

Juuso Valkama

IoT-etäohjauksen soveltaminen sähkösaunan ohjauksessa

Opinnäytetyö

Kevät 2019

SeAMK Tekniikka

Tietotekniikan Tutkinto-ohjelma



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: SeAMK Tekniikka

Tutkinto-ohjelma: Tietotekniikka

Suuntautumisvaihtoehto: Ohjelmistotekniikka

Tekijä: Juuso Valkama

Työn nimi: IoT-etäohjauksen soveltaminen sähkösaunan ohjauksessa

Ohjaaja: Alpo Anttonen

Vuosi: 2019 Sivumäärä: 40 Liitteiden lukumäärä: 3

Esineiden internetin yleistymisen on ollut kovassa nousussa jo vuosikymmenen ajan ja nyt se alkaa saavuttaa jo tavallisia kuluttajia. Älyn lisääntyminen arkipäiväisissä laitteissa tuo uusia mahdollisuuksia ja uusia turvallisuusriskejä.

Tässä opinnäytetyössä tarkoituksena oli soveltaa IoT-tekniikkoja arkipäiväiseen asiaan, tässä tapauksessa saunan kiukaaseen, ja hallita sitä verkkosivun yli. Verkkosivu toimii Raspberry Pillä ja käyttää relemoduulia saunan kiukaan hallitsemiseen.

Työssä tulkittiin lakia tietosuojan ja evästeiden kohdalla, käytettiin neljää eri ohjelmointikieltä ja luotiin käyttäjäystävällinen käyttöliittymä, jolla hallita saunaa. Esineiden internetin historiaa tutkittiin aina internetin synnystä lähti tähän päivään saakka.

Työ oli osittain onnistunut. Vaikka itse järjestelmää ei koskaan fyysisesti asennettu, ohjelmistot ja testaaminen osoitti järjestelmän toimivan kuten sen oli tarkoitettu. Tätä järjestelmää voitaisiin käyttää myös missä tahansa saunassa, jossa on tavallinen kiuas, jos jatkokehitystä halutaan tehdä.

Työn lopussa pohdittiin mahdollisia parannuksia järjestelmään, kuten automaattista kalenterisynkronointia ja antureiden avulla kerättävää dataa. Järjestelmän skaalaaminen suuremmaksi tai pienemmäksi olisi myös mahdollista sen yksinkertaisuuden vuoksi.

Avainsanat: IoT, JavaScript, PHP, HTML, CSS, Raspberry Pi

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Software engineering

Author: Juuso Valkama

Title of thesis: Implementing an IoT-Remote Control System in an Electric Sauna

Supervisor: Alpo Anttonen

Year: 2019 Number of pages: 40 Number of appendices: 3

The Internet of Things or IoT has been on the rise for the last decade and now it is starting to reach consumer markets. These so-called smart appliances bring new possibilities and new security risks.

The goal of the thesis was to apply IoT-technology to an everyday item, in this case a sauna stove, and control it over a website. The website was being run on a Raspberry Pi single-board computer and used a relay module to control the sauna stove.

The thesis studied data protection laws and cookie laws, used four different programming languages and created a user-friendly UI for managing the sauna. The history of the Internet of Things was also studied thoroughly all the way from the birth of the World Wide Web to this day.

The thesis was a partial success. Even though the system was never physically implemented, the programming and testing proved to be working as intended. This system could also be used with any regular sauna stove if further development occurs.

In the end of the thesis, possible upgrades were considered, such as automatic calendar synchronization and data-gathering sensors. Upscaling and downscaling the system could also be possible given the simplicity of the system.

Keywords: IoT, JavaScript, PHP, HTML, CSS, Raspberry Pi

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract	3
SISÄLTÖ.....	3
Kuvaluettelo	6
Käytetyt termit ja lyhenteet	7
1 JOHDANTO.....	9
1.1 Työn tausta	9
1.2 Työn tavoite.....	9
1.3 Työn rakenne	9
1.4 Yritysesittely	10
2 MITÄ ON IOT?	11
2.1 IoT:n historiaa	11
2.2 IoT nykyään ja sen tulevaisuus	13
2.3 Turvallisuus.....	13
3 KOKOONPANO, OHJELMOINTI JA ASENNUS	16
3.1 Osaluettelo.....	16
3.2 Kokoonpanon diagrammi	19
3.3 Käytettävät ohjelmistot ja ohjelmointikielet.....	19
3.3.1 Ohjelmallinen diagrammi.....	20
3.3.2 Visual Studio Code.....	21
3.3.3 Python.....	21
3.3.4 PHP.....	22
3.3.5 JavaScript	22
4 SAUNAN KIUKAAN HALLINTA	23
4.1 Ohjelmallinen puoli	23
4.1.1 Verkkosivusto (HTML/CSS)	23
4.1.2 Verkkosivuston ohjelmallinen puoli (JavaScript)	28
4.1.3 Palvelimen ohjelmallinen puoli (PHP ja Python)	30
4.1.4 Muut ohjelmalliset asiat.....	32
4.2 Asentaminen	34

4.2.1 Raspberry Pi ja XCSOURCE TE213 -relemoduuli	34
4.2.2 Kotelointi ja nappulat.....	35
5 LOPPUYHTEENVETO, POHDINTAA JA JATKOKEHITYSIDEOITA	
.....	36
5.1 Vastoinkäymisiä	36
5.2 Kehitysideoita.....	36
5.3 Loppuyhteenveto.....	37
LÄHTEET	38
LIITTEET	40

Kuvaluettelo

Kuva 1. John Romkey ja Internet-leivänpaahdin.....	12
Kuva 2. The Trojan Room Coffee Pot -IoT kahvinkeitin tarkkailuikkuna.....	12
Kuva 3. Raspberry Pi Model B ja 3D-tulostettu kotelo	16
Kuva 4. XCSOURCE TE213-relemoduuli	17
Kuva 5. Raspberry Pin SD-kortti, WiFi-USB–adapteri ja Bluetooth-adapteri.....	17
Kuva 6. Häätäseis/Tappokytin.....	18
Kuva 7. Kokoonpanon diagrammi	19
Kuva 8. Ohjelmallinen diagrammi	20
Kuva 9. Yksinkertainen versio verkkosivusta saunan ollessa pois päältä.	24
Kuva 10. Yksinkertainen versio verkkosivusta saunan ollessa päällä.....	25
Kuva 11. Lopullinen verkkosivu saunan ollessa pois päältä.....	26
Kuva 12. Lopullinen verkkosivu saunan ollessa päällä.	27
Kuva 13. Raspberry Pin nastojen tiedot.....	35

Käytetyt termit ja lyhenteet

Bluetooth	Lyhyen matkan langaton peer-to-peer-tiedonsiirto.
CSS	Cascading Sheet Style eli verkkosivujen tyylimääritys. Käytetään HTML:n ulkoasun muokkaukseen.
DHCP	Dynamic Host Configuration Protocol, jakaa IP-osoitteita verkossa oleville laitteille.
HTML	Hypertext Markup Language eli visuaalinen ohjelmointikieli, johon kaikki verkkosivut perustuu.
HTTP	Hypertext Transport Protocol, protokolla, jolla HTML-sivujen dataa siirretään internetissä.
Hypesana	Tunnetaan myös nimellä buzzword, iskevä sana, jolla pyritään tuomaan esiin uusia tuotteita ja palveluita
IDE	Ohjelmointiympäristö, eli ohjelma, jolla helpotetaan ohjelmointia visuaalisilla ominaisuuksilla sekä kielitarkistuksella.
IoT	Internet of Things, esineiden internet.
IP-osoite	Verkossa olevan laitteen määrätty osoite.
Kirjasto	Kirjasto eli library, esimerkiksi Pythonissa käytettävä kokoelma funktioita ja metodeja, jotka voidaan tuoda koodiin ulkopuolelta vähentäen tarvittavan ohjelmoinnin määrää.
Rekursiivisuus	Vaikuttaa kaikkiin kansiossa ja alikansioissa oleviin tiedostoihin ja kansioihin.
TCP/IP	Transmission Control Protocol / Internet Protocol. Yleinen tiedonsiirtoprotokolla internetissä.

Viitekehys	Englanniksi framework, muun muassa CSS-tyylimäärittysten kanssa käytettäviä "lisäosia", joilla saadaan ulkoasua helpommin esteettisesti miellyttävämmäksi ja responsiiviseksi.
WiFi	Wireless Fidelity, eli langaton verkko. WiFin alle kuuluvat muun muassa Bluetooth ja WLAN.
WLAN	Wireless Local Area Network, langaton lähiverkko. IEEE 802.11 -standardin mukainen verkkojärjestelmä.

1 JOHDANTO

1.1 Työn tausta

Alkujaan ajatus verkon yli ohjattavasta saunasta lähti vitsistä, että jonain päivänä kaikkia asioita ohjataan internetin välityksellä. Tämän vitsin pohjalta innovoinnin lopputuloksena kehitettiin mahdollinen kehityskohde. Squash & Bowling Centerin miesten sauna on päällä aina keskuksen ollessa auki, mutta naisten sauna on päällä vain silloin kun sitä tarvitaan. Alkuperäinen idea oli rakentaa yksinkertainen nappi, jota painamalla sauna olisi päällä kaksi tuntia, ja sammuisi sen jälkeen. Tämä oli kuitenkin hieman liian helppo toteuttaa, joten päätettiin ottaa suurempi haaste vastaan ja ohjata saunaa verkon yli selaimessa. Yksinkertaiselle napille tuli kuitenkin uusi tarkoitus hätäseis-painikkeena.

1.2 Työn tavoite

Tavoitteena tässä opinnäytetyössä on rakentaa Squash & Bowling Centerille lähiverkon yli toimiva saunanhallintajärjestelmä käyttämällä Raspberry Pi -pienoistietokonetta ja sen avulla ohjattavaa relettä. Tätä varten rakennetaan myös verkkopalvelin Raspberry Pihin ja tarvittavat verkkosivut sekä ohjelmat. Lisäksi suunnitellaan ja tehdään kotelot laitteistolle käyttäen 3D-tulostinta tai muita saatavilla olevia vaihtoehtoja.

