

# CAN-väyläohjattu PWM-ohjain



Ammattikorkeakoulututkinnon opinnäytetyö  
Tieto- ja viestintätekniikka, insinööri (AMK), Riihimäen kampus  
Kevät 2021  
Kenneth Blomstedt

## TIIVISTELMÄ

Tämän opinnäytetyön tavoitteena oli suunnitella sekä toteuttaa CAN-väyläohjattu PWM-ohjain. Ohjaimen oli tarkoitus soveltua ensisijaisesti henkilöauton polttoainepumpun ohjaukseen.

Työ aloitettiin kartoittamalla ohjaimen halutut ensisijaiset sekä toissijaiset toiminnot, jonka jälkeen ohjaimen elektroniikka suunniteltiin komponenttikokonaisuus kerrallaan.

Elektroniikkasuunnittelun jälkeen komponenttikokonaisuuksia testattiin koekytkentöjen avulla, ja suunniteltiin ohjaimelle piirilevy sekä kotelointi. Ohjaimelle kirjoitettiin ohjelma, joka toimintoinen vastasi ohjaimelle asetettuja vaatimuksia. Ohjaimen asetusten määrittämistä varten suunniteltiin sekä toteutettiin Android-mobiililaittealustalle säätöohjelma, jonka tärkeimpiä ominaisuuksia oli helppokäyttöisyys. Työn päätteeksi ohjaimelle rakennettiin testiympäristö, jossa ohjaimen toimintaa testattiin keinokuorman avulla.

Työn toteutuksessa ilmeni haasteita komponenttien saatavuusongelmien sekä suunnitteluvirheiden takia, mutta ne saatiin ratkaistua. PWM-ohjain ylitti testauksessa sille asetetut vaatimukset virtamäärän suhteen komponenttien lämpötilan pysyessä siedettävänä jopa ilman jäähdytystä. Testauksen tulosten perusteella voidaan todeta ohjaimen suunnittelun onnistuneen.

Avainsanat mikro-ohjaimet, ohjelmointi, väylät

Sivut 49 sivua ja liitteitä 1 sivu

---

Author Kenneth Blomstedt

Year 2021

Subject CAN-bus PWM-controller

Supervisors Timo Karppinen

---

ABSTRACT

The objective of this thesis was to design and build a CAN-bus -based PWM-controller. The purpose of the controller was mainly to serve as a fuel pump controller in a passenger car.

The work was started by mapping primary and secondary functions for the controller, after which the controller electronics were designed one function at a time. After finishing the electronics design, the functions were tested on a breadboard. A printed circuit board (PCB) and enclosure were also designed. A program which met design requirements was then written for the controller. For easy modification of configuration parameters in the controller, an easy-to-use Android app was designed and developed. Testing on the finished controller was carried out in a self-made test environment with the aid of an artificial load.

During the development, there were some challenges regarding component availability and minor design errors, but they were overcome. In testing, the PWM-controller proved to fulfill the design requirements, even without any additional component cooling. Based on the test results it can be concluded that the controller design was successful.

Keywords microcontrollers, programming, communication buses

Pages 49 pages and appendices 1 page

## Sisällys

1	Johdanto .....	1
2	Teoria.....	2
2.1	CAN-väylätekniikka .....	2
2.1.1	Viestirakenne .....	3
2.1.2	CAN-väyläohjain .....	4
2.1.3	Väylän nopeus .....	5
2.2	Pulssinleveysmodulaatio (PWM) .....	5
2.3	Tehoelektroniikka (transistorit) .....	6
2.4	Signaalin suodatus .....	7
3	Ohjaimen suunnittelu.....	8
3.1	Toiminnot.....	8
3.2	Mikrokontrolleri.....	9
3.3	Komponenttisuunnittelu.....	10
3.3.1	PWM-lähtö .....	11
3.3.2	Jännitteen mittaus .....	12
3.3.3	CAN-lähetin-vastaanotin (CAN-transceiver) .....	14
3.3.4	FlexFuel-anturi .....	15
3.3.5	Lämpöanturit.....	16
3.3.6	Analogiatulot.....	17
3.3.7	Virran mittaus.....	19
3.3.8	Relelähdöt .....	21
3.3.9	Kiihtyvyyssanturi.....	22
3.3.10	Bluetooth-sarjaliikenne (HC-05).....	23
3.3.11	Mikrokontrolleri .....	24
4	Ohjaimen rakentaminen.....	26
4.1	Kotelo .....	26
4.2	Koekytkenät .....	27
4.2.1	Kytkenät .....	27
4.2.2	PWM-lähtö .....	28
4.2.3	Jännitteen mittaus .....	29
4.2.4	Alipäästösuodatin.....	29
4.2.5	Analogiatulot .....	30
4.2.6	Lämpöanturit.....	32

4.2.7	FlexFuel-anturi .....	33
4.3	Piirilevy.....	34
4.3.1	Suunnittelu .....	34
4.3.2	Komponentit .....	35
4.3.3	Kokoonpano ja testaus.....	36
4.4	Ohjelma.....	36
4.4.1	PWM-lähtö .....	38
4.4.2	Anturitulot ja relelähdöt .....	39
4.4.3	CAN-viestit.....	39
4.4.4	Bluetooth-sarjaliikenne.....	41
4.4.5	Säätöohjelma .....	41
5	Testaus.....	43
5.1	Testiympäristö.....	43
5.2	CAN-väylä .....	43
5.3	PWM-lähtö .....	44
5.4	Tulot ja lähdöt .....	45
5.5	Lämmönhallinta .....	45
5.6	Asennus autoon .....	46
6	Yhteenveto .....	47
	Lähteet.....	48

## Kuvat, taulukot ja kaavat

Kuva 1. CAN-väylän topologia (Frei, 2015, s. 10, muokattu).....	2
Kuva 2. CAN-väyläviestin viestikehys (Frei, 2015, s.11, muokattu).....	3
Kuva 3. CAN-väyläohjaimen tasot (Frei, 2015, s. 16, muokattu).....	4
Kuva 4. Kanavatransistorityypit (Tutorialspoint, 2021).....	6
Kuva 5. RC-alipäästösuodatin (Ahonen, Haiko & Salonen, 2016, s. 234).....	7
Kuva 6. NUCLEO-F072RB -kehityskortti (Partco, n.d.–a).....	10
Kuva 7. Lohkokaavio ohjaimen komponenteista. ....	11
Kuva 8. KytKentäkaavio: PWM-lähtö.....	12
Kuva 9. KytKentäkaavio: Syöttöjännitteen mittaus. ....	14
Kuva 10. KytKentäkaavio: CAN-lähetin-vastaanotin.....	15
Kuva 11. FlexFuel-anturi (Fuel Flex Finland, 2017).....	15
Kuva 12. KytKentäkaavio: FlexFuel-anturi. ....	16
Kuva 13. KytKentäkaavio: Lämpöanturi.....	16
Kuva 14. KytKentäkaavio: Analogiatulo. ....	18
Kuva 15. KytKentäkaavio: +5V virransyöttö.....	19
Kuva 16. KytKentäkaavio: Virranmittauspiiri. ....	21
Kuva 17. KytKentäkaavio: Relelähtöpiiri.....	22
Kuva 18. KytKentäkaavio: Kiihtyvyyssanturi.....	23
Kuva 19. KytKentäkaavio: Bluetooth-moduuli.....	23
Kuva 20. KytKentäkaavio: +3V3 virransyöttö. ....	24
Kuva 21. KytKentäkaavio: Mikrokontrollerin ohjelmointiliitin. ....	25
Kuva 22. KytKentäkaavio: Ulkoiset kellopiirit. ....	25
Kuva 23. Alumiininen kotelo liittimellä. ....	26
Kuva 24. Transistori jäähdytys-elementillä (Conrad, 2020.).....	27
Kuva 25. PWM-lähdön signaali oskilloskoopin lukemana. ....	28
Kuva 26. Suodattimen tulo ja lähtö ennen muutosta sekä muutoksen jälkeen. ....	30
Kuva 27. Analogiatulon jännite mikrokontrollerin laskemana.....	31
Kuva 28. Lämpöanturitulon laskenta mikrokontrollerin ohjelmassa.....	33
Kuva 29. FlexFuel-anturin signaaliarvot eri polttoainepitoisuuksilla. ....	34
Kuva 30. Mallinnuskuva valmiista piirilevystä.....	35

Kuva 31. Pääohjelman rakenne ja ohjelmalohkojen suoritusmalli. ....	37
Kuva 32. Käyttöliittymän aloitusnäkyvä, päänäkyvä sekä asetusnäkyvä.....	42
Kuva 33. Asetusnäkyvän alavetovalikot sekä lisätiedot. ....	43
Taulukko 1. Ohjaimen suunnitellut toiminnot. ....	9
Taulukko 2. Koekytkenän vastusarvot.....	29
Taulukko 3. Jännitemittauspiirin tulokset. ....	29
Taulukko 4. NTC-lämpöanturin muuntotaulukko.....	32
Taulukko 5. Analogiatulon muuntotaulukko eräälle paineanturille. ....	39
Taulukko 6. CAN-viestitaulukko.....	40
Kaava 1. CAN-väylän tiedonsiirtonopeus (Frei, 2015, s. 17). ....	5
Kaava 2. Pulssisuhde.....	6
Kaava 3. Vaikuttava jännite. ....	6
Kaava 4. Kapasitiivinen reaktanssi (Ahonen, Haiko & Salonen, 2016, s. 234). ....	8
Kaava 5. RC-alipäästösuodattimen mitoitus (OKAWA Electric Design, 2021). ....	8
Kaava 6. Jännitteen jakaja. ....	12
Kaava 7. Sähköteho. ....	19
Kaava 8. Anturipiirin signaalijännite. ....	20
Kaava 9. Virranmittauksen tarkkuus. ....	20
Kaava 10. Vastusarvo jännitteen jakopiirissä. ....	32
Kaava 11. Etanolipitoisuus prosentteina FlexFuel-anturisignaalin taajuudesta. ....	33
Kaava 12. Polttoaineen lämpötila FlexFuel-anturisignaalin pulssisuhteesta. ....	33

## **Liitteet**

Liite 1	Testauspöytäkirja
---------	-------------------

## 1 Johdanto

Tämän opinnäytetyön tarkoituksena on suunnitella sekä toteuttaa CAN-väyläohjattu PWM-ohjain henkilöauton polttoainepumpun portaattomaan ohjaukseen. Työssä on tarkoitus tutkia ohjaimen toiminnan kannalta tärkeitä osa-alueita ja toimintaperiaatteita, kuten CAN-väylätekniikan perusteita sekä PWM-signaalin ominaisuuksia, jonka jälkeen suunnitella ja rakentaa PWM-ohjain. Ohjaimen on tarkoitus määrittellä mahdollisuuksien mukaan sellaisia toissijaisia toimintoja, joiden katsotaan olevan hyödyllisiä, ohjaimelle lisäarvoa tuovia ominaisuuksia. Ohjaimen asetusten määrittämistä varten on myös tarkoitus suunnitella ja toteuttaa säätöohjelma, jonka avulla asetusten määrittäminen on käyttäjälle yksinkertaista. Työssä on tarkoitus perehtyä edellä mainittujen asioiden lisäksi myös piirilevysuunnitteluun sekä -kokoontamiseen. Kaikkia osa-alueita ei aiheen laajuuden takia ole tarkoitus käydä työssä läpi.

Ajatus oman ohjaimen kehittämisestä heräsi, kun opinnäytetyön tekijän harrasteajoneuvoon ei markkinoilla ollut saatavilla valmista kohtuuhintaista ratkaisua, joka olisi ominaisuuksiltaan sopinut ajateltuun käyttötarkoitukseen. Tämän opinnäytetyön tuloksena valmistuva PWM-ohjain on tarkoitus asentaa kyseiseen harrasteajoneuvoon, jolloin ohjaimen toimintaa sekä toimivuutta voidaan arvioida pitkällä aikavälillä, ja näin ollen jatkaa ohjaimen kehitystä myös työn valmistumisen jälkeen.



## 2 Teoria

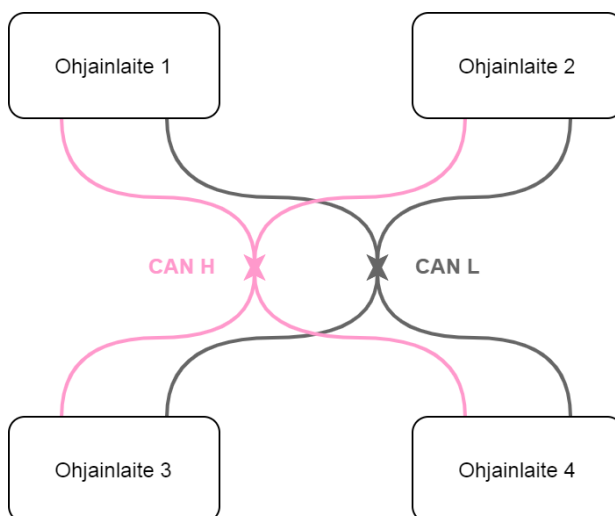
### 2.1 CAN-väylätekniikka

CAN-väylä on ajoneuvoissa yleisesti käytetty väylätyyppi, jossa tietoa (dataa) kuljetetaan kahta kuparijohdinta pitkin erinäisten ohjainlaitteiden välillä. Kaikki CAN-väylän ohjainlaitteet ovat kytkettynä samoihin kuparijohtimiin, jolloin lähetetty data saavuttaa jokaisen laitteen (Kuva 1). Johtimia on kaksi, CAN H (High) sekä CAN L (Low), ja ne ovat kierretty toistensa ympärille ”Twisted Pair” -tyyppisesti. Tällä menetelmällä saavutetaan parempi häiriösuojaus. (Frei, 2015, ss. 10–23)

CAN-väylätyyppejä on olemassa kaksi, CAN High Speed sekä CAN Low Speed. High Speed -väylässä tiedonsiirtonopeus on 125–1000 kilobittiä sekunnissa, kun se Low Speed -väylässä jää alle 125 kilobittiin sekunnissa. (Frei, 2015, ss. 10–23)

CAN-väylässä jokainen laite on isäntälaitte eli ”master”. Tämä tarkoittaa sitä, että kaikilla laitteilla on samat oikeudet lähettää sekä vastaanottaa viestejä. Jokainen väylässä oleva laite näkee kaikki väylässä liikkuvat viestit, ja on laitekohtaisesti määritettävissä, mitkä viestit halutaan vastaanottaa luettavaksi. (Frei, 2015, ss. 10–23)

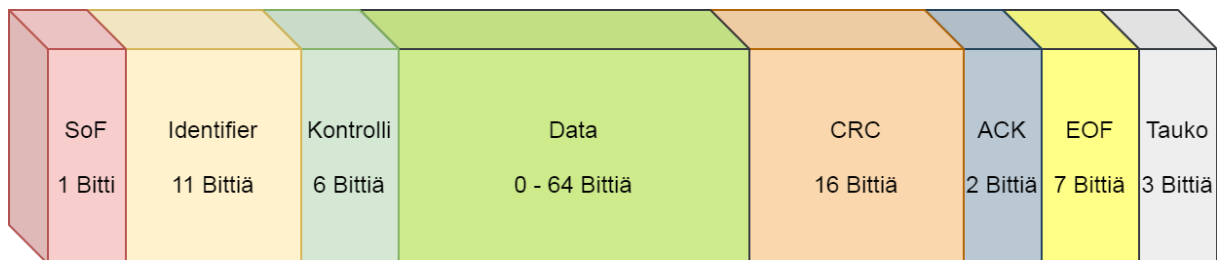
Kuva 1. CAN-väylän topologia (Frei, 2015, s. 10, muokattu).



### 2.1.1 Viestirakenne

CAN-väylän viestiprotokollia on kaksi: yleisin käytössä oleva standardiprotokolla sekä suuren viestimäärän sovelluksissa laajennettu protokolla. Standardiprotokollan CAN-väyläviesti koostuu viestikehyksestä, jonka sisältö on tarkasti määritelty (Kuva 2). Viesti alkaa käynnistysbitillä (SoF, Start of Frame), jonka jälkeen tulee tunnistuskenttä (Identifier). Tunnistuskenttä on yksilöllinen, joten kaksi eri laitetta eivät saa lähettää saman sisällön sisältävää tunnistuskenttää. Tunnistuskentän jälkeen tulee kontrollikenttä, joka pitää sisällään viestin datan pituuden (DLC). Seuraavaksi lähetetään itse data, joka voi olla maksimissaan 64 bittiä (8 tavua) pitkä. Datan jälkeen lähetetään tarkistussumma (CRC), jotta väylän muut ohjainlaitteet voivat tarkistaa, että viesti on säilynyt eheänä vastaanottavalle laitteelle asti. Tässä vaiheessa lähettävä laite jää odottamaan muilta väylän ohjainlaitteilta kuittausta (Acknowledge), että viesti on toimitettu. Lopuksi lähetetään päätöskenttä (EOF) joka on seitsemän bittiä pitkä. Tämän jälkeen tulee kolmen bitin väylätauco, jonka aikana väylässä ei liiku mitään tietoa. Tauon jälkeen mikä tahansa laite voi taas aloittaa viestin lähetyksen. (Frei, 2015, ss. 11–13)

Kuva 2. CAN-väyläviestin viestikehys (Frei, 2015, s.11, muokattu).

