

Henri Tölli

**STECAGRID 4803 -INVERTTERISTÄ SAATAVIEN TIETOJEN
TUTKIMINEN**

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Sähkötekniikan koulutusohjelma
Joulukuu 2017**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	Aika Joulukuu 2017	Tekijä/tekijät Henri Tölli
Koulutusohjelma Sähkötekniikka		
Työn nimi STECAGRID 4803 -INVERTTERISTÄ SAATAVIEN TIETOJEN TUTKIMINEN		
Työn ohjaaja Hannu Puomio		Sivumäärä 27
Työelämäohjaaja		
<p>Opinnäytetyössä tutustutaan StecaGrid 4803 -invertteristä saataviin tietoihin. Työssä käydään läpi aurinkopaneelijärjestelmä, invertterin yhdistäminen internetiin sekä tutustutaan invertterin käyttöliittymään ja tutkitaan, kuinka siitä saadaan sähköntuotantoon liittyviä tietoja.</p> <p>Opinnäytetyöhön liittyvä tieto on etsitty internetistä. Työssä käytettiin apuna olemassa olevaa aurinkopaneelijärjestelmää ja Raspberry Pi -mikrotietokonetta.</p>		
Asiasanat Aurinkosähkö, invertteri, Python, Raspberry		

ABSTRACT

Centria University of Applied Sciences	Date December 2017	Author Henri Tölli
Degree programme Electrical Engineering		
Name of thesis ANALYZING DATA FROM STECAGRID 4803 SOLAR PANEL INVERTER		
Instructor Hannu Puomio	Pages 27	
Supervisor		
<p>This thesis focused on how to get data out of the StecaGrid 4803 solar panel inverter. The thesis includes a demonstration about the basic parts of the whole solar panel system and how to connect the inverter online. In addition, the interface of the inverter and how to get data related to electricity production out of the inverter are discussed in this thesis.</p> <p>All information that was used in this thesis was found in the internet. The existing solar panel system and Raspberry Pi computer were utilized in doing the thesis.</p>		

Key words Inverter, Python, Raspberry, solar energy.
--

TIIVISTELMÄ
ABSTRACT
SISÄLLYS

1 JOHDANTO	1
2 AURINKOPANEELIJÄRJESTELMÄ	2
2.1 Aurinkopaneelit.....	2
2.2 Invertteri.....	3
2.3 Asennustelineet, kiinnikkeet, kaapelit ja liittimet	3
3 STECAGRID 4803 -KÄYTTÖLIITTYMÄ	5
4 INVERTTERILTÄ SAATAVIEN TIETOJEN TUTKIMINEN	11
4.1 Wireshark	11
4.2 Raspberry Pi	12
4.3 Lähdekoodi	13
4.3.1 HTML (Hypertext Markup Language)	13
4.3.2 Javascript.....	14
4.4 Raspberry Pi 3	14
4.5 Python -ohjelmointikieli	16
5 INVERTTERIN KÄYTTÖLIITTYMÄN LÄHDEKOODI JA PARSEROINTI	18
5.1 Lähdekoodin haku ja parserointi	20
5.1.1 URL ja urllib	21
5.1.2 Lähdekoodin haku	21
5.1.3 Lähdekoodin parserointi	22
6 TIETOJEN TALLENTAMINEN	24
6.1 Tietojen tallennus testiympäristössä	24
6.2 Tietojen tallennus Google Spreadsheets.....	25
7 YHTEENVETO	26
LÄHTEET	28

KUVAT

KUVA 1. Aurinkopaneelijärjestelmän osat	2
KUVA 2. Kuvia aurinkopaneelijärjestelmän asennuksesta	4
KUVA 3. Invertteri seinälle asennettuna. Näytössä näkyy hetkellinen sähkön tuotanto.	5
KUVA 4. Invertterin käyttöliittymän päänäkymä tietokoneen selaimella avattuna	5
KUVA 5. Measurements, mittaustulos sivu invertterin käyttöliittymässä.....	6
KUVA 6. Kuva 6. Info -näkymä invertterin käyttöliittymässä.....	7
KUVA 7. Internet-yhteys WLAN -reitittimeltä sillan kautta invertterille.....	8
KUVA 8. Selaimen avautuvat WET54G -sillan konfigurointi asetukset.....	9
KUVA 9. Invertteri valmistajan tarjoaman seurantal palvelun päänäkymä	10
KUVA 10. Osa Wireshark tallenteesta	12
KUVA 11. Yksinkertainen esimerkki HTML -lähdekoodista ja kuva, miltä kyseinen esimerkki näyttää selaimessa.....	14

KUVA 12. Raspberry Pi 3 -piirilevy	15
KUVA 13. Esimerkki Python -koodista	17
KUVA 14. Koodiesimerkki Python -simulointiohjelmalla ajettuna	17
KUVA 15. Osa käyttöliittymän päänäkymän (Home) lähdekoodista.....	18
KUVA 16. Measurements sivun lähdekoodi ja mittaustulosten Javascripti	19
KUVA 17. Web -käyttöliittymän kuukausituotantonäkymä.....	20
KUVA 18. Measurements JavaScriptin lähdekoodi Python 3.6.1 -ohjelmalla haettuna tietokoneen kova-levyltä.....	22
KUVA 19. Numerotiedot measurements.js lähdekoodista	23
KUVA 20. Testiympäristön sähköntuotantotiedot tallennettuna tekstitiedostoon Python -ohjelmalla ..	24

TAULUKOT

TAULUKKO 1. Raspberry Pi 3 Model B ominaisuudet	15
---	----

1 JOHDANTO

Opinnäytetyön aiheena oli tutustua aurinkopaneeliin liitettävästä invertteristä saataviin tietoihin. Invertteri on laite, joka kytketään aurinkopaneelien ja sähköverkon väliin. Invertterin tehtävänä on muuttaa aurinkopaneelilta tuleva tasajännite verkkoon syötettäväksi vaihtojännitteeksi. Markkinoilla on tarjolla usean eri valmistajan inverttereitä. Työssäni tutkin StecaGrid 4803 -mallia.

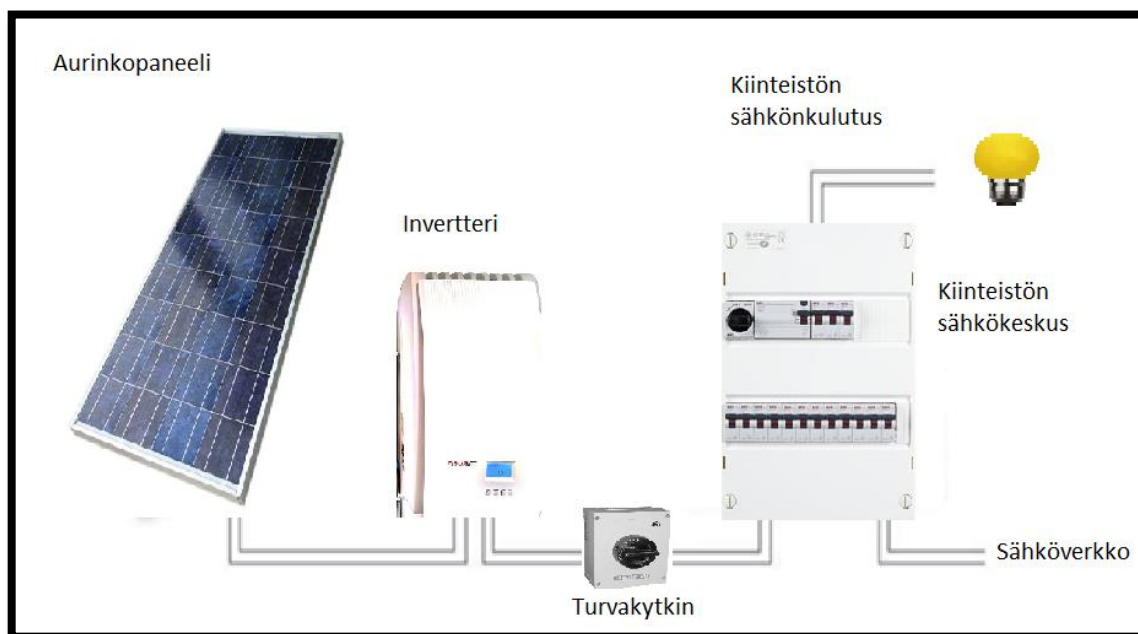
Ongelmana oli ensinnä saada yhdistettyä invertteri internetiin. Rakennuksessa, johon invertteri oli asennettu, ei ollut internet liityntää ollenkaan. Läheisessä talossa oli WLAN-tukiasema, josta oli mahdollista saada yhteys invertteriin. Invertterissä ei ollut sisäänrakennettua langatonta vastaanotinta, joten jouduin asentamaan siihen ulkopuolisen WLAN-vastaanottimen, josta sain RJ45-liitynnän kautta internet-yhteyden invertterille.

Internet-yhteyden kautta invertteri lähettää tietoja valmistajansa palvelimelle, josta niitä voi seurata etänä tietokoneella, tabletilla tai matkapuhelimella selaimen kautta. Invertterin valmistaja tarjoaa seuranta-palvelun kahdeksi vuodeksi ilmaiseksi, jonka jälkeen se muuttuu maksulliseksi. Tavoitteena oli selvittää, voisiko invertterin tietoja kerätä itse, jolloin tulevaisuudessa ei tarvitsisi maksaa etäseuranta-palvelusta.

Opinnäytetyön ensimmäisessä osiossa käydään läpi pääpiirteittäin aurinkopaneelijärjestelmä ja sen asennus. Toisessa osiossa tarkastellaan invertterin käyttöliittymää valikoineen, sekä yhdistetään invertteri internetiin. Kolmannessa osiossa perehdytään tarkemmin invertteriltä saataviin sähköntuotantoon liittyviin tietoihin, sekä tutustutaan, että kuinka niihin päästään käsiksi. Osiossa esitellään myös tarvittavat työkalut invertterin tietojen tutkimiseen. Työn neljännessä osiossa tarkastellaan selainpohjaisen käyttöliittymän lähdekoodia, lähdekoodin hakua ja parserointia. Viidennessä osiossa pohditaan mahdollisia tietojen tallennusmuotoja. Lopussa yhteenveto tehdystä työstä sekä mahdollisista työn jatkokehityskohteista.

