

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2021

Rami Takala

MITTAUSDATAN MUUNTAMINEN


TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

Rami Takala

MITTAUSDATAN MUUNTAMINEN

Opinnäytetyön tavoitteena oli ohjelmoida muuntaja, joka muuntaa eri tiedostomuodot yhdeksi valituksi tiedostomuodoksi automaattisesti. Tämän tarkoitus oli yksinkertaistaa Turun ammattikorkeakoulun Health Tech Labilla tehtävää tutkimustyötä, ja siellä on tarkoituksena ylläpitää tietokantaa mittausdatasta, jotka saadaan siellä olevista mittauslaitteista.

Opinnäytetyö käsitteli Polar Ignite -fitnesskellosta saatavien CSV-, TCX- ja GPX-tiedostojen muuntamista JSON-tiedostoiksi automaattisesti. Ohjelmointikielenä toimi Python-ohjelmointikieli ja alustana Raspberry Pi 4 -tietokone, jolle oli alustettu verkkokansiojako.

Muuntaja toteutettiin niin, että Python-ohjelmisto etsi erikseen CSV-, TCX- ja GPX-tiedostoja määritetystä verkkohakemistosta. Tämä mahdollistaa sen, että Windows-tietokoneelta voi siirtää mittausdatatiedosto jaettuun verkkohakemistoon. Tämän jälkeen ohjelmisto muunsi löydetyt tiedostot JSON-tiedostomuotoon käyttämällä jo valmiiksi ohjelmoituja muuntajia, jotka ovat vapaasti käytettävissä, sekä siirsi muunnetun JSON-tiedoston määritettyyn hakemistoon.

Tulevaisuudessa muuntajaan pitäisi lisätä muitakin tiedostomuotojen muuntajia, kuin opinnäytetyössä on mainittu. Myös datan anonymisoinnin lisääminen ohjelmistoon tai erillisenä ohjelmalla on jatkokehityksessä tärkeää, koska käsitellään henkilöstä saatavaa tietoa ja tieto voi olla sellaista, mistä mitatun henkilön voi tunnistaa.

ASIASANAT:

Python, data, muuntaja, JSON, Health Tech Lab, tiedosto, muoto

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2021 | 24 of pages

Rami Takala

CONVERSION OF MEASUREMENT DATA

The aim of the thesis was to program a converter which automatically converts different file formats into one selected file format. The purpose of this thesis was to simplify the research work carried out at Turku University of Applied Sciences' Health Tech Lab, and to maintain a database of measurement data obtained from the measurement devices.

The thesis dealt with the automatic conversion of CSV, TCX and GPX files from the Polar Ignite fitness watch to JSON files. The programming language was the Python programming language, and the platform was a Raspberry Pi 4 computer with network folder sharing.

The converter was implemented so that Python software separately searched for CSV, TCX and GPX files in the specified network directory. This allows the measurement data file to be transferred from a Windows computer to a shared network directory. The software then converted the found files to the JSON file format using already pre-programmed converters that are freely available and moved the converted JSON file to the specified directory.

In the future, other than previously mentioned file format converters need to be added to the converter. The addition of data anonymization to the software or as a separate program is also important in further development, because the measured person can be identified from the already processed information.

KEYWORDS:

Python, data, converter, JSON, Health Tech Lab, file, format

SISÄLTÖ

KÄYTETYT LYHENTEET	6
1 JOHDANTO	7
2 POLAR IGNITE-FITNESS KELLON MITTAUSDATAN ERI MUODOT JA SEN ANONYMISOINTI	8
2.1 Polar Ignite -fitnesskello	8
2.2 Raspberry Pi 4 -tietokone	8
2.3 Python ohjelmointikieli	9
2.4 Mittausdatan eri muodot	9
2.4.1 Mittausdatan määritelmä	9
2.4.2 CSV-tiedosto	10
2.4.3 TCX-tiedosto	11
2.4.4 GPX-tiedosto	11
2.4.5 JSON-tiedosto	12
2.5 Anonymisoitu tieto	13
3 MUUNTAJA JA SEN AUTOMATISOINTI	14
3.1 Python ohjelmointikoodin alustus	14
3.2 Muuntajan ohjelmointi	15
3.2.1 Muuntajan hakemistorakenne	15
3.2.2 Ohjelmiston lokitietojen tallentaminen	16
3.2.3 TCX-tiedoston muuntaminen JSON-tiedostoksi	17
3.2.4 GPX-tiedoston muuntaminen JSON-tiedostoksi	18
3.3 Raspberry Pi -tietokone muuntajana	19
3.3.1 Raspberry Pi -tietokoneen verkkohakemisto	19
3.3.2 Ohjelmiston automatisointi Raspberry ympäristössä	20
4 LOPUKSI	22
LÄHTEET	23

KUVAT

Kuva 1. Esimerkki CSV-tiedoston sisällöstä.	10
Kuva 3. Python ohjelmistokoodin alustus.	14
Kuva 4. Hakemistopolut ohjelmistoa varten.	16
Kuva 5. Time moduulin alustus.	16
Kuva 6. Loki tiedoston alustus.	17
Kuva 7. TCX-tiedoston muuntaminen CSV-tiedostoksi.	17
Kuva 8. CSV-tiedoston muuntaminen JSON-tiedostoksi.	18
Kuva 9. GPX-tiedoston muuntaminen JSON-tiedostoksi.	19
Kuva 10. PiShare moduulin määrittäminen smb.conf tiedostossa. (Delpport 2021.)	20
Kuva 11. Jaetun hakemiston näkymä Windows-tietokoneella.	20

KÄYTETYT LYHENTEET

CSV	Tiedostomuoto, jolla tallennetaan yksinkertaista taulukkomuotoista tietoa tekstitiedostoon. (Comma Separated Values)
GPS	Yhdysvaltain puolustusministeriön kehittämä ja rahoittama satelliittipaikannusjärjestelmä (Global Positioning System)
GPX	Tekstitiedosto, joka sisältää siihen tallennettuja maantieteellisiä tietoja, kuten reittipisteitä, jälkiä ja reittejä (GPS Exchange)
JSON	On yksinkertainen avoimen standardin tiedostomuoto tiedonvälitykseen (JavaScript Object Notation)
SMB	Verkon tiedostojen jakaminen ja data fabric protokolla (Server Message Block)
TCX	XML-tiedosto, jota käytetään harjoitustietojen siirtämiseen kuntolaitteiden välillä (Training Center XML)
XML	Merkintäkielien standardi, joka määrittää tietojen merkintämuodon loogisella rakenteella (eXtensible Markup Language)

1 JOHDANTO

Terveysteknologian laitteiden moninaisuus ja mittausdatan tutkimisen yleisyys on ollut kasvavaa. Laitteita on moneen lähtöön, niin kuumemittarista sormuksiin kuin ranneke-loihin. Laitteista saatava mittausdata on myös erilaisissa tiedostomuodossa, joka hanka-loittaa tutkimista.

