

Opinnäytetyö (AMK)
Tietotekniikan koulutusohjelma
Sulautetut ohjelmistot
2013

Juha Mäkelä

WWW-HALLINTAPANEELI RASPBERRY PIN GPIO- PINNIEN OHJAUKSEEN



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma | Sulautetut ohjelmistot

2013 | Sivumäärä: 37

Ohjaaja: TkL Jari-Pekka Paalassalo

Juha Mäkelä

WWW-HALLINTAPANEELI RASPBERRY PIN GPIO-PINNIEN OHJAUKSEEN

Tämän opinnäytetyön tavoitteena oli luoda internetissä, lähiverkossa tai paikallisesti toimiva hallintapaneeli Raspberry Pin GPIO-pinnien asetusten määrittämiseksi selaimen kanssa. Hallintapaneelista tuli pystyä määrittämään jokaisen pinnin tiedon kulkusuunta (data direction), sekä pinnille arvo (high/low). Lisäksi tavoitteena oli edellämainittujen tietojen esittäminen ja automaattinen päivittäminen hallintapaneelissa. Paneelia piti myös olla mahdollista käyttää mobiililaitteilla, jotka eivät tue JavaScript-komentosarjakieltä. Hallintapaneeli tehtiin helpottamaan GPIO-pinnien ohjausta ja tarkkailua Raspberry Pi -laitteella.

Hallintapaneeli toteutettiin käyttäen HTML:ää, CSS:ää, JavaScriptiä ja PHP:tä. Varsinainen GPIO-pinnien ohjaus suoritettiin Raspberry Pillä ajettavilla Python-ohjelmilla. Hallintapaneelia varten piti Raspberry Pille asentaa Apache WWW-palvelin, MySQL-tietokanta ja PHP-ohjelmointikieli.

Lopputuloksena saatiin toimiva järjestelmä, jossa Raspberry Pi toimii palvelimena web-hallintapaneelille. GPIO-pinnien ohjaus onnistuu helppokäyttöisestä hallintapaneelista internetin yli.

ASIASANAT:

GPIO, HTML, JavaScript, MySQL, ohjelmointi, PHP, Python, Raspberry Pi

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Information Technology | Embedded systems

2013 | Total number of pages: 37

Instructor: Jari-Pekka Paalassalo Lic. Sc. (Tech.)

Juha Mäkelä

WEB-BASED CONTROL PANEL FOR RASPBERRY PI GPIO-PINS

The goal of this thesis was to develop a web-based control panel for Raspberry Pi GPIO (General Purpose Input Output) pins. The control system had to meet the following requirements: accessible locally, through local area network or through the internet, modifiable GPIO pin information including the data direction and the state of the pin, automatic updating of the GPIO pin information, usable with mobile devices which do not support JavaScript.

The control panel was created using HTML, CSS, JavaScript and PHP. The actual controlling of the GPIO-pins was carried out with Python scripts. In addition the Apache WWW server, the MySQL database and the PHP scripting language had to be installed.

The outcome of the project was a functional system in which the control panel is hosted by the Raspberry Pi web-server, and the GPIO pins can be easily controlled with the control panel over a network.

KEYWORDS:

GPIO, HTML, JavaScript, MySQL, programming, PHP, Python, Raspberry Pi

SISÄLTÖ

KÄYTETYT LYHENTEET	6
1 JOHDANTO	7
2 RASPBERRY PI	8
3 RASPBIANIN TYÖKALUT	11
3.1 Samba	11
3.2 Apache	12
3.3 PHP	13
3.4 MySQL-tietokanta	14
3.5 Python	16
4 WWW-HALLINTAPANEELI	18
4.1 HTML ja CSS	19
4.2 JavaScript	21
4.3 MySQL-tietokanta	25
4.4 PHP	26
4.5 Python	29
5 YHTEENVETO	36
LÄHTEET	37

KUVAT

Kuva 1. Raspberry Pi B-malli [1].	9
Kuva 2. Samba asetustiedosto.	12
Kuva 3. Apache-asetustiedosto	13
Kuva 4. phpMyAdmin aloitusnäkyvä.	16
Kuva 5. GPIO-pinnien hallintapaneeli.	20
Kuva 6. GPIO-pinnien ohjausprosessi.	30
Kuva 7. Prosessien tarkkailua Top-komentoriviohjelmalla.	35

KUVIOT

Kuvio 1. PHP:n käyttö palvelimissa 2002-2012 [8].	14
---	----

TAULUKOT

Taulukko 1. MySQL-tietokannan gpio_pin-taulu.	25
---	----

KOODIESIMERKIT

Koodiesimerkki 1. CSS tyylimäärittely.	19
Koodiesimerkki 2. JavaScript-tiedoston linkittäminen HTML-dokumenttiin	21
Koodiesimerkki 3. JavaScript-kirjastojen linkittäminen HTML-dokumenttiin	21
Koodiesimerkki 4. JavaScript-funktio GPIO-pinnin ohjaukseen	22
Koodiesimerkki 5. Hallintapaneelin päivittämiseen tarkoitettu JavaScript-funktio	23
Koodiesimerkki 6. GPIO-pinnin kuvauksen päivitys tietokantaan JavaScriptillä	24
Koodiesimerkki 7. Tietokantaan yhdistäminen PHP:llä	27
Koodiesimerkki 8. GPIO-pinnien ohjauspyynnöt PHP:llä	28
Koodiesimerkki 9. GPIO-pinnien ohjaus Pythonilla	31
Koodiesimerkki 10. Python-tarkkailuohjelman käynnistys	33

KÄYTETYT LYHENTEET

Ajax	Asynchronous JavaScript And XML, joukko sovelluskehitystekniikoita JavaScriptille.
ARM	Advanced RISC Machines, 32-bittinen mikroprosessoriarkkitehtuuri.
CSS	Cascading Style Sheet on tekniikka, jolla voidaan määritellä tyyliohjeita HTML-dokumenteille
FAT32	FAT (File Allocation Table) on Microsoftin kehittämä tiedostojärjestelmä.
GPU	Graphics Processing Unit eli grafiikkaprosessori.
HDMI	High Definition Multimedia Interface on kuvan ja monikanavaäänen siirtämiseen suunniteltu liitäntästandardi.
HTML	Hypertext Markup Language on avoimesti standardoitu kuvauskieli varsinkin internet-sivujen tuottamiseen.
HTTP	Hypertext Transfer Protocol on protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon.
IP	Internet Protocol on TCP/IP-mallin Internet-kerroksen protokolla.
PHP	Hypertext Preprocessor on ohjelmointikieli, jota käytetään erityisesti Www-palvelinympäristöissä dynaamisten www-sivujen luonnissa.
RCA	Radio Corporation of America on audio- ja videolaitteissa käytetty liitäntätapa.
SD	Secure Digital, muistikorttityyppi
SoC	Järjestelmäpiiri
SQL	Structured Query Language (SQL) on IBM:n kehittämä standardoitu kyselykieli.
USB	Universal Serial Bus on sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen.

1 JOHDANTO

Tässä opinnäytetyössä toteutetaan Raspberry Pin GPIO-pinneille Www-hallintapaneeli. Työhön kuuluu käyttöjärjestelmän asennus, kaikkien tarvittavien ohjelmistojen ja työkalujen asennus Raspberry Pille sekä varsinaisen hallintapaneelin toteutus. Hallintapaneelin tekoon käytettäviä kieliä ovat

- HTML
- CSS
- JavaScript
- PHP
- Python.

Tärkeä palanen on myös MySQL-tietokanta, jota tarvitaan hallintapaneelin toiminnassa.

Tavoitteena oli saada aikaiseksi www-hallintapaneeli, jonka avulla on helppo tarkkailla ja hallita Raspberry Pin GPIO-pinnejä. Pinneihin voi kytkeä erilaisia laitteita, järjestelmiä tai ohjauspiirejä, joiden toimintaa voi ohjata käyttäen www-hallintapaneelia paikallisesti, lähiverkon tai internetin kautta. Raspberry Pillä ja hallintapaneelilla voidaan toteuttaa helposti esimerkiksi kodin automatisointia. Esimerkiksi ulkovalojen ohjaus voidaan kytkeä kiinni GPIO-pinniin, minkä jälkeen valot voidaan laittaa päälle ja sammuttaa hallintapaneelin avulla vaikkapa langattomasti internetin kautta.

2 RASPBERRY PI

Raspberry Pi on pieni, luottokortin kokoinen, yhden piirilevyn tietokone johon on mahdollista kytkeä kiinni mm. hiiri, näppäimistö ja näyttö, kuten tavallisiin pöytätietokoneisiin. Raspberry Pin laitteisto perustuu Broadcomin BCM2835-järjestelmäpiiriin (SoC). [1]

BCM2835-järjestelmäpiiri sisältää:

- ARM1176JZF-S 700 MHz suorittimen
- VideoCore IV grafiikkaprosessorin
- 512 MB keskusmuistia.