1.3 Työn rakenne

Työ koostuu kolmesta pääkohdasta; Ohjelmistojen luomisesta, ohjelmistojen asentamisesta sekä fyysisestä asentamisesta. Lisäksi tutkitaan IoT:n historiaa ja selvitetään, mitä IoT oikeasti on.

1.4 Yritysesittely

SQB-keskuksella on Seinäjoella toimiva vuonna 2013 perustettu vapaa-aika-, urheilu- ja viihdekeskus. Nimensä mukaan SQB-keskuksella on squash-kenttiä neljä kappaletta ja 12 Brunswick-keilarataa sekä kuntosali saunatiloineen ja solarium. Lisäksi SQB-keskuksella on Seinäjoen Kapernaumissa Tuottajantiellä sulkapallohalli, jossa on viisi sulkapallokenttää. (Squash & Bowling Center [Viitattu 2.5.2019].)

2 MITÄ ON IOT?

IoT eli Internet of Things tarkoittaa suomeksi esineiden (ja asioiden) internetiä. Yksinkertaistettuna IoT on arkipäiväisen esineen liittämistä verkkoon ja sen hallinnointia verkon yli. IoT liittyy myös vahvasti tiedonkeruuseen ja tiedon analysointiin; kuinka usein laitetta käytetään, millä aikavälillä laitetta käytetään, kuinka pitkään laitetta käytetään kullakin käyttökerralla. Tietoa analysoimalla voidaan koneoppimista hyödyntämällä oppia ennakoimaan käyttöä ja tekemällä esivalmisteluja käyttöä varten. (What Is IoT? – A Simple Explanation of the Internet of Things 6.1.2019.)

Teollisella puolella on jo vuosikaudet ollut olemassa teollisuusautomaatiota joka ei itsessään ole esineiden internetiä vaikka aina välillä sitä yritetäänkin markkinoida sinä. Tässä tapauksessa IoT on vain hypesana, jolla pyritään luomaan kiinnostusta yritystä kohtaan. Teollisuusautomaation ja IoT:n tärkein ero on jo aikaisemmin mainittu tiedonkeruu ja datan hyödyntäminen. Tiedonkeruu on kuitenkin helppo lisätä teollisuusautomaation laitteisiin antureiden avulla, ja on hyötyynsä nähden yleensä hyvin pieni investointi. (What is Industrial Automation? 15.10.2013.)

2.1 IoT:n historiaa

Yleisesti ensimmäisenä IoT-laitteena pidetään John Romkeyn internet-leivänpaahdinta. Tämä tapahtui vuonna 1989 eli hyvin samoihin aikoihin kuin WWW (World Wide Web) alkoi muodostua. Kyseessä oli tavallinen leivänpaahdin, jonka virtoja pystyi kytkemään päälle ja pois TCP/IP-yhteyden yli. Leivänpaahdinten liittäminen internettiin lähti vitsimäisenä haasteena, mutta Romkey tarttui tähän, ja loi samalla kysymyksen, johon tänä päivänäkin etsitään uusia vastauksia: "Mitä kaikkea voidaan liittää internetiin?" (The Internet Toaster 7.1.2000.)



Kuva 1. John Romkey ja Internet-leivänpaahdin. (The Internet Toaster 7.1.2000.)

Seuraava suuri askel IoT:n maailmassa oli The Trojan Room Coffee Machine. Tämä kahvinkeitin ei ollut itsessään mikään sähkötekniillisesti erityinen keksintö, mutta se oli tiedonkeruun kannalta tärkeä. Kyseessä oli Quentin Stafford-Fraserin vuonna 1992 rakentama kahvinkeitin tarkkailujärjestelmä, jossa kamera otti kuvan kolmisen kertaa minuutissa verkkoon, ja sitä pystyttiin tarkkailemaan yrityksen tietokoneilta. Stafford-Fraserin sanojen mukaan hänen yksikössään työskenteli 15 tutkijaa ja heillä oli vain yksi kahvinkeitin, joten kuvan laittaminen sisäverkkoon kertoi oliko kahvia vai ei. Kamera toimi näin ollen anturina ja ihmiset käsittelivät tietoa. (The Trojan RoomCoffee Pot 1995.)



Kuva 2. The Trojan Room Coffee Pot, IoT-kahvinkeitin tarkkailuikkuna. (The Trojan RoomCoffee Pot 1995.)

Vuosituhannen vaihteessa tapahtui paljon edistystä, vuonna 1999 Kevin Ashton mainitsi ensimmäistä kertaa termin ”Internet of Things” esityksessään (That ’Internet of Things’ Thing 22.6.2009) ja vuonna 2000 LG julkisti ensimmäisen internet-jääkaappinsa. (No need for PCs with intelligent fridges 13.12.2000). Cisco IBSG arvioi IoT:n ”syntymäpäivän” olevan vuosien 2008 ja 2009 välillä, jolloin IoT-laitteita oli enemmän kuin ihmisiä. (Evans 2011.)

2.2 IoT nykyään ja sen tulevaisuus

Internet-jääkaapin jälkeen arkipäiväisiin laitteisiin on lisätty IoT-ominaisuuksia paljon. Yhdysvaltalainen Griffin julkisti CES-tavaramessuilla 2017 Griffin Connected –malliston, johon kuuluu kahvinkeitin, leivänpaahdin sekä älypeili. Näistä kahvinkeitin ja leivänpaahdin on Bluetoothin kautta yhteydessä mobiililaitteisiin sekä älypeiliin. Älypeili on taas kiinni talouden WiFi-verkossa, mikä mahdollistaa etäkäytön asunnon ulkopuolelta. (Griffin Technology Unveils Griffin Home 14.1.2017.)

Mobiililaitteiden yleistymisen jälkeen IoT on siirtynyt hypesanasta seuraavaksi askeleeksi kohti tieteisfiktion tulevaisuutta, jossa kaikki laitteet ovat yhteydessä toisiinsa. Jääkaappi voi kertoa että, täältä puuttuu munia, jolloin se tilaa munia verkkokaupasta, ja munat toimitetaan kotiovelle. Vessanpönttö voi tutkia virtsan ja ulosteen koostumusta, päätellä sen perusteella, onko käyttäjä kenties raskaana tai kipeänä, ja ottaa yhteyttä terveydenhoitoon. Se mikä kuulosti täysin absurdilta ja järjenvastaiselta 20 vuotta sitten, on kohta jo arkipäivää. (Tokyo Toilets deliver data dump by very personal RSS. [Viitattu 23.4.2019].)

2.3 Turvallisuus

Kun puhutaan internetin yli toimivista laitteista, jotka vaikuttavat arkipäivän elämään niinkin henkilökohtaisesti kuin kodin IoT-laitteet, tulee väistämättäkin mieleen kysymys: ”Entä jos?”. Entä jos joku hakkeroi laitteet ja kerää tietoja? Entä jos joku murtautuu järjestelmään ja tilaa omistajan nimissä rekkakuormallisen munia? Entä

jos joku polttaa saunan lähettämällä jonkun koodatun skriptin? Nämä ovat kaikki täysin valideja uhkia ja samalla syy miksi tehtaiden IoT-järjestelmät ovat käytännössä kaikki suljetussa sisäverkossa. IoT-laitteet ovat siis yhteydessä pääpalvelimeen, joka taas on yhteydessä internetiin. Pääpalvelimessa tai reitittimessä/kytkimessä on omat palomuurinsa, jotka estävät ulkopuoliset luvattomat yhteydenotot. Teollisuudessa on ihan ymmärrettävää että raskas kalusto vaatii raskaat turvatoimet, esimerkiksi hitsausrobotti voi tehdä pahaa jälkeä työntekijöille ja laitteille. Henkirikosten mahdollisuus on myös olemassa, sillä nykyään on jo olemassa sydämentahdistimia, jotka on liitetty verkkoon. (FDA confirms that St. Jude's cardiac devices can be hacked 9.1.2017.)

Kodeissa ei yleensä ole pääpalvelinta tai erillisiä palomuuureja, joten suojaus täytyy toteuttaa jollain muulla tavalla. Yksi varmimmista tavoista suojata IoT-laitteet on luoda niille oma verkkoympäristö, eli periaatteessa sama kuin teollisuudessakin. Tähän toki saatetaan tarvita toinen reititin tai palvelin väliin, joten se ei välttämättä ole maallikolle sopiva ratkaisu. Toinen ratkaisu on liittää kaikki IoT-laitteet vierasverkkoon, jossa niillä on vähemmän oikeuksia toimia kodin verkossa.

Turvallisuuteen liittyy myös toimintavarmuus, joka IoT-laitteiden kohdalla koostuu manuaalisista ohjaimista, joilla voidaan laitetta käyttää vikatilanteen selvittämiseksi käsin. Myös pääkytkimet tulisi olla selkeästi näkyvillä. Mitä tulee muihin hallintalaitteisiin, älypuhelin on ideaalilaitte IoT-järjestelmien tarkkailuun ja hallintaan johtuen sen käyttöliittymän helppoudesta ja käyttöjärjestelmän takaamasta turvallisuudesta. Esimerkiksi Applen iPhone -sarjan puhelimissa on pitkälle kehitetty turvallisuuskerros perustuen sen suljettuun järjestelmään (laitteeseen voi ladata vain hyväksytyjä sovelluksia vain yhdestä paikasta) ja pitkään kehitykseen. (Gilchrist 2017, 29.)

