

Henri Harju

HN ECU-MODUULIN TESTILAITTEEN TUOTEKEHITYSPROJEKTI

Diplomityö
Sähkötekniikka
Prof. Jukka Vanhala
Prof. Karri Palovuori
Maaliskuu 2022

TIIVISTELMÄ

Henri Harju: HN ECU-moduulin testilaitteen tuotekehitysprojekti
Diplomityö
Tampereen yliopisto
Sähkötekniikan diplomi-insinöörin tutkinto-ohjelma
Maaliskuu 2022

Tämä diplomityö tehtiin Patria Aviation Oy:n Systems-liiketoiminnan tuotekehitysprojektina. Patria Aviationin laitehuollon piiriin kuuluu Suomen ilmavoimien Boeing F/A-18 Hornetin (HN) huolto. Työn tarkoituksena on suunnitella laitehuollon tueksi sopiva testilaitteen prototyyppi, jolla voisi testata Hornetin moottorinohjausyksikön (ECU) moduulin toimintaa. Työ antaa myös tietoa siitä, millainen projekti mittajärjestelmän suunnittelu olisi laajemmalle kokonaisuudelle sekä millaiseen suorituskäyttöön päästään ja olisiko siihen investointi kannattavaa.

HN ECU-moduuleille suoritettavat lentokelpoisuushyväksyntätestit on tehty manuaalisesti. Tämä on melko työläs ja aikaa vievä toimenpide suorittaa, mutta mittauksen automatisoinnilla ne voitaisiin suorittaa huomattavasti nopeammin. Tavoitteena oli suunnitella ja toteuttaa mittalaite, joka ei olisi sidottu tiettyyn paikkaan eikä käyttäjään ja jota olisi tarvittaessa helppo siirtää paikasta toiseen.

Mittalaitteen prototyyppi toteutettiin suunnitteleamalla mittauselektronikkakortti, johon liitetään STMicroelectronicsin STM32F746NG -kehitysalustakortti. Käyttäjä kytkee moduulin mittausjärjestelmään ja valitsee suoritettavan testin. Laite suorittaa moduulille testispesifikaation mukaiset testit ja näyttää testitulokset näytöllä.

Mittauskortin laitteisto- että ohjelmistosuunnittelussa on kaksi merkittävää osa-aluetta: ohjausjännitteiden tuottaminen sekä jännitteiden mittaaminen. Näiden toteuttamiseksi tehdyt suunnitteluratkaisut käydään läpi perusteluineen. Mikrokontrollerin ohjelmoinnissa hyödynnettiin reaaliaikaisen käyttöjärjestelmän ominaisuuksia.

Projektin aikana mittauselektronikkakortista tehtiin kaksi tuotekehitysversiota. Työssä perehdytään näiden toimintaan vaikuttaneista asioista ja syistä, sekä esitetään lukuisia parannusehdotuksia jatkokehitystä ja tuotteistamista varten.

Mittalaitteen suunnittelu osoittautui haastavaksi projektiksi. Suurin ongelma oli järjestelmässä esiintyvät häiriöpiikit ja yleinen kohinataso, joiden takia laitteistolta vaadittua suorituskäyttöä lentokelpoisuushyväksyntätestien suorittamiseen ei aivan saavutettu. Testivaatimukset täyttävän laitteen suunnittelu ja toteutus tuotantoversioksi olisi vaatinut vielä yhden kehityskierroksen, ja suunnitteluun joitain merkittäviä muutoksia. Projektin aikataulu- ja kustannussyistä tuotteistusta ei tehty. Projekti antoi silti vastaukset sille annettuihin tavoitteisiin.

Avainsanat: HW-suunnittelu, elektroniikka, jännitteen mittaus, tuotekehitys, mikrokontrollerin ohjelmointi

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

Henri Harju: HN ECU module tester product development project
Master of Science Thesis
Tampere University
Master's Degree Program in Electrical Engineering
March 2022

This thesis was a product development project for Patria Aviation Oy's Systems business. Patria Aviation's equipment maintenance area includes the maintenance of the Finnish Air Force's Boeing F / A-18 Hornet. The purpose of this work is to design a suitable test device prototype to support the maintenance of the Hornet engine control unit (ECU) module. The work also provides information on what kind of project the design of the measurement system would be for a larger entity, as well as what kind of performance could be achieved and whether the investment would be worth the investment.

The airworthiness approval tests performed on the HN ECU modules have been performed manually. This is a rather laborious and time-consuming procedure to perform, but by automating the measurements, those could be performed much faster. The aim was to design and implement a measuring device that would not be tied to a specific location or user and could be easily moved from one place to another if necessary.

The prototype of the measuring device was implemented by designing a measuring electronics card to which STMicroelectronics' STM32F746NG development platform card is connected. The user connects the module to the measurement card and selects the test to be performed. The device performs tests on the module according to the test specification and displays the test results on the screen.

There are two important areas in the hardware and software design of a measurement card: generating control voltages and the measurement of the voltages. The design solutions made to implement these are reviewed with justification. The features of the real-time operating system were utilized in the programming of the microcontroller.

During the project, two product development versions of the measurement electronics card were made. The work examines the issues and reasons that affected the operation and suggests numerous improvements for further development and productization.

The design of the measuring device proved to be a challenging project. The main problem was the interference peaks in the system and the overall noise level, which did not quite achieve the performance required of the equipment to perform the airworthiness approval tests. The design and implementation of a test-compliant device into a production version would have required another round of development with some significant design changes. Due to the schedule and budget of the project, no productization was done. The project still provided answers to the goals given to it.

Keywords: hardware design, electronics, voltage measurement, product development, microcontroller programming

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tämä diplomityö on tehty Patria Aviation Oy:n Systems- liiketoiminnan rahoittamana tuotekehitysprojektina. Kiitokset työnantajalle tästä mahdollisuudesta ja osoittamastaan ymmärryksestä ja kärsivällisyydestä. Kiitokset työn tarkastajille Karri Palovuorelle ja Jukka Vanhalalle. Kiitos ystävälleni Matti Lukkariselle monista keskusteluhetkistä asian tiimoilta ja työn oikoluvusta viimeistelyvaiheessa. Erityiskiitos äidilleni, joka on jaksanut tukea ja kannustaa minua tässä haasteellisessa, loputtoman tunteisen pitkässä projektissa ja tutkintoni loppuunsaattamisessa.

Tampereella, 28.2.2022

Henri Harju

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. TYÖN TAUSTAA JA JÄRJESTELMÄN VAATIMUKSET	3
2.1 Testausjärjestelmän vaatimukset	4
2.2 Väyläohjattava mittausjärjestelmä	5
2.3 Kehitysalustakortin valinta	6
3. HW-SUUNNITTELU	8
3.1 Pulssinleveysmodulaatio eli PWM	8
3.1.1 PWM-asetukset	9
3.1.2 PWM-ohjauksen suunnittelu	10
3.2 PWM-signaalin suodattaminen	11
3.2.1 Sallen-Key aktiivisuodatin	12
3.3 Summaava vahvistin	14
3.4 Jännitteen skaalaus AD-muuntimelle	17
3.5 Negatiivisten jännitteiden skaalaus AD-muuntimelle	18
3.6 Sähkönsyöttö	20
3.7 Piirilevyn suunnittelu	21
3.8 Piirilevyn kokoonpano	22
3.9 Mittausjärjestely	23
4. OHJELMISTO	25
4.1 Super-loop	25
4.2 FreeRTOS	26
4.2.1 Tehtävien synkronointi ja niiden välinen tiedonsiirto	29
4.3 Model-View-Presenter -malli	30
4.4 Ohjelman rakenne ja toiminta	31
4.4.1 PWM-ohjausarvojen laskenta	32
4.4.2 Jännitteen skaalaus AD-muuntimelle	33
4.4.3 Negatiivisen jännitteen skaalaus AD-muuntimelle	34
4.4.4 Mittaussäie	34
5. TULOSTEN ANALYSOINTI	38
5.1 Ensimmäinen tuotekehitysversio	38
5.2 Toinen tuotekehitysversio	39
5.3 Kehitysehdotuksia	43
6. YHTEENVETO	46
LÄHTEET	47

KUVALUETTELO

Kuva 1. A11-moduuli	3
Kuva 2. A11 moduulin testikytkentä.	5
Kuva 3. STM32F746G -Discovery	7
Kuva 4. Lohkokaavio mittausjärjestelmästä.	7
Kuva 5. PWM-signaali	8
Kuva 6. Yleiskäyttöisen ajastinpiirin lohko-kaavio [6].	9
Kuva 7. RC-alipäästösuodatin.	11
Kuva 8. Kondensaattorin jännite ajan funktiona [7].	12
Kuva 9. Sallen-Key aktiivisuodatin.	13
Kuva 10. 3-asteen alipäästösuodatin.	14
Kuva 11. Vahvistin ja bufferiaste.	14
Kuva 12. Summaava vahvistin.	14
Kuva 13. PWM -ohjausjännite, summaava vahvistin ja invertoiva yksikkövahvistin.	16
Kuva 14. Vahvistinkytkennän ulostulojännite 50% PWM ohjauksella.	16
Kuva 15. Vahvistinkytkennän ulostulojännite PWM-ohjauksen funktiona.	17
Kuva 16. Ohjausjännitteen V7_fb skaalaus AD-muuntimelle.	18
Kuva 17. Negatiivisen jännitteen skaalaus mikrokontrollerille.	19
Kuva 18. Järjestelmän virran- ja tehonkulutus.	21
Kuva 19. Piirilevyn kerrosrakenne.	22
Kuva 20. Valmis piirilevy kalustettuna. Suodatinkondensaattoreita asennettu "jälkiasennuksena".	23
Kuva 21. Mittausjärjestely.	23
Kuva 22. Super-loop.	25
Kuva 23. RTOS toimintaperiaate	26
Kuva 24. Säikeiden mahdolliset tilat FreeRTOS käyttöjärjestelmässä [11].	27
Kuva 25. Ennakoiva aikataulutus [14].	28
Kuva 26. Käyttöliittymän MVP- malli [18].	30
Kuva 27. Projektin luokkien hierarkia [18]	31
Kuva 28. Lohkokaavio ohjelman toiminnasta.	32
Kuva 29. Ohjelman päävalikko.	32
Kuva 30. Kytkentä sisääntulojännitteen -15...9,9V skaalaamiseksi välille 0...3,3V.	34
Kuva 31. Piirilevyn etupuolella vain liitännät STM-kortille.	39
Kuva 32. Käyttöjännitteessä näkyvä häiriö.	40
Kuva 33. +15V käyttöjännite, lisätty 470uF kondensaattori.	41
Kuva 34. Ulostulo V9, 0V ohjauksella.	41
Kuva 35. Ulostulo V9, ohjausjännite > 6,4V.	42
Kuva 36. Mittaustuloksia	43

LYHENTEET JA MERKINNÄT

ADC	Analog to Digital Converter, analogia-digitaalimuunnin
API	Application Programming Interface, ohjelmointirajapinta
ARM	Advanced RISC Machines, 32-bittinen prosessoriarkkitehtuuri
CPU	Central Processing Unit, prosessori
DAC	Digital to Analog Converter, digitaali-analogiamuunnin
DCMI	Digital Camera Interface
DMA	Direct Memory Access
ECU	Engine Control Unit, moottorinohjausyksikkö
GPIO	General Purpose Input/Output, sisään- ja ulostulo pinni
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HW	Hardware
IDE	Integrated Design Environment
ISR	Interrupt Service Routine, keskeytyksen palvelurutiini
LCD	Liquid Crystal Display
LSB	Least significant bit
MUX	Multiplexer
PCB	Printed Circuit Board
RAM	Random Access Memory
RTOS	Real-Time Operating System, reaaliaikainen käyttöjärjestelmä
SAI	Serial Audio Interface
SD	Secure Digital
SMD	Surface Mount Device
SW	Software
Telok ry	Tampereen Elektroniikan Opiskelijoiden Kerho ry
UART	Universal Asynchronous Receiver Transmitter
UI	User Interface
USB	Universal Serial Bus
A	Ampeeri
f	taajuus
Hz	Hertsi
R	resistanssi
U	jännite
I	virta

1. JOHDANTO

Sähköisten järjestelmien ja elektroniikkalaitteiden käyttö integroituu yhä tiiviimmin osaksi nyky-yhteiskuntaa. Näiden laitteiden luotettavuuden merkitys korostuu entisestään. Laitteen vikaantuminen tai epäluotettava toiminta saattaa lievimmillään aiheuttaa vain jonkin toiminnon viivästymistä ja harmistusta käyttäjälle. Toisessa ääripäässä ovat turvallisuuskriittiset järjestelmät, kuten autojen lukkiutumattomat jarrut (ABS), ydinreaktorien tai lentokoneiden ohjausjärjestelmät tai vaikka jotkin lääketieteelliset laitteet. Näissä vika tai liian suuri viive saattaa johtaa katastrofaaliseen virheeseen, jonka seurauksena voi olla mittavaa tuhoa omaisuudelle ja ennen kaikkea ihmishengille.

Avioniikkajärjestelmät ovat niin sanottuja turvallisuuskriittisiä järjestelmiä, joihin kohdistuu tiukat viranomaisvaatimukset. Ne joutuvat kestämaan toistuvaa altistumista lämpötilan vaihteluille, paineelle, ilmankosteudelle ja mekaaniselle tärinälle. Elektronisten laitteiden ominaisuuksiin kuuluu valitettavasti vikaantuminen ja ominaisuuksien heikentyminen ja edellä mainitut vaativat käyttöolosuhteet kiihdyttävät näitä. Lentokoneiden elinkaari on tavallista nykyajan kuluttajaelektroniikkaa huomattavasti pitkäikäisempää, jopa n. 30 vuotta. Komponenttien ikääntyminen yhdessä vaativien käyttöolosuhteiden kanssa aiheuttavat väistämättä tarvetta säännöllisille huoltotoimenpiteille [1,2].

Tämän työn tarkoituksena oli suunnitella ja toteuttaa Boeing F/A-18 Hornetin huoltotoiminnan tueksi sopiva testilaitteen prototyyppi, jolla voi testata Hornetin moottorinohjausyksikön (ECU) moduulin toimintaa. Työ antaa myös tietoa siitä, millainen projekti mittajärjestelmän suunnittelu olisi laajemmalle kokonaisuudelle, sekä millaiseen suorituskykyyn päästään ja onko siihen investointi kannattavaa.

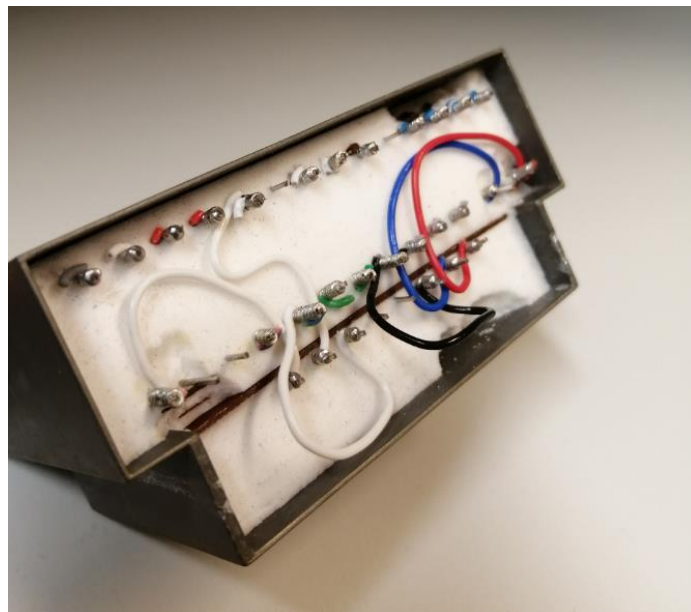
2. luvussa käydään läpi työn taustaa ja esitellään laitteiston vaatimukset sekä toteutustapa. Mikrokontrolleri- ja moduulikortin välille tarvittavan ohjauselektroniikan suunnittelu- ja mitoitusperiaatteet esitellään luvussa 3. Mikrokontrollerin ohjelmointia ja siihen liittyviä asioita käsitellään luvussa 4. Projektin aikana saatuja tuloksia ja näiden syitä käydään läpi luvussa 5. Yhteenveto työn tuloksista on viimeisessä luvussa 6. Laitesuunnittelun dokumentaatio on liitteinä. Kokonaisia ohjelmalistauksia ei ole sisällytetty niiden laajuuden vuoksi.

Tässä dokumentissa oletetaan, että lukijalla on perustiedot elektroniikan suunnittelusta ja sulautettujen järjestelmien ohjelmoinnista. Laitesuunnittelun ja ohjelmiston laajuuden vuoksi jokaista yksityiskohtaa ei käydä läpi.

2. TYÖN TAUSTAA JA JÄRJESTELMÄN VAATI- MUKSET

Patria Aviation Oy on Suomen Puolustusvoimien strateginen kumppani, ja vastaa muun muassa ilmavoimien Boeing F/A-18 Hornetin huollosta. Patrian laitehuollon toimintaan kuuluu huoltoon tulevien laitteiden testaus. Yksinkertaisten laitteiden testaus manuaalisesti mittaamalla on perusteltua, mutta laitteen toiminnallisuuden kasvaessa inhimillisen virheen mahdollisuus kasvaa, ja on hidasta. Laitteiden testispesifikaatioiden testisekvenssit voivat olla monimutkaisia, jolloin manuaalisesta testauksesta tulee työlästä, tai käytännössä mahdotonta suorittaa. Automaattisella- tai puoliautomaattisella testauksella voidaan mittaukset suorittaa huomattavasti nopeammin ja luotettavammin kuin manuaalisesti. Tässä työssä suunnitellaan ja toteutetaan huoltotoiminnan tueksi mittalaitteen prototyyppi, jolla voi mitata Hornetin ECU:n A11-moduulin toimintaa.

Hornetin moottorinohjausyksikkö sisältää useita toiminnaltaan erilaisia moduuleja, kuten Kuva 1 oleva A11-moduuli. Moduulia ei ole suunniteltu sen huollettavuuden kannalta, sillä sen irrottaminen vaatii johtimien katkaisun. Moduuli ja sen liitinpinnit on suojattu tärinää vaimentavalla kumimaisella massalla. Massausta ei voida helposti poistaa liitinpinneistä eikä siihen ei voida kytkeä suoraan mitään liitintä, mikä hankaloittaa sen mittaamista.



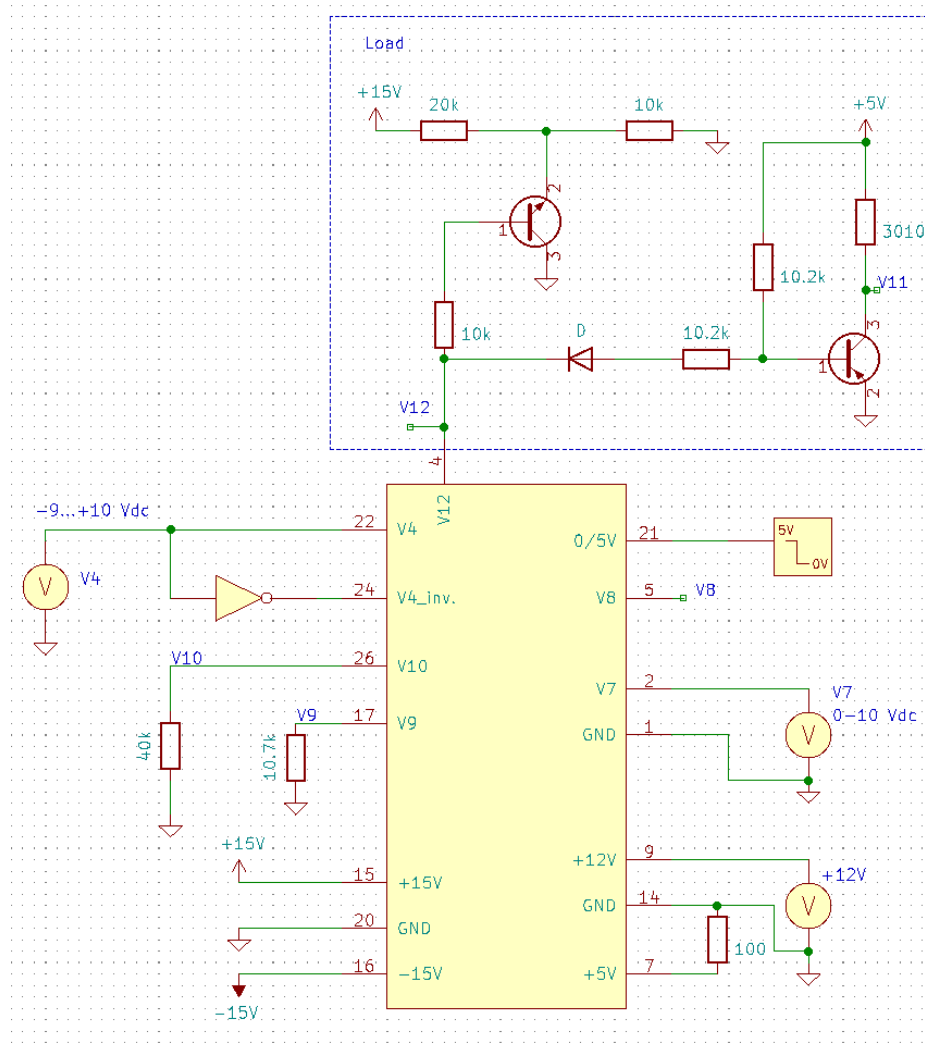
Kuva 1. A11-moduuli

2.1 Testausjärjestelmän vaatimukset

Suunnittelu alkoi tutustumalla A11-moduulin testistandardin vaatimuksiin ja siihen millainen funktionaalinen testi tulisi moduulikortille suorittaa. Testispesifikaation raja-arvot olivat annettu jopa 1 mV:n tarkkuudella, mikä asetti vaatimustason laitteiston suorituskyvystä haastavaksi. Laitteen vaatimusmäärittely oli seuraava:

- A11-moduuli tarvitsee ± 15 V:n ja +12 V:n käyttöjännitteet
- Kolme eri funktionaalista testiä, joiden ohjausjännitteet vaihtelevat välillä n. -9V...+10V
- Aktiivikuorman käyttö
- Moduulin inputin logiikkatason ohjaus (0 tai 5 V)
- Moduulin oman regulaattorin ulostulojännitteen testaus
- Mittaustulosten näyttö jollain päätteellä
- Ilmoitus testin onnistumisesta. Mikäli laite ei läpäise testiä ilmoitus mikä tai mitkä mittaukset eivät menneet läpi. Lisäksi toivottiin mahdollisuutta tallentaa mittaustulokset muistikortille.