Turun ammattikorkeakoulun Health Tech Labille on monia terveysteknologian mittaus-laitteita. Ongelmaksi on ilmennyt, että erilaisista mittauslaitteista saadaan ulos erilaisia tiedostomuotoja. Erilaisten mittaustiedostojen tutkiminen sekä kääntäminen on aikaa vievää, ja siihen halutaan nyt ratkaisu.

Health Tech Labilla on tarkoituksena saada kerättyä mittausdata tietokantaan jatkokäsit-telyä varten. Tämänhetkinen tilanne on, että siellä on kovalevy, mihin tiedostot tullaan tallentamaan. Health Tech Labissa on päätetty, että helpointa on muuntaa kaikki mit-tausdata samaan muotoon, jolloin jatkokäsittely on helpompaa.

Opinnäytetyön tarkoituksena on käsitellä mittausdatan eri tiedostomuotoja ja niiden muuntamista yhteen valikoituun tiedostomuotoon automatisoidusti. Päätetty tiedosto-muoto on JSON, ja Python-ohjelmiston alustana toimii Raspberry PI 4 -tietokone. Tässä opinnäytetyössä käsitellään CSV-, TCX- ja GPX-tiedostoja, jotka on ladattu Polar Ignite -fitnesskellosta ja tiedostojen muuntamista JSON-tiedostomuotoon.

2 POLAR IGNITE-FITNESS KELLON MITTAUSDATAN ERI MUODOT JA SEN ANONYMISOINTI

2.1 Polar Ignite -fitnesskello

Polar Ignite on vedenkestävä fitnesskello, jossa on sykkeenmittaus ranteesta ja integroitu GPS. Ignite on kuntoon keskittynyt älykello, joka opastaa päivittäisessä liikunnassa sisäänrakennettujen antureiden avulla. IPS-näytön ansiosta se on hyvännäköinen kello, joka päiväseen käyttöön. (Martin 2021.)

Polar Ignite tarjoaa tietoa, minkälaista liikuntaa käyttäjän kannattaisi tehdä juuri silloin perustuen uneen ja viimeisempään liikuntasuoritukseen sekä kuntotasoon. Ignitessä on yli sata erilaista liikuntaprofiilia, jotka antavat erilaista tietoa suorituksesta. (Martin 2021.)

Opinnäytetyöhön Polar Ignite -fitnesskello valikoitu, koska tästä kellosta saa paljon erilaista dataa jokaisesta harjoituksesta samanaikaisesti, esim. syke, aika, matka, kesto ja koordinaattipisteitä. Kellossa on sykkeenmittaus suoraan ranteesta, eli ei tarvitse erillistä sykevyötä mittausta varten, mikä helpottaa käyttöä todella paljon.

2.2 Raspberry Pi 4 -tietokone

Raspberry Pi 4 on uusin versio Raspberry Pi -tietokoneesta. Raspberry eroaa paljon muista tietokonelaitteista: Raspberry Pi on yksinkertaisuudessaan luottokortin kokoinen piirikortti, josta löytyy tarvittavat komponentit, jotka löydät myös tavallisesta tietokoneesta tai kannettavasta. (Heath 2019.)

Raspberryllä voi tehdä paljon erilaisia asioita. Käyttäjät rakentavat Pi-levyistä reitittimiä, verkkotason mainosten estäjiä, retropelikonsoleita, mediakeskuksia ja tiedostopalvelimia. Tämä on kuitenkin vain jäävuoren huippu siitä, mikä on Raspberryn kanssa mahdollista. Internetistä löytyy monia projekteja, missä muut ovat rakentaneet Raspberrystä tabletteja, puhelimia, robotteja, kuvien ottamiseen avaruuden reunoilta sekä kokeiden suorittamiseen kansainvälisellä avaruusasemalla. (Heath 2019.)

Pi-tietokoneen käytön aloittamiseen tarvitaan virtalähde, näyttö, hiiri ja näppäimistö, koska Pi-tietokone on vain piirikortti. Helpoin tapa ottaa Pi käyttöön on ladata NOOBS

(New Out-of-Box Software) -asennusohjelma ja asentaa käyttöjärjestelmä ohjeiden mukaan, jotta käytön voi aloittaa. (Heath 2019.)

2.3 Python ohjelmointikieli

Python on korkean tason ohjelmointikieli, jossa on dynaaminen semantiikka sekä se on oliokeskeinen ja tulkittu. Tietorakenteet ovat sisäänrakennettu yhdistettyinä dynaamiseen kirjoittamiseen ja sidontaan, jotka tekevät siitä houkuttelevan sovelluskehityksessä sekä käytettäväksi komentosarjakielenä komponenttien yhdistämiseen. (Python.org 2021.)

Python on helposti opittavan yksinkertainen, syntaksit korostavat luettavuutta ja vähentävät ylläpitokustannuksia. Ohjelmointikieli tukee moduuleita ja paketteja, joka kannustaa uudelleenkäyttöön ja modulaarisuuteen. Laaja standardikirjasto on saatavilla niin binaari- kuin lähdemuodossa, kaikille tärkeimmille alustoille ja ovat vapaasti jaettavissa. (Python.org 2021.)

