

Vesikiertoisen lattialämmityksen elektroninen säädin

Harri Valkonen

Opinnäytetyö

Joulukuu 2015

Tekniikan ja liikenteen ala

Insinööri (AMK), Tietotekniikan (Tietoverkkotekniikan) koulutusohjelma

Tekijä(t) Valkonen, Harri	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 06.12.2015
	Sivumäärä 59	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Vesikiertoisen lattialämmityksen elektroninen säädin		
Tutkinto-ohjelma Insinööri (AMK), Tietotekniikan (Tietoverkkotekniikan) koulutusohjelma		
Työn ohjaaja(t) Mika Rantonen, Olli Väänänen		
Toimeksiantaja(t) SolidNet Oy		
Tiivistelmä <p>Opinnäytetyön toimeksiantajana toimi SolidNet Oy. Toimeksiantajalla oli tarve toteuttaa vesikiertoisen lattialämmityksen digitaalinen ohjausjärjestelmä. Opinnäytetyön tavoitteena oli toteuttaa prototyyppi ohjausjärjestelmästä, pilvipalvelusta ja mobiilisovellus ohjausjärjestelmän hallintaa varten.</p> <p>Aluksi työssä tutkittiin servomoottorin mahdollista kytkentätapaa. Toimeksiantaja oli määritellyt käytettäväksi 1-Wire lämpötila-antureita, joten 1-Wire väylän sopimista myös servomoottorin väyläksi tutkittiin. Työssä perehdyttiin myös pilvipalvelutekniikkaan, tietokantarajapintoihin ja mobiilisovelluksen toteutustekniikkaan.</p> <p>Vaatusmäärittely työlle oli toteuttaa prototyyppi mittaus- ja säätöyksiköstä, pilvipalvelu ja mobiilisovellus mittaus- ja säätöyksikön hallintaa varten. Toimeksiantaja oli määritellyt käytettäväksi RaspBerry Pi pienoistietokonetta mittausyksikön alustana ja digitaalisia 1-Wire lämpötila-antureita.</p> <p>Opinnäytetyön lopputulos oli toimiva prototyyppi mittaus- ja säätöyksiköstä, joka on asennettavissa kokeiluympäristöön testikäyttöön. Lisäksi saatiin toteutettua myös pilvipalvelu datan tallentamiseen ja mobiilisovellus mittaus- ja säätöyksikön hallintaa varten.</p> <p>Mobiilisovellus ei ole vielä ladattavissa Google Play kaupasta, mutta on käännettävissä suoraan uusimpiin Android-puhelimiin.</p>		
Avainsanat (asiasanat) RaspBerry Pi, 1-Wire, LAMP, MySQL, PHP, pilvipalvelu, Cordova, tietokanta		
Muut tiedot		

Author(s) Valkonen, Harri	Type of publication Bachelor's thesis	Date 06.12.2015
		Language of publication: Finnish
	Number of pages 59	Permission for web publication: x
Title of publication Electronic thermostat for hydronic underfloor heating		
Degree programme Degree Programme in Information Technology (Data Network Technology)		
Supervisor(s) Rantonen Mika, Väänänen Olli		
Assigned by SolidNet Oy		
Abstract <p>The thesis was assigned by SolidNet Oy, a company which had a need for an electronic thermostat for hydronic underfloor heating. The aim of this study was to implement a prototype of the control system as well as a cloud service and management software application for a mobile environment.</p> <p>Initially, the work examined a possible connection method for a servo motor. The Company was defined to use one-Wire temperature sensors, therefore the suitability of one-Wire bus gateway for the servo motor was also examined. The study also discusses cloud service technology, database interfaces and implementation technology of a mobile application.</p> <p>The specifications for the work required to implement the prototype of the measurement and control unit as well as a mobile application for system management. The administration takes place via the cloud service the creation of which was part of the work.</p> <p>The client had specified to use Raspberry Pi minicomputer as a platform for unit of measurement platform, and digital one-Wire temperature sensors. The technical implementation was able to meet the requirements set to it.</p> <p>The outcome of the thesis is a functional prototype that can be installed in a pilot environment for testing purposes.</p> <p>The mobile application is not yet downloadable from Google Play store, but it can be compiled directly to the latest Android phones.</p>		
Keywords/tags (subjects) RaspBerry Pi, 1-Wire, LAMP, MySQL, PHP, cloud service, Cordova, database		
Miscellaneous		

Sisältö

Lyhenteet.....	6
1 Työn lähtökohdat	8
2 Ympäristö	8
2.1 Raspberry Pi.....	8
2.1.1 Yleistä.....	8
2.1.2 GPIO-väylä	9
2.2 1-Wire	10
2.2.1 Yleistä.....	10
2.2.2 Verkon rakenne	11
2.2.3 1-Wire-väylä.....	12
2.2.4 Adapteri	15
2.2.5 Master hub	16
2.2.6 Hub.....	16
2.2.7 Lisävirransyöttö	16
2.2.8 Liittimien kytkennät käytettäessä RJ12/J45 -liittimiä.....	17
2.3 XML.....	18
2.3.1 XML-tiedoston rakenne	18
2.3.2 XML-tiedoston puurakenne ja elementit	19
2.3.3 XML-elementin attribuutin käyttö.....	20
2.4 JSON.....	20
2.4.1 JSON-tiedoston rakenne	20
2.4.2 JSON:n käyttötavat	21
2.5 LAMP.....	22
2.5.1 Asennus.....	22
2.5.2 Linux.....	23
2.5.3 Apache	24

	2
2.5.4 MySQL.....	25
2.5.5 PHP.....	30
2.6 Pilvipalvelu.....	32
2.6.1 Pilvipalvelutyypit.....	32
2.6.2 Tiedon tallentaminen pilvipalveluun.....	32
3 Toteutus	33
3.1 Toteutetun järjestelmän kuvaus	33
3.2 Mittaus- ja säätöyksikkö.....	34
3.2.1 Lämpötila-antureiden kytkeminen.....	35
3.2.2 Servomoottorin kytkeminen	35
3.2.3 Merkkivalot.....	37
3.2.4 Piirilevy	38
3.2.5 GPIO-väylä	39
3.2.6 Mittaus- ja säätöyksikön ohjelmisto.....	40
3.2.7 Mittaus- ja säätöyksikön käyttöönotto	40
3.3 Pilvipalvelu.....	41
3.3.1 Tietokanta, tietokannan käyttö ja hallinta	41
3.3.2 Pilvipalvelun JSON -rajapinnat.....	42
3.3.3 Sääennusteet	42
3.3.4 Venttiilin ohjauslämpötilan laskeminen.....	43
3.4 Mobiilisovellus.....	44
3.4.1 Cordova.....	45
3.4.2 Sovelluksen toiminta	45
4 Verkko liikenteen tietoturva	46
4.1 Anti-DDOS.....	46
4.2 JSON-rajapinnat.....	46
5 Jatkokehitys.....	47

5.1	Pilvipalvelun kehitykset.....	47
5.1.1	Asiakastilin luominen.....	47
5.1.2	Mittaus ja säätöyksikön rekisteröiminen palveluun	48
5.2	Mittaus ja säätöyksikön kehitys	48
5.2.1	1-Wire-verkon kehittäminen ja langattomuus.....	49
5.2.2	Toiminnallisuuden laajentaminen.....	49
5.2.3	Vikasietoisuuden parantaminen.....	50
5.3	Mobiilisovelluksen kehitysideat	50
5.3.1	Asiakkuuden luominen	50
5.3.2	Mittaus- ja säätöyksikön rekisteröiminen ja hallinta	50
5.3.3	Sääennusteen näyttäminen.....	51
5.3.4	Teknisiä muutoksia	51
5.3.5	Uusien ominaisuuksien tuomat muutokset	51
6	Lopputulos.....	52
7	Yhteenveto	53
	Lähteet.....	54
	Liitteet	56
	Liite 1. Mittaus- ja säätöyksikön Python-ohjelmistokoodi.....	56

Kuviot

Kuvio 1. GPIO-väylä	10
Kuvio 2. 1-Wire-laitteiden ketjutettu verkko	11
Kuvio 3. 1-Wire-laitteiden tähtiverkko.....	11
Kuvio 4. Hubin käyttö 1-Wire-verkossa.....	12
Kuvio 5. Aikaväli 1-Wire-väylässä	13
Kuvio 6. Nollauspulssi ja slave-laitteiden ilmoittama (presence) ”läsnäolo” -pulssi ...	14
Kuvio 7. Kirjoittaminen ja lukeminen väylästä.....	14
Kuvio 8. Adapterin kytkentäkaavio	16
Kuvio 9. RJ12-liittimen ja RJ45-liittimen 1-Wire-kytkentä	17
Kuvio 10. XML:n rakenne	18
Kuvio 11. XML:n laajennettu rakenne.....	19
Kuvio 12. XML:n puurakenne	19
Kuvio 13. JSON:n rakenne	21
Kuvio 14. JSON JavaScript muuttujassa ja sen käsittely.....	21
Kuvio 15. Linux-käyttöjärjestelmän kuvaus	24
Kuvio 16. HTTP palvelinsovellusten käyttö kuvaaja	25
Kuvio 17. MySQL Fabric.....	26
Kuvio 18. MySQL Cluster	27
Kuvio 19. Taulun luominen tietokantaan.....	28
Kuvio 20. PHPMyAdmin-käyttöliittymä.....	29
Kuvio 21. MySQL Workbench-käyttöliittymä.....	30
Kuvio 22. Yksinkertainen PHP-scripti	31
Kuvio 23. Tiedon haku MySQL-tieokannasta PHP:llä	31
Kuvio 24. Toteutetun järjestelmän kuvaus	34
Kuvio 25. Lämpötila-antureiden kytkentä GPIO-väylään.....	35
Kuvio 26. Pulssin pituus modulaatiossa	36
Kuvio 27. PWM käyttöön Python-ohjelmointikielellä.....	36
Kuvio 28. Servomoottorin kytkeminen GPIO-väylään	37
Kuvio 29. Piirilevy	38
Kuvio 30. Piirilevy kalustettuna ja liitettynä Raspberryyyn	39

Kuvio 31. Esimerkki pilvipalvelun sensor_data-taulusta.....	42
Kuvio 32. Mittaus- ja säätöyksikön ohjaus JSON.....	42
Kuvio 33. Esimerkki pilvipalvelun forecast-taulusta	43
Kuvio 34. Venttiilin laskennallisen lämpötilan laskentakaava	44
Kuvio 35. Mobiilisovellus.....	44
Kuvio 36. Suunniteltu laitekotelo (30,5 mm x 103 mm x 160 mm)	49

Taulukot

Taulukko 1. Listausta ROM-tason käskyistä	15
Taulukko 2. Merkkivalojen toiminta eri laitteen tiloissa.....	37
Taulukko 3. Projektissa käytetyt GPIO-väylän pinnit ja niiden käyttötarkoitus.....	39

Lyhenteet

AJAX (Asynchronous JavaScript And XML) joukko web-sovelluskehityksen tekniikoita, joiden avulla web-sivun sisältöä voidaan ladata ilman koko sivun lataamista.

CSS (Cascading Style Sheets) web-sivun tyylitiedosto.

CGI (Common Gateway Interface) tekniikka, jonka avulla selain voi välittää dataa palvelimella suoritettavalle ohjelmalle.

DDOS (Distributed Denial of Service) hajautettu palvelunestohyökkäys.

DHCP (Dynamic Host Configuration Protocol) verkkoprotokolla, jonka tehtävä on jakaa ip-osoite verkkoon liittyneille laitteille.

DOM (Document Object Model) tapa kuvata HTML:n, XML:n tai XHTML:n rakenne puuna.

GPIO (General Purpose I/O) yleiskäyttöinen portti mikro-ohjaimissa ja mikroprosessoreissa.

GPL (General Public License) vapaiden ohjelmistojen julkaisemiseen tarkoitettu lisenssi.

HDMI (High Definition Multimedia Interface) kuvan ja monikanavaäänien siirtämiseen suunniteltu digitaalinen liitäntästandardi.

HTML (Hypertext Markup Language) kuvauskieli, jota käytetään web-sivustoissa.

HTTP(Hypertext Transfer Protocol) protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon.

IaaS (Infrasructure as a Services) pilvipalvelu käyttöjärjestelmille, kuten virtuaalikoneille.

IP (Internet Protocol) TCP/IP-verkon Internet-kerroksen protokolla, joka huolehtii IP-tietoliikennepakettien toimittamisesta perille pakettikytkentäisessä verkossa.

JSON (JavaScript Object Notation) kevyt tiedonvälitysmuoto.

LAMP (Linux, Apache, MySQL, PHP) kokoelma avoimen lähdekoodin ohjelmia.

NAT (Network Address Translation) TCP/IP-verkon osoitteenmuunnostekniikka, joka mahdollistaa yhden julkisen osoitteen käytön useamman sisäverkon osoiteella.

OS X Applen kehittämä ja markkinoima käyttöjärjestelmäperhe Macintosh-tietokoneisiin.

PaaS (Platform as a Service) ympäristö, jossa voi suorittaa pilvialustalle suunniteltuja ohjelmistoja.

PWM (Pulse-Width Modulation) pulssinleveysmodulaatio. Kuormaan menevää jännitettä säädetään muuttamalla pulssin suhdetta.

ROM (Read Only Memory) pysyväismuisti, johon ei tehdä muutoksia normaalikäytön aikana.

S3 (Simple Storage Service) Amazonin tiedostopalvelinjärjestelmä.

SaaS (Software as a Service) ympäristössä kaikki palvelun osat ovat pilvessä ja palvelun tarjoaja huolehtii palvelusta.

SDHC (Secure Digital High Capacity) muistikorttityyppi yli 2 GB:n tallennuskapasiteetilla.

SSL (Secure Sockets Layer) salausprotokolla, jolla voidaan suojata Internet-sovellusten tiedonsiirto.

USB (Universal Serial Bus) sarjamuotoinen väylä oheislaitteiden liittämiseen.

VAC (Valve Anti-Cheat System) useasta laitteesta koostuva palvelunsuojausjärjestelmä.

Windows 10 IoT (The Internet of Things) Microsoft Windows 10 käyttöjärjestelmän pelkistetty teollisuusversio esineiden Internetin toteuttamiseen.

WWW (World Wide Web) Internet-verkossa toimiva hypertekstijärjestelmä.

XML (Extensible Markup Language) tiedonsiirtoon suunniteltu kuvauskieli.

1 Työn lähtökohdat

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa prototyyppi vesikiertoisen lattialämmityksen digitaalisesta ohjausjärjestelmästä. Järjestelmän tarkoituksena on ohjata säätöventtiiliä halutun sisälämpötilan saavuttamiseksi. Säättämisessä otetaan huomioon sisälämpötila, ulkolämpötila ja ennuste tulevasta lämpötilasta.

Tavoitteena oli suunnitella laite, joka pystyy keräämään tiedon lämpötila-antureista ja samalla ohjaamaan venttiilin servomoottoria. Laitteen käyttöönotto tuli myös olla yksinkertainen. Lisäksi tuli suunnitella pilvipalvelu, jonne lämpötilatiedot tallentuvat ja josta laitteen ohjaustiedot tulevat. Laitteen säätäminen tuli tapahtua mobiilisovelluksen kautta.

Laitteen alustaksi oli määritelty käytettäväksi Raspberry Pi yhden piirilevyn tietokone ja DS18B20 1-Wire -lämpötila-antureita. Mobiilisovelluksen tuli toimia Android-puhelimissa.

2 Ympäristö

2.1 Raspberry Pi

2.1.1 Yleistä

Raspberry Pi on halpa noin luottokortin kokoinen pienoistietokone. Se voidaan kytkeä monitoriin tai televisioon HDMI-(High Definition Multimedia Interface) väylän kautta. Näppäimistönä ja hiirenä voidaan käyttää normaalia USB-(Universal Serial Bus) portin kautta kytkettävää näppäimistöä ja hiirtä. Käyttöjärjestelmäksi asennetaan yleensä Linux-pohjainen käyttöjärjestelmä, mutta keväällä 2015 markkinoille tullut uusi Raspberry Pi 2 tukee myös Windows 10 IoT (The Internet of Things) -järjestelmää.

Raspberry Pin on kehittänyt Raspberry Pi Foundation. Ajatuksena oli luoda pieni ja halpa tietokone opetuskäyttöön. Laitteisto perustuu Broadcomin BCM2835 -järjestelmäpiiriin (ARM1176JZF-S). Se sisältää suorittimen, muistin ja integroidun grafiikkapiiriin. Helmikuussa 2015 julkaistu Raspberry Pi 2 sai uuden neliytimisen

BCM2836 ARM Cortex-A7 -suorittimen. Samalla kellotaajuus kasvoi 700 MHz:stä 900 MHz:iin ja muistin määrä kaksinkertaistui 1 GB:iin. Uusi suoritin on suorituskyvyltään noin kuusinkertainen entiseen nähden, vaikka virran kulutus pysyykin pienenä (4 W). (What is a raspberry pi ? N.d.)

Raspberry Pi on opetuskäytössä yleinen ja sen myötä alan harrastajatkin ovat löytäneet laitteen. Pienen mediapalvelimen tai pelilaitteen alustana on Raspberry. Onpa se löytänyt monen robotin ohjaukseenkin. Myös automaatioissa on alettu käyttää mm. säätötekniikan ohjauksessa.

