Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Ida Koponen

Olosuhdevalvontaratkaisu

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan tutkinto

Insinöörityö

8.4.2019

Tekijä Otsikko Sivumäärä Aika	Ida Koponen Olosuhdevalvontaratkaisu 40 sivua + 4 liitettä 8.4.2019
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Avoimenlähdekoodin järjestelmät, tietoliikenne
Ohjaajat	Ohjaava päällikkö Mikko Olli Ohjaava opettaja Erik Pätynen

Insinöörityön tarkoituksena on tuottaa yksinkertainen, edullinen, helposti asennettavissa ja laajennettavissa oleva ratkaisu konesalitilojen olosuhdevalvontaan. Valvottavia konesalitiloja on kalusteiltaan ja tietoliikenneympäristöiltään erilaisia, mikä estää tavallisten markkinoilta saatavien valvontatuotteiden käytön. Rakennettavan valvontajärjestelmän tulee toimia itsenäisesti ja sijainnistaan riippumattomasti, eikä se saa vaatia muutoksia konesalitilan tietoliikenteessä. Valvontalaitteen arvo tulee olla enimmillään 200 € jokaista konesalitilaa kohti.

Valvontalaitteena käytetään Raspberry Pi 3, johon on kytketty digitaalinen kosteus- ja lämpötilasensori (DHT22). Laitteen tietoliikenneyhteys muodostetaan 3G-modeemin avulla ja vpn-tunnelin kautta valvontapalvelimelle. Valvontajärjestelmänä käytetään avoimeen lähdekoodiin perustuvaa Nagios Core 4:ää, mikä lähettää olosuhteisiin liittyviä hälytyksiä sähköpostitse kohdehenkilöille. MRTG kerää lokitietoa konesalitilan kosteudesta ja lämpötilasta. Nagios ja MRTG hyödyntävät SNMP-protokollaa tietojen keräämisessä.

Insinöörityön tuloksena saatiin toimiva olosuhdevalvontajärjestelmä. Valvontajärjestelmä lähettää varoitus- ja hälytysviestejä sähköpostitse kohonneesta ilmankosteudesta ja lämpötilasta.

Avainsanat	Olosuhdevalvonta, Nagios, SNMP, Raspberry Pi

Author Title Number of Pages Date	Ida Koponen Environmental monitoring 40 pages + 4 appendices 8 April 2019
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Opensource software, networks
Instructors	Mikko Olli, Head of Unit Erik Pätynen, Principal Lecturer

The aim of this Bachelor's thesis was to design and produce a simple, cheap, easily wired and expandable system for monitoring of datacenter room's temperature and humidity. There are many rooms with different types of racks and existing systems, which prevented the use of commercial appliances. The cost of the designed system should be under 200 € for a datacenter room.

The monitoring appliance produced in this thesis project is made of Raspberry Pi 3 computer board with digital humidity and temperature sensor (DHT22). Communications go via a 3G modem and a VPN-tunnel to the monitoring server. Nagios Core 4 is used as a monitoring system. Environmental failure messages will be sent to administrators via email. MRTG is used to get log data of the datacenter room's temperature and humidity. Nagios and MRTG both take advantage of SNMP for collecting the data from an appliance.

Keywords	Environmental monitoring, Nagios, SNMP, Raspberry Pi

Sisällys

Lyhenteet

IETF Internet Engineering Task Force. protokollien standardoimisesta vastaava organisaatio.

SNMP Simple Network Management Protocol. Hallinta- ja valvontaprotokolla.

OID Object identifier. SNMP-objektin tunniste.

MIB Management Information Base. SNMP-tietokanta.

NRPE Nagios Remote Protocol Executive. Nagioksen NRPE-lisäosa.

1-wire Datan siirtoon käytetty sarjaliikenteinen yksittäinen väylä.

Single bus Sama kuin 1-wire.

IC2 Datan siirtoon käytetty kellolla tahdistettu sarjaliikenteinen väylä.

DHT22 Digitaalinen kosteus- ja lämpötilasensori. Sama kuin Aosong AM2302.

RH Relative humidity. Suhteellinen kosteus.

IANA Internet Assigned Numbers Authority. OID-tunnisteiden myöntämisestä vastaava organisaatio.

SPF Sender Policy Framework. Tekstimuotoinen nimipalvelintietue.

VPN Virtual Private Network. Salattu tietoliikenne tunneli.

TCP Transmission Control Protocol. Yhteydellinen tietoliikenne protokolla.

UDP User Datagram Protocol. Yhteydetön tietoliikenne protokolla.

DMZ Demilitarized Zone. Demilitarisoitu alue.

1 Johdanto

Tämä insinöörityö on toteutettu Valtion tieto- ja viestintätekniikkakeskus Valtorille. Valtori tarjoaa toimialariippumattomia ict-palveluja valtionhallinnolle, ja se aloitti toimintansa vuonna 2014. Valtorin hallintaan siirrettiin useita konesalitiloja, jotka sijaitsevat eri puolella Suomea.

Insinöörityön tarkoituksena on suunnitella Valtorin asiakkaiden konesalitilojen olosuhdevalvontaan keskitetysti toimiva, yksinkertainen, edullinen ja laajennettavissa oleva valvontaratkaisu. Konesalitilat sijaitsevat eri puolella Suomea ja niitä on useita kymmeniä. Tilat ovat keskenään hyvin erilaisia niin kalustukseltaan kuin tietoliikenteeltään. Yhteneväistä tiloille on 230 V:n sähkö ja vähintään 3G-verkon kuuluvuusalue.

Olosuhderatkaisu ilmoittaa konesalitilojen ilmankosteuden ja lämpötilan sallitun raja-arvon ylittämisestä, ja siten se antaa hetken aikaa ennakoida tulevaa. Ilmoituksen avulla pystytään reagoimaan ajoissa, jolloin vältytään laitteiden ylikuumenemiselta, sammumiselta ja vikaantumiselta. Kuumenevalle konesalitilalle saadaan järjestettyä hätäjäähdytys ja mahdollisesti hajonneelle jäähdytyslaitteelle huolto ennen kuin lämpötilat kohoavat kriittisen korkealle. Konesalitila saattaa myös lämmetä vahingossa liiallisen lämpökuorman takia.

Olosuhderatkaisulla estettäisiin huomaamaton lämpötilan nousu ja kosteuden lisääntyminen ennen kuin ne ehtivät aiheuttamaan haittaa. Pahimmallaan lämmin ilma vanhentaa laitteita ennen aikaisesti lyhentäen käyttöikää ja liiallinen kosteus aiheuttaa ruostumista.

Teoriaosuudessa kuvataan lyhyesti insinöörityössä käytetyt välineet, sensorit, ohjelmat ja protokollat lyhyesti. Tekniseen osuuteen on dokumentoitu työnvaiheet palvelimen asennuksesta valvontalaitteen käyttöönottoon asti.

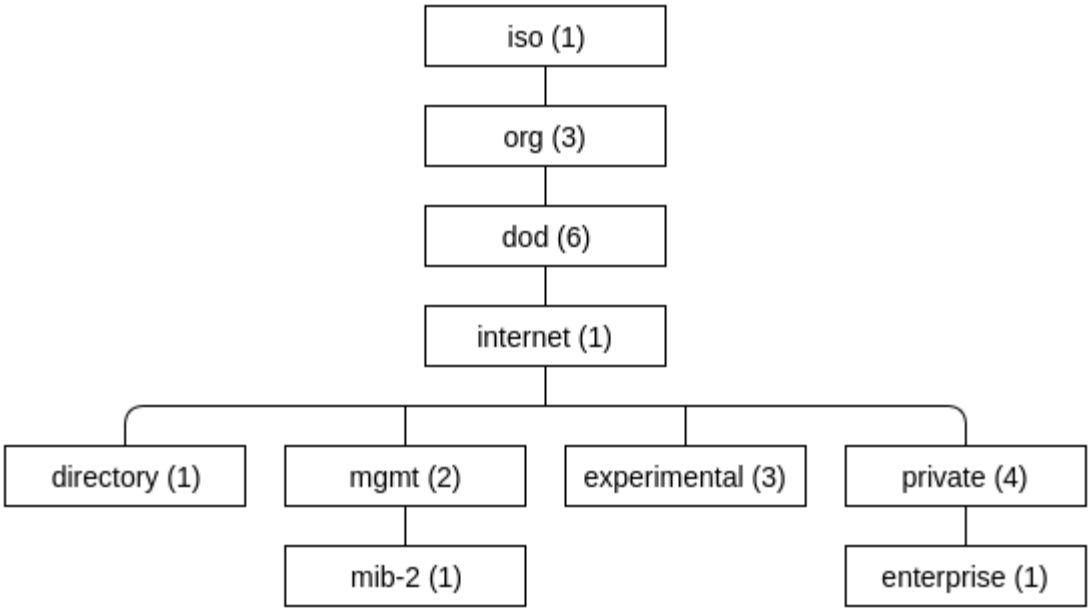
2 Tekniikoita, protokollia ja sensoreita

2.1 SNMP

SNMP on ip-laitteiden hallintatarkoitukseen kehitelty, IETF:n standardoima protokolla, joka esiteltiin vuonna 1988. Protokollasta on julkaistu useampia versioita, joista tuorein on SNMPv3. [1, s. 2.]

SNMP-hallinta koostuu vähintään kahdesta elementistä. SNMP-palvelin (manager) valvoo tekemällä kyselyjä (queries) valvonta-asemalle (agent). Lyhyesti ilmaistuna snmp-kyselyt koostuvat get-, getnext-, set-kyselyistä, joista get- ja getnext-kyselyt palauttavat vain valvonta-aseman tietoa. Set-kyselyllä pystytään muuttamaan valvonta-aseman asetuksia. Kyselyt tehdään ja vastaanotetaan porttiin 161/udp. Valvonta-asema pystyy omatoimisesti lähettämään hälytyksiä SNMP-palvelimelle (trap). Hälytykset osoitetaan porttiin 162/udp. [1, s. 19.]

SNMP käyttää kyselyihin puumaisesta rakenteesta koostuvia OID-yksilöintitunnuksia (Object identifier, OID). OID-tunnus on numerosarja, joka muodostuu kuvassa 1 esitetyn kaavion pohjalta. OID-tunnukset on määritelty MIB-taulukoihin (Management Information Base), joissa säilytetään kyselyihin liittyviä tietoja, kuten esimerkiksi palautuvan tiedon tyyppiä (gauge, string, jne.). MIB-taulukko on selvätekstinen tiedosto, joita laitevalmistajat ja harrastelijat pystyvät kirjoittamaan tarvitsemilleen valvonta-asemille. [1, s. 23.]



Kuva 1. OID-rakenne

Esimerkiksi laitenimi tiedustellaan snmpget-komennolla 1.3.6.1.2.1.1.5.0 -oid:ia käyttäen. Kyselyn tyyppi (string) ja oid (1.3.6.2.1.1.5.0) on määritelty SNMPv2-MIB.tx.t nimisessä MIB-taulukossa.

```
snmpget -v 2c -c <community-salaisuus> 10.8.0.22 1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: raspi2
```

Kuten kuvassa 1 on havainnollistettu, OID-tunnukset on jaettu eri ryhmiin. MGMT(2)-ryhmään kuuluvat yleiseen laitehallintaan liittyvät MIB-taulukot. Experimental(3)-ryhmä on tarkoitettu kokeelliselle kehitykselle, eikä sitä pidä käyttää tuotannossa. Yksityisille organisaatioille on tarkoitettu private(4):n enterprise(1)-ryhmä, josta pystytään hakemaan ilmaiseksi OID-tunnusta. OID-tunnus anotaan https://pen.iana.org/pen/PenApplication.page-sivuston kautta. Kaikki myönnetyt tunnukset ovat listattuina https://www.iana.org/assignments/enterprise-numbers/enterprise-numbers-sivustolla.

SNMP:stä on olemassa kolme versiota. SNMPv1 kehiteltiin 1980-luvulla, ja se on edelleen käytössä. SNMPv1 on omaisuuksiltaan suppea ja tietoturvaltaan heikko. Kyselyt ja hälytysten tekeminen rajoittuu get-, getnext-, set- ja trap-ominaisuuksiin. Palvelimen ja valvonta-aseman välinen luottamussuhde perustuu selväkieliseen community-salaisuuteen.

SNMPv2 koostuu useammasta eri muunnoksesta, kuten SNMPv2p:stä, SNMPv2u:sta, SNMPv2sec:stä ja SNMPv2c:stä. Muunnoksien eroavaisuus on lähinnä turvallisuuteen liittyvissä seikoissa. Suosituimmaksi tuli SNMPv2c, jonka tietoturva perustuu community-salaisuuteen SNMPv1:n tavalla. SNMPv2:ssa on enemmän ominaisuuksia verrattuna SNMPv1:een (mm. getbulk, report, inform -kyselyt). SNMP1 ja SNMPv2 ovat formaatiltaan ja protokollan toiminnaltaan erilaisia.

SNMPv3 on SNMP-versioista tietoturvallisin tarjoten vahvaa autentikointia ja tiedon salausta. Protokollassa on yhteneväisyyksiä SNMPv2:n kanssa, kuten mm. paketin sisältö.