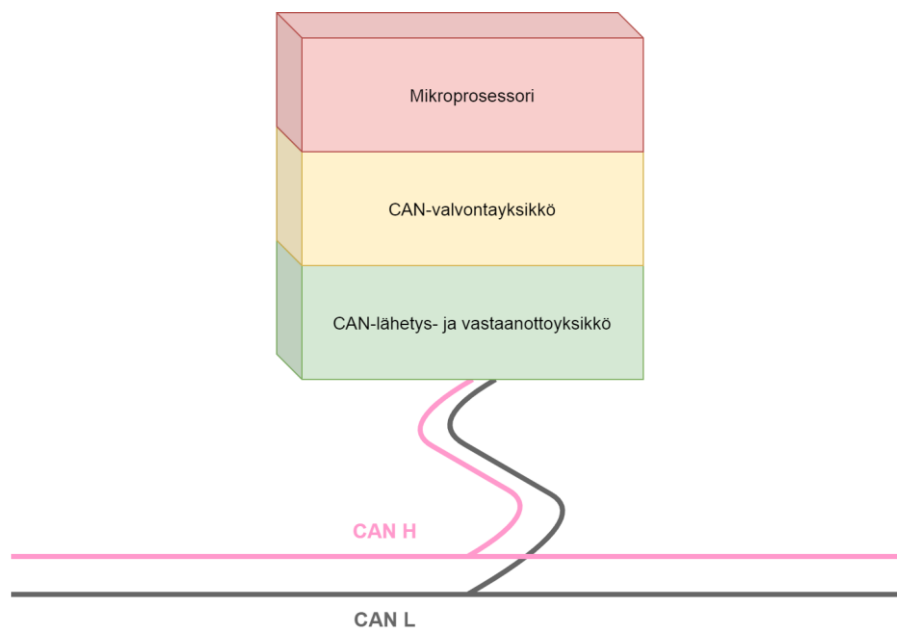


Jokaisen CAN-väylässä olevan ohjainlaitteen ollessa samanarvoinen, on pitänyt määrittää säännöt, joilla estetään konfliktit viestiliikenteessä laitteiden välillä. Tämä on ratkaistu luomalla viestien priorisointijärjestys. Viestin tunnistuskenttä (Identifier) määrittää viestin prioriteetin; mitä tärkeämpi viesti, sitä pienempi luku. Kommunikaatiossa tämä tarkoittaa sitä, että jos tapahtuu yhtäaikainen lähetyseritys, saa se ohjainlaite, jonka tunnistuskentän luku on pienempi, jatkaa lähetystä, ja muut lähettämistä yrittäneet ohjainlaitteet peruuttavat lähettämisen ja jäävät lukemaan tärkeimmän prioriteetin viestiä. (Frei, 2015, ss. 12–13)

### 2.1.2 CAN-väyläohjain

CAN-väyläohjaimen toiminta koostuu kolmesta eri tasosta (Kuva 3). Ensimmäisellä tasolla on mikroprosessori, jonka sisältämä tieto halutaan lähettää CAN-väylään tiedoksi muille ohjaimille. Toinen taso on CAN-valvontayksikkö (CAN Controller), jossa tämän tiedon pohjalta luodaan viestikehys CAN-väylään. CAN-valvontayksikkö voi olla sisäänrakennettu mikrokontrolleriin, tai erillinen mikropiiri. Kolmannella tasolla on lähetys- ja vastaanottoyksikkö (CAN Transceiver), joka muuntaa viestikehyksen rautatasolla sopivaksi signaaliksi CAN-väylään sekä lähettää ja vastaanottaa viestit CAN-väylästä. (Frei, 2015, ss. 13–15)

Kuva 3. CAN-väyläohjaimen tasot (Frei, 2015, s. 16, muokattu).



Esimerkiksi voidaan ottaa auton moottorin lämpötila, jolle on tarvetta muissakin ohjainlaitteissa moottorin ohjainlaitteen lisäksi. Moottorin ohjainlaite saa lämpötilatiedon lämpötila-anturilta. Omien sisäisten toimintojensa lisäksi moottorin ohjainlaitteen prosessori ohjaa tiedon CAN-valvontayksikölle, joka luo lämpötilatiedon sisältävän viestikehyksen. Tämä viestikehys ohjataan CAN-lähetys- ja vastaanottoyksikölle, joka toimittaa viestikehyksen CAN-väylään. Muiden väylässä olevien ohjainlaitteiden lähetys- ja vastaanottoyksiköt lukevat viestikehyksen ja välittävät tiedot CAN-valvontayksikön kautta ohjainlaitteen mikroprosessorille. Jos lämpötilatietoa ei jossain ohjainlaitteessa tarvita, CAN-valvontayksikkö jättää viestin huomioimatta. (Frei, 2015, ss. 15–16)

### 2.1.3 Väylän nopeus

CAN-väylän nopeus määrittää kuinka paljon dataa voidaan enimmillään siirtää tietyssä ajanjaksossa. Väylän nopeus vaikuttaa myös väylän maksimipituuteen, mitä nopeampi tiedonsiirtonopeus, sitä lyhyempi maksimipituus. CAN Low Speed -ratkaisussa väylän pituus voi maksimissaan olla noin 500 metriä, kun taas CAN High Speed -väylässä maksimipituus on noin 130 metriä. Tiedonsiirtonopeus sekunneissa bittikohtaisesti  $t_{Bitti}$  sekä yhdelle kehykselle  $t_{Kehys}$  voidaan laskea kaavan 1 avulla, kun tiedetään väylän nopeus  $TDS$  ja bittien määrä  $n_{Bitti}$  kehyksessä. (Frei, 2015, ss. 17–18)

Kaava 1. CAN-väylän tiedonsiirtonopeus (Frei, 2015, s. 17).

$$t_{Bitti} = \frac{1 \text{ Bit}}{TDS}$$

$$t_{Kehys} = t_{Bitti} \times n_{Bitti}$$

## 2.2 Pulssinleveysmodulaatio (PWM)

Pulssinleveysmodulaatio, eli pulssisuhdesäätö on nimensä mukaisesti pulsseina syötettyä tasavirtajännitettä. Pulssien taajuus on kiinteä, ja pulssin leveys muuttuva. Muuttamalla pulssin leveyttä saadaan aikaan haluttu keskimääräinen vaikuttava jännite, jota säätämällä voidaan vaikuttaa kytketyn toimilaitteen tehoon. (Schneehage, 2018, ss. 51–53)

Pulssisuhdesäätöä käytetään yleisesti ajoneuvojen toimilaitteiden ohjauksessa. Hyviä esimerkkejä ovat erinäiset magneettiventtiilit sekä sähkömoottorit, joita voidaan pulssisuhdesäädön avulla ohjata portaattomasti ja näin ollen mahdollistaa laaja säätöalue pelkän auki/kiinni- tai päällä/pois-toiminnan sijaan. (Schneehage, 2018, ss. 51–53)

Pulssisuhdesäätö on neliöaalto, jonka amplitudi on yleisesti syöttöjännitteen suuruinen. Pulssisuhde  $D$  on riippuvainen pulssin pituudesta  $M$  verrattuna aallon pituuteen  $T$  (Kaava 2). Pulssisuhde voi vaihdella 0 ja 1 välillä, ja yleisesti se ilmaistaan prosenteissa, 0–100%. (Ibrahim, 2014)

Kaava 2. Pulssisuhde.

$$D = \frac{M}{T}$$

Keskimääräinen toimilaitteeseen vaikuttava jännite  $V_{eff}$  voidaan laskea, kun tiedetään neliöaallon amplitudi  $A$  volteissa sekä pulssisuhde  $D$  (Kaava 3).

Kaava 3. Vaikuttava jännite.

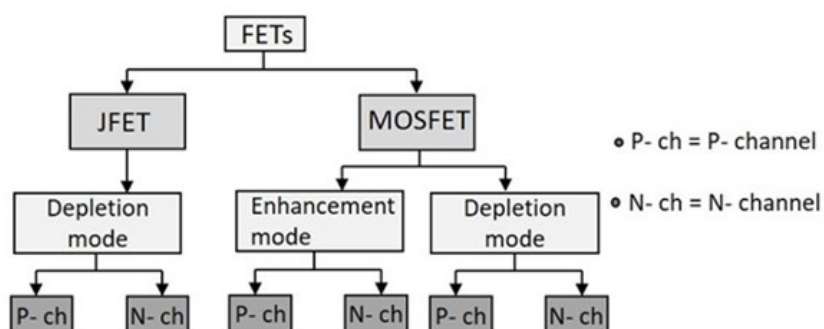
$$V_{eff} = D \times A$$

## 2.3 Tehoelektroniikka (transistorit)

Transistori on elektroniikan puolijohdekomponentti, jota voidaan hyödyntää moneen eri käyttötarkoitukseen, kuten signaalin vahvistukseen, virtojen kytkentään, oskillaattorina, säädettävänä vastuksena tai jopa lämpötila-anturina. Transistoreita on olemassa kahta eri päätyyppiä, BJT sekä FET. Huomattavin ero on transistorin ohjaustavassa. BJT:n, eli bipolaaritransistorin ohjaus tapahtuu kannan (B) kautta kulkevalla virralla, kun taas FET:in, eli kanavatransistorin ohjaus tapahtuu tuomalla hilalle (G) jännite. (Haiko, 2013, ss. 235–239)

Kanavatransistoreiden kaksi päätyyppiä ovat liitoskanavatransistorit (JFET) ja eristehilatransistorit (MOSFET) (Kuva 4). Eristehilatransistoreita on vielä kahta eri tyyppiä, sulkukanavatransistorit (Depletion) sekä avauskanavatransistorit (Enhancement). Kaikki kanavatransistorit jaotellaan vielä transistorin rakenteen mukaan N- tai P-kanavatransistoreiksi. (Silvonen, 2009, s. 197)

Kuva 4. Kanavatransistorityypit (Tutorialspoint, 2021).



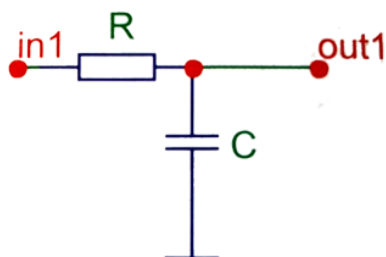
## 2.4 Signaalin suodatus

Suodattimet ovat elektronisia piirejä, jotka vaikuttavat niiden läpi menevän signaalin voimakkuuteen tai taajuuteen, jolloin voidaan aikaansaada tasaisempi, häiriötön signaali tai suodattaa tietty osa-alue (taajuus) kokonaan pois. Yleinen esimerkki suodattimien käyttökohteista on viihde-elektroniikka, joissa suodattimia voidaan käyttää esimerkiksi kohinan poistamiseen ja äänensävyyn muuttamiseen. Kaiuttimien jakosuotimilla voidaan suodattaa äänisignaali sopivaksi basso- ja diskanttikaiutinelementtien toiminta-alueelle (taajuusvaste). (Silvonen 2009, s. 297)

On olemassa analogisia sekä digitaalisia suodattimia. Analogiset suodattimet jaotellaan kahteen luokkaan: aktiiviset ja passiiviset. Aktiiviset suodattimet sisältävät aktiivisia komponentteja kuten operaatiovahvistimia, kun taas passiiviset suodattimet koostuvat yksinkertaisimmillaan vastuksesta sekä kondensaattorista (passiivinen RC-suodatin) tai kelasta sekä kondensaattorista (passiivinen LC-suodatin). Digitaalisia suodattimia käytetään nykyään enenemissä määrin ja niitä ovat esimerkiksi näytejonosuodattimet sekä kytkin-kapazitanssi-suodattimet (SC-suodattimet). (Silvonen 2009, s. 297)

Passiivinen RC-alipäästösuodatin koostuu kahdesta komponentista, vastuksesta  $R$  sekä kondensaattorista  $C$ . RC-alipäästösuodattimen toimintatapa on sama kuin jännitteenjakajassa, jossa signaali tuodaan sisään suodattimelle ( $in1$ ), mutta vain osa siitä jatkaa ulos ( $out1$ ) (Kuva 5). (Ahonen, Haiko & Salonen, 2016, s. 234)

Kuva 5. RC-alipäästösuodatin (Ahonen, Haiko & Salonen, 2016, s. 234).



Vastuksen resistanssi  $R$  on vakio, mutta signaalin taajuudesta  $f$  riippuen kondensaattorin kapasiivinen reaktanssi  $X_C$  muuttuu kaavan 4 mukaisesti.

Kaava 4. Kapasitiivinen reaktanssi (Ahonen, Haiko & Salonen, 2016, s. 234).

$$X_c = \frac{1}{2\pi fC}$$

Taajuuden noustessa kapasitiivinen reaktanssi pienenee, jolloin yhä suurenevampi osuus signaalista reitittyy kondensaattorin  $C$  läpi maattoon ulos menevän signaalin (*out1*) sijaan (Kaava 5). Tällä tavalla saadaan aikaiseksi signaalin vaimennus halutulta taajuudelta ylöspäin. Taajuutta  $f_c$  kutsutaan rajataajuudeksi. Rajataajuudella tarkoitetaan taajuutta, jossa siirrytään suodattimen päästökaistalta estokaistalle tai päinvastoin. (Ahonen, Haiko & Salonen, 2016, s. 235)

Kaava 5. RC-alipäästösuodattimen mitoitus (OKAWA Electric Design, 2021).

$$f_c = \frac{1}{2\pi RC}$$

### 3 Ohjaimen suunnittelu

#### 3.1 Toiminnot

Ohjaimen suunnittelu aloitettiin määrittämällä tarvittavat toiminnot (Taulukko 1). Näiden toimintojen pohjalta muodostettiin vaatimukset ohjaimen elektronikalle sekä koteloinnille. Kriittisin askel oli mikrokontrollerin valinta, koska mikrokontrolleri määrittäi pitkälti myös sen, mitä ohjelmointiympäristöä tulisi käyttää. Ohjelmointiympäristö taas määrittäi sen, miten laajalti valmiita kirjastoja voitaisiin hyödyntää mikrokontrollerin ohjelman kirjoituksessa. Mikrokontrollerin valinta vaikutti myös muiden tarvittavien mikropiirien sekä muunto- ja anturipiirien kytkentään sekä ohjelmointiin.

Ohjaimen tärkeimmät toiminnot olivat vähintään 30 ampeerin PWM-signaalin tuottaminen polttoainepumpun portaaton ohjausta varten sekä toimiva kommunikaatio muiden laitteiden kanssa CAN-väylän avulla. Toissijaisia toimintoja valitun mikrokontrollerin ominaisuuksia hyväksikäyttäen olivat erinäiset jännitemittaukset, PWM-lähdön

virranmittaus, FlexFuel-anturin kytkentä, lämpöanturin kytkentä, analogiatulon kytkentä, relelähdön kytkentä, kiihtyvyyssanturin kytkentä sekä bluetooth-kommunikaatio.

Taulukko 1. Ohjaimen suunnitellut toiminnot.

Toiminto	Määrä	Lisätiedot	Kytkentä mikrokontrolleriin
PWM-lähtö	1	> 30 A	Digitaalilähtö
Virran mittaus	1	> 30 A	Analogiatulo
Syöttöjännitteen mittaus	1	0 - 15 V	Analogiatulo
PWM-jännitteen mittaus	1	0 - 15 V	Analogiatulo
FlexFuel-anturi	1	50 - 150 Hz	Digitaalitulo (taajuus)
Analogiatulo	2	0 - 5V	Analogiatulo
Lämpöanturi	2	NTC / vastaava	Analogiatulo
Relelähtö	2	12V / 100mA	Digitaalilähtö
Kiihtyvyyssanturi	1	-	Väylä
Bluetooth	1	USART	Sarjaliikenneportti

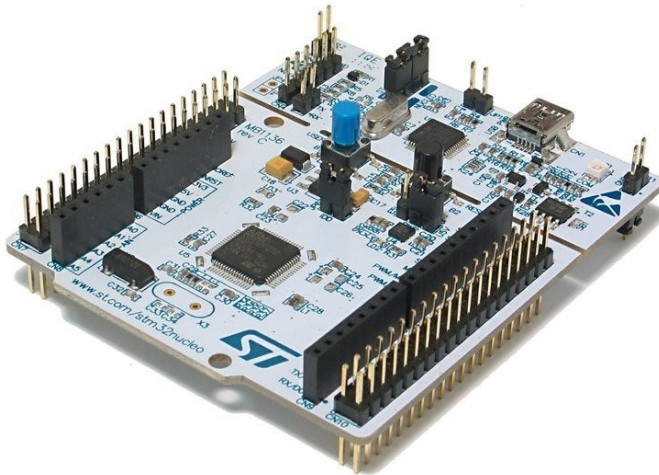
### 3.2 Mikrokontrolleri

Ohjaimen vaatimukset täyttäviä mikrokontrollereita sekä ohjelmointiympäristöjä on olemassa lukematon määrä, joista tunnetuimpia ovat aloittelijoille sekä harrastelijoille suunnatut Arduino sekä Mbed OS. Arduino hyödyntää pääsääntöisesti Atmelin valmistamia ATmega-mallisia 8-bittisiä mikrokontrollereita ja sen ohjelmointiympäristönä toimii Arduino IDE. (Arduino, 2021) Mbed OS tukee monen eri valmistajan mikrokontrollereita, joista suurin osa perustuu ARM Cortex 32-bittiseen arkkitehtuuriin. Eri valmistajia on kymmeniä, ja tehokas nykyaikainen mikrokontrolleri mahdollistaa monen ominaisuuden ja toiminnon integroinnin suoraan prosessorin sisään. (Mbed OS, 2021–b)

Tässä opinnäytetyössä päädyttiin käyttämään STM32-pohjaista mikrokontrolleria Mbed OS -ohjelmointiympäristön kanssa. Mikrokontrolleriksi valikoitui STM32F072RBT6, joka sisälsi kaikki opinnäytetyössä tarvittavat ominaisuudet. Kyseiselle mikrokontrollerille oli myös saatavilla kehityskortti NUCLEO-F072RB, joka helpottaisi kytkentöjen sekä ulkoisten piirien testausta käytännössä ennen oman piirilevyn valmistusta (Kuva 6).



