei ole can-piiriä, joten kokeilin poistaa sen käytöstä koodista. Tämän tein muuttamalla rivin, jossa on `#define CAN_ENABLE 1` kommentiksi. Tämän jälkeen kääntäminen antoi virhettä siitä, että yritettiin käyttää määrittelemättömiä muuttujia, jotka ovat koodissa siten, että ne määritetään vain, jos `CAN_ENABLE` on määritetty. Korjasin sen lisäämällä funktioihin, jotka antavat virheen alkuun `#if CAN_ENABLE`, ja loppuun rivit, joissa on `#else, return 0;` ja `#endif`, joka tekee sen, että jos `CAN_ENABLE` ei ole määritelty, kääntäjä jättää funktion sisälle ainoastaan `return 0;`. Silloin ne eivät enää anna virhettä. Tämän jälkeen kun koodin käänsi ja lähetti piirilevyllä, VESC tool tunnisti sen.

6 TESTAUS

6.1 Kytkeä sähköpotkulautaan

Kokeilin kytkeä piirin moottoriin, joten juotin piiriin kiinni 3 johtoa, jotka kytketään moottorin vaiheisiin. Yritin ajaa moottorin tunnistusta vesc toolista, mutta mitään ei tapahtunut. Sitten huomasin, että STM32-piirin jalkojen 63 ja 64 välissä näkyi punainen hehku. Pinniin 64 tulee 3,3v, ja pinniin 63 tulee maa, niiden väliin oli tullut oikosulku. Yritin kokeilla sulattaa niissä olevaa tinaa kolvilla ja varmistaa, ettei tina yhdistä niitä, mutta oikosulku ei lähtenyt pois. Kun kokeilin vielä kytkeä virrat, huomasin, että punainen hehku näyttää tulevan jostain pinnien takaa. Irrotin STM32-piirin piirilevytä, puhdistin piirilevyn, ja mittasin, että siinä ei ole oikosulku. Sen jälkeen juotin STM32-piiriin takaisin. Tällä sain oikosulun hetkeksi pois, mutta se ilmestyi takaisin testaillessani 36 voltilla. Tällä kertaa havaitsin, että itse STM32-piirin sisällä oli oikosulku, joten ei auttanut muu kuin vaihtaa se uuteen. Tämän jälkeen piiri toimi sen verran, että se yhdistää tietokoneeseen, eikä mennyt enää oikosulkuun testattaessa 36v akulla. Moottoria se ei alkanut vielääkään pyörittää. Päätin vielä kokeilla kytkeä kaasukahvan piirilevyyn, jonka kytkin PWR_COMM-liittimen 3,3v, maa, ja SCK_ADC_EXT-pinneihin. Kuvassa 31 on kaasukahva ja moottorin vaiheet kytkettynä piirilevyyn.



Kuva 31. Moottorin vaiheet ja kaasukahva kytkettynä ohjaimen.

Vaikka olin aiemmin käyttänyt kaasukahvaa 5v jännitteellä, se näkyi toimivan oikein myös 3,3v jännitteellä, sillä tällä kytkennällä kaasukahvan signaali tuli oikein VESC Tooliin. Jännite vaihteli noin 0,8 ja 2,6 voltin välillä riippuen kaasukahvan asennosta. Käytin siihen 3,3v siksi, että STM32-piirin sisääntulot eivät kestä 5v jännitettä [10].

Seuraavaksi kasasin ja kokeilin toista piirilevyä. Ohjelmointi onnistui, minkä jälkeen tälläkin kävi niin, että 3,3v meni oikosulkuun. Tällä kertaa STM32-piiri lämpeni selvästi. Mittailtuani komponentteja huomasin, että yhdessä kondensaattorissa oli oikosulku. Vaihdettuani sen uuteen oikosulku oli poissa, mutta tämän jälkeen STM32-piiri ei vaikuttanut enää toimivan, joten vaihdoin senkin uuteen. Tämän jälkeen piiriin sai ohjelmoitua, mutta kun sen kytkee USB-liitäntään, tietokone ei tunnista sen olevan kytkettynä. Piirillä oleva vihreä valo, joka kertoo ohjelmakoodin tilasta, värisi ja vilkkui. Hetken päästä se lakkasi palamasta kokonaan. Normalisti toimivassa piirissä vihreän valon pitäisi pysyä päällä.

