



LÄMPÖTILAN SEURANTAJÄRJES- TELMÄ

Joni Piittala

Opinnäytetyö
Toukokuu 2015
Tietotekniikka
Tietoliikennetekniikka ja
tietoverkot

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikka
Tietoliikennetekniikka ja tietoverkot

PIITTALA, JONI:
Lämpötilan seurantajärjestelmä

Opinnäytetyö 32 sivua, joista liitteitä 3 sivua
Toukokuu 2015

Tämän opinnäytetyön tavoitteena oli toteuttaa lämpötilan seurantajärjestelmä käyttäen Raspberry Pi -mikrotietokonetta. Työn tarkoituksena oli näyttää oman oppimisen tulos ja soveltaa opittuja tietoja ja taitoja sulautettujen järjestelmien, ohjelmoinnin sekä tietokoneverkkojen osalta. Opinnäytetyönä suunniteltiin ja toteutettiin toimiva järjestelmä, joka koostuu lämpötilan mittauksesta, tiedon tallentamisesta tietokantaan ja tiedon visualisoinnista. Työssä selvitetään käytettyjen laitteistojen ja ohjelmistojen toimintaa teoriassa, mutta työn pääpaino on järjestelmän käytännön toteutuksessa.

Työn käytännön osuus koostuu mittalaitteen ja mittausdatan visualisointiin tarkoitetun web-sivun kehityksestä. Varsinainen mittalaitte päätettiin toteuttaa käyttäen Raspberry Pi -mikrotietokonetta ja sen GPIO-porttiin liitettävällä lämpötila-anturilla. Mittaus tulokset tallennetaan laitteelle tietokantaan käyttäen MySQL-tietokantapalvelinta. Tietokantapalvelun tuottamiseksi tehtiin LAMP-asennus, joka on Linux-käyttöjärjestelmään pohjautuva palvelin, johon on asennettu Apachen HTTP-palvelin, MySQL-tietokantapalvelin ja PHP-ohjelmointirajapinta.

Mittaustulosten tarkastelua varten kehitettiin yksinkertainen web-sivu, jonka avulla mittaus tulokset saadaan käyttäjälle taulukko- ja graafimuodossa. Mittausdatan tarkasteluun tarkoitetun web-sivun toteutuksessa käytettiin HTML-, JavaScript-, ja PHP-ohjelmointikieliä. Mittausdatan visualisointiin käytettiin Google Charts -piirtokirjastoa.

Työn tuloksena saatiin kehitettyä ja rakennettua toimiva lämpötilan seurantajärjestelmä, joka täytti työlle annetut vaatimukset. Työn lopussa pohditaan myös järjestelmän jatko-kehitys mahdollisuuksia ja mitä parannuksia ja uusia ominaisuuksia järjestelmään olisi mahdollista toteuttaa.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Telecommunication and Networks

PIITTALA, JONI:
Temperature monitoring system

Bachelor's thesis 32 pages, appendices 3 pages
May 2015

The purpose of this thesis was to develop a temperature monitoring system using Raspberry Pi microcomputer. The goal was to have a functioning entity which measure temperature, store the measured values in a database and then make data available via a web page. In this work, theory of used hardware and software is explained, but the main focus in this work is in the manufacture of application.

The practical part of the thesis consists of developing and creating a temperature monitor device and web page that visualize the data. The instrument was chosen to be built on Raspberry Pi microcomputer using temperature sensor connected to its GPIO port. Measurement data are saved in the MySQL database. Database service was implemented as an LAMP installation which is Linux based server environment with Apache HTTP-server, MySQL database server and PHP programming interface.

The web page was developed to allow users to inspect saved measurements in numeric and graphical form. The programming languages used in development of website were HTML, JavaScript and PHP. Graphical representations of measured data were implemented with Google Charts JavaScript library.

As a result, a functioning temperature monitor system was developed and built and the goals of the thesis were accomplished. There are still many improvements and features that could be implemented to the system in the future.

Key words: raspberry pi, linux, gpio, sensor, lamp-server, database

SISÄLLYS

1	JOHDANTO.....	6
2	RASPBERRY PI: KÄYTTÖÖNOTTO	7
2.1	Laitteisto	7
2.1.1	Raspberry Pi -tietokone.....	8
2.1.2	MicroSD-muistikortti.....	9
2.1.3	Muut tarvittavat laitteet.....	9
2.2	Käyttäjärjestelmä	10
2.2.1	Raspbian-käyttäjärjestelmä.....	10
2.2.2	Muistikortin alustus.....	10
2.2.3	Käyttäjärjestelmän asennus.....	11
2.3	Verkkoasetukset.....	12
2.3.1	Staattinen IP-osoite	12
2.3.2	SSH-etäyhteys	12
3	TIETOKANTAPALVELINYMPÄRISTÖ	14
3.1	LAMP -serveri	14
3.2	Apache-palvelinohjelmisto ja PHP	14
3.3	Tietokanta	15
3.3.1	MySQL.....	15
3.3.2	phpMyAdmin	16
3.3.3	Tietokannan luonti	16
4	LÄMPÖTILAN MITTAAMINEN	18
4.1	Lämpötila-anturi	18
4.2	Anturin kytkentä	19
4.3	Anturin lukeminen	20
4.4	Mittausdatan tallennus	21
5	MITTAUSDATAN VISUALISOINTI	24
5.1	WWW-sivu	24
5.2	PHP-koodi.....	25
5.3	Google Chart API	25
5.4	Testaus	26
6	POHDINTA.....	28
	LÄHTEET.....	29
	LIITTEET	30
	Liite 1. Python ohjelma	30
	Liite 2. WWW-sivun lähdekoodi	31

ERITYISSANASTO

Apache	WWW –palvelin
APT	Advanced Package Tool, Linux pakettinhallintatyökalu
ARM	Advanced RISC Machines, 32-bittinen mikroprosessori- arkkitehtuuri
CRC	Cyclic Redundancy Check, virheetarkistusalgoritmi
CRON	Ajastuspalvelu Unix-pohjaisille käyttöjärjestelmille.
Google Charts	JavaScript-piirtokirjasto datan visualisointiin
GPIO	General Purpose I/O, yleiskäyttöinen portti mikroproses- soreissa
HTML	Hypertext Markup Language, WWW-sivujen rakenteen ku- vaava sivunkuvauskieli
IP	Internet Protocol, Internet protokolla
JavaScript	Pääasiassa web-ympäristössä käytettävä dynaaminen komen- tosarjakieli
Kernel	Käyttöjärjestelmän ydin, joka hallinnoi ohjelmien I/O- pyyntöjä ja kääntää ne prosessorille sekä muille tietokoneen elektronisille komponenteille sopiviksi käskyiksi.
LAMP	Tietokantapalvelin (Linux, Apache, MySQL, PHP)
Linux	Linux-ydintä käyttävä Unixin kaltainen käyttöjärjestelmä
MySQL	Tietokantapalvelinsovellus
PHP	Hypertext Preprocessor, ohjelmointikieli
PuTTY	SSH-asiakasohjelma
Python	Ajonaikaisesti tulkittava ohjelmointikieli
SD	Secure Digital, yleisesti käytetty muistikorttityyppi.
SSH	Secure Shell, salattu tietoliikenneprotokolla
USB	Universal Serial Bus, tietokoneen sarjaliikenneväyläarkki- tehtuuri

1 JOHDANTO

Nykyään on hyvin edullisesti saatavilla erilaisia kehitysalustoja ja mikrokontrollereita, sekä elektroniset komponentit ja anturit ovat halventuneet koko ajan. Lisäksi internetistä on saatavilla valtava määrä tietoa, avointa lähdekoodia ja ohjelmistokirjastoja. Kun on tarpeeksi tietoa ohjelmoinnista, elektroniikasta ja web-tekniikoista, kenellä tahansa on mahdollisuuksia kehittää mielenkiintoisia sulautettuja järjestelmiä. Tämän opinnäytetyön tavoitteena oli toteuttaa järjestelmä, joka mittaa ympäristöään, tallentaa tiedon ja esittää kerätyn tiedon käyttäjälle. Tällainen järjestelmä, joka sisältää antureita ja on yhtedessä tietoverkkoihin, liittyy vahvasti esineiden internettiin.

Esineiden internetissä laitteet keskustelevat keskenään. Laitteet pystyvät erilaisten antureiden avulla havainnoimaan ympäristöään ja jotkin laitteet pystyvät jopa vaikuttamaan ympäristöönsä. Yksi esineiden internetin suurista mahdollisuuksista liittyy kodinautomaatioon ja erilaisiin älykodin järjestelmiin. Älykodin tärkeimpiä toimintoja ovat esimerkiksi automatisoitu valaistus ja lämmitys, sekä kodin toimintojen ja tapahtumien seuranta etänä, esimerkiksi älypuhelimien kautta. Tärkeä osa kotiautomaatiojärjestelmiä ovat erilaiset anturit. Kun anturi on kytketty järjestelmään, joka on yhteydessä internetiin, voidaan tätä mittaustietoa tarkastella etänä internetin kautta.

Mitä tarvitaan tiedon keräämiseen, kuinka tieto saadaan tallennettua ja kuinka tietoa voidaan tulkita ja visualisoida? Tämän työn tarkoituksena oli toteuttaa lämpötilan seurantajärjestelmä, joka mahdollistaa lämpötilan tarkkailun internetin kautta. Järjestelmä päätettiin toteuttaa Raspberry Pi -mikrotietokoneella. Raspberry Pi:n GPIO-porttiin yhdistettiin lämpötila-anturi. Prosessoitu anturitieto tallennetaan järjestelmässä tietokantaan. Raspberry Pi -tietokoneella suoritetaan myös palvelinohjelmistoa, jonka avulla ylläpidetään verkkosivuja, jotka näyttävät tietokantaan tallennetut mittaustiedot.