2.2 Raspberry Pi

Raspberry Pi -alusta on Raspberry Pi Foundationin kehittelemä yhden piirilevyn pientietokone. Ensimmäinen Raspberry Pi -tietokone julkaistiin vuonna 2012. Alun perin Raspberry Pi oli kehitelty opetustarkoitukseen, mutta pientietokoneen edullinen hinta ja Linuxiin pohjautuva käyttöjärjestelmä teki alustasta mielenkiintoisen kehityskohteen erilaisille projekteille. [2.]

Raspberry Pi:lle on saatavilla erilaisia lisävarusteita, kuten 7”:n kosketusnäyttö, kameramoduuli sekä Digital DVB-T2 TV adapteri. Erilaisia komponentteja voidaan käyttää GPIO-pinnien kautta, kuten digitaalisia kosteus- ja lämpötilasensoreita. Ethernet-verkkoon liitettynä Raspberry Pi:stä voidaan tehdä mm. NAS-jako tai mediapalvelin. [3.]

Raspberry Pistä on julkaistu kolme sukupolvea (generation 1, 2 ja 3), jotka voidaan jakaa Model A ja Model B -sarjaan. Model A -sarjan Raspberry Pi:t on rakennettu mahdollisimman alhaisilla kustannuksilla, mikä mahdollistaa huokeamman myyntihinnan. Model A ja Model B -sarjan piirilevyjen erot näkyvät: Model A -sarjan Raspberry Pi:t ovat varustukseltaan hiukan karsitumpia.

Raspberry Pi 1 Model B oli ensimmäinen julkaistu Raspberry Pi -alusta (kesäkuu 2012). Se sisälsi RAM-muistia 512 MB ja ARM1176JZF-S 700 MHz 32-bittisen prosessorin. Tämän jälkeen on julkaistu useita Raspberry Pi -alustoja useilla eri kokoonpanoilla. Halvin julkaistu Raspberry Pi oli marraskuussa 2015 myyntiin tullut 32-bittinen Raspberry Pi

Zero, jonka sai parhaimmillaan 5 USD:n hinnalla omaksi. Tällä hetkellä sukupolvista tuorein on Raspberry Pi 3, jonka merkittävin muutos vanhempiin sukupolviin verrattuna on 64-bittinen prosessori.

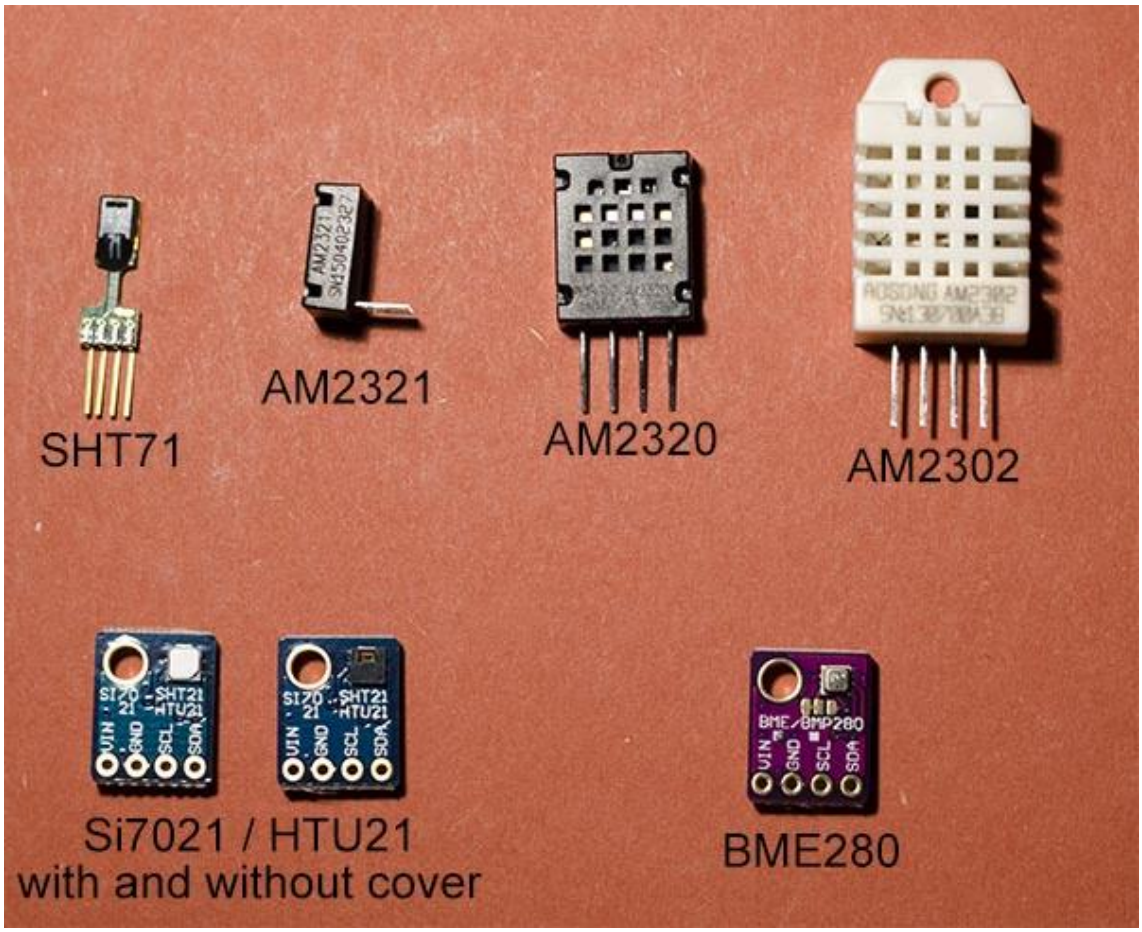
Raspberry Pi Model B+ on kirjoitushetkellä tuorein Raspberry Pi -alusta (julkaistu maaliskuussa 2018). Raspberry Pi 3 sisältää neliytimisen 64-bittisen 1.4 GHz:n Broadcom BCM2837B0 -prosessorin ja 1 GB RAM-muistia. Raspberry Pi 3:ssä on 10/100M ethernet-portti, neljä USB2.0-porttia, HDMI-portti, nelinapainen stereoulostulo ja micro-SD-korttipaikka. Lisäksi piirilevyltä löytyy 40-pinninen GPIO-paikka, CSI-portti Raspberry Pi -kameralle ja DSI-portti Raspberry Pi -kosketusnäytölle. Raspberry Pi 3 tukee myös wlan- ja bluetooth -yhteyksiä. [4.]

Raspberry Pi:lle on tarjolla useita erilaisia Linux-pohjaisia käyttöjärjestelmiä, kuten Raspbian, Pidora, OpenElec, Arch jne. Lisäksi saatavilla on Microsoftin Windows IoT Core ja RISC OS Open Limited -yrityksen kehittelemä RISC OS Pi. [5.]

2.3 Digitaaliset kosteus- ja lämpötilasensorit

Digitaalisia kosteus- ja lämpötilasensoreita löytyy erilaisia kooltaan kuin tekniikaltaan. Kuvassa 2 on näytetty muutamia eri tyyppisiä sensoreita, mm. Aosong AM2302 (ts. DHT22), Aosong AM2320, Aosong AM2321, Sensirion SHT71, Measurement Specialties HTU21D, Silicon Labs Si7021 ja Bosch Sensortec BME280. Sensorit käyttävät 1-wire tai I2C bus -teknologiaa tiedon siirtämiseen. I2C-väylä mahdollistaa useamman orjalaitteen käyttämisen samassa väylässä, sillä isäntälaite tahdistaa väylän kellolla.

Sensorit ovat yhteen sopivia isäntälaitteiden kanssa, jotka tukevat digitaalista tiedon siirtoa ja antavat tarvittavan käyttöjännitteen. Joitakin valmistajakohtaisia rajoituksia saattaa ilmetä. [6.]

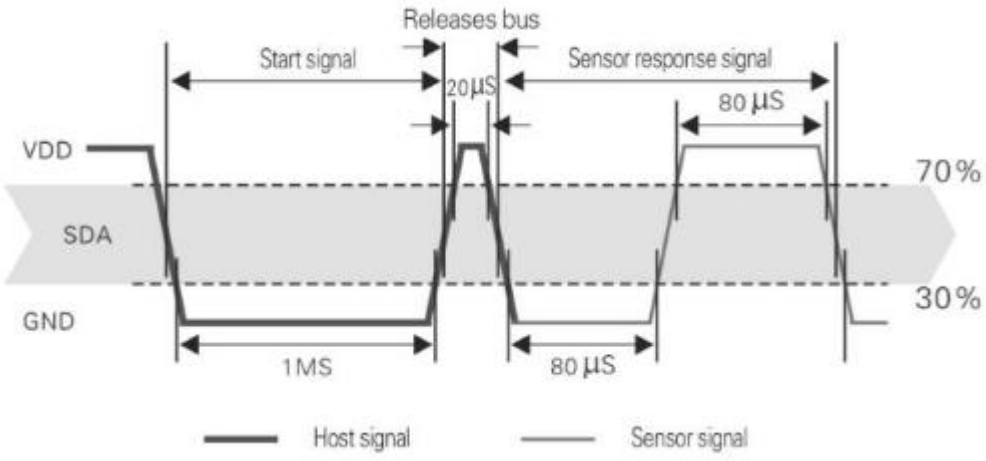


Kuva 2. Digitaalisia kosteus- ja lämpötilasensoreita [6.]

DHT22 (tunnetaan myös AM2302) on pieni, valkoinen, digitaalinen kosteus- ja lämpötilasensori, joka mittaa suhteellista ilmankosteutta 0-100 %:n alueelta ja lämpötilaa -40-+80 celsiusasteen väliltä. DHT22-käyttöjännite on 3.3-5.5 V, suositeltu käyttöjännite on 5 V. Mittaustietojen siirtoon käytetään yhden väylän periaatetta (1-wire, single bus). Tietojen lukeminen perustuu master-slave-asetelmaan, joten isäntälaite tekee pyynnön mittaustietojen lähettämisestä sensorille. Kuvassa 3 DHT22-sensori on levyllä, joka sisältää yleensä vastuksen ja kondensaattorin vakauttamaan tiedon siirtoa.

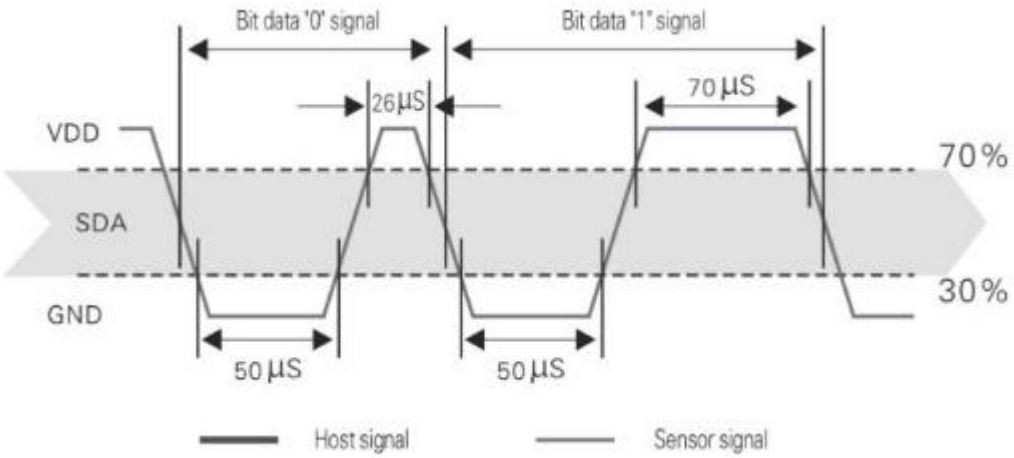
Kuva 3. DHT22:lla on kolme jalkaa käytettävissä

Tietojen lukemiseen ja lähettämiseen kuuluu noin 2 sekuntia. Väylä on vapaana, kun siinä on yhtämittainen jännite. Isäntälaite antaa merkin sensorille asettamalla väylän signaalin maihin 1-10 ms:n ajaksi, jonka jälkeen isäntälaite nostaa väylän jännitteen käyttöjännitteen tasolle 20-40 µs hetkeksi. DHT22 vastaa isännälle tiputtamalla väylän jännitteen maihin 80 µs ja nostamalla ylös 80 µs:n ajaksi. Tämän jälkeen DHT22 on valmis lähettämään kosteus- ja lämpötilalukemat sekä pariteettibitit. Kättely on esitelty kuvassa 4.



Kuva 1: Isäntälaitteen merkkisignaali DHT22-sensorille [7.]

DHT22 lähettää bitit sarjoissa siten, että jokaisen bitin lähetys aloitetaan aina signaalijännitteen ollessa alhaalla 50 µs. Bitit määräytyvät signaalin jännitteen keston mukaan. ”0” lähetettäessä signaalin jännite kuluu aikaa 26-28 µs, kun taas ”1” lähetys kestää 70 µs. Kuvassa 5 asia on havainnollistettu aikajanalle. [7; 8.]



Kuva 2: DHT22:n tiedon välitys isäntälaitteelle [7.]

