

Aleksanteri Virtaniemi

RASPBERRY PI -POHJAINEN VERKKOKELLORADIO

Tietotekniikan koulutusohjelma

2017

RASPBERRY PI -POHJAINEN VERKKOKELLORADIO

Virtaniemi, Aleksanteri
Satakunnan ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Toukokuu 2017
Sivumäärä: 28
Liitteitä: 0

Asiasanat: Raspberry Pi, Python, rinnakkaisohjelmointi, palvelinohjelmointi

Raspberry Pi on Linux-pohjainen sulautettu järjestelmä, jolle on ominaista pieni koko ja monipuoliset liitännät. Tässä työssä käytettiin Raspberry Pistä Mallia Raspberry Pi 1 Model B, johon liitettiin lisälaitteena 3,5 tuuman kosketusnäyttö. Ohjelmistokehitys tapahtui käyttämällä Pythonia, jota laajennettiin erikseen ladattavilla kolmannen osapuolen kirjastoilla.

Työssä tutkittiin Raspberry Pin ohjelmistokehitystä toteuttamalla verkkopohjainen kelloradio. Ohjelmistolle asetettiin tavoitteeksi toteuttaa ohjelmisto monipuolisilla ominaisuuksilla, kuten mahdollisuus toistaa musiikkia useista eri lähteistä ja hyödyntää näyttöä lisäominaisuuksiin, kuten säätietojen ja kalenterimerkintöjen näyttäminen.

Lopputuloksena havaittiin, että Raspberry Pi on hyvin yhteensopiva tämänkaltaisen projektin alustaksi. Laitteen pieni koko ja matala virrankulutus ovat etu tämän työn kannalta. Lisäksi monipuoliset liitännät mahdollistavat useiden ominaisuuksien toteutuksen.

NETWORK CLOCK RADIO BASED ON RASPBERRY PI

Virtaniemi, Aleksanteri

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information Technology

May 2017

Number of pages: 28

Appendices: 0

Keywords: Raspberry Pi, Python, parallel programming, server programming

Raspberry Pi is a Linux-based embedded system, characterized by small size and versatile interfaces. The model of Raspberry Pi that was used in this project was Raspberry Pi 1 Model B, that had additional 3.5-inch touchscreen display attached to it. Software development took place using Python, which was extended with separately downloadable third-party libraries.

The project studied software development on Raspberry Pi by implementing a network-based clock radio. The software was set with the goal of implementing software with versatile features such as the ability to play music from multiple sources, and utilize touchscreen display for additional features such as displaying weather information and calendar entries.

As a result, it was found that Raspberry Pi is very compatible with this kind of project. The small size and low power consumption of the device is an advantage for this project. In addition, versatile interfaces make it possible to implement multiple features.

SISÄLLYS

1	JOHDANTO	5
2	RASPBERRY PI	6
2.1	Yleistä.....	6
2.2	Raspberry Pi 1 Model B	8
3	RASPBERRY PIN KÄYTTÖÖNOTTO	10
3.1	Tarvittavat lisälaitteet ja tarvikkeet.....	10
3.2	Raspbian-käyttöjärjestelmän asentaminen	11
3.3	Raspbianin asetusten määrittäminen.....	12
3.4	Tarvittavien ohjelmistojen ja lisäosien asentaminen	16
4	OHJELMA	20
4.1	Yleistä.....	20
4.2	HTTP-palvelin	21
4.3	Pääsilmukka.....	22
4.4	Säätiedot.....	24
4.5	Tietokanta.....	24
4.6	Kalenteritiedot.....	25
4.7	Musiikkisoitin	26
4.8	Hälytyksen ajastaminen	26
5	LOPPUSANAT	27
	LÄHTEET	28

1 JOHDANTO

Raspberry Pi on Linux-pohjainen yhden piirilevyn tietokone. Se on noin luottokortinkokoinen ja siinä on monipuoliset liitännät. Laitteeseen on saatavilla useita eri käyttöjärjestelmiä ja sitä voi ohjelmoida useilla eri ohjelmointikielillä. Yksi suosituimmista ohjelmointikielistä Raspberry Pille on tässäkin työssä käytettävä Python.

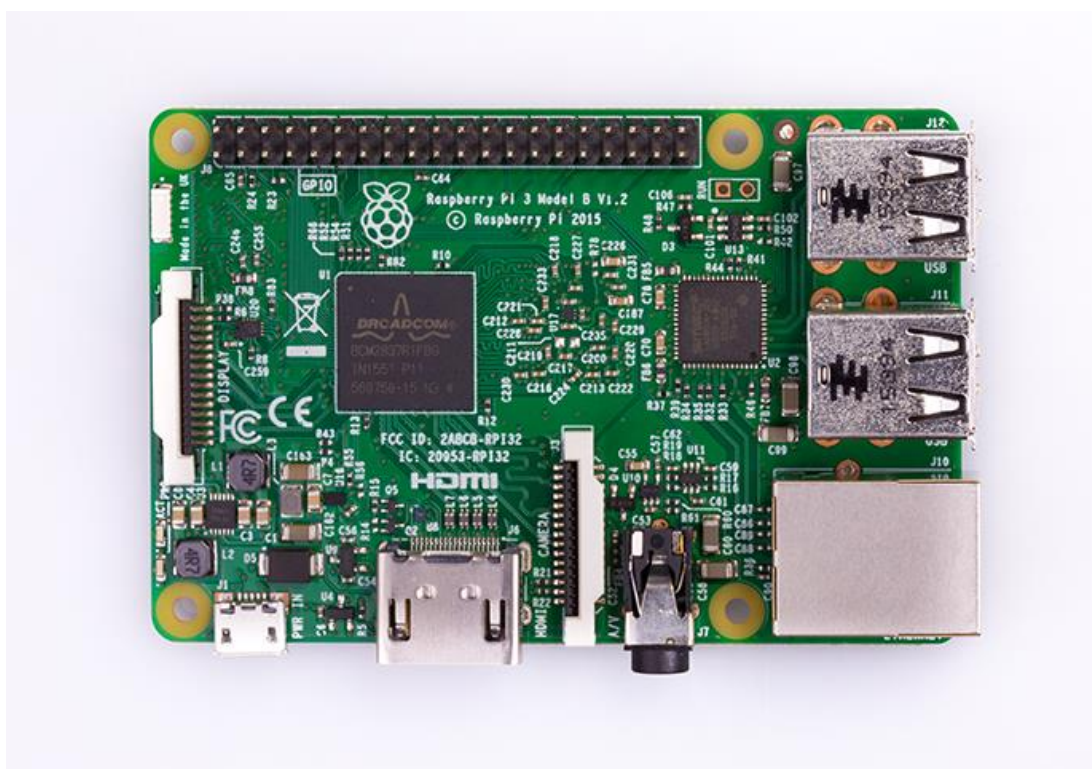
Tämä opinnäytetyö kuvaa Raspberry Pi -pienoistietokoneen päälle rakennetun verkkokelloradio toteutuksen. Työn laitteistoon kuuluu Raspberry Pin lisäksi sen GPIO-liittimeen liitettävä 3,5 tuuman TFT-näyttö. Sovelluksen ensisijainen tehtävä on aloittaa musiikin toisto ajastetusti. Toissijaisia tehtäviä on esittää näytöllä muuta informaatiota, kuten säätiedot ja seuraavat merkinnät Google Calendarista.