Opinnäytetyön alussa kerrotaan tarkemmin Raspberry Pi -tietokoneen ominaisuuksista ja käyttöönotosta. Sen jälkeen perehdytään palvelinohjelmiston ja tietokannan asennukseen. Sitten kerrotaan kuinka anturit saadaan liitettyä GPIO-porttiin ja kuinka anturin tietojen lukeminen ja tallennus tietokantaan tapahtuu. Lopuksi keskitytään web-sivun toteutukseen ja mittausdatan visualisointiin.

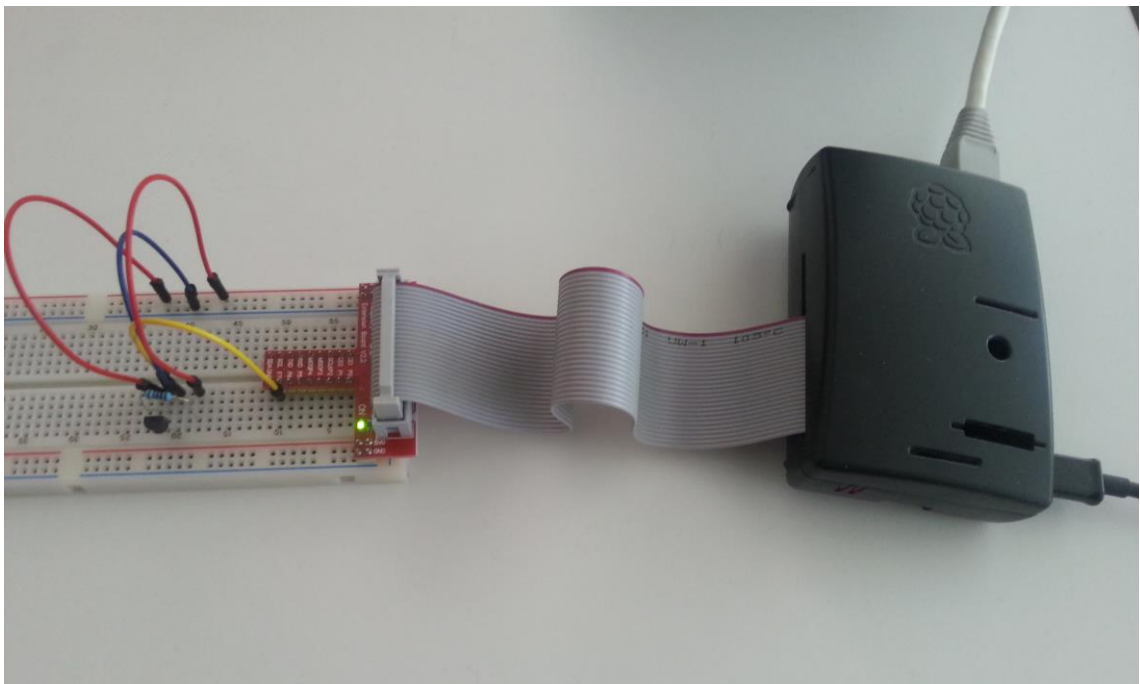
2 RASPBERRY PI: KÄYTTÖÖNOTTO

Tässä luvussa käsitellään lyhyesti Raspberry Pi -tietokoneen historiaa, teknisiä ominaisuuksia sekä perehdytään laitteen käyttöönottoon lähtien käyttöjärjestelmän asentamisesta. Käyttöjärjestelmän asennuksen jälkeen perehdytään verkkoasetuksiin, jotka ovat olennaisia Raspberry Pi:n etäohjauksen ja web-serverin toiminnan kannalta.

Tämän luvun jälkeen päästään toteuttamaan varsinaista lämpötilan seurantajärjestelmää, joka koostuu palvelinohjelmistosta, tietokannasta ja GPIO-porttiin kytketystä anturista.

2.1 Laitteisto

Raspberry Pi on täysin toimiva tietokone, joka tarvitsee lisäksi muutaman lisävarusteen jotta se saadaan toimintakuntoon. Raspberry Pi:n käyttöönottoa varten tarvitaan vähintään MicroSD-muistikortti, USB-virtalähde, näyttö ja näppäimistö. Lämpötilan seurantajärjestelmän toteuttamiseen tarvittavaan anturiin perehdytään luvussa neljä. Toimintavalmis järjestelmä sisältää Raspberry Pi -tietokoneen ja kytkentäalustan, johon lämpötila-anturi on kytketty. Lisäksi Raspberry Pi:hin on kytketty virtajohto ja verkkokaapeli (kuva 1).



KUVA 1. Lämpötilan seurantajärjestelmän laitteisto

2.1.1 Raspberry Pi -tietokone

Raspberry Pi on edullinen ja pienikokoinen yhden piirilevyn tietokone. Se toimii ARM-mikroprosessorilla ja suunniteltu käytettäväksi Linux-käyttöjärjestelmällä. Raspberry Pi -mikrotietokone on tarkoitettu lähinnä ohjelmoinnin opetteluun, mutta Linux-käyttöjärjestelmän ja monipuolisten liitännöiden ansiosta sen käyttömahdollisuudet ovat lähes rajattomat. Tietokoneen on kehittänyt brittiläinen Raspberry Pi Foundation. Tällä hetkellä laitteesta on julkaistu viisi erilaista mallia: Raspberry Pi Model A, Model A+ Model B, Model B+ sekä Raspberry Pi 2 Model B. (Raspberry Pi –säätio 2015.) Tässä työssä käytettiin Raspberry Pi -tietokoneen B+-mallia (kuva 2).



KUVA 2. Raspberry Pi B+ -tietokone (Raspberrypi.org)

Raspberry Pi -tietokone valittiin työssä käytettäväksi alustaksi, Linux-käyttöjärjestelmän, laajan ohjelmistotuen ja edullisen hinnan vuoksi. Linux-käyttöjärjestelmä oli tarpeen, koska mittaustulokset haluttiin tuoda web-sivulle. Myös anturien luku onnistuu suoraan GPIO-portin kautta. Raspberry Pi:n eri malleista käyttöön valikoitui Model B+ lähinnä kattavampien liitännöiden vuoksi. Uudempi Raspberry Pi 2 -malli ei ollut työn aloitushetkellä vielä saatavilla. Alla olevassa taulukossa on esitelty kyseisen mallin ja hieman vanhemman B-mallin keskeisiä ominaisuuksia.

TAULUKKO 1. Raspberry Pi B ja B+ mallien tekniset tiedot. (Raspberrypi.org)

Raspberry Pi	Model B	Model B+
Järjestelmäpiiri	Broadcom BCM2835	Broadcom BCM2835
RAM	512 MB SDRAM	512 MB SDRAM
Massamuisti	SD-muistikortti	MicroSD-muistikortti
USB 2.0 -portit	2	4
Virrankulutus	750 mA (3,5 W)	600 mA (3,0 W)
GPIO	26	40
koko	85,60 mm x 56 mm	85,60 mm x 56 mm

2.1.2 MicroSD-muistikortti

Raspberry Pi käyttää tallennusmediana MicroSD-muistikorttia. Muistikortti on tallennusmediana pienikokoinen, äänetön ja vähävirtainen mutta suorituskyky on huomattavasti heikompi kuin perinteisellä kovalevyllä. Sopivan muistikortin valinnassa olennaisinta on tallennustilan koko ja kirjoitusnopeusluokka. On suositeltavaa käyttää ainakin neljän gigatavun muistikorttia. Kirjoitusnopeusluokka vaikuttaa laitteen kokonaissuorituskykyyn. Tässä työssä käytettiin SanDisk Ultra 8Gt Class 10 -MicroSD-muistikorttia, jonka vähittäiskirjoitusnopeudeksi on määritetty kymmenen megatavua sekunnissa.

2.1.3 Muut tarvittavat laitteet

Virtalähteenä toimii kännykkälaturi MicroUSB-liitännällä. Malli B+ käyttää virtaa vähintään 600 mA. Virrankäyttö voi kuitenkin nousta huomattavasti, jos käytetään useita USB-lisälaitteita. Suositeltavaa onkin käyttää virtalähdettä, joka antaa virtaa 2 A 5 voltin jännitteellä. Internet yhteyttä varten tarvitaan ethernet-verkkokaapeli. Käyttöjärjestelmän asennuksen ajaksi tarvitaan USB-hiiri ja -näppäimistö, sekä HDMI-kaapeli ja yhteensopiva näyttö. Kun käyttöjärjestelmä on asennettu, voidaan ottaa käyttöön SSH-yhteys. Tällöin laitteeseen ei tarvi olla kytkettyä näyttöä, hiirtä ja näppäimistöä, koska laitetta voidaan ohjata etänä toisella tietokoneella.