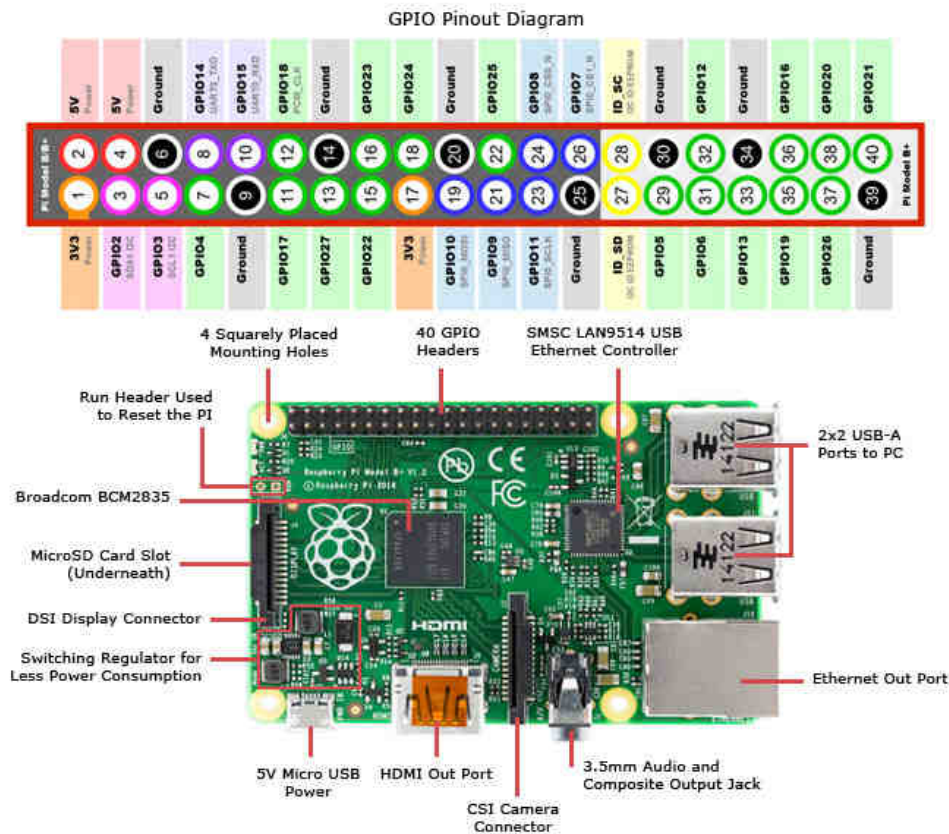
Raspberryn käyttöjärjestelmä on ladattavissa verkosta ja helposti asennettavissa. Asennusta helpottamaan on Raspberry Pi Foundation tuonut NOOBS -asennus ohjelman. Ohjelma kopioidaan micro SDHC (Secure Digital High Capacity) -muistikortille ja ensimmäisen käynnistymisen yhteydessä voidaan valita valikosta, mikä jakelupaketti asennetaan. Itse asennus on automaattinen.

Raspberrystä voidaan käyttää joko tekstipohjaisena tai graafisen käyttöliittymän kautta Linuxin tavoin. Raspberryn löytyy myös valmiita mediapalvelin paketteja, mutta niiden käyttöönotto vaatii hieman omaa aktiivisuutta. (Linnake, 2013.)

Halvan hinnan ja helppokäyttöisyyden myötä Raspberry Pi on myydyin Britanniassa valmistettu tietokone. Raspberrystä oli myyty 17. helmikuuta 2015 mennessä yli viisi miljoonaa kappaletta. (Raspberry Pi becomes best selling British computer, N.d.)

2.1.2 GPIO-väylä

Raspberry Pi tarjoaa kehittäjille tehokkaan GPIO (general purpose input/output) väylän. Väylä tarjoaa ohjelmoitavien pinnien lisäksi 5 voltin ja 3.3 voltin ulostulot vähävirtaisille oheislaitteille. Enemmän virtaa tarvitseville laitteille kannattaa tuoda ulkoinen käyttöjännite. Raspberry on herkkä jännitteen vaihteluille, ja lisälaitteen virran käyttö voi aiheuttaa Raspberryn uudelleen käynnistymisen. Kuviossa 1 Raspberry Pi 2 -laitteen GPIO-väylän sijainti ja pinnien selitykset.



Kuvio 1. GPIO-väylä (GPIO Pinout Diagram, N.d.)

Ohjelmoitavat pinnit voidaan ohjelmoida ulostuloksi (output) tai sisäänmenoksi (input). Raspberryn ulostulon tila voi olla joko 0 voltia (LOW) tai 3,3 voltia (HIGH). Ulostuloon voidaan ohjelmallisesti tehdä myös pulssinleveysmodulaatio (PWM, Pulse-Width Modulation), jolloin saadaan mm. led-polttimon valotehoa säädettyä. Pinnin toimiessa sisäänmenona täytyy usein käyttää vastusta selkeyttämään pinnin tilaa. Sisäänmenotilassa voidaan lukea ainoastaan kaksi tilaa: 0 voltia tai 3,3 voltia. Jos jännite ei ole selkeästi esim. 3,3 voltia, pinnin tilaa ei välttämättä kyetä lukemaan oikein. Tällöin tarvitsee kytkeä jännitettä nostattava vastus (pull up resistor). (GPIO: Raspberry Pi Models A and B, N.d.)

2.2 1-Wire

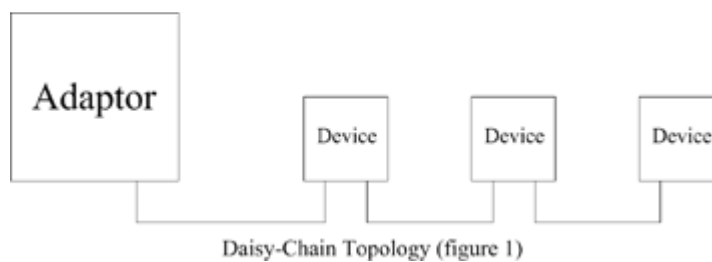
2.2.1 Yleistä

1-Wire on yksinkertainen tapa liittää laitteita ja antureiden tietokoneeseen tai muihin ohjaimiin. Laitteet muodostavat verkon, jonka avulla saadaan siirrettyä helposti

mm. lämpötila, kosteus, sademäärä, ilmanpaine ja paljon muuta tietoa. Verkko koostuu pääohjaimesta (master), esim. tietokone tai mikrokontrolleri, ja yhdestä tai useammasta orjalaitteesta (slave). Tietokoneeseen verkon saa kytkettyä sarjaliikenne- tai USB-portin kautta. Master-laite kommunikoi slave-laitteiden kanssa Dallas Semiconductorin kehittämällä 1-Wire-protokollalla. Laitteet kytketään toisiinsa kahdella johtimella, joista toinen on maa ja toista käytetään datan siirtoon. Samaan johdinpariin voidaan kytkeä useampi slave-laite ja ne voidaan tunnistaa valmistusvaiheessa annetun yksilöllisen osoitteen avulla. (1-Wire Tutorial, N.d.)

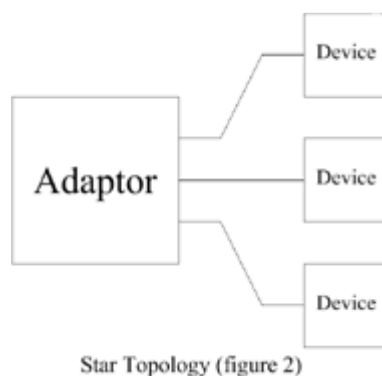
2.2.2 Verkon rakenne

Verkko jakaantuu pääasiassa kahdenlaiseen verkkorakenteeseen: ketjutettuun verkkoon ja tähtiverkkoon. Ketjutetussa verkossa master-laitteesta kytketään slave-laitteeseen ja slave-laitteesta seuraavaan slave-laitteeseen (ks. kuvio2).



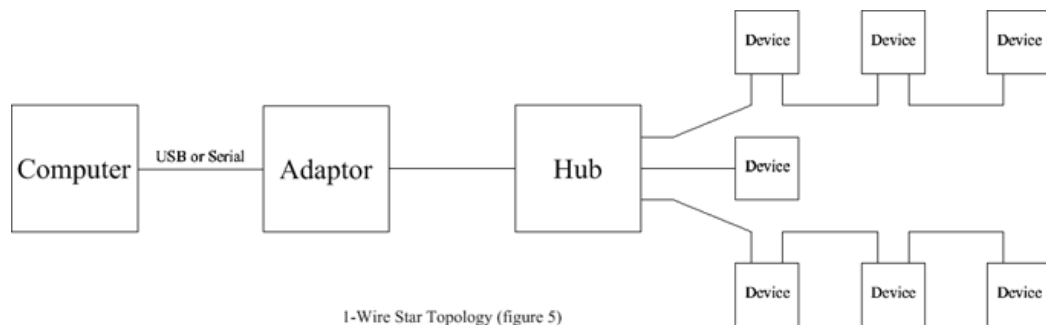
Kuvio 2. 1-Wire-laitteiden ketjutettu verkko (Network Topology, N.d.)

Tähtiverkossa jokainen slave-laite kytketään suoraan master-laitteeseen (ks. kuvio3).



Kuvio 3. 1-Wire-laitteiden tähtiverkko (Network Topology, N.d.)

Verkon rakenteeseen voivat myös vaikuttaa ulkoiset tekijät, kuten esimerkiksi laitteiden fyysinen sijainti, jolloin laitteiden kytkeminen sarjaan voi olla mahdotonta. Verkko- rakenteessa voidaan käyttää tähtiverkon ja sarjaan kytketyn verkon yhdistelmää. Tällöin verkot kytketään hubin kautta master-laitteeseen (ks. kuvio4).



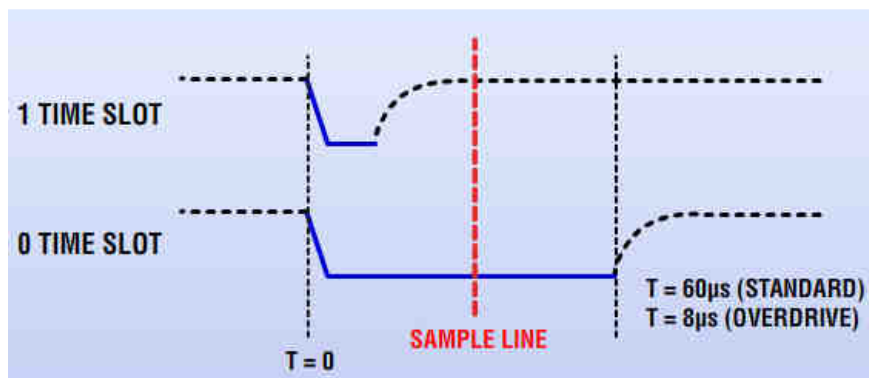
Kuvio 4. Hubin käyttö 1-Wire-verkossa (Network Topology, N.d.)

Verkkoa suunnitellessa kannattaa ottaa huomioon, että master-laitteeseen liitettävän verkon kokonaismitta ei saa ylittää 300 m ja on suositeltavaa käyttää CAT5-kaapelia. Myös muunlaisen kaapelin käyttö on mahdollinen, mutta kaapelin ominaisuudet voivat vaikuttaa verkon maksimimitaan. Myöskään kaapelin sijoittamista virtajohtojen läheisyyteen ei suositella (voi aiheuttaa häiriöitä). 1-Wire-verkkoon voidaan kytkeä noin 100 slave-laitetta, mutta suunnittelussa kannatta kuitenkin ottaa huomioon kyselyyn kuluva aika. Monessakin tapauksessa kyselyyn käytetyllä ajalla ei ole merkitystä, mutta jos käsitellään esimerkiksi liikkeentunnistimia tai muuta vastaavaa reaaliaikaista dataa vaativaa järjestelmää, kannattaa slave-laitteiden määrä pitää pienenä, jolloin kyselyyn käytetty aika pysyy pienenä. (Network Topology, N.d.)

2.2.3 1-Wire-väylä

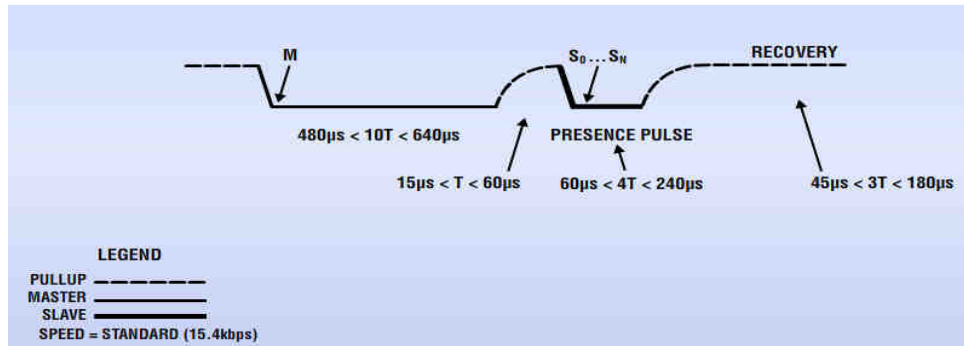
1-Wire väylä on kaksisuuntainen, ja väylänopeutena vakiona on 15,4 kbps. Väylää on myös mahdollista ajaa "overdrive" -tilassa, jolloin väylän nopeus kasvaa 125 kbps. Väylää ohjataan master-laitteella, yleensä mikrotietokoneella. Väylässä voi olla vain yksi master-laite, joka huolehtii tiedonsiirron aloittamisesta. Väylään kytketyt slave-laitteita voi olla useampikin. Käytettäessä yksinkertaisia slave-laitteita, kuten lämpö-

tila-antureita, laitteiden kytkentään riittää 2 johdinta, jolloin laitteet saavat käyttövir-
tansa dataväylästä ja sisäisestä kondensaattorista väylän ollessa alatilassa. Tiedonsiir-
ron aikana väylä käy lyhyitä aikoja alatilassa, muutoin se on aina ylätilassa. Master- ja
slave-laitteiden dataliitännän tulee olla avokollektortyyppinen, joka mahdollistaa
väylän maadoittamisen master- tai slave-laitteen toimesta. Väylän liikennöinti perus-
tuu aikaväleihin (timeslot), ja dataa siirrettäessä aikavälin aikana luettu arvo väylän
tilasta määrää, onko bitti 1 vai 0 (ks. kuvio 5).



Kuvio 5. Aikaväli 1-Wire-väylässä (1-Wire Tutorial, N.d.)

Väylän liikennöinti alkaa aina nollauspulssilla (reset pulse). Master-laite vetää väylän
alatilaan 480...640 us:n ajaksi. Tämän jälkeen master-laite alkaa kuuntelemaan väy-
lää. Jos väylällä on slave-laitteita, ne vastaavat nollauspulssiin vetämällä väylän alati-
laan 60 us:n kuluttua 60...240 us:n ajaksi. Vaikka väylällä on useampi kuin yksi slave-
laite, master-laite ei tiedä tässä vaiheessa, kuin että väylällä on ainakin yksi slave-
laite (ks. kuvio 6).

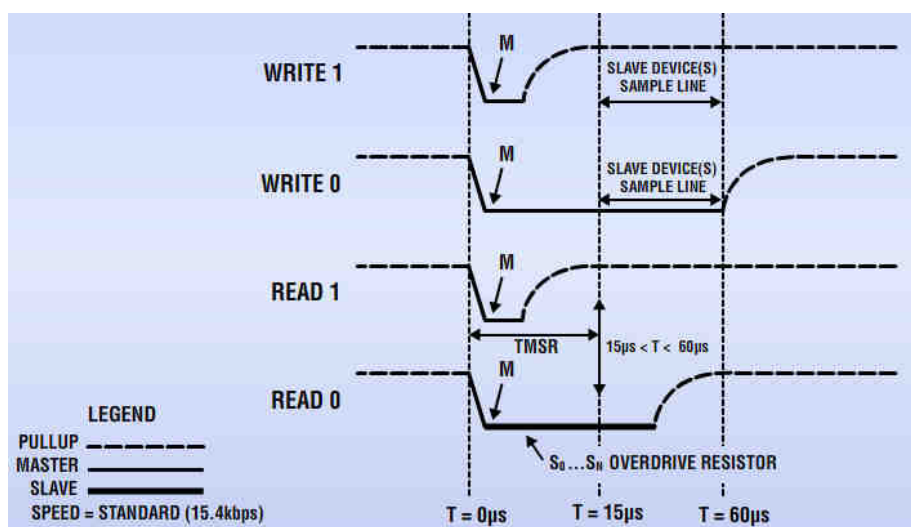


Kuvio 6. Nollauspulssi ja slave-laitteiden ilmoittama (presence) ”läsnäolo” -pulssi (1-Wire Tutorial, N.d.)

Kuviossa 7 on esitetty master-laitteen ja slave-laitteen välinen tiedonsiirto. Tiedon siirron aloittaa aina master-laite asettamalla väylän alatilaa.

Lähettyessä 1-bitin master-laite palauttaa väylän tilan 15 us:n aikana, jolloin slave-laitteen lukiessa väylän tilan 15 -60 us:n aikana, väylän on ylätilassa. Lähettyessä 0-bitin master-laite pitää väylän alatilassa koko 60 us:n ajan. Tällöin slave-laitteen lukiessa väylän tila on alhaalla.

Master-laitteen lukiessa väylää, se asettaa väylän alatilaa ja palauttaa tilan heti. Jos slave-laite ei vedä väylää ennen 15 us:a aloituksesta, niin tällöin slave-laitteesta luetaan 1-bitti. Jos taas slave-laite vetääkin väylän alatilaa ennen 15 us:a aloituksesta, niin slave-laitteesta luetaan 0-bitti. (1-Wire Tutorial, N.d.)



Kuvio 7. Kirjoittaminen ja lukeminen väylästä

Nollaus-pulssin jälkeen suoritetaan jokin ROM(Read Only Memory) -tason käsky (ks. taulukko 1.)