Kuva 6. NUCLEO-F072RB -kehityskortti (Partco, n.d.–a).

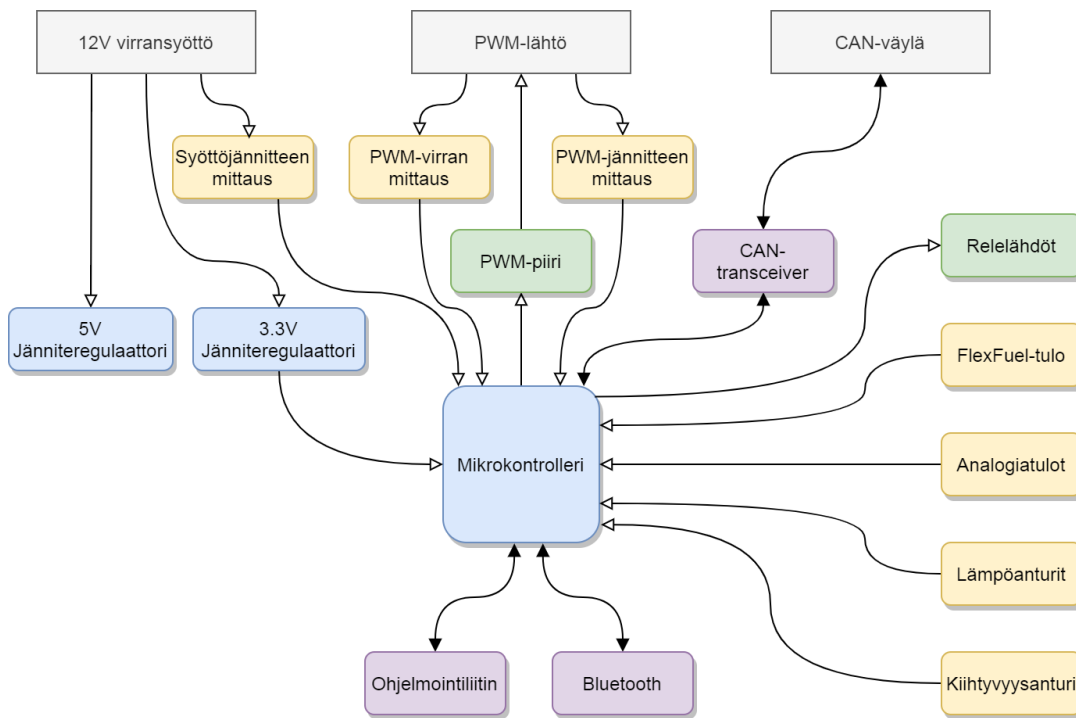


STM32F072RBT6 on ARM-arkkitehtuuriin pohjautuva 32-bittinen mikrokontrolleri, joka sisältää 128 kilobittiä Flash-muistia ja 16 kilobittiä RAM-muistia. Mikrokontrollerin maksimitaajuus on 48 MHz ja tuloja/lähtöjä on konfiguraatiosta riippuen jopa 51 kappaletta. Mikrokontrolleri sisältää kaksitoista ajastinta, 12-bittisen analogia-digitaalimuuntimen kuudelletoista kanavalle sekä myös yhden digitaal-analogiamuuntimen. Mikrokontrolleriin sisäänrakennettuja kommunikointiliittymiä ovat muun muassa I2C, USART, SPI, USB sekä CAN. Mikrokontrollerin pakkaustyyppi on LQFP64. (STMicroelectronics, 2019)

### 3.3 Komponenttisuunnittelu

Komponenttisuunnittelua varten luotiin ensin lohkoakaavio, jonka avulla pystyttiin muodostamaan selkeä kuva siitä, minkälaisia komponenttikokonaisuuksia ohjaimen mikrokontrollerin lisäksi tarvitaan (Kuva 7). Tämän jälkeen piirit suunniteltiin yksi kerrallaan. Suunnitteluun käytettiin EasyEDA-suunnitteluohjelmistoa, jolla on mahdollista tehdä komponenttisuunnittelua sekä piirilevysuunnittelua. Komponenttisuunnittelun jälkeen piirit testattiin mahdollisuuksien mukaan kehityskortilla.

Kuva 7. Lohkokaavio ohjaimen komponenteista.

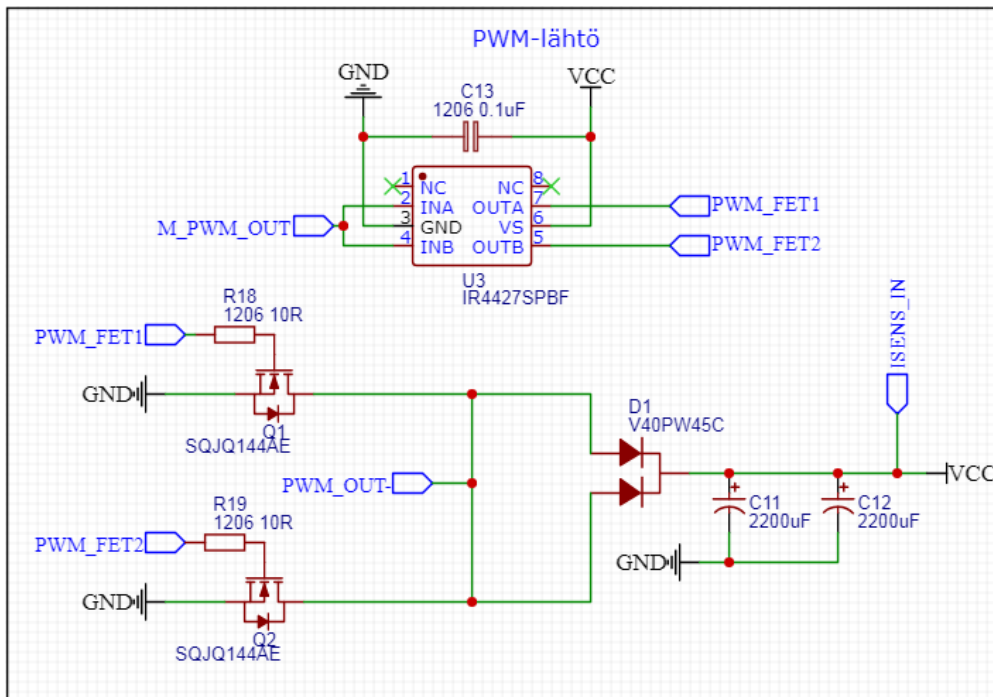


### 3.3.1 PWM-lähtö

PWM-lähtöä varten tarvittiin komponenttikokonaisuus, joka kestäisi suuria virtamääriä, nopeita tilanvaihtoja sekä tuottaisi mahdollisimman vähän hukkalämpöä. Transistoreita päätettiin laittaa kaksi kappaletta rinnankytkettynä, jolloin ne jakavat kuorman. Transistoreiksi valikoituivat suuren tehonkeston sekä pienen kytkentähäviön omaavat N-kanava MOSFET-transistorit.

Transistoreiden lämpöhäviön minimoimiseksi tarvitaan mahdollisimman nopea kytketyminen, joka käytännössä saavutetaan suurella hilan ohjausjännitteellä. Mikrokontrollerin matala logiikkatason jännite ei ollut paras mahdollinen vaihtoehto, joten transistoreiden ohjausta varten päädyttiin käyttämään erillistä vahvistinpiiriä. Käytetty vahvistinpiiri mahdollistaa transistoreiden ohjauksen ohjaimen syöttöjännitteellä, jolloin transistorin kytkentähäviöt ovat pienemmät. Kytkentään lisättiin myös estodiodipari sekä kondensaattoreita tasaamaan transistoreiden kytketyksestä aiheutuvia virtapiikkejä (Kuva 8).

Kuva 8. Kytentäkaavio: PWM-lähtö.



### 3.3.2 Jännitteen mittaus

Jännitteen mittaus haluttiin suorittaa ohjainlaitteen syöttöjännitteestä sekä PWM-lähdön jännitteestä. Nämä jännitteet ovat noin 14 voltia auton ollessa käynnissä, mutta mikrokontrollerin analogiatulo pystyy lukemaan maksimissaan 3.3 voltin jännitettä. Näin ollen jännitetulon ja mikrokontrollerin väliin piti tehdä muuntopiiri, jotta vältetään raja-arvojen ylittymiseltä sekä komponenttien rikkoontumisilta. Tässä tapauksessa päätettiin käyttää yksinkertaista jännitteen jakopiiriä, jolla saatiin tiputettua jännite turvalliselle tasolle mikrokontrolleria varten (Kaava 6). Vastusarvoiksi valittiin 510 ohmia sekä 2000 ohmia.

Kaava 6. Jännitteen jakaja.

$$V_{out} = \frac{V_s \times R_2}{(R_1 + R_2)}$$

$$V_{out} = \frac{14 \times 510}{(2000 + 510)}$$

$$V_{out} \approx 2,845 \text{ V}$$

Tulokseksi saatiin neljäntoista voltin jännitteellä alle kolmen voltin jännite mikrokontrollerilla. Lopuksi tarkistettiin vielä suurin sallittu tulojännite mikrokontrollerin 3,3V tulolle laskemalla saatu tulos samoilla vastusarvoilla kaavan 6 mukaisesti.

$$3.3 = \frac{V_s \times 510}{(2000 + 510)}$$

$$V_s \approx 16,2 \text{ V}$$

Jännitteen mittauspiireihin lisättiin myös passiivinen alipäästösuodatin. Tämä oli pakollista PWM-jännitteen mittaukselle jo pelkästään pulssinleveysmodulaation toimintatavan takia. Syöttöjännitteen mittaukselle käytettiin samanlaista alipäästösuodatinta mahdollisia virransyötössä esiintyviä häiriöitä varten, vaikkei se välttämättä olisi ollut pakollista (Kuva 9). Haluttu taajuus  $f_c$  suodattimelle oli noin 400 hertsiä ja kondensaattorin  $C$  lähtöarvona pidettiin 100 nanofaradia, joka vaikutti olevan yleisesti käytetty mitoitus. Komponenttien arvot laskettiin kaavan 5 mukaisesti.

$$f_c = \frac{1}{2\pi RC}$$

$$400 = \frac{1}{2\pi \times R \times 0,0000001}$$

$$R \approx 3979 \Omega$$

Lähin saatavilla oleva vastus oli 3,9 kilo-ohmia, joten suodatustaajuus varmistettiin vielä laskemalla uudelleen tätä vastusarvoa hyväksi käyttäen.

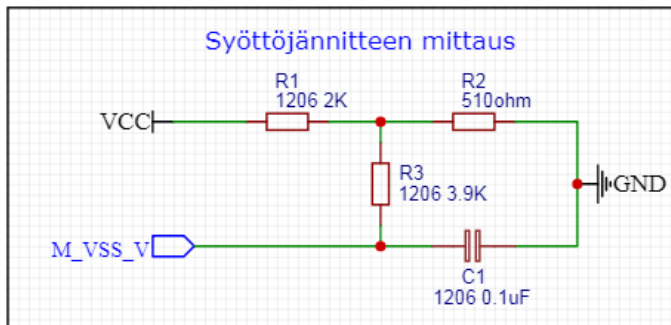
$$f_c = \frac{1}{2\pi RC}$$

$$f_c = \frac{1}{2\pi \times 3900 \times (1 \times 10^{-7})}$$

$$f_c \approx 408 \text{ Hz}$$

Passiivinen alipäästösuodatin on siis viritetty noin neljänsadan hertsin taajuuteen. Tällä tavalla saadaan nopealla taajuudella muuttuvasta PWM-signaalista tasainen jännite mikrokontrollerille luettavaksi.

Kuva 9. Kytchentäkaavio: Syöttöjännitteen mittaus.

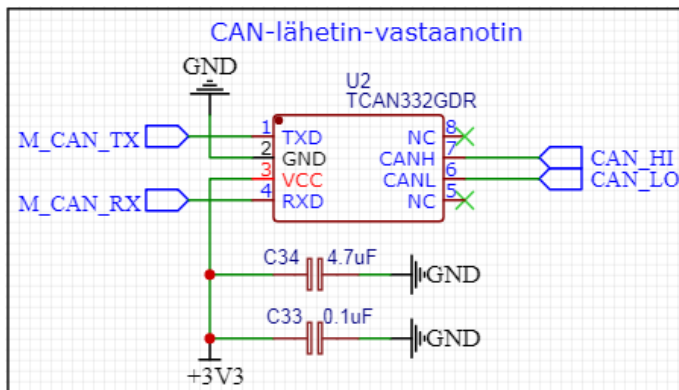


### 3.3.3 CAN-lähetin-vastaanotin (CAN-transceiver)

Vaikka mikrokontrollerissa onkin CAN-piiri, ei sitä sellaisenaan voi suoraan kytkeä CAN-väylään. Syynä tähän on muun muassa jännite-erot mikrokontrollerin 3,3V signaalitasoissa sekä CAN-väylän 12V jännitteen välillä. Tätä varten on olemassa erilaisia CAN-lähetin-vastaanottimia, jotka muuntavat signaalitasot CAN-väylän sekä mikrokontrollerin välillä. Mikrokontrolleri sekä CAN-lähetin-vastaanottimia muodostavat yhdessä CAN-noodin, eli CAN-väylään kytketyn laitteen.

CAN-lähetin-vastaanotinpiiriksi valittiin Texas Instruments:in TCAN332-piiri, jonka logiikkataso vastasi mikrokontrollerin logiikkatasoa, eli 3,3 voltia. Näin ollen se voitiin kytkeä suoraan mikrokontrolleriin (Kuva 10). Piirin virransyöttöön lisättiin valmistajan suosittelemat ohituskondensaattorit häiriönpoistoa varten.

Kuva 10. Kytentäkaavio: CAN-lähetin-vastaanotin.



### 3.3.4 FlexFuel-anturi

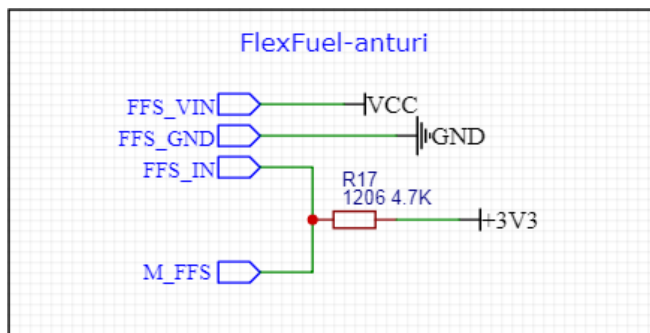
FlexFuel-anturilla mitataan polttoaineen etanolipitoisuutta (Kuva 11). Polttoaineen etanolipitoisuuden perusteella voidaan moottorinohjausyksikössä säätää polttoaineen ruiskutusmäärää. Tällä tavoin voidaan esimerkiksi kompensoida mahdollisia polttoaineen laatuvariaatioita.

Kuva 11. FlexFuel-anturi (Fuel Flex Finland, 2017).



Etanolianturi asennetaan polttoainelinjaan, jossa se mittaa läpimenevän polttoaineen etanolipitoisuuden sekä lämpötilan, ja koostaa mitatut arvot neliöaalto-signaaliin. Signaalin vaihteluväli on 50 hertsistä 150 hertsiin, ja se kertoo etanolipitoisuuden prosentteina nollasta sataan prosenttiin. Neliöaalto-signaalin aallonleveys sisältää tiedon polttoaineen lämpötilasta. Anturin signaalilähdön kytkentä mikrokontrolleriin toteutettiin lösvetovastuksen avulla (Kuva 12).

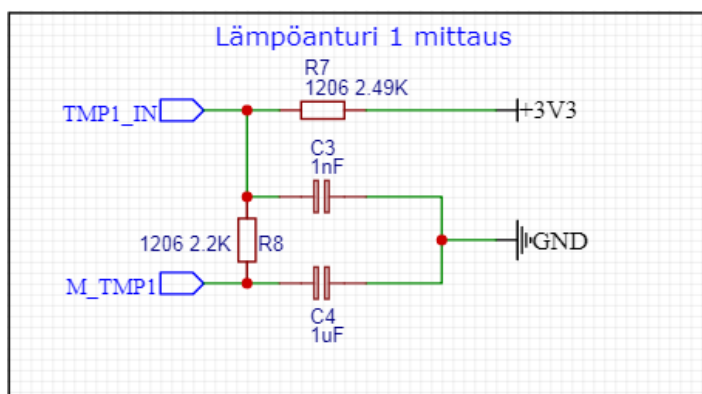
Kuva 12. KytKentäkaavio: FlexFuel-anturi.



### 3.3.5 Lämpöanturit

Lämpöantureita varten suunniteltiin mittaussiiri, jonka avulla ohjaimen voitaisiin kytkeä erinäisiä NTC-lämpöantureita (Kuva 13). Piiri koostuu jännitteen jakopiiristä, jossa on tiedetyn arvon omaava vertailuvastus. Tämän vertailuvastuksen avulla voidaan mikrokontrollerilla laskea lämpöanturilta mitattu vastusarvo ja muuntaa se lämpöanturin valmistajan taulukon (lämpötiläkäyrän) mukaiseksi lämpötilaksi. Lisäksi lämpöanturin mittaussiiriin lisättiin kondensaattori tasaamaan mahdollisia ulkoisista häiriöistä johtuvia jännitepiikkejä ja alipäästösuodatin, jonka viritystaajuus laskettiin noin 70 hertsiksi. Näytteenottotaajuuden ollessa alle 50 hertsiä olisi suodattimen voinut mitoittaa paljon matalammallekin taajuudelle, mutta mahdollisen signaalin vääristymisen takia taajuus päätettiin jättää korkeammaksi.

