

Domoticz-kotiautomaatiojärjestelmä: Rakennusautomaatiolaboratorion laite- ja käyttöharjoitus

Niko Karhunen

Opinnäytetyö
Joulukuu 2017
Teollisuustekniikka
Insinööri (AMK), automaatiotekniikan tutkinto-ohjelma

| | | |
|--|-------------------------------------|------------------------------------|
| Tekijä(t) Karhunen, Niko | Julkaisun laji Opinnäytetyö, AMK | Päivämäärä Joulukuu 2017 |
| | Sivumäärä 149 | Julkaisun kieli Suomi |
| | | Verkkojulkaisulupa myönnetty: x |
| Työn nimi Domoticz-kotiautomaatiojärjestelmä: Rakennuslaboratorion laite- ja käyttöharjoitus | | |
| Tutkinto-ohjelma automaatiotekniikan tutkinto-ohjelma | | |
| Työn ohjaaja(t) Olli Väänänen, Markku Ström | | |
| Toimeksiantaja(t) Jyväskylän ammattikorkeakoulu | | |
| Tiivistelmä <p>Automaatioteknologian ratkaisuja tekevät yritykset ovat huomanneet helposti muokattavien avoimen lähdekoodin toteutuksien edut, kun niitä verrataan isojen laitevalmistajien tarjoamiin kokonaisratkaisuihin. Tähän liittyen Jyväskylän ammattikorkeakoulu tilasi opinnäytetyönä rakennusautomaatiolaboratorioon opiskelijoiden käytettäväksi laboratorioharjoituksen, joka soveltaa avoimen lähdekoodin ohjelmistoja ja laitteistoja.</p> <p>Työhön valittiin käytettäväksi Domoticz-kotiautomaatio-ohjelmisto. Käyttöliittymään yhdistettiin laajalle levinnyt kuluttaja-asiakkaille suunnattu Z-Wave-järjestelmä sekä teollisuudessa yleisesti käytetty mittainstrumenttiväylä Modbus RTU. Z-Wave valittiin näyttämään koti- ja rakennusautomaation mahdollisuudet ja Modbus-väyläesimerkki mallintamaan vaihtoehtoista toteutusta perinteisille teollisuuden laiteratkaisuille. Tavoitteena oli kartoittaa mainittujen järjestelmien toimivuus ja mahdollisuudet annetussa laajuudessa.</p> <p>Työhön kuului opiskelijaharjoitukseen sopivien laitekokonaisuuksien valinta, niiden kytkentä ja asennus sekä osakokonaisuuksien liittäminen yhden käyttöliittymän alle. Työssä tutustuttiin valittujen laitteiden ja järjestelmien toimintaan sekä niiden käyttömahdollisuuksiin. Työ toi esiin useita avoimen lähdekoodin toteutuksille tyypillisiä ongelmia, joihin etsittiin samalla vaihtoehtoisia ratkaisuja.</p> <p>Työn tuloksena syntyi kattava laboratorioharjoituskokonaisuus, jonka avulla rakennusautomaatiokursseilla kaikenlaiset opiskelijat voivat tutustua avoimen lähdekoodin mahdollisuuksiin erilaisissa käyttöympäristöissä.</p> <p>Harjoitusmateriaali hyväksyttiin toimeksiantajan puolesta suorittamalla ohjeistuksen mukainen laitteiston asennus.</p> | | |
| Avainsanat (asiasanat) Modbus RTU, RS-485, Z-Wave, sarjaliikenne, Raspberry Pi, Arduino, MySensors. | | |
| Muut tiedot (salassa pidettävät liitteet) | | |

| | | |
|--|--|--|
| Author(s) Karhunen, Niko | Type of publication Bachelor's thesis | Date December 2017 Language of publication: Finnish |
| | Number of pages 149 | Permission for web publication: x |
| Title of publication Domoticz homeautomation HMI Exercise for building automation laboratory | | |
| Degree programme Electrical and Automation Engineering | | |
| Supervisor(s) Väänänen Olli, Ström Markku | | |
| Assigned by JAMK University of Applied Sciences, Flyktman Teppo | | |
| Abstract <p>Companies offering automation solutions have noticed increasing benefits from choosing various open source applications over standard automation industry equipment and software providers in non-critical implementations. To reflect on this shift in opinion, JAMK University of Applied Sciences assigned a Bachelor's thesis study using readily available open source applications.</p> <p>The main platform chosen to be used was an open source home automation interface called Domoticz, into which an industry standard instrument serial bus Modbus RTU and a commercial application solution Z-Wave were to be attached – both approaching the same agenda of home and building automation from different aspects. The end goal was to assess and determine the value of open source solutions in the given scale.</p> <p>The research was carried out by choosing and installing appropriate equipment for student exercise and by combining those applications under one user interface. The work also involved studying given information on the connected devices and used protocols. The combined work process brought up some unpredicted difficulties when using open source solutions at given scale and forced to find alternative solutions.</p> <p>As the result of the research, a comprehensive student exercise portfolio was compiled. These exercises were tested and approved for use by the department of Building Automation.</p> | | |
| Keywords/tags (subjects) Modbus RTU, RS-485, Z-Wave, Serial interface, Raspberry Pi, Arduino, MySensors. | | |
| Miscellaneous (Confidential information) | | |

Sisältö

| | | |
|----------|--|-----------|
| 1 | Opinnäytetyön lähtökohdat | 7 |
| 2 | Opinnäytetyön toteutus..... | 7 |
| 3 | Rakennusautomaatio yleisesti..... | 8 |
| 4 | Sarjaliikenneviestintä..... | 10 |
| | 4.1 Asynkroninen sarjaviestintä | 11 |
| | 4.2 RS-485-standardi | 12 |
| 5 | Modbus-sarjaliikenneprotokola | 13 |
| | 5.1 Modbus RTU -väylän viestirakenne..... | 14 |
| | 5.2 Modbus-funktiot | 15 |
| | 5.3 Käytettävät Modbus RTU-väylälaitteet | 16 |
| | 5.3.1 HK Instruments: CDT-MOD-2000-hiilidioksidilähetin..... | 16 |
| | 5.3.2 Davis Vantage Pro 2 -sääasema..... | 17 |
| 6 | Z-Wave-langatonviestintäprotokolla | 18 |
| | 6.1 Toimintaperiaate | 18 |
| | 6.2 Topologia ja viestintä | 19 |
| | 6.3 Z-Way-käyttöohjelmisto | 20 |
| | 6.4 Z-Way: RaZberry2-isäntäohjain..... | 20 |
| | 6.5 Käytettävät Z-Wave-laitteet..... | 21 |
| | 6.5.1 AEOTEC: Multisensor 6 | 21 |
| | 6.5.2 Philio-PAN11-1B: Älykäs energiamittari ja kytkin..... | 22 |

| | | |
|-----------|---|-----------|
| 7 | Domoticz-kotiautomaatiokäyttöliittymä | 22 |
| 8 | MySensors-laiteverkko..... | 23 |
| 9 | Graafinen ohjelmointikieli Blockly..... | 25 |
| 10 | Raspberry Pi -mikrotietokone..... | 25 |
| | 10.1 Raspbian-käyttöjärjestelmä | 26 |
| 11 | Arduino-kehitysalusta | 27 |
| | 11.1 Arduino Nano -kehitysalusta..... | 27 |
| | 11.2 Arduino IDE -ohjelmistopaketti..... | 28 |
| | 11.2.1 ModbusMaster-kirjasto | 28 |
| | 11.3 RS-485 (MAX485) Arduino -ohjainkortti. | 29 |
| 12 | Työn toteus | 30 |
| | 12.1 Laboratorioharjoituksen laitekytkennät | 31 |
| | 12.2 Rasbian-käyttöjärjestelmän asennus | 34 |
| | 12.3 Rasbian-käyttöjärjestelmän ohjelmistoasennukset..... | 34 |
| | 12.3.1 Domoticz-käyttöliittymän asennus..... | 35 |
| | 12.3.2 Arduino IDE -ohjelmiston asennus | 36 |
| | 12.3.3 Verkkoselaimen asennus | 37 |
| | 12.4 Arduino-Modbus-sillan ohjelmointi ja liittäminen Domoticz- käyttöliittymään | 38 |
| | 12.4.1 Arduino-Modbus-sillan ohjelmointi | 38 |
| | 12.4.2 Arduino-Modbus-sillan liittäminen Domoticz-käyttöliittymään..... | 41 |
| | 12.4.3 Z-Wave-laiteverkon asennus Domoticz-käyttöliittymään..... | 42 |
| | 12.5 Domoticz-käyttöliittymän tapahtumaohjelmointi | 42 |
| | 12.6 Työn toteutuksessa ilmenneitä ongelmia | 43 |

| | |
|---|-----------|
| 13 Johtopäätökset ja pohdinta..... | 45 |
| Lähteet | 49 |
| Liitteet | 52 |
| Liite 1. Rasbian käyttöjärjestelmän asennusvalmistelut..... | 52 |
| Liite 2. Rasbian manuaaliasennus Raspberry Pi'lle, täysversio | 56 |
| Liite 3. Rasbian NOOBS-asennus Raspberry Pi'lle, täysiversio | 59 |
| Liite 4. Rasbian Lite-version valmistelu ja asennus | 62 |
| Liite 5. Rasbian Lite-version ohjelmistoasennukset | 70 |
| Liite 6. Rasbian-ohjelmistojen asennus..... | 81 |
| Liite 7. Domoticz-käyttöliittymä: Z-Waven asennus | 87 |
| Liite 8. Arduino-Modbus-sillan kytkennät ja ohjelmointi | 97 |
| Liite 9. Domoticz-käyttöliittymä: Arduino-modbus-sillan asennus..... | 109 |
| Liite 10. Domoticz-käyttöliittymä: Blockly-tapahtumaohjelmointi | 119 |
| Liite 11. Domoticz-käyttöliittymä: LUA-tapahtumaohjelmointi..... | 128 |
| Liite 12. Arduino-Modbus-silta: ohjelmakoodi kokonaisuudessaan | 134 |

Kuviot

| | |
|--|----|
| Kuvio 1. Laitteistoharjoituksen kokoonpano | 8 |
| Kuvio 2. Rakennusautomaation hierarkia | 9 |
| Kuvio 3. Esimerkki asynkronoidusta sarjaviestistä | 12 |
| Kuvio 4. Bittien siirtäminen RS-485-väylässä | 13 |
| Kuvio 5. Modbus RTU-viestikehys | 14 |
| Kuvio 6. Modbus ASCII- ja RTU viestien vertailu | 15 |
| Kuvio 7. HK Instruments -hiilidioksidilähetin: CDT-MOD-2000..... | 16 |
| Kuvio 8. Davis Vantage Pro 2 -sääaseman ohjainpaneeli | 17 |
| Kuvio 9. Davis Vantage Pro 2 -sääaseman ulkoilman mittauslaitteisto..... | 17 |
| Kuvio 10. Davis Vantage Pro 2 -sääaseman Modbus RTU -väyläadapteri | 18 |
| Kuvio 11. Yksinkertaistettu malli Z-Wave-topologiasta | 19 |
| Kuvio 12. RaZberry2-lisäkortti Raspberry Pi-mikrotietokoneelle | 21 |
| Kuvio 13. Aeotec Multisensor 6 -mittalaite | 22 |
| Kuvio 14. Philio PAN11-1B -energiamittari ja kytkin | 22 |
| Kuvio 15. MySensors-isäntäohjain | 24 |
| Kuvio 16. MySensors-laitepiste-esimerkki | 24 |
| Kuvio 17. Blockly-käyttöliittymä..... | 25 |
| Kuvio 18. Raspberry Pi 3 Model B -kehitysalusta | 26 |
| Kuvio 19. Rasbian-käyttöjärjestelmän Jessie-jakeluversion työpöytä..... | 26 |
| Kuvio 20. Arduino Nano. | 27 |
| Kuvio 21. ModbusMaster-ohjelmakoodin esimerkki..... | 29 |
| Kuvio 22. RS-485-väyläadapteri. | 29 |
| Kuvio 23. Raspberry Pi'n kytkennät | 31 |

Taulukot

| | |
|---|----|
| Taulukko 1. Modbus-yleisfunktio- taulukko | 16 |
| Taulukko 2. Arduino Unon ja Nanon vertailu..... | 28 |

Käsitteitä

| | |
|------------------|--|
| Anturi | Mitattavassa suuressa tapahtuvien muutoksien sähköinen viestintälaitte |
| ASCII | American Standard Code Information Interchange. 256 merkin toimintaohjeistus prosessorien hallintaan |
| Avoin lähdekoodi | Ohjelmakoodi ja/tai ohjelmistokokonaisuus joiden muok- kausta, käyttöä ja levitystä ei ole rajoitettu |
| Baud | Tiedonsiirtonopeuden suure. lähetettävien bittien määrä jaettuna lähetysajalla, bits/second. |
| CRC | Cyclic Redundancy Check – Modbus-väylän viestinvarmen- nussumma |
| Debug | Ohjelmoinnissa käytetty vianselvitysprosessi. |
| GPIO | General Purpose Input Output. Tarkoittaa tietotekniikassa vapaasti ohjelmoitavia kytkentäpisteitä. |
| Hex | Heksadesimaali, tietotekniikassa yleisesti käytetty numeroin- tijärjestelmä. |
| HTML 5 | Hypertext Markup Language, versio 5. Elementeistä muo- dostettua rakenteellista ohjelmointikoodia. |
| HTTP(S) | Hypertext Transfer Protocol (Secure) |
| ID | Identifier, yksilöintitunnus. |
| IP | Internet Protocol |
| Käyttöliittymä | Käyttäjän ja toimintojen välinen hallintarajapinta |
| Linux | Windows-käyttöjärjestelmään verrattavissa oleva avoimen lähdekoodin käyttöjärjestelmäydin. |
| Lux | Valaistusvoimakkuuden mittasuure, SI-järjestelmän mittayk- sikkö 1/lx |

| | |
|-------------------|--|
| LVIS-järjestelmät | Yleinen lyhennys lämpö-, vesi-, ilma- ja sähköjärjestelmien kokonaisuudesta. |
| NOOBS | New Out Of the Box Software. Suoraan käyttöön otettava ohjelmisto. Tavoitteena yksinkertaistaa ohjelmistojen asennus |
| OpenSSL | Vapaan lähdekoodin tietoturvakirjasto Secure Socket Layer ja Transport Layer Security protokollille. |
| ppm | Parts per million. partikkelia miljoonasta. |
| RTU | Remote Terminal Unit |
| TTL | Transistor–transistor logic, 5 voltin transistorilogiikka. |

1 Opinnäytetyön lähtökohdat

Automaatioteknologian alalla on hiljalleen herännyt mielenkiintoa korvata kalliita, laajalti tunnettuja automaatoratkaisuja kustannustehokkaammilla vaihtoehdoilla. Yksi tapa on korvata ei-kriittisissä kohteissa toimittajien valmistajakohtaiset automaatiosovellukset avoimen lähdekoodin ratkaisulla. Avoimen lähdekoodin käyttöliittymät ovat kaikkien vapaasti käytettävissä eikä ohjelmistokoodia ole lukittu kalliin lisenssin tai vaikeasti tavoitettavan teknisen tuen taakse. Tällainen lähestymistapa avaa mahdollisuuksia pienissä ja keskisuurissa sovelluksissa, joissa koodia voi muokata vapaasti käyttötarkoitusta varten.

Jyväskylän ammattikorkeakoulun rakennusautomaatiolaboratorio tilasi opinnäytetyön aiheeseen liittyen. Voidaanko teollisia automaatoratkaisuja korvata avoimen lähdekoodin kotiautomaatiosovelluksella? Tähän kysymykseen etsittiin ratkaisua hyödyntämällä avoimen lähdekoodin Domoticz-kotiautomaatiokäyttöliittymää.

Domoticz on pienen kehitystiimin luoma ilmainen käyttöliittymä. Siinä on tuki useille käyttöympäristöille (Raspberry Pi, Windows, Linux, Mac OS, sulautetut järjestelmät). Se tukee myös useita muita avoimen koodin anturiverkkoja. Kehitystiimin tarkoituksena on ollut luoda helposti lähestyttävä ympäristö koti- ja rakennusautomaation tarkoituksiin.

Raportissa tutustutaan Domoticzin käyttömahdollisuuksiin laitteistoesittelyn ja sen pohjalta tehdyn laboratorioharjoituksen kautta.

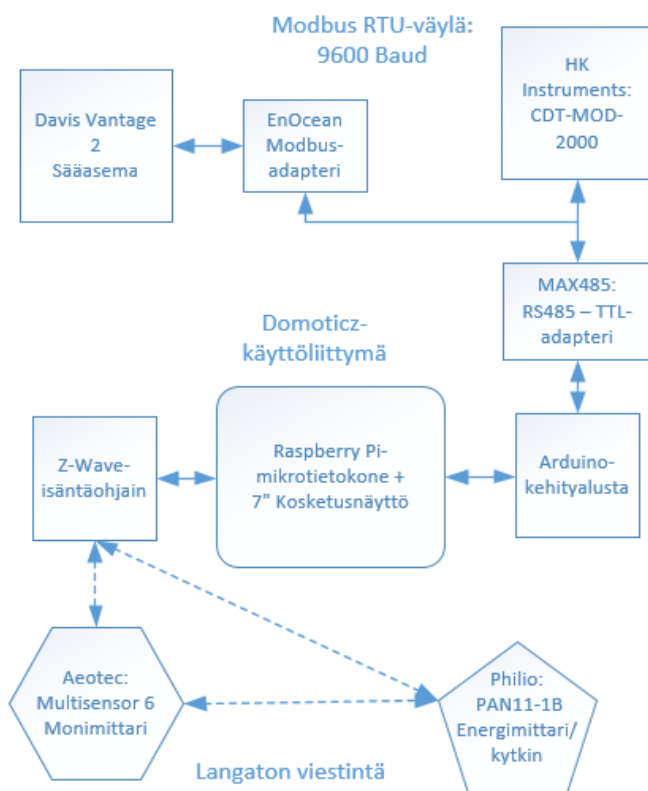
2 Opinnäytetyön toteutus

Jyväskylän ammattikorkeakoulun rakennusautomaatiolaboratorioon tehtävä laitteistoesittely päädyttiin rajamaan kahteen hyvinkin erilaiseen laitteistoon: Z-Wave-tekniologia on tarkoitettu kotiautomaatiosta kiinnostuneille arkipäivän kuluttajille. Modbus RTU -protokollalla toimivat laitteet ovat taas laajalti teollisuus- ja rakennusautomaation käyttämä ratkaisu erinäisissä mittaustarpeissa.

Molempien laitteistojen ohjauksesta vastaa laajalti harrastekäytössä oleva avoimen lähdekoodin Raspberry Pi -mikrotietokone, jonka päälle Domoticz-käyttöliittymä asennetaan. Z-Wave on Domoticzin luonnostaan tukema protokolla, joka tarvitsee

toimiakseen erikseen hankittavan Z-Wave-yhdyskortin. Modbus-sarjaliikennevies-tintä ei ole Domoticziin sisäänrakennettu ominaisuus, ja tätä varten laitteistoestitel-lyyn pitää rakentaa Arduino-kehitysalustan päälle Modbus-usb-silta, jolla Modbus RTU -väyläliikenne muunnetaan Raspberry Pi'n tuloporttiin Domoticz-sovelluksen ymmärtämään muotoon.

Kuvion 1 mukaisella laitteistolla pyrittiin selvittämään, miten eri rajapintojen yhteen-sovittaminen onnistuisi. Laitteasennusten ja toimintatestauksien pohjalta luotiin opis-kelijoille tarkoitettu harjoituskokonaisuus, jolla oppinäytetyön tavoitteita pystyttiin konkreettisesti havainnollistamaan.



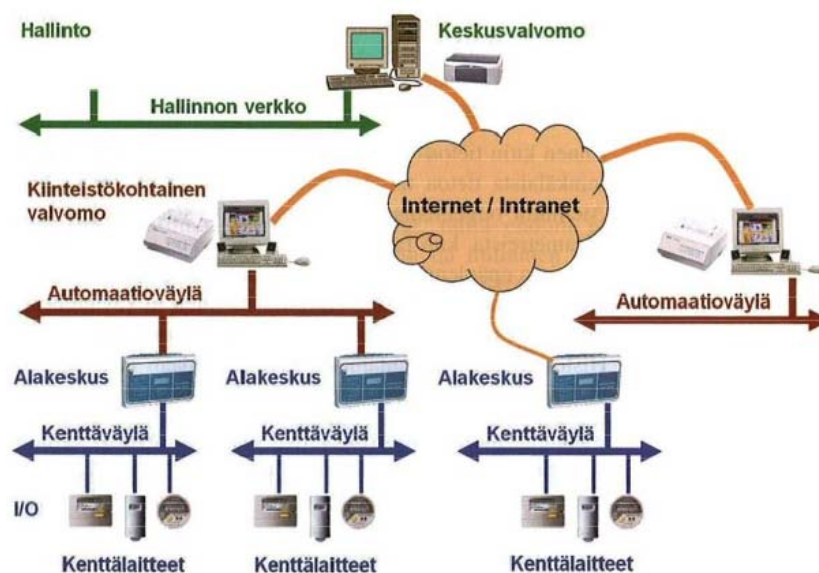
Kuvio 1. Laitteistoharjoituksen kokoonpano.

