



TAMPEREEN
AMMATTIKORKEAKOULU

SILTAMITTAUS

Tommi Perälä

Walteri Penttilä

Opinnäytetyö

Joulukuu 2016

Tietotekniikan koulutusohjelma
Elektroniikka ja sulautetut järjestelmät
Tietoliikennetekniikka ja tietoverkot



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Elektroniikka ja sulautetut järjestelmät
Tietoliikennetekniikka ja tietoverkot

TOMMI PERÄLÄ & WALTTERI PENTTILÄ:
Siltamittaus

Opinnäytetyö 38 sivua, joista liitteitä 3 sivua
Joulukuu 2016

Tämän opinnäytetyön tarkoituksena oli saada toteutettua toimiva sekä luotettava järjestelmä Tampereella Vuoreksen alueella sijaitsevan Tervassillan mittausdatan keruuta varten. Mittausdatan keruun tarkoituksena on mitata sillan kuormitusta käyttäen venymäliuska- sekä siirtymä-antureita. Datan keruu tapahtuu siihen tarkoitettulla dataloggerilla paikallisella keskusyksikölle, jossa data käsitellään haluttuun muotoon ja lähetetään mobiiliverkkoa käyttäen etäpalvelimelle. Järjestelmän automatisointiin vaadittiin omien ohjelmien kirjoittamista Powershell- sekä komentoriviympäristöjä apuna käyttäen.

Tehtävänä oli suunnitella ja toteuttaa valmiiksi hankittuja laitteita käyttäen tämä datan mittaus- ja siirtojärjestelmä sekä pyrkiä tekemään siitä mahdollisimman automatisoitu. Myös etätyöpöytäyhteys keskusyksikölle täytyi mahdollistaa, mahdollisten ongelmatilanteiden etäratkaisun sekä datan keruun asetusten muuttamiseen.

Järjestelmän testaus suoritettiin jokainen järjestelmän komponentti sekä ohjelmisto kerrallaan sekä lisäksi koko järjestelmä testattiin huolellisesti virheiden varalta. Lopuksi tehtiin johtopäätökset testien onnistumisesta sekä pohdittiin mittauksen erilaisia mahdollisia toteutustapoja ja omaa näkökulmaa projektin toteuttamisesta.

ABSTRACT

Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Electronics and embedded systems
Telecommunication and Networks

TOMMI PERÄLÄ & WALTTERI PENTTILÄ:
Bridge load measurement

Bachelor's thesis 38 pages, appendices 3 pages
December 2016

The purpose of this thesis was to design as well as implement functional and reliable system for the collection of measurement data from Tervassilta which is located in Vuores, Tampere. The function of collecting measurement data was to measure bridge load by using strain gauge and displacement sensors. Data collection takes place in the data logger which is connected to the local computer, where data will be handled to desired format and sent to remote server using mobile network. Automation of the system required to write your own programs by using Powershell and command-line environments.

The task was to engineer this data measurement and transmission system plus strive to make it as automated as possible using pre-acquired measuring devices. It was also important to make a remote desktop connection possible for the central unit in case of fixing problems as well as for changing data measurement settings.

System testing was performed on each component of the system and software individually. After individual component testing whole system was thoroughly tested for errors. Finally, the conclusion of the tests success was drawn, as well as consideration about variety of possible methods and own perspective on the implementation of the project was made.

Key words: strain gauge, tervassilta, powershell, data transfer

SISÄLLYS

1	JOHDANTO.....	6
2	KOHTEEN ESITTELY	7
	2.1 Tervassilta	7
	2.2 Isokuusen alue.....	8
3	LAITTEISTON KUVAUS.....	9
	3.1 Keskusyksikkö	9
	3.2 PCD-400A anturisoitin	10
	3.2.1 Mittaustarkkuus ja virhetarkastelu	11
	3.2.2 CST-menetelmä.....	11
	3.3 E5776-mobiililaajakaistareitin.....	12
	3.4 Venymäliuska	13
	3.4.1 Venymäliuskan asennus	14
	3.4.2 Wheatstonen silta	14
4	JÄRJESTELMÄN KUVAUS	16
	4.1 Järjestelmän vaatimukset	16
	4.2 Järjestelmän toiminta	17
5	OHJELMISTON KUVAUS	18
	5.1 DCS-100A	18
	5.2 WinSCP	19
	5.3 PowerShell.....	21
	5.4 Task Scheduler.....	22
	5.5 TeamViewer.....	23
6	OHJELMAT	25
	6.1 Synkronointiskripti	25
	6.2 Synkronoinnin käynnistysohjelma.....	27
7	JÄRJESTELMÄN TESTAUS	28
	7.1 Laajakaistareitittimen testaus.....	28
	7.2 Dataloggerin testaus.....	30
8	OHJELMIEN TESTAUS	31
	8.1 Synkronointiskriptin testaus	31
	8.2 Synkronoinnin aloitusohjelman testaus	31
9	POHDINTA.....	33
	LÄHTEET.....	34
	LIITTEET	36
	Liite 1. Valmistunut silta	36
	Liite 2. Synkronointiskripti	37

LYHENTEET JA TERMIT

WLAN	Wireless local area network
USB	Universal serial bus
TEKES	Innovaatorahoituskeskus
SFTP	Secure file transfer –protocol
FS	Full scale, tarkkuus koko näytteenottoalueella
SSH	Secure shell
RSA	Julkisen avaimen salausalgoritmi
AES	Advanced Encryption Standard
NAT	Network Address Translation
OSI	Open System Interconnection
ASU	Arbitrary Strength Unit
LTE	Long-Term Evolution
IP40	IP-suojausluokitus
ISE	Integrated Scripting Environment
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
HTTP	Hypertext Transfer Protocol

1 JOHDANTO

Tervassilta on rakennettu yhdistämään kaksi Vuoreksen alueella Tampereella sijaitsevaa kaupunginosaa. Silta on liikkumisen kannalta tärkeä osa kaupunginalueen puista infrastruktuuria. Työn tarkoituksena on saada kerättyä sillasta dataa ja saada siirrettyä se luotettavasti paikalliselta keskusyksiköltä etäpalvelimelle.

Aluksi opinnäytetyössä käydään lävitse käytössä olleet laitteet, jokaisen laitteen käyttötarkoitus, sekä tutustutaan tarkemmin käytössä oleviin tekniikoihin. Kappaleessa analysoidaan mahdolliset mittausvirheet ja tutustutaan venymäliuskan oikeaoppiseen asennukseen. Järjestelmän kuvauksessa tutustutaan järjestelmän toimintaan yleisesti, tarkastellaan järjestelmän vaatimuksia vallitsevien sääolosuhteiden kannalta, sekä tarkastellaan vaiheittain datan eteneminen paikalliselta keskusyksiköltä TAMK:n palvelimelle.

Seuraavassa kappaleessa käsitellään käytössä olleet ohjelmistot sekä miksi juuri näitä ohjelmistoja käytettiin tässä työssä. Kappaleessa tutustutaan myös ohjelmistojen ominaisuuksiin, hyviin ja huonoihin puoliin, sekä tutkitaan eri ohjelmistojen tietoturva.

Työssä tutustutaan myös projektia varten luotuihin ohjelmiin sekä niiden rakenteisiin ja toimintaan.

Järjestelmän testaus – osiossa tarkastellaan järjestelmän testausprosessia yksittäisten komponenttien ja koko järjestelmän osalta, kun taas ohjelmien testauksessa tutkittiin omien ohjelmien testausprosessia samanaikaisesti koko järjestelmän pidempiaikaisen testin yhteydessä

Lopuksi pohdittiin erilaisia toteutusmenetelmiä, joilla projekti olisi voitu toteuttaa toisin sekä tarkasteltiin opinnäytetyön aikana kohdattuja ongelmia ja niiden ratkaisuja.

2 KOHTEEN ESITTELY

Projektissa kohteena olleen puusillan tarkoituksena on yhdistää Vuoreksen alueella sijaitsevan Isokuusen alueen keskiset ja läntiset osat. Silta on Tampereen kaupungin rakennuttama ja Suunnittelukide Oy:n suunnittelema.

Tampereen ammattikorkeakoulussa on jo toteutettu ja tullaan toteuttamaan opinnäytetöitä ja tutkimuksia sillan rakenteesta ja siihen liittyvistä lujuusmittauksista. Tulevaisuudessa tutkitaan myös materiaalisissa tapahtuvia muutoksia sekä sillan elinkaaren aikana aiheutuneita kustannuksia.