Taulukko 1. Listausta ROM-tason käskyistä

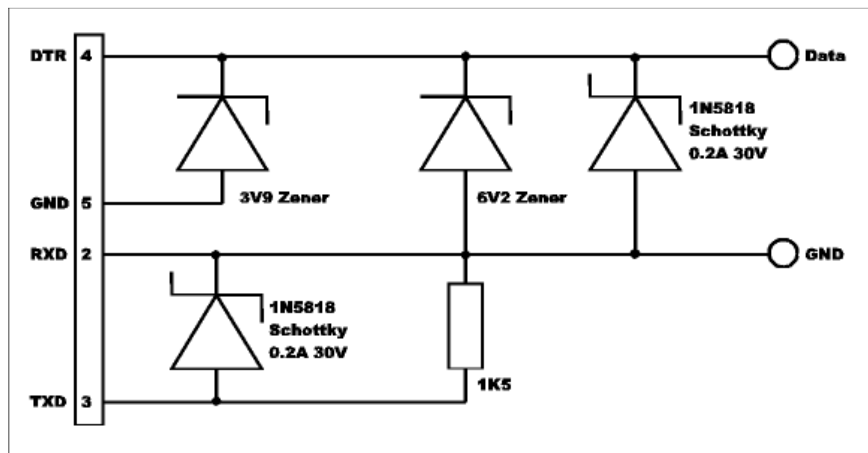
ROM-tason kasky	toiminto
skip ROM	jättää väliin laitteen valinnan (yksi laite väylässä)
read ROM	lukee 64 bit ROM ID:n (yksi laite väylässä)
match ROM	valitsee laitteen jo tiedetyn ROM ID:n perusteella
resume	aloittaa keskustelun uudestaan jo valitun laitteen kanssa
overdrive-Skip ROM	jättää väliin laitteen valinnan ja ottaa käyttöön overdrive-tilan
search ROM	etsii laitteet 1-Wire-väylästä

ROM-tason käskyn jälkeen suoritetaan laitteen omat käskyt. Esimerkiksi pyydetään lämpötila-anturilta sen hetkinen tilatieto.(1-Wire Tutorial, N.d.)

2.2.4 Adapteri

Jos verkko ollaan kytkemässä tietokoneeseen, adaptereita löytyy valmiina sarjaliikenneporttiin tai USB-porttiin. 1-Wire-verkon saa kytkettyä adapteriin RJ12-liittimellä käyttäen keskimmäistä paria. Liittimen kytkennästä tarkemmin liittimien kytkennät osiossa luvussa 2.2.8. Valmiissa adapterissa käytetään DS9097U -mikropiiriä, eli ohjelmassa täytyy olla tuki tälle piirille.

PC:n sarjaliikenneporttiin voidaan muuttaa jännitetasoiltaan antureille sopivaksi toteuttamalla kytkentä muutamasta komponentista ja liittimistä (ks. kuvio 8). Kytkennässä käytetään: 3.9 V zener-diodia, 6.2 V zener-diodia, 1.5 kohm vastusta ja kahta IN5818 schottky-diodia. Komponentit mahtuvat hyvin D9-liittimen sisälle. Rakennettun adapterin saa toimimaan ohjelmassa DS9097E -adapterina. (Adaptors, N.d.)



Kuvio 8. Adapterin kytkentäkaavio (Adaptors, N.d.)

2.2.5 Master hub

Vaihtoehtoisesti voidaan käyttää master hubia 1-Wire-verkon liittämiseksi tietokoneeseen. Etuna pelkkään adapteriin on, se että hubiin voidaan kytkeä kuusi erillistä 1-Wire-verkkoa tähtiverkoksi. Hubi poikkeaa adapterista myös siinä, että se vaatii ulkoisen virtalähteen. Master hubi ei vaadi erillistä adapteria tietokoneeseen, vaan se kytketään suoraan sarjaliikenneporttiin. Hubissa itsessään on sama Dallasin (DS2480B) mikropiiri kuin sarjaliikenne adapterissakin. Tästä syystä voidaankin käyttää samoja ohjelmia kuin pelkkää adapteria käytettäessä. (1-Wire Tutorial, N.d.)

2.2.6 Hub

Hubin ja master hubin erona on, että hubi vaatii adapterin, että se voidaan kytkeä tietokoneeseen. Hubissa on 8 porttia, joista in-porttiin tulee tietokone kiinni ja out-porttiin voidaan kytkeä toinen hubi tai vaihtoehtoisesti virransyöttö verkolle. Eli yhteen hubiin saadaan kytkettyä 6 erillistä 1-Wire verkkoa kuten master hubiinkin. (1-Wire Tutorial, N.d.)

2.2.7 Lisävirransyöttö

Yksinkertaisissa verkoissa ei lisävirransyötölle ole tarvetta. Esimerkiksi lämpötila antureille riittää dataväylästä saatava virta. Kuitenkin on laitteita, jotka vaativat lisävirtaa, että ne pystyvät toimimaan. Onkin syytä käyttää aina lisävirtaa, jos vaan jokin laite sitä pystyy käyttämään, ja näin saadaan dataväylän kuormitusta pienennettyä.

On myös suositeltavaa käyttää lisävirtaa, jos 1-Wire-verkko on pitkä ja verkossa on laitteita, jotka pystyvät lisävirran hyödyntämään, ja vaikka ei sitä välttämättä tarvitsikaan. Lisävirta-adapteri syöttää 5 voltia reguloitua jännitettä ja 15 voltia ei reguloitua jännitettä verkkoon. Lisävirtaa ei syötetä dataväylään, eli jos laite ei kykene käyttämään kuin dataväylää, lisävirta on tarpeeton. (1-Wire Tutorial, N.d.)

2.2.8 Liittimien kytkennät käytettäessä RJ12/J45 -liittimiä

1-Wire-verkon kytkennöissä käytetään RJ12- tai RJ45-liittimiä. Kaapeloinnissa suositellaan käytettäväksi Cat-5- tai Cat-6-kaapelia. Kytkentäjärjestys Cat-5- tai Cat-6-kaapeleilla tehdään kuvion 9 mukaisesti. Kuviossa 9 RJ45-liitännässä näkyvät lisävirsyöttöön käytetyt parit. (Wiring Diagram, N.d.)

1	2	3	4	5	6
		Blue	White Blue		
Not Connected	Not Connected	1-Wire Data DQ	1-Wire Return GND	Not Connected	Not Connected

Adaptors RJ12

1	2	3	4	5	6	7	8
White Green	Green	White Orange	Blue	White Blue	Orange	White Brown	Brown
Power Return GND	+5v DC Regulated +5v	Power Return GND	1-Wire Data DQ	1-Wire Return GND	Not Connected	+12v to +24v Unregulated	Power Return GND

RJ45

Kuvio 9. RJ12-liittimen ja RJ45-liittimen 1-Wire-kytkentä (Wiring Diagram, N.d.)

2.3 XML

Extensible Markup Language (XML) on kuvauskieli. Se on suunniteltu tiedonsiirtoon ja tallentamiseen. XML tallentaa tiedon tekstimuodossa ja tarjoaa laitteisto riippumattoman tavan tallentaa tietoa. XML on helposti luettavissa, eikä vaadi välttämättä mitään laitetta. (XML Tutorial, N.d.)

2.3.1 XML-tiedoston rakenne

XML:ssä itsessään ei ole mitään toiminnallisuutta. XML on rakenteeltaan itse kuvaileva. Kuviossa 10. esitetty XML-rakenteesta on luettavissa, että kyse on viestistä, keneltä viesti on, kenelle menossa, viestin otsikko ja itse viesti. XML alkuun kannattaa lisätä tieto XML:n versiosta ja missä merkistössä tiedosto on. Tämä helpottaa tiedoston käyttöä ja välttää merkistövirheitä.

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Harri</to>
  <from>Mika</from>
  <heading>Muistutus</heading>
  <body>Muista palauttaa opinnäytetyö</body>
</note>
```

Kuvio 10. XML:n rakenne

XML on HTML (Hypertext Markup Language) tyylinen kuvauskieli. Erona lähinnä se, että XML on suunniteltu tiedon välittämiseen ja HTML tiedon näyttämiseen. Ja XML:ssä käytetyt tagit eivät ole ennalta määrättyt, kuten HTML:ssä. Tästä syystä XML:stä saadaan tehtyä kuvailevampaa ja sidottua tiedon sisältöön paremmin, kuten kuviossa 10 on tehty. XML:n rakenteesta ja tageista vastaa XML:n suunnittelija.

XML on laajennettavissa myöhemmin. Kuvion 10 XML voidaan laajentaa sisältämään esimerkiksi päivämääräkentän (ks. kuvio 11) ja vaikka ohjelmaa, millä luetaan XML-tietue, ei päivitetäisikään uuteen versioon, niin uusi XML on kuitenkin luettavissa vanhalla ohjelmalla. Ainoastaan uuden kentän arvot jää käyttämättä. (XML Tutorial, N.d.)

```

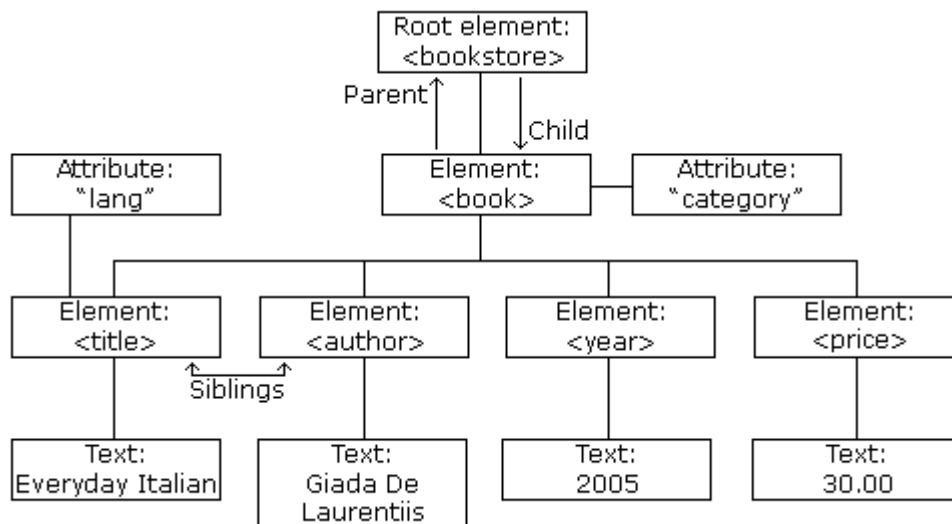
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <date>2015-12-05</date>
  <to>Harri</to>
  <from>Mika</from>
  <heading>Muistutus</heading>
  <body>Muista palauttaa opinnäytetyö</body>
</note>

```

Kuvio 11. XML:n laajennettu rakenne

2.3.2 XML-tiedoston puurakenne ja elementit

XML-tiedosto on rakenteeltaan puumainen (ks. kuvio 12).



Kuvio 12. XML:n puurakenne (XML Tutorial, N.d.)

Tiedosto alkaa juurielementillä (root) ja sisältää lapsielementtejä (child) , joilla voi olla myös lapsielementtejä (subchild) . Juuri elementti on aina pakollinen. Lapsielementtien määrä voi vaihdella ja jokaisella elementillä voi olla lapsielementtejä. Elementti alkaa aina aloitustagilla ja loppuu lopetustagiin. Kuten esimerkiksi kuviossa 11 (<note>...</note>). Tagi voi olla myös itsensä lopettava tagi (self-closing tag) (<element />) tällöin ei erillistä lopetustagia tarvita .

Elementti voi sisältää tekstiä, ominaisuuksia (attributes), muita elementtejä ja näiden yhdistelmiä. Elementin nimessä kirjainten koolla on merkitystä. Nimi tulee aloittaa kirjaimella tai alaviivalla, kuitenkin nimi ei saa alkaa kirjainyhdistelmällä "xml" eikä sisältää välilyöntiä. Ongelmien välttämiseksi kannattaa skandinaavisien merkkien käyttöä välttää. (XML Tutorial, N.d.)

2.3.3 XML-elementin attribuutin käyttö

Elementti voi sisältää attribuutin (`<note id="1">...</note>`) Attribuutin arvo tulee aina olla lainausmerkkien sisällä. Attribuutin käytössä ei ole sääntöä, milloin käyttää attribuuttia, ja milloin XML elementtiä. Attribuutin käyttöä puoltaa, jos ei tallenneta kuin yksi arvo tai attribuutti toimii metatietona, eikä ole tarvetta myöhemmin laajentaa. Huomioitavaa on, ettei attribuuttiin voi lisätä elementtejä. (XML Tutorial, N.d.)

2.4 JSON

JavaScript Object Notation (JSON) on kevyt tiedonvälitys muoto. Se on helposti ymmärrettävää ja itse kuvaavaa. JSON:n muoto vastaa rakenteeltaan JavaScript objektin muotoa. Samankaltaisuudesta johtuen JavaScript voi muuntaa JSON-tiedon suoraan JavaScript-objektiksi. Tästä huolimatta JSON on vain tekstiä.

JSON kuten XML ovat molemmat helposti luettavissa ilman erillistä ohjelmistoa. Rakenteeltaan hierarkkinen ja käytettävissä eri ohjelmointikielien kanssa. JSON:n ja XML:n erona on, ettei JSON-tietuetta tarvitse jäsentää kuten XML-tietuetta. Elementtejä ei merkitä tageilla ja kirjoitusmuoto on lyhyempää, kuin XML:ssä. JSON voi myös sisältää ryhmiä (arrays). Rakenteensa ansiosta JSON on nopeampaa käyttää, kuin XML. Tästä syystä JSON:n käyttö on yleistynyt varsinkin AJAX (Asynchronous JavaScript And XML) sovelluksissa. (JSON tutorial, N.d.)

2.4.1 JSON-tiedoston rakenne

JSON-tiedosto on rakenteeltaan avain/arvo pareja. Tietueen avain ja arvo tulee olla lainausmerkkien sisällä ja avaimen ja arvon väliin tulee kaksoispiste. JSON-tietueen arvot monesti kuuluvat objektiin, jolloin niitä ympäröi aaltosulkeet. Hakasulkeilla

merkitään ryhmän tietueita. Kuviossa 13 on kuvattu JSON-tiedoston rakenne, missä "employees" avain sisältää kolme erillistä ryhmää. Ryhmän sisältö on määritetty objektiksi. (JSON tutorial, N.d.)

```
"employees": [
  {"firstName": "John", "lastName": "Doe"},
  {"firstName": "Anna", "lastName": "Smith"},
  {"firstName": "Peter", "lastName": "Jones"}
]
```

Kuvio 13. JSON:n rakenne (JSON tutorial, N.d.)

2.4.2 JSON:n käyttötavat

Yleisin käyttötapaus on lukea JSON palvelimelta ja näyttää luettu tieto verkkosivulla JavaScriptiä käyttäen. JSON:n syntaksia voidaan käyttää myös esimerkiksi JavaScriptin muuttujassa (ks. kuvio 14). Varsinkin kehitysympäristössä muuttujan käyttö voi olla tarpeen, jos varsinaista tietoa ei ole saatavilla. JSON-tietueiden käyttö myös PHP ohjelmointikielessä on yleistynyt. JSON voidaan tallentaa suoraan MySQL tietokantaan sellaisenaan, jolloin sen käyttö JSON muodossa ei vaadi mitään käsittelyä. (JSON tutorial, N.d.)