3 Rakennusautomaatio yleisesti

Nykyaikainen rakennusautomaatio käsitteenä sisältää kattavan laitteisto- ja ohjaus-järjestelmän, joilla pyritään automatisoimaan rakennussuunnittelun määrittelemät

LVIS-järjestelmät (Lämpö, Vesi, Ilma ja Sähkö). Kunkin osa-alueen yhtenäinen hallinta tarkoituksenmukaisella automaatiolla edesauttaa ylläpitämään esimerkiksi suunniteltuja energia- ja sisäilmatavoitteita. Rakennusautomaatio sisältää myös kasvavassa määrin kulunvalvonta- ja turvallisuusjärjestelmiä. (Vesänen 2016)

Rakennusautomaatiojärjestelmä voidaan jaotella kolmeen osa-alueeseen (ks. kuvio 2). Alimmalta tasolta löytyvät yksittäiset kenttä- ja toimilaitteet ja niiden käyttämät tiedonsiirtotavat. Kenttätason kokoonpanot liitetään automaatiokeskuksiin, joissa suurin osa automaatio-ohjauksista suoritetaan. Keskuksilta on yhteys valvomotason laitteistoon. Tähän kuuluvat yleensä PC-pohjaiset käyttöliittymät. Toteutuskohteesta riippuen valvomotaso voidaan vielä yhdistää ylemmän tason keskusvalvomon alle. (Mäntylä 2011, 7.). Kulunvalvonta- ja turvallisuusohjaukset voidaan integroida suoraan vartiointiliikkeen valvomoon ja paloilmoitukset hätäkeskuksen järjestelmään. Tällöin hälytykseen vastaavat saavat reaaliaikaista tilannetietoa ja voivat näin ennakoita tilannetta aikaisempaa paremmin (Hakkarainen N.d. 8.). Kulunvalvonnalla voidaan myös ohjata tilakohtaisia turvallisuustekijöitä. Huoltohenkilöstön kirjautuminen tilaan voi kytkeä paikallisen kaasusammutusjärjestelmän pois päältä (Alanen 2014, 33.).



Kuvio 2. Rakennusautomaation hierarkia (Kiinteistöjen valvomojärjestelmät 2008, 22.)

Valvomotason hallintajärjestelmät toimivat rajapintana automaatioon päin. Niistä voidaan paikallisten tai keskitettyjen etävalvomoiden kautta hallita automaatioprosesseja. Kunnossapidon ja prosessin raportoinnin toiminnot suoritetaan yleensä tällä tasolla. Kiinteistöhuollolla voi olla koottu etähallinta useasta kiinteistöstä. Laaja teollisuusprosessi voi olla koottuna yhden valvomopisteen yhteyteen. (Rajala 2014, 7.)

Automaatiotason keskuksat suorittavat automaatioprosessit siihen liittyvien kenttälaiteiden antamien mittausten ja valvomotason asetuksien mukaan. Automaatiokeskusten välinen tiedonsiirto tapahtuu pääsääntöisesti LAN-verkossa TCP-IP-protokollalla. Jotkin yksittäiset mittaukset voidaan liittää langattomasti samaan verkkoon. Esimerkiksi lämmityksen säätöön yleensä tarvittava ulkolämpötila on hyvä tuoda järjestelmään langattomasti. (Rajala 2014, 7.)

Kenttätason laitteet tuovat automaation käyttöön reaaliaikaista mittaustietoa. Laitteet toimivat yleensä jonkin sarjaväylän kautta. Laitteet eivät yleensä sisällä omia prosessinohjauksia. Joissakin tilanteissa on kuitenkin tarpeellista varmistaa toiminta yksinkertaisella kenttätasonohjauksella. Kenttätason kommunikointi tapahtuu kenttäväylän välityksellä. Tunnetuimpia rakennusautomaation väyliä ovat Modbus, KNX, LON, EIB jne. Väylän valintaan vaikuttavat sovelluksen rakenne, asiakkaan vaatimukset ja automaatiotoimittajan tarjoamat vaihtoehdot. (Rajala 2014, 8.)

Rakennusautomaation alueelle ei ole muodostunut yksittäisiä, muita selkeästi suosittuja toteutustapoja. Laite- ja järjestelmätoimittajia on runsaasti, ja kaikkien väyläratkaisujen yhteen liittäminen muodostuu helposti ongelmaksi. Varsinkin vanhempien järjestelmien uudistaminen voi olla hankalaa, ja monesti koko järjestelmän uusimisen sijaan järjestelmät pyritään integroimaan toisiinsa. Laitteiden fyysiset yhteydet, tiedonsiirtotavat ja tiedonesittäminen valvomossa ovat keskeisimmät tekijät tässä prosessissa. (Alanen 2014, 33.)

4 Sarjaliikenneviestintä

Sarjaliikennekommunikointi erilaisten laitteiden välillä perustuu jaksotettuun keskusteluun yhteysväylässä, jossa jokainen bitti lähetetään vastaanottajalle hallitussa järjestyksessä. Vuosien varrella standardeja on muodostunut tarpeen ja käyttökohteen

mukaan, esimerkkinä mm. USB (Universal Serial Bus), Ethernet, RS-232 ja RS-485. (Serial Communication N.d.)

Toteutustavasta huolimatta kaikki sarjaliikennekeskustelu voidaan jakaa kahteen kastiin, joilla väyläkeskustelu saadaan pidettyä yhtenäisenä. Puhutaan synkronoidusta tai asynkronisesta kommunikoinnista. Synkronoitu viestintä tarvitsee aina ulkopuolisen ”porttikellon”, joka pitää huolen, että jokainen bitti lähtee ja saapuu ajallaan. Viestintä ei onnistu, jos lähtö- ja tulokello ovat eri ajassa. Asynkroninen viestintä korvaa ulkoisen kellon muodostamalla helposti hallittavia sanomia lähetettävästä tiedosta. Sanoman tarkka rakenne takaa, että lähettävä ja vastaanottava laite ymmärtävät toisiaan. (Module7: Serial Communication – Part 1 (Basics) 2016.)

Työn kannalta tärkeintä on hahmottaa asynkronisen viestinnän sanomarakenne.

4.1 Asynkroninen sarjaviestintä

Ilman ulkoista ohjauskelloa asynkroninen viestintä perustuu sanomanviestin tarkkaan määrittämiseen. Sanoma koostuu itse viestistä (Data bits), alku- ja lopputahdistusbiteistä (Synchronization bits) sekä tarkistusbiteistä (Parity bit). Sarjaväylän kannalta tärkeää on myös väylän Baud-nopeus (bits per second) eli se, kuinka nopeasti bitit liikkuvat väylässä. Kaikilla väylään liitetyillä laitteilla tämän nopeuden pitää olla sama, jotta viestit eivät vääristy matkalla. Tarkistusbitit jätetään nykyisin suurelta osin käyttämättä parantuneen fyysisen väylätekniikan (esim. kaapelointi) takia. Häiriölle herkässä ympäristössä, jossa linjassa on paljon melua, voidaan tarkistusbitillä parantaa huomattavasti viestintävarmuutta. (Module7: Serial Communication – Part 1 (Basics) 2016.)

Asynkronisen viestinnän edut selviävät parhaiten tutkimalla asiaa visuaalisesti. Kuviossa 1 näytetään, miten ASCII-merkistön avulla voidaan muodostaa kirjaimet O ja K sisältävä sanoma. Kirjaimet muutetaan ASCII-merkistön mukaiseen bittimuotoon, O = 79 = ”01001111” ja K = 75 = ”01001011”. Tämän välittämiseen tarvitaan kaksi sanomaa. Väylän viestintä tahdistuu automaattisesti laitteiden tunnistaessa aloitus- ja lopetusbitit. (Serial Communication N.d.)



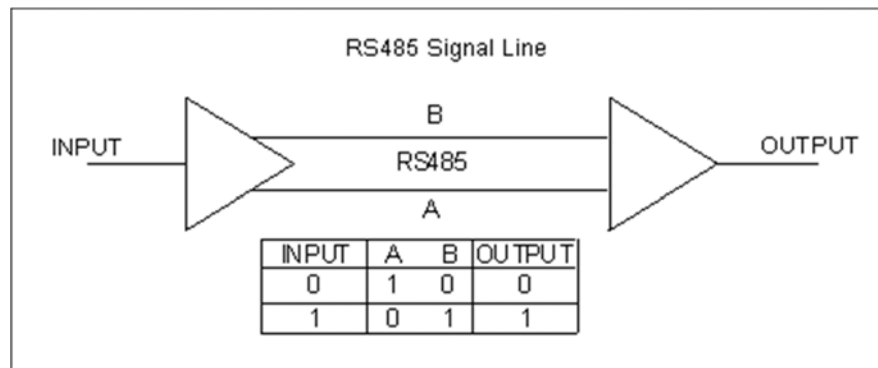
Kuvio 3. Esimerkki asynkronoidusta sarjaviestistä (Serial Communication N.d.)

4.2 RS-485-standardi

RS-485 on laajalti käytetty sarjaliikennestandardi, joka määrittelee tarkoin viestien ja kaapeloinnin fyysisen rakenteen. Fyysisenä määritelmänä RS-485 ei ota kantaa siihen, millä protokollalla sanoman sisältö on koodattu, onko esimerkiksi Modbus vai Profibus. (RS485 (2-wire, half-duplex, differential, multi-drop (32 nodes), communications standard for distances up to 4000ft.) N.d.)

RS-485-standardin mukaisella väylällä voidaan toteuttaa maksimissaan 32 laitteen kaksisuuntainen sarjaliikenne aina 1200 metrin kaapeloinnilla. Väylän signaalitaso on määritelty välille -7 V... +12 V ja signaalierojen on oltava vähintään 200 mV. Signaalinen differentiaalivertailu ja kierretty parikaapelointi sallivat varmemman viestinnän pitkillä ja häiriöherkillä alueilla. Väylässä ei verrata jännitetasoa suoraan maapotentiaaliin vaan viestintäkaapeleiden väliseen jännitevaihteluun. Koska häiriö muodostuu molempiin johtimiin ja viestien välittämiseen riittää johtimien välinen jännite-ero, sietää väylä huomattavia häiriötekijöitä, ennen kuin viestien siirtyminen vaarantuu. Tästä syystä standardi on suosittu erityisesti teollisuussovelluksissa. (RS-232, RS-422, RS-485 Serial Communication General Concepts N.d.)

RS-485-väylän kaapeloinnissa käytetään yleisesti merkintöjä A tai - ja B tai +. A-johtimeen kytketään negatiivinen jännite ja B-johtimeen positiivinen jännite. Oikein kytkettynä A-johtimen negatiivinen jännite ja B-johtimen positiivinen jännite vastaa arvoa 1. Kun signaalitasot johtimien välillä vaihdetaan, väylä on arvossa 0 (ks. kuvio 4). (Karttunen 2009, 13.)



Kuvio 4. Bittien siirtäminen RS-485-väylässä (Studholm, J N.d.)

RS-485-väylää voidaan laajentaa useiden kilometrien alueelle käyttämällä väylätoistimia, ja yhden väyläverkon 32 laitteen normaalia voidaan nostaa käyttämällä korkean impedanssin vastaanottovastuksella varustettuja laitteita, jolloin ylärajaksi saadaan 256 laitetta yhdessä verkossa. Linkittämällä toisiinsa tällaisia verkkoja voidaan toteuttaa tuhansien laitteiden laaja-alainen väyläratkaisu. (Karttunen 2009, 13.)

5 Modbus-sarjaliikenneprotokola

Modbus-väyläprotokolla on laajasti käytetty, vuonna 1979 kehitetty viestintärakennemuoto. Modbus-protokollan avoin koodi on tehnyt siitä laitevalmistajille suotuisan vaihtoehdon toteuttaa isäntä-orja (master-slave) -keskusteluja älykkäiden laitteiden välillä. Eri laitevalmistajien komponentit voidaan liittää suoraan väylään ja suorittaa väylänsäätö laitteen osalta ilman erillisiä laitevalmistajakohtaisia lisälaitteita. (Modbus FAQ: About the Protocol 2017.)

Isäntä-orja-keskustelussa isäntäohjain hallitsee väyläviestintää lähettämällä kyselyitä ja vastaanottamalla viestejä orjalaitteilta. Kaikille laitteille tarkoitettuun kyselyyn ei odoteta vastausta. Laitekohtaiseen kyselyyn odotetaan protokollan mukainen vastaus. Vastaus sisältää aina kenttälaitteen osoitteen, jotta viestien yksilöiminen olisi mahdollista. Jos viestin välityksessä tai itse viestissä havaitaan ongelma, vastauksena lähetetään virheilmoitus. (Karttunen 2009, 14.)

5.1 Modbus RTU -väylän viestirakenne

Modbus RTU -väylässä (Remote Terminal Unit) sanomat muodostetaan kahdeksan bitin tavuista. Tavut on jaettu kahteen neljän bitin heksadesimaalimerkkiin. Tavu alkaa aina aloitusbitillä. Tätä seuraa kahdeksan tietobittiä, minkä jälkeen tulee varmenusbitti valinnaisena ja lopetusbitti. Tavu päättyy kahteen lopetusbittiin, jos tarkistusbittiä ei käytetä. Tavuista muodostetaan RTU-viesti kuvion 5 mukaisesti. Viestin perään lisätään sisällöstä laskettu 16-bittinen CRC-tarkistussumma (Cyclical Redundancy Check), jolla varmistetaan viestin yhtenäinen siirtyminen väylässä. (Modicon: Modbus Protocol Reference Guide 1996, 16.)

| START | ADDRESS | FUNCTION | DATA | CRC CHECK | END |
|-------------|---------|----------|-------------------|-----------|-------------|
| T1-T2-T3-T4 | 8 BITS | 8 BITS | $n \times 8$ BITS | 16 BITS | T1-T2-T3-T4 |

Kuvio 5. Modbus RTU -viestikehys (Modicon: Modbus Protocol Reference Guide. 1996, 18.)

Viestien välittämisen kannalta on tärkeää, että laitteet tunnistavat, milloin viesti loppuu ja uusi viesti voidaan aloittaa. Tätä varten protokolla määrittelee erottavaksi tekijäksi aikajakson (T1-T2-T3-T4) (ks. kuvio 3), jolloin laite ei lähetä väylään informaatiota. Tämä hiljainen aika on pituudeltaan vähintään "3.5 x merkkiaika". 9600 baudin väylässä yhden heksadesimaalimerkin lähettäminen kestää noin 1 ms. Tällöin pienin hiljaisuusaika on 3.5 ms. Samalla tavalla varmistetaan viestien jatkuvuus. Käytännössä hiljainen aika on kymmenien ja satojen millisekuntien luokkaa viestintävarmuuden takaamiseksi. Jos taas viestin keskellä on pidempi ajanjakso kuin "1.5 x merkkiaika", viesti katsotaan vialliseksi ja se unohdetaan. Tämän jälkeen laite jää odottamaan uutta osoitekentällä alkavaa viestiä. (Modbus Messaging Time 2017.)

RTU-viestin suurin etu on kompakti viestirakenne. Viestin lyhempi muoto lisää väylän tiedonsiirtokapasiteettia. Esimerkiksi otettaessa yhteyttä laitteeseen "06" (Hex), ASCII-muotoisena osoitteen ilmoittamiseen tarvitaan kaksi merkkiä "0" ja "6". Nämä muutetaan väylään binäärimuotoon "00110000 0001010" ja " 00110110 0001010". Molemmat merkit ilmaistaan standardin mukaisesti kahdella kahdeksan bitin tavulla.

Vastaavasti RTU-viestinä heksadesimaaliluvut voidaan kääntää suoraan binäärimuotoon "0000 0110" (ks. kuvio 6). (Modicon: Modbus Protocol Reference Guide 1996, 15.)

| QUERY | | | |
|---------------------|---------------|------------------|-----------------|
| Field Name | Example (Hex) | ASCII Characters | RTU 8-Bit Field |
| Header | | : (colon) | None |
| Slave Address | 06 | 0 6 | 0000 0110 |
| Function | 03 | 0 3 | 0000 0011 |
| Starting Address Hi | 00 | 0 0 | 0000 0000 |
| Starting Address Lo | 6B | 6 B | 0110 1011 |
| No. of Registers Hi | 00 | 0 0 | 0000 0000 |
| No. of Registers Lo | 03 | 0 3 | 0000 0011 |
| Error Check | | LRC (2 chars.) | CRC (16 bits) |
| Trailer | | CR LF | None |
| Total Bytes: | | 17 | 8 |

Figure 7 Master Query with ASCII/RTU Framing

| RESPONSE | | | |
|---------------|---------------|------------------|-----------------|
| Field Name | Example (Hex) | ASCII Characters | RTU 8-Bit Field |
| Header | | : (colon) | None |
| Slave Address | 06 | 0 6 | 0000 0110 |
| Function | 03 | 0 3 | 0000 0011 |
| Byte Count | 06 | 0 6 | 0000 0110 |
| Data Hi | 02 | 0 2 | 0000 0010 |
| Data Lo | 2B | 2 B | 0010 1011 |
| Data Hi | 00 | 0 0 | 0000 0000 |
| Data Lo | 00 | 0 0 | 0000 0000 |
| Data Hi | 00 | 0 0 | 0000 0000 |
| Data Lo | 63 | 6 3 | 0110 0011 |
| Error Check | | LRC (2 chars.) | CRC (16 bits) |
| Trailer | | CR LF | None |
| Total Bytes: | | 23 | 11 |

Kuvio 6. Modbus ASCII- ja RTU viestien vertailu (Modicon: Modbus Protocol Reference Guide 1996, 19.)

5.2 Modbus-funktiot

Funktio on standardissa määritelty komento, jolla isäntälaitte kertoo orjalaitteelle, mitä sen pitää tehdä. Funktio voi olla kokonaisluku väliltä 1–255. Komentoalue sisältää ennalta määrättyjä komentoja, jotka ovat yhteisiä kaikille laitteille, laitekohtaisia erikoiskomentoja, joilla voidaan suorittaa laitteelle ominaisia tehtäviä, sekä komento-osio, joka on varattu mahdollisille laajennuksille itse standardiin. (Modicon: Modbus Protocol Reference Guide 1996, 19.)

Kaikille laitteille yhteiset yleiskomennot tiedon lukemiseen ja kirjoittamiseen on määritelty taulukossa 1.