Käyttöjärjestelmänä toimii näytön valmistajan toimittama versio Raspbian -distribuutiosta. Raspbianista on käytössä Lite-versio, jossa ei ole graafista työpöytäympäristöä mukana, tästä johtuen näytön ohjelmoiminen tapahtuu suoraan frame bufferiin. Työssä käytetty ohjelmointikieli on Python, jota on täydennetty tarvittavilla moduuleilla.

2 RASPBERRY PI

2.1 Yleistä

Raspberry Pi on brittiläisen Raspberry Pi Foundation vuonna 2012 julkaisema yhden piirilevyn tietokone. Laite on suunnattu opetuskäyttöön kouluihin ja kehitysmaihin, mutta se on kuitenkin suosittu kaikenlaisessa muussa käytössä. Raspberry Pi toimitetaan ilman mitään lisälaitteita, kuten virtalähde, näppäimistö ja hiiri. Vuoteen 2017 mennessä Raspberry Pin eri versioita on myyty yli 10 miljoonaa maailmanlaajuisesti. (Upton & Halfacree 2013, 24.)



Kuva 1. Raspberry Pi 3 Model B (Raspberry Pi Foundationin www-sivut 29.4.2017)

Raspberry Pistä on julkaistu useita eri versioita. Ensimmäinen julkaistu malli on Raspberry Pi 1 Model B, joka julkaistiin tammikuussa 2012. Ensimmäisen mallin seuraaja on vuonna 2014 julkaistu karsittu ja edullisempi malli Model A. Vuonna 2015 julkaistiin Raspberry Pi Zero, joka on fyysisiltä mitoiltaan pienempi kuin edeltäjänsä. Zero on edeltäjiinsä verrattuna selvästi karsitumpi ja on myös kaikista malleista edullisin ja maksaa vain noin 5 dollaria. Samana

vuonna julkaistiin vielä Raspberry Pi 2 jossa on edeltäjiinsä verrattuna mm. tehokkaampi prosessi ja kaksi kertaa enemmän muistia. Tuorein vuonna 2016 julkaistu malli on Raspberry Pi 3 Model B, jonka suurimmat uudistukset ovat 64-bittinen suoritin ja sisäänrakennettu WLAN-sovitin. Raspberry Pi Zeroa lukuun ottamatta kaikki muut mallit ovat fyysisiltä mitoiltaan saman kokoisia, eli suunnilleen luottokortin kokoisia.

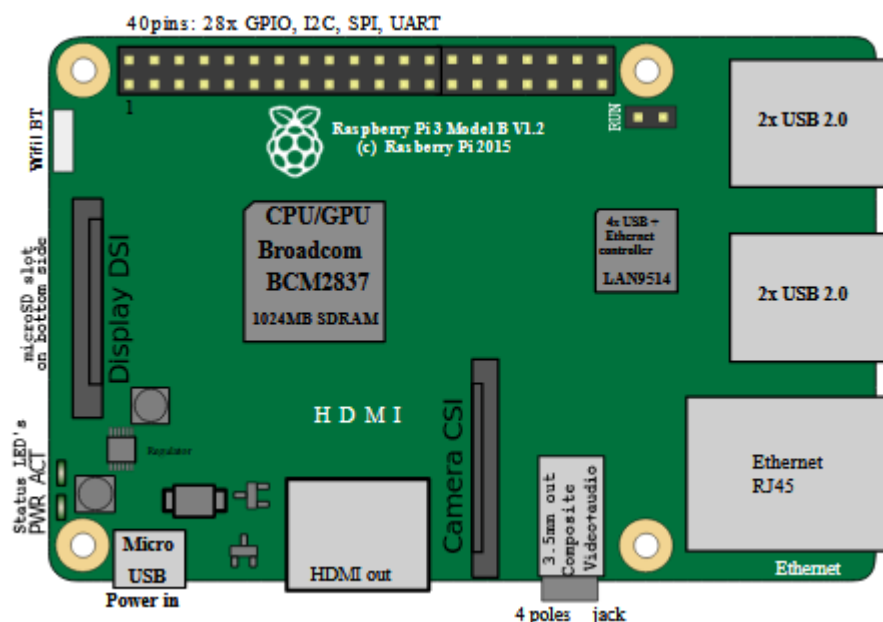
Taulukko 1. Raspberry Pin mallien vertailu

Malli	Raspberry Pi 1 Model B	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B
Suoritin	700 MHz 32-bit single-core	900 MHz 32-bit quad-core	1,2 GHz 64-bit quad-core
Muisti	256 MB	1 GB	1 GB
USB-porttien määrä	2	4	4
WLAN	ei	ei	kyllä
Virrankulutus	700 mA	600 mA	800 mA
Suositus hinta	35\$	35\$	35\$

Raspberry Pi ei sisällä käyttöjärjestelmää, vaan se pitää ladata ja kirjoittaa SD-muistikortille erikseen. Käyttöjärjestelmä tulee valita käyttötarkoituksen mukaan ja vaihtoehtoja käyttöjärjestelmäksi on useita. Suurin osa käyttöjärjestelmistä on Linux-pohjaisia, mutta myös mm. BSD-pohjaisia käyttöjärjestelmiä on olemassa ja myös ilmainen Windows-pohjainen Windows 10 IOT Core on saatavilla. Raspberry Pi Foundation suosittelee Debian-pohjaista Raspbian-käyttöjärjestelmää, josta on olemassa kaksi versiota Raspbian ja Raspbian Lite. Raspbian on työpöytäkäyttöön soveltuva graafisen työpöytäympäristön sisältävä käyttöjärjestelmä ja Lite-versio ei sisällä graafista työpöytäympäristöä. Raspberry Pin käyttöjärjestelmät voidaan jakaa kahteen kategoriaan: yleiskäyttöiset ja tiettyyn käyttötarkoitukseen kehitetyt käyttöjärjestelmät. Tiettyyn käyttötarkoitukseen soveltuvia käyttöjärjestelmiä ovat esim. multimediaseläin käyttöön tarkoitettu OSMC ja retropelikonsolien emulaatioon kehitetty Lakka.

2.2 Raspberry Pi 1 Model B

Tässä luvussa keskitytään tarkemmin opinnäytetyössä käytettävään malliin Raspberry Pi 1 Model B. Kyseinen laite on ensimmäinen versio Raspberry Pistä. Laite pohjautuu Broadcomin BCM2835-järjestelmäpiiriin. Suoritin on ARMv6Z-arkkitehtuuriin perustava 700 MHz:n yksiytiminen 32-bittinen ARM1176JZF-S. Näytönohjaimena on Broadcomin VideoCore IV, joka toimii 250 MHz:n kellotaajuudella. Muistia laitteessa on 256 MB vanhemmissa malleissa ja toukokuun 2016 jälkeen valmistetuissa malleissa muistia on 512 MB. Muisti on jaettu näytönohjaimen kanssa, ja muistinjakosuhdetta on myös mahdollista muuttaa itse konfigurointityökalun avulla.