Moduulikortti vaatii toimiakseen ± 15 V:n sekä +12 V:n käyttöjännitteet. Lisäksi moduuliin tulee erillinen kuormakytKentä, johon tarvitaan +12 V:n ja +5 V:n käyttöjännitteet. Mitattavalle A11-moduulikortille testistandardin mukaisia testejä on neljä. Testeissä sisääntulojännitteitä parametrisoidaan määrättyjen rajojen välillä, jolloin ulostulojännitteiden tulee olla niille määriteltyjen rajojen sisällä. Sallitut poikkeamat tavoitearvoissa vaihtelevat ± 200 mV:sta ± 50 mV:in riippuen sisääntulojännitteiden suuruudesta, sekä testistä. Testistandardin mukainen testikytkentä A11-modulille on Kuva 2.



Kuva 2. A11 moduulin testikytkentä.

2.2 Väyläohjattava mittausjärjestelmä

Ensimmäisenä vaihtoehtona oli testausjärjestelmän tekeminen väyläohjattavilla laboratoriotehonlähteillä. Lisäksi tarvittaisiin sopiva mittausohjelmisto, kuten esimerkiksi National Instruments:in LabVIEW. LabVIEW on graafiseen G-kieleen perustuva helppokäyttöinen ohjelmointiympäristö, joilla voidaan suunnitella automaattisia mittausjärjestelmiä. Mittauksen ohjaukseen tarvitaan tietokone, jossa on LabVIEW ohjelmisto, sekä väyläohjattava tehonlähde tai useampia. Hyvinä puolina tässä toteutuksessa on sen yksinkertainen toteutus, sekä muunneltavuus. Myös suorituskyvyn voi olettaa olevan hyvällä tasolla, sillä NI:n järjestelmiä on käytetty lentokone-teollisuudessa jo useamman vuosikymmenen ajan. Toisaalta toteutus vaatisi investointeja laadukkaisiin väyläohjattaviin laboratoriotehonlähteisiin sekä ohjelmistolisenssien hankkimisen. Koska mittauksesta haluttiin sellainen, että sen voisi

tehdä kuka tahansa huollon henkilökunnasta ja paikasta riippumatta, päätettiin se toteuttaa mikrokontrolleripohjaisesti erillisenä laitteena. Lisäksi haluttiin selvittää millaisella panostuksella oman mittalaitteen tuotekehitys onnistuisi ja tietoa sen soveltuvuudesta mittauksiin.

2.3 Kehitysalustakortin valinta

Tavoitteena oli tehdä mittalaite, joka ei ole sidoksissa tiettyyn käyttäjäryhmään tai paikkaan. Laitteen prototyyppi päätettiin toteuttaa käyttäen sopivaa kehitysalustakorttia, johon voisi liittää mittauselektronikan sisältävän piirikortin. Kehitysalustakortteja on markkinoilla paljon eri vaihtoehtoja ja aluksi päädyttiin käyttämään STMicroelectronicsin STM32F469 -discovery kehitysalustakorttia [3]. Valintaperusteina oli riittävä määrä (8 kpl) suoria liityntöjä A/D-muuntimille, sekä käytettävissä olevat 2 DAC lähtöä. Lisäksi kortissa on 800x480 pikselin LCD-kosketusnäyttö ja mikro-SD muistikorttipaikka. Valinnassa painottui vahvasti myös STMicroelectronicsin tarjoama ilmaisohjelma graafisen käyttöliittymän suunnitteluun, sekä graafinen integroitu kehitysympäristö (IDE) mikrokontrollerin ohjelmointiin. Käyttämällä kehitysalustakorttia voitiin myös aloittaa ohjelmiston kehitys ennen mittauskortin prototyyppin valmistumista, käyttämällä koekytkentälevyä ja erilliskomponentteja.

Graafisen käyttöliittymän ohjelmisto TouchGFX on melko uusi ja sen integrointi ohjelmistotyökalun (CubeMX) kanssa oli vielä tätä työtä aloittaessa kesken. Projektin edetessä selvisi, että ohjelmisto ei tarjoa vielä tukea kyseiselle kortille. Se jouduttiin vaihtamaan saman valmistajan STM32F746-discovery kehitysalustaan (Kuva 3) [4], joka on ominaisuuksiltaan hyvin lähellä STM32F469:sää. Kortilla on tämän työn kannalta paljon tarpeettomia ominaisuuksia (muun muassa Ethernet, DCMI, SAI), mutta oleellisia tietoja tämän työn kannalta ovat:

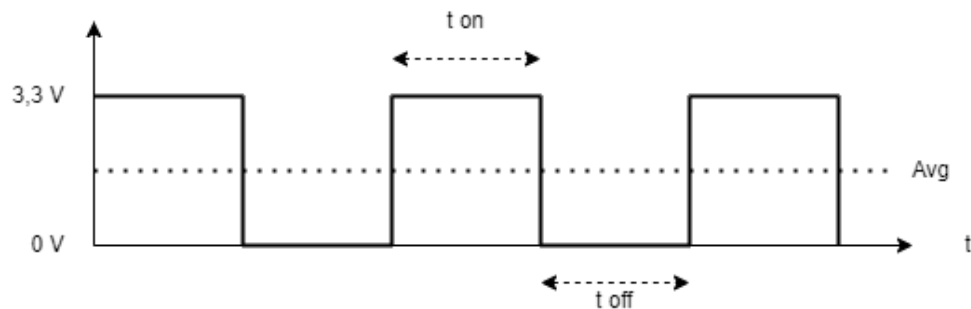
- Mikrokontrolleri STM32F746NGH6, 1Mb Flash -muistia, 340 Kb RAM
- 4,3" RGB 480x272 kapasitiivinen kosketusnäyttö
- 128 Mb Quad-SPI Flash-muistia
- 128 Mb SDRAM
- 3 kpl 12-bittistä AD-muunninta
- ARDUINO Uno V3 yhteensopivat liittimet

3. HW-SUUNNITTELU

Kun vaatimusmäärittelyyn oli perehdytty ja toteutustapa valittu, mittausjärjestelmän lohkokaaviosta (Kuva 4) suunniteltiin kytkentäkaavio. Suunnittelu on jaoteltu kahteen tärkeään osa-alueeseen. Osa-alueista ensimmäinen sisältää mittausjärjestelmässä tarvittavien ohjausjännitteiden tuottamisen vaatimuksien mukaisilla tasoilla. Toisena on suunnitella mittausjärjestelmän jännitteiden mittaaminen ja skaalaaminen mikrokontrollerin AD-muuntimelle. Lisäksi tarvitaan ohjaukset kahdelle kytkimelle sekä multiplekserille. Järjestelmän eri osioita testattiin koekytkentälevyllä, jolloin näiden toimivuutta kehitysalustakortin kanssa päästiin kokeilemaan ennen prototyypin valmistamista.

3.1 Pulssinleveysmodulaatio eli PWM

Valitussa STM32F746G- kehitysalustakortissa DAC-lähdöt ovat kortin muiden oheislaitteiden käytettävissä, joten ohjausjännitteet tuli tehdä käyttämällä PWM:ää. Pulssinleveysmodulaatiossa tasajännitelähtöä moduloidaan muuttamalla sen päällä- ja poissaoloaika eli pulssisuhdetta tietyllä taajuudella (Kuva 5). PWM-signaalin jännitteen keskiarvo on suoraan verrannollinen pulssisuhteeseen.



Kuva 5. PWM-signaali

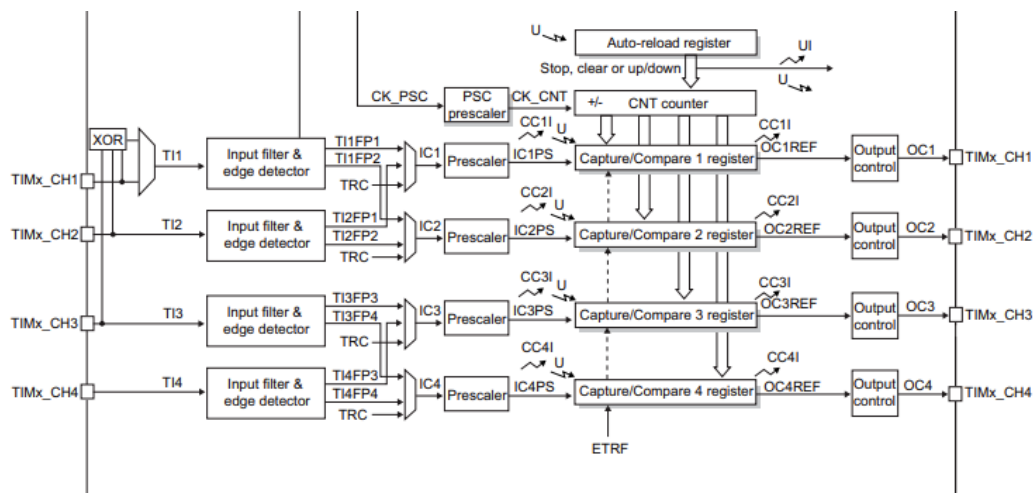
STM32F746G -mikrokontrollerissa on 18 ajastinpiiriä (Timer), joita voidaan käyttää mm. PWM -ohjauksen tuottamiseen. Mikrokontrollerin erilaisia ajastintyyppejä ovat perus- (Basic), yleiskäyttöinen- (General-purpose) ja kehittynyt- (Advanced) ajastin. Lisäksi on erityinen ajastin pienen virrankulutuksen tilaan (Low-power timer), sekä korkean resoluution ajastin (High-resolution timer). Perus -ajastimia voidaan käyttää vain ohjelmallisesti ajastuksien toteuttamiseen ja tehtävien synkronointiin. Niillä ei ole I/O-ominaisuuksia. Yleiskäyttöisillä ajastimilla voidaan toteuttaa monipuolisesti erilaisia ajoituksia, aikalaskureita, aikavertailuja ja PWM-ohjauksia. Kehittyneillä-ajastimilla on

erityisesti moottorinohjauksiin ja tehoelektronikan sovellutuksiin liittyviä lisäominaisuuksia [5,6].

3.1.1 PWM-asetukset

Yleiskäyttöisen ajastinpiirin lohkokaavio on Kuva 6, josta olennaisimpia asioita ovat kellon esijakaja (prescaler), Auto-reload -rekisteri ja Capture/Compare -rekisteri. Ajastimen tuottaman PWM-signaalin pulssisuhde määräytyy Auto-reload -rekisterin ja Capture/compare -rekisteriin asetettujen arvojen perusteella (1) [6].

$$\text{pulssisuhde}[\%] = \frac{CCR_x}{ARR_x} [\%] \quad (1)$$



Kuva 6. Yleiskäyttöisen ajastinpiirin lohkokaavio [6].

Pulssisuhteen lisäksi PWM-ohjauksien suunnittelussa tulee ottaa huomioon PWM-taajuus ja resoluutio, jotka kaikki linkittyvät toisiinsa.

PWM –taajuus (f_{PWM}) määräytyy seuraavasti (2):

$$f_{PWM} = \frac{f_{CLK}}{(ARR+1)(PSC+1)} \quad (2)$$

, missä f_{CLK} on ajastinväylän kellotaajuus, ARR (Auto Reload Register) on rekisteriin määritetty asetusarvo, ja PSC on ajastimen kellon esijakajan arvo [6].

PWM:n resoluutio ($Resolution_{PWM}$) voidaan puolestaan laskea seuraavasti (3):

$$Resolution_{PWM} = \frac{\log\left(\frac{f_{CLK}}{f_{PWM}}\right)}{\log 2} [Bits] \quad (3)$$

Yhdistämällä kaavat (2) ja (3), PWM-resoluutioksi saadaan (4):

$$Resolution_{PWM} = \frac{\log(Arr+1)(PSC+1)}{\log 2} [Bits] \quad (4)$$

Kellon, PWM-signaalin taajuuden ja resoluution välillä on seuraava yhteys (5):

$$Timer\ frequency = PWM_{f_max} * 2^{PWM\ resolution} \quad (5)$$

Asetusarvojen rajat riippuvat rekisterin maksimiarvosta, joka 16-bittisellä rekisterillä on $2^{16} - 1$, eli 65535. Lisäksi rajan asettaa väylän maksimikellotaajuus (*Timer frequency*) [6].

Taulukossa 1 on PWM- maksimitaajuuden ja resoluution välinen riippuvuus, kun ajastinväylän kellotaajuus on 200 MHz.

Taulukko 1. PWM- resoluution ja taajuuden välinen riippuvuus väylän kellotaajuuden ollessa vakio.

<i>Timer frequency</i>	<i>PWM resolution [bits]</i>	<i>PWM max frequency</i>
200 MHz	16	~3,05 kHz
200 MHz	14	~12,2 kHz
200 MHz	12	~48,8kHz
200 MHz	10	~195,3 kHz
200 MHz	8	~781 kHz
200 MHz	6	~3,125 MHz
200 MHz	4	~12,5 MHz

Taulukosta 1. huomataan, että mikäli halutaan tarkempaa resoluutiota, ei voida käyttää niin suurta PWM-taajuutta, mikä ei välttämättä käy kyseiseen sovellukseen, tai tekee sen suodattamisesta haasteellisempaa.

3.1.2 PWM-ohjauksen suunnittelu

Tässä työssä käytettyjen ohjausjännitteiden vaatimuksena oli tarkka, muutaman millivolttin tarkkuudella asetettava ohjausjännite. Tähän tarvittavan PWM-ohjauksen resoluutiota voi arvioida laskemalla, kuinka moneen osaan PWM-ohjauksen maksimiulostulojännite ($V_{in_max} = 3,3\text{ V}$) täytyy jakaa, jotta päästään tarvittavaan noin 2 mV:n resoluutioon. Laskutoimituksien perusteella askelien määräksi valittiin 5000. Resoluution suunnittelussa pitää ottaa huomioon myös resoluution kertautuminen

ohjauksen vahvistuksen mukaan (6). PWM-ohjauksen vahvistuksen suunnittelu käsitellään kappaleessa 3.2.1.

$$resolution = \frac{V_{in_max}}{pwm_resolution} * G_A = \frac{3,3}{5000} * 3,1 = 2,0 \text{ mV} \quad (6)$$

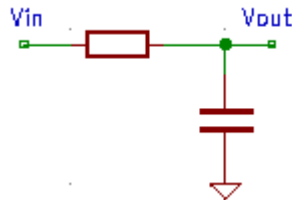
Käytettävän ajastimen väylän kellotaajuus on 200 MHz ja kellotaajuuden esijakajaa ei käytetä (PSC = 0). Näillä arvoilla PWM-signaalin taajuus saadaan kaavan (2) mukaan:

$$f_{PWM} = \frac{200 \text{ MHz}}{(4999 + 1)(0 + 1)} = 40,000 \text{ kHz}$$

Huomiona, että 0 on ensimmäinen arvo, joten ARR rekisteriin sijoitetaan (askelmäärä -1), eli 4999.

3.2 PWM-signaalin suodattaminen

PWM-signaalia ei usein sellaisenaan voida käyttää ohjaukseen. Yksi syy tähän on PWM:n tuottamat harmoniset taajuudet, jotka kulkeutuvat ilman alipäästösuodattamista järjestelmään. Yksinkertaisimmillaan alipäästösuodin voi olla pelkkä RC-suodatin (Kuva 7).



Kuva 7. RC-alipäästösuodatin.

Suodattimen rajataajuudella tarkoitetaan sitä taajuutta, jolloin ulostulosignaali on vaimentunut 3 dB, toisin sanoen sen teho on puolittunut. Rajataajuus (f_c) voidaan laskea seuraavasti:

$$f_c = \frac{1}{2\pi RC} \quad (7)$$

Kaavasta (7) nähdään, että mitä suurempi vastuksen tai kondensaattorin kapasitanssin arvo, sitä matalampi on rajataajuus.

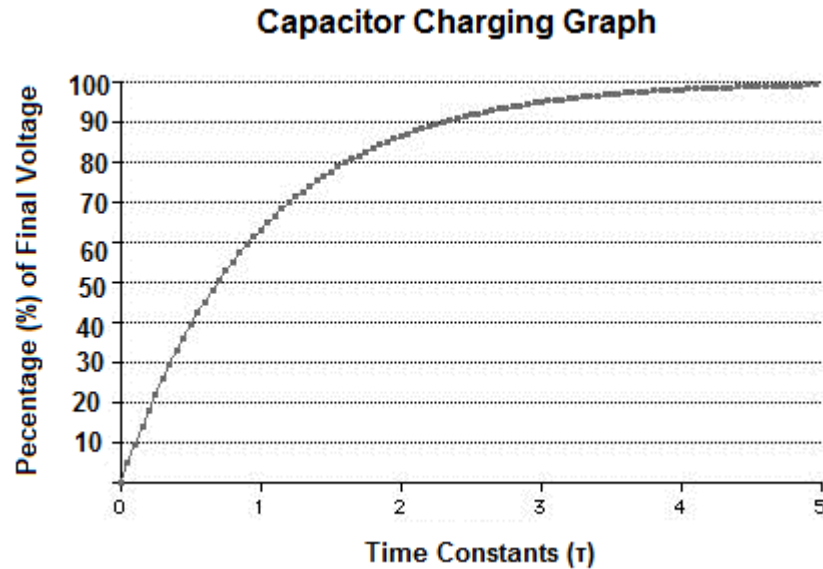
Aikavakion τ perusteella määräytyy, kuinka kauan kestää, että suodattimen ulostulojännite saavuttaa lopullisen arvonsa. Suodattimen aikavakio määräytyy kaavan (8) mukaan seuraavasti:

$$\tau = RC \quad (8)$$

Kondensaattorin jännite voidaan laskea kaavan 9 mukaan:

$$V_C = V_{in}(1 - e^{(-t/RC)}) \quad (9)$$

, missä V_{in} on sisääntulojännite, t on aika, joka on kulunut jännitteen kytkemisestä ja RC on aikavakio.



Kuva 8. Kondensaattorin jännite ajan funktiona [7].

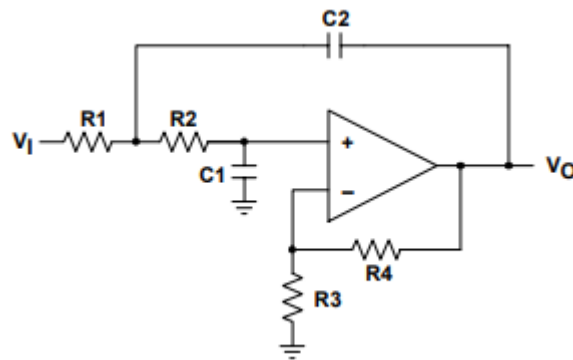
Kuva 8 on kondensaattorin jännite askelmaisen herätteen jälkeen. Kun aikavakio on 5 kondensaattori on saavuttanut 99,3 % tulojännitteestä. Tällöin voidaan käytännössä katsoa, että kondensaattori on täysin latautunut [8]. Kaavan (9) perusteella mitä suurempi vastuksen tai kondensaattorin kapasitanssin arvo, sitä kauemmin kestää suodattimen ulostulojännitteen asettuminen lopulliseen arvoonsa.

Suodattimen rajataajuuden valintaan liittyy PWM signaalin modulaatiotaajuus. Jos suodattimen vasteen on oltava nopea (pieni aikavakio), PWM- signaalin modulaatiotaajuuden on oltava korkeampi. Tässä järjestelmässä mittauksen nopeudella ei ole juuri merkitystä, joten voidaan valita suhteellisen matala rajataajuus (≈ 33 Hz).

3.2.1 Sallen-Key aktiivisuodatin

Ensimmäisen ohjausjännitteen tarvittava dynaaminen jännitealue on 0 - 10 V. Mikrokontrollerin referenssijännite on sama kuin sen maksimiulostulojännite, joka tässä tapauksessa on 3,3 V. Tämän perusteella tarvittava jännitevahvistus on $10\text{ V}/3,3\text{ V} = 3,03$. Tähän otettiin hieman marginaalia, jotta varmasti päästään 10 V:in, joten vahvistukseksi valittiin 3,1.