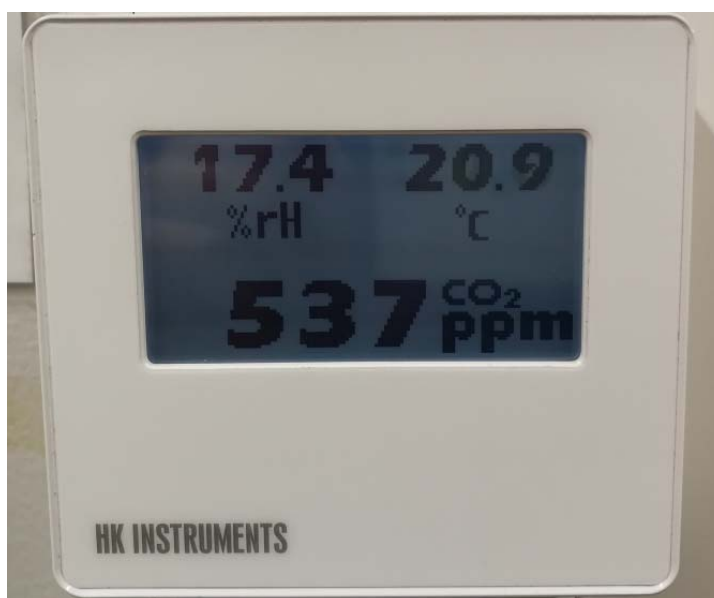
Taulukko 1. Modbus-yleisfunktioitaulukko (What is a function code 2017.)

| Function Code | Action | Table Name |
|---------------|----------------|---------------------------------|
| 01 (01 hex) | Read | Discrete Output Coils |
| 05 (05 hex) | Write single | Discrete Output Coil |
| 15 (0F hex) | Write multiple | Discrete Output Coils |
| 02 (02 hex) | Read | Discrete Input Contacts |
| 04 (04 hex) | Read | Analog Input Registers |
| 03 (03 hex) | Read | Analog Output Holding Registers |
| 06 (06 hex) | Write single | Analog Output Holding Register |
| 16 (10 hex) | Write multiple | Analog Output Holding Registers |

5.3 Käytettävät Modbus RTU-väylälaitteet

5.3.1 HK Instruments: CDT-MOD-2000-hiilidioksidilähetin

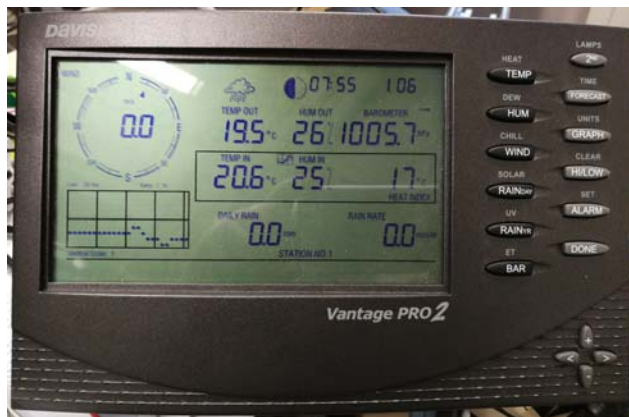
HK Instrumentsin valmistamalla CDT-MOD-2000-sarjan (ks. kuvio 7) mittapaneelilla voidaan lähettää Modbus RTU -väylään mittatiedot asennuspisteen lämpötilasta, suhteellisesta kosteudesta ja ilman hiilidioksidipitoisuudesta. Paneelin ensiohjelointi suoritetaan paneelin kosketusnäyttövalikon kautta. Ensiasennuksessa tärkeintä on valita Modbus RTU -väylässä käytettävä väyläosoite ja -nopeus, minkä jälkeen laitetta voidaan säätää väylän kautta. (Asennusohje CDT-MOD-2000 2016)



Kuvio 7. HK Instruments -hiilidioksidilähetin: CDT-MOD-2000

5.3.2 Davis Vantage Pro 2 -sääasema

Davisin valmistamaan Vantage Pro 2 -sarjan sääasemaan kuuluu kuvion 8 mukainen konsolipaneeli, sekä kuvion 9 mukainen ulkoilman mittalaittekokonaisuus. Ulkotilaan asennettava mitta-asema lähettää langattomana säätietoja lämpötilasta, ilmankosteudesta, ilmanpaineesta, sademäärästä, tuulensuunnasta ja nopeudesta. Sisätiloihin asennettava paneeli mittaa asennuspisteessä lämpötilan ja ilmankosteuden. Paneeli toimii myös sääaseman isäntänä ja huolehtii muun muassa kellonajoista, mittausten laatumuunnoksista ja viestinnästä ulkoiseen käyttöliittymään. (User manual 2015.)

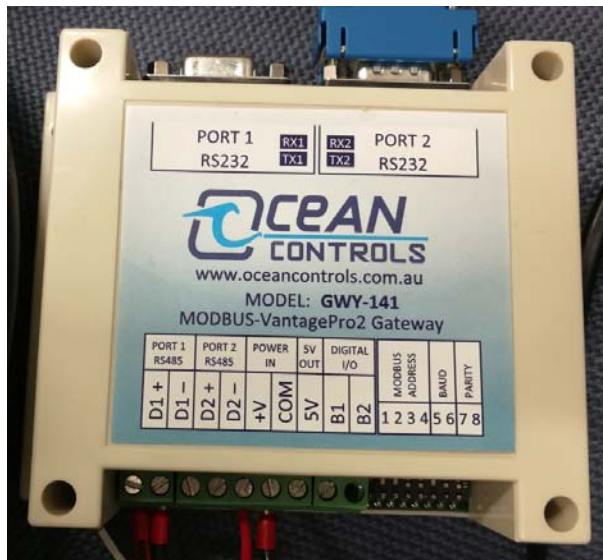


Kuvio 8. Davis Vantage Pro 2 -sääaseman ohjainpaneeli



Kuvio 9. Davis Vantage Pro 2 -sääaseman ulkoilman mittauslaitteisto

Sääseman ohjainpaneeli on oletusarvoisesti tarkoitettu liitettäväksi suoraan yksittäiseen tietokoneeseen, minkä vuoksi paneelin liittämiseksi Modbus RTU -väylään tarvitaan kuvion 10 mukainen signaaliadapteri. Adapterista säädetään Modbus RTU -väylän mukaiset asetukset asetuskymillä (Instruction manual 2011).



Kuvio 10. Davis Vantage Pro 2 -sääseman Modbus RTU -väyläadapteri

6 Z-Wave-langatonviestintäprotokolla

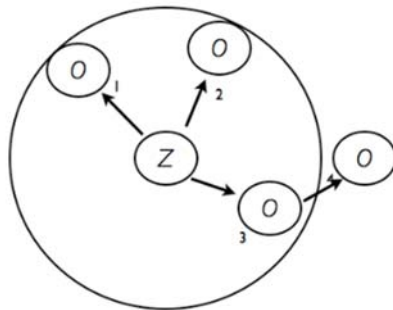
Tanskalainen Zensys toi kotiautomaatiomarkkinoille vuonna 2001 langattoman Z-Wave-järjestelmän ensimmäisen version. Useiden uudistusten ja monien yhteistyökumppaneiden jälkeen Z-Wave Alliance on hyväksynyt 1700 laitetta 350 eri valmistajalta Z-Wave-sertifikaatilla. Sertifikaatti takaa kuluttaja-asiakkaille järjestelmän laadun. (Z-Wave 2017.)

6.1 Toimintaperiaate

Z-Wave-järjestelmä käyttää hyväkseen vapaan käytön radiokaistaa, joka mantereesta riippuen toimii 866–921 MHz:n alueella. Euroopassa pääsääntöisesti käytössä on taajuuskaista 868.42 MHz. Valitsemalla järjestelmän viestintäväyläksi matalat radiotaajuudet Z-Wave-kommunikointi pystyy välittämään viestit laitteiden välillä pienellä viiveellä luotettavasti. (Z-Wave 2017.)

6.2 Topologia ja viestintä

Yksi suurimmista eduista Z-Wave-järjestelmässä on sisäänrakennettu laitteiden risiinkeskusteluominaisuus (Meshed Network Technology). Järjestelmässä olevat laitteet muodostavat automaattisesti itsenäisen verkon, jonka avulla kaikki laitteet saavuttavat isäntäohjaimen joko suoraan tai välillisesti toisen tai useamman mittalaitteen toimiessa välittäjänä. Tällöin järjestelmään ei tarvitse asentaa erillisiä reitittämiä tai muita lisälaitteita. Esimerkiksi isäntälaitte voi välittää viestin laitteelle 4 käyttäen välittäjänä laitetta 3, vaikka laite 4 on isäntäohjaimen oman kantaman ulkopuolella (ks. kuvio 11). Perinteisessä langattomassa toteutuksessa tähän väliin tarvittaisiin reititin. (Z-Wave 2017.)



Kuvio 11. Yksinkertaistettu malli Z-Wave-topologiasta. (Z-Wave Explained 2017.)

Yhden isännän verkko tukee enimmillään 232 laitteen kokoonpanoa ja tätä voidaan laajentaa siltaamalla kaksi verkkoa toisiinsa. Laitteiden lisääminen verkkoon tapahtuu laittamalla isäntäohjain ja lisättävä laite paritustilaan, jolloin isäntä antaa yhdistettävälle laitteelle verkotunnukset. Isäntälaitte arvioi samalla laitteiden välisen signaalitason, joten toiminnan kannalta järjestelmä olettaa laitteiden olevan niille tarkoitettuilla pysyvillä paikoilla. Parittaminen suoritetaan vain kerran, minkä jälkeen isäntä tunnistaa laitteen, vaikka sen paikkaa vaihdettaisiin. Joissain isäntälaitteissa on akulla varmennettu virran saanti, jolloin se voidaan viedä paritettavan laitteen luo ja palauttaa sen jälkeen omalle paikalleen. Paritus voidaan purkaa samanlaisella prosessilla, jolloin laitteen verkkotunnukset palautetaan ja se voidaan liittää toiseen verkkoon. (Z-Wave 2017.)

Isäntälaitteella ylläpidetään koko verkon toimintaa. Verkolle luodaan oma ID-tunnus ja yksittäisille laitteille liityntä-ID. Vain laitteet, joilla on parittamisen yhteydessä annettu verkkotunnus, voivat keskustella kyseisen verkon kanssa. Kahta samanlaista laitetunnusta ei hyväksytä samassa verkossa. (Z-Wave 2017.)

6.3 Z-Way-käyttöohjelmisto

Vuonna 2016 Z-Wave-lähdekoodi vapautettiin suurelta osin avoimeen käyttöön, ja vuonna 2008 aloitettu harrasteprojekti Z-Way sai vuonna 2017 Z-Wave Alliancen virallisen sertifiikaatin. Kuitenkin jo vuonna 2014 Z-Wayn silloinen päivitysversio sisälsi uusimman Z-Wave Plus -standardin mukaisen toteutuksen, jota kuluttaja-asiakkaille tuotteita toimittavat isot tietotekniikkayhtiöt eivät pystyneet vielä tuolloin tekemään. Z-Wave Plus paransi tiedonsiirtotehokkuutta sekä päivitti merkittävästi järjestelmän tietosuojaa määrittelemällä uudet tietoturva vaatimukset (Security 2, S2) sertifiikaatin saamiseksi. Vuonna 2017 Z-Way oli jälleen ensimmäinen Z-Wave-toteusmalli, jolla pystyttiin tarjoamaan kokonainen "Security 2"-tietoturvaso. (Paetz, C 2017.)

Z-Way on täydellinen ja ensimmäinen sertifioitu ohjelmistopohjainen toteutustapa Z-Wave-laiteverkolle (Z-Way the Gold-Standard Z-Wave Controller 2017.). Z-Way tarvitsee kuitenkin toimiakseen Z-Way-lisensoidun Z-Wave-kommunikointilaitteiston. Tiettyissä tilanteissa Z-Way toimii kuluttaja-asiakaslaitteistolla, mutta vaatii usein laitevalmistajan vapauttamaan laitteiston lukituksen. Tarkoituksenmukaisessa kehitysympäristössä Z-Wayn voi muokata toimimaan ilman lisenssejä, jolloin ohjelmiston ja laitteiston tuotekehitystyö onnistuu vapaasti. Z-Way-lisenssin voi hakea valmiille toteutukselle erikseen. (Paetz, C 2017.)

6.4 Z-Way: RaZberry2-isäntäohjain

RaZberry2 (ks. kuvio 12) on uusi iteraatio Raspberry Pi -mikrotietokoneelle tarkoitusta Z-Wave-ohjaimesta. RaZberry2-ohjain sisältää kaiken tarvittavan, jolla Raspberystä saadaan muodostettua Z-Wave-isäntä. Laitteen kantama on noin 200 metriä. Laitteen asennuksen mukana tulee käyttövalmis verkkoselainpohjainen käyttöliit-

tymä, jolla pääsee hallitsemaan laiteverkkoa. Käyttöliittymää voi tarvittaessa muokata omiin tarkoituksiin sopivaksi avoimen lähdekoodin ansiosta. (Z-Wave.Me RaZberry2 EU 2017.)



Kuvio 12. RaZberry2-lisäkortti Raspberry Pi -mikrotietokoneelle (Z-Wave.Me RaZberry2 EU 2017.)

6.5 Käytettävät Z-Wave-laitteet

6.5.1 AEOTEC: Multisensor 6

Aeotec Multisensor 6 (ks. kuvio 13) on nimensä mukaisesti usean mittauksen yksittäinen laite. Laitetta voidaan käyttää paristoilla tai USB-muuntajalla. Laitteessa on edistynyt omadiagnostiikka, joka valvoo esimerkiksi laitteen värähtelyanturilla, onko laitetta käyty käsittelemässä ilman lupaa. Tätä tietoa voidaan käyttää käyttöliittymän kautta laukaisemaan etähälytyksen lähettäminen omistajan haluamaan viestivälineeseen (puhelin, sähköposti). Myös muista mittauksista saadaan reaaliaikainen tieto käyttöliittymään. Laitteella suoritettavat mittaukset:

- Liike
- Lämpötila
- Valoisuus (LUX)
- Ilmankosteus
- Tärinä
- UV, ultravioletti säteily

(Multisensor 6 2017.)



Kuvio 13. Aeotec Multisensor 6 -mittalaite (Multisensor 6 2017.)

6.5.2 Philio-PAN11-1B: Älykäs energiamittari ja kytkin

PAN11-1B on älykäs pistorasiaan kytkettävä energiamittari etäohjattavalla kytkimellä. Laitteen kautta voidaan kytkeä 3000 W kuorma, ja laitteeseen voidaan ohjelmoida etähälytyksiä esimerkiksi energiakulutuksen mukaan (ks. kuvio 14). (Philio PAN11-1B manual 2017.)



Kuvio 14. Philio PAN11-1B -energiamittari ja kytkin (Philio PAN11-1B manual 2017.)

7 Domoticz-kotiautomaatiokäyttöliittymä

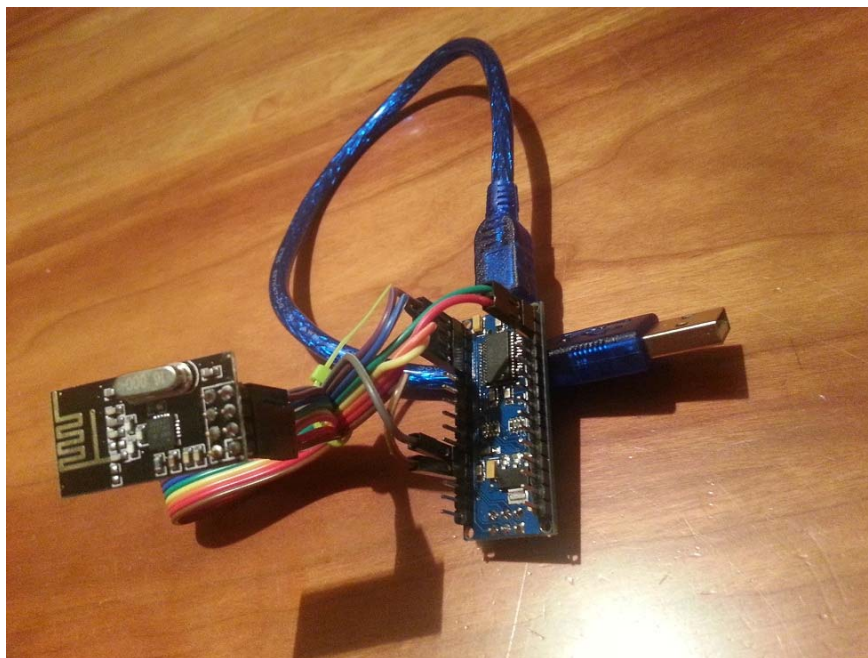
Domoticz on pienen kehitystiimin vuonna 2012 julkaisema kokonaisvaltainen avoimen lähdekoodin kotiautomaatiokäyttöliittymä, joka tukee useita käyttöjärjestelmiä

(Windows, Linux, Mac OS, sulautetut). Käyttöliittymä on HTML 5 -pohjainen, automaattisesti skaalautuva selainympäristö. Ohjelmiston taustalla pyörii C++ -toteutus yhteensopivuus- ja lisäosaongelmien välttämiseksi. (About Domoticz 2017.)

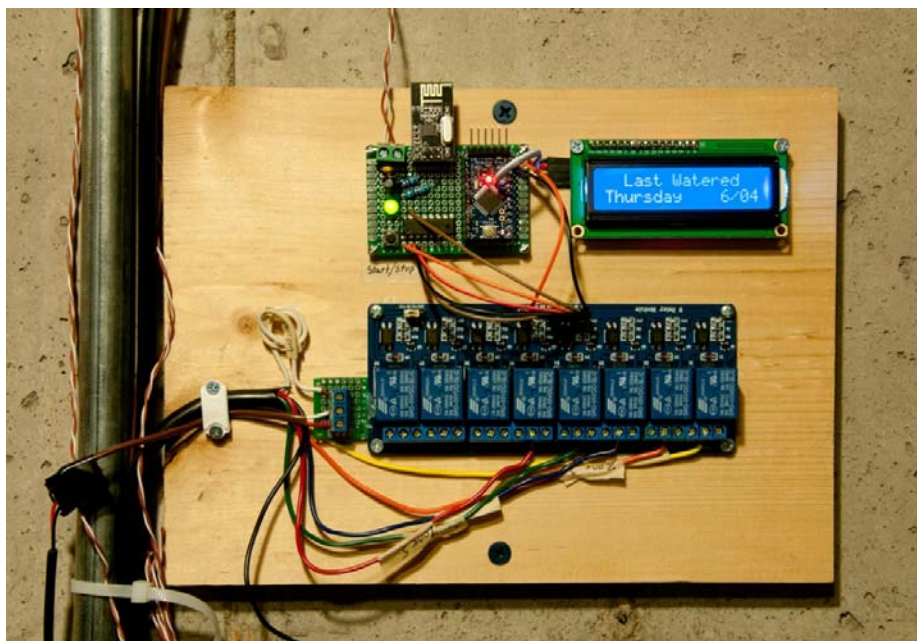
Domoticz on pyritty toteuttamaan mahdollisimman käyttäjäystävällisellä tavalla, mutta samalla se kuitenkin kannustaa kotiautomaation syvempää oppimista. Pelkän mittaustiedon näyttämisen ohella käyttöliittymän kautta voidaan toteuttaa esimerkiksi yksinkertainen kellonajan tai muun tapahtuman mukaan säätyvä valaistuksen ohjaus. Myös monimutkaisempia energiankäyttöön liittyviä säätöjä voidaan ohjelmoida: esimerkiksi milloin aurinkoenergiaa kannattaa varastoida ja milloin käyttää kodin tarpeisiin. Domoticz tukee useimpia ohjelmointikieliä (Python, Lua, Blockly, PHP, Perl), joten osaamisen tai tarpeen mukaan voi valita sopivimman. Avoimen lähdekoodin toteutuksena Domoticzin ohjesivuilla on valmiita malleja, joista voi muokata omaan tarkoitukseen sopivia. (Domoticz and Scripting 2017.)

8 MySensors-laiteverkko

MySensors on avoimen lähdekoodin ohjelmistokirjasto, jonka avulla voidaan luoda erilaisia laiteverkkoja. MySensors toteuttaa viestinnän samalla periaatteella Z-Wave-järjestelmän kanssa. Laitepisteet (Node) voivat toimia verkossa myös viestin välittäjänä, jolloin suoraa kommunikointiyhteyttä isännän ja kohdelaitteen välillä ei tarvita. Kirjasto sisältää tarvittavat ohjelmakoodit isäntäohjaimen ja erillisten laitepisteiden ohjelmoimiseen. Viestintä voi tapahtua langattomasti tai langallisesti asennuskohdeesta ja tarpeesta riippuen. Kuviossa 15 on esimerkkinä MySensors-isäntä käyttäen Arduino Nanoa, jolla laitepisteiden kanssa viestitään langattomasti. Käyttöliittymään päin viestintä tapahtuu USB-portin kautta. Samaa kokoonpanoa voidaan käyttää vaihtamalla ohjelmakoodisto laitepisteeksi ja liittämällä Nanoon mittalaitteita (ks. kuvio 16). (MySensors 2017.)