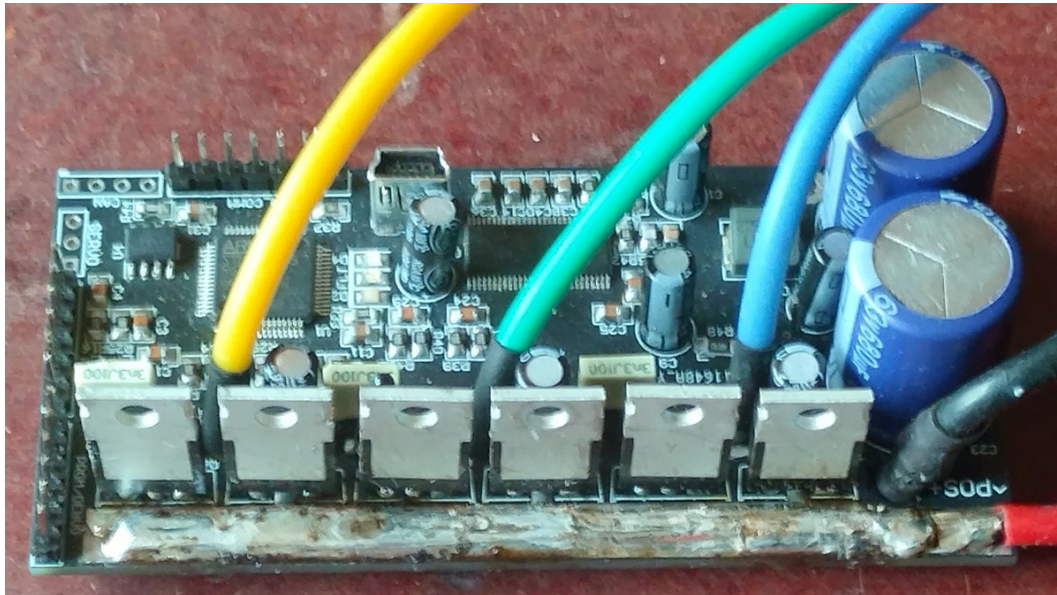
Seuraavaksi kasasin lähes loppuun ja kokeilin kolmatta piirilevyä. Myös sen kanssa kävi siten, että piirin saa ohjelmoitua, mutta vihreä valo ei pala normaalisti, eikä siihen saa yhteyttä USB-liitännän kautta.

Aloin epäillä vikaa kondensaattoreissa, sillä se vaikutti mielestäni siltä, että piirissä saattaa olla epävakautta. Tutkimista vaikeutti se, että minulla olevassa Mastech MY64-yleismittarissa kapasitanssin mittausta on omassa liittimessä, joka on tarkoitettu käytettäväksi siten, että kondensaattorin jalat laitetaan liittimeen. Sitten keksin laittaa johtojen päähän metalliliuskat, että saan ne yhdistettyä siihen liittimeen. Tällä sain mitattua kondensaattoreita piirilevyltä. Mittailtuani piirilevyjä hetken löysin, että näissä kahdessa piirilevyssä, joissa USB-yhteys ei toimi, kondensaattorit C17 ja C19, jotka on kytketty kiteeseen, näyttivät mitatessa isompaa kapasitanssia, kuin siinä piirilevyssä jossa se toimi. Kyseistä kokoa täytyi tilata lisää, joita sitten minulla oli ainoastaan 2 oikean kokoista.

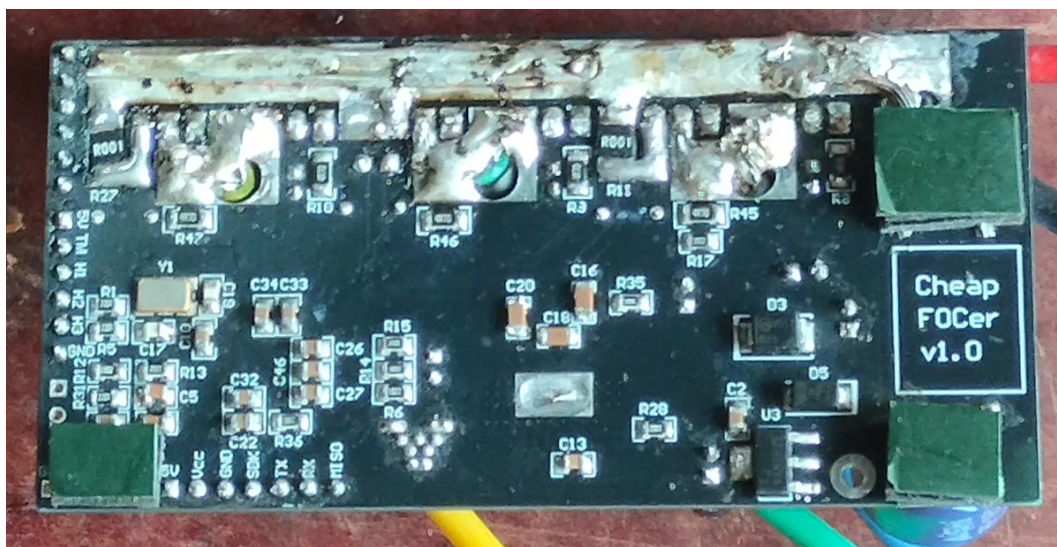
Komponentteja odotellessa mittailin vielä piirilevyjä, jolloin ensimmäisestä testaamastani piirilevystä löytyi että mitatessa vastusta R42, jonka kuuluisi olla 220 kilo-ohminen näytti noin 90 kilo-ohmia. Tarkistin muista piirilevyistä, että mittaus näyttää niissä oikeaa, vaihdoin vastuksen uuteen. Tämän jälkeen mittaus näytti noin 220 kilo-ohmia kaikissa piirilevyissä.