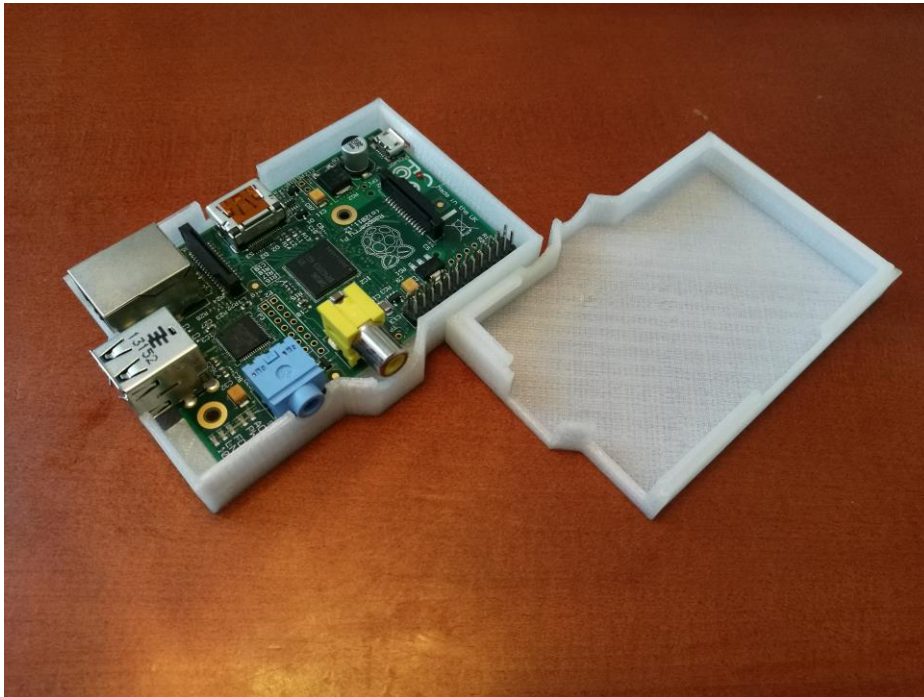
IoT:n turvallisuuteen kuuluu myös yksi oleellinen elementti, nimittäin luotettavuus. Tämä luotettavuus on käyttäjän, kehittäjän ja valmistajan välistä luotettavuutta, eli vuotaako joku tietoa tai kerääkö joku tietoa toisen tietämättä. Tämä on itsessään IoT:n pelottavimpia puolia. Sillä aina kun kerätään tietoja, voidaan kerätä henkilökohtaisia tietoja, mikä saattaa rikkoa henkilötietoja koskevaa EU-maiden GDPR -asetusta. (Usein kysyttyä EU:n tietosuojaa-asetuksesta [Viitattu 20.4.2019].) IoT:n konsepti perustuu luotettavuuteen ja tiedon jakoon, joten jos luotettavuus

menee, niin yrityksen maine kärsii. Datamäärän kasvaessa myös tietomurtojen vakavuus kasvaa, 2015 vuoden lopulla kaksi tietomurtoa paljasti satojen tuhansien lasten henkilötietoja ja tietokannan rakenteesta johtuen myös heidän vanhempiansa henkilötiedot osoitteineen. Näiden tietomurtojen takia tulisi aina rakentaa järjestelmät mahdollisimman yksinkertaisiksi, mutta silti tietoturvatuiksi. (Glichrist 2017, 169-174.)

3 KOKOONPANO, OHJELMOINTI JA ASENNUS

3.1 Osaluettelo

Tässä työssä käytettiin Raspberry Pi Model B:tä. Se on hieman vanhempi versio, mutta riittää tähän projektiin.



Kuva 3. Raspberry Pi Model B ja 3D-tulostettu kotelo

XCSOURCE TE213 -relemoduuli. Tämä on geneerinen kiinalainen relemoduuli yhdellä releellä. Vastaavia relemoduuleja löytyy myös useammalla releellä.



Kuva 4. XCSOURCE TE213 -relemoduuli

Työssä käytettiin myös SD-korttia, WiFi-USB adapteria ja Bluetooth-adapteria. Vanhemmat Raspberry Pit käyttävät täysikokoista SD-korttia, kun taas uudemmat käyttävät puhelimista tutumpaa Micro-SD-korttia. WiFi-USB-adapteri on kiinalainen 802.11n-standardin USB-laite ja Bluetooth-adapteri on myös kiinalainen Bluetooth 4.0 -standardin USB-laite. Tässä projektissa Bluetoothia ei käytetä, mutta se on silti asennettuna mahdollista päivitystä varten.



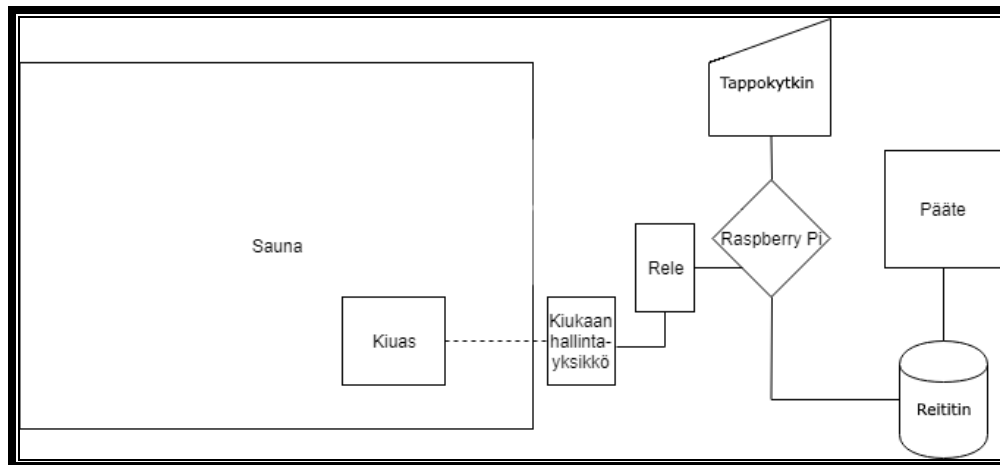
Kuva 5. Raspberry Pin SD-kortti, WiFi-USB-adapteri ja Bluetooth-adapteri

Työssä on myös Hätäseis/Tappokytin. Se on suuri, kymmenen senttimetriä halkaisijaltaan oleva punainen painike lampulla varustettuna ja varmasti huomattavissa pimeässäkin tilassa.



Kuva 6. Hätäseis/Tappokytin.

3.2 Kokoonpanon diagrammi



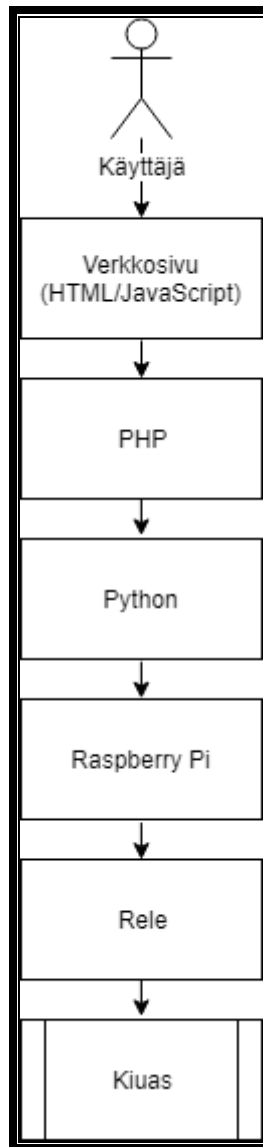
Kuva 7. Kokoonpanon diagrammi

3.3 Käytettävät ohjelmistot ja ohjelmointikieliet

Tässä opinnäytetyössä käytetään suhteellisen montaa eri ohjelmointikieltä, mikä sinänsä on yleistä kun yritetään yhdistää internetiä ja fyysistä maailmaa toisiinsa. Selainohjelmointi on tarkoitettu käyttäjän nähtäväksi ja koettavaksi, joten sillä on vaikea suoraan lukea laitteista tietoa. Laiteohjelmointi on tarkoitettu laitteen hallintaan ja se ei välttämättä ole esteettisesti miellyttävää lukea raakana datana selaimelta. Tämän takia tarvitaan useampaa eri ohjelmointikieltä.

3.3.1 Ohjelmallinen diagrammi

Kuvassa on yksinkertaistettu näkymä, kuinka ohjelmallinen puoli rakentuu työssä, ja kuinka sivuston ohjelmallinen rakenne välittää käskyä käskyä eteenpäin seuraavalle tasolle.



Kuva 8. Ohjelmallinen diagrammi

3.3.2 Visual Studio Code

Visual Studio Code on tällä hetkellä suosituin ohjelmointiympäristö ja sen suosion taustalla on automaattinen koodintäydennys ja syntaksin korostus, jotka toimivat sadoilla eri ohjelmointikielillä (Most Popular Development Environments, 2019.) Tuettujen ohjelmointikielien määrä kasvaa jatkuvasti johtuen Visual Studio Coden yhteisön panostuksesta itse ohjelmaan. Muita hyviä puolia Visual Studio Codessa on sen vakiona oleva tumma teema, joka säästää silmiä, ja siinä on suhteellisen looginen ja intuitiivinen käyttöliittymä. Suurin syy, miksi Visual Studio Code on suosituin, on kuitenkin sen hinta: Visual Studio Code on ilmainen. (Visual Studio Code [Viitattu 4.4.2019].)

3.3.3 Python

Python on Linux-maailman käytetyimpiä ohjelmointikieliä ja Raspberry Pin käyttämä Debian Linux -pohjainen Raspbian -käyttöjärjestelmä tukee sitä todella hyvin. Pythonista on olemassa kaksi käytössä olevaa versiota, vanhempi 2.7 ja uudempi 3. Syy miksi käytössä on kaksi eri versiota johtuu versioiden välisestä yhteensopimattomuudesta, ja siitä että vanhemman version ohjelmien päivittäminen uudempaan versioon on todella työlästä. Versiomuunnokseen on olemassa työkalu joka muuntaa osan koodista uudempaan versioon mutta se ei ole täydellinen. Ohjelmointikielenä Python on todella nopea oppia sen loogisuuden vuoksi ja valmiiksi tehtyjen kirjastojen avulla. Mihinkään suurempaan ohjelmistoprojektiin Pythonia ei suositella sen tulkattavuuden vuoksi, eli tietokone joutuu tekemään ylimääräistä työtä, jotta laitteisto ymmärtää kieltä. Pienissä projekteissa Python on kuitenkin parhaita vaihtoehtoja. (Why Python is Popular Despite Being (Super) Slow [Viitattu 23.4.2019].)