Kuvio 15. MySensors-isäntäohjain (MySensors 2017)

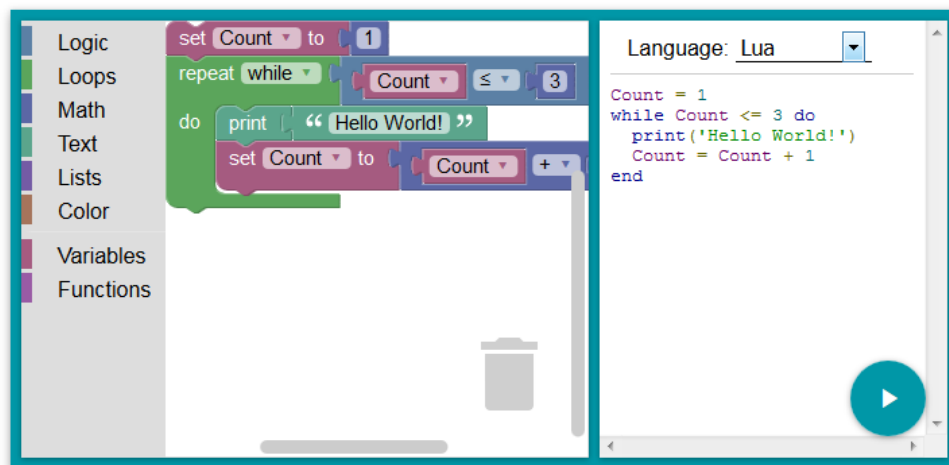


Kuvio 16. MySensors-laitepiste-esimerkki (MySensors 2017)

9 Graafinen ohjelmointikieli Blockly

Blockly on Googlen ylläpitämä avoimen lähdekoodin graafinen ohjelmointikäyttöliittymä ohjelmointikielille (ks. kuvio 17). Perinteiset ohjelmakoodin komennot on visualisoitu helposti käsiteltäviksi palikoiksi, joista voi yksinkertaisesti rakentaa ohjelmakokonaisuuksia. Esimerkiksi MIT (Massachusetts Institute of Technology) käyttää Blocklya App Inventor -ohjelmistossaan, jolla luodaan Android-pohjaisia sovelluksia. (Blockly 2017.)

Try Blockly



Kuvio 17. Blockly-käyttöliittymä (Google for Education – Blockly 2017.)

10 Raspberry Pi -mikrotietokone

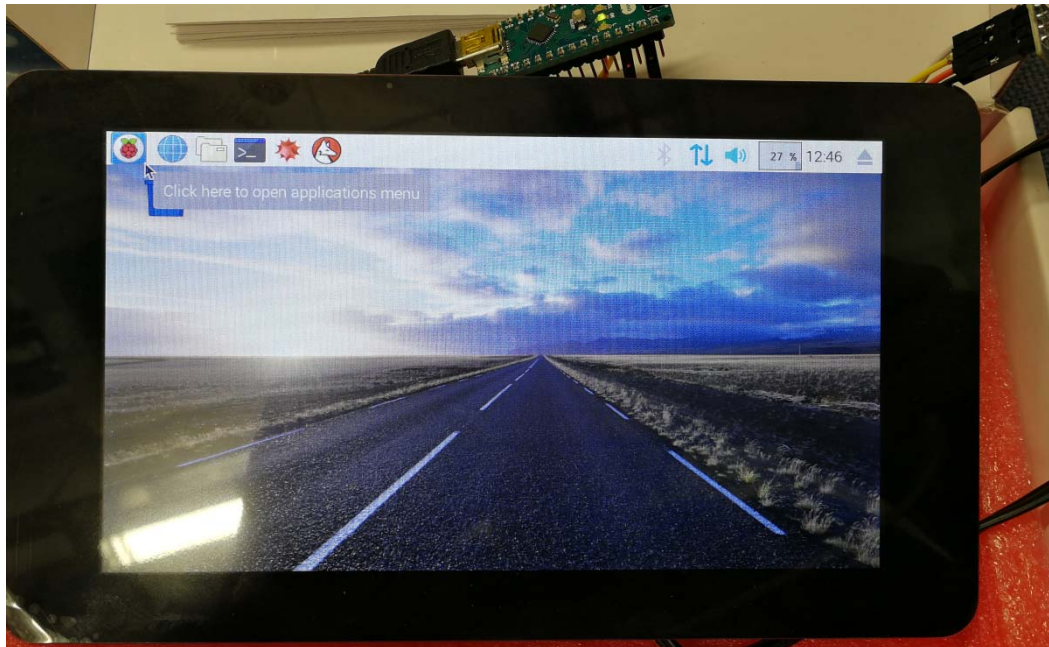
Raspberry Pi (ks. kuvio 18) on Raspberry Foundation ylläpitämä ja luoma kehitysalusta, jolla haluttiin tuoda esiin uusi malli opettaa ja oppia tietotekniikkaa niin kotiin kuin kouluympäristössä. Raspberry Foundation ylläpitää myös Rasbian-käyttöjärjestelmäjulkaisua, joka on suositus käyttöjärjestelmä Raspberry Pi -alustalle. Käyttöjärjestelmä sisältää kaiken tarvittavan, jolla tietokone toimii sellaisenaan (verkkoseläimen, tekstin- ja kuvakäsittelyohjelmat, ohjelmointiohjelmiston jne.). (Raspberry Pi 2017.).



Kuvio 18. Raspberry Pi 3 Model B -kehitysalusta (Raspberry Pi 2017.)

10.1 Raspbian-käyttöjärjestelmä

Raspbian on Raspberry Pi'ille optimoitu Debian (Linux) -pohjainen käyttöjärjestelmä. Käyttöjärjestelmä sisältää kaiken tarvittavan Raspberryn käyttämiseen. Käyttöjärjestelmää ylläpitävä Raspberry Pi Foundation on sisällyttänyt käyttöjärjestelmään myös oppimistarkoitukseen paljon itsenäistä työskentelyä ja oppimista tukevaa ohjelmistoa ja esimerkkejä. Kuviossa 19 näkyy Raspbian Jessie -käyttöjärjestelmä 7" kosketusnäytöllä varustetussa Raspberry Pi'ssa. (Raspbian FAQ 2017.)



Kuvio 19. Raspbian-käyttöjärjestelmän Jessie-jakeluversion työpöytä

11 Arduino-kehitysalusta

Arduino on avoimen lähdekoodin ohjelmointi- ja kehitysalustakokoonpano. Alkujaan aloittelijoille tarkoitettu prototyypisovellus levisi nopeasti maailmanlaajuisesti tee-se-itse-valinnaksi helposti lähestyttävän ja moniulotteisen ohjelmointikielen ansiosta. Arduinon käyttäjäyhteisö ylläpitää avoimen lähdekoodin kirjastoa, josta voi vapaasti käyttää projektiinsa sopivia elementtejä ja muokata niitä omiin tarpeisiin. Arduino taipuu yksinkertaisesta ”paina nappia ja valo syttyy”-komentoista aina monimutkaisiin ohjauksiin, kuten kauko-ohjattava quad-lennokki. Alustat skaalautuvat toimintojen kasvaessa lähes postimerkin kokoisesta (Mini) noin A5-arkin kokoiseksi. Yleisimmät mallit ovat Uno, Nano/Mini ja Mega. (Introduction 2017.)

11.1 Arduino Nano -kehitysalusta

Arduino Nano (kuvio 20) on tuoteperheen pienimpiä alustoja. Se sisältää kuitenkin samat ominaisuudet kuin isompi Uno-versio (ks. taulukko 2). Nanoa käytetään yleensä valmiissa kokoonpanossa sen jälkeen, kun projekti on testattu toimivaksi kokonsa puolesta helpommin käsiteltävällä Unolla. (Introduction 2017.)



Kuvio 20. Arduino Nano.

Taulukko 2. Arduino Unon ja Nanon vertailu (Compare board specs 2017.)

| Nimi | Proessori | Käyt- tö- jän- nite | CPU MHz | A:I/O | D:I/O PWM | EEPROM kB | SRAM kB | FLASH USB | USB |
|---------------------|--------------------------------------|------------------------------|------------|-------|--------------|----------------|------------|--------------|------|
| UNO 68x57 mm | AT- mega328P | 5 V / 7-12 V | 16 | 6/8 | 14/6 | 1 | 2 | 32 | Reg. |
| NANO 18x45 mm | AT- mega168 AT mega328P | 5 V / 7-9 V | 16 | 8/0 | 14/6 | 0,512 1 | 1 2 | 16 32 | Min. |

11.2 Arduino IDE -ohjelmistopaketti

Arduino IDE -ohjelmistopaketti sisältää tarvittavat sovellukset ja ajurit Arduino-kehitysalustojen ohjelmointiin. Ohjelmistosta on olemassa käyttöjärjestelmään asennettava tai verkkoselaimessa toimiva versio. Käyttäjäläheisenä toteutuksena ohjelmisto sisältää valmiiksi suuren määrän kirjastoja projektien aloittamiseen. Erikoisempia kirjastoja on mahdollista asentaa lisää projektikohtaisesti. Kirjasto on kokoelma valmiita ohjelmointiesimerkkejä, joita voi vapaasti soveltaa omiin tarkoituksiin. Esimerkiksi LiquidCrystal-kirjasto sisältää tarvittavat koodiosiot LCD-näytön ohjaamiseen. (Introduction 2017.)(Installing Additional Arduino Libraries 2017.).

11.2.1 ModbusMaster-kirjasto

ModbusMaster-kirjasto sisältää kattavan ohjelmakokonaisuuden Modbus-väyläkeskusteluun orjalaitteiden kanssa. Ohjelmakoodi on hyvin kommentoitu ja sisältää yleisesimerkit, joiden avulla pääsee helposti alkuun (ks. kuvio 21). (ModbusMaster Library 2017.)

```

#include <ModbusMaster.h>

// instantiate ModbusMaster object
ModbusMaster node;

void setup()
{
  // use Serial (port 0); initialize Modbus communication baud rate
  Serial.begin(19200);

  // communicate with Modbus slave ID 2 over Serial (port 0)
  node.begin(2, Serial);
}

void loop()
{
  static uint32_t i;
  uint8_t j, result;
  uint16_t data[6];

  i++;

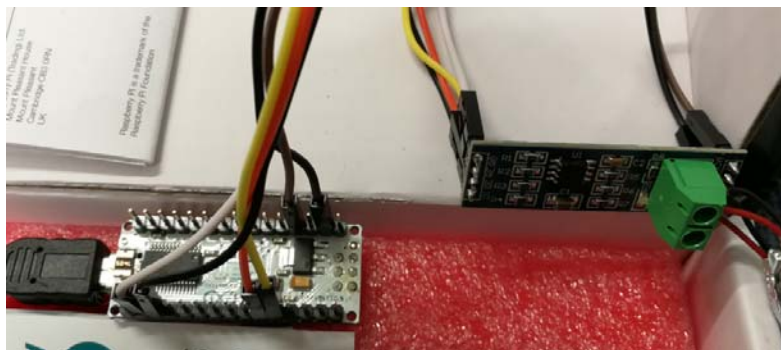
  // set word 0 of TX buffer to least-significant word of counter (bits 15..0)
  node.setTransmitBuffer(0, lowWord(i));
}

```

Kuvio 21. ModbusMaster-ohjelmakoodin esimerkki

11.3 RS-485 (MAX485) Arduino -ohjainkortti.

RS-485 Arduino -kortti, kuviossa 22 oikean puoleinen laite, käyttää Max485-mikro-ohjainta muuntamaan Arduinon TTL-muotoisen sarjaliikenteen RS-485-yhteensopivaksi. Laite on täysin läpinäkyvä järjestelmän kannalta ja sisältää tarvittavat päätevastukset. Laite ottaa käyttövirran Arduinolta. (RS-485 module for Arduino (MAX485) 2017.)



Kuvio 22. RS-485-väyläadapteri.

12 Työn toteus

Opinnäytetyö lähti liikkeelle selvittämällä, minkä tasoinen käyttöliittymä Domoticz on ja mitä sillä voidaan toteuttaa. Tässä vaiheessa käyttöliittymä oli kaikille osapuolille tuntematon. Samalla käytiin läpi rakennuslaboratoriosta löytyviä laitteita ja mahdollisuutta käyttää näitä laitteita tulevassa kokoonpanossa. Myös käytettävissä oleva aika rajasi toteutettavan laitteiston ominaisuuksia.

Koska laitteisto tulisi rakennusautomaatio-osaston laboratoriokäyttöön, sovittiin, että laitteistolle kirjoitettaisiin myös laboratorioharjoitus. Tällöin laitteisto olisi helppo sisällyttää aiheeseen liittyville kursseille harjoituksena tai oppilaiden omatoimisen opiskelun tueksi. Harjoitukselle ei määritelty tilaajan toimesta tiettyä toteutustapaa. Harjoitukset kuitenkin testattiin ja hyväksyttiin henkilökunnan toimesta.

Taustatutkimuksen pohjalta esiteltiin tilaajalle laitteistovaihtoehtoja, jotka mahtuivat budjettiin ja toteuttaisivat annettuja toimintoja. Ehdotelman kautta päädyttiin raportin alussa esitettyyn kokoonpanoon, joka palvelee rakennusautomaatiolaboratorion oppimisympäristöä. Tässä vaiheessa myös sovittiin, että laboratorio-ohjeistus keskittyisi Domoticz-käyttöliittymän ohjelmallisiin ominaisuuksiin.

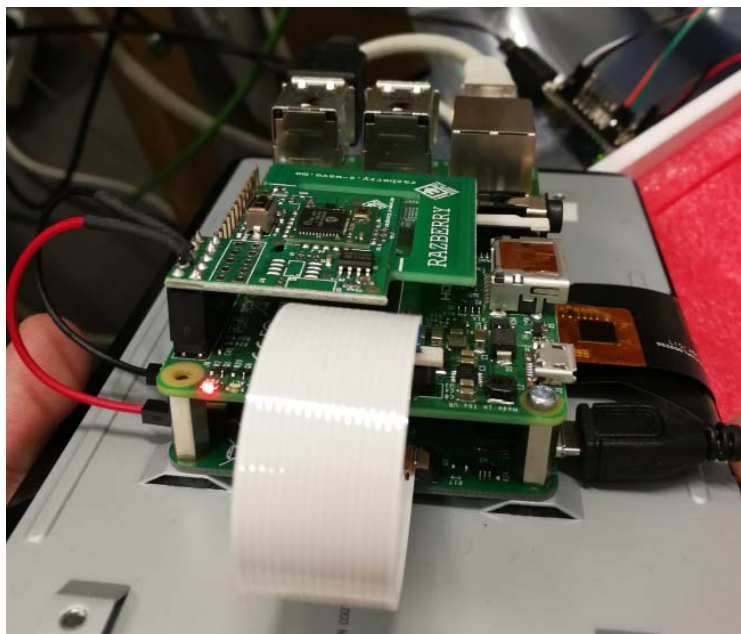
Työn edistyessä ja laitteiston mahdollisuuksien selkiytyessä työlle asetettiin lisätavoitteita, joita laitteistolla pitäisi pystyä tekemään. Laitteistolla tulisi pystyä kirjoittamaan Modbus-väylään, jotta sillä voitaisiin mallintaa oikeaa käytännön toteutusta. Kävi myös ilmi, että rakennuslaboratoriosta löytyvät Davisin sääasemat voitaisiin liittää erillisellä adapterilla Modbus-väylään. Toinen asemista varattiin laitteistoon liitettäväksi.

Opinnäytetyön laajin tulos on opiskelijoille tarkoitettu laboratorioharjoituskokonaisuus, jossa eritasoisten tehtävien kautta pyritään näyttämään harjoituksen tekijälle laitteiston mahdollisuudet. Laboratorio-ohjeistukseen on sisällytetty vaihtoehtoisia toteutuksia käytettävissä olevan ajan sekä halutun vaikeus- tai osaamistason mukaan. Harjoitukset pohjautuvat työn aika tehtyihin havaintoihin. Harjoitukset on jaoteltu liitteinä osa-alueittain ohjeistuksiin, joista voi koota haluamansa kokonaisuuden.

Asennukset suoritetaan Windows-ympäristössä, joten käyttäjältä edellytetään Windowsin perusosaamista.

12.1 Laboratorioharjoituksen laitekytkennät

Kuviosta 23 selviävät Raspberry Pi -laitteiden kytkennät. Raspberry Pi -mikrotietokoneen lisälaitteet on suunniteltu toimimaan keskenään, jolloin yksittäiset osat eivät tarvitse omaa käyttöjännitettään. Kosketusnäytön ohjainkortti toimii käyttöjännitteen jakelijana. Ohjaimen mikro-USB-porttiin tuodaan virta, joka jaetaan kuviossa näkyvillä GND (mus.) ja +5 V (pun.) johtimilla Z-Wave-ohjaimen vastaaviin nastoihin. Nämä ovat suoraan yhteydessä Raspberry Pi'n GPIO:n pinneihin 4 (+5 V) ja 6 (GND). Käyttöjännite voitaisiin yhtä hyvin tuoda laitteille Raspberry Pi'n USB-virtapistokkeen kautta. Tällöin liitin kuitenkin jäisi ilman tukea ja voisi aiheuttaa laitteiden rikkoutumisen taipuessaan. Kosketusnäyttö on liitettyä valkoisella lattakaapelilla. Taustalla näkyvät USB- ja Ethernet-pistokkeet, joihin on kytketty näppäimistö, hiiri ja verkko-kaapeli.



Kuvio 23. Raspberry Pi'n kytkennät

Arduino Nano -kehitysalusta liitetään RS-485-väyläadapteriin kuvion 24 mukaisesti. Kuviossa 24 ovat vasemmalla Nano ja oikealla RS-485-adapteri.



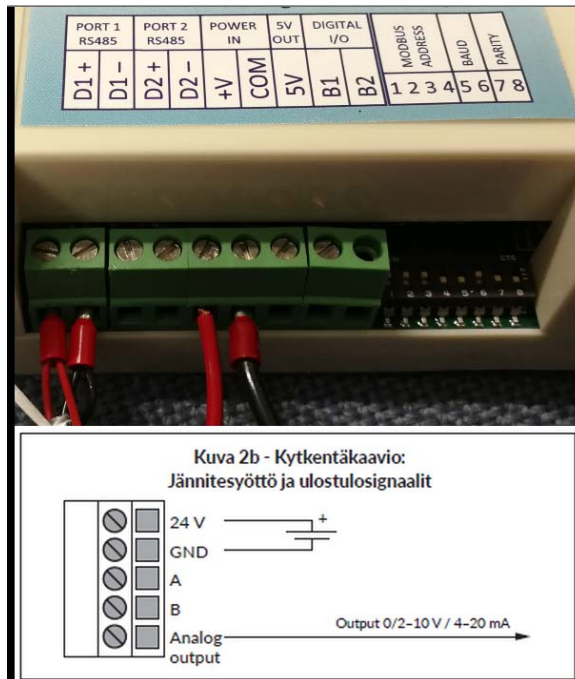
Kuvio 24. Arduino Nano Modbus -siltakytkennät.

Tarkemmat johtimien kytkennät:

- Nano: D3 – MAX: DE
- Nano: D2 – MAX: RE
- Nano: D11 – MAX: R0
- Nano: D10 – MAX: DI
- Nano: 5 V – MAX: VCC
- Nano: GND – MAX: GND

RS-485-adapteri ei tarvitse erillistä virtalähdettä. Käyttöjännite tulee Arduino Nanon kautta. Adapterin vihreään liittimeen kytketään väyläkaapeli niin, että punainen johdin kytketään liittimeen A ja musta liittimeen B. Kaapelissa olevaa suojajohdinta ei tarvitse kytkeä. Suojajohdinta käytetään tarvittaessa häiriöherkissä- ja/tai pitkänmatkan asennuksissa.

Käytettävät Modbus-laitteet kytketään kuvion 25 mukaisesti. Kuviossa yläosan EnOcean-adapterin D1+ -liittimeen kytketään väyläkaapelin punainen A-johdin ja liittimeen D1- väyläkaapelin musta B-johdin. Liittimeen +V kytketään erilliseltä virtalähteeltä +24 v ja COM-liittimeen kytketään virtalähteen musta maajohdin. Kuvion 25 alaosassa HK Instrumentsin Modbus-paneelin kytkennät.



Kuvio 25. Modbus-laitteiden EnOcean-adapterin ja HK Instruments -mittapaneelin kytkennät.

EnOcean-adapterin väyläasetuksia säädetään dippikytkimillä kuvion 26 mukaisesti. Harjoituksessa dippikytkimet on säädetty vastaamaan osoitetta 8, 9600 Baud ja Parity: None.

Table 4a: DIP Switches - Modbus Address

| Modbus Address | Switch 1 | Switch 2 | Switch 3 | Switch 4 |
|----------------|----------|----------|----------|----------|
| 1 | OFF | OFF | OFF | OFF |
| 2 | ON | OFF | OFF | OFF |
| 3 | OFF | ON | OFF | OFF |
| 4 | ON | ON | OFF | OFF |
| 5 | OFF | OFF | ON | OFF |
| 6 | ON | OFF | ON | OFF |
| 7 | OFF | ON | ON | OFF |
| 8 | ON | ON | ON | OFF |
| 9 | OFF | OFF | OFF | ON |
| 10 | ON | OFF | OFF | ON |
| 11 | OFF | ON | OFF | ON |
| 12 | ON | ON | OFF | ON |
| 13 | OFF | OFF | ON | ON |
| 14 | ON | OFF | ON | ON |
| 15 | OFF | ON | ON | ON |
| 16 | ON | ON | ON | ON |

Table 4b: DIP Switches - Baud Rate

| Baud Rate | Switch 5 | Switch 6 |
|-----------|----------|----------|
| 2400 | OFF | OFF |
| 4800 | ON | OFF |
| 9600 | OFF | ON |
| 19200 | ON | ON |

Table 4c: DIP Switches - Parity

| Parity | Switch 7 | Switch 8 |
|--------|----------|----------|
| None | OFF | OFF |
| Even | ON | OFF |
| Odd | OFF | ON |
| None | ON | ON |

Please note if DIP Switches are altered while power is on then no change will take effect until power is re-applied.