Yleensä ohjelmoijat tykätsevät Pythonin tarjoamaan tuottavuuden vuoksi. Edit-test-debug-sykli on nopea, koska käännösvaiheita ei ole. Ohjelmiston virheenkorjaus on helppoa, kun huono syöte tai virhe ei koskaan aiheuta segmentointivirhettä. Ohjelmisto aiheuttaa poikkeuksen, kun tulkki havaitsee virheen. Jos tulkki ei havaitse virheitä, tulostetaan pinojälki. (Python.org 2021.)

Paikallisten ja globaalinen muuttujien tarkistamisen mahdollistaminen on toteutettu lähdetason virheen korjaajalla, joka läpikäy koodin rivikerrallaan. Virheen korjaaja on kirjoitettu myös Python-kielellä, mikä kertoo introspektiivisen voiman, jota Python tarjoaa. Nopein tapa usein on lisätä muutama tulostettava lauseke ohjelmistoon, niin tekee edit-test-debug sykli tästä tehokkaan ja yksinkertaisen lähestymistavan. (Python.org 2021.)

2.4 Mittausdatan eri muodot

2.4.1 Mittausdatan määritelmä

Mittausdata on kokoelma erilaisia havaintoja tai mittauksia. Mittausdatan voi jakaa neljään tiedonmittausasteikkoon (nimellistaso, järjestysaste, intervallitaso ja suhdetaso)

Nimellistasolla tarkoitetaan muuttujien nimeämistä ilman määrällistä arvoa ja voidaan vain luokitella, esimerkiksi henkilön sukupuoli. Voidaan luokitella merkitsemällä tiedot poissulkeviin ryhmiin, mutta luokkien välillä ei ole järjestystä. (Bhandari 2020.)

Järjestysasteikolla arvot voidaan luokitella ja järjestää. Data voidaan järjestää ja luokitella järjestyksessä, mutta ei voida sanoa mitään sijoitusten välisistä aikaväleistä. Esimerkiksi voidaan sijoittaa viisi parasta olympiamitalistia, mutta tämä asteikko ei kerro, kuinka kaukana tai lähellä ne ovat voittojen määrässä. (Bhandari 2020.)

Intervallitasolla voidaan datalle päätellä ja luokitella yhtä suuria aikavälejä datapisteiden välille, mutta todellista nollapistettä ei ole. Esimerkiksi kahden vierekkäisen lämpötilan ero on sama: yksi aste. Nolla astetta määritellään eri tavalla asteikosta riippuen. (Bhandari 2020.)

Suhdetasossa datan tiedot voidaan järjestää ja luokitella tasaisesti, sekä niillä on luonnollinen nolla. Todellisella nollalla tarkoitetaan muuttujan absoluuttista puutetta. Esimerkiksi Kelvinin lämpötila-asteikolla ei ole negatiivisia lämpötiloja eli nolla tarkoittaa absoluuttista lämpöenergian puutetta. (Bhandari 2020.)

2.4.2 CSV-tiedosto

CSV (Comma Separated Values) -tiedosto on pelkkää tekstiä sisältävä tiedosto, joka sisältää luettelon tiedoista. Näitä tiedostoja käytetään usein tietojen vaihtamiseen eri sovellusten välillä. Esimerkiksi tietokannat tukevat usein CSV-tiedostoja. (Hoffman 2018.)

Comma Separated Values on suomeksi pilkuilla erotetut tiedot. Ja CSV-tiedosto on juuri nimensä veroinen, eli tiedot on erotettu pilkuilla. CSV-tiedostolla on yksinkertainen rakenne, kun se on luettelo tiedoista, jotka on erotettu pilkuilla. Kuvassa 1. on esimerkki CSV-tiedoston rakenteesta. (Hoffman 2018.)

Nimi,Sähköposti,Puhelinnumero,Osoite
Etu Sukunimi,etu@esimerkki.fi,123-456-7890,123 Testikatu
Mika Testi,mika@esimerkki.fi,098-765-4321,321 Esimerkkitie

Kuva 1. Esimerkki CSV-tiedoston sisällöstä.

Kuvassa 1. näkyy oikeastaan se, mitä CSV-tiedostosta löytyy. Tietenkin tiedosto voi olla monimutkaisempi ja sisältää monia tuhansia riviä, enemmän erilaisia tietoja rivillä tai tekstijonoja. Joissakin CSV-tiedostoissa ei välttämättä ole edes nimettynä yläotsikoita. (Hoffman 2018.)

Tuo CSV-tiedoston yksinkertaisuus on sen ominaisuus. Ne ovat suunniteltu tapa viedä tietoja helposti ja tuoda niitä muihin ohjelmiin. Niitä voidaan selata tekstieditorilla helposti ja ne ovat luettavissa. (Hoffman 2018.)

2.4.3 TCX-tiedosto

TCX-tiedosto on XML-tiedosto, jota käytetään liikuntadatan siirtämiseen laitteiden välillä. Sisältäen tiedot käyttäjän harjoittelusta, esim. juoksun/pyöräilyn reitistä, sykkeestä, poljinnopeudesta, kierrosajasta ja kulutetuista kaloreista. (Fileinfo n.d.)

Garmin julkaisi TCX-tiedoston vuonna 2007 Garmin Training Centerin julkaisun yhteydessä. TCX nimi tulee sanoista Training Center XML. TCX-tiedostot ovat GPX-tiedostojen kanssa samanlaisia mutta TCX-tiedosto tallentaa myös käyttäjän toiminnan tyyppin, kierrosajat, maksiminopeuden, kulutetut kalorit ja liikunnan voimakkuuden. (Fileinfo n.d.)

2.4.4 GPX-tiedosto

GPX-tiedosto on GPS-datatieosto, joka on tallennettu GPS Exchange-muodossa. Monet GPS-ohjelmat käyttävät sitä, koska se on avoin standardi. GPX-tiedosto sisältää pituus- ja leveysasteiden sijaintitietoja, jotka sisältävät jälkiä, reittejä ja reittipisteitä. GPX-tiedostot tallennetaan XML-muodossa, tämä mahdollistaa eri verkkopalveluissa ja ohjelmissa GPS-tietojen lukemisen ja tuomisen. (Fileinfo n.d.)