2.4 Nagios Core

Nagios Core (myöhemmin pelkkä Nagios) on Linux- ja Unix-alustalle kehitetty ilmainen avoimen lähdekoodin valvontaohjelmisto laitteiden ja palveluiden valvontaan. Valvonnan toteuttaminen tehdään Nagioksessa kahdella menetelmällä: SNMP-protokollalla tai NRPE-agentilla. NRPE-agentti voidaan asentaa linux-pohjaisille alustoille, kun taas SNMP-protokollalla pystytään valvomaan kaikkia SNMP:tä tukevia laitteita. SNMP tekee valvonnasta laitteistorajoja ylittävän valvontamenetelmän.

Nagios näyttää valvottavat palvelut ja laitteet html-sivustolla joko ryhmittäin tai yksittäin. Valvonnan tarkemmat määrittelyt tehdään tekstipohjaisiin tiedostoihin komentorivin avulla. Nagioksen hakemistorakenne on selkeä, ja se pystytään tarvittaessa määrittelemään halutulla tavalla.

Valvottavat palvelut ja laitteet voidaan jaotella Nagioksessa erilaisiin ryhmiin, mikä helpottaa hallintaa ja ongelmien laajuuden selvittämistä isoissa ympäristöissä. Tämä mahdollistaa ryhmäkohtaisten ilmoitusten, hälytystasojen ja hälytettävien henkilöiden määrittelemisen.

Nagios tekee valvottavalle kohteelle kyselyjä ja reagoi kohdelaitteen vastauksen tai vastaamattomuuden perusteella, mahdollisesti tehden hälytyksen. SNMP:n trap-ominaisuutta käytettäessä on mahdollista hyödyntää kohdelaitteen itse generoimaa hälytystä. Nagios tekee hälytykset ja ilmoitukset sähköposti- tai sms-viestillä.

Nagioksella on yleisesti ottaen kolme hälytystasoa laitteille ja alustoille. Laite voi olla alhaalla (down), saavuttamattomissa (unreachable) tai toipunut (recovery, host up). Lisäksi Nagios voi tehdä ilmoituksia myös huojumisesta (flapping) ja ajoitetusta alasajosta ja sen päättymisestä (scheduled downtime). Ilmoitukset ja hälytykset ovat säädettävissä halutulla tavalla.

Nagios voidaan määritellä ilmoittamaan ja hälyttämään palveluiden toiminnasta halutulla tavalla. Palvelulle säädetään tietyt raja-arvot, joiden ylittäminen johtaa varoitukseen (warning) tai kriittiseen hälytykseen (critical). Muut ilmoitukset ovat palvelun toipuminen (recovery), palvelun tuntematon tila (unknown), alustan huojuminen (flapping) ja alustan ajoitettu alasajo alkaa tai loppuu (scheduled downtime). [9.]

Nagioksesta on olemassa laajennettu kaupallinen Nagios XI -ohjelmistoversio, jossa on Nagios Corea enemmän ominaisuuksia. Nagios Corea ja Nagios XI:tä kehittää vuonna 2007 Ethan Galdstatin perustama Nagios Enterprise -yritys. Nagioksen alkutaival alkoi NetSaint -nimisenä valvontaohjelmistona 1990-luvulla Ethan Galdstatin ja muiden kehittäjien toimesta. Nagios Core on rekisteröity GNU GPL V2 -lisenssin alle. [10; 11.]

2.5 MRTG

MRTG (Multi Router Traffic Grapher) on avoimen lähdekoodin sovellus (GNU GPL), joka kerää SNMP-protokollaa käyttäen tietoa laitteilta ja piirtää sen perusteella web-sivustolle kuvaajia png-kuvien avulla. MRTG on saatavilla Unix/Linux- ja Windows -järjestelmille. [12.]

2.6 OpenVPN

OpenVPN on ilmainen VPN-ohjelmisto, jolla luodaan salattu tunneli VPN-palvelimen ja asiakaskoneen välille turvattoman internetverkon ylitse.

VPN-tunneli voidaan määritellä bridged- tai routed -tyyppiseksi. Bridged-tyyppinen tunneli mahdollistaa asiakaskoneelle kohdeverkosta IP-osoitteen. Tällöin asiakaskone on kuin olisi kytketty paikallisesti verkkoon. Routed-tyyppisessä tunnelissa tunneli toimii omassa aliverkossaan, josta pääsee vain aliverkon yhdyskäytävää pitkin ulos. Suurin ero tunnelityyppien välillä on, että routed-tunneli ei päästä broadcast-viestejä leviämään tunneliin, kun taas bridged-tunnelissa tätä estettä ei ole. [13.]

VPN-tunnelin pakettiliikenne voidaan määritellä toimimaan TCP- tai UDP-protokollaa hyödyntäen. Oletusarvoisesti tunneli on määritelty käyttämään UDP-protokollaa ja oletusportti on 1194. TCP-protokollaa käytettäessä tunnelin eheyttä testataan jatkuvasti ping-liikenteellä, ellei tunnelissa satu kulkemaan muuta liikennettä. [14; 15.]

VPN-palvelin ja asiakaskone tunnistautuu toisilleen sertifikaattien avulla. Palvelimelle luodaan ca.key, jonka avulla luodaan VPN-palvelimen ja asiakaskoneen salainen ja julkinen avain. Ca.key tulee säilyttää turvallisessa paikassa. [16.]

VPN-palvelimella käytetään Diffie-Hellman parametriä, joka mahdollistaa avaimen turvallisen vaihtamisen VPN-palvelimen ja asiakaskoneen välillä. [17.]

Ta.key-avain antaa suojaa palvelunestohyökkäyksiä vastaan, ja se perustuu ennalta jaettuun avaimeen. Avain tulee pysyä samana kaikilla asiakaskoneilla. VPN-palvelin tarkistaa ta.key-avaimen avulla kaikkien sisään tulevien pakettien allekirjoitukset. Allekirjoituksesta poikkeavat paketit pudotetaan pois varhaisessa vaiheessa, eikä niihin käytetä enempää prosessorin laskenta tehoa. [18.]

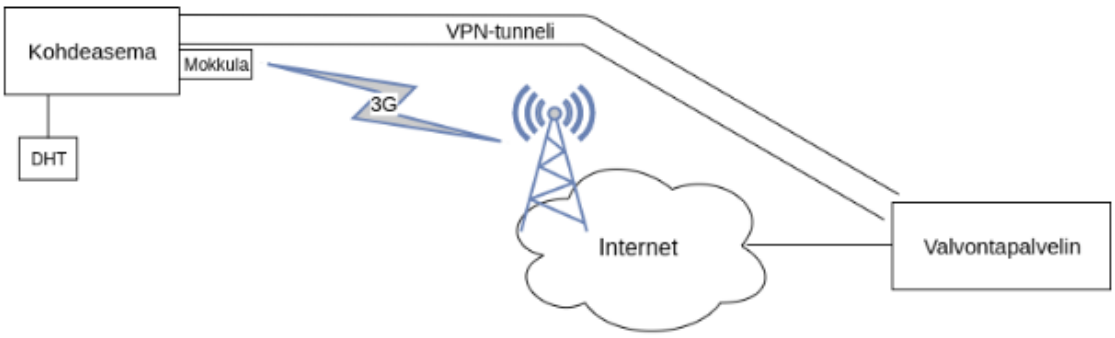
3 Ratkaisu

Olosuhdevalvontaratkaisua lähdettiin pohtimaan valvonnan tarpeellisuudesta ja kohteiden asettamista rajoitteista. Valvonta tulisi olla helposti ja kustannustehokkaasti toteutettavissa. Valvottavia kohteita on eri puolella Suomea, joten valvontalaitteen tulee olla ominaisuuksiltaan itsenäisesti toimiva ja erilaisiin ympäristöihin soveltuva. Valvottavia ominaisuuksia on ensi alkuun ainakin kosteus ja lämpötila, mutta laajennusmahdollisuus muihin sensoreihin tulee olla mahdollista. Laite postitetaan kohdekonesaliin, jossa se vaatii vain paikalle asentamisen ja sähköihin kytkemisen. DHT-sensori asennetaan räkin etupuolelle ylimmän laitteen kohdalle, jolloin saadaan mitattua jäähdytykseen käytetyn ilman lämpötila ja kosteus.

Alustavaihtoehtona oli tietokonepohjainen piirilevy tai mikrokontrolleri. Mikrokontrolleri vaikutti kuitenkin liian suppealta ja hankalalta ylläpidettäväksi, sillä alustan täytyy olla päivitettävissä ja ylläpidettävissä mahdollisimman helposti. Joten alustana päädyttiin käyttämään Raspberry Pi -pientietokonetta ja avoimen lähdekoodiin perustuvaa Rasbian-käyttöjärjestelmää. Rasbian on suosittu ja tuettu käyttöjärjestelmä kaikille Raspberry Pi -alustoille ja sitä kehittää Raspberry Pi -säätiö (Raspberry Pi Foundation).

Valvontaohjelmaksi valikoitui Nagios. Nagios on ilmainen, linux-pohjainen, SNMP-protokollaa tukeva avoimen lähdekoodin valvontaohjelma, jonka valvottavissa laitemäärissä ei ole rajoitteita. Muita valvontaohjelmistoja on mm. Zabbix, jossa valvontaa voidaan tehdä mm. tietoturvallisesti SSH-protokollalla, tai Incinga, jolla on yhteneväisyyksiä Nagioksen kanssa. Olosuhdevalvontajärjestelmän protokollana käytetään SNMP:tä, Nagios soveltuu valvontaohjelmaksi sen helppokäyttöisyyden takia. Valvonta-asemana toimii Raspberry Pi 3 -pientietokone ja tietoliikenne hoituu mokkulalla 3G-verkon avulla VPN-tunnelin kautta. Valvontalaite käynnistyy plug and play -periaatteen mukaisesti ja ottaa automaattisesti yhteyden Nagios-palvelimeen.

Nagios valvoo valvontalaitetta tekemällä SNMP-kyselyjä ja hälyttää raja-arvon ylittäneistä lukemista sähköpostitse. Kohdekonesalitilan lämpötila- ja kosteuslukemat kerätään talteen MRTG:llä myöhempää tarkastelua varten.



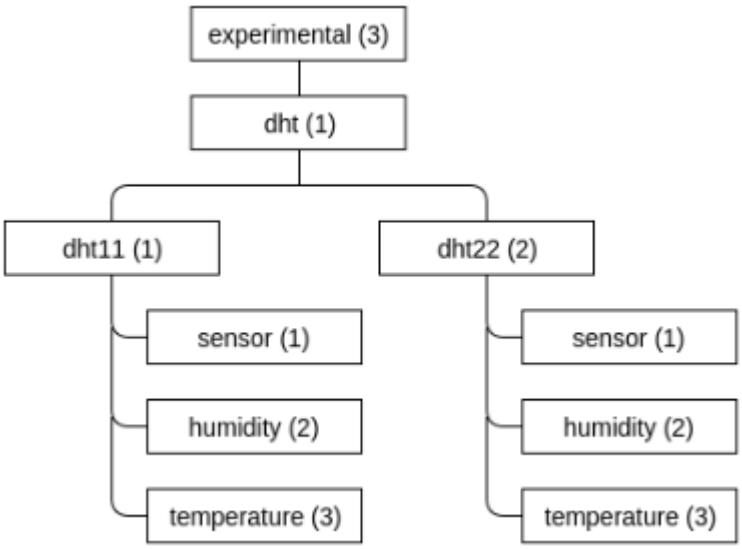
Kuva 3: Olosuhdevalvonnan komponentit ja verkko

Valvontajärjestelmää varten palvelimelle asennetaan Nagios, Snmpd, OpenVPN ja MRTG sekä niiden vaatimat riippuvuudet. Palvelinalustana toimii CentOS 7.5 -virtuaalipalvelin, jolle on annettu muistia 2 GB ja levytilaa 10 GB.

Kohdelaitteelle asennetaan Snmpd-, Wvdial-, Ppp-, OpenVPN- ja DHT22-sensorin vaatimat skriptit ja Adafruitin kirjasto. Alustana käytetään Raspberry Pi 3:sta, johon on kytketty DHT22-sensori, 16G:n SD-kortti, Huawei E169 -mokkula ja sim-kortti.

Olosuhdevalvontajärjestelmän SNMP-palvelua varten lähdettiin kehittelemään MIB-taulukkoa koekäyttömielessä experimental OID:n alaisuuteen (1.3.6.1.3), mikä on vain kehittelyä ja tutkimista varten. DHT22-sensorin kosteuden ja lämpötilan kyselyyn käytetään 1.3.6.1.3.1.2.2 ja 1.3.6.1.3.1.2.3 -OID-tunnuksia.

MIB-taulukoon ja OID-suunnitteluun on jätetty laajentumiselle varaa, joten siihen saadaan lisättyä helposti useampia sensoreita. Kuvassa 7 on esitettynä olosuhdevalvonnan OID-rakenne.