```
<script>
var text = '{"name":"John Johnson","street":"Oslo West 16","phone":"555 1234567"}';

var obj = JSON.parse(text);

document.getElementById("demo").innerHTML =
obj.name + "<br>" +
obj.street + "<br>" +
obj.phone;
</script>
```

Kuvio 14. JSON JavaScript muuttujassa ja sen käsittely (JSON tutorial, N.d.)

2.5 LAMP

LAMP (Linux, Apache, MySQL, PHP) lyhennettä käytetään kokoelmasta avoimeen lähdekoodiin pohjautuvista ohjelmista. Kokoelma sisältää Linux-käyttöjärjestelmän, Apache HTTP-(Hypertext Transfer Protocol) palvelinsovelluksen, MySQL- tietokantarakajapinnan ja PHP-, Perl- tai Python- ohjelmointikielen. LAMP on saatavilla liki jokaisessa palvelinhotellipaketissa. Sillä voidaan tehdä mutkikkaitakin ja monipuolisia verkkopalveluita. Yksi tunnetuimmista verkkopalveluista, mikä on tehty LAMP -ympäristöön on Facebook.

Hyötynä LAMP-kokoelman käytöstä on, ettei kustannuksia tule ohjelmistoista. Toki maksullisia Linux-versioita löytyy, jolloin laajemmat tukipalvelut on käytettävissä. Ohjelmisto on myös räätälöitävissä tarpeen mukaan. (LAMP, N.d.)

2.5.1 Asennus

Yleensä LAMP asennetaan yhdelle palvelimelle. Nykyään käytetään usein virtuaalista pilvipalvelinta. Yhden palvelimen järjestelmällä voidaan ylläpitää pienen kävijämäärän sivustoja. Isompi kävijämäärä aiheuttaa palvelimelle enemmän prosesseja ja sivuston latautumisajat pitenevät aiheuttaen ongelmia palvelulle. Pahimmillaan palvelun kaatumisen.

LAMP voidaan myös hajauttaa useammalle koneelle, jolloin kaikkien koneiden käyttöjärjestelmä on Linux, mutta itse palvelut voidaan hajauttaa kahdelle koneelle. Yhdelle palvelimelle asennetaan Apache ja PHP. Tämä palvelin huolehtii WWW- (World Wide Web) liikenteestä ja tarjoaa dynaamiset WWW-sivut käyttäjille. Toiselle palvelimelle asennetaan MySQL-tietokanta. Tarvittaessa voidaan WWW-palvelin kahdentaa käyttäen samaa tietokantaa. Tällöin nimipalveluun lisätään kaksi WWW- palvelinta. Palvelimien välistä kuormantasausta voidaan toteutetaan erillisellä laitteella.

Käytettävissä on myös Amazon tyyllisiä palveluita, missä palvelu jaetaan vähintään kolmelle palvelimelle. WWW palvelu voidaan toteutetaan Linux-ympäristössä PHP tuella. Palvelun ero perinteisempään on, että palvelimia voidaan kuormituksen yhteydessä lisätä automaattisesti. Tällöin palvelinresurssia on aina käytettävissä tarvittava määrä. Tietokantapalvelu toteutetaan erillisellä tietokantapalvelimella ja tiedostot omalla S3 -(Simple Storage Service) palvelulla. (Amazon WEB Service, N.d.)

2.5.2 Linux

2.5.2.1 Yleistä

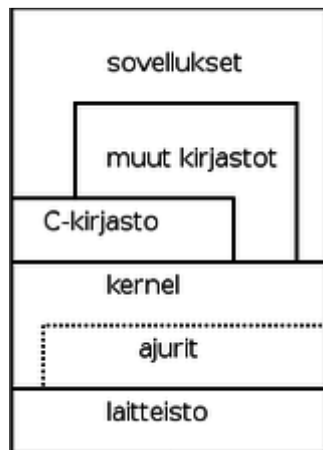
Linux on maailman yleisin palvelinkäyttöjärjestelmä. Linux pohjautuu Unix käyttöjärjestelmään. Linux on avoimen lähdekoodin järjestelmä ja kuuluu GPL (General Public License) lisenssin alaisuuteen. Se on vapaasti kopioitavissa ja muokattavissa oleva käyttöjärjestelmä. Linux-ytimen alkuperäinen kehittäjä oli Linus Torvalds, mistä Linux nimikin juontaa juurensa. Linux on saatavilla eri jakelupaketteina. Yleisimpiä ilmaisia jakelupaketteja ovat mm. Debian, Fedora, openSuse, Ubuntu ja CentOS. Maksullisia jakelupaketteja on mm. Suse ja RedHat.

Linuxia käytetään myös kämmentietokoneissa ja matkapuhelimissa. Lisäksi mukaan on tullut teollisuuden sulautetut järjestelmät. (Linux, N.d.)

2.5.2.2 Lyhyt kuvaus Linux käyttöjärjestelmän rakenteesta

Käyttöjärjestelmässä lähinnä prosessoria on käyttöjärjestelmän ydin. Linux- käyttöjärjestelmissä puhutaan tällöin kernelistä. Kerneliin kuuluu olennaisena osana laitteistoajurit. Linuxin ydin on ohjelma, mikä huolehtii prosessorin käytöstä, ja että ohjelmat saavat prosessoriaikaa tarvittavan määrän. Moniajossa prosessorin käyttöä vaihdellaan eri ohjelmien kesken. Ydin tarjoaa ohjelmille rajapinnan kaikkiin järjestelmän resursseihin, kuten muistiin, äänikortille, näytölle, kiintolevyille, jne. Ytimen tarjoama rajapinta ohjelmille on samanlainen huolimatta siitä esim. minkälainen äänikortti on koneeseen asennettu. Ytimelle voisi kirjoittaa C-kielellä ohjelman suoraankin, mutta yleensä välissä käytetään kirjastoja, jotka helpottavat ohjelman kirjoittamista.

Kirjastojen tehtävänä on helpottaa ohjelmien kirjoittamista. Kirjastoja voidaan käyttää tekemään joku ennalta määrätty toiminto. Kirjastoa tehtäessä määritellään, mihin toiminnallisuuteen kirjasto liittyy. Esim. Raspberry Pi:n GPIO väylään on kirjoitettu kirjasto, joka osaa esim. muuttaa pinnin tilan 1 ja 0 välillä. Ohjelmaa kirjoittaessa käytetään valmiiksi kirjoitettuja kirjastoja, jolloin esim. GPIO väylän pinnin tilan vaihto on ohjelmallisesti yksinkertaisempi tehdä. Kirjastojen tehtävänä onkin helpottaa ohjelmoijaa. Kuviossa 15 on kuvattu, missä kohdassa eri osa-alueet ovat laitteistoon nähden. (Linux-järjestelmän rakenne, N.d.)



Kuvio 15. Linux-käyttöjärjestelmän kuvaus (Linux-järjestelmän rakenne, N.d.)

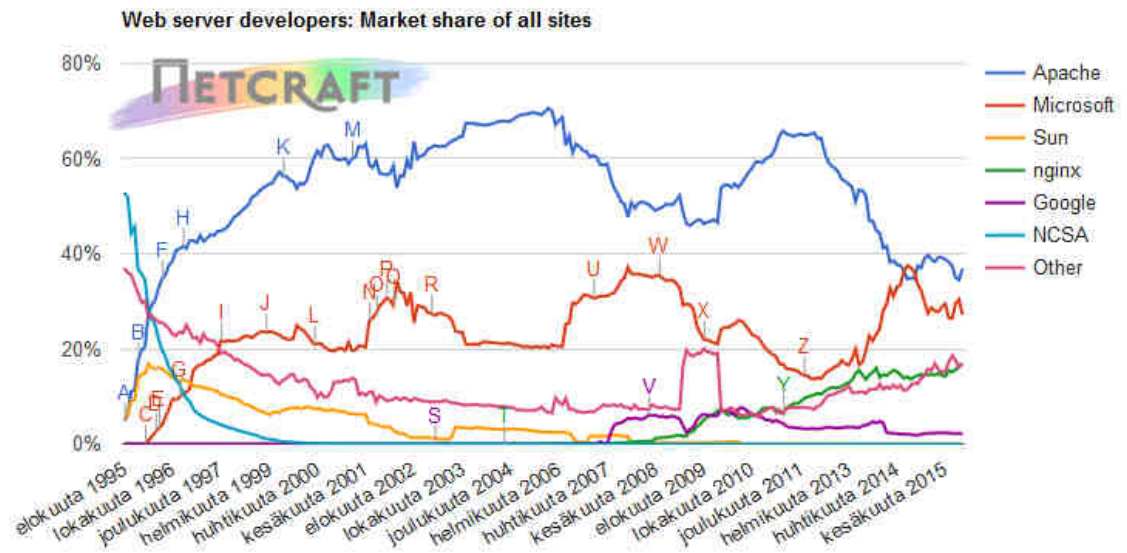
2.5.3 Apache

Apache on avoimen lähdekoodin HTTP -palvelinsovellus. Se on Apache Software Foundationin tunnetuin tuote. Apache on saatavilla mm. Unix-, Linux- ja Windows-käyttöjärjestelmille. Lisäksi se on integroitu OS X- käyttöjärjestelmään.

Apachen ensimmäinen versio oli joukko päivityksiä ja korjauksia NCSA:n palvelimelle, mutta version 2.0 myötä se on kirjoitettu kokonaan uudestaan. Ensimmäinen Apache on julkaistu 1995.

Apache itsessään tukee pelkästään staattisia WWW-sivuja, mutta eri moduuleilla voidaan räätälöidä palvelin vastaamaan käyttötarkoitusta. Esimerkiksi `mod_cgi` moduuli mahdollistaa CGI -(Common Gateway Interface) scriptien ajamisen palvelimella, `mod_perl` lisää Perl tuen ja `mod_rewrite` moduulilla pystytään manipuloimaan verkko-osoitetta. (Apache, N.d.)

Apache on ollut pitkään suosituin HTTP-palvelin sovellus. Toki pudotusta on tullut parhaimmilta vuosilta 71%:sta tämän päivän 37%:n. Microsoftin IIS tulee toisena ja nginx kolmantena kuvion 16 mukaisesti. (November 2015 Web Server Survey, N.d.)



Kuvio 16. HTTP palvelinsovellusten käyttö kuvaaja (November 2015 Web Server Survey, N.d.)

2.5.4 MySQL

MySQL on relaatiotietokantaohjelmisto. Ohjelmiston on alunperin kehittänyt MySQL AB, nykyään ohjelmiston oikeudet omistaa Oracle Corporation. Ohjelmistoa on saatavilla kahdella eri lisenssillä, GPL- ja kaupallisella lisenssillä. Ohjelmisto on yksi johtavista avoimen lähdekoodin WEB-tietokantaohjelmistoista, kuten myös pilvipalvelutietokantana. MySQL-tietokantaa käyttää mm. Facebook, Twitter ja LinkedIn. Pilvipalvelutietokantana MySQL käyttää mm. Amazon Web Services ja Google. (Top 10 Reasons to Choose MySQL for Next Generation Web Applications, N.d.)

2.5.4.1 Tietokannan kahdentaminen

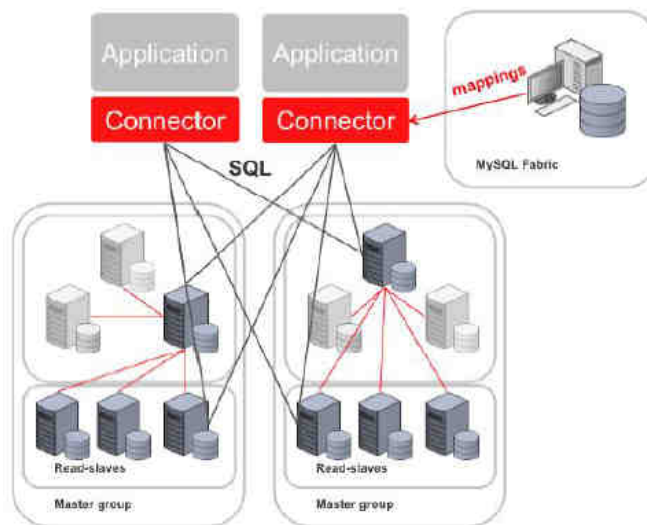
Skaalautuvuutensa ansiosta MySQL soveltuu hyvin myös suuriin tietomääriin. Tietokantaa voidaan skaalata kahdentamalla (replication), jolloin tieto kopioidaan master-tietokannasta slave-tietokantaan tai tietokantoihin. Tietokannan kahdentamista voidaan tehdä myös usealta master-tietokannalta yhdelle slave tietokannalle. Yleensä tällainen kahdentaminen toteutetaan varmuuskopioinnin takia. (Top 10 Reasons to Choose MySQL for Next Generation Web Applications, N.d.)

2.5.4.2 Tietokannan reitittäminen

Reititystä käyttäen voidaan tietokannan saatavuutta parantaa. Käytettäessä kahdennettua tietokantaa, reitityksen avulla voidaan liikennettä ohjata eripalvelimille liikennemäärien mukaan ja tasata kuormaa tietokantojen välillä (load balancing). Reititys ottaa huomioon myös, jos jokin tietokanta vikaantuu, niin liikenne ohjataan ehjille palvelimille. (Top 10 Reasons to Choose MySQL for Next Generation Web Applications, N.d.)

2.5.4.3 Hajautettu ja korkea saatavuus

MySQL Fabric (ks. kuvio 17) on suunniteltu hallinnoimaan MySQL -palvelinryhmiä ja hajautettua tietokantaa. Tietokannan hajauttaminen useammalle MySQL- palvelimelle ja eri fyysiseen kohteeseen takaa tietokannan korkean saatavuuden, vikasietoisuuden ja skaalautuvuuden. (Top 10 Reasons to Choose MySQL for Next Generation Web Applications, N.d.)

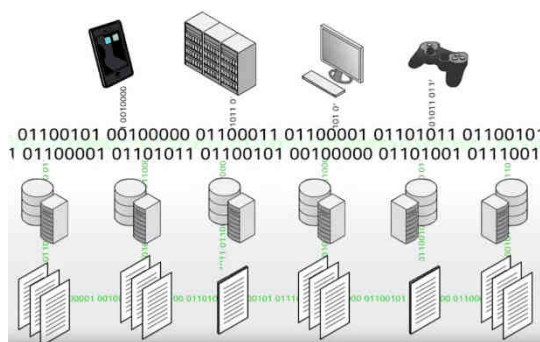


Kuvio 17. MySQL Fabric (Top 10 Reasons to Choose MySQL for Next Generation Web Applications, N.d.)

2.5.4.4 My SQL klusteri

Klusteri on suunniteltu parantamaan reaaliaikaisen muistin suorituskykyä ja 99.9999%:n saatavuuteen. MySQL klusteri on skaalautuva ja siinä voi olla monta master-tietokantaa (ks. kuvio 18). Klusteriin voi kuulua eri datakeskusten palvelimia

ja skaalautumista voidaan tehdä eri mantereiden välilläkin. Klusteri on myös vikasetoinen. Yhden solmun (node) vikaantuessa, se poistetaan klusterista ja jäljelle jäävät huolehtivat vikaantuneen liikenteestä. Kun vikaantuneen tilalle saadaan nostettua uusi solmu, liikenne ohjataan alkuperäiselle reitille. Kaikissa solmuissa on sama tieto ja se on kopioitavissa uuteen solmuun. Erilaiset sovellukset kuten esim. pelit ja mobiili maksaminen vaatii hyvän saatavuuden ja lyhyen vasteajan, näihin palveluihin klusteri on oikea valinta. (Top 10 Reasons to Choose MySQL for Next Generation Web Applications, N.d.)



Kuvio 18. MySQL Cluster (Top 10 Reasons to Choose MySQL for Next Generation Web Applications, N.d.)

2.5.4.5 Tietokannan rakenne

Tietokantaa luodessa valitaan tietokannalle haluttu nimi ja merkistö, missä tietokannan sisältö tullaan tallentamaan. Käytettäessä ulkoista palvelua, tietokantaa ei välttämättä pysty itse luomaan, vaan se on luotu jo palveluntarjoajan toimesta.

Tietokanta sisältää tauluja ja taulujen sisässä on kenttiä. Taulua luotaessa annetaan taulun nimi ja mahdollisesti kenttien määrä. Määrää toki voi muuttaa myöhemmin. Kentille annetaan haluttu nimi ja määritellään tyyppi. Kuviossa 19 on luotu yksinkertainen taulu, mistä näkee yleisimmin käytetyt tyypit ja niitä vastaavat pakolliset arvot PHPMYAdminia käytettäessä.

Taulun nimi: demotaulu Add 1 column(s) Suorita

Rakenne									
Nimi	Tyyppi	Pituus/Arvot*	Oletusarvo	Aakkosjärjestys	Attribuutit	Tyhjä	Indeksi	A_I	Kommentit
id	INT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
nimi	VARCHAR	255	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
kuvaus	TEXT		None			<input type="checkbox"/>	---	<input type="checkbox"/>	
hinta	DECIMAL	10,2	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
aikaleima	TIMESTAMP		CURRENT_TIMESTAMP			<input type="checkbox"/>	---	<input type="checkbox"/>	

Taulun kommentit:

Tallennusmoottori: InnoDB

Aakkosjärjestys:

Kuvio 19. Taulun luominen tietokantaan

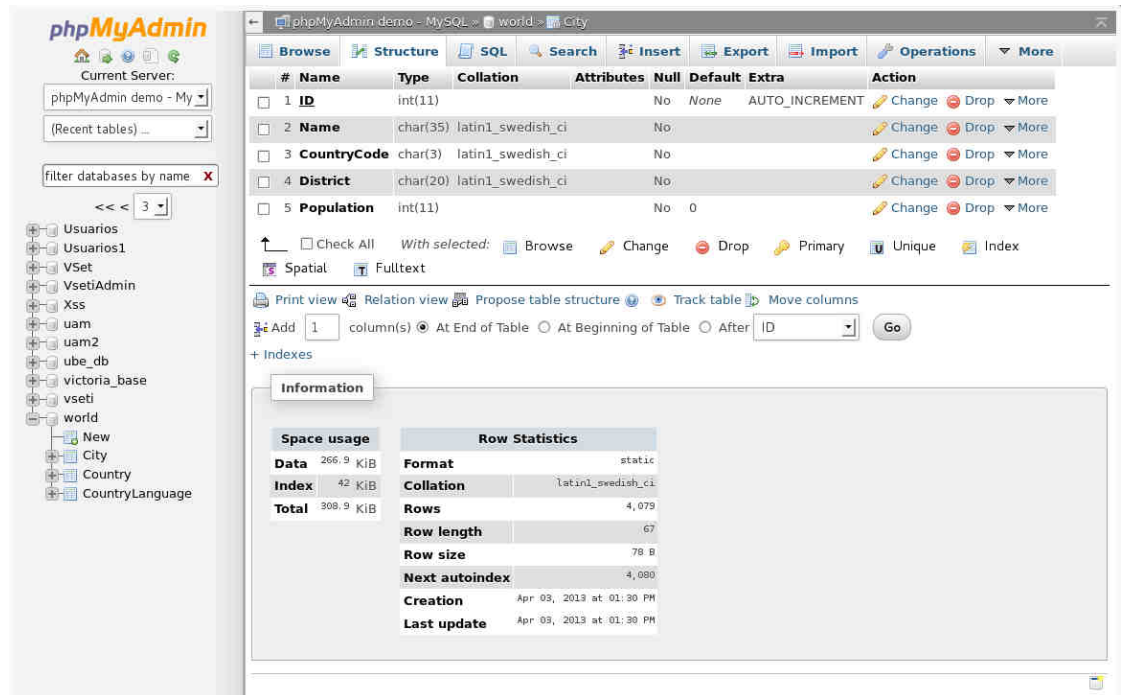
2.5.4.6 Tietokannan hallintasovelluksia

MySQL-tietokantaa varten on useampia hallintasovelluksia. Kaksi yleisintä hallintaan käytettyä sovellusta on PHPMyAdmin ja MySQL Workbench. Hallintasovelluksilla tietokannan hallinta onnistuu vaivattomasti ja sillä näkee taulujen rakenteen ja sisällön selkeästi. Hallintaohjelmaa käyttäen voidaan myös tietokannasta ottaa varmuuskopio koko tietokannasta tai halutuista tauluista. Osittainen kopiointi on suotavaa, kun tehdään kehitystyötä ja on olemassa pienikin riski taulun sisällön vääristymiseen tai tuhoutumiseen.

2.5.4.6.1 PHPMyAdmin

PHPMyAdmin lienee yleisin MySQL-tietokannan hallinta ohjelma. Se on selaimen kautta käytettävä ja se on toteutettu PHP:llä. PHPMyAdmin-ohjelma on alunperin Tobias Ratchillerin vuonna 1998 aloittama projekti, mutta luopui vuonna 2000 kehittämisestä ajan puutteen vuoksi. Nytemmin kehityksestä vastaa PHPMyAdmin-yhteisö. PHPMyAdmin on helppo asentaa projektin käyttöön, jos palvelun tarjoaja ei sitä valmiiksi tarjoa. Ohjelmisto on ladattavissa yhteisön sivustolta ja asennettavissa ilmaiseksi palvelimelle. Ohjelmisto on rakennettu siten, että se pystyy ottamaan paikalliseen palvelimeen (local host) suoraan yhteyden ilman mitään erillisiä asennuksia. Käyttöön vaaditaan tietokannan käyttöoikeus tunnukset.

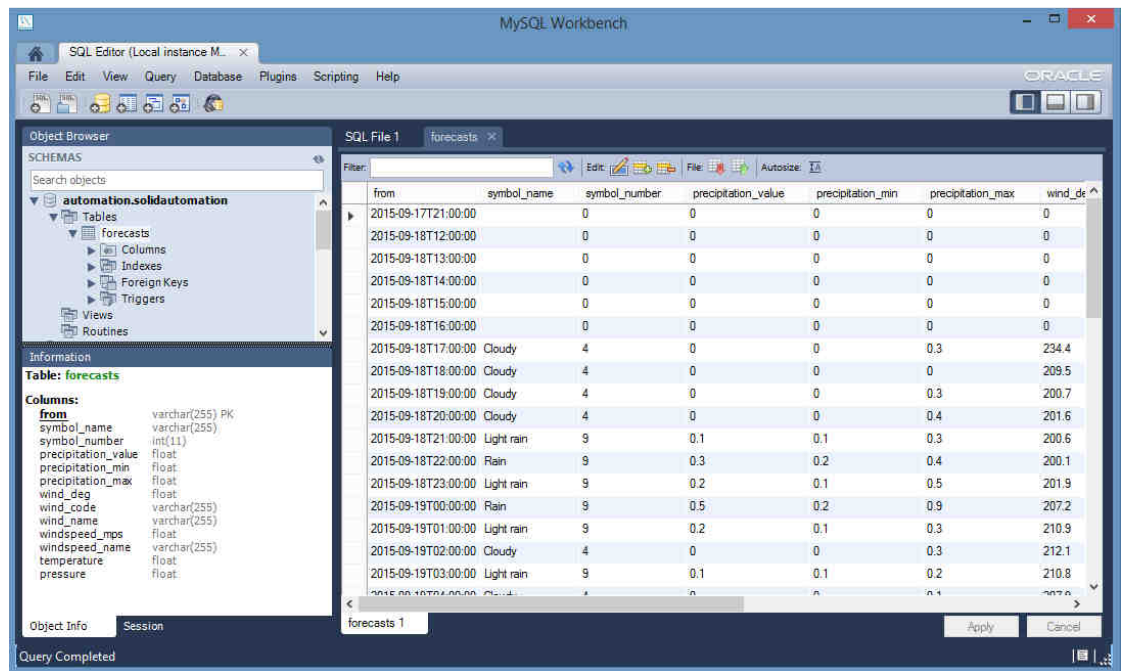
PHPMyAdminilla pystytään hallinnoimaan tietokannan koko sisältöä ja monitoroimaan kyselyiden määrää, yhteyksien määrää ja palvelimen muistin ja suorittimen kuormitusta. Kuviossa 20 on esitetty näkymä taulun rakenteen hallinnasta. (PHPMyAdmin, N.d.)



Kuvio 20. PHPMyAdmin-käyttöliittymä

2.5.4.6.2 MySQL Workbench

Oracle tarjoaa MySQL Workbench-työpöytäsovelluksen tietokannan hallintaan. Sovelluksen toiminnan edellytyksenä on, että MySQL-palvelin sallii tietokannan etäkäytön. Monissa projekteissa etähallinta on tietoturvasyistä kielletty. Tällöin käyttöliittymä täytyy olla samalla palvelimella, kuin missä itse tietokantakin on. Oraclen sovelluksella voidaan hallinnoida tietokannan kaikkia toiminnallisuuksia ja monitoroida käyttöä. Kuviossa 21 on esitetty näkymä tietokannan taulun sisällöstä. (MySQL Workbench, N.d.)



Kuvio 21. MySQL Workbench-käyttöliittymä

2.5.5 PHP

Hypertext Preprocessor (PHP) on Perlin kaltainen skripti ohjelmointikieli. PHP:tä käytetään palvelinympäristössä dynaamisten WWW-sivujen luomiseen. PHP-scriptit suoritetaan aina palvelimella ja luotu WWW-sivu palautetaan HTML muodossa käyttäjälle. PHP-scriptejä voidaan myös ajaa palvelimen taustasovelluksissa kutsumalla niitä esim. ajastettuna Linuxin cronia käyttäen. (PHP Tutorial, N.d.)

2.5.5.1 PHP tiedoston rakenne

PHP-tiedosto voi sisältää tekstiä, HTML-koodia, CSS- (Cascading Style Sheets) tyylitietoja, JavaScriptejä ja PHP-koodia. Kuviossa 22 on esitetty yksinkertainen PHP tiedosto, joka sisältää myös HTML-koodia. Taustasovelluksissa ei välttämättä tuoteta mitään ruudulle tulostettavaa, jolloin tiedosto ei sisällä kuin pelkän PHP koodin. PHP:llä voidaan toteuttaa myös esim. JSON-rajapinta, jolloin palautetaan haluttu tieto JSON-syötteenä. (PHP Tutorial, N.d.)