2 AURINKOPANEELIJÄRJESTELMÄ

Tutkittava aurinkopaneelijärjestelmä asennettiin Pyhäjoelle Yppäriin huhtikuun puolella välissä. Olin mukana asentamassa järjestelmää, jolloin se tuli hyvin tutuksi. Valmiiseen pakettiin kuului 20 kappaletta Solarwatt Blue 60P 260Wp -aurinkopaneelia, StecaGrid 4803 -invertteri, katolle asennettavat aurinkopaneelien asennustelineet sekä tarvittavat kiinnikkeet, kaapelit ja liittimet. Paketin lisäksi asentamiseen tarvitsi vain turvakytkimen, jolla invertterin ja aurinkopaneelit voi kytkeä irti sähköverkosta, pätkän MMJ 5x2,5S kaapelia turvakytkimen asentamiseen sekä paneelien ja telineiden maadoitukseen MK6 KEVI kaapelia.



KUVA 1. Aurinkopaneelijärjestelmän osat.

2.1 Aurinkopaneelit

Aurinkopaneelit olivat Solarwatt:in valmistamia. Solarwatt on Saksassa toimiva aurinkoenergiaan erikoistunut yritys. Paneelit olivat nimellisteholtaan 260 wattisia. Kaikki 20 paneelia asennettiin sarjaan. Yhteensä ne kykenevät tuottamaan yli 800 voltin tyhjäkäyntijännitteen.

2.2 Invertteri

StecaGrid 4803 -invertteri on kolmivaiheinen invertteri. Invertteri kykenee syöttämään verkkoon maksimissaan 4800 watin teholla sähköä. Teho jakautuu tasan kaikille kolmelle vaiheelle. Invertteristä löytyy itsessään LCD-näyttö, josta voi muokata laitteen asetuksia ja seurata sähkön tuotantoa.

Laitteen pohjasta löytyy kolme RJ45 -porttia ja yksi RJ10 -portti. RJ45 -kytkimistä Ethernet-portin kautta laitteen saa yhdistettyä internettiin sekä pääsee käsiksi laitteen käyttöjärjestelmään. Kaksi muuta RJ45 -porttia ovat RS485 -standardin sarjaliikenneväyläportteja, joiden kautta voidaan esimerkiksi yhdistää useampi invertteri yhteen, jolloin ne kommunikoiivat toistensa kanssa. RS485 -sarjaliikenneväylään voidaan myös kytkeä kolmannen osapuolen valmistamia ulkoisia laitteita, kuten esimerkiksi tiedonkeruulaitteita. RJ10 -liitäntäiseen Modbus RTU -porttiin on mahdollista yhdistää tiettyjä energiamittareita, jotka kykenevät lukemaan invertteriltä tulevaa dataa.

2.3 Asennustelineet, kiinnikkeet, kaapelit ja liittimet

Paketin mukana tuli oikean mittaiset asennustelineet paneeleille, peltikattokiinnikkeet telineille sekä tarvittavat ruuvit, pultit ja mutterit. Paneelit tulivat telineisiin kiinni pikakiinnikkeillä, jotka kiristettiin kuusiokoloruuvilla.



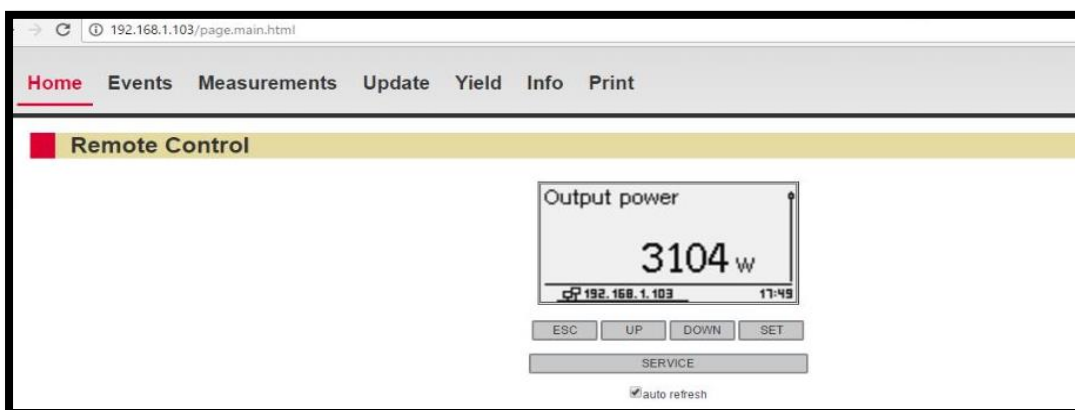
KUVA 2. Kuvia aurinkopaneelijärjestelmän asennuksesta. Vasemmalla ylhäällä paneelien asennuskiskot kiinnitettyinä peltikattoon. Vasemmalla alhaalla paneelit asennettuina katolle. Oikealla invertteri asennettuna seinälle.

3 STECAGRID 4803 -KÄYTTÖLIITTYMÄ

Invertterissä on sisäänrakennettu käyttöliittymä, johon pääsee RJ45 Ethernet-portin kautta tietokoneen internet selaimella käsiksi verkkokaapelin välityksellä. Käyttöliittymän päävalikkoon avautuu sama kuva, joka invertterin näytössä itsessään näkyy. Selainpohjaisen käyttöliittymän päävalikkoon on tehty virtuaaliset painikkeet, joilla voi selaila invertterin valikoita ja muuttaa asetuksia. Invertteriä voi siis etäkäyttää selaimen kautta.




KUVA 3. Invertteri seinälle asennettuna. Näytössä näkyy hetkellinen sähkön tuotanto.



KUVA 4. Invertterin käyttöliittymän päänäkymä tietokoneen selaimella avattuna.

Home -päänäkymän lisäksi käyttöliittymässä pääsee valikoihin Events, Measurements, Update, Yield, Info ja Print. Events -valikon takaa avautuvat hälytykset, mikäli niitä on laitteen muistissa. Measurements sivulla näkyvät laitteelta saatavat mittaustulokset yhtenäisessä taulukossa. Update -sivulle voi ladata tietokoneelta tiedoston, jolla invertterin ohjelmiston voi päivittää. Yield -valikosta löytyy laitteen kokonaissähköntuotanto, sekä vuosi-, kuukausi- ja päiväkohtaiset tuotannot.



Inverter		
Name	Value	Unit
P DC	3047.23	W
U DC	610.30	V
I DC	4.99	A
U AC1	236.65	V
U AC2	233.41	V
U AC3	239.00	V
I AC1	4.24	A
I AC2	4.21	A
I AC3	4.19	A
F AC	50.03	Hz
F AC1	50.03	Hz
F AC2	50.03	Hz
F AC3	50.03	Hz
P AC	2987.96	W
P AC1	1005.02	W
P AC2	992.70	W
P AC3	992.61	W

KUVA 5. Measurements, mittaustulos sivu invertterin käyttöliittymässä.

Info -sivulta löytyvät invertterin tiedot ja sähköntuotantoon liittyvät raja-arvot sekä ohjelmiston ja laitteiston versiotiedot. Invertterin käyttöönotossa valikoista valitaan maa, mihin invertteri asennetaan. Maan perusteella invertteri asettelee sähkön syöttöön liittyvät suojaus- ja sähkönlaatuasetukset. Viimeiseltä Print -sivulta voi tulostaa laitteiston tietoja.

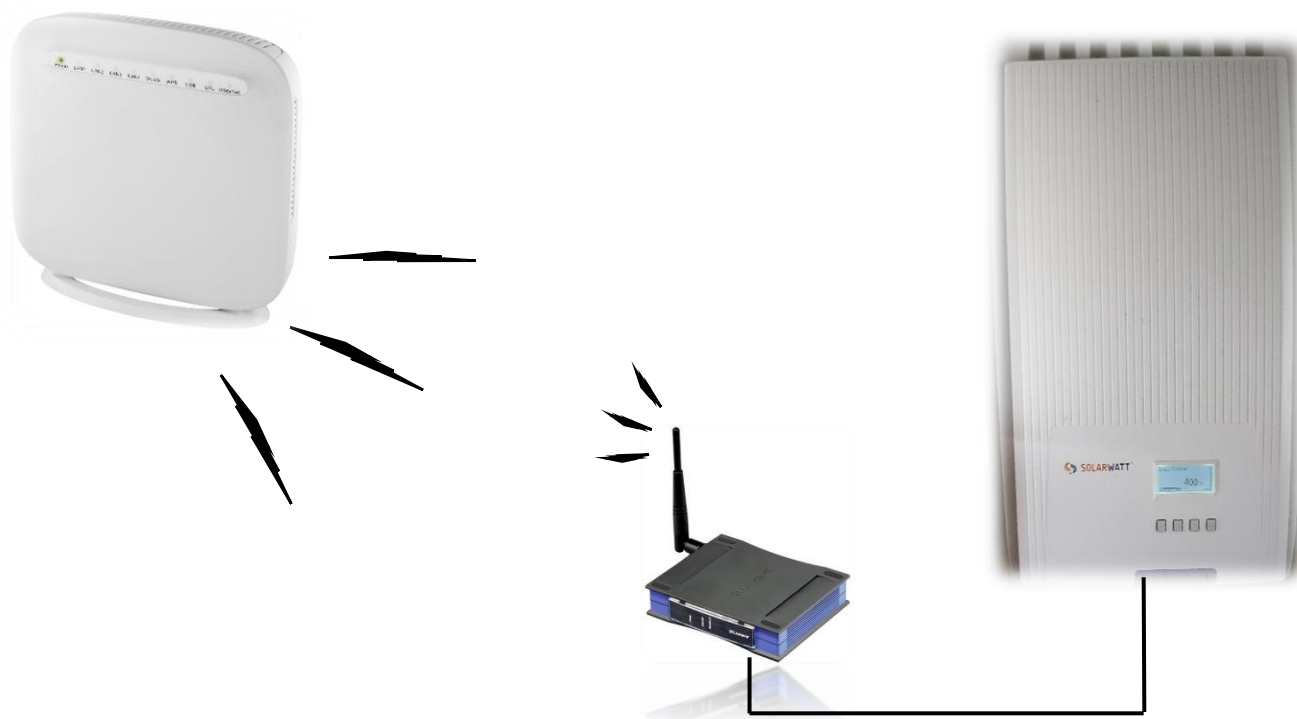
Home Events Measurements Update Yield Info Print									
System Information									
System Info		Country Settings		Software Versions		Hardware Versions			
Type	Value	#	Name	Value	Unit	Name	Version	Name	Version
Name	StecaGrid 4803		Code	35800		HMI BFAPI	2.5.4	HMI	3
Serial Number	753773BC005940570001		Country	Suomi		HMI FBL	2.2.0	PU	3
Power Limit	---	0	U Nominal	230.00	V	HMI APP	2.8.0	ENS1	3
Country Limit	---	1	U 1LoLimit	85.00	%	HMI PAR default	2.0.4	ENS2	3
Nominal Power	4800W	2	U 1LoTime	0.20	s	HMI OEM SG4803	1.0.4		
RS485 Address	1	3	U 1UpLimit	110.00	%	PU BFAPI	2.5.4		
Web Portal	Meteocontrol	4	U 1UpTime	0.20	s	PU FBL	2.5.0		
Supported Portals	Meteocontrol	11	F Nominal	50.00	Hz	PU APP	2.7.0		
		12	F 1LoLimit	48.00	Hz	PU PAR default	2.0.0		
		13	F 1LoTime	0.20	s	PU OEM 4800W-IP21-25A	2.0.2		
		14	F 1UpLimit	51.00	Hz	ENS1 BFAPI	2.5.4		
		15	F 1UpTime	0.20	s	ENS1 FBL	1.5.0		
		23	F UpReErr	0.00	Hz	ENS1 APP	2.7.0		
		24	Recon Time	30.00	s	ENS1 PAR default	3.0.1		
		25	Recon Grad	100.00	%	ENS1 BRAM default	2.0.0		
		27	F BackGrad	0.00		ENS2 BFAPI	2.5.4		
		28	F ExitDelay	0.00	s	ENS2 FBL	1.5.0		
		29	Istanding	1.00		ENS2 APP	2.7.0		
		30	PhaseAngle	1.80	°	ENS2 PAR default	3.0.1		
		31	F Rocof	0.00	Hz	ENS2 BRAM default	2.0.0		
		32	T Rocof	1.00	s				
		33	I DcRellmax	0.00	%				
		34	I DcRellnom	0.00	%				
		35	T IDc rel	0.00	s				
		36	I DC abs	1.00	A				
		37	T IDc abs	0.20	s				
		38	R iso	1000000.00	Ohm				
		39	P UserLimit	4800.00	W				
		40	Tolerant	0.00					
		41	Selftest	0.00					
		44	P UserOnOff	0.00					
		47	Q CR_Time	1.00	s				
		48	RapidShut	0.00					
		57	P Shutdown	0.00	%				
		58	Q fix var	0.00	var				
		59	U LineUpLim	115.00	%				
		60	U LineLoLim	80.00	%				
		61	U LineTime	0.20	s				
		63	Qmax / Smax	0.00	%				