3.3.4 PHP

PHP on vuonna 1994 luotu ohjelmointikieli josta on muodustunut yksi internetin kulmakivistä. Käytännössä PHP pystyy toimimaan välikätenä verkkosivujen ja tietokantojen välillä sekä käskemään palvelimen suorittaa ohjelmia ilman että käyttäjä näkee varsinaista koodia koskaan. Tämä on tietoturvan nimissä ollut pitkään parhaita ratkaisuja, mutta myöhemmin sen turvallisuus on kyseenalaistettu useampaan otteeseen muiden ohjelmointikielien kehittyessä. Lähestulkoon kaikki päivitykset tällä vuosikymmenellä ovat liittyneet jollain tavalla PHP:n turvallisuuteen. (PHP 23.4.2019.)

3.3.5 JavaScript

JavaScript on verkkoselainten ohjelmallisen puolen kulmakivi, sen avulla saadaan luotua interaktiivisia elementtejä, kuten pelejä, pieniä ohjelmia sekä voidaan muokata sivun graafista ulkonäköä. JavaScript on vanhahko ohjelmointikieli ja sitä on pyritty korvaamaan HTML5-julkaisun myötä vuodesta 2014 lähtien. JavaScriptin parhaat puolet piilevät sen helppoudessa ja monipuolisuudessa, mutta samalla siitä on vuosien saatossa löydetty turvallisuusriskejä. Siinä missä PHP suoritetaan palvelimen päässä, suoritetaan JavaScript käyttäjän päässä, jolloin sitä hyväksikäyttämällä voidaan mahdollisesti saada käyttäjän tietoja kolmannelle osapuolelle. Tämän potentiaalisen turvallisuusriskin ja verkkosivun ylimääräisen kuorman vuoksi JavaScript on joissain selaimissa vakiona poistettu. Nimestään huolimatta JavaScriptillä ei ole oikeastaan mitään tekemistä Java-ohjelmointikielen kanssa. (How is JavaScript different from Java? [Viitattu 20.4.2019].)

4 SAUNAN KIUKAAN HALLINTA

4.1 Ohjelmallinen puoli

Suurin kysymys ohjelmallisen puolen toteutuksessa on, mitä kaikkea tarvitaan sen toiminnan varmistamiseen ja käyttömukavuuteen eli järjestelmän käyttöä tulee miettiä käyttäjän näkökulmasta. Kysymykseksi siis jääkin, millainen on käyttäjä?

Oletetaan että käyttäjä ei ole tekniikan alan asiantuntija, joten järjestelmän tulee olla yksinkertainen sekä graafisesti että mekaanisesti eli käyttäjälle jää mahdollisimman vähän arvailun varaan. Yksinkertaisin ratkaisu olisi kahden verkkosivun sivusto, jossa toisessa olisi iso ON-nappula ja toisessa iso OFF-nappula, sekä jonkinlainen visuaalinen indikaattori, josta voi nopeasti päätellä onko saunan kiuas päällä vai ei. ON-nappulaa painettaessa sivu käynnistäisi PHP:n avulla Python-skriptin joka antaisi Raspberry Pille komennon yhdistää nastat 13 (BCM 27) ja 14 (maa) kahdeksi tunniksi, jotka taas kontrolloisi relettä, joka on kiinni kiukaan ohjaimessa.

4.1.1 Verkkosivusto (HTML/CSS)

Verkkosivun rakentaminen alkaa elementtien listaamisesta, eli mitä kaikkia ominaisuuksia sivu tarvitsee. Työ aloitetaan yksinkertaisimmasta versiosta jossa on vain ON-nappula ja punainen laatikko. Tässä tapauksessa ei tarvitse vielä miettiä ohjelmallista puolta vaan pelkästään sitä miltä sivu näyttää käyttäjälle. Sivun kontrasti on hieman monimutkaisempi asia sillä ergonomian näkökulmasta sivun tulisi olla tummasävytteinen silmien säästämiseksi, mutta selkeyden vuoksi siinä tulisi silti olla jonkin verran kontrastia. Kontrastia ei kuitenkaan saisi hakea räikeillä väreillä eikä myöskään liian samanlaisilla väreillä. Väreistä puhuttaessa harmaa ja musta ovat neutraaleja ja sopivat käytännössä mihin tahansa väripalettiin joten niillä on hyvä lähteä rakentamaan sivuston väripalettia. (An Introduction to Color Theory for Web Designers 12.7.2012.)

Indikaattorin väri on punainen kiukaan ollessa pois päältä ja vihreä kiukaan ollessa päällä. Tässä on kuitenkin ongelma, sillä miehistä 8 % ja naisista 0,5 % on

punavihersoikeita, joten täytyy olla toinen indikaattori joka kertoo saunan tilan (Värisokeus ja poikkeava värinäkö 25.8.2018). Turvallisin ratkaisu tähän on vain kirjoittaa saunan tila verkkosivuun selkokielellä.



Kuva 9. Yksinkertainen versio verkkosivusta saunan ollessa pois päältä.



Kuva 10. Yksinkertainen versio verkkosivusta saunan ollessa päällä.

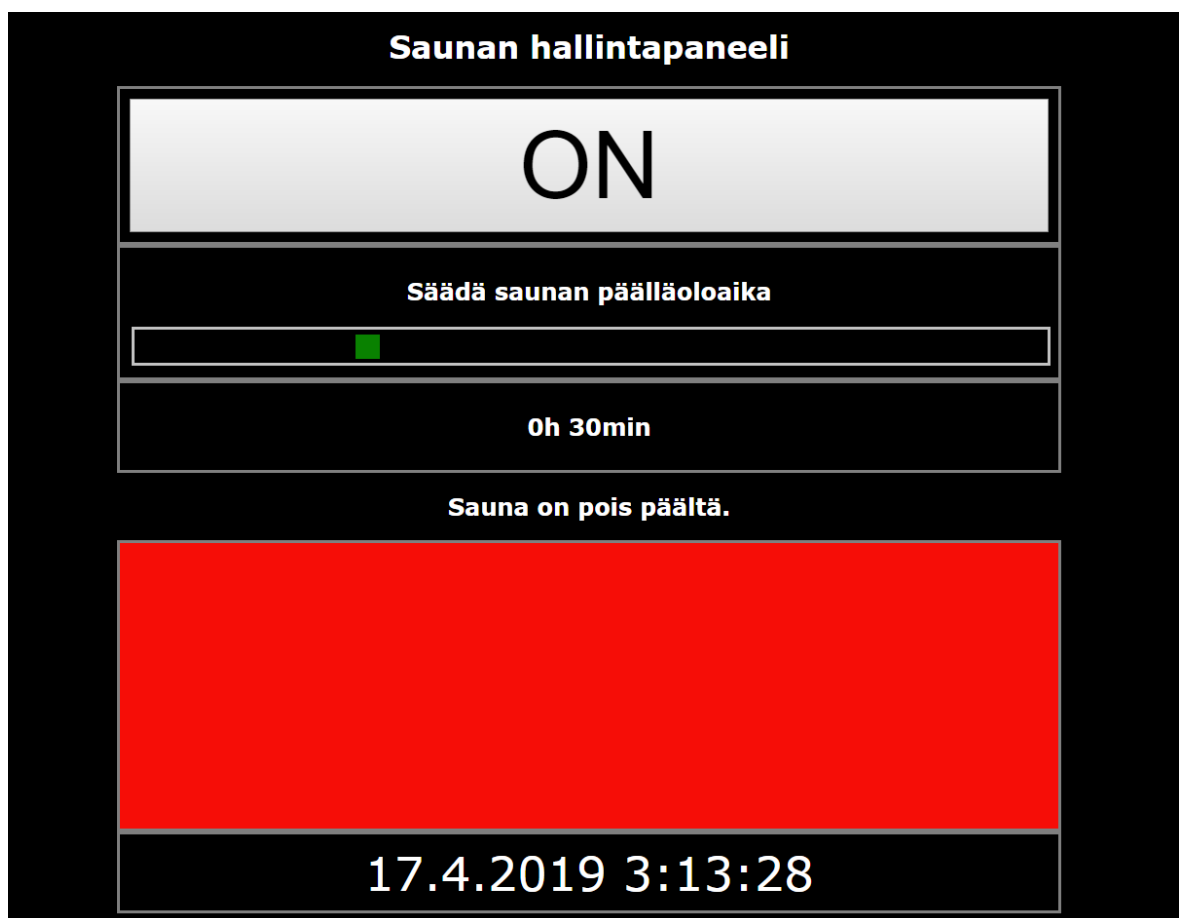
Kun yksinkertainen versio on valmiina, voidaan miettiä, mitä kaikkea muuta käyttäjä voi haluta nähdä ja tehdä:

- Säättää saunan päälläoloaika manuaalisesti
- Tietää nykyinen kellonaika ja päivämäärä
- Nähdä kauanko sauna on vielä päällä.

