

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Andres Nigols 155141IASB

RF FILTRIPANGA TESTIMISE RAKENDUS

Bakalaureusetöö

Juhendaja: Olev Märten
Professor
Danel Pähnapuu

Tallinn 2018

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Andres Nigols

18.05.2020

Annotatsioon

Töö eesmärgiks on dokumenteerida tarkvara, mille autor valmistas seoses tööülesannetega. Kasutades tarkvara on võimalik juhtida autori töökoha poolt toodetavat filtripanka ning teostada teste. Kasutades teste tuvastatakse katkised tooted, mis vajavad komponentide asendust või ümberjootmist.

Töö dokumenteerimiseks kirjeldatakse esimeses peatükis, mida toode peab tegema ning kuidas vana ja uut versiooni tootest testida. Teise peatükiga kirjeldatakse kuidas tarkvara suhtleb tootega ja millised iseärasused tekivad seoses toote ehitusega. Kolmandas peatükis kirjeldatakse kergelt millist platvormi kasutatakse rakenduse loomiseks. Neljandas peatükis tutvutakse kasutatava mõõteseadmega ning kirjeldatakse kuidas sellega suheldakse. Viimases osas kirjutatakse, millises formaadis salvestatakse läbi viidud testi tulemused ja kuidas tulemused salvestatakse.

Töö tulemusena muudeti olulisel määral Windowsi keskkonna rakendust, mis juhtis vana toodet. Lisaks kohandati uue testimise tarvis eksisteerivat tarkvara arendusplaadi, mis juhtis toodet ja releed. Rakenduse kirjutamise käigust leiti tootest ehitus viga, mida oli võimalik tarkvaraliselt peita. Kasutades loodud tarkvara kasvas oluliselt testimise kiirus. Viimaseks valmis see töö.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 5 peatükki, 17 joonist, 0 tabelit.

Abstract

Test Application for a Switched Filter Bank

Purpose of this thesis is to document software, that was created in connection with author's work responsibilities. The created software can manage the switched filter bank produced by authors workplace and is able to it test thoroughly. Using these test results, it is possible to detect broken components and bad solder joints on the bord.

In order to document the software first chapter describes switched filter bank functionality and how to test both new and old version of the filter bank. The second chapter describes how software communicates with the filter bank and what sort of problems arise from this implementation of this product. In the third chapter it is described on what platform Windows application was written on and how the problem of transportation is solved, when using this platform. In the fourth chapter measuring devise used in testing is explored. In addition, how to communicate to the measuring devise, is discussed in this chapter. In the last chapter format of the test report is explored. As the report is made by modifying an existing file, this chapter describes how it is done.

As a result of this thesis existing software for controlling old switched filter bank was heavily modified to accommodate features of convenience and changes in the new filter bank. Not only was the desktop application modified, but also software on development board, used to interface the filter bank, was modified. In the process of developing this application a fault in design was discovered. Luckily, a software patch can prevent any ill effects. Using this software testing of filter banks was sped up considerably.

The thesis is in Estonian and contains 25 pages of text, 5 chapters, 17 figures, 0 tables.

Lühendite ja mõistete sõnastik

LNA	Low noise amplifier, vähese müra võimendi.
atenuaator	Seade, mis võimaldab vähendada läbiva signaali võimsust.
SPI	<i>Synchronous serial communication</i> , sünkroonse järjestiksuhtluse liides.
USB	<i>Universal serial bus</i> , seadmete arvuti külge ühendamise standard.
HID	<i>Human interface device</i> , arvutikülge ühendtav seade, mis on mõeldud inimestega suhtlemiseks.
GPIO	<i>General-purpose input/output</i> , mitme otstarbeline sisend/väljund.
SCPI	<i>Standard Commands for Programmable Instruments</i> , standardiseeritud suhtluskeel mõõteseadmetele.
ODS	<i>OpenDocument spreadsheet</i> , avatud dokumendistandardi töövihik.
XML	<i>Extensible markup language</i> , laiendatav märgistuskeel.
ZIP	Failide pakkimise formaat.

Sisukord

1 Sissejuhatus	8
2 Testiprotseduur	9
2.1 Vana testiprotseduur	9
2.2 Uus testiprotseduur	10
3 Suhtlus riistvaraga	14
3.1 Raadiosagedusliku relee juhtimine	14
3.2 GPIO laienduse juhtimine	15
3.3 Atenuaatori juhtimine	17
3.4 GPIO laienduse ja atenuaatori konflikt	18
3.5 Suhtlus mikrokontrolleriga	20
4 QT	22
4.1 Mis on qt	22
4.2 Staatiline qt	23
5 Mõõteseadmega suhtlus	25
5.1 Mida vektoranalüsaator võimaldab	25
5.2 Mida kasutatakse testi programmis	27
6 Testi dokumenteerimine	29
6.1 ODS dokument	29
6.2 Kuidas tulemusi talletatakse	30
6.3 Kuidas veel on võimalik talletada	31
7 Kokkuvõte	33
Kasutatud kirjandus	34

Jooniste loetelu

Joonis 1. Uue toote lihtsustatud joonis	9
Joonis 2. Vana toote testimine	10
Joonis 3. Kuvatõmmised uuest programmist.....	11
Joonis 4. Uue toote testimine.....	13
Joonis 5. Relee sisemine ehitus	14
Joonis 6. GPIO laiendi loogiline ülesehitus	15
Joonis 7. GPIO laiendusega suhtlemise kaks esimest baiti	16
Joonis 8. Atenuaatori loogiline ülesehitus	17
Joonis 9. Atenuaatori juhtimise baidid SPI siinil	18
Joonis 10. SPI seadmete konflikt.....	19
Joonis 11. Toote kontrolleri arvuti küljes	20
Joonis 12. Signaalide ja pistikute kasutus	23
Joonis 13. Qt Creator staatiline	24
Joonis 14. Filtri otsing markeritega	26
Joonis 15. Kõigi portidega vektoranalüsaatori taga paneel	27
Joonis 16. ODS fail avatud ZIP failina.....	30
Joonis 17. Märghendid raportis	31

1 Sissejuhatus

Autori töökoha RF filtri panga uuema versiooni väljatöötamisega tekkis vajadus uuendada filtri panga juhtimise tarkvara. Tarkvara kasutati toote testimiseks enne tellijale saatmist. Varasemast kogemusest oli oodata vigasid atenuaatoritega ja hulgaliselt jootmisevigasid. Eelnevalt käis toote testimine käsitsi kasutades juhtimise tarkvaraga. Kuna selline testimine võtab väga kaua aega ja on inimestele kurnav, siis pandi autorile lisaülesandeks muuta testimise protsess automaatseks.

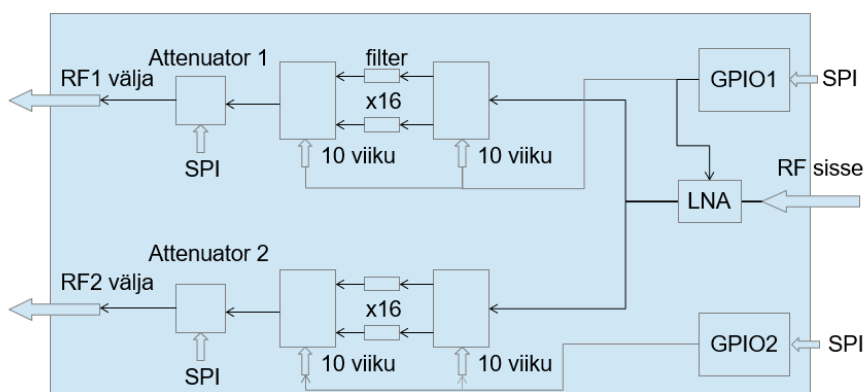
Selle töö eesmärgiks on dokumenteerida rakendus, mis suudab juhtida uut toodet ja suudab viia läbi teste kui füüsiline testimise keskkond on õigesti ülesse seatud.

Esimeses peatükis kirjeldatakse filtri panga ehitust, vana ja uut testi protseduuri ning kirjeldatakse milline on uus toote juhtimise rakendus. Teises peatükis kirjeldatakse kuidas suheldakse kontrolleri ja kuidas kontrolleri juhib releed ja filtri panka. Kolmandas peatükis kirjeldatakse millega kasutaja keskkond on loodud ning kuidas kergemini rakendusi arendus ja testi masina vahel liigutada. Neljandas peatükis kirjeldatakse milleks on võimeline testimise seade ja kuidas käib suhtlus sellega. Viimases peatükis kirjeldatakse testide raporti failiformaati, millest koosneb testi raport ise ning kuidas raport kirjutatakse kasutades eelnevaid teadmisi.

Töö tulemusena saab loodud tarkvara komplekt filtripanga testimiseks ja juhtimiseks ning dokumenteeritud tarkvara töö põhimõtted.