KUVA 6. Info -näkyä invertterin käyttöliittymässä.

Selainpohjaisen käyttöliittymän käyttö RJ45 -portista ja sitä kautta tietojen seuranta tietokoneella ei ole kovin järkevää. Langattomasti tietokoneen voisi sijoittaa muualle kuin invertterin läheisyyteen. Invertterin käyttöliittymään taasen ei saa yhteyttä langattoman verkon ulkopuolelta. Tästä johtuen tietokoneen, jolla invertterin tietoja halutaan tutkia, täytyy olla samassa verkossa kuin invertteri.

Varastorakennuksessa, johon invertteri asennettiin, ei ollut internet liityntää. Tästä johtuen jouduin rakentamaan invertterille WLAN -vastaanottimen. Varaston läheisyydessä sijaitsevassa talossa oli WLAN -tukiasema, jonka signaali kantoi juuri invertterille saakka.

WLAN -vastaanottimeksi hankin Linksys WET54G Wireless-G Ethernet Bridge -sillan. Siltaan asennin mukana tulleen ympäriseiteilevän antennin tilalle suuntaavan antennin, joka suunnattiin varastolta taloa kohti. Silta tarvitsee antennin lisäksi yksivaiheisen pistorasian, johon sen virtalähde kytketään, sekä verkkokaapelin jolla silta yhdistetään laitteeseen, johon internet yhteys halutaan.



KUVA 7. Internet-yhteys WLAN -reitittimeltä sillan kautta invertterille.

Sillan kytkeminen ei onnistu pelkästään kytkemällä johdot kiinni. Ennen siltausta laitteelle täytyy määrittellä käsin langattoman yhteyden eri parametreja. Sillan asetuksiin pääsin käsiksi kytkemällä sen verkkokaapelilla suoraan tietokoneeseen kiinni. Tämän jälkeen muuttamalla tietokoneen IP-osoitteen verkko ja jakamiskeskuksesta 192.168.1.x (x = 1-255) voi internet-selaimen kautta muuttaa sillan asetuksia. IP-osoite ei saa olla sama, kuin laitteella johon halutaan saada yhteys. Sillan oletus IP-osoite (192.168.1.226) sekä käyttäjätunnus ja salasana löytyivät sillan konfigurointi manuaalista.

Asetuksissa sillalle pitää kertoa käytettävän yhteyden nimi (SSID), käytettävä salaustyyppi ja salausavain. Linksysin selaimen avautuvassa konfigurointinäkyvässä laitteella voi hakea käytettävissä olevia langattomia WLAN -yhteyksiä. Tällöin nimeä ei tarvitse itse syöttää, ja varmistuu, että signaali ylittää varmasti laitteelle asti. Yhteyden salaustyyppi ja salausavain syötetään käsin. Mikäli niitä ei ole muutettu tukiaseman asetuksista, ne löytyvät yleensä tukiasemaan liimatusta tarrasta. Käytettävissä olevia salaustyyppejä olivat WEP, WPA sekä WPA2. Tässä tapauksessa käytössä oli WPA2.

Sillan asetuksia muuttaessa kohtasin useita ongelmia. Käytettävä laite oli ollut pitkään muussa käytössä ja yli 10 -vuotta vanha. Monta kertaa laite meni jumiin syöttäessä sinne tietoja. Laitteen resetoinnin jälkeen sen sai taas toimimaan, mutta kaikki sisälle syötetyt tiedot hävisivät. Pohdimme opinnäytetyön ohjaajan kanssa useaa eri laite vaihtoehtoa, mutta päädyimme kuitenkin lopulta käyttämään reistailevaa vanhaa siltaa. Linksys WET54G on todella pelkistetty silta, jota on kohtuullisen helppo käyttää. Mahdollinen ongelmien lähde ja jumiutumisen syyllinen oli laitteen vanhuus. Laite ei pystynyt kommunikoimaan kunnolla uudempien tukiasemien kanssa, jonka seurauksena se meni jumiin.

The screenshot displays the Linksys WET54G web configuration utility interface. The page is titled "Setup" and includes a navigation menu with options: Setup, Password, Advanced, Status, and Help. A note at the top states: "This screen allow changes to basic settings. Click the **Apply** button to save any changes. If this page does not automatically refresh after clicking **Apply**, then click the **Refresh** button of your web browser."

The interface is divided into two main sections: LAN and Wireless.

LAN Section:

- Firmware:** v 1.05, April 18, 2003
- MAC Address:** (empty field)
- Device Name:** WET54G (You may specify a device name up to 19 characters long.)
- Configuration Type:**
 - Automatic Configuration - DHCP
 - Static IP Address:

IP Address:	192	168	1	226
Subnet Mask:	255	255	255	0
Gateway:	192	168	1	1

The above settings will not be applied if Automatic Configuration - DHCP is selected.

Wireless Section:

- SSID:** linksys (Site Survey button: Search for available wireless network(s).)
- Network Type:** Infrastructure
- Channel:** 6
- Mode:** Mixed
- WEP:** Enable Disable (Edit WEP Settings button)

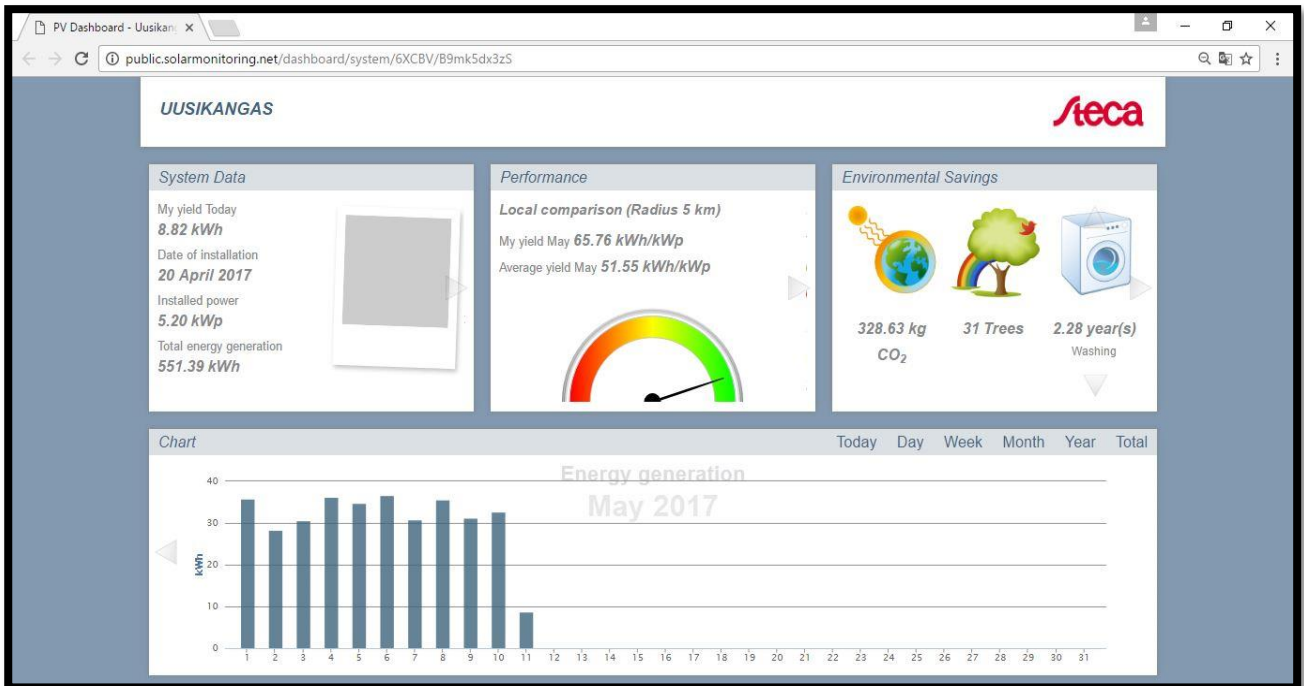
Note: All devices must use the same settings in order to communicate.

Buttons: Apply, Cancel, Help.

Footer: Wireless-G Ethernet Bridge Web Configuration Utility

KUVA 8. Selaimen avautuvat WET54G -sillan konfigurointi asetukset.

Kun invertteri oli yhdistetty internetiin, täytyi valmistajan verkkosivuille rekisteröityä, jotta valmistajan tarjoamaa selainpohjaista seurantasivua pystyi käyttämään. Sivuille tuli tehdä käyttäjä, jolla pääsi myöhemmin tutkimaan invertterin tuottoa ja muokkaamaan laitteiston tietoja. Käyttäjää luodessa täytyi syöttää invertterin sarjanumero, jonka perusteella sivuille päivitetään tiedot oikealta invertteriltä.