GPS-sijaintitietojen vaihtamiseen käytetään GPX-tiedostoja, sisältäen geokätkötiedot, reitit ja kartat. Kuntosovellukset ja -laitteet käyttävät GPX-tiedostoja harjoitusreittien tuonnissa ja viennissä. Myös Google Mapsin verkkoversio ja Google Earth Pron työpöytäversio käyttävät ja luovat GPX-tiedostoja. (Fileinfo n.d.)

2.4.5 JSON-tiedosto

JSON tulee sanoista JavaScript Object Notation ja on strukturoidun datan tekstipohjainen, skeematon esitys, joka perustuu järjestettyihin luetteloihin. JSONia tuetaan joko kirjastojen kautta tai alkuperäisesti useimmilla ohjelmointikielillä, niin silti JSON on johdettu JavaScriptistä. Yleisesti JSONia käytetään tietojen vaihtamiseen web-palvelimien ja web-asiakkaiden välillä, mutta JSONia pystyy kuitenkin käyttämään myös muuhun tarkoitukseen. (Freeman 2019.)

JSONsta on levittänyt viimeisten 15 vuoden aikana kaikkialle verkossa. Se on melkein jokaisella julkisesti saatavilla olevalle verkkopalvelulle valintamuotona ja usein sitä käytetään myös verkkopalveluihin yksityisellä puolella. (Freeman 2019.)

Suosio on johtanut monien tietokantojen alkuperäiseen JSON-tukeen. Relaatietokannat, kuten MySQL ja PostgreSQL tarjoavat JSON-tietojen tallentamiseen ja kyselyihin tuen. Myös NoSQL-tietokannat, kuten Neo4j ja MongoDB tukevat JSONia, vaikkakin MongoDB käyttää hieman muunnettua binääristä versiota JSONsta. (Freeman 2019.)

Kuvassa 2. oleva rakenne määrittelee selvästi jotkut henkilön ominaisuuksista sisältäen etu- ja sukunimen, iän ja luettelon lemmikeistä. Tämä sama rakenne voidaan siirtää verkkoselaimeen tai mobiilisovellukseen palvelimelta. Verkkoselain tai mobiilisovellus suorittaa jonkin toiminnon, kuten esimerkiksi tallentaa myöhäisempää käyttöä varten tai vastaavasti näyttää tiedot. (Freeman 2019.)

```
{
  "etuNimi": "Jonathan",
  "sukuNimi": "Freeman",
  "ika": 4,
  "lemmikki": [
    {
      "nimi": "Eetu",
      "tyyppi": "koira"
    }
  ]
}
```

Kuva 2. Esimerkki JSON-tiedoston sisällöstä.

JSON on yleinen tiedostomuoto, jossa on minimaalinen määrä arvotyyppejä: numerot, loogiset arvot, luettelot, objektit, merkkijonot ja nolla. Vaikka JavaScriptin merkintätapa on osajoukko, nämä tyypit ovat kaikilla tavallisilla ohjelmointikielillä, mikä tekee JSONsta hyvä tavan siirtämään tietoja yli kielirajojen. (Freeman 2019.)

JSON-tiedot tallennetaan JSON-päätteisiin tiedostoihin. Tiedostot ovat yksinkertaisia tekstitiedostoja, jotka voidaan helposti avata ja tutkia. Tämä on myös avain, miksi JSONsta on laajempaan yhteen toimivuuteen eri koodikielillä, koska melkein jokainen koodikieli voi lukea ja käsitellä pelkkää tekstiä sisältäviä tiedostoja. Ja ne on helppo lähettää internetin kautta. (Freeman 2019.)

2.5 Anonymisoitu tieto

Kun puhutaan anonymisoidusta tiedosta, tarkoitetaan sellaista henkilökohtaista tietoa, josta yksilöä ei voida tunnistaa. Eli kun tietoa anonymisoidaan, niin yksilöä koskevat henkilökohtaiset tunnisteet ja muut tiedot on poistettava siten, että tietoja ei voida välillisestikään yhdistää yksittäiseen henkilöön. Tällaisenaan saatu tieto ei ole enää luonteeltaan sellaista tietoa, mistä pystyttäisiin tunnistamaan henkilöä. (Komulainen 2018.)

Anonymisointi tulee ottaa myös huomioon tässä opinnäytetyössä, koska Polar Ignite -fitnesskellossakin on käyttäjän henkilökohtaisia tietoja, kuten nimi, ikä, paino ja pituus, joiden perusteella henkilö on tunnistettavissa. Itse harjoituskerrasta saatavassa datassa on näkyvissä nimi, pituus, paino, maksimi syke ja aktiviteetinnimi mukana. Tietenkin myös syke ja koordinaatti pisteet näkyvät harjoittelun datassa, mutta pelkästään sykkeestä ei pystytä suoraan tunnistamaan henkilöä, koordinaatti pisteistä pystytään rajamaan, millä alueella kyseinen henkilö voisi asua. (Polar 2021.)

Jos ladataan koko data, mitä Polar on kerännyt henkilöstä, nähdään pituuden ja painon kehitykset, aktiviteetit sekä henkilön unen mittauksien tuloksen. Tämä koko data pitää erikseen ladata Polarin sivuilta, kun kyseinen henkilö on kirjautunut omilla tunnuksillaan. (Polar 2021.)

3 MUUNTAJA JA SEN AUTOMATISOINTI

Data-tiedostojen muuntaminen tehdään sen takia, että Turku AMK Health Tech Labilla on monia erilaisia mittauslatteita, joista saadaan erilaista dataa mitattavasta henkilöstä. Mitattava data esitellään erilaisilla tiedostomuodoilla ja tähän opinnäyte työhön valikoitui CSV-, GPX- ja TCX-tiedostomuodot. Muuntajan tarkoitus on muuntaa edellä mainitut tiedostomuodot JSON-tiedoksi, jolloin kaikki kerätty data on saman muotoista ja silloin helpompaa jatko käsitellä saatua dataa.

3.1 Python ohjelmointikoodin alustus

Kun aloitetaan muuntajan ohjelmointi, pitää Python-koodille lisätä systeemille spesifejä parametreja ja funktioita. Kuvassa 3. esitellään ohjelmistossa tarvittavat parametrit ja funktiot.

```
import sys
import glob, os
import shutil
import csv, json
from os import listdir
from os.path import isfile, join
import xml.etree.ElementTree as et
from gpx_converter import Converter
import time
from tcx2csv import tcxtocsv
```

Kuva 2. Python ohjelmistokoodin alustus.

