jamk.fi

Mittarin testiautomaation tuotteistaminen

Asko Kämäräinen

Opinnäytetyö
Helmikuu 2021
Tietojenkäsittely ja tietoliikenne
Insinööri (AMK), tieto- ja viestintätekniikan tutkinto-ohjelma
Ohjelmistotekniikka

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Kämäräinen, Asko	Opinnäytetyö, AMK	Helmikuu 2021
	Sivumäärä 30	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi

Mittarin testiautomaation tuotteistaminen

Tutkinto-ohjelma

Insinööri (AMK), tieto- ja viestintätekniikka

Työn ohjaaja(t)

Matti Mieskolainen, Jouni Huotari

Toimeksiantaja(t)

Aidon oy, Tarmo Hyttinen

Tiivistelmä

Aidon Oy:llä oli tarve saada sähkömittarin testiautomaatio nostettua samalle tasolle kuin samaisen sähkömittarin kommunikaatiomoduulissa. Tätä varten suunniteltiin ja toteutettiin mittarin testiautomaatiopenkki, joka käsittää hardware-osuuden ja pikaisen testin, että kaikki toimii.

Tavoitteena oli saada tuotteistettua testipenkki nopeasti päivitettäväksi ja plug & play -periaatteella toimivaksi. Tämä vaatia hieman penkin suunnittelulta aikaa, jotta kytkettävyys ja testattavan laitteen vaihto saadaan sujuvaksi.

Testipenkki toteutettiin liittämällä ympäristöön pc, NI-USB-6525 I/O -laite ja itse testattava mittari. Ohjelmistopuolella merkittävässä osassa olivat Robot Framework, Donbot, Labview ja Glogg login lukemiseen.

Penkin suunnittelu ja rakennus vaati perehtymistä sähkötekniikkaan ja elektroniikkaan sekä ohjelmistojen hallintaa ja konfiguroimista. Useat digitaalisesti ohjattavat fyysiset toiminnot vaativat toimiakseen erilaisten jännitteiden käsittelyä NI-USB-6525 input -porteissa ja tähän piti pystyä laskemaan sopivat vastukset ja/tai kondensaattorit toiminnan varmistamiseksi.

Lopputulemana saatiin erillinen mittari, testiympäristö ja robottia pyörittävä pc-kombinaatio, joka on helposti vaihdettavissa testin alla olevalle laitteelle. Alustavissa testeissä laite toimii hyvin ja luotettavasti. Tähän tullaan tulevaisuudessa kehittämään lisänä uusia ominaisuuksia testejä varten ja itse testikirjasto, joka jätettiin ulos projektin tavoitteista sen laajuuden vuoksi.

Avainsanat (asiasanat)

Robot Framework, testiautomaatio, sähkömittari, sulautetut järjestelmät

Muut tiedot (Salassa pidettävät liitteet)

jamk.fi

Description

Author(s)	Type of publication	Date
Kämäräinen, Asko	Bachelor's thesis	February 2021
		Language of publication: Fin
	Number of pages: 30	Permission for web publication: x

Title of publication

Meter test automation and its productization

Degree programme

Software engineering

Supervisor(s)

Matti Mieskolainen, Jouni Huotari

Assigned by

Aidon oy, Tarmo Hyttinen

Abstract

Aidon Oy had a need for an automated test environment for their electricity meter to match their communication module test automation. For this it was needed to go thru the designing and completing the testbench environment including the hardware and a quick test to ensure that everything is working correctly.

The goal was to achieve a working product for plug & play -functionality and easy upgradability for the varied meters tested in the environment.

This was achieved by integrating the pc, NI-USB6525 I/O -device and the device under testing to the bench. From the software side the essentials are Robot Framework, Donbot, Labview and Glogg -log parser.

The design and implementation took some understanding of electronics and electricity as well as understanding the used software and configuring them for the job. Most of the physical functions that were done with digitalized switches and so on needed some altering of the voltages so it would be readable from the NI-USB_6525 device. Some of the connections needed resistors and/or transistors to work properly.

At the end was fully functional test environment consisting of a computer to run the testing software, testing hardware and the device under testing. Preliminary tests show that everything seems to work properly, and the future implementations of new added features and the test library can begin.

Keywords/tags (subjects)

Robot framework, test automation, electricity meter, embedded systems

Miscellaneous (Confidential information)

Sisältö

Kuviot

Lyhenteet ja käsitteet

Bugi

Ohjelmointivirhe tai virheellinen toiminto.

Bugzilla

Bugien ja vikojen raportointiportaali.

CB

Circuit Breaker, katkorele, katkaisee tai kytkee sähköt mittarille.

Donbot

Aidonin oma ohjelmisto, joka yhdistää Robot Frameworkin mittariin.

DUT

Device Under Testing, Sähkömittari tässä tapauksessa.

Falcon

Aidonin uusimman mittarisukupolven työnimi.

Flashaaminen

Mittarin ohjelmiston päivittäminen.

Gitlab

Versionhallintajärjestelmä.

HES

Head End System, käsittelee mittareiden lähettämän datan.

Jenkins

CI automatisaatio työkalu.

Labview

Ohjelmointiympäristö National Instruments laitteiden käyttöön.

OBIS koodi

Obis standardin mukainen teksti muotoinen arvo.

PTS

Portable Test System, Ulkoinen ohjattava virtalähde.

Python

Ohjelmointi/scriptauskieli.

RFA

Robot Framework.

RPA

Robotic Process Automation, ohjelmistorobottien käyttö toistettavien tehtävien automatisointiin.

RTC

Real Time Clock – Pitää kellon ajan, vaikka virransyöttö katkeaa.

RxD

Receive data, datan vastaanotto.

Tamperingkytkin

Kytkin mittarin kannessa, joka kytkeytyy päälle, jos kansi avataan. Lähettää tapahtuman HESiin kun näin käy.

TxD

Transmit Data, datan lähetys.

1 Taustat ja tavoitteet

1.1 Aihe

Opinnäytetyön aihe tuli suoraan yritykseltä, jossa olin suorittamassa työharjoitteluani kesällä 2020. Aihe tuli tarpeesta automatisoida sähkömittarin ohjelmiston testausta. Tätä ennen mittarin ohjelmisto on testattu manuaalisesti ja se syö todella paljon työtunteja. On arvioitu, että yhden mittarin testaamiseen menee aikaa noin 1.5-2 viikkoa.