(Tampereen kauppakamarilehti, Puukaupunki nousee Vuoreksen Isokuuseen, 2015)

2.1 Tervassilta

Tervassilta on Vuoreksen kaupunginosassa Isokuusen alueella Tampereella sijaitseva puinen silta, kokonaispituudeltaan silta on 41,82 m ja leveydeltään 11,23 m. Tervassillan rakennusprojekti on osa TEKES:n INKA-innovatiiviset kaupungit-ohjelmaa. Sillan ylittää 2-kaistainen ajorata sekä kevyen liikenteen väylä, jonka kapasiteetti on suunniteltu riittävän suureksi kantamaan jopa usean kymmenen tonnin painoisia kuormia. (Kuva 1) Venymä- ja siirtymä-antureiden lisäksi kohteesta mitataan myös lämpötilaa, kastepistettä sekä suhteellista ilmankosteutta.



KUVA 1. Tervassilta (Suunnittelukide Oy / kuva Vianova Finland Oy)

2.2 Isokuusen alue

Tulevaisuuden visio Isokuusen alueesta on rakentaa noin 4000 asukkaan suuruinen asutuskeskus. Arkkitehtuuri alueella on ekologinen sekä hiilineutraali.

Tampereen kaupungin sekä tutkimus- ja kehitysyhtiö Finnish Wood Researchin Oy:n tavoitteena on suunnitella ja toteuttaa suomen suurin nykyaikainen ekotehokas puukaupunki, joka sisältää tarpeelliset perusinfrastruktuurit puisine liikerakennuksineen.

Toteutuessaan Isokuusen alueen korkeat puukerrostalot, puiset liikerakennukset sekä puinen pientalorakentaminen muodostavat yhtenäisen ja eheän kokonaisuuden, jonka suuri, puinen silta täydentää (Kuva 2.)

(Tampereen kaupunki, Isokuusen alue, 2015.)



KUVA 2. Isokuusen alueen suunniteltu asuinalue (Vianova Finland Oy)

3 LAITTEISTON KUVAUS

Laitteiston tarkoituksena on saada suoritettua sillan kuormitusmittaus onnistuneesti sekä tarkasti. Valintakriteereinä laitteistolle oli mittauksen luotettavuus vaihtelevissa sääolosuhteissa, mittaolosuhteet saattavat vuoden aikana vaihdella lämpötilan osalta jopa -30 °C ja 30 °C asteen välillä. Tässä kappaleessa käydään lävitse työssä käytettyjä laitteita

3.1 Keskusyksikkö

Mittausjärjestelmän keskusyksikkönä toimii Advantechin UNO-2174G, joka on teollisuuskäyttöön suunniteltu IP40-suojatulla koteloinnilla varustettu sulautettuihin järjestelmiin tarkoitettu tietokone. IP40-luokitus tarkoittaa, että laite on suojattu halkaisijaltaan yli 1mm kokoisia kappaleita vastaan.



KUVA 3. UNO-2174G–keskusyksikkö (Advantech)

Keskusyksikkö on suunniteltu kestäämään vaihtelevia käyttölämpötiloja, tukemaan useampia käyttöjärjestelmiä sekä kuluttamaan mahdollisimman vähän tehoa. Tämän vuoksi se soveltuu hyvin mittauksen vaatimiin olosuhteisiin. Pienen kokonsa ja passiivisen jäähdyttimen ansiosta laitteen voi myös asentaa helposti pieniin ja suljettuihin tiloihin.

Käyttöjärjestelmänä keskusyksikössä toimii Windows 7, 32-bittinen versio tutun käyttöympäristön sekä hyvän ohjelmistojen yhteensopivuuden vuoksi. Käyttöjärjestelmänä oli yrityksille ja tehokäyttäjille tarkoitettu professional-versio.

3.2 PCD-400A anturisoitin

Antureiden ja keskusyksikön välisenä sovittimena käytettiin Kyowa:n valmistamaa PCD-400A-anturisoitinta (Kuva 4.) Anturisoitin on kytketty keskusyksikköön käyttämällä USB-johtoa.



KUVA 4. PCD-400A-anturisoitin (KYOWA)

Useampaa anturisoitinta käytettäessä anturisoittimet pinotaan päällekkäin, jolloin ne kytkeytyvät yhteen, mikä mahdollistaa useamman anturin käyttämisen. Rajoitteena kuitenkin on kytkeä enintään neljä anturisoitinta yhteen, jolloin käytössä on jopa 16 kanaavaa antureille.

Sovittimella on mahdollista käyttää 2-johdon sekä 3-johdon mittauksissa 120 Ω :n antureita, sekä half bridge-mittauksessa kahta samanlaista anturia, arvoltaan 120 Ω –1000 Ω .

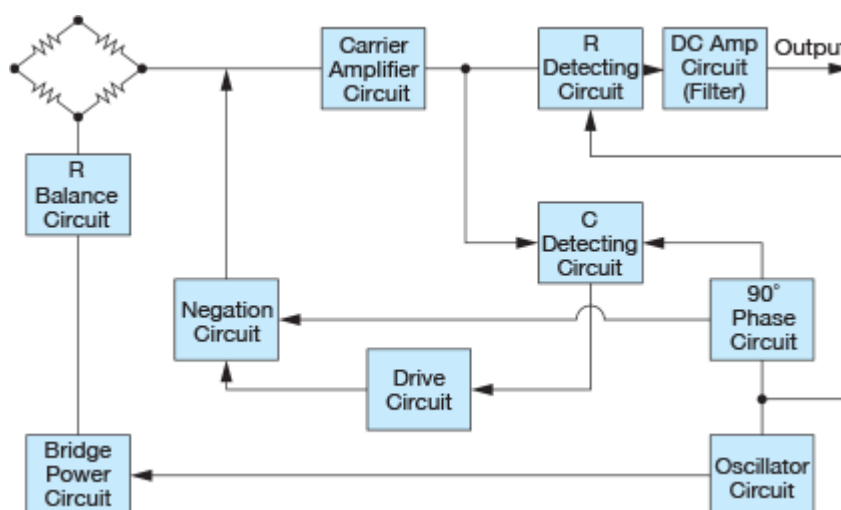
3.2.1 Mittaustarkkuus ja virhetarkastelu

Dataloggeri muuntaa mitatun datan käyttäen 24-bittistä A/D-muunninta, sekä suodattaa datan toisen asteen Butterworth-alipäästösuotimella. Laitteen näytteenottotaajuus on suurimmillaan 10kHz, maksimimittaustarkkuutena voidaan käyttää portaittain 200, 500, 1000, 2000, 5000, 10000, 20000 $\mu\text{m/m}$ arvoja. Mittaustarkkuus koko seitsenportaisella mittausalueella on $\pm 0.5\%$ FS, jonka vuoksi suurin mahdollinen mittausrvirhe käytettäessä 20000 $\mu\text{m/m}$ mittausaluetta on ± 100 $\mu\text{m/m}$.

Epälineaarisuuden aiheuttama mahdollinen virhe on $\pm 0.1\%$ FS, joka aiheuttaa suurimmillaan ± 20 $\mu\text{m/m}$ lisävirhettä mittaustulokseen. Mittaustuloksen maksimikokonaisvirhe näin ollen on ± 120 $\mu\text{m/m}$.

3.2.2 CST-menetelmä

Mittaustulosten suodattamiseen on varauduttu rakentamalla anturisovittimeen CST-menetelmää käyttävä balansoimaton kapasitanssia mittaauksen aikana suodattava järjestelmä. (Kuva 5) Menetelmässä kapasitanssin vaihe invertoidaan ja invertoitua signaalia käytetään kumoamaan balansoimattoman kapasitanssin vaikutus.