KUVA 9. Invertteri valmistajan tarjoaman seurantapalvelun päänäkökulma.

Valmistajan tarjoamat seurantasivut ovat helppokäyttöiset ja selkeät. Sivulta löytyy tuotannon lisäksi tietoa, minkä verran aurinkopaneelien tuottamalla sähköllä on säästetty ympäristöä. Muun muassa kuinka paljon aurinkopaneelijärjestelmän tuottamalla sähköllä on vähennetty hiilidioksidi päästöjä ja kauanko tuotetulla sähköllä olisi käytetty tiettyjä sähkölaitteita, esimerkiksi pyykinpesukonetta.

4 INVERTTERILTÄ SAATAVIEN TIETOJEN TUTKIMINEN

Invertterin sähköntuotantoon liittyviä tietoja olisi varmasti mahdollista tutkia ja kerätä sarjaliikenneväylistä. RS485- ja Modbus-portteihin on mahdollista liittää kolmannen osapuolen energiamittareita, jolloin niihin tulee varmasti invertteriltä tietoa sähköntuotantoon liittyen. Päätin kuitenkin lähteä tutkimaan, onnistuisiko tietoja kerätä millään muulla tavalla.

4.1 Wireshark

Aloitin invertteriltä lähtevän tiedon tutkimisen nauhoittamalla invertterin ja WLAN -tukiaseman välistä tietoliikennettä. Nauhoittamiseen käytin Wireshark -ilmaisohjelmaa, joka on kohtuullisen helppokäyttöinen verkkoliikenteen tarkkailuun ja analysointiin kehitetty ohjelma. Ohjelmalla voi tallentaa kaiken tietokoneen verkkokortin kautta kulkevan tietoliikenteen. Wireshark listaa verkossa liikkuneet paketit aikajärjestyksessä. Jokaisesta paketista tallennetaan lähettäjän sekä vastaanottajan IP-osoite, tiedonsiirtoprotokolla, paketin koko sekä tietoja paketista.

Invertterin valmistajan tarjoaman seurantapalvelun internet-sivuilla sähkön tuotantomäärät näkyivät 10 minuutin tarkkuudella, joten nauhoitin kerrallaan 10 minuutin pätkiä. Nauhoituksia tutkiessa ohjelmassa voi kytkeä suodattimia, jotka suodattavat esimerkiksi vain tietystä IP -osoitteesta saapuneet tai lähteneet tiedot. Kyseinen toiminto tulee tarpeeseen, jos verkossa on paljon liikennettä. Verkossa voi sekunnissa liikkua jopa tuhansia paketteja, joten lista olisi todella työläs tutkittava ilman suodattimia


```

29 2.736395 192.168.1.106 192.168.1.105 TCP 60 80 → 56767 [ACK] Seq=1029 Ack=377 Win=2544 Len=0
30 5.248741 192.168.1.106 192.168.1.105 TCP 60 80 → 56764 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0 MSS=1460
31 5.249001 192.168.1.105 192.168.1.106 TCP 54 56764 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
32 7.265555 fe80::643e:9aff:feb2... ff02::1 ICMPv6 110 Router Advertisement from 8c:79:67:e2:2d:60
33 7.451005 192.168.1.105 216.58.209.109 SSL 55 Continuation Data
34 7.507352 216.58.209.109 192.168.1.105 TCP 66 443 → 56683 [ACK] Seq=1 Ack=2 Win=351 Len=0 SLE=1 SRE=2

```

```

> Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
> Ethernet II, Src: Zte_e2:2d:60 (8c:79:67:e2:2d:60), Dst: IPv6mcast_01 (33:33:00:00:00:01)
> Internet Protocol Version 6, Src: fe80::643e:9aff:feb2:3b7e, Dst: ff02::1
> Internet Control Message Protocol v6

```

```

0000 33 33 00 00 00 01 8c 79 67 e2 2d 60 86 dd 60 00 33.....y g.-`...
0010 00 00 00 38 3a ff fe 80 00 00 00 00 00 00 64 3e ...8:....d>
0020 9a ff fe b2 3b 7e ff 02 00 00 00 00 00 00 00 00 ...;~.....
0030 00 00 00 00 00 01 86 00 c1 69 40 00 00 1e 00 00 .....i@.....
0040 00 00 00 00 00 00 03 04 40 c0 00 00 01 2c 00 00 .....@.....
0050 00 78 00 00 00 00 20 01 14 bb 01 40 27 37 78 88 .x....@'7x.

```

KUVA 10. Osa Wireshark tallenteesta.

Wireshark tallenteista en löytänyt apua tietojen keräämiseen invertteriltä. Tutkiessa tallenteita tuli opinnäytetyön ohjaajalla mieleen, että tiedot voisi yrittää kerätä invertteriltä selaimen kautta. Invertterin selaimen avautuvan käyttöliittymän lähdekoodista saisi tarvittavat tiedot, kunhan vain keksii keinon, millä tavalla.

4.2 Raspberry Pi

Lähdin verkkoliikenteen kuuntelun jälkeen tutkimaan, kuinka tietojen keruu onnistuisi Raspberry Pi:n avulla. Tarkoituksena oli tutkia kuinka invertteriltä onnistuisi kerätä tarpeelliset tiedot selaimelle tehdyn käyttöliittymän lähdekoodista, tallentaa ne väliaikaisesti johonkin ja lähettää edelleen haluttuun tiedostosisajaintiin.

4.3 Lähdekoodi

Tietokoneohjelmien lähdekoodi sisältää ohjelmien käskyt ja toiminnot. Käskyjen avulla tietokone tai ohjelma saadaan toimimaan halutulla tavalla. Lähdekoodista on nähtävissä, kuinka ohjelma toimii. Muokkaamalla lähdekoodia ohjelman toimintaa voidaan muuttaa. Useilla ohjelmointikielillä lähdekoodi on käännettävä erillisellä kääntäjäksi kutsutulla tietokoneohjelmalla suoritettavaan muotoon ennen kuin sitä voidaan käyttää. (Coss.fi 2017.)

Internet sivujen lähdekoodia pääsee tutkimaan selaimella. Internet sivut on rakennettu HTML -koodilla. HTML -koodilla www -selaimelle kerrotaan millaisia elementtejä sivu sisältää, ja millaisena sivu esitetään.

4.3.1 HTML (Hypertext Markup Language)

HTML on avoimesti standardoitu kuvauskieli. Kielellä voidaan kuvata hyperlinkkejä sisältävää tekstiä. HTML -koodi muodostuu sisäkkäisistä ja peräkkäisistä elementeistä. Jokaisessa elementissä on erikseen aloitus- ja lopetustunniste kulmasulkeissa. Kulmasulkeissa olevilla elementeillä kuvataan www-sivuilla näkyvää tekstiä, linkkejä, kuvia yms. Usein HTML -sivuille luodaan jokin tyyli CSS (Cascading Style Sheets) tiedostoon. CSS -tiedosto haetaan HTML -koodin avulla palvelimelta. CSS tiedostossa voidaan kertoa sivun eri elementeille värit, fontit, elementtien sijoittelu, marginaalit ja reunukset, ynnä muut tyylitiedot. (Wikipedia 2017a.)

HTML -koodia voi kirjoittaa esimerkiksi Windows -käyttöjärjestelmän Muistio -sovelluksella. Oikein kirjoitettu ja oikeaan muotoon tallennettu tiedosto avautuu millä tahansa selaimella. HTML -koodin ja www -sivujen tekemiseen on olemassa useita eri sovelluksia. Jopa web -pohjaisiakin sovelluksia löytyy, mutta niiden avulla koodista saattaa tulla sekavaa ja vaikeasti tulkittavaa.



KUVA 11. Yksinkertainen esimerkki HTML -lähdekoodista ja kuva, miltä kyseinen esimerkki näyttää selaimessa.

4.3.2 Javascript

Www-sivujen HTML -koodiin sisällytetään usein Javascriptejä. Javascriptit voivat olla sisällytettynä HTML -koodiin, tai ne voidaan erinäisten komentojen avulla hakea eri sijainnista kuin HTML -koodi. Javascriptiä käytetään yleensä luomaan verkkosivuille dynaamista toiminnallisuutta. Scriptien avulla voidaan esimerkiksi muuttaa verkkosivuilla näkyviä elementtejä, tuoda uusia tai piilottaa jo näkyviä elementtejä. (W3schools.com 2017.)

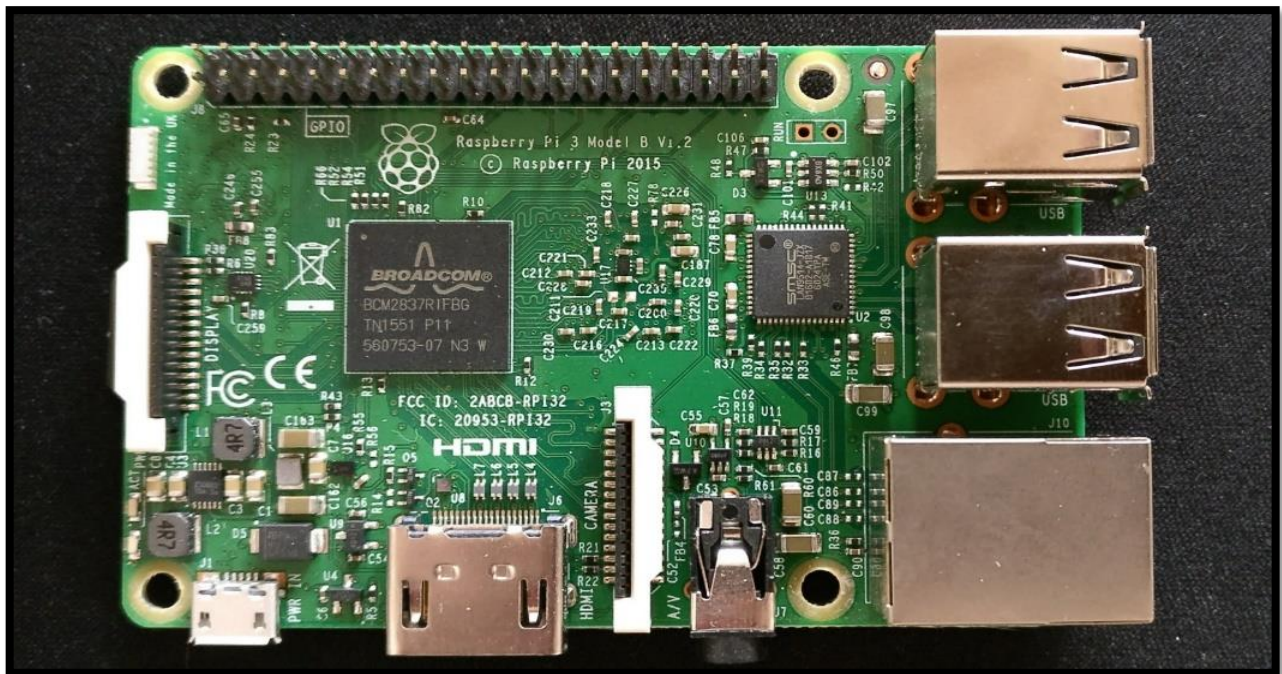
4.4 Raspberry Pi 3

Pi 3 on tällä hetkellä tuorein versio suositusta Raspberry -tuoteperheen yhden piirilevyn tietokoneista. Raspberry Pi 3 on hieman luottokorttia suurempi. Sen on kehittänyt brittiläinen Raspberry Pi Foundation -yhtiö. Yhtiön tavoitteena on saada ihmiset ympäri maailmaa ymmärtämään paremmin digitaalista maailmaa. Halpojen Raspberry Pi -tietokoneiden avulla koodaamisen opettaminen esimerkiksi kouluissa onnistuu kohtuullisen halvalla, eikä vaadi suuria investointeja. Raspberry Pi 3:n hinta on noin 45 euroa. Tämä kolmannen sukupolven tietokone julkaistiin 2016 helmikuussa. (Raspberry 2017.)