2.2 Käyttöjärjestelmä

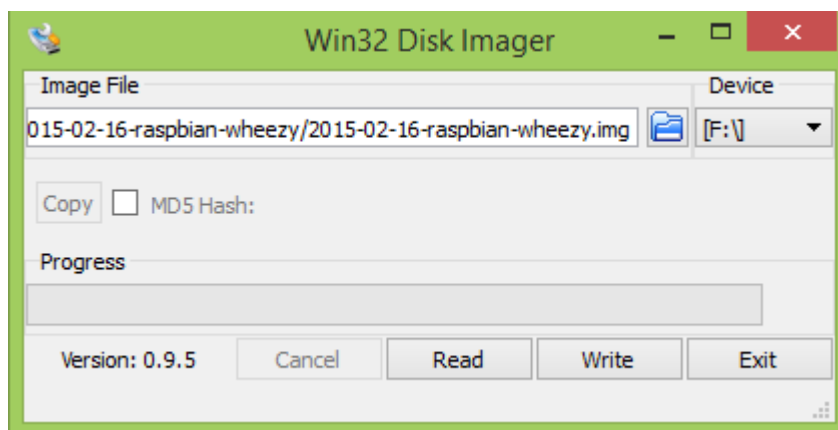
Raspberry Pi toimii ARM-mikroprosessorilla ja on suunniteltu käytettäväksi Linux-käyttöjärjestelmällä. Raspberry Pi:lle on saatavilla laaja valikoima erilaisia Linux-pohjaisia käyttöjärjestelmiä. Suosituin käyttöjärjestelmä Raspberry Pi:lle on Raspbian ja sitä käytetään myös tässä työssä.

2.2.1 Raspbian-käyttöjärjestelmä

Raspbian on Debian-käyttöjärjestelmästä muokattu ja optimoitu versio Raspberry Pi:lle. Debian on eräs vanhimpia ja käytetyimpiä Linux-jakeluita ja se tarjoaa erittäin laajan pakettivalikoiman. Raspbian-käyttöjärjestelmän mukana tulee valtava määrä valmiiksi käännettyjä ohjelmistoja, jotka on optimoitu parhaan suorituskyvyn takaamiseksi. Raspbian asennetaan muistikortille, jolta Raspberry Pi käynnistää sen. (Raspbian jakelupaketin kotisivut 2015.)

2.2.2 Muistikortin alustus

Muistikortin alustusta varten tarvitaan toinen tietokone ja SD-kortinlukija. Ensin ladataan viimeisin versio käyttöjärjestelmästä esim. www.raspberrypi.org/downloads -sivustolta. Tämän jälkeen tarvitaan Win32DiskImager-ohjelma, jolla käyttöjärjestelmän image-tiedosto asennetaan muistikortille (kuva 3).



KUVA 3. Käyttöjärjestelmän asennus muistikortille

2.3 Verkkoasetukset

Verkkoyhteys saadaan helposti käyttöön kytkemällä laitteeseen ethernet-verkkokaapeli, jonka jälkeen Raspberry Pi osaa hakea IP-osoitteen automaattisesti. Tämän jälkeen voidaan tarkistaa IP-osoite, jonka se on saanut reitittimeltä komennolla "ifconfig". Raspberry Pi:n reitittimeltä saama IP-osoite on vakiona tyypiltään dynaaminen. Tämä kannattaa vaihtaa staattiseksi, koska tässä työssä Raspberryn on tarkoitus toimia web-serverinä. Dynaaminen IP-osoite voi vaihdella tulevaisuudessa, mikä voi vaikeuttaa etäyhteyden muodostamista tai yhteyden muodostamista web-serveriin internetistä käsin. (Penguin Tutor 2014.)

2.3.1 Staattinen IP-osoite

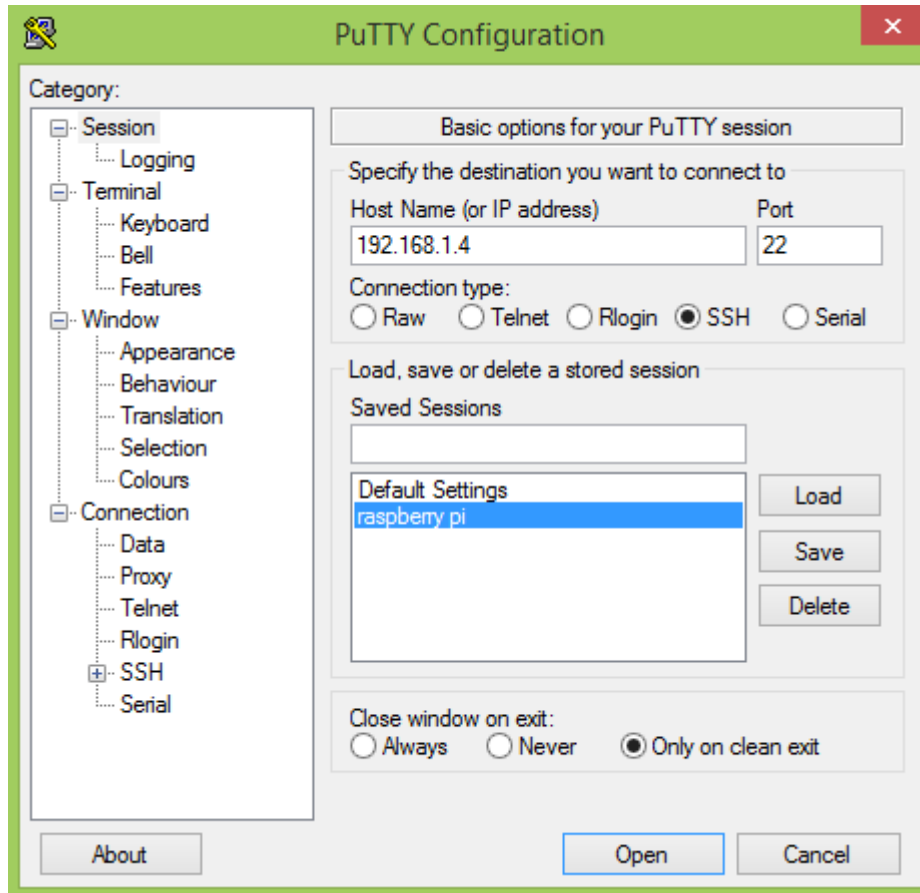
Laitteen verkkoasetuksia päästään tarkastelemaan komennolla "sudo nano /etc/network/interfaces". Tässä esimerkissä laitteelle annetaan staattinen IP-osoite: 192.168.1.4. Dynaaminen IP-osoite saadaan muutettua staattiseksi vaihtamalla asetus-tiedostosta rivi "iface eth0 inet dhcp" alla olevaksi:

```
iface eth0 inet static
address 192.168.1.4
netmask 255.255.255.0
gateway 192.168.1.1
```

2.3.2 SSH-etäyhteys

SSH (Secure Shell) on tietoliikenneprotokolla, joka mahdollistaa salatun tietoliikenne yhteyden luomisen kahden tietokoneen välille. Tässä tapauksessa etäyhteys luodaan samassa lähiverkossa olevan Raspberry Pi –tietokoneen ja Windows-tietokoneen välille. Windows-koneessa käytetään PuTTY –nimistä SSH-asiakasohjelmaa, joka mahdollistaa kirjautumisen ja komentojen antamisen komentorivillä Raspberry Pi:lle. Kun SSH-yhteys on käytössä, ei Raspberry Pi:ssä tarvitse olla kytkettynä näyttöä, näppäimistöä ja hiirtä. Riittää kun siihen on kytketty virtajohto ja verkkokaapeli. Kaikki tässä työssä tehtävät ohjelmien asennukset ja konfiguroinnit voidaan tehdä toiselta tietokoneelta komentori-

viltä käsin. Alla on kuva Putty-ohjelman käyttöliittymästä ja asetuksista joilla kirjaudutaan Raspberry Pi –tietokoneelle. (Penguin Tutor 2014.)



KUVA 5. Putty SSH-asiakasohjelman käyttöliittymä.

SSH-palvelimen pitäisi olla valmiiksi asennettuna Raspbian-käyttöjärjestelmässä ja SSH-yhteyden käyttö sallittuna vakiona. Onko SSH-yhteys käytössä, voidaan tarkastaa antamalla Raspberry Pi:n komentorivillä käsky: `"/etc/init.d/ssh status"`. Mikäli SSH on käytössä, komentoriville pitäisi ilmestyä vastaus `"[ok] sshd is running"`. Jos SSH-yhteys ei ole käytössä, se saadaan käyttöön "Raspi-config" asetussivulta, johon päästään komennolla `"raspi-config"`.

3 TIETOKANTAPALVELINYMPÄRISTÖ

Lämpötilan mittausjärjestelmä koostuu kolmesta eri elementistä, jotka ovat lämpötilan mittaus, tiedon tallentaminen tietokantaan ja mittausdatan visualisointi. Tietokantapalvelinympäristön tarkoituksena on mittausdatan tallennus sekä datan visualisointiin tarkoitettun web-sivun ylläpito. Järjestelmän palvelinosa toteutettiin LAMP-asennuksena. Tässä luvussa käydään läpi eri ohjelmisto komponenttien asentaminen ja toiminnan testaaminen.

3.1 LAMP -palvelin

Lyhennettä LAMP käytetään kuvaamaan erittäin yleistä web-palvelinteknologian kokonpanoa, jossa palvelinkoneen käyttöjärjestelmänä on Linux, palvelinohjelmistona Apache, tietokantaohjelmistona on MySQL ja dynaamisuuden ja vuorovaikutteisuuden mahdollistavana palvelinpuolen skriptikielenä on PHP. LAMP-tekniikoilla on mahdollista toteuttaa hyvin monipuolisia verkkopalveluja. Tässä työssä näitä tekniikoita käytetään osana lämpötilan seurantajärjestelmän toteuttamista. (Linux-wiki 2014.)