KUVA 5. CST-menetelmän lohkokaavio (KYOWA)

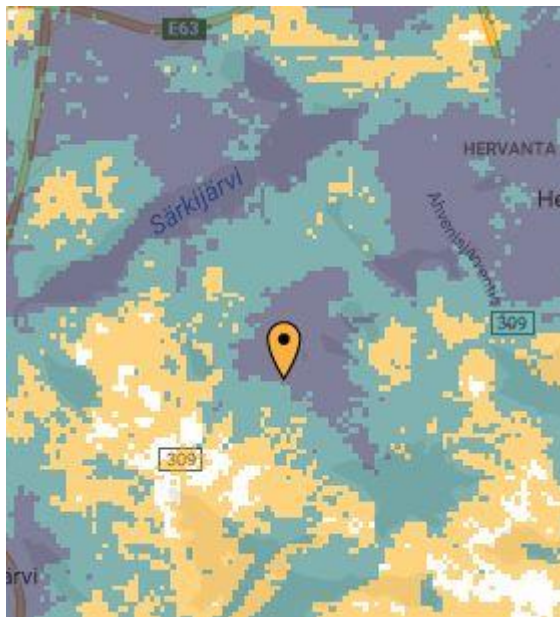
3.3 E5776-mobiililaajakaistareititin

Internet-yhteyden muodostamista varten oli hankittu langaton mobiililaajakaistareititin, sillä kiinteän laajakaistayhteyden toteuttaminen kohteeseen olisi ollut erittäin haastavaa. Yhteys reitittimeen oli mahdollista langallisesti USB:n avulla tai langattomasti WLAN-yhteyttä käyttäen. Mittauksessa yhteys muodostettiin USB:n avulla, koska sitä pidettiin luotettavampana tapana muodostaa yhteys.



KUVA 6. HUAWEI-mobiililaajakaistareititin (SONERA)

Yhteystyypeistä reititin tukee 2G-, 3G- ja 4G-yhteyksiä. Vuoreksen alue sijaitsee 4G-verkon peittoalueella lähes kokonaan, ja suurin osa vuoreksesta sijaitsee erinomaisen yhteyden alueella (kuvassa violetti)

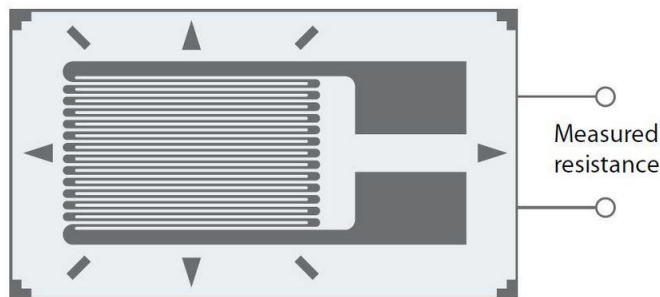


KUVA 7. Vuoreksen alueen 4G-verkon peittoaluekartta (SONERA)

3.4 Venymäliuska

Venymäliuska-anturilla mitataan pääasiassa kappaleen yhdensuuntaista venymistä. Venymäliuskoja käytetään teollisuudessa mm. paineantureissa sekä erilaisissa vaakasoveluksissa. Venymäliuskan toiminta perustuu sen venymisen aiheuttamaan muutokseen vastuslangan pituudessa, liuskaa venytettäessä vastuslangan poikkipinta-ala pienenee, jonka seurauksena johdinlangan pituus kasvaa sekä vastusarvo kohoaa.

Venyttäminen toimii kuitenkin vain pituussuuntaisesti.



KUVA 8. Venymäliuska-anturi (United Electronic Industries)

Koska venymäliuskalla mitattavat venymiset saattavat olla hyvinkin pieniä, täytyy muuttosta vahvistaa jonkinlaisella vahvistimella. Perinteisesti vahvistukseen on käytetty Wheatstonen siltakytkentää.

Häiriön poistamiseksi käytetään yleisesti kolmijohdinkytkentää, minkä avulla voidaan kompensoida mittaustuloksesta anturin kaapeloinnin aiheuttama resistanssi.

Siltaan asennettuna antureiden kaapeloinnit ovat useasti monien kymmenien metrien pituisia, mikä aiheuttaa kytkentään häiriötä pitkien kaapeleiden aiheuttaman resistanssin muodossa. Testimittauksissa käytettiin kyseistä kolmijohdinkytkentää, sillä jo käytössä olleissa noin 30 cm mittaisissa johtimissa oli liikaa mittausta häiritsevää resistanssia.

Varsinaisessa siltamittauksessa vaihtelevien lämpötilojen vuoksi kolmijohdinkytkentä ei anna riittävän luotettavaa arvoa, sillä tämä kytkentä ei ota huomioon lämpölaajenemisen vaikutusta mittaustulokseen. Tämän vuoksi käytetään ns. ”dummy”-kytkentää, joka toisen eri suuntaisesti asennetun venymäliuskan avulla kompensoi tuloksista myös lämpölaajenemisen aiheuttaman virheen.

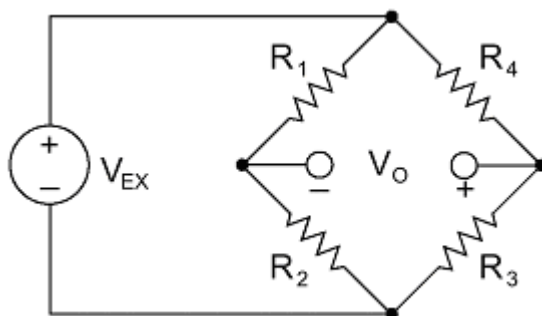
3.4.1 Venymäliuskan asennus

Venymäliuskan asennuksessa pitää ottaa huomioon monia asioita, ehkä tärkein näistä on liuskan kiinnitys mitattavaan kohteeseen. Liuskaa asennettaessa täytyy pohjatyöt tehdä huolellisesti, asennuspinta täytyy hioa erittäin hienolla hiekkapaperilla sekä puhdistaa pinnasta mahdolliset epäpuhtaudet tehokkaalla liuottimella. Tämän jälkeen suoritetaan välittömästi itse liimaus, jottei kohteeseen ehdi muodostua minkäänlaista korroosiota tai hapettumista.

Liimauksessa käytetään hyvin useasti siihen tarkoitettua erikoisliimaa, jotta kiinnitys pintaan olisi optimaalinen ja kestävä. Yleensä liimaus tapahtuu epoksipohjaisella liimalla, jonka kovettamiseen tarvitaan korkeaa lämpötilaa.

Lyhyissä, alle muutaman viikon kestävässä mittauksissa kuitenkin on mahdollista käyttää tavallista, syanoakryliin pohjautuvaa pikaliimaa.

3.4.2 Wheatstonen silta



KUVA 9. Wheatstonen siltakytkentä (NATIONAL INSTRUMENTS)

Wheatstonen siltakytkentä perustuu kytkentään, jossa käytetään kolmea samanarvoista vastusta $R_2=R_3=R_4$ sekä mitattavaa resistanssin muutokseen pohjautuvaa anturia, kuten esimerkiksi tässä tapauksessa venymäliuskaa. Siltakytkentään jännitteeksi V_{EX} kytketään vakiojännite, jolloin venymäliuskan vastusarvon muuttuessa voidaan mitata muutos jännitteestä V_0 .

Tarkempia mittaustuloksia haluttaessa pitää kytkentää kuitenkin muuttaa hieman. Kytkeä muutetaan vastuksen R_2 tilalle asennettavalla säädettävällä vastuksella, potentiometrillä, sekä jännitteen V_0 mittaamiseen tarkoitettulla herkällä galvanometrillä. Galvanometrin herkkyyden ansiosta kytkennällä on mahdollista saada mitattua hyvin pieniä muutoksia mitattavasta resistanssista. Mitattavan kohteen ollessa kytkettynä säädetään potentiometrillä galvanometrin näyttämä virta nolaksi, jolloin kohteeseen tuleva muutos ilmenee virran muutoksena. Mittaustuloksen tarkkuus riippuu täysin galvanometrin tarkkuudesta.

4 JÄRJESTELMÄN KUVAUS

Siltamittaukseen käytetään venymäliuska-antureiden lisäksi myös siirtymä-antureita, joiden avulla saadaan muodostettua kokonaiskuva sillan kuormitustilanteesta. Järjestelmän täytyy olla luotettava, sekä nopeisiin muutoksiin reagoiva, jotta sillan mahdolliset värinät ja muut nopeat ilmiöt havaittaisiin. Ongelmaksi tästä muodostuu hyvinkin nopeasti tallennuskapasiteetin rajallisuus, sillä jo muutaman tunnin mittaus mittaamalla vain yhtä kanavaa 50 hertsin taajuudella käytti tallennustilaa usean kymmenen megatavun verran.

Lopullisessa mittauksessa käytetään jopa kahdeksaa eri kanavaa datan keruuseen ja mikäli jokaisella kanavalla käytettäisiin samaa 50 Hz taajuutta, tallennustila loppuisi jo muutaman viikon kuluessa. Tämän vuoksi mittauksen optimointi on tärkeää, esimerkiksi siirtymä-antureiden arvoa mitattaessa ei tarvita samaa herkkyyttä kuin venymäliuskojen kanssa.