1,2GHz 64-bittinen neliydinprosessori	1 Gt RAM muistia
Sisäänrakennetut WLAN sekä Bluetooth	Broadcom VideoCore IV -näytönohjain
4 USB-porttia	HDMI-portti
Ethernet -portti	3,5mm audio/video lähtö
Micro SD korttipaikka	40 GPIO pinniä esim. antureille.

TAULUKKO 1. Raspberry Pi 3 Model B -ominaisuudet.

Raspberry Pi:hin on olemassa useita eri käyttöjärjestelmiä. Rasperryn omien Raspbian sekä NOOBS - käyttöjärjestelmien lisäksi tarjolla on myös kolmansien osapuolien tarjoamia käyttöjärjestelmiä, kuten esimerkiksi Windows 10 IoT Core, Android, Ubuntu MATE ja RISC OS. Omaan testikäyttöön hankittuun Raspberry Pi 3:en asensin Raspbian -käyttöjärjestelmän, jota valmistaja suosittelee aloittelijoille. Käyttöjärjestelmän voi ladata Rasperryn kotisivuilta. Käyttöjärjestelmän .zip tiedosto puretaan tietokoneelle, jonka jälkeen purettu image -tiedosto kirjoitetaan muistikortille. Tiedoston kirjoitin Win32DiskImager -ohjelmalla. Raspberry -tietokoneissa ei ole lainkaan sisäistä muistia, joten niiden käyttöjärjestelmä sekä muut tiedostot tallennetaan muistikortille, joka asennetaan Rasperryn muistikorttipaikkaan.



KUVA 12. Raspberry Pi 3 -piirilevy.

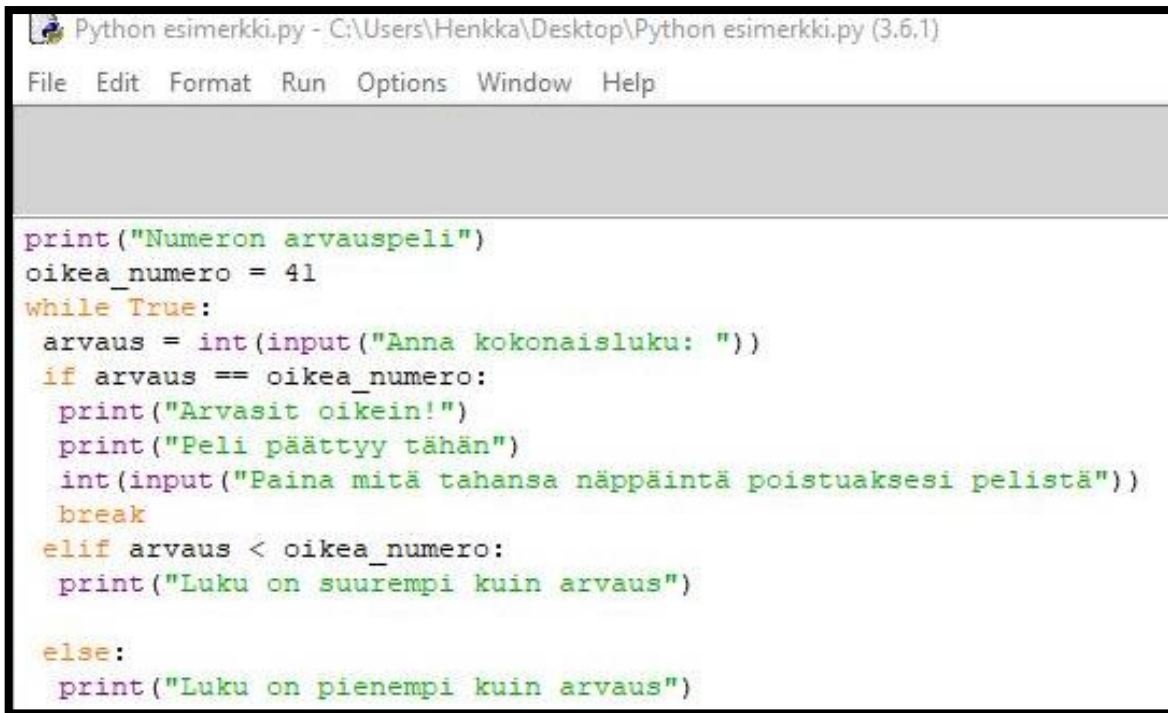
Raspberry Pi:ssä ei ole sisäänrakennettua akkua, eikä paketin mukana tule virtalähdettä. Laitteessa on micro-USB portti, jonka kautta Raspberry saa tarvitsemansa sähkön. Laitteen tarvitseman sähkön syöttöön riittää esimerkiksi matkapuhelimen laturi. Pi käynnistyy heti, kun se liitetään sähköverkkoon. Siinä ei ole erillistä on/off -kytkintä. Laitevalmistaja suosittelee 2 ampeerin laturia, jotta virtaa riittää Pi:n lisäksi myös mahdollisille lisälaitteille, joita GPIO -pisteisiin ja USB -portteihin on mahdollista kytkeä.

4.5 Python -ohjelmointikieli

Raspberryllä on mahdollista ohjelmoida useilla eri ohjelmointikielillä. Laajaan valikoimaan kuuluvat mm. C, C+, Java, Javascript ja Python. Python -ohjelmointikieli laitteesta löytyy valmiiksi asennettuna. Pythonilla on mahdollista tehdä monimutkaisiakin ohjelmia, joita voi ajaa Raspberryllä.

Python on monipuolinen, kohtuullisen helppokäyttöinen ja yksinkertainen ohjelmointikieli. Sitä pidetään helppona opeteltavana yksinkertaisen syntaksin, eli lauseopin takia. Python on tulkettava kieli, ja se on kirjoittamisen jälkeen suoraan valmis ajoon. Python ei siis vaadi erillistä ohjelmaa kääntämään kirjoitettua koodia ennen käyttöä. Python -ohjelmointikieli on syntynyt 1980 -luvun lopussa. Kieli on alun perin kehitetty ABC -ohjelmointikielen jatkoksi. (Wikipedia 2017b.)

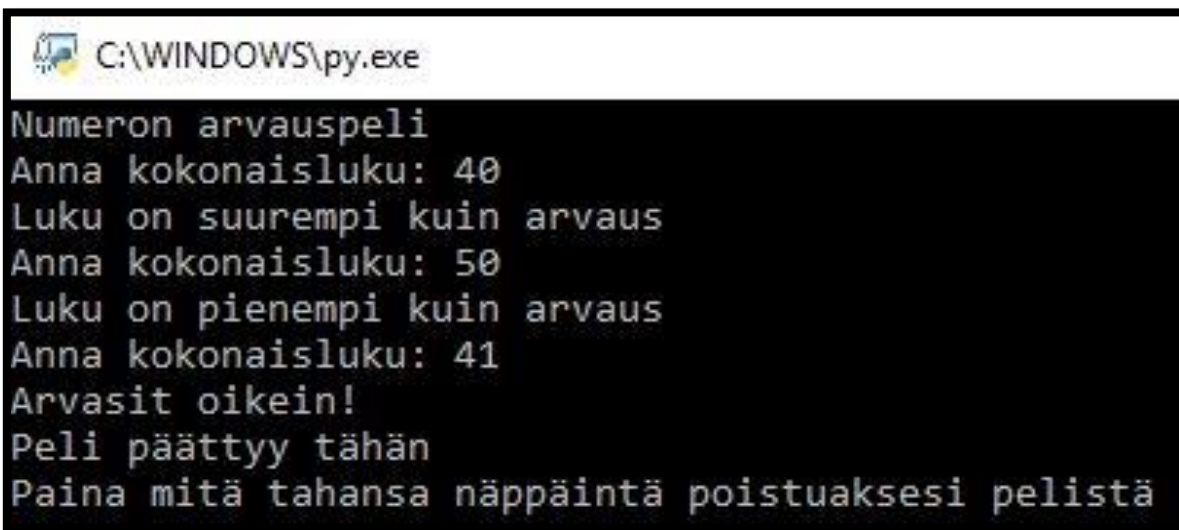
Pythonia voidaan ohjelmoida monilla eri käyttöjärjestelmillä. Esimerkiksi Windowsille voi ladata erillisen Python -ohjelman Pythonin kotisivuilta. Windowsin ohjelmaan sisältyy Python -tulkki, jolla voidaan simuloida ja testata kirjoitettuja ohjelmia.



```
Python esimerkki.py - C:\Users\Henkka\Desktop\Python esimerkki.py (3.6.1)
File Edit Format Run Options Window Help

print("Numeron arvauspeli")
oikea_numero = 41
while True:
    arvaus = int(input("Anna kokonaisluku: "))
    if arvaus == oikea_numero:
        print("Arvasit oikein!")
        print("Peli päättyy tähän")
        int(input("Paina mitä tahansa näppäintä poistuaksesi pelistä"))
        break
    elif arvaus < oikea_numero:
        print("Luku on suurempi kuin arvaus")
    else:
        print("Luku on pienempi kuin arvaus")
```

KUVA 13. Esimerkki Python -koodista.