”Import sys” -komento sisällyttää sys-moduulin ohjelmiston käyttöön, joka tarjoaa pääsyn joihinkin Python-kielen käyttämiin ja ylläpitämiin muuttujiin ja toimintoihin, jotka ovat vahvasti vaikutuksessa ohjelmointi kieleen. (Python 2021.)

”Import glob, os” -komento sisällyttää glob-moduulin, jota tarvitaan hakemistopolkujen etsimiseen, jotka käyttävät Unix-komennon käyttämien sääntöjen mukaisesti. Os-moduuli tarjoaa tavan käyttää käyttäjärjestelmästä riippuvia toimintoja. (Python 2021.)

"Import shutil" lisää shutil-moduulin, joka tarjoaa useita toimintoja tiedostoille ja tiedostokokoelmille. Erityisesti tarjotaan toimintoja, jotka tukevat tiedostojen kopiointia ja poistamista. (Python 2021.)

"Import csv", CSV-moduuli toteuttaa luokat taulukkotietojen lukemiseksi ja kirjoittamiseksi CSV-muodossa. Sen avulla ohjelmoijat voivat määritellä koodin kirjoittamaan tiedoston tiedot excel-tiedostoon tai lukea tietoja excel-tiedostosta. (Python 2021.)

"From os import listdir" palauttaa luettelon, joka sisältää polun antaman hakemistosta merkintöjen nimet. Luettelo on mielivaltaisessa järjestyksessä, eikä se sisällä erityisiä merkintöjä "." ja "...", vaikka ne olisivatkin osa hakemistoa. (Python 2021.)

"Os.path import isfile, join" isfile-moduuli palauttaa True:n, jos polku on olemassa oleva tavallinen tiedosto. Tämä seuraa symbolisia linkkejä, joten sekä islink(), että isfile() voivat olla totta samalla polulla. Join-moduuli yhdistää yhden tai useamman polkukomponentin älykkäästi. Palautusarvo on polun ja kaikkien polkujen jäsenten ketjutus, jossa on täsmälleen yksi hakemistoerotin jokaista ei-tyhjää osaa lukuun ottamatta viimeistä. (Python 2021.)

"import xml.etree.ElementTree as et" moduuli toteuttaa yksinkertaisen ja tehokkaan API:n XML-tietojen jäsentämiseen ja luomiseen. (Python 2021.)

"from gpx_converter import Converter" antaa jo asennetun gpx_converter moduulin Converter toiminnon ohjelmiston käyttöön. (Baccouri 2021.)

"Import time" moduuli tarjoaa erilaisia aikaan liittyviä toimintoja. Vaikka tämä toiminto on aina saatavilla eivät kuitenkaan ole käytettävissä kaikilla alustoilla. (Python 2021.)

"From tcx2csv import tcxtocsv" moduuli ottaa käyttöön tcxtocsv funktion tiedostosta tcx2csv.py, joka on lisätty ohjelmiston hakemisto rakenteeseen. (Coreysiegel 2016.)

3.2 Muuntajan ohjelmointi

3.2.1 Muuntajan hakemistorakenne

Seuraava määritellään datatiedostojen hakemistot eli se, mistä datatiedosto halutaan hakea ja minne käännetty datatiedosto siirretään.

Kuvassa 4. nähdään, miten hakemistopolut on määritelty, "filePath" määrittää mistä polusta halutaan datatiedostoja etsiä muuntamista varten. "tcxCsvPath" määrittelee apukansion TCX-tiedostojen muuntamiseen JSON-tiedostoksi. Seuraavaksi määritellään, minne mistäkin tiedostomuodosta muunnettu JSON-tiedosto siirretään alkuperäisen tiedoston mukaisesti. Koska opinnäytetyöhön valikoidusta laitteesta tuleva mittausdata yhdestä harjoituksesta sisältää kaikki nämä tiedostomuodot, muuntamisen jälkeen ne ovat samannimisiä, vaikkakin sisältävät erilaista mittausdataa. Tällöin saadaan enemmän irti jokaisesta harjoituksesta saatavasta datasta.

```
# Annetaan hakemisto rakenteet, mistä etsiä ja minne siirretään valmis tiedosto
filePath = "/home/pi/Share/Data/"
tcxCsvPath = "/home/pi/Share/Data/csv/"
tcxJsonFilePath = "/home/pi/Share/Json/tcx/"
csvJsonFilePath = "/home/pi/Share/Json/csv/"
gpxJsonFilePath = "/home/pi/Share/Json/gpx/"
```

Kuva 3. Hakemistopolut ohjelmistoa varten.

3.2.2 Ohjelmiston lokitietojen tallentaminen

Kuvassa 5. alustetaan aika ja päivämäärä muuttujat lokitusta varten. "Now = datetime.now()" antaa muuttujalle now määrittymisenä juuri sen hetkisen kellonajan. "Today = date.today()" antaa muuttujalle today määrittymisenä sen hetkisen päivän. "D1 = today.strftime("%d/%m/%Y")" määrittää muuttujalle d1 missä muodossa muuttuja ilmoittaa päivämäärän. "Current_time = now.strftime("%H:%M:%S")" määrittää muuttujalle current_time, missä muodossa muuttuja ilmoittaa kellonajan.

```
now = datetime.now()
today = date.today()
d1 = today.strftime("%d/%m/%Y")
current_time = now.strftime("%H:%M:%S")
```

Kuva 4. Time moduulin alustus.