Kuva 4: DHT-sensoreiden OID-rakenne

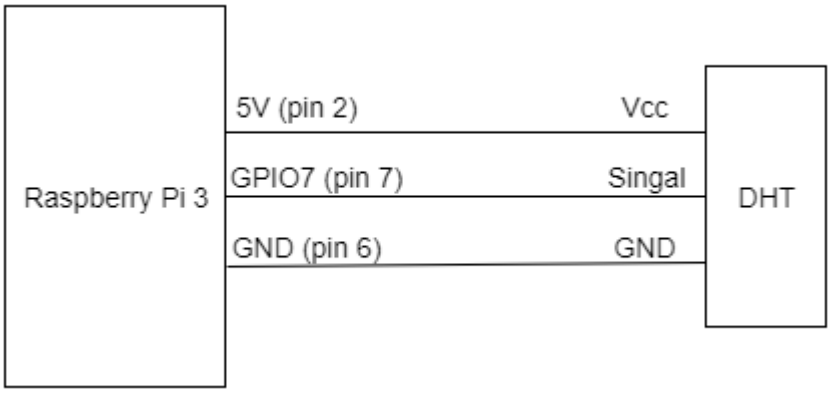
3.1 Valvontalaitteen asentaminen

Valvontalaite on olosuhteita mittaava Raspberry Pi, johon on asennettu Rasbian-käyttöjärjestelmä.

Seuraavissa luvuissa tehdään valvontalaitteelle tarvittavat asennukset. Ensimmäisenä tarkastellaan DHT22-sensoriin liittyviä kytkentöjä ja dht-kirjastojen asentamista. Seuraavaksi tehdään snmpd-palvelun asentaminen ja olosuhdevalvontaan liittyvien skriptien määrittelyt siten, että saadaan toimiva SNMP-valvonta. Viimeisenä asennetaan 3G-modeemilla toimiva tietoliikenneyhteys ja OpenVPN tunnelia varten. Valvontalaitteelle määritellään palvelinyhteyttä valvova skripti, joka auttaa toipumaan yhteyden menetyksestä. Lopuksi määritellään valvontalaitteelle Firewalld-palomuuri.

3.1.1 DHT22-sensori

DHT22-sensori kytketään Raspberry Pi 3:n GPIO-portteihin kuvan 8 mukaisella tavalla. Tarkemmat DHT22-sensorin ja Raspberry Pi 3:n GPIO -pinnikartat ovat liitteenä.



Kuva 5: DHT22-sensorin kytkentäkartta

Sensorin vaatimat kirjastot ladataan GitHubista, jonka jälkeen kirjastot asennetaan valvontalaitteelle.

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
sudo apt-get update
sudo apt-get install build-essential python-dev python-openssl
sudo python Adafruit_Python_DHT/setup.py install
```

DHT22-sensorille luodaan kansio /usr/local/rpirackmon, jonne tallennetaan lämpötila- ja kosteuden lukemista varten tarvittavat python-tiedostot ja skriptit. Pass_dht22-humidity ja pass_dht22-temperature kyselevät DHT22_humidity.py:ltä ja DHT22_temperature.py:ltä kosteus- ja lämpötilatietoja. Esimerkkikoodissa 1 on esitettynä DHT22_humidity.py-tiedoston sisältö. Arvot tulostetaan kokonaislukuina.

```
#!/usr/bin/python
import sys
import Adafruit_DHT

pin = "7"
sensor = Adafruit_DHT.DHT22;

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

if humidity is not None and temperature is not None:
     humidity = round(humidity, 0)
     print(int(humidity))
else:
    print('Failed to get reading. Try again!')
    sys.exit(1)
```

Esimerkkikoodi 1. DHT22_humidity.py

DHT22_humidity.py ja DHT22_temperature.py tulostaa arvot suoritettaessa.

```
./usr/local/rpirackmon/DHT22_humidity.py
19
./usr/local/rpirackmon/DHT22_temperature.py
21
```

SNMP:n kosteuden ja lämpötilan arvot luetaan pass_dht22-humidity ja pass_dht22-temperature -skripteillä. Skripti suoritetaan vain ”-g” -option ollessa valittuna, ja se tulostaa standardin mukaisesti oid-tunnuksen, gauge32-tyypin ja kosteus- tai lämpötilalukeman. Esimerkkikoodissa 2 on pass_dht22-humidityn skripti.

```
#!/bin/bash
#
if [ "$1" = "-g" ]
  then
  echo .1.3.6.1.3.1.2.2
  echo gauge
  python /usr/local/rpirackmon/DHT22_humidity.py
fi
exit 0
```

Esimerkkikoodi 2. pass_dht22-humidity

3.1.2 SNMP

Kohdelaitteelle asennetaan snmpd- ja snmp-palvelut apt-get install -komennolla komentorivillä.

```
apt-get install snmp snmpd
```

Snmpd.conf-tiedostoon määritellään esimerkkikoodi 3:n mukaisesti pass-lausekkeet, jotka linkittävät oid-tunnuksen suoritettaviin skripteihin. Skriptit suoritetaan vain ”-g” -optiolla.

```
pass .1.3.6.1.3.1.2.1 /bin/bash /usr/local/rpirackmon/pass_dht22-sensor -g
pass .1.3.6.1.3.1.2.2 /bin/bash /usr/local/rpirackmon/pass_dht22-humidity -g
pass .1.3.6.1.3.1.2.3 /bin/bash /usr/local/rpirackmon/pass_dht22-temperature -
g
```

Esimerkkikoodi 3. Pass-lausekkeet

Gpio-ryhmään lisätään Debian-snmp-käyttäjä ja snmp-palvelu käynnistetään uudelleen.

```
usermod -aG gpio Debian-snmp
systemctl snmpd restart
```

Valvontalaitteen SNMP-kyselyt onnistuvat paikallisesti tehtynä.

```
snmpget -v 2c -c olosuHd3 localhost 1.3.6.1.3.1.2.2
iso.3.6.1.3.1.2.2 = Gauge32: 33
```

3.1.3 Tietoliikenneyhteys

Valvontalaitteelle asennetaan 3G-modeemia varten Wvdial- ja Ppp-ohjelmat apt-get -komennolla alla olevan mukaisesti.

```
apt-get install wvdial ppp
```

Wvdial on rajapintasovellus, joka käskyttää Ppp:tä, ja Ppp hoitaa 3G-yhteyden luomisen modeemia hyödyksi käyttäen. Toimivaa 3G-yhteyttä varten tarvitaan operaattorin vaatimat asetukset /etc/wvdial.conf-tiedostoon. Parametrit ovat operaattorikohtaisia. Seuraavana ovat Saunalahden vaatimat parametrit. [19.]

```
[Dialer defaults]
Modem = /dev/ttyUSB0
Init = AT+CGDCONT=1,"IP","internet.saunalahti"
Phone = *99***1#
Stupid Mode = 1
Username = " "
Password = " "
```

Wvdial.conf-tiedoston virheet tarkistetaan wvdialconf-komentoa hyödyksi käyttäen.

```
wvdialconf
```

Kun virheitä ei löydy, niin wvdialconf-komento käsittelee tiedostoa ja muokkaa sitä lisäämällä tietoja ja muuttamalla hiukan merkintätapaa, kuten esimerkkikoodi 4:ssa on mainittu.

```
[Dialer defaults]
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
Init3 = AT+CGDCONT=1,"IP","internet.saunalahti"
Phone = *99***1#
New PPPD = yes
Modem Type = Analog Modem
```

```
Stupid Mode = 1
Auto DNS = Off
Modem = /dev/ttyUSB0
ISDN = 0
Username = { }
Password = { }
Baud = 9600
```

Esimerkkikoodi 4. Wvdial.conf-tiedoston sisältö

3G-yhteyden toimivuutta testataan suorittamalla wvdial-komento.

```
wvdial
```

Onnistunut yhteys havaitaan Rasbianin ppp0-liitäntänä ja sen kautta pystytään liikennöimään internettiin päin.

3G-yhteyden tulee käynnistyä automaattisesti valvontalaitteen käynnistyessä. Wvdialle tehdään /etc/init.d/-kansioon käynnistysskripti. [20.]

OpenVPN asennetaan apt-get-komennolla ja sen sallitaan käynnistyä valvontalaitteen käynnistyksen yhteydessä.

```
apt-get install openvpn
systemctl enable openvpn
```

OpenVPN:n client.conf-tiedostoon määritellään esimerkkikoodin 5 mukaiset asetukset tunnelin muodostusta varten.

```
client
dev tun
proto udp
remote <palvelimen ip> 3990
resolv-retry infinite
nobind
persist-key
persist-tun
ca /etc/openvpn/client/ca.crt
cert /etc/openvpn/client/raspi2.crt
key /etc/openvpn/client/raspi2.key/
ramote-crt-tls server
key-direction 1
tls-auth /etc/openvpn/client/ta.key 1
cipher AES-256-CBS
verb 3
```

Esimerkkikoodi 5. Client.conf-asetukset

Valvontalaite tarvitsee VPN-tunnelia varten raspi2.key-, raspi2.crt-, ca.crt- ja ta.key-avaimet ja -sertifikaatit /etc/openvpn/client-hakemistoon. Avaimet ja sertifikaatit toimitetaan sinne.

Wvdial muokaa reititystaulukkoa ja asentaa sinne oletusreitin internetiin päin, mikä ei sovellu sellaisenaan OpenVPN:n käyttöön. Reitti voidaan määritellä käsin, mutta parhaiten se kannattaa asentaa skriptillä. Skriptiä voidaan myös hyödyntää jälkikäteen.

OpenVPN:ää varten luodaan esimerkki 6 mukainen skripti. Skripti asentaa reititystaulukkoon suoran reitin valvontapalvelimelle. Jos reitti on jo asennettu reititystaulukkoon, niin skriptiä ei ajeta loppuun asti.

```
#!/bin/bash
route_found=$(/sbin/route -n | /bin/grep -c ^<palvelimen_ip>)
ppp_on=$(/sbin/ifconfig | /bin/grep -c ppp0)
if [ $route_found -eq 0 ] && [ $ppp_on -eq 1 ]
  then /sbin/ip route add <palvelimen_ip>/32 via 10.64.64.64
fi
```

Esimerkkikoodi 6. Route2server-skripti

OpenVPN:n oletusreitin asennusskripti suoritetaan taustalla Crond:lla kahden minuutin välein, kuten esimerkkikoodi 7:ssa on määritelty.

```
*/2 * * * *     root /usr/local/inssi/route2server &
```

Esimerkkikoodi 7. /etc/cron.d/route2server-tiedosto

Uudelleen käynnistyksen jälkeen valvontalaitteen 3G-yhteys ja VPN-tunneli käynnistyvät automaattisesti. Yhteys palvelimelle on toimiva.

3G-modeemin yhteys voi katketa yllättäen operaattorista tai jostain muusta ulkoisesta häiriöstä johtuen. Raspi2:lle lisätään esimerkkikoodin 9 mukainen skripti, joka palauttaa yhteyden valvontapalvelimelle. Skriptillä tarkistetaan yhteyden toimivuus kolmen minuutin välein crond:llä, kuten esimerkkikoodi 8:ssa on mainittu.

Esimerkkikoodi 8:n mukaisesti tallennetaan /etc/cron.d/recovery-tiedostoon.

```
*/3 * * * * root /usr/local/inssi/recovery
```

Esimerkkikoodi 8. Recoveryn crond-tiedosto

Esimerkkikoodi 9:n skripti tallennetaan /usr/local/inssi/recovery-tiedostoon. Skripti tiedustelee ping-komennolla yhteyttä. Jos yhteyttä ei ole, niin ensimmäiseksi käynnistetään Wvdial uudelleen, seuraavaksi asennetaan reitti serverille ja viimeisenä käynnistetään vielä OpenVPN uudelleen. Yhteyden uudelleen käynnistyksestä kirjataan lokitieto /var/log/inssirecovery.log-tiedostoon.

```
#!/bin/bash
# Modem Hang Up Recovery
/bin/ping -q -c5 10.8.0.1 >/dev/null

if [ $? -eq 0 ]
then
      echo ”VPN-tunnel ok!”
else
      echo ”VPN-tunnel is broken!”
      systemctl restart wvdial
      sleep 30
      /usr/local/inssi/route2server
      systemctl restart openvpn@client
      /bin/date >> /var/log/inssirecovery.log
fi
```

Esimerkkikoodi 9. Recovery-skripti

Valvontalaitteelle asennetaan Firewalld-palomuuri apt-get komennolla.

```
apt-get install firewalld
```

Valvontalaitteen Firewalld-palomuurilla käytetään public ja dmz-nimisiä zoneja. Zonella voidaan määritellä sääntöjä, jotka myönnetään zonen interfacelle. Interfacena toimii fyysinen portti, tunnelin portti tai 3G-modeemin portti. Valvontalaitteen palomuurissa käytetään kahta zonea: public ja dmz. Public-zonelle lisätään ppp0-interface. Oletuksena siihen on jo liitetty eth0. Public-zonea käytetään julkisessa verkossa.

Public-zonelle avataan SSH-yhteys hallintaa varten firewall-cmd-komennolla ja add-service-parametrilla. Zone liitetään ppp0-interfaceen add-interface parametrilla. Eth0-inteface on valmiiksi liitetty järjestelmän puolelta public-zoneen.