Kuva 2. Raspberry Pi 3 Model B liittimet (Raspberry Pi Foundationin www-sivut 29.4.2017)

Raspberry Pi:ssä on monipuoliset liittännät. Virtaliittimenä toimii MicroUSB-liitin, joten virtalähteeksi kelpaa esim. matkapuhelinlaturi. Virtalähteen tulee olla laadukas, sillä huonolaatuinen virtalähde saattaa aiheuttaa muistikortin korruptoitumista tai laitteen kaatumista, erityisesti jos laitteen suorittinta ajetaan ylikellotettuna. Laitteessa ei ole sisäistä tallennusmuistia, vaan muistina käytetään SD-muistikorttia. USB-liittimiä on käytettävissä kaksi kappaletta, laitteessa on kolmeporttinen sisäänrakennettu USB-hubi, joista jäljelle jäävä kolmas portti

on käytössä USB-Ethernet-sovittimelle. Videoliittiminä on käytettävissä HDMI ja RCA-liitin komposiittivideosignaalille. Audioliittimenä on yksi 3,5 mm jakkiliitin, lisäksi audio kulkee HDMI-liittimen kautta. Lisälaitteiden ja ohjelmoitavan elektronikan kytkemiseen laitteesta löytyy 40-pinninen GPIO-liitin. Osa pinneistä voidaan käyttää toteuttamaan tekniikoita, kuten UART, mutta pinnit toimivat myös yleiskäyttöisinä GPIO-pinneinä. (Raspberry Pi Foundationin www-sivut 2017.)

3 RASPBERRY PIN KÄYTTÖÖNOTTO

Tässä luvussa käydään läpi tarvittavat asennustoiminnot, laitteen käyttökuntoon saattamiseksi. Raspberry Pin käyttöönotto on hyvin selkeää ja yksinkertaista. Käydään läpi tarvittavat lisälaitteet ja tarvikkeet, ja niiden asentaminen. Lisäksi käydään läpi käyttöjärjestelmän asentaminen ja konfigurointi ja tarvittavien ohjelmistojen ja lisäosien asentaminen.

3.1 Tarvittavat lisälaitteet ja tarvikkeet

Raspberry Pihin voidaan liittää monenlaisia oheislaitteita, riippuen mihin käyttötarkoitukseen laitetta käytetään. Yksinkertaisimmillaan riittävät pelkkä virtalähde ja muistikortti. Tässä työssä käytetään seuraavia oheislaitteita: virtalähde, muistikortti, GPIO-liitäntäinen kosketusnäyttö, verkkokaapeli ja kaiuttimet.

Tässä työssä virtalähteenä on kehitysvaiheessa käytetty tietokoneen USB-porttia USB-MicroUSB-kaapelia. Lopullisessa käyttöönotettavassa versiossa virtalähteenä toimii matkapuhelimen laturi. Muistikortille tallennetaan käyttöjärjestelmä ja se toimii laitteen tallennusmuistina. Tässä työssä muistikorttina käytetään PNY:n 8 gigatavun SD-muistikorttia.

Kosketusnäytön asentaminen on yksinkertaista, sen liittämiseksi riittää näytön liittimien työntäminen GPIO-nastoihin, mitään ylimääräisiä johtoja ei tarvita.

Verkkoyhteyden muodostamiseksi voidaan yksinkertaisesti liittää verkkokaapeli verkkoliittimeen. Vaihtoehtoisia tapoja USB-liitäntäiset WLAN-sovittimet tai mokuulat. Tässä työssä on kehitysvaiheessa käytetty verkkokaapeliliitäntä, mutta käyttöönottaessa verkkoliityntä korvataan USB-liitäntäisellä WLAN-sovittimella. WLAN-sovitinta käytetään koska laitteen verkko, virta ja audioliittimen ovat kaikki eri puolilla laitetta, tästä syystä laitteen ja johtojen sijoittaminen on haastavaa ja johtojen minimoiminen on

järkevää.



Kuva 3. Tarvittavat lisälaitteet ja tarvikkeet

3.2 Raspbian-käyttöjärjestelmän asentaminen

Tässä työssä käyttöjärjestelmä toimii Raspbian, joka on yksi suosituimmista käyttöjärjestelmistä Raspberry Pille. Raspbian on Debian Linux-jakeluun pohjautuva Raspberry Pille optimoitu käyttöjärjestelmä. Raspbianin asentamiseksi tarvitaan Raspbian-levykuva, muistikortti, ohjelmat zip-paketin purkuun ja levykuvan kirjoittamiseen muistikortille.

Raspbian-levykuva on ladattavissa ilmaiseksi Raspberry Pi Foundationin verkkosivuilta. Siitä on ladattavissa tavallinen ja lite-versio, joka ei sisällä graafista työpöytäympäristöä. Ladattu levykuva on zip-pakattu, joten se pitää purkaa ennen kuin se voidaan kirjoittaa muistikortille. Purkamiseen voidaan käyttää Linux Mintistä löytyvää Archive Manageria.

Purettu levykuva on valmis kirjoitettavaksi muistikortille. Ensimmäiseksi pitää selvittää millä nimellä muistikortin osiot näkyvät järjestelmässä. Tämä

tapahtuu seuraavalla komennolla:

```
df -h
```

Tässä tapauksessa osiot ovat nimillä `/dev/mmcblk0p1` ja `/dev/mmcblk0p2`.

Seuraavaksi osiot tulee irrottaa järjestelmästä seuraavilla komennoilla:

```
umount /dev/mmcblk0p1
```

```
umount /dev/mmcblk0p2
```

Seuraavaksi voidaan kirjoittaa levykuva muistikortille, käyttäen Linux Mintissä olevaan kopiointityökalua `dd`. `Dd` tekee kopio tiedoston sisällön bitintarkasti.

Levykuva kirjoitetaan muistikortille seuraavalla komennolla:

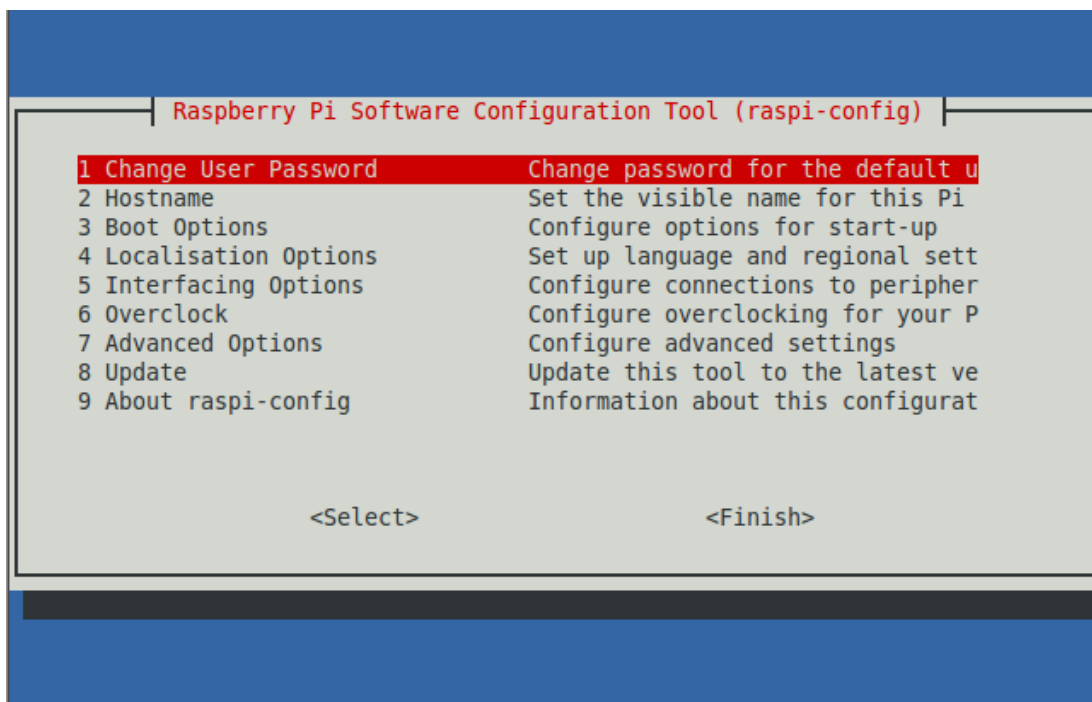
```
sudo dd bs=4M if=2017-03-02-raspbian-jessie-lite.img of=/dev/mmcblk0
```

Kirjoittamisen jälkeen muistikortti on valmis liitettäväksi Raspberry Pihin ja se on käyttövalmis sellaisenaan.

3.3 Raspbianin asetusten määrittäminen

Vaikka Raspberry Pi on käyttövalmis käyttöjärjestelmämuistikortin asettamisen jälkeen, täytyy joitain asetuksia muuttaa käyttötarkoituksen mukaan. Tähän tarkoitukseen Raspbianista löytyy ”`raspi-config`”-työkalu, joka käynnistetään komennolla:

```
sudo raspi-config
```



Kuva 4. raspi-config päävalikko

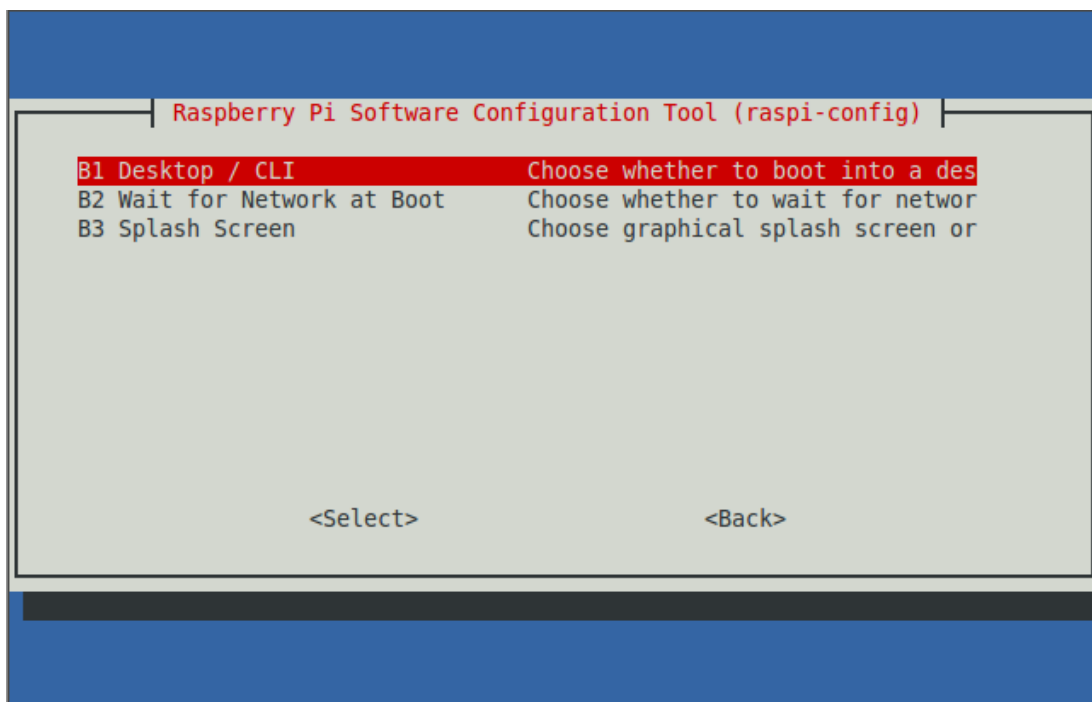
Työkalulla voidaan muuttaa niin yleisiä käyttöjärjestelmäasetuksia kuin Raspberry Pin laiteasetuksia. Asetustyökalussa navigointi tapahtuu nuolinäppäimillä ja enter-näppäimellä voidaan hyväksyä valinta. Konfiguraatiovaiheessa laitteeseen liitetään näppäimistö ja näyttö, kun konfiguraatio on saatu tehtyä, voidaan laitetta hallita etänä. Seuraavaksi käydään läpi tämän työn edellyttämät asetukset.

Change User Password -alivalikossa voidaan vaihtaa sisään kirjautuneen käyttäjän salasana. Salasanan vaihtaminen on suositeltavaa, etenkin jos laite kytketään julkiseen verkkoon. Oletussalasanan jättäminen ennalleen muodostaa melkoisen tietoturvariskin. Uutta salasanaa kirjoittaessa ruudulle ei ilmesty mitään merkkejä Linux-tyylisesti.

Hostname -alivalikossa voidaan vaihtaa tietokoneen nimi, jolla tietokone näkyy verkossa muille laitteille.

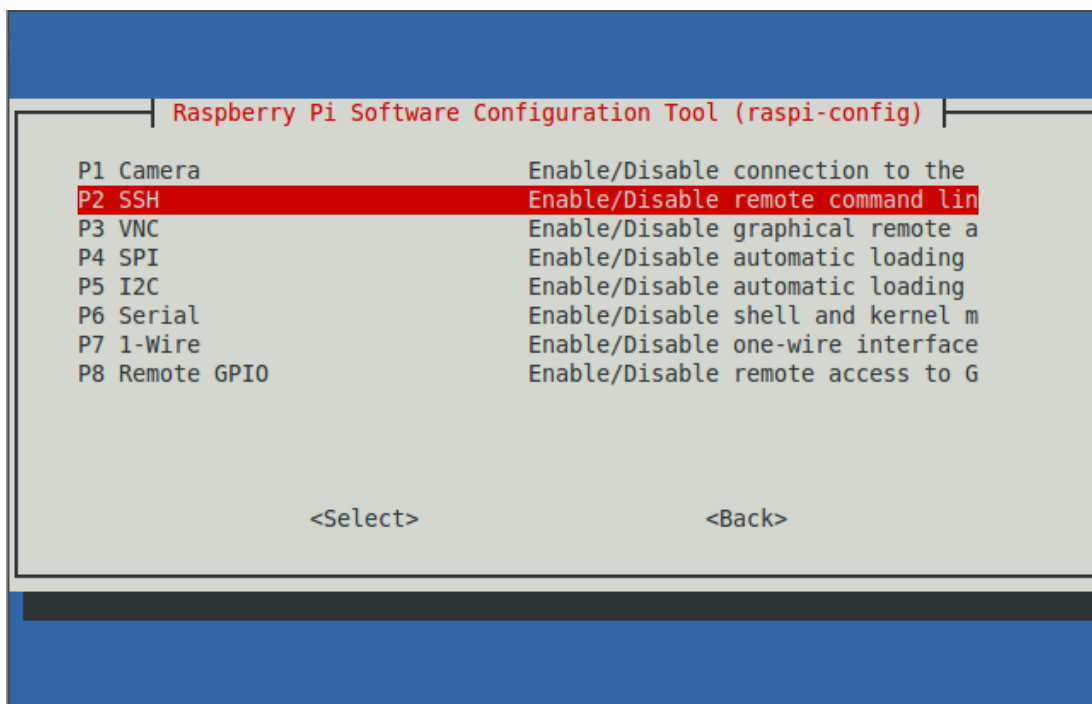
Boot options -alivalikosta löytyy asetuksia, jotka vaikuttavat siihen mitä käynnistämisen yhteydessä ladataan. Alivalikossa *Desktop / CLI* on valittavissa neljästä vaihtoehdosta *Console*, *Console Autologin*, *Desktop* ja