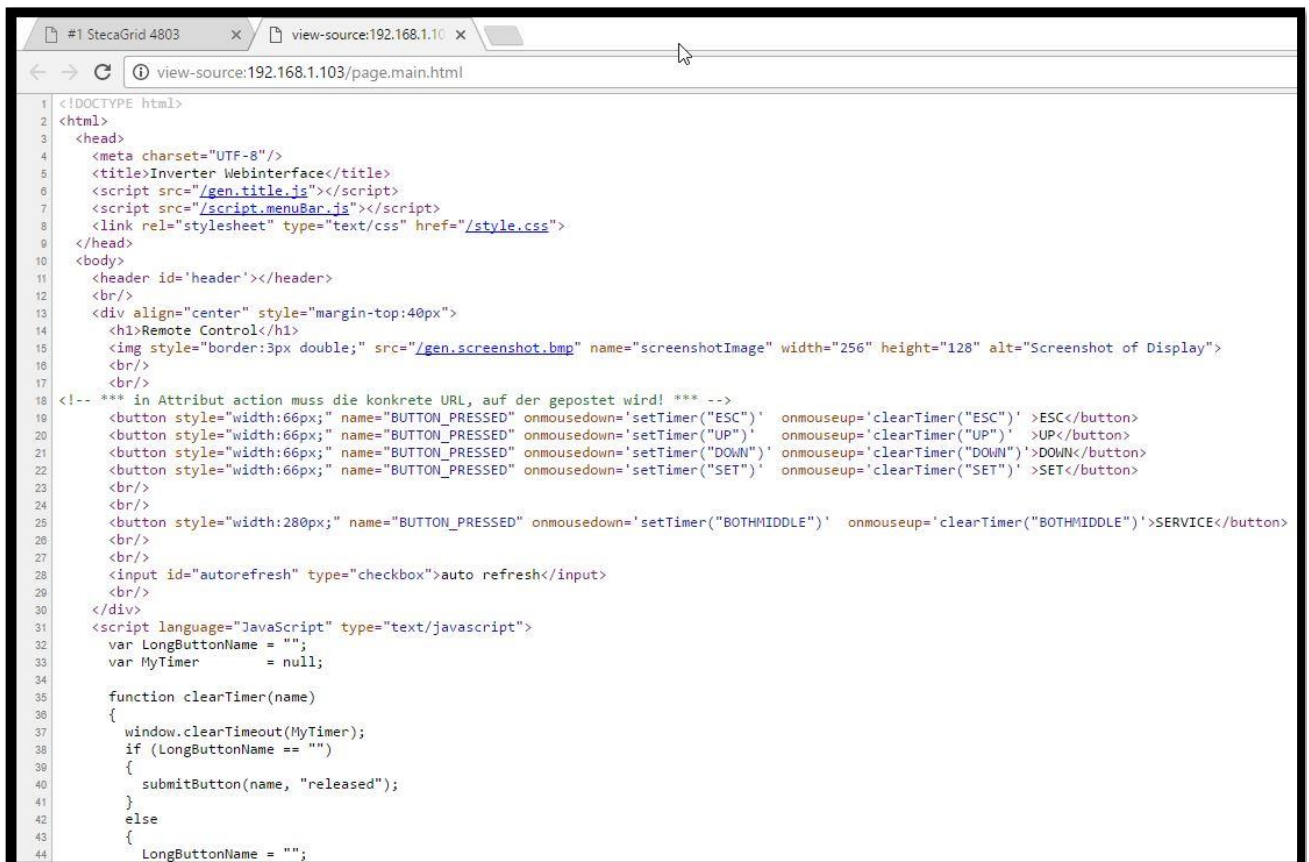
```
C:\WINDOWS\py.exe
Numeron arvauspeli
Anna kokonaisluku: 40
Luku on suurempi kuin arvaus
Anna kokonaisluku: 50
Luku on pienempi kuin arvaus
Anna kokonaisluku: 41
Arvasit oikein!
Peli päättyy tähän
Paina mitä tahansa näppäintä poistuaksesi pelistä
```

KUVA 14. Koodiesimerkki Python -simulointiohjelmalla ajettuna.

5 INVERTTERIN KÄYTTÖLIITTYMÄN LÄHDEKOODI JA PARSEROINTI

Käyttöliittymän HTML -lähdekoodin tutkimiseen käytin kannettavaa tietokonetta ja Google Chrome -selainta. Chromella pääsee tutkimaan internet-sivujen lähdekoodeja painamalla yhtä aikaa Ctrl + U. Sivun lähdekoodi avautuu selaimessa uuteen välilehteen. Käyttöliittymän pääsivun lähdekoodin head -osioon on sisällytetty kaksi Javascriptiä sekä CSS -tiedosto (rivit 6-8). Ensimmäisellä JavaScriptillä (gen.title.js, rivi 6) kerrotaan sivun nimi, joka näkyy selaimen yläpalkissa (1# StecaGrid 4803). Toinen skripti (script.menuBar.js, rivi 7) sisältää komentoja sivun ylhäällä oleville painikkeille (KUVA 15).

Pääsivulta pääsee selaamaan tietoja invertteristä, mutta niiden keräämiseen en keksinyt keinoa. Tiedot näkyvät pienessä ikkunassa, joka on kuvakaappaus invertterissä sijaitsevan näytön kuvasta. Kuvakaappauksesta olisi hankala kerätä tietoja, eikä siitä saisi irti muuta kuin sillä hetkellä näkyvät tiedot. Näyttöön saa kerrallaan näkymään maksimissaan 3 eri arvoa.



```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8"/>
5     <title>Inverter Webinterface</title>
6     <script src="/gen.title.js"></script>
7     <script src="/script.menuBar.js"></script>
8     <link rel="stylesheet" type="text/css" href="/style.css">
9   </head>
10  <body>
11    <header id='header'></header>
12    <br/>
13    <div align="center" style="margin-top:40px">
14      <h1>Remote Control</h1>
15      
16      <br/>
17      <br/>
18      <!-- *** in Attribut action muss die konkrete URL, auf der gepostet wird! *** -->
19      <button style="width:66px;" name="BUTTON_PRESSED" onmousedown='setTimer("ESC")' onmouseup='clearTimer("ESC")' >ESC</button>
20      <button style="width:66px;" name="BUTTON_PRESSED" onmousedown='setTimer("UP")' onmouseup='clearTimer("UP")' >UP</button>
21      <button style="width:66px;" name="BUTTON_PRESSED" onmousedown='setTimer("DOWN")' onmouseup='clearTimer("DOWN")' >DOWN</button>
22      <button style="width:66px;" name="BUTTON_PRESSED" onmousedown='setTimer("SET")' onmouseup='clearTimer("SET")' >SET</button>
23      <br/>
24      <br/>
25      <button style="width:280px;" name="BUTTON_PRESSED" onmousedown='setTimer("BOTHMIDDLE")' onmouseup='clearTimer("BOTHMIDDLE")' >SERVICE</button>
26      <br/>
27      <br/>
28      <input id="autorefresh" type="checkbox">auto refresh</input>
29      <br/>
30    </div>
31    <script language="JavaScript" type="text/javascript">
32      var LongButtonName = "";
33      var MyTimer = null;
34
35      function clearTimer(name)
36      {
37        window.clearTimeout(MyTimer);
38        if (LongButtonName == "")
39        {
40          submitButton(name, "released");
41        }
42        else
43        {
44          LongButtonName = "";

```

KUVA 15. Osa käyttöliittymän päänäkymän (Home) lähdekoodista.

Lähdekoodien head -osio on lähes samanlainen kaikissa käyttöliittymän valikoissa. Head -osioon sijoitetaan JavaScriptien ja CSS -tiedostojen linkit. Joissain valikoissa on muitakin JavaScriptejä, kuin pelkät title.js ja menuBar.js Skriptit. Body -osioon taas sijoitetaan itse sivun rungon muodostava koodi. Body -osio on jokaisella käyttöliittymän sivulla luonnollisesti erilainen.

Events, Update, Info ja Print -valikoiden lähdekoodeista en löytänyt mitään tärkeää tietojen keräämiseen liittyen. Sen sijaan Measurements ja Yield -valikoiden lähdekoodeista löytyi kaikki sähköntuotantoon liittyvät tiedot, mitä invertteriltä on mahdollista saada. Measurements -valikon lähdekoodissa on JavaScripti (gen.measurements.table.js), joka hakee invertteriltä tietoja hetkellisestä sähköntuotannosta, jännitteestä (V), virrasta(A), tehosta (W) ja taajuudesta (Hz, KUVA 16). Tiedoista skripti tekee kuvan 5 mukaisen taulukon. Tiedot sähköntuotannosta näkyvät yhteenlaskettuina, sekä jokaiselta vaiheelta erikseen.

```

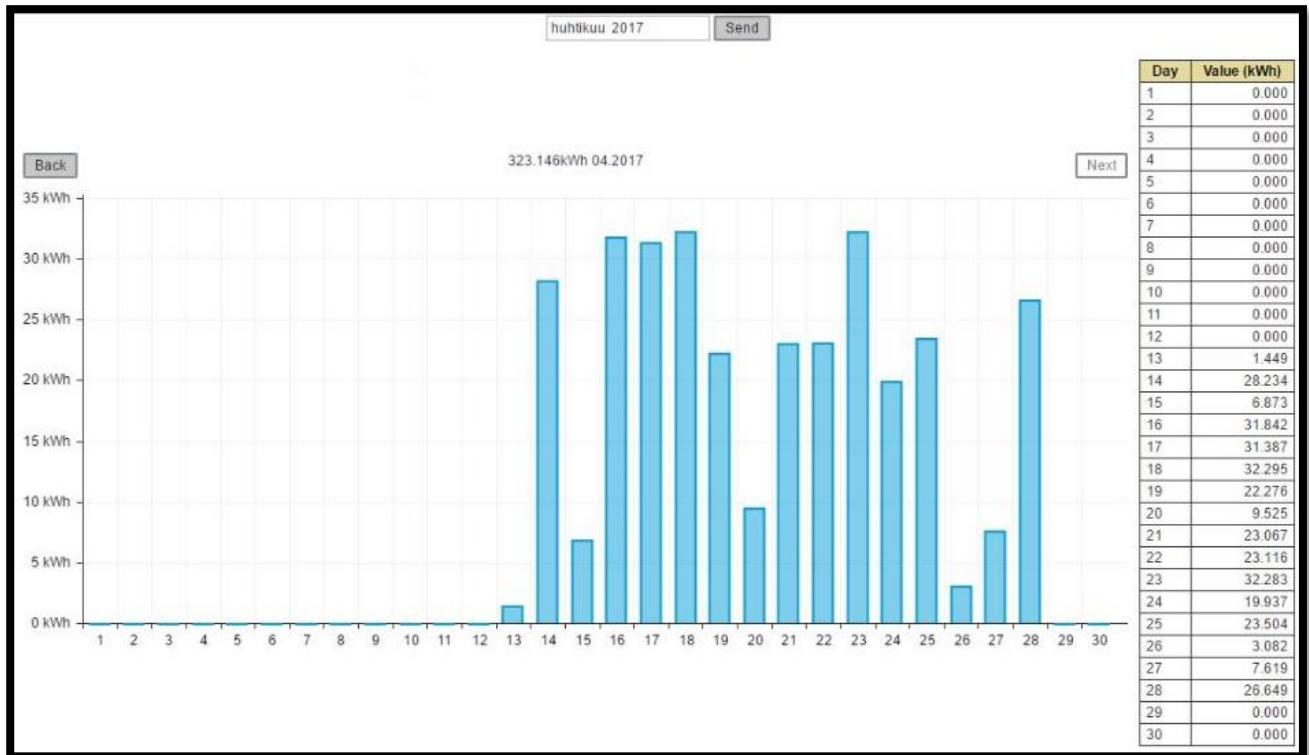
2 <html>
3   <head>
4     <meta charset="UTF-8"/>
5     <title>Inverter Webinterface</title>
6     <meta http-equiv="refresh" content="10" >
7     <script src="/gen.title.js"></script>
8     <!-- <script src="/script.jquery.min.js"></script> -->
9     <script src="/script.menuBar.js"></script>
10    <link rel="stylesheet" type="text/css" href="/style.css">
11  </head>
12  <body>
13    <header id='header'></header>
14    <br/>
15    <div align="center" style="margin-top:40px">
16      <h1>Measurements</h1>
17      <script src="/gen.measurements.table.js"></script>
18    </div>
19  </body>
20  <noscript>
21    <div id="pgm-no-js-text">
22      <p>JavaScript is required to access this website.</p>
23      <p>Please enable JavaScript or use a browser that supports JavaScript.</p>
24    </div>
25  </noscript>
26  <script>createMenu("Measurements"); </script>
27 </html>