```
firewall-cmd --zone=public --add-service=ssh --permanent
firewall-cmd --zone=public --add-interface=ppp0 --permanent
```

Tun0-interface liitetään palomuurin dmz-zoneen ja siihen avataan portit ssh ja snmp-yhteyksiä varten. Lopuksi palomuuri käynnistetään uudelleen reload-parametrilla.

```
firewall-cmd --zone=dmz --add-service=ssh --permanent
firewall-cmd --zone=dmz --add-service=snmp --permanent
firewall-cmd --zone=dmz --add-interface=tun0 --permanent
firewall-cmd --reload
```

3.2 Valvontapalvelimen asentaminen

Valvontapalvelin on CentOS 7-palvelin, jolla on vähintään yksi julkinen IP-osoite.

Valvontapalvelimen asennus koostuu useammasta vaiheesta. Ensimmäisenä asennetaan tietoliikennettä varten Firewalld, OpenVPN ja sen tarvitsemat sertifikaatit ja avaimet. Seuraavana vaiheena asennetaan toimiva SNMP-palvelu ja siihen tarvittava MIB-taulukko. Kolmantena asennetaan Nagios-palvelu, johon säädetään tarvittavat parametrit valvontalaitteen valvomista varten. Viimeisenä asennetaan MRTG lokien keräystä varten.

3.2.1 Tietoliikenne

OpenVPN käyttää oletuksena vpn-tunnelissa 1194/udp -porttia. Tietoturvan kannalta sovellus siirretään 3390-porttiin. Palvelimella olevan Firewalld-sovelluksen palomuuriasetuksia varten muutetaan /usr/lib/firewalld/services/openvpn.xml-tiedostoa esimerkkikoodin 10 mukaisesti.

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>OpenVPN</short>
  <description>OpenVPN is a virtual private network (VPN) solution. It is used
to create encrypted point-to-point tunnels between computers. If you plan to
provide a VPN service, enable this option.</description>
  <port protocol="udp" port="3990"/>
</service>
```

Esimerkkikoodi 10. Openvpn.xml-tiedosto

Firewalld määritellään käyttämään asetuksia.

```
firewall-cmd --zone=public --add-service=openvpn --permanent
```

```
firewall-cmd --reload
```

3.2.2 OpenVPN

Valvontapalvelimelle asennetaan yum-komennolla OpenVPN- ja EasyRSA-ohjelma. EasyRSA:lla luodaan VPN-tunnelin tarvitsemat avaimet ja sertifikaatit valvontapalvelimelle ja kohdelaitteelle.

```
yum install openvpn easy-rsa
```

OpenVPN varten kopioidaan vars.example-mallitiedosto, jota lähdetään muokkaamaan.

```
cp -p /usr/share/doc/easy-rsa-3.0.3/vars.example /etc/openvpn/easy-rsa/vars
```

Mallitiedoston sisältö muutetaan vastaamaan paremmin järjestelmään liittyviä tietoja.

```
set_var EASYRSA_REQ_COUNTRY     "FI"
set_var EASYRSA_REQ_PROVINCE    "Uusimaa"
set_var EASYRSA_REQ_CITY        "Espoo"
set_var EASYRSA_REQ_ORG         "insinöörityö"
set_var EASYRSA_REQ_EMAIL       "<sähköpostiosoite>"
set_var EASYRSA_REQ_OU          "insinöörityö"
set_var EASYRSA_REQ_CN          "palvelin"
```

Esimerkkikoodi 11. Vars-tiedoston sisältöä

Sertifikaattien ja avaimien luominen aloitetaan suorittamalla /etc/openvpn/easy-rsa/easyrsa-tiedosto, jonka jälkeen luodaan kansiot ja sertifikaatti.

```
./easyrsa
./easyrsa init-pki
./easyrsa build-ca
```

Common name -parametriksi määritellään palvelimen FQDN-nimi. Sertifikaatti ja avain ovat easy-rsa/pki/ca.crt ja easy-rsa/private/ca.key. Ca.key tulee säilyttää huolellisesti, koska sitä käytetään palvelimen ja valvontalaitteiden sertifikaattien luomisessa. Diffie-Hellman-parametri luodaan gen-dh-parametrillä ja sen nimi muutetaan dh2048.pem:ksi. Kyseistä parametriä käytetään tietoliikenneyhteyden avaimien salaukseen.

```
./easyrsa gen-dh
mv dh.pem dh2048.pem
```

Palvelimen sertifikaattia luodessa määritellään ”-nopass”-optio, jotta palvelin ei kysy salasanaa OpenVPN:n käynnistyessä.

```
./easy-rsa build-server-full palvelin -nopass
```

Ta.key-avain luodaan openvpn-komentoa käyttäen.

```
openvpn --genkey --secret ta.key
```

Seuraavat tiedostot siirretään /etc/openvpn-kansioon:

```
ca.crt
dh2048.pem
palvelin.crt
palvelin.key
ta.key.
```

Palvelimen server.conf-tiedostoon määritellään asetukset esimerkkikoodin 12 mukaisesti.

```
port 3990
proto udp
dev tun
ca ca.crt
cert palvelin.crt
dh dh2048.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
client-config-dir ccd
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.4.4"
push "dhcp-option DNS 8.8.8.8"
keepalive 10 120
cipher AES-256-CBC
compress lz4-v2
push "compress lz4-v2"
user nobody
group nobody
persist-key
persist-tun
status openvpn-status.log
verb 3
```

Esimerkkikoodi 12. server.conf-tiedosto

OpenVPN-palvelu myöntää kohdelaitteelle IP-osoitteet dynaamisesti VPN-yhteyden muodostuessa. Jotta IP-osoite ei ole sertifikaattiin sidottu, niin ”Client-config-dir ccd”:n

kohdalla varmistetaan IP-osoitteen annettavan laitenimen perusteella. /etc/openvpn/ccd-kansiossa määritellään laitenimi ja sen IP-osoite esimerkkikoodin 13 mukaisesti.

```
more /etc/openvpn/ccd/raspi2
ifconfig-push 10.8.0.22 10.8.0.1
```

Esimerkkikoodi 13. Raspi2:n IP-osoitteen asetukset

OpenVPN määritellään käynnistymään.

```
systemctl enable openvpn@server
systemctl start openvpn@server
```

3.2.3 SNMP

Palvelimelle asennetaan snmpd-palvelu yum-komennolla.

```
yum install snmpd
```

Snmpd.conf-tiedostoon määritellään community-salaisuus ja .1.3.6.1.3.1 -OID-tunnus.

```
com2sec notConfigUser  default       olosuHd3
view    systemview    included   .1.3.6.1.3.1
```

Snmpd-palvelu tarvitsee DHT-sensorille tehtäviä kyselyjä varten MIB-taulukon. Tätä varten luodaan /usr/share/snmp/mib/RPI-RACKMON-MIB.txt-tiedosto, jonka sisältö on eriteltynä kokonaisuudessaan liitteessä. Esimerkkikoodi 14:ssa on lyhyesti MIB-taulukon sisältöä. Palautettava arvo on kokonaislukutyyppinen (Gauge32) ja rpiRackmonDht22 -object indentifier on 1.3.6.1.3.2.1, jonka alaisuudesta löytyy kosteuden (.1.3.6.1.3.1.2.2) ja lämpötilan (.1.3.6.1.3.1.2.3) kohdat.

```
rpiRackmonDht22         OBJECT IDENTIFIER ::= { rpiRackmon 2 }

rpiRackmonDht22sensor OBJECT-TYPE
  SYNTAX                DisplayString (SIZE (0..255))
  MAX-ACCESS            read-only
  STATUS                current
  DESCRIPTION           "DHT22 Digital humidity and temperature sensor."
  ::= { rpiRackmonDht22 1 }

rpiRackmonDht22Hum OBJECT-TYPE
  SYNTAX                Gauge32
  MAX-ACCESS            read-only
  STATUS                current
```

```
  DESCRIPTION           "DHT22 humidity value (%)."
  ::= { rpiRackmonDht22 2 }

rpiRackmonDht22Temp OBJECT-TYPE
  SYNTAX                Gauge32
  MAX-ACCESS            read-only
  STATUS                current
  DESCRIPTION           "DHT22 temperature value (celsius)."
  ::= { rpiRackmonDht22 3 }
```

Esimerkkikoodi 14. RPI-RACKMON-MIB.txt-tiedoston sisältöä

Snmpd:lle määritellään käynnistysasetukset, ja tämän jälkeen snmp-kyselyt ovat toimivia.

```
systemctl enable snmpd
systemctl start snmpd
```

3.2.4 Nagios

Insinöörityössä käytettävä Nagioksen versio on 4.3.4 ja nagios-pluginien versio 2.2.1.

3.2.5 Nagios-ohjelmiston asentaminen

Nagios vaatii toimiakseen asennettavaksi seuraavat kirjastot ja ohjelmat.

```
yum install gcc make imake binutils cpp postgresql-devel mysql-libs mysql-de-
vel openssl openssl-devel pkgconfig dg dg-devel dg-progs libpng libpng-devel
libjpeg lib-jpeg-devel perl perl-devel net-snmp net-snmp-devel net-snmp-perl
net-snmp-utils httpd php
```

Nagiosta varten luodaan nagios- ja nagios-cmd-ryhmät. Ryhmiin liitetään jäseneksi nagios-käyttäjä, jonka kotihakemistona on /opt/nagios-kansio.

```
groupadd nagios
groupadd nagioscmd
useradd -g nagios -G nagioscmd -d /opt/nagios nagios
```

Nagioksen asennusmedia ja lisäosat ladataan www.nagios.org/download-sivustolta. Asennusmedia tallennetaan ja puretaan Nagiokselle luotuun /usr/src/nagios4-kansioon.

```
tar -xvf Nagios-4.0.tar.gz
tar -xvf nagios-plugins-1.4.16.tar.gz
```

Asennusta varten luodaan tiedostot.

```
mkdir -p /opt/nagios /etc/nagios /var/nagios
chown nagios:nagios /opt/nagios /etc/nagios /var/nagios
```

Nagios käännetään ja asennetaan.

```
sh configure \
--prefix=/opt/nagios \
--sysconfdir=/etc/nagios \
--localstatedir=/var/nagios \
--libexecdir=/opt/nagios/plugins
--with-command-group=nagioscmd
make all
make install
make install-commandmode
make install-config
make install-init
```

Nagios säädetään käynnistymään automaattisesti palvelimen käynnistyksen yhteydessä.

```
systemctl enable nagios
```

Nagioksen lisäosat asennetaan.

```
sh configure \
     --prefix=/opt/nagios \
     --sysconfdir=/etc/nagios \
     --localstatedir=/var/nagios \
     --libexecdir=/opt/nagios/plugins
make all
make install
```

Nagioksen websivusto tarvitsee toimiakseen luku- ja suoritusoikeudet /opt/nagios-kansiossa.

```
drwxr-xr-x 7 nagios nagios 4096 Mar 24  2018 nagios
```

Sivuston asetukset määritellään /etc/httpd/conf.d/nagios.conf-tiedostoon. Sivustolle pääsyä rajoitetaan htacces-kirjautumisella.

```
ScriptAlias /nagios/cgi-bin /opt/nagios/sbin
Alias /nagios /opt/nagios/share

<DirectoryMatch /opt/nagios/share>
```

```
        Options FollowSymLinks
        AllowOverride AuthConfig
        Order Allow,Deny
        Allow From All
        AuthName "Nagios Access"
        AuthType Basic
        AuthUserFile /etc/nagios/htpasswd.users
        AuthGroupFile /etc/nagios/htpasswd.groups
        require valid-user
</DirectoryMatch>

<DirectoryMatch /opt/nagios/sbin>
        Options ExecCGI
        AllowOverride AuthConfig
        Order Allow,Deny
        Allow From All
        AuthName "Nagios Access"
        AuthType Basic
        AuthUserFile /etc/nagios/htpasswd.users
        AuthGroupFile /etc/nagios/htpasswd.groups
        require valid-user
</DirectoryMatch>
```

Websivuston kirjautumisessa käytetään Nagioksen htpasswd.group ja htpasswd.users -tiedostoja. Htpasswd.users -tiedosto luodaan ”htpasswd -c” -komentoa käyttäen. Samalla luodaan nagiosadmin-käyttäjä. Myöhemmin käyttäjiä lisättäessä ”-c”-optio jätetään pois.

```
cp /dev/null /etc/nagios/htpasswd.groups
htpasswd -c /etc/nagios/htpasswd.users nagiosadmin
```

Htpasswd.group- ja htpasswd.users-tiedostojen omistus-, luku- ja kirjoitusoikeudet määritellään alla olevan mukaisesti.

```
-rw-r----- 1 root   nagioscmd     0 Mar 24  2018 htpasswd.groups
-rw-r--r-- 1 root   nagioscmd    85 Sep 26 18:18 htpasswd.users
```

Websivuston käyttöoikeudet on määritelty /etc/nagios/cgi.cfg-tiedostossa oletusarvoisesti nagiosadmin-käyttäjälle. Uusia käyttäjiä voi lisätä merkitsemällä käyttäjänimen pilkulla eroteltuna.

```
authorized_for_system_information=nagiosadmin
authorized_for_configuration_information=nagiosadmin
authorized_for_system_commands=nagiosadmin
authorized_for_all_services=nagiosadmin
authorized_for_all_hosts=nagiosadmin
authorized_for_all_service_commands=nagiosadmin
authorized_for_all_host_commands=nagiosadmin
```

Nagios-palvelu käynnistetään systemctl-komennolla.

```
systemctl start nagios
```

Nagios käynnistyy ja web-sivusto vastaa pyytäen käyttäjätunnusta ja salasanaa.