Desktop Autologin. Tähän työhön sopivin vaihtoehto on *Console Autologin*, joka tarkoittaa sitä, että laite käynnistyy konsoliin ja oletuskäyttäjä kirjautuu sisään automaattisesti. Toisessa alivalikossa *Wait for Network at Boot* voidaan määrittää, että jääkö käyttöjärjestelmän käynnistysprosessi odottamaan, että se saa yhteyden verkkoon. Tässä työssä verkkoyhteys on olennainen osa sovellusta, joten tämä asetus laitetaan päälle.



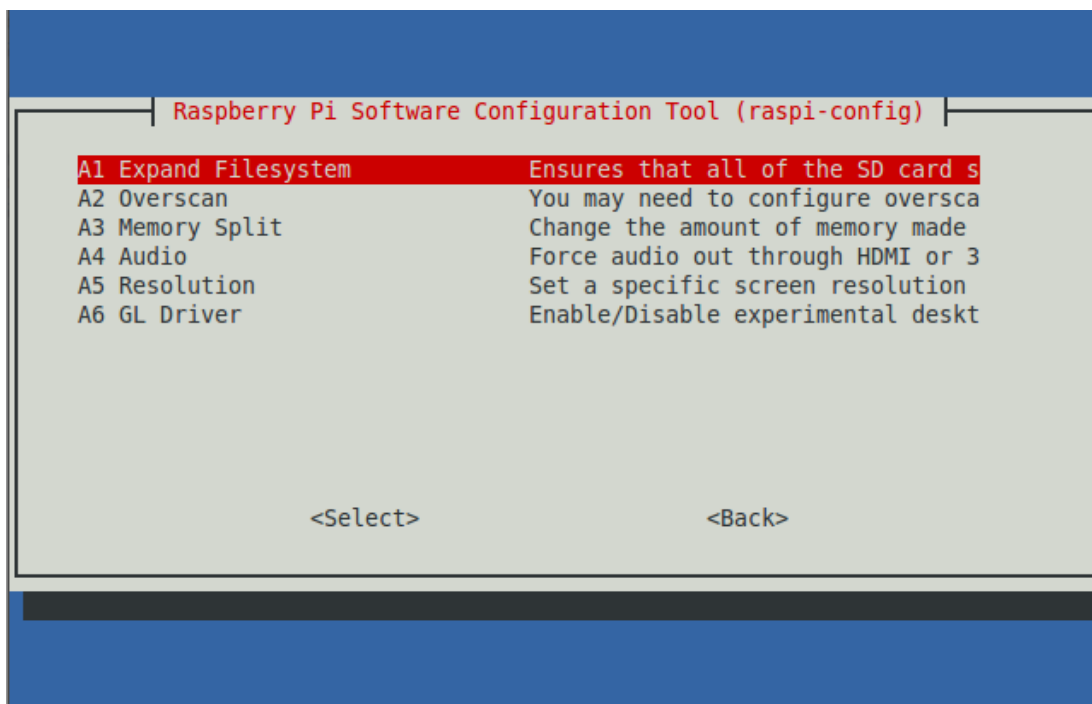
Kuva 5. Boot Options -alivalikko

Interfacing Options -alivalikosta löytyy asetuksia liitännöiden hallitsemiseen. Näistä asetuksista työn kannalta ainoa olennainen asetus on *SSH*, joka otetaan käyttöön, jotta laitetta voidaan hallita etänä. SSH:n käyttöönottamisen jälkeen ja kun muutkin asetukset on saatu kuntoon ja laite uudelleenkäynnistettyä, voidaan irrottaa näyttö ja näppäimistö laitteesta ja laitteen hallinta voidaan suorittaa etänä. Muut alivalikon asetukset voidaan ottaa pois päältä viemästä resursseja turhaan, sillä niitä ei tässä työssä tarvita.



Kuva 6. Interfacing Options -alivalikko

Advanced options -alivalikosta löytyy joukko sekalaisia asetuksia. Näistä asetuksista olennaisia ovat *Expand Filesystem* ja *Audio*. *Expand Filesystem* -asetuksella voidaan laajentaa tiedostojärjestelmä koko muistinkortin kokoiseksi, sillä kun levykuva kirjoitetaan muistikortille, niin muistikortilta varataan vain levykuvan koon edestä tilaa. Tällä asetuksella varataan muistikortin käyttämättä jäänyt tila käyttöjärjestelmän käyttöön. *Audio* -asetuksella voidaan määrittää mitä audioulostuloa laite käyttää. Vaihtoehtoina on *Auto*, *Force 3.5mm ('headphone') jack* tai *Force HDMI*. Tähän työhön sopiva vaihtoehto on *Force 3.5mm ('headphone') jack*.



Kuva 7. Advanced Options -alivalikko

Asetuksien määrittämisen jälkeen laite tulee käynnistää uudelleen komennolla:

[sudo reboot now](#)

3.4 Tarvittavien ohjelmistojen ja lisäosien asentaminen

Alkuasetuksien jälkeen voidaan siirtyä tarvittavien ohjelmistojen ja lisäosien asentamiseen. Ainoa varsinainen asennettava ohjelmisto on Mopidy, joka on music player daemon, eli musiikkisoitin jota ajetaan taustalla ja sen hallinta vaatii erillisen asiakasohjelmiston. Tämän lisäksi tarvitaan muutamina Python-moduuleja. Python-moduulien asentamiseksi asennetaan pip-paketinhallintajärjestelmä, jonka avulla Python-moduulien lataaminen ja asentaminen onnistuu yhdellä komennolla. Pip voidaan asentaa antamalla komento:

[sudo apt-get install python-pip](#)

Mopidyn asentaminen ei ole aivan näin yksinkertaista. Mopidyn asentaminen tapahtuu seuraavilla komennoilla:


```
wget -q -O - https://apt.mopidy.com/mopidy.gpg | sudo apt-key add -
```

Ladataan ja asetetaan salausavain paketinhallintajärjestelmään.

```
sudo wget -q -O /etc/apt/sources.list.d/mopidy.list
```

```
https://apt.mopidy.com/jessie.list
```

Ladataan ja asetetaan pakettilähde paketinhallintajärjestelmään.

```
sudo apt-get update
```

Päivitetään pakettilähde, jotta edellisessä vaiheessa lisätyn pakettilähteen paketit näkyvät paketinhallintajärjestelmässä.

```
sudo apt-get install mopidy
```

Ladataan ja asennetaan Mopidy paketinhallintajärjestelmästä.