2 Testiprotseduur

Tooteks on ühe sisendi ja kahe väljundiga raadiosagedus filtripank. Toote väljunditeks on sisend signaal, millel kummalgil on valitud oma filter. Sisendil saab sisse lülitada 20dB vähese müraga võimenduse. Uuel tootel saab lisaks filtri valimisele väljundi võimsust vähendada kuni 31,75dB võrra 0,25dB sammuga. Toote juhtimine käib üle ühe SPI siini, kus seadme valiku viigud on kokku ühendatud ja asendatud andmebaitidega. Joonis 1 on kirjeldatud uus toode lihtsustatud joonisega



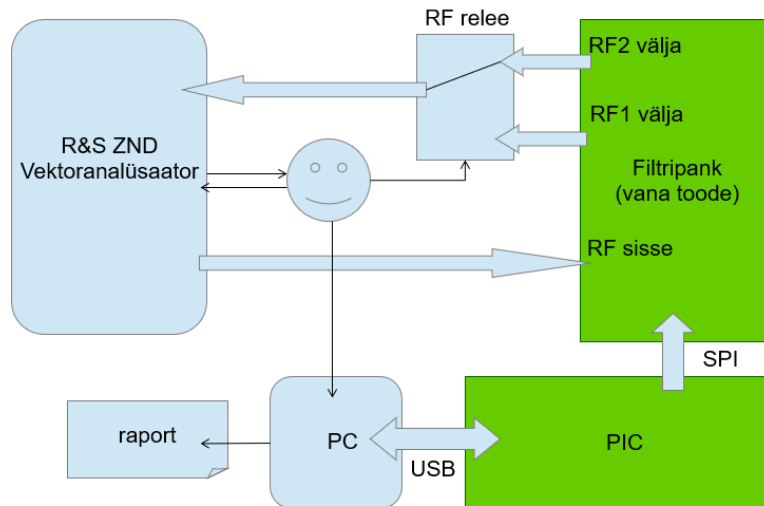
Joonis 1. Uue toote lihtsustatud joonis

Toote testimiseks on kasutada vektoranalüsaator, kahe sisendi vahel vahetav relee, mikrokontrolleriga arendusplaat ja arvuti Windows seitsmega. Vektoranalüsaator oli paari aasta vanune Rohde & Schwartz vektoranalüsaator ZND perekonnast, millel jooksis Windows seitse analüsaatori tarkvaraga. Arendusplaat on PIC24FJ64GB004 kontrolleri enda loodud arendusplaat. Sama funktsionaalsust saab realiseerida suvalisel arendusplaadil, millel on USB ja SPI suhtlus võimekus ning vähemalt üks GPIO viik.

2.1 Vana testiprotseduur

Toote valmides saadeti toode testija juurde. Enne toote testimist kalibreeritakse vektoranalüsaator. Seejärel ühendas toote testija sisendi ja kaks väljundit läbi relee vektoranalüsaatori külge. Peale toote ja relee toite sisse lükkamist ühendas testija toote SPI juhtotsad arendusplaadi külge ja arendusplaadi USB kaabliga arvuti külge. Siis avas

testija arvutis toote juhtimise tarkvara ja veendus, et juhtimine toimib. Kasutades vana tarkvara testis testija sisendi võimenduse ja filtrite töötamise. Saadud tulemused kirjutas testija arvutis ODS faili, mis prinditi välja ja pandi tootega kaasa. Vea leides saadeti toode parandusse. Parandusest naastes testiti toodet uuesti. Tüüpilisteks vigadeks tootes olid jootmisvead ja katkised komponendid. Joonis 2 illustreerib vana testimise keskkonda.



Joonis 2. Vana toote testimine

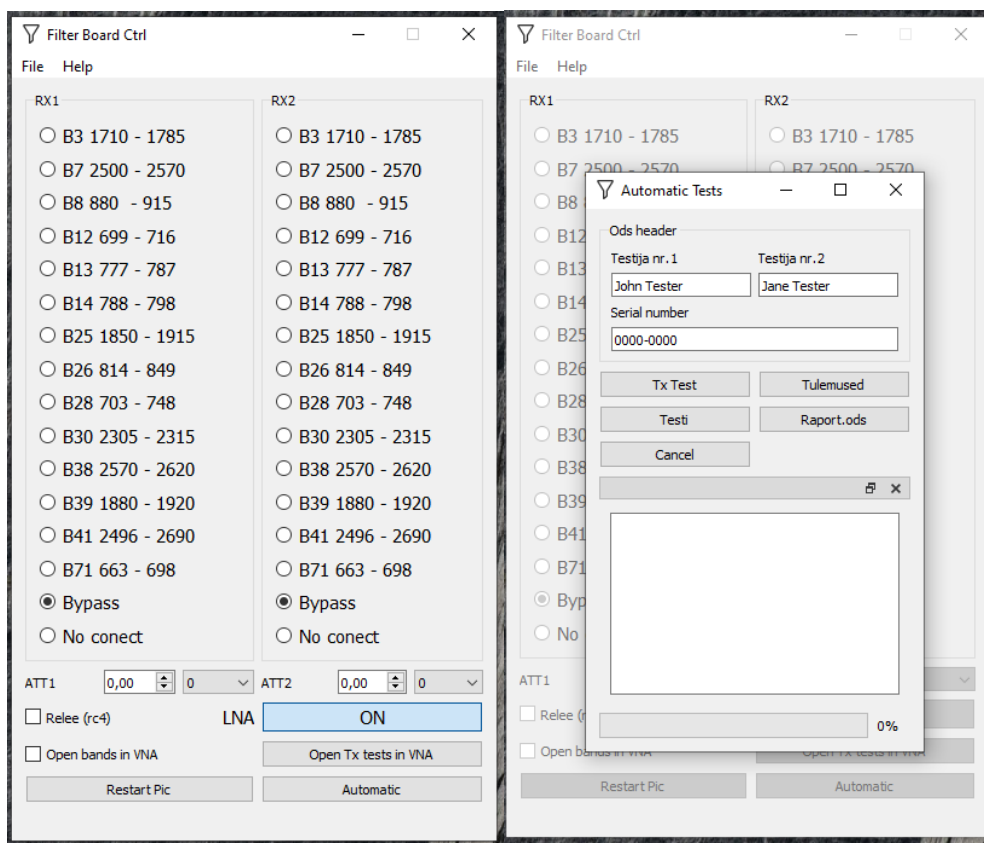
Võimenduse testimiseks valiti releega esimene väljund ning valiti filtripangast filter, mis oli otseühendus väljundiga. Seejärel avas testija vektoranalüsaatoris salvestatud seaded nägemaks tervet tööpiirkonda. Vaadates vektoranalüsaatorit nägi testija, kas selles ahelas esines jootmisvigu ja märkis ülesse toote sumbumuse. Testija lükkas sisse sisendi võimenduse ja vaatas, et toote tööpiirkonnas toimuks 20dB-ine võimendus. Sarnaselt esimesele väljundile kontrolliti jootmisvigu teisel väljundil.

Filtrite testimiseks avati vektoranalüsaatoris salvestatud seaded testitava filtri testimiseks. Seadetes oli salvestatud piirjoon, mis näitas kui tugev peab olema töötava võimendusega filtreeritud väljund. Kasutades piirjoont ja vana tarkvara kontrollis testija iga filtri töökorralisust mõlemal väljundil. Erandiks olid otseühendus ja lahti ühendus mille töökorralisust kontrolliti terve tööpiirkonnaga. Nõnda tulid välja vigased filtrid ja jootvead erinevate filtrite ahelates.

2.2 Uus testiprotseduur

Vanas tarkvaras oli võimalik valida toote filtrit ja toote sisendi võimendust sisse lükata. Uues tarkvaras on lisaks eelnevale võimalik lisada sumbumust väljundile, lülitada ümber

releed läbi arendusplaadi, taaskäivitada arendusplaat, avada vektoranalüsaatoris valitud filtrile vastavad salvestatud seaded ja avada aken automaatseks testimiseks. Automaatse testi aknas on võimalik alustada testi, peatada testi läbiviimine, kuvada testi tulemused eraldi aknas ja lasta arvutil testi tulemuste põhjal kirjutada raport, muuta raportisse kirjutatavaid testija ja toote andmeid. Kui testi läbiviimist ei peatata kirjutatakse testi lõppedes automaatselt raport. Testi läbi viies kirjutab rakendus tekstikasti järjest, mida rakendus tegi. Automaatse testi akna avamisel ei ole võimalik manuaalse juhtimise akent kasutada. Joonis 3 on kuvatõmmised uuest programmist.



Joonis 3. Kuvatõmmised uuest programmist

Sarnaselt vanale testiprotseduurile ühendati toode vektoranalüsaatori, arendusplaadi ja relee külge. Erinevalt varasemale tuleb enne arendusplaadi arvuti külge ühendamist ühendada relee toide ja relee juhtimise ots arendusplaadi külge. Peale USB kaabli ühendamist tuleb ühendada netikaabliga arvuti ja vektoranalüsaator. Sellises olekus saab uut tarkvara kasutades juhtida toodet manuaalselt või sooritada automaatne test. Uut toodet testides tuleb sooritada lisaks filtrite ja võimenduse testile ka sumbumuse test.

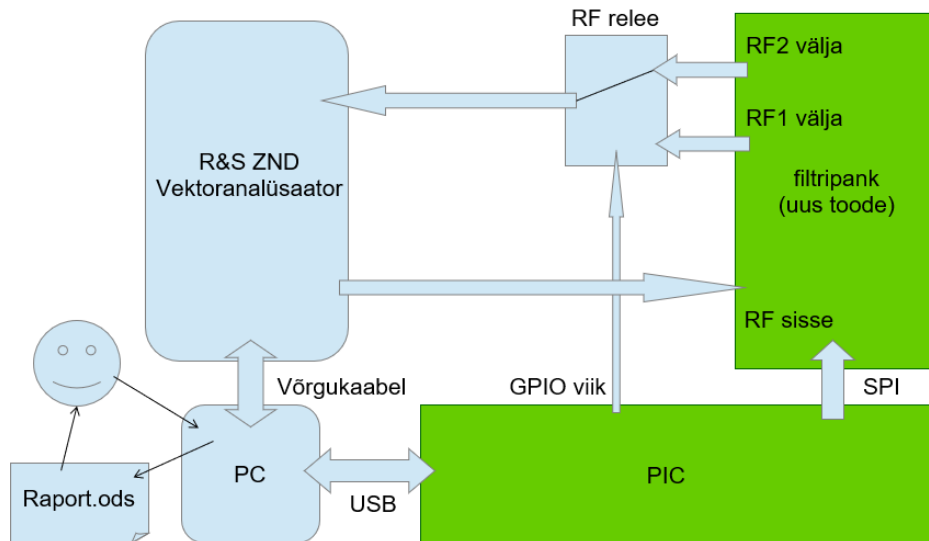
Automaatne testimine algab võimenduse testimisega. Selleks valib rakendus tootes filtriteks otseühenduse väljundiga, nullib atenuaatori, avab vektoranalüsaatoris

salvestatud seaded toote tööpiirkonnaga ning salvestab viie strateegilise sageduse läbivuse. Peale signaali tugevuse salvestamist lülitatakse sisse sisend signaali võimendus ning salvestatakse nende sama viie sageduse läbivus uuesti. Relee asendit ei kontrollita kuna tootel on üks vähese müraga võimendi ja jootevigade puhul kukuvad ka teised testid läbi.