```

<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>

```

Kuvio 22. Yksinkertainen PHP-scriptti (PHP Tutorial, N.d.)

2.5.5.2 PHP:n käyttö

PHP:n yleisin käyttö on tuottaa dynaamisia WEB-sivuja. Yleensä sivun tuottamiseen liittyy taustatoiminnallisuutta esim. tietokannasta lukemista tai tiedostojen lukemista. PHP:llä voidaan ajaa tehokkaasti eri komentoja palvelimella ja esim. MySQL:n käyttö on yksinkertaista (ks. kuvio 23). PHP:llä voidaan tietokannan lisäksi mm. käsitellä tiedostoja, lähetetyn lomakkeen syötettyä tietoa, lukea ja kirjoittaa evästeisiin, hallinnoida käyttöoikeuksia ja salata ja purkaa salauksia. (PHP Tutorial, N.d.)

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

Kuvio 23. Tiedon haku MySQL-tietokannasta PHP:llä (PHP Tutorial, N.d.)

2.5.5.3 PHP:n tuettuja yleisimpiä järjestelmiä

PHP on asennettavissa useammalle käyttöjärjestelmälle. Windows, Linux, Unix, Mac OS X ovat mm. tuettuja käyttöjärjestelmiä. PHP tukee melkein kaikkia WWW-palvelin ohjelmistoja, kuten esim. Apache, IIS. Tietokantojen tuki on myös laaja. Yleisimpiä tuettuja tietokantoja ovat MySQL, Microsoft SQL, Postgre SQL, SQLite. (PHP Tutorial, N.d.)

2.6 Pilvipalvelu

2.6.1 Pilvipalvelutyypit

Pilvipalveluita on kolmea eri tyyppistä palvelua. IaaS (Infrastructure as a Services) tarjoaa suoritusympäristön käyttöjärjestelmille, kuten virtuaalikoneet. PaaS (Platform as a Service) on ympäristö, missä voi suorittaa pilvialustalle suunniteltuja ohjelmistoja. SaaS (Software as a Service) ympäristössä kaikki palvelun osat ovat pilvessä ja palvelun tarjoaja huolehtii palvelusta. SaaS palvelua käytetään asiakasohjelmalla. Yksi tunnetuimmista pilvipalveluista on Microsoftin Office 365. Microsoft on kehittänyt myös palvelun, missä työasema on sijoitettu pilvipalveluun. Pilvipalvelun etuna on, että ne on käytettävissä kaikkialla Internetissä.

Pilvipalvelun etuna on, että palveluntarjoaja pystyy hajauttamaan käyttökuorman useammalle fyysiselle koneelle. Loppukäyttäjä ei tätä näe käytössään, vaan palvelu toimii kuin olisi paikallisesti tuotettu. Hajauttaminen antaa myös vikasietoisuutta, koska palvelu on skaalautuva ja yksittäinen hajonnut kone korvataan uudella automaattisesti. (Hyödynnä pilvipalvelua viisaasti, N.d.)

Pilvipalvelu jakaantuu julkiseen ja yksityiseen pilveen. Julkinen pilvipalvelu on jaettava muiden palveluntarjoajan asiakkaiden kanssa. Yksityinen pilvi varataan kokonaan yhdelle asiakkaalle ja se muodostaa oman suljetun verkon. (Pilvipalveluiden turvallisuus organisaatioiden näkökulmasta, N.d.)

2.6.2 Tiedon tallentaminen pilvipalveluun

Pilvipalveluun tallentaessa käyttäjän tulee varmistua siitä, ettei riko oman maan lakia, minkä maan lakia sovelletaan mahdollisissa poikkeustapauksissa ja minkä maan viranomaisilla on oikeus tutkia tätä aineistoa. Esimerkiksi henkilötietojen tallentami-

nen pilvipalveluun voi rikkoa lakia. Useat pilvipalveluntarjoajat käyttävät maantieteellistä hajautusta palvelun varmistamiseksi, tällöin palvelinten välinen liikennöinti pitää olla salattua. (Pilvipalveluiden turvallisuus organisaatioiden näkökulmasta, N.d.)

3 Toteutus

Työhön kuuluu kolme toisistaan erillään olevaa kokonaisuutta. Mittaus- ja säätöyksikkö, joka pohjautuu Raspberry Pi -alustalla toteutettuun laitteeseen. Pilvipalvelu huolehtii tiedon tallentamisesta, sääennusteiden lataamisesta ja ohjaustiedon laskennasta. Mobiilisovellus mittaus- ja säätöyksikön hallintaa varten.

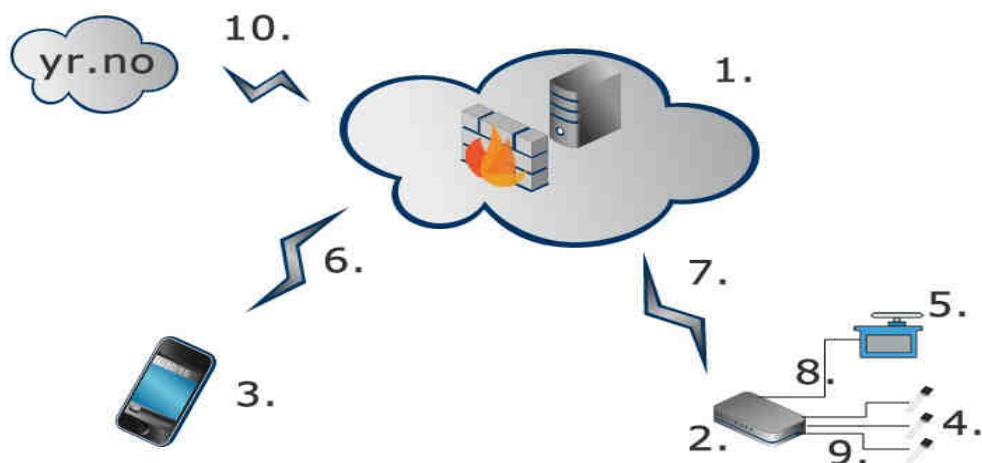
3.1 Toteutetun järjestelmän kuvaus

Toteutettu järjestelmä koostuu eri kokonaisuuksista ja niiden välisistä yhteyksistä. Kuviossa 24. kuvataan järjestelmän kokonaisuutta. Kuvioon liittyvät selitteet:

1. Pilvipalvelu on toteutettu virtuaalipalvelimella. Käyttöjärjestelmänä Centos 6 Linux-käyttöjärjestelmä. Palvelimen palveluina (services) asennettuna on Apache laajennettuna PHP-ohjelmointikielellä ja MySQL-tietokannalla (LAMP). Virtuaalipalvelin toteutettiin laas-pilvipalvelun päälle.
2. Mittaus- ja säätöyksikkö pohjautuu Raspberry Pi -alustaan. Käyttöjärjestelmänä toimii Raspbian, joka on Raspberryn optimoitu versio Debian Linux- käyttöjärjestelmästä. Raspberryn on asennettu GPIO- ja 1-Wire-kirjastot ja Python-ohjelmointikieli.
3. Mobiilisovellus Android-käyttöjärjestelmässä, joka on toteutettu Cordova-mobiilisovellusten kehitysympäristöllä.
4. DS18S20 1-Wire digitaalinen lämpötila-anturi (3 kpl).
5. Alturn USA ADS-966HTG digitaalinen servomoottori.
6. Mobiilisovellus suorittaa kyselyn pilvipalvelimen JSON-rajapintaan. Kyselyssä lähetetään lämmityksen säätöarvo. Palvelin palauttaa halutun tiedon mobiilisovellukselle. Yhteys on SSL (Secure Sockets Layer) suojattu ja lähetetty ja vastaanotettu JSON-tietueen oikeellisuus varmennetaan laskemalla tarkistussumma ja vertaamalla sitä saatuun tarkistesummaan. Tarkistesumman laskennassa käytetään salaista avainta. Mobiilisovellus voi sijaita NAT -(Network

Address Translation) palomuurin takana. Ei vaadi julkista IP -(Internet Protocol) osoitetta.

7. Mittaus- ja säätöyksikkö avaa yhteyden ja lähettää mittaustiedot pilvipalvelimelle. Palvelin palauttaa venttiilin lasketun lämpötila-arvon mittaus- ja säätöyksikölle. Tiedot lähetetään JSON-tietueena SSL-yhteyden yli. JSON-tietueen oikeellisuus varmennetaan vielä laskemalla tarkistesumma ja vertaamalla sitä saatuun tarkistesummaan. Tarkistesumman laskennassa käytetään salaista avainta. Mittaus- ja säätöyksikkö voi sijaita NAT palomuurin takana. Ei vaadi julkista IP-osoitetta.
8. Mittaus- ja säätöyksikkö ohjaa servomoottoria pulssileveysmodulaation (PWM, Pulse-Width Modulation) avulla
9. Lämpötila-antureiden tiedonsiirtoon käytetään 1-Wire-verkkoa.
10. Pilvipalvelin hakee säätiedot yr.no -palvelusta



Kuvio 24. Toteutetun järjestelmän kuvaus

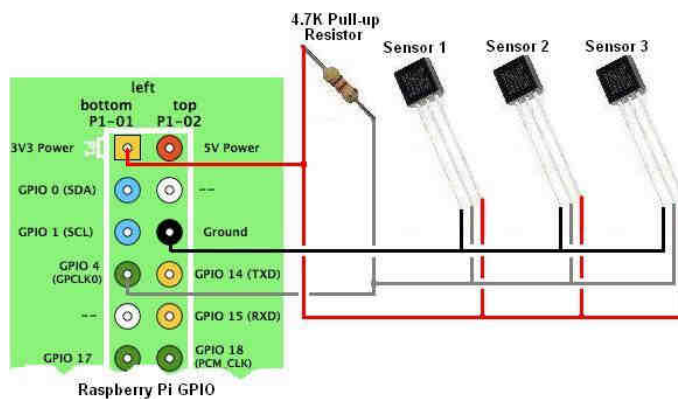
3.2 Mittaus- ja säätöyksikkö

Järjestelmän tehtävänä on välittää lämpötilatiedot lämpötila-antureilta pilvipalveluun ja säätää venttiili oikeaan lämpötilaan. Järjestelmä määriteltiin tilaajan toimesta rakennettavaksi Raspberry Pi -alustalle ja lämpötila-antureiksi oli valittu DS18B20.

3.2.1 Lämpötila-antureiden kytkeminen

Valitut lämpötila-anturit ovat digitaalisia antureita ja ne voidaan kytkeä suoraan 1-Wire-väylään. Antureihin on valmistuksen yhteydessä määritetty yksilöllinen id, jonka perusteella se voidaan tunnistaa 1-Wire-verkossa. Samaan väylään voidaan kytkeä useampikin anturi kerrallaan.

Raspberry ei tue 1-Wire-väylää suoraan, vaan väylää varten oli tehtävä muunnos kytkentä. 1-Wiren dataväylä voidaan kytkeä suoraan Raspberryn GPIO-väylään. GPIO -väylän kytketty pinni asetetaan kuuntelemaan väylän arvoa. Koska väylän tila on lepotilassa 3,3 voltia ja data välitetään maadoittamalla väylä, niin GPIO väylään liitettävä dataväylä tulee liittää GPIO väylän 3,3 voltin ulostuloon vastuksen avulla. Kytkentä pitää väylän jännitteen oikeassa arvossa ja mahdollistaa kuitenkin väylän maadoittamisen. Lisäksi antureille tulee kytkeä käyttöjännite 3,3 voltia tai 5,0 voltia ja maadoitus. Antureiden käyttöjännite voidaan ottaa suoraan GPIO-väylästä. Antureiden kytkentäkaavio esitetty kuviossa 25.

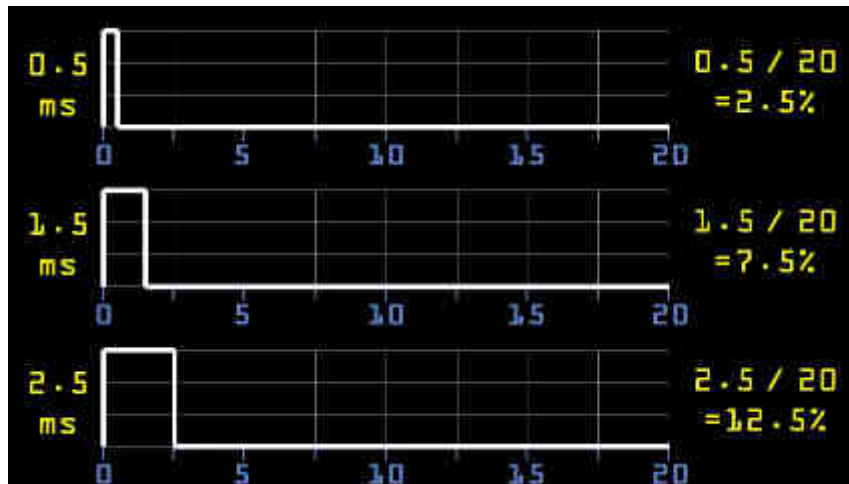


Kuvio 25. Lämpötila-antureiden kytkentä GPIO-väylään (Connect Multiple Temperature Sensors With Raspberry Pi, N.d.)

3.2.2 Servomoottorin kytkeminen

Venttiilin säätö päätettiin toteuttaa servomoottorilla. Käytettävissä oli Alturn USA ADS-966HTG radio-ohjattavan digitaalinen servomoottori. Moottorin käyttöjännite on 4,8 - 6,0 voltia, joka soveltuu hyvin raspberryn 5 voltin järjestelmään.

Moottoria ohjataan pulssileveysmodulaation avulla, jossa pulssin leveys muuttaa moottorin asentoa. Moottorin pulssinleveysalue on 0,5ms - 2,5ms. Kyseiset arvot vastaavat moottorin ääriasentoja. Kun taajuudeksi valitaan 50Hz, niin pulssin pituus on 20ms. Tällöin 0,5ms on 2,5% 20ms ja 2,5ms on 12,5% (ks. kuvio 26).



Kuvio 26. Pulssin pituus modulaatiossa (Servo Control with the Raspberry Pi, N.d.)

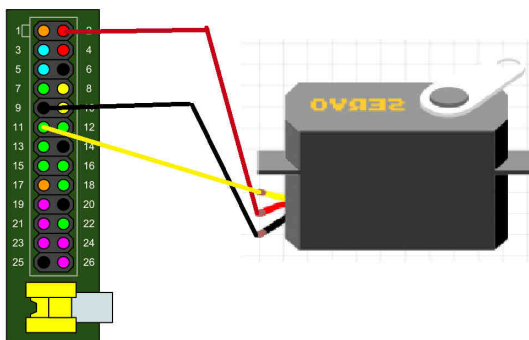
Servomoottorin säätöä varten raspberryn GPIO-väylän pinnin ulostulo asetetaan pulssileveysmodulaatio tilaan ja taajuudeksi valitaan 50Hz. Moottoria voidaan tämän jälkeen ohjata antamalla pulssin pituus prosentiarvona. Kuviossa 27 on esitetty Python-käskyt, miten saadaan servomoottori käyttöön ja käännettyä ääriasennosta toiseen.