On sanottava, että Aidonilla on sulautettujen järjestelmien testiautomaatio hyvällä tolalla jo kansainvälisellä tasollakin. Tiimimme muut jäsenet ovat pitäneet huolta siitä, että moduulipuolella testiautomaatio on mahdollisimman kattava ja luotettava. Testilaitteitakin löytyy jo satoja. Nyt tarkoituksena on ottaa harppaus mittaripuolella ja saada se samaan tasoon moduulitestauksen tehokkuuden kanssa.

Mittarin testaus on siitä erikoinen osa-alue Aidonilla, että mittarin ohjelmistoa ei päivitetä kovin usein, joten testaaminen tulee aalloittain. Kun tulee uusi ohjelmistoversio, niin kaikki mittarimallit testataan kertaalleen läpi. Hyväksyttyjen testien jälkeen mittariohjelmisto menee tuotantoon eikä ohjelmistoa sen jälkeen enää voida päivittää ilman sinettien murtamista. Tämä toimii eri lailla kuin mittarissa oleva kommunikaatiomoduuli, joka voidaan päivittää langattomasti.

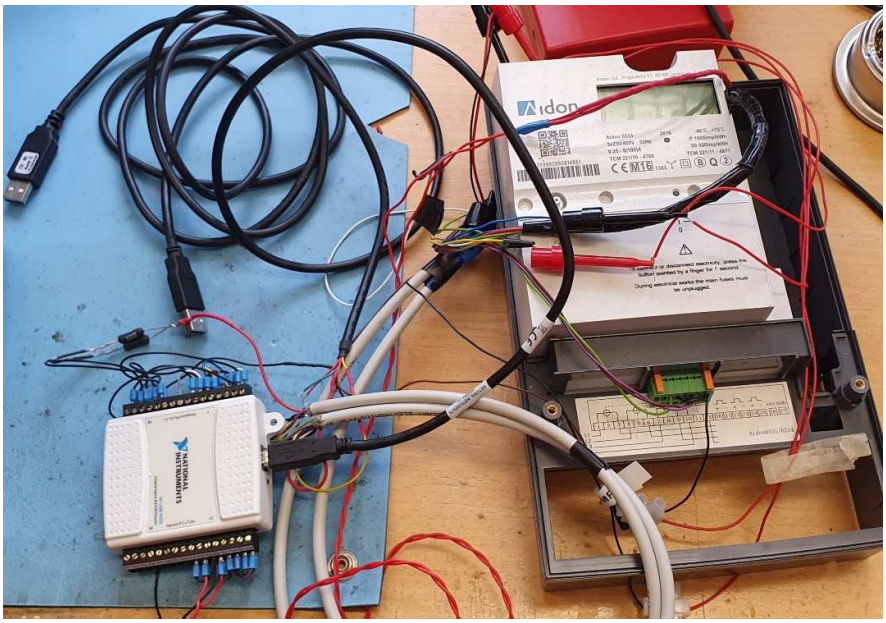
Mittarimalleja alkaa Aidonilla olla kymmeniä, joten automaation kehittäminen on ajankohtaista ja taloudellisesti järkevää. Kaikkia testejä ei tosin ole mahdollista automatisoida, ainakaan tämän opinnäytetyön raameissa koska osa testeistä vaatii datan lukemista mittarin näytöltä, eikä esim. koenäön implementaatio ole työmäärään nähden toistaiseksi järkevää.

Raaka-arvio automaation hyödystä on lyhentää testiaikaa per mittari 4-5 päivään. Testeissä on useita ns. ”yön yli” testejä, jotka kestävät noin 16 tuntia. Silti jos mittarin testiautomaatio pyörii ympärivuorokauden jatkuvasti, arvioin että testit ajetaan läpi 4-5 vuorokaudessa. Opinnäytetyön piiriin ei testitapausten laajuuden takia sisällytetä itse testejä kuin ehkä muutama esimerkin vuoksi.

Tavoitteena oli suunnitella ja toteuttaa testipenkki, jolla tulevaisuudessa testaus suoritetaan. Testitapauksia oli hieman alle 300 per mittari, joten kaikkien testien automatisointi vie paljon aikaa. Testejä joutuu muun lisäksi vielä yksilöimään jokaiselle mittarille erikseen, koska mittarimallit eroavat toisistaan fyysisillä ja ohjelmallisilla ominaisuuksiltaan paljon.

Yrityksessä on aloitettu mittarin testiautomaation rakentaminen pari kertaa aikaisemminkin, mutta jokaisella kerralla kehitys on jäänyt kesken joko aika- tai resurssipuutteiden takia. Kuviossa 1 on esimerkki yhdestä prototyypistä.

Edellisessä prototyypissä oli perustoiminnallisuutta rakennettu lähinnä mittarikommunikaatioon ja saatu se toimivaksi. Edellisessä versiossa kaikki johdot oli juotettu suoraan mittarin piirilevyyn ja näin ollen mittarin vaihtaminen testipenkkiin oli hidasta ja hankalaa. Tämän työn aiheena oli nopeuttaa prosessia ja parantaa toiminnallisuuksia.



Kuvio 1. Prototyyppi, joka oli opinnäytetyön lähtökohta

Testipenkissä käytettävä rauta:

- Windows työpöytätietokone
- PPS3.3C -ohjattava virtaemulaattori
- National Instruments -NI-USB-6525 I/O-laite
- Sähkömittari (DUT)

Testipenkissä käytettävät ohjelmistot:

- Windows 10
- Robot Framework 3.2.2
- Donbot
- National Instruments NI-MAX
- Python 3.9
    - Robotframework requests
    - Websocket-client
    - Pytz
    - Pyserial
    - Pythoncrc

- Jsonschema
- robotbackgroundlogger

Tämän opinnäytetyön kohteena oli uusi Aidon ”Falcon” 7534 DC (Direct Current) -mittari, jonka pohjalta oli helpointa lähteä rakentamaan muiden mittareiden testejä, koska tarkoituksena on, että mittarit ovat taaksepäin yhteensopivia. Uusimmassa mittarissa on kaikki ominaisuudet ja vanhempiin mittareihin ei niitä ole ainakaan lisää tullut.