Järjestelmä mittaa myös omaa ohjausjännitettään, jotta varmistetaan sen oikeellisuudesta ja tarkkuudesta. Ensimmäisissä testeissä havaittiin, ettei pelkkä RC-alipäästösuodatin vaimentanut riittävästi PWM-signaalin rippelijännitettä. Suodatuksen jälkeen ohjaussignaalissa oli 150 mV:n suuruisia jännitepiikkejä, jotka näkyivät AD-muunnoksen mittaustuloksissa. Tämän vuoksi suodatuksen astelukua jouduttiin nostamaan lisäämällä RC-suodattimen perään Sallen-Key tyyppinen aktiivisuodatin [8] (Kuva 9). Tällä suodatinrakenteella saavutettiin huomattavasti parempi vaimennus, jonka johdosta ohjausjännitteen rippelijännite vaimeni alle 10 mV:iin.



Kuva 9. Sallen-Key aktiivisuodatin.

Sallen-Key suodattimen komponenttiarvojen mitoittamiseksi käytetään kytkennän siirtofunktiota hyväksi. Kirjallisuudesta löytyy lukuisia analyysejä suodattimen toiminnasta [9,10], ja sen siirtofunktio voidaan esittää kaavan (10) mukaan:

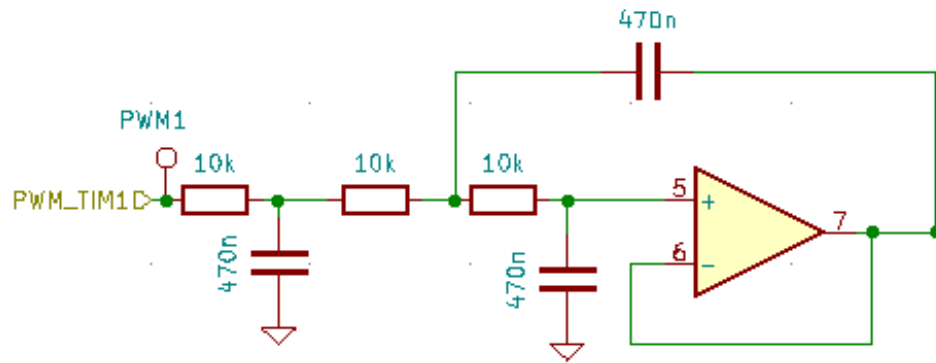
$$H(s) = \frac{V_o}{V_i}(s) = \frac{K}{s^2(R_1R_2C_1C_2) + s(R_1C_1 + R_2C_1 + R_1C_2(1-K)) + 1} \quad (10)$$

Q-arvo saadaan kaavan (11) mukaan:

$$Q = \frac{\sqrt{R_1R_2C_1C_2}}{R_1C_1 + R_2C_1 + R_1C_2(1-K)} \quad (11)$$

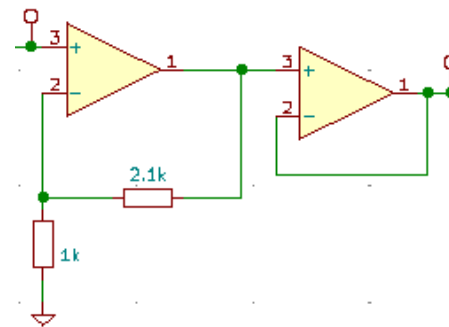
, missä $K = 1 + \frac{R_4}{R_3}$ (eli DC-vahvistuskerroin) ja $s = j\omega$. Suunnittelua yksinkertaistaa, jos komponenttiarvot valitaan siten, että $R_1 = R_2$ ja $C_1 = C_2$. Tästä seuraa, että kaavat yksinkertaistuvat muotoon $f_c = \frac{1}{2\pi RC}$ ja $Q = \frac{1}{3-K}$.

Edellä suunniteltu vahvistus on 3,1. Tällä vahvistuksella huomataan, että suodattimen Q-arvosta tulee negatiivinen, mikä tarkoittaa sen olevan tällöin epästabiili. Tämän vuoksi päädyttiin käyttämään yksikkövahvisteista Sallen-Key suodinta (Kuva 10).



Kuva 10. 3-asteen alipäästösuodatin.

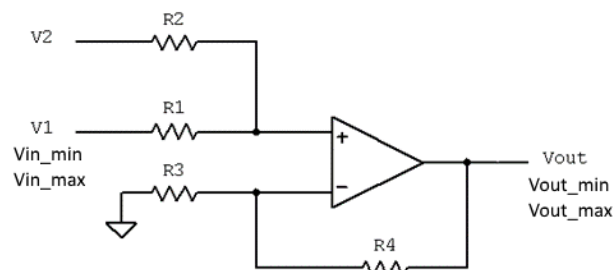
Tarvittava vahvistus ($K = 3,1$) tehdään suodatuksen jälkeen (Kuva 11), jolloin vahvistus ei vaikuta Sallen-Key suodattimen hyvytyteen tai stabiiliuteen. Vahvistimen perässä on vielä bufferiaste, joka erottaa kytkennän muusta piiristä.



Kuva 11. Vahvistin ja bufferiaste.

3.3 Summaava vahvistin

Toisen ohjausjännitteen tarvittava dynamiikka oli $-9\text{ V} - 9\text{ V}$. Tämän toteuttamiseksi suunniteltiin summaava vahvistinkytkentä (Kuva 12). Vahvistimen toisena sisäänmenona on offset-jännite (V2), ja toisena sisääntulona on suodatettu PWM-jännite (V1).



Kuva 12. Summaava vahvistin [11].

Vahvistimen siirtofunktio on seuraava (12)[11]:

$$V_{out} = \left(V1 \frac{R2}{R1 + R2} + V2 \frac{R1}{R1 + R2} \right) \left(1 + \frac{R4}{R3} \right) \quad (12)$$

Ulostulojännite voidaan ilmaista minimi- ja maksimisisääntulojännitteiden avulla seuraavasti (13):

$$\begin{cases} V_{out_min} = \left(V_{in_min} \frac{R2}{R1 + R2} + V2 \frac{R1}{R1 + R2} \right) \left(1 + \frac{R4}{R3} \right) \\ V_{out_max} = \left(V_{in_max} \frac{R2}{R1 + R2} + V2 \frac{R1}{R1 + R2} \right) \left(1 + \frac{R4}{R3} \right) \end{cases} \quad (13)$$

Vastuksia on neljä tuntematonta, mutta riittää jos niiden suhde ratkaistaan. Merkitään $k1 = R2/R1$, $k2 = R4/R3$ ja sijoitetaan ne yhtälöön (13), jolloin saadaan (14):

$$\begin{cases} V_{out_min} = \left(V_{in_min} \frac{k1}{1 + k1} + V2 \frac{1}{1 + k1} \right) (1 + k2) \\ V_{out_max} = \left(V_{in_max} \frac{1}{1 + k1} + V2 \frac{1}{1 + k1} \right) (1 + k2) \end{cases} \quad (14)$$

Tästä yhtälöparista voidaan ensiksi ratkaista algebrallisesti tuntematon $k1$ (15).

$$k1 = V2 \frac{V_{out_min} - V_{out_max}}{V_{out_max} V_{in_min} - V_{out_min} V_{in_max}} \quad (15)$$

Sijoittamalla $k1$ yhtälöön (14) saadaan $k2$ seuraavaan muotoon (16):

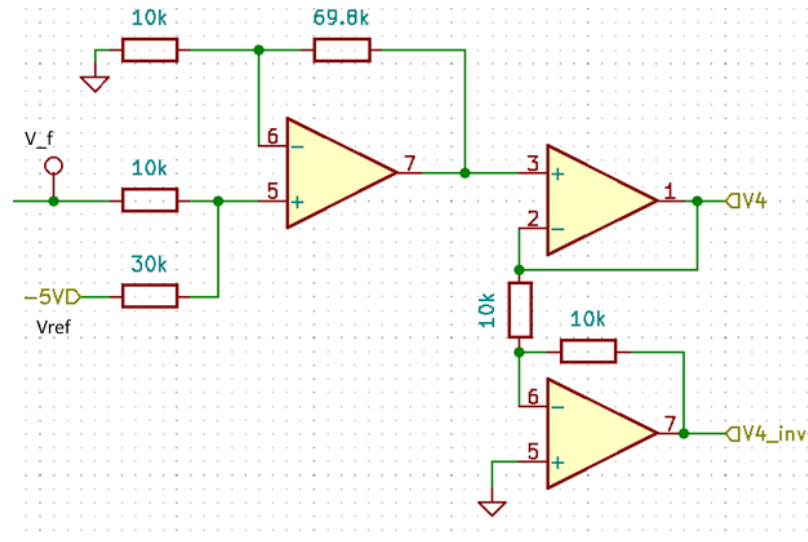
$$k2 = \frac{V_{out_min} - V_{out_max}}{V_{in_min} - V_{in_max}} + \frac{V_{out_max} V_{in_min} - V_{out_min} V_{in_max}}{V2(V_{out_min} - V_{out_max})} - 1 \quad (16)$$

Nyt vastuksien suhteet voidaan kirjoittaa muotoon (17):

$$\begin{cases} R2 = R1 V2 \left(\frac{V_{out_min} - V_{out_max}}{V_{out_max} V_{in_min} - V_{out_min} V_{in_max}} \right) \\ R4 = R3 \left(\frac{V_{out_min} - V_{out_max}}{V_{in_min} - V_{in_max}} + \frac{V_{out_max} V_{in_min} - V_{out_min} V_{in_max}}{V2(V_{out_min} - V_{out_max})} - 1 \right) \end{cases} \quad (17)$$

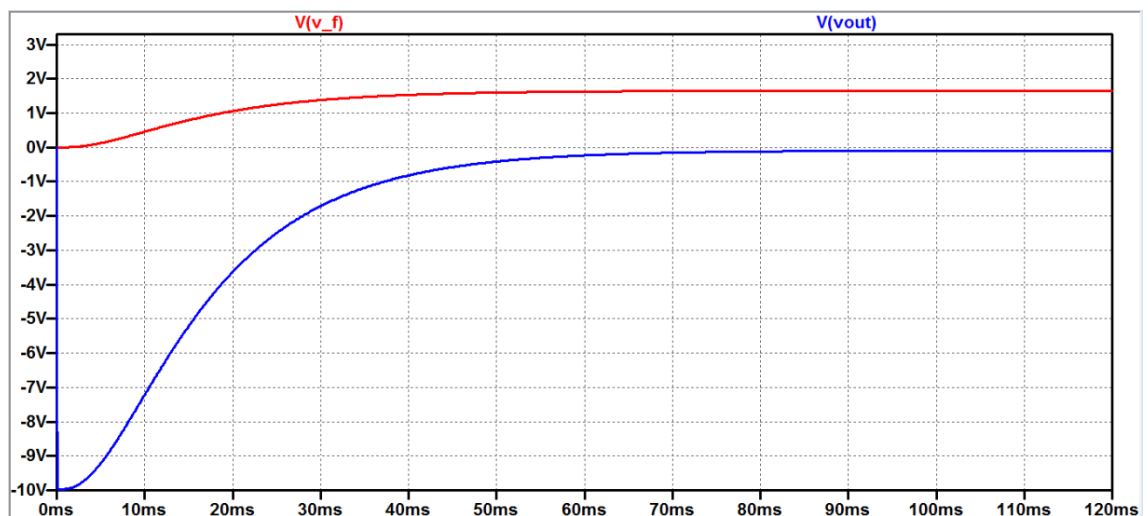
Halutut sisään- ja ulostulojännitteet tiedetään: $V_{in_min} = 0 \text{ V}$, $V_{in_max} = 3,3 \text{ V}$, $V_{out_min} = -9 \text{ V}$, $V_{out_max} = 9 \text{ V}$. Offset-jännitteen $V2$ tulee olla negatiivinen, jotta negatiivinen ulostulojännite on mahdollinen. Vastusarvojen määrittäminen tapahtui laskemalla ensin yhtälön (17) perusteella sopivia arvoja, ja sen jälkeen simuloimalla kytkentää LTspice- ohjelmalla. Kytkentä haluttiin pitää yksinkertaisena, joten mieluummin valittiin yksi vastus, kuin useamman vastuksen sarja- ja/tai rinnankytkentä, jolla olisi saanut toteutetuksi sopivan resistanssiarvon. Ulostulojännitteen dynaamista aluetta oli helpompi

muuttaa, kuin yrittää sovittaa vastusarvoja tiettyyn jännitealueeseen. Lopullisilla komponentti ja offset-jännitteen valinnoilla ulostulojännitteen dynamiikka on $-10,0 - 9,8$ V. Kuva 13 on suunniteltu summaava vahvistinkytkentä. Vahvistimen perässä on puskurina jänniteseuraaja, sekä invertoiva yksikkövahvistin toiselle ohjaukselle.



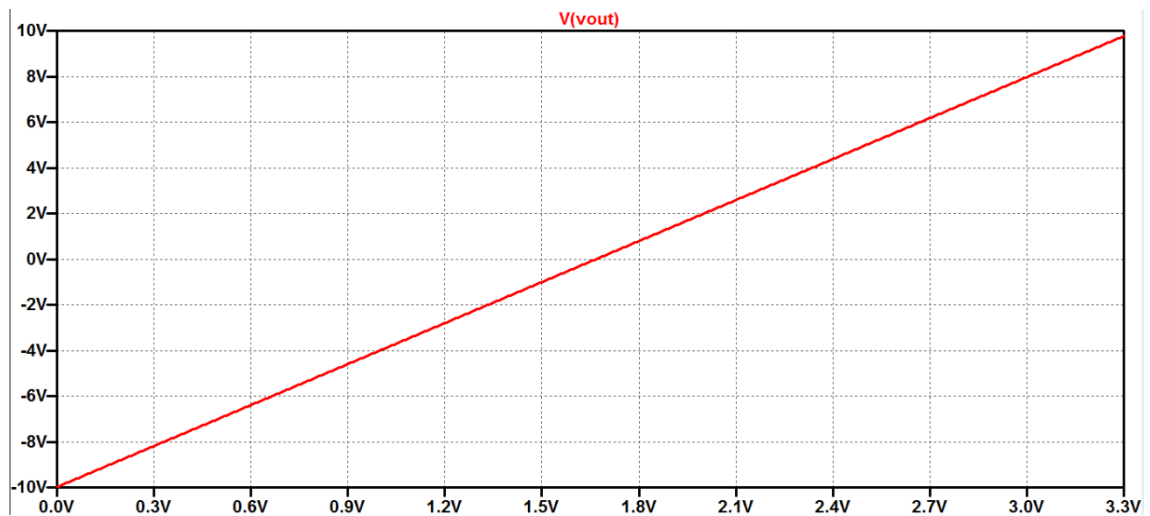
Kuva 13. PWM -ohjausjännite, summaava vahvistin ja invertoiva yksikkövahvistin.

Aktiivisuodattimen ja summaavan vahvistimen yhteistoimintaa testattiin simuloinneilla. Kuva 14 on transienttisimulointi vahvistinkytkennän ulostulojännitteestä. PWM-ohjaussignaalin pulssisuhde on asetettu 50 %:in. Kuva 14 näkyy, kuinka ulostulojännitteen asettumisaika on yli 60 ms. Tämä on otettava huomioon ohjelmoitaessa jännitteen mittausta. Ohjausjännitteitä muutettaessa on varattava riittävä asettumisaika ennen AD-muunnoksen ottamista.



Kuva 14. Vahvistinkytkennän ulostulojännite $V4$ 50% PWM ohjauksella.

Vahvistinkytkennän ulostulojännitteen kuvaaja tulojännitteen funktiona on Kuva 15. 0 V:n ohjauksella ulostulo on -10,0 V ja maksimi 3,3 V:n ohjauksella ulostulo on 9,80 V.



Kuva 15. Vahvistinkytkennän ulostulojännite PWM-ohjauksen funktiona.

Kytkenän vahvistus on sama kuin sen kulmakerroin, eli ulostulojännitteen suhde sisäänmenojännitteeseen (18).

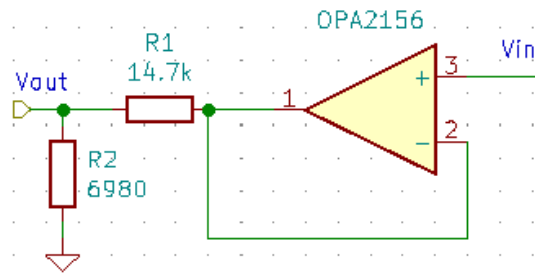
$$G_A = \frac{\Delta V_{out}}{\Delta V_{in}} = \frac{(9,8 - (-10))V}{3,3V} = 6,0 \quad (18)$$

Ohjauksen resoluutio on tällä ohjauksignaaliolla noin 4,0 mV. Resoluutiota olisi voinut lisätä pienentämällä PWM -taajuutta, mutta sitä ei katsottu tarpeelliseksi tehdä.

3.4 Jännitteen skaalaus AD-muuntimelle

Mikrokontrollerin AD-muuntimelle tuleva jännite täytyy skaalata sille sopivalle jännitealueelle 0 - 3,3 V. Jännitteen skaalaus sovitetaan tarkasti AD-muuntimen mittausalueen mukaiseksi, jotta saadaan käytettyä AD-muuntimen maksimaalinen mittausresoluutio [12]. Laitteen ohjauksignaalin V7 (Kuva 2) riittävä dynaaminen alue on 0 - 10,0 V, joten mittauksen dynamiikaksi valittiin 0 - 10,25 V, jotta yläraja pystytään mittaamaan varmasti oikein. Jännitteen skaalaus tapahtuu bufferiasteen jälkeen jännitejaolla kahden vastuksen kytkennän avulla (Kaava (19), Kuva 16).

$$V_{out} = \frac{R_2}{R_1 + R_2} * V_{in} = G_{fb} * V_{in} \quad (19)$$



Kuva 16. Ohjaujännitteen V7_fb skaalaus AD-muuntimelle.

Valitut vastukset ovat 0,1 % tarkkuusvastuksia. Tarkkuusvastusten käytön voi perustella tutkimalla niiden merkitystä mittaustarkkuuteen.

Taulukko 2 on vertailu AD- muunnoksen arvoista eri jännitteillä ideaalitapauksessa, sekä vastusarvojen 1% ja 0,1% poikkeamilla Worst-case tapauksessa. AD-muunnosta vastaava jännitearvo saadaan seuraavasti (20):

$$\frac{ADC_{reading}}{ADC_{Fullscale}} = \frac{V_{in}}{V_{Fullscale}} \quad (20)$$

Jännitealue on edellä mainittu 0 - 10,25 V ja AD -muunnoksen suurin arvo on 4095.

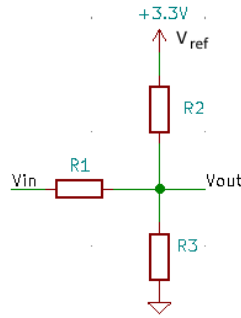
Taulukko 2. Vastuksien toleranssin vaikutus AD-muunnosten tarkkuuteen Worst-case tapauksessa.

Vastusten toleranssi	AD-muunnoksen arvo			AD-muunnosta vastaava jännite [V]		
	1 V	4 V	10 V			
Referenssi	400	1598	3995	1,001	4,000	10,00
1 %	394	1576	3941	0,986	3,945	9,865
0,1 %	399	1596	3990	0,999	3,995	9,987

Taulukon 2. arvoista nähdään virheen lisääntyvän erityisesti suuremmilla jännitteillä, jolloin tuloksessa saattaa olla jo yli 130 mV:n virhe 1% toleranssin vastuksilla.

3.5 Negatiivisten jännitteiden skaalaus AD-muuntimelle

Mikrokontrollerin sisääntuloihin ei saa syöttää negatiivista jännitettä, koska se voi vaurioitua siitä. Mikrokontrollerin AD-muunnin ei pysty myöskään mittaamaan negatiivista jännitettä. Yksinkertainen tapa skaalata negatiivinen jännite positiiviseksi on käyttää Kuva 17 esitettyä kolmen vastuksen kytkentää.



Kuva 17. Negatiivisen jännitteen skaalaus mikrokontrollerille.

Vastukset mitoitetaan siten, että pisteen V_{out} jännite on 0 V, kun sisääntulossa V_{in} on sen pienin jännite. Tällöin vastuksen R3 yli oleva jännite on 0 V, eikä sen läpi kulje virtaa. Kirchoffin virtalain mukaan pisteeseen tulevien ja siitä lähtevien virtojen summa on sama (21).

$$\frac{0 V - V_{in_min}}{R1} = \frac{V_{ref}}{R2} \leftrightarrow R2 = R1 \frac{V_{ref}}{-V_{in_min}} \quad (21)$$

Toinen tapaus on sisääntulon maksimi, jolloin $V_{out} = V_{ref}$. Nyt vastuksen R2 läpi ei mene virtaa koska potentiaaliero on 0 V. Vastaavasti voidaan kirjoittaa seuraava yhtälö (22):

$$\frac{V_{in_max} - V_{out}}{R1} = \frac{V_{out}}{R3} \leftrightarrow R3 = R1 \frac{V_{ref}}{V_{in_max} - V_{ref}} \quad (22)$$

Kytkenän vahvistuskerroin G (23) saadaan seuraavasti:

$$G = \frac{\Delta V_{out}}{\Delta V_{in}} = \frac{(R2 || R3)}{(R1 + (R2 || R3))} \quad (23)$$

Vastusarvojen valinnassa on otettava huomioon kytkennän impedanssi. AD -muuntimen sisääntuloimpedanssin kasvaessa signaalin näytteistys vaatii pidemmän ajan [10]. Mikäli impedanssi on liian pieni, AD-muuntimen oma sisääntuloimpedanssi kuormittaa kytkentää. Kaavan (24) mukaan AD-muuntimen sisääntuloimpedanssille voidaan määrittää maksimiarvo, jolloin sen aiheuttama virhe on pienempi kuin $1/4$ LSB :

$$R_{AIN} = \frac{(k - 0,5)}{f_{ADC} * C_{ADC} * \ln(2^{N+2})} - R_{ADC} \quad (24)$$

, missä k on näytteistykseen kuluvien AD-muuntimen kellojaksojen määrä, f_{ADC} on AD-muuntimen näytteistystaajuus, C_{ADC} on AD-muuntimen sisäinen kapasitanssi, N on resoluutio (bittien lukumäärä, $N=12$) ja R_{ADC} on AD-muuntimen näytteistystapahtuman aikainen resistanssi (Sampling switch resistance) [3]. Valmistajan ilmoittamien arvojen ($C_{ADC} = 5,5, pF, R_{ADC} = 6 k\Omega$) [3], sekä näytteistysjaksojen (480) ja AD-muuntimen

kellotaajuuden (25 MHz) perusteella maksimi sallittu impedanssi on 50 k Ω . Kytkennän vastusarvot määritetään tämä huomioiden.