Võimenduse testimise järel alustab rakendus atenuaatori testimist. Atenuaatori ehitusest sõltuvalt on tavaline viga, et element töötab ainult alla või üle 16dB-ise sumbumuse valikul. Vastavalt sellele testitakse atenuaatorit väärtustel 0, 10, 16 ja 30. Atenuaatori testimiseks lülitatakse relee esimese väljundi mõõtmise asendisse, veendutakse võimendi sees olekus, antakse väljundile üks neljast testitavast sumbumuse väärtusest ning salvestatakse toote läbivus ühel sagedusel. Nõnda testitakse mõlema väljundi atenuaator nelja erineva väärtusega.

Filtrite testimiseks valitakse releega esimene väljund, valitakse üks filter tootes, avatakse sama filtri salvestatud seaded vektoranalüsaatoris, küsitakse vektoranalüsaatorist kui kõrgel on piirjoon ja kas toote väljund ületab piirjoone väärtust. Nõnda testitakse iga filter mõlemal väljundil.

Peale testide lõpetamist kirjutab rakendus ODS faili kõik salvestatud andmed. Faili kirjutatakse lisaks võimenduse, atenuaatori ja filtrite testi andmetele ka testijate ja toote andmed automaatse testi aknast. Kasutades raportit näeb testija millised filtrid on vigased, kas atenuaatorid on vigased, kas võimendus töötab ja kus võib esineda jooteviga. Joonis 4 illustreerib uut testikeskkonda.



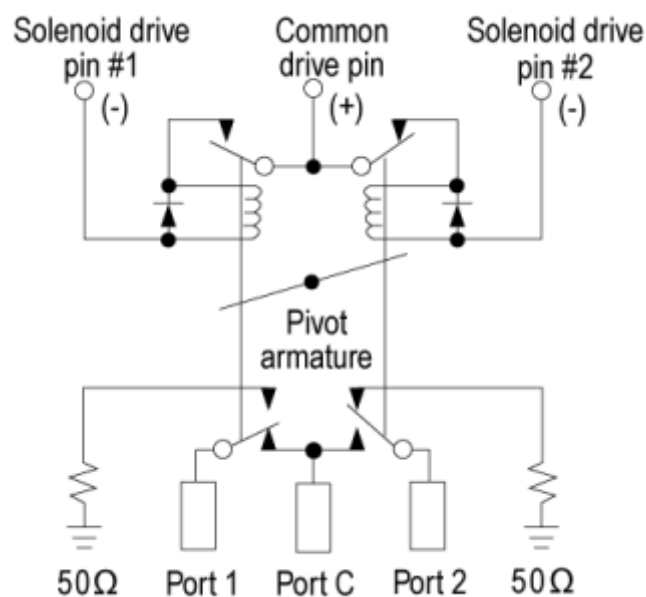
Joonis 4. Uue toote testimine

3 Suhtlus riistvaraga

Arendusplaadi kontrolleris jookseb programm, mis võtab vastu arvutilt käske ning vastavalt käskudele muudab toote käitumist või vektoranalüsaatorisse minevat väljundit. Selles peatükis kirjeldatakse täpsemalt kuidas käib suhtlus arendusplaadi ja tootega.

3.1 Raadiosagedusliku rele juhtimine

Releeks toote testimisel on valitud 8762A. Joonis 5 on rele dokumentatsiooni joonis, mis näitab rele sisemist ehitust. Vastavalt sisend pinnidele on rele võimeline ühendama ühe kahest pordist ühise pordiga. Teine port ühendatakse viiekümne oomise terminaatori otsa. Sisend pinnide ühine pinn ühendatakse toitepingesse ja vastavalt kumba porti soovitatakse ühendada ühendatakse vastav pinn null nivooaga. Kui mõlemad pinnid ühendada nulli, siis hakkab rele kiirelt portide vahel vahetama. Korraliku ühenduse loomiseks portide vahel tuleb pinni hoida nullis vähemalt kolmkümmend millisekundit. Kui pinn on null nivool kauem siis ühendatakse peale kolmekümmet millisekundit ühine pinn lahti sisend pinnist, et vältida liigset voolu tarbimist. [1]

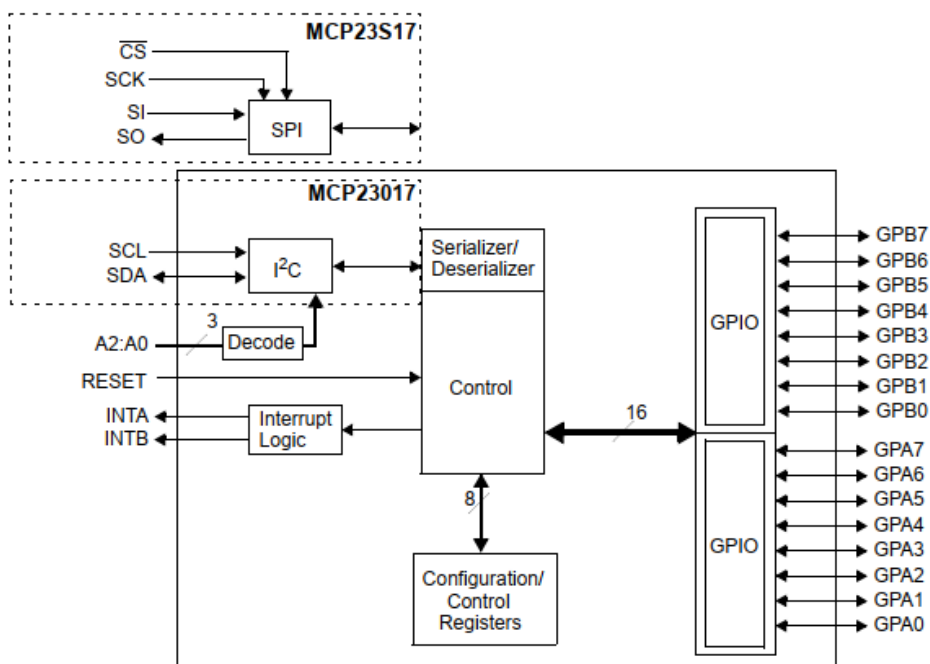


Joonis 5. Rele sisemine ehitus

Relee juhtimiseks kasutatakse isetehtud relee juhtimisseadet, mis saab sisendiks tõusva või langeva frondi ja vastavalt frondile ühendab null nivooga esimese või teise pinni kolmekümneks millisekundiks. Eelnevalt kasutati frontide tekitamiseks lülitit, nüüd kasutatakse selleks kontrolleri GPIO pinni. Kontrolleri ei saa relee juhtimisseadelt tagasisidet relee oleku kohta ja seega relee toite kadumisel jätkab testiprogramm oma tegevusi teadmata probleemist.

3.2 GPIO laienduse juhtimine

GPIO laienduse kiibiks on tootes valitud MCP23S17, mis võimaldab üle SPI siini muuta või lugeda kuueteistkümne viigu väärtust. Lisaks võimaldab kiip signaali saatmiseks loetava viigu muutudes ühest olekust teise või minnes kindlasse olekusse. Igat viiku on võimalik seadistada eraldi sisendiks või väljundiks ja võimaldada viigu muutumise signaal. Joonis 6 on laiendi lihtsustatud joonis dokumentatsioonist. [2]

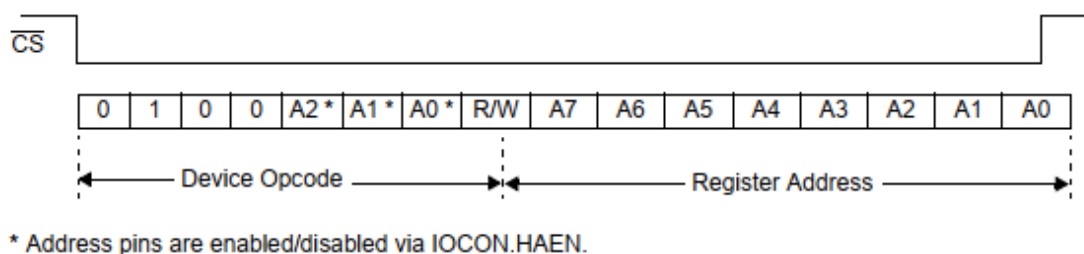


Joonis 6. GPIO laiendi loogiline ülesehitus

Viikude seaded salvestatakse kiibi mälu, mille maksimaalne adresseeritav bait on kuueteistkümnendsüsteemis 0x1A. kiibi mälu on jagatud esimese baidi jagu viikude juhtimise seadeteks ja teise baidi jagu viikude juhtimise seadeteks. Lisaks on mälus ühes baidis kiibiüleused seaded, mille poole on võimalik pöörduda kahel aadressil, millest üks on alumiste viikude seadete grupis ja teine ülemiste viikude grupis. Viikude gruppe on

võimalik adresseerida ühe viikude grupi seaded järjest või viikude grupid vaheldumisi samad seaded järjest. Kuidas adresseerida on kirjas kiibiüleses baidis. [2]