LAMP ei tänä päivänä enää edusta uusinta mahdollista teknologiaa. Apachen kanssa kilpailee monia uusia paljon nopeampia palvelinohjelmistoja (esim. NodeJS ja Nginx), suosittuja tietokantaohjelmistoja MySQL lisäksi ovat esimerkiksi MongoDB ja PostgreSQL ja ohjelmointikielipuolella on nykyisin suosittumpaa käyttää Javascriptiä, Ruby on Railisia tai Pythonia. LAMP on kuitenkin edelleen paljon käytössä ja edelleen toimiva tekniikka. Tähän työhön se valikoitui käytettäväksi pääasiassa aiemman kokemuksen perusteella. (Linux-wiki 2014.)

3.2 Apache-palvelinohjelmisto ja PHP

Raspberry Pi -tietokoneella suoritetaan palvelinohjelmistoa, jonka avulla ylläpidetään verkkosivuja, jotka esittävät tietokantaan tallennettua mittausdataa. Apache httpd on Apache-säätiön tuottama avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma. Apache2 ja PHP-ohjelmointikieli voidaan asentaa yhdellä komennolla komentorivin

kautta. Asennus onnistuu komennolla "sudo apt-get install apache2 php5 libapache2-mod-php5", mikä asentaa em. ohjelmat ja kaikki niiden riippuvuudet. Kun asennus on valmis, käynnistetään Apache uudestaan, jotta PHP aktivoituu. Tämä onnistuu komennolla "sudo /etc/init.d/apache2 restart". Asennuksen onnistuminen voidaan testata kirjoittamalla selaimen osoiteriville Raspberryn IP-osoite, jolloin selaimen pitäisi avautua palvelimen oletussivu, jos palvelinohjelmisto on käynnissä. (Install LAMP on Ubuntu 12.04.)

Tämän jälkeen testataan php-asennuksen onnistuminen. Luodaan Apachen oletus hakemistoon var/www/ tiedosto info.php komennolla "sudo nano /var/www/info.php" ja kirjoitetaan tiedoston sisällöksi: " <?php phpinfo(); ?> ". Tämän jälkeen selaimella samasta verkosta osoitteessa ip-osoite/info.php pitäisi näkyä PHP-infosivu. (Install LAMP on Ubuntu 12.04.)

3.3 Tietokanta

Tietokanta on paikka, jonne anturin prosessoima mittausdata tallennetaan. Kun tiedot on tallennettu tietokantaan, niitä voidaan tarkastella ja käsitellä myöhemmin. Tässä työssä käytetään MySQL-tietokantaa, joka asennetaan Raspberry Pi -tietokoneelle. Tietokanta konfiguroidaan niin, että mittausprosessi tallentaa tiedot suoraan sinne. Varsinaisen tietokantaohjelmiston lisäksi asennetaan myös phpMyAdmin-ohjelmisto, joka auttaa tietokannan käsittelyssä.

3.3.1 MySQL

MySQL on hyvin suosittu avoimen lähdekoodin tietokantaohjelmisto, joka on ominaisuuksiltaan erittäin monipuolinen ja suorituskykyinen. MySQL-tietokanta on hyvin suosittu web-sovellusten tietokantana. Tietokannan päälle rakennettava ohjelmalogiikka tehdään usein PHP-, Python- tai Perl-ohjelmointikielellä. (Linux-wiki 2014). Tässä työssä mittausdatan tallennus tietokantaan on totoutettu Python-kielellä ja tiedon hakeminen tietokannasta web-sivun käyttöön tapahtuu PHP-kielellä.

MySQL-tietokannan asennus onnistuu komennolla "sudo apt-get install mysql-server". Asennuksen aikana Raspberry Pi kehottaa asentamaan valikoiman muita paketteja, jotka ovat oleellisia tietokannan toiminnan kannalta. Asennuksen loppuvaiheessa käyttäjältä pyydetään root-salasanaa tietokannalle. Tähän on syytä keksiä jokin turvallinen salasana. Lisäksi on asennettava muutama muu ohjelmistokomponentti, jotta PHP- ja Python-ohjelmointikielet toimivat tietokannan kanssa. Tarvittavat ohjelmistot saadaan asennettua komennolla "sudo apt-get install mysql-client php5-mysql python-mysqldb". (Install LAMP on Ubuntu 12.04.)

3.3.2 phpMyAdmin

phpMyAdmin on PHP:llä toteutettu sovellus, joka helpottaa tietokannan käyttöä. SQL-kyselyiden suorituksen lisäksi phpMyAdmin sisältää moneen toimintoon valmiin selaimen kautta käytettävän käyttöliittymän. Sen asennus onnistuu komennolla "sudo apt-get install phpmyadmin". Asennuksen aikana käyttäjältä kysytään käytössä oleva palvelinohjelmisto, tietokantaohjelmisto ja tietokannan hallintaan tarvittava salasana. (Bringing MySQL to the web).

3.3.3 Tietokannan luonti

Luodaan tietokanta, joka sisältää lämpötilamittaukset ja mittausajankohdat. Ensiksi luodaan tietokanta nimeltä TempLog, joka sisältää taulun DS18B20. Taulu sisältää viisi kenttää, jotka ovat measurement_id, sensor_id, date, time ja value. Ensimmäinen kenttä on jokaiselle mittaukselle annettava yksilöllinen id-numero, joka sisältää valmiin AutoIncrement-ominaisuuden, jota käyttämällä id-numerosta ei tarvitse huolehtia itse, vaan jokainen tauluun lisättävä rivi saa suoraan seuraavan järjestyksessä olevan numeron. Seuraava kenttä kertoo anturin yksilöllisen id-numeron. Seuraavaksi ovat date- ja time-tyyppiset kentät, jotka huolehtivat mittaushetken päivämäärän ja tarkan kellonajan tallennuksesta. Viimeisenä on value-kenttä, johon tallennetaan lämpötila-arvo celsiusasteina (kuva 6).

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 measurement_id	int(11)			No	None	AUTO_INCREMENT	Change Drop More ▼
<input type="checkbox"/>	2 sensor_id	int(11)			No	None		Change Drop More ▼
<input type="checkbox"/>	3 date	date			No	None		Change Drop More ▼
<input type="checkbox"/>	4 time	time			No	None		Change Drop More ▼
<input type="checkbox"/>	5 value	varchar(50)	latin1_swedish_ci		Yes	NULL		Change Drop More ▼

↑ Check All / Uncheck All With selected: Browse Change Drop Primary Unique Index

KUVA 6. Tietokannan ja taulukon luonti phpMyAdminin avulla

Kun tietokanta ja taulukko on luotu, tarvitaan vielä sovellus-käyttäjä jolla on oikeudet lukea ja kirjoittaa dataa tietokantaan. Tämä onnistuu myös phpMyAdminin avulla. Anturidatan tallennusta ja tarkastelua varten voidaan luoda esimerkiksi käyttäjä nimeltä "sensor-writer" ja antaa tälle oikeudet "SELECT" ja "INSERT".

4 LÄMPÖTILAN MITTAAMINEN

Lämpötilan seurantajärjestelmän tärkein yksittäinen komponentti on anturi, joka reagoi ympäristön kanssa ja muuntaa mitattavan suureen helpommin käsiteltävään sähköiseen muotoon. Tässä luvussa kerrotaan käytetystä anturista, anturin kytkentä Raspberry Pi:n GPIO-porttiin sekä ohjelmakoodista, jolla anturin lukeminen ja mittausdatan tallennus tietokantaan tapahtuu.

4.1 Lämpötila-anturi

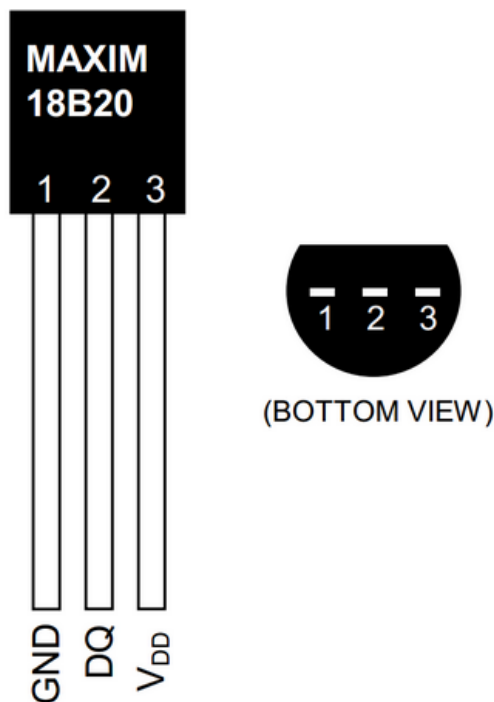
Lämpötilan mittaukseen käytetään DS18B20-anturia, jonka valmistaja on Dallas Semiconductor. DS18B20 on digitaalinen lämpötila-anturi, joka käyttää 1-wire-protokollaa kommunikointiin, joten Raspberry Pi:n GPIO-portista tarvitaan vain yksi I/O-pinni anturin lukemiseen. Anturin toimintaperiaate perustuu piioskiolaattoriin, jonka värähtelytaajuus riippuu voimakkaasti lämpötilasta. Anturin elektroniikka laskee värähtelyjen määrän tietyllä aikajaksolla ja lähettää vastaukset digitaalisessa muodossa. Jokaisella anturilla on yksilöllinen 48-bittinen sarjanumero. Sarjanumero mahdollistaa useiden DS18B20-antureiden kytkemisen sarjaan, jolloin lämpötila-anturiverkon hallintaan riittää yksi signaalijohdin. Tässä työssä käytetään kuitenkin vain yhtä anturia. Lisää anturin ominaisuuksia löytyy alla olevasta taulukosta. (Maxim Integrated DS18B20 datasheet.)