Kuvassa 6. ensimmäisenä tyhjenetään tekstitiedosto, jonka nimi on Log.txt, johon koodin edetessä jokainen "print" -komento kirjoitetaan, jotta nähdään, mitä koodin aikana on tapahtunut. Tämän jälkeen Log.txt-tiedosto avataan, jotta koodiin sisällytetyt "print" -komennot kirjataan suoraan Log.txt-tiedostolle. Ensimmäisenä kirjoitetaan lokille, että millä päivämäärällä ja kellonaikana ohjelmisto on viimeksi ajettu. Ohjelmiston edetessä kirjoitetaan lokille, onko muunnettu tiedosto jo olemassa vai onko muunnettu tiedosto siirretty kohde hakemistoon. Lopuksi kun koodi on ajettu kokonaan läpi, tulee kirjoitus "Valmis"

```
#Tyhjenetään log.txt tiedosto
open("/home/pi/Share/Log.txt", "w").close()

#Avataan log.txt tiedosto, johon kaikki print komennot kirjoitetaan
f = open("/home/pi/Share/Log.txt", "a")
sys.stdout = f

#Kirjoitetaan logille minä päivänä ja kellonaikana ohjelmisto on viimeksi ajettu
print("Ohjelmisto ajettu viimeksi " + d1 + " " + current_time)
```

Kuva 5. Loki tiedoston alustus.

3.2.3 TCX-tiedoston muuntaminen JSON-tiedostoksi

Kuvassa 7. aloitetaan ajamalla kansio for silmukkaan ja etsitään tiedostoja, if-lausekkeet määrittelevät koodin edetessä, mitä tiedostoja tällä hetkellä etsitään. Tässä haetaan TCX-loppuisia tiedostoja, jotka nidotaan jonoon, jotta kaikki löydetty tiedostot muunnetaan CSV-tiedosto muodoksi. Ensimmäisenä luodaan CSV-tiedosto käyttäen tiedoston nimeä ottamalla TCX tiedostopäätte pois ja lisäämällä CSV tiedostopäätteeksi. Sen jälkeen tcxtocsv koodi ajaa muuntamisen TCX-tiedostosta CSV-tiedostoksi, jonka jälkeen poistaa muunnetun TCX-tiedoston. (Coreysiegel 2016.)

```
# Etsitään tiedosto hakemistosta
for file in os.listdir(filePath):

    if file.lower().endswith(".tcx"):
        # Jos tiedosto päätte on .tcx tehdään tämä

        tcx = os.path.join(filePath, file)
        tcxfile = os.path.splitext(file)[0]
        csvfile = os.path.join(tcxCsvPath, tcxfile + '.csv')
        # Muutetaan tcx tiedosto csv tiedostoksi
        tcxtocsv(tcx, csvfile)
        # Poistetaan muunnettu tcx tiedosto
        os.remove(tcx)
```

Kuva 6. TCX-tiedoston muuntaminen CSV-tiedostoksi.

Tcxtocsv ohjelmisto on muuntanut TCX-tiedoston CSV-tiedostoksi ja siirtänyt apukansioon on vuorossa muunnetun CSV-tiedoston muuntaminen JSON-tiedostoksi. Kuvassa 8. nähdään, että aluksi annetaan tiedostolle nimi niin kuin edellisessä vaiheessa. Avataan kyseinen CSV-tiedosto ja rivitetään sen sisältämä teksti. Sen jälkeen rivitetty teksti siirretään luotuun samannimiseen JSON-tiedostoon.

```
# Lisäosa tcx muuntimelle, muutetaan luotu csv tiedosto json tiedostoksi
for file2 in os.listdir(tcxCsvPath):
    # Tsekataan onko kansiossa .csv tiedostoja
    if file2.endswith('.csv'):

        csv3 = os.path.join(tcxCsvPath, file2)
        csvfile3 = os.path.splitext(file2)[0]
        # Annetaan siirrettävälle json tiedostolle sama nimi
        jsonfile2 = csvfile3 + '.json'

        # Muunnetaan csv tiedosto json tiedostoksi
        with open(csv3) as f:
            reader = csv.DictReader(f)
            rows = list(reader)

        with open(jsonfile2, 'w') as f:
            json.dump(rows, f)

        # Tarkistetaan onko jo samanniminen json tiedosto olemassa, jos on siirrytään eteenpäin, jos ei siirretään json tiedosto
        if os.path.isfile(tcxJsonFilePath+jsonfile2):
            print("Tiedosto " + jsonfile2 + " on jo olemassa")
            # Poistetaan ylimääräinen json tiedosto ja samalla muunnettu csv tiedosto
            os.remove(jsonfile2)
            os.remove(csv3)
            continue
        else:
            print("Siirretään " + jsonfile2 + " kohde kansioon")
            dest = shutil.move(jsonfile2, tcxJsonFilePath)

        # Poistetaan muunnettu csv tiedosto
        os.remove(csv3)
```

Kuva 7. CSV-tiedoston muuntaminen JSON-tiedostoksi.

Tämän jälkeen tarkistetaan, onko tämäniminen JSON-tiedosto jo olemassa vai ei. Jos samanniminen tiedosto löytyy, poistetaan luotu JSON-tiedosto ja muunnettu CSV-tiedosto sekä siirrytään eteenpäin. Kun samannimistä JSON-tiedostoa ei ole, siirretään luotu JSON-tiedosto oikeaan hakemistoon ja kirjoitetaan lokille tarvittava tieto tapahtumasta. Lopuksi poistetaan muunnettu CSV-tiedosto.

Tätä samaa koodia tullaan käyttämään myös ohjelmiston toisessa kohdassa, jossa muunnetaan laitteesta saadut CSV-tiedosto JSON-tiedostoiksi. Laitteesta saatavassa CSV-tiedostosta ei löydy TCX-tiedostossa olevia koordinaattipisteitä.

3.2.4 GPX-tiedoston muuntaminen JSON-tiedostoksi

Kuvassa 9. ohjelmisto hakee kaikki .GPX-päätteiset tiedostot, laitetaan ne myös jonoon, jotta kaikki tiedostot tulee muunnettua. Tässä osiossa käytetään jo valmista koodia,

jonka pystyy asentamaan suoraan komentokehotteella konsolissa komennolla "pip install gpx-converter". Sisällytys ohjelmistoon vaatii komennon "from gpx_converter import Converter" Python ohjelmiston alustukseen, jotta kyseistä asennettua funktiota pystyy käyttämään. (Baccouri 2021.)

```

if file.lower().endswith(".gpx"):
    # Jos tiedosto päätte on .gpx tehdään tämä
    gpx2 = os.path.join(filePath, file)
    gpxfile = os.path.splitext(file)[0]
    # Annetaan siirrettävälle json tiedostolle sama nimi
    jsonfile2 = gpxfile + '.json'

    Converter(input_file=gpx2).gpx_to_json(output_file=jsonfile2)
    # Tarkistetaan onko jo samanniminen json tiedosto olemassa, jos on siirrytään eteenpäin, jos ei siirretään json tiedosto
    if os.path.isfile(gpxJsonFilePath+jsonfile2):
        print("Tiedosto " + jsonfile2 + " on jo olemassa")
        # Poistetaan ylimääräinen json tiedosto ja samalla muunnettu gpx tiedosto
        os.remove(jsonfile2)
        os.remove(gpx2)
        continue
    else:
        print("Siirretään " + jsonfile2 + " kohde kansioon")
        dest = shutil.move(jsonfile2, gpxJsonFilePath)
    # Poistetaan muunnettu gpx tiedosto
    os.remove(gpx2)