Kuvio 26. EnOcean-adapterin asentokytkinkaavio (Instruction manual 2011.)

12.2 Rasbian-käyttäjärjestelmän asennus

Helpoin tapa asentaa Rasbian-käyttäjärjestelmä on noudattaa Rasbianin asennusohjeita osoitteesta ”<https://www.raspberrypi.org/downloads/>”. Ensisijainen vaihtoehto aloittelijoille on käyttää NOOBS-pakettia. Seuraava aste on asentaa täydellinen jakeluversio ilman NOOBS-lisäosia. Kummassakin tapauksessa Rasbianin asennusohjetta käytettäessä asennetuksi tulee aina uusin jakeluversio.

Opinnäytetyössä ilmeni yhteensopivuusongelmia uusimman Rasbian Stretch-version ja Domoticz-käyttöliittymän kanssa. Tätä aihetta käsitellään tarkemmin raportin loppupuolella. Tästä syystä asennusesimerkkinä käytetään vaihtoehtoista asennustapaa. Jakeluversioiden vaihtoehtoiset versiot ovat saatavilla erillisen arkiston kautta. Joissain tilanteissa ei myöskään ole tarkoituksenmukaista asentaa täyttä käyttäjärjestelmää, jolloin voidaan valita jakeluversioista Lite-versio. Lite-versio sisältää käyttäjärjestelmän ytimen, eikä siinä ole asennettuna esimerkiksi työpöytänäkyä. Halutut ominaisuudet asennetaan manuaalisesti ensikäynnistyksen jälkeen avautuvasta konsolista.

Käyttäjärjestelmän asentamiseen tarvittavat liitteet ovat

- Liite 1: Rasbian-jakeluversion lataaminen ja asentaminen muistikortille.
- Liite 2: Rasbian-manuaaliasennus Raspberry Pi -alustalle, täysversio
- Liite 3 Rasbian NOOBS -asennus Raspberry Pi -alustalle, täysversio.
- Liite 4: Rasbian Lite -muistikortin valmistelu ja asennus.

Liitteistä voi valita kulloiseenkin laboratorioharjoitukseen tai harjoituskentekijän osaamistasoon parhaiden sopivan osion. Näillä ohjeistuksilla saadaan käyttövalmis Raspberry Pi, ja ohjeita voidaan soveltaa myös muissa Raspberry Pi’ta käyttävissä harjoituksissa kuin raportissa esitelty tehtävänannon mukainen sovellus.

Rasbian Lite-käyttäjärjestelmän asennus käydään tarkemmin läpi liitteessä 5.

12.3 Rasbian-käyttäjärjestelmän ohjelmistoasennukset

Rasbian-käyttäjärjestelmän asennuksen jälkeen järjestelmään piti asentaa tarvittavat lisäohjelmistot, joilla laitteistokokoonpanoa päästäisiin testaamaan. Tämä ei ollut aivan niin yksiselkoinen vaihe kuin työn alussa muodostunut mielikuva antoi olettaa,

eivätkä aikaisemmin mainitut käyttäjärjestelmän- ja Domoticz-ohjelmistoversioiden väliset ongelmat olleet ainoat ilmenneet yllätykset. Aihetta käsitellään tarkemmin raportin loppupuolella.

Raportin liitteessä 6 käydään läpi tarkemmin vaihe vaiheelta tarvittavat ohjelmistoasennukset.

12.3.1 Domoticz-käyttöliittymän asennus

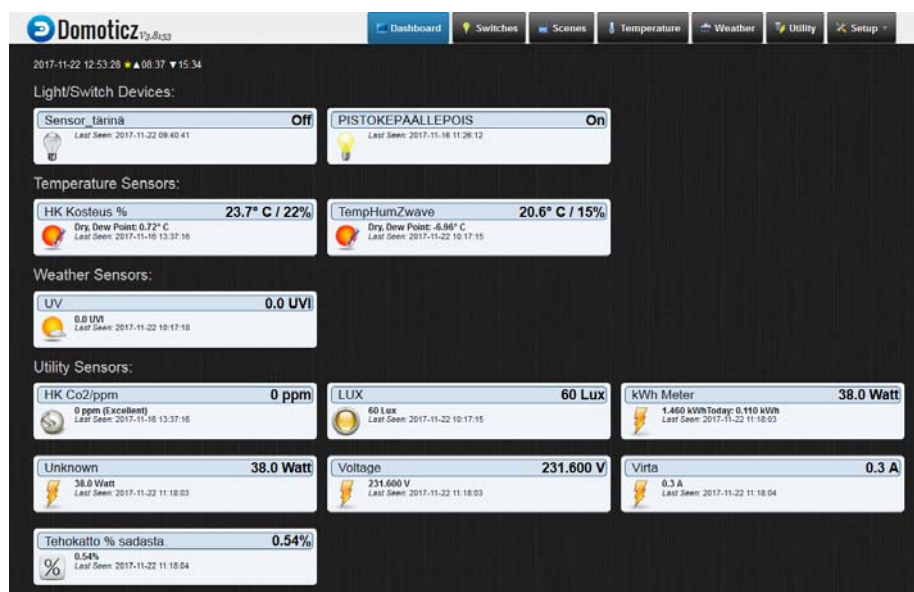
Domoticz-käyttöliittymä voidaan asentaa yksinkertaisesti yhdellä komentorivikäskyllä (ks. kuvio 27).

```
curl -L install.domoticz.com | sudo bash
```

Kuvio 27. Domoticz-käyttöliittymän asennuskomento

Mainitulla käskyllä asennus on yksinkertaista ja helppoa asennusohjelman ohjeistuksella avustettuna.

Kuviossa 28 on toimivan Domoticz-käyttöliittymän esimerkki-ikkuna, josta nähdään liitettyjen Z-Wave- ja Modbus-järjestelmien reaaliaikaista mittatietoa.



Kuvio 28. Domoticz-käyttöliittymän esimerkinäkymä valmiista kokoonpanosta

12.3.2 Arduino IDE -ohjelmiston asennus

Työn alussa tarkoituksena oli asentaa Arduino IDE -ohjelmointipaketti suoraan Raspberry Pi -alustalle. Tällöin Arduino-Modbus-siltaa voisi ohjelmoida suoraan harjoituslaitteistosta, eikä USB-kaapelia tarvitsisi kytkeä jatkuvasti edestakaisin ohjelmointitietokoneen ja Raspberry Pi'n välillä. Ongelmaksi tässä kuitenkin osoittautui automaattisesti asentuvan Arduino IDE -ohjelmiston versio 1.0.5. Ohjelmistoversio on huomattavan vanha. Työn toteutusvaiheessa uusin versio oli kuitenkin 1.8.5. Raportin liitteessä 8 ohjelmistoasennus käydään tarkemmin läpi.

Arduino IDE -ohjelmisto voidaan asentaa komentoriviltä komennolla:

```
sudo apt-get install arduino
```

Komento asentaa ohjelmistosta uusimman saatavana olevan ohjelmistoversion käytössä olevalle käyttöjärjestelmälle. Rasbian Jessie -jakeluversiolle ei ole julkaistu Arduino IDE -ohjelmistosta uudempaa ohjelmistoversiota kuin 1.0.5. Linux-pohjaiset käyttöjärjestelmät ovat kuitenkin varsin muokattavia, ja tätä versiorajoitusta yritin manuaalisesti ohittaa Arduino- ja Domoticz-käyttäjyhteisöjen forum-keskustelujen avulla. Tämä osoittautui kuitenkin varsin työlääksi sekä aikaa vieväksi, eikä tämä myöskään soveltunut opinnäytetyön aiheajauksen sisään.

Vanhentunut ohjelmistoversio ei tukenut käytettäviä ohjelmointikirjastoja. Vanhasta versioista voitiin kuitenkin käyttää hyväksi Serial Monitor -ominaisuutta Arduino-Modbus-sillan välittämän sarjaliikenteen tutkimiseen. Tästä syystä ohjelmiston asennusta ei pudotettu kokonaan pois laboratorioharjoituksen ohjeista. Kuviossa 29 esitellään kuvakaappaus valmiin järjestelmän sarjaliikenneviestinnästä käyttäen Serial Monitor -työkalua. Serial Monitor -työkalua voidaan käyttää paikallisena vianselvitysvälineenä. Esimerkiksi kuvioista 29 löytyvä "Result write Ok: 0" ei näy Domoticz-käyttöliittymässä, mutta tällä viestillä todennetaan Modbus-väylälaitteelle onnistunut rekisterikirjoitus.

```
COM6
|
0;255;3;0;9;MCO:BGN:INIT GW,CP=R-NGA--,VER=2.1.1
0;255;3;0;14;Gateway startup complete.
0;255;0;0;17;2.1.1
0;255;3;0;11;DOMOTICZ Harjoitus, RakLa
0;255;3;0;12;1.0
0;1;0;0;22;HK Co2-ppm
0;2;0;0;6;HK temperature
0;6;0;0;7;HK Kosteus %
0;3;0;0;6;Vantage Sääasema Ulko
0;12;0;0;7;Vantage rH% ulko
0;5;0;0;8;Vantage Barometri hPa
0;255;3;0;9;MCO:BGN:STP
Result write ok:
0
Result write ok:
0
Result write ok:
0
0;255;3;0;9;MCO:REG:NOT NEEDED
0;255;3;0;9;MCO:BGN:INIT OK,TSP=NA
ModbusSlave 1
0;2;1;0;0;23.40
0;1;1;0;37;487
0;6;1;0;1;12.50
ModbusSlave 8
0;5;1;0;4;1014.20
0;3;1;0;0;19.80
0;12;1;0;1;19.00
```

Kuvio 29. Serial Monitor -työkalun kuvakaappaus toimivasta sarjaliikennekeskustelusta Arduino-Modbus-sillan ja Raspberry Pi'n välillä

12.3.3 Verkkoselaimen asennus

Rasbian-käyttöjärjestelmän Lite-versio ei sisällä mitään ylimääräistä. Kävi kuitenkin ilmi, että harjoituksia tehdessä käyttöjärjestelmään tarvitaan verkkoselain. Rasbian-käyttöjärjestelmälle on tehty useita eri selainsovelluksia. Esimerkiksi Chromium-se-lain on Googlen ylläpitämä Linux-pohjainen versio suositusta Chrome-selaimesta. La-boratorioharjoitukseen kuitenkin valittiin Epiphany-selain, joka on Raspberry Founda-tionin ylläpitämä selainsovellus ja joka näin ollen takasi parhaimman soveltuvuuden käytetyn käyttöjärjestelmän jakeluversion kanssa.

Selain asennetaan komentoriviltä komennolla:

```
sudo apt-get install epiphany-browser
```


12.4 Arduino-Modbus-sillan ohjelmointi ja liittäminen Domoticz-käyttöliittymään

Arduino Nanolla ja MAX-RS-485-adapterilla toteutettavan Modbus RTU -sillan toteuttaminen osoittautui odotettua monimutkaisemmaksi. Modbus-väylälle löytyy useampi kirjasto, ja ensimmäiseksi valitsemani SimpleModbus-kirjasto osoittautui liian pelkistetyksi, eikä sen avulla pystynyt suorittamaan tarvittavia tehtäviä laboratorioharjoitusten kannalta. Kirjasto on tarkoitettu esimerkiksi keskusteluun kahden Arduino-kehitysalustan välillä. Lopulta aikaa vieneen käyttäjäyhteisön forum-keskustelujen perusteella käytettäväksi kirjastoksi valikoitui ModbusMaster-kirjasto, johon on sisällytetty kaikki Modbus-standardin määrittelemät ominaisuudet sekä lisäominaisuutena kattava vianselvityskoodisto, jota aikaisemmin mainitussa SimpleModbus-kirjastossa ei ollut lainkaan. ModbusMaster-kirjaston valinta ei kuitenkaan ollut ongelmaton. Ongelma kuitenkin johtui suurelta osin valinnasta käyttää pelkästään Arduino Nanoa.

Alkuperäisen oletuksen mukaan Domoticz-käyttöliittymän olisi saanut ymmärtämään suoraan sopivan muotoista sarjaliikennettä. Tämä oletamus kuitenkin osoittautui testauksien jälkeen vääräksi. Ongelman sai ratkaistua sisällyttämällä sovellukseen MySensors-laiteverkon ohjelmointikirjaston. MySensors-kirjastolla Modbus-väylän laitteet voitiin mallintaa MySensors-laiteverkon mukaisella määrittelyllä. Domoticz-käyttöliittymässä on vastaavasti sisäänrakennettuna MySensors-laiteverkkotuki.

Näihin aiheisiin perehdytään tarkemmin raportin loppupuolella.

12.4.1 Arduino-Modbus-sillan ohjelmointi

Arduino-kehitysalustoja ohjelmoidaan käyttämällä aikaisemmin mainittua Arduino IDE -ohjelmistopakettia ja sen versiota 1.8.5. Aihetta käsitellään tarkemmin raportin liitteessä 8 ”Arduino-Modbus-sillan kytkennät ja ohjelmointi”.

Ohjelmisto osoittautui yksinkertaiseksi käyttää, ja tarvittavien kirjastojen asentaminen onnistui vaivatta ohjelmiston omalla kirjastonhallintatyökalulla.

Arduino-Modbus-sillan ohjelmointi aloitettiin kirjoittamalla testikoodi, jolla viestintä Modbus-väylään pystyttiin todentamaan RS-485-adapterin läpi. ModbusMaster-kirjastosta löytyy malliohjelma, joka on kirjoitettu käyttäen vastaavaa RS-485-adapteria kuin työhön valittu väyläadapteri. Ohjelmakoodissa oli kuitenkin paljon ylimääräistä pelkän testauksen tarpeisiin, joten sitä riisuttiin ja muokattiin työn tarkoituksiin sopivaksi. Lopullinen ohjelmakoodi kommentoituna löytyy raportin liitteestä 12

ModbusMaster-kirjaston merkittävin etu työn kannalta oli sen sisältämä sisäänrakennettu vianselvitysohjelma. Kirjaston avulla luotu ohjelma laskisi automaattisesti jokaisesta ohjelman väyläkomennosta tarkistusarvon, jolla mahdollisia ongelmia voitiin selvittää.

Testauksessa Arduino Nanolla ajettiin väylästä lukukomentoa HK Instrumentsin mittapaneelille:

```
ModbusMaster node;

Serial.begin(9600);

node.begin(1, Serial);

result = node.readInputRegisters(1, 16);

if (result != node.ku8MBSuccess)
{
    Serial.println("READInputTfail");
    Serial.println(result);
}

if (result == node.ku8MBSuccess)
{
    Serial.println("READINPUTSOK");
    Serial.println(result);
}
```

Ohjelmakoodi lähettää ennalta määritetylle väylälaitteelle komennon lukea ja lähettää takaisin tiedot rekisteripaikasta 1 alkaen seuraavat 16 rekisteripaikkaa. Komento "readInputRegisters" vastaa Modbus-standardin funktiota 4. ModbusMaster-kirjaston komennot palauttavat aina sisäisen tarkistussumman, joka tallennetaan muuttujaan result. Tätä arvoa voidaan sitten verrata sisäisiin vianselvitysarvoihin. Testauksessa käytettiin kuitenkin hyväksi yleistä ".ku8Success"-arvoa, joka onnistuneella viestillä saa arvon 0. Väyläkomennon jälkeen saatua arvoa verrataan oletukseen onnistuneesta viestistä vertailulla "==" ja epäonnistuneeseen vertailulla "!=" , ja molemmissa tapauksissa saatu tarkistussumma tulostettiin sarjaporttiin. Testauksessa käytettiin kaikkia kirjaston lukukomentoja. Kun kaikista saatiin onnistuneet tarkistussummat, ohjelmaan lisättiin ohjelmakoodia, jolla laitteen lähettämät rekisteriarvot voitiin tulostaa sarjaporttiin tutkittavaksi. Tällä varmistettiin väylän kunnollinen toiminnallisuus.

ModbusMaster-kirjaston lukukomentojen laitevastaukset tallentuvat kirjaston käyttämään ResponseBuffer-muuttujaan, jota käyttämällä voitiin tulostaa väylästä saatu konkreettinen tieto sarjaporttiin.

Testauksessa käytettiin seuraavaa komentoa väylätiedon tulostukseen:

```
Serial.println("Lämpötila:");  
  
Serial.println(node.getResponseBuffer(0));  
  
Serial.println("Ilmankosteus:");  
  
Serial.println(node.getResponseBuffer(1));  
  
jne.
```

Komennon vastauspuskuripaikkaa osoittavaa numeroa 0 muuttamalla voitiin tutkia kaikki luetut 16 rekisteripaikkaa ja tulostaa ne sarjaporttiin. Tämä tehtiin tulostamalla kaikki paikat omalla komentorivillä allekkain, jolloin uutta ohjelmakoodia ei tarvinnut ladata jatkuvasti Arduino-alustalle uudestaan. Tulostettavien rivien väliin lisättiin tulostettavaksi mittausuureen tyyppi helpottamaan testausta.

Väyläviestinnän varmistuttua ohjelmaan lisättiin MySensors-kirjaston mukainen viestintärakenne. Sen toimivuus ModbusMaster-kirjaston kanssa testattiin lukemalla yllämainittuja rekistereitä väylästä ja kirjoittamalla saadut tiedot MySensors-kirjaston kautta sarjaporttiin, sekä seuraamalla viestintälinkin toimintaa Serial Monitorin kautta. MySensors-kirjastolla testausohjelmaan lisättiin mittauksen esittely ja mitaustiedon välittäminen käyttöliittymään. Kommentojen tarkempi selvitys ja lopullinen ohjelmakoodi on sisällytetty laboratorioharjoitukseen liitteessä 8.

ModbusMaster- ja MySensors-kirjastojen yhdistelmäohjelman testauskoodi:

```
MyMessage msgnode_Temp(node_ID_TEMP, V_TEMP);

present(node_ID_TEMP, S_TEMP, "HK temp");

result = node.readInputRegisters(0, 6);

if (result == node.ku8MBSuccess)
{
  Serial.println("node: HK-paneeli:");

  send(msgnode_Temp.set(node.getResponseBuffer(0x03) / 10.0f, 2));
}
```

Testausohjelman toimivuuden todentamisen jälkeen ohjelmaan muokattiin lähemmäksi lopullista muotoaan (ks. liite 12). Väylään lisättiin toinen testilaite, ja lukukomentoja lisättiin. Tämän jälkeen siirryttiin testaamaan viestinnän toimintaa Domoticz-käyttöliittymään.

12.4.2 Arduino-Modbus-sillan liittäminen Domoticz-käyttöliittymään

Asennuksessa ja testauksessa seurattiin Domoticz-käyttöliittymän MySensors-laiteverkon asennusohjeita. Arduino-Modbus-sillan liittäminen Domoticz-käyttöliittymään on käsitelty tarkemmin liitteessä 9.

Testausvaiheessa ilmeni muutamia mainittavia ongelmia, jotka liittyivät Domoticzin tapaan käsitellä eri laitejärjestelmistä tulevaa tietoa. Tämä ongelma oli sama Z-Wave-järjestelmän kanssa.

Domoticz kerää samankaltaisia mittauksia yhden mittapisteen alle kuvion 30 mukaisesti. Tämä on Domoticzin ohjelmistoon sisäänrakennettu ominaisuus, johon ei opinäytetyön aikana pystytty puuttumaan. Omaisuus on tietyissä tilanteissa ymmärrettävä, mutta tähän kaivattaisiin mahdollisuutta valita, käytetäänkö yhdistettyjä mittapisteitä vai näytetäänkö kaikki erillisenä. Ongelma vaikuttaa vain käyttöliittymässä näytettäviin tietoihin, joten tietoja voidaan ohjelman sisällä käsitellä yksittäisenä mittauksina.