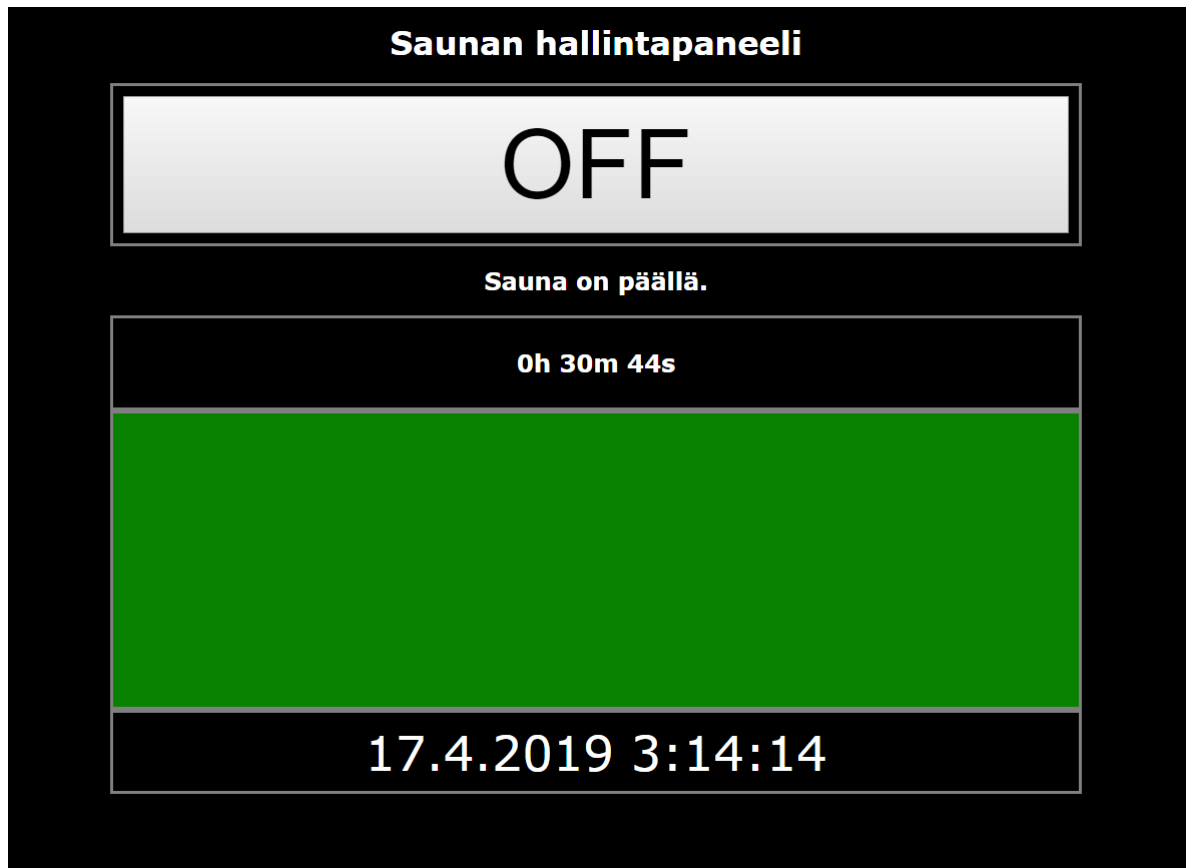
Manuaaliseen säätöön on olemassa aika monta eri ratkaisua, kuten ajan syöttö käsin kirjoittamalla, ajan valinta valintavalikosta tai vaikkapa ajan säätäminen linkeistä. Tässä tapauksessa käytetään slider-elementtiä eli liukuvalitsinta. Liukuvalitsin on hyvä, koska se on yksinkertainen ja sopii lineaariseen verkkosivusuunnitteluun hyvin. Huonona puolena liukuvalitsimessa on sen numeraalisen arvon indikaattorin puuttuminen, eli käyttäjälle täytyy tehdä erikseen

jokin elementti, joka näyttää liukuvalitsimen arvon. Tämä ei ole onneksi kovinkaan monimutkainen ohjelmoida.

Nykyisen kellonajan ja päivämäärän lisääminen verkkosivulle ei ole välttämättömyys, mutta tässä tapauksessa on ihan tarpeellista, tietää mitä kello on ja mikä päivä on. Päivämäärän sijoittelu sivulle on myös hyvin tärkeää, sillä se on tässä tapauksessa vain lisäelementti eikä pääelementti. Näin ollen kello sijoitetaan sivun alareunaan, missä se ei vie huomiota pois pääelementeistä. Saunan päälläoloaika on kuitenkin relevantti tieto, joten sen sijoittaminen keskemmas sivua on suositeltavaa.



Kuva 11. Lopullinen verkkosivu saunan ollessa pois päältä.



Kuva 12. Lopullinen verkkosivu saunan ollessa päällä.

Täytyy myös miettiä sivuston joustavuutta, entä jos käyttäjä haluaakin käyttää sivustoa esimerkiksi puhelimella tai tabletilla? Tähän joustavaan verkkosivusuunnitteluun on olemassa valmiita viitekehysjä, kuten Bootstrap tai Pure.css, joita hyödyntämällä pystyy rakentamaan responsiivisen verkkosivuston nopeasti. Tässä tapauksessa ei kuitenkaan ole varsinaisesti tarvetta käyttää mitään erillistä viitekehystä, sillä sivusto on hyvin pieni ja vain yhteen tarkoitukseen. Responsiivisuus on tässä sivustossa ratkaistu käyttämällä suhteellisia arvoja kiinteiden arvojen asemesta (liite 1).

Lisäksi turvallisuuden ja selkeyden vuoksi verkkosivujen osoitteita ei näytetä osoitepalkissa ollenkaan käyttämällä index.html-pääsivussa iframe-elementtiä, jonka sisällä kaikki sivut näytetään. Iframella pystytään myös auttamaan skaalauksessa muille laitteille.

4.1.2 Verkkosivuston ohjelmallinen puoli (JavaScript)

JavaScript suoritetaan käyttäjän selaimessa, joten sitä pitäisi pyrkiä käyttämään mahdollisimman vähän, ettei päätelaitteen suorituskyky jää pullonkaulaksi. Yksinkertaiset koodinpätkät pystyy suorittamaan melkein millä tahansa laitteella, joka on tehty vuoden 2010 jälkeen, mutta raskaamman koodin toimimiseen voi tarvita jo tehokkaan pöytätietokoneen. Näistä syistä JavaScriptin määrä tässä sivustossa on suhteellisen pieni.

Ensimmäinen funktio on `getClock()`, jonka tehtävä on hakea käyttäjän sijainnin aikavyöhykkeen mukainen aika ja päivämäärä, sekä näyttää se käyttäjälle. JavaScriptissä on useita valmiiksi määritettyjä sisäisiä funktiota, ja `Date()` on yksi niistä. Yksinkertaisuudessaan koodinpätkä `new Date d = Date();` tallentaa senhetkisen päivämäärän ja kellonajan muuttujaan, jonka jälkeen siitä pystytään erikseen ottamaan ulos päivämäärän ja kellonajan eri ominaisuuksia (tunnit, viikonpäivä, vuosi jne.) valmiilla metodeilla. Ainoa esteettinen muutos, mikä pitää manuaalisesti tehdä on formatoida kellonaika oikein (13:07 vs. 13:7). Tämä tapahtuu tarkastamalla onko muuttujan arvo alle 9 ja lisäämällä nollan muuttujan eteen. `getClock()`-funktion kutsumisen lisäksi täytyy myös HTML-puolella lisätä `setInterval(getClock, 1000)`-funktio joka automaattisesti suorittaa `getClock()`-funktion tuhannen millisekunnin eli yhden sekunnin välein. Funktio `getClock()` sijoittaa arvonsa `clockContainer`-elementin sisälle.

Seuraava funktio, `setTimeBox()`, toimii yhteysväylänä liukuvalitsimen ja aikaindikaattorin välillä saunan ollessa pois päältä. Liukuvalitsin saa arvoja väliltä 0 ja 7201, eli aika sekunteina kauanko sauna on päällä. `Date()`-funktioita voi käyttää myös syöttämällä sinne aikaa millisekunteina, jolloin se laskee millisekuntien perusteella kokonaisajan. Eli liukuvalitsimen arvo kerrotaan tuhannella ja syötetään `Date()`-funktioon jonka jälkeen lasketaan tunnit ja minuutit, korjataan formatointi oikeaan muotoon ja syötetään arvot `timeOn`-elementtiin.

Viimeinen funktio on kaikista raskain ja monimutkaisin. Funktio `timeOnCountDown()` laskee aikaa alaspäin saunan ollessa päällä ja aloittaa laskunsa edellisen funktion arvon mukaan. Tässä tuleekin ensimmäinen ongelma, sillä funktiot toimivat eri sivuilla eli ne eivät pysty lukemaan toistensa arvoja suoraan. Onneksi

JavaScriptissä on olemassa localStorage ja sessionStorage, joista ensimmäinen tallentaa tiedon selaimeen, kunnes se tieto poistetaan ja jälkimmäinen tallentaa tiedon selaimeen, kunnes selain tai välilehti suljetaan. Kyseessä ei kuitenkaan ole varsinaisesti eväste, sillä evästeitä käytetään yleensä palvelimen päässä ja localStorageea sekä sessionStorageea käytetään käyttäjän päässä. Tästä huolimatta selaimeen tallennettavat tiedot ovat EU:n evästelain alla (Cookies, 10.12.2018).

Tässä tapauksessa käytetään localStorageea, jolloin ikkunan sulkeminen ei poista tietoa. Funktio hakee localStorageesta nimen timeLeft kohdalla olevan arvon, jonka se sen jälkeen lisää senhetkiseen aikaan, ja tallentaa muuttujaan countdownDate. Tämän jälkeen alkaa silmukka, jota toistetaan kerran sekunnissa. Silmukan sisällä lasketaan erotus countdownDateen ja senhetkisen ajan välillä, jolloin saadaan muuttuja distance eli etäisyys. Tästä muuttujasta sitten puretaan tunnit, minuutit ja sekunnit laskemalla jakojäännös jokaisen aikamuuttujan kohdalla ja jakamalla jakojäännös ajan määrällä. Esimerkiksi yksi minuutti on 60 000 millisekuntia, tunnissa on 3 600 000 millisekuntia joten jakojäännökseksi jää 60 000 millisekuntia joka jaetaan millisekuntien määrällä minuutissa. Näiden aikamuuttujien selvittyä ne syötetään elementtiin timeLeftBox. Jos distance-muuttujan arvo on alle yksi, funktion lopussa oleva ehto käy toteen. Ehtolauseen sisältö lopettaa funktion päivittämisen automaattisesti, tyhjentää localStorageen ja uudelleenohjaa selaimen saunaTurnOff.php-sivulle, josta se uudelleenohjataan takaisin etusivulle. (Liite 2)

4.1.3 Palvelimen ohjelmallinen puoli (PHP ja Python)

Palvelimen ohjelmallisesta puolesta vastaavat PHP-sivut ja Python-skripti toimivat toistensa kanssa käytännössä täysin erillään muusta järjestelmästä. PHP-sivut eivät periaatteessa tee muuta kuin käskyttävät palvelinta käynnistämään Python-skriptin ja uudelleenohjaavat selaimen eteenpäin on- tai off-sivulle. Python puolestaan tekee suuremman työn tässä järjestelmässä, se ottaa PHP:n antamat käskyt vastaan ja hallitsee ohjausnastoja. Tässä projektissa on kaksi PHP-sivua ja yksi Python-skripti.