Kun lisää 15pF -kondensaattoreita saapui, juotin niitä piirilevyihin. Tämän jälkeen 2 piirilevyä, joita tietokone ei aiemmin tunnistanut USB-liittimessä, alkoivat tunnistua. Sen jälkeen kokeilin kytkeä toisen niistä moottoriin ja ajaa VESC toolista moottorin tunnistuksen. Moottori yritti pyörähtää. Kun nostin virtalähteestä virran rajoitusta ja jännitteen 24 volttiin, tunnistus meni läpi onnistuneesti. Tämän jälkeen juotin piirilevylle palat kuparikaapelista, jotta se kestää varmasti virtaa joka sieltä kulkee läpi, kun moottorilla ajetaan. Laitoin myös palat kaksipuolista teippiä piirilevyn pohjaan tukemaan sitä, kun se

laitetaan koteloon. Piirilevyn eri puolet olivat tämän jälkeen kuvien 32 ja 33 mukaiset.



Kuva 32. Piirilevyn sisääntulojännitteen positiivinen puoli vahvistettuna.

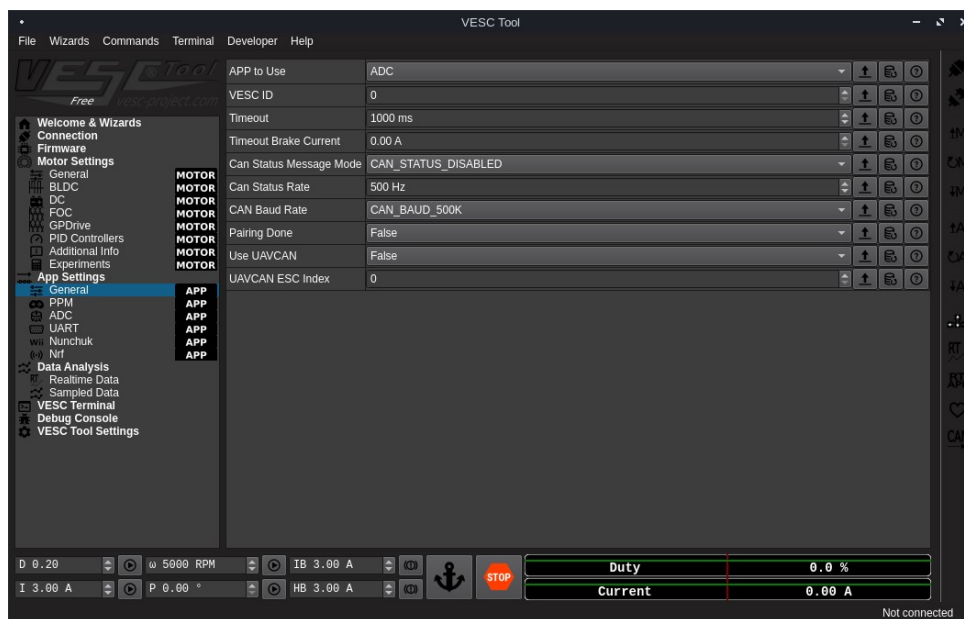


Kuva 33. Piirilevyn sisääntulojännitteen maa vahvistettuna

Kokeilin ajaa ohjaimella testiajon, ja se toimi kuten pitää.