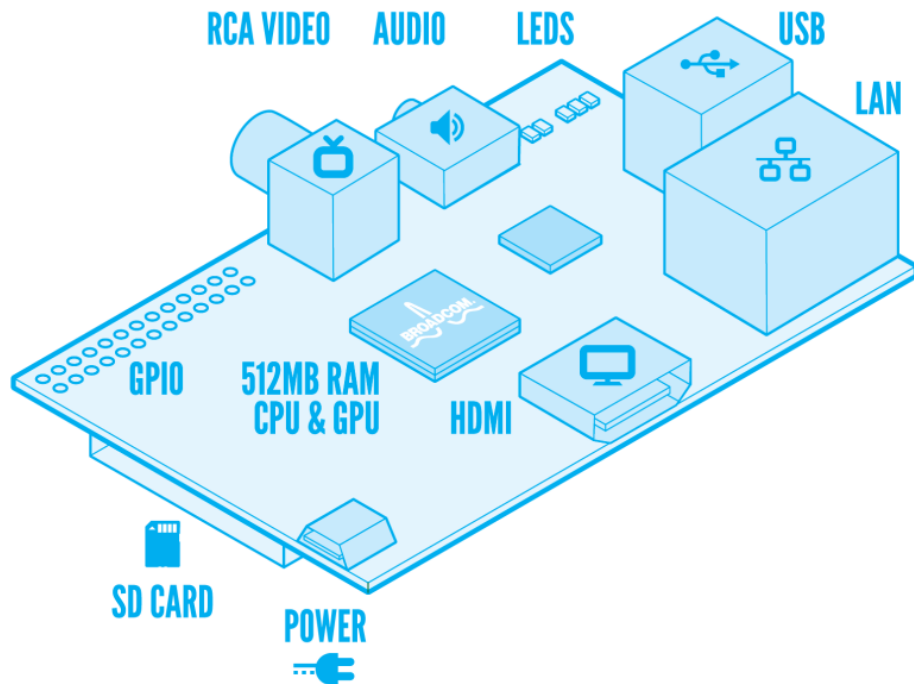
Tietokoneen on kehittänyt englantilainen Raspberry Pi Foundation –järjestö, joka aikoo julkaista Raspberry Pistä kaksi eri mallia. Tässä opinnäytetyössä käytettiin mallia B, joka on paremmin varusteltu ja hieman kalliimpi kuin A-malli. [1]

B-mallissa saatavilla olevia oheislaitteiden liitännät ovat

- RCA-videoliitäntä
- 3,5 mm:n ääniliitäntä
- kaksi USB-porttia
- Ethernet-liitäntä
- HDMI-liitäntä.

Kuvassa 1 näkyy liitännöjen sijoittelu Raspberry Pin B-mallissa. Käytettävissä on myös ohjelmoitavia GPIO-pinnejä, joita käytettiin tässä opinnäytetyössä esimerkiksi valojen ohjaukseen. Virtaa Raspberry Pi saa micro USB:n kautta verkkovirrasta jännite on 5 V ja virta hieman suorituskykyisemmässä B-mallissa 700 mA. Virtalähteenä on myös mahdollista käyttää paristoja, jolloin suosituksena on neljä AA-paristoa. Varsinaista virtanappulaa laitteessa ei ole, vaan sammutus ja käynnistys tapahtuu suoraan virtajohdon avulla. [1]

RASPBERRY PI MODEL B



Kuva 1. Raspberry Pi B-malli [1].

Raspberry Pi:ssä ei itsessään ole minkäänlaista massamuistilaitetta, mutta siinä on paikka SD-muistikortille. Muistikortille asennetaan käyttöjärjestelmä, joka vaatii tilaa noin 2 GB, joten vähintään 4 GB:n muistikortti on suositeltava. Käyttöjärjestelmän voi valita Linuxin eri jakeluversioista. [2]

Käyttöjärjestelmän asennus

Käyttöjärjestelmäksi suositellaan Linuxin Debian jakelupakettia, mutta valita voi minkä tahansa muun ARM yhteensopivan jakelupaketin. Käyttöjärjestelmäksi valittiin tätä projektia varten Raspbian-käyttöjärjestelmä, joka on Debianista Raspberry Pi:lle optimoitu versio. [2]

Käyttöjärjestelmä asennetaan muistikortille ja se vie tilaa noin 2 GB. Jotta tallennustilan kanssa ei tulisi ongelmia projektin edetessä, muistikortiksi valittiin

Kingstonin valmistama 8 GB:n muistikortti. Myös muistikortin nopeusluokkaan kiinnitettiin huomiota, jottei se hidasta laitteen toimintaa. Valitun kortin nopeusluokka on Class 10, jolloin kirjoitusnopeuden pitäisi olla vähintään 10 MB/s. [3]

Raspbianin levykuva kirjoitettiin muistikortille käyttäen Win32DiskImager-työkalua. Ennen levykuvan kirjoittamista varmistettiin, että muistikortissa on vain yksi osio (FAT32), muutoin muistikortti saattaa korruptoitua Win32DiskImager-työkalua käytettäessä. Kirjoitusprosessin valmistuttua kortti voidaan laittaa Raspberry Pin muistikorttipaikkaan, minkä jälkeen laite on valmis käynnistettäväksi. Ensimmäisen käynnistyksen yhteydessä konfiguroitiin käyttöjärjestelmän asetuksia. Osio, jossa Raspbian-käyttöjärjestelmä sijaitsee, laajennettiin koko kortin tallennuskapasiteetin suuruiseksi, jotta kortista saatiin kaikki tila käyttöön. Laajennus suoritettiin käynnistyksen yhteydessä ajettavalla raspi-config komentorivityökalulla. [4]

3 RASPBIANIN TYÖKALUT

Työkaluilla tarkoitetaan tässä yhteydessä ohjelmistoja ja ohjelmointikieliä, jotka asennettiin Raspbianille ja joita tarvittiin tämän opinnäytetyöprojektin toteuttamiseen. Raspbian-käyttöjärjestelmässä on valmiina Python-ohjelmointikieli. Se ei kuitenkaan sisällä oletuksena GPIO-pinnien ohjaukseen vaadittavaa moduulia, joka piti asentaa erikseen. Pythonin lisäksi toinen ohjelmointikieli, jota opinnäytetyöprojektissa käytettiin on PHP, jolla hoidettiin www-ohjelmointi. Muita tarvittavia työkaluja olivat verkkotyökalu Samba, WWW-palvelin Apache, tietokantaohjelmisto MySQL ja phpMyAdmin eli graafinen käyttöliittymä tietokannan hallintaan. Seuraavissa luvuissa käydään läpi kyseisten työkalujen asennus Raspbian-käyttöjärjestelmään. Projektissa käytettiin myös HTML:ää ja JavaScriptiä, joita ei tarvitse palvelinkoneelle erikseen asentaa.

3.1 Samba

Vaikka Raspbianilla pystyy käyttämään ohjelmointiympäristöjä ja kirjoittamaan ohjelmia, tässä työssä Raspberry Pin tiedostojärjestelmää käytetään Windows-pöytä tietokoneelta, joka helpottaa tiedostojen hallintaa ja ohjelmien kirjoittamista. Tämä onnistuu liittämällä Raspberry Pi samaan verkkoon Windows-koneen kanssa ja tekemällä siitä verkkolevy. Tähän tarkoitukseen asennetaan Raspbianille Samba-ohjelmisto.

Samba on vapaa ohjelmisto, jonka avulla Unix-koneet, kuten Linux-tietokoneet saadaan samaan verkkoympäristöön Microsoftin Windowsia käyttävien tietokoneiden kanssa. Samban asennus suoritetaan terminaalissa komennolla `sudo apt-get install samba smbfs`. Asennuksen jälkeen piti muokata Samban asetustiedostoa (config file), jotta Samba toimi suunnitellulla tavalla. Asetustiedosto sijaitsee polussa `/etc/samba/smb.conf`. Tiedoston loppuun tehtiin Kuvan 2 osoittamat muutokset. [5]

```

333 ; preexec = /bin/mount /cdrom
334 ; postexec = /bin/umount /cdrom
335
336 [RPI]
337     comment = Raspberry Pi
338     path = /
339     public = yes
340     writable = yes
341     guest ok = yes
342     guest only = yes
343     guest account = nobody
344     browseable = yes
345     read only = no
346     create mask = 0777
347     directory mask = 0777
348     force create mode = 0777
349     force directory mode = 0777
350

```

Kuva 2. Samba asetustiedosto.

Tiedoston muokkaamisen jälkeen Samba täytyi vielä käynnistää uudelleen komennolla `sudo /etc/init.d/samba restart`, jotta asetukset tulivat voimaan. Tämän jälkeen Windows-koneelta pystyi käyttämään Raspberry Pin tiedostoja.

3.2 Apache

Tässä opinnäytetyössä Raspberry Pin piti pystyä jakamaan HTML-dokumentteja HTTP-protokollan välityksellä, jotta www-pohjaisen GPIO-pinnien hallintapaneelin tekeminen oli mahdollista. Toisin sanoen tietokoneelle piti asentaa WWW-palvelin, joka hoitaa dokumenttien välityksen. Palvelimen tehtävä on vastaanottaa pyyntöjä muualta verkosta ja sitten vastata tähän pyyntöön lähettämällä kohdelaitteelle esimerkiksi HTML-dokumentin [6]. Tähän projektiin valittiin WWW-palvelimeksi avoimen lähdekoodin Apache HTTP Server –palvelinohjelma. Apache on palvelinohjelmista suosituin, ja sitä käyttää noin 65 % kaikista internetin palvelimista [7].