Kiibi juhtimine käib üle SPI siini, kus esimene saadetav bait on kontrollsõna. Kontrollsõnas esimesed neli baiti on fikseeritud väärtustega, järgmised kolm biti on adresseerimise bitid ja viimane bitt määrab ära, kas teises baidi väärtuse aadressilt loetakse või kirjutatakse andmed. Kiibiülese seadete baidiga määratakse ära, kas aadressi bitte kasutatakse või mitte. Aadressi bitte kasutades ei vasta kiibi SPI suhtlusele, kui kiibi aadressiviigud ja bitid ei ole kooskõlas. Veel on seadetes kirjas, kas peale ühe baidi kirjutamist või lugemist on võimalik järgmiste baitidega kirjutada või lugeda. Joonis 7 on kiibi dokumentatsioonist joonis demonstreerimaks kahe esimese saadetava baidi väärtusi. [2]



Joonis 7. GPIO laiendusega suhtlemise kaks esimest baiti

Toote tellijast sõltuvalt on kasutada vaid üks kiibi valiku viik. Seetõttu on tootel mõlemale GPIO laienduse kiibile antud fikseeritud aadress ja nende poole pöördumine käib kontrollsõnaga. Kuigi riistvara seda ei nõua, kasutatakse kontrolleri tarkvaras ühe viikude grupi seaded järjest adresseerimist. Kontrolleri tarkvaras seadistatakse kiipi lugema või kirjutama rohkem kui ühte baiti, kuid tarkvaras seda funktsionaalsust ei kasutata. Need kolm seadet saadetakse peale mikrokontrolleri seadete kirjutamist, enne tarkvara põhitsükli. GPIO laienduse kiip alustab toite saamisel olekust, kus aadressi bitte ei kasutata, viikude grupi seaded on adresseeritud vaheldumisi ja mitme baidi kirjutamine on lubatud [2]. Seetõttu saab ühe SPI suhtlusega ära seadistada mõlemad laiendid.

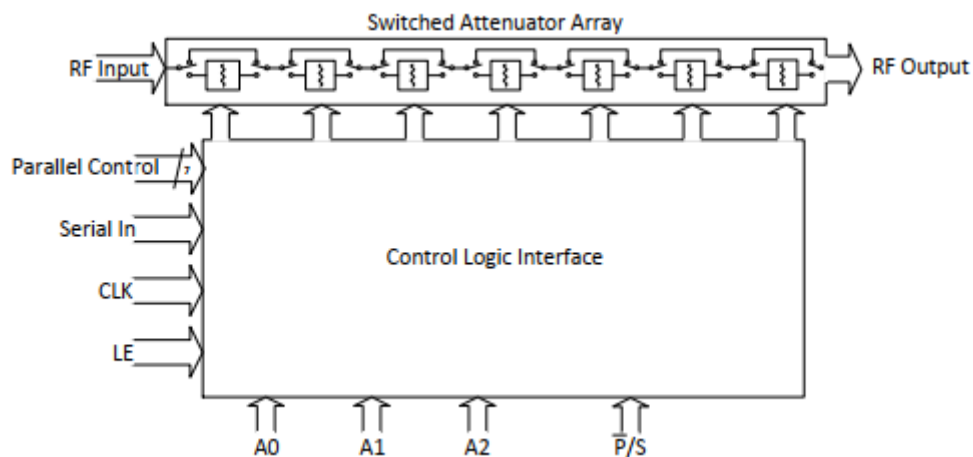
Filtrite vahel valimine on teostatud kahes astmes nelja kanali vahel vahetava kiibiga. Ühe filtri valimiseks kasutatakse kuute signaali, millest kolm on kolme teise filtri puhul sama väärtusega. Kokku kasutatakse ühe toote väljundi juhtimiseks GPIO laiendi viiteteist viiku. Ühel pool filtrit on filtri valimiseks kokku kasutuses kümme signaali, millest viite kasutatakse ka teisel pool filtrit. Esimese väljundi GPIO laiendi viimast kasutamata viiku kasutatakse võimenduse juhtimiseks. Vahetult peale eelmise peatüki seadistuse saatmist

seatakse mõlema kiibi kõik viigud väljunditeks ja antakse neile väärtus, millega valitakse filtri otsühendus väljundisse.

Filtri muutmiseks saadab testimise tarkvara mikrokontrolleri tarkvarale üle USB käsu muuta mõlema GPIO laienduste kõiki väärtusi. Selleks pööratakse mõlema laienduse poole kaks korda. Esimese pöördumisega kirjutatakse ühe poole viikude väärtused vastavaks USB andmetele ja teise korraga kirjutatakse teise poole andmed. Vähesel määral võimendi käivitamise puhul saadetakse samamoodi kõikidele viikudele uus väärtus, kuid testimise tarkvaras muudetakse vaid võimendi bitti.

3.3 Atenuaatori juhtimine

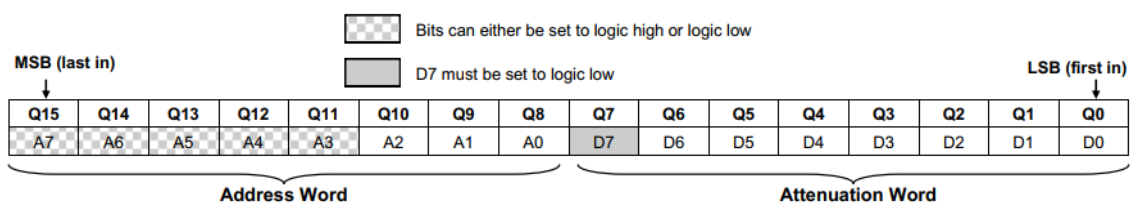
Atenuaatoriks on tootes valitud PE43705, mis võimaldab vähendada kiibi läbiva raadiosignaali tugevust. Kui palju nõrgemaks muutub signaal on võimalik juhtida SPI siinilt tulevate andmetega või seitsme viiguga. Seitsme viigu kasutamiseks tuleb viia nullnivoole kiipi juhtimist reguleeriv viik. Atenuaator võimaldab vähendada signaali tugevust 0dB-ist kuni 31,75dB-ini 0,25dB-ise sammuga. Joonis 8 on dokumentatsiooni joonis näitamaks atenuaatori lihtsustatud ülesehitust. [3]



Joonis 8. Atenuaatori loogiline ülesehitus

Tootel on ette nähtud kaks atenuaatorit, millest mõlemat peab olema võimalik eraldi juhtida. Kuna seitsme viiguga juhtimine kasutaks liiga palju kontrolleri viike siis on tootes kasutatud SPI siiniga juhtimist. SPI siini lugemine on atenuaatoris lahendatud kahe baidise nihkeregistriga, mille nihutamist lubab kiibi valiku signaal. Valiku signaali kadudes loeb atenuaator nihkeregistrisse kirjutatud andmed ja korrektsete andmete puhul muudab signaali tugevuse vähendamise suurust. SPI siinilt loetakse kõigepealt sisse

signaali vähendamise suurus, väikseim bitt esimesena. Baidi väärtus jagatud neljaga saab olema signaali vähendamise suurus detsibellides. Seitsmes bitt baidis peab olema null, kuna baidist saadud väärtus jääb atenuaatori tööalast välja. Dokumentatsioonis ei ole kirjeldatud atenuaatori käitumist, kui seitsmes bitt on üks. Seejärel loetakse adresseerimise bait, millest on kasutusel vaid kolm esimest biti. Kolme biti väärtuste kattumise atenuaatori aadressi viikude väärtusega loetakse nihkeregistris olevad andmed korrektseks. Joonis 9 on dokumentatsiooni joonis, mis kirjeldab, kuidas atenuaatoriga suhelda üle SPI siini. [3]



Joonis 9. Atenuaatori juhtimise baidid SPI siinil

Testi tarkvaras atenuaatori väärtust muutes saadetakse üle USB vastava väljundi atenuaatori väärtuse muutmise käsuga uus väärtus. Mõlemad käsud saadavad selle peale oma aadressiga SPI siinile uue atenuaatori väärtuse. Enne põhiprogrammi tsükklisse jõudmist antakse atenuaatoritele väärtus null, kuna atenuaatori väärtus toite saamisel on maksimaalne.

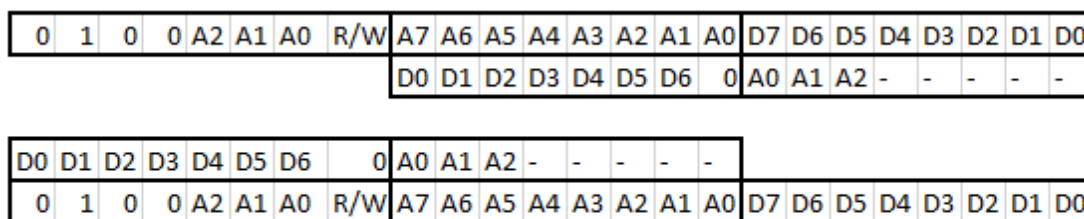
3.4 GPIO laienduse ja atenuaatori konflikt

Tootes on kõik atenuaatorid ja GPIO laiendid sama SPI siini peal. Igale kiibile on aadressi viikudega antud unikaalne aadress, mida kasutatakse kiibile mõeldud andmete ära tundmiseks. Atenuaatori ja GPIO laiendi keerulisuse vahest ja tootjate erinevuse tõttu on kiipides realiseeritud SPI siini lugemine erinevalt.