```
# Set up pin 11 for PWM
GPIO.setup(11, GPIO.OUT) # Sets up pin 11 to an output (instead of an input)
p = GPIO.PWM(11, 50)     # Sets up pin 11 as a PWM pin
p.start(0)               # Starts running PWM on the pin and sets it to 0

# Move the servo back and forth
p.ChangeDutyCycle(3)     # Changes the pulse width to 3 (so moves the servo)
sleep(1)                 # Wait 1 second
p.ChangeDutyCycle(12)    # Changes the pulse width to 12 (so moves the servo)
sleep(1)
```

Kuvio 27. PWM käyttöön Python-ohjelmointikielellä (Raspberry Pi Learning Resources, N.d.)

Servomoottoriin on kytketty kolme johdinta. Musta maadoitusta, punainen moottorin käyttöjännitettä ja keltainen ohjaussignaalia varten. Käyttöjännite voidaan ottaa raspberryn GPIO-väylästä, mutta on suositeltavaa, että moottorilla olisi oma virtalähde. Moottorin aiheuttamat vaihtelut jännitteessä aiheuttaa helposti raspberryn uudelleen käynnistymisen. Signaalijohdin voidaan kytkeä suoraan GPIO-väylään kuvion 28 mukaisesti.



Kuvio 28. Servomoottorin kytkeminen GPIO-väylään (Raspberry Pi Learning Resources, N.d.)

3.2.3 Merkkivalot

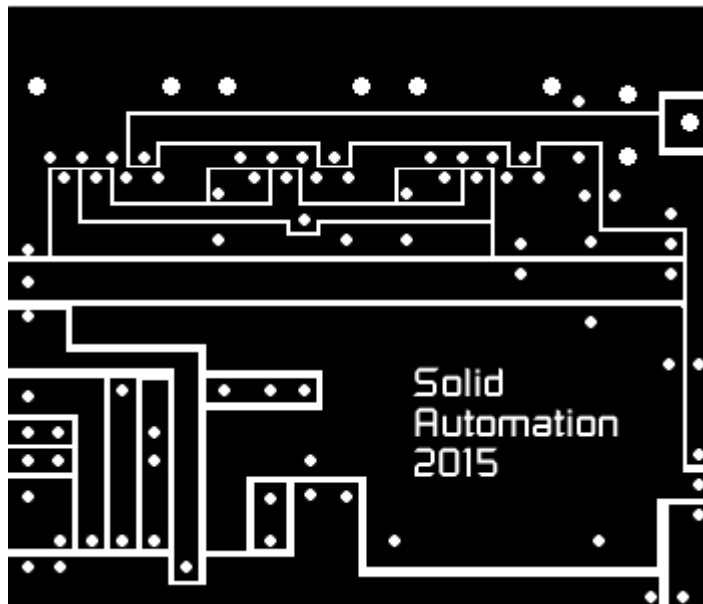
Raspberryn GPIO väylään liitettiin myös kaksi led-diodia merkkivaloksi ilmaisemaan laitteen tilaa. Tämä mahdollistaa laitteen vikatilän selvittämisen ja toiminnallisuuden kuvaamisen ilman ulkoista näyttöä. Vihreä merkkivalo näyttää verkkoyhteyden tilan ja punainen merkkivalo milloin servomoottoria säädetään ja mihin suuntaan. Taulukossa 2 on kuvattu merkkivalojen eri toiminnot ja niitä vastaavat laitteen tilat.

Taulukko 2. Merkkivalojen toiminta eri laitteen tiloissa

Laitteen tila	Merkkivalojen toiminta
laite käynnistyy	molemmat merkkivalot palavat
laite käynnistynyt	molemmat merkkivalot sammuvat
laite kytkeytynyt pilvipalvelimeen	vihreä merkkivalo palaa
venttiiliä säädetään kiinni	punainen merkkivalo vilkkuu nopeasti
venttiiliä säädetään auki	punainen merkkivalo palaa hetken yhtäjaksoisesti
verkkoyhteys ei toimi	punainen merkkivalo vilkkuu tai palaa, mutta vihreä merkkivalo ei pala
tiedonsiirto käynnissä	vihreä merkkivalo vilkkuu

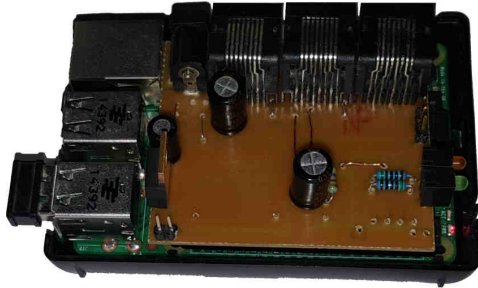
3.2.4 Piirilevy

Eri komponenttien liittämiseksi Raspberryn GPIO-väylään, päätettiin syövyttää piirilevy, johon on helppo kytkeä kaikki toiminnallisuus (ks. kuvio 29). Lämpötilantureita varten piirilevylle lisättiin RJ45-liittimet ja servomoottoria varten kolme pinninen liitin, johon servomoottorin oma liitin käy suoraan. Moottoria varten tuotiin myös lisävirtaliitäntä piirilevylle. Merkkivalot saatiin myös hyvin sijoitettua piirilevylle.



Kuvio 29. Piirilevy

Kuviossa 30 nähdään miten komponentit sijoittuvat tehdylle piirilevyille. Piirilevyn asennus Raspberryyn onnistuu painamalla liittimen GPIO-väylään.



Kuvio 30. Piirilevy kalustettuna ja liitettynä Raspberryyn

3.2.5 GPIO-väylä

Koska Raspberry Pi 2:n GPIO-väylässä on 40 pinniä, väylää varten piirilevyyn juotettiin 40 pinniä sisältävä liitin. Liittimestä poistettiin käyttämättömien pinnien vastakohtat ja juotettiin ainoastaan käytössä olevat kohdat. Näin saatiin piirilevy toteutettua hieman yksinkertaisemmin. Taulukossa 3 on kuvattu käytetyt pinnit ja niiden käyttötarkoitus.

Taulukko 3. Projektissa käytetyt GPIO-väylän pinnit ja niiden käyttötarkoitus

pinni	nimi	käyttö
2	5v	1-wire käyttöjännite
3	GPIO 2	vihreä led ohjaus
4	5v	1-wire käyttöjännite
5	GPIO 3	punainen led ohjaus
7	GPIO4	1-wire dataväylä
9	ground	maadoitus
12	GPIO 18	PWM - servomoottorin ohjaus
17	3,3v	1-wire dataväylän jännitteen syöttö
25	ground	maadoitus
39	ground	maadoitus

3.2.6 Mittaus- ja säätöyksikön ohjelmisto

Raspberryn GPIO-väylää voidaan ohjelmoida suoraan komentoriviltä, c- kielellä tai Pythonilla. Laitteen ohjelmointikieleksi valittiin Python. 1-Wire väylää varten on oma kerneli tason ajuri ja se tallentaa datan suoraan muistikortille `/sys/bus/w1/devices/` kansioon. Laitteen ohjelmisto hyödyntää tätä ja lukee lämpötila-arvot suoraan tiedostoista.

Servomoottorin ohjausta varten GPIO-väylän pinni asetettiin PWM (pulssinleveysmodulaatio) tilaan (`GPIO.PWM(12,50)`). Taajuudeksi valittiin 50Hz, jolloin saatiin servomoottorin käyttöalueeksi 0,2 - 14,8. Käyttöalueen arvoilla voidaan moottori kääntää ääriasennosta toiseen. Ohjelmiston käynnistyessä servomoottori käännetään puoleen väliin (`ChangeDutyCycle(7.5)`). Tämä takaa sen, että jos laite ei saa säätöarvoa, niin venttiili ei ole kokonaan kiinni. Venttiilin säätöä varten saadaan pilvipalvelimelta laskennallinen venttiilin lämpöarvo. Tätä arvoa verrataan venttiilin lämpötilanturin arvoon. Jos laskennallinen arvo on isompi kuin lämpötilanturin arvo, niin venttiiliä avataan servomoottorilla ja jos pienempi venttiiliä säädetään kiinni.

Merkkivalojen ohjaus toteutettiin yksinkertaisesti muuttamalla GPIO portin tilaa 1 ja 0 välillä riippuen miten ohjelma etenee ja saako vastauksen esim. pilvipalvelimelta.

Ohjelma välittää lämpötilantureiden arvon pilvipalvelimelle ja saa samalla laskennallisen säätöarvon. Lämpötila-arvoista ja salaisesta avaimesta lasketaan ohjelmallisesti tarkistesumma. Summa luodaan md5-algoritmilla. Lämpötila-arvot ja tieto miltä anturilta arvo on, sekä tarkistesumma välitetään JSON muodossa pilvipalvelimelle. Palvelin palauttaa JSON-vastauksen, joka sisältää laskennallisen arvon ja tarkistesumman. Ohjelma laskee laskennallisesta arvosta ja salaisesta avaimesta tarkistesumman, mitä verrataan palvelimelta saatuun. Jos tarkistesummat ovat samat, niin arvo hyväksytään järjestelmään. Mittaus- ja säätöyksikön ohjelmisto kokonaisuutena liitteessä 1.

3.2.7 Mittaus- ja säätöyksikön käyttöönotto

Laitteen käyttöönotto on tehty mahdollisimman helpoksi. Suunnittelun lähtökohtana oli, ettei mitään asetuksia tarvitse syöttää mittaus- ja säätöyksikköön. Laitteen kytkemiseen riittääkin oheislaitteiden liittäminen liittimillä, ja virran ja verkkokaapelin

kytkeminen. Laite hakee verkosta DHCP-(Dynamic Host Configuration Protocol) palvelusta osoitetiedot ja käyttöönotto on valmis. Laitteen sijoittaminen laajakaistareitittimen NAT- palomuurin taakse on mahdollinen. Toimii myös mobiililaajakaistareitittimen kytkettynä.

3.3 Pilvipalvelu

Pilvipalvelu toteutetaan virtuaalipalvelimella LAMP-kokoonpanolla. Linux käyttöjärjestelmänä on Centos 6 -käyttöjärjestelmä laajennettuna Parallels Plesk 12 -sovelluksella, joka helpottaa palvelimen ylläpitoa tarjoamalla mm. mobiilisovelluksen palvelimen hallintaa ja seuranta varten.

3.3.1 Tietokanta, tietokannan käyttö ja hallinta

Tietokanta on toteutettu avoimeen lähdekoodiin pohjautuvalla MySQL- relaatiotietokanta ohjelmistolla. Tietokantaan kirjoittaminen ja lukeminen toteutettiin PHP rajapinnan kautta. Tietokannan hallintaan käytössä oli myös phpMyAdmin, joka on selaimen kautta käytettävä tietokannan hallintatyökalu.

Tietokannan taulut suunniteltiin laajempaa käyttöä ajatellen. Rakenteesta löytyy mm. mahdollisuus liittää lämpötila-anturit eri alueisiin ja valita antureiden käyttökohte (ulkoanturi, huoneanturi tai venttiilianturi). Taulujen lisääminen jatkossa on helppoa ja tietokanta kestävä suurempaakin käyttöä. Esimerkki tietokantaan tallennetusta antureiden arvoista (ks. kuvio 31).

id	sensor	value	timestamp
281299	10-00080138f9be	4625	2015-12-06 15:17:01
281300	28-000006aad8b8	23937	2015-12-06 15:17:01
281301	28-000006aaf9d1	23750	2015-12-06 15:17:01
281296	10-00080138f9be	4562	2015-12-06 15:16:48
281297	28-000006aad8b8	23937	2015-12-06 15:16:48
281298	28-000006aaf9d1	23750	2015-12-06 15:16:48
281293	10-00080138f9be	4562	2015-12-06 15:16:34
281294	28-000006aad8b8	23937	2015-12-06 15:16:34
281295	28-000006aaf9d1	23687	2015-12-06 15:16:34
281290	10-00080138f9be	4562	2015-12-06 15:16:20
281291	28-000006aad8b8	23875	2015-12-06 15:16:20
281292	28-000006aaf9d1	23687	2015-12-06 15:16:20
281287	10-00080138f9be	4625	2015-12-06 15:16:06
281288	28-000006aad8b8	23875	2015-12-06 15:16:06
281289	28-000006aaf9d1	23750	2015-12-06 15:16:06
281284	10-00080138f9be	4625	2015-12-06 15:15:52

Kuvio 31. Esimerkki pilvipalvelun sensor_data-taulusta

3.3.2 Pilvipalvelun JSON -rajapinnat

Pilvipalvelin tarjoaa JSON rajapinnat tiedon siirtoon. Rajapinnan kautta palvelin saa tiedon eri antureiden lämpötila-arvot ja tallentaa ne tietokantaan. Mittaus- ja säätöyksikölle lasketaan venttiilin haluttu lämpötila-arvo ja se palautetaan säätöyksikölle. Mobiilisovellusta varten on myös oma JSON-rajapinta, joka tarjoaa lämpötila-arvot 10 h ajanjaksolta sekä reaaliaikaiset lämpötila-arvot. Kuviossa 32 esimerkki mittaus- ja säätöyksikölle lähetettävästä JSON-tietueesta.