TAULUKKO 2. Lämpötila-anturin tekniset tiedot. (DS18B20 datasheet)

Anturin tyyppi	digitaalinen
Toiminta jännite	3 V ~ 5.5 V
Resoluutio	12 bit
Tarkkuus	$\pm 0.5^{\circ}\text{C}$ ($\pm 2^{\circ}\text{C}$)
Toiminta-alue	-55°C ~ 125°C
Väyläprotokolla	1-Wire

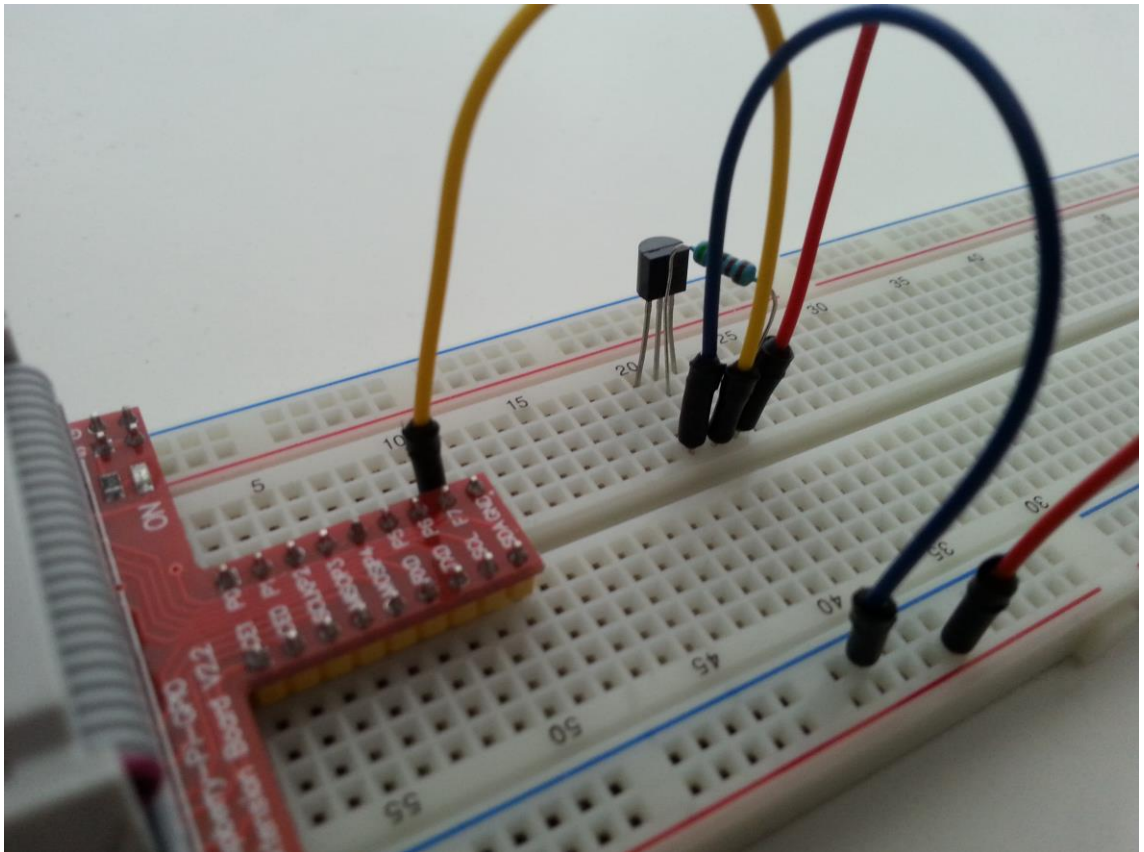
4.2 Anturin kytkentä

DS18B20-anturilla on kolme pinniä: GND, käyttöjännite (VDD) ja dataliitin (DQ) (kuva 7). Kytkentää tehdessä Raspberry Pi kannattaa olla sammutettuna. Kytkentää tehdessä on oltava huolellinen, sillä virheellinen kytkentä saattaa rikkoa anturin. Anturin GND-pinni kytketään GPIO-portin GND-liittimeen. Anturin käyttöjännite (VDD) voidaan kytkeä GPIO-portin 3.3 V tai 5 V lähtöön. Anturin dataliitin (DQ) kytketään GPIO4-pinniin. Anturi vaatii myös ylösmenovastuksen. Dataliittimen ja käyttöjännitteen välille täytyy kytkeä 4.7 k Ω vastus (kuva 8).



KUVA 7. DS18B20-anturin pinnijärjestys (Maxim Integrated DS18B20 datasheet)

Kytkenän tekemisessä on hyödynnetty kytkentäalustaa ja lattaakaapelia, joka yhdistää Raspberry Pi:n GPIO-portin pinnit kytkentäalustaan. Lisäksi kytkennästä löytyy muutama hyppylanka, 4.7 k Ω vastus ja tietenkin itse anturi (kuva 8).



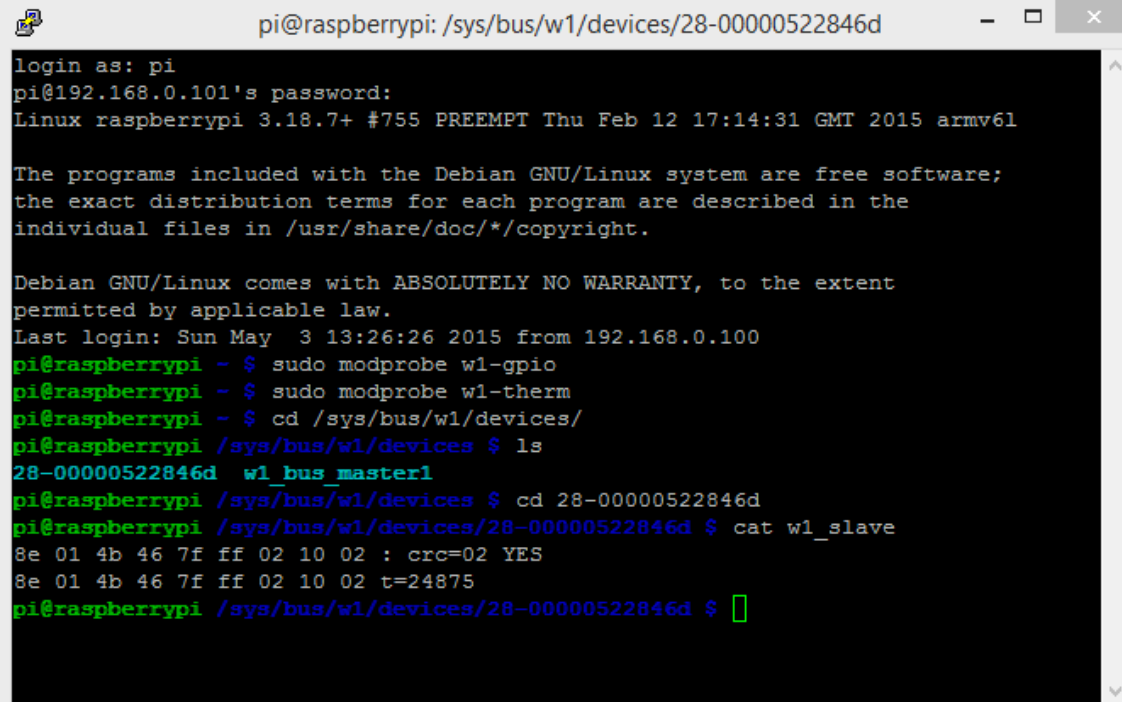
KUVA 8. Lämpötila-anturin kytkentä

4.3 Anturin lukeminen

Anturin lukeminen onnistuu komentorivin kautta muutamalla komennella ilman mitään ohjelmakoodin kirjoittamista. Tämä on mahdollista, koska Raspbian-käyttöjärjestelmässä on valmiina suuri määrä erilaisia ajureita ja myös anturin käyttämä 1-wire protokolla on tuettuna. Näitä kaikkia ajureita ei kuitenkaan ladata automaattisesti käyttöön järjestelmän käynnistyksen yhteydessä, koska se pidentäisi käynnistys aikaa merkittävästi ja veisi turhaan järjestelmän resursseja. Tarvittaessa ajurit pystytään ottamaan käyttöön modprobe-ohjelmalla, joka liittää ne ajon aikana Linux-ytimeen. (Adafruit 2014.)