Asentamisen jälkeen asennetaan vielä yksi Mopidy-lisäosa. Oletuksena Mopidy tallentaa musiikkikirjaston JSON-formaatissa, tällä lisäosalla voidaan tallennusformaatti vaihtaa SQLite-tietokantaan. Tallennusformaattia vaihdetaan, koska oletustallennusformaattissa oli ongelmia mp3-tiedostojen ID3-tagien kanssa. Lisäosa voidaan asentaa komennolla:

```
sudo pip install Mopidy-Local-SQLite
```

(Mopidyn www-sivut 2017.)

Samalla asennetaan muut työssä vaadittavat Python-moduulit. Näytön ohjaamiseen käytetään PyGame-pelikirjastoa, joka voidaan asentaa komennolla:

```
sudo pip install pygame
```

(PyGamen www-sivut 2017)

Mopidyn hallintaan käytettävä MPDClient-moduuli, joka toteuttaa asiakasohjelmistorajapinnan, jonka avulla Mopidyn hallitseminen onnistuu helposti. Moduuli asennetaan komennolla:

```
sudo pip install python-mpd2
```

Asennetaan vielä Google Client Library, jota käytetään Google Calendarin

kalenterimerkintöjen hakemiseen. Asennetaan komennolla:

```
pip install google-api-python-client
```

Mopidyn asentamisen jälkeen täytyy vielä muokata muutamia asetuksia, ennen kuin Mopidy voidaan ottaa käyttöön. Yleisten asetusten lisäksi säädetään asetukset niin, että Mopidy käynnistyy palveluna laitteen käynnistymisen yhteydessä. Asetuksia muutetaan muokkaamalla asetustiedostoa, joka sijaitsee tiedostosijainnissa */etc/mopidy/mopidy.conf*. Mopidyn ajaminen palveluna onnistuu komennolla:

```
sudo systemctl enable mopidy
```

Asetustiedostosta muutetaan seuraavat rivit:

```
[mpd]
enabled = true
hostname = localhost
port = 6600

[local]
enabled = true
library = sqlite
```

Mpd-kohdassa otetaan käyttöön Mopidyn etäohjaukseen käytettävä mpd-rajapinta. Lisäksi määritellään yhteysasetukset, *hostname*-asetukseksi määritetään localhost, koska Mopidyä ohjataan ainoastaan Python-skriptin kautta. Oletusportti jätetään ennalleen. *Local*-kohdassa otetaan käyttöön paikallisten tiedostojen käyttö ja otetaan käyttöön aikaisemmin asennettu SQLite-tallennusformaatti.

Asetetaan vielä lopuksi laite päivittämään dynaaminen domain-osoite. Tähän tarkoitukseen käytetään ilmaista suomalaista dy.fi-palvelua, jonka avulla voidaan asettaa alidomain ohjaamaan haluttuun osoitteeseen. Laite on kytketty julkiseen verkkoon ja se saa IP-osoitteen DHCP:n kautta, jonka takia laitteen IP-osoite saattaa vaihtua, jonka takia alidomain pitää automaattisesti päivittää. Palvelu myös vapauttaa alidomainin jos sitä ei uudisteta 7 päivään. Alidomainin päivittäminen onnistuu yhdellä komennolla wget-ohjelman avulla. Käytetty komento on:

```
wget --delete-after --no-check-certificate --no-proxy --user=[käyttäjätunnus] --password=[salasana] https://www.dy.fi/nic/update?hostname=[alidomain].dy.fi
```

Komento suoritetaan laitteen käynnistyessä ja ajastettuna kerran päivässä. Suoritus laitteen käynnistyessä toteutetaan lisäämällä komento */etc/rc.local* -tiedostoon. Päivittäinen ajastettu suoritus voidaan toteuttaa käyttämällä crontab-ohjelmaa ja lisäämällä sinne rivi:

```
0 0 * * * wget --delete-after --no-check-certificate --no-proxy --user=[käyttäjätunnus] --password=[salasana] https://www.dy.fi/nic/update?hostname=[alidomain].dy.fi
```

4 OHJELMA

Tässä luvussa käydään läpi ohjelmoidun sovelluksen eri osat. Ohjelmointikielenä on käytetty Python-ohjelmointikieltä, joka on valmiiksi asennettuna Raspbian-käyttöjärjestelmässä. Python on tulkittava kieli, eli ohjelmaa ei tarvitse kääntää vaan se on ajettavissa välittömästi. Python sisältää lukuisia eri kirjastoja, ja se on laajennettavissa kolmannen osapuolen kirjastoilla. Python on dynaamisesti tyyppittävä ohjelmointikieli, eli muuttujan tyyppiä ei erikseen tarvitse esitellä. Python tukee useita eri ohjelmointiparadigmoja ja sitä voidaan käyttää oliopohjaisena, funktionaalisen tai proseduraalisen ohjelmointikielenä. Tässä työssä käytetään Pythonin versiota 2.7. (Pythonin [www-sivut](http://www.python.org) 2017.)

4.1 Yleistä

Toteutettu sovellus on verkkokelloradio, jota ohjataan mobiiliapplikaatiolla. Sovellus toimii kuten tavallinenkin kelloradio, mutta verkkoliityntää hyödyksi käyttäen musiikkia voidaan toistaa esim. nettiradioista tai Spotifystä. Verkkoliityntää käytetään myös hakemaan dataa verkosta, joka esitetään laitteeseen liitettyllä näytöllä. Verkosta haettava on säätiedot Ilmatieteenlaitoksen API:sta ja seuraava kalenterimerkintä Google Calendar API:sta. Lisäksi verkkoliityntä mahdollistaa etäohjauksen.

Sovellus voidaan jakaa kahteen osaan. Ensimmäinen osa on HTTP-palvelin, joka toteutetaan Pythonin vakiokirjastoihin kuuluvalla HTTP-palvelimella. HTTP-palvelinta käytetään rajapintana vastaanottamaan ohjaukset ja antamaan tietokantaan tallennettujen herätysten tiedot. HTTP-palvelin käynnistyy omaan säikeeseen.

Toinen osa on pääsilmukka, joka toimii samalla näytön piirtosilmukkana. Silmukan sisällä päivitetään ja piirretään näytölle piirrettävät grafiikat. Piirtämisen lisäksi silmukan sisällä päivitetään verkosta haettavat säätiedot ja kalenterimerkinnät. Silmukka vastaa vielä tallennettujen herätysten

suorittamisesta ajastettuna.