Kappaleessa 3.3 esitetyn summaavan vahvistimen dynaaminen alue on mitoitettu välille -10,0 V:sta 9,8 V:in ja V_{ref} jännite on 3,3 V. Vastuksia on kolme tuntematonta, joten yksi pitää valita, jotta muut voidaan määrittää. Valitaan $R_1 = 10 \text{ k}\Omega$, ja nyt sijoittamalla lähtötiedot yhtälöihin (21) ja (22) saadaan $R_2 = 3,3 \text{ k}\Omega$ ja $R_3 = 5 \text{ k}\Omega$. Vastusarvot määritettiin interpoloimalla siten, että kyetään toimimaan halutulla toiminta-alueella.

Suunnittelussa on otettava huomioon, ettei kytkennän sisääntulon jännite alita tai ylitä sille määritettyjä minimi- ja maksimijännitteitä. Mittaustulokset ovat tällöin virheellisiä (saturoituneita) ja pahimmassa tapauksessa yli- tai alijännite voi hajottaa mikrokontrollerin. Kontrollerin GPIO-portit kestävät paremmin yli- kuin alijännitettä, mutta sisääntuloporttienjännite ei tule ylittää 4,0 V tai alittaa -0,3 V missään tilanteessa [13]. Mittauksen testispesifikaation rajat yhden testin dynamiikalle oli -9...6.4V. Testauksessa ilmeni, että ilman ohjausta lepotilassa sisääntulon jännite olikin n. -12V, eli kontrollerin AD-muuntimen sisääntulossa oli muutaman voltin suuruinen negatiivinen jännite. Tämä ei onneksi vielä rikkonut mikrokontrolleria ja seuraavassa suunnitteluvaiheessa jännitteen skaalaus mitoitettiin -15V:in.

3.6 Sähkönsyöttö

Sähkönsyötön suunnittelussa vaihtoehtona oli laitteen tekeminen akkukäyttöisenä. Se olisi teettänyt jonkin verran lisätyötä, joten prototyypin osalta tämä idea jätettiin toteuttamatta. Sähkönsyöttöön käytetään laboratoriotehonlähdettä, josta otetaan $\pm 15 \text{ V}$:n käyttöjännitteet. Näistä reguloidaan lisäksi tarvittavat +12V:n, +5V:n, 3,3V:n ja -5V:n käyttöjännitteet. Valitut regulaattorit ovat kaikki lineaariregulaattoreita niiden pienemmän kohinan takia (piirikaavio liitteessä B). Kehitysalustakortille viedään 12 V:n käyttöjännite, josta kortin omat regulaattorit tuottavat tarvitsemansa jännitteet. Koko järjestelmän virrankulutus kuormituksen aikana oli maksimissaan noin 650 mA (Kuva 18). Regulaattorit kestivät hyvin tämän kuormituksen, eikä niiden lämpötilan kanssa ilmennyt ongelmia. Regulaattoreiden tarvitsemat kondensaattorit mitoitettiin niiden datalehtien mukaan, mutta niitä jouduttiin vaihtamaan kapasiteetiltaan suuremmiksi, josta lisää kappaleessa 5.



Kuva 18. Järjestelmän virran- ja tehonkulutus.

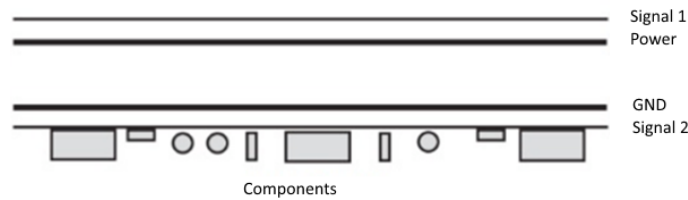
3.7 Piirilevyn suunnittelu

Piirilevyn koko oli vapaasti valittavissa ja ensimmäinen versio tehtiin standardin Eurokortin (100 mm x 160 mm) kokoisena. Kooltaan se oli jo tarpeettoman iso, mutta ajatuksena oli, että siihen kokoon oli useita saatavilla olevia valmiita kotelovaihtoehtoja.

Piirilevyn komponenttien osasijoittelussa määräävä tekijä oli siihen kytkettävän kehitysalustakortin liittimet (Kuva 3). Näihin tulevien vastaliittimien kohdistus tuli mitoittaa tarkasti. Tarvittavat I/O-liittimet sijoitettiin levyn reunoille. Komponenttisijoittelussa pyrittiin pitämään signaalien reititykset lyhyinä ja välttämään ylimääräisiä silmukoita. Reitityksien pitäminen lyhyinä osoittautui melko haasteelliseksi tehtäväksi. Tähän vaikutti STM-kortin vastaliittimet, joiden sijoitukseen ei voinut vaikuttaa. Komponentteja oli noin 100 kpl, sekä 6 eri reititettävää käyttöjännitettä. Lisäksi kaksipuolisen piirilevyn toinen puoli yritettiin pitää mahdollisimman yhtenäisenä maatasoa varten. Piirrettäessä layout:ia kytkentään lisättiin tarpeen tullen 0Ω :n vastuksia, jos niillä pystyi lyhentämään merkittävästi reititystä tai välttämään useampia läpivientejä.

Prototyyppejä valmistettiin kaksi eri versiota. Ensimmäinen levy teetätettiin kaksipuolisenä. Komponenttien asennus tehtiin Patrian toimesta. Levy tilattiin ilman juotteenestopinnoitetta, jotta mahdolliset muutostyöt olisivat helpoimmin tehtävissä. Tässä versiossa oli muutamia suunnitteluvirheitä, jotka tulivat esille testauksessa. Näitä käydään tarkemmin läpi kappaleessa 5. Tästä saatiin kuitenkin paljon informaatiota, mihin asioihin piti kiinnittää huomiota ja korjata. Myös ohjelman toimintaa pystyttiin puutteista huolimatta testaamaan.

Toiseen versioon levyn layout suunniteltiin kokonaan uudestaan, joka tehtiin 4-kerros levynä. Piirilevyn kerrosrakenne on Kuva 19 mukainen yleinen suunnittelutapa, jossa kerrokset ovat: signaali- maataso- käyttöjännitteet- signaali.

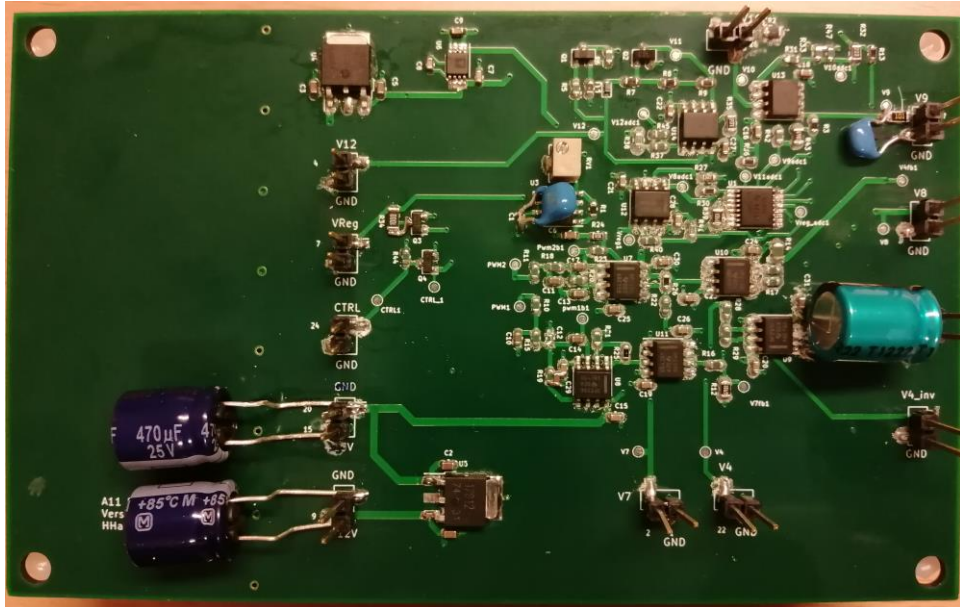


Kuva 19. Piirilevyn kerrosrakenne.

Levyn kokoa muutettiin siten, että se oli ulkomitoiltaan samankokoinen kuin STM-kortti (80 mm x 130 mm). Suunnitteluperiaatteena oli sijoittaa kaikki komponentit piirilevyn pohjapuolelle ja levyn etupuolelle pelkästään liittimet STM-korttia varten. Muutoin mittapisteiden mittaaminen olisi mahdotonta tai ainakin hyvin hankala, koska kortit ovat niin lähellä toisiaan. Liittimet sijoitettiin piirilevyn reunoille. Kunkin regulaattorin ja operaatiovahvistimen käyttöjännitteiden suodatuskondensaattorit sijoitettiin mahdollisimman lähelle komponentin käyttöjännitepinniä. Käyttöjännitteille ja maatasolle varattu oma kerroksensa helpotti reitityksien suunnittelua. Liitteissä A, B ja C ovat piirikaaviokuvat kytkennästä ja liitteissä D ja E ovat layout kuvat.

3.8 Piirilevyn kokoonpano

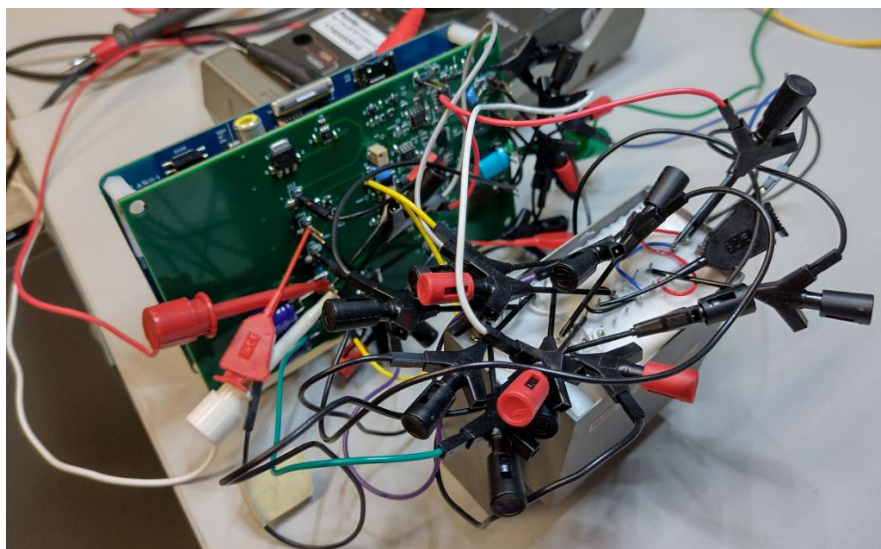
Toinen piirilevy kalustettiin itse Telok ry:n tiloissa. Kaikki komponentit olivat pintaliitoskomponentteja, lukuun ottamatta muutamaa jälkiasennuksena lisättyä suodatuskondensaattoria. Juotospastan levityksessä käytettiin stensiiliä, jonka avulla juotospasta levitettiin levyille. Tämän jälkeen komponentit aseteltiin paikoilleen. Asettelu piti tehdä huolellisesti, mutta silti riittävän nopeasti. Mikäli levitetty juotospasta ehtisi kuivua komponenttien asettelu hankaloituisi, koska ne eivät enää tarttuisi juotospastaan eivätkä siten pysyisi paikallaan. Tämän jälkeen tehtiin reflow -juotos. Reflow-prosessi jouduttiin tekemään muutaman kerran ennen sopivan lämpötilaprofiilin löytymistä, koska tekoaikaan kyseistä reflow-uunia ei oltu vielä testattu. Kuva 20 on piirilevy kalustettuna.



Kuva 20. Valmis piirilevy kalustettuna. Suodatinkondensaattoreita asennettu "jälkiasennuksena".

3.9 Mittausjärjestely

Kuva 21 näkyy laitteen mittajärjestely. Moduulista tulee 15 I/O-liitäntää, joiden kytkeminen vaatii huolellisuutta ja tarkkuutta. Mittauksissa käytettiin pienikokoisia "hauen leukoja", jotka kytkettiin A11-moduuliin. Oskilloskoopin ja yleismittarin mittapäät olivat näitä isompia ja painavampia, ja niiden kytkeminen ilman tukea aiheuttivat joidenkin mittakortin pinnien vääntymisen (Kuva 20).



Kuva 21. Mittausjärjestely.

On selvää, että käyttökelpoisuuden vuoksi laite tulisi koteloida. Liityntöjen osalta haasteellisinta olisi moduulin I/O-pinnien kytkeminen laitteeseen. Edellä (Kappale 1)

mainittiin, kuinka moduulin irrottaminen vaatii johtimien katkaisun. Ylimääräiset tinat on mahdollista puhdistaa, mutta pinnit ovat hieman erimittaiset. Suunnitelmana oli käyttää niin sanottuja Pogo-pinnejä, jotka ovat jousikuormitteisia. Kotelon kanteen tulisi Pogo-pinnit, ja mitattava moduuli painettaisiin näitä vasten. Tällöin ulkoisia mittajohtimia ei tarvittaisi lainkaan ja moduulin kytkeminen olisi nopeaa, eikä sitä voisi myöskään asentaa väärin paikoilleen. Moduulin paikoillaan pysyminen vaatisi myös jonkin kiinnityssysteemin suunnittelun. Kotelointi oli tarkoitus valmistaa 3D-tulostamalla kun laite olisi muutoin täyttänyt kaikki sille asetetut vaatimukset. Valitettavasti projektin budjetti- ja aikataulu syistä koteloinnin toteutus jäi suunnitteluasteelle.

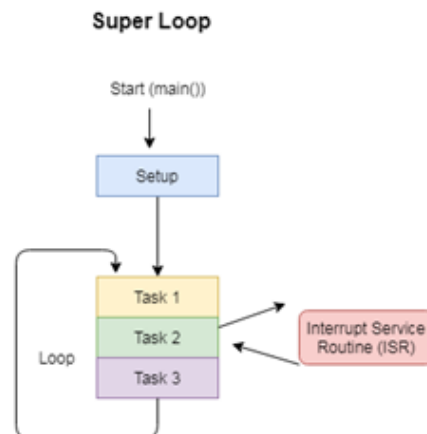
4. OHJELMISTO

Laitteiston ohjelmointiin käytettiin ST-Microelectronicsin CubeIDE -ohjelmistoa. Se on ilmainen kehitysympäristö ST:n valmistamille mikrokontrollereille ilman ohjelman kokorajoituksia, kuten joissain muissa ohjelmistoissa. CubeIDE:n yhtenä etuna on sen graafinen käyttöliittymä. Tämän avulla voi tehdä kaikki mikrokontrollerin perusasetukset mitä tarvitaan. Käyttäjä valitsee mikrokontrollerin ja mitä oheislaitteita, kelloasetuksia, I/O-pinnejä tai muita toimintoja käytetään. Ohjelma generoi näiden perusteella tarvittavat alustukset ja kirjastot valmiiksi ohjelmakoodiin, jotka se tarvitsee valittujen oheislaitteiden väliseen kommunikointiin. CubeIDE:ssä on mahdollista suorien rekisterikäskyjen sijasta (tai näiden lisäksi) käyttää ohjelmoinnissa abstraktointikerrosta (Hardware Abstraction Layer, HAL) ja sen käskykanta, joka yksinkertaistaa vielä ohjelmakoodia.

ST on integroinut kehitysympäristöön myös graafisen käyttöliittymän suunnitteluohjelmiston, TouchGFX:n, jolla voidaan suunnitella käyttöliittymän ulkoasu ja rakenne. Ohjelmalla käyttöliittymäympäristöön voidaan liittää erilaisia pienoishjelmia (widget), joilla saa tehtyä esimerkiksi tekstialueita, painonappeja, valikoita ym. Kun ulkoasu ja rakenne ovat tehty, ohjelma luo tiedostot, joita muokataan halutun toiminnan toteuttamiseksi. Ohjelmoijan tehtäväksi jää määrittää mitä dataa ja miten se siirretään kosketusnäytön ja mikrokontrollerin välillä.

4.1 Super-loop

Perinteisesti sulautetun järjestelmän ohjelman suoritus tapahtuu päättymättömässä silmukassa (super-loop), jossa tehtäviä suoritetaan järjestyksessä yksi kerrallaan (Kuva 22).

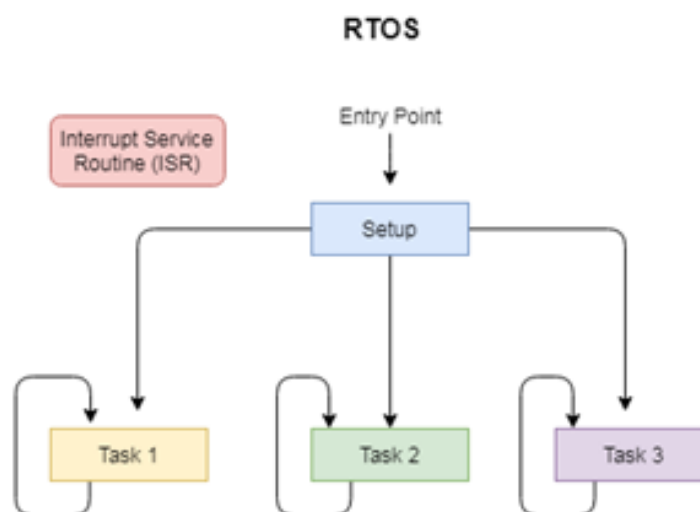


Kuva 22. Super-loop [14].

Yksinkertaisissa sovellutuksissa tämä on usein paras ratkaisu, mutta koska sulautetut järjestelmät ovat monesti yhä monimutkaisempia, niiden resurssien hallinta ja ajoitus käyvät hankalaksi hallita tällä silmukkarakenteella. Aikakriittiset tapahtumat täytyy käsitellä keskeytysten käsittelijöissä (ISR), jolloin niistä tulee monimutkaisempia ja niiden suoritusajaksi kasvaa. Lisäksi sisäkkäiset keskeytykset (nested interrupts) vaikeuttavat suoritusajan ja muistivaatimuksien ennakoitavuutta. Erityisesti ajoituksen hallinta käy haastavaksi, mikäli järjestelmän tulee kyetä moniajioon (multitasking). Jos järjestelmässä vielä on useita erilaisia ajastinsyklejä, näiden hallitseminen käy entistä vaikeammaksi ja monimutkaisemmaksi. Vaikka tällaisen ohjelman saisikin toimimaan optimaalisesti, siitä tulee helposti sekava ja vaikeasti ymmärrettävä. Ohjelman mahdolliset muutokset uusien tarpeiden mukaisiksi voivat tuoda suuria haasteita ohjelman toiminnallisuuden suhteen, ja niiden vaikutusta voi olla hankala arvioida [14,15,16].

4.2 FreeRTOS

Tehtävien moniajo ja niiden ajoituksen hallinta voidaan jättää käyttöjärjestelmän tehtäväksi. Koska suorituksen ajoitus on monesti ratkaisevan tärkeää sulautetuissa järjestelmissä, niissä käytetään niin sanottuja reaaliaikaisia käyttöjärjestelmiä. Reaaliaikaisella käyttöjärjestelmällä tarkoitetaan, että se takaa tietyn vasteajan, jonka sisällä se reagoi käskyihin tai tapahtumiin [19]. Reaaliaikainen on sinänsä hieman harhaanjohtava termi, että se ei tarkoita välttämättä nopeutta vaan sitä, että sen reagointi-aika tapahtumiin on deterministinen [18], joka on tärkeä ominaisuus erityisesti kaikissa turvallisuuskriittisissä sovellutuksissa. Kuva 23 on reaaliaikaisen käyttöjärjestelmän toimintaperiaate.

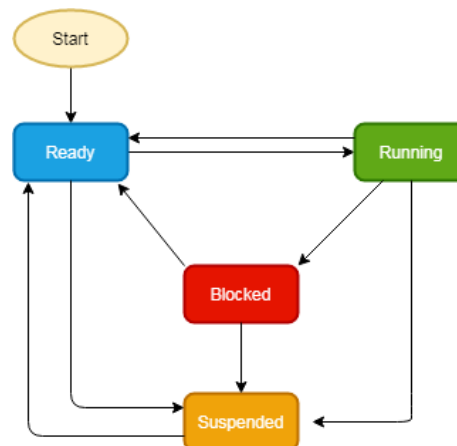


Kuva 23. RTOS toimintaperiaate [20]

Reaaliaikaiset käyttöjärjestelmät voidaan jakaa tapahtumavetoisiin (event-driven) tai aikaa jakaviin (time slicing) järjestelmiin. Tapahtumavetoisessa järjestelmässä tehtävien suoritusjärjestys määräytyy niiden prioriteetin (ja tapahtumien) perusteella, kun taas aikaa jakavissa suoritettava tehtävä vaihtuu kellon keskeytysten mukaan. Järjestelmä voi olla myös näiden yhdistelmä, kuten tässä työssä käytetty FreeRTOS [15,18,19].