Komennot, jotka tarvitaan anturin lukemiseen komentoriviltä, näkyvät kuvassa 9. Ensin "modprobe" komennolla ladataan tarvittavat ajurit käyttöön. Sitten siirrytään w1-laite kansioon, jossa pitäisi näkyä GPIO4-pinniin kytkettyjen anturien sarjanumerot. Nämä ovat kansioita, joiden sisällä olevaan w1_slave -tiedostoon anturit kirjoittavat mittausdatan. Tämän tiedoston pystyy lukemaan cat-komennolla, joka tulostaa tiedoston ruu-

dulle. Tämä komento palauttaa kaksi riviä, joista ensimmäisen perässä pitäisi lukea YES onnistuneen CRC-virheentarkistuksen vuoksi. Toisen rivin lopussa pitäisi näkyä mitattu lämpötila celsiusasteina tuhannesosan tarkkuudella. Tässä esimerkissä lämpötilaksi saatiin 24,875 °C (kuva 9).



```

pi@raspberrypi: /sys/bus/w1/devices/28-00000522846d
login as: pi
pi@192.168.0.101's password:
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:14:31 GMT 2015 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May 3 13:26:26 2015 from 192.168.0.100
pi@raspberrypi ~$ sudo modprobe w1-gpio
pi@raspberrypi ~$ sudo modprobe w1-therm
pi@raspberrypi ~$ cd /sys/bus/w1/devices/
pi@raspberrypi /sys/bus/w1/devices $ ls
28-00000522846d  w1_bus_master1
pi@raspberrypi /sys/bus/w1/devices $ cd 28-00000522846d
pi@raspberrypi /sys/bus/w1/devices/28-00000522846d $ cat w1_slave
8e 01 4b 46 7f ff 02 10 02 : crc=02 YES
8e 01 4b 46 7f ff 02 10 02 t=24875
pi@raspberrypi /sys/bus/w1/devices/28-00000522846d $

```

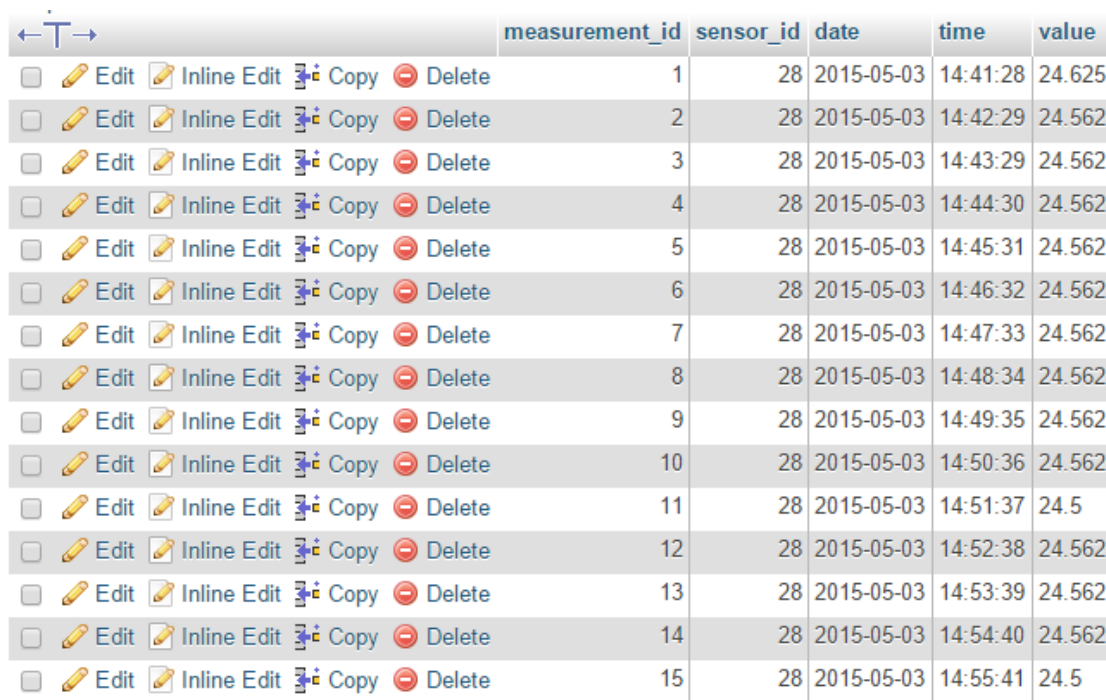
KUVA 9. Lämpötila-anturin testausta

Anturin lukeminen onnistuu komentoriviltä muutaman komennon avulla, mutta tämä ei ole kuitenkaan kovin kätevä tapa lämpötilan tarkkailuun. Tällä tavalla voidaan kuitenkin testata, että lämpötila-anturi toimii. Anturin lukemisesta voidaan tehdä helpompaa kirjoittamalla Python-ohjelma, joka lukee anturin arvon automaattisesti tietyn ajan välein ja tulostaa arvon komentoriville. Python-ohjelmaa voidaan vielä kehittää niin, että se osaa tallentaa mittaustiedot automaattisesti tietokantaan.

4.4 Mittausdatan tallennus

Python-ohjelma, joka lukee lämpötila-anturia ja tallentaa arvot tietokantaan, aloitetaan liittämällä ohjelman käyttämät moduulit import-komennolla, sekä lataamalla w1-ajurit käyttöön modprobe-komennolla. Sen jälkeen kirjoitetaan koodi, joka hakee anturin w1_slave-tiedoston ja lukee sen sisällön. Jos tiedosto sisältää lämpötila-arvon ja onnistuneen CRC-virheentarkistuksen, arvo tallennetaan muuttujaan. Lopuksi tarvitaan vielä

toiminto, joka muodostaa yhteyden tietokantaan ja tallentaa arvoja taulukkoon tietyn ajan välein. Tehty ohjelmakoodi löytyy kokonaisuudessaan työn lopusta liitteistä (liite 1). Koodin pohjana käytettiin Adafruitin sivuilta löytyvää tutoriaalia, jonka on kirjoittanut Simon Monk. Koodia muokattiin hieman ja siihen lisättiin toiminto, joka mahdollistaa mittausdatan tallentamisen tietokantaan. Python-koodin toiminta voidaan tarkistaa, tarkastelemalla Templog-tietokannan sisältöä phpMyAdmin-sovelluksen avulla (kuva 10).



	measurement_id	sensor_id	date	time	value
<input type="checkbox"/> Edit Inline Edit Copy Delete	1	28	2015-05-03	14:41:28	24.625
<input type="checkbox"/> Edit Inline Edit Copy Delete	2	28	2015-05-03	14:42:29	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	3	28	2015-05-03	14:43:29	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	4	28	2015-05-03	14:44:30	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	5	28	2015-05-03	14:45:31	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	6	28	2015-05-03	14:46:32	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	7	28	2015-05-03	14:47:33	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	8	28	2015-05-03	14:48:34	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	9	28	2015-05-03	14:49:35	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	10	28	2015-05-03	14:50:36	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	11	28	2015-05-03	14:51:37	24.5
<input type="checkbox"/> Edit Inline Edit Copy Delete	12	28	2015-05-03	14:52:38	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	13	28	2015-05-03	14:53:39	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	14	28	2015-05-03	14:54:40	24.562
<input type="checkbox"/> Edit Inline Edit Copy Delete	15	28	2015-05-03	14:55:41	24.5

KUVA 10. Mittaustulosten tarkastelu phpMyAdminin avulla

Mittausarvojen lukeminen ja tallentaminen automaattisesti minuutin välein voidaan toteuttaa lisäämällä Python-koodiin esimerkiksi silmukkarakenne ja silmukan loppuun sleep-funktio, joka odottaa silmukan sisällä olevan ohjelmakoodin suorituksen jälkeen aina esimerkiksi 60 sekuntia. Työn lopussa olevassa Python-ohjelmakoodissa (liite 1) on käytetty sleep-funktiota ajastukseen. Toinen vaihtoehto olisi käyttää Linux-käyttöjärjestelmässä olevaa Crontab-ohjelmaa, jonka avulla voidaan suorittaa erilaisia komentoja ajastetusti. Crontab:in muokkaaminen onnistuu komennolla: "crontab -e". Crontab-tiedoston rivi koostuu yleensä kahdesta kentästä, viisiosaisesta aikakentästä ja komennosta, joka on tarkoitus suorittaa ajastetusti. Crontab-ajastusta käytettäessä Python-koodissa ei tarvita sleep-funktiota.

Mittausdatan tallennukseen tarkoitettu python-ohjelma voitaisiin ajastaa kirjoittamalla crontab-tiedostoon alla olevat rivit.

```
# m h dom mon dow  command
*/1 * * * * /usr/bin/python /home/pi/ds18b20/temp_monitor1.py
```