Hyvin optimoitu järjestelmä myös vähentää tarvittavaa datansiirtoa, mikä vähentää osaltaan verkon kuormitusta sekä kustannuksia pienemmän kovalevykapasiteetin tarpeen osalta. Datansiirtoa käytetään tiedostojen lähettämiseen etätietokoneelta TAMK:n palvelimelle. Tämä pienentää entisestään etätietokoneen kovalevykapasiteetin tarvetta ja data on helposti saatavilla TAMK:n palvelimelta tulosten analysointia varten.

4.1 Järjestelmän vaatimukset

Laitteet asennetaan ulkotiloissa sijaitsevaan kytkentäkaappiin, joten laitteiston täytyy kestää vaihtelevia sääolosuhteita. Tämän vuoksi laitteistoksi on valittu vain kestäviä ammattilaistason laitteita. Järjestelmän tarkoituksena on saada luotettavasti mitattua sillan kuormitustekijöitä erilaisissa kuormitustilanteissa, esimerkiksi kuorma-auton ylittäessä siltaa ja lämpötilavaihteluiden aiheuttamat muutokset.

4.2 Järjestelmän toiminta

Venymäliuskoilta saadut mittausarvot välitetään anturisoittimelle, joka muuntaa signaalin digitaaliseksi viestiksi, joka on luettavissa tietokoneella olevalla Kyowa:n ohjelmistolla. Ohjelmisto asetettiin tallentamaan mittaustulos tietokoneen kovalevyllä sopivin väliajoin, jotta tiedostojen määrä ja koko pysyisivät kohtuullisina. Kovalevyllä tallennetut tiedostot lähetetään edelleen SFTP-yhteyden avulla kovalevyllä TAMK:n palvelimelle, josta data on helposti saatavilla millä tahansa laitteella.

5 OHJELMISTON KUVAUS

Muutaman valmiin ohjelmiston käyttö on projektissa pakollista datan lukemiseksi data-loggerilta, tiedonsiirron suorittamiseksi, sekä etähallinnan mahdollistamiseksi.

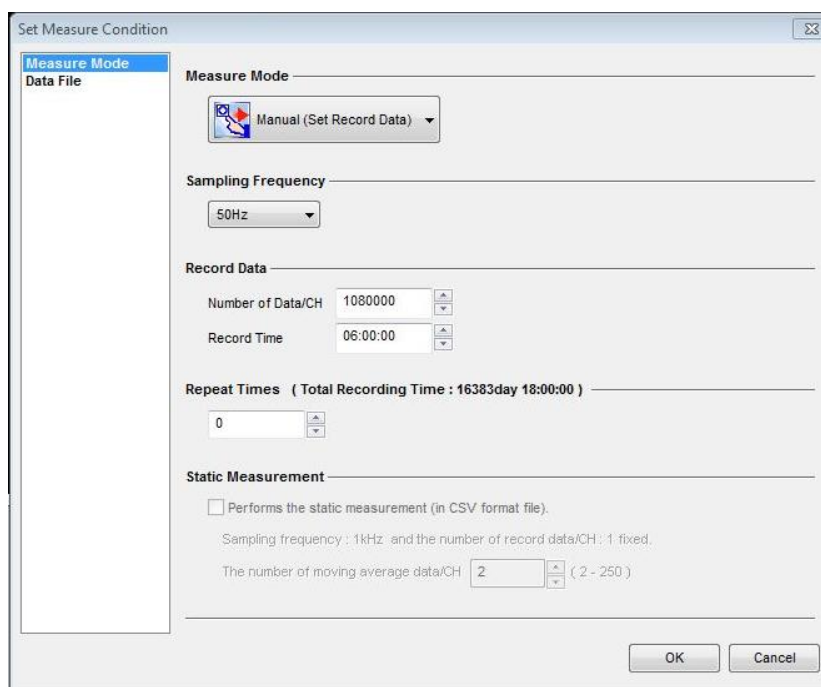
Kappaleessa käydään lävitse kaikki valmiit ohjelmistot, joita projekti vaatii toimiakseen halutulla tavalla. Kappaleessa kerrotaan myös, kuinka ohjelmistojen konfigurointi tapahtuu ja mitä ominaisuuksia ohjelmat sisältävät sekä mitkä ominaisuuksista ovat projektissa tärkeitä.

5.1 DCS-100A-ohjelmisto

Kyowa:n DCS-100A-ohjelmisto on datan graafiseen esitykseen suunniteltu ympäristö.

Ohjelmisto on tarkoitettu käytettäväksi Kyowa:n omien anturisoittimien kanssa, kuten tässä tapauksessa PCD-400A.

Ohjelmiston asennuksen jälkeen, sitä ensi kertaa käynnistettäessä, määritellään käytössä oleva anturisoitin. Kun oikea anturisoitin on määritelty, ohjelmisto on täysin käyttövalmis. Ohjelmiston tärkeimmät ominaisuudet tässä tapauksessa olivat näytteenottoaajuus, näytteenoton kesto sekä toistokerrat (KUVA 10).



KUVA 10. DCS-100A mittausasetukset

Ohjelmisto mahdollistaa datan esityksen monella erilaisella visuaalisella tavalla, esimerkiksi käytössä olleella venymä-aika-akselilla, jolla saadaan visualisoitua data ajan suhteen. Muita vaihtoehtoja olisivat pylväsdiaagrammi, numeraalinen esitystapa, viisarimitari, sekä monta muuta vaihtoehtoa joita ei kuitenkaan tässä tapauksessa tarvittu.

Oman visuaalisen esitystavan lisäksi voidaan data viedä excel-tiedostoksi, minkä avulla voidaan tehdä syvällisempää analyysia saadusta datasta.

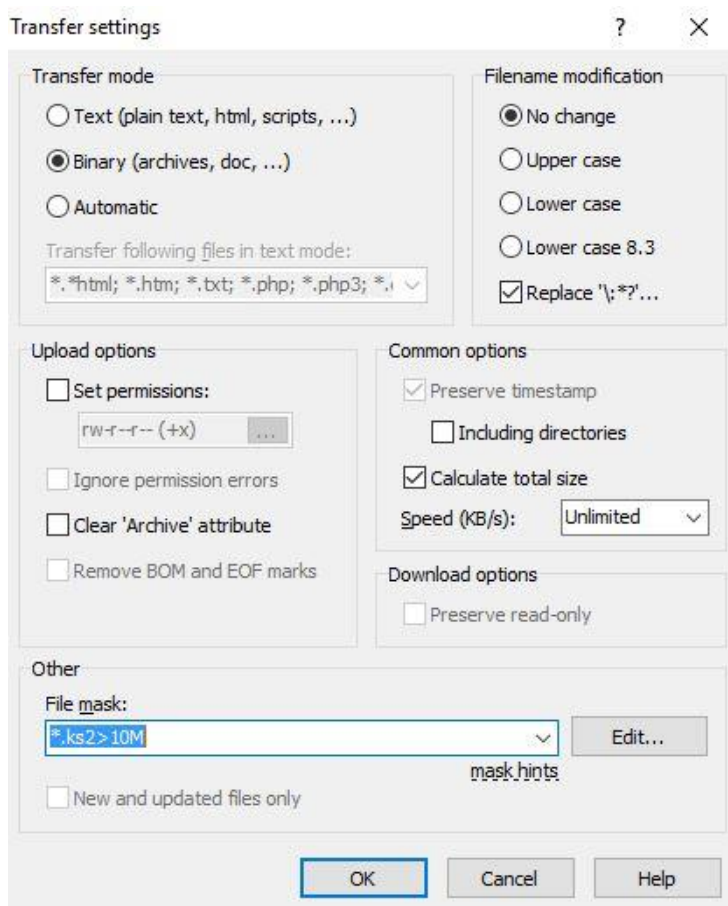
5.2 WinSCP-ohjelmisto

WinSCP on FTP- ja SFTP-protokollia SSH:n välityksellä käyttävä ohjelmisto, joka mahdollistaa tiedonsiirron kahden tietokoneen, paikallisen ja etätietokoneen välillä. WinSCP sisältää helposti lähestyttävän käyttöliittymän, jonka avulla yhteys on sujuva muodostaa. Graafisen käyttöliittymän lisäksi WinSCP mahdollistaa omien skriptien kirjoittamisen ja käyttämisen erillisten skripti/konsolikäyttöliittymien kautta.

Vaihtoehtoisista protokollista käytössä oli SFTP-protokolla, koska se mahdollistaa paremmin suojatun tiedonsiirron laitteiden välillä ja TAMK:n palvelimella on mahdollista hyödyntää tätä tiedonsiirtotapaa.