```



KUVA 16. Measurements -sivun lähdekoodi ja mittaustulosten JavaScripti.

Yield -sivulta pääsee näkemään invertterin tuotannosta graafisia esityksiä. Kokonais-, vuosi-, kuukausi- ja päivätuotantoa pääsee tarkastelemaan taulukoista sekä pylväs- että viivadiagrammeista. Taulukot ja diagrammit on tehty JavaScriptillä haettujen tietojen pohjalta. Skriptit ovat saman tyyliisiä, kuin edellä esitelty mittaustulosten JavaScripti (KUVA 16).



KUVA 17. Web -käyttöliittymän kuukausituotantonäkymä.

5.1 Lähdekoodin haku ja parserointi

Tutkiessani kannettavalla tietokoneella lähdekoodeja tallensin ne tekstitiedostoiksi tietokoneelle. Lähdekoodin hakemisen Python -ohjelmilla aloitin Windows -tietokoneella. Internetistä löysin muutamia eri vaihtoehtoisia komentoja, joilla lähdekoodin haku www -sivuilta onnistuu. Lyhyen vertailun ja tutkimisen jälkeen päädyin käyttämään urllib -kirjastoa.

5.1.1 URL ja urllib

Pythoniin asennettavasta urllib -kirjastosta löytyy komennot mm. URL -osoitteiden lähdekoodin luke- miseen ja parserointiin. URL (Uniform Resource Locator) on jonkin Internet-resurssin osoite. URL - osoite koostuu käytännössä kahdesta osasta:

1. Protokolla
2. Hakupolku

Esimerkiksi googlen URL -osoitteen `https://www.google.fi` alkuosa `https` on protokolla ja `www.google.fi` hakupolku. Protokolla kertoo, että mihin alueeseen objekti liittyy. HTTP (Hyper Text Transfer Protocol) on yleisesti käytetty protokolla HTML -dokumenteille. HTTPS on HTTP ja TLS/SSL -protokollan yhdistelmä, jossa tiedot salataan ennen lähettämistä. Muita protokollia ovat mm. seuraavat:

- FTP (File Transfer Protocol), tiedonsiirtomenetelmä kahden tietokoneen välille
- MAILTO, käytetään sähköpostiosoitteiden osoittamiseen
- FILE, käytetään paikallisten tiedostojen sijainnin osoittamiseen.

Kaksoispiste (:) erottaa protokollan ja hakupolun toisistaan. Vinoviivoja (/) käytetään objektin sijain- nin tarkentamiseen. Jälkimmäinen osa URL -osoitteesta kertoo objektin sijainnin. Esimerkkinä toimi- nut `https://www.google.fi/` ohjaa google -hakukoneen etusivulle. Kun hakukoneella etsii jotain, muuttuu URL -osoitteen jälkimmäinen osa, eli hakupolku.

Python 3 version urllib kirjasto sisältää useita komentoja URL -osoitteiden tutkimiseen. Esimerkiksi `urllib.request.urlopen()` komennolla voi avata jonkin URL -osoitteen luettavaksi Python -ohjelmalla. Urllib kykenee avaamaan myös tietokoneen kovalevyllä sijaitsevat tiedostot.

5.1.2 Lähdekoodin haku

Invertterin lähdekoodia tutkiessa tein itselle testiympäristön tietokoneen kovalevyllä. Testeissä käytin Pythonin kotisivuilta ladattavaa Python 3.6.1 -version simulointiohjelmaa. Kopion invertterin lähde- koodista löytyvän `measurements.js` JavaScriptin kansioon `E:\Koulu\Oppari\Python testit`. Seuraavaksi

lähdin hakemaan Python ohjelmalla JavaScriptin lähdekoodia. Onnistuin testiympäristössä hakemaan lähdekoodin kuvassa 18 esitetyllä yksinkertaisella ohjelmalla.

```
import urllib.request

url = "file:///E:/Koulu/Oppari/Python testit/measurements.js"
vastaus = urllib.request.urlopen(url)
data = vastaus.read()
print(data)

===== RESTART: E:\Koulu\Oppari\Python testit\Oppari testi.py =====
b'\xef\xbb\xbfdocument.write("<table class=\'invisible\'><tr class=\'invisible\'><th class=\'invisible\'><h3>Inverter</h3></th><th class=\'invisible\'><h3></h3></th></tr><tr class=\'invisible\'><td\r\n<table><tr><th>Name</th><th>Value</th><th>Unit</th></tr><tr><td>P DC</td><td align=\'right\'>3060.75</td><td>W</td></tr><tr><td>U DC</td><td\r\nalign=\'right\'> 61 0.20</td><td>V</td></tr><tr><td>I DC</td><td align=\'right\'> 5.02</td><td>A</td></tr><tr><td>U AC1</td><td align=\'right\'> 236.53</td><td>V</td></tr><tr><td>U AC2</td><td\r\nalign=\'right\'> 232.88</td><td>V</td></tr><tr><td>U AC3</td><td align=\'right\'> 239.14</td><td>V</td></tr><tr><td>I AC1</td><td align=\'right\'> 4.26</td><td>A</td></tr><tr><td>I AC2</td><td\r\nalign=\'right\'> 4.24</td><td>A</td></tr><tr><td>I AC3</td><td align=\'right\'> 4.22</td><td>A</td></tr><tr><td>F AC</td><td align=\'right\'> 50.03</td><td>Hz</td></tr><tr><td>F AC1</td><td\r\nalign=\'right\'> 50.03</td><td>Hz</td></tr><tr><td>F AC2</td><td align=\'right\'> 50.03</td><td>Hz</td></tr><tr><td>F AC3</td><td align=\'right\'> 50.03</td><td>Hz</td></tr><tr><td>P AC</td><td align=\'right\'> 3000.97</td><td>W</td></tr><tr><td>P AC1</td><td align=\'right\'> 1009.75</td><td>W</td></tr><tr><td>P AC2</td><td align=\'right\'> 996.93</td><td>W</td></tr><tr><td>P AC3</td><td align=\'right\'> 999.18</td><td>W</td></tr></table></td><td class=\'invisible\' valign=\'top\' align=\'center\'></table></td></tr></table>");'
```

KUVA 18. Measurements JavaScriptin lähdekoodi Python 3.6.1 -ohjelmalla haettuna tietokoneen kovalevyltä.

Tuloksena sain lähes samanlaisen lähdekoodin, kuin miltä se näyttää selaimella avattuna. Koodi näyttää vielä tässä vaiheessa todella epäselvältä, koska siihen sisältyy taulukoiden eri osia. HTML -elementtien tunnukset <tr> ja <td> kertovat joko rivin tai sarakkeen vaihdosta. Elementtien välistä voi kuitenkin nähdä invertterin mittauskohteita, niiden tuloksia ja yksiköitä. Tämä lähdekoodi muodostaa aiemmin kuvassa 5 esitetyn taulukon, kun sille annetaan oikeat tyylitiedot CSS -tiedostosta.

5.1.3 Lähdekoodin parserointi

Lähdekoodin parseroinnilla tarkoitetaan merkkijonon sisältämien muuttujien ja symbolien käsittelyä. Tarkoituksena on muuttaa merkkijonoa helpommin luettavaksi ja ymmärrettäväksi. Parserointiin löytyy Python ohjelmointiympäristöstä useita työkaluja ja menetelmiä. Päädyn kokeilemaan re -kirjastoa, jolla voi hakea ja jaotella objektista löytyviä tietoja.

Ensimmäisenä lähdin erottelamaan lähdekoodista pelkästään numeroarvoja. Komennolla `'luvut = re.findall(r"[0-9]+", str(data))'` sain lähdekoodista kaikki numeroarvot poimittua jonoksi. Ongelmaksi muodostuivat taulukossa esiintyvät mittauskohteet, mm. vaihekohtaiset jännitteet U AC1, U AC2 ja U AC3, joista ohjelma poimi numerot mukaan. Myös desimaalit jäivät mittaustuloksista pois.

Kokeiltuani useita eri vaihtoehtoja, en saanut mieleistä tulosta millään aikaan. Lopulta tyydyin tulokseen, jossa erottelin ensin lähdekoodista elementtien `<td> </td>` sisällä olevat merkit komennolla `'tulokset = re.findall(r"td(?:.*?)td", str(data))'`. Kyseisen komennon tuloksista erottelin vielä numeroarvot ja pisteet seuraavalla komennolla `'luvut = re.findall(r"[0-9.]+", str(tulokset))'`. Tämä antaa kuvan 19 näköisen tuloksen.

```

===== RESTART: E:\Koulu\Oppari\Python testit\Oppari testi.py =====
['3060.75', '610.20', '5.02', '1', '236.53', '2', '232.88', '3', '239.14', '1',
'4.26', '2', '4.24', '3', '4.22', '50.03', '1', '50.03', '2', '50.03', '3', '50.
03', '3000.97', '1', '1009.75', '2', '996.93', '3', '999.18']
>>> |

```

KUVA 19. Numerotiedot measurements.js lähdekoodista.