Ensimmäinen PHP-sivu on saunan käynnistävä sivu, joka vastaanottaa POST-metodilla lähetetyn datan. POST ja GET ovat siis HTTP-metodeja joilla voidaan hakea tai lähettää tietoa sivulta palvelimelle. Tässä tapauksessa saunaOff.html-sivulla on form-elementti, jonka sisällä on liukuvalitsin timeOnRange. Liukuvalitsimen arvo lähetetään POST-metodilla saunaTurnOn.php-sivulle, jossa se sijoitetaan muuttujaan \$time. Tämän jälkeen luodaan muuttuja \$command, johon kirjoitetaan palvelimen, tässä tapauksessa Raspberry Pin suorittama komento. Komento käskyy laitetta suorittamaan setOn.py-skriptin ja antaa siihen argumentiksi ajan \$time sekä sijoittaa skriptin tulostaman datan /dev/null-tyhjiöön. Tämän /dev/null-tyhjiön idea on se, että kaikki data mitä sinne tallennetaan, poistetaan automaattisesti. Tässä tapauksessa ei tulostetta tarvita joten, sitä on turha säilyttää.

Toinen PHP-sivu on saunan päälläolon keskeyttävä saunaTurnOff.php-sivu. Python-skripti tarkistaa, onko olemassa tiedosto nimeltä stop-script. Jos sen niminen tiedosto on olemassa annetussa osoitteessa, silmukka katkeaa ja sauna sammuu. Tämä tiedosto luodaan saunaTurnOff.php-sivun komennossa touch /home/pi/python/stop-script samaan kansioon Python-skriptin kanssa. Komento touch luo tyhjän tiedoston annettuun sijaintiin, mikä on vanha komento Linuxin alkuajoilta, mutta se on yksinkertainen ja toimintavarma komento.

Python-skripti on viimeinen välikäsi ennen relettä ja itse saunaa joten sen toimintavarmuus on erityisen kriittinen asia. Python on kielenä onneksi aika yksinkertainen ja selkeälukuinen, vaikka hieman outo rivinpäätteen merkinnän puutteen vuoksi. Raspberry Pin kanssa Python on vielä turvallisin vaihtoehto

RPi.GPIO-kirjaston vuoksi, jolla pystytään ohjaamaan laitteen ohjausnastoja. Skriptiin lisätään neljä kirjastoa, jotka ovat:

RPi.GPIO	-	GPIO-ohjausnastojen käsittely
time	-	Ajanhallinta, tauottaminen
argparse	-	Argumenttien lisääminen skriptiin
os	-	Laitteen käyttöjärjestelmän käsittely

Alkuun täytyy muistaa määrittää, käytetäänkö Broadcomin (Raspberry Pin prosessorin valmistajan) vai Raspberry Pi Foundationin määriytyksiä nastoille. Valinnalla ei ole käytännön kannalta mitään väliä, se vain numeroi nastat eri tavalla. Tässä tapauksessa käytetään Raspberry Pin määriytyksiä ja valitaan relettä ohjaavaksi nastaksi nastan numero 13 sekä tappokytkintä varten nasta numero 15. Seuraavaksi luodaan parseri, jolla tuodaan argumentit koodiin sisälle ulkopuolelta. Argumentit lisätään järjestyksessä, tyypitetään ja nimetään. Tämän jälkeen argumentteja voidaan käyttää muuttujina koodissa.

Edellä mainittu stop-script-tiedosto sulkee Python-skriptin, joten ennen kuin voidaan aloittaa silmukan, käyttö täytyy se poistaa. Tähän tarvitaan os-kirjastoa, eli käyttöjärjestelmää hallitsevaa kirjastoa, jolla pystytään tarkistamaan, onko tiedosto olemassa, ja poistaa tiedosto, jos se on olemassa. Itse silmukka suoritetaan niin monta kertaa, kuin argumentissa timeOn on määritetty. Aina silmukan alussa tarkistetaan, mikä on killSwitch-muuttujan arvo eli onko tappokytkin pohjassa vai ei. Silmukassa on kolme eri vaihtoehtoa:

- Ensimmäisen kerran kun silmukka suoritetaan kytketään rele päälle, tulostetaan "OK" ja laitetaan ohjelma sekunniksi tauolle. Ehtoina tämän vaihtoehdon toteutumiselle on, että tappokytkin ei ole pohjassa, silmukan kierroslaskuri on 0 ja tiedostoa stop-script ei ole olemassa.
- Jos tappokytkin on pohjassa tai stop-script on olemassa, kytketään rele pois päältä, poistetaan stop-script-tiedosto, tulostetaan "TERMINATED BY KILLSWITCH" ja suoritetaan gpio.cleanup(), jolla tyhjennetään GPIO-nastojen määriytykset. Tämän jälkeen silmukka katkaistaan.

- Muissa tapauksissa tulostetaan "ON" ja prosenttimäärä, kauanko ajasta sauna on ollut päällä sekä tauotetaan ohjelma.

Silmukan suorituksen jälkeen ohjelma vielä tyhjentää GPIO-nastojen määitykset etteivät ne jää järjestelmään muita mahdollisia ohjelmia häiriten. Tämän jälkeen tulostetaan "Done." ja ohjelma on suoritettu loppuun. (Liite 3)

4.1.4 Muut ohjelmalliset asiat

Raspberry Pihin pitää asentaa järjestelmän lisäksi Apache eli HTTP-palvelinohjelmisto ja PHP. Tämä tapahtuu yksinkertaisesti terminaalissa komennolla `sudo apt-get install apache2 php`, ja konfigurointia ei tarvitse tehdä. Jos Raspberry Pi liitetään langattomasti, täytyy se erikseen konfiguroida `wpa_supplicant.conf`-tiedostossa, joka sijaitsee `/etc/wpa-suppllicant`-kansiossa. Tiedostoon lisätään tiedot seuraavanlaisesti:

```
network={  
  
    ssid="verkon nimi"  
  
    psk="verkon salasana"  
  
}
```

Tämän jälkeen Raspberry Pi yrittää aina yhdistää tähän verkkoon automaattisesti käynnistyksen yhteydessä. Yhdistettävä johdon kanssa tätä ei tarvitse tehdä, vaan se tapahtuu automaattisesti. Verkkoasetuksia voi halutessaan säätää vielä niin että Raspberry Pi saa staattisen (kiinteän) IP-osoitteen lähiverkossa muokkaamalla `/etc/network/interfaces`-tiedostoon DHCP:n kautta tulevan IP-osoitteen kiinteäksi seuraavalla tavalla:


```
allow-hotplug wlan0
```

```
iface wlan0 inet static
```

```
address 192.168.1.111 //osoite jonka haluat palvelimelle antaa
```

```
netmask 255.255.255.0 //verkkomaski
```

```
gateway 192.168.1.1 //reitittimen osoite
```

```
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Tämän jälkeen verkkosivustoon pääsee käsiksi selaimella osoitteessa 192.168.1.111. Se ei muutu ellei tiedostoa muuteta.

Raspberry Pin puolella joudutaan tekemään vielä muutoksia käyttöoikeuksiin. Apache-ohjelmistolla on oma käyttäjä, www-data, Raspbian-käyttöjärjestelmässä. Tällä käyttäjällä ei ole vakiona oikeuksia käsitellä käyttöjärjestelmän tiedostoja. Tämän takia joudutaan luomaan ensiksi ryhmä groupadd verkko -komennolla, ja lisätään www-data ryhmään verkko

```
usermod -a -G verkko www-data
```

Tämän jälkeen annetaan ryhmälle oikeudet käyttää sekä /var/www-kansiota (verkkosivujen sijainti) että /home/pi/python-kansiota (setOn.py- ja stop-script-tiedostojen sijainti). Oikeuksien antaminen tapahtuu chmod-komennolla