Apache on saatavilla useimmille käyttöjärjestelmille, kuten Windowsille, Linuxille ja Applen Os X:lle. Apachen asennus Raspbianille hoitui terminaalissa komennolla `apt-get install Apache2`. Asennuksen jälkeen palvelin oli

käyttövalmiina, ja toimivuutta voitiin testata ottamalla HTTP-yhteys Raspberry Pin IP-osoitteeseen selaimen kanssa. Lisämoduleita ei tämän projektin kannalta joutunut Apacheen asentamaan. Apache luo automaattisesti palvelimen käytössä olevan oletuskansion polkuun /var/www/.

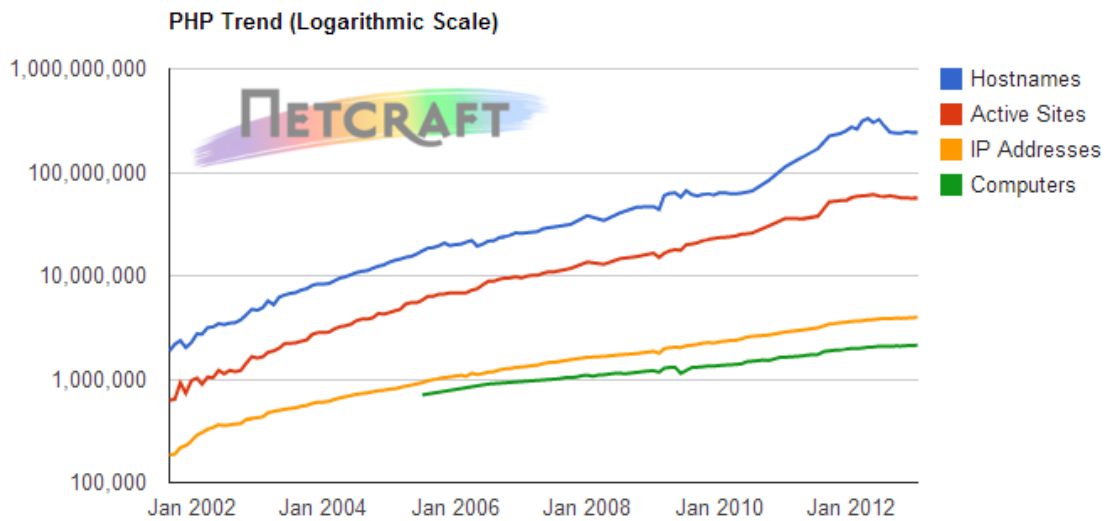
```
269  
270 ServerName localhost  
271
```

Kuva 3. Apache-asetustiedosto

Lopuksi Apachen asetustiedoston loppuun lisättiin rivi `ServerName localhost`, kuvan 3 osoittamalla tavalla. Tiedosto löytyy polusta `/etc/apache2/apache2.conf`. Rivin kanssa määritellään WWW-palvelimelle nimi, jonka avulla siihen voi ottaa yhteyden.

3.3 PHP

PHP on ohjelmointikieli, joka on yleensä käytössä www-palvelinympäristöissä, jossa sitä käytetään dynaamisten internet-sivujen luomiseen. Yleisesti PHP:n rinnalla käytetään jotain tietokantaa, usein MySQL:ää, kuten tässäkin opinnäytetyössä. PHP 5 on tällä hetkellä (2013) uusin PHP:n kehitysversio ja on käytössä arviolta kymmenissä tai sadoissa miljoonissa palvelimissa ympäri maailmaa. Kuviossa 1 on esitetty PHP:n käytön yleistymisen verkkotunnuksissa ja verkko-osoitteissa vuodesta 2002 vuoteen 2012 asti. [8]



Kuvio 1. PHP:n käyttö palvelimissa 2002-2012 [8].

PHP asennettiin Raspbianille suorittamalla terminaalissa komento `sudo apt-get install php5`. Koska varsinkin projektin kehitysvaiheessa oli tärkeää, että PHP varoitti virheistä virheviestein, oli tarpeellista muokata `php.ini`-tiedostoa ja laittaa virheilmoitukset näkyviin. Ilmoitukset laitettiin näkyviin muokkaamalla kyseisestä tiedostosta rivi `display_errors = Off` muotoon `display_errors = On`. Samaa tiedostoon piti lisätä rivi `doc_root = /var/www`, jotta PHP-kääntäjä tietää mistä etsiä PHP-tiedostoja. Asennuksen jälkeen PHP:n toiminta varmistettiin tekemällä polkuun `/var/www/` lyhyt PHP-komentosarja ja kokeiltiin näkyykö se selaimessa oikein.

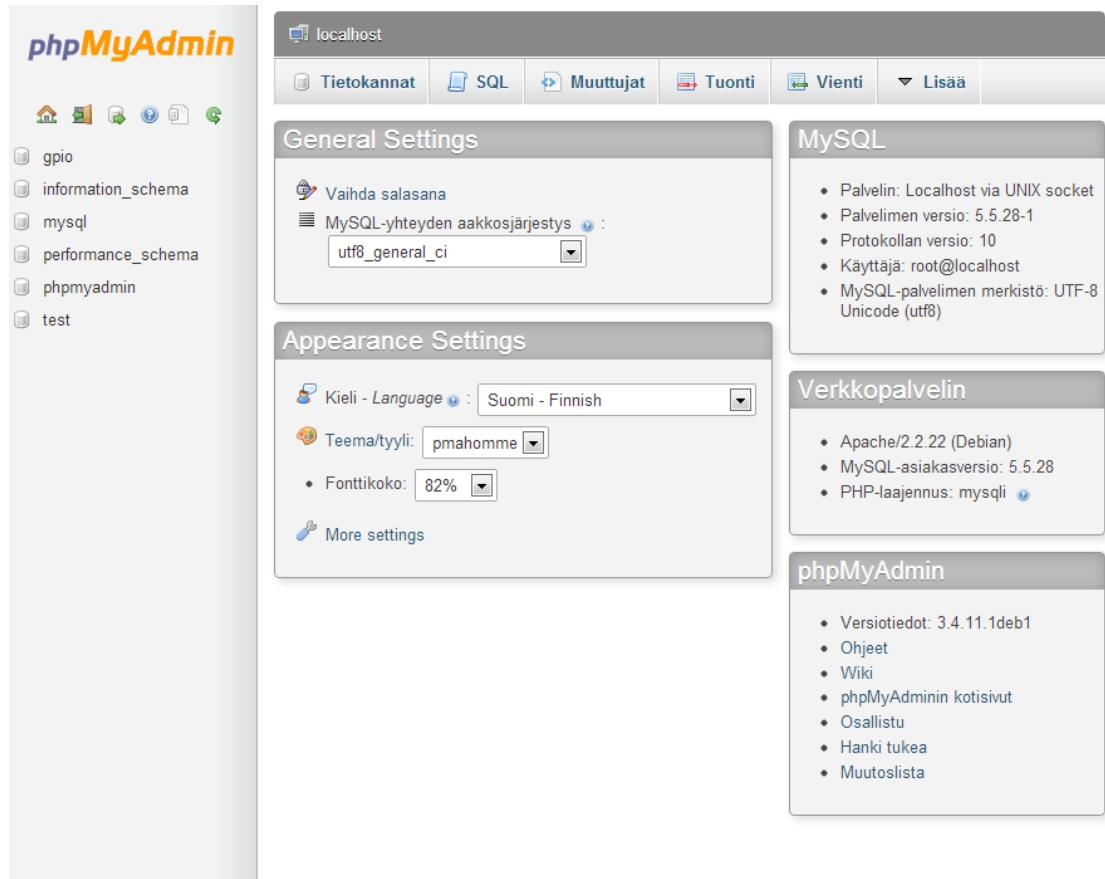
PHP:tä käytettiin tässä opinnäytetyössä tietokannasta lukemiseen, kirjoittamiseen ja päivittämiseen. Esimerkiksi GPIO-pinnien tilan lukeminen MySQL-tietokannasta hoidettiin kyselyllä, joka tehtiin PHP-komentosarjan kanssa. PHP myös käynnistää Python ohjelmakoodeja joita käytettiin mm. ohjaamaan GPIO-pinneihin jännitettä.

3.4 MySQL-tietokanta

Tietojen tallennus ja hakeminen on tärkeä osa lähes kaikissa interaktiivisissa www-sovelluksissa. Yleensä tähän tarkoitukseen käytetään jonkinlaista

tietokantaa. Tässäkin projektissa esimerkiksi GPIO-pinnien tila tallennettiin tietokantaan, josta se sitten haettiin tarpeen tullessa. Tietokantajärjestelmiä on tarjolla monia, joista suosituimpia ovat mm. MySQL, Microsoft SQL Server, SQLite, Oracle ja Microsoft Office Access. Projektiin valittiin MySQL-tietokantajärjestelmä, joka sopii käytettäväksi etenkin PHP:n kanssa ja on lisäksi sopivan kevyt ja nopea tietokanta. Raspberry pin suhteellisen vaatimattoman suorituskyvyn takia projektin MySQL-tietokannan samanaikaisten yhteyksien lukumäärä rajoitettiin 30:een yhteyteen. [9]

MySQL on saatavissa lähes kaikille alustoille, mukaan lukien Windows, Linux ja Os X. MySQL:n asennus Raspbianille suoritettiin terminaalissa komennolla `sudo apt-get install mysql-server`. Asennuksen jälkeen tietokanta on valmis käytettäväksi ja sitä voi hallita käyttämällä komentorivityökaluja. Usein rinnalle asennetaan kuitenkin jonkinlainen graafinen käyttöliittymä, joka nopeuttaa ja helpottaa suuresti tietokannan hallintaa. Etäkäytettävyyden kannalta erinomainen valinta tähän projektiin MySQL:n graafiseksi käyttöliittymäksi oli phpMyAdmin, joka on selaimen kautta käytettävä hallintatyökalu. Lisäksi se on vapaa ja avoimen lähdekoodin ohjelmisto. PhpMyAdmin sisältää kaikki tarvittavat työkalut tietokannan ylläpitoon ja on perusominaisuuksiltaan helppo käyttää. Kuvassa 4 näkyy phpMyAdminin aloitusnäkyvä, jossa vasemmalla näkyvät olemassa olevat tietokannat, yläpalkissa on käytettävissä olevat työkalut sekä käyttäjäoikeuksien hallinta ja oikealla MySQL serverin tiedot. [10]



Kuva 4. phpMyAdmin aloitusnäkyvä.