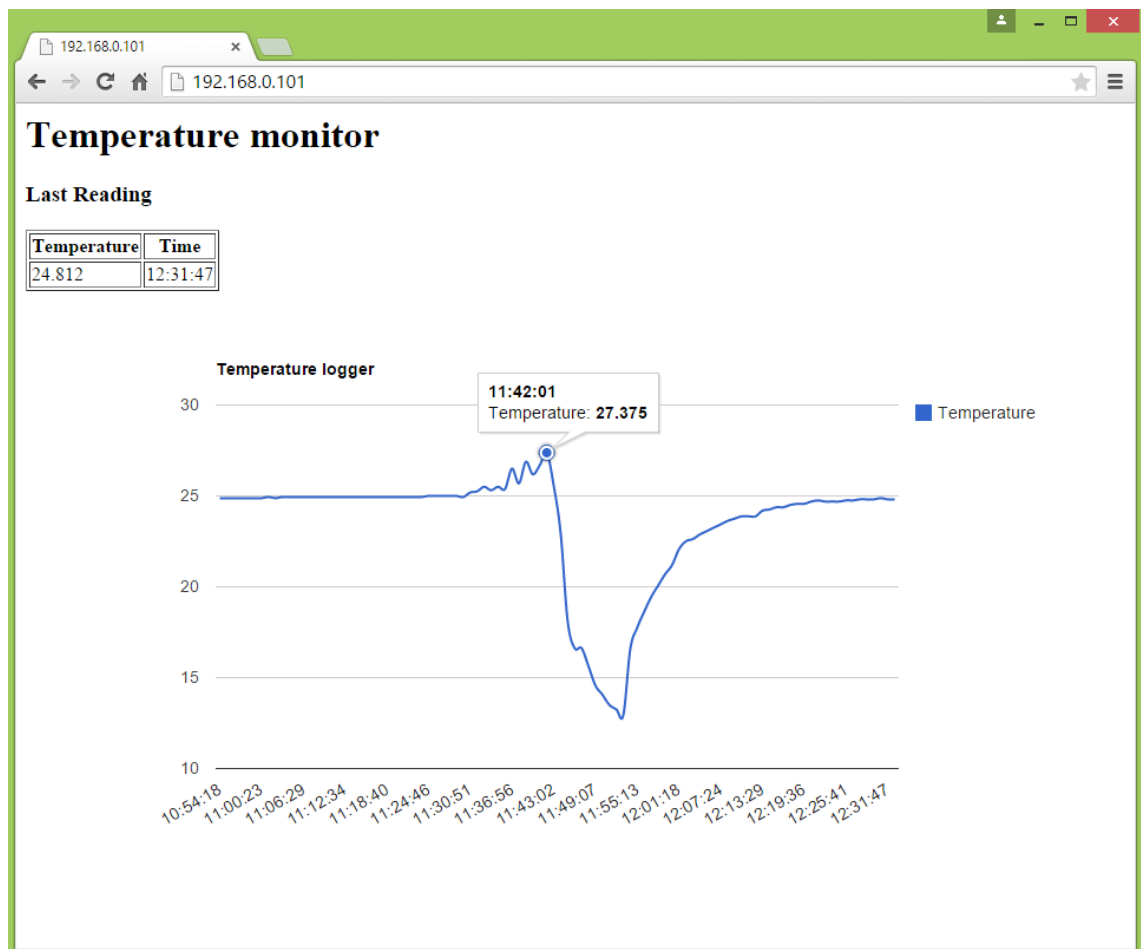
Yllä oleva esimerkki suorittaa komennon jatkuvasti minuutin välein. Komennolla "sudo service cron status" voidaan tarkistaa onko Crontab käytössä. Jos Crontab ei ole käytössä, se voidaan käynnistää komennolla "sudo service cron start". Crontab ajastusta käytettäessä on hyvänä puolena se, että se käynnistää lämpötilan monitorointiin käytettävän ohjelman automaattisesti, kun Raspberry Pi käynnistyy. (Linux-box 2006.)

5 MITTAUSDATAN VISUALISOINTI

Työn viimeisessä vaiheessa toteutettiin web-sivu, joka mahdollistaa mittusdatan tarkastelun selkeässä visuaalisessa muodossa. Sivu hakee automaattisesti viimeisimmät lämpötila-arvot tietokannasta, ja esittää ne käyttäjälle graafisessa muodossa. Sivu toteutettiin HTML-, PHP-, ja Javascript-kielillä.

5.1 WWW-sivu

Sivusto koostuu yhdestä PHP-tiedostosta, joka on sijoitettu Apache-palvelimen www-kansioon. PHP-tiedosto sisältää HTML-koodia, joka määrittää sivun rakenteen. Mittausdatan hakeminen MySQL-tietokannasta tapahtuu PHP:n avulla. Sivu on rakenteeltaan hyvin yksinkertainen: otsikko, taulukko ja lämpötilakuvaaja (kuva 11).



KUVA 11. Valmis web-sivu

Sivun alussa on taulukko, johon haetaan viimeisin tietokantaan tallennettu lämpötila-arvo ja tiedon mittausajankohta. SQL-selektoreita muokkaamalla voidaan taulukossa näyttää useampiakin arvoja. Taulukon jälkeen tulee kuvaaja, joka on toteutettu Google Charts -kirjastolla. Se on Javascript-kirjasto, joka mahdollistaa tyylikkään kuvaajan piirtämisen mittausdatan perusteella. Kuvaaja esittää lämpötila vaihtelun viimeisen vuorokauden ajalta.

5.2 PHP-koodi

PHP-tiedostolle annetaan nimi `index.php` ja se sijoitetaan Apache-palvelimen `www`-kansioon. PHP-tiedosto sisältää PHP-koodin lisäksi HTML-koodin, joka määrittää sivun rakenteen ja ulkoasun. HTML-koodin sisällä on myös Javascript-koodia, jota tarvitaan kuvaajan piirtämiseen. PHP:n tehtävä on hakea tietokannasta haluttu data ja tallentaa se muuttujaan, jota voidaan käyttää myöhemmin esimerkiksi taulukon tai kuvaajan muodostamiseen tämän datan perusteella.

PHP-koodi suoritetaan palvelimella, jossa PHP-tiedosto sijaitsee. Kun sivujen käyttäjän selain lähettää pyynnön sivusta, palvelin käyttää koodin PHP-tulkille. Se palauttaa valmiin sivun palvelimelle, joka lähettää sen edelleen selaimelle. Selaimelle lähetettävä sivu ei sisällä PHP-koodia, vaan PHP:n tuottaman HTML-koodin. Tämä on hyvä asia tässä tapauksessa, koska PHP-koodi sisältää tietokannan käyttöön tarvittavan käyttäjän nimen ja salasanan. Toteutetun web-sivun lähdekoodi löytyy kokonaisuudessaan liitteestä (liite 2).

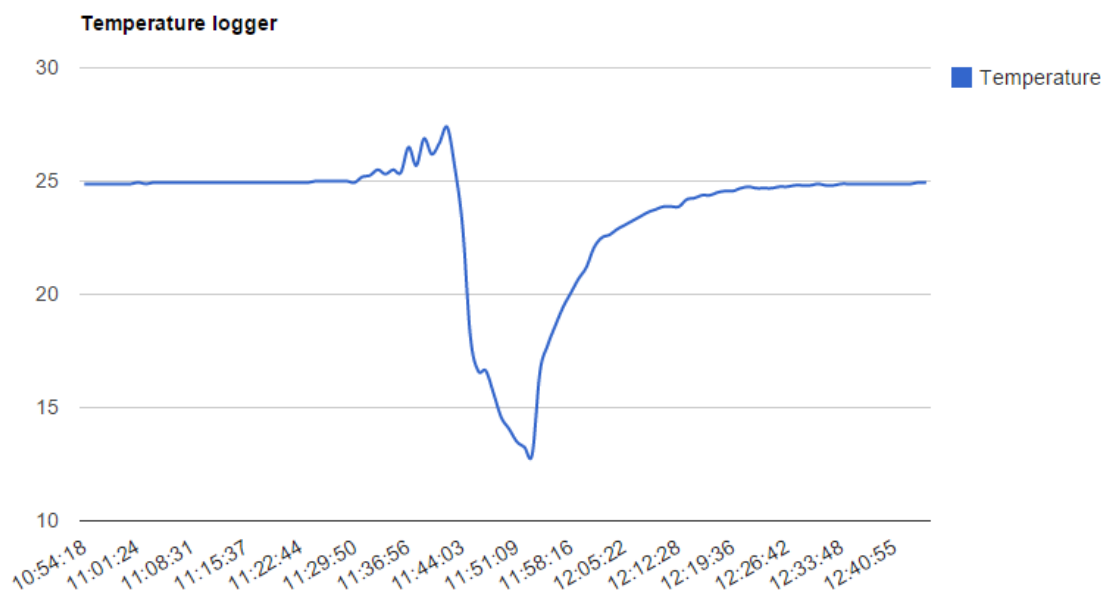
5.3 Google Chart API

Google Charts -kirjasto tarjoaa helpon ja tehokkaan tavan visualisoida dataa nettisivuilla. Sen avulla voidaan luoda eri tyyppisiä dynaamisia ja interaktiivisia kuvaajia. Kuvaajat renderöidään SVG-vektorikuva tekniikalla, joka tarjoaa hyvän yhteensopivuuden eri alustojen ja selaimien kanssa. Kirjasto sisältää paljon erilaisia ominaisuuksia, kuten datan yhdistämistä ja lajittelua SQL-tyylisillä komennoilla. Lisäksi kirjaston käyttö on täysin ilmaista. (Google Developers 2015.)

Google Charts -kirjaston käyttöä varten tarvitsee se ensin sisällyttää www-sivuun linkittämällä paikalliseen kopioon tai Googlen julkisen palvelimen kopioon. Sitten haetaan visualisoitava data PHP:llä ja tallennetaan se taulukkoon. Sen jälkeen määritellään tarvittavat asetukset kuten kuvaajan tyyppi sekä kuvaajan leveys ja korkeus. Sitten kuvaajasta luodaan objekti ja sille annetaan id ja lopuksi luodaan www-sivulle div-elementti, joka sisältää kuvaajan id:n. Kuvaajan toteutukseen tarvittava Javascript-koodi löytyy liitteestä (liite 2). (Google Developers 2015.)