3.2.6 Nagioksen tiedostohierarkian luominen

Nagioksessa käytettävät kansiot määritellään /etc/nagios/nagios.cfg-tiedostossa. Selkeyden vuoksi tiedostoja on hyvä olla eriteltynä laitteille ja palveluille sekä niiden ryhmille.

```
cfg_dir=/etc/nagios/commands
cfg_dir=/etc/nagios/timeperiods
cfg_dir=/etc/nagios/contacts
cfg_dir=/etc/nagios/contactgroups
cfg_dir=/etc/nagios/hosts
cfg_dir=/etc/nagios/hostgroups
cfg_dir=/etc/nagios/services
cfg_dir=/etc/nagios/servicegroups
```

Nagios-palvelun asetusten oikeellisuus testataan erillisellä komennolla, jonka jälkeen palvelu käynnistetään uudelleen muutosten voimaan saamiseksi.

```
/opt/nagios/bin/nagios -v /etc/nagios/nagios.cfg
service nagios restart
```

3.2.7 Hälytykset

Hälytyksille määritellään /etc/nagios/contactgroups/contactgroups.cfg-tiedostoon admins-ryhmä, joka vastaanottaa olosuhdevalvontaan liittyviä hälytyksiä. Jäseneksi liitetään nagiosadmin käyttäjä.

```
define contactgroup{
  contactgroup_name     admins
  alias                 Administrators
  members               nagiosadmin
  }
```

Jäsenen yhteystiedot määritellään /etc/nagios/contacts/contacts.cfg-tiedostoon. Nagiosadmin ottaa vastaan laitteisiin ja palveluihin liittyviä hälytyksiä sähköpostitse viikon jokaisena hetkenä.

```
define contact{
  contact_name                          nagiosadmin
  alias                                 Nagiosadmin
  email                                 <admins.email@here>
  contactgroups                         admins
  host_notification_period              24x7
  service_notification_period           24x7
  host_notification_options             d,u,r
  service_notification_options          w,u,c,r
  host_notification_commands            notify-host-by-email
  service_notification_commands         notify-service-by-email
 }
```

Palvelimella täytyy olla toimiva Postfix tai jokin muu sähköpostisovellus ilmoitusten ja hälytysten lähettämiseksi. Postfixille määritellään seuraavat asetukset. Asetukset määritellään /etc/postfix.main.cf-tiedostoon.

```
myhostname = <palvelimen FQDN>
mydomain = <domain>
myorigin = $myhostname
inet_interfaces = $myhostname
home_mailbox = Maildir/
```

Tämän jälkeen Postfix käynnistetään uudelleen.

```
systemctl restart postfix
```

DNS-palvelimelle lisätään valvontapalvelimen SPF-tietue. Tietueen avulla verkkotunnuksen omistaja voi määritellä palvelimet, jotka voivat lähettää viestiä verkkotunnuksen nimissä. Tietueen tarkoitus on karsia roskapostia. Tietueeseen määritellään palvelimen IPv4- ja IPv6-osoitteet. ”-all”-parametri määrittelee hyväksyttävät palvelimen IP-osoitteet.

```
v=spf1 a ip4:<IPv4-osoite> ip6:<IPv6-osoite> -all"
```

SPF-tietueen oikeellisuus varmistetaan dig-komennolla TXT-parametrin kanssa, joka palauttaa palvelimen SPF-tietueen IPv4- ja IPv6-osoitteiden kanssa.

```
dig <palvelin> TXT
<palvelin> 3591  IN  TXT "v=spf1 a ip4:<ipv4-osoite> ip6:<ipv6-osoite> -all"
```

Testisähköpostiviestin lähetys mailx-komennolla kohdeosoitteeseen onnistuu palvelimella.

```
mailx -s “testi” <kohde sähköpostiosoite>
Tämä on testi.
```

.

3.2.8 Kohdelaitteiden valvonta

/etc/nagios/hosts-kansion alaisuuteen määritellään jokaiselle valvottavalle kohdelaitteelle oma tiedosto, josta selviää kohdelaitteen hostname, alias, ip-osoite jne. Check-host-alive-parametrilla Nagios pingaa valvontalaitetta, muussa tapauksessa valvontalaitteen oletetaan olevan toiminnassa ja saavutettavissa.

```
Define host{
  host_name                       raspi2
  alias                           Raspi2
  address                         10.8.0.22
  check_command                   check-host-alive
  check_interval                  1
  retry_interval                  1
  max_check_attempts              3
  check_period                    24x7
  contact_groups                  admins
  notification_interval           30
  notification_period             24x7
  notification_options            d,u,r
}
```

Valvottaville laitteille luodaan ryhmä (hostgroup) nimeltä raspit, johon liitetään kaikki samankaltaiset valvontalaitteet. Määrittelyt tehdään /etc/hostgroups/raspit.cfg-tiedostoon. Ryhmän jäsenet erotellaan pilkulla, kuten esimerkkikoodissa 15 on esitelty.

```
define hostgroup{
  hostgroup_name                  raspit
  alias                           raspit
  members                         raspi1,raspi2
  }
```

Esimerkkikoodi 15. raspit.cfg-tiedosto

Ryhmälle määritellään valvottava kosteus- ja lämpötilavalvontapalvelu (service). Kosteuden valvontapalvelu määritellään /etc/nagios/services/raspit_check-hum.cfg-tiedostossa.

```
define service{
  hostgroup_name                  raspit
  service_description             DHT22 humidity
  check_command                   check-dht22-hum!-C Olosuhd3 -w 60 -c 70
  check_interval                  1
  check_period                    24x7
  retry_interval                  1
```

```
  max_check_attempts            3
  notification_interval         60
  notification_period           24x7
  notification_options          w,c,u,r
  contact_groups                admins
}
```

Lämpötilan valvontaparametrit määritellään /etc/nagios/services/raspit_check-temp.cfg-tiedostossa.

```
define service{
  hostgroup_name                raspit
  service_description           DHT22 temperature
  check_command                 check-dht22-temp!-C Olosuhd3 -w 26 -c 30
  check_interval                1
  check_period                  24x7
  retry_interval                1
  max_check_attempts            3
  notification_interval         60
  notification_period           24x7
  notification_options          w,c,u,r
  contact_groups                admins
}
```

Valvonnan Check-dht-hum ja check-dht-temp-komennot määritellään /etc/nagios/commands/check-dht.cfg-tiedostossa.

```
# Check DHT22 humidity
define command{
        command_name    check-dht22-hum
        command_line    $USER1$/check_snmp -H $HOSTADDRESS$ $ARG1$ -P 2c -o
.1.3.6.1.3.1.2.2
}


# Check DHT22 temperature
define command{
        command_name    check-dht22-temp
        command_line    $USER1$/check_snmp -H $HOSTADDRESS$ $ARG1$ -P 2c -o
.1.3.6.1.3.1.2.3
}
```

Lopuksi Nagioksen määrittelytiedostot tarkastetaan ja palvelu käynnistetään uudelleen muutosten voimaan saamiseksi.

```
/opt/nagios/bin/nagios -v /etc/nagios/nagios.cfg
service nagios restart
```

Valvontalaitteet ovat toiminnassa ja Nagioksen valvonnassa, kuten kuvassa 9 on esitetty.

raspi2	DHT22 humidity	OK	01-23-2019 21:14:43	0d 1h 9m 40s	1/3	SNMP OK - 26
	DHT22 temperature	OK	01-23-2019 21:15:18	0d 1h 10m 25s	1/3	SNMP OK - 22

Kuva 6: Raspi2:n status

3.2.9 MRTG

Palvelimelle asennetaan MRTG yum-komennolla.

```
yum install mrtg
```

MRTG:lle määritellään seuraavat asetukset /etc/mrtg/mrtg.conf-tiedostoon.

```
HtmlDir: /var/www/html/mrtg
ImageDir: /var/www/html/mrtg
LogDir: /var/log/mrtg
ThreshDir: /var/log/mrtg
```

Websivustoa varten /var/www/mrtg-kansio siirretään /var/www/html/mrtg-kansioon ja httpd-palvelu käynnistetään uudelleen asetusten voimaan saamiseksi. MRTG-sivusto tiedustelee käyttäjätunnusta ja salasanaa, joka on sama kuin Nagioksen web-sivustolla.

```
Alias /mrtg /var/www/html/mrtg

<Location /mrtg>
    AllowOverride AuthConfig
    AuthType Basic
    AuthName "Password Required"
    AuthUserFile /etc/nagios/htpasswd.users
    AuthGroupFile /etc/nagios/htpasswd.groups
    require valid-user

</Location>
```

Valvontalaitteille luodaan /etc/mrtg/osv-kansio, jonne sijoitellaan kohdelaitteiden cfg-tiedostot, johon on määritelty OID-tunnukset, Olosuhd3-community ja raspi2-laitenimi, kuten esimerkkikoodi 16 on merkitty.

```
### Humidity

Target[raspi2_hum]: .1.3.6.1.3.1.2.2.0&.1.3.6.1.3.1.2.2.0:Olosuhd3@raspi2
MaxBytes1[raspi2_hum]: 25
MaxBytes2[raspi2_hum]: 65
Title[raspi2_hum]: Raspi2 | Hum
```

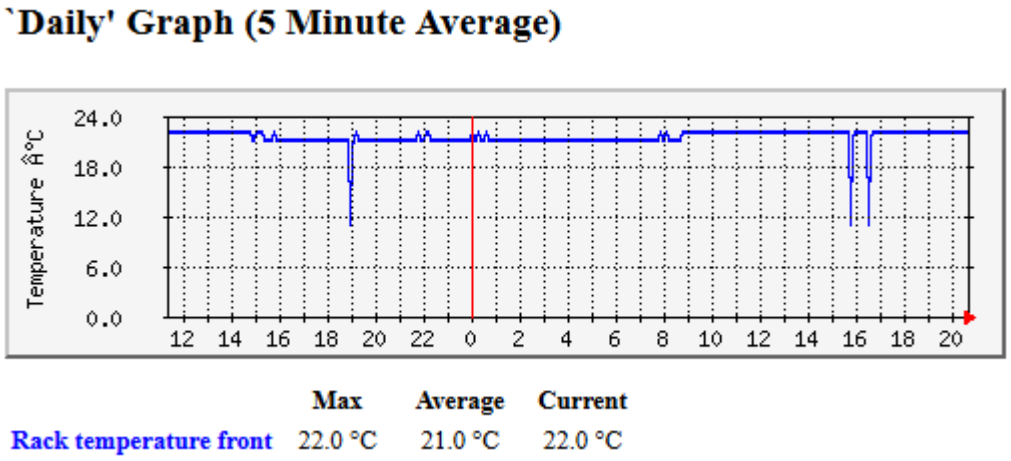
```
Options[raspi2_hum]: gauge, nopercent, growright, unknaszero, noi
YLegend[raspi2_hum]: Humidity %
ShortLegend[raspi2_hum]: %
Factor[raspi2_hum]: 1
Legend2[raspi2_hum]: Rack humidity in %
LegendO[raspi2_hum]: Rack humidity front
PageTop[raspi2_hum]: <H1>Rack Humidity of Raspi2 (3G)</H1>


### Temperature

Target[raspi2_temp]: .1.3.6.1.3.1.2.3.0&.1.3.6.1.3.1.2.3.0:Olosuhd3@raspi2
MaxBytes1[raspi2_temp]: 25
MaxBytes2[raspi2_temp]: 40
Title[raspi2_temp]: Raspi2 | Temp
Options[raspi2_temp]: gauge, nopercent, growright, unknaszero, noi
YLegend[raspi2_temp]: Temperature °C
ShortLegend[raspi2_temp]: °C
Unscaled[raspi2_temp]: n
Factor[raspi2_temp]: 1
Legend2[raspi2_temp]: Rack temperature in °C
LegendO[raspi2_temp]: Rack temperature front
PageTop[raspi2_temp]: <H1>Rack Temperature of Raspi2 (3G)</H1>
```

Esimerkkikoodi 16. Raspi2.cfg-tiedosto

MRTG suoritetaan cronilla viiden minuutin välein. Määrittely tehdään /etc/cron.d/mrtg-tiedostoon seuraavalla tavalla:

```
# Raspi2 3G
*/5 * * * * root LANG=C LC_ALL=C /usr/bin/mrtg /etc/mrtg/osv/raspi2.cfg --
lock-file /var/lock/mrtg/raspi2 –confcache -file /var/lib/mrtg/mrtg.ok
```

MRTG näyttää kosteus- ja lämpötilakuvaajat, kuten kuvassa 10 on esitettynä raspi2:n sensorin mittaama lämpötila. Kuvaajan muutamat notkahdukset johtuvat joko yhteysongelmasta tai lukuvirheestä.