Kuvio 30. Domoticz-käyttöliittymän mittapisteongelma

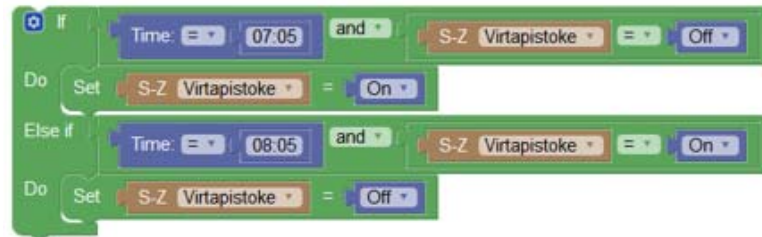
12.4.3 Z-Wave-laiteverkon asennus Domoticz-käyttöliittymään

Z-Wave-laitejärjestelmän asennus oli työn toteutuksen kannalta helpoin osuus, eikä sen käyttöönotto poikennut Domoticzin käyttöohjeiden mukaisesta asennuksesta. Tämä oli positiivinen yllätys, koska tilatun RaZberry2-ohjaimen toimituksessa ilmenneiden viivästysten takia se saatiin käyttöön vain hetkeä ennen raportin palautusta. Z-Wave-laitteiden liittäminen Domoticz-käyttöliittymään käsitellään kokonaisuudessaan laboratorio-ohjeistuksessa liitteessä 7.

12.5 Domoticz-käyttöliittymän tapahtumaohjelmointi

Sellaisenaan Domoticz-käyttöliittymä ja siihen liitetyt laitejärjestelmät toimivat hyvin tiedonkeruu- ja tallennusohjelmistona. Käyttöliittymä on kuitenkin pohjimmiltaan tarkoitettu automaatio-ohjaimeksi. Automaatioon tutustumista Domoticz tukee sisällyttämällä yksinkertaisen ja toimivan tapahtumaohjelmointityökalun. Työkalulla mittatietojen avulla voidaan ohjelmoida erilaisia tapahtumia. Liiketunnistin laukaisee omistajan poissa ollessa murtohälytyksen päälle tai ilmankosteus voi automaattisesti kytkeä poistoilmakojeen päälle ja pois. Esimerkkejä on monia. Laboratorioharjoitukseen on sisällytetty harjoitukset kahdesta yleisimmästä lähestymistavasta aiheeseen.

Tarkemmin aiheet käydään läpi liitteissä 10 ja 11. Liitteissä käsitellään Blockly-ohjelmointi kuvion 31 esimerkillä ja monimuotoisempi LUA-script-ohjelmointi kuvion 32 esimerkillä. Näihin harjoituksiin on sisällytetty myös muutamia tehtävään liittyviä kysymyksiä, eikä kaikkia vaiheita ole selitetty liian yksityiskohtaisesti – tavoitteena rikkoo putkinäköinen ohjeiden suorittaminen ilman asian sisäistämistä.



Kuvio 31. Blockly-ohjelmoinnin esimerkkiohjelma

```
time = os.date("!*t")
if time.hour == ReportHour and time.min == ReportMinute then

    if (otherdevices[sensor2] > '20') then

        commandArray[sensor1] = 'On'

    end

end

if time.hour == ReportHour_2 and time.min == ReportMinute_2 then

    if (otherdevices[sensor1] == 'On' and otherdevices[sensor2] < '10') then

        commandArray['Virtapistoke'] = 'Off'

    end

end

return commandArray
```

Kuvio 32. LUA-script-ohjelmoinnin esimerkkiohjelma

12.6 Työn toteutuksessa ilmenneitä ongelmia

Työn edetessä ilmaantui useita mainittavia ongelmia. Ensimmäinen vastaan tullut, oleellisesti työhön vaikuttanut ongelma liittyi työn toteutuksen aikaan julkaistuihin Rasbian-käyttöjärjestelmän ja Domoticz-käyttöliittymän ohjelmistopäivityksiin. Domoticzin uusin versio julkaistiin 30.7.2017. Rasbian Stretchin ensijulkaisu oli

17.7.2017. Ongelmat tulivat yleisesti esiin vasta Rasbian Stretchin ensimmäisen päivityksen 8.9.2017 jälkeen. Tuolloin Rasbian-käyttöjärjestelmän käyttämiä OpenSSL-turvallisuuskirjastoja päivitettiin, ja vanhentuneet kirjastot poistettiin käytöstä kaikilta osin. Domoticzin käyttäjyhteisö suositteli asentamaan vanhat openSSL-kirjastot käsin, mutta tämäkin vaihtoehto oli poistunut käytöstä ennen opinnäytetyön aloittamista. Varmuuden vuoksi tätäkin vaihtoehtoa kokeiltiin, kuitenkin tuloksetta, jolloin päädyttiin valitsemaan vanhempi Rasbian Jessie -jakelu, jonka viimeisin päivitys oli 5.7.2017 ja jolle 30.7.2017 julkaistu Domoticzin uusin vakaa versio julkaistiin.

Toinen merkittäviä muutoksia alkuperäiseen suunnitelmaan aiheuttanut ongelma syntyi valinnasta käyttää Arduino Nano -kehitysalustaa. Ongelmaa ei olisi syntynyt, jos väyläviestinnän pystyisi hoitamaan pelkästään käyttäen ModbusMaster-kirjastoa ja viestien välitystä sellaisenaan Domoticzille, tai Arduino Nanoa käytettäisiin pelkästään MySensors-kirjaston mukaisesti sen omassa laiteverkossa isäntänä. Molempien kirjastojen yhdistäminen kulutti Nanolta lähes kaiken käytettävissä olleen muistin. Pelkkä kahden väylälaitteen mittatietojen luku ja ilmoittaminen Domoticzin käytettäväksi vei 92 % käytettävissä olevasta muistista. Tällaisesta kuormasta tulee ohjelmaa ladatessa varoitus, ja sen pystyi käytännön sarjaporttiliikenteellä todentamaan. Ohjelmaa joutui suurelta karsimaan, ja alkuperäisessä tehtävänannossa vaadittua väylään kirjoittamista jouduttiin miettimään uudestaan. Tavoite oli saada suora kontakti Domoticzin käyttöliittymästä Arduino-Modbus-sillan väylään kirjoituskomentoihin. Tämä osuus jouduttiin toteuttamaan karsittuna, liitteen 8 ohjeiden mukaisesti.

Kolmas merkittävä ongelma johtui myös valinnasta käyttää Arduino Nano -kehitysalustaa Modbus-siltana. Ongelman syy selvisi liian myöhään pitkän vianselvityksen jälkeen, jotta käyttöön valittu Arduino olisi voitu vaihtaa järeämpään versioon. Ongelma liittyy Nanon laitetason sarjaporttien määrään: niitä on vain yksi, ja se on linkitetty myös kehitysalustan USB-porttiin, jota myös käytetään keskusteluun Domoticz-käyttöliittymän kanssa. Kaikkien ohjelmaosioiden mahduttaminen yhteen laitetason-porttiin osoittautui mahdottomaksi, ja Nanolla täytyi yrittää simuloida sarjaporttia ohjelmallisesti käyttäen GPIO-pinnejä. Tarkoituksena oli käyttää ohjelmallista sarjaporttisimulaatiota keskusteluun Modbus-väylään. Kaikki toimi hyvin, kunnes väylään yritettiin kirjoittaa rekistereitä. Ohjelmallinen sarjaportti on riippuvainen ohjelmalli-

sista ajastimista, joita se käyttää pulssittamaan kytkentäpinnejä. Ohjelmalliset ajastimet ovat riippuvaisia Nanon sisäisistä keskeytyskomennoista. ModbusMaster ja MySensors-kirjastot käyttävät samoja keskeytyskomentoja toiminnoissaan. Näin ollen pitkät viestit, joita väylään kirjoittaminen vaatii, eivät pysyneet kaikkien ominaisuuksien ollessa yhtä aikaa käytössä ajallisesti kasassa, ja ModbusMaster-kirjaston sisäinen tarkastus hylkäsi viestit aikaviiveongelmien takia. Vianselvityksen lomassa kävi kuitenkin ilmi, että laitteet kirjoittivat rekistereihin tietoa ohjelmakomentojen mukaan. Modbus-väylä tarvitsee kuitenkin vastauksen, ja siitä lasketun tarkistussumman tulee täsmätä. Ajastusongelmista johtuen ohjelmallinen sarjaportti ei pystynyt pitämään tuloportteja auki tai kiinni oikeaan aikaan, jolloin tarkistussummat eivät täsmänneet.

Davisin Vantage-sääaseman Modbus-adapteri ilmoittaa mittaukset oletuksena tuumamitoilla, vaikka itse sääaseman käyttökonsoli olisi asennettu metrisenä. Ominaisuus ilmeni testausvaiheessa. Adapterin asetuksia pystyy muuttamaan kirjoittamalla manuaalissa mainittuihin rekistereihin sopivat arvot. Adapterin asetuksia kirjoitettaessa kävi kuitenkin ilmi, että vaikka Arduino-väyläohjelma ilmoitti viestintävirheestä rekistereitä kirjoitettaessa, rekistereihin lopulta tallentui halutut muutosarvot. Tämä ilmeni, kun rekistereitä tarkistettiin lukemalla ja tulostamalla niiden arvot sarjaporttiin. Ongelmaa tutkittiin lisää, ja ohjelmakoodia voitiin optimoida niin, että väylään kirjoittaminen ja sen todentaminen voitiin sisällyttää laboratorioharjoituksiin.

Näistä syistä rakennusautomaatiolaboratorion harjoitustehtävät ovat siinä muodossaan, kuin ne laitteistolla annetussa aikataulussa oli mahdollista toteuttaa.

13 Johtopäätökset ja pohdinta

Työ oli monilta osin hyvin palkitsevaa, vaikkei kaikkiin tavoitteisiin päästy toivotussa laajuudessa. Varsinkin reaaliaikaisen väylään kirjoittamisen puuttuminen käyttöliittymästä jäi henkilökohtaisesti vaivaamaan. Nykyisellään väylään kirjoittaminen on monimutkaista eikä palvele esimerkkinä oikeaa käyttöliittymän kautta tapahtuvaa laitehallintaa. Väylään reaaliaikaisen kirjoittamisen voisi toteuttaa esimerkiksi vaihtamalla käytettävä Arduino-kehitysalusta tuotevalikoiman Mega-versioon. Megassa on Nanosta poiketen useampi laitetason sarjaporttiliittymä, joista väyläviestinnälle voi-

taisiin varata portti ja käyttöliittymään päin toinen. Tämä mallintaisi samalla käytännönsovelluksia paremmin. Megan neljästä portista kolmea voitaisiin käyttää väyläviestintään eri väyläverkkojen kanssa ja yhtä käyttöliittymään päin. Megan suuremmat muistiresurssit myös edesauttaisivat laajempien laiteverkkojen ylläpitoa. Myös käytettyjä ohjelmakirjastoja voisi optimoida käyttämään vähemmän Arduinon muistiresursseja. Ohjelman optimoinnilla järjestelmän laajuutta voitaisiin kasvattaa säilyttämällä samalla järjestelmän vakaus.

Työssä ilmennyt Rasbian-käyttöjärjestelmän ja Domoticz-käyttöliittymän versioiden yhteensopivuusongelma toi esiin hyvin yhden isoista avoimen lähdekoodin ongelmista: Sovelluksia ylläpidetään usein pienellä, muutaman hengen kehitystiimillä, eivätkä ne välttämättä pysy perässä liittyvien ohjelmistojen suurissa muutoksissa. Versio-ongelmien ydin olivat Rasbian-käyttöliittymän päivittyneet openSSL-turvallisuuspäivitykset. Vanhentuneet openSSL-turvallisuuskirjastomääritykset ovat Domoticz-käyttöliittymään syväle sisäänrakennettuna, ja niiden päivittäminen vaatisi todennäköisesti merkittäviä muutoksia käyttöliittymän ohjelmakoodiin. Turvallisuuskirjastojen versioerot estivät Domoticzin käynnistymisen Rasbian-käyttöjärjestelmässä. Ongelman selvittäminen ja kiertäminen asentamalla vanha Rasbian-käyttöjärjestelmä toi työn toteusvaiheeseen turhaa monimutkaisuutta ja kulutti aikaa muista työlle asetetuista tavoitteista. Ilman päivitettyjä turvallisuuskirjastoja Domoticzia ei voi suositella käytettäväksi etäyhteyden kautta sisäisen verkon ulkopuolelta. Etäohjattavuuden puuttuminen turvallisuusongelmien takia on merkittävä huomioon otettava tekijä, jos sovellusta haluaisi laajentaa käytännön teollisuus- tai rakennusautomaatioteutukseen.

MySensors-laiteverkon ohjelmakirjasto oli pakollinen valinta ohjelmakoodiin työn aikataulun kannalta. Sen käyttäminen voitaisiin kuitenkin välttää kirjoittamalla Modbus-standardin mukainen järjestelmäyhteys Domoticzin sisäiseen ohjelmistoon. Kokonaisen ohjelmiston kirjoittaminen toisen järjestelmän sisään sopisi sellaisenaan ohjelmistotekniikan harjoitus- tai opinnäytetyön aiheeksi. Modbus-yhteyden ohjelmointi käyttöliittymän päähän vapauttaisi edelleen Arduino-kehitysalustan rajallisia resursseja. MySensors-ohjelmiston eduksi voidaan työssä esitellyssä sovelluksessa laskea sen tapa toteuttaa mittauksista muodostettavat virtuaalilaitteet ja niiden esit-

täminen käyttöliittymään. Tavoitteena oli tämän ominaisuuden kautta luoda reaaliaikainen kirjoitusyhteys Modbus-väylään käyttämällä virtuaalikytkimiä, joilla eri Modbus-väylän funktioita olisi voinut ajaa erillään syklisestä ohjelmakoodista. Kirjoituskäsky menisi käyttöliittymästä väylänsillan komentoa vastaavalle virtuaalikytkimelle, joka pysäyttäisi syklisen ohjelman ja kirjoittaisi väylän vapauduttua halutun komenon väylään ModbusMaster-kirjaston avulla. Tällaisella toteutuksella väyläviestintä voitaisiin varmistaa ModbusMaster-kirjaston sisältämän viantarkistusfunktion kautta. Tavoite jäi kuitenkin saavuttamatta eteen tulleiden ongelmien kulutettua työlle varattua aikaa.

Kuluttaja-asiakkaille kotikäyttöön tarkoitettu Z-Wave-laitejärjestelmä osoittautui järjestelmän mainontaa vastaavaksi. Vaikka työssä käytettiin Domoticz-käyttöliittymän omaa vaihtoehtoista Openzwave-rajapintaa (vastaa laitetoimittajan omaa ohjelmistoa pienillä muutoksilla laajemman yhteensopivuuden takaamiseksi), järjestelmän käyttöönotto oli yksinkertaista ja laitteiden hallinta helppoa. Ottaen huomioon työn muiden osakokonaisuuksien tuomat ongelmat ja Z-wave-laitteiden myöhäinen toimitus odotin tästä laiteosuudesta tulevan paljon monimutkaisempi. Odotukset olivat kuitenkin turhia ja Z-Wave-järjestelmää voi suositella sellaisenaan sen omalla käyttöliittymällä tai sulautettuna työssä esitellyllä tavalla yhtenäisen käyttöliittymän alla. Laitejärjestelmän kustannukset voivat kotikäytössä tosin nousta varsin korkealle. Yksittäisten laitteiden hinnat vaihtelevat 40–100 euroon ominaisuuksista riippuen. Korkea hinta kuitenkin takaa vaivattoman järjestelmän koti- ja rakennusautomaation toteutuksiin. Verrattavissa oleva MySensors-laiteverkko lähestyy automaatiota enemmän tee-se-itse-näkökulmasta ja vaatii keskitason osaamista Arduino IDE -ohjelmoinnista sekä laitepisteiden käsin rakentamista irrallisista komponenteista. Vaikkei tätä ominaisuutta työssä käsitelty, vaatii se erityismaininnan vaihtoehtoisena valintana Z-Wave-järjestelmälle. Muutaman mittauksen MySensors-laitepiste irto-osina maksaa 20–30 euroa. Järjestelmä kuitenkin vaatii enemmän aikaa ja kiinnostusta laiteläheisempään harrastukseen.

Domoticzin vakioasennus on sellaisenaan varsin toimiva käyttöliittymä, ja tapahtumaohjelmointi odotettua monimuotoisempi. Käyttöliittymää voidaan kuitenkin helposti muokata tapauskohtaisesti paremmin sopivaksi. Esimerkiksi Domoticzin

Floorplan-ominaisuutta voisi hyvin käyttää hyväksi laajemmassa rakennusautomaatiosovelluksessa. Rakennuksen pohjapiirustusten lataaminen sopivassa muodossa harjoituksessa esitellyllä tavalla Domoticzin tiedostoihin ja pohjasuunnitelmaan liitettävien laitepisteiden yhdistelmä poistaisi tarpeen irrallisten piirustusten selailuun. Lisäksi laitteet voisi ongelmatilanteissa paikallistaa suoraan käyttöliittymästä ilman erillistä ristiintarkastusta muista tiedostoista. Domoticzin valmiita laite- ja mittauskoneja voi myös korvata omilla kuvapohjilla, jolloin käyttöliittymästä voidaan muokata paremmin toimiva kokonaisuus kulloiseenkin sovellukseen.

Työssä perehdyttiin tarkemmin Modbus-väylän käyttöön ja sen liittämiseen käyttöliittymään. Toteutuksesta saadun tiedon pohjalta voidaan kuitenkin hahmottaa Domoticzin potentiaali laajemmassakin käyttötarkoituksessa. Sovelluksen laajentaminen laboratorioharjoituksesta käytännön sovellukseen vaatii kuitenkin esille tuotujen ongelmakohtien pohdintaa käyttökohteen tarpeet huomioon ottaen.

Opinnäytetyö vahvisti myös hyvin omaa aikaisempaa harrastetason tietämystä käytetyistä Arduino-kehitysaluista ja Linux-pohjaisesta käyttöjärjestelmästä. Työn alussa oman osaamisen riittävyys herätti epäilyjä tavoitteiden saavuttamisesta. Epäilyt kuitenkin osoittautuivat liioitelluiksi. Vaikka osaamistasoon nähden ohjelmointiin liittyvät ongelmat olivat haastavia, niihin löytyneet ratkaisut palkitsivat ja kannustivat pitämään tavoitteista kiinni.

Työn tuloksena syntyneet laboratorioharjoitukset suoritettiin toimeksiantajan puolesta ja hyväksyttiin rakennusautomaatiolaboratorion käyttöön.

Lähteet

About Domoticz, 2017. Wikipedia. Domoticz. Viitattu 24.10.2017.

https://www.domoticz.com/wiki/About_Domoticz

Alanen, S. 2014. Integraatio rakennusautomaatiossa. Opinnäytetyö.

Automaatiotekniikka. Jyväskylän ammattikorkeakoulu. Viitattu 24.11.2017.

Asennusohje CDT-MOD-2000, 2016. PDF-asennusohje. HK Instruments. Viitattu 21.11.2017.

<http://hkinstruments.fi/wp-content/uploads/2017/08/CDT-MOD-Asennusohje-4.0.pdf>

Asennusohjeet, 2017. Verkko-ohjeistus. Raspberry Pi Foundation. Viitattu 2.11.2017

<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

Blockly, 2017. Wikipedia. Viitattu 15.11.2017.

<https://en.wikipedia.org/wiki/Blockly>

Compare board specs, 2017. Arduino alustojen vertailutaulukko. Arduino.cc. Viitattu 24.10.2017.

<https://www.arduino.cc/en/Products/Compare>

Domoticz and Scripting, 2017. Wikipedia. Domoticz. Viitattu 24.10.2017.

https://www.domoticz.com/wiki/Domoticz_and_Scripting

Google for Education – Blockly, 2017. Verkko-ohjeistus. Google. Viitattu 15.11.2017.

<https://developers.google.com/blockly/>

Hakkarainen, T. N.d. Talo- ja turvatekniikka tulipalossa: Nykytilanne ja tulevaisuus. Tietopaketti. VTT. Viitattu 21.11.2017.

Installing Additional Arduino Libraries, 2017. Verkko-ohjekirja. Arduino.cc. Viitattu 31.10.2017

<https://www.arduino.cc/en/Guide/Libraries>

Instruction manual, 2011. GWY-141 VantagePro2 Modbus Gateway V5.4

Käyttöohjekirja, PDF. EnOcean. Viitattu 21.11.2017

Introduction, 2017. Verkkoesite. Arduino Viitattu 24.10.2017.