Yhteyden muodostamisen jälkeen kansiot olivat mahdollista synkronoida käyttäen sovelluksen graafisessa käyttöliittymässä olevaa automaattista synkronointia.

Etuina tässä synkronointitavassa oli sen käyttöönoton helppous, sillä tarvittavat siirtoasetukset saatiin määritellä suodattimen (file mask) avulla (KUVA 11.)



KUVA 11. WinSCP synkronointiasetukset

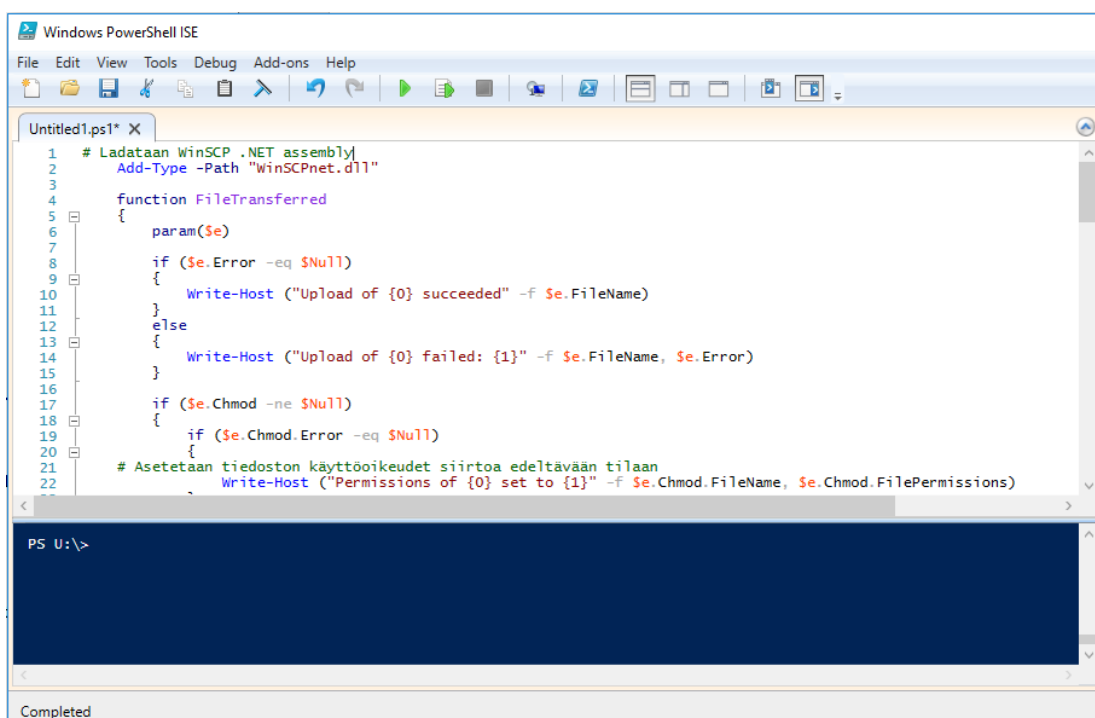
Tämä tapa oli käytössä aluksi, mutta siinä ilmeni muutamia huonoja puolia, jonka vuoksi siirryttiin käyttämään toista synkronointitapaa. Ongelmana tässä synkronointitavassa oli puutteelliset säätömahdollisuudet. Asetuksista puuttui kokonaan synkronoinnin tiheyden säätö, synkronointi oli jatkuvaa, mikä aiheutti turhaa kuormitusta prosessorille, sekä virheitä ohjelmiston yrittäessä siirtää keskeneräisiä tiedostoja.

Tätä kuormitusta pyrittiin vähentämään käyttämällä sopivaa suodatinta, kuten kuvassa 11 olevaa suodatinta, jolla saatiin suodatettua alle 10 megatavun kokoiset tiedostot, sekä muut kuin ks2-päätteiset tiedostot. Kuitenkaan nämä suodattimet eivät riittäneet poistamaan kokonaan virheitä, joten täytyi turvautua muihin menetelmiin.

5.3 PowerShell-ohjelmisto

PowerShell on Microsoftin kehittämä ohjelmointiympäristö, minkä käyttötarkoituksena on selkeyttää järjestelmän hallintaa. Ohjelmistoa käytetään usein tehtävien automatisointiin, sekä muiden käyttöä auttavien skriptien kirjoittamiseen. PowerShell sisältää komentorivipohjaisen komentotulkin, minkä ansiosta skriptien testaaminen kirjoitusvaiheessa on helppoa.

Ohjelmisto valittiin käyttöön, koska kaikki Windows käyttöjärjestelmät Windows 7:stä alkaen sisältävät kyseisen ohjelmiston integroituna käyttöjärjestelmään. Skriptien kirjoittaminen tapahtuu halutulla tekstieditorilla ja valmis skripti tallennetaan .ps1-muotoon, mikä on PowerShellin käyttämä skriptien tiedostomuoto.



```

1 # Ladataan WinSCP .NET assembly
2 Add-Type -Path "WinSCPnet.dll"
3
4 function FileTransferred
5 {
6     param($e)
7
8     if ($e.Error -eq $Null)
9     {
10        Write-Host ("Upload of {0} succeeded" -f $e.FileName)
11    }
12    else
13    {
14        Write-Host ("Upload of {0} failed: {1}" -f $e.FileName, $e.Error)
15    }
16
17    if ($e.Chmod -ne $Null)
18    {
19        if ($e.Chmod.Error -eq $Null)
20        {
21            # Asetetaan tiedoston käyttöoikeudet siirtoa edeltävään tilaan
22            Write-Host ("Permissions of {0} set to {1}" -f $e.Chmod.FileName, $e.Chmod.FilePermissions)
23        }
24    }
25 }
  
```

Completed

KUVA 12. PowerShell ISE

Ohjelmien kirjoittamisen helpottamiseksi on saatavilla WinSCPnet.dll kirjasto, mikä sisältää hyvän dokumentoinnin esimerkkeineen. Ohjelmistoa käytetään työssä DCS-100A-sovelluksella kerätyn datan synkronointiin mittaustietokoneelta TAMK:n palvelimelle. Synkronointi tapahtuu automaattisesti kerran vuorokaudessa Task Scheduleria apuna käyttäen. Synkronointiohjelmaa käsitellään tarkemmin Ohjelmat-kappaleessa.

5.4 Task Scheduler

Windowsin esiasennettuihin ohjelmistoihin kuuluu ohjelmisto nimeltään Task Scheduler, ohjelmistoa käytetään suorittamaan haluttuja ajastettuja toimintoja, kuten esimerkiksi käynnistämään ohjelmia tai skriptejä esimerkiksi tiettyyn kellonaikaan, tietyn ajan välein tai tietokoneen käynnistyksen jälkeen. Työssä Task Scheduleria käytetään synkronointiskriptin suorittamisen ajastamiseen.

Tällä hetkellä synkronointiskriptin suoritus on ajastettu tapahtumaan kerran vuorokaudessa, kolmelta aamuyöstä. Tähän ajankohtaan päädyttiin koska oletus on, että verkkoliikenne TAMK:n palvelimella on tällöin pienimmillään eikä synkronoinnin aiheuttama pieni lisäkuormitus tällöin aiheuta ylimääräistä rasitusta järjestelmään.

Ajankohta ja synkronointitiheys ovat helposti jälkikäteen muokattavissa haluttuihin arvoihin.

5.5 TeamViewer-ohjelmisto

TeamViewer on yleisesti käytetty tietokoneen ja mobiililaitteiden etähallintaan suunniteltu ohjelmisto. Ohjelmisto mahdollistaa esimerkiksi työpöydän jakamisen, videopuhelut, tiedostonsiirron sekä paljon muita etähallinta-ominaisuuksia, tässä työssä ominaisuuksista oli käytössä vain työpöydän jakaminen. Etähallintaa projektissa tarvitaan mitausasetusten muuttamiseen sekä mahdollisten ongelmien diagnosointiin.

Projektiin ohjelmisto valittiin koska käytössä olevassa mobiililaajakaistareitittimessä ei ole operaattorin puolesta tarjolla julkista IP-osoitetta, vaan se on osoitteenmuunnoksen (NAT) takana. Tämän vuoksi yhteys pitää muodostaa käyttämällä kolmannen osapuolen, tässä tapauksessa TeamViewerin, ylläpitämää palvelinta.