Koska phpMyAdmin on tehty PHP:llä ja on selainpohjainen työkalu, täytyy sen olla Apache-palvelimen saavutettavissa, jotta sitä voi käyttää. PhpMyAdmin asentuu oletuksena polkuun `/usr/share/phpmyadmin/`, joka ei ole Apachen ulottuvissa. Asian korjattiin tekemällä symbolinen linkki Apachen `www`-kansioon. Toimivuutta testattiin selaimella, antamalla osoitteeksi Raspberry Pin IP-osoite ja phpMyAdmin-kansio, esimerkiksi `http://192.168.0.111/phpMyAdmin/`.

3.5 Python

Projektin kannalta keskeisten GPIO-pinnien hallintaan käytettiin Python-ohjelmointikieltä. Python on korkean tason ohjelmointikieli ja on saatavilla kaikille yleisimmille käyttöjärjestelmille. Raspbianiin Python sisältyy valmiiksi

asennettuna. GPIO-pinnien hallintaan tarvittava RPi.GPIO-moduuli, sekä Python Developer Package –kehityspaketti täytyi kuitenkin asentaa erikseen. RPi.GPIO-moduuli on vapaasti ladattavissa Pythonin internet-sivuilta ja asennus tapahtui suorittamalla paketin mukana tuleva setup.py-tiedosto. Python kehityspaketti asennettiin komennolla apt-get install python-dev. [11]

Moduulin asennuksen jälkeen käytettävissä on monia funktioita, joista keskeisimpiä ovat pinnien tietosuunnan määrittäminen, sekä pinniin kirjoittaminen ja pinnistä lukeminen. Mukana on tietenkin myös muita tarpeellisia funktioita, joita käsitellään myöhemmin tässä opinnäytetyössä. Python ei kuitenkaan sovi reaaliaikaisten sovelluksien kehittämiseen tai sovellusten kehittämiseen, joissa ajoitus on kriittisessä osassa. Tämä johtuu mm. Pythonin suorittamasta arvaamattomasta roskienkeruusta. [12]

Projektissa otettiin Pythonilla yhteys MySQL-tietokantaan, jota varten piti asentaa MySQLdb-moduuli. Moduulissa tulevat kaikki tarvittavat komennot tietokannan hallintaa varten. Moduulin asennus suoritettiin terminaalissa komennolla sudo apt-get install mysqlldb. [13]

Opinnäytetyöprojektissa piti myös ajaa Python ohjelmaa käyttöjärjestelmän taustalla taustaohjelmana, eli niin sanottuna daemon-ohjelmana. Pythonille löytyy tätä tarkoitusta varten daemon-ohjelmakirjasto, joka asennettiin Raspbianille internetistä ladatun paketin avulla. [14]

4 WWW-HALLINTAPANEELI

Hallintapaneeli toteutettiin www-sivuna, josta on mahdollista määrittää jokaisen Raspberry Pin GPIO-pinnin tiedon kulkusuunta (data direction), sekä pinnille arvo (high/low). Pinneille on myös paneelin kautta mahdollisuus tallettaa kommentteja, jotka helpottavat pinnien tunnistusta paneelia käytettäessä. Hallintapaneelin tekemiseen käytettiin JavaScript, PHP ja Python-ohjelmointikieliä, sekä HTML-kuvauskieltä ja CSS-tyyliohjeita. Tiedon varastointiin käytettiin MySQL-tietokantaa. Hallintapaneelista tehtiin kaksi eri versiota. Ensimmäinen versio on tarkoitettu käytettäväksi tietokoneilta ja uudemmilta tehokailta mobiililaitteilta. Toisessa versiossa ei ole käytössä jQuery-kirjastoa, eikä JavaScript-komentosarjakieltä ja sopii näin ollen käytettäväksi vanhemmilla mobiililaitteilla.

Hallintapaneeli suunniteltiin siten, että pinnien tilamuutokset päivittyvät paneeliin, vaikka niiden tila olisi muuttunut jonkin muun kuin hallintapaneelin toimesta. Esimerkiksi jos jokin Raspberry Pillä ajettava Python-komentosarja muuttaa pinnien tilaa, tulisi sen näkyä hallintapaneelissa. Edellä mainittu ominaisuus toteutettiin Pythonilla kirjoitetulla ohjelmalla, joka tarkkailee tietyin väliajoin GPIO-pinnien tilaa, ja päivittää mahdolliset muutokset tietokantaan. Kyseinen ohjelma on mahdollista suorittaa heti Raspberry Pin käynnistyksen yhteydessä, jolloin se jää taustalle tarkkailemaan GPIO-pinneissä tapahtuvia muutoksia. Myös kaikki hallintapaneelista tapahtuvat pinnien muutokset suoritetaan Python-ohjelmilla.

Www-sivu eli varsinainen käyttäjälle näkyvä hallintapaneeli tehtiin käyttäen HTML:ää sekä CSS:ää ulkoasun toteutukseen, ja toiminnallisuus saavutettiin käyttämällä PHP ja JavaScript-ohjelmointikieliä. Taulukkomuotoisen ulkoasun todettiin olevan selkein ja käytännöllisin vaihtoehto GPIO-pinnien esittämiseen. Tiedot haetaan taulukkoon PHP:n avulla MySQL-tietokannasta. Laajemmassa hallintapaneeliversiossa käytetään JavaScriptiä ja jQueryä taulukon automaattiseen päivitykseen. Suppeammassa mobiiliversiossa taulukkoa ei

päivitetä automaattisesti, vaan käyttäjän pitää päivittää www-sivu saadakseen ajantasaisen tiedon GPIO-pinnien tilasta.

4.1 HTML ja CSS

Www-hallintapaneelin rakenne ja ulkoasu tehtiin käyttämällä HTML-kuvauskieltä ja tyylit määriteltiin CSS:n avulla. CSS:n kanssa voidaan määritellä HTML-objekteille, kuten taulukoille, omia tyylejä. Esimerkiksi voidaan asettaa mm. taulukon solujen koko, reunus ja taustaväri. Yleensä CSS-tyylitiedosto otetaan käyttöön linkittämällä se HTML-dokumentin alkuun.

GPIO-pinnejä on hallintapaneelissa kaikkiaan 13, joten parhaaksi esitystavaksi valikoitui taulukkorakenne. Taulukot ovat muokattavuutensa ja selkeytensä ansiosta HTML-kielessä hyvin yleinen tapa esittää tietoa. Taulukot koostuvat riveistä ja sarakkeista, joita jokaista voi muokata erikseen. Esimerkissä 1 on hallintapaneelin taulukon otsikkorivin tyylimäärittely CSS-tiedostossa.

Koodiesimerkki 1. CSS tyylimäärittely.

```
tr.header {  
    height: 40px;  
    text-transform: uppercase;  
    color: #fff;  
    background-color: #5368AD;  
}
```

Koodissa määritellään otsikkorivin korkeudeksi 40 pikseliä, teksti muunnetaan isoiksi kirjaimiksi, tekstin väri valkoiseksi ja taustaväri määritellään heksadesimaaliarvona tumman siniseksi, kuten Kuvassa 5 on nähtävissä.

Projektissa taulukolle määriteltiin jokaiselle GPIO-pinnille oma rivinsä sekä pinnin tiedoille omat sarakkeensa. Kuvassa 5 on hallintapaneelin laajemmassa versiossa käytetty HTML-tilukko.

1	0	PIN NUMBER	DATA DIRECTION	HI/LOW STATUS	NOTES
<input type="checkbox"/>	<input type="checkbox"/>	GPIO4	input	0	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO7	input	0	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO8	input	0	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	GPIO9	output	0	LED
<input type="checkbox"/>	<input type="checkbox"/>	GPIO10	input	0	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO11	input	0	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO17	input	0	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO18	input	0	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO22	input	0	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO23	input	0	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO24	input	0	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	GPIO25	output	1	
<input type="checkbox"/>	<input type="checkbox"/>	GPIO27	input	0	

Kuva 5. GPIO-pinnien hallintapaneeli.

Ensimmäisellä rivillä on kuvaus kunkin sarakkeen sisältämästä tiedosta, esimerkiksi DATA DIRECTION sarakeesta voidaan vaihtaa halutun pinnin tietosuunta (input/output). Muita sarakkeita ovat

- 1, josta voi asettaa pinnin tilan ykköseksi
- 0, josta voi asettaa pinnin tilan nolaksi
- PIN NUMBER, jossa on pinnin nimi
- HI/LOW STATUS, joka tarkoittaa pinnin tilaa
- NOTES, johon voi tallettaa kuvauksen kyseisestä pinnistä.