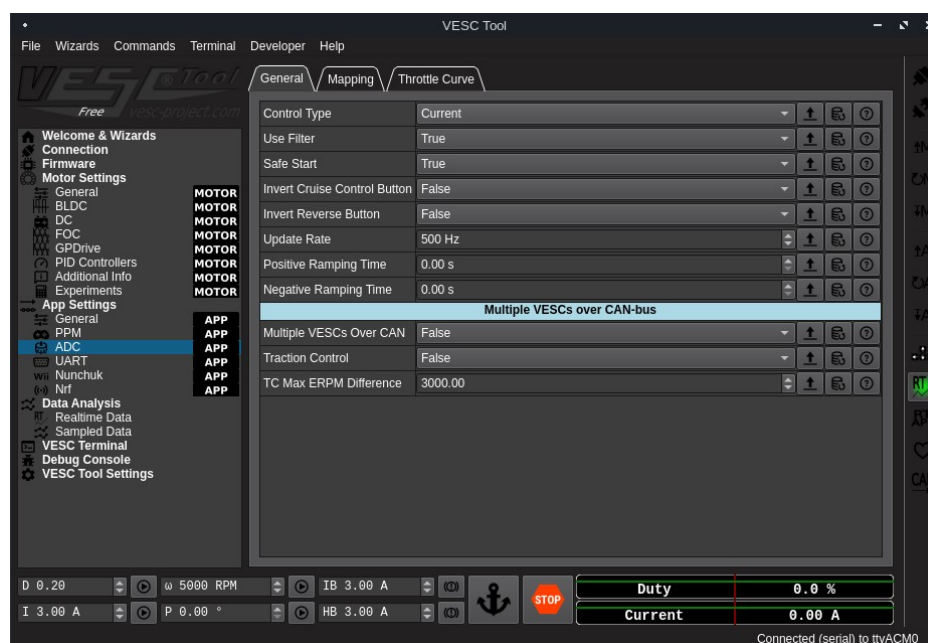
6.2 Asetuksien säätö

Asetuksista valitsin käyttöön ADC. Se on oikea vaihtoehto kaasukahvalle, jolta tuleva jännite muuttuu kaasukahvan asennon mukaan. Kuvassa 34 näkyy ADC valittuna VESC toolista.



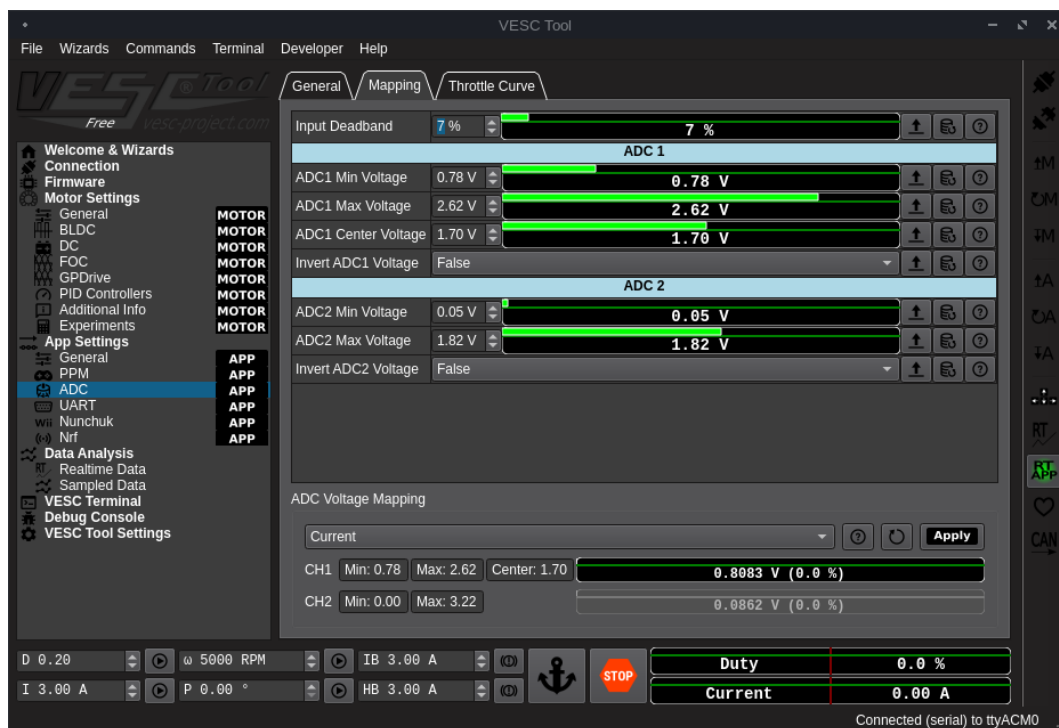
Kuva 34. ADC valittuna kaasukahvalle vesc toolissa.

Kaasukahvalle täytyy valita ohjaustyyppi, valitsin siihen currentin kuvan 35 mukaisesti.



Kuva 35. Kaasun ohjaustyyppi valittuna.