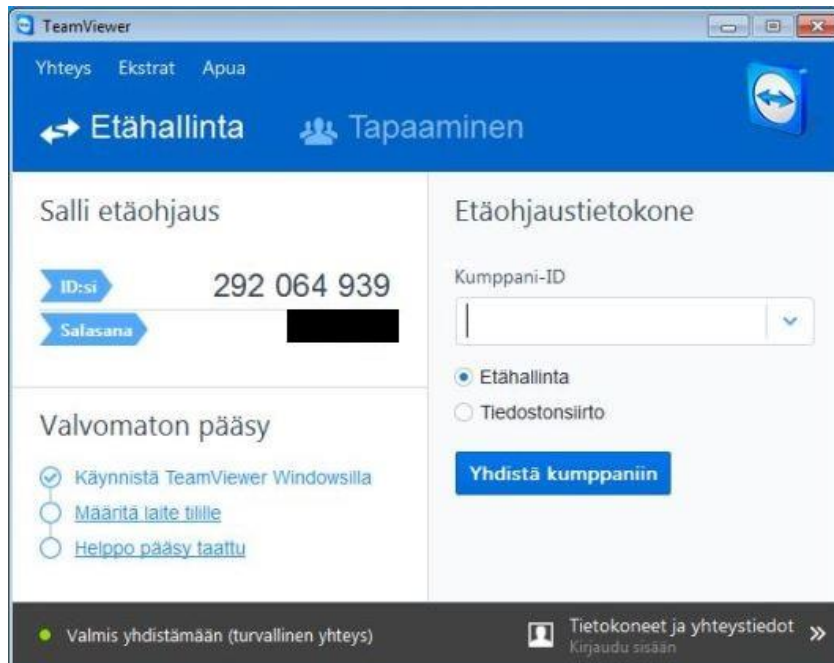
TeamViewer tukee useaa käyttöjärjestelmää kuten esimerkiksi Windows, Mac, Linux, Chrome OS, Android ja Blackberry. Myös selaimella käyttö on mahdollista tietyissä tapauksissa. Teamviewer on ilmainen sovellus ei-kaupalliseen sekä yksityiskäyttöön. Kaupallisessa käytössä ohjelmisto vaatii lisenssin.

Yhteyden muodostamisessa käytetään käyttäjäkohtaista ID:tä sekä ohjelman generoimaa tai itsemääritelyä salasanaa. Oletusasetuksilla ohjelmisto käyttää yhtä TeamViewerin palvelimista reitittääkseen yhteyden paikallisen ja etähallittavan laitteen välillä.

Yhteyden salauksen muodostamiseen käytetään 2048-bittistä RSA julkisen ja yksityisen avaimen vaihtoon perustuvaa salausta. Istunnon salauksessa käytetään AES:n 256-bittistä salausta. Samaa tekniikkaa käytetään myös HTTPS/SSL-suojauksessa, ja se on tämänhetkisten kriteerien mukaan arvioitu täysin turvalliseksi. Tässä menetelmässä yksityinen avain ei koskaan siirry pois client-tietokoneelta. Näin voidaan varmistaa, että keskenään yhdistetyt tietokoneet, mukaan lukien TeamViewerin reititinpalvelimet, eivät pysty tulkitsemaan salattua dataa.

(TeamViewer)

Ohjelmisto päättää kuinka yhteys luodaan, 70% tapauksissa "kädenpuristuksen" jälkeen yhteys on luotu UDP:n tai TCP:n avulla. Toinen yhdistystapa on TeamViewerin hajautettujen reitittimien lävitse reititetty yhteys (TCP- tai HTTP-tunnelin avulla.)



KUVA 13. TeamViewer-käyttöliittymä

6 OHJELMAT

Tässä kappaleessa käsitellään projektia varten luotuja ohjelmia ja niiden käyttötarkoituksia. Ohjelmien pääkohdat ja rakenteet käydään lävitse sekä selitetään oleelliset asiat koodista.

6.1 Synkronointiskripti

Ohjelmassa käytetään ns. try-, catch-, finally-rakennetta, jonka avulla saadaan ohjelman alustus, suoritus, virhetarkastelu ja lopetus suoritettua kolmen ohjelmablokin avulla.

```
try
{
    $sessionOptions = New-Object WinSCP.SessionOptions -Property @{
        Protocol = [WinSCP.Protocol]::Sftp
        HostName = "shell.tamk.fi"
        UserName = "*****"
        Password = "*****"
        SshHostKeyFingerprint = "ssh-rsa 2048
7d:36:24:d8:6b:37:9f:e5:92:25:03:b4:9b:f3:17:74"
    }
}
```

Yhteys alustetaan määrittämällä käytettävä protokolla (SFTP), palvelimen osoite, käyttäjätunnus palvelimelle sekä salasana. Myös yhteyden salaamiseen käytetty julkinen salausavain pitää määrittellä, ja siihen käytetty avain generoitiin yhdistettäessä palvelimelle ensimmäistä kertaa. Käyttäjänimi sekä salasana on sensuroitu opinnäytetyöstä mahdollisen väärinkäytön estämiseksi. Myös julkinen salausavain on muokattu, vaikkakin julkisen avaimen avulla salauksen purkaminen on nykytekniikalla käytännössä mahdotonta.

```
$session = New-Object WinSCP.Session
```

Ladataan WinSCP.Session luokka, joka sisältää käytettävät toiminnot

```
$session.add_FileTransferred( { FileTransferred($_) } )
```

Add_FileTransferred-komennolla tulostetaan onnistunut siirto

```
$session.Open($sessionOptions)
```

Open-käskyllä avataan yhteys käyttäen edellä määriteltyjä asetuksia (\$sessionOptions.)

```
$synchronizationResult = $session.SynchronizeDirectories(  
    [WinSCP.SynchronizationMode]::Remote, "C:\Users\Omistaja\Desktop\MIT-  
TAUSDATA\", "/mnt/projects/Tervassilta/mittaukset/", $False)  
  
    $synchronizationResult.Check()  
}  
finally  
{  
    $session.Dispose()  
}  
  
exit 0  
}  
catch [Exception]  
{  
    Write-Host $_.Exception.Message  
    exit 1  
}
```

KUVA 14. Synkronointiin käytettyä koodia

6.2 Synkronoinnin käynnistysohjelma

Turvallisuussyistä Windows-käyttöjärjestelmässä on PowerShell-skriptien suoritus oletuksena kytketty pois päältä. Tämän vuoksi se täytyi kytkeä päälle erillisellä komennolla:

```
Set-ExecutionPolicy Unrestricted
```

Komentoriville kirjoitettu komento sallii kaikkien Powershell skriptien suorittamisen ilman rajoituksia, ongelmana tässä kuitenkin on kaikkien skriptien suorituksen salliminen, mikä aiheuttaa tietoturvariskin. Ratkaisuksi tähän käytettiin skriptin yksittäiseen suoritukseen tarkoitettua komentoa:

```
powershell.exe -ExecutionPolicy Bypass -Command "C:\Users\Omistaja\Sync.ps1"
```

Tämä komento ohittaa Windows-järjestelmän suorituskäytännöt antaen oikeuden suorittaa polulla viitattu skripti kuitenkaan hyväksymättä muiden ulkopuolisten skriptien käyttöä. Komennon suorituksen automatisoimiseksi kyseisestä komentorivikomennosta tehtiin .bat-tiedosto, jonka suorittaminen saatiin ajastetuksi. Lisäksi .bat-tiedostoon lisättiin @ECHO OFF ja exit -komennot. @ECHO OFF -komento estää terminaali-ikkunan luomisen, sillä ikkunan näkyminen ei tässä tapauksessa ole tarpeellista. Exit-komento lopettaa koodin suorittamisen.