```

Kuva 8. GPX-tiedoston muuntaminen JSON-tiedostoksi.

Muuntamisen jälkeen tehdään tarkistus, onko kyseinen tiedosto jo olemassa vai ei, sen jälkeen siirretään luotu JSON-tiedosto haluttuun hakemistoon jatkokäsittelyä varten, jos sen nimistä JSON-tiedostoa ei ole jo olemassa. Lopuksi poistetaan muunnettu GPX-tiedosto.

3.3 Raspberry Pi -tietokone muuntajana

Raspberry Pi -tietokone on osana opinnäytetyön koodia toimimalla ohjelmiston suorittimena ja tiedon siirtäjänä tietokoneilta Turku Health Tech Labin tietokantaan. Raspberry Pi valikoitu järjestelmäksi sen helppokäyttöisyyden takia. Lisäksi se on pieni ja helppo siirrellä paikasta paikkaan. Tietenkin ohjelmisto pitää muokata sen hetkisen ympäristön mukaan.

3.3.1 Raspberry Pi -tietokoneen verkkohakemisto

Aluksi Raspberryyn tarvitsee asentaa ja konfiguroida tarvittavat asennukset, ennen kuin verkkohakemiston saa jaettua. Ensimmäisenä asennetaan Samba ohjelmistopaketti, joka asentaa SMB protokollan ja antaa tuen yhdistyä Windowsin työryhmäverkkoon. Asennuksen jälkeen avataan smb.conf tiedosto komennolla "sudo nano smb.conf", johon

lisätään "workgroup = WORKGROUP" ja "wins support = yes" rivit, workgroup määrittämiseen tulee sen ympäristön Windowsin työryhmäverkon nimi. Windowsin työryhmäverkon nimi on oletuksena Workgroup. (Delport 2021.)

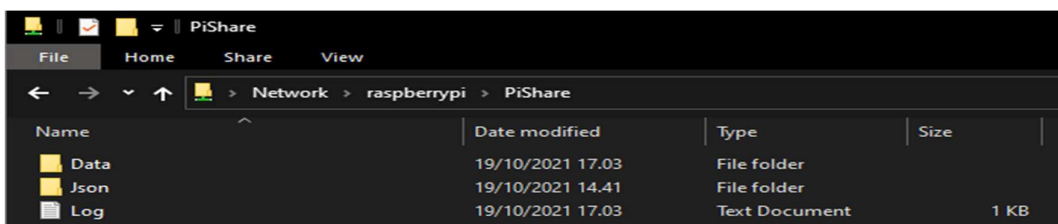
Tämän jälkeen luodaan haluttu jaettava hakemisto "mkdir ~/share" komennolla, jolloin jaettavan hakemiston nimi on share. Kuvassa 10. on avattu uudestaan smb.conf tiedosto ja lisätty PiShare määrittymiset. (Delport 2021.)

```
[PiShare]
comment=Raspberry Pi Share
path=/home/pi/share
browseable=Yes
writeable=Yes
only guest=no
create mask=0777
directory mask=0777
public=no
```

Kuva 9. PiShare moduulin määrittäminen smb.conf tiedostossa. (Delport 2021.)

"Public = no" rivi määrittää, että kun halutaan mennä jaettuun kansioon, vaaditaan kirjautuminen Raspberry Pi tietokoneeseen määritetyillä tunnuksilla. Jos rivi määrittää "Public = yes", kirjautumista ei enään vaadita hakemistoon pääsyä varten.

Lopputuloksena saamme yhteyden haluttuun jaettuun verkkohakemistoon Windows koneelta. Kuvassa 11. on esimerkki, miltä Windows hakemiston näkymä voi näyttää.



Kuva 10. Jaetun hakemiston näkymä Windows-tietokoneella.

3.3.2 Ohjelmiston automatisointi Raspberry ympäristössä

Ohjelmiston automatisointi tehtiin crontab-lisäosalla, asennus tehdään Pi -tietokoneen konsolilla komennolla "pip install python-crontab". Tämän jälkeen avataan crontab

tiedosto komennolla "sudo crontab -e" ja lisätään tiedostoon `"*/10 * * * * /usr/bin/python3 /path/to/script.py"` rivi. (Radečić 2021.)

`"*/10 * * * *"` kohta määrittää, milloin ohjelmisto ajetaan ja luvut tulevat minuutti/tunti/päivä/kuukausi/viikonpäivä. `"/usr/bin/python3"` määrittää Python3-ohjelmiston hakemistopolun, millä ohjelmisto tullaan suorittamaan. Ohjelmiston hakemisto polku `"/path/to/script.py"` vaihdetaan muotoon `"home/pi/Data_converter/dataconverter.py"`, koska ajettava ohjelmisto sijaitsee tuossa hakemistopolussa. (Radečić 2021.)

4 LOPUKSI

Opinnäytetyön tavoitteena oli, miten saadaan rakennettua automaattinen mittausdata-tiedostojen muuntaja. Tuloksena saatiin automaattinen mittausdatatiedostojen muuntaja, joka etsii määritetystä verkkohakemistosta mittausdatatiedostoja ja muuntaa tiedostot haluttuun tiedostomuotoon sekä siirtää valmiit tiedostot määritettyyn hakemistoon kymmenen minuutin välein.