<https://www.arduino.cc/en/Guide/Introduction>

Karttunen, H. 2009. Kiinteistöautomaation etähallinta. Opinnäytetyö. Lahden ammattikorkeakoulu. Ohjelmistotekniikka. Viitattu 31.10.2017

http://www.theseus.fi/bitstream/handle/10024/3453/Karttunen_Henri.pdf;jsessionid=003B0FBFC69F19103F9D6ABE42729B19?sequence=1

Modbus FAQ: About the Protocol, 2017. FAQ. Modbus Organization inc. Viitattu 19.10.2017

<http://www.modbus.org/faq.php>

Modbus Messaging Time, 2017. Asiakaspalveluviesti. Continental Control Systems, LLC. Viitattu 19.10.2017

https://ctlsys.com/support/modbus_message_timing/

ModbusMaster Library, 2017. verkko-ohjeistus. Github. Viitattu 15.11.2017.

<https://github.com/4-20ma/ModbusMaster>

Modicon: Modbus Protocol Reference Guide. 1996. Modbus Organization inc. PDF-ohjekirja. Viitattu 19.10.2017

http://modbus.org/docs/PI_MBUS_300.pdf

Module7: Serial Communication – Part 1 (Basics), 2016. Embedded Systems Course. Verkko-ohjeistus. Eeherald. Viitattu 12.10.2017

<http://www.eeherald.com/section/design-guide/esmod7.html>

Multisensor 6, 2017. Laite-esittely. Aeotec. Viitattu 15.11.2017

<https://aeotec.com/z-wave-sensor>

MySensors, 2017. Verkko-ohjeistus. Sensnology AB. Viitattu 15.11.2017.

<https://www.mysensors.org/>

Mäntylä, L. 2011. Kastellin tutkimuskeskuksen automaatioasaneerauksen tarkastelua. Opinnäytetyö. Automaatiotekniikka. Oulun seudun ammattikorkeakoulu. Viitattu 21.11.2017.

Paetz, C. 2017. Z-Way PDF-manuaali. Z-Way.Me. Viitattu 23.10.2017.

<http://rasberry.z-wave.me/index.php?id=5>

Philio PAN11-1B manual, 2017. PDF-manuaali. Z-Wave Alliance. Viitattu 15.11.2017
https://www.google.fi/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwihdsjdvMDXAhVpEpoKHUwXBCsQFggIMAA&url=https%3A%2F%2Fproducts.z-wavealliance.org%2FProductManual%2Ffile%3Ffolder%3D%26filename%3DManuals%2F1088%2FPAN11-1B%2520manual-20140816.pdf&usg=AOvVaw02C_5K65bYhV-QLffO-PfR

Rajala, A. 2014. Rakennusautomaatiojärjestelmän WEB-liitäntä. Opinnäytetyö. Automaatiotekniikka. Jyväskylän ammattikorkeakoulu. Viitattu 24.11.2017.

Raspbian FAQ, 2017. Verkko-ohjeistus. Raspbian.org. Viitattu 15.11.2017.

http://www.raspbian.org/RaspbianFAQ#What_is_Raspbian.3F

Raspberry Pi, 2017. Wikipedia. Viitattu 24.10.2017

https://en.wikipedia.org/wiki/Raspberry_Pi

RS-232, RS-422, RS-485 Serial Communication General Concepts, N.d. Artikkel. National Instruments. Viitattu 12.10.2017.

<http://www.ni.com/white-paper/11390/en/#toc4>

RS485 (2-wire, half-duplex, differential, multi-drop (32 nodes), communications standard for distances up to 4000ft.). N.d. Artikkel. RESmith Inc. Viitattu 12.10.2017

<http://www.rs485.com/pfaq.html#RS485Anchor>

RS-485 module for Arduino (MAX485), 2017. Laite-esite. Yourduino. Viitattu 31.10.2017

http://yourduino.com/sunshop//index.php?l=product_detail&p=323

Serial Communication, N.d. SparkFun. Oppimateriaali. Viitattu 12.10.2017.

<https://learn.sparkfun.com/tutorials/serial-communication>

Studholm, J. N.d. RS485 Reference Guide. Artikkel. Windmill Software Ltd. Viitattu 12.10.2017.

<http://www.windmill.co.uk/rs485.html>

User manual, 2015. Käyttöohjekirja, PDF. Davis Instruments. Viitattu 21.11.2017

<https://www.davisnet.com/support/vantage-pro2-console/>

Vesanen, J. 2016. Rakennusautomaatio tietomallissa. Opinnäytetyö. Metropolia. Automaatiotekniikka. Viitattu 21.11.2017

http://www.theseus.fi/bitstream/handle/10024/105021/Vesanen_Juuso.pdf?sequence=1&isAllowed=y

What is a function code, 2017. Taulukko, Simply Modbus, 2017. Viitattu 20.10.2017

<http://www.simplymodbus.ca/FAQ.htm#FC>

Z-Wave Explained, 2017. Kuvio. MAS-UK.COM, 2017. Viitattu 23.10.2017

<http://www.moretonalarms.com/fibarozwave.html>

Z-Wave, 2017. Wikipedia. Viitattu 19.10.2017

<https://en.wikipedia.org/wiki/Z-Wave>

Z-Wave.Me RaZberry2 EU, 2017. Laite-esittely. CeBit. Viitattu 24.10.2017

<http://www.cebit2017.de/product/z-wave.me-razberry2-eu/2273623/E155780>

Z-Wave the Gold-Standard Z-Wave Controller, 2017. Valmistajan laite-esittely. Z-Wave.Me. Viitattu 23.10.2017

<https://www.z-wave.me/index.php?id=22>

Liitteet

Liite 1. Rasbian käyttöjärjestelmän asennusvalmistelut

Asennus aloitetaan lataamalla tietokoneelle tarvittava asennustiedosto. Uusimman version löytää osoitteesta:

`"https://www.raspberrypi.org/downloads/"`.

Valittavana on tällöin NOOBS-asennus tai suora käyttöjärjestelmän asennus. Laboratorioharjoituksen sujuvuuden kannalta tarvittavat tiedostot jaetaan erillisellä tallenteella, jolloin tiedostoja ei tarvitse ladata ja voidaan siirtyä kohtaan 2 "Muistikortin valmistelu".

NOOBS-asennus poikkeaa hieman seuraavaksi esiteltävistä asennuksista ja käsitellään erikseen omalla ohjeistuksella.

Uusimman Rasbian Stretch-version ja käytettävän Domoticz-version välisistä yhteensopivuusongelmista johtuen työssä käytetään vanhempaa Jessie jakeluversiota. Jakeluversio on valmiiksi ladattuna annetulla tallenteella. On kuitenkin hyvä tietää mistä tarvittaessa löytää omiin tarpeisiin vanhemmat jakeluversiot. Näiden lataaminen ja ohjeistus käsitellään kohdassa 1.

Kohta 1. Jakeluversioiden vaihtoehtoiset asennustiedostot.

Jakelutiedostot löytyvät osoitteesta ”<http://downloads.raspberrypi.org/>”, kansiossa ”rasbian/”.

| | | |
|---|-------------------|---|
|  rasbian/ | 08-Sep-2017 10:58 | - |
|  rasbian lite/ | 08-Sep-2017 10:59 | - |

Alakansiossa ”images” on listattuna julkaisupäivämäärän mukaiset versiot rasbian käyttöjärjestelmästä. Jessie -jakelun viimeinen versio julkaistiin 5.7.2017. (Kuvio 9)

| | |
|---|-------------------|
|  rasbian-2017-07-05/ | 05-Jul-2017 17:43 |
|  rasbian-2017-08-17/ | 17-Aug-2017 09:17 |

Kansio sisältää muutamia tiedostoja. (Kuvio 10) Isoin ”.zip” pakattu tiedosto sisältää asennustiedostot. Toinen tärkeä mainitsemisen arvoinen on ”.sha1” päätteellä varustettu tiedosto. ”.sha1” tiedosto sisältää turvallisuustarkistusta varten laskentasumat, jolla käyttäjä voi erillisen ohjelman kautta tarkistaa asennustiedoston oikeellisuuden.

Index of /raspbian/images/raspbian-2017-07-05

| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|--|----------------------|-------------|--------------------|
|  Parent Directory | | | - |
|  2017-07-05-raspbian-jessie.img.info | 05-Jul-2017 17:19 | 179K | |
|  2017-07-05-raspbian-jessie.zip | 05-Jul-2017 13:05 | 1.5G | |
|  2017-07-05-raspbian-jessie.zip.sha1 | 05-Jul-2017 17:40 | 73 | |
|  2017-07-05-raspbian-jessie.zip.sha256 | 05-Jul-2017 17:40 | 97 | |
|  2017-07-05-raspbian-jessie.zip.sig | 05-Jul-2017 17:02 | 473 | |
|  2017-07-05-raspbian-jessie.zip.torrent | 05-Jul-2017 17:41 | 31K | |

Apache/2.2.22 (Debian) Server at downloads.raspberrypi.org Port 80

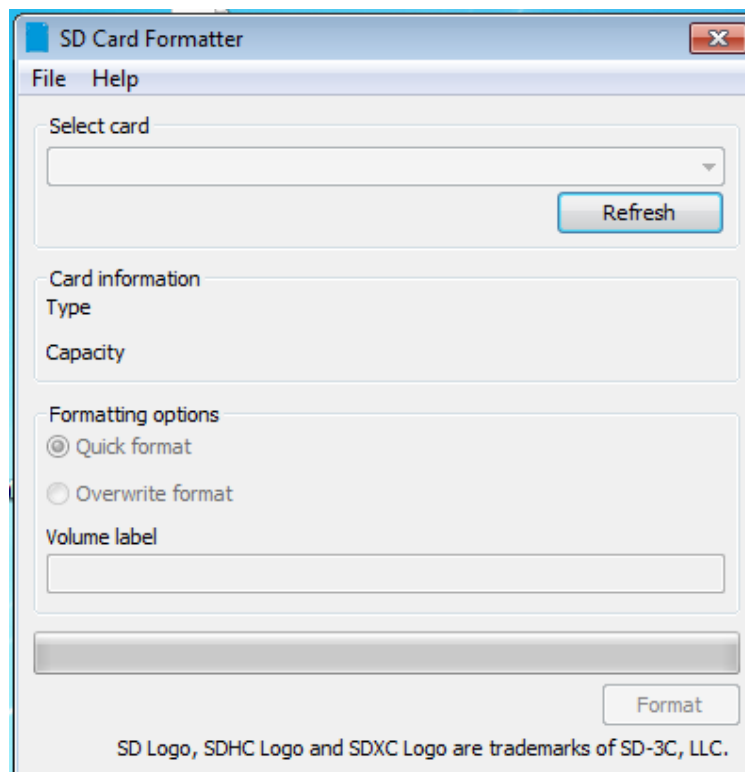
Painetaan linkistä ”2017-07-05-raspbian-jessie.zip” Linkkistä aukeaa selaimen latausvalikko, josta tiedosto tallennetaan haluttuun kansioon.

Kohta 2. Muistikortin valmistelu.

Ennen asennustiedostojen lataamista muistikortille se on hyvä alustaa. Näin varmistetaan, ettei kortilla ole mitään ylimääräistä eikä kortilla ole asennusta estäviä virheitä.

Alustukseen suositellaan erillistä ohjelmaa. Työssä käytetään ilmaista ja hyväksi todettua ohjelmaa SD Card Formater.

Ylemmästä alasvetovalikosta valitaan alustuksen kohde. Varmuudeksi ennen ohjelman käynnistämistä koneesta on suositeltavaa irrottaa muut mahdolliset siirrettävät tallennusvälineet jolloin valikossa ei näy muita tallennusvälineitä Kuvio 11.



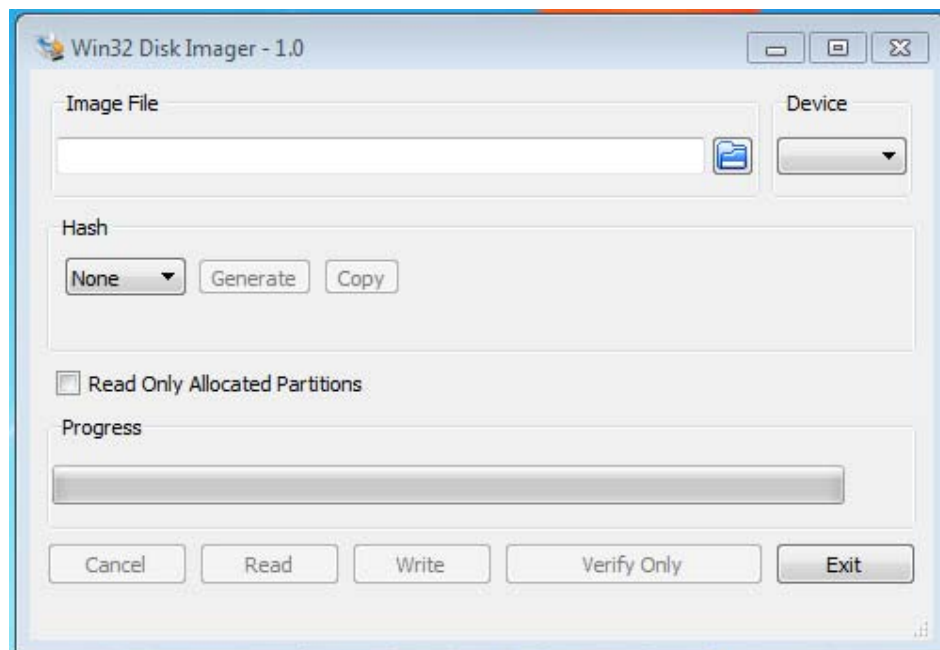
Kohta 3 Muistikortin asennus.

Onnistuneen formatoinnin jälkeen voidaan ladatut käyttöjärjestelmätiedostot asentaa muistikortille. Tämä asennusvaihe on sama kaikille Rasbianin versioille tehtäessä asennus windows-ympäristössä.

Ladataan, asennetaan ja käynnistetään Win32Diskimager

Sinisestä kansio-ikonista valitaan tietokoneelta löytyvä levykuva, joka halutaan asentaa muistikortille. Vieressä olevasta pudotusvalikosta valitaan haluttu muistikortti.

Kun asetukset on valittu, painetaan alarivin write-komentoa. Ohjelma esittää varmistuksen kirjoituksesta. Tämän voi kuitata ja jatkaa.



Liite 2. Rasbian manuaaliasennus Raspberry Pi:lle, täysversio

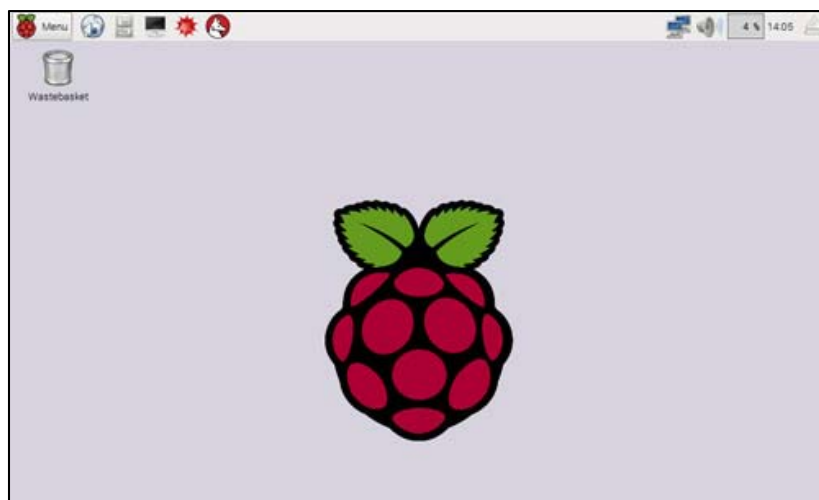
Erillisen ohjeen mukaan valmisteltu asennustallenne asetetaan Raspberry Pi alustan päästä löytyvään muistikortti paikkaan.

Liitetään Raspberry Pi:hin tarvittavat lisälaitteet.

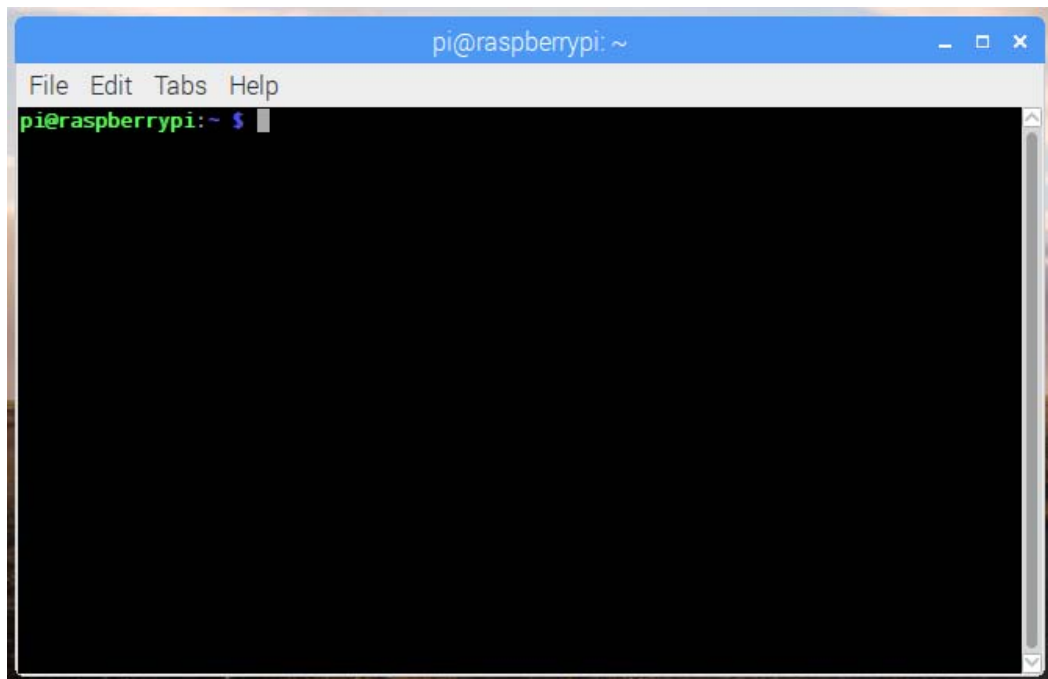
Näppäimistö ja hiiri ovat asennusvaiheessa pakollisia. Nämä on tässä vaiheessa hyvä liittää langallisesti USB-portteihin. Näin vältetään mahdollisilta ajuriongelmilta langattoman USB-adapterin kanssa.

Näiden toimenpiteiden jälkeen Pi:hin voidaan kytkeä virta.

Käynnistettäessä Raspberry Pi käynnistysohjelma käy läpi tarvittavia tiedostoja ja näytöllä rullaa nopeaan tahtiin komentoja. Käynnistys jatkaa suoraan käyttöjärjestelmän työpöydälle kuvio X. Vanhemmissa versioissa käynnistys pysähtyy komentoriville, jossa kysytään käyttäjätunnukset ennen työpöydän käynnistystä.

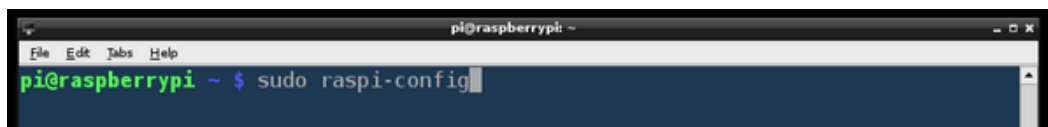


Ensimmäisenä on hyvä tehdä perusasetuksien tarkistus. Kuviossa X vasemmassa yläkulmassa näkyvästä mustasta ikonista avataan linux-ympäristölle tyypillinen terminälinäkymä (komentorivi), jossa suurin osa käyttöjärjestelmään tehtävistä asennuksista suoritetaan. Painamalla mainittua ikonia avautuu seuraavanlainen näkymä.



Kirjoitetaan komentoriville

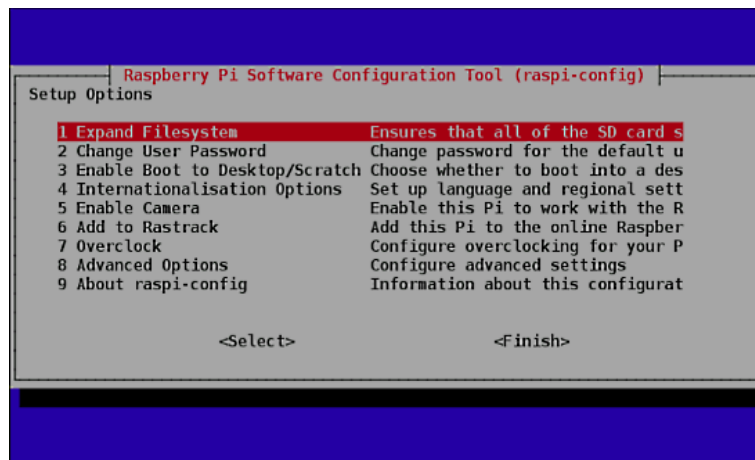
"Sudo raspi-config" ,



Komennolla avautuu näkymä, jossa Raspberry Pi:n perusasetuksia voidaan muokata. Tavallisen käytön kannalta tärkeä asetus löytyy kohdasta 1 "Expand Filesystem". Koska jakeluversiot voidaan asentaa erikokoisille tallenteille, niihin on määriteltä verratavaksi vain käynnistyksen kannalta tarvittava tila tallenteelta. "Expand Filesys-

tem” vapauttaa lopun tallenteella olevan tilan käyttöjärjestelmän käyttöön. Käytettäessä NOOBS-asennusversiota tämä vaihe on tehty valmiiksi. NOOBS-asennus käsitellään erillisellä ohjeella.