Ydin (Kernel) on käyttöjärjestelmän keskeisin osa, joka ylläpitää sen prosesseja, muistin hallintaa, keskeytysten käsittelyä, I/O-dataliikennettä, tiedostojärjestelmää ym. [17,19]. Tehtävien (tasks) suoritusta hallinnoi ytimen komponentti nimeltään *vuorontaja* (scheduler), joka määrittää mikä tehtävä kulloinkin on suorituksessa [18,19]. Vuorontajan tehtävänä on maksimoida tehokas prosessoriaika siten, että tehtävien välinen odottelu-aika on mahdollisimman lyhyt. Tällöin tehtäviä voidaan suorittaa rinnakkain samanaikaisesti. Tehtävien aikataulukukseen vuorontaja käyttää tietonaan kolmea parametria: tehtävän vaatimaa suoritusaikaa (execution time), aikaa jolloin tehtävän tulee olla suoritettu (deadline) ja aikaa milloin tehtävää voidaan alkaa suorittaa (release time) [15,18,19,20].

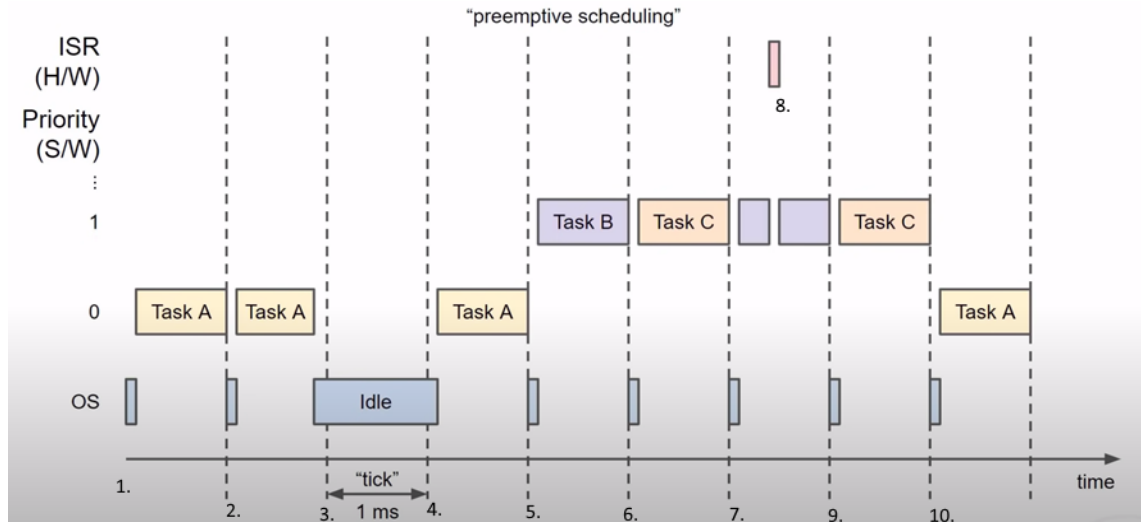
Tehtäviä kutsutaan myös säikeiksi. Käyttöjärjestelmä varaa jokaiselle säikeelle sen tarvitseman muistialueen. Ohjelma ei koskaan saa poistua säikeestä, koska sillä ei ole paluuarvoa tai osoitetta, mikä aiheuttaisi ohjelman kaatumisen. Säikeen voi kyllä halutessaan tuhota, jos siihen on tarvetta sille varatun muistin vapauttamiseksi. Säikeet voivat olla 4:ssä eri tilassa: ready, running, blocked tai suspended (Kuva 24)



Kuva 24. Säikeiden mahdolliset tilat FreeRTOS käyttöjärjestelmässä [19].

Kontekstin vaihdoksi sanotaan tapahtumaa, kun säikeen suoritus vaihtuu toiseen. Suorituksessa olevan säikeen tila (funktioparametrit, paluusoitteet ja väliaikaiset muuttujat) tallennetaan muistiin ja vastaavasti suorituksen aloittavan säikeen tila haetaan muistista [15,19].

Käyttöjärjestelmän vuorontaja tarkistaa joka kellojakso (tässä työssä käytetty kellojakso = 1 ms), onko säikeitä valmiina suoritukseen. (Huomiona, että tämä käyttöjärjestelmän kellojakso ei ole tietenkään sama kuin prosessorin kellojakso). Seuraavassa esimerkki tehtävien aikataulutuksesta (Kuva 25) :



Kuva 25. Ennakoiva aikataulutus [20].

1. Ajastin antaa vuorontajalle käskyn tarkistaa, onko tehtäviä valmiina suoritukseen. Esimerkissä tehtävä A on valmiustilassa, ja sen suoritus alkaa.
2. Käyttöjärjestelmä tarkistaa, onko muita tehtäviä valmiina suoritukseen. Ei ole, joten tehtävän A suoritus jatkuu.
3. Tehtävä A suoritus loppuu, ja se menee "blocked" tilaan. Käyttöjärjestelmä suorittaa "tyhjäkäynti tehtävää" (idle task).
4. Tehtävä A on uudelleen valmiina, ja sen suoritus alkaa. Suorituksen aikana tehtävät B ja C tulevat valmiustilaan. Niiden suoritus ei kuitenkaan ala ennen kuin seuraavaa OS:n kellojaksoa.
5. Tehtävät B ja C ovat valmiina suoritukseen. Ne ovat korkeamman prioriteetin tehtäviä, joten tehtävä A jää valmiustilaan odottamaan.
- 6-7. Tehtävä B:llä ja C:llä on sama prioriteetti, joten OS vuorottelee näiden suoritusta (time slicing).
8. Keskeytyksellä (ISR) on korkein prioriteetti, joten suoritus siirtyy keskeytyksen palvelurutiiniin, jonka jälkeen tehtävän suoritus jatkuu.
10. Tehtävät B ja C ovat suorittaneet tehtävänsä ja ovat joko "blocked" tai "suspended" tilassa. Tehtävän A suoritus voi jatkua [20].

Sulautetuissa järjestelmissä keskeytykset voivat olla joko laitteistopohjaisia tai ohjelmallisia. Laitteistokeskeytyks antaa tiedon oheislaitteen tapahtumasta mikrokontrollerille, johon järjestelmän tulee reagoida niin pian kuin mahdollista. Ohjelman ei siis tarvitse odottaa vuorontajan kellojaksoa vaihtaakseen tehtävää, vaan ohjelma tallentaa muistiin (stack) sen hetkisen tilansa ja siirtyy välittömästi keskeytyksen palvelurutiiniin (ISR). Keskeytysten palvelurutiinit tulee pitää suoritusajallisesti lyhyinä, koska ne pysäyttävät ohjelman muun suorituksen keskeytyksen ajaksi. Keskeytyksen jälkeen ohjelman suoritus jatkuu siitä mihin se jäi. Myös keskeytyksillä voi olla prioriteetteja, joten keskeytysten keskeytykset ovat mahdollisia. Ohjelmallisia keskeytyksiä käytetään yleensä vain joidenkin virhetilanteiden käsittelyyn, tai ohjelman kehitysvaiheessa laitteistokeskeytysten ohjelmalliseen testaamiseen [15,17,19,21].

4.2.1 Tehtävien synkronointi ja niiden välinen tiedonsiirto

Ohjelman suoritus vaatii yleensä datan tai tapahtumien jakoa eri tehtävien välillä. Yksinkertaisinta olisi käyttää globaaleja muuttujia, mutta tämä ei monestikaan ole suositeltavaa. Jos useampi tehtävä yrittää lukea tai kirjoittaa jaettuun resurssiin samanaikaisesti, syntyy kilpailutilanne, mikä johtaa datan korruptoitumiseen. FreeRTOS sisältää useampia tiedonsiirto-objekteja, jotka mahdollistavat tehtävien välisen turvallisen dataliikenteen [17,19].

Viestijono (Queue) on yksinkertainen ja joustava tapa tehtävien väliseen tiedonsiirtoon. Viestijonon voi lähettää myös keskeytykseen. Oletuksena viestijono toimii FIFO –periaatteella, ja data lähetetään kopiona jonoon. Ideana tässä on, että dataa lähettävä tehtävä voi ottaa jo uudelleen muuttujan käyttöön, vaikka vastaanottaja ei olisi vielä ehtinyt lukea sitä. Data säilyy viestijonossa, kunnes se on luettu. Mikäli lähetettävä datan määrä on suuri, sen voi myös lähettää osoittimena. Tällöin pitää vain varmistaa, että data säilyy, kunnes se on luettu [19,22].

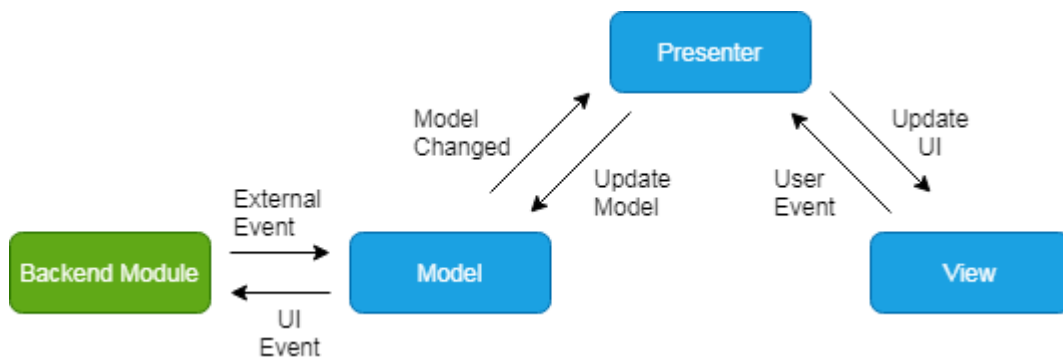
Semafori (semaphore) on mekanismi jaetun resurssin suojaamiseen tai tehtävien synkronointiin. Semafori voi olla binäärinen tai laskeva. Laskevassa semaforissa on tietty määrä tehtäviä, joilla on pääsy yhtä aikaa jaettuun resurssiin. Yleensä tätä ominaisuutta ei käytetä, vaan käyttökelpoisempi tapa on binäärisemafori, jota käytetään ohjelman suorituksen synkronointiin [19,23].

Mutex on lukkomekanismi jaetun resurssin suojaamiseen. Tehtävän ottaessa mutexin, mikään toinen tehtävä ei voi käyttää sitä ennen kuin mutex on vapautettu. Mutex on kuten binäärisemafori, mutta sisältää prioriteetin perintämekanismiin, joka nostaa mutexin haltijan prioriteettia. Tämä minimoi mahdollisuuden prioriteetin inversion

(priority inversion), jolla tarkoitetaan tilannetta, jossa korkeamman prioriteetin tehtävä joutuu blokatuksi jonkin alemman tehtävän toimesta. Tämä voi tapahtua tilanteessa, jossa korkean prioriteetin tehtävä odottaa (blocked) mutexin vapautumista alemman prioriteetin tehtävältä. Jos tänä aikana jokin kolmas tehtävä (joka on prioriteetiltaan näiden välistä) tulee valmiustilaan, suoritus vaihtuu tähän tehtävään. Näin ollen korkean prioriteetin tehtävä joutuu blokatuksi alemman prioriteetin tehtävän toimesta. [19,21,24].

4.3 Model-View-Presenter -malli

Ohjelman monimutkaisuuden kasvaessa, sen hallittavuuden ja loogisen toiminnan kannalta on hyvä erotella ohjelman toiminallisuus eri osiin, eli moduuleihin. TouchGFX:n arkkitehtuurimalli on niin sanottu MVP-malli (Model-View-Presenter) (Kuva 26).



Kuva 26. Käyttöliittymän MVP- malli [25].

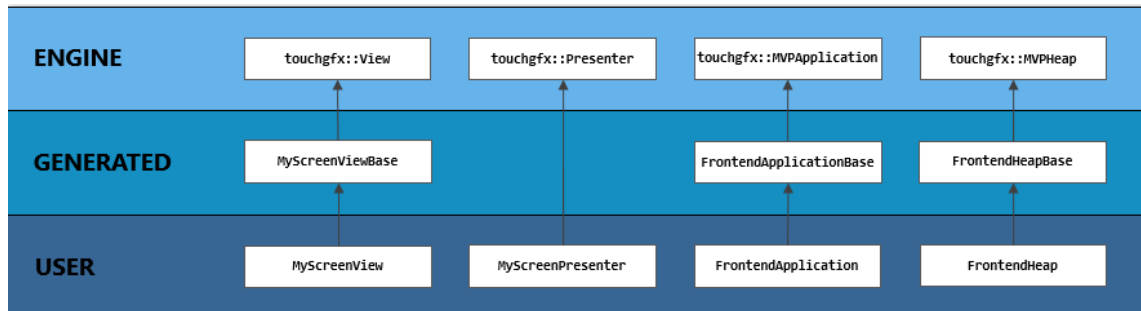
Malli (Model) on rajapinta mikrokontrollerin ja näytön välissä. Se hallinnoi dataliikennettä näiden välillä. Se toimii myös datan tallennuspaikkana rajapintojen välillä. Mikäli ohjelmassa on useampi ikkuna, vain yhden ikkunan data on kerrallaan allokoituna kyseisen ikkunan esittelijässä ja näkymässä. Ikkunan vaihtuessa sille varattu muisti vapautetaan, joka minimoi RAM-muistin tarpeen.

Esittelijä (Presenter) toimii dataliikenteen välikätenä rajapintojen välillä, ja muuntaa sen näytölle sopivaan muotoon.

Näyttö (View) rajapinta huolehtii nimenmukaisesti näytöstä ja sen toiminnasta, kuten esimerkiksi mitä ja mihin kohtaa näyttöä teksti esitetään. Se välittää myös tiedon kosketusnäytön tapahtumista esittelijälle.

Ohjelmassa model- rajapintoja on vain yksi, mutta jokaiselle eri ohjelmassa käytettävälle näyttöruudulle on oma esittelijä- ja näyttö- luokkansa. UI on näin pilkottu pieniin osakokonaisuuksiin, jolloin niiden hallitseminen on yksinkertaisempaa ja mahdolliset

muutokset eivät aiheuta niitä kaikissa osioissa. TouchGFX- ohjelmiston projektirakenne on Kuva 27 [25].

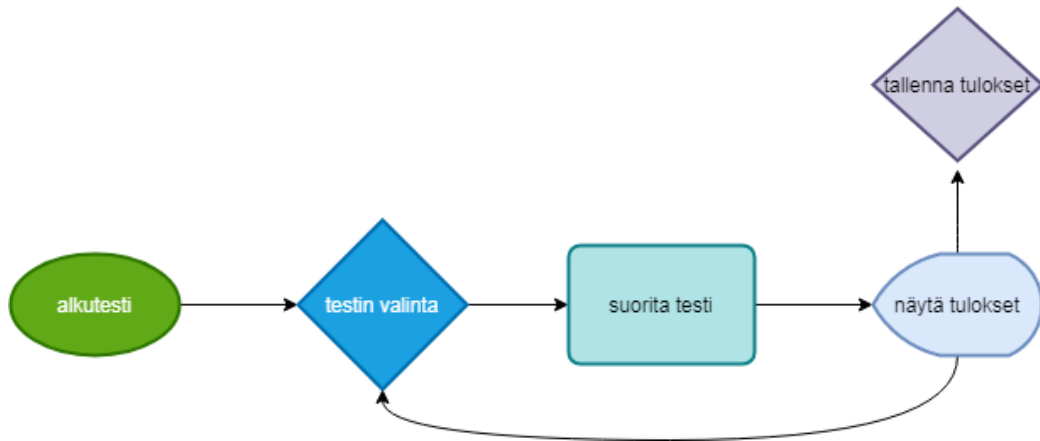


Kuva 27. Projektin luokkien hierarkia [25]

Ohjelmistokehys (Engine) sisältää TouchGFX:n luomat kantaluokat luotaville näyttöikkunoille. Generated-tasolla on TouchGFX:n koodigeneraattorin luomat aliluokat, jotka perityvät kantaluokastaan. TouchGFX luo edellä mainitut luokat, eikä niitä tulisi editoida, koska ne generoidaan uudelleen, mikäli GUI:hin tehdään muutoksia. Sen sijaan kaikki käyttäjän haluama toiminnallisuus toteutetaan User- luokissa, joihin TouchGFX ei tee mitään muutoksia [25].

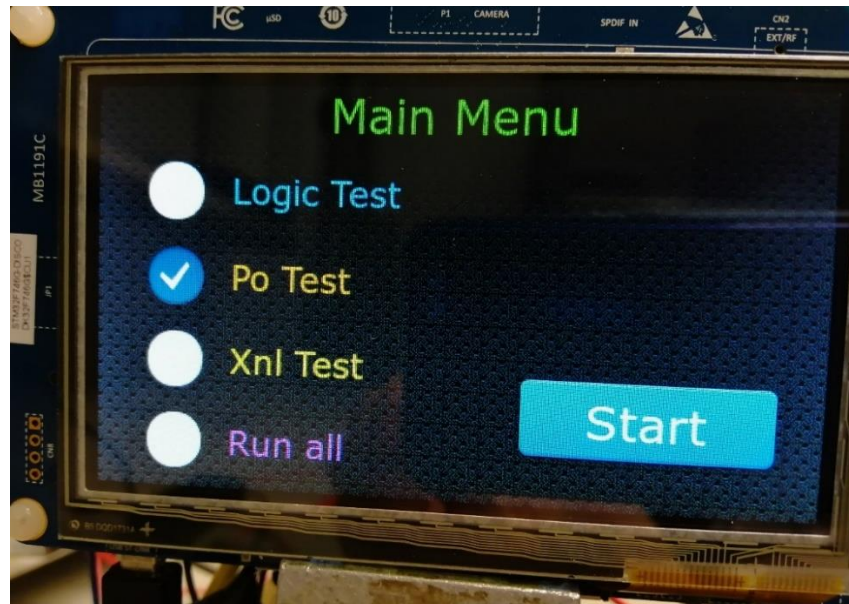
4.4 Ohjelman rakenne ja toiminta

Kuva 28 on vuokaavio ohjelman toiminnasta. Laitteen käynnistyessä suoritetaan tarvittavat alustukset oheisliitännöille sekä muisteille. Käynnistyksen yhteydessä testataan moduulin sisäisen regulaattorin toiminta, sekä tehdään ohjainkortin ohjausjännitteille muutama testimittaus, joilla testataan PWM- ohjauksien toiminta. Aloitusruudulle tulee tieto näiden testien onnistumisesta. Tämän jälkeen aloitusnäyttö vaihtuu ohjelman päävalikkoon, josta valitaan haluttu testi. Testin suorituksen jälkeen mittaustulokset sekä tieto testin läpäisystä tulevat näytölle. Tässä vaiheessa on mahdollisuus tallentaa mittaustulokset muistikortille. Haluttaessa voidaan siirtyä eri mittaustulosten välillä, palata päävalikkoon ja suorittaa testi uudelleen tai valita uusi testi.



Kuva 28. Lohkokaavio ohjelman toiminnasta.

Ohjelman päävalikosta (Kuva 29) valitaan suoritettava testi. On myös mahdollista valita kaikki testit suoritettavaksi peräkkäin.



Kuva 29. Ohjelman päävalikko.

4.4.1 PWM-ohjausarvojen laskenta

PWM-ohjausarvojen laskemisessa lähtötietoina tarvitaan kytkennän vahvistus ja PWM-resoluutio. Järjestelmässä on 2 PWM-ohjausjännitettä, joiden suunnittelu on käsitelty kappaleissa 3.2.1 ja 3.3. PWM:n resoluution määrittäminen on esitetty kappaleessa 3.1.2. Ohjausjännitteiden PWM arvot on laskettu kaavan (25) mukaan:

$$V_{out} = DutyCycle_{PWM}[\%] * G_A V_{REF} \leftrightarrow \frac{CCR_x}{ARR_x} [\%] = \frac{V_{out}}{G_A V_{REF}} \leftrightarrow CCR_x = \frac{ARR_x V_{out}}{G_A V_{REF}} \quad (25)$$

, missä G_A on kytkennän vahvistus, V_{REF} on 3,3 V, CCR_x on Capture/Compare rekisteriin sijoitettava arvo, ARR_x on PWM-resoluutio ja V_{out} on haluttu ulostulojännite. Arvot on ensin laskettu, jonka jälkeen yleismittarilla ja oskilloskoopilla mitattiin kalibrointimittaus ja tarvittaessa säädettiin arvot kohdalleen.