4.2 HTTP-palvelin

Sovelluksen etäohjauksen käskyjen vastaanottamiseen käytetään Pythonin vakiokirjastoihin kuuluvaa HTTP-palvelinta, jonka käyttöönotaminen on hyvin yksinkertainen toimenpide. HTTP-palvelin pyörii omassa säikeessä erikseen muusta sovelluksesta. Sovelluksen ohjaamiseen käytetään JSON-enkoodattuja sanomia. Erilaisia sanomia on yhteensä kuusi, joista osa on sovelluksen toimintojen hallintaan ja osa tietokantaan tallennettujen herätysten lisäämiseen, poistoon ja muuttamiseen.

Seuraavaksi käydään läpi käytetyt JSON-sanomat. Kaikille sanomille on yhteistä "action"-tietue, joka määrittää mitä sovellus tekee vastaanottaessaan sanoman. Listatut sanomat on täytetty esimerkkidatalla.

```
{"action": "turn_off"}
```

Lopettaa käynnissä olevan hälytyksen, eli käytännössä lopettaa musiikintoiston, lähettämällä musiikkisoittimelle "playback_stop"-käskyn. Tämän jälkeen haetaan tietokannasta seuraava hälytys ja ajastetaan se hälyttämään seuraavaksi.

```
{"action": "snooze", "time": "5"}
```

Torkku-toiminto joka toimii muuten samalla tavalla kuin "turn_off"-sanoma, mutta seuraavaan hälytyksen ajastamisen sijaan, ajastamiseen käytetään "time"-tietuetta, joka on ilmoitettu minuutteina.

```
{'action': 'add_alarm', 'm': 0, 'h': 9, 'dom': '*', 'mon': '*', 'dow': '*',  
'disabled': 'false'}
```

```
{'action': 'modify_alarm', 'm': 0, 'h': 9, 'dom': '*', 'mon': '*', 'dow': '*',  
'disabled': 'false'}
```

```
{'action': 'delete_alarm', 'm': 0, 'h': 9, 'dom': '*', 'mon': '*', 'dow': '*',
```

'disabled': 'false'}

Kolmea yllä olevaa sanomaa käytetään lisäämään, poistamaan ja muuttamaan hälytyksiä, jotka ovat tallennettu tietokantaan. Aikaformaatti on lainattu Linuxin cron-ohjelmasta, jolla voidaan ajastaa toimintoja käyttöjärjestelmässä. Tietueet "m" ja "h" ovat kellonajan minuutti- ja tuntiosat. Seuraavat tietueet määrittävät päivää, "dom" on "day of month" eli kuukaudenpäivä, "mon" on "month" eli kuukausi ja "dow" on "day of week" eli viikonpäivä. Tähtimerkit tarkoittavat tässä tapauksessa, sitä että hälytys suoritetaan joka päivä. Tietue "disabled" määrittää onko hälytys otettu pois käytöstä, jolloin sitä ei suoriteta.

{"action": "get_alarms"}

Hakee ja palauttaa kaikki tietokantaan tallennetut hälytykset.

4.3 Pääsilmut

HTTP-palvelinta lukuun ottamatta muut sovelluksen osat suoritetaan pääsilmut sisällä. Pääsilmut toimii myös näytön piirtosilmukkana. Pääsilmut sisällä haetaan dataa verkosta ja tietokannasta, käsitellään sitä ja lopuksi piirretään data näytölle. Näytölle piirrettävät tiedot ovat: kellonaika, päivämäärä, aika seuraavaan hälytykseen, seuraava kalenterimerkintä, säätiedot ja musiikkisoittimen tiedot.

Ennen silmut suorittamista pitää kuitenkin tehdä joitain valmisteluja. Alustetaan tarvittavat muuttujat ja luodaan käytettävät oliot. Näytön piirtämiseen määritetään asetukset kuten resoluutio ja kokoruuduntila. Lisäksi piirtämistä varten määritetään fontit ja fonttikoot, ja ladataan käytettävät kuvat muistiin.

Pääsilmut alussa kutsutaan PyGame-olion "event.get"-metodia, koska jos mitään PyGamen metodia ei kutsuta PyGame tulkitsee sovelluksen

kaatuneen ja lopettaa toimintansa. Silmukkaan lisätään pieni viive "sleep"-komennolla, sillä muussa tapauksessa sovellus kuormittaa prosessoria 100-prosenttisesti. Lisäksi silmukan alussa tallennetaan kellonaika muuttujaan, jota käytetään viivästyttämään komentoja. Esim. kellonaika päivitetään kerran sekunnissa, mutta säätiedot kerran 10 minuutissa, sillä Ilmatieteenlaitoksen API päivittää säätiedot vain kerran 10 minuutissa.

Ensimmäisen vaiheen jälkeen silmukassa päivitetään näytölle piirrettävät tiedot. Tietoja ei päivitetä jokaisella silmukan kierroksella vaan vertaamalla aikaisemmin määritettyä aikamuuttujaa, jokaiselle päivitettävälle kohdalle määritettyyn viivemuuttujaan. Pythonissa "datetime"-objekteja voidaan vertailla "<"-operaattorilla, jos viivemuuttuja on suurempi kuin sen hetkinen aika, komento suoritetaan ja lisätään viivemuuttujaan haluttu viive.

Tietojen päivittämisen jälkeen suoritetaan piirto-operaatiot, joissa aikaisemmin päivitetty data piirretään näytölle. Ensimmäiseksi näyttö tyhjennetään käyttämällä "fill"-metodia, joka täytyy näytön määritetyllä värillä. Seuraavaksi piirretään tekstit ja kuvat. Lopuksi Näyttö päivitetään näyttämään aikaisemmin piirretyt asiat "display.update"-metodilla.



Kuva 8. Kuva piirretystä näytöstä

4.4 Säätiiedot

Yksi näytölle piirrettävä tieto on säätiiedot, jotka haetaan Ilmatieteenlaitoksen API:sta. API on käytettävissä ilmaiseksi, mutta vaatii tunnuksen luomisen, jonka jälkeen saa API-avaimen, jota käytetään, kun haetaan dataa API:sta. Tässä työssä API:sta noudetaan dataa kahdella pyynnöllä.

Ensimmäisellä pyynnöllä noudetaan lämpötila, joka päivittyy kerran 10 minuutissa. Hakua rajataan sisällyttämällä pyyntöön haku-parametreja. Haku saadaan rajattua tietylle paikkakunnalle antamalla paikkakunnan nimi "place"-parametrille. Lisäksi pyynnöllä haetaan vain tuorein lämpötilahavainto, antamalla "starttime"-parametriksi, aika-arvo joka on muodostettu vähentämällä nykyhetkestä 10 minuuttia.

Toisella pyynnöllä noudetaan "Weather Symbol", eli sääsymboli joka kuvaa säätilaa, esim. auringon tai pilven kuva. Sääsymbolit ilmoitetaan kokonaislukuina, joita vastaavat selitykset ja kuvat löytyvät Ilmatieteenlaitoksen sivuilta. Hakua rajataan paikkakunnan perusteella, kuten edellisessäkin pyynnössä. Sen lisäksi "starttime"- ja "endtime"-parametreiksi annetaan aika-arvot, jotka ovat puoli tuntia vähennetty ja lisätty nykyhetkestä. Sääsymboli perustuu sääennusteeseen, jonka takia noudetaan edellinen ja seuraava sääsymboli, joista valitaan se, joka on lähempänä nykyhetkeä.