GPIO laiendi puhul loetakse sisse kontrollbait saaja füüsilise aadressiga, manipuleeritava andmebaidi aadress ja siis kirjutatakse või loetakse baite. Atenuaatori puhul loetakse sisse tagurpidine uus atenuaatori väärtus, saaja füüsiline aadress ning kiibi valiku signaali lõpedes muudetakse aadressi kokku langedes väärtust. Kui atenuaatorisse jõuab kolm baiti, siis nihutatakse esimene bait välja ja aadressi kontrollitakse viimase saadud baidi peal. Nõnda võib juhtuda, et filtri valikul GPIO laiendisse kirjutatav väärtus aktiveerib atenuaatori ning see saab väärtuseks tagurpidise aadressibaidi. Filtri valikul kirjutatava

baidi aadresside viimased bitid on väärtusega null ja seega ei teki dokumenteerimata olukorda. Joonis 10 ülemine osa näitab kuidas langevad bitid kokku mainitud olukorras.

Ka atenuaatori juhtimisega on võimalik kogemata pöörduda GPIO laiendi poole, kuid selline pöördumine ei ole võimaline muutma laiendi käitumist. Kui anda atenuaatori väärtuseks laiendi füüsilise aadressi spetsiifiline arv, siis pöördutakse kirjutamise režiimis atenuaatori aadressiga laiendi mälu aadressi poole. Kui atenuaator mille poole pöördutakse ei ole aadressil null, siis toimub laiendis pöördumine aadressi poole, mida mälus pole. Null aadressiga pöördumise puhul otsustavad bitid, mida atenuaator ei vaata, millise mälu aadressi poole pöördutakse. Kui toimub GPIO laiendi mälu aadressi poole pöördumine, siis ei toimu kirjutamist tolele aadressile tänu selle, et atenuaatori käsk on kaks baiti pikk ja kirjutatavad andmed on laiendil kolmandas baidis. Joonis 10 alumine osa näitab kuidas töötleb GPIO laiendi atenuaatorile mõeldud andmeid.



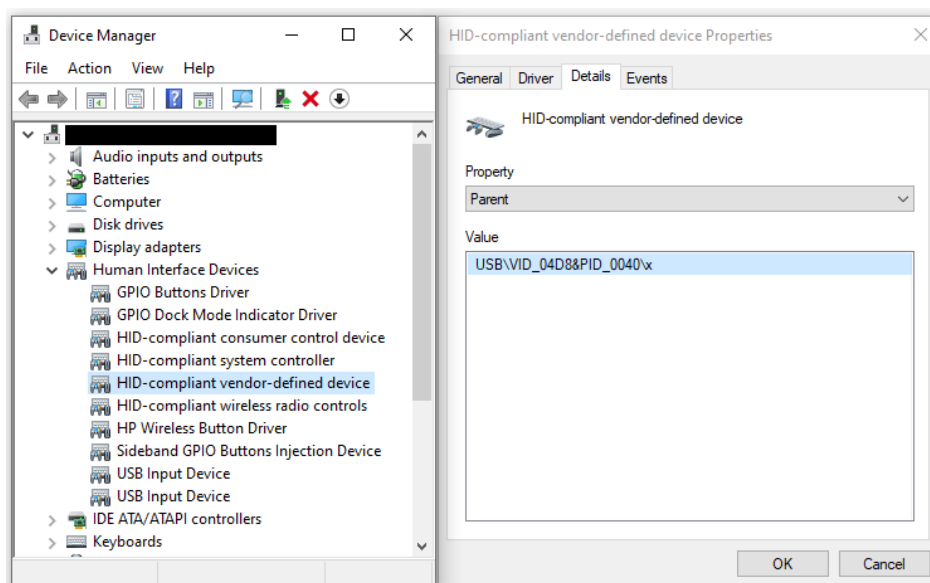
- on suvaline biti väärtus

Joonis 10. SPI seadmete konflikt

Tootes ei ole atenuaatoritel ega GPIO laienditel antud aadressi viikudele väärtus null, seega ei ole vaja karta atenuaatorile mõeldud pöördumisi. Kuid leidub filtreid, mille valimisel muutub üks kahest atenuaatori väärtusest. Lisaks toimub atenuaatori poole pöördumine filtrite järjestikuse valimise kombinatsioonidega, kui filtrite valimiseks muudetakse ainult filtri juhtimise jalgu. Viimane probleem on võimalik lahendada algväärtustades kõik kasutamata viigud. Teiste problemaatiliste filtrite kasutamisest tekkivaid atenuaatori väärtuste muutmisi ei ole võimalik ennetada toodet ümber tegemata. Seega tuleb peale filtrite kasutamist muuta tagasi atenuaatorite väärtused. Testimise programmis rakendatakse mõlemat meetodit.

3.5 Suhtlus mikrokontrolleriga

Vanas testitarkvaras oli arendusplaat arvuti külge ühendatud kui USB HID seade ja sellega suhtlemiseks kasutati kolmandate isikute koodi nimega hidapi¹. Iga kord kui oli vaja valida uus filter avati hidapiga ühendus kontrolleri tootjakoodiga ja suvalise tootja poolt kasutamata tootekoodiga seadmega. Ühenduse loomise läbi kukkumisel viskas testiprogramm ekraanile veateate. Õnnestumisel saadeti arendusplaadile GPIO laiendite juhtimise käsk nelja uue laiendite viikude olekuga. Eduka saatmise puhul suleti hidapiga avatud ühendus ja vabastati kasutatud mälu. Joonis 11 näitab edukalt arvuti külge ühendatud arendusplaati.



Joonis 11. Toote controller arvuti küljes

Uues testitarkvaras on lahendatud atenuaatori väärtuste saatmine sarnaselt GPIO laiendiga suhtlemisele. Ainukeseks erinevuseks on saadetavad andmed. Atenuaatori väärtuse muutmiseks saadetakse andmepaket käsubaadi ja uue väärtusega. Mõlemad antakse kaasa funktsiooni muutujatega, seega saab sama funktsiooniga juhtida mõlemat atenuaatorit, kuid nende juhtimise käsud on kontrolleri erinevad. Lisaks atenuaatoritele saab sama funktsiooniga juhtida relee olekut ja taaskäivitada kontrolleri. Relee juhtimiseks tuleb saata õige käsubaad ja relee uus olek. Taaskäivitamiseks tuleb saata õige käsubaad praht andmetega.

¹ <http://github.com/signal11/hidapi>

Mikrokontrolleril jooksev programm on oma ehituselt sarnane kontrolleri tootja poolt pakutavale USB isevalmistatud HID seadme kasutamise näitekoodile¹. Kontrolleri programm seadistab USB mooduli käituma kui HID seade. USB pöördumisel toimub katkestus ja kontrolleri tootja kood loob vajadusel ühenduse arvutiga ning salvestab arvutist tulnud andmed. Enne põhiprogrammi tsükli algust konfigureeritakse lisaks USB moodulile SPI siin, GPIO viigud, üks taimer ja toote komponendid. Põhiprogrammi tsükli loendatakse nelja tulukesega sekundeid. Kui arvutiga on loodud ühendus ning andmepakett on vastu võetud, siis töödeldakse arvutist saadud andmepakete. Vajadusel saadetakse arvutisse tagasi informatsiooni ja lastakse tootja koodil valmistada järgmise andmepaketi vastuvõtmiseks.

Arvutist saadud andmete esimene bait tähistab käsku, mida mikrokontroller peab täitma. Varasemalt olid programmis käsud tagastamaks seadet kirjeldavat stringi, kõikidele GPIO laiendite viikudele uute väärtuste andmiseks, kaks erinevat kontrolleri magama panemise käsku, kontrolleri taaskäivitamise käsk ja muud käsud, millel pole pistmist toote testimisega. Uue toote juhtimiseks lisati käskude hulka kumagi atenuaatori juhtimise käsk ja releed juhtiv käsk.

Saates kontrollerisse GPIO laiendite juhtimise käsk tuleb järgmise nelja baidiga kaasa anda esimese väljundi juhtimise laiendi esimese kaheksa viigu ja teise kaheksa viigu väärtused eraldi baitides ning seejärel teise laiendi viikude väärtused. Mõlema atenuaatori käsuga tuleb kaasa saata uus atenuaatori väärtus, ning käsk kannab hoolt atenuaatoriga õigesti suhtlemise eest. Relee juhtimise käsule järgnev bait määrab ära kumb väljund on releega valitud. Kui baidi väärtuse ei ole võrdne nulliga valitakse esimene väljund.

¹ <https://www.microchip.com/mplab/microchip-libraries-for-applications>

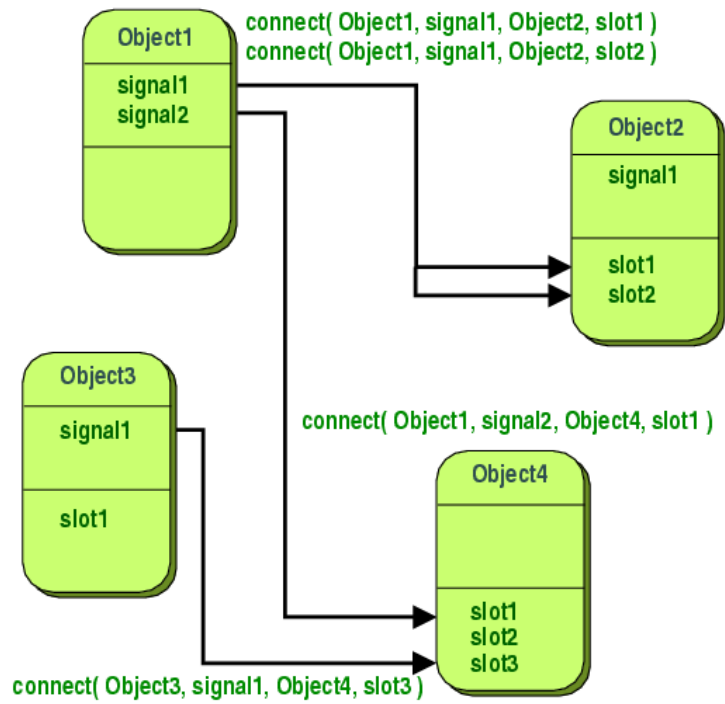
4 Qt

Testrakendus on loodud kasutades Qt raamistiku. Selles peatükis kirjutatakse lühidalt, mida raamistik võimaldab ja kuidas on rakendus kompileeritud kergemaks transpordiks.