4.4.2 Jännitteen skaalaus AD-muuntimelle

Laite mittaa A11-moduulin ulostulojännitteiden lisäksi myös omia ohjausjännitteitään. Mitattavat jännitteet pitää skaalata mikrokontrollerille sopivalle välille 0 - 3,3 V. Raja-arvot, joiden sisällä jännitteiden tulee olla, on määritelty annettujen testispesifikaatioiden mukaan. Raja-arvot on laskettu kaavan (26) mukaan seuraavasti:

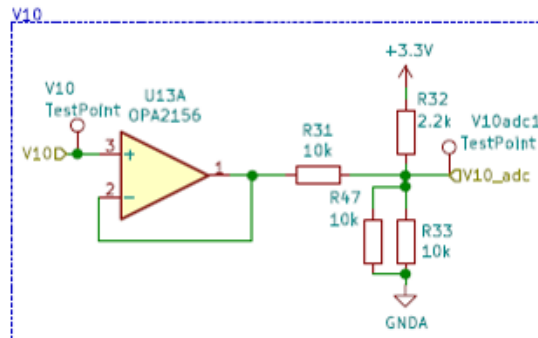
$$\text{Measured Voltage} = \frac{\text{ADC reading} * V_{ref}}{\text{ADC resolution} * G_{fb}} \quad (26)$$

, missä AD- muuntimen resoluutio on 4095 ja G_{fb} on skaalauskerroin, joka saadaan vastusjaon perusteella (Kaavat 19 ja 22). Eri jännitelähteille on omat kytkentänsä, jolloin myös skaalauskerroin on eri (Liite A). AD-muunnoksen lukemat saadaan tutkimalla CubeIDE:n debuggerilla AD-muuntimen toimintaa. Ohjelman toimintaa suoritetaan yksi mittaus kerrallaan. Jokaisen mittauksen AD-muunnoksen arvot merkitään muistiin ja samalla oskilloskoopilla ja yleismittarilla mitataan jännitteen referenssiarvo. AD-muuntimen arvoista lasketaan arvot mitatuille jännitteille kaavan (26) mukaan. Referenssiarvoja verrataan mittaustuloksiin ja lineaarisen regressioanalyysin perusteella määritetään mittauksen korjauskerroin. Mittauksissa on huomioitava, että vahvistuksessa on epälineaarisuutta ja korjauskertoimen voi joutua määrittämään erikseen kunkin mittauksen osalta. Kun mittauksen referenssiarvoja vastaavat AD-muunnoksien arvot on määritelty, lasketaan raja-arvot sallituille poikkeamille, jotka on ilmoitettu testispesifikaatiossa.

Ohjelmalistauksen 2. Po-testissä on kolme mitattavaa jännitettä: V7, V8 ja V9. Näistä V7 ja V9 omaavat samat skaalauskerroimet. V8 on invertoitu jännite V7:stä. V8:n dynaaminen jännitealue on siten 0...-10,25 V ja sen skaalaus AD-muuntimelle tapahtuu helpoiten invertoivalla operaatiovahvistinkytkennällä (Liite A). Valituilla vastusarvoilla laskennallinen V7:n skaalauskerroin on 0.321956 ja V8:n 0,318182.

4.4.3 Negatiivisen jännitteen skaalaus AD-muuntimelle

Useat järjestelmän mitattavista jännitteistä saavat sekä positiivisia- että negatiivisia jännitearvoja, joten niiden skaalaus AD-muuntimelle ei onnistu invertoivalla operaatiovahvistimella. Negatiivisten jännitteiden skaalaukseen käytetään kappaleessa 3.5 esitettyä kolmen vastuksen kytkentää. Esimerkkinä kytkennässä sisääntulojännitteen (V10) jännitealue on -15V...9,9V (Kuva 30).



Kuva 30. Kytkentä sisääntulojännitteen -15...9,9V skaalaamiseksi välille 0...3,3V.

Kytkenän raja-arvot lasketaan samaan tapaan, kuin kappaleessa 4.4.2 tuotiin esille. Laskennasta tulee ottaa huomioon se, että AD-muunnoksen arvo 0 vastaa -15,0 V, joten tuloksesta vähennetään 15,0 V:n offset-jännite (28).

$$\text{Measured Voltage} = \frac{\text{ADC reading} * V_{ref}}{\text{ADC resolution} * G_{fb}} - V_{offset} \quad (27)$$

Kytkenän skaalauskerroin G_{fb} lasketaan kaavan (23) mukaan. Esimerkin komponenttiarvoilla kytkennän skaalauskerroimeksi saadaan 0,132530.

4.4.4 Mittaussäie

Ohjelma sisältää kolme säiettä. Yksi mittauksille ja toinen LCD -näytön dataliikenteelle. Kolmas säie on FreeRTOS:in oma Idle-task (tyhjäkäynti). Seuraavassa mittaussäie:

Ohjelma 1.

```
void StartTestTask(void const * argument)
{
    uint32_t NotifyValue=0;

    V6Test();

    for(;;)
    {
        xTaskNotifyWait(0, 0xffffffff, &NotifyValue, portMAX_DELAY);
```

```

        if ((NotifyValue & 0x01) != 0)
        {
            LogicTest();
        }
        else if ((NotifyValue & 0x02) != 0)
        {
            PoTest();
        }
        else if ((NotifyValue & 0x04) != 0)
        {
            XnlTest();
        }
        else if ((NotifyValue & 0x07) != 0)
        {
            RunAll();
        }
    }
}

```

Funktio V6Test on alkutesti, joka suoritetaan kertaalleen ohjelman käynnistyessä. Alkutesti mittaa A11-moduulin regulaattorin ulostulojännitteen sekä ohjausjännitteiden toiminnan. Muutoin säie odottaa kosketusnäytöltä tulevaa viestiä (NotifyValue). Viesti nollataan, kun se on luettu, jotta testin suorittavaa funktiota ei kutsuta uudestaan. Säikeessä itsessään ei ole montakaan riviä koodia, vaan toteutus on sisällytetty sen kutsumiin funktioihin.

Seuraavassa ohjelmalistauksessa on esimerkkinä funktio PoTest.

Ohjelma 2.

```

void PoTest() {

    uint32_t V7Po[12]={0,875,899,920,1070,1265,1461,1634,1645,1665,1951,4886}; //PWM
    // Ylä- ja alarajojen raja-arvot
    uint32_t V7bLow[12]=
        {0,700,719,737,859,1019,1179,1318,1329,1345,1578,3976};
    uint32_t V7bHigh[12]=
        {20,739,759,777,899,1059,1219,1359,1369,1385,1618,4016};
    uint32_t V8Low[] =
        {0, 692, 711, 729,850,1008,1166,1305,1315,1331,1562,3936};
    uint32_t V8High[] =
        { 20,732,752,769,890,1048,1206,1345,1355,1371,1602,3976};
    uint32_t V9Low[] =
        { 0, 0, 0, 0, 765, 1809, 2847, 3763, 3820, 3925, 3884, 3884 };
    uint32_t V9High[] =
        { 16, 16, 16, 40, 845, 1898, 2950, 3870, 3936, 4017, 4082, 4082};
    uint32_t V7b[12],V8[12],V9[12];
    uint32_t buffer[12][3] ={0};

    // MUX settings for V7, V8, and V9
    HAL_GPIO_WritePin(GPIOG, GPIO_PIN_6, GPIO_PIN_RESET); //ENABLE MUX
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_4, GPIO_PIN_RESET); // A D3
    HAL_GPIO_WritePin(GPIOG, GPIO_PIN_7, GPIO_PIN_SET); // B D4
    HAL_GPIO_WritePin(GPIOH, GPIO_PIN_6, GPIO_PIN_SET); // C D6

    //HAL_TIM_Base_Start(&htim5);
    for(int i=0; i<12; i++)

```

```

    {
        htim1.Instance->CCR1=V7Po[i]; //arvoksi V7Po[i]
        HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1); //Pin D10
        osDelay(200); // pitkä 200 ms viive
        HAL_ADC_Start_DMA(&hadc3, buffer[i], 3);

// Otetaan 4096 arvoa jokaisesta mitatusta arvosta
        for (int k=0; k<4096; k++)
            for (int k=0; k<3; k++) {
                results.PoAdcValues[i][k] += buffer[i][k];
            }
    }
    osDelay(100);
    HAL_ADC_Stop_DMA(&hadc3);
    htim1.Instance->CCR1=0; //PWM stop
// Jaetaan tulokset 4096 keskiarvon laskemiseksi.
    for (int a=0; a<3; a++) {
        for (b=0; b<12; b++) {
            results.PoAdcValues[b][a] = results.PoAdcValues[b][a]>>12;
        }
    }

// Järjestetään mitatut arvot omiin taulukoihin
    for (int k=0; k<12; k++)
    {
        V7b[k]=results.PoAdcValues[k][0];
        V8[k]=results.PoAdcValues[k][1];
        V9[k]=results.PoAdcValues[k][2];
    }

// Testataan ovatko arvot sallituissa rajoissa
    for (int j=0; j < 12; j++)
    {
        if ((V7b[j] <= V7bHigh[j]) && (V7b[j] >= V7bLow[j])
            &&(V8[j] <= V8High[j]) && (V8[j] >= V8Low[j])
            && (V9[j] >= V9Low[j]) && (V9[j] <= V9High[j]))
        {
            results.PoResults[j] = 1;
        }
        else
        {
            results.PoResults[j] = 0;
        }
    }
    xQueueSend(DataQ,&presults,portMAX_DELAY); //lähetetään tietueen
osoite jonoon
    return;
}

```

AD- muunnoksen näytteistysajan voi valita väliltä 3 - 480 kellojaksoa [6], joka tässä tapauksessa asetettiin maksimiin tarkkuuden parantamiseksi. AD -muunnos suoritetaan kertapyyhkäisynä (scan mode, single conversion), jolloin AD-muunnos suoritetaan peräkkäin 3 eri kanavalle. Tiedonsiirto muistiin tapahtuu DMA:n avulla. DMA ei käytä prosessorin resursseja vaan hoitaa datan siirron itsenäisesti [6,13].

Mittaustulokset tallennetaan globaaliin tietueeseen, jonka määrittely on alla. Tietueen koko on 365 tavua (byte), ja se lähetetään osoittimena jonoon. Vastaanottajana on kosketusnäytön rajapintaluokka (model).

```
#ifndef INC_RESULTS_H_
#define INC_RESULTS_H_

typedef struct results{
    uint32_t PoAdcValues[12][3];
    uint32_t LogicTestAdc[2][3];
    uint32_t XnlAdcValues[14][3];
    uint8_t PoResults[12];
    uint8_t XnlTestResults[14];
    uint8_t LogicTestResults[2];
    uint8_t selfChkResult[1];
}Results_t;

#endif /* INC_RESULTS_H_ */
```

Tieto testin onnistumisesta lähetetään lukuarvona 0 tai 1. Se muunnetaan tekstiksi vasta tulosten näytön yhteydessä.

5. TULOSTEN ANALYYSINTI

Tuotekehityksessä hyvin usein ensimmäinen prototyyppi tai vielä toinenkaan ei osoittaudu täysin toimivaksi versioksi. Ne voivat olla silti erittäin hyödyllisiä tuoden esiin asioita, joita ei suunnitteluun käytettävän rajallisen ajan ja resurssien puitteissa osata ottaa huomioon. Projektin aikana rakennettiin kaksi eri prototyyppiversiota. Tässä luvussa esitellään näiden tuloksia ja havaittuja asioita.

5.1 Ensimmäinen tuotekehitysversio

Projektin edetessä ilmeni monia asioita, joita ei osattu etukäteen huomioida, ja jotka paljastuivat vasta testauksessa. Ensimmäisessä prototypissä oli useita suunnittelu- ja olettamavirheitä, jotka paljastuivat komponenttien asennuksessa tai vasta testauksessa. Ensimmäinen asia ilmeni komponenttien kalustusvaiheessa. Kahden tehonsyötön regulaattorin layout oli väärä. Piirilevyn layout ja komponentin läpivientireiät oli tehty TO247-tyyppiselle layoutmallille (footprint), vaikka tarkoitus oli käyttää TO-220 tyyppistä. Regulaattorien jalkoja ei voinut taivuttaa, joten niiden kiinnittämiseksi jalkoja jatkettiin kuparilangalla.

Toinen suunnitteluvirhe oli operaatiovahvistimien negatiivisen käyttöjännitteiden puuttuminen niistä operaatiovahvistimista, joiden ei tarvitse tuottaa tai käsitellä negatiivisia jännitteitä. Toimiakseen oikein työssä käytetty operaatiovahvistin OPA2156 tarvitsee sekä positiivisen, että negatiivisen käyttöjännitteen. Tämä korjattiin käyttämällä hyppylankoja negatiivisen jännitteen reitittämiseen operaatiovahvistimille.

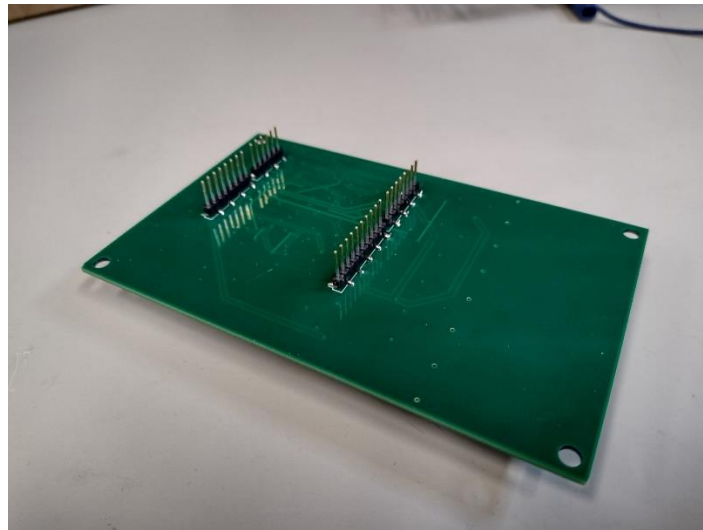
Kappaleessa 3.2.1 esitetty Sallen-Key aktiivisuodatin oli suunniteltu virheellisesti 3,1 jännitevahvistuksella, jolloin sen ulostulo oli epästabiili. Tämän vuoksi suodatinkytkeä tehtiin koekytkelevylle, joka yhdistettiin hyppylangoilla piirilevylle.

Seuraava asia mitä ei osattu ottaa huomioon oli, että A11-moduulin lähtöjännitteiden arvot olivat oletettua suuremmat, kuin mitä testispesifikaation perusteella pääteltiin. Jos testispesifikaatiossa oli ilmoitettu jonkin signaalin jännitealueeksi esimerkiksi -9,0 V...10.0 V, jännite saattoi silti olla selvästi yli ilmoitettujen arvojen. Tällöin jännitteen skaalaus mikrokontrollerille ei toimi oikein. Suunnittelun mitta-alueen ylittävät jännitteet satureituvat AD-muuntimen näytteistyksessä, jolloin niiden todellista jännitettä ei pystytä mittaamaan. Mittausalueen ylittäminen aiheuttaa myös sen, että mikrokontrollerin AD-muuntimen sisääntulossa näkyy joko ylijännite tai negatiivinen alijännite, mikä saattaa vaurioittaa mikrokontrolleria.

Testaamista hankaloittavana tekijänä oli, että liittimet joihin moduuli kytkettäisiin, sekä näyttökortin liittimet olivat sijoitettu samalle puolelle levyä. Tämä teki mittapisteiden mittaamisen erittäin hankalaksi tai jossain kohtaa mahdottomaksi.

5.2 Toinen tuotekehitysversio

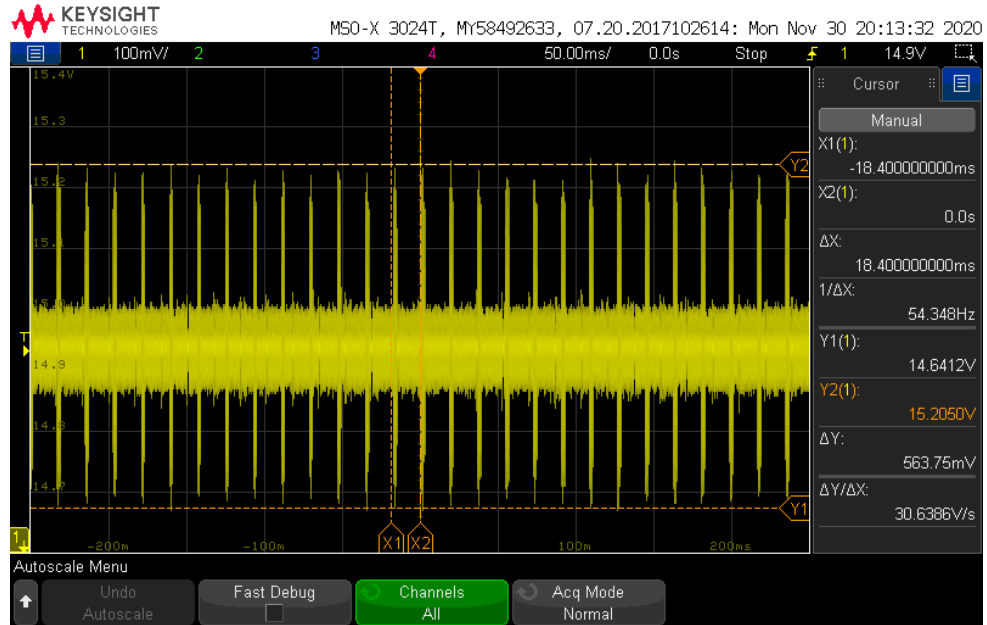
Toiseen prototyypiversioon kaikki edellä mainitut ongelmat korjattiin (Liitteet A, B ja C). Jännitesyötön regulaattorit vaihdettiin SMD-tyyppisiin. PWM-suodatukseen tehtiin muutokset, jotka on kuvattu tarkemmin kappaleessa 3.3.1 ja kytkentä mitoitettiin ottaen huomioon riittävän suuret jänniterajat. Layout suunniteltiin uusiksi siten, että toiselle puolelle tuli näyttökortin liittimet ja levyn toiselle puolelle kaikki muu, jolloin testipisteisiin päästiin helposti käsiksi (Kuva 31). Testipisteiden määrää myös lisättiin.



Kuva 31. Piirilevyn etupuolella vain liitännät STM-kortille.

Tehdyistä korjaustoimenpiteistä huolimatta järjestelmässä esiintyi edelleen ongelmia. Jotkin havaitut viat tai häiriöt olivat myös hankalasti paikannettavia ja niiden löytäminen oli erittäin työlästä. Eräs vaikeasti paikannettava asia oli n. 300 mV:n suuruiset häiriöpiikit, jotka esiintyivät järjestelmässä. Näiden lähde yritettiin selvittää poissulkemalla asioita yksi kerrallaan. Asiaa tutkittua häiriönlähteen epäiltiin olevan työssä käytettävä laboratoriotehonlähde, mutta edes tämän vaihto ei korjannut ongelmaa. Lopulta häiriön aiheuttajaksi paljastui laboratorion pöytävalaisin, jonka tuottamat häiriöt kulkeutuivat mittaussuunnitelmaan johtamalla sähköverkon ja tehonlähteen kautta. Kytkemällä valaisin pois päältä nämä häiriöt poistuivat. Tämä oli hyvä esimerkki siitä, että pelkkä CE-merkintä ei takaa laitteen laatua. Tämän häiriön löytämiseen meni valitettavan paljon aikaa ja työtä.

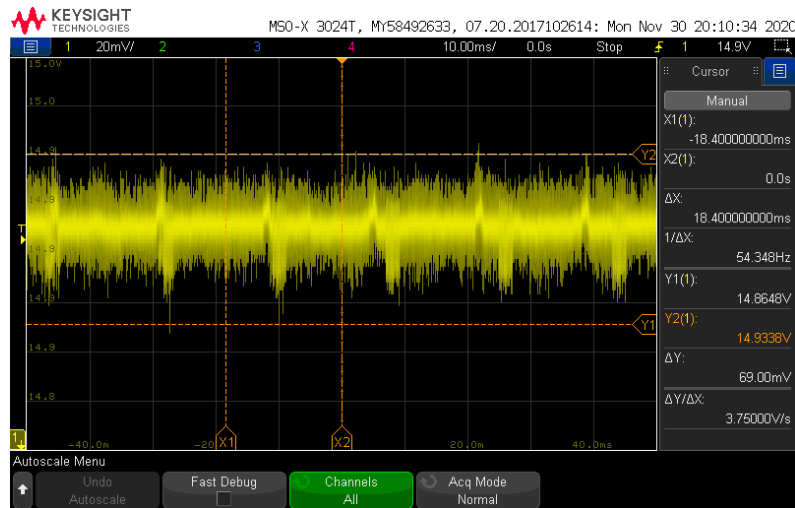
Ongelmallisin asia oli käyttöjännitteissä näkyvä n. 55 Hz jaksollinen häiriösignaali, kun koko järjestelmä oli toiminnassa (Kuva 32). Todennäköisesti häiriön lähteenä on STM-kortin näytön taustavalon ajuriipiiri. Kytettäessä järjestelmä päälle menee aikaa n. sekunti, kun näyttö käynnistyy ja samaan aikaan oskilloskoopin näytölle ilmaantuu näkyviin kyseinen häiriösignaali. Häiriöpiikit ovat jopa yli 0,5 V ja ne aiheuttivat pahoja ongelmia mittaustuloksissa.



Kuva 32. Käyttöjännitteessä näkyvä häiriö.