Samassa luokassa lasketaan myös koska auringonnousu ja -lasku tapahtuvat. Ajat lasketaan käyttäen algoritmia. Saatuja tietoja käytetään, koska sääsymboleiden kuvista on olemassa päivä- ja yöversiot. Saatujen aikojen avulla määritetään, kumpia kuvia käytetään. (Ilmatieteenlaitoksen www-sivut 2017.)

4.5 Tietokanta

Laitteeseen tallennetut hälytykset tallennetaan tietokantaan. Työssä käytettäväksi tietokantajärjestelmäksi on valittu SQLite, joka pienikokoinen ja

käyttää vain vähän järjestelmän resursseja. Muista tietokantajärjestelmistä eroten SQLite ei tarvitse erillistä tietokantapalvelinta, tietokannanhallintaohjelmaa tai ODBC-yhteyttä. SQLiten tietokanta voidaan tallentaa yhteen tiedostoon tai säilyttää kokonaan tietokoneen muistissa. (SQLiten www-sivut 2017.)

Tietokannan hallintaa helpottamaan on ohjelmoitu luokka, joka toteuttaa työssä vaadittavat tietokantaoperaatiot. Luokka on hyvin yksinkertainen ja siinä on toteutettu yleisimmät operaatiot kuten lisäys, muutos ja poisto-operaatiot. Tietokannan rakenne on seuraava:

Taulukko 2. Tietokannan rakenne

Nimi	Tyyppi	Lisäasetukset
id	integer	primary key autoincrement
m	text	not null
h	text	not null
dom	text	not null
mon	text	not null
dow	text	not null
disabled	integer	default 0

4.6 Kalenteritiedot

Yksi laitteen näytölle piirrettävä tieto on seuraava kalenterimerkintä Google Calendarista. Kalenterimerkintä noudetaan Googlen API:n avulla. Google tarjoaa ohjeita ja esimerkkejä useille ohjelmointikielille, joista yksi on työssä käytettävä Python. Google tarjoama esimerkkikoodi toteuttaa suurin osin kaiken tarvittavan, ja sitä on muutettu vain hyvin vähän.

Esimerkkikoodia on muutettu kahdella tavalla. Ensimmäiseksi noudettujen merkintöjen määrä on rajoitettu yhteen. Tämän lisäksi noudettuja merkintöjä pitää käsitellä, koska ne eivät aina ole samassa formaatissa, esim. koko päivällä merkitty merkintä ei sisällä lainkaan kellonaikaa. Tästä syystä noudetut merkinnät muunnetaan yhtenevään muotoon.

4.7 Musiikkisoitin

Tietokoneen musiikin toistamisesta huolehtii Mopidy, joka on säädetty käynnistymään taustalla pyörivänä palveluna tietokoneen käynnistymisen yhteydessä. Mopidyä ohjataan MPDClient-Python-kirjaston avulla. Mopidyn ohjaamisen helpottamiseksi on luotu luokka, jossa on toteutettu metodit Mopidyyn yhdistämiseen, yhteyden tarkkailuun ja tarvittaville musiikkisoitintoiminnoille.

Ensimmäiseksi muodostetaan yhteys Mopidyyn ja skannataan mediakirjasto ja lisätään löytyneet kappaleet soittolistalle. Tämän lisäksi muutetaan muutamia asetuksia, kuten satunnaistoisto ja loputon toisto. Mopidyn tila ja yhteys tarkistetaan pääsilmukan sisällä. Silmukassa tarkistetaan, onko yhteys vielä auki ja tarvittaessa luodaan se uudestaan. Lisäksi noudetaan tietoja, kuten soiva kappale ja onko musiikintoisto käynnissä. Muut käytettävät toiminnot ovat musiikintoiston aloittaminen ja pysäyttäminen, jotka suoritetaan, kun hälytys suoritetaan tai lopetetaan.

4.8 Hälytyksen ajastaminen

Hälytysten ajastaminen tapahtuu Pythonin "Timer"-funktioilla, jolle annetaan parametreiksi funktio, joka suoritetaan halutun viiveen jälkeen ja kokonaisluku joka määrittää kuinka monen sekunnin kuluttua annettu funktio suoritetaan. Ajastin alustetaan ohjelman käynnistymisen yhteydessä ja sille annetaan sekuntimäärä, joka muodostetaan vähentämällä seuraavaksi suoritettavan hälytyksen aika nykyhetkestä. Muut ajastustoiminnot suoritetaan, kun aktiivinen hälytys sammutetaan tai käytetään torkkutoimintoa. Hälytyksen sammutuksessa toistetaan sama prosessi kuin ohjelman käynnistyessä. Torkkutoimintoa käytettäessä ajastaminen sekuntimääräksi asetetaan torkkutoiminolle asetettu sekuntimäärä.

5 LOPPUSANAT

Projekti oli hyvä keino oppia uutta ja saada ylimääräiseksi jääneelle Raspberry Pille käyttötarkoitus. Python ja sen omat ja kolmannen osapuolen kirjastot osoittautuivat loistavaksi valinnaksi tähän työhön. Kaikkiin työn eri osa-alueisiin tarvittavat lisäkirjastot löytyivät helposti ja ne hoitivat tehtävänsä kunnialla, eikä kompromissiratkaisuihin tarvinnut turvautua.

Työn ainoa osa, jota olisi voinut parantaa on työssä käytettävä näyttö. Näytön kosketusominaisuuden käyttäminen vaatisi erillisen kosketuskynän käyttämistä. Tämän lisäksi näytön valmistaja ei tarjoa minkäänlaista ohjekirjaa tai dokumentaatiota, joten kosketusominaisuuksien ohjelmoiminen olisi saattanut olla haastavaa. Kuitenkin tämä rajoite pakottaa keksimään vaihtoehtoisen ratkaisun ja projektista tulee haastavampi.

Kaupallisiin ratkaisuihin verrattuna tässä työssä on etunsa. Samaan hintaan tuskin saisi vastaavaa laitetta ostettua valmiina ratkaisuna. Laitteessa on kuitenkin monipuolisesti ominaisuuksia ja se on laajennettavissa ja muokattavissa, jos myöhemmin tulee tarvetta muille käyttötarkoituksille.

LÄHTEET

Ilmatieteenlaitoksen www-sivut 2017. Viitattu 10.4.2017
<https://www.ilmatieteenlaitos.fi>

Mopidyn www-sivut 2017. Viitattu 8.4.2017.
<https://www.mopidy.com/>

PyGamen www-sivut 2017. Viitattu 15.4.2017
<https://www.pygame.org>

Pythonin www-sivut 2017. Viitattu 10.4.2017.
<https://www.python.org/>

Raspberry Pi Foundation www-sivut 2017. Viitattu 1.4.2017.
<https://www.raspberrypi.org>

SQLiten www-sivut 2017. Viitattu 18.4.2017
<https://www.sqlite.org/>

Upton, E. & Halfacree, G. 2013. Raspberry Pi User Guide. Hoboken: Wiley.
Viitattu 28.3.2017. <http://ebookcentral.proquest.com/lib/samk/detail.action?docID=1574363>