4.1 Mis on Qt

Qt on C++ keele raamistik, mis võimaldab luua samasuguse väljanägemisega kasutajaaknaid erinevatel platvormidel. Testrakendus koosneb kolmest aknast, millest esimesega saab juhtida toodet, teisega saab alustada automaatset testi ja viimane näitab testi tulemusi. Testi tulemuste aken koostatakse dünaamiliselt kasutades iga testi jaoks koostatud graafilist komponenti. Igasse komponenti kirjutatakse loomisel hetke testi tulemused, kui tulemusi pole mõõdetud siis on testi tulemused null ja läbikukkunud. Juhtimise aken koosneb põhiliselt nuppudest ja automaatse testi aken koosneb nuppudest ja tekstikastidest.

Dokumentatsioonile kohaselt lisatakse raamistiku poolt signaalid ja pistikud, mida kasutatakse objektide vahel suhtlemiseks. Signaalid on mõeldud näitama teistele objektidele, et objektiga on toimunud midagi, millest keegi teine võib olla huvitatud. Näiteks on nupu vajutamine ja tekstikasti kirjutamine signaalid. Signaalid ei tohi tagastada midagi. Pistikud on funktsioonid, mis kutsutakse välja nendega seotud signaalide saatmise peale. Üks signaal võib kutsuda välja mitu pistikut ning need kutsutakse välja ühendamise järjekorras. Joonis 12 näitab kuidas võib pistikuid ühendada. Pistikud kutsutakse välja vastavalt pistiku tüübile, kas koheselt peale emiteerimise käsku või peale emiteeriva funktsiooni lõppu. Peale esimest pistikute töötlemise varianti jätkub kood peale pistiku töötlust emiteerimise käsus juurest. Testimise rakenduses kasutatakse erinevaid nupu vajutuse signaale kõikide tegevuste algatamiseks. [4]



Joonis 12. Signaalide ja pistikute kasutus

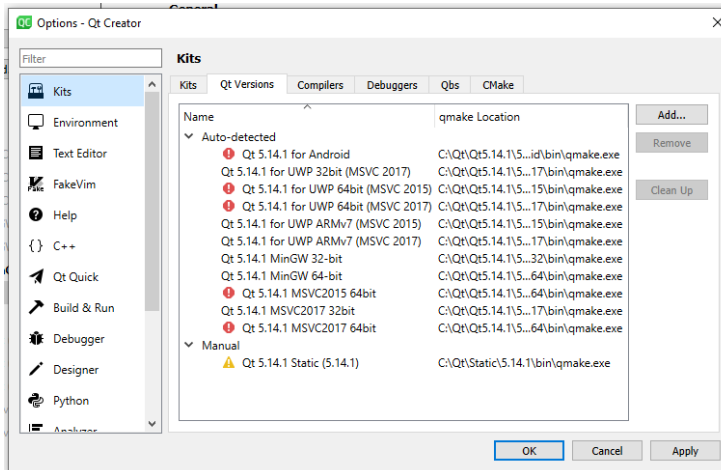
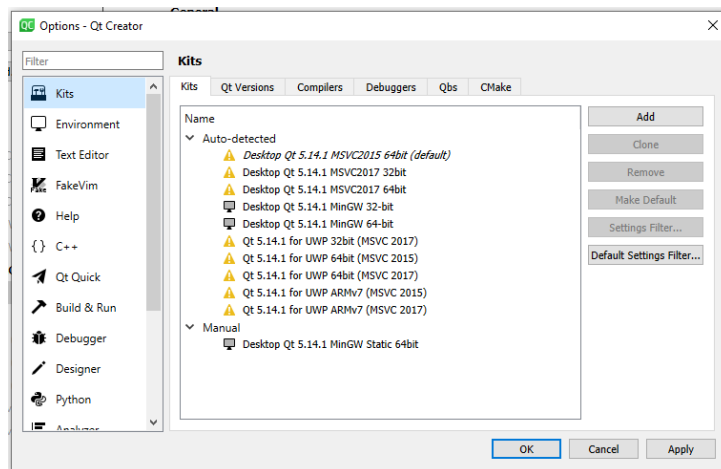
4.2 Staatileine Qt

Qt rakendused on tavaliselt lingitud dünaamiliselt ning kõigi vajalike teekide kaasamine on raskendatud kuna osad neist laetakse programmi töö käigus. Nõnda on raske rakendust arvutis, kuhu pole raamistiku installeeritud tööle saada. Selleks tuleb kaasata programmiga trobikond teeke või kompileerida programm staatiliselt. Staatiliselt kompileeritud rakendus on suurem kuna rakendusega pannakse kaasa kood, mis on dünaamilise rakenduse puhul teekides. Staatiliselt kompileerimiseks on veebileht¹, mis kirjeldab kuidas luua staatileine rakendus Windowsi masinal. Lisaks kirjeldatakse raamistiku loojate arenduskeskkonda kasutades staatilise rakenduse loomist. Viimasel viisil kompileeriti see rakendus. Joonis 13 näitab kompileerimise viiside valiku akent raamistiku loojate arenduskeskkonda kasutades ning kompileerimise viiside muutmise aknaid, kuhu on käsitsi lisatud staatileine kompileerimise viis.

¹ https://wiki.qt.io/Building_a_static_Qt_for_Windows_using_MinGW

Build & Run

- Desktop Qt 5.14.1 MSVC2015 64bit
- Desktop Qt 5.14.1 MSVC2017 32bit
- Desktop Qt 5.14.1 MSVC2017 64bit
- Desktop Qt 5.14.1 MinGW 32-bit
- Desktop Qt 5.14.1 MinGW 64-bit
- Desktop Qt 5.14.1 MinGW Static 64bit**
- Build
- Run
- Qt 5.14.1 for UWP 32bit (MSVC 2017)
- Qt 5.14.1 for UWP 64bit (MSVC 2015)
- Qt 5.14.1 for UWP 64bit (MSVC 2017)
- Qt 5.14.1 for UWP ARMv7 (MSVC 2015)
- Qt 5.14.1 for UWP ARMv7 (MSVC 2017)



Joonis 13. Qt Creator staatile

5 Mõõteseadmega suhtlus

Selles peatükis kirjeldatakse, mida on võimalik vektoranalüsaatoriga teha ja kuidas käib suhtlus vektoranalüsaatoriga.

5.1 Mida vektoranalüsaator võimaldab

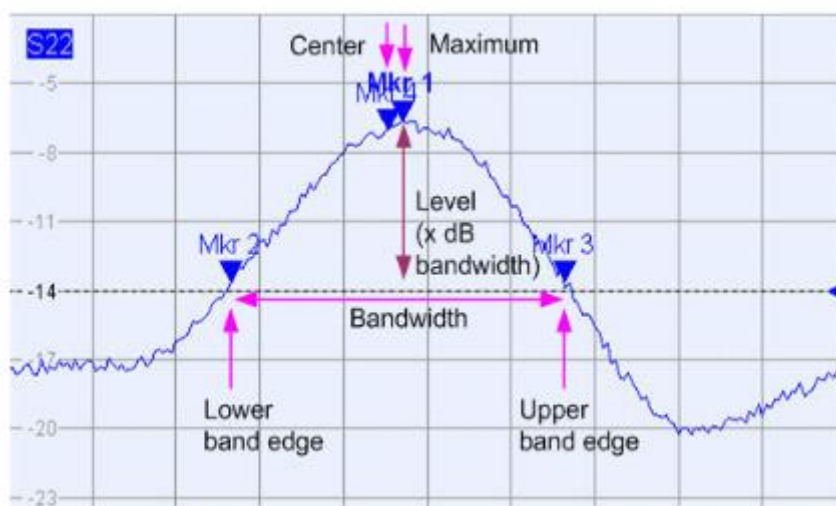
Vektoranalüsaator on seade, mis võimaldab mõõta toote läbivust, näivtakistust ja sisendi peegeldust. Tüüpiline vektoranalüsaatori variant võimaldab mõõta ühesuunaliselt vahemikus 100 kHz kuni 4,5 GHz. Teiste variantidega on võimalik mõõta kahesuunaliselt või suurendada mõõtmise vahemiku 8,5 GHz-ni. [5]

Analüsaator võimaldab salvestada mõõtmisi ja seadeid, mida kasutati mõõtmiseks. Salvestatud seadeid kasutades on võimalik sooritada uusi mõõtmisi. Lisaks mõõtmisega seotud seadetele on olemas globaalsed seaded, mis võimaldavad muuta andmete ja seadete salvestuskohta, osasid kalibreerimise andmeid ja väliste seadete juhtimise seadeid. [5]

Salvestatud seaded on jagatud kolmeks andmete kuvamise seaded, kuvatavate andmete seaded ja mõõtmise seaded. Mõõtmise seaded kirjeldavad kui tihti või millal andmeid mõõdetakse, millist korrektsiooni andmetele rakendatakse ja milline on mõõtmiseks saadav signaal. Kuvatavate andmete seaded näitavad, mida mõõdetakse analüsaatoriga, mis on piirangud saadud andmetele, millistel väärtusel mõõdetakse andmeid, kuidas leitakse ekstreemumeid ja milline töödeldud andmetest on hetkel aktiivne. Kuvamise seadetega kirjeldatakse, millise diagrammiga andmeid kuvatakse ja milliseid mõõdetuid andmeid kuvatakse ning millisel diagrammil. [5]