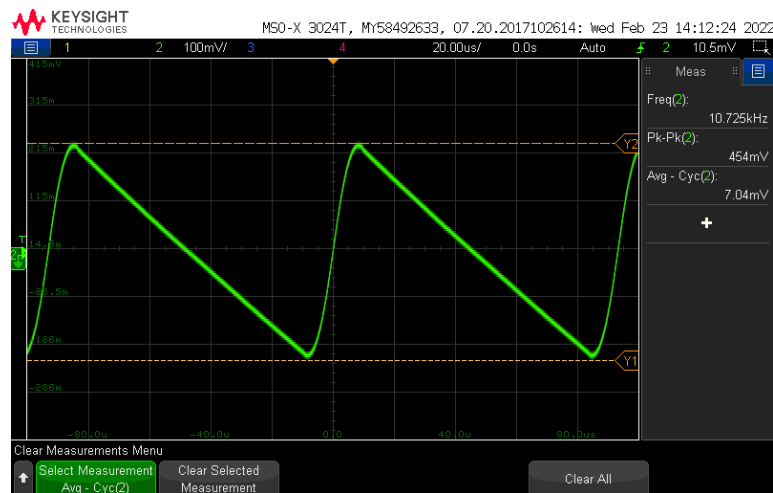
Lopulta sitä saatiin merkittävästi pienennettyä lisäämällä +15V:n ja +12V:n käyttöjännitteiden syöttöön 470 μ F:n elektrolyyttikondensaattorit (Kuva 33).

Häiriöt näkyivät myös 12V:n regulaattorin ulostulossa, joka menee näyttökortille (Liite F). Kokonaan sitä ei kuitenkaan saatu poistettua. Häiriölähdettä selvitettyä testattiin kortteja myös erikseen. Kehitysalustakortti irrotettiin mittakortista ja A11-moduulista, jolloin kyseisiä häiriöitä ei esiintynyt lainkaan (Liite G).



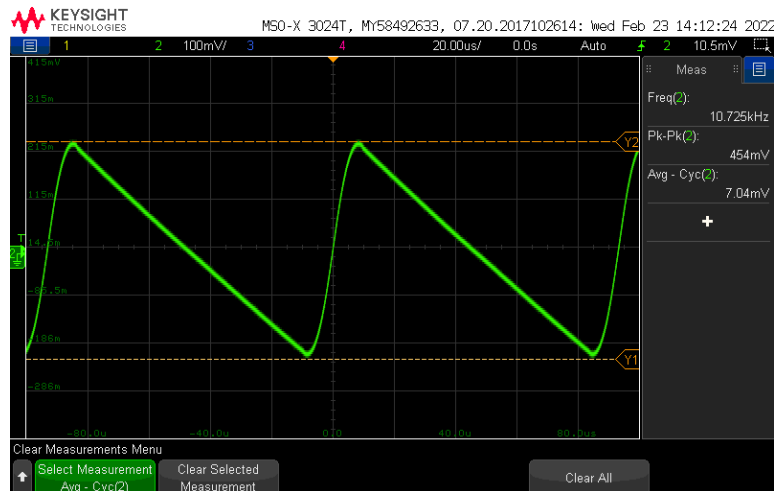
Kuva 33. +15V käyttöjännite, lisätty 470uF kondensaattori.

Kolmas asia oli yhden testimittauksen tulokset, joissa ilmeni erityisen paljon varianssia tietyillä ohjauksjännitteillä. A11-moduulin testispesifikaatioissa oli mainittu vain DC-arvoja, joiden rajoissa ulostulojen tulisi olla. Niinpä suunnittelussa oletettiin ulostulojännitteen olevan DC- jännitetä. Ongelma oli siinä, että moduulin ulostulojännite tietyillä ohjauksjännitteillä ei ollut DC-jännitetä. 0 V:n ohjauksella ulostulojännitteen aaltomuoto oli n. 10,7 kHz:n kolmioaaltoa (Kuva 34).



Kuva 34. V9:n ulostulojännite 0V ohjauksella.

Jännitteen amplitudi asettui 0 V:n tason molemmin puolin siten, että sen aritmeettinen keskiarvo oli lähes 0 V. Eri mittauskerroilla tämä kuitenkin vaihteli, jolloin läheskään aina tulos ei näyttänyt nolaa. Ohjauksjännitettä kasvattamalla ulostulo asettui DC-jännitteeksi. Kun ohjausta lisättiin n. 6.4 V:in, ulostulojännite muuttui jälleen kolmioaaloksi (Kuva 35).



Kuva 35. Ulostulojännite V9, ohjausjännite > 6,4V.

Riippuen mittauksen ajanhetkestä, tulos saattoi olla yli ± 200 mV ilmoitetusta DC-jännitteen tavoitearvosta. Tätä vaihtelua oli ilmeisesti yritetty ottaa huomioon testispesifikaatiossa, koska raja-arvot oli asetettu juuri ± 200 mV tavoitearvosta. Eri mittauskertojen välillä saattaa tuloksissa esiintyä huomattavia eroja riippuen ajanhetkestä, milloin AD-muuntimen näytteistys tapahtuu.

Mittaustulosten ja raja-arvojen kalibrointi kohdalleen osoittautui työlääksi toimenpiteeksi. Jokainen ADC:n arvo piti tutkia debuggerilla yksitellen ja mitata käsin yleismittarilla sekä oskilloskoopilla sitä vastaava jännitelukema. Tämän jälkeen tuloksista laskettiin uudet arvot korjauskertoimelle, tehtiin tarvittavat muutokset ja suoritettiin uudet mittaukset. Tätä iterointia jouduttiin tekemään useita kertoja jokaiselle eri mittaukselle. Järjestelmässä esiintyvän kohinan ja häiriöiden takia jotkin mittaukselliset tulokset ovat epätarkkoja. Varsinkin mitä lähempänä laitteen toiminta on raja-arvoa, sen helpommin mitatut arvot ajautuvat mittaushetkellä rajojen ulkopuolelle, vaikka yleismittarilla mitattuna arvo olisi edelleen rajojen sisällä. Yleismittari keskiarvottaa mittauksen ja testispesifikaatiossakin oli määritelty vain DC-jännitearvoja, joten ohjelmaan sisällytettiin vielä mittausten keskiarvon laskeminen. Jokaisesta mitatusta jännitteestä lasketaan 4096:n näytteen aritmeettinen keskiarvo. Sopiva määrä mittauksille löydettiin kokeilemalla eri parametrioita. Tällöin parannusta keskihajontaan ei havaittu enää mittaustuloksissa. Huolimatta näytteiden keskiarvottamisesta tuloksissa esiintyi edelleen epätarkkuutta eri mittauskertojen välillä. Syynä on todennäköisesti häiriöiden kulkeutuminen mikrokontrollerin AD-muuntimen referenssijännitteeseen, jonka vaihtelu aiheuttaa keskihajontaa tuloksissa. Lisäksi häiriöt kulkeutuvat myös muualle järjestelmään sekä mitattavaan A11-moduuliin ja sitä kautta moduloivat koko järjestelmän toimintaa, mikä heijastuu mittaustuloksiin.

Järjestelmän tehonkulutus on noin 10W (Kappale 3.6). Laitteiston ollessa kauan päällä siinä esiintyi jonkin verran lämpötilaryömintää, joka hankaloitti tarkkojen asetusten löytämistä. Kun edellisenä päivänä oli tehty kalibrointi ja säädetty kaikki kohdalleen, seuraavana päivänä niissä todettiin olevan jälleen epätarkkuutta. Aiheutuva ero mittaustuloksiin on suuruusluokaltaan n. 10mV. Kuva 36 on esillä mittaustulokset, jotka ovat menneet rajojen sisään. Mikäli mittausta tehdäisiin uudestaan, jotkin tulokset todennäköisesti olisivat yli rajojen. Mittaustulokset eivät siis ole riittävällä varmuudella toistettavia, eikä lentokelpoisuushyväksyntätestejä valitettavasti voida suorittaa tällä mittalaitteella. Testispesifikaatioiden erittäin tarkat raja-arvot herättivät jonkin verran skeptisyyttä niiden tarkoituksenmukaisuudesta.



Kuva 36. Mittaustuloksia

5.3 Kehitysehdotuksia

Laitteen jatkokehitys sellaiseksi, että voitaisiin varmistua sen antamien tulosten oikeellisuudesta ja siten todentaa A11-moduulin lentokelpoisuus, vaatisi vielä ainakin yhden prototyypikierroksen. Seuraavassa on kehitysehdotuksia sekä mitä asioita seuraavan prototyypin suunnittelussa on suositeltavaa ottaa huomioon.

- Suurin ongelma on järjestelmässä esiintyvät häiriöjännitepiikit ja kohina, jotka tulee minimoida laitteen suorituskyvyn parantamiseksi. Jaksollisten 55 Hz:n

häiriöjännitepiikkien poistamiseen voitaisiin kokeilla digitaalista suodatusta, jolla tehtäisiin jyrkkä kaistanestosuodatus kyseiselle taajuusalueelle.

- Mikäli digitaalisella suodatuksella ei pystytä ratkaisemaan ongelmaa, kehitysalustakortista tulisi luopua kokonaan ja suunnitella näyttökortti itse. Kaikki ylimääräiset ja tarpeettomat liitännät sekä oheislaitteet voisi ottaa pois. Näytön ja tarvittavan mittauselektronikan integrointi samalle kortille mahdollistaisi paremman suunnittelun maavirtojen reititykselle häiriöiden minimoimiseksi.
- Mittauselektronikka kannattaa suunnitella siten, että laite vaihtaa automaattisesti sopivaan mitta-alueeseen. Periaate on samantapainen, kuin esimerkiksi joissain digitaalisissa yleismittareissa. Tämä mahdollistaisi paremman mittausresoluution, sekä laitteen soveltuvuuden eri moduulien mittauksiin.
- Mikrokontrollerin 12-bittisen AD- muuntimien sijasta voitaisiin käyttää ulkoista tarkempaa AD-muunninta. Tämä yhdistettynä digitaaliseen signaalinkäsittelyyn (suodattamiseen) antaisi tarkempia tuloksia. Laskennan ei edes tietysti tarvitse tapahtua reaaliaikaisesti, joten on mahdollisuus käyttää korkea-asteisia suodattimia.
- Laitteen käytettävyyden kannalta se tulisi ehdottomasti koteloida. Liitännät mitattavaan moduulin voisi toteuttaa kappaleessa 3.9 kuvatuilla Pogo-pinneillä. Toisaalta tämä ei ehkä olisi paras ratkaisu, jos laitteesta halutaan tehdä monikäyttöinen. ECU-moduulit ovat ulkoisesti samanlaisia, mutta niiden pinnijärjestys on eri, joten ei voida olettaa esimerkiksi käyttöjännitepinnien olevan samoja. Monikäyttöisyyden kannalta parempi ratkaisu on, jos laite varustetaan riittävällä määrällä ohjausjännitteitä ja sisääntuloja, joiden toiminta määritellään ohjelmallisesti.
- Lämpötilaryömintää voisi pienentää käyttämällä tarkkaa AD-muuntimen lämpötilakompensoitua referenssijännitettä. Laitteen koteloon asennettava tuuletin olisi myös suositeltavaa.
- Tarkkuusvastuksien käyttöä tärkeämpää on suorittaa mittauksille kalibrointi ja määrittää korjauskertoimet. Mittausketjussa olevilla vastuksilla tulisi olla alhainen ja sama lämpötilakerroin.
- Sähkönsyötön muuttaminen akkukäyttöiseksi. Järjestelmän maksimi virrankulutukseksi mitattiin n. 650 mA (mukaan lukien mitattava moduuli), joten esimerkiksi 2400 mAh akkukapasiteetti olisi riittävä.

- LCD-kosketusnäyttö voisi olla vielä suurempi kuin 4,3". Ohjelmaan voisi myös lisätä numeeristen tulosten lisäksi kuvaajan, joka näyttäisi ulostulojännitteen ohjausjännitteen funktiona.
- Testituloksien näyttämisen voisi muuttaa niin, että mikäli mittaustulos ei mene läpi, tuloksista näkyisi suoraan mikä tai mitkä jännitteet eivät läpäisseet testiä. Nyt testitulos näkyy vain 3:n mittauksen ryhmä kerrallaan.
- Ohjausjännitteiden tarkkuuden parantamiseksi ohjelmaan voitaisiin lisätä PID-säätöalgoritmi.

6. YHTEENVETO

Tässä diplomityössä suunniteltiin HN ECU A11-moduulille puoliautomaattisen testilaitteen prototyyppi. Työ toteutettiin suunnitteleamalla A11-moduulin ja STM32F746-kehitysalustakortin välille mittauselektronikkakortti. Työssä käydään läpi laitesuunnittelun eri vaiheita, ja perehdytään valittujen toteutustapojen teoriaan ja perusteisiin. Laitteen ohjelmistosta esitetään vain pääpiirteet sen laajuuden vuoksi.

Tuotekehitysprojektit sisältävät aina jonkin asteisia riskejä. Joko lopputulos ei vastaa odotuksia tai niiden saavuttaminen vaatisi enemmän aikaa ja rahaa, kuin on varauduttu käyttämään. Tämän projektin osalta saatiin vastaus ainakin siihen, että tarkan mittausjärjestelmän suunnittelu tiettyä HN ECU-moduulia varten on haasteellista. Mikäli laite soveltuu vain tietyn tyyppisen moduulin mittaamiseen, se on todennäköisesti myös taloudellisesti kannattamatonta. Projektiin käytettävän aikataulun puitteissa laitteesta ei saatu sellaista, että sillä olisi pystytty tekemään A11-moduulille tarvittavia lentokelpoisuushyväksyntätestejä. Laitteella pystyisi kyllä tekemään suuntaa antavia mittauksia, mutta järjestelmässä olevan kohinan vuoksi sen tarkkuus ja mittauksen toistettavuus eivät ole riittävällä tasolla. Tämän vuoksi toteutusta ei tuotteistettu ja sen kotelointi jäi suunnitteluasteelle.

Järjestelmän suurin ongelma johtuu STM32F746NG-kehitysalustakortista, jonka ominaisuudet tarkan ja kohinattoman mittajärjestelmän alustaksi paljastuivat riittämättömiksi. Ongelman lähde on kehitysalustakortin näytön taustavalonajuripiiri, jonka tuottamat jaksolliset häiriöt kulkeutuvat ympäri järjestelmää. Häiriöt ilmenevät, kun näyttökortti ja ohjauselektronikkakortti yhdistetään, mutta erikseen mitattuna niitä ei havaita.

Laitteesta saataisiin todennäköisesti toimivampi, jos kehitysalustakortin sijasta näyttöohjainkortti suunniteltaisiin itse. Erityisesti maavirtojen kulkureitit tulisi suunnitella huolellisesti. Lisäksi mittauselektronikka kannattaisi toteuttaa siten, että laite automaattisesti valitsisi sopivan mittausvälin, kuten monet digitaaliset yleismittarit. Tämä mahdollistaisi laitteen monikäyttöisyyden, eikä sitoisi sitä vain yhteen tiettyyn moduuliin.

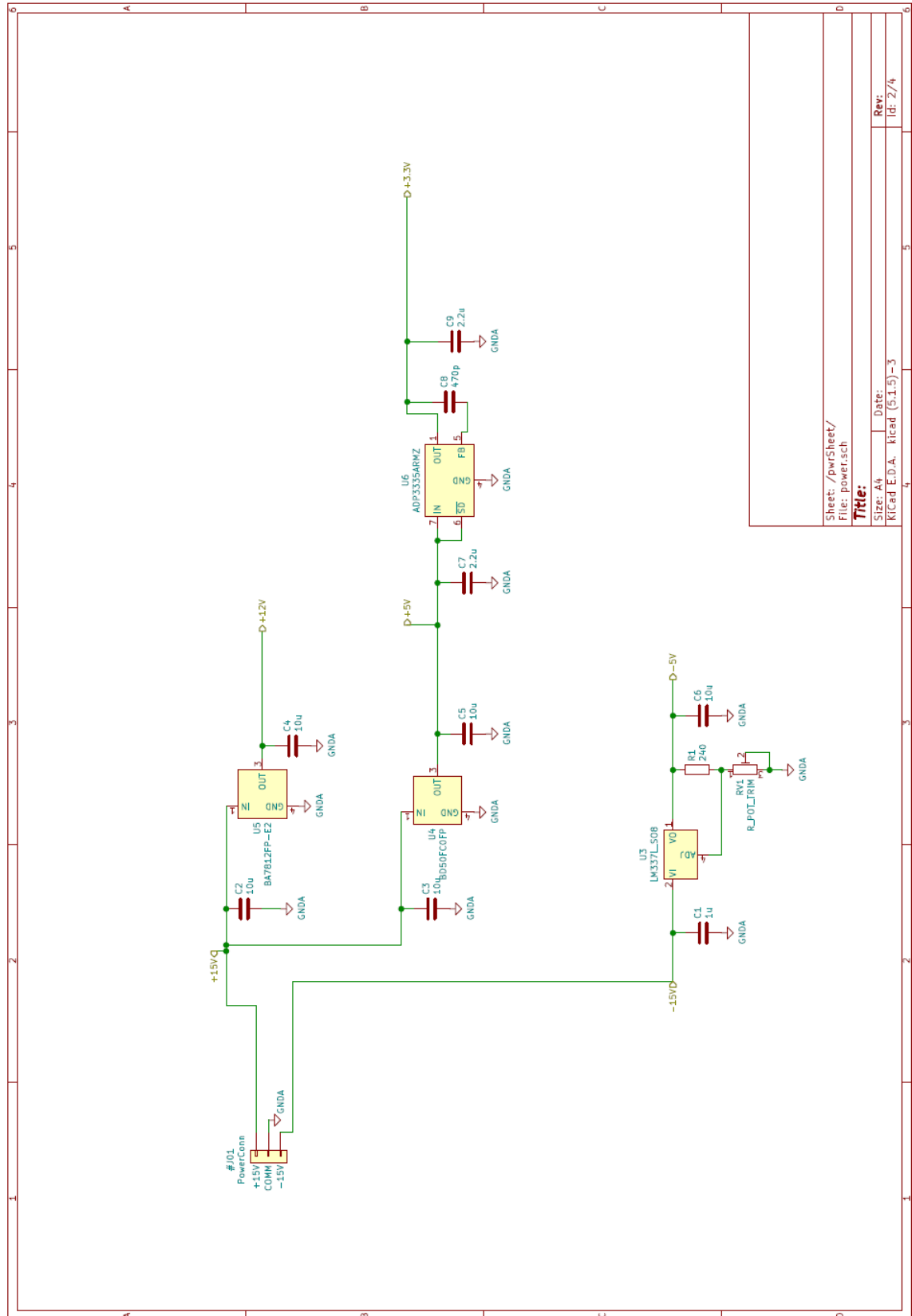
Monikäyttöisen ja helposti muokattavan mittajärjestelmän saisi tehtyä myös Labview ympäristöön, joka oli yksi vaihtoehto tämän projektin toteutustapaa mietittäessä. Se vaatisi tietysti alkuinvestoinnin mittalaitteisiin ja tietokoneeseen sekä ohjelmistolisenssin hankkimisen, mutta suorituskyky olisi varmasti riittävä ja ohjelma muokattavissa suhteellisen helposti monikäyttöiseksi.