Lopullinen suorituskomento näyttää seuraavalta:

```
@ECHO OFF  
powershell.exe -ExecutionPolicy Bypass -Command "C:\Users\Omistaja\Sync.ps1"  
exit
```

7 JÄRJESTELMÄN TESTAUS

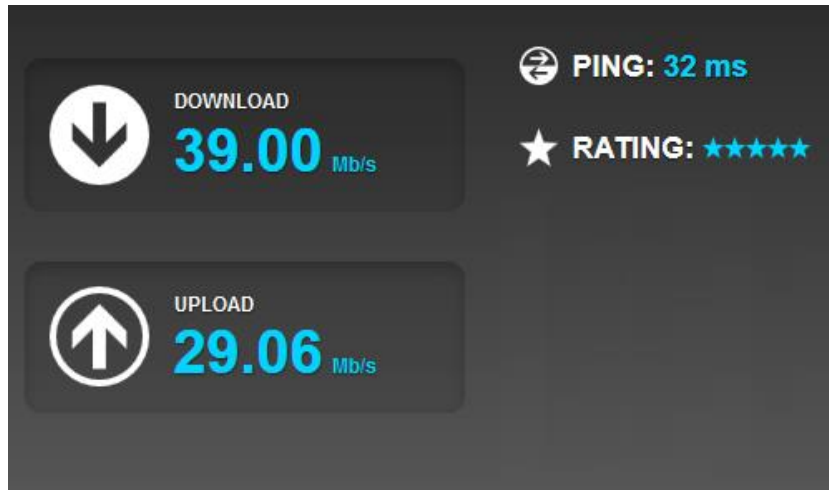
Järjestelmän testaus suoritettiin testaamalla yksilöllisesti jokainen järjestelmän komponentti mahdollisten virheiden varalta. Keskusyksikön käyttöjärjestelmän vakauden testaus tapahtui muiden järjestelmän osien testauksen yhteydessä, sillä kaikki komponentit ovat yhteydessä joko suorasti tai epäsuorasti keskusyksikköön. Koko järjestelmän pitkäaikainen testaus suoritetaan ohjelmien testauksen yhteydessä.

7.1 Laajakaistareitittimen testaus

Testauksen aloittaminen tapahtui kytkemällä keskusyksikkö internetiin käytettävissä olevan, aiemmin esitellyn 4G-mobiililaajakaistareitittimen avulla. Aluksi testattiin selaimella, että yhteyden muodostaminen internettiin onnistuu. Kun havaittiin yhteyden muodostamisen onnistuneen, testattiin yhteyden nopeutta, signaalin voimakkuutta ja viivettä.

Yhteyden nopeutta ja viivettä testattiin <http://www.speedtest.net>-osoitteessa, testi käyttää http-protokollaa käyttäen OSI-mallin tasoa 7. Speedtest.net:n testipalvelimia löytyy suomestakin useita, Tampereen seudulla testipalvelimia on testihetkellä 2 kappaletta, Tampereen puhelimen sekä Tamico Oy:n ylläpitäminä.

Testipalvelimena käytettiin Tamico:n ylläpitämää palvelinta (KUVA 15), koska fyysiseltä sijainniltaan tämä on lähin käytettävissä oleva palvelin.



KUVA 15. Speedtest.net testin tulokset

Datan siirtonopeudet ovat hyvin tyypillisiä 4G-yhteydelle sekä viive on normaali mobiiliverkkoyhteydelle, näin ollen yhteys täyttää projektin vaatimat vaatimukset riittävän hyvin. Signaalin voimakkuus testiolosuhteissa oli -91 dBm 50 ASU, eli signaalin voimakkuus on kohtalainen, mutta riittäisi projektin vaatimaan yhteyteen. Signaalin voimakkuutta usein kuvataan termillä ASU, joka on yksinkertainen signaalin voimakkuuden muunnos desibeliasteikolta kokonaisluvuksi. LTE-yhteyden ollessa toiminnassa ASU-arvon täytyy olla väliltä 0-97, suuremman luvun tarkoittaessa pienempää vaimennusta. ASU-arvojen 1-96 välillä ASU saadaan laskettua käyttäen kaavaa 1:

$$1) \quad (\text{ASU} - 141) \leq \text{dBm} < (\text{ASU} - 140)$$

Signaalin vaimennuksen ollessa -140 dBm signaalin ASU-arvo on 0, kun taas signaalin voimakkuuden ollessa suurempi kuin -44 dBm ASU-arvo on 97. Mahdolliseksi ongelmaksi kuitenkin muodostuu lopullinen asennus peltiseen kytkentäkaappiin, sillä kaappi saattaa aiheuttaa huomattavaa vaimennusta signaalin voimakkuuteen. Ongelmaan ratkaisuksi kytkentäkaappiin voitaisiin sijoittaa ulkopuolinen 4G-antenni, minkä avulla signaali saataisiin vahvistettua yhteyden pysyvyyden takaamiseksi.

7.2 Dataloggerin testaus

Dataloggeria testattiin käyttämällä testimittauksessa kaikkia neljää kanavaa suurimmalla mahdollisella näytteenottotaajuudella, eli 10 kHz taajuudella. Varsinaisessa mittauksessa ei pitäisi olla tarvetta missään olosuhteissa yli 1 kHz taajuuksiin, eikä näin ollen testitilanteen suuruista kuormitusta pitäisi missään tilanteessa syntyä.

Kuuden tunnin pituisella kokeilujaksolla syntyi n. yhden gigatavun suuruinen mittausdata, eikä mittauksessa ollut mitään ongelmia, joten voidaan todeta dataloggerin toimivan hyvin jopa suurella kuormituksella.

8 OHJELMIEN TESTAUS

Tässä kappaleessa käydään lävitse yhteenveto kirjoitettujen ohjelmien testausvaiheista, ohjelmien suorituksesta seuranneista ongelmista ja niiden ratkaisuksista. Ohjelmien testauksessa pyrittiin mahdollisimman pitkäaikaiseen yhtämittaiseen testausajanjaksoon, jotta havaittaisiin mahdolliset poikkeamat ohjelmien toiminnassa.

Ohjelmien testauksessa tarkkailtiin myös ohjelmien kuormittavuutta tutkimalla järjestelmän muistinkäyttöä, prosessorin kuormitusastetta sekä levyn käyttöä. Testauksessa havaittiin ohjelmien olevan erittäin kevyitä suorittaa, joten ongelmia ei järjestelmän rasituksen näkökulmasta ole.

8.1 Synkronointiskriptin testaus

Testauksen aloittaminen tapahtui synkronointiskriptin testauksella. Synkronointiskriptiä testattiin PowerShell-ohjelmointiympäristön ”Run”-komennolla, jolla nähdään ohjelman suoritus vaiheittain. Ohjelman suorittaminen sujui halutulla tavalla ja tiedostojen kopiointi onnistui. Ohjelman ajastettua suorittamista yritettiin käyttämällä Task Scheduler-ohjelmaa, ohjelman suoritus ei kuitenkaan suoraan .ps1-tiedostoa käyttämällä onnistunut, sillä Task Scheduler ei tue kyseistä tiedostformaattia. Tämä ongelma ratkaistiin käyttämällä erillistä synkronoinnin aloitusohjelmaa, jolla .ps1-tiedosto saatiin suoritettua.

8.2 Synkronoinnin aloitusohjelman testaus

Synkronoinnin aloitusohjelman testaus tapahtui suorittamalla kirjoitettu .bat-ohjelma. Ohjelman suoritus ei kuitenkaan onnistunut, vaan tietokone antoi virheilmoituksen, joka ilmoitti ettei käyttäjällä ole riittäviä oikeuksia skriptin suoritukseen. Internetistä haettiin tietoa kyseisestä virheilmoituksesta ja todettiin että koodiin täytyi lisätä komento, joka antaisi oikeudet suorittaa .ps1-skriptin järjestelmänvalvojan oikeuksilla. Komennon lisäyksen jälkeen ohjelma toimi kuten oletettiin, ja tiedostojen synkronointi onnistui ajastetusti.

Ohjelmaa testattiin n. 10 päivän ajan onnistuneesti ajastetusti Task Scheduleria käyttäen, mittaustulokset siirtyivät haluttuun kohteeseen ja tiedostot säilyivät muuttumattomina. Ohjelman luotettavuuden testausta jatkettiin vielä muutaman päivän ajan, jolloin havaittiin, ettei tiedostot synkronoidu, eikä etätietokoneeseen saa muodostettua yhteyttä. Tällöin käytiin paikan päällä tutkimassa etätietokonetta ja todettiin että joku ulkopuolinen henkilö oli irroittanut tietokoneesta virtakaapelin. Kaapeli kytkettiin takaisin ja tiedon siirtoa testattiin vielä viikko ongelmitta.

9 POHDINTA

Työ oli sopivan haasteellinen, koska aiempaa kokemusta vastaavista mittauslaitteistoista ei ollut ennen tätä projektia. Laitteistot projektiin oli hankittu valmiiksi, eikä tähän näin ollen päästy vaikuttamaan. Projektin toteuttamisesta olisi saanut mielenkiintoisemman toteuttamalla valmiin järjestelmän sijasta mittausdatan keräämisen Raspberry Pi-, Beaglebone- tai jollain vastaavalla pienellä Linux-pohjaisella I/O-liitäntöjä sisältävällä kortti-tietokoneella.