Väärtuse mõõtmise kohtasid nimetatakse markeriteks. Markereid on kolme tüüpi absoluutväärtusega marker, võrdluseks kasutatav marker ja marker suhtelise väärtusega eelmisest. Kokku on töödeldud andmete kohta kümme markerit, millest üks on aktiivne. Aktiivse markeriga on võimalik sooritada otsingut. Otsida võib maksimume ja miinimume, väärtust või pääsu ja takistus filtreid. Otsides liigutatakse aktiivne marker järgmise leitud väärtuse juurde. Kui puudub aktiivne marker siis kasutatakse esimest

markerit. Otsida võib terves andmete vahemikus või enne otsimist antud vahemikus. Kõigi kümne markeriga on võimalik otsida maksimume ja minime. Nelja markeriga tähistatakse pääsu ja takistus filtreid. Esimene tähistab filtri ekstreemumi. Teise ja kolmanda markeriga tähistatakse filtri algus ja lõppu, mis on defineeritud kasutaja antud väärtuse võrra kukumise või tõusmisega. Neljas marker on teise ja kolmanda markeri keskpunkt. Joonis 14 on analüsaatori dokumentatsioonist joonis, mis näitab markerite paigutust filtrite otsimisel. [5]



Joonis 14. Filtri otsing markeritega

Andmetele pandavad piirangud jagunevad kolmeks. Esimene on x-telje vahemiku väärtuste suurem või väiksem kui määratud väärtus piirang. Piirangu algus ja lõppväärtus võivad olla erinevad, kuid piirang on nende punktide vahel lineaarne. Piirangud järjestatakse loomise järjekorras. Piirangud võivad kattuda x-telje suhtes, kuid mõlemat kontrollitakse andmete saamisel. Teine on vahemiku väärtuste kõikumise maksimaalne väärtus. Sarnaselt eelmisele piirangule järjestatakse kõikumise piirangud loomise järjekorras ja piirangud võivad kattuda. Viimaseks on kompleksdiagrammil kasutatav ringikujuline piirang. [5]

Vektoranalüsaatori tagumisel paneelil on kaks, valikuliselt neli porti, vektoranalüsaatori kaugjuhtimiseks ja teiste seadmetega suhtlemiseks. Kaks tavalist on netikaabli ühendus ja kahekümne viie pinniga port nimega user port. Kaks valikulist porti on kolmekümne kuue pinniga port ja GPIB siini port. Joonis 15 näitab kuidas on paigutatud pordid taga paneelil. Kasutataval analüsaatoril ei ole valikulisi porte. Testirakendus kasutab

analüsaatori juhtimiseks netikaablit ja tootja poolt pakutavat teeki, mis võib suhelda analüsaatoriga üle VXI-11, HiSLIP või RSIB protokollide. [5]



Joonis 15. Kõigi portidega vektoranalüsaatori taga paneel

5.2 Mida kasutatakse testi programmis

Automaattesti aknas testimist alustades luuakse tootja teegiga ühendus analüsaatoriga, kasutades `rsvna_init` funktsiooni. Funktsioon võimaldab luua ühenduse erinevate portide ja protokollidega vastavalt ette antud stringile. Lisaks on võimalik kasutada ühenduse loomiseks seadetele antud nimesid. Järgmine funktsiooni parameeter lubab või keelab seadme ID kontrolli. Eelviimane parameeter kirjeldab, kas seadmega ühendust luues viiakse analüsaator kindlasse olekusse. Viimane parameeter on viide instrumendi sessioonile, mida kasutatakse teegis analüsaatoriga suhtlemiseks. Automaattesti jaoks valitakse netiport VXI-11 protokolliga. Instrumendi olekut ei muudeta ja ID ei kontrollita. Funktsioon tagastab väärtuse funktsiooni töötlemise käigus tekkinud probleemidega. Kui probleeme ei tekkinud siis tagastatakse väärtus null.

Peale iga teegi funktsiooni kasutamist teostatakse veakontroll. Selleks vaadatakse, kas eelnev funktsioon tagastas veakoodi ja kui tagastas, siis kasutatakse teeki funktsiooni veastringi saamiseks. Veafunktsioonile antakse analüsaatori sessiooni viide, veakood ja veastringi puhver. Saanud stringi kirjutatakse veateade automaatse testimise aknasse veateade. Peale veateate kirjutamist suletakse sessioon analüsaatoriga. Kui viga tekib sessiooni loomisel või sulgemisel, siis ei ole vaja sessiooni sulgeda.

Võimenduse ja atenuaatori testimiseks avatakse salvestatud seaded terve toote tööpiirkonnaga ja viie markeriga. Iga filtri testimiseks kasutatakse filtrile vastavat seadete faili, kus on piirang filtri töösagedustel testiinseneri poolt määratud tugevusega ja markerid töösageduste alguses ning lõpus. Kõigi testidega mõõdetakse toote läbivust, kuid tagaplaanil kuvatakse ka sisendi peegeldus. Terve vahemiku seaded jäätakse lahti kuid filtrite seaded suletakse peale nende kasutamist. Seadete avamiseks ja sulgemiseks kasutatakse `rsvna_RecallSetManager` funktsiooni, mis võimaldab nii avada kui sulgeda seadeid. Funktsioon vajab sessiooni viidet, teostatavat operatsiooni ning failinime ja asukohta. Peale avamise ja sulgemise operatsioonile on funktsiooniga võimalik tekitada uus seadete fail, salvestada muudetud seaded ja valida milline salvestatud seadete fail on parasjagu aktiivne.

Teegiga on võimalik muuta markeri kasutamist, asukohta ja tüüpi ning teostada erinevaid otsinguid. Teegist kasutatakse ainult markeri väärtuse lugemise funktsiooni, millega loetakse viie salvestatud absoluutse väärtusega markerit. Funktsiooni kasutamiseks tuleb sellele ette anda sessioon, millist kanalit mõõdetakse, millist markerit küsitakse ja viide numbrile kuhu funktsioon kirjutab real ja imaginaar osa markeri väärtusest. Markerit lugemist kasutatakse võimenduse ja atenuaatori testimisel. Filtri testimisel loetakse sisse kaks markerit, kuid raportisse neid ei kirjutata.

Filtri testi läbimine on kirjeldatud filtri testifaili limiidiga. Limiidi lugemiseks kasutatakse teegi funktsiooni kõigi faili limiitide väärtuste lugemiseks. Selleks antakse funktsioonile ette sessioon ja muutuja kuhu kirjutada limiitide testimise tulemus. Lisaks kirjutatakse raportisse millisel väärtusel on limiidid. Teegi funktsiooniga loetakse sisse kõik limiidid. Funktsioon vajab sessiooni, millist kanalit mõõta, mitu limiiti maksimaalselt loetakse, muutuja kuhu kirjutada mitu limiiti loeti, list kuhu kirjutada limiidi tüüp, listid x-telje algus- ja lõppväärtuste kirjutamise kohaga ning listid y-telje algus- ja lõppväärtuste kirjutus kohaga. Kirja pannakse esimese limiidi y-telje algusväärtus. Igas filtri failis on üks limiit, mille y-telje algus- ja lõppväärtus on võrdsed.

6 Testi dokumenteerimine

Selles peatükis kirjeldatakse, mis on ODS fail, kuidas seda muuta ja, mida tehakse testiprogrammis testiraporti genereerimiseks.

6.1 ODS dokument

ODS fail on avatud standardiga töövihik, mille on välja aretanud OASIS. Fail koosneb avatud dokumendi pakendist, mille sisse on pakitud avatud standardi dokumendi failid. Kõik XML dokumendid pakendis ja dokumendis vastavad XML1.0 spetsifikatsioonidele.

Avatud dokumendi pakendi standardi kohaselt peab dokumendi pakend olema pakitud või pakkimata ZIP fail, kus fail nimega mimetype ja kaust META-INF XML failiga, mille nimi on manifest. Mimetype fail peab olema pakkimata paigutatud esimeseks failiks ZIP kaustas. Failis endas on kirjas millise avatud dokumendiga on tegemist. Manifest failis on kirjas kõik pakendis olevad failid ja nende tüübid. Kui mõni fail pakendis on krüpteeritud siis on manifestis kirjas dekrüptimiseks vajalik informatsioon. Kausta thumbnails pannakse võimalusel dokumendi eelvaade. Eelvaate fail peab olema nimega thumbnail ja PNG formaadis. [6]

Avatud dokumendi standardi kohaselt peab pakendisse pakitud dokumendi pakendis sisalduma sisu või stiilide XML fail. Tüüpiliselt on pakendis mõlemad koos seadete ja metaandmete XML failiga. Dokumendi sisu kirjutatakse sisu faili kere elementi. Sisu elementi võidakse kirjutada vorminguid, skripte ja fontide andmeid vähendatud mahus. [7] Joonis 16 näitab tüüpilise pakendatud dokumendi sisu. Pildi kaustas on pakitult pildid, mida on dokumendis kasutatud.