LÄHTEET

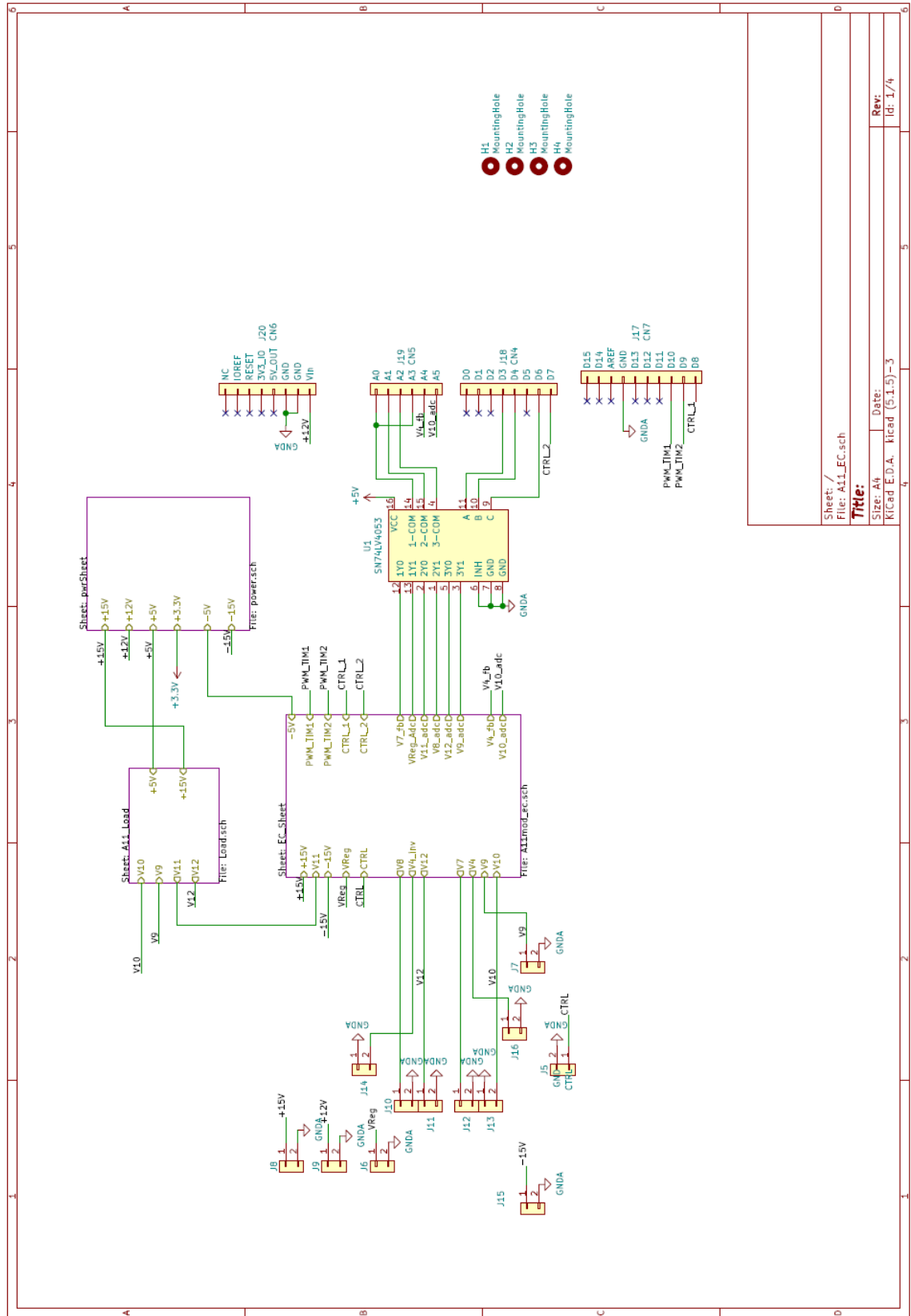
- [1] M. Ohring, L. Kasprzak, Reliability and failure of electronic materials and devices, 2nd ed. Academic Press is an imprint of Elsevier, Amsterdam, 2015.
- [2] R.R. Tummala, Fundamentals of microsystems packaging, McGraw-Hill, New York, 2000.
- [3] User manual, Discovery kit with STM32F469NI MCU, https://www.st.com/resource/en/user_manual/dm00218846-discovery-kit-with-stm32f469ni-mcu-stmicroelectronics.pdf.
- [4] User manual, Discovery kit for STM32F7 Series with STM32F746NG MCU, https://www.st.com/content/ccc/resource/technical/document/user_manual/f0/14/c1/b9/95/6d/40/4d/DM00190424.pdf/files/DM00190424.pdf/jcr:content/translations/en.DM00190424.pdf.
- [5] STM32 cross-series timer overview AN4013, https://www.st.com/resource/en/application_note/an4013-stm32-crossseries-timer-overview-stmicroelectronics.pdf.
- [6] Reference manual, STM32F75xxx and STM32F74xxx advanced Arm®-based 32-bit MCUs, https://www.st.com/content/ccc/resource/technical/document/reference_manual/c5/cf/ef/52/c0/f1/4b/fa/DM00124865.pdf/files/DM00124865.pdf/jcr:content/translations/en.DM00124865.pdf.
- [7] Capacitor Charging Graph, <http://www.learningaboutelectronics.com/Articles/Capacitor-charging-graph>.
- [8] RC Charging Circuit, https://www.electronics-tutorials.ws/rc/rc_1.html.
- [9] D. Bucci, Analog Electronics for Measuring Systems, 1st ed. John Wiley & Sons, Incorporated, Newark, 2017.
- [10] J. Karki, Analysis of the Sallen-Key Architecture, Texas Instruments, 2002, 1-18 p. Available: <https://www.ti.com/lit/an/sloa024b/sloa024b.pdf>.
- [11] A.S. Nastase How to Design a Summing Amplifier Calculator, <https://masteringelectronicsdesign.com/design-a-summing-amplifier-calculator/>.
- [12] AN2834 Rev.4 How to get best accuracy in STM32 microcontrollers, STMicroelectronics, https://www.st.com/resource/en/application_note/cd00211314-how-to-get-the-best-adc-accuracy-in-stm32-microcontrollers-stmicroelectronics.pdf.
- [13] STM32F746xx Datasheet <https://www.st.com/content/ccc/resource/technical/document/datasheet/96/ed/61/9b/e0/6c/45/0b/DM00166116.pdf/files/DM00166116.pdf/jcr:content/translations/en.DM00166116.pdf>.
- [14] S. Hymel Introduction to RTOS Part 1 - What is a Real-Time Operating System (RTOS)? | Digi-Key Electronics, <https://youtu.be/F321087yYy4>.
- [15] J. Onkila, Käyttöjärjestelmän edut langattomassa anturiverkossa , TAMPEREEN TEKNILLINEN YLIOPISTO, 2011
- [16] RTOS Advantages, https://www.keil.com/rl-arm/rtx_rtosadv.asp.
- [17] D. Ibrahim, ARM-Based Microcontroller Multitasking Projects: Using the FreeRTOS Multitasking Kernel, Elsevier Science & Technology, San Diego, 2020.

- [18] General RTOS Concepts, <https://www.renesas.com/eu/en/document/apn/general-rtos-concepts?language=en>.
- [19] R. Barry, Mastering the FreeRTOS™ Real Time Kernel, © Real Time Engineers Ltd., 2016, 371 p.
- [20] S. Hymel Introduction to RTOS Part 3 - Task Scheduling | Digi-Key Electronics, <https://youtu.be/95yUbClyf3E>.
- [21] J.H. Saltzer, M.F. Kaashoek, Principles of Computer System Design: An Introduction, 2009.
- [22] s. Hymel Introduction to RTOS Part 5 - Queue | Digi-Key Electronics, <https://youtu.be/pHJ3lxOoWel>.
- [23] s. Hymel Introduction to RTOS Part 7 - Semaphore | Digi-Key Electronics, <https://youtu.be/5JcMtbA9QEE>.
- [24] s. Hymel Introduction to RTOS Part 6 - Mutex | Digi-Key Electronics, <https://youtu.be/l55auRpbiTs>.
- [25] STMicroelectronics, TouchGFX Documentation, <https://support.touchgfx.com/4.18/docs/introduction/welcome>.

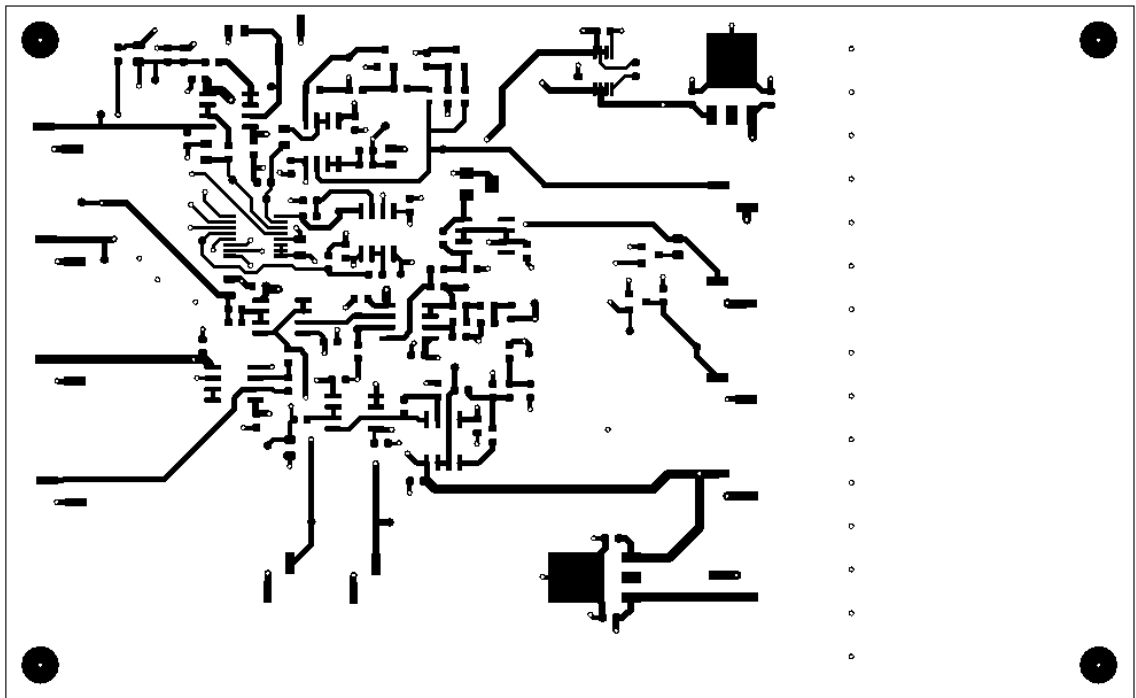
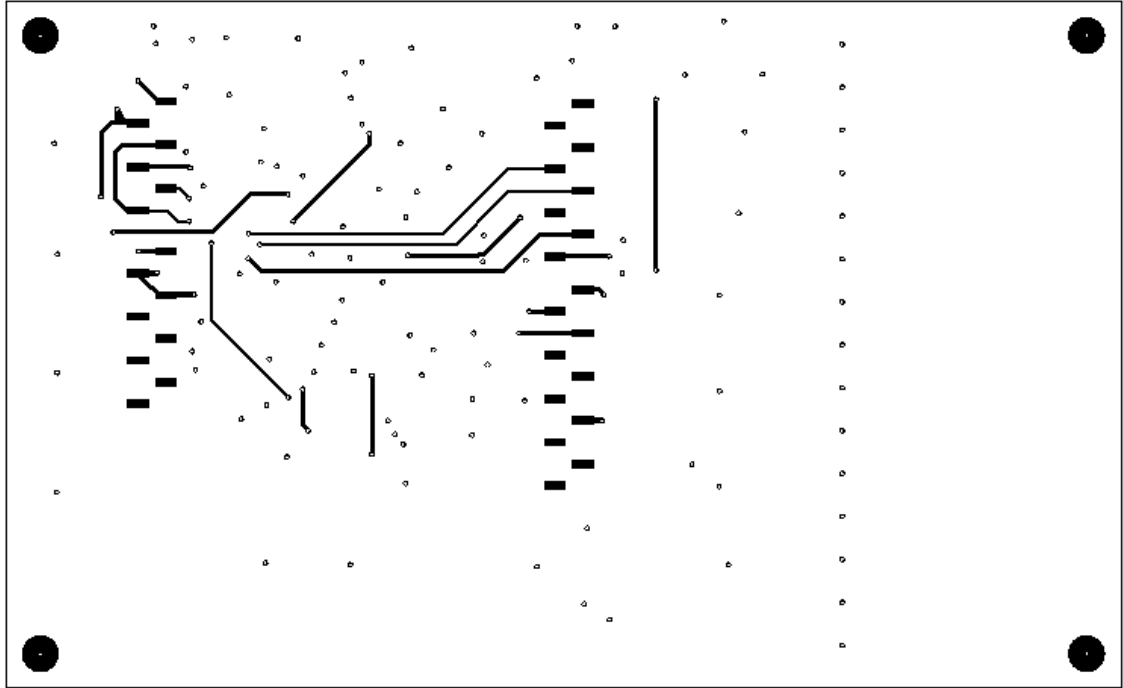
LIITE B: REGULAATTORIT.

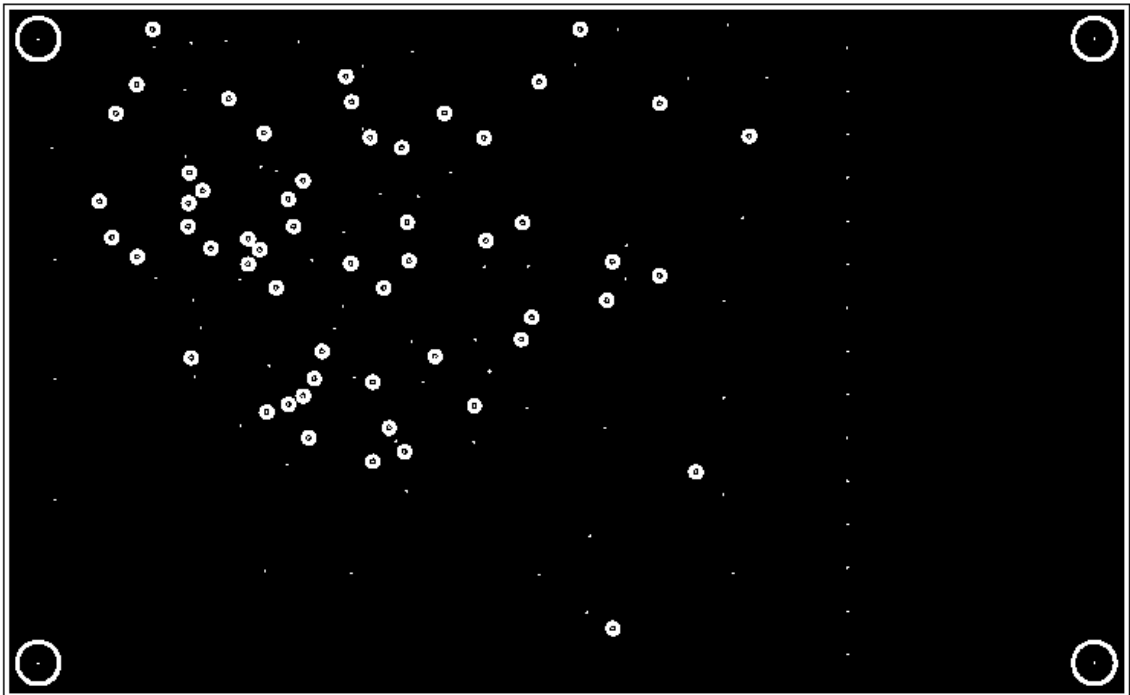
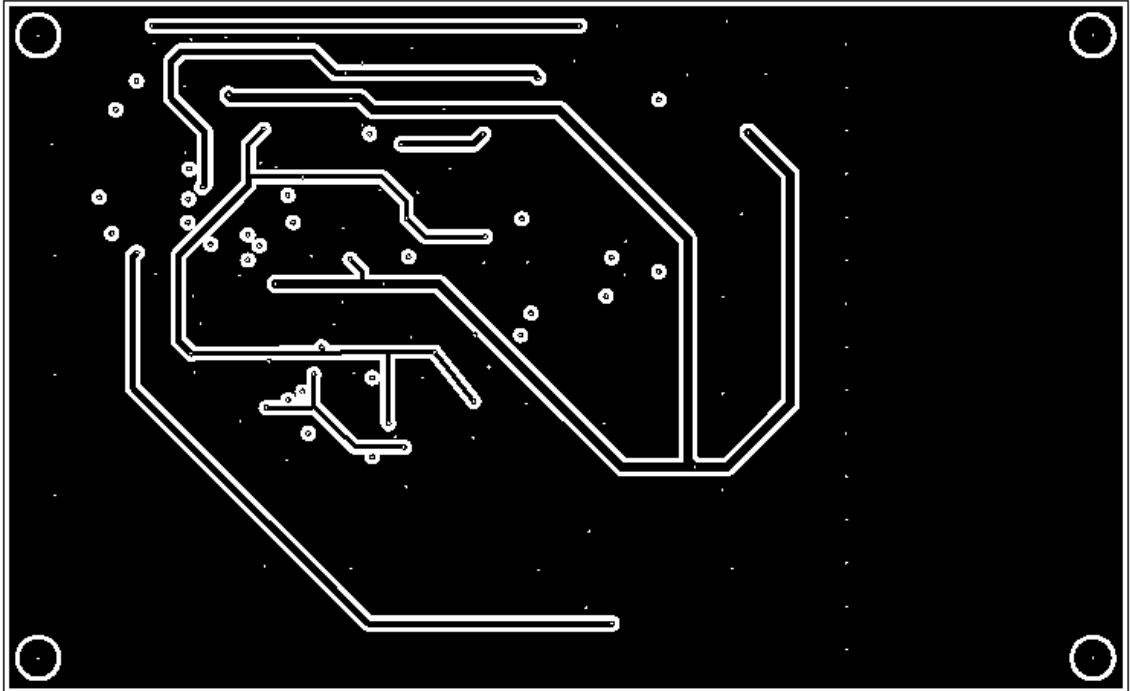


LIITE C: I/O-LIITÄNNÄT.

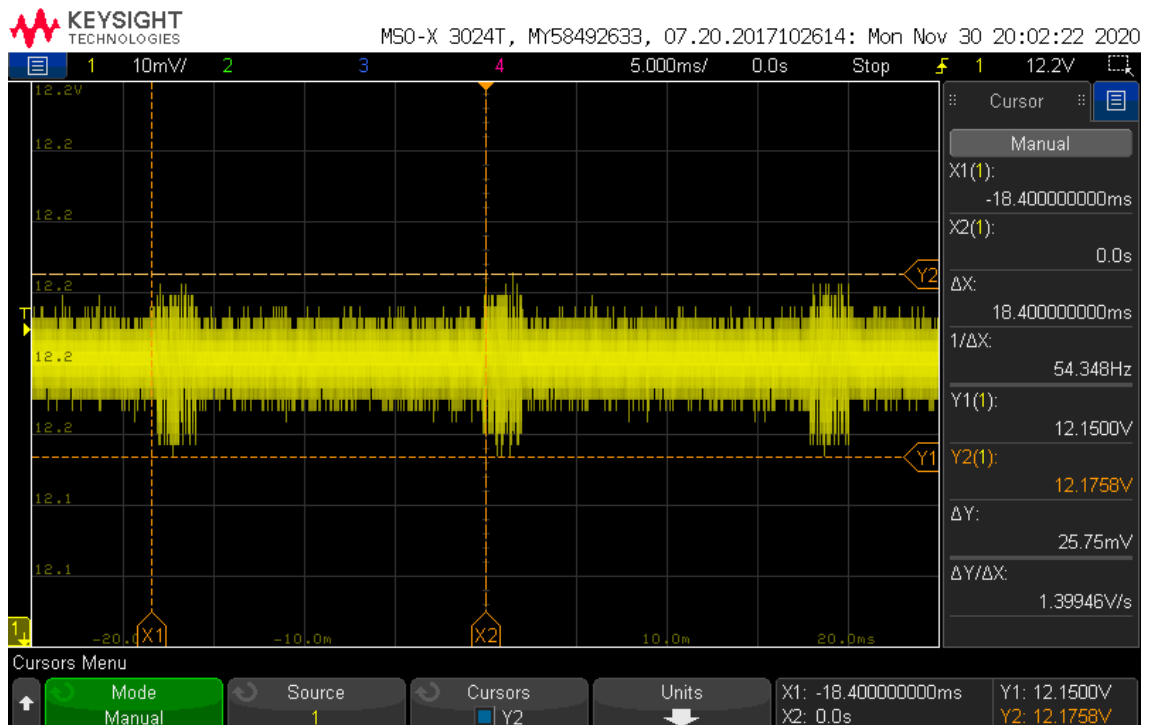
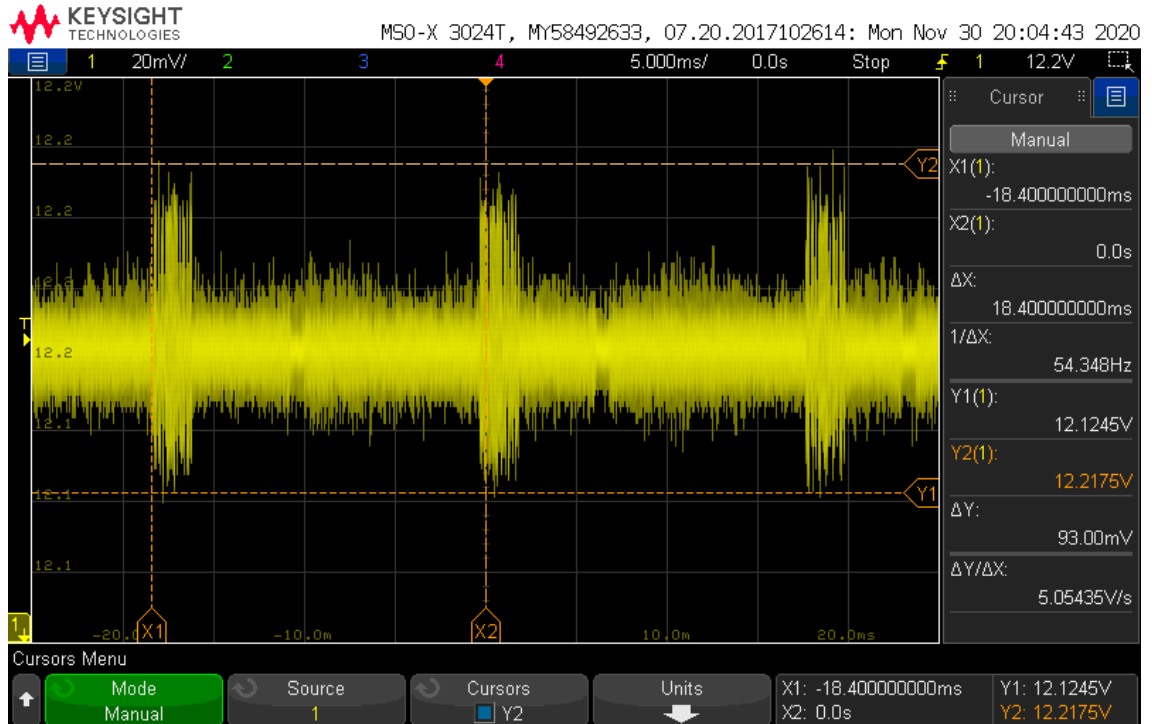


LIITE D: VALOTUSMASKIT, KERROKSET 1&4.



LIITE E: VALOTUSMASKIT, KERROKSET 2-3.

LIITE F: +12 V KÄYTTÖJÄNNITE SUODATUSKONDENSAATTOREIDEN ASENNUSTA ENNEN JA SEN JÄLKEEN.



LIITE F: +15 V KÄYTTÖJÄNNITE ILMAN STM-KORTTIA