Työliien avulla pyrittiin korostamaan taulukon selkeyttä. Rivit, joiden pinnit ovat asetettu ulosmeno tilaan, näytetään sinertävällä värillä, muutoin väri on vihertävä. Samaa ideaa käytettiin myös HI/LOW STATUS –sarakeen kohdalla, jossa väri muuttuu sen mukaan, onko pinnin tila 1 vai 0. Kuvassa 5 näkyy esimerkit edellämainituista ominaisuuksista rivien GPIO9 ja GPIO25 kohdalla.

Hallintapaneeli kirjoitettiin käyttäen HTML5-versiota ja testattiin päteväksi sekä standardien mukaiseksi käyttäen W3C:n Markup Validation Service –palvelua.

4.2 JavaScript

JavaScript on komentosarjakieli, jota käytetään www-sivuilla mm. lisäämään toiminnallisuutta, tarkistamaan www-lomakkeita sekä keskustelemaan palvelimen kanssa [15]. JavaScript otetaan käyttöön www-sivulla, joko kirjoittamalla sitä suoraan HTML:n sekaan käyttäen `<script>` -tagia, taikka linkittämällä HTML-dokumentin alkuun JavaScript-tiedosto. JavaScript koodia kirjoitettiin projektissa kohtalaisen paljon, joten siitä tehtiin oma tiedostonsa. Tiedosto `script.js` linkattiin HTML-dokumenttiin koodiesimerkki 2:n kaltaisesti.

Koodiesimerkki 2. JavaScript-tiedoston linkittäminen HTML-dokumenttiin

```
<script type="text/javascript" src="./js/script.js"></script>
```

Lisäksi käytettiin JavaScriptille saataville olevaa jQuery-kirjastoa, jonka avulla mm. ajax-sovelluksien teko on huomattavasti helpompaa, kuin pelkällä JavaScriptillä. JQuerystä käytettiin versiota 1.9.1. Ennen kuin käyttäjä lataa www-sivun piti myös tarkistaa onko käyttäjä selaamassa sivua tietokoneelta vai mobiililaitteelta. Tähän tarkoitukseen käytettiin avoimen lähdekoodin mobiililaitteen tunnistusta, joka on toteutettu JavaScriptillä. Lähdekoodi on saatavilla osoitteesta <http://detectmobilebrowsers.com/>. JQuery sekä mobiililaitteen tunnistus lisättiin HTML-dokumenttiin samalla tavalla kuin `script.js`-tiedosto – mallina koodiesimerkki 3.

Koodiesimerkki 3. JavaScript-kirjastojen linkittäminen HTML-dokumenttiin

```
<script type="text/javascript" src="./js/jquery-1.9.1.min.js"></script>
<script type="text/javascript"
src="./js/detectMobile.js"></script>
```

Tässä opinnäytetyöprojektissa JavaScriptiä käytettiin monella tavalla, kuten keräämään tietoa käyttäjältä, lähettämällä tietoa palvelimelle ja hakemalla tietoa palvelimelta. Käyttäjältä kerättyyn tietoon sisältyvät napinpainallukset, alavetovalikon valinnat ja tekstikenttään kirjoitettu teksti. Edellämainitut tiedot kerättiin JavaScriptillä ja lähetettiin palvelimelle PHP-komentosarjojen käsiteltäviksi. Toisaalta PHP-komentosarjoilta myös haettiin tietoa JavaScriptin kanssa ja päivitettiin sitten tieto www-sivulle.

JavaScript-funktiot hallintapaneelissa

Hallintapaneelin script.js-tiedostossa on seuraavat kuusi funktiota:

- pageReady()
- GPIO(cmd, pin)
- pollData()
- disableButtons(buttonName, value)
- clickEdit(pin)
- clickOK(pin).

Näistä toiminnallisuuden kannalta tärkeimpiä JavaScript-funktioita ovat GPIO(cmd, pin), polldata() ja clickOK(pin). Loput funktiot ovat tarkoitettu ulkoasun ja HTML-elementtien päivittämiseen.

Kun käyttäjä muuttaa hallintapaneelin alavetovalikosta jonkin pinnin tietosuuntaa, tai muuttaa pinnin tilaa klikkaamalla nappia, suoritetaan funktio GPIO(cmd, pin).

Koodiesimerkki 4. JavaScript-funktio GPIO-pinnin ohjaukseen

```
function GPIO(cmd, pin) {
    $.ajax({
        url: 'gpio.php?action=' + cmd + '&pin=' + pin,
        complete: null
    });
}
```

Koodiesimerkki 4:n funktiossa käytetään jQuery-komentoa \$.ajax, joka suorittaa sille annetun URL-kohteen. Ajaxin käytön suuri etu verrattuna perinteiseen JavaScriptiin on, että komentosarjan suoritus etenee, vaikkei Ajax-funktiota ole vielä suoritettu loppuun [16]. Komentosarjan suoritus etenee, ja samanaikaisesti gpio.php käsittelee sille annetut arvot ja suorittaa tarvittavat toiminnot. Tällä tavoin käyttäjän kokemus hallintapaneelistä on sulava eikä www-sivua tarvitse tietojen päivittämiseksi ladata uudelleen.

Tietokannasta haettavan tiedon hakuun käytettiin funktiota pollData, joka määriteltiin ajettavaksi 500 ms:n välein. Funktiota kutsutaan automaattisesti, ilman käyttäjän pyyntöä, heti www-sivun latauduttua. Kyseistä funktiota käytetään päivittämään hallintapaneelin taulukon GPIO-pinnien tiedot.

Koodiesimerkki 5. Hallintapaneelin päivittämiseen tarkoitettu JavaScript-funktio

```
function pollData() {
    $.get("db.php", function(data) {
        for (i=0; i<data.length; i++) {
            /* luetaan db.php-tieodostosta haetut tiedot muuttujiin */
            var pin = data[i][0],
                direction = data[i][1],
                status = data[i][2],
                note = data[i][3];
            .
            .
            .
        }
        /* ajastetaan funktion suoritus 500 millisekunnin päähän */
        timeout = setTimeout(function() { pollData() }, 500);
    }, "json");
}
```

jQuery-komento \$.get on ajax komento, jolla ladataan tietoa palvelimelta käyttäen HTTP GET –pyyntöä, kuten koodiesimerkki 5:ssä. Tässä tapauksessa

komennon kanssa haettiin tietokannasta tiedot pinnien tilasta. Haetut tiedot (pin, direction, status, note) päivitetään sen jälkeen JavaScript-komennoin hallintapaneelitaulukkoon. Triviaalisempi HTML-elementtien päivitys on jätetty koodiesimerkki 5:stä pois ja korvattu for-silmukassa kolmella allekkaisella pisteellä. Kun tiedot on saatu haettua tietokannasta, kutsutaan setTimeout-funktiolla samaa pollData-funktiota uudelleen 500 ms:n kuluttua. [17]

Hallintapaneelissa on mahdollisuus pinnin muuttamisen lisäksi tallettaa tietokantaan pieni kuvaus pinnistä tai siihen liitetystä laitteesta tai toiminnasta. Pinnien kuvaukset näkyvät taulukossa NOTES-sarakkeessa, kuten esimerkiksi kuvassa 5 (s. 20) solun GPIO9 kohdalla. Kuvausta pääsee muuttamaan klikkaamalla kyseistä taulukon solua, minkä jälkeen ilmestyy tekstikenttä, johon voi kirjoittaa haluamansa kuvauksen. Teksti tallennetaan tietokantaan kun käyttäjä painaa OK-nappia. Napin painallus kutsuu clickOK-funktiota.

Koodiesimerkki 6. GPIO-pinnin kuvauksen päivitys tietokantaan JavaScriptillä

```
function clickOK(pin) {
    if ($('#div#div_note_' + pin).text() != $('#textarea#ta_' +
    pin).val()) {
        /* päivitetään uusi kuvaus tietokantaan */
        $.ajax({
            url: 'editNote.php?note=' + 'textarea#ta_' + pin).val() +
            '&pin=' + pin,
            success: null
        });
    }
}
```

Koodiesimerkissä 6 clickOK-funktio saa paremetriksi pinnin numeron, minkä kuvausta halutaan muokata. Tietokannan rasituksen pienentämiseksi tarkistetaan ennen tietokantaan päivittämistä, onko kuvaus muuttunut alkuperäisestä muodosta, kun käyttäjä painaa OK-nappia. Tietokannan päivitys tapahtuu ajax-komennolla, joka suorittaa editNote.php-komentosarjan, joka päivittää tietokantaan uuden kuvauksen kyseessä olevalle pinnille.

JavaScript siis toimii niin sanottuna välikätenä käyttäjän ja palvelimen välillä. Sen avulla otetaan vastaan käyttäjän syöttämiä tietoja, jotka lähetään eteenpäin palvelimen eli Raspberry Pin hoidettavaksi. Toisaalta JavaScript myös hakee itsenäisesti tietoa palvelimelta ja päivittää sen hallintapaneeliin käyttäjän nähtäville.