```
{
  "data": [{"valveTemp": "23750", "valveSensorId": "28-000006aaf9d1",
    "groupId": 1, "countedValveTemp": -5045}],
  "secure": "195eedb35407c9557de68afc7413d83"}

```

Kuvio 32. Mittaus- ja säätöyksikön ohjaus JSON

3.3.3 Sääennusteet

Venttiilin säätöön käytetään lämpötila-arvojen lisäksi myös ennustetta tulevasta säästä. Ennusteen käyttö mahdollistaa tasaisemman lämmön pitämisen huoneistos-

sa. Ennusteen avulla pystytään reagoimaan pakkasen kiristymiseen lisäämällä lämmitystä, ennen kuin huoneiston lämpö alenee ja sään lämmitessä vähennetään lämmitystä.

Ennusteet haetaan Norjalaisesta verkkopalvelusta (yr.no). Palvelu tarjoaa XML-rajapinnan, mistä voidaan kysyä halutun paikan viiden vuorokauden sääennuste tunnin tarkkuudella. Ennusteet tallennetaan tietokantaan myöhempää käyttöä varten (ks. kuvio 33).

from	symbol_name	symbol_number	precipitation_value	precipitation_min	precipitation_max	wind_deg	wind_code	wind_name	windspeed_mps	windspeed_name	temperature	pressure
2015-12-08 17:00:00	Cloudy	4	0	0	0	261.3	W	West	4.9	Gentle breeze	-1	1026.1
2015-12-08 16:00:00	Partly cloudy	3	0	0	0	271.3	W	West	5.3	Gentle breeze	-1	1026.2
2015-12-08 15:00:00	Partly cloudy	3	0	0	0	273.9	W	West	5.7	Moderate breeze	-1	1026
2015-12-08 14:00:00	Fair	2	0	0	0	283.4	WNW	West-northwest	5.7	Moderate breeze	0	1025.6
2015-12-08 13:00:00	Clear sky	1	0	0	0	291.3	WNW	West-northwest	6	Moderate breeze	0	1024.8
2015-12-08 12:00:00	Clear sky	1	0	0	0	291.7	WNW	West-northwest	6.6	Moderate breeze	0	1024
2015-12-08 11:00:00	Clear sky	1	0	0	0	296.2	WNW	West-northwest	7.6	Moderate breeze	0	1022.9
2015-12-08 10:00:00	Clear sky	1	0	0	0	300.1	WNW	West-northwest	8.9	Fresh breeze	0	1022.1
2015-12-08 09:00:00	Clear sky	1	0	0	0	301.6	WNW	West-northwest	10.3	Fresh breeze	0	1021.1
2015-12-08 08:00:00	Fair	2	0	0	0	300.2	WNW	West-northwest	11.1	Strong breeze	1	1019.9
2015-12-08 07:00:00	Partly cloudy	3	0	0	0	296.5	WNW	West-northwest	11.7	Strong breeze	1	1018.8
2015-12-08 06:00:00	Cloudy	4	0	0	0	294.2	WNW	West-northwest	11.8	Strong breeze	1	1017.9
2015-12-08 05:00:00	Fair	2	0	0	0	291.2	WNW	West-northwest	11.9	Strong breeze	1	1017.3
2015-12-08 04:00:00	Clear sky	1	0	0	0	289.5	WNW	West-northwest	11.3	Strong breeze	0	1017.4
2015-12-08 03:00:00	Clear sky	1	0	0	0	290.8	WNW	West-northwest	10.9	Strong breeze	0	1017.4
2015-12-08 02:00:00	Clear sky	1	0	0	0	287.9	WNW	West-northwest	11	Strong breeze	0	1017.1
2015-12-08 01:00:00	Clear sky	1	0	0	0	290.1	WNW	West-northwest	11.8	Strong breeze	0	1016.6
2015-12-08 00:00:00	Clear sky	1	0	0	0	295.2	WNW	West-northwest	10.7	Fresh breeze	0	1016
2015-12-07 23:00:00	Clear sky	1	0	0	0	299.4	WNW	West-northwest	10.2	Fresh breeze	0	1015.5
2015-12-07 22:00:00	Clear sky	1	0	0	0	298.9	WNW	West-northwest	9.4	Fresh breeze	0	1014.4

Kuvio 33. Esimerkki pilvipalvelun forecast-tilusta

3.3.4 Venttiilin ohjauslämpötilan laskeminen

Palvelimen palauttama venttiilin haluttu lämpötila lasketaan kuvion 34 mukaisella kaavalla. Julkaistu kaava ei sisällä korjauskertoimia, mitkä vaikuttavat lopputulokseen. Korjauskertoimia tulee mm. kohteen tiiveyden mukaan, energiatodistuksen mukaan, tuulen suunnan ja voimakkuuden mukaan. Laitteiston ollessa pidempään käytössä saadaan mukaan myös laitteiston oppimat kertoimet.

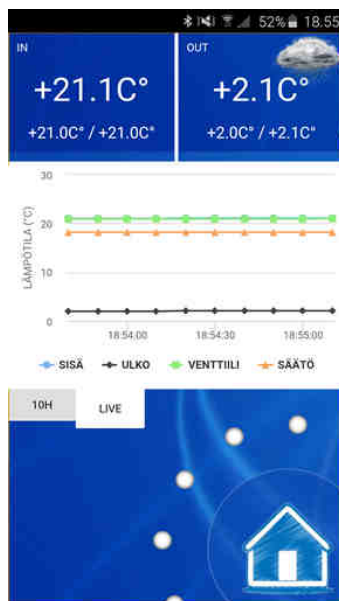
$$y = x - 10 - (y - z) - 1/6 * \frac{k1 + k2}{2}$$

y = huoneen haluttu lämpötila
 x = venttiilin lämpötila
 z = huoneen lämpötila
 k1 = ulkolämpötila
 k2 = ennusteen alin lämpötila 4h sisällä

Kuvio 34. Venttiilin laskennallisen lämpötilan laskentakaava

3.4 Mobiilisovellus

Järjestelmän hallintaa varten tehtiin mobiilisovellus, millä pystyy säätämään huoneiston haluttua lämpötilaa. Sovellus näyttää myös 10 h ja 2 min kaaviot eri antureiden lämpötiloista. Kaavioissa näkyy myös säätöarvo, mitä pystytään muuttamaan sovelluksen säädinelementillä. Sovellus näyttää myös sisä- ja ulkolämpötilan ja 24 h ajanjaksolta minimi ja maksimi arvot (ks. kuvio35).



Kuvio 35. Mobiilisovellus

3.4.1 Cordova

Sovellus toteutettiin Cordova -kehitysympäristöllä. Se tarjoaa työkalut sovelluksen toteuttamiseen HTML5, CSS ja JavaScript ohjelmointia käyttäen. Ohjelman lähdekoodi on käännettävissä Android, ios ja windows phone laitteille. Cordova on avoimeen lähdekoodiin pojautuva, joten sen käyttö ei ole maksullinen. Kustannuksia aiheuttaa lähinnä tarve käyttää ulkoisia palveluita, mitä kautta ohjelmiston saa jaettua asiakkaille. Projektissa toteutetun sovelluksen jakelua ei ollut tarvetta toteuttaa vielä tässä vaiheessa, mutta julkaisumahdollisuuksia tutkittiin valmiiksi mahdollista tarvetta varten. Sovelluksen prototyypin alustaksi valittiin Android käyttöjärjestelmä, jolloin sovellus saatiin helposti siirrettyä puhelimeen USB-kaapelin avulla ilman ulkoista palvelua. Ohjelman kääntäminen vaatii vain Androidin SDK -paketin asentamisen tietokoneeseen ja puhelimen aktivoimisen kehitystilaan. Tämän jälkeen Cordova pystyy kääntämään sovelluksen suoraan puhelimeen.

3.4.2 Sovelluksen toiminta

Sovellusta varten käyttöön otettiin kaksi JavaScript- kirjastoa. HighCharts -kirjasto tarjoaa helposti ohjelmoitavat graafiset kaaviot. Kirjasto sisältää itsessään monta erityylistä kaaviota. Tyylejä on mahdollista myös muokata projektikohtaisesti. Kaaviot voidaan toteuttaa interaktiivisina, jolloin voidaan valita mitä arvoja kaaviossa näytetään. Sovelluksen kaavioista voidaan valita mitä lämpötila-anturia seurataan. HighCharts on yritys käytössä maksullinen, mutta demoympäristössä sitä voi käyttää vapaasti. Maksullisuus on otettava huomioon kaupallistumisen myötä.

jQuery tarjoaa DOM- (Document Object Model) elementtien, AJAX-kyselyjen ja JSON-tietueiden käsittelyyn hyvän avoimen lähdekoodin kirjaston. jQueryn syntaksi on helposti ymmärrettävä ja vähentää perinteisen JavaScriptin kirjoittamista. Siksi sitä käytetäänkin monissa vastaavissa projekteissa.

Sovellus hakee JSON rajapinnan kautta pilvipalvelimelta lämpötila-arvot. Tiedon oikeellisuus tarkistetaan vertaamalla palvelimen laskemaa tarkistesummaa ja sovelluksen laskemaa tarkistesummaa keskenään. Tarkistesumman laskennassa käytetään salausavainta, mitä ei siirretä palvelimen ja sovelluksen välillä. Hakua tehtäessä sovellus lähettää samalla säätöarvon palvelimelle.

4 Verkkoliikenteen tietoturva

Pilvipalvelin on ainoa kokonaisuus, minkä täytyy olla julkisessa verkossa. Mittaus- ja säätöyksikkö, sekä mobiilisovellus voivat olla palomuurin takana. Näin saadaan suojattua suurin osa järjestelmää hyökkäyksiä vastaan. Samalla saadaan helpotettua mittaus- ja säätöyksikön asentamista, kun ei ole tarvetta konfiguroida kiinteää IP-osoitetta. Mittaus- ja säätöyksikön ainoa vaatimus verkolle, että verkko tarjoaa DHCP- palvelun ja jaettavalla osoitteella on pääsy Internetiin. Tämä mahdollistaa mittaus- ja säätöyksikön suojaamisen esim. NAT -palomuurilla.

Mobiilisovellus toimii mittaus- ja säätöyksikön tavoin. Sovellus avaa yhteyden pilvipalvelimelle ja tarvittaessa tallentaa mm. säätöarvon palvelimelle ja saa samalla palvelimelta mm. lämpötila-arvot. Sovellus voi olla NAT-palomuurin takana ja suojassa hyökkäyksiltä. Mittaus- ja säätöyksikkö ja mobiilisovellus ottavat yhteyden pilvipalvelimen JSON-rajapintaan.

4.1 Anti-DDOS

Palvelunestohyökkäyksellä (DDOS, Distributed Denial of Service) halutaan lamauttaa hyökkättävän kohteen palvelut kokonaan. Pilvipalvelussa käytössä oleva Anti-DDOS-palvelu on suunniteltu tunnistamaan hyökkäykset ja estämään ne. Verkkoliikennettä analysoidaan koko ajan ja poikkeava liikenne pystytään tunnistamaan ja suorittamaan automaattinen mitigaatio.

DDOS-hyökkäyksiä vastaan käytettynä mitigaatio merkitsee oikeudettoman liikenteen suodattamisen ja imuroinnin VACin kautta, samalla päästään oikeiden pakettien jatkamaan kulkuaan. VAC (Valve Anti-Cheat System) koostuu useista laitteista, jotka kukin hoitavat tietyn osa-alueen suojatakseen palveluita tietyltä tai usealta erilaiselta hyökkäykseltä kuten DDOS ja Flood.

4.2 JSON-rajapinnat

Pilvipalvelin tarjoaa kaksi erillistä SLL-salattua JSON-rajapintaa. Mittaus- ja säätöyksikkö avaa yhteyden pilvipalvelimeen ja lähettää keräämänsä arvot ja tarkistesum-

man. Arvojen pohjalta laskettava tarkistesumma sisältää myös salaisen avaimen, joten suoraan lähetetyistä arvoista tarkistesummaa ei voi laskea.

5 Jatkokehitys

Ohjelmiston suunnittelussa oli jo ajatuksia jatkokehityksestä. Tietokantaan tehtiin valmiiksi määrittämiä, joilla voidaan määrittellä huoneistoon eri alueita. Alue muodostuu lämmitettävien alueiden mukaan. Pienemmissä huoneissa on yleensä yksi alue, mutta isommissa huoneissa voi olla useampikin lämmityspiiri. Alue voidaan nimetä haluksi ja alueeseen voidaan liittää tarvittavat lämpötila-anturit. Lähtökohta on, että alueeseen kuuluu vähintään huoneistoanturi, kyseisen piirin venttiilianturi ja ulkolämpötila-anturi. Sama huoneistoanturi voidaan liittää useampaankin alueeseen, mutta suotavaa olisi, että jokaisessa huoneessa olisi ainakin yksi lämpötila-anturi. Ulkoantureita ei tarvitse olla kuin yksi, joka liitetään kaikkiin alueisiin. Lisäksi voidaan käyttää lattian lämpötilaa seuraavaa anturia. Tällöin voidaan lattialle määrittää mukavuuslämpötila, joka voidaan ottaa huomioon kokonaislämmityksessä.

5.1 Pilvipalvelun kehitykset

Palvelua täytyisi vielä laajentaa asiakaskohtaisilla tiedoilla. Tällöin palveluun voitaisiin kytkeä useampikin huoneisto. Asiakastietoihin tulisi määrittellä asiakaskohtainen salausavain, jolloin tietojen oikeellisuus olisi taattu. Sääennusteiden hakeminen vaatisi myös lisää suunnittelua. Kehitysympäristössä on käytössä kiinteään paikkatietoon perustuvaa ennustetietoa, mutta asiakasmäärän lisääntyessä myös ennusteiden hakeminen eri maantieteellisiin kohteisiin tulisi automatisoida. Paikkatiedon tallentamisen voisi lisätä mobiilisovellukseen ja automaattisesti voitaisiinkin tarjota mobiililaitteen sijaintiin perustuvaa paikkatietoa.

5.1.1 Asiakastilin luominen

Palvelun aloittaminen tulisi alkaa uuden asiakastilin luomisella. Kun mobiililaitteeseen ladataan sovellus, tulisi rekisteröitymisen olla pakollinen. Tilin perustamisen yhteydessä luotaisiin satunnaisgeneraattorin ja aikaleimaan perustuva yksilöllinen salausavain. Salausavain voitaisiin määrittää tietokantaan uniikki avaimeksi, jolloin

kahta samanlaista avainta ei voi tallentaa tietokantaan. Tällöin avain olisi varmasti yksilöllinen. Salausavainta voitaisiin rekisteröitymisen jälkeen käyttää tarkistesumman laskemisessa.

5.1.2 Mittaus ja säätöyksikön rekisteröiminen palveluun

Mittaus ja säätöyksikön asennuksen yhteydessä laite liitetään palveluun laitteen yksilöllisellä tunniste numerolla. Liittäminen tapahtuisi mobiilisovelluksella, jolloin antamalla tunnistenumero mittaus- ja säätöyksikkö voitaisiin liittää asiakkaan tiliin ja samalla välittää asiakkaan oma salainen avain mittaus- ja säätöyksikölle. Omistajuuden vaihtuessa, laite tulisi vapauttaa vanhan omistajan toimesta ja tämän jälkeen uusi omistaja voi rekisteröidä laitteen omaan tiliinsä.

Mittaus- ja säätöyksiköitä voi olla useampikin asiakasta kohden. Mittaus- ja säätöyksiköt lähettää kaikkien antureiden arvon ja anturin yksilöllisen ID:n palvelimelle. Asiakas pystyy näkemään kaikki rekisteröimänsä yksiköiden antureiden tiedot ja muodostamaan haluamansa alueet ja liittämään anturit haluamaansa alueeseen. Antureille määritellään niiden paikkatieto (ulkoanturi, huoneanturi, venttiilianturi tai lattia-anturi) ja samalla voidaan anturit nimetä helpottamaan tunnistamista myöhemmin. Yksi ulkolämpötila-anturi voi toimia useammankin yksikön tiedonlähteenä. Määritettyä aluetta kohden lasketaan laskennallinen venttiilin lämpötila, joka välitetään sille säätöyksikölle, missä on kyseisen alueen venttiilianturi.

Omistajan vaihdon yhteydessä säätöyksikön antureiden tiedot säilyvät, mutta on tyhjennettävissä helposti. Tällöin ei huoneiston oston yhteydessä tarvitse määritellä säätöyksiköitä, eikä antureita uudelleen.

5.2 Mittaus ja säätöyksikön kehitys

Säätöyksikön kehitysajatuksia nousi piirilevyn valmistamisen myötä. Laitteen kasaa- mista alumiinikoteloon suunniteltiin ja testattiin osittain jo projektin yhteydessä. Alumiinikotelossa saataisiin mm. jänniteregulaattoreiden lämpö siirrettyä kotelon runkoon. Testatussa kotelossa (ks. kuvio36.) oli reunoilla urat, joihin piirilevyn voi mitoittaa ja erillistä muuta kiinnitystä ei tarvita. Alumiinikotelo antaisi myös hyvän suojan laitteistolle teknisessä tilassa.



Kuvio 36. Suunniteltu laitekotelo (30,5 mm x 103 mm x 160 mm) (Hammond)

5.2.1 1-Wire-verkon kehittäminen ja langattomuus

Prototyypissä 1-Wire toteutettiin kytkemällä data väylä suoraan GPIO-väylään. Uudessa versiossa voisi käyttää välissä Maxim DS2482-800 1-Wire-master-piiriä. Tällä saataisiin laajennettua 1-Wire-laitteiden määrää ja jokainen säätöyksikön 1-Wire-portti olisi oma 1-Wire-väylä. Tällöin vikasietoisuus paranisi.

Servomoottoreiden kytkentää voisi kehittää siten, että ne olisi liitettävissä myös 1-Wire-verkkoon. Tällöin riittäisi yksinkertainen kaapelointi ja kaikki venttiilit saataisiin ketjutettua peräkkäin. Servomoottoreiden tarvitsema lisävirta voidaan syöttää samassa kaapelissa, joten erillisiä virtalähteitäkään ei tarvita. Ongelmana on ettei 1-Wire-tarjoa suoraan komponenttia kytkentään vaan rajapinta täytyy suunnitella erikseen. Lisäksi voisi kartoittaa, miten voitaisiin kytkeä esimerkiksi huoneistoanturit langattomasti. Langattomissa yksiköissä voisi olla myös 1-Wire-liitäntä, jolloin voisi siirtää tarvittaessa useammankin anturin tiedot langattoman linkin yli.

5.2.2 Toiminnallisuuden laajentaminen

1-Wire-verkko tarjoaa helposti laajennettavan verkon ilman lisä kaapelointia mittaus- ja säätöyksiköltä asti. Riittää kun ketjuttaa lähimmän 1-Wire-laitteen kanssa. Verkkoa laajentamalla muihin antureihin, se antaa mahdollisuuden seurata mahdollisia vesivahinko kohteita. Astianpesukoneen, pyykinpesukoneen ja varaajan alle sijoitettavalla kosteusanturilla saadaan helposti mitattua kosteuden muutos ja ilmoitettua mobiililaitteeseen mahdollinen vesivahinkovaara. Ovien ja ikkunoiden magneettikytkimien liittäminen järjestelmään takaa, ettei kukaan käy salaa huoneistossa.

5.2.3 Vikasietoisuuden parantaminen

Mittaus- ja säätöyksikköä voisi parantaa myös ohjelmallisesti. Yksiköllä voisi olla tallennettuna enemmän tietoa venttiilin säätöön liittyen. Tällöin verkkoyhteyden katkeaminen ei vaikuttaisi perussäädön toimintaan. Vikasietoisuutta voisi lisätä myös tallentamalla antureiden tiedon paikallisesti, mistä siirto tehtäisiin yhteyden ollessa toiminnassa.

1-Wire-antureiden tunnistus tehdään prototyypissä laitteen käynnistyksen yhteydessä. Tunnistus pitäisi tehdä joka kerta, kun käsitellään verkkoa. Tällöin anturin vaurioituminen ei aiheuttaisi ongelmaa kuin ainoastaan kyseisen alueen säätöön ja 1-Wire-laitteita voisi kytkeä ilman järjestelmän uudelleen käynnistämistä.

Useamman säätöyksikön liittäminen yhteen huoneiston tai kiinteistön sisäisesti takaisi tiedon siirtymisen ilman ulkoista linkkiä. Tämä ominaisuuden tärkeys korostuu mm. uuden mahdollisen varashälytintoiminnallisuuden myötä.

5.3 Mobiilisovelluksen kehitysideat

Mittaus- ja säätöyksikön laajennuksen ja pilvipalvelun monipuolistumisen myötä vaatimukset mobiilisovelluksellekin muuttuu. Osa muutoksista on järkevää tuoda mobiilisovellukseen, vaikka muut järjestelmät eivät laajenisikaan.

5.3.1 Asiakkuuden luominen

Mobiilisovelluksen asennuksen jälkeen tulisi avautua kirjautumisikkuna, jonka kautta olisi mahdollista luoda myös uusi asiakkuus. Asiakkuuden yhteydessä kysyttäisiin kohteen osoitetiedot ja laskutukseen liittyvät yhteystiedot. Rekisteröitymisen yhteydessä palvelin luo uuden salausavaimen, jonka mobiilisovellus tallentaa laitteeseen.

5.3.2 Mittaus- ja säätöyksikön rekisteröiminen ja hallinta

Sovelluksen rekisteröimisen jälkeen avautuisi mittaus- ja säätöyksikön rekisteröintiin liittyvät lomakkeet. Rekisteröintiin käytetään laitteen yksilöllistä tunnistenumeroa. Tunnistenumero varmentaa, ettei väärää laitetta rekisteröidä. Rekisteröinnin jälkeen olisi mahdollista luoda alueet ja sijoittaa lämpötila-anturit oikeaan alueeseen. Omis-

tajuuden vaihtuessa sovelluksen kautta pitäisi pystyä vapauttamaan mittaus- ja säätöyksikkö uudelleen rekisteröitäväksi.

5.3.3 Sääennusteen näyttäminen

Pilvipalvelin kerää tietoa kohteen sääennusteesta. Tätä tietoa kannattasi käyttää myös mobiilisovelluksessa. Lämpötilan lisäksi ennusteista tallennetaan myös sademäärät, tuulen nopeus ja suunta, ilmanpaine ja pilvisyyden tila. Ennusteista pystyy rakentamaan sääennusteen kolme vuorokautta eteenpäin tunnin aikavälillä.

5.3.4 Teknisiä muutoksia

Mobiilisovelluksen pohjaututtua HTML5 ja JavaScript ohjelmointiin, lisää suorituskykyä saataisiin JavaScript React kirjaston käyttöönnotolla. Kirjasto tuo näytettävän sisällön päivittämiseen tehokkuutta. Ideana on päivittää ainoastaan näytön se alue missä muutosta tarvitaan ei koko näyttöä kuten tavallisemmin tehdyissä sovelluksissa on tapana.

Näyttöön ilmestyvien ilmoitusten (push notification) tuominen osaksi sovellusta tulisi toteuttaa ensimmäisien kehityksien joukossa. Ilmoituksilla saadaan tuotua virhetilanteen tehokkaammin käyttäjän tietoisuuteen. Tästä olisi hyötyä pelkän lämmityksen säädöissäkin esim. anturivian ilmoittamisessa. Uusien ominaisuuksien esim. kosteusmittauksen mukaan tulo lisäisi toiminnallisuuden tarvetta.

Ominaisuuksien kasvaessa myös tarve hallinnoida ja näyttää eri sisältöä laajenee. Prototyyppi sovelluksessa ei ole toteutettuna valikkoa lainkaan, mutta säätöpyörä on muutettavissa toimimaan valikon tavoin. Säätöpyörässä olevien pallojen tilalle voidaan laittaa ikonit ja linkitys toiseen toimintoon.

5.3.5 Uusien ominaisuuksien tuomat muutokset

Kosteuden mittauksen tai varashälytínjärjestelmän myötä mobiilisovelluksen täytyy laajentua myös näiden osalta. Miten ja mitä tietoa uudet ominaisuudet tuovat täytyy selvittää, kun laajennukset on ensin toteutettu mittaus- ja säätöyksikköön.

6 Lopputulos

Opinnäytetyön aihe syntyi todellisesta tarpeesta kehittää pieni ja kustannustehokas vesikiertoisen lattialämmityksen ohjainyksikkö, pilvipalvelu ja hallintasovellus mobiiliympäristöön.

Opinnäytetyössä tutkittiin 1-Wire-väylän soveltuvuutta servomootorin dataväyläksi. 1-Wire-väylän etuna olisi ollut mm. servomootorin käyttövirran saaminen samasta kaapeloinnista. 1-Wire-väylää käytettiin lämpötila-antureiden tiedonsiirtoon. Servomootorin ohjaus päädyttiin tekemään suoraan Raspberry Pi GPIO-väylän kautta. Servomootorin ohjausta varten väylään otettiin käyttöön pulssinleveysmodulaatio. Opinnäytetyössä tutkittiin myös miten ulkolämpötilan, huoneiston lämpötilan ja sääennusteen avulla voidaan toteuttaa ennakoiva lämmitysjärjestelmä. Ennustetta käytettiin lämpötilan säädön ennakointiin takaamaan tasaisempi lämpö huoneistossa.

Opinnäytetyön yksi osa oli toteuttaa pilvipalvelu (SaaS) JSON rajapinnoilla. Pilvipalvelun tehtävänä oli tallentaa lämpötila-antureiden arvot, huolehtia sääennusteiden automaattisesta hakemisesta ja mittaus- ja säätöyksikön ohjaus arvojen laskemisesta. Sääennusteet haettiin norjalaisesta sääpalvelusta (yr.no) XML-tietueina ja tallennettiin tietokantaan käytettäväksi myöhemmin ohjausarvon laskennassa.

Mittaus- ja säätöyksikkö toteutettiin Raspberry PI -alustalla ja toteutetulla laajennuskortilla. Korttiin oli mahdollista liittää lämpötila-anturit ja servomoottori. Kortissa oli myös kaksi led-merkkivaloa ilmaisemassa laitteen toiminnallisen tilan. Mittaus- ja säätöyksikkö huolehti tiedonsiirrosta salatun ja varmennetun yhteyden yli pilvipalveluun.

Mobiilisovellus toteutettiin Cordova -sovelluskehitysympäristöllä. Sovellus käännettiin suoraan USB-liitännän kautta Android -puhelimeen. Toteutuksessa käytettiin HTML5 ja JavaScript tekniikoita. Sovellus näytti lämpötila-antureiden lämpötilat ja huoneiston säätöarvon viivakaaviona. Huoneiston lämpötilan säätö onnistui sovelluksen säätöpyörällä. Sovelluksen tehtävänä oli myös huolehtia tiedonsiirrosta pilvipalvelusta salatun ja varmennetun yhteyden yli.

Opinnäytetyössä otettiin huomioon toteutetun järjestelmän tietoturva. Työssä käsiteltiin pilvipalvelun suojausta palvelunestohyökkäykseltä. Pilvipalvelun ja mobiiliso-

velluksen, sekä mittaus- ja säätöyksikön tiedonsiirto tehtiin salatun yhteyden (SSL) yli ja siirtyvä data varmennettiin käyttäen laskettuja varmenneavaimia. Varmenteen laskemiseen käytettiin salausavainta, mitä ei laitteiden kesken siirretty.

7 Yhteenveto

Opinnäytetyön aihe oli mukaansa tempaava ja uusia ideoita järjestelmän kehittämiseksi syntyi paljon. Opinnäytetyön toteuttaminen palkkatyön ohessa ja perheellisenä antoi haasteita ajan käytön suhteen, mutta mielenkiintoinen aihe oli hyvä motivaattori projektille.

Mobiilisovelluksen tekeminen tuli itselle täysin uutena, mutta HTML5 ja JavaScriptit kun oli jo tuttua, niin ilman isompia ongelmia tästäkin selvittiin.

Prototyyppi tullaan ottamaan käyttöön ja toiveissa olisi, että jatkokehitystä päästään vielä tekemään.

Lähteet

1-Wire Tutorial. N.d. Viitattu 29.11.2015.

<https://www.maximintegrated.com/en/products/1-Wire/flash/overview/index.cfm>

Adaptors. N.d. Viitattu 29.11.2015.

<https://www.hobby-boards.com/store/pages.php?pageid=7>

Amazon WEB Service. N.d. Viitattu 6.12.2015. <https://aws.amazon.com>

Apache. N.d. Viitattu 6.12.2015.

https://fi.wikipedia.org/wiki/Apache_%28palvelinohjelma%29

Connect Multiple Temperature Sensors With Raspberry Pi. N.d. Viitattu 29.11.2015.

<http://www.reuk.co.uk/Connect-Multiple-Temperature-Sensors-with-Raspberry-Pi.htm>

GPIO Pinout Diagram. N.d. Viitattu 1.12.2015.

http://www.jameco.com/Jameco/workshop/circuitnotes/raspberry_pi_circuit_note.html

GPIO: Raspberry Pi Models A and B. N.d. Viitattu 29.11.2015.

<https://www.raspberrypi.org/documentation/usage/gpio/>

Hammond 1455L1601 Enclosure. N.d. Viitattu 2.12.2015.

<http://fi.farnell.com/hammond/1455l1601/enclosure-aluminium-end-plate/dp/4272882>

Hyödynnä pilvipalvelua viisaasti. N.d. Viitattu 6.12.2015.

<https://technet.microsoft.com/fi-fi/ff934854.aspx>

JSON Tutorial. N.d. Viitattu 6.12.2015. <http://www.w3schools.com/json/default.asp>

LAMP. N.d. Viitattu 6.12.2015. <https://www.linux.fi/wiki/LAMP>

Linnake, T. 2013. Pieni suuri Raspberry Pi tukkurijakeluun Suomessa. Viitattu

22.11.2015. <http://www.digitoday.fi/vimpaimet/2013/02/05/pieni-suuri-raspberry-pi-tukkurijakeluun-suomessa/20131979/66>

Linux. N.d. Viitattu 6.12.2015. <https://fi.wikipedia.org/wiki/Linux>

Linux-järjestelmän rakenne. N.d. Viitattu 6.12.2015. https://www.linux.fi/wiki/Linux-j%C3%A4rjestelm%C3%A4n_rakenne

MySQL Workbench. N.d. Viitattu 6.12.2015.

<http://dev.mysql.com/downloads/workbench/>

Network Topology. N.d. Viitattu 29.11.2015. <https://www.hobby-boards.com/store/pages.php?pageid=5>

November 2015 Web Server Survey. N.d. Viitattu 6.12.2015.

<http://news.netcraft.com/archives/category/web-server-survey/>

PHPMyAdmin. N.d. Viitattu 6.12.2015. <https://www.phpmyadmin.net/about/>

PHP Tutorial. N.d. Viitattu 6.12.2015. <http://www.w3schools.com/php/default.asp>

Pilvipalveluiden turvallisuus organisaatioiden näkökulmasta. N.d. Viitattu 6.12.2015.
https://www.viestintavirasto.fi/attachments/tietoturva/Pilvipalveluiden_tietoturva_organisaatioille.pdf

Raspberry Pi becomes best selling British computer. N.d. Viitattu 29.11.2015.
<http://www.theguardian.com/technology/2015/feb/18/raspberry-pi-becomes-best-selling-british-computer>

Raspberry Pi Learning Resources. N.d. Viitattu 29.11.2015.
<https://www.raspberrypi.org/learning/grandpa-scarer/worksheet/>

Servo Control with the Raspberry Pi. N.d. Viitattu 29.11.2015.
<https://www.youtube.com/watch?v=ddIDgUymbxc>

Top 10 Reasons to Choose MySQL for Next Generation Web Applications. N.D. Viitattu 6.12.2015. <http://www.mysql.com/why-mysql/white-papers/top-10-reasons-to-choose-mysql-for-next-generation-web-applications/>

Wiring Diagram. N.d. Viitattu 29.11.2015
<https://www.hobby-boards.com/store/pages.php?pageid=11>

What is a raspberry pi. N.d. Viitattu 21.11.2015.
<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>

XML Tutorial. N.d. Viitattu 6.12.2015. <http://www.w3schools.com/xml/default.asp>

Liitteet

Liite 1. Mittaus- ja säätöyksikön Python-ohjelmistokoodi