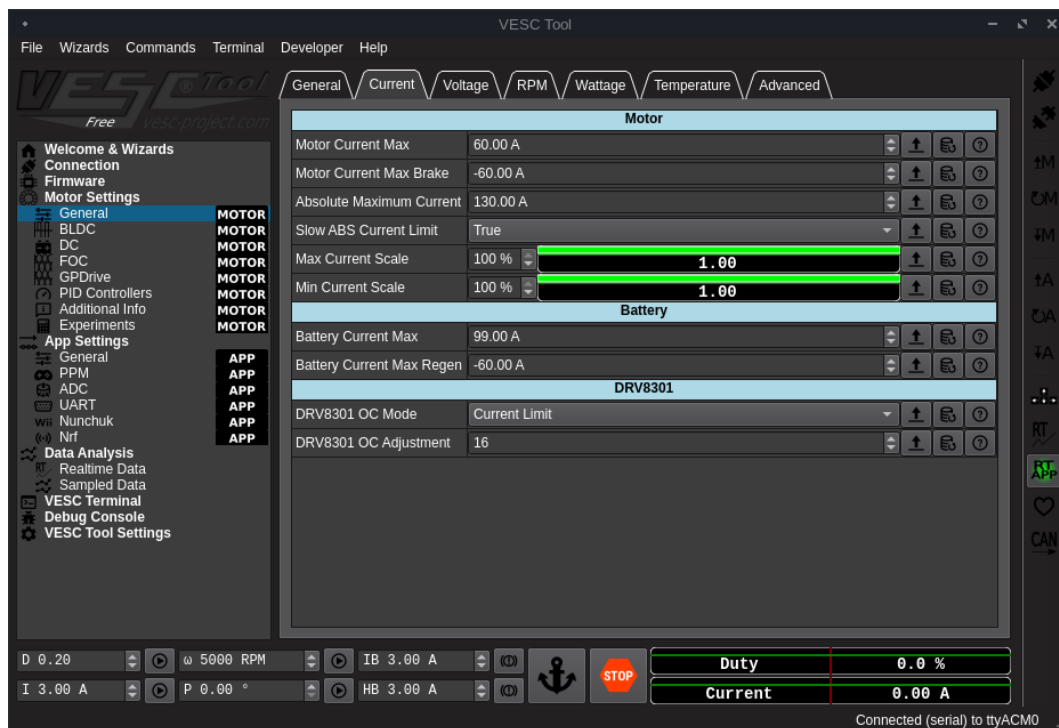
Mapping-asetuksista voidaan säätää kaasun jännitealue käytössä olevan kaasukahvan mukaiseksi. Kun oikeasta reunasta painaa RT APP -painiketta se alkaa näyttää tämänhetkisen kaasun jännitteen. CH1-kohdassa näkyy min-, max- ja center-jännitteet, jotka saa sopiviin arvoihin painamalla kaasun pohjaan. Tämän jälkeen kun painaa apply-painiketta se laittaa arvot ADC1 min-, max- ja center-jännitteiksi. Laskin myös input deadband -asetusta 7 prosenttiin oletuksesta, joka oli 15, sillä mielestäni kaasukahvassa oleva tyhjä alue kääntäessä kaasua oli turhan suuri. Tämän jälkeen mapping-asetukset olivat kuvan 36 mukaiset.



Kuva 36. Kaasukahvan mapping-asetukset säädettynä.

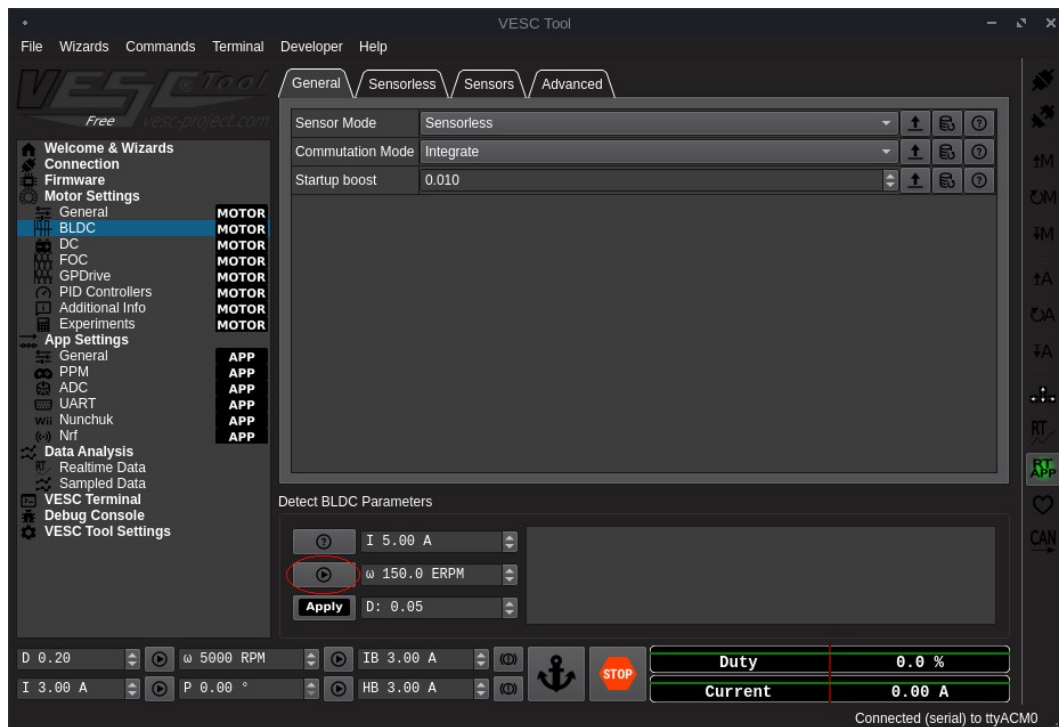
Asetukset tallennetaan ohjaimen painamalla oikeassa reunassa olevaa write app configuration -painiketta.