Muuntajan yksinkertainen käyttö on yksi suurimmista onnistumisista, myös ohjelmiston yksinkertaisuus ja kommentointi onnistuivat hyvin. Muuntajan asentaminen Raspberry Pi -tietokoneelle ja verkkohakemiston jakaminen Windows-tietokoneelle oli myös onnistunut valinta. Raspberry Pi -tietokonetta pystyy siirtämään nopeasti ja vaivattomasti paikasta toiseen. Tietenkin silloin pitää tarkistaa verkkohakemiston jako uudelleen, joka tuo lisätyön mukauttamisessa toiseen ympäristöön. Ajan käyttöä olisi voinut käyttää tehokkaammin, koska opinnäytetyön olisi saanut jo keväällä valmiiksi.

TCX- ja GPX-tiedostostoista löytyi tosi vähän tietoa. Näiden tiedostojen tieto perustuu pelkästään yhteen lähteeseen, josta ei kuitenkaan saanut niin paljoa tietoa kuin olisin halunnut. CSV- ja JSON-tiedostoista löytyi paljon erilaisia lähteitä ja oli vaikeaa valikoida juuri se oikea lähde. Kuitenkin valikoitui oikea lähde ja tietoa sai paljon, joka oli myös luotettavaa. Myös Raspberry Pi -tietokoneesta ja Python-ohjelmointikielestä löytyi helposti paljon tietoa, joita pystyi hyödyntämään luotettavasti.

Tuloksia voidaan hyödyntää myös muualla helposti. Ohjelmiston käyttämät valmiit muuntajat ovat julkisia, joten niiden käytölle ei ole estettä. Hyödyntäminen isommassa kuvassa, jossa suora yhteys mittauslaitteesta muuntajalle, on myös mahdollista. Tässä pitää ottaa kuitenkin huomioon myös mittauslaitteen tuottama mittausdatatiedosto ja miten sen saa ladattua.

Ohjelmistoon on paljonkin kehittämismahdollisuuksia. Muiden mittausdatatiedostojen mukauttaminen ohjelmaan ja anonymisoinnin lisääminen pitää ottaa huomioon, kun käsitellään henkilöstä mitattua dataa, joka voi sisältää sellaista tietoa mistä henkilön voi tunnistaa. Ohjelman automatisoinnin kehittäminen, että ohjelmisto on käynnissä taustalla ja kun ohjelmisto huomaa uuden tiedoston määritetyssä hakemistossa käynnistää heti muuntajan, jolloin ei tule turhaa odottelua.

LÄHTEET

Baccouri, N 2021: gpx-converter 1.7.4 27.05.2021 Viitattu 19.10.2021 <https://pypi.org/project/gpx-converter/>

Bhandari, P 2020: Levels of measurement: Nominal, ordinal, interval, ratio. 16.07.2020 Viitattu 09.06.2021 <https://www.scribbr.com/statistics/levels-of-measurement/>

Coreysiegel 2016: tcx-gpx-csv 11.04.2016 Viitattu 19.10.2021 <https://github.com/coreysiegel/tcx-gpx-csv>

Delpont, R 2021: Sharing a Raspberry Pi directory on a local area network 18.01.2021 Viitattu 26.10.2021 <https://behind-the-scenes.net/sharing-a-raspberry-pi-directory-on-a-local-area-network/>

Fileinfo N.d: .GPX File Extension Viitattu 10.05.2021 <https://fileinfo.com/extension/gpx>

Fileinfo N.d: .TCX File Extension Viitattu 10.05.2021 <https://fileinfo.com/extension/tcx>

Freeman, J 2019: InfoWorld. What is JSON? A better format for data exchange 25.10.2019 Viitattu 10.06.2021 <https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html>

Hofman, C 2018: How To Geek. What Is a CSV File, and How Do I Open It? 17.04.2018 Viitattu 10.05.2021 <https://www.howtogeek.com/348960/what-is-a-csv-file-and-how-do-i-open-it/>

Komulainen, J 2018: EDK-2018-AK-229673 - Asiantuntijalausunnon liite 05.12.2018 Viitattu 14.06.2021 <https://www.eduskunta.fi/FI/vaski/JulkaisuMetatieto/Documents/EDK-2018-AK-229673.pdf>

Martin, A 2021: Polar Ignite review: Frustratingly close to brilliance 1.2.2021 Viitattu 15.6.2021 <https://www.expertreviews.co.uk/wearable-technology/1410047/polar-ignite-review>

Polar 2021: Polar Flow Viitattu 08.10.2021 https://support.polar.com/en/how-to-download-all-your-data-from-polar-flow?product_id=63839&category=other

Polar 2021: Polar Ignite Viitattu 09.06.2021 <https://www.polar.com/fi/ignite>

Python 2021: Python Standard Library csv 2021 Viitattu 26.10.2021 <https://docs.python.org/3/library/csv.html>

Python 2021: Python Standard Library glob 2021 Viitattu 26.10.2021 <https://docs.python.org/3/library/glob.html>

Python 2021: Python Standard Library sys 2021 Viitattu 26.10.2021 <https://docs.python.org/3/library/sys.html?highlight=sys#module-sys>

Python 2021: Python Standard Library shutil 2021 Viitattu 26.10.2021 <https://docs.python.org/3/library/shutil.html>

Python 2021: Python Standard Library os 2021 Viitattu 26.10.2021 <https://docs.python.org/3/library/os.html>

Python 2021: Python Standard Library os.path.isfile 2021 Viitattu 26.10.2021 <https://docs.python.org/3/library/os.html>

Python 2021: Python Standard Library xml.etree.ElementTree 2021 Viitattu 26.10.2021 <https://docs.python.org/3/library/xml.etree.elementtree.html?highlight=xml#xml.etree.ElementTree.XML>

Python 2021: Python Standard Library time 2021 Viitattu 26.10.2021 <https://docs.python.org/3/library/time.html>

Python.org 2021: What is Python? Executive Summary Viitattu 1.11.2021 <https://www.python.org/doc/essays/blurb/>

Radečić, D 2021: How to Schedule Python Scripts With Cron 07.05.2021 Viitattu 28.10.2021 <https://towardsdatascience.com/how-to-schedule-python-scripts-with-cron-the-only-guide-youll-ever-need-deea2df63b4e>