4.3 MySQL-tietokanta

Hallintapaneeli hakee tietokannasta tietoja GPIO-pinneistä. Tarvittavia tietoja ovat pinnin tietosuunta, pinnin tila ja käyttäjän pinnille antama kuvaus. Tietojen hakemisen lisäksi hallintapaneeli myös tallentaa edellämainittuja tietoja tietokantaan. Projektia varten luotiin MySQL-tietokantaan taulu nimeltä gpio_pin, johon kyseiset pinnien tiedot talletetaan (taulukko 1). Tiedot haetaan taulusta pinnin numeron mukaan, joka on jokaisella GPIO-pinnillä yksilöllinen.

Taulukko 1. MySQL-tietokannan gpio_pin-taulu.

pin_number	data_direction	hi_lo_status	note
4	output	0	
7	input	0	
8	input	0	
9	output	1	LED
10	input	0	
11	input	0	
17	input	0	
18	input	0	
22	input	0	
23	input	0	
24	input	0	
25	input	1	
27	input	0	

Taulukossa 1 on esitetty gpio_pin-tietokantataulu, kun GPIO-pinnien 4 ja 9 tietosuunta on asetettu ulosmenoksi, sekä pinnien 9 ja 25 tila on 1. Pinnille 9 on myös annettu kuvaus LED, muille pinneille ei ole asetettu kuvausta. Enempää tauluja ei ollut projektin kannalta tarpeen luoda.

4.4 PHP

PHP on palvelimella suoritettava komentosarjakieli, joka on suunniteltu www-käyttöön. Onkin hyvin yleistä, että samaan tiedostoon PHP:n kanssa kirjoitetaan myös HTML-koodia. PHP kirjoitetaan `<?php-` ja `?>`-tagien väliin, jolloin palvelin tietää, mikä osa koodista on PHP:tä. Samaa tiedostoon kirjoitettu HTML-koodi tulee siis olla edellämainittujen tagien ulkopuolella. Koska koodin PHP-osuus suoritetaan palvelimella, sitä ei pysty tarkastelemaan esimerkiksi internet-sivujen lähdekoodin kautta. Jos PHP:llä halutaan tuottaa jotain sisältöä HTML-sivulle, tehdään se PHP-komentojen `echo` tai `print` avulla. Esimerkiksi tässä projektissa hallintapaneelin taulukon rivien HTML-koodi luotiin käyttäen PHP:n `echo`-komentoa. [18]

Hallintapaneelissa PHP:tä käytettiin muutamaa eri tarkoitukseen. PHP:n kanssa automatisoitiin HTML-koodin tuottoa. Esimerkiksi taulukon rivit ja taulukon solujen sisältö tuotetaan PHP:n kanssa. PHP:tä käytettiin myös tietokantakyselyiden tekemiseen, joissa sekä haettiin tietokannasta tietoa että tallennettiin tietoa. Myös GPIO-pinnejä ohjaavien python-ohjelmien käynnistys tapahtui PHP:llä.

Tietokannan käsittely

Projektia varten tehtiin viisi eri PHP-tiedostoa, joissa jokaisessa otetaan jossain vaiheessa yhteys tietokantaan. `index.php` ja `mobile.php`-tiedostot tuottavat selaimelle HTML-koodia, ja täyttävät hallintapaneelitaulukon `db.php`-tiedoston avulla tietokannasta haetuilla tiedoilla. Kun taas `gpio.php` ja `editNote.php`-tiedostoissa tietokantaan päivitetään vain jokin tietty hallintapaneelin käyttäjän tekemä muutos. Tietokantaan yhdistämisessä käytettiin PHP:n MySQL-ohjelmointirajapintaa.

Koodiesimerkki 7. Tietokantaan yhdistäminen PHP:llä

```
//muodostetaan yhteys tietokantapalvelimeen
$connection = mysql_connect('localhost', 'root', '*****') or
die("Could not connect to server!");

//valitaan tietokanta "gpio"
mysql_select_db("gpio", $connection) or die("Could not find da-
tabase!");

//määritellään SQL-kysely
$sql = "SELECT * FROM gpio_pin";

//suoritetaan kysely
$query = mysql_query($sql, $connection) or die("Virhe kyse-
lyssä!");

// haetaan tiedot tietokannan ensimmäiseltä riviltä
$pin_number = mysql_result($haku, 0, "pin_number");
$data_direction = mysql_result($haku, 0, "data_direction");
$hi_lo_status = mysql_result($haku, 0, "hi_lo_status");
$note = mysql_result($haku, 0, "note");
```

Koodiesimerkissä 7 yhdistetään tietokantaan *gpio*, jossa on kaikkien GPIO-pinnien tila tallennettuna tietokantataulussa *gpio_pin*. Esimerkissä taulusta otetaan vain ensimmäisen rivin tiedot. Kaikkien rivien tiedot voidaan hakea for-silmukalla, jossa jokainen taulun rivi käydään läpi. Tällä tavoin tehdään mm., kun hallintapaneelin taulukon kaikki tiedot päivitetään *index.php* ja *mobile.php*-tiedostoissa, sekä JavaScript-funktion pyytäessä tietoja *db.php*-tiedostosta. Jos tietokantaan halutaan päivittää, jokin GPIO-pinnin muutos, onnistuu se muuttamalla *\$sql*-muuttujan sisältämää SQL-kyselyä.

GPIO-pinnien ohjauskäskyt

PHP:llä ei varsinaisesti ohjata GPIO-pinnejä, vaan käynnistetään tarvittava Python-ohjelma, joka hoitaa pinnien ohjauksen. Kun hallintapaneelin käyttäjä muuttaa vaikkapa pinnin tietosuuntaa, kutsuu JavaScript-funktio ajax-tekniikkaa käyttäen gpio.php-tiedostoa. JavaScript-funktio myös antaa gpio.php-tiedoston muuttujille arvot, HTTP GET –metodilla, minkä jälkeen tiedosto suoritetaan käyttämällä kyseisiä muuttujia. Esimerkiksi, kun käyttäjä muuttaa pinnin GPIO4 tietosuunnan tilaan ouput, JavaScript-funktio kutsuu gpio.php:tä nimellä *gpio.php?action=output&pin=4&mobile=false*, josta voi hakea HTTP GET –metodilla muuttujille arvot ja suorittaa tarvittavat toiminnot. Mikäli käyttäjän on tunnistettu käyttävän mobiililaitetta, muutetaan gpio.php-kutsun loppuun *&mobile=true*.

Projekti suunniteltiin siten, että kaikki hallintapaneelistä tehdyt GPIO-pinneihin liittyvät muutokset menevät gpio.php-tiedoston koodin kautta. Mahdollisia muutoksia ovat pinnin tilan asettaminen ykköseksi tai nollaksi, sekä tietosuunnan muuttaminen (input/output). Näille muutoksille määriteltiin käsittelytavat gpio.php-tiedostossa, käyttäen switch-case rakennetta.

Koodiesimerkki 8. GPIO-pinnien ohjauspyynnöt PHP:llä

```
// HTTP GET
$action = $_GET['action'];
$pin = $_GET['pin'];
$mobile = $_GET['mobile'];

switch ($action)
{
    case "hi":
        $ret_val = exec("sudo python ./python/hi_lo.py hi " . $pin);
        break;
    case "low":
        $ret_val = exec("sudo python ./python/hi_lo.py lo " . $pin);
        break;
}
```

```

case "input":
    $ret_val = exec("sudo python ./python/data_direction.py in
put " . $pin);
    break;
case "output":
    $ret_val = exec("sudo python ./python/data_direction.py out
put " . $pin);
    break;
}

// jos mobiilililaite, ohjataan takaisin mobiilisivulle
if ($mobile) header("Location: ./mobile.php");

```

Koodiesimerkistä 8 on poistettu MySQL-komennot, switch-case –rakenteen käytön esittämisen selkeyttämiseksi. Switch-case –rakenteen käyttöön päädyttiin, koska oli tarpeen vertailla vain yhtä muuttujaa (\$action) erilaisten merkkijonojen kanssa. Vertailun ollessa tosi, koodissa kutsutaan Python-ohjelmaa, joka asettaa pinnin tilan parametreissa anettujen arvojen mukaisesti. HTTP GET –metodilla haetaan myös arvo \$mobile-muuttujalle. Mikäli muuttujan arvo on true, ohjataan lopuksi käyttäjä takaisin mobile.php-tiedostoon. Normaalin käyttäjän tapauksessa näin ei tarvitse tehdä, sillä gpio.php-tiedosto suoritetaan ajax-tekniikalla.