Tuloksessa näkyy vieläkin ylimääräisiä numeroarvoja, mm. edellä mainittuihin vaihekohtaisiin jännitteisiin liittyvät 1, 2 ja 3. Nämä eivät kuitenkaan lopullisessa versiossa haitanneet mitään. `Re.findall` komennolla erotellut numeroarvot ovat jokainen omia elementtejään, jotka voidaan hakea myöhemmin erikseen. Näin ollen ylimääräiset numerot voidaan jättää huomioimatta.

6 TIETOJEN TALLENTAMINEN

Tietojen tallennus onnistuu usealla eri tavalla. Tiedot voisi tallentaa esimerkiksi Raspberryyn, samassa verkossa olevaan tietokoneeseen, jonnekin Internet-palvelimelle tai -serverille. Tietojen keruun ja tallennuksen tarkkuutta olisi syytä mieltiä tarkoin, jottei tallenna tietoa turhan tiheään ja näin täytä tiedoille varattua kapasiteettia liian aikaisin. Myöskään tietojen tallennusmuotona ei ole yhtä ja oikeaa ratkaisua.

6.1 Tietojen tallennus testiympäristössä

Testiympäristössä hain Python -ohjelmalla tiedot lähdekoodin JavaScript -tiedostosta, jonka jälkeen tallensin ne ohjelmalla luotuun tekstitiedostoon (KUVA 20). Tekstitiedostona ne on helppo siirtää eteenpäin, esimerkiksi internet -sivuille. Tekstitiedostoa järkevämpi voisi olla tallentaa tiedot Excel -taulukkoon, johon ne voisi listata aikajärjestyksessä. Muun muassa invertterin hetkellisiä arvoja voisi tallentaa esimerkiksi 10-minuutin välein. Tämän lisäksi kokonaistuotantoa, tuotantohuippuja ja katkoksia tuotannossa voitaisiin seurata.



```

import urllib.request
import re

url = "file:///E:/Koulu/Oppari/Python testit/measurements.js"
vastaus = urllib.request.urlopen(url)
data = vastaus.read()

tulokset = re.findall(r"td(?:.*?)td", str(data))
luvut = re.findall(r"[0-9.]+", str(tulokset))
a = luvut

with open("Invertterin tiedot.txt", "w") as tekstitiedosto:

    print ("Paneeliteho =", a[0], "W", file=tekstitiedosto)
    print ("Paneelijännite =", a[1], "V", file=tekstitiedosto)
    print ("Paneelivirta =", a[2], "A", file=tekstitiedosto)
    print ("L1 jännite =", a[4], "V", file=tekstitiedosto)
    print ("L2 jännite =", a[6], "V", file=tekstitiedosto)
    print ("L3 jännite =", a[8], "V", file=tekstitiedosto)
    print ("L1 virta =", a[10], "A", file=tekstitiedosto)
    print ("L2 virta =", a[12], "A", file=tekstitiedosto)
    print ("L3 virta =", a[14], "A", file=tekstitiedosto)
    print ("Taaajuus =", a[15], "Hz", file=tekstitiedosto)
    print ("L1 taaajuus =", a[17], "Hz", file=tekstitiedosto)
    print ("L2 taaajuus =", a[19], "Hz", file=tekstitiedosto)
    print ("L3 taaajuus =", a[21], "Hz", file=tekstitiedosto)
    print ("Invertterin teho =", a[22], "W", file=tekstitiedosto)
    print ("L1 teho =", a[24], "W", file=tekstitiedosto)
    print ("L2 teho =", a[26], "W", file=tekstitiedosto)
    print ("L3 teho =", a[28], "W", file=tekstitiedosto)

```

Tiedosto	Muokkaa	Muotoile	Näytä	Ohje
Paneeliteho				
Paneeliteho				
Paneelijännite				
Paneelijännite				
Paneelivirta				
L1 jännite				
L2 jännite				
L3 jännite				
L1 virta				
L2 virta				
L3 virta				
Taaajuus				
L1 taaajuus				
L2 taaajuus				
L3 taaajuus				
Invertterin teho				
L1 teho				
L2 teho				
L3 teho				

KUVA 20. Testiympäristön sähköntuotantotiedot tallennettuna tekstitiedostoon Python -ohjelmalla.

Tietojen tallentaminen paikalliselle kovalevyille, tai Raspberryn muistikortille onnistuisi samalla tavalla kuin testiympäristössä. Python -ohjelmalle täytyisi vain tehdä komento, joka toistaisi annetun ohjelman tietyin väliajoin, ja tallentaisi haetut tiedot edellisten jatkoksi. Jatkuvaan tallennukseen Excel -taulukko olisi helpommin luettavissa kuin tekstitiedosto. Excel -taulukosta olisi myös kohtuullisen helppoa hakea tietyn päivän tai kellonajan mukaan tallennettuja tietoja.

6.2 Tietojen tallennus Google Spreadsheets

Google Docs on Googlen kehittämä ilmainen toimisto-ohjelma. Ohjelmaan kuuluu mm. tekstinkäsittely- ja taulukkolaskentaohjelma. Taulukkolaskentaohjelma on Googlen palvelussa nimellä Spreadsheets. Spreadsheets -tiedoston avaaminen onnistuu myös Microsoft Excel -ohjelmalla.

Python -ohjelman avulla olisi mahdollista tallentaa invertteriltä saadut tiedot Googlen pilvipalveluun Spreadsheetsiä hyödyntäen. Pilvipalveluun tallennettuun tiedostoon voisi sallia pääsyn vain tietyille henkilöille, jolloin esimerkiksi tietojen haku kotisivuille ei vaatisi erillistä yhteyttä tiedonkerääjän ja kotisivujen haltijan välille. Kotisivujen ylläpitäjä kykenisi hakemaan tiedot Spreadsheets -tiedostosta ja lähettämään ne kotisivuille.

7 YHTEENVETO

Opinnäytetyössäni tavoitteena oli keksiä, kuinka StecaGrid 4803 -invertteristä onnistuisi hakea sähköntuotantoon liittyviä tietoja. Onnistuin keksimään ratkaisun hakea tietoja invertterin selainpohjaisen käyttöliittymän lähdekoodista. Invertterin tietojen hakeminen lähdekoodista onnistuu huomattavasti kolmannen osapuolen tarjoamia energiamittareita halvemmalla. Tietojen keräämiseen riittäisi esimerkki kohteessa pelkästään Raspberry Pi -tietokone pientarvikkeineen.

Tietojen keruuseen tarvittavan ohjelman tekeminen ei loppupeleissä vaadi ihmeitä. Ohjelmointi on paljon internetistä löytyvien valmiiden ohjelmien muokkausta ja testaamista. Valmiita ohjelmia selaimella ja foorumeita lukemalla löytää paljon vinkkejä omia ohjelmia varten. Pidempää ohjelmaa tehdessä sitä kannattaa alkaa ohjelmoimaan pienissä palasissa, jolloin niitä on helpompi testata ja muokata, mikäli jokin osa ei toimi toivotulla tavalla. Invertterin tietojen keruu ohjelmaksi kävisi opinnäytetyössä esitetty ohjelma pienellä muokkaamisella. Jäljelle jäisi enää tietojen tallennukseen ja ohjelman toistamiseen liittyvä ohjelmointi, mikäli haluaisi tehdä lopullisen valmiin järjestelmän, ja testata sitä oikeassa ympäristössä.

Invertterin internettiin yhdistäminen olisi helpompi toteuttaa esim. erillisellä mokkalalla. Tällöin se ei olisi riippuvainen lähellä sijaitsevasta WLAN -verkosta. Näin ollen invertterin seurantaan kehitettyä järjestelmää voisi hyödyntää myös kohteissa, joiden läheisyydessä ei ole mahdollista käyttää olemassa olevia verkkoja. Tällaisia kohteita voisivat olla kauempana sijaitsevat mökit yms. loma-asunnot. Nykyään useat operaattorit tarjoavat liittymiä, jotka ovat tarkoitettuja IoT -laitteiden (Internet of Things) yhdistämiseen internettiin. Nämä ovat peruskuukausimaksultaan edullisia, rajoitetulla tiedonsiirtonopeudella olevia nettiliittymiä. Invertteriltä kerättävän datan lähettämiseen kyseinen liittymä olisi varmasti riittävä.

Lähtiessäni tutkimaan invertterin tietojen keruuta, tuntui tehtävä alkuun lähes mahdottomalta. En ollut aiemmin perehtynyt Python -ohjelmointiin, enkä Raspberryyn ollenkaan. HTML ja siihen liittyvät asiat olivat myös minulle täysin tuntemattomia. Uusien ohjelmointikielien opiskeluun, ja ohjelmointiin yleensäkin auttoi suuresti koulussa opitut asiat, sekä internetistä löytyvät oppaat. Internetissä olevat ohjeet olivat lähes poikkeuksetta englannin kielellä, mikä hieman hankaloitti opiskelua. Pidän sitä kuitenkin tulevaisuuden kannalta parempana vaihtoehtona, että oppii opiskelun yhteydessä englannin kieliset termit ynnä muut sanat.

Opinnäytetyö kokonaisuudessaan oli erittäin opettava projekti. Sen yhteydessä tuli opittua paljon uutta aurinkoenergiaan ja ohjelmointiin liittyen. Työ sisälsi paljon tietokoneella istumista ja internetin selausta, mutta myös hieman käytännön koodaamista ja omin käsin tekemistä. Sain paljon tietoa ja opein taitoja, mitä tulen tulevaisuudessa tarvitsemaan.

LÄHTEET

Coss.fi 2017, Avoin lähdekoodi, saatavilla: <https://coss.fi/avoimuus/avoin-lahdekoodi/> luettu: 20.7.2017

Raspberry 2017, Raspberry Pi 3 Model B, saatavilla: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> luettu: 1.9.2017

Wikipedia 2017a, HTML, saatavilla: <https://fi.wikipedia.org/wiki/HTML> luettu: 1.9.2017

Wikipedia 2017b, Python (ohjelmointikieli), saatavilla: [https://fi.wikipedia.org/wiki/Python_\(ohjelmointikieli\)](https://fi.wikipedia.org/wiki/Python_(ohjelmointikieli)) luettu: 1.6.2017

W3schools.com 2017, JavaScript Introduction, saatavilla: https://www.w3schools.com/js/js_intro.asp luettu: 17.5.2017