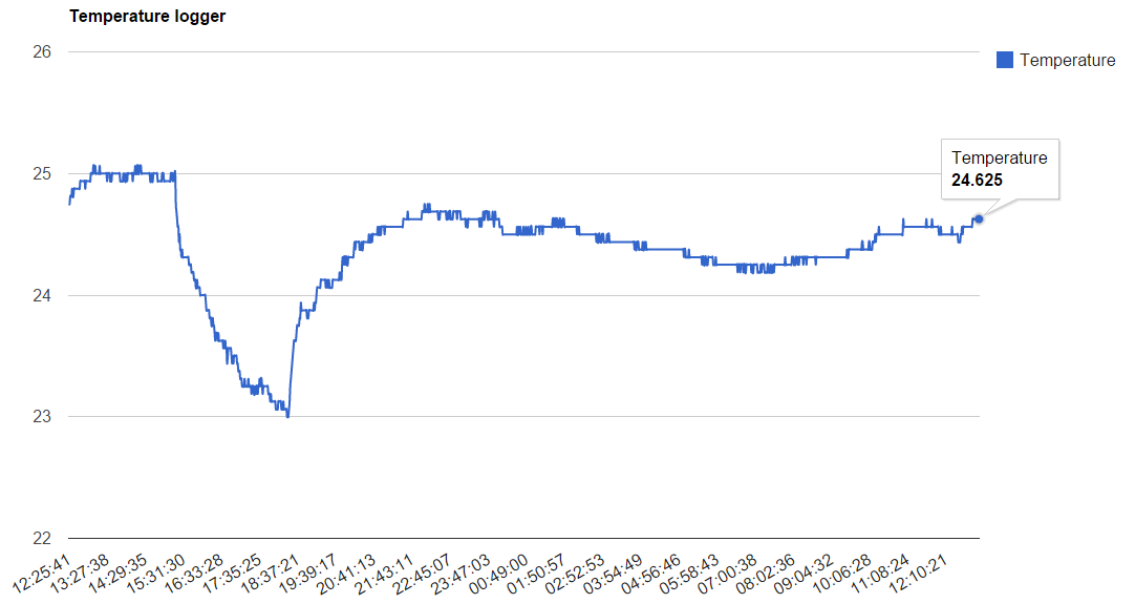
5.4 Testaus

Lämpötilan seurantajärjestelmän toimivuutta ja tarkkuutta testattiin manipuloimalla anturia eri tavoin. Tuloksena saatiin kuvaaja, jossa näkyy lämpötilan vaihtelu noin tunnin ajalta (kuva 12). Mittausympäristössä lämpötila oli noin 25 °C, mutta anturia manipuloitiin puhaltamalla siihen ensin lämmintä ilmaa, jolloin kuvaajasta nähdään lämpötilan nousevan kahden asteen verran. Sen jälkeen lämpötila-anturin vieressä pidettiin jääpala kymmenen minuutin ajan, jolloin kuvaajan mukaan lämpötila laski noin 14 astetta. Tämän jälkeen lämpötilalukeman annettiin taas tasaantua normaaliin huoneenlämpötilaan, johon kului kuvaajan mukaan noin 30 minuuttia.



KUVA 12. Lämpötila-kuvaaja tunnin ajalta

Seuraavaksi järjestelmää testattiin pidemmällä testijaksolla, jossa mittalaite oli yhtäjaksoisesti toiminnassa yli vuorokauden ajan. Tuloksena saatiin kuvaaja, joka näyttää tietokantaan tallennetut lämpötilalukemat viimeisen vuorokauden ajalta eli yhteensä noin 1440 lukemaa (kuva 13). Tässä testissä lämpötila-anturia ei manipuloitu millään tavalla, vaan mittalaitteen annettiin mitata huoneilman lämpötilaa.



KUVA 13. Lämpötila-kuvaaja vuorokauden ajalta

6 POHDINTA

Työn tavoitteena oli tutustua Raspberry Pi -tietokoneeseen, siinä käytettävään Linux-käyttöjärjestelmään, GPIO-portin käyttöön anturin lukemiseen, LAMP-tietokantapalvelinympäristöön ja toteuttaa toimiva lämpötilan seurantajärjestelmä. Lopputuloksena saatiin toteutettua toimiva järjestelmä, joka täytti opinnäytetyölle asetetut tavoitteet. Työ oli mielenkiintoinen ja sopivan haastava. Työn tekeminen edellytti monipuolisesti opittujen tietojen ja taitojen soveltamista käytännössä. Uutta opittavaa oli myös erityisesti anturitekniikan ja tietokannan käsittelyn kanssa.

Esineiden internetin yleistyessä tulee olemaan yhä enemmän järjestelmiä, jotka havainnoivat ympäristöään antureiden avulla ja ovat kytkettynä verkkoon. Kerättävää tietoa voidaan analysoida ja tulkita eri tavoilla. Tietoa voidaan myös hyödyntää esimerkiksi laitteiden etäseurantaan ja hallintaan. Tällaisille esineiden internetin ja teollisen internetin sovelluksille on varmasti kysyntää tulevaisuudessa eri teollisuuden aloilla. Tässä työssä toteutettu järjestelmä on pieni esimerkki järjestelmästä, joka kerää tietoa ympäristöstä, tallentaa tiedon ja esittää sen käyttäjälle.

Vaikka toteutettu järjestelmä todettiin toimivaksi, on siinä vielä kehitettävää varsinkin luotettavuuden ja tietoturvan osalta. Järjestelmä käyttää tallennustilanaan ainoastaan flash-muistiin perustuvaa MicroSD-korttia, jonka elinikä eli uudelleenkirjoituskertojen määrä on rajallinen. Tietoturva puoleen ei kiinnitetty erityistä huomiota, joten järjestelmä sopii paremmin lähiverkosta kuin internetin kautta käytettäväksi. Järjestelmää voisi parantaa vielä lisäämällä siihen uusia antureita esimerkiksi ilmankosteuden ja ilmanpaineen mittaamiseen. Web-sivulle voisi toteuttaa lisäominaisuutena esimerkiksi mahdollisuuden valita tietyn ajankohdan, jolta mittausdataa halutaan tarkastella.

LÄHTEET

Google developers. Google Charts -kirjaston kotisivut. Luettu 14.5.2015.
<https://developers.google.com/chart/>

Linux-käyttöjärjestelmää käsittelevä blogi. Luettu 13.5.2015.
<http://linux-box.blogspot.fi/2006/11/tehtvien-suorittaminen-ajastetusti.html>

Linux-ohjesivusto. LAMP-ympäristö. Luettu 2.5.2015.
<http://www.linux.fi/wiki/LAMP>

Linux-ohjesivusto. Raspberry Pi. Luettu 14.4.2015.
http://www.linux.fi/wiki/Raspberry_Pi

Maxim Integrated DS18B20 datasheet. Luettu 4.5.2015.
<http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

MySQL kotisivut. Luettu 3.5.2015.
<https://www.mysql.com/>

Penguin Tutor -sivuston tarjoama ohje LAMP-serverin asentamiseen. Luettu 2.5.2015.
<http://www.penguintutor.com/linux/raspberrypi-webserver>

phpMyAdmin contributors. (Bringing MySQL to the web). Luettu 3.5.2015.
<http://www.phpmyadmin.net/>

Raspberry Pi -säätiön kotisivut. Luettu 14.4.2015.
<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>

Raspbian Linux -jakelupaketin kotisivut. Luettu 15.4.2015.
<http://www.raspbian.org/>

Simon Monk. Ohje DS18B20-anturin liittämistä Raspberry Pi:hin Luettu 7.5.2015.
<http://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/>

Tutorial Arena. Install LAMP on Ubuntu 12.04. Luettu 3.5.2015.
<http://www.tutorialarena.com/blog/install-linux-apache-mysql-php-on-ubuntu-12.php>

LIITTEET

Liite 1. Python ohjelma

```

import os
import glob
import time
import MySQLdb as mdb

# Load the modules
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

# Read w1_slave file
def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c

#get readings from sensors every minute and store them to MySQL
while True:

    try:
        pi_temp = read_temp()
        con = mdb.connect('localhost', 'sensor_writer', 'password',
'TempLog');
        cur = con.cursor()
        cur.execute("""INSERT INTO DS18B20(date, time, value) \
VALUES(%s,%s,%s)""", time.strftime("%Y-%m-%d"),
time.strftime("%H:%M:%S"), (pi_temp))
        con.commit()

    except mdb.Error, e:
        con.rollback()
        print "Error %d: %s" % (e.args[0],e.args[1])
        sys.exit(1)

    finally:
        if con:
            con.close()

# print(read_temp())
time.sleep(60)

```

Liite 2. WWW-sivun lähdekoodi

1 (2)

```

<?php
$con=mysqli_connect("localhost","sensor_writer","password","TempLog");
// check connection
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$result = mysqli_query($con,"SELECT * FROM DS18B20 ORDER BY measurement_id DESC LIMIT 1");

echo "<table border='1'>".
"<h1>Temperature monitor</h1>".
"<p></p>".
"<h3>Last Reading</h3>".
"<tr>".
"<th>Temperature</th>".
"<th>Time</th>".
"</tr>";

while($row = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['value'] . "</td>";
    echo "<td>" . $row['time'] . "</td>";
    echo "</tr>";
}
echo "</table>";

mysqli_close($con);

echo "
<html>
<head>
<script type=\"text/javascript\"
src=\"https://www.google.com/jsapi\"></script>
<script type=\"text/javascript\">
google.load(\"visualization\", \"1\", {packages:[\"corechart\"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
var data = google.visualization.arrayToDataTable([
['Time', 'Temperature'], ";

$con=mysqli_connect("localhost","sensor_writer","password","TempLog");
// check connection
if (mysqli_connect_errno())
echo "Failed to connect to MySQL: " . mysqli_connect_error();
$result = mysqli_query($con,"SELECT * FROM DS18B20 ORDER BY measurement_id ASC LIMIT 1440");
while($row = mysqli_fetch_array($result))
{
    echo "[" . $row['time'] . "," . $row['value'] . "],\n";
    $value = $row['value'];
}

echo "
[', $value ]
)];

```

2 (2)

```
var options = {
  title: 'Temperature logger', curveType: 'function'
};
var chart = new
google.visualization.LineChart(document.getElementById('chart_div'));
chart.draw(data, options);
}
</script>
</head>
<body>
<div id="chart_div" style="width: 900px; height: 500px;"></div>
</body>
</html>
";
```