Ennen kuin testasin moottorin pyörittämistä, halusin säätää virran rajoitukset pienelle. Oletuksena ohjaimen virran rajoitukset olivat olivat VESC toolissa kuvan 37 mukaiset.



Kuva 37. Virran rajoituksen oletusasetukset.

Ensiksi testatessa laitoin akun ja moottorin virran rajoitukset 5 ampeeriin. Tämän jälkeen kokeilin ajaa moottorin tunnistusta BLDC-kohdasta, jonka tein ensin anturittomassa tilassa, koska en ollut vielä kytkenyt hall-antureita ohjaimen. Kuvassa 38 näkyy anturiton tila valittuna ja painike, josta moottorin tunnistus ajetaan on ympyröity punaisella.



Kuva 38. BLDC-asetukset ja moottorin tunnistus.

Ensiksi testatessa 12 voltilla tunnistus ei mennyt läpi, mutta kun nostin virtalähteestä virran rajoitusta ja jännitteen 24 volttiin, sain tehtyä sen onnistuneesti. Silloin tunnistetut asetukset laitetaan käyttöön apply-painikkeella. Sitten kytkin vielä hall-anturit ohjaimeen, jolloin täytyy vaihtaa sensed-tila käyttöön, ja ajaa moottorin tunnistus uusiksi. Asetukset tallennetaan ohjaimeen oikeassa reunassa olevasta write motor configuration -painikkeesta.

Kokeilin tehdä ohjaimella testiajon, jolloin olin laittanut virran rajoitukset aluksi akulle 10A ja moottorille 20A, joilla se toimi, mutta oli tehoton. Kun nostin akun virran rajoituksen 20 ampeeriin ja moottorin virran 40 ampeeriin se alkasi liikkua jo vähän paremmin. Ennen kuin testasin maksimivirralla, jota käyttämästäni akusta saa ottaa, laitoin MOSFETien ja ohjaimen kotelon väliin lämpötahnaa, että ne pysyisivät viileämpinä. Sen jälkeen nostin akun virran rajoituksen 30 ampeeriin, joka on maksimi suositeltu tälle akulle, joka minulla on käytössä, ja moottorin virran 80 ampeeriin. Tällä kiihtyvyys parani taas

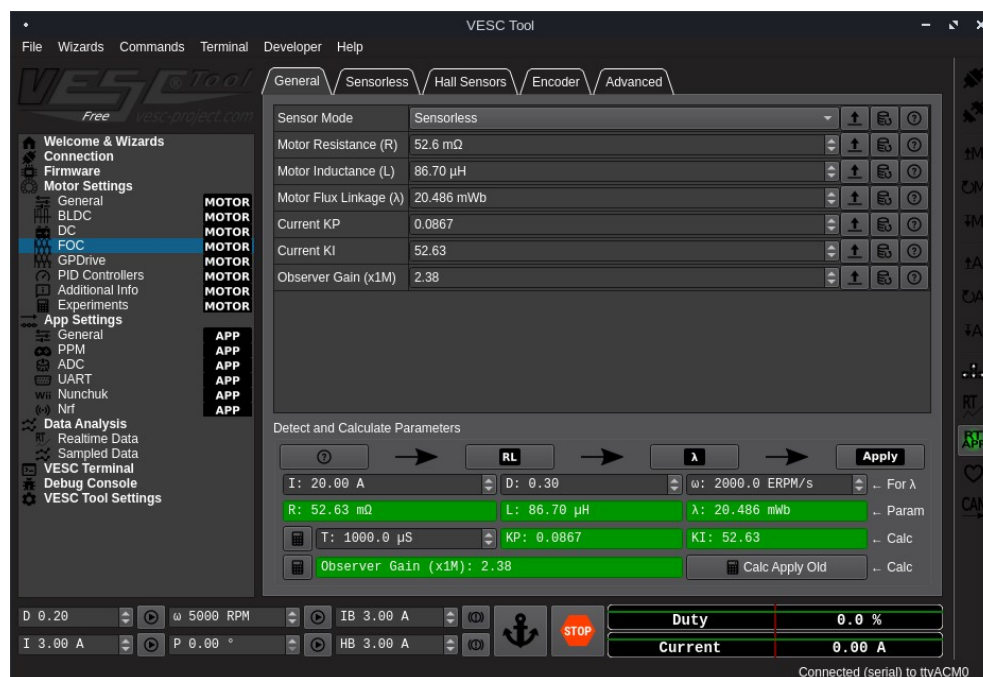
lisää. Ohjain vaikutti toimivan ongelmitta näillä asetuksilla, ja päätin ajaa sillä vähän pidemmästäkin, ja se toimi ongelmitta noin 15 km ajoa, ja siihen tuli joku ongelma, että moottori ei käynyt kokoajan tasaisesti. Kun aloin tutkimaan vikaa, USB-yhteys ei enää toiminutkaan. Ilmeni, että toinen 680 μ F-kondensaattoreista oli löystynyt ja välillä tuli valokaarta löystyneen jalan kohdalla. Korjattuani sen huomasin, että siihen oli tullut sellainen vika, että sisääntulojännitteen ja 5 voltin välillä oli oikosulku; mittari näytti vain 12 ohmia niiden välillä. Tutkittuani sitä hieman. epäilin, että DRV-piiriin oli tullut oikosulku.