```
chmod -R g+rx /var/www
```

Tämä komento rakentuu siis neljästä argumentista: itse komento chmod, -R eli rekursiivisuus, g+rx eli ryhmälle (g) annetaan oikeudet lue (r) ja suorita (x) sekä kansion sijainti. Tämän jälkeen oikeuksia ei tarvitse muuttaa enempää. Tämä on turvallisuustekijöistä yksi suurimpia. Jos www-datalle annettaisiin kaikki oikeudet tehdä mitä tahansa, se voisi kirjoittaa tiedostoa uusiksi ja tehdä pahaa jälkeä koko järjestelmälle, pahimpana pelkona ollen saunan vaurioituminen.

4.2 Asentaminen

Tässä luvussa käsitellään itse laitteen fyysinen asennus, johdotukset ja varotoimenpiteet. Onneksi järjestelmän johdotus on suhteellisen yksinkertainen ja se voidaan sijoittaa kosteiden tilojen ulkopuolelle.

4.2.1 Raspberry Pi ja XCSOURCE TE213 -relemoduuli

Relemoduulissa on kolme lähtöä; NC, NO ja COM. Näistä NC eli Normally Closed tarkoittaa normaalitilassa suljettua piiriä, NO eli Normally Open tarkoittaa normaalisti avointa piiriä ja COM-lähtöön tulee hallittavan piirin toinen johto. Tässä tapauksessa hallitaan saunan kiukaan ajastinta tai pikemminkin ohitetaan ajastin ajamalla sitä digitaalisesti mekaanisen kellon asemesta. Kuitenkin, saunan kiuas on normaalitilassa pois päältä eli piiri on avoin eli johdotukset ajastimesta tulevat relemoduulin COM-lähtöön ja NC-lähtöön. Sulakkeesta ei tarvitse huolehtia, se on kiukaan tehonsäädössä, ajastin ei vaikuta kuin virransyötön keston.

Python-koodissa määritettiin relettä ohjaavaksi nastaksi nasta numero 13, joten se kytketään relemoduulin IN-tuloon. XCSOURCE TE213 -moduuli toimii viiden voltin jännitteellä, joten Raspberry Pi:n nasta numero 2 kytketään moduulin VCC-tuloon, ja maa-nasta numero 6 kytketään moduulin GND-tuloon. Raspberry Pihin tarvitaan vielä viiden voltin tulovirta micro-USB-johdolla, sekä USB-WiFi-adapteri, jolla se voi yhdistää langattomaan verkkoon.

Raspberry Pi2 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power	⬛	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	⬢	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	⬢	Ground	06
07	GPIO04 (GPIO_GCLK)	⬢	(TXD0) GPIO14	08
09	Ground	⬛	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	⬢	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	⬢	Ground	14
15	GPIO22 (GPIO_GEN3)	⬢	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	⬛	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	⬢	Ground	20
21	GPIO09 (SPI_MISO)	⬢	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	⬢	(SPI_CE0_N) GPIO08	24
25	Ground	⬛	(SPI_CE1_N) GPIO07	26
<hr/>				
27	ID_SD (I ² C ID EEPROM)	⬢	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	⬢	Ground	30
31	GPIO06	⬢	GPIO12	32
33	GPIO13	⬢	Ground	34
35	GPIO19	⬢	GPIO16	36
37	GPIO26	⬢	GPIO20	38
39	Ground	⬛	GPIO21	40

Early Models

Late Models

Rev. 1
26/01/2014

<http://www.element14.com>

Kuva 13. Raspberry Pin nastojen tiedot. (Sparkfun 2019.)

4.2.2 Kotelointi ja nappulat

Kotelointi on väliaikaisesti toteutettu vain teipatulla pahvilaatikolla, mutta asentamishetkellä siihen rakennetaan jostain kosteudenkestävästä materiaalista kotelo. Kotelossa on reikä, josta näkee Raspberry Pi:n virta- ja kirjoitusvalon, eli tavallaan visuaalista virheenkorjausta. Raspberry Pin kotelo on 3D-tulostimella läpikuultavasta PLA-muovista tulostettu yksinkertainen kotelo, josta pystyy näkemään läpi Raspberry Pin led-indikaattorit. Koko järjestelmän kotelon keskellä on tappokytin johon ei ole kytketty valoa ainakaan tässä versiossa.

5 LOPPUYHTEENVETO, POHDINTAA JA JATKOKEHITYSIDEOITA

Tässä osassa käsitellään tekijän kokemuksia, projektin etenemistä ja projektin mahdollista jatkoa. Opinnäytetyön tekemisen aikana ehti tapahtumaan aika paljon sitä viivytäviä ja haittaavia asioita, mutta myös syntyi monta kehitysideaa.

5.1 Vastoinkäymisiä

Toukokuun alussa vuonna 2018 SQB Centerissä sattui tulipalo, joka sai alkunsa saunasta. Tämä luonnollisesti häiritsti opinnäytetyön edistymistä ja tekeminen viivästy entisestään, kun ei ollut mahdollista päästä asentamaan pitkään aikaan. Tulipalon seurauksena päätettiin, ettei ole turvallista tehdä saunaan omia virityksiä ennen kuin saadaan vakuutusasiat selville.

Ongelmat vain jatkuivat, yritys ei saanut täyttä korvausta vakuutuksesta ja joutui yrityssaneeraukseen. Samoihin aikoihin tämän opinnäytetyön yhteyshenkilö lopetti työnsä yrityksessä ja tämä opinnäytetyö jäi jälleen vaiheeseen.

5.2 Kehitysideoita

Tulipalo toi kuitenkin paljon turvallisuuskysymyksiä ja ideoita sekä kysymyksen ”Miten tämän olisi voinut estää?”. Lämpötila-anturin ja häikäanturin asentaminen järjestelmään on täysin mahdollista, ja on helppoa lisätä jonkinlainen hälytysjärjestelmä sekä lämpötilantarkkailujärjestelmä. Muutenkin antureiden lisääminen ja datan kerääminen voisi edesauttaa saunatilojen ominaisuuksien tulkintaa.

Toinen asia, jota pohdittiin jo projektin alussa, oli varausjärjestelmän lisääminen saunanhallintaan, mutta se ei tässä skaalassa ole välttämättä tarpeellinen. Naisten puolen sauna ei ole läheskään niin usein käytössä kuin miesten sauna (miesten sauna on koko päivän päällä), mutta kenties jonain päivänä tätä voitaisiin skaalata eteenpäin. Tietojen syöttäminen kalenterista tietokantaan ei ole kovin

monimutkainen prosessi eikä tietokannan lukeminen PHP:n avulla ole myöskään mahdollisuus.

Kotelointi on yksi kysymys, nyt kun on mahdollista 3D-tulostaa esimerkiksi Seinäjoen kaupunginkirjaston Apila-verstaalla halvalla. Kotelosta voi tehdä melkein vesitiiviin ja näin ollen sijoittaa muihinkin tiloihin. Koteloon voisi lisätä myös manuaalisen käytön: painike, jota painamalla sauna on automaattisesti kaksi tuntia päällä, tai painike, jota painamalla sauna on puoli tuntia pidempään päällä. Mahdollisuudet ovat pitkälti rajattomat.

5.3 Loppuyhteenveto

Alkuun täytyy mainita opinnäytetyön olleen haastava, jos joskus kymmenen vuotta sitten olisi joku sanonut, että saunatiloja ohjataan internetin yli, olisi moni nauranut, mutta nyt se on arkipäivää. Erona vain tässä opinnäytetyössä oli se, että tämä toimii vanhoillakin kiukailla. Myönnetään, kaikkea mitä haluttiin saada aikaan ei saatu aikaan, mutta toimiva järjestelmä on aina toimiva järjestelmä.

Minimalistinen web-suunnittelu on hieman katoamassa oleva asia erilaisten viitekehysten kehittyessä ja verkkosivukoneiden yleistyessä, mutta tätä tehdessä sitä sai opetella aivan kiitettävästi. Funktionaalisen verkkosivun rakentaminen on helppoa kunnes se ei ole, se tätä tehdessä opittiin. Responsiivinen versio syntyi puhtaasti vahingossa kun yritettiin keskittää elementtejä sivulle.

Vaikka tätä opinnäytetyötä ei saatu koskaan oikeasti implementoitua niin sen tekeminen ei silti jää harmittamaan. Tässä työssä opittiin niin paljon eri ohjelmointikielien välisistä vuorovaikutuksista. Tärkeimpänä opittiin kuitenkin fakta, että kaikkea voi sattua kesken projektin, jopa tulipalo.

LÄHTEET

- An Introduction to Color Theory for Web Designers. 12.7.2012. [www-sivu]. Envato Pty Ltd. [Viitattu 2.5.2019]. Saatavilla: <https://webdesign.tutsplus.com/articles/an-introduction-to-color-theory-for-web-designers--webdesign-1437>
- Cookies. 10.12.2018. [www-sivu]. European Commission. [Viitattu 20.4.2019]. Saatavilla: http://ec.europa.eu/ipg/basics/legal/cookies/index_en.htm
- Evans. 2011. The Internet of Things. [www-dokumentti]. Cisco Systems, Inc. [Viitattu 20.4.2019]. Saatavilla: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
- FDA confirms that St. Jude's cardiac devices can be hacked. 9.1.2017. [www-sivu]. Cable News Network. [Viitattu 2.5.2019]. Saatavilla: <https://money.cnn.com/2017/01/09/technology/fda-st-jude-cardiac-hack/>
- Gilchrist, A. 2017. IoT Security Issues. 1. painos. Boston: De|G Press.
- Griffin Technology Unveils Griffin Home. 4.1.2017. [www-sivu]. Griffin Technology. [Viitattu 20.4.2019]. Saatavilla: <https://press.griffintech.com/release/griffin-technology-unveils-griffin-home-a-collection-of-smart-apppowered-appliances-that-simplify-and-enhance-everyday-routines-at-ces-2017/>
- How is JavaScript different from Java? Ei päiväystä. [www-sivu]. Oracle Corporation. [Viitattu 20.4.2019]. Saatavilla: https://www.java.com/en/download/faq/java_javascript.xml
- Most Popular Development Environments. 2019. [www-sivu]. Stack Exchange Inc. [Viitattu 20.4.2019]. Saatavilla: <https://insights.stackoverflow.com/survey/2019#technology--most-popular-development-environments>
- No need for PCs with intelligent fridges. 13.12.2000. [Verkojulkaisu]. ITWeb South Africa. [Viitattu 20.4.2019]. Saatavilla: <https://www.itweb.co.za/content/KA3WwqdlozkqrydZ>
- PHP. 4.4.2019 [www-sivu]. The PHP Group. [Viitattu 23.4.2019]. Saatavilla: <https://www.php.net/>
- Squash & Bowling Center. Ei päiväystä. Sulkapallo/Pickleball. [www-sivu]. Seinäjoen Palloiluhallit Oy. [Viitattu 2.5.2019]. Saatavilla: <https://www.sqb.fi/sulkapallo>

- Sparkfun. Ei päiväystä. [www-lähde]. SparkFun Electronics. [Viitattu 19.4.2019]. Saatavilla: <https://learn.sparkfun.com/tutorials/raspberry-gpio/gpio-pinout>
- The Internet Toaster. 7.1.2000. [verkkokirja]. Living Internet. [Viitattu 20.4.2019]. Saatavilla: https://www.livinginternet.com/i/ia_myths_toast.htm
- The Trojan Room Coffee Pot. 5.1995. [www-sivu]. Quentin Stafford-Fraser. [Viitattu 20.4.2019]. Saatavilla: <https://www.cl.cam.ac.uk/coffee/qsf/coffee.html>
- That 'Internet of Things' Thing. 22.6.2009. [www-sivu]. Emerald Expositions, LLC. [Viitattu 20.4.2019]. Saatavilla: <https://www.rfidjournal.com/articles/view?4986>
- Tokyo Toilets deliver data dump by very personal RSS. 2009. [www-sivu]. Digital World Tokyo. [Viitattu 23.4.2019]. Saatavilla: http://www.digitalworldtokyo.com/index.php/digital_tokyo/articles/tokyo_toilets_deliver_data_dump_by_very_personal_rss/
- Usein kysyttyä EU:n tietosuoja-asetuksesta. Ei päiväystä. [www-sivu]. Tietosuojavaltuutetun toimisto. [Viitattu 20.4.2019]. Saatavilla: <https://tietosuoja.fi/gdpr>
- Visual Studio Code. 4.4.2019. [www-sivu]. Microsoft Corporation. [Viitattu 23.4.2019]. Saatavilla: <https://code.visualstudio.com/>
- Värisokeus ja poikkeava värinäkö. 25.8.2018. [www-sivu]. Kustannus Oy Duodecim. [Viitattu 20.14.2019]. Saatavilla: https://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_artikkeli=dlk00347
- What is Industrial Automation? 15.10.2013. [www-sivu]. Sure Controls. [Viitattu 24.4.2019]. Saatavilla: <https://www.surecontrols.com/what-is-industrial-automation/>
- What Is IoT? – A Simple Explanation of the Internet of Things. 6.1.2019. [www-sivu]. IoT For All. [Viitattu 24.4.2019]. Saatavilla: <https://www.iotforall.com/what-is-iot-simple-explanation/>
- Why Python is Popular Despite Being (Super) Slow. [www-sivu]. Medium Programming. [Viitattu 23.4.2019]. Saatavilla: <https://medium.com/@trungluongquang/why-python-is-popular-despite-being-super-slow-83a8320412a9>

LIITTEET

Liite 1. CSS-tyylimääritykset

Liite 2. JavaScript

Liite 3. Python

LIITE 1 CSS-tyylimääritykset