1.2 Toimeksiantaja

Aidon Oy on pohjoismaiden johtava älykkäiden energianmittausratkaisujen ja älyverkkosovellusten kehittäjä ja toimittaja ja on perustettu vuonna 2004 Jyväskylässä. Toimipisteitä on kolmessa maassa: Suomessa, Ruotsissa ja Norjassa. Pääkonttori sijaitsee Jyväskylässä. Aidonilla on vankka osaaminen sulautettujen järjestelmien kehittämiseen, testaamiseen ja käyttöönottoon. (Tämä on Aidon n.d.)

1.3 Tutkimusmenetelmät

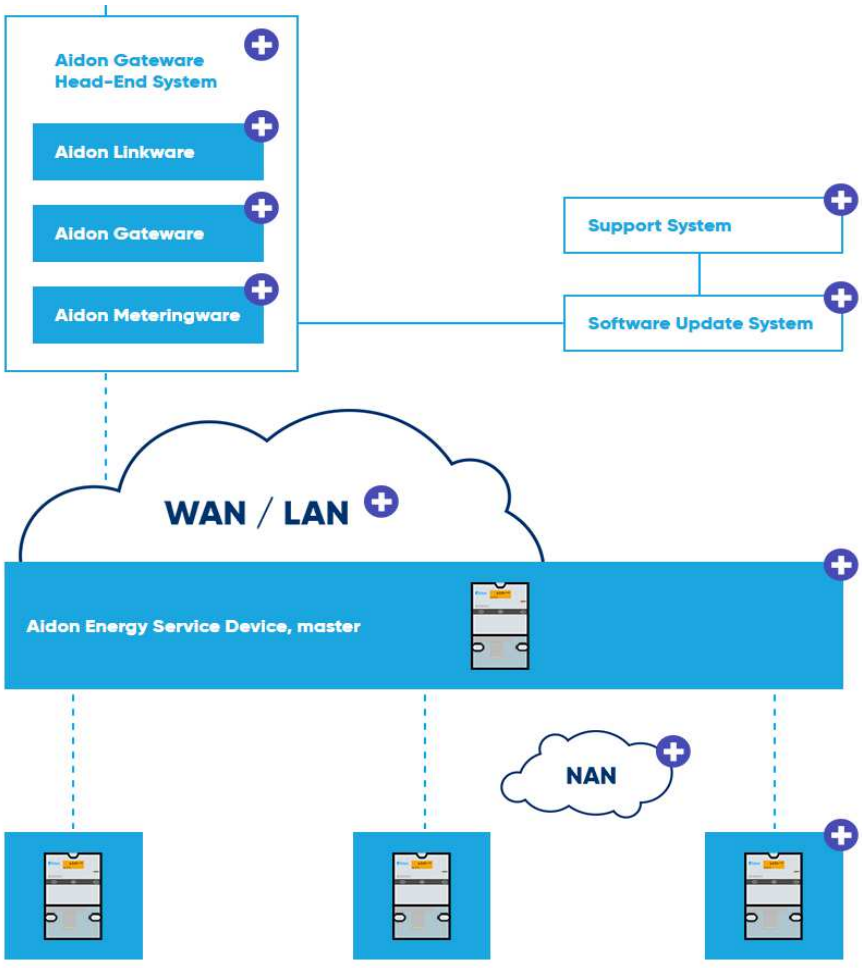
Projektissa käytettiin soveltavaa tutkimusmenetelmää, joka tarkoittaa tiedon hankkimista ja omaksumista toteuttamalla jokin käytännön sovellus. Taustalla käytettiin olemassa olevaa tietoa ja sovellettiin sitä kyseessä olevaan ongelmaan. Soveltavan tutkimuksen kohteena on usein jonkin asian kehittäminen tai parantaminen. Tutkimustuloksien vertailu on tärkeää teorian ja käytettyjen menetelmien välillä, jotta päästään haluttuun lopputulokseen. (Soveltavasta tutkimuksesta n.d.)

Projektissa toteutettiin testiautomaatio-ympäristö, joka kehittää edelleen olemassa olevaa testiautomaatiota ja tutkimuksen pohjana on käytetty toimeksiantajan dokumentaatiota ja internetistä löytyvää tietoa.

2 Ympäristön ja testauksen kuvaus

2.1 Ympäristö

Opinnäytetyön aiheena oleva sähkömittari on yksi osa Aidonin luentajärjestelmää (ks. kuvio 2). Sähkömittarin tehtävänä on mitata mittauspaikan sähkönkulutusta, havaita mahdollisia virhetilanteita sekä mitata sähkön laatua. Kommunikointi luentajärjestelmään tapahtuu järjestelmämoduulin välityksellä. (Luentajärjestelmän.d.)



Kuvio 2. Aidonin luentajärjestelmä (Luentajärjestelmä n.d)

Sähkömittari koostuu kahdesta moduulista: mittarista ja järjestelmämoduulista (ks. kuvio 3).

Kuvio 3. Sähkömittarin moduulit

2.2 Testaus Aidonilla

Aidonilla käytetään sekoitusta vaatimus- ja riskipohjaisesta testaamisesta, joka on koettu joustavaksi tavaksi kokonaisuuden kannalta.

Vaatimuspohjainen testaaminen on juuri sitä miltä se kuulostaa. Vaatimusmäärittelyn pohjalta voidaan määritellä testattavat ominaisuudet ja tilanteet. Tämä pitää sisällään toiminnalliset ja ei-toiminnalliset testit, kuten myös

käytettävyys, luotettavuus ja suorituskykytestaamisen. (What is Requirements based Testing? n.d.)

Riskipohjainen testaaminen on menetelmä, jossa pyritään tunnistamaan mahdolliset riskit tuotettavassa ohjelmistossa tai tuotteessa. Menetelmään kuuluu määrittäminen tuotteen monimutkaisuudesta, kriittisyydestä liiketoiminnalle, käyttömäärästä ja mahdollisista heikoista kohdista. Riskipohjainen testaaminen asettaa etusijalle avain featureiden ja toimintojen testaamista mahdollisten ongelmien löytämiseksi. (Risk Based Testing: Approach, Matrix, Process & Examples n.d.)