Kuva 13. KytKentäkaavio: Lämpöanturi.



### 3.3.6 Analogiatulot

Analogiatulot haluttiin määritellä 0–5 voltin alueelle, koska suurin osa autoihin tarkoitetuista antureista toimivat tällä alueella. STM32-mikrokontrollerin ADC-muuntimen suurin sallittu referenssijännite on kuitenkin vain 3,6 volttia, joten analogiatulojen signaalitasot piti muuntaa mikrokontrollerille sopiviksi. Tätä varten päätettiin käyttää jännitteen jakopiiriä. Jännitteen jakopiirissä käytettävien vastuksien tarvitsisi anturin kannalta olla mahdollisimman korkean impedanssin (vastusarvo) omaavia, jotta anturin signaalilähtö ei kuormittuisi. Mikrokontrollerin analogiatulolle suositeltavaa taas on mahdollisimman matala impedanssi. Jotta molemmat kriteerit toteutuisivat, lisättiin kytkentään operaatiovahvistin. Operaatiovahvistimen korkea tuloimpedanssi ei kuormita kytketyn anturin signaalilähtöä, eikä operaatiovahvistimelta lähtevä matalan impedanssin signaali kuormita mikrokontrollerin ADC-muunninta. Operaatiovahvistin kytkettiin jänniteseuraaja-tyyppisesti, ja jännitteen jakopiiri tehtiin mikrokontrollerin puolelle (Kuva 14). Operaatiovahvistinpiiriksi valittiin LM358-piiri, joka sisältää kaksi operaatiovahvistinta integroituna yhteen koteloon. Suunnittelussa hyödynnettiin mikropiirivalmistajan ohjeita (Microchip Technology, 2008, s.14). Analogiatulopiireihin on suositeltavaa lisätä ylijännitesuojaus, jotta tulon mahdollinen ylijännite ei pystyisi vahingoittamaan laitetta. Tulon ylijännite voi johtua esimerkiksi anturin rikkoutumisesta tai väärinkytkenästä. Tämän ohjaimen ensimmäiseen kehitysversioon ei toteutettu ylijännitesuojausta.

ADC-muuntimen referenssijännitteenä päätettiin käyttää mikrokontrollerin sisäistä jännitereferenssiä, joka on 3,3 volttia. Näin ollen jännitteen jakopiirin vastusarvot laskettiin näitä arvoja hyväksikäyttäen. Jakopiirin vastuksina päätettiin käyttää standardivastuksia, joiden toleranssi on 1%. Vastuksen  $R_1$  arvona käytettiin 332 ohmia.

$$V_{out} = \frac{V_s \times R_2}{(R_1 + R_2)}$$

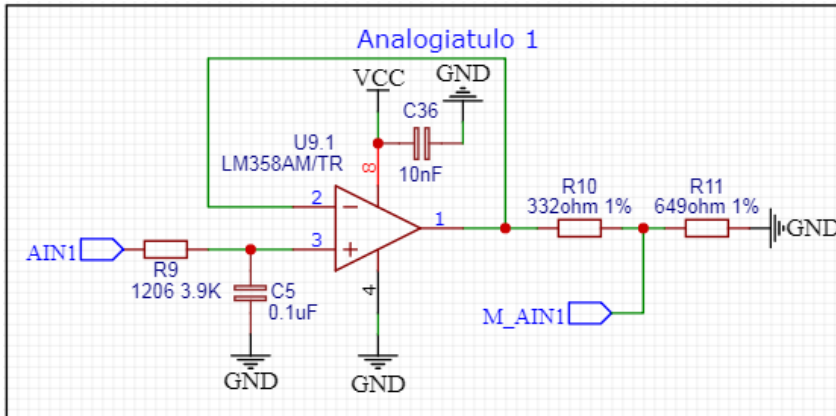
$$3,3 = \frac{5 \times R_2}{(332 + R_2)}$$

$$R_2 \approx 644 \Omega$$



Laskemalla saatiin  $R_2$  arvoksi 644 ohmia, joten päätettiin käyttää lähimmäksi osuvaa 1% standardivastusta, jonka arvo on 649 ohmia. Signaalin häiriönsuodatukseen käytettiin samaa RC-piiriä kuin jännitteenmittauksessakin, eli noin 400 hertsin alipäästösuodatinta, joka sijoitettiin operaatiovahvistimen tulopuolelle.

Kuva 14. Kytchentäkaavio: Analogiatulo.



ADC-muuntimen tarkkuus  $ADC_{res}$  STM32-mikrokontrollerissa on 12 bittiä, jolloin nolajännitteen ja referenssjännitteen  $V_{adc}$  välillä on 4096 mahdollista arvoa. Näin ollen mittaustarkkuus  $V_{step}$  on alle 1 millivoltia.

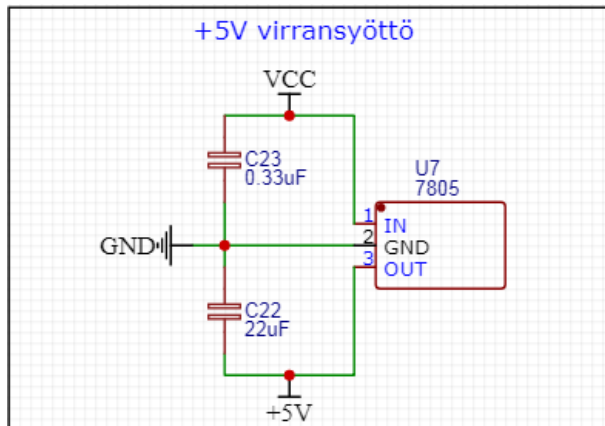
$$V_{step} = \frac{V_{adc}}{ADC_{res}}$$

$$V_{step} = \frac{3,3}{4096}$$

$$V_{step} \approx 0,0008 \text{ V} \approx 0,8 \text{ mV}$$

Analogiatuloja varten tarvittiin ohjaimen kytkettäville antureille sekä operaatiovahvistimille 5 voltin syöttöjännite. Tätä varten kytkentään lisättiin sopivanlainen jänniteregulaattori (Kuva 15). Jänniteregulaattorin kytkentä tehtiin jänniteregulaattorin datalehden esimerkin mukaisesti.

Kuva 15. Kytkentäkaavio: +5V virransyöttö.



### 3.3.7 Virran mittaus

PWM-lähdön kuluttamaa virtaa haluttiin mitata, jotta virrankulutusta voitaisiin seurata mikrokontrollerin ohjelmassa. Tietyn raja-arvon ylittyessä voitaisiin ryhtyä toimenpiteisiin, jolloin pystyttäisiin välttämään ylikuormitustilanteet, jotka saattaisivat vaurioittaa ohjaimen elektroniikkaa. Mittauksen avulla voitaisiin myös laskea PWM-lähdön ottama sähköteho wateissa, kun tiedetään myös jännite (Kaava 7).

Kaava 7. Sähköteho.

$$P_{out} = U_{out} \times I_{out}$$

Virran mittaamiseen on tarjolla lukematon määrä vaihtoehtoja. Tässä tapauksessa tarvittiin piirilevylle sijoitettava, yli 30 ampeerin tasavirran mittaukseen kykenevä komponentti, jonka lähtösignaali olisi mikrokontrollerin ymmärtämässä muodossa. Komponentin saatavuus huomioiden päädyttiin lopulta Allegro Microsystems:in valmistamaan ACS780xLR-sarjan anturipiiriin, jota on saatavilla PSOF-pintaliitospakettina, ja jonka mittauskapasiteetti on mallista riippuen jopa 100 ampeeria jatkuvaa virtaa. Anturipiiriä on saatavilla sekä unipolaarisena (yksisuuntainen) että bipolaarisena (kaksisuuntainen). Anturipiiri tuottaa jännitelähtöön  $V_{out}$  analogisen signaalin, joka on suhteessa mitattuun virtaan  $I$ . (Allegro MicroSystems, n.d.). Tähän projektiin sopivin vaihtoehto oli unipolaarinen 0–50 ampeerin mittausalueella toimiva anturipiiri, jonka vahvistus  $S_{sens}$  on 60 millivolttia ampeeria kohden

ja nollassa jännite  $V_{start}$  0,5 voltia. Anturiin tuottama jännite voidaan laskea kaavan 9 avulla.

Kaava 8. Anturiin signaalijännite.

$$V_{out} = (S_{sens} \times I) + V_{start}$$

$$I = \frac{(V_{out} - V_{start})}{S_{sens}}$$

Anturiin valinnassa oli tärkeää ottaa huomioon mikrokontrollerin analogiatulon suurin mahdollinen tulojännite, joka on sama kuin analogi-digitaalimuuntimen referenssijännite  $V_{adc}$ , 3,3 voltia.

$$I = \frac{(3,3 - 0,5)}{0,06}$$

$$I = 45 \text{ A}$$

Näin ollen 30 ampeerin suunnitellulla maksimivirralla pystytään ongelmitta käyttämään jännitelähtöä mikrokontrollerin kanssa ilman erillistä jännitteen jakopiiriä, joka saattaisi vaikuttaa mittauksen tarkkuuteen. Aiemmassa kappaleessa lasketun mikrokontrollerin analogiatulon mittaustarkkuutta  $V_{step}$  sekä anturiin vahvistusta  $S_{sens}$  hyödyntäen voidaan myös määrittää Kaava 9, jonka avulla voidaan selvittää virran mittauksen tarkkuus  $I_{step}$  ampeereissa. Näillä komponenteilla päästään melkein milliampeerin tarkkuuteen.

Kaava 9. Virranmittauksen tarkkuus.

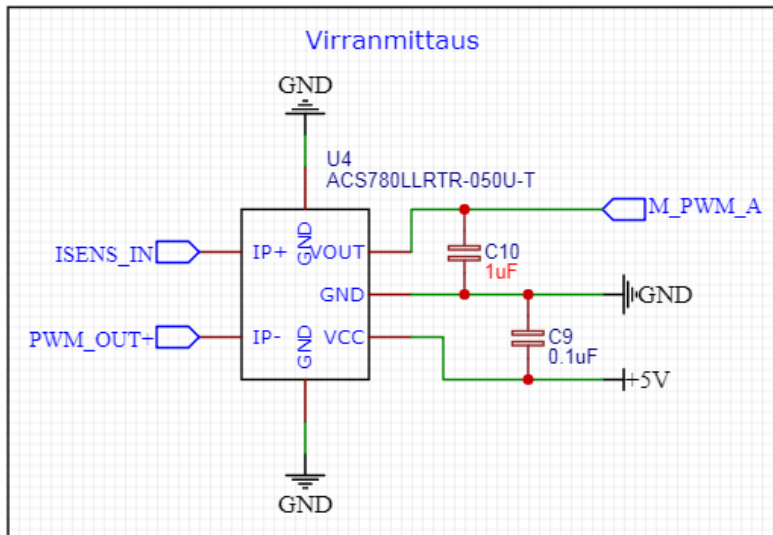
$$I_{step} = \frac{I}{(S_{sens} / V_{step})}$$

$$I_{step} = \frac{1}{(0,06 / 0,0008)}$$

$$I_{step} \approx 0,013 \text{ A}$$

Anturipiirin kytkennässä hyödynnettiin suurelta osin valmistajan datalehden esimerkkikytkentää, mutta häiriöpoistokondensaattorin arvoa suurennettiin tasaisemman signaalin aikaansaamiseksi (Kuva 16).

Kuva 16. Kytkentäkaavio: Virranmittauspiiri.

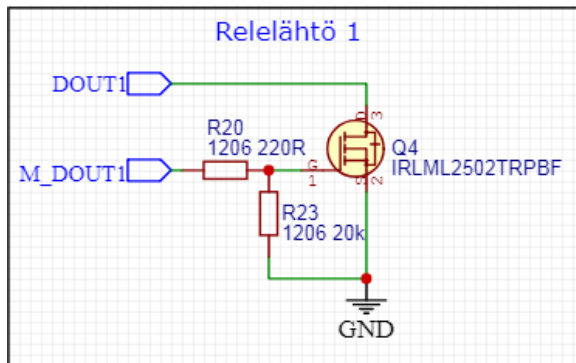


### 3.3.8 Relelähdöt

Jotta ohjaimella olisi mahdollista ohjata erinäisiä ulkoisia toimilaitteita, päätettiin ohjaimeen lisätä relelähdepiiri (Kuva 17). Releen kelan ottama virta riippuu täysin releen ominaisuuksista, yleinen 12 voltin autorele asettuu 150–200 milliampeerin väliin (12 Volt Planet, 2021.). Tässä työssä käytetyn STM32-mikrokontrollerin digitaalilähdön maksimivirta on 25 milliampeeria lähtöä kohden. Mikrokontrollerin ja ulkoisen toimilaitteen väliin tarvittiin komponentti, joka sallisi suuremman virran ohjauksen pienellä ohjausvirralla, joten päädyttiin käyttämään transistoria. Transistorin mitoituksessa oli otettava huomioon mikrokontrollerin digitaalilähdön alhainen 3,3 voltin jännitetaso. Tavallinen transistori ei välttämättä jaksaisi kytkeytyä täysin johtavaan tilaan niin matalalla jännitteellä. Lopulta päädyttiin IRLML2502-transistoriin, joka on logiikkatasolla toimiva N-kanava MOSFET-transistori. Kyseinen transistori on SOT-23-pintaliitoskomponentti joka on täysin johtava jo 3,3 voltin jännitteellä. Transistorin ohjauspiiriin lisättiin sarjavastus rajoittamaan transistorin ohjauksen ottamaa virtaa sekä alavetovastus, jolla varmistetaan, ettei transistori jää johtavaan tilaan.

Tällä tavalla toteutettuna relelähtö on maadoittava, eli kytketylle toimilaitteelle (esimerkiksi releen kela) kytketään jatkuva positiivinen jännite, ja kelan maadoitus kulkee relelähdön kautta. Kun transistori ajetaan johtavaan tilaan, kela maadoittuu, virta kulkee kelan läpi, ja rele aktivoituu.

Kuva 17. Kytchentäkaavio: Relelähtöpiiri.



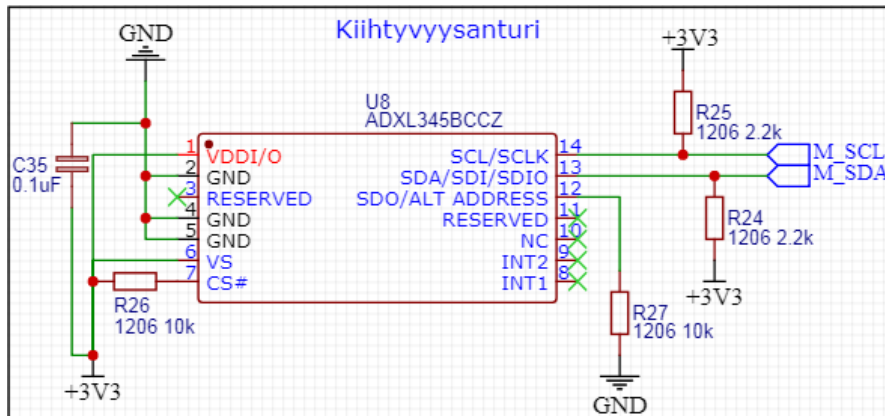
### 3.3.9 Kiihtyvyyssanturi

Ohjaimella haluttiin mitata ajoneuvoon kohdistuvia voimia eri ajotilanteissa. Tätä varten päätettiin käyttää kiihtyvyyssanturia, joka rekisteröi kiihtyvyyden kolmella eri akselilla (kolmiulotteisesti) ja välittää nämä tiedot mikrokontrollerille.

Tässä työssä päädyttiin käyttämään ADXL345-kiihtyvyyssanturia, joka on 3,3 voltin käyttöjännitteellä toimiva kolmen akselin ja maksimissaan 16 G:n kiihtyvyyksiä mittaava anturi. Anturin kommunikointi mikrokontrollerin kanssa on väylätyyppinen, ja valittavissa on SPI- sekä I2C-väylävaihtoehdot. Anturin pakkaustyyppi on 14-pinninen LGA ja se on pintaliitoskomponentti. (Analog Devices, 2015.)

Kiihtyvyyssanturin kytkentä tehtiin datalehden suositusten mukaisesti, ja kommunikaatiotavaksi valittiin I2C-väylä kytkemällä piirin CS-pinni käyttöjännitetasoon (Kuva 18).

Kuva 18. KytKentäkaavio: Kiihtyvyyssanturi.

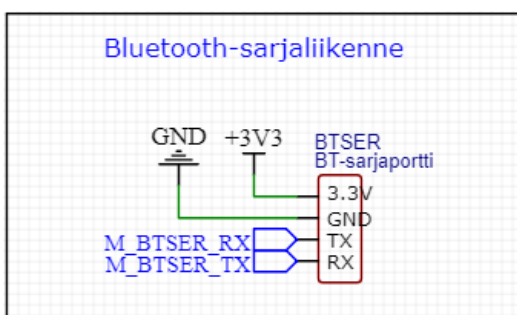


### 3.3.10 Bluetooth-sarjaliikenne (HC-05)

Ohjaimen konfigurointia varten haluttiin kehittää mahdollisimman käyttäjäystävällinen tapa, joka kuitenkin olisi helppo toteuttaa. Bluetooth-sarjaliikenteessä yhdistyy sarjaliikenteen helppous ohjelmoinnin kannalta ja langattomuus, joka on käyttäjää helpottava ominaisuus. Yksinkertaisuuden takia päätettiin käyttää valmista moduulia, jota kutsutaan yleisesti nimellä "HC-05".