Nimi	Maht	Maht ti...	Muudetud	L...	A...	...	K...	K...	CRC	Meetod	Karakteristikud	O...	Ve...	/...	Nihe
Configurations2	0	0							00000000						
META-INF	1 199	345							DC5D16A5						
Pictures	7 345	7 345							E6937C17						
Thumbnails	11 177	11 177							2347F016						
content.xml	63 195	4 664	2020-02-27 09:52				-		17EBCDEB	Deflate	Descriptor UTF8	FAT	20	0	3 496
manifest.rdf	899	261	2020-02-27 09:52				-		D268F7B4	Deflate	Descriptor UTF8	FAT	20	0	591
meta.xml	871	433	2020-02-27 09:52				-		74EC3E48	Deflate	Descriptor UTF8	FAT	20	0	3 009
mimetype	46	46	2020-02-27 09:52				-		8A396C85	Store	UTF8	FAT	20	0	0
settings.xml	11 623	1 768	2020-02-27 09:52				-		FEeee389	Deflate	Descriptor UTF8	FAT	20	0	26 876
styles.xml	11 506	2 043	2020-02-27 09:52				-		F189523D	Deflate	Descriptor UTF8	FAT	20	0	910

Joonis 16. ODS fail avatud ZIP failina

Töövihiku dokumendi puhul kirjutatakse kere elementi töövihiku element. Iga töölehe kohta vihikus kirjutatakse üks tabeli element. Tabeli alguses on tabeli tulba elemendid, mille atribuutidega kirjeldatakse tulba vorming. Tulba elementidele järgnevad rea elemendid, mille sees on tabeli kasti elemendid. Kasti elementide atribuutidega kirjeldatakse andmeid ja elementide sees on tekstielement tabeli sisuga. [7]

6.2 Kuidas tulemusi talletatakse

Testide tulemuste talletamiseks kasutatakse ODS faili, kus on raporti struktuur märgenditega, mille asemele kirjutatakse testi andmed. Selleks avatakse testimise lõppedes eraldatakse failid ODS failist ajutisse kausta. Seejärel tehakse märgenditega raporti sisu failist koopia, avatakse see lugemiseks ning avatakse vana sisu fail kirjutamiseks kustutades seal olnud info. Liikudes rida haaval läbi vana sisu faili kirjutatakse iga märgendi asemele testi tulemused. Joonis 17 näitab filtrite testi märgendeid raporti struktuuri failis. Asendanud kõik märgendid valmistatakse ette raporti nimi liites kokku tootenimi, tootekood ja testimiseaeg. Seejärel luuakse ZIP fail raporti nimega mimetype failist, mis lisatakse pakkimata. Seejärel lisatakse ZIP faili kõik teised failid ajutisest kaustast peale originaalse sisu faili. Viimaseks muudetakse ZIP faililaiend töövihiku faililaiendiks ning kustutatakse ajutine kaust.

Filter tests			
filter	Reference dB	Rx1	Rx2
?Filter_0?	?Ref_0?	?Status_1	?Status_2_0?
?Filter_1?	?Ref_1?	?Status_1	?Status_2_1?
?Filter_2?	?Ref_2?	?Status_1	?Status_2_2?
?Filter_3?	?Ref_3?	?Status_1	?Status_2_3?
?Filter_4?	?Ref_4?	?Status_1	?Status_2_4?
?Filter_5?	?Ref_5?	?Status_1	?Status_2_5?
?Filter_6?	?Ref_6?	?Status_1	?Status_2_6?
?Filter_7?	?Ref_7?	?Status_1	?Status_2_7?
?Filter_8?	?Ref_8?	?Status_1	?Status_2_8?
?Filter_9?	?Ref_9?	?Status_1	?Status_2_9?
?Filter_10?	?Ref_10?	?Status_1	?Status_2_10?
?Filter_11?	?Ref_11?	?Status_1	?Status_2_11?
?Filter_12?	?Ref_12?	?Status_1	?Status_2_12?
?Filter_13?	?Ref_13?	?Status_1	?Status_2_13?

Joonis 17. Märgendid raportis

Testiraporti koostamiseks võeti eeskujuks teise toote testiprogrammi, mis kirjutas sarnaselt autori koostatud programmile kirjutas raporti. Nii uues kui teises testitarkvaras asendatakse märgendid kasutades tavalist tekstiasendust. Nõnda on võimalik muuta ainult tekstiks vormitud kasti sisu. Selleks et muuta kasti olevaid andmeid numbriteks tuleb muuta tekstikasti elemendi atribuuti väärtusetüüp ja lisada uus atribuut väärtus, kus on kirjas tekstikasti väärtus. Vormingu muutmiseks tuleb muuta vormingu elemente, millele kasti element viitab või muuta millisele vormingu elemendile viidatakse. Rakenduses ei muudeta tekstikaste numbriga kastideks ega muudeta vormingut.

Raporti sisu faili kopeerimiseks, ZIP faili ümbernimetamiseks ODS failiks ja ajutise kausta kustutamiseks kasutatakse Windowsi käsurea käsked copy, rename ja rd. ODS faili avamiseks ja ZIP faili loomiseks kasutatakse 7-Zip eraldiseisvat konsoolirakendust. Teises testitarkvaras kasutati quazip¹ teeki ZIP faili lugemiseks ja kirjutamiseks. Liikudes uue tarkvara arendusel Linuxilt Windowsile tekis probleeme nimetatud teegi kompileerimisega ja seega võeti kasutusele 7-Zip.

6.3 Kuidas veel on võimalik talletada

Kui ei piisa dokumendi kirjutamisel ainult tekstiks vormitud lahtrite teksti muutmisest, siis tasub ära XML töötlemine. Qt raamistikus, milles rakendus on kirjutatud, on

¹ <https://github.com/stachenov/quazip>

suhteliselt lihtne töödelda XML faile. Selleks tuleks lisada XML moodul projekti, avada ja töödelda dokumendi sisu fail. Seejärel tuleks leida tekstikast, kus on märgend, mida asendatakse ning muuta märgend andmeteks. Vormingu muutmise soovi korral on võimalik muuta tekstikasti atribuute, millest üks on vormingu nimi. Viimaseks tuleb muudetud XML puu salvestada sisu faili.

Apache OpenOffice ja LibreOffice pakuvad liidest, millega on võimalik kasutada programmi funktsionaalsust teises rakenduses. Nõnda on võimalik muuta liidese kaudu spetsiifiliste lahtrite sisu. Lisaks sisu muutmisele on võimalik muuta vormingut. Liideselega oleks võimalik luua terve raporti fail nullist, kuid see oleks tükk tühja tööd. Liidese kasutamiseks peab olema installeeritud testimise masinas vastav kontoritarkvara ja arendus arvutis tarkvara komplekt liidest kasutava tarkvara kompileerimiseks.

7 Kokkuvõte

Töö kirjutamise tarvis loodi rakendus, mis on võimeline juhtima autori töökoha toote uut versiooni ja suudab läbi viia teste selgitamaks, kas toode on töökorras. Tänu rakendusele on muutunud toote testimine paarikümne minutilisest protsessist kolme minutiliseks. Töö eesmärgiks oli dokumenteerida loodud rakendus.

Esimeses peatükis kirjutati, et vanal tootel oli üks võimendatav sisend, mis jagati kaheks ja saadeti väljunditesse peale individuaalset filtreerimist. Uuele tootele lisati mõlemale väljundile atenuaator. Vana toote testimiseks kontrolliti võimendust ja mõlemal väljundil kõiki filtreid. Uue toote testimiseks lisati mõlema atenuaator testimine. Lisaks kirjeldati millist funktsionaalsust lisati uue toote juhtimise programmi.

Teises peatükis kirjutati, millist releed kasutati väljundite mõõteseadmesse juhtimiseks ja kuidas seda kontrolleriga juhti. Filtri valimiseks kasutati kuute signaali, mis saadi üle SPI siini juhitava GPIO laiendilt. Atenuaatori juhtimine oli oluliselt erinev, kuid käis üle sama SPI siini. Tulenevalt samal siinil olevate laiendite ja atenuaatorite juhtimise erinevutest tuleb peale iga filtri muutmist muuta atenuaatorite väärtusi. Relee ja toote juhtimiseks kasutatava kontrolleriga suheldakse kui HID-USB seadmega.

Kolmandas peatükis kirjutatakse, et programm on kirjutatud c++ keeles ja Qt raamistikuga. Programmi ühest arvutist teise saamiseks kasutatakse staatilist kompileerimist.

Neljandas peatükis kirjeldatakse mõõteseadme funktsionaalsust ja kuidas seadmega suhelda. Mõõteseadme on võimeline mõõtma toote läbivust, näivtakistust ja peegeldust. Testimisel kasutatakse salvestatud mõõtmise seaded. Seadmega suhtlus käib üle VXI-11 protokolliga ja kasutades tootja teeki.

Viimases peatükis kirjeldatakse raporti kirjutamise protsessi. Raporti kirjutamiseks muudetakse ettevalmistatud töövihikusse, mis on ODS formaadis. ODS fail on kindla struktuuriga ZIP fail, kus dokumendi sisu on XML failis. Lahtrite muutmiseks asendatakse kohahoidja tekst testi tulemustega sisu XML failis.

Kasutatud kirjandus

- [1] „Coaxial Switch Technical Overview,“ [Võrgumaterjal]. Available: <https://www.keysight.com/zz/en/assets/7018-06709/technical-overviews/5952-1873.pdf>. [Kasutatud mai 2020].
- [2] „MCP23017/MCP23S17 Datasheet,“ Microchip Technology Inc., [Võrgumaterjal]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/20001952C.pdf>. [Kasutatud aprill 2020].
- [3] „PE43705 Datasheet,“ pSemi Corporation, [Võrgumaterjal]. Available: <https://www.psemi.com/pdf/datasheets/pe43705ds.pdf>. [Kasutatud aprill 2020].
- [4] „Qt Documentation,“ [Võrgumaterjal]. Available: <https://doc.qt.io/qt-5/signalsandslots.html>. [Kasutatud mai 2020].
- [5] „ZND User Manual,“ [Võrgumaterjal]. Available: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/pdm/cl_manuals/user_manual/1173_9557_01/ZND_UserManual_en_51.pdf. [Kasutatud mai 2020].
- [6] „Open Document Format for Office Applications (OpenDocument) Version 1.2 Part 3: Packages,“ OASIS, [Võrgumaterjal]. Available: <http://docs.oasis-open.org/office/v1.2/OpenDocument-v1.2-part3.pdf>. [Kasutatud aprill 2020].
- [7] „Open Document Format for Office Applications (OpenDocument) Version 1.2 Part 1: OpenDocument Schema,“ OASIS, [Võrgumaterjal]. Available: <http://docs.oasis-open.org/office/v1.2/OpenDocument-v1.2-part1.pdf>. [Kasutatud aprill 2020].