Aidonilla testaaminen on pitkälti automatisoitu Jenkinsin ja Gitlabin päälle CI/CD ketterän työtavan mukaisesti. Jenkin alla pyörii useita runnereita, joissa testit suoritetaan käyttäen Robot Frameworkia. Testauksen kulku näkyy kuviossa 4. Testiautomaation lisäksi joudutaan jonkin verran tekemään manuaalista testaamista paikoissa, joihin automaation rakentaminen ei ole järkevää.

Kuvio 4. Testauksen vuokaavio

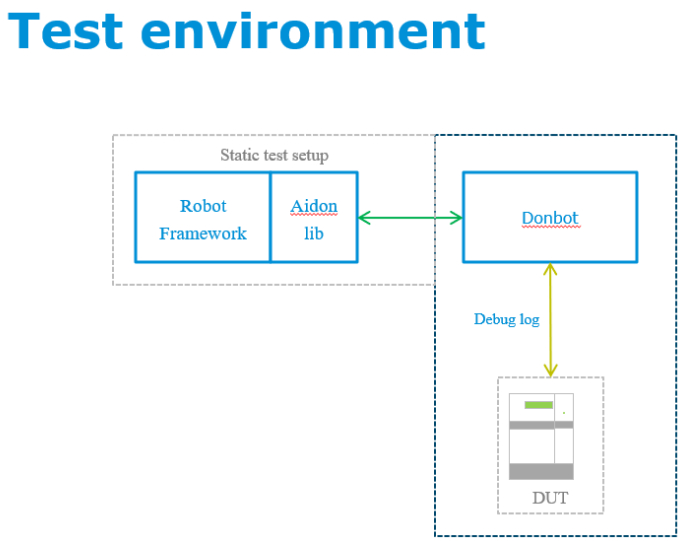
Mittaritestauksen tapauksessa mittari ja mittarin ohjelmisto tuotetaan Kiinassa ja siellä tehdään jonkinlainen alustava regressiotestaus, mutta Aidonilla tehdään perusteellisempi ja kattava regressio ja sisäinen hyväksyntätestaus. Jos softasta löydetään bugeja, raportoidaan ne Bugzillaan, Kiinan päässä tehdään korjaus ja sitten

testataan uudestaan. Tässä alkaa nopeasti nähdä hyödyt testauksen automatisoinnista.

Kiinassa tehdään rautapuolelle vanhennustestausta, joka suoritetaan altistamalla mittari 80 celsiuksen lämpötilalle viideksisadaksi tunniksi tai enemmän. Samaan aikaan mittarista mitataan energiankulutusta ja hälytyksiä seurataan ja verrataan referenssimittariin normaalissa huoneenlämmössä. Nyrkkisääntönä voidaan pitää, että noin 1000 tuntia vastaa yhtä vuotta näissä olosuhteissa. Käyttöaikaa mittareille kentällä ollaan luvattu 15 vuotta.

3 Työkalut

Käyn tässä läpi tärkeimmät työkalut, joita tarvittiin työn toteuttamiseen. Kuviossa 5 on esitelty toimintaympäristö.



Kuvio 5. Lohkokaavio testiympäristöstä

3.1 Robot Framework

Robot Framework on avoimen lähdekoodin RPA (Robotic Process Automation) prosessien automatisointiin tarkoitettu testaus työkalu, jota ylläpitää suomalaislähtöinen Robot Framework Foundation. (Robot Framework n.d.)

RFA:n kehitys perustuu Pekka Klärckin diplomityöhön vuodelta 2005 nimeltään ” Data-Driven and Keyword-Driven Test Automation Frameworks”, jonka ensimmäinen julkaisu tapahtui saman vuonna Nokian Networksilla. Avoimeksi lähdekoodiksi RFA julkaistiin ensimmäisen kerran vuonna 2008. (Robot Framework n.d.)

Robot Framework käyttää erittäin helppolukuista syntaksia käyttäen avainsanoja ja on integroitavissa lähes mihin tahansa ympäristöön, käyttöjärjestelmästä riippumatta, tehden siitä näin ollen suosituimman testiautomaatiotyökalun alalla. RFA on kirjoitettu Pythonilla ja tukee Pythonilla ja Javalla tehtyjä kirjastoja, jotka lisäävät toiminnallisuutta valtavasti. Myös omien kirjastojen luominen projektikohtaisesti on vaivatonta. (Robot Framework n.d.)

Robot Framework on julkaistu avoimen lähdekoodin Apache License 2.0 -lisenssin alla ja on ilmainen käyttää kaikille (Robot Framework n.d.).

3.2 Donbot

Donbot on Aidonin tilaama ja Devecton kehittämä Labview-pohjainen ohjelmisto, joka toimii Websocket-linkkinä Robot Frameworkin ja DUT:in välillä. Donbot täytyy asentaa ja konfiguroida tarkoitusta varten, koska sitä käytetään työpaikan sisällä laajasti erilaisissa automaatioissa. Systeemimoduuli automaatiotestauksessa Donbot

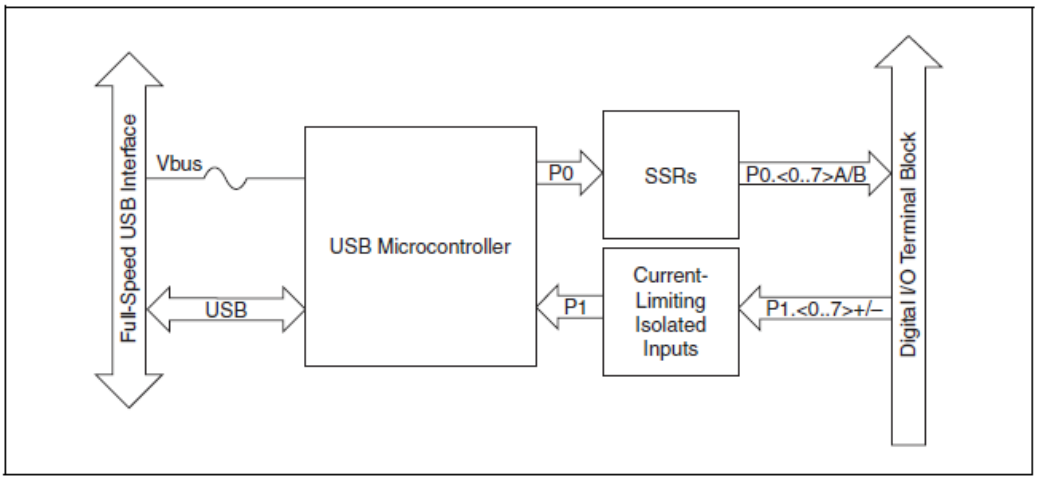
toimii linkkinä Robot Frameworkin ja Dexter Morganin välillä, joka simuloi Head End Systemiä (HES) automaatioympäristössä.