Kuva 7: Raspi2:n ympäristön lämpötila

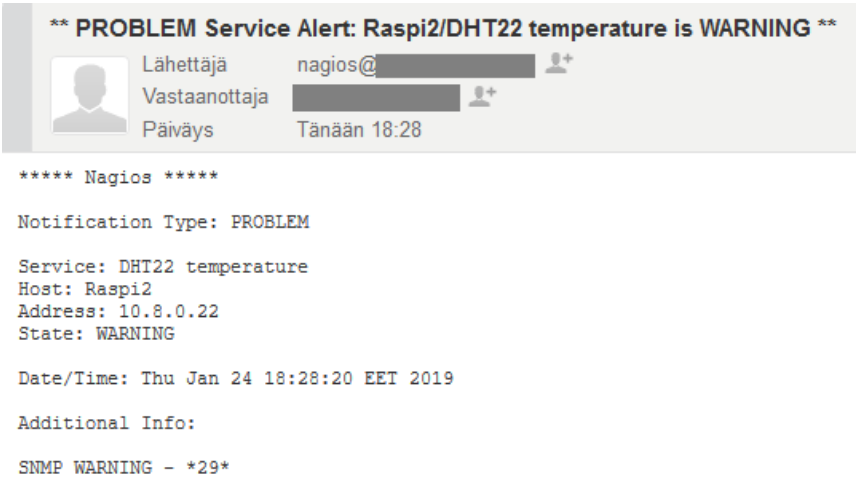
4 Testaus ja johtopäätökset

4.1 Testaus

Olosuhdejärjestelmän toimivuus testataan altistamalla DHT22-sensori kuumalle lämpötilalle. Järjestelmän toimivuutta seurattiin selaimella Nagios-palvelimen sivustolta ja sähköpostin varoitusviesteistä. Varoituksen hälytysraja on 27 °C ja kriittisen 30 °C. Palvelin tarkistaa kohonneen lämpötilan aina kolmesti minuutin välein ja tekee tämän jälkeen hälytyksen. Ensimmäinen testi suoritettaan 28-29 °C lämpötilassa varoitushälytyksen saamiseksi. Toinen testi suoritettaan yli 30 °C lämpötilassa kriittisen hälytyksen saamiseksi.

Lämpötilan noustessa 28 °C:een Nagios ilmoittaa kuvan 11 mukaisesti kohonneesta lämpötilasta. Sähköpostiviesti tuli hetki tämän jälkeen, kuvassa 12.

Kuva 8: Varoitus lämpötilan noususta yli raja-arvon sähköpostiviesti

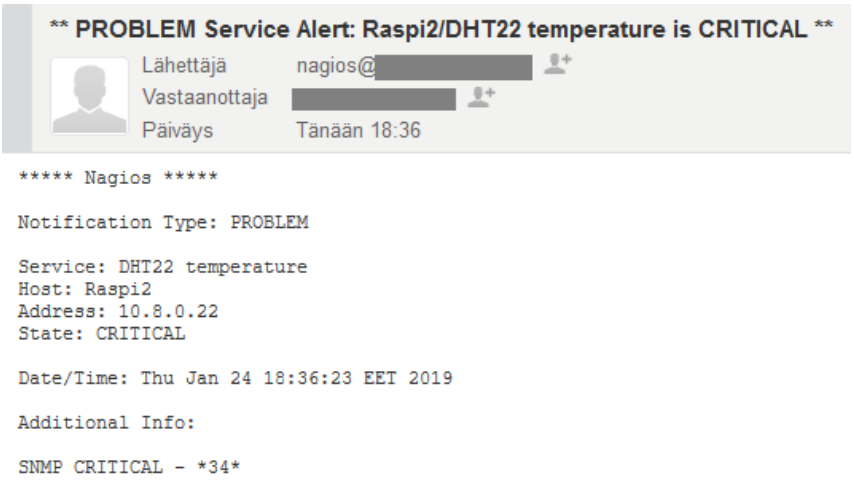


Kuva 9: Varoituksen sähköpostiviesti

Kun DHT22-sensorin lämpötilaa nostettaan yli 30 °C asteeseen, Nagios ilmoittaa kriittisestä lämpötilasta kuvan 13 mukaan. Lisäksi Nagios ilmoittaa asiasta sähköpostitse, kuten kuvassa 14 ilmenee.

raspi2	DHT22 humidity	OK	01-24-2019 18:35:43	0d 12h 16m 1s	1/3	SNMP OK - 18
	DHT22 temperature	CRITICAL	01-24-2019 18:36:18	0d 0h 0m 26s	3/3	SNMP CRITICAL - *34*

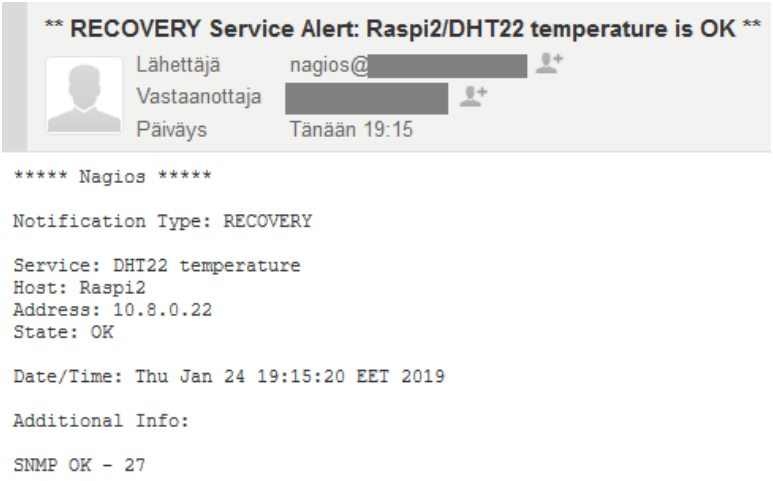
Kuva 10: Varoitus lämpötilan noususta yli kriittisen raja-arvon



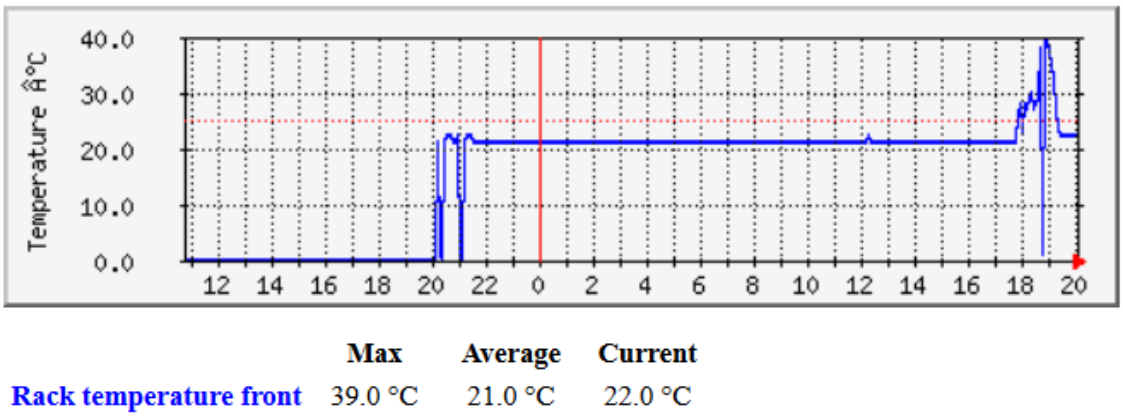
Kuva 11: Kriittisen varoituksen sähköpostiviesti

Lämpötilan laskiessa Nagios lähettää ilmoituksen toipumisesta, kuten kuvassa 15 on nähtävillä. Lämpötilatestaus sujui hyvin. Kaikki varoitusviestit ja toipumisilmoitukset saapuivat perille, ja Nagios reagoi odotetulla tavalla.

MRTG:n piirtämästä kuvaajasta kuvassa 16 pystytään havainnollistamaan testin kulku. Testin aikana sattui yksi virhelukema.



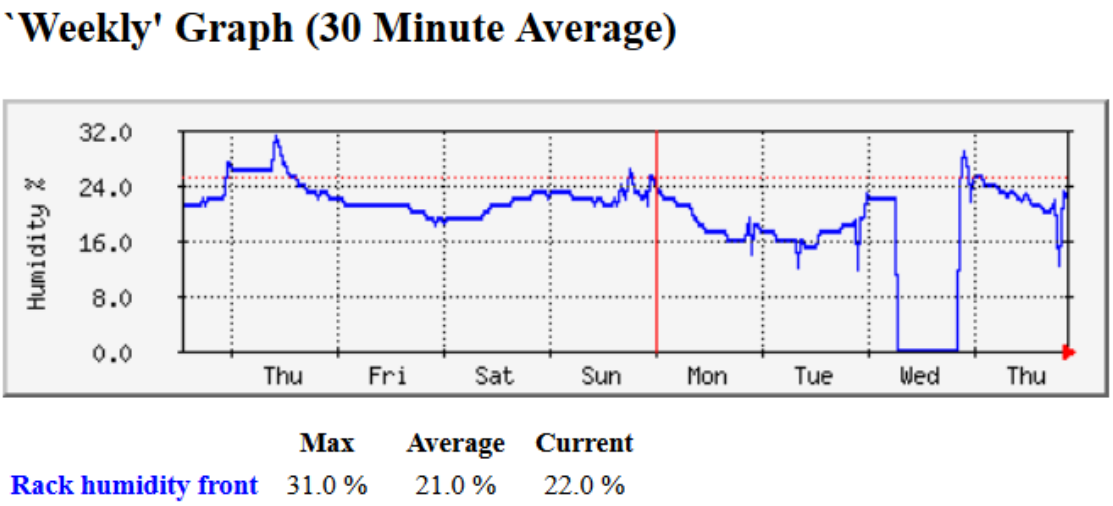
Kuva 12: Sähköpostiviesti lämpötilan palautumisesta normaalille tasolle



Kuva 13: MRTG:n lämpötilan kuvaajassa näkyy testaushetki klo 17.45-19.15

4.2 Lokitiedostot

MRTG tallentaa kuvaajia varten lukemia. Kuvaajista on havaittavissa päivän, viikon, kuukauden tai vuodenajalta lämpötila- ja kosteuslukemat. Kuvassa 17 on Raspi2:n viikon ajalta kertyneet kosteuslukemat. Kuvaajasta näkyy myös ajankohta, jolloin laite oli sammuneena.



Kuva 14: Raspi2:n kosteuskuvaaja viikon ajalta

4.3 Järjestelmän laajentaminen

Valvontalaitteeseen pystytään asentamaan useampia DHT22-sensoreita tai muita Raspberry Pi 3:n kanssa yhteensopivia antureita. Rajoituksen muodostaa sähkötehon ja kapasiteetin riittäminen sekä pinnipaikkojen määrä. Enimmillään Raspberry Pi 3:een voidaan liittää kaksi 3.3 V:n sensoria ja kaksi 5 V:n sensoria.

Olosuhdevalvonnan MIB-taulukko on suunniteltu siten, että sitä pystytään laajentamaan eri tyyppisille sensoreille. Taulukon laajentaminen onnistuu käyttämällä jo olemassa olevaa OID-tunnistetta tai luomalla uuden OID-tunnisteen. Valvontalaite vaatii snmpd.conf-tiedostoon tarvittavat muutokset ja sensorin käyttöä varten mahdolliset skriptit ja kirjastot. Sensorin antamista lukemista on mahdollista piirtää kuvaaja MRTG:llä.

4.4 Kustannusarvio

Insinöörityön antajan mukaan valvontalaitteen enimmäisarvo saa olla 200 € konesalitilaa kohden. Yksi laite riittänee yhteen konesalitilaan. Raspberry Pi 3:n varusteineen saa hankittua noin 70-90 €:lla (alv 0%), 3G-modeemi maksaa noin 30-40 €, DHT22-sensorin hinnaksi arvioidaan tulevan noin 10-20 €. Lisävarustukset maksavat noin 30 €. Yhteishinta-arvio on 140-180 € laitetta kohti. Tähän lisätään vielä 3G-liittymä ja postikustannukset. Kilpailutuksella ja suurella sisäänostomäärällä hankinnan arvoa saadaan pienennettyä, mikä tekee laitteesta toteutuskelpoisen. [21.]

4.5 Johtopäätökset

Olosuhdevalvontaratkaisuksi ehdotettu järjestelmä on testien perusteella toimiva ja sitä voidaan käyttää konesalitilojen valvomiseen. Järjestelmä hälyttää raja-arvojen ylittävistä kosteus- ja lämpötilalukemista. Lisäksi valvontalaitteen hankintahinta on kohtuullinen, eikä sen arvioida ylittävän budjettia.

Valvonta-palvelimen VPN-yhteys on toimiva lukuun ottamatta muutamia paketin häviämisiä UDP-pohjaisessa tunnelissa. Valvontalaitteella oli aikaisemmin ongelmia 3G-mokkulan yhteyden ylläpitämisessä. Recovery-skriptin lisäämisen jälkeen valvontalaitteen yhteys ei ole katkennut siten, että se olisi jäänyt ulottumattomiin, vaan valvontalaite on kyennyt toipumaan yhteyden menetyksestä täysin. Näistä kerroista Nagios ei ole ehtinyt tekemään ilmoituksia laitteen kaatumisesta, mikä on ollut haluttu tahtotila.