Seuraavaksi kasasin toisen piirilevyn jossa USB-yhteys toimi loppuun, voidakseni kokeilla toimiiko se. Siinä oli sellainen ongelma, että virran mittaus näytti, että moottoriin menee virtaa, vaikka näin ei ole. Tutkittuani asiaa löysin, että vastukseen R41, jonka toinen pää on yhteydessä shunttivistukseen, DRV-piirin puoleiseen päähän vuotaa jostain hieman virtaa. Mitatessa siellä oli noin 7-20 millivolttia, kun taas 3 muuta virtamittauksen vastusta näyttivät vain 3 millivolttia. Se selittää miksi virran mittaus näyttää väärin. Vastukseen vuotava jännite oli vaihtelevaa, ja kun se oli pieni, onnistuin ajamaan moottorin tunnistuksen. Mutta heti sen jälkeen siihen tuli vika, alkoi antaa virhettä DRV-piiristä.

Myöhemmin saatuani lisää DRV-piirejä, vaihdoin piirilevyihin uudet ja sen jälkeen piirilevy, joka oli toiminut noin 15km, alkoi toimia taas normaalisti. Toisessa piirilevyssä, jossa oli ollut virran mittauksessa ongelmaa, oli tullut joku oikosulku. Havaitsin että vika oli 0,1 μ F pintaliitoskondensaattorissa C21, mutta huomasin että niitä ei ollut enempää. Päätin silti kokeilla sitä, ja moottorin tunnistus meni läpi onnistuneesti, mutta ADC ei toiminut, joten kaasukahva ei toiminut siinä.

Päätin testata toimivalla ohjaimella ajoa anturittomana. Kun BLDC-ohjaustekniikka oli käytössä, se tuntui nykivän liikkeellehdössä aika paljon.

Päätin kokeilla ajaa sitä FOC-ohjaustekniikalla, jolle pitää tehdä erikseen moottorin tunnistus FOC-asetuksista. Ensiksi painetaan RL-painiketta jolloin ohjain mittaa moottorilta tarvittuja arvoja ja tehdessä sitä moottori voi hieman nykäistä ja pitää ääntä. Sitten painetaan λ -painiketta, joka pyörittää moottoria ja mittaa sille lisää sopivia arvoja. Nämä laitetaan käyttöön apply-painikkeella, jonka jälkeen asetukset olivat kuvan 39 mukaiset.



Kuva 39. FOC-asetukset moottorin tunnistuksen jälkeen.

Lisäksi jos haluaa käyttää hall-antureita FOC-ohjaustekniikan kanssa, pitää ajaa hall-anturien tunnistus hall sensors-kohdasta. Asetukset pitää muistaa myös tallentaa ohjaimen. FOC-ohjaustekniikalla moottorin vääntö tuntui olevan suurempi samoilla virran rajoituksilla kuin BLDC-ohjaustekniikalla, se on hiljaisempi, ja anturittomana liikkeellelähdössä tuli vähemmän nykimistä.

Seuraavaksi kytkin jarrukahvassa olevan kytkimen ohjaimen MISO_ADC_EXT2-pinniin. Kun se oli kytkettynä vain 3,3 volttiin kytkimen kautta, pinniin tulevassa jännitteessä oli liikaa vaihtelua, jarru saattoi mennä päälle vaikka sitä ei paina, joten kytkin sen 10 kilo-ohmin vastuksen kautta maahan, ja

kytkimen 1 kilo-ohmin vastuksen kautta 3,3 volttiin. Sen jälkeen pinniin tuleva jännite pysyi lähes 0 voltissa kun jarrua ei paineta, ja 3,3 voltissa kun sitä painetaan. Tämän jälkeen kokeilin ajaa sitä, ja moottorin jarrutusvirran ollessa asetettu -5 ampeeriin sähköisen jarrun jarrutusteho oli hyvin pieni. Kun laitoin sen -22 ampeeriin sen jarrutusteho alkoi olla mielestäni sopiva siihen, että suuren osan ajasta pelkkä sähköinen jarru riittää, mutta kovempaan jarrutukseen tarvitsee mekaanista jarrua.

Tämän jälkeen laitoin vielä kuitenkin hall-anturit käyttöön, sillä niiden kanssa nykimistä, joka voi tapahtua liikkeellelähdössä tai jarruttaessa hyvin pienessä vauhdissa ei tule.