3.3 PTS3.3C

PTS3.3C on ulkoinen ohjattava virtalähde, joka kytketään mittariin ja jolla simuloidaan sähköverkkoa. PTS voidaan syöttää mittariin virtaa, jännitettä, simuloida sähkökatkoja ja esimerkiksi muuttaa virran taajuutta. Parasta PTSsä on että, sitä voidaan ohjata suoraan Robot Frameworkilla määrittämällä ensin sille ohjausportti ja baudinopeus. PTS tottelee omaa syntaksiaan, joka käy ilmi virtalähteen manuaalista. (PTS3.3C Manual 2017.)

3.4 National Instruments NI-USB-6525

NI-6525 on National Instrumentsin valmistama digitaalinen input/output-laite, jolla voidaan ohjata ja lukea signaaleja. Käytimme kyseistä purkkia ohjaamaan normaalisti manuaalisia kytkimiä ja lukemaan mittarin moduuliterminaalista tulevia signaaleja. NI-6525 on 16-kanavainen, joista 8 on output-ohjausta ja 8 on input-lukemiseen (ks. kuvio 6). (USB-6525 n.d.)



Kuvio 6. NI-USB-6525 block diagram

4 Toteutus

4.1 Testipenkki

4.1.1 Yleisesti testipenkistä

Testipenkin toteutus piti aloittaa tutustumalla laajaan dokumentaatioon mitä Aidonilla on tietokannassaan. Nämä tiedot ovat ymmärrettävästi liikesalaisuuksia, joten kovin yksityiskohtaisesti ei voi avata mittarin toimintaa ja rakennetta. Dokumentaatiosta löytyi kuitenkin kattavasti tiedot mittarin ominaisuuksista ja rakenteesta, joita tarvittiin ympäristön toteutukseen kuten, Systeemimoduulin terminaalin pinnijärjestyksen, kytkentäkaaviot ja sähkökaaviot. Koska poiketen Aidonin aikaisemmista testiautomaatioympäristöistä mittaritestiympäristössä ei tulla käyttämään systeemimoduulia ollenkaan, jotta mahdollisia virheitä aiheuttavat järjestelmät ympäristössä saadaan minimoitua. Tämä aiheuttaa toisaalta sen, että

Robot Frameworkille joudutaan rakentamaan kirjastot OBIS ja E2 koodeille, joita mittari lukee. Normaalisti tämän hoitaisi systeemimoduuli.

Jotta mittarin testien automatisointi olisi sujuvaa, jouduttiin miettimään erilaisia keinoja, joilla automatisoida ennen fyysiset suoritteet testaamisessa. Näitä ovat mittarin superkondensaattorin tyhjentäminen, joka ylläpitää RTC mittarissa sähkökatkon aikaan, CB toiminta, ohjausreleen asento, program pin ja tampering kytkin. Näistä toiminnoista program pin ja tampering kytkin päätettiin jättää tulevaisuuden kehityskohteiksi ja sitä varten juotettiin valmiiksi kaapelit, joten lisäys on tulevaisuudessa mahdollista vain testikirjastoja kehittämällä. Fyysisten toimintojen automatisoinnin päätettiin tehdä käyttäen NI-USB-6525 IO-ohjainta, koska näitä löytyi toimistolta valmiiksi ja niiden toiminnot ovat riittävät.

4.1.2 Mittari

Koska tarkoituksena oli tuotteistaa mittarin testiautomaatio, päätettiin tehdä tästä mallia: Plug & Play, joka tosin tarvitsi hieman muokkausta ennen kuin se otettiin käyttöön, mutta tarkoitus olisi tulevaisuutta ajatellen, että mittari tarvitsee avata vain kerran. Dokumentaatiosta selvitettiin etukäteen tarvittavat liitännät ja suunniteltiin systeemimoduulin paikalle tulevaan muovikanteen paikat liittimille D25 ja D9, jotta DUT:in vaihtaminen olisi tulevaisuudessa helppoa ja nopeaa. Koska ennen uusi ohjelmisto flashattiin mittariin avaamalla ensin kuoret ja sitten käyttäen erillistä Flash moduulia, ajateltiin nopeuttaa tätä toimenpidettä juottamalla kaapelit D9-liittimeltä mittarin kommunikaatiopinneihin. Näin ollen mittarin ohjelmistoa päivittäessä ei tarvitse enää kytkeä laitteita kuin kannessa olevaan liittimeen.