DHT22-sensori on testausten perusteella toimiva ja tarkka-arvoinen. Sensorin lukemia on verrattu kaupasta ostettuun langattomaan lämpö- ja kosteusmittariin (Art. No 7340, K-Rauta, arvo noin 40 €). Lukemat ovat lähes identtiset.

4.6 Jatkotoimenpiteet ja kehitysehdotukset

Jatkotoimenpiteinä SNMP-palvelun käyttämää MIB-taulukkoa varten olisi hyvä hakea yksityisen sektorin 1.3.6.1.4.1 -OID-tunnistetta. Insinöörityön MIB-taulukko on rakennettu

testaukseen käytettävällä .1.3.6.1.3 -OID-tunnisteella, eikä sitä suositella otettavan käyttöön tuotantoympäristöön. Organisaatiokohtaisia OID-tunnuksia pystyy anomaan ilmaiseksi IANA:n web-sivustolta.

Kehitysehdotuksena Nagioksen voisi asentaa sopivasta repositorysta, koska repositoryn käyttö helpottaisi päivityksien ja mahdollisesti valmiiden SELinux-asetusten saamisen. Nagios Core 4 (viimeisin versio 4.4.2, julkaistu 15.1.2019) on saatavilla EPEL-repositorysta. Tämän vaihtoehdon käyttöä tulisi harkita huolella, sillä pakettien päivittyminen ei välttämättä tule aina olemaan ajan tasalla. Kaikkein paras ratkaisu ylläpidon kannalta olisi asentaa kaupallinen Nagios XI, jolle löytyy oma repository https://repo.nagios.com/ -sivulta. Tällöin versiopäivityksien voidaan olettaa olevan lähes heti saatavilla.

Valvontalaitteiden asentamisessa ja ylläpidossa voisi hyödyntää Ansiblea. Ansible on ilmainen Linux-järjestelmien ylläpitoa varten tehty automatisaatiotyökalu. Valvontalaitteen sd-kortille siirrettäisiin sopiva Rasbian-image, johon on ennalta asennettu vähintään Ansiblen kanssa yhteensopiva Python (2.7 tai 3.5) ja toimivan SSH-yhteyden parametrit. Tämän jälkeen Ansiblea voidaan käyttää valvontalaitteen asennuksen viimeistelyyn.

Lähteet

1 Douglas R. Mauro & Kevin J. Schmidt. 2005b. Essential SNMP. O'Reillys.

2 A History of The Raspberry Pi. <http://novadigitalmedia.com/history-raspberry-pi/>. 4.3.2015. Luettu 26.11.2018.

3 Featured Products. <www.raspberrypi.org/products>. Luettu 26.11.2018.

4 Raspberry Pi 3 Model B. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Luettu 26.11.2018.

5 The Best Operating Systems for Your Raspberry Pi Projects. <https://lifehacker.com/the-best-operating-systems-for-your-raspberry-pi-project-1774669829>. 5.5.2016. Luettu 26.11.2018.

6 Wide range of Hydrometers: DHT22, AM2302, AM2320, AM2321, SHT71, HTU21D, Si7021, BME280. <http://www.kandrsmith.org/RJS/Misc/Hygrometers/calib_many.html>. 26.2.2018. Luettu 16.12.2018.

7 Temperature and humidity module AM2303 Product Manual. <https://akizukidenshi.com/download/ds/aosong/AM2302.pdf>. Luettu 16.12.2018.

8 Digital relative humidity & temperature sensor AM2302/DHT22. <https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>. Luettu 16.12.2018.

9 Wojciech Kocjan. 2014b. Learning Nagios 4. PACKT publishing open source.

10 Get to known Nagios. <www.nagios.com/about-nagios-enterprises>. Luettu 26.11.2018.

11 Nagios. <https://en.wikipedia.org/wiki/Nagios>. Luettu 26.11.2018.

12 Tobi Oetiker's MRTG. <https://oss.oetiker.ch/mrtg/> 3.5.2017. Luettu 26.11.2018.

13 What are the fundamental differences between bridging and routing in terms of configuration? <https://openvpn.net/faq/what-are-the-fundamental-differences-between-bridging-and-routing-in-terms-of-configuration/>. Luettu 5.3.2019.

14 Why down OpenVPN use UDP and TCP? <https://openvpn.net/faq/why-does-openvpn-use-udp-and-tcp/>. Luettu 5.3.2019.

15 2x how to. <https://openvpn.net/community-resources/how-to/>. Luettu 5.3.2019.

16 OpenVPN Introduction. <https://community.openvpn.net/openvpn/wiki/HOWTO>. Luettu 5.3.2019.

17 Diffie-Hellman Key Agreement Protocol. <https://linuxconfig.org/vpn-virtual-private-network-and-openvpn#h16-diffie-hellman-key-agreement-protocol>. Luettu 5.3.2019.

18 Hardening OpenVPN. <https://community.openvpn.net/openvpn/wiki/Hardening>. Luettu 5.3.2019.

19 Wvdial. <https://www.linux.fi/wiki/Wvdial>. Luettu 12.3.2019.

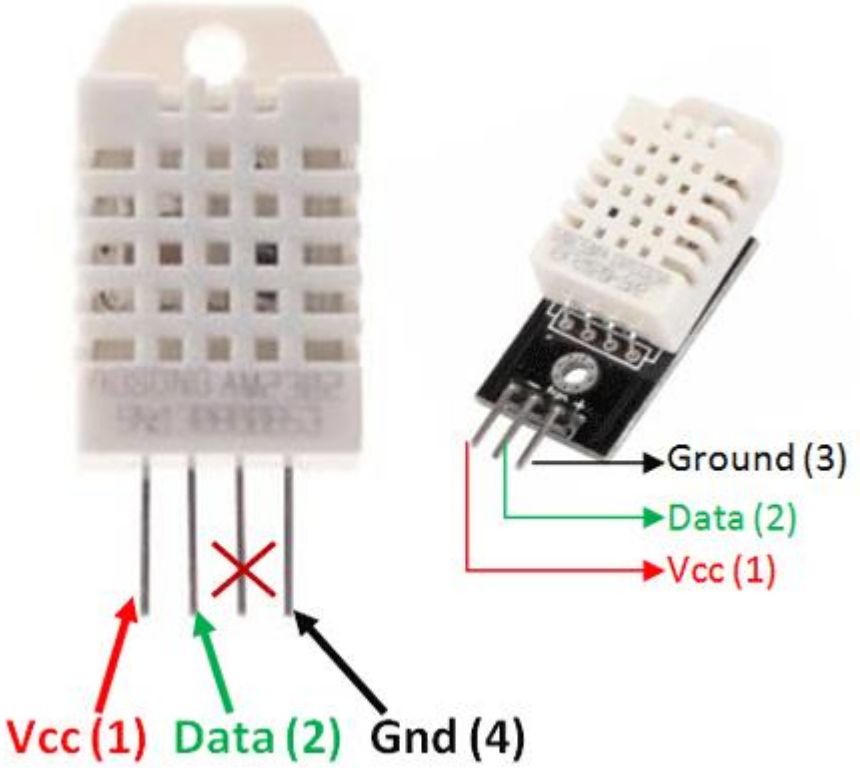
20 GitHub. <https://github.com/donaldbales/ecospec/blob/master/etc.init.d.wvdial>. Luettu 26.11.2018.

21 Dustin. <www.dustin.fi>. Luettu 24.1.2019.

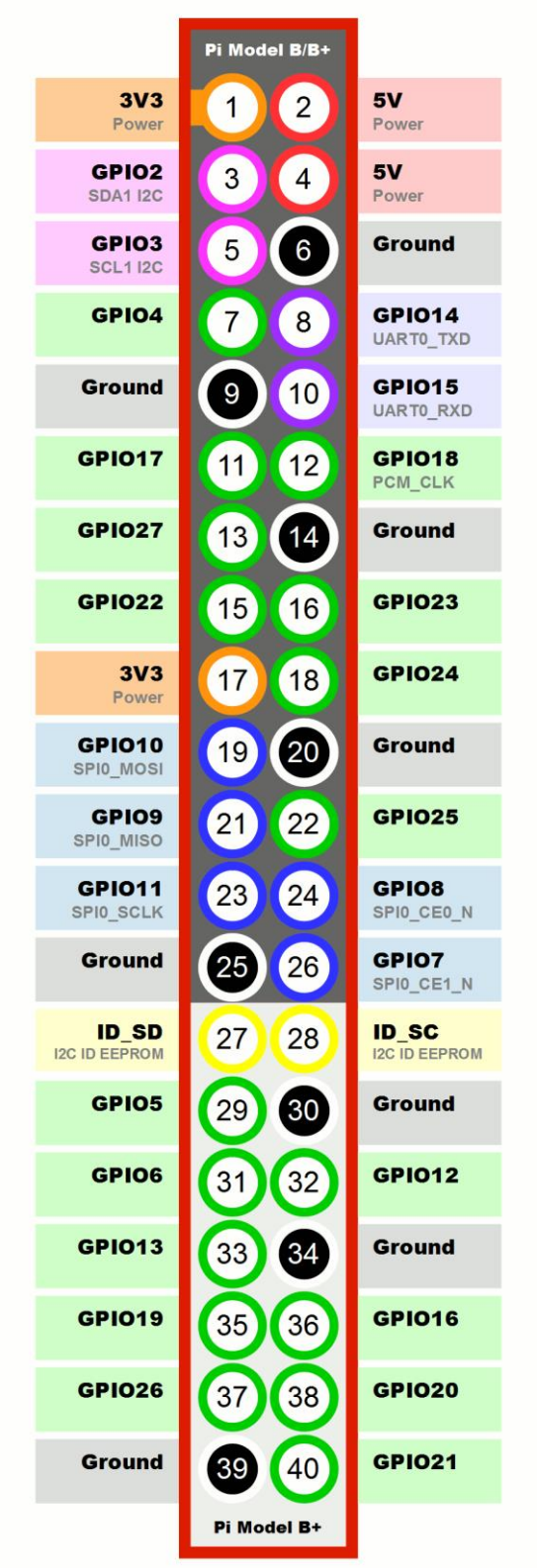
22 DHT22 - Temperature and Humidity Sensor. <https://components101.com/sensors/dht22-pinout-specs-datasheet>. Luettu 26.11.2018

23 Simple Guide to the Raspberry Pi GPIO Header and Pins. <https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/#prettyPhoto>. Luettu 26.11.2018.

DHT22-sensorin pinnit [22.]



Raspberry Pi 3 Model B GPIO-pinnit [23.]



PRI-RACKMON-MIB.txt

```
RPI-RACKMON-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Gauge32, Unsigned32, experimental
    FROM SNMPv2-SMI
    DisplayString       FROM SNMPv2-TC
;

rpiRackmon MODULE-IDENTITY
    LAST-UPDATED        "201801210000Z"
    ORGANIZATION        "www.valtori.fi"
    CONTACT-INFO
         " email:       jonkun@sähköposti"
    DESCRIPTION
         "A MIB file for RPI DHT11/22 sensors."
    REVISION            "201801210000Z"
    DESCRIPTION         "Third Release"
    REVISION            "201712030000Z"
    DESCRIPTION         "Second Release"
    REVISION            "201711160000Z"
    DESCRIPTION         "First Release"
    ::= { experimental 1 }

rpiRackmonDht11         OBJECT IDENTIFIER ::= { rpiRackmon 1 }

rpiRackmonDht11Sensor OBJECT-TYPE
  SYNTAX                DispalyString (SIZE (0..255))
  MAX-ACCESS            read-only
  STATUS                current
  DESCRIPTION           "DHT11 Digital humidity and temperature sensor"
  ::= { rpiRackmonDht11 1 }


rpiRackmonDht11Hum OBJECT-TYPE
  SYNTAX                Gauge32
  MAX-ACCESS            read-only
  STATUS                current
  DESCRIPTION           "DHT11 humidity value (%)."
  ::= { rpiRackmonDht11 2 }

rpiRackmonDht11Temp OBJECT-TYPE
  SYNTAX                Gauge32
  MAX-ACCESS            read-only
  STATUS                current
  DESCRIPTION           "DHT11 temperature value (celsius)."
  ::= { rpiRackmonDht11 3 }

rpiRackmonDht22         OBJECT IDENTIFIER ::= { rpiRackmon 2 }

rpiRackmonDh22Sensor OBJECT-TYPE
  SYNTAX                DisplayString (SIZE (0..255))
  MAX-ACCESS            read-only
  STATUS                current
  DESCRIPTION           "DHT22 humidity and temperature sensor"
  ::= { rpiRackmonDht22 1 }

rpiRackmonDht22Hum OBJECT-TYPE
  SYNTAX                Gauge32
  MAX-ACCESS            read-only
```

```
  STATUS                current
  DESCRIPTION           "DHT22 humidity value (%)."
  ::= { rpiRackmonDht22 2 }

rpiRackmonDht22Temp OBJECT-TYPE
  SYNTAX                Gauge32
  MAX-ACCESS            read-only
  STATUS                current
  DESCRIPTION           "DHT22 temperature value (celcius)."
  ::= { rpiRackmonDht22 3 }


END
```