7 JOHTOPÄÄTÖKSET

Työssä käytiin läpi, miten harjaton moottori ja sen ohjaus toimii teoriassa, valmistettiin ohjain ja kytkettiin se sähköpotkulautaan. Ohjain rakennettiin valmiiksi suunnitellusta piiristä tilaamalla sitä varten piirilevy ja komponentit, ja juottamalla ne piirilevyyn. Sen kasauksessa oli paljon haastetta, sillä pintaliitoskomponenttien juottamisesta minulla oli hyvin vähän aiempaa kokemusta. Piirilevyjen toimimaan saamisessa oli paljon ongelmia, tuli oikosulkuja ja hajosi komponentteja. Lopulta kuitenkin sain tehtyä toimivan piirilevyn, jolla sähköpotkulauta toimi kuten pitää, ja kokeilin, mitä eroa sillä käytännössä tulee anturittomassa ja hall-anturillisessa ohjauksessa sekä BLDC- ja FOC-ohjaustekniikoissa. Anturittomasti ajaessa saattaa tulla vähän nykimistä liikkeellelähdössä tai jarruttaessa sähköisesti pienessä vauhdissa, jota ei tule käyttäessä hall-antureita. FOC-ohjaustekniikka taas teki moottorista hiljaisemmän verrattuna BLDC-ohjaustekniikkaan ja lisäsi moottorin vääntöä.

Tässä työssä oppi harjattoman moottorin toiminnasta ja ohjauksesta, asioita elektroniikasta. kuten pintaliitoskomponenttien juottamisesta, vikojen etsimisestä piirilevyiltä ja mikro-ohjaimen ohjelmoinnista. Olen tyytyväinen siihen, millainen ohjaimesta tuli kun se toimii, mutta en päässyt aivan siihen mitä tavoittelin, että olisi 3 toimivaa ohjainta. Lopussa tuli kiire saada työ valmiiksi, joten ehdin saada vain yhden ohjaimen toimimaan. Aion kuitenkin vielä jatkaa ongelmien selvittämistä ja yrittää saada lisää ohjaimia toimimaan. Lisäksi pitäisi tehdä kytkentä, jolla saan ohjattua jarrukahvalta tulevalle 0-3,3v signaalilla syötetäänkö jarruvalolle akulta tuleva jännite, jotta saan jarruvalon toimimaan. Toiseksi on ideana tehdä ohjaimen kytkettävä näyttö, jolle saa näkyviin tietoja, kuten akun jännite ja nopeus. Siihen voi laittaa myös painikkeet, joilla saa säädettyä ohjaimen asetuksia.

LÄHTEET

1. Hosseini M. Understanding BLDC Motor Controller. [Internet]. [viitattu 9.6.2025]. Saatavilla: <https://arshon.com/blog/understanding-blcd-motor-controller-brushless-dc/>
2. Cheap-FOCer v1.0. [Internet]. [viitattu 1.6.2025]. Saatavilla: <https://github.com/shamansystems/Cheap-FOCer>
3. Brushless DC Motor (BLDC) – Construction, Working, and Applications. [Internet]. [viitattu 9.6.2025]. Saatavilla: <https://www.electricaltechnology.org/2016/05/blcd-brushless-dc-motor-construction-working-principle.html>
4. Joner E. How Brushless Motors Work and How to Test Them. [Internet]. [viitattu 6.2.2025]. Saatavilla: <https://www.tytorobotics.com/blogs/articles/how-brushless-motors-work>
5. Udhayakumar S. Brushless-DC Motor 1: Basics. [Internet]. [viitattu 6.2.2025]. Saatavilla: https://www.ti.com/content/dam/videos/external-videos/en-us/4/3816841626001/6088830019001.mp4/subassets/brushless_dc_basics_.pdf
6. Zhao J, Yu Y. Brushless DC Motor Fundamentals Application Note. [Internet]. [viitattu 9.6.2025]. Saatavilla: https://www.monolithicpower.com/media/document/Brushless_DC_Motor_Fundamentals.pdf
7. Brushless-DC Motor Driver Considerations and Selection Guide. [Internet]. [viitattu 17.2.2025]. Saatavilla: <https://www.ti.com/lit/an/slvaes1a/slvaes1a.pdf>
8. Kim, S. 2017. Electric Motor Control: DC, AC, and BLDC Motors. Chantilly: Elsevier Science & Technology.

9. Hall Effect Sensor Types: Unipolar, Bipolar, Omnipolar, and Linear.

[Internet]. [viitattu 15.2.2025]. Saatavilla:

<https://www.rfwireless-world.com/Terminology/Unipolar-hall-sensor-vs-Bipolar-hall-sensor-vs-Omnipolar-hall-sensor.html>

10. STM32F405xx STM32F407xx datasheet. [Internet]. [viitattu 9.6.2025].

Saatavilla: <https://www.st.com/resource/en/datasheet/stm32f405rg.pdf>