HC-05-moduuli on 3,3V jännitteellä toimiva luokan 2 bluetooth-sarjaliikennemoduuli, jonka datan maksimisiirtonopeus on 2 Mbps (Partco, n.d.–b). Moduulin kytkentä on yksinkertainen, käyttöjännitteen ja maan lisäksi tarvitaan mikrokontrollerin sarjaliikenneportilta TX- (lähetys) sekä RX- (vastaanotto) -linjat. Moduulin ohjelmointia (EN) sekä tilatietoa (STATE) varten on omat pinninsä, mutta niitä ei tarvitse kytkeä käyttöä varten. Kytkennässä pitää huomioida, että lähetys- sekä vastaanottolinjat kytketään ristiin, mikrokontrollerin TX on siis HC-05-moduulin RX (Kuva 19).

Kuva 19. KytKentäkaavio: Bluetooth-moduuli.

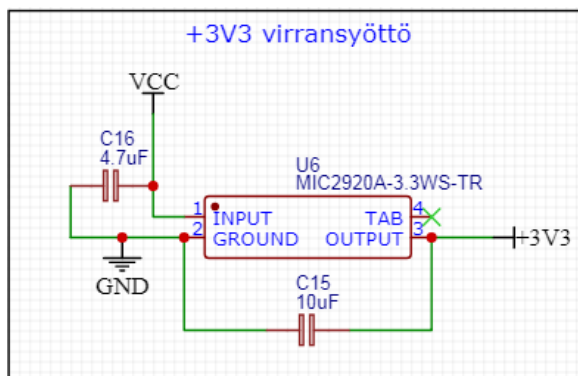


### 3.3.11 Mikrokontrolleri

Mikrokontrollerin kytkentä koostui virransyötöstä, ohjelmointiliittimestä, ulkoisista kellopiireistä sekä edellisissä kappaleissa suunniteltujen komponenttikokonaisuuksien kytkennöistä mikrokontrollerin tuloihin ja lähtöihin.

Tässä työssä käytetty mikrokontrolleri tarvitsee toimiakseen 3,3 voltin käyttöjännitteen sekä analogia-digitaalimuuntimen jännitereferenssin. Tämän käyttöjännitteen luontia varten käytettiin MIC2920-jänniteregulaattoria, joka luo tulojännitteestä 3,3 voltin tasajännitteen, ja jonka suurin lähtövirta on 400 milliampeeria. Jänniteregulaattorin kytkentä tehtiin datalehden mukaisesti, jonka lisäksi tulojännitepuolelle lisättiin kondensaattori häiriöiden poistoa varten (Kuva 20).

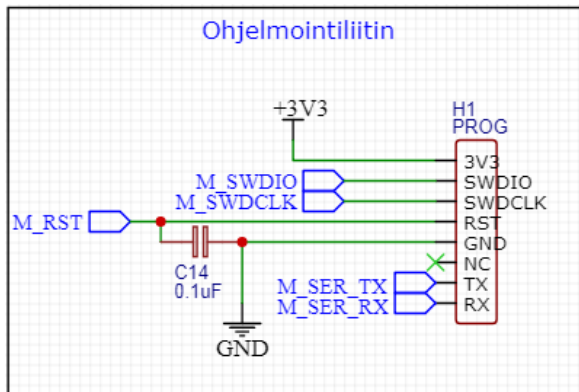
Kuva 20. Kytkenäkaavio: +3V3 virransyöttö.



Mikrokontrollerin ohjelmointia varten päätettiin käyttää ulkoista ohjelmointikorttia/laitetta (ST-Link v2) tai STM32-kehityskorttia, jossa on sisäänrakennettu ohjelmointipiiri.

Mikrokontrollerin ohjelmointi tapahtuu SWD-liittymän (Serial Wire Debug) kautta, ja vaatii viiden johtimen kytkennän mikrokontrollerin lähtöihin. Nämä ovat käyttöjännite (+3V3), datalinja (SWDIO), kellolinja (SWDCLK), reset (RST) sekä maa (GND). Kytkennän suunnittelussa noudatettiin tältä osin STM32:n MB1136-dokumenttia, jossa on kuvattu tässä työssä käytetyn kehityskortin kaikki kytkennät (Kuva 21). Lisäksi liittimeen lisättiin mikrokontrollerin sarjaportti UART2, jotta voidaan tarvittaessa reitittää sarjaliikenneviestit ohjelmointiympäristöön.

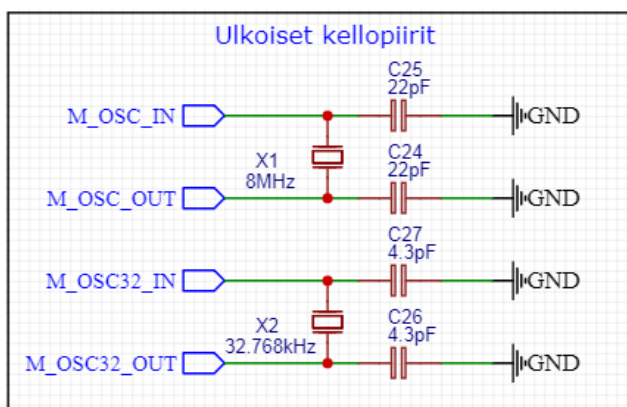
Kuva 21. KytKentäkaavio: Mikrokontrollerin ohjelmointiliitin.



Mikrokontrollerissa on liitännät kahdelle eri ulkoiselle kellopiirille, LSE (Low Speed External) sekä HSE (High Speed External). Mikrokontrollerissa on myös sisäinen kellopiiri HSI (High Speed Internal), jota voidaan käyttää, jos ulkoisia piirejä ei ole kytketty. Mbed OS määrittää oletusarvoisesti mikrokontrollerin käyttämään ensisijaisesti ulkoista 8 megahertsin taajuudella toimivaa kellopiiriä. (mBed OS, 2021–a) STM32-kehityskorteissa ulkoisen kellopiirin sijasta käytetään sisäänrakennetun ohjelmointipiirin kellotaajuuslähtöä kellopiirinä mikrokontrollerille, ja näin ollen kehityskortilla ei ole erillistä HSE-kellopiiriä.

Tätä mikrokontrolleria varten suunniteltiin sekä HSE- että LSE-kellopiirit, jotta voitaisiin tarpeen vaatiessa muuttaa komponentteja ja konfiguraatiota suunnittelematta piirilevyä uudelleen (Kuva 22).

Kuva 22. KytKentäkaavio: Ulkoiset kellopiirit.





## 4 Ohjaimen rakentaminen

### 4.1 Kotelo

Ohjaimen rakennusprosessi aloitettiin valitsemalla kotelo, jonka ominaisuuksien perusteella voitaisiin suunnitella sopivan kokoinen piirilevy komponentteineen. Koteloksi valikoitui alumiininen kotelo, jossa oli paikka liittimelle sekä kiinnityskorvakkeet, jotka helpottaisivat asennusta ajoneuvoon (Kuva 23). Kotelon ulkomitat ilman liittintä ovat 96x36x114mm ja sen sisälle mahtuisi suurimmillaan 98x76mm piirilevy.

Kuva 23. Alumiininen kotelo liittimellä.



Materiaaliltaan alumiinisen kotelon valinta muovisen kotelon sijaan perusteltiin alumiinin paremmalla lämmönjohtavuudella. PWM-lähdössä käytettävä tehoelektronikka tarvitsee lähes poikkeuksetta jäähdytystä, koska komponenteissa on aina tehohäviöitä, jotka muuttuvat lämmöksi. Suuret virtamäärät sekä korkeat kytkentätaajuudet lisäävät tehohäviöitä ja näin ollen komponenttien lämpökuormitusta. Yleisin tapa johtaa lämpö pois komponenteista on käyttää komponenttiin kiinnitettävää jäähdytyslementtiä, joka sitoo komponentin lämpöä itseensä ja luovuttaa sitä ympäröivään ilmaan (Kuva 24). Tämä voidaan toteuttaa passiivisesti pelkällä jäähdytyslementillä tai aktiivisesti esimerkiksi lisäämällä jäähdytyslementtiin puhallin. Nämä elementit ovat yleensä rivoitettuja alumiinikappaleita.

Kuva 24. Transistori jäähdytys-elementillä (Conrad, 2020.).



Tässä tapauksessa kotelossa ei ollut jäähdytys-elementeille tilaa, joten päätettiin kokeilla toteutustapaa, jossa kotelo toimii jäähdytys-elementtinä tehoelektronikan komponenteille. Tämä toteutustapa vaatii erityistä huomiota komponenttien sijoittelussa piirilevyllä, jotta aikaansaadaan suurin mahdollinen pinta-ala ja näin ollen paras lämmönjohtavuus komponentin ja kotelon välille. Lisäksi tulee huomioida mahdollinen komponentin jäähdytyspintaan johdettu potentiaali, jolloin jäähdytyspinnan ja -elementin väliin tarvitaan lämmönjohtavuuden lisäksi sähköjohtavuutta estävä kerros.

Kotelon toimitussisältöön kuului piirilevyllä sopiva liitin sekä kaapeloitava vastakappale. Liittimessä oli 24 pinniä, joista kuusi kappaletta olivat kooltaan isompia kuin muut, joten ne kestäisivät suurempia virtamääriä. Isojen pinnien suurin mahdollinen kaapelikoko oli kuitenkin vain  $2,5\text{mm}^2$  eikä näin ollen riittävä 30 ampeerin virtamäärälle. Tämä rajoitus oli otettava huomioon piirilevyn pinnijärjestystä suunniteltaessa.

## 4.2 Koekytkenät

### 4.2.1 Kytkenät

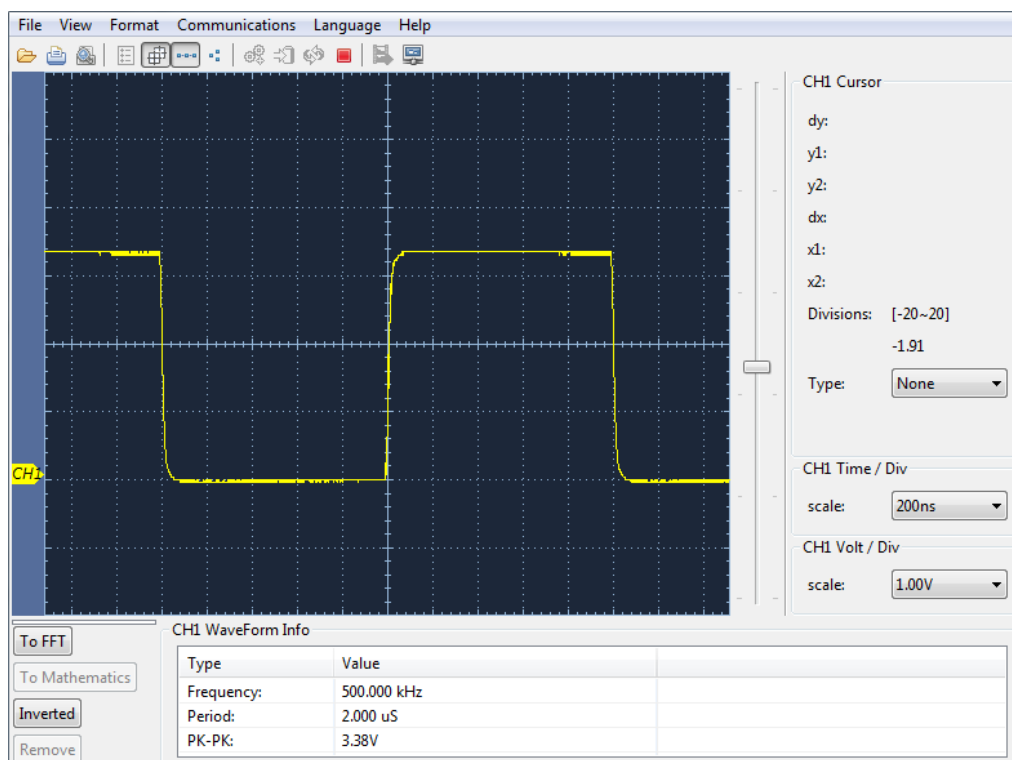
Ennen piirilevyn piirtämistä suoritettiin koekytkenät niiden komponenttikokonaisuuksien osalta, joihin oli saatavilla kytkentälevylle sopivat pintaliitoskomponenttien vastineet. Komponenttikokonaisuudet rakennettiin kytkentälevylle yksi kerrallaan. Mikrokontrollerille kirjoitettiin kytkentäkohtaisia ohjelmalohkoja, joilla koekytkenöjä voitiin testata. Koekytkenöjen testauksessa käytettiin lisäksi digitaalioskilloskooppia ja yleismittaria. Koekytkenää ei voitu puuttuvien komponenttien takia toteuttaa CAN-lähetin-vastaanotinpiirille, kiihtyvyyssanturille, relelähdöille eikä virran mittaukselle. Bluetooth-

moduulin kytkentä mikrokontrolleriin ei vaatinut erillisiä komponentteja, joten sitä ei koettu aiheelliseksi kokeilla.

#### 4.2.2 PWM-lähtö

PWM-lähdön koekytkentää varten kehityskortin mikrokontrollerin digitaalilähtöön kytkettiin digitaalioskilloskooppi. Kehityskortille kirjoitettiin ohjelma, joka aktivoi digitaalilähtöä 50 prosentin pulssisuhteella eri taajuuksilla. Ohjelmassa käytettiin mBed OS:n rajapintakirjaston *PwmOut* -luokkaa, joka tarjoaa ohjelmoijalle yksinkertaiset komennot pulssinleveysmodulaation aikaansaamiseksi (mBed OS, 2021–c). Kokeessa todettiin PWM-signaalin olevan oikeanlainen neliöaalto, jonka jännite vaihteli nollan ja 3,3 voltin välillä (Kuva 25). PWM-lähdön suurin toimiva taajuus oli 500 kilohertsiä. PWM-ohjaimelle sopiva taajuus jäisi reilusti tätä maksimitaajuutta matalammaksi. Lopullinen käytettävä taajuus selviäisi testauksen perusteella.

Kuva 25. PWM-lähdön signaali oskilloskoopin lukemana.



### 4.2.3 Jännitteen mitta

Jännitteen mittausspiirin koekytkentä suoritettiin kytkentäkaavion mukaisesti metallikalvovastuksiin perustuvalla jännitteenjakajalla sekä passiivisella alipäästösuodattimella (Kuva 9). Koe aloitettiin mittaamalla koekytkennän vastusarvot (Taulukko 2). Vastusarvot mitattiin kahdella laadukkaalla yleismittarilla, jotta mittaustulosta voitaisiin pitää luotettavana.

Taulukko 2. Koekytkennän vastusarvot.

Vastus	Suunniteltu arvo	Mitattu arvo	Ero
R1	2 k $\Omega$	1,983 k $\Omega$	0,9 %
R2	510 $\Omega$	512 $\Omega$	0,4 %
R3	3,9 k $\Omega$	3,863 k $\Omega$	0,9 %

Tämän jälkeen mittausspiirin tuloon syötettiin jännitettä porrastetusti. Samanaikaisesti mitattiin mittausspiirin lähdön jännitettä. Tällä tavalla voitiin todentaa, ettei mittausspiirin lähtöjännite ylitä mikrokontrollerin analogiatulon maksimijänniterajaa. Mittaustulokset merkittiin taulukkoon 3.

Taulukko 3. Jännitemittausspiirin tulokset.

Tulojännite nimellinen	Tulojännite mitattu	Lähtöjännite mitattu
0 V	0 V	0 V
1 V	1,00 V	0,20 V
3 V	3,01 V	0,62 V
5 V	5,01 V	1,03 V
7 V	6,99 V	1,43 V
9 V	8,99 V	1,84 V
11 V	11,01 V	2,25 V
13 V	13,01 V	2,66 V
15 V	15,01 V	3,06 V

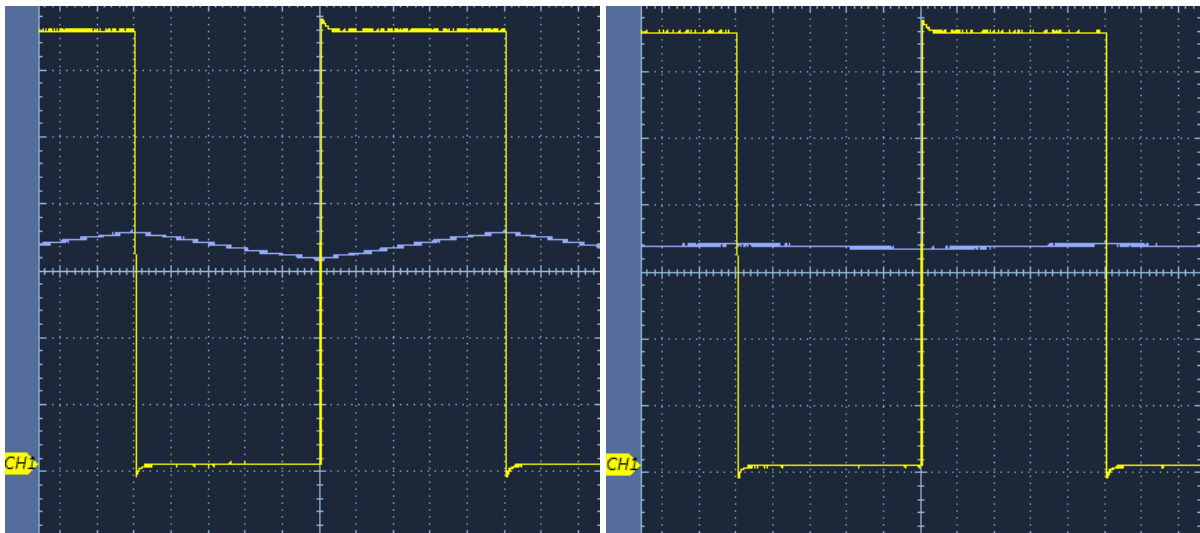
### 4.2.4 Alipäästösuodatin

Mittausspiirien alipäästösuodattimille tehtiin koekytkentä käyttäen suunnitteluosion laskennassa saatuja arvoja. Koekytkentää testattiin ajamalla suodatinpiiriin PWM-signaalia mikrokontrollerilta 20 kilohertsin taajuudella ja 50 prosentin pulssisuhteella. Tuloa sekä lähtöä mitattiin digitaalioskilloskoopilla. Kokeessa kävi ilmi, että 400 hertsin taajuudelle

viritetty alipäästösuodatin ei kykene täysin tasaamaan neliöaalto-signaalia, vaan suodatettuun signaaliin jää niin sanottua ”sahalaitaa”. Tämä ilmiö korostuu matalammilla PWM-taajuuksilla, jolloin suodattimella on enemmän aikaa reagoida PWM-signaalin värähtelyyn.