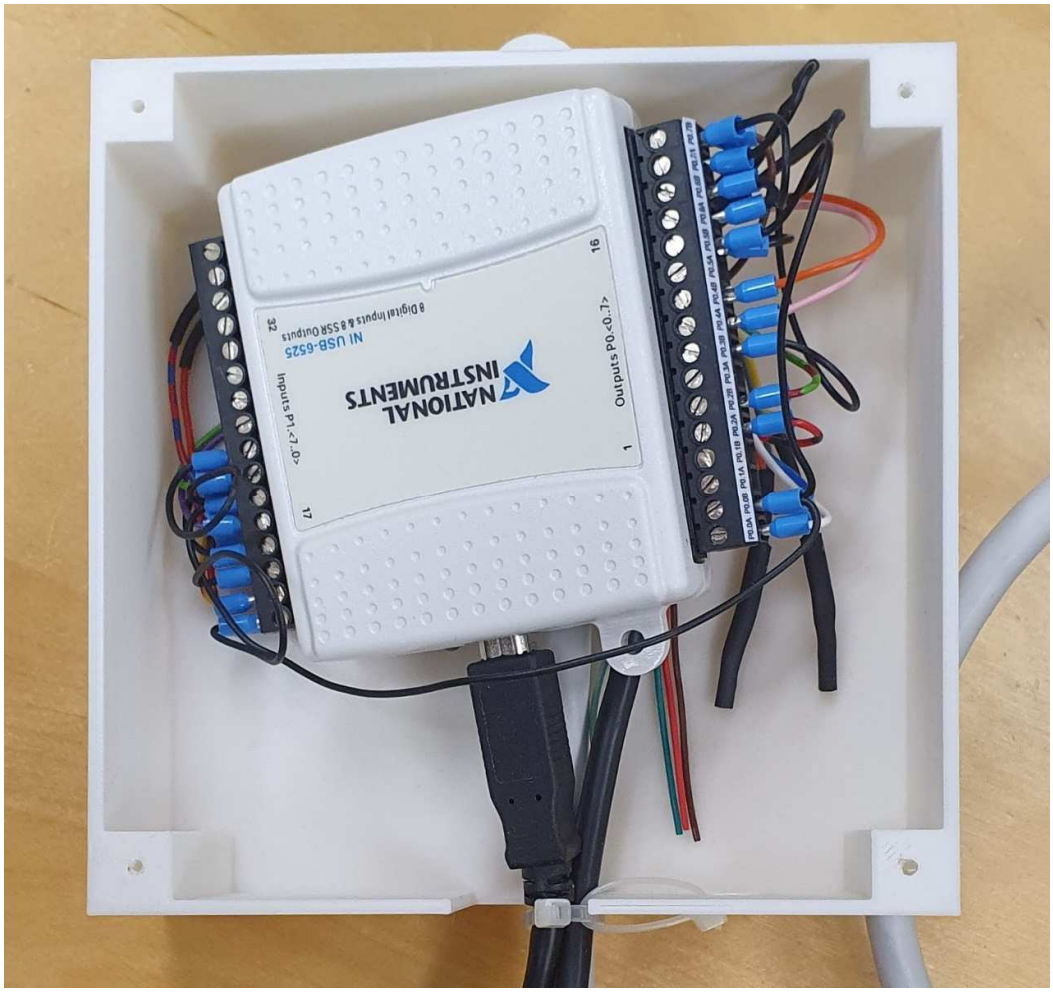
Samaa periaatetta käytettiin, kun toteutettiin ohjauksen digitalisaatiota fyysisille toimenpiteille. Kanteen tuli D25 liitin, josta kaapelit juotettiin mittarin piirilevylle tarvittavia toimintoja varten. Tärkeimmät toiminnot ovat CB:n ohjaaminen digitaalisesti ja superkondensaattorin tyhjentäminen. CB ohjaaminen on normaalisti tehtävä fyysisellä painikkeella, jolla voidaan kytkeä mittari johtavaksi tai

johtamattomaksi. CB:tä voidaan ohjata joko ohjelmistolla tai fyysisesti, joten oli tärkeää saada fyysiset painallukset tehtyä digitaalisesti. Testejä, jotka liittyvät Circuit Breakerin toimintaan ja superkondensaattorin tyhjentämiseen, on useita, joten tämä oli prioriteettilistan kärjessä toiminnallisuuksissa.

Kommunikaatio mittariin suoritetaan systeemimoduuli terminaalin kautta, jonka toiminta selvisi mittarin dokumentaatiosta. Koska tiedot ovat liikesalaisuksia ja tietoturvariski, voidaan sanoa vain, että kommunikaatio suoritetaan käyttämällä poikkileikattua USB-kaapelia, josta tarvitaan ainoastaan maadoitus, TxD ja RxD -johdot. Toinen pää menee tietokoneeseen ja toinen mittarin terminaaliin. Tämäkin järjestettiin kulkemaan samaa yhdyskaapelia pitkin kuin mitä NI-USB-6525 ohjaus kulkee, jotta vedettävien johtojen määrä pysyisi minimissä. Tähän tarkoitukseen riittää 18x.25 monijohdekaapeli.

4.1.3 NI-USB-6525

National Instrumentsin I/O-ohjainta käytettiin fyysisten toimintojen digitalisoimiseen, mittarin ohjausreleen asennon lukemiseen ja mittarin POUT-tilojen lukemiseen(ks. kuvio 7). Ensimmäiset fyysiset ohjaukset olivat CB:n napin painaminen, superkondensaattorin tyhjentäminen kytkemällä kondensaattori oikosulkuun digitaalisesti ja ohjelmointi pinnin päälle/pois kytkeminen. Tulevaisuudessa tähän tullaan kytkemään vielä peukalointikytkimen ohjaus päälle/pois ja ADE mittauspiirin oikosulkeminen. Input puolella luetaan mittarilta tulevat signaalit kuten POUT_1, POUT_2 ja PIN_1, jotka kertovat erinäisiä asioita mittarin tilasta. NI-USB toimii yhdessä Robot Frameworkin ja Donbotin kanssa ja se vaatii omat ajurinsa toimiakseen. Asennettiin uusimmat ajurit, jotka National Instrumentsin sivuilta löysin ja niiden mukana tuli ohjelmisto NI-MAX, jolla voitiin kokeilla sisään ja ulostulojen toimivuus. Ulostulojen toimivuus oli helppo kokeilla yleissähkömittarilla. Kun portti on kytkettynä, mittari piippaa. Sisääntulopuolta testastattiin kytkemällä porttiin plus- ja miinusnapojen väliin 9voltin patterin ja tutkailemalla NI-MAXista, onko portti päällä vai ei.



Kuvio 7. NI-6525 kytkennät

4.1.4 Donbot ja Robot Framework

Donbot toimii siis yhdyskäytävänä DUT:in ja Robot frameworkin välillä ja vaatii configuroinnin toimiakseen. Configuraatio suoritetaan NI-MAX ohjelmasta luetuilla tiedoilla, jotka saadaan ”Hardware” tabin alta. Vakiona ensimmäinen laite on nimetty ”Dev1” ja sieltä löytyy porttien määrittely.