```
h1,h2 { /*Tekstien muotoilut*/
  text-align: center;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  color: white;
}
body { /*Koko sivuston taustaväri*/
  background-color: black;
}
iframe{ /*iframe-elementin määritykset, ei reunoja ja täyttää koko ruudun*/
  width: 100%;
  height: 98%;
  border: none;
}
.buttons { /*Nappuloiden containerin määritykset*/
  margin: auto; /*marginin ollessa auto container on keskitettynä
ruutuun*/
  width: 50%;
  border: 3px solid gray;
  padding: 10px;
}
.button { /*Nappuloiden määritykset*/
  width: 100%; /*Täyttää koko containerin*/
  height: 15%;/*Täyttää 15% ruudusta*/
  font-size: 6em; /*Nappulan ON/OFF -tekstin koko*/
}
/*statusBoxOn ja statusBoxOff on indikaattorin määrityksiä, eivät eroa
toisistaan kuin värillä*/
.statusBoxOn {
  margin: auto;
  width: 50%;
  height: 30%;
  border: 3px solid gray;
  padding: 10px;
  background-color: green;
}
.statusBoxOff {
  margin: auto;
  width: 50%;
  height: 30%;
  border: 3px solid gray;
  padding: 10px;
  background-color: red;
}
```

```
.slidecontainer { /*Slider-elementin containerin määrittelyt*/
  width: 50%;
  margin: auto;
  border: 3px solid gray;
  padding: 10px;
}
.slider {
  -webkit-appearance: none; /*Ohittaa oletustyyli-määrittelyt*/
  appearance: none; /*Poistaa oletus slider-elementin näkyvyyden*/
  width: 100%; /* Täyttää koko containerin */
  height: 3em; /* Määrätty korkeus */
  background: black; /* Taustan väri */
  opacity: 1; /* Läpinäkyvyyden määrittely osoittimen ollessa päällä */
  transition: opacity .2s; /*Aika, joka kuluu läpinäkyvyyden muutokseen*/
  border-color: gray; /*Reunan väri*/
  border: solid; /*Reunan muotoilu*/
}
.slider::-webkit-slider-thumb { /*Slider-elementin osoittimen muotoilu
webkit-pohjaisissa selaimissa kuten Googlen Chrome*/
  -webkit-appearance: none;
  appearance: none;
  width: 25px;
  height: 25px;
  background: green;
  cursor: pointer;
}
.slider::-moz-range-thumb { /*Sama kuin yllä, mutta Mozillan Firefox-
selaimelle*/
  width: 25px;
  height: 25px;
  background: green;
  cursor: pointer;
}
.timeOn{ /*Kiukaan päälläoloajan containerin määrittelyt*/
  margin: auto;
  width: 50%;
  border: 3px solid gray;
  padding: 10px;
}
#clockContainer { /*Päivämäärän/kellonajan ulkoasun määrittelyt*/
  margin: auto;
  width: 50%;
  border: 3px solid gray;
  padding: 10px;
  text-align: center;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  font-size: 3em;
  color: white;
}
```

LIITE 2 JavaScript

```

function getClock(){
    //Date() on tämänhetkinen päivämäärä/aika
    var d=new Date();
    //Päivämäärän muuttujat
    var nmonth=d.getMonth(),ndate=d.getDate(),nyear=d.getFullYear();
    //Kellonajan muuttujat
    var nhours=d.getHours(),nmins=d.getMinutes(),nsecs=d.getSeconds();
    //Formatoinnin korjaaminen, hh:mm:ss
    if(nmins<=9) nmins="0"+nmins;
    if(nsecs<=9) nsecs="0"+nsecs;
    //Rakennetaan muuttuja syötettäväksi elementtiin
    var clockTime=""+ndate+"."+(nmonth+1)+"."+nyear+"
"+nhours+":"+nmins+":"+nsecs+"";
    //Syötetään muuttuja elementtiin jonka id on clockContainer
    document.getElementById('clockContainer').innerHTML=clockTime;
}
function setTimeBox(){
    //Sliderin arvon määrittäminen
    var time = document.getElementById("timeOnRange").value;
    //Tallennetaan localStorageiin arvo timeOnCountDownia varten
    window.localStorage.setItem("timeLeft", time);
    //Date(millis) ottaa ajan millisekunteina ja laskee siitä
päivämäärän/ajan
    var outPutTime = new Date(time*1000);
    //Tuntien määrittäminen
    var hours = outPutTime.getHours();
    //Minuuttien määrittäminen
    var minutes = outPutTime.getMinutes();
    //Formatoinnin korjaaminen
    if(minutes<=9) minutes="0"+minutes;
    //Näyttää sliderin arvon tunteina/minuutteina timeOn elementissä
    document.getElementById("timeOn").innerHTML = hours - 2 +"h "+minutes +
"min ";
}
function timeOnCountDown(){
    var d = new Date();
    //Määritetään päättöaika
    var countDownDate=d.getTime()+window.localStorage.getItem("time-
left")*1000;

    //Päivitetään koodia kerran tuhannessa millisekunnissa eli sekunnissa
    var x = setInterval(function() {

```

```
//Haetaan tämänhetkinen päivämäärä
var now = new Date().getTime();
//Lasketaan etäisyys tämänhetkisen päivämäärän ja päättöajan välillä
var distance = countdownDate - now;

//Lasketaan tunnit, minuutit ja sekunnit etäisyydestä
var hours = Math.floor((distance % (1000*60*60* 4)) / (1000*60*60));
var minutes = Math.floor((distance % (1000*60*60)) / (1000*60));
var seconds = Math.floor((distance % (1000*60)) / 1000);

//Syötetään jäljellä oleva aika timeLeftBox-elementtiin
document.getElementById("timeLeftBox").innerHTML = hours + "h "
+ minutes + "m " + seconds + "s ";

//Kun aika menee alle yhden sekunnin niin suoritetaan alla oleva
if (distance < 1) {
    //Lakataan päivittämästä funktiota
    clearInterval(x);
    //Poistetaan aikamääre localStorageesta ettei tule ongelmia
    window.localStorage.removeItem("timeLeft");
    //Uudelleenohjataan saunaTurnOff.php-sivulle
    window.location.href = 'saunaTurnOff.php';
}
}, 1000);
}
```

LIITE 2 Python

```
#!/usr/bin/env python
import RPi.GPIO as gpio
import time
import argparse
import os
#Pinnimaaritykset releelle

gpio.setmode(gpio.BOARD)
gpio.setup(13, gpio.OUT)
gpio.setup(15, gpio.IN) #killSwitchin pin
#Argumenttien parsiminen
parser = argparse.ArgumentParser(description='Kiukaan hallintaohjelma')
parser.add_argument('timeOn', type=float, help='Aika sekunteina kauanko
kiuas kay')
args = parser.parse_args()
if os.path.isfile('/home/pi/python/stop-script'):
    os.remove('/home/pi/python/stop-script')
ok = 0

for x in range(int(args.timeOn)):
    killSwitch = gpio.input(13)
    if killSwitch == 0 and x == 0 and not os.path.isfile('/home/pi/py-
thon/stop-script'): #Korjaa PATH ku siirrat naa raspiin
        ok = 1
        print ("OK " + str ((x+1)*100 / args.timeOn) + "%")
        gpio.output(13, gpio.HIGH)
        time.sleep(1)
    elif killSwitch == 1 or os.path.isfile('/home/pi/python/stop-script'):
#Tappokytkimen tapaus
        gpio.output(13, gpio.LOW)
        os.remove('/home/pi/python/stop-script')
        print ("TERMINATED BY KILLSWITCH")
        gpio.cleanup()
        break
    else:
        #Normaalitila
        print ("ON " + str ((x+1)*100 / args.timeOn) + "%")
        time.sleep(1)
gpio.cleanup()
print ("Done.")
```