4.5 Python

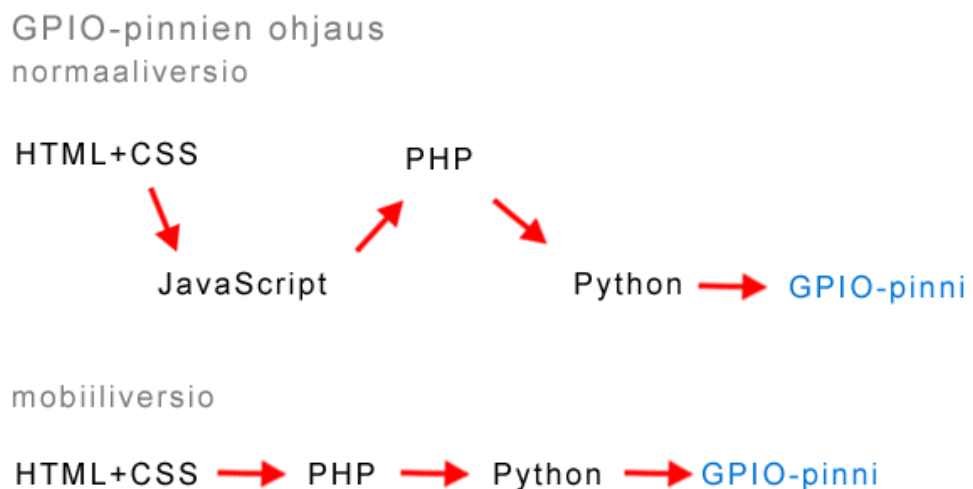
Python on korkean tason ohjelmointikieli, jonka syntaksi on moneen muuhun ohjelmointikieleen verrattuna suhteellisen helppo. Python on myös valmiiksi asennettuna Raspbian-käyttöjärjestelmässä. Edellämainittujen seikkojen pohjalta päätettiin tässä opinnäytetyöprojektissa käyttää GPIO-pinnien ohjaukseen Pythonia. Pinnien ohjauksen lisäksi oli myös tarpeellista pystyä päivittämään tietokantaa, joka on Pythonia käyttäen yksinkertaista. [11]

Projektissa Pythonia käytettiin suoraan www-hallintapaneelin yhteydessä, jolloin PHP-komentosarjan kanssa käynnistetään Python-ohjelma, joka ohjaa pinnejä

PHP:ltä saamiensa arvojen mukaisesti. Pinnien ohjaukseen kirjoitettiin kaksi erillistä python-tiedostoa. Tiedostossa `data_direction.py` olevan ohjelman kanssa muutetaan hallintapaneelin pyynnöstä pinnin tietosuuntaa ja `hi_lo.py`-tiedoston kanssa pinnin tila asetetaan ykköseksi tai nollaksi. Pythonin kanssa kirjoitettiin myös hallintapaneelista riippumaton ohjelma, joka tarkkailee GPIO-pinneissä tapahtuvia muutoksia ja päivittää mahdolliset muutokset tietokantaan. Koska kyseinen ohjelma tarkkailee jatkuvasti muutoksia, täytyy sen olla päällä pitkiä aikoja, minkä vuoksi ohjelmasta tehtiin myös käyttöjärjestelmän taustalla toimiva versio, eli niin sanottu daemon-ohjelma. Ohjelma on mahdollista suorittaa heti käyttöjärjestelmän käynnistyttyä, jolloin on suositeltavaa käyttää ohjelman daemon-tilaa.

GPIO-pinnien ohjaus

GPIO-pinnien ohjaukseen kirjoitettiin siis kaksi erillistä Python-tiedostoa – `hi_lo.py` ja `data_direction.py`. `Hi_lo.py`-tiedostolla suoritetaan pinnien tilan muutos ja `data_direction.py`-tiedostolla vaihdetaan pinnin tietosuuntaa. Kyseisten Python-ohjelmien ajo käynnistyy, kun hallintapaneelin käyttäjä vaihtaa hallintapaneelista pinnin tilaa taikka tietosuuntaa. Kuvassa 6 esitellään sekä normaalista pöytäkoneille tarkoitettu hallintapaneelista suoritettu pinnin muutos että mobiiliversiosta tehty muutos.



Kuva 6. GPIO-pinnien ohjausprosessi.

Kuvan 6 nuolet kuvaavat pinnin muutosprosessin kulkua käyttäjälle näkyvältä HTML-sivulta, aina GPIO-pinniin asti. Huomioitavaa on ero mobiiliversion ja normaaliversion välillä, joka on JavaScriptin puuttuminen mobiiliversiosta. Normaaliversiossa JavaScript hoitaa keskustelun PHP:n kanssa ajax-tekniikalla, jolloin HTML-sivua ei tarvitse ladata uudelleen jokaisen PHP-komentosarjan suorituksen jälkeen.

Kun käyttäjä muuttaa hallintapaneelistä pinnin tilan ylös tai alas, käynnistyy Raspberry Piillä hi_lo.py-tiedostosta Python-ohjelma. Ohjelmaa käynnistettäessä pitää sille määrittää argumenteiksi pinnin ohjaukskäsky, sekä ohjattavan pinnin numero. Esimerkiksi jos halutaan muuttaa pinnin GPIO25 tila ykköseksi, käynnistetään Python-ohjelma käskyllä `sudo python hi_lo.py hi 25`. Tiedostojen hi_lo.py ja data_direction.py koodien rakenne on samanlainen, niissä käytetään vain eri funktioita pinnien ohjaamiseen.

Koodiesimerkki 9. GPIO-pinnien ohjaus Pythonilla

```
import RPi.GPIO as GPIO
import sys
# luetaan argumentit muuttujiin
arg = sys.argv[1]
pin = int(sys.argv[2])

GPIO.setmode(GPIO.BCM)
GPIO.setup(pin, GPIO.OUT)

# suoritetaan haluttu GPIO-pinnin ohjaus
if arg == "hi":
    GPIO.output(pin, GPIO.HIGH)
    print GPIO.input(pin)
elif arg == "lo":
    GPIO.output(pin, GPIO.LOW)
    print GPIO.input(pin)
elif arg == "state":
    print GPIO.input(pin)
```

GPIO-pinnien ohjaukseen käytetään moduulia RPi.GPIO. Koodiesimerkissä 9:n verrataan vastaako argumentiksi annettu merkkijono oikeita arvoja, minkä jälkeen suoritetaan RPi.GPIO-moduulin kanssa GPIO-pinnien ohjaus. Funktiolla GPIO.output(pin, GPIO.HIGH) asetetaan halutun pinnin tila ykköseksi, minkä jälkeen palautetaan, käyttäen print-funktiota, pinnin tila takaisin PHP:lle, joka tarkistaa onnistuiko tilan muutos. Tiedoston data_direction.py ohjelma toimii samalla periaatteella, mutta käyttää eri funktioita pinnien tietosuunnan muuttamiseen.

GPIO-pinnien tilan tarkkailu

Hallintapaneelissa näkyy GPIO-pinneille asetettu tietosuunta ja pinnin tila. Tiedot päivittyvät aina lyhyin väliajoin tuoreisiin tietoihin. Pinnin tietojen pitää muuttua hallintapaneelissa, vaikka pinnin tilaa ei hallintapaneelista käsin muutettaisikaan. Toisin sanoen Raspberry Pillä pitää pyöriä ohjelma, joka tarkkailee GPIO-pinnien tilaa ja muutosten tapahtuessa päivittää tiedot tietokantaan, josta ne sitten päivittyvät hallintapaneeliin. Kyseinen ohjelma toteutettiin Pythonilla ja sen lisäksi käytettiin MySQLdb- ja python-daemon-moduuleita.

Ohjelmaa varten kirjoitettiin kolme tiedostoa: db.py, event.py ja startup.py. Ohjelma käynnistetään suorittamalla terminaalissa komento `sudo python startup.py normal` tai jos ohjelma halutaan käynnistää daemon-tilassa, korvataan sana `normal` sanalla `daemon`. Ohjelman voi myös laittaa käynnistymään käyttöjärjestelmän käynnistyksen yhteydessä, joka tehdään lisäämällä edellämainittu terminaalikommento tiedostoon `/etc/rc.local`, tällöin on suositeltavaa käyttää ohjelman daemon-tilaa. Käynnistyttyään ohjelma jää tarkkailemaan muutoksia GPIO-pinneissä ja sen sulkemiseksi pitää kyseessä oleva Python-prosessi lopettaa. Jos ohjelma on käynnistetty daemon-tilassa pitää se lopettaa erikseen kirjoitettavalla terminaalikomennolla, muutoin ohjelman lopetus suoritetaan painamalla näppäimistöä `ctrl+c`

näppäinyhdistelmää. Koodiesimerkki 10:ssä näkyy startup.py-ohjelman toiminta.

Koodiesimerkki 10. Python-tarkkailuohjelman käynnistys

```
import daemon, sys, db, event

# ajetaan ohjelma daemonina
def run_daemon():
    with daemon.DaemonContext():
        db.resetDB()
        event.listener()

# ajetaan ohjelma normaalisti
def run_normal():
    db.resetDB()
    event.listener()

if __name__ == "__main__":
    try:
        if sys.argv[1] == 'daemon':
            run_daemon()
        elif sys.argv[1] == 'normal':
            run_normal()
        else:
            print 'Accepted arguments: "normal" or "daemon". Try
            again.'
    except IndexError:
        print 'Give an argument. Accepted arguments: "normal" or
        "daemon". Try again.'
    except KeyboardInterrupt:
        print 'Exit program.'
```

Ohjelman käynnistyessä asettaa se kaikki GPIO-pinnit samaan, ennalta määrättyyn tilaan, jolloin vastaavasti MySQL-tietokantaan pitää päivittää kaikkien pinnien tiedot samaan tilaan. Tämän vuoksi ohjelma suorittaa aluksi

db.py-tiedostosta resetDB-funktion, joka alustaa tietokannan. Tietokannan käsittelyn jälkeen ohjelma käynnistää varsinaisen pinnien tarkkailuvaiheen, joka kirjoitettiin event.py-tiedostoon. Event.py-tiedosto koostuu kahdesta Python-funktiosta, jotka ovat updateDB(pin, value, mode) ja listener(). Pääohjelma kutsuu aluksi listener-funktiota, joka alkaa suorittamaan while-silmukkaa ja vertailee 300 millisekunnin välein kaikissa pinneissä tapahtuneita mahdollisia muutoksia. Muutoksen tapahtuessa kutsutaan updateDB-funktiota, joka päivittää tietokantaan kyseessä olevan GPIO-pinnin kohdalle uudet arvot. Jos ohjelma on käynnistetty normaali tilassa eikä daemonina, tulostaa se pinnien muutokset myös terminaali-ikkunaan. Ohjelma suoritus jatkuu, kunnes käyttäjä lopettaa suorituksen. Mitkään ohjelmat eivät kuitenkaan ole täydellisen luotettavia, ja mahdollisia ikäviä virhetilanteita voi syntyä ja ohjelman suoritus voi pysähtyä ennalta-arvaamattomista syistä. Ohjelman pysähtyttyä hallintapaneelin toiminta ja luotettavuus kärsivät, sillä GPIO-pinnien muutokset eivät enää päivity tietokantaan. Tämän opinnäytetyön tarkoituksena ei kuitenkaan ole perehtyä toimintavarmuuden parantamiseen Linux-ympäristössä, joten mahdollisen ongelman sattuessa täytyy ohjelma käynnistää uudelleen.