Donbotin konfiguraatio tiedostossa portit määritetään antamalla niille nimi, kuten kuviossa 8 nähdään. Tässä tapauksessa käytämme nimeä Digital1-10, joka seuraa

Aidonin nimeämiskäytäntöjä muissa testiautomaatioissa. Konfiguraatiossa pitää myös nimetä käytettävät com-portit, jotta liikenne saadaan ohjattua oikeaan osoitteeseen. Tarvittavat com-portit ovat PPS ja DUT. NI-USB-6525 toimii kutsumalla Dev1 nimeä.

```
[DI meter]
DIGITAL1 = "Dev1/port1/line0"
invert1 = TRUE
DIGITAL2 = "Dev1/port1/line1"
invert2 = TRUE
DIGITAL3 = "Dev1/port1/line2"
invert3 = TRUE
DIGITAL4 = "Dev1/port1/line3"
invert4 = FALSE
[DO meter]
DIGITAL5 = "Dev1/port0/line0"
invert5 = FALSE
wait5 = 0
DIGITAL6 = "Dev1/port0/line1"
invert6 = FALSE
wait6 = 0
DIGITAL7 = "Dev1/port0/line2"
invert7 = FALSE
wait7 = 0
DIGITAL8 = "Dev1/port0/line4"
invert8 = FALSE
wait8 = 0
DIGITAL9 = "Dev1/port0/line6"
invert9 = FALSE
wait9 = 0
DIGITAL10 = "Dev1/port0/line7"
invert10 = FALSE
wait10 = 0
```

Kuvio 8. Donbotin configuraatio esimerkki

Robot Frameworkia käytetään testien luomiseen ja ajamiseen. Koska mittarin kommunikaatio suoritetaan suoraan kommunikaatio terminaaliin muokatulla USB-kaapelilla, joudutaan luomaan Robotille kirjastot -OBIS ja E2 koodien käsittelyyn.

4.2 Testikirjaston toteutus

Testinkirjaston luominen on rajattu ulos opinnäytetyön raameista, koska se vie todella paljon aikaa. Tähän viittaavia töitä löytyy myös Theseuksesta monia, joten keskityimme tässä ainoastaan rautapuoleen ja testipenkin toteutukseen.

Yleisesti voidaan kuitenkin sanoa, että kirjasto tulee noudattamaan karkeasti samaa rakennetta kuin muut testiautomaatiokirjastot Aidonilla. Työni Aidonilla tulee jatkossa keskittymään testikirjaston toteuttamiseen ja ylläpitoon.

Edellisen prototyypin aloittajalta on jäänyt talteen joitain valmiita testejä testikirjastoon, joita voidaan kokeilla heti sikäli, kun tutkitaan että ne ovat yhteensopivia uudemman mittarisukupolven kanssa. Näin ollen rautapuolen kytkentöjä voidaan testata heti. Kävi ilmi, että nämäkin vaativat vielä paljon työtä toimiakseen.

Mitä kannattaa automatisoida?
Tästä käytiin paljon keskusteluja työn tilaajan kanssa, koska työn luonne on manuaalisen testaamisen muuttaminen automaatioksi. On paljon sellaisia testejä, joiden automatisoiminen olisi kohtuuttoman vaikeaa ja veisi paljon resursseja. Silloin tämä ei tietenkään olisi järkevää. Osa testeistä esimerkiksi vaatii mittarin näytöltä lukemista ja verifioimista, joten konenäön virittäminen tähän tarkoitukseen olisi liikaa.

Kustannustehokkuudesta on ollut jonkun verran puhetta ja aihetta tutkittiin. Voitiin laskea, että testauksen hinta on noin 50€/tunti jos se ostetaan alihankkijalta ja mittarin läpitestaukseen menee arviolta noin 70 tuntia/mittari, joten yhden mittarin testaus maksaisi alihankkijalta ostettuna palveluna noin 3500€. Jos tästä 70 tunnista saa resurssit ohjattua automaation seurauksena muihin töihin, on se huomattavan

tehokasta. Tavoitteena on lopulta saada manuaalitestauksesta siirrettyä automaation piiriin noin 80%, joten testaus nopeutuisi noin 56 tuntia.

Manuaaliseen testaamiseen on käytetty Focus Terminal nimistä ohjelmistoa, jolla on ajettu ennalta määritettyjä skriptejä, jotka ajetaan OBIS- ja E2-muodossa mittariin. Sama ohjelmisto lukee ulostulevaa liikennettä ja sitä on aiemmin jouduttu lukemaan manuaalisesti ja se on todella hidasta. Tulevan tekstin määrä on valtavaa ja sieltä tärkeiden asioiden löytäminen ja analysoiminen on työlästä. Tein tätä työtä kesällä 3 kuukautta ja tästä oppineena saatoin aloittaa automatisaation suunnittelun, kun tiesin mitä kannattaa ja mitä ei kannata alkaa automatisoimaan. Paljon testeissä oli semmoista dataa, joita joutui käsin laskemaan ja näiden automatisointi nopeuttaa testiä huomattavasti.

Tulevaisuuden töihini kuuluu manuaalisten testien läpikäyminen ja kartoittaminen testien kattavuudesta. Näistä voidaan tehdä jonkinlainen lista ja määrittää prioriteetit tehtäville automaatiotesteille. Tavoite olisi saavuttaa ainakin 80% kattavuus automaatiotesteille.

5 Jatkokehitys

5.1 Tampering-kytkin

Tämä jätettiin tulevaisuuden kehityskohteiksi, koska toteutus olisi ollut tarpeettoman hankalaa. Edellisissä mittarimalleissa kytkin oli helpommin ohjattava 2-vaihekytkin, mutta uudessa Falcon mallissa tämä on erilainen, turvallisempi ja toimintavarmempi, mutta tämä tuo mukanaan ongelmia sen testiautomaatioon integroimiseen. Nyt resurssit eivät tähän riittäneet ja aikataulu oli muutenkin liian tiukka. Tampering-kytkintä on tarkoitus tulevaisuudessa ohjata myös NI-6525 USB ohjaimella ja ehkä jonkinlaisella relekombinaatiolla.

5.2 ADE-mittauspiiri

ADE-mittauspiirin oikosulkemista tehdään vain yhdessä manuaalitestitapauksessa ja sen kanssa on ollut ongelmia jopa manuaalisestikin. Mittauspiiri on niin herkkä, että kaikenlainen ylimääräisten johdinten juottaminen siihen aiheuttaa koko mittarin menon virhetilaan. Tätä ongelmaa on yritetty ratkaista aiemminkin tuloksetta.