Kokeen löydösten perusteella päätettiin muuttaa PWM-suodatinpiirin komponentteja niin, että saataisiin aikaan tasaisempi signaali. Tehokkain tapa vahvistaa suodattimen toimintaa oli suurentaa kondensaattorin kapasitanssia. Kondensaattorin kapasitanssi kymmenkertaistettiin, jolloin alipäästösuodattimen taajuus pieneni kymmenkertaisesti. Tällä yksinkertaisella muutoksella suodatinpiirin lähdön jännite tasaantui merkittävästi ja tarve signaalin ohjelmalliselle suodatukselle pieneni (Kuva 26).

Kuva 26. Suodattimen tulo ja lähtö ennen muutosta sekä muutoksen jälkeen.



Havaittu ja korjattu ongelma koski lähinnä PWM-jännitteen mittaussiirä, jonka signaali on neliöaalto-tyyppinen. Muissa ohjaimen komponenttikokonaisuuksissa suodatinpiirien tulosaali on neliöaallon sijaan tasajännitteinen, ja niissä käytettyjen suodatinpiirien pääasiallinen tarkoitus on häiriöiden poistaminen.

#### 4.2.5 Analogiatulot

Analogiatulon koekytkentä toteutettiin komponenttien puuttumisen takia ilman operaatiovahvistinta pelkkää jännitteen jakopiiriä käyttäen, josta lähtösignaali ohjattiin

mikrokontrollerin analogiatuloon. Myös signaalin suodatus jätettiin koekytkennästä pois. Jännitteen jakopiirin lähtö  $V_{out}$  laskettiin tuloon kytkettävän anturin oletetulle maksimijännitteelle  $V_s = 5 \text{ V}$  Kaava 6 mukaan.

$$V_{out} = \frac{V_s \times R_2}{(R_1 + R_2)}$$

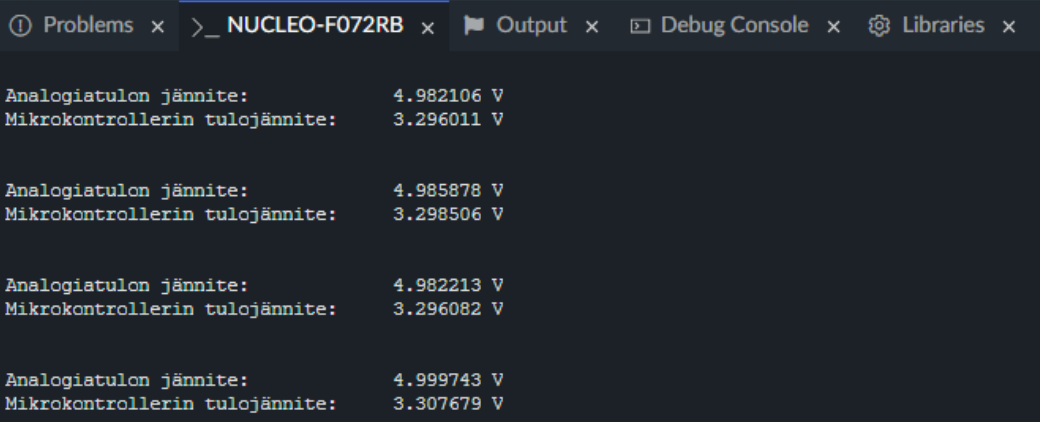
$$V_{out} = \frac{5 \times 649}{(332 + 649)}$$

$$V_{out} \approx 3,31 \text{ V}$$

Seuraavaksi jännitteenjakopiirin tuloon syötettiin jännitelähteestä 5 voltin tasajännite sekä mitattiin yleismittarilla lähtöjännite. Mittaustulos vastasi laskettua tulosta, joten piiri todettiin toimivaksi.

Tämän jälkeen mikrokontrollerille kirjoitettiin yksinkertainen ohjelma, joka sekunnin välein lukee mikrokontrollerin analogiamuuntimen arvon ja tekee tarvittavat muunnoslaskelmat. Lopuksi ohjelma tulostaa sarjaportin kautta ohjelmointikonsoliin mikrokontrollerin analogiatulon jännitteen sekä lasketun anturijännitteen (Kuva 27). Tällä tavoin voitiin todeta myös mikrokontrollerin analogiatulo toimivaksi.

Kuva 27. Analogiatulon jännite mikrokontrollerin laskemana.



```

① Problems x >_ NUCLEO-F072RB x Output x Debug Console x Libraries x

Analogiatulon jännite:      4.982106 V
Mikrokontrollerin tulojännite: 3.296011 V

Analogiatulon jännite:      4.985878 V
Mikrokontrollerin tulojännite: 3.298506 V

Analogiatulon jännite:      4.982213 V
Mikrokontrollerin tulojännite: 3.296082 V

Analogiatulon jännite:      4.999743 V
Mikrokontrollerin tulojännite: 3.307679 V

```

#### 4.2.6 Lämpöanturit

Lämpöantureiden mittauspiirin koekytkentä tehtiin suunnittelun mukaisesti käyttämällä jännitteen jakopiiriä ja tiedetyn arvon vertailuvastusta  $R_1$ . Näin oli mahdollista laskea mitatusta lähtöjännitteestä  $V_{out}$  piiriin kytketyn lämpöanturin vastusarvo  $R_2$  (Kaava 10). Koekytkennässä käytettiin NTC-tyyppistä lämpötila-anturia, jolle oli saatavilla valmistajan ilmoittama vastusarvotaulukko eri lämpötiloilla. Vertailuvastuksena käytettiin kahta vastusta sarjaan kytkettynä oikeanlaisen vastuksen puuttumisen takia. Vastusarvot sekä piirin tulo- ja lähtöjännite tarkistettiin yleismittarilla, jonka jälkeen lämpöanturin vastusarvo laskettiin sekä todettiin anturista mittaamalla oikeaksi.

Kaava 10. Vastusarvo jännitteen jakopiirissä.

$$R_2 = \frac{V_{out} \times R_1}{V_s - V_{out}}$$

$$R_2 = \frac{V_{out} \times R_1}{V_s - V_{out}}$$

$$R_2 = \frac{1,416 \times 2623}{3,3 - 1,416}$$

$$R_2 \approx 1971 \Omega$$

Mikrokontrollerille kirjoitettiin ohjelmalohko, joka lukee analogiatulon arvoja, muuntaa ne jännitteeksi sekä laskee tiedetyn vertailuvastuksen arvon ja jännitteen avulla lämpötila-anturin vastusarvon (Kuva 28). Tätä laskettua vastusarvoa hyväksikäyttäen ohjelmaan voitiin lisätä anturivalmistajan vastusarvotaulukko, jonka avulla vastusarvo muunnettiin lämpötilaksi lineaarisen interpoloinnin avulla (Taulukko 4).

Taulukko 4. NTC-lämpöanturin muuntotaulukko.

<b>Vastusarvo <math>\Omega</math></b>	45340	15473	5894	2501	1175	596	323	187	113	89
<b>Lämpötila <math>^{\circ}\text{C}</math></b>	-40	-20	0	20	40	60	80	100	120	130

Kuva 28. Lämpöanturitulon laskenta mikrokontrollerin ohjelmassa.

```

Problems x Output x Debug Console x Libraries x >_ NUCLEO-F072RB x
Lämpöanturitulon jännite: 1.417289 V
Lämpöanturin vastusarvo: 1974.573486 ohmia
Lämpöanturin lämpötila: 26.630056 °C

Lämpöanturitulon jännite: 1.416459 V
Lämpöanturin vastusarvo: 1972.546875 ohmia
Lämpöanturin lämpötila: 26.655581 °C

Lämpöanturitulon jännite: 1.415629 V
Lämpöanturin vastusarvo: 1970.522217 ohmia
Lämpöanturin lämpötila: 26.681080 °C

Lämpöanturitulon jännite: 1.417289 V
Lämpöanturin vastusarvo: 1974.573486 ohmia
Lämpöanturin lämpötila: 26.630056 °C

```

Mikrokontrollerin ohjelman laskemia arvoja verrattiin yleismittarilla mitattuun lämpöanturin vastusarvon sekä lämpömittarilla mitattuun huonelämpötilaan. Näiden arvojen vertailujen perusteella koekytkentä sekä ohjelmalohko todettiin toimiviksi.

#### 4.2.7 FlexFuel-anturi

FlexFuel-anturille tehtiin koekytkentä suunnittelun mukaisesti. Tällä tavoin anturin signaalilähdöstä saatiin ulos 3,3 voltin neliöaalto-muotoinen signaali, joka voitiin kytkeä mikrokontrollerin digitaalituloon. Mikrokontrollerille kirjoitettiin ohjelmalohko, joka laskee anturisignaalin pulssisuhteen  $p_{sens}$  sekä taajuuden  $f_{sens}$  keskeytysrutiinien ja ajastintoiminnoiden avulla. Näitä tietoja hyödyntämällä voidaan laskea myös polttoaineen etanolipitoisuus  $E_{cont}$  sekä lämpötila  $T_{fuel}$  (Kaava 11 & Kaava 12). Laskettuaan arvot ohjelma tulostaa tulokset konsoliin (Kuva 29).

Kaava 11. Etanolipitoisuus prosentteina FlexFuel-anturisignaalin taajuudesta.

$$E_{cont} = f_{sens} - 50$$

Kaava 12. Polttoaineen lämpötila FlexFuel-anturisignaalin pulssisuhteesta.

$$T_{fuel} = (41,25 \times p_{sens}) - 81,25$$



Koekytkenän toimivuus varmistettiin digitaalioskilloskoopilla, jonka antamien arvojen perusteella polttoaineen etanolipitoisuus ja lämpötila laskettiin käsin. Koe tehtiin ensin ilman polttoainetta sekä arvojen ja laskelmien todentamisen jälkeen etanolipolttoaineella (RE85), jonka etanolipitoisuuden vaihteluväli on valmistajan mukaan 70–85 prosenttia. Koekytkenä sekä ohjelmalohko todettiin mittaustulosten perusteella toimiviksi.

Kuva 29. FlexFuel-anturin signaaliarvot eri polttoainepitoisuuksilla.

Parameter	0.037529 % Ethanol	84.390533 % Ethanol
FlexFuel-signaalin taajuus:	50.040035 Hz	134.390533 Hz
FlexFuel-signaalin pulssinleveys:	2.606000 ms	2.408000 ms
Polttoaineen Etanolipitoisuus:	0.037529 %	84.390533 %
Polttoaineen lämpötila:	26.857500 °C	18.640503 °C

Parameter	0.037529 % Ethanol	84.444748 % Ethanol
FlexFuel-signaalin taajuus:	50.037529 Hz	134.462814 Hz
FlexFuel-signaalin pulssinleveys:	2.605000 ms	2.407000 ms
Polttoaineen Etanolipitoisuus:	0.037529 %	84.444748 %
Polttoaineen lämpötila:	26.857500 °C	18.681999 °C

## 4.3 Piirilevy

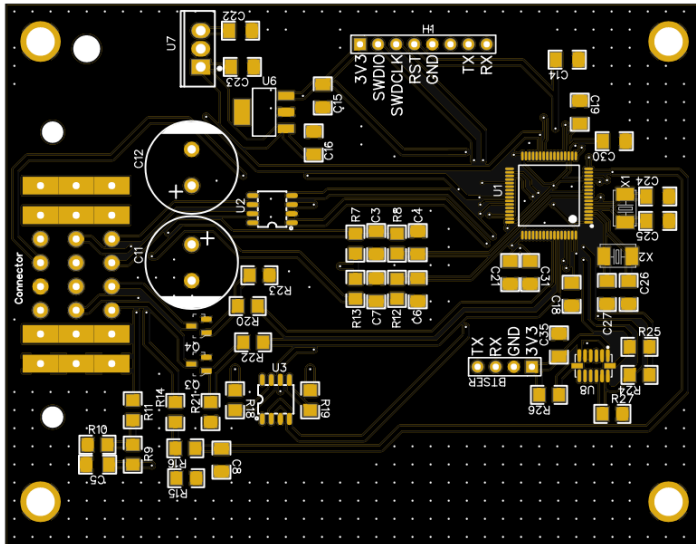
### 4.3.1 Suunnittelu

Piirilevyprototyypin suunnittelu aloitettiin luomalla suunnitteluohjelmaan tyhjä kaksipuolinen piirilevy, joka sopisi mitoiltaan kotelon sisään. Piirilevyn kulmiin mitoitettiin reiät kiinnitysruuveille. Suurin osa komponenttisuunnittelussa määritetyistä komponenteista voitiin generoida suoraan piirilevylle. Kotelon liittimelle ei ollut olemassa piirilevymallia, joten sille luotiin malli itse, jonka jälkeen se sijoitettiin piirilevylle niin, että kotelon kansi sopisi paikalleen. Komponentit sijoitettiin piirilevylle komponenttivalmistajien suositusten mukaisesti sekä yleisiä elektroniikkasuunnittelun lainalaisuuksia noudattaen. Kokoonpanon helpottamiseksi passiivikomponenteissa (vastukset ja kondensaattorit) pyrittiin käyttämään yhtä kokoluokkaa.

Piirilevyn johteiden leveys komponenttien välillä mitoitettiin tarvittavan virran mukaan. Tämä oli ensisijaisen tärkeää virransyötön komponenteissa, jotta ylikuumentumisesta

johtuvaa vikaantumista ei tapahtuisi. Piirilevy-suunnittelun valmistuttua ensimmäinen piirilevy-prototyyppi teetettiin suunnitteluohjelmiston tarjoamaa tuotantokanavaa hyväksikäyttäen (Kuva 30).

Kuva 30. Mallinnuskuva valmiista piirilevystä.



#### 4.3.2 Komponentit

Piirilevyn kokoonpanoon tarvittavien komponenttien, kuten mikrokontrollerin ja kiihtyvyysanturin, saatavuus oli suunnitteluvaiheessa tarkistettu ja todettu hyväksi. Tilanne oli kuitenkin tilausvaiheeseen mennessä muuttunut täysin. Käynnissä olleen maailmanlaajuisen pandemian takia elektroniikan komponenteissa oli saatavuusongelmia, eikä yhdeltäkään maailmanlaajuiselta elektroniikkatukkurilta ollut saatavilla kaikkia tarvittavia komponentteja. Lisäksi saatavilla olevien komponenttien hinnat olivat nousseet pahimmillaan jopa viisinkertaisiksi.

Ohjaimen tärkeintä komponenttia, mikrokontrolleria, ei ollut saatavilla yhdeltäkään toimittajalta. Asiaa tutkittiin ja selvitettiin mikrokontrollerivalmistajan vaihtoehtoiset mallit, mutta niitäkään ei ollut saatavilla. Ongelma ratkottiin hankkimalla toinen STM32-kehityskortti, josta mikrokontrolleri juotettiin irti ja siirrettiin PWM-ohjaimen piirilevylle. Muut ohjaimen komponentit tilattiin kahdelta eri toimittajalta.

### 4.3.3 Kokoonpano ja testaus

Kokoonpanossa käytettiin pintaliitoskomponenttien juottamiseen kuumailmapuhallinta sekä juotospastaa. Läpilevyn komponentit (THT) juotettiin tavallisella juottimella käyttäen juotostinaa. Mikrokontrollerin 0.5 mm johdinjako oli käsin tehtävänä juotoksena haasteellinen, mutta onnistui lopulta juotinta sekä juoksuteainetta (flux) käyttäen. Juotosten laatu varmistettiin visuaalisella tarkastuksella digitaalimikroskoopin avulla sekä yleismittarilla mittaamalla.

Testausta varten valmiin ohjaimen ohjelmointiliitäntään kytkettiin STM32-kehityskortin sisäänrakennetulta ohjelmointipiiriltä (ST-Link) ohjelmointijohtimet, sarjaliikenne sekä virransyöttö. Ohjaimelle lähetettiin onnistuneesti koekytkenöissä käytettyjä ohjelmalohkoja, mutta sarjaporttiin ei tulostunut mitään. Ongelmaa tutkiessa kävi ilmi, ettei mikrokontrolleri jostain syystä käynnistänyt ohjelmalohkoa.