Koska ohjelman suorituksen on tarkoitus jatkua pitkiä aikoja ja tasaisin väliajoin tehdä prosessoria ja muuta laitteistoa rasittavia komentoja, oli kehitysvaiheessa hyvä tarkkailla sen aiheuttamaa kuormaa järjestelmälle. Tarkkailuun käytettiin kuvan 7. alaosassa näkyvää Top-komentoriviohjelmaa. Kuvan yläosassa on käynnissä samanaikaisesti GPIO-pinnien tarkkailuohjelma.

```

pi@raspberrypi ~ $ sudo python /var/www/python/startup.py
Reset database...
done.
Reset GPIO pins...
Done.

Listening for changes in GPIO pins:
UPDATE gpio_pin SET data_direction="output" WHERE gpio_pin.pin_number=9
UPDATE gpio_pin SET hi_lo_status=1 WHERE gpio_pin.pin_number=9
█

0 python listener
top - 17:51:43 up 27 min, 2 users, load average: 0.26, 0.21, 0.19
Tasks: 77 total, 1 running, 76 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.3 us, 0.7 sy, 0.0 ni, 97.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 448996 total, 170924 used, 278072 free, 15252 buffers
KiB Swap: 102396 total, 0 used, 102396 free, 81392 cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1880 www-data  20   0 40040 6564 2344  S   1.0   1.5   0:00.18 apache2
 2865 pi         20   0 4656 1352 1024  R   1.0   0.3   0:03.01 top
 2395 mysql    20   0 319m  37m 5932  S   0.7   8.5   0:11.68 mysqld
    1 root      20   0 2144   724  616  S   0.3   0.2   0:01.69 init
 1455 root      20   0 1744   508  416  S   0.3   0.1   0:00.91 ifplugd
 2847 pi        20   0 4804 1596  960  S   0.3   0.4   0:00.92 screen
 2881 root      20   0 85392 5756 2940  S   0.3   1.3   0:01.06 python
    2 root      20   0     0     0     0  S   0.0   0.0   0:00.00 kthreadd
    3 root      20   0     0     0     0  S   0.0   0.0   0:00.00 ksoftirqd/0
    5 root      20   0     0     0     0  S   0.0   0.0   0:00.23 kworker/u:0

1 top

```

Kuva 7. Prosessien tarkkailua Top-komentoriviohjelmalla.

Top-komentoriviohjelma näyttää tietoa käynnissä olevista prosesseista ja järjestelmän resursseista, kuten suorittimen ja muistin käytöstä [19]. Prosessilistasta näkyy, että Apache-palvelin, sekä MySQL-tietokanta vei suurimmat määrät muistista ja prosessorikäytöstä testin aikana. Kyseiset prosessit kuitenkin rasittivat Raspberry Piä testausvaiheessa suhteellisen vähän, sillä molemmilla oli suoritinkäyttö vain yhden prosentin luokkaa. GPIO-pinnien tilaa tarkkaileva, jatkuvasti käynnissä oleva Python-ohjelma oli listauksessa seitsemäntenä (PID 2881), ja suoritinkäyttö vain 0,3 prosenttia. Testin perusteella Raspberry pi suoriutuu hallintapaneelin ylläpitämisestä.

5 YHTEENVETO

Hallintapaneelin toteutus vaati erilaisten ohjelmistojen ja työkalujen asennusta Raspberry Pi –tietokoneelle. Kaikki tarvittavat ohjelmistot eivät olleet projektin alussa vielä tiedossa, joten tutkimustyötä piti jonkin verran suorittaa. Raspberry Pi saatiin kuitenkin käyttövalmiiksi tavoitteidein mukaisesti. Lähdemateriaalia asennusten tueksi löytyi mm. internetistä runsaasti, joka helpotti työn etenemistä.

Hallintapaneelin ohjelmoinnissa käytettiin monia eri tekniikoita - HTML:stä Pythoniin. Työ oli monipuolinen kokonaisuus, sillä se vaati internet-sivujen luomista, dynaamisuuden kehittämistä sivuille, tietokannan kanssa työskentelyä sekä Raspberry Pin GPIO-pinnien ohjausta Pythonilla. Monet käytetyistä tekniikoista eivät olleet entuudestaan tuttuja, joten työ oli opettavainen projekti varsinkin JavaScript-ohjelmoinnin ja Linuxin käytön kannalta. Tavoitteena oli toteuttaa toimiva www-hallintapaneeli, josta piti pystyä määrittämään GPIO-pinnien asetuksia. Tavoitteisiin päästiin onnistuneesti ja tuotoksena on toimiva www-hallintapaneeli.

Projektia on helppo jatkojalostaa käytettäväksi johonkin täsmällisempään tarkoitukseen, kuten erilaisten laitteiden tai moottoreiden ohjaukseen. Koska hallintapaneelin käyttöliittymä on tehty www-muotoon, on sitä helppo muokata sopivaksi moniin erilaisiin ympäristöihin. Kokonaisuudessaan työ eteni suunnittelussa aikataulussa ja lopputuloksena saatiin toimiva tuotos.

LÄHTEET

- [1] Raspberry Pi, FAQs, [www-dokumentti], saatavilla: <http://www.raspberrypi.org/faqs> (Luettu: 14.2.2013)
- [2] Raspberry Pi, Downloads, [www-dokumentti], saatavilla: <http://www.raspberrypi.org/downloads> (Luettu: 14.2.2013)
- [3] Wikipedia, Secure digital, [www-dokumentti], saatavilla: http://en.wikipedia.org/wiki/Secure_Digital (Luettu: 16.2.2013)
- [4] Embedded Linux Wiki, RPi Easy SD Card Setup, [www-dokumentti], saatavilla: http://elinux.org/RPi_Easy_SD_Card_Setup (Luettu: 16.2.2013)
- [5] Linux.com, Easy Samba Setup, [www-dokumentti], saatavilla: <http://www.linux.com/learn/tutorials/296391-easy-samba-setup> (Luettu: 18.2.2013)
- [6] Wikipedia, WWW-palvelin, [www-dokumentti], saatavilla: <http://fi.wikipedia.org/wiki/WWW-palvelin> (Luettu: 19.2.2013)
- [7] W3Techs, Usage statistics and market share of Apache for websites, [www-dokumentti], saatavilla: <http://w3techs.com/technologies/details/ws-apache/all/all> (Luettu: 19.2.2013)
- [8] PHP, Usage Stats for January 2013, [www-dokumentti], saatavilla: <http://www.php.net/usage.php> (Luettu: 20.2.2013)
- [9] dbconvert, Popular Database Management Systems Overview, [www-dokumentti], saatavilla: <http://dbconvert.com/overview.php> (Luettu: 21.2.2013)
- [10] phpMyAdmin, About, [www-dokumentti], saatavilla: http://www.phpmyadmin.net/home_page/index.php (Luettu: 23.2.2013)
- [11] Python, Python Programming Language, [www-dokumentti], saatavilla: <http://www.python.org/> (Luettu: 24.2.2013)
- [12] Python, RPi.GPIO 0.5.1a, [www-dokumentti], saatavilla: <https://pypi.python.org/pypi/RPi.GPIO> (Luettu: 24.2.2013)
- [13] MySQL-Python, MySQLdb User's Guide, [www-dokumentti], saatavilla: <http://mysql-python.sourceforge.net/MySQLdb.html> (Luettu: 24.2.2013)
- [14] Python, python-daemon 1.5.5, [www-dokumentti], saatavilla: <https://pypi.python.org/pypi/python-daemon/> (Luettu: 24.2.2013)
- [15] w3schools.com, JavaScript Tutorial, [www-dokumentti], saatavilla: <http://www.w3schools.com/js/default.asp> (Luettu: 1.3.2013)
- [16] w3schools.com, Ajax Tutorial, [www-dokumentti], saatavilla: <http://www.w3schools.com/ajax/default.asp> (Luettu: 3.3.2013)
- [17] jQuery, jQuery.get(), [www-dokumentti], saatavilla: <http://api.jquery.com/jQuery.get/> (Luettu: 3.3.2013)
- [18] Wikikirjasto, PHP, [www-dokumentti], saatavilla: <https://fi.wikibooks.org/wiki/PHP> (Luettu: 6.3.2013)
- [19] About.com, Linux / Unix Command: top, [www-dokumentti], saatavilla: http://linux.about.com/od/commands/l/blcmdl1_top.htm (Luettu: 12.3.2013)