5.3 Testikirjasto

Manuaalisten testitapausten muuttaminen automaattiseksi vaatii valtavan määrän työtunteja, eikä sitä minun tai toimeksiantajan mielestä ollut järkevää sisällyttää tämän opinnäytetyön raameihin. Robot Frameworkista ja sen kirjastojen käytöstä on jo tehty monia opinnäytetöitä, joten päätimme työn tilaajan kanssa jättää sen pois työn tavoitteista. Päättelin, että lisäämällä testiautomaatiota sulautettuun järjestelmään ja ottamalla käyttöön digitaalista ohjausta ja signaalin lukemista, olisi se tehokkaampaa resurssien käyttöä. Tulen joka tapauksessa jatkamaan aiheen parissa, rakentaen testikirjastoa ja kasvattamalla testikattavuutta työni puitteissa.

6 Yhteenveto ja arviointi

Loppujen lopuksi tästä saatiin toimiva paketti, jonka jatkokehittämistä voidaan jatkaa luontevasti suunnittelemalla ja toteuttamalla testejä testikirjastoon kattavuuden lisäämiseksi. Työn suunnittelu ja toteutus oli erittäin opettavaista niin tiedon hakemisen, kuin itse tekemisen kannaltakin. Jouduin opettelemaan paljon uusia asioita sähkötekniikan ja elektroniikan saralta, jotta tavoitteeseen päästäisiin. Sulautettujen ohjelmistojen ja rautapuolen yhteensovittaminen olikin antoisinta koko projektissa ja siitä sai suurimpia onnistumisen tunteita, kun tekemisensä näkee konkreettisesti, esimerkiksi fyysisen napin toimintona tietokoneelta ohjattuna.

Koko ympäristö on suunniteltu sitä silmällä pitäen, että mittarin vaihto ympäristöön on mahdollisimman helppoa ja nopeaa. Siitä syystä mittariin juotetaan johdot mittarin sisälle kaikkiin ohjauksiin. Tavoitteena on, että mittari tarvitsee avata vain kerran ja sen jälkeen voi vain kytkeä liittimet mittariin, kytkeä virtalähteen kaapelit ja aloittaa testaamisen. Uusia mittarimalleja on tulevaisuudessa tulossa arviolta 2-4, joten ajansäästö tulee olemaan huomattava testaamisen automatisoinnilla.

Suurimmat ongelmat koko projektissa olivat sähkötekniikan ja elektroniikan ymmärtäminen ja toteuttaminen. Pohjatietämys tältä alalta oli todella vähäistä ja sen toimintaan saattaminen oli usein yrityksen ja erehdyksen kautta saavutettua. Vastusten ohmien laskeminen tuli ainakin tutuksi rakentaessa kuormareleohjauksia.

Kuviossa 9 näkyy kaksi erillistä mittaria, jotka on laitettu valmiiksi testaamista varten. Toinen on Direct Current ja toinen Current Transformer mallia. Testit tulevat olemaan mittarimalleilla hieman erilaiset, joten tähän ongelmaan joudutaan pohtimaan Robot Frameworkin puolella ratkaisuja, kuinka tunnistaa testattava mittarimalli ja sitä myöten mallille soveltuvat testit. Testien olisi myös oltava taaksepäin yhteensopivia edellisen mittarisukupolven kanssa, joten haasteita riittää jatkossakin. Pohja on kumminkin vakaa ja testejä on hyvä lähteä rakentamaan, kun on luottoa, että rautapuoli toimii vakaasti.

Onnistuin mielestäni hyvin projektissa ja oppimisen määrä oli valtava. Tiedonhakutaidot ja projektin suunnittelun ja läpiviennin kehitys kaikkine haasteineen antoi itseluottamusta tulevaisuuden haasteisiin työelämässä. Vaikka kokemusta edellisistä työpaikoista ja harrastuksista on monelta osa-alueelta, on elektroniikan ja ohjelmistojen yhdistäminen vielä vierasta.

Kuvio 9. Testiseinä

Kun projekti saatiin valmiiksi, ajettiin sille pienimuotoista prototestaamista, jotta voitiin varmistua ominaisuuksien toiminnasta. Proton testaamiseksi luotiin karkeat Robot Framework testit, joilla testattiin kommunikaation mittarin ja Robotin välillä. Jokaiselle fyysiselle ominaisuudelle tehtiin toimintatesti, jolla varmistuttiin toimivuus. Työn tilaaja oli todella tyytyväinen tuloksiin ja testeissä kaikki tuntui toimivan odotetusti. Jotain pientä parannettavaa totta kai aina jää, mutta niistä selvittiin pienillä muokkauksilla ja korjauksilla.

Lähteet

Luentajärjestelmä. N.d. Aidonin esite. Viitattu 13.1.2021. https://www.aidon.com/fi/asiakasratkaisut/#luentajarjestelma.

PTS3.3C Manual. 2017. PTS käyttöohje. Viitattu 10.1.2021. https://www.mte.ch/data/files/PTS%203.3%20C%20Overview%20English_R03%20(09.2017).pdf.

Risk Based Testing: Approach, Matrix, Process & Examples. N.d. Guru99 sivusto. Viitattu 10.2.2021. https://www.guru99.com/risk-based-testing.html.

Robot Framework. 2021. Robot framework kotisivu. Viitattu 10.1.2021. https://robotframework.org/rpa/.

Soveltavasta tutkimuksesta. N.d. Opinnäytetyön ohjaajan käsikirja, Jyväskylän ammattikorkeakoulun sivut. Viitattu 24.1.2021. https://oppimateriaalit.jamk.fi/yamk-kasikirja/soveltavat-tutkimusmenetelmat/.

Tämä on Aidon. N.d. Aidonin verkko sivusto. Viitattu 24.1.2021. https://www.aidon.com/fi/aidon/.

USB-6525. N.d. National instruments tuoteseloste. Viitattu 10.1.2021. https://www.ni.com/fi-fi/support/model.usb-6525.html.

What is Requirements based Testing?. N.d. Tutorialspoint sivusto. Viitattu 10.2.2021. https://www.tutorialspoint.com/software_testing_dictionary/requirements_based_testing.htm.