Laitteiston osalta suuria ongelmia ei ollut, sillä käytössä ollut laitteisto on suunniteltu juuri tämän tyyppisiä mittauksia varten. Ongelmia sen sijaan projektissa aiheutti järjestelmän automatisointi, sillä valmista ratkaisua tähän ei ollut saatavilla vaan automatisointi täytyi toteuttaa itse. Etätyöpöydän muodostaminen aiheutti myös hieman ongelmia, sillä alun perin käytettäväksi suunniteltua kevyttä VNC-ohjelmistoa, jonka käytöstä olisi ollut kokemusta, ei voitu käyttää.

Opinnäytetyön kirjoittamisen aikana laitteistoa ei vielä oltu saatu kytkettyä, joten varsinaisia mittaustuloksia ei saatu analysoitavaksi. Lopulliset järjestelmän testaukset jäivät tämän osalta myös opinnäytetyön aihealueen ulkopuolelle. Järjestelmän testaukset sujui- vat testiolosuhteissa kuitenkin odotetusti, joten varsinaisessa mittaustilanteessa tuskin ongelmia ilmenee.

Kokonaisuudessaan projekti oli mielenkiintoinen ja työn edetessä uusia asioita tuli opit- tua jo aikaisemmin käytössä olleista ohjelmista ja uusista ohjelmista, joita ei ole aiemmin käytetty.

LÄHTEET

T.Ruotsila. PUUSILLAN SYRJÄLANKKUKANNEN PITKÄAIKAISKESTÄVYYS SÄÄRASITUKSESSA. Theseus. Luettu 5.7.2016

https://www.theseus.fi/bitstream/handle/10024/109578/Ruotsila_Tommi.pdf.pdf?sequence=1

Tampereen kauppakamarilehti. PIKKUKAUPUNKI NOUSEE VUOREKSEN ISOKUUSEEN. Luettu 5.7.2016

<http://www.tampereenkauppakamarilehti.fi/uutispoyta/puukaupunki-nousee-vuoreksen-isokuuseen-112>

H.Viljanen. SILTOJEN PUISET LAMELLIKANNET. Theseus. Luettu 6.7.2016

https://theseus32-kk.lib.helsinki.fi/bitstream/handle/10024/107903/Viljanen_Heli.pdf?sequence=1

Tekes. INKA-projekti. Luettu 12.7.2016

<http://www.tekes.fi/ohjelmat-ja-palvelut/ohjelmat-ja-verkostot/inka/>

Tampereen kaupunki, Isokuusen alue. Luettu 12.7.2016

<http://www.tampere.fi/vuores/asuminenjarakentaminen/isokuusi.html>

PCD-400A Sensor Interface manual. Luettu 26.7.2016

http://www.straintech.fi/pdf/pcd-430a_catalog_1501_01_eng.pdf

Advantech UNO-2174G – keskusyksikkö. Luettu 26.7.2016

http://www.advantech.com/products/1-2mli9a/uno-2174g/mod_797f3e7a-3ef3-4df6-8afe-5ab353002983

Measuring Strain with Strain Gages. Luettu 29.7.2016

<http://www.ni.com/white-paper/3642/en/>

PCD-400A Kyowa. Luettu 29.7.2016

<http://www.kyowa-ei.com/eng/product/category/acquisition/pcd-400a/index.html>

HUAWEI, E5776 Specifications. Luettu 5.8.2016

<http://consumer.huawei.com/au/mobile-broadband/mobile-wifi/features/e5776-au.htm>

Sonera peittoaluekartta. Luettu 5.8.2016

<https://www.sonera.fi/asiakastuki/ohjeet/Peittoaluekartta?id=1398>

Oulu, teknologiakasvatus. Luettu 8.8.2016

<http://www.oulu.fi/teknokas/tehtavakortit/venymaliuska.pdf>

Wheatstonen bridge. Luettu 15.8.2016

<http://www.electronics-tutorials.ws/blog/wheatstone-bridge.html>

WinSCP. Luettu 18.8.2016

<https://winscp.net/eng/index.php>

TeamViewer. Luettu 24.8.2016

<https://www.teamviewer.com/en/>

TeamViewer, TeamViewer Security Statement. Luettu 16.9.2016

<https://downloadeu2.teamviewer.com/docs/fi/TeamViewer-Security-Statement-fi.pdf>

Microsoft, What is PowerShell? Luettu 30.9.2016

<https://msdn.microsoft.com/en-us/powershell/mt173057.aspx>

Ookla Speedtest. Luettu 10.11.2016

<http://www.speedtest.net/>

How Is Temperature Affecting Your Strain Measurement Accuracy?. National Instruments. Luettu 6.12.2016

<http://www.ni.com/white-paper/3432/en/>

LIITTEET

Liite 1. Valmistunut silta



Liite 2. Synkronointiskripti

```

# Ladataan WinSCP .NET assembly
1. Add-Type -Path "WinSCPnet.dll"
2.
3.
4.
5. function FileTransferred
6. {
7.     param($e)
8.
9.     if ($e.Error -eq $Null)
10.    {
11.        Write-Host ("Upload of {0} succeeded" -f $e.FileName)
12.    }
13.    else
14.    {
15.        Write-Host ("Upload of {0} failed: {1}" -f $e.FileName, $e.Error)
16.    }
17.
18.    if ($e.Chmod -ne $Null)
19.    {
20.        if ($e.Chmod.Error -eq $Null)
21.        {
22.            # Asetetaan tiedoston käyttöoikeudet siirtoa edeltävään tilaan
23.            Write-Host ("Permissions of {0} set to {1}" -f $e.Chmod.File-
Name, $e.Chmod.FilePermissions)
24.        }
25.        else
26.        {
27.            Write-Host ("Setting permissions of {0} failed: {1}" -f $e.Chmod.File-
Name, $e.Chmod.Error)
28.        }
29.    }
30.    }
31.    else
32.    {
33.        Write-Host ("Permissions of {0} kept with their defaults" -f $e.Destination)
34.    }
35.
36.    if ($e.Touch -ne $Null)
37.    {
38.        if ($e.Touch.Error -eq $Null)
39.        {
40.            # Asetetaan tiedoston aikaleima siirtoa edeltävään arvoon
41.            Write-Host ("Timestamp of {0} set to {1}" -f $e.Touch.File-
Name, $e.Touch.LastWriteTime)
42.        }
43.        else
44.        {
45.            Write-Host ("Setting timestamp of {0} failed: {1}" -f $e.Touch.File-
Name, $e.Touch.Error)
46.        }
47.    }
48.    }
49.    else
50.    {
51.        # Mahdollinen virhetilanne, ei pitäisi tapahtua siirrettäessä paikallisesta
#kansioista etäkansioon
52.        Write-Host ("Timestamp of {0} kept with its default (current time)" -f $e.Des-
tination)
53.    }
54. }
55.
56. # Itse siirtoskripti alkaa tästä
57.
58. try
59. {

```

```

60.     $sessionOptions = New-Object WinSCP.SessionOptions -Property @{
61.         Protocol = [WinSCP.Protocol]::Sftp
62.         HostName = "shell.tamk.fi"
63.         UserName = "*****"
64.         Password = "*****"
65.         SshHostKeyFingerprint = "ssh-rsa 2048
7d:36:24:d8:6b:37:9f:e5:92:25:03:b4:9b:f3:17:74"
66.     }
67.
68.     $session = New-Object WinSCP.Session
69.     try
70.     {
71.         # Ilmoitetaan onko tiedoston siirto onnistunut vai ei
72.         $session.add_FileTransferred( { FileTransferred($_) } )
73.
74.         # Avataan yhteys
75.         $session.Open($sessionOptions)
76.
77.         # Synkronointi
78.         $synchronizationResult = $session.SynchronizeDirectories(
79.             [WinSCP.SynchronizationMode]::Remote, "C:\Users\Omistaja\Desktop\MIT-
TAUSDATA\", "/mnt/projects/Tervassilta/mittaukset/", $False)
80.
81.         # Tulostaa mahdolliset virheet
82.         $synchronizationResult.Check()
83.     }
84.     finally
85.     {
86.         # Katkaistaan yhteys
87.         $session.Dispose()
88.     }
89.
90.     exit 0
91. }
92. catch [Exception]
93. {
94.     Write-Host $_.Exception.Message
95.     exit 1
96. }

```