Mikrokontrolleri tarkistaa käynnistysvaiheessa BOOT0-tulon, ja suorittaa tulon arvosta (low/high) riippuen joko käyttäjäohjelman, tai muita toimenpiteitä, kuten esimerkiksi alkulatausohjelman (bootloader). Ongelman aiheuttajaksi paljastui piirilevyiltä puuttuva kytkentä mikrokontrollerin BOOT0-tulolle. Tämä kytkentä oli jäänyt suunnitteluvaiheessa huomioimatta. Vika korjattiin lisäämällä piirilevylle hyppylanka BOOT0-tulon ja maajohtimen väliin, jolloin mikrokontrolleri valitsee käynnistymisvaiheessa käyttäjäohjelman suorituksen. Tämän korjauksen jälkeen ohjain suoritti ladattuja ohjelmalohkoja onnistuneesti sekä tulosti sarjaporttiin dataa.

## 4.4 Ohjelma

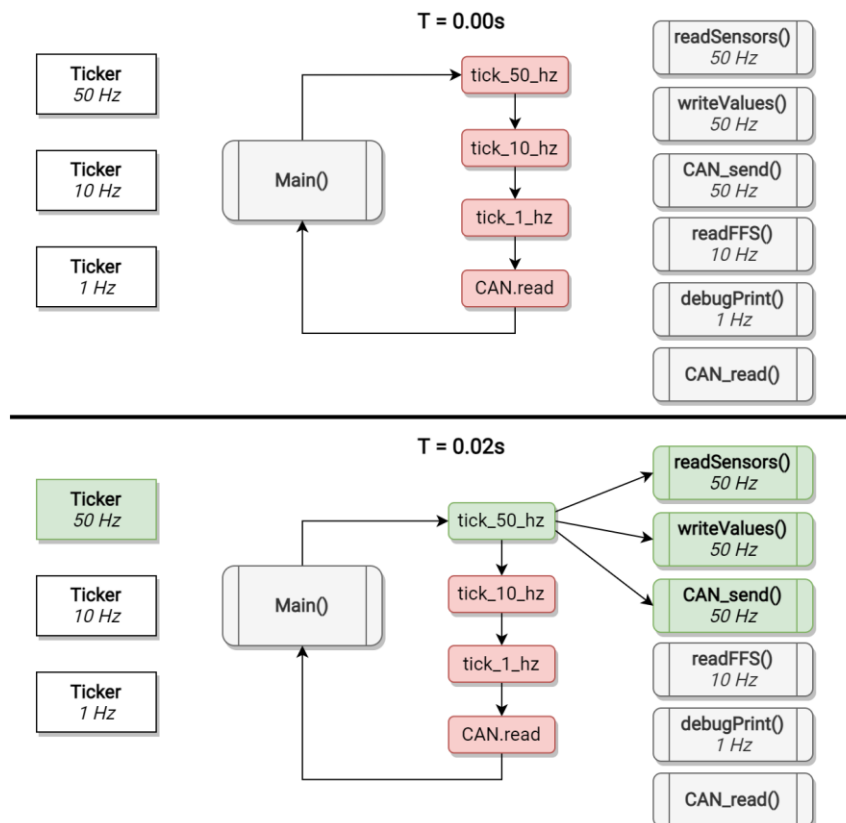
Mikrokontrollerin ohjelman tekoa varten laadittiin ensin suunnitelma tarpeellisista ohjelmalohkoista. Ohjaimen haluttiin toimivan 50 Hz taajuudella, eli sen tulisi lukea anturidataa sekä lähettää tietoa CAN-väylään 50 kertaa sekunnissa. Tiettyjen antureiden, kuten FlexFuel-anturin kohdalla 10 Hz näytteenottotaajuus riittäisi. Ohjelmaan haluttiin luoda myös ohjelmointivaiheessa ohjelmointiporttiin tulostuvaa dataa, joka helpottaisi vianetsintää. Tälle datalle 1 Hz päivitystaajuus riittäisi hyvin. Näiden kriteereiden perusteella

ohjelmaan luotiin kolme mBed-rajapintakirjaston *Ticker*-ajastinta, jotka automaattisesti aktivoivat keskeytyspyyntöjä edellä mainituilla aikaväleillä.

Keskeytyspyynnöillä voidaan suorittaa määrättyjä funktioita. Keskeytyspyynnön keskeyttävän toimintatavan vuoksi suoritettava funktio kannattaa pitää mahdollisimman lyhyenä, jotta mikrokontrollerin muulle toiminnalle ei keskeytyksen takia aiheutuisi haittaa. Näin ollen ohjelmaan määriteltiin kolme totuusarvomuuttujaa (tosi/epätosi), jotka kertovat ohjelmalle tarpeesta suorittaa tiettyjä toimintoja, ja joiden arvoa muutetaan keskeytyspyyntöjen avulla. Pääohjelma määritettiin lukemaan näitä muuttujia jatkuvasti ja suorittamaan tarvittaessa ohjelmalohkoja muuttujien arvojen perusteella (Kuva 31).

*Ticker*-ajastimien avulla aktivoitujen ohjelmalohkojen lisäksi pääohjelmaan sisällytettiin tarkastus CAN-väylästä saapuneille viesteille. Jos viestejä on, suoritetaan viestien luentalohko. Saapuneesta CAN-viestistä olisi myös voinut aktivoida keskeytyspyynnön, jolloin mikrokontrolleri lukisi viestin heti kun se saapuu. Tässä toteutuksessa kyseistä toimintoa ei viestien pienen lukumäärän eikä alhaisen prioriteetin kannalta koettu tarpeelliseksi.

Kuva 31. Pääohjelman rakenne ja ohjelmalohkojen suoritusmalli.



Ohjaimen asetusten tallennusta varten oli tarkoitus kirjoittaa tietyt asetusarvot ohjaimen pysyvään muistiin. Näitä asetusarvoja olivat esimerkiksi CAN-väylän nopeus, PWM-lähdön taajuus ja asennetut anturit. Ongelmaksi muodostui kuitenkin pysyvän EEPROM-muistin puuttuminen mikrokontrollerista. Tämän takia asetusarvot olisi pitänyt kirjoittaa Flash-muistiin, joka olisi ollut toimiva ratkaisu, mutta monimutkaisempi sekä muistia kuluttavampi prosessi kuin EEPROM-muistin kirjoittaminen. Tämän ominaisuuden puutteen takia asetusten tallennuslohko jätettiin ensimmäisen kehitysversion ohjelmasta pois, mutta sisällytettiin kuitenkin sarjaliikenteeseen sekä säätöohjelmaan, jotta seuraavaan kehitysversioon voitaisiin lisätä ulkoinen EEPROM-muisti ja ottaa asetusten tallennus käyttöön.

#### **4.4.1 PWM-lähtö**

PWM-lähtö toteutettiin ohjelmallisesti samalla tavalla kuin koekytkenässä, mutta ohjelmalohkoon lisättiin ylivirran suojatoiminto, pehmokäynnistystoiminto sekä esikäyttötoiminto.

Ylivirran suojatoiminto lisättiin suojelemaan elektroniikkaa tilanteissa, joissa PWM-lähtö ottaa sallittua enemmän virtaa. Toiminto toteutettiin kaksiportaisena. Ensimmäisen portaan ylityessä ohjelma aktivoi ajastintoiminnon, jonka päätteeksi ottovirta tarkistetaan uudelleen. Jos virta vieläkin ylittää asetetun rajan, PWM-lähtö sammutetaan. Toisen portaan ylityessä PWM-lähtö sammutetaan heti. Molemmissa tapauksissa PWM-lähdön sammutus aiheuttaa lukitustilan, joka ei salli PWM-lähdön uudelleen aktivoimista. Tämä tarkoittaa sitä, että jos lukitustila aktivoituu, ohjain on käynnistettävä uudelleen. Portaiden arvoiksi asetettiin 30 A sekä 48 A ja ajastintoiminnon ajaksi kolme sekuntia.

Pehmokäynnistystoiminto on toiminto, joka hidastaa PWM-lähdön reagointia ohjauspyynnön äkillisesti muuttuessa. Tätä toimintoa voidaan tarvita hitaasti käynnistyvissä suurissa kuormissa, kuten esimerkiksi puhaltimissa, joiden käynnistysvirta saattaisi muuten olla liian suuri ja näin ollen aiheuttaa ylivirtasuojatoiminnon aktivoitumisen. Polttoaine- tai sähköisen vesipumpun ohjauksessa ei pyörivän massan pienuuden takia yleensä esiinny suuren käynnistysvirran aiheuttamia ongelmia.

Esikäyttötoiminto on suunnattu polttoainepumppukäyttöön. Esikäyttötoiminto aktivoi ohjaimen käynnistyessä PWM-lähtöä kolmen sekunnin ajan. Toiminnon tarkoitus on esipaineistaa polttoainejärjestelmä ennen auton käynnistystä, jolloin auton käynnistysvaiheessa polttoainejärjestelmä olisi heti toimintakuntoinen ja näin ollen auton käynnistyminen nopeutuu. Suurimmassa osassa ruiskutusjärjestelmällä varustetuista henkilöautoista käytetään vastaavanlaista toimintoa.

#### 4.4.2 Anturitulot ja relelähdöt

Antureiden lukemista varten kirjoitettiin ohjelmalohko, joka lukee mikrokontrollerin tuloja ja laskee niiden perusteella antureille arvoja. Lämpötila-antureiden sekä FlexFuel-anturin lukemisessa hyödynnettiin koekytkennöissä toimiviksi todettuja osioita. Analogiatuloille lisättiin omat muuntotaulukot, joiden avulla mitattu jännite voidaan muuttaa eri suureeksi (Taulukko 5). Analogiatuloille kirjoitettiin myös vaihtoehtoinen ohjelmalohko, jonka avulla tuloja voi käyttää analogiatulojen sijasta käyttää digitaalituloina kahden voltin kynnyksjännitteellä.

Taulukko 5. Analogiatulon muuntotaulukko eräälle paineanturille.

<b>Jännite V</b>	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5
<b>Paine kPa</b>	0	0	100	200	300	400	500	600	700	800	800

Kiihtyvyyssanturin tietojen lukemiseen käytettiin *ADXL345* mBed-kirjastoa, jonka avulla anturilta voitiin lukea tietoja yksinkertaisia komentoja käyttäen. Anturille kirjoitettiin lisäksi mikrokontrollerin käynnistymisen yhteydessä ajettava kalibrointilohko, jolla anturilta tällöin tulevat kiihtyvyyssarvot määritetään nollapisteiksi. Relelähdöille rakennettiin yksinkertainen CAN-väyläviestin lukemiseen perustuva toiminto, joka aktivoi/deaktivoi halutun relelähdön tulevan viestin sisällön (looginen yksi tai nolla) perusteella.

#### 4.4.3 CAN-viestit

CAN-viestejä varten tehtiin taulukko ohjaimen lähtevistä sekä saapuvista arvoista, ja määriteltiin viesteille osoitteet sekä paikka viestirakenteessa (Taulukko 6). Ohjaimen ydintoimintoa, PWM-lähtöä, varten haluttiin ohjausviesti, joka määritettäisiin prosenteissa.



#### 4.4.4 Bluetooth-sarjaliikenne

Bluetooth-sarjaliikennettä varten ohjelmaan luotiin toiminnot tietojen lähetyksestä sekä vastaanottoa varten. Mobiililaitteelle suunnitellun säätöohjelman kanssa kommunikointia varten luotiin ohjelmalohko, joka aktivoituu, kun säätöohjelma yhdistyy bluetooth-yhteyden avulla ohjaimen. Ohjelmalohko kommunikoi säätöohjelman kanssa lähettämällä säätöohjelmalle ohjaimen käyntiarvoja sekä tallentamalla säätöohjelman lähettämiä asetusviestejä. Tällä tavalla ohjaimen toimintoja sekä toimintaa voidaan säätöohjelman avulla määrittää. HC-05 bluetooth-moduuliin tehtiin sarjaportti-komentoja käyttäen tarvittavat määritykset moduulin konfigurointitilassa. HC-05-moduuliin määritettiin ohjainta kuvaava tunniste, ”PWM Module”, jotta se olisi helppo tunnistaa bluetooth-laitteita parittaessa. Lisäksi moduulin muut asetukset tarkistettiin ja tarvittaessa muutettiin sopiviksi ohjaimelle sekä säätöohjelmalle.

#### 4.4.5 Säätöohjelma

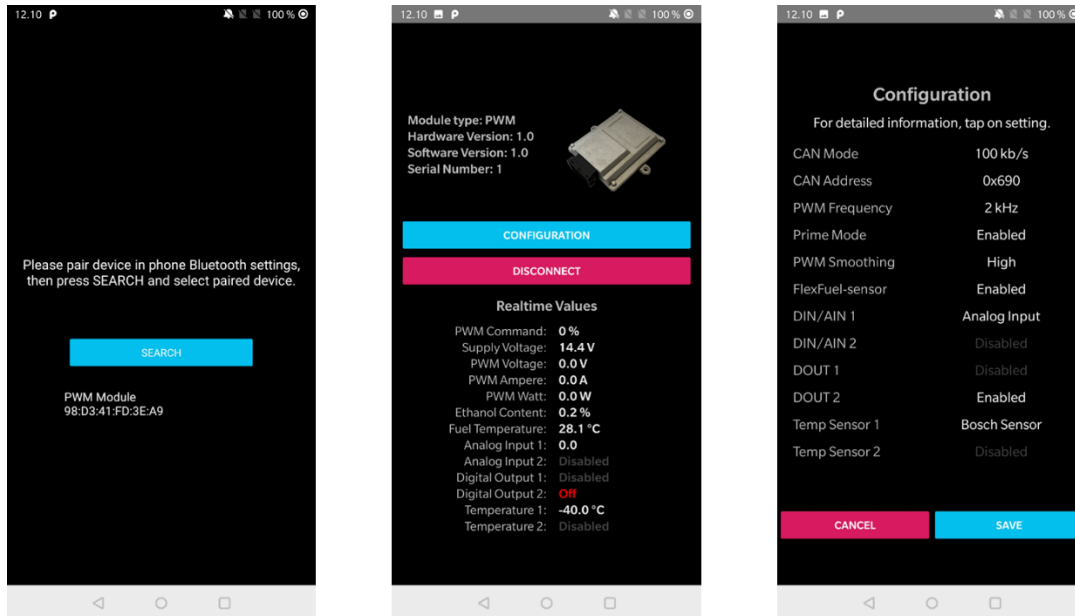
Ohjaimen kanssa kommunikointia varten olisi riittänyt mobiililaitteella käytettävä terminaaliohjelma, jonka kautta lähetetyillä komennoilla laite suorittaisi haluttuja toimintoja. Kommunikointia varten haluttiin kuitenkin luoda käyttöliittymä, joka ei vaatisi käyttäjältä minkäänlaista ammatillista tietämystä, vaan olisi itseselittyvä ja käyttäjäystävällinen. Näiden kriteereiden perusteella päätettiin kirjoittaa säätöohjelma (”app”) Androidille, joka on maailman laajimmin käytetty mobiililaittealusta. Ohjelma kirjoitettiin Android Studio -ympäristöllä käyttäen hyväksi ympäristön tarjoamia oletuskirjastoja sekä käyttöliittymän että kommunikaation osalta. Säätöohjelman käyttöliittymässä käytettiin kielenä englantia.

Säätöohjelman käyttöliittymä koostuu kolmesta eri päänäkymästä (Kuva 32). Ohjelma käynnistyy aloitusnäytön (”splash screen”) kautta aloitusnäköön, jossa käyttäjä voi etsiä saatavilla olevia paritettuja bluetooth-laitteita ja yhdistää niihin. Onnistuneen yhdistämisen jälkeen käyttäjä ohjataan päänäkymään, jossa esitetään ohjaimen tiedot (tyyppi, versio, sarjanumero) sekä reaaliaikaisesti päivittyvät käyntiarvot. Päänäkymästä käyttäjä voi joko katkaista yhteyden ohjaimen (*Disconnect*), jolloin käyttäjä palautuu aloitusnäköön, tai



siirtyä asetusnäkyyn (*Configuration*), jolloin käyttäjä pääsee muokkaamaan laitteen asetuksia.

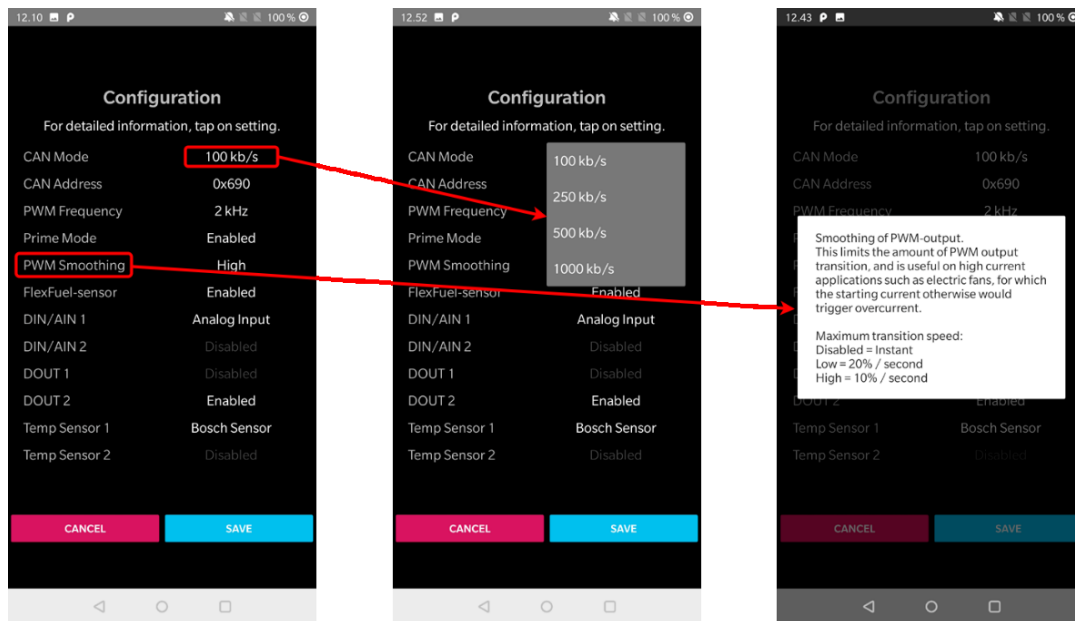
Kuva 32. Käyttöliittymän aloitusnäky, päänäky sekä asetusnäky.