```
import json
import urllib
import urllib2
import hashlib
import os
import glob
import time
import RPi.GPIO as GPIO
from urllib2 import HTTPError

secureKey = "sdjhasert45676sdfsdfd712kjlaksjd9";
device_id = "demo1234"
url = 'https://automation.solidautomation.fi'

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

GPIO.setup(3,GPIO.OUT)
GPIO.setup(5,GPIO.OUT)
GPIO.setup(12,GPIO.OUT)

GPIO.output(3, 0)
GPIO.output(5, 0)

valve_position = 7.5

p = GPIO.PWM(12,50)
p.start(valve_position)

os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

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

def isEmpty(s):
    return not bool(s and s.strip())
```

```

def read_temp_raw(dFile):
    f = open(dFile, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp(dFile):
    lines = read_temp_raw(dFile)
    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 = int(temp_string) / 1000.0

    return temp_c

def chkSecure(resp_dict):
    chk_data =
    '{"valveTemp":'+str(resp_dict['data'][0]['valveTemp'])+'", "valveSensorId":'+str(resp
    _dict['data'][0]['valveSensorId'])+'", "groupId":'+str(resp_dict['data'][0]['groupId'])+', "
    countedValveTemp":'+str(resp_dict['data'][0]['countedValveTemp'])+'}'
    chk_secure = hashlib.md5(chk_data+secureKey).hexdigest()
    if resp_dict['secure'] == chk_secure:
        return True
    else:
        return False

def setValve(values):
    global valve_position
    if int(values[0]['valveTemp']) > int(values[0]['countedValveTemp']) and
    (int(values[0]['valveTemp']) - int(values[0]['countedValveTemp'])) > 500:

        if valve_position > 0.4 :
            valve_position = valve_position - 0.2
            p.ChangeDutyCycle(valve_position)

        print "---- "+str(valve_position)
        GPIO.output(5, 1)
        time.sleep(0.05)
        GPIO.output(5, 0)
        time.sleep(0.05)
        GPIO.output(5, 1)
        time.sleep(0.1)

```

```

        GPIO.output(5, 0)
        time.sleep(0.05)
        GPIO.output(5, 1)
        time.sleep(0.05)
        GPIO.output(5, 0)

        elif int(values[0]['countedValveTemp']) > int(values[0]['valveTemp'])
and (int(values[0]['countedValveTemp']) - int(values[0]['valveTemp'])) > 500:

            if valve_position < 14.8 :
                valve_position = valve_position + 0.2
                p.ChangeDutyCycle(valve_position)

            print "++++ "+str(valve_position)
            GPIO.output(5, 1)
            time.sleep(0.30)
            GPIO.output(5, 0)

def send_values(oneWireData):
    data = '{"device":"' + device_id + "', "oneWire":{' + oneWireData + '}}'
    secure = hashlib.md5(data+secureKey).hexdigest()
    data = '{"data":'+data+', "secure":"' + secure + '"}'

    try:
        f = urllib2.urlopen(url, data)
        response = f.read()
        f.close()

        resp_dict = json.loads(response)

        if chkSecure(resp_dict) == True:
            setValve(resp_dict['data'])

            try: print resp_dict
            except: pass

    except:
        GPIO.output(3, 0)
        pass

    else:
        GPIO.output(3, 0)
        time.sleep(0.05)
        GPIO.output(3, 1)
        time.sleep(0.05)
        GPIO.output(3, 0)
        time.sleep(0.1)
        GPIO.output(3, 1)

```

```

        time.sleep(0.05)
        GPIO.output(3, 0)
        time.sleep(0.05)
        GPIO.output(3, 1)

try:
    while True:
        oneWireData= ""

        #for device in device_file:
        for (i, device) in enumerate(device_file):
            device_temp = read_temp(device)
            #print "sensor ", i,":", device_temp
            print
            ""+device.split("/") [5]+"": ""+str(device_temp)+""
            #print "sensor "+i+": "(read_temp(device))

            if isEmpty(oneWireData):
                oneWireData = oneWireData+', '
            oneWireData = oneWireData +
            ""+device.split("/") [5]+"": '+str(device_temp)+'"

            #time.sleep(0.5)
            #GPIO.output(3, 1)
            #GPIO.output(5, 0)

            send_values(oneWireData)

            time.sleep(10)

except KeyboardInterrupt:
    p.stop()

    GPIO.cleanup()

```