Aetusnäkyssä käyttäjä voi muokata ohjaimen asetuksia sekä saada lisätietoa halutusta asetuksesta. Asetuksen muokkaus tapahtuu painamalla asetusarvoa, jolloin kyseisen asetuksen kohdalta aukeaa alavetovalikko sisältäen kaikki kyseisen asetuksen sallitut asetusarvot. Asetusten lisätiedot ovat asetustekstiä painamalla näkymän päälle aukeavia ruutuja, joissa on selitetty kyseisen asetuksen vaikutus ohjaimen toimintaan (Kuva 33). Aetusnäkyä käyttäjä voi palata päänäkymään *Cancel/Save*-painikkeilla, joilla määritetään, tallennetaanko uudet, mahdollisesti muokatut asetukset ohjaimeen.

Säätöohjelmassa kiinnitettiin erityistä huomiota ohjelman käytettävyyteen. Käyttöliittymän ulkoasu, käytetty väriteema sekä kirjasintyyli ja -koko pyrittiin valitsemaan niin, että käyttäjän olisi mahdollisimman vaivatonta käyttää ohjelmaa.

Kuva 33. Asetusnäytön alavetovalikot sekä lisätiedot.



## 5 Testaus

### 5.1 Testiympäristö

Ohjaimen ohjelman valmistuttua prototyyppiä varten rakennettiin testiympäristö. Testiympäristö koostui erinäisistä mittauslaitteista, tietokoneesta sekä CAN-väyläanalyysoijasta, joiden avulla pystyttiin seuraamaan testausta reaaliaikaisesti. PWM-lähdön kuormana käytettiin henkilöauton jäähdytinpuhaltimia, joita oli käytettävissä kahta eri mallia. Puhaltimien ottama kokonaisvirta oli suurimmillaan yli 37 ampeeria, joka ylitti reilusti suunnitellun 30 ampeerin kokonaisvirran määritteen, joten ne soveltuivat siltä osin testaukseen mainiosti.

### 5.2 CAN-väylä

CAN-väylän testaus aloitettiin tarkastamalla, että ohjain lähettää CAN-väylään viestejä. Tarkastukseen käytettiin tietokoneeseen kytkettävää CAN-väyläanalyysoijaa, jonka avulla voidaan seurata väylän liikennettä. Viestiliikenteen toimivuuden todentamisen jälkeen käytiin läpi standardit väylänopeudet 125kbit/s ja 1Mbit/s väliltä ja varmistettiin, että CAN-kommunikaatio toimii kaikilla nopeuksilla.

CAN-väyläkomennot testattiin lähettämällä ohjaimelle viestitaulukon mukaisesti eri PWM-ohjausarvoja sekä relelähtöjen tiloja. Komennot toimivat sekä näkyivät myös säätöohjelmassa. Ohjaimen väylään lähettämät anturitiedot tallennettiin ja tietojen paikkansapitävyys todennettiin viestitaulukon (Taulukko 6) avulla.

### 5.3 PWM-lähtö

PWM-lähdön testaus suoritettiin viiden minuutin sykleissä eri pulssisuhteilla sekä yhdellä että kahdella puhaltimella (Liite 1). Testaus suoritettiin neljällä eri taajuudella 1000 – 20 000 Hz väliltä. Jokainen testi kesti yhteensä 20 minuuttia. Testausta varten PWM-lähtöön kytkettiin digitaali-oskilloskooppi, pihiampeerimittari sekä yleismittari jännitteenmittausta varten. Testilaitteiston lämpökuormitusta seurattiin lämpökameran avulla.

Yhden puhaltimen testissä suurin ottovirta oli sadan prosentin pulssisuhteella 23,2 ampeeria. 13,8 voltin syöttöjännitteellä sähkötehoa oli 320 wattia. Kahdella puhaltimella suurin ottovirta oli 37,2 ampeeria, jolloin sähkötehoa oli noin 510 wattia. Kahden puhaltimen testissä ohjaimen ylivirtasuojastoiminto aktivoitui ylivirta-tilanteessa asetetun raja-arvon mukaisesti kolmen sekunnin kuluttua, joten ylivirtasuojastoiminto todettiin toimivaksi. Toiminnon raja-arvoja muutettiin tämän jälkeen väliaikaisesti kahden puhaltimen testausta varten, jotta välttyttäisiin ohjaimen ylivirtasuojauksen aiheuttamalta lukitustilalta.

Testauksessa kävi ilmi, että jäähdytinpuhaltimien poiskytkentävaiheessa ilmenevä suuri induktanssi aiheuttaa elektroniikan estodiodille kuormitusta. Estodiodi olikin näin ollen jokaisessa testausvaiheessa selkeästi eniten häviölämpöä tuottava komponentti. Lämpenemisen määrä riippui suurelta osin myös käytetystä PWM-taajuudesta. Suuremman taajuuden myötä kytkentätiheyden kasvaessa lämpöhäviötkin suurenivat.

Testauksessa huomattiin myös, että PWM-signaali aiheutti jäähdytinpuhaltimissa taajuutensa mukaista sivuääntä. Ainoa testattu äänettömäksi havaittu taajuus oli 20 000 Hz, joka on ihmisen kuuloaistille liian korkea taajuus kuultavaksi. Kuuluvilla taajuuksilla havaittu sivuääni ei millään testatulla taajuudella ollut häiritsevän voimakasta, eikä sitä luultavasti kuulisi auton moottorin käydessä lainkaan.

PWM-lähdön testauksen tulosten perusteella päätettiin ohjaimessa sallia käytettävien taajuuksia 1000 Hz ja 20 000 Hz väliltä.

#### 5.4 Tulot ja lähdöt

Kiihtyvyyssanturin toiminta varmistettiin muodostamalla ”nollatilanne”, jossa ohjain oli tasaisella pinnalla täysin suorassa. Tässä nollatilanteessa kiihtyvyyssanturin X- sekä Y-akselin kiihtyvyys oli 0 G, ja Z-akseli näytti painovoimasta johtuen noin 1 G. Tämän jälkeen ohjainta käännettiin vuorollaan kuvitteellisen kuution kaikille kuudelle sivulle ja jokaisessa asennossa tarkistettiin anturin antamien arvojen todenmukaisuus.

Analogiatulot testattiin syöttämällä tuloihin laboratoriovirtalähteestä 0–5 voltin tasavirtaa. Lämpöanturituloon kytkettiin koekytkenässäkin käytetty NTC-tyyppinen lämpöanturi. Tulojen toiminta todennettiin seuraamalla säätöohjelman arvoja sekä ohjaimen CAN-väylään lähettämien viestien sisältöä.

Relelähtöihin kytkettiin kuormaksi 5 watin hehkulankapolttimo. Lähtöjen toiminta testattiin lähettämällä ohjaimelle relelähtöjä aktivoivia/deaktivoivia CAN-väyläviestejä. Hehkulankapolttimo syttyi relelähdön aktivoituessa ja sammui relelähdön deaktivoituessa, joten relelähdöt todettiin toimiviksi.

#### 5.5 Lämmönhallinta

PWM-lähdön ollessa ohjaimen eniten lämpöhäviötä aiheuttava ominaisuus, seurattiin PWM-lähtöä testatessa tarkasti komponenttien lämpötilaa lämpökameran avulla. Piirilevy irrotettiin testauksen ajaksi kotelosta, jolloin komponenttien ainoa jäähdytys oli ympäröivä ilma. Syy irrotukseen oli komponenttien sijoittelu piirilevyn alapuolella, jolloin koteloon kiinnitettynä ei olisi ollut mahdollista seurata tarkasti yksittäisten komponenttien lämpötilaa. Näin ollen testaustilanne oli lisäksi komponenttien jäähdytyksen kannalta huonoin mahdollinen saavutettavissa oleva.

Komponenttien lämpötilat pysyivät testauksessa alle valmistajien maksimisuositusten ilman erillistä jäähdytystä, joten näin ollen komponentit toimisivat myös luotettavasti kotelossa,

jossa lämpöhäviöt johdetaan komponenteista koteloon. Tämä johtopäätös todennettiin suorittamalla erillinen kahden puhaltimen testaus käyttäen aiempien testien perusteella suurimman lämpökuormituksen aiheuttanutta taajuutta/pulssisuhdetta, jolloin viiden minuutin syklin jälkeen kotelon pintalämpötila komponenttien kohdalta oli noin 45°C. Tämä oli yli 40 astetta matalampi lämpötila kuin varsinaisissa testauksissa saavutetut yksittäisten komponenttien lämpötilat.

## **5.6 Asennus autoon**

Testauksen päätteeksi ohjaimen ensimmäinen kehitysversio todettiin puutteista huolimatta toimivaksi, ja asennettiin suunnittelun mukaisesti henkilöautoon tarkoituksena ohjata polttoainepumppua portaattomasti. Kyseisen henkilöauton ollessa etanolikäyttöinen, asennettiin autoon myös FlexFuel-anturi, joka kytkettiin ohjaimeen. Ohjaimen käyntiarvot poimittiin CAN-väylästä auton moottorinohjausyksikköön, johon myös luotiin kartta sekä lähetettävä CAN-arvo PWM-lähdön ohjausta varten. Ohjaimen tallennusmuistin (EEPROM) puutteen takia halutut säätö- sekä asetusrvot ohjelmoitiin kiinteinä arvoina ohjaimen Flash-muistiin.

## 6 Yhteenveto

Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa CAN-väyläohjattu PWM-ohjain henkilöauton polttoainepumpun portaatonta ohjausta varten. PWM-ohjaimelle määriteltiin PWM-signaalin tuottamisen lisäksi myös toissijaisia toimintoja sekä säätöohjelma, joiden tarkoitus oli tuoda lisäarvoa laitteen käyttöön.

Opinnäytetyötä tehdessä törmättiin erinäisiin haasteisiin. Komponenttien mitoituksen osalta jouduttiin tekemään muutoksia, jotta laite toimisi paremmin. Saatavuusongelmien takia jouduttiin osin hyödyntämään käytettyjä komponentteja. Myös suunnittelussa puuttumaan jääneet mikrokontrollerin kytkennät ja niiden lisääminen jälkeinpäin pintaliitoskomponentteihin aiheuttivat haasteita. Näistä haasteista kuitenkin selvittiin ja onnistuttiin rakentamaan toimiva ohjain.

Opinnäytetyön tavoite voidaan todeta toteutuneeksi laitteen ensimmäisen kehitysversion ollessa ensisijaisilta toiminnoiltaan tarkoituksenmukainen, toimiva kokonaisuus. Laitteen kehitystyötä on tarkoitus jatkaa korjaamalla opinnäytetyön aikana ilmi tulleet puutokset sekä virheet laitteen suunnittelussa. Laitteesta tulevaisuudessa kertyvä käyttökokemus tulee myös osaltaan ohjamaan laitteen kehittämisen tarvetta sekä suuntaa.

## Lähteet

12 Volt Planet. (2021). *Automotive Relay Guide*. <https://www.12voltplanet.co.uk/relay-guide.html>

Ahonen, J. Haiko, T. Salonen, K. (2016). *Elektroniikka* Helsinki: Sanoma Pro.

Allegro MicroSystems. (n.d.). *ACS780xLR High-Precision Linear Hall-Effect-Based Current Sensor IC With 200 $\mu\Omega$  Current Conductor*. <https://docs.rs-online.com/a249/0900766b815e937a.pdf>

Analog Devices. (2015). *ADXL345 Digital Accelerometer Datasheet*. <https://www.analog.com/media/en/technical-documentation/datasheets/ADXL345.pdf>

Arduino. (2021). *Arduino Products*. Haettu 25.2.2021 osoitteesta <https://www.arduino.cc/en/Main/Products>

Conrad. (2020). *Fischer Elektronik SK 95 25 TO 220 Heat sink 40 K/W*. <https://www.conrad.com/p/fischer-elektronik-sk-95-25-to-220-heat-sink-40-kw-l-x-w-x-h-25-x-126-x-65-mm-to-220-188158>

Frei, M. (2015). *Verkotettujen järjestelmien vikadiagnosi*. Saarijärven Offset.

Fuel Flex Finland. (2017). *Smart Fuel Control 5.0*. Haettu 26.3.2021 osoitteesta <https://fuelflexfinland.fi/2019/11/12/blog-title-goes-here/>

Haiko, T. (2013). *Analoginen elektroniikka*. Sanoma Pro.

Ibrahim, D. (2014). *Designing Embedded Systems with 32-Bit PIC Microcontrollers and MikroC*. E-julkaisu. <https://www.sciencedirect.com/topics/engineering/pulse-width-modulation>

Mbed OS. (2021-a). *Automatic clock configuration*. Haettu 31.3.2021 osoitteesta

<https://os.mbed.com/teams/ST/wiki/Automatic-clock-configuration>

Mbed OS. (2021-b). *Development boards*. Haettu 25.2.2021 osoitteesta

<https://os.mbed.com/platforms/>

Mbed OS. (2021-c). *PwmOut – API references and tutorials*. Haettu 24.4.2021 osoitteesta

<https://os.mbed.com/docs/mbed-os/v6.9/apis/pwmout.html>

Microchip. (2008). *3V Tips 'n Tricks*.

<http://ww1.microchip.com/downloads/en/DeviceDoc/chapter%208.pdf>

OKAWA Electric Design. (2021). *RC Low-pass Filter Design Tool*. Haettu 20.2.2021 osoitteesta

<http://sim.okawa-denshi.jp/en/CRLowkeisan.htm>

Partco. (n.d.-a). *Evaluointilauta STM32F0 ARM Cortex-M0*. [KUVA]

<https://www.partco.fi/fi/elektroniikan-komponentit/aktiivit/mikrokontrollerit/arm/3890-nucleo-f072rb.html>

Partco. (n.d.-b). *HC-05 Bluetooth-moduuli*. [https://www.partco.fi/fi/arduino/arduino-](https://www.partco.fi/fi/arduino/arduino-leikkikenttae/20054-bt-hc-05.html)

[leikkikenttae/20054-bt-hc-05.html](https://www.partco.fi/fi/arduino/arduino-leikkikenttae/20054-bt-hc-05.html)

Scheehage, G. (2018). *Moottorinohjaus Toimilaitteet*. Hansaprint.

Silvonen, K. (2009). *Elektroniikka ja puolijohdekomponentit*. Otatieto.

STMicroelectronics. (2019). *STM32F072x8 STM32F072xB datalehti*.

<https://www.st.com/resource/en/datasheet/stm32f072rb.pdf>

Tutorialspoint. (2021). *Basic Electronics – Types of Transistors*.

[https://www.tutorialspoint.com/basic\\_electronics/basic\\_electronics\\_types\\_of\\_transistors.htm](https://www.tutorialspoint.com/basic_electronics/basic_electronics_types_of_transistors.htm)



## Liite 1: Mittauspöytäkirja

## Testauspöytäkirja PWM-ohjain

Syöttöjännite: **13,8V**Testaus sykli: **4 x 5 min**

Yksi puhallin 23,2 A @ 100%

Taajuus	Nopeus	Virta A	Jännite V	Maks. Lämpötila °C
1000 Hz	10 %	1,65	1,7	40
1000 Hz	25 %	4,1	3,8	49,5
1000 Hz	50 %	8,58	6,6	62,1
1000 Hz	75 %	15,35	9,8	67,5
5000 Hz	10 %	1,23	1,1	36,8
5000 Hz	25 %	3,05	3,1	43
5000 Hz	50 %	8,25	6,6	61,1
5000 Hz	75 %	15,25	9,9	66,7
10 000 Hz	10 %	1,32	1,1	38,3
10 000 Hz	25 %	3,02	3,1	43,8
10 000 Hz	50 %	8,36	6,6	62,2
10 000 Hz	75 %	15,17	9,9	68,2
20 000 Hz	10 %	1,4	1,2	40,1
20 000 Hz	25 %	3,12	3,1	47,1
20 000 Hz	50 %	8,25	6,6	64,5
20 000 Hz	75 %	15,18	9,9	72,5

Kaksi puhallinta 37,2 A @ 100%

Taajuus	Nopeus	Virta A	Jännite V	Maks. Lämpötila °C
1000 Hz	10 %	3,01	1,7	53,3
1000 Hz	25 %	8,31	3,8	61
1000 Hz	50 %	15,91	6,5	81,8
1000 Hz	75 %	28,65	9,8	74,1
5000 Hz	10 %	3,13	1,1	57
5000 Hz	25 %	8,22	3,1	62,9
5000 Hz	50 %	15,1	6,5	82,1
5000 Hz	75 %	28,61	9,8	75,3
10 000 Hz	10 %	3,21	1,1	58,1
10 000 Hz	25 %	8,25	3,1	62,2
10 000 Hz	50 %	15,63	6,5	85
10 000 Hz	75 %	28,73	9,9	80,2
20 000 Hz	10 %	3,19	1,2	58,1
20 000 Hz	25 %	8,35	3,1	63,1
20 000 Hz	50 %	15,77	6,6	87,5
20 000 Hz	75 %	28,95	9,8	81,6