Valitaan näppäimistöllä valikko 1 ja painetaan ”Enter”. Asennusohjelma tekee tarvittavat asetukset itsestään eikä muita komentoja tarvita.



Omaan käyttöön asennettaessa tässä vaiheessa on hyvä myös määritellä käyttöjärjestelmälle uusi salasana valikon 2 ”Change User Password” kautta. Harjoituksen sujuvuuden kannalta on kuitenkin helpompaa säilyttää järjestelmän oletustunnukset.

Kun halutut asetukset on tehty, valitaan alakulman painike ”<Finish>” ja painetaan ”enter”.

Asennus ohjelma varmistaa halutaanko sulkea ohjelma vai käynnistää käytöjärjestelmä uudestaan. Koska on tehty käyttöjärjestelmän kannalta merkittäviä muutoksia, valitaan uudelleen käynnistys ”Reboot”.

Rasbian Jessie käyttöjärjestelmä käynnistyy uudelleen takaisin työpöydälle.

Liite 3. Rasbian NOOBS-asennus Raspberry Pi'lle, täysiversio

NOOBS-asennus suoritetaan formatoimalla muistikortti SD Card Formater-ohjelmalla FAT-tiedosto muotoon. Kortin tulee olla vähintään 8 Gt kokoinen. NOOBS-paketti sisältää valmiiksi ensikäynnistykseen tarvittavat osiot, joten pakettia ei tarvitse asentaa muistikortille kuten muissa asennustavoissa.

Puretaan NOOBS.zip tiedosto ja siirretään kansion sisältämät tiedostot muistikortille. Tässä vaiheessa on tärkeää siirtää kaikki tiedostot sellaisenaan suoraan muistikortin juureen eikä kansiossa, jossa tiedostot mahdollisesti purun jälkeen ovat.

Muistikortti voidaan nyt liittää Raspberry Pi:n muistikortti paikkaan.

Näppäimistö ja hiiri ovat asennusvaiheessa pakollisia. Nämä on tässä vaiheessa hyvä liittää usb-portteihin. Näin vältetään mahdollisilta ajuriongelmilta langattoman usb-adapterin kanssa.

Näiden toimenpiteiden jälkeen Pi:hin voidaan kytkeä virta.

NOOBS-asennus käynnistyy erilliseen asennusohjelmaan.



Ylä-valikosta löytyy käyttöpainikkeet. Sulussa olevat kirjaimet tarkoittavat näppäimistön pikanäppäin määritelmää.



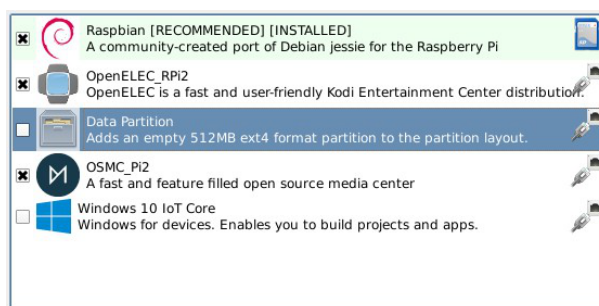
Install: Asentaa valitut ohjelmistot.

Wifi Networks: Voidaan yhdistää langattomaan verkkoon lankayhteyden sijaan.

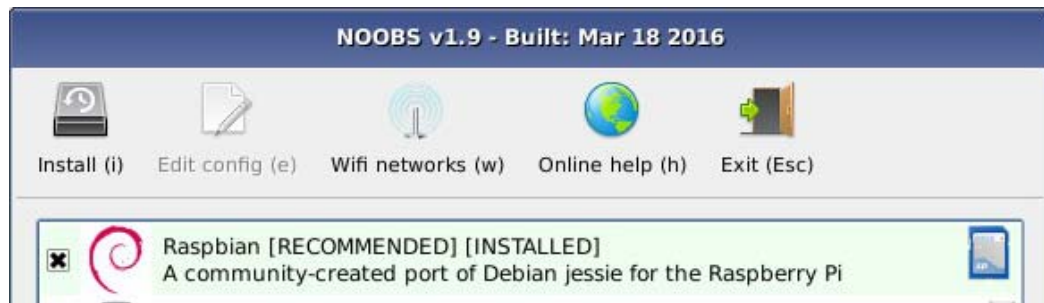
Online Help: Avataan yhteys verkossa olevaan ohjekansioon.

Exit: Sulkee asennuksen ja sammuttaa Raspberry-alustan.

Jos Raspberry PI on kytkettynä verkkoon, keskivalikkoon latautuu verkosta lista tarjolla olevista lisäohjelmista. Asennuslähteen tunnistaa oikealta löytyvästä ikonista. Muistikortin kuva tarkoittaa paikallista tallennetta ja verkkokytkeä ikoni ladattavia lisäosia. Tässä harjoituksessa käytetään muistikortilta löytyvää Rasbian käyttöjärjestelmää. Tämä vaihtoehto on aina ylimpänä listalla.



Valitaan ylin rivi "Raspbian [Recommended]" aktiiviseksi ja painetaan yläkulman install-painiketta.



NOOBS-asennus asentaa Rasbian käyttöjärjestelmän ja tekee tarvittavat asetukset automaattisesti. Tämän jälkeen Raspberry käynnistyy uudestaan suoraan käyttöjärjestelmän työpöydälle.

Liite 4. Rasbian Lite-version valmistelu ja asennus

Aina projektin kannalta ei ole tarpeellista asentaa käyttöjärjestelmästä täydellistä kokoonpanoa. Tällöin voidaan valita jakeluversion Lite-versio, josta on poistettu kaikki ylimääräinen sisältäen vain käyttöjärjestelmän ytimen. Asennus voidaan suorittaa NOOBS-asennuksella (vaihtoehto 1), jolloin asennetaan aina uusin Lite-jakeluversio tai manuaalisesti (vaihtoehto 2) haluttu Lite-jakeluversio.

Vaihtoehto 1 on harjoituksen kannalta hyvä tietää. Uusimman Rasbian Stretch:in ja uusimman Domoticz-version väliset yhteensopivuusongelmat estävät toistaiseksi käyttämästä vaihtoehtoa 1 laboratorioharjoituksessa. Vaihtoehdossa 2 asennetaan vanhempi Jessie Lite-jakeluversio yhteensopivuusongelmien välttämiseksi.

Käy vaihtoehto 1 läpi ja siirry sen jälkeen jatkamaan harjoitusta vaihtoehdon 2 mukaisesti.

Vaihtoehto 1: NOOBS, Rasbian Lite-asennus.

NOOBS-asennus suoritetaan formatoimalla muistikortti SD Card Formater-ohjelmalla FAT-tiedosto muotoon. Kortin tulee olla vähintään 8 Gt kokoinen. NOOBS-paketti sisältää valmiiksi ensikäynnistykseen tarvittavat osiot, joten pakettia ei tarvitse asentaa muistikortille kuten muissa asennustavoissa.

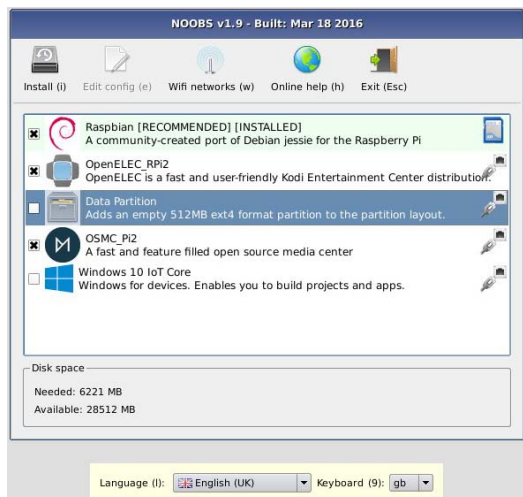
Puretaan NOOBS.zip tiedosto ja siirretään kansion sisältämät tiedostot muistikortille. Tässä vaiheessa on tärkeää siirtää kaikki tiedostot sellaisenaan suoraan muistikortin juureen eikä kansiossa, jossa tiedostot mahdollisesti purun jälkeen ovat.

Muistikortti voidaan nyt liittää Raspberry PI:n muistikortti paikkaan.

Näppäimistö ja hiiri ovat asennusvaiheessa pakollisia. Nämä on tässä vaiheessa hyvä liittää usb-portteihin. Näin vältetään mahdollisilta ajuriongelmilta langattoman usb-adapterin kanssa.

Näiden toimenpiteiden jälkeen Pi:hin voidaan kytkeä virta.

NOOBS-asennus käynnistyy erilliseen asennusohjelmaan. Näkysissä voi olla poikkeava lista lisäohjelmia. Kuva



Ylä-valikosta löytyy käyttöpainikkeet. Suluissa olevat kirjaimet tarkoittavat näppäimistön pikanäppäin määritelmää.



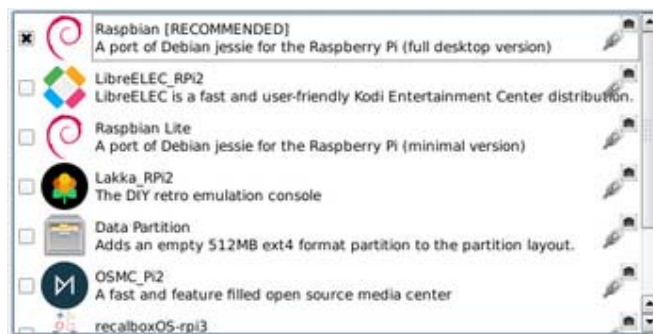
Install: Asentaa valitut ohjelmistot.

Wifi Networks: Voidaan yhdistää langattomaan verkkoon lankayhteyden sijaan.

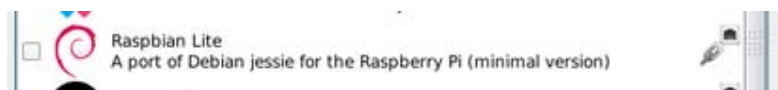
Online Help: Avataan yhteys verkossa olevaan ohjekansioon.

Exit: Sulkee asennuksen ja sammuttaa Raspberry-alustan.

Raspberry Pi pitää olla Lite-asennusta varten kytkettynä verkkoon, joko langalla tai yhdistämällä langattomaan verkkoon ”Wifi Networks” painikkeen kautta. Verkkoon yhdistämisen jälkeen keskivalikkoon latautuu lista tarjolla olevista lisäohjelmista. Asennuslähteen tunnistaa oikealta löytyvästä ikonista. Muistikortin kuva tarkoittaa paikallista tallennetta ja verkkokytkentä ikoni ladattavia lisäosia. KUVA X



Valitaan rivi ”Raspbian Lite” aktiiviseksi ja painetaan yläkulman Install-painiketta.



NOOBS-asennus asentaa Raspbian Lite käyttöjärjestelmän ja tekee tarvittavat asetukset automaattisesti. Tämän jälkeen Raspberry Pi käynnistyy uudelleen Lite-version terminaali näkymään.

```

[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
      Starting Hold until boot process finishes up...
      Starting Terminate Plymouth Boot Screen...
[ OK ] Started LSB: Start NTP daemon.

Raspbian GNU/Linux 8 raspberrypi tty1

raspberrypi login: pi
Password:
Linux raspberrypi 4.4.34-v7+ #930 SMP Wed Nov 23 15:20:41 GMT 2016 armv7l

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ raspi-config_

```



Vaihtoehto 2: Manuaalinen Lite-version asennus.

Jakelutiedostot löytyvät osoitteesta "<http://downloads.raspberrypi.org/>".

Kuvan mukaisesta kansioista "rasbian lite/".

| | | |
|---|-------------------|---|
|  rasbian/ | 08-Sep-2017 10:58 | - |
|  rasbian lite/ | 08-Sep-2017 10:59 | - |

Alakansiossa "images" on listattuna julkaisupäivämäärän mukaiset versiot rasbian lite käyttöjärjestelmästä. Jessie lite-jakelun viimeinen versio julkaistiin 5.7.2017. (Kuvio 9)

| | | |
|--|-------------------|---|
|  rasbian lite-2017-06-23/ | 23-Jun-2017 07:16 | - |
|  rasbian lite-2017-07-05/ | 05-Jul-2017 17:44 | - |

Kansio sisältää muutamia tiedostoja. (Kuvio 10) Isoin ".zip" pakattu tiedosto sisältää asennustiedostot. Toinen tärkeä mainitsemisen arvoinen on ".sha1" päätteellä varustettu tiedosto. ".sha1" tiedosto sisältää turvallisuustarkistusta varten laskentasumat, jolla käyttäjä voi erillisen ohjelman kautta tarkistaa asennustiedoston oikeellisuuden.

Index of /raspbian_lite/images/raspbian_lite-2017-07-05

| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|---|----------------------|-------------|--------------------|
|  Parent Directory | | | - |
|  2017-07-05-raspbian-jessie-lite.img.info | 05-Jul-2017 17:19 | 52K | |
|  2017-07-05-raspbian-jessie-lite.zip | 05-Jul-2017 12:47 | 297M | |
|  2017-07-05-raspbian-jessie-lite.zip.sha1 | 05-Jul-2017 17:42 | 78 | |
|  2017-07-05-raspbian-jessie-lite.zip.sha256 | 05-Jul-2017 17:42 | 102 | |
|  2017-07-05-raspbian-jessie-lite.zip.sig | 05-Jul-2017 17:02 | 473 | |
|  2017-07-05-raspbian-jessie-lite.zip.torrent | 05-Jul-2017 17:42 | 47K | |

Apache/2.2.22 (Debian) Server at downloads.raspberrypi.org Port 80

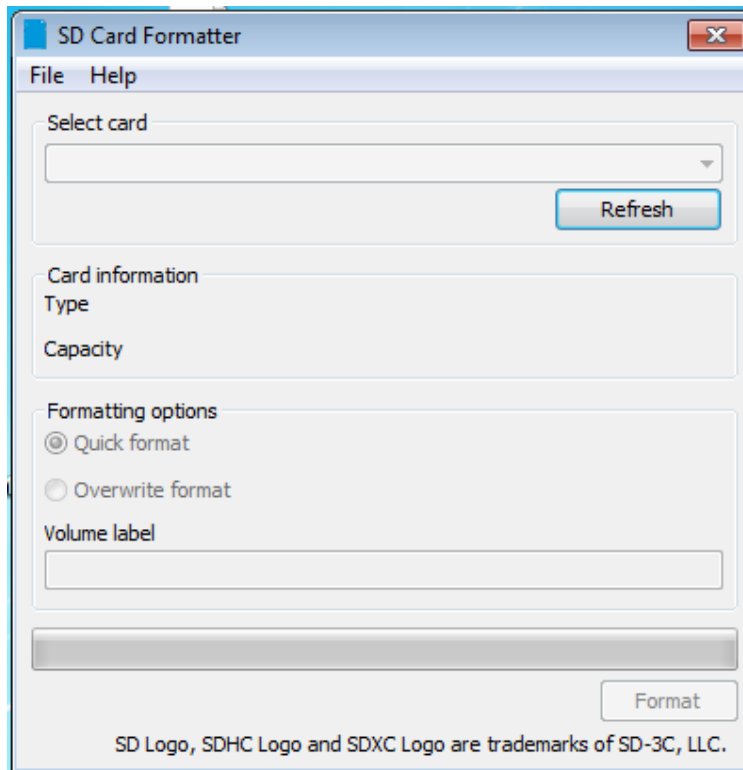
Painetaan linkistä "2017-07-05-raspbian-jessie-lite.zip" Linkkistä aukeaa selaimen latausvalikko, josta tiedosto tallennetaan haluttuun kansioon.

Kohta 2. Muistikortin valmistelu.

Ennen asennustiedostojen lataamista muistikortille se on hyvä alustaa. Näin varmistetaan, ettei kortilla ole mitään ylimääräistä eikä kortilla ole asennusta estäviä virheitä.

Alustukseen suositellaan erillistä ohjelmaa. Työssä käytetään ilmaista ja hyväksi todettua ohjelmaa SD Card Formater.

Ylemmästä alasetoalistasta valitaan alustuksen kohde. Varmuudeksi ennen ohjelman käynnistämistä koneesta on suositeltavaa irrottaa muut mahdolliset siirrettävät tallennusvälineet jolloin valikossa ei näy muita tallennusvälineitä.



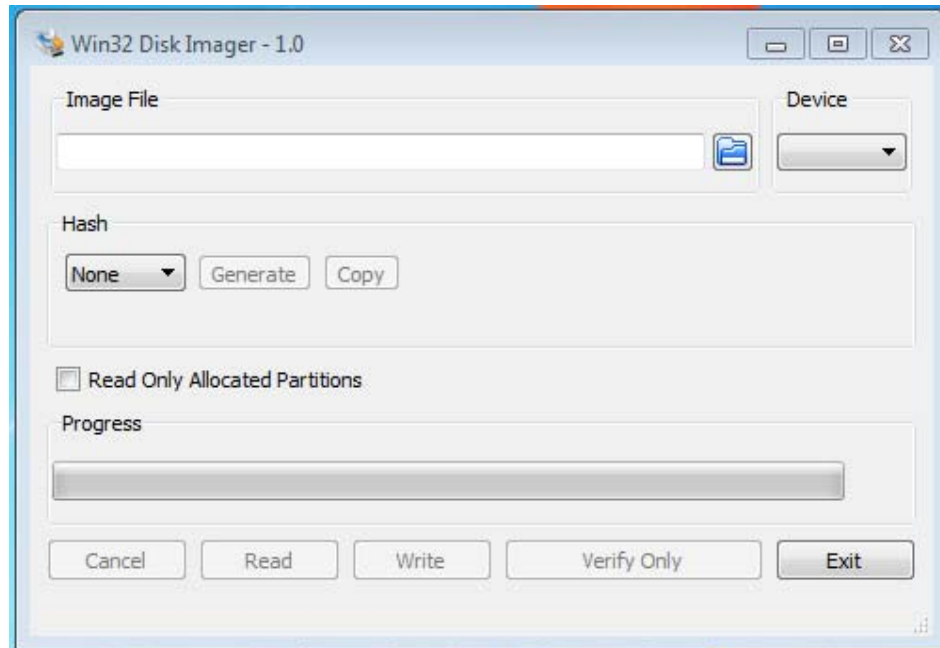
Kohta 3 Muistikortin asennus.

Onnistuneen formatoinnin jälkeen voidaan ladatut käyttöjärjestelmätiedostot asentaa muistikortille. Tämä asennusvaihe on sama kaikille Rasbianin versioille tehtäessä asennus windows-ympäristössä.

Ladataan, asennetaan ja käynnistetään Win32Diskimager.

Sinisestä kansio-ikonista valitaan tietokoneelta löytyvä levykuva, joka halutaan asentaa muistikortille. Vieressä olevasta pudotusvalikosta valitaan haluttu muistikortti.

Kun asetukset on valittu, painetaan alarivin write-komentoa. Ohjelma esittää varmistuksen kirjoituksesta. Tämän voi kuitata ja jatkaa.



Liitetään Raspberry PI:hin tarvittavat lisälaitteet.

Näppäimistö ja hiiri ovat asennusvaiheessa pakollisia. Nämä on tässä vaiheessa hyvä liittää langallisesti usb-portteihin. Näin vältetään mahdollisilta ajuriongelmilta langattoman usb-adapterin kanssa.

Näiden toimenpiteiden jälkeen Pi:hin voidaan kytkeä virta.

Käynnistettäessä Raspberry PI käynnistysohjelma käy läpi tarvittavia tiedostoja ja näytöllä rullaa nopeaan tahtiin komentoja. Lite-version käynnistys pysähtyy aina terminaalinäkymään kuvan X mukaan.

```
g W address to 1101:2800:2200:3000::110:3000:1110:2300
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
      Starting Hold until boot process finishes up...
      Starting Terminate Plymouth Boot Screen...
[ OK ] Started LSB: Start NTP daemon.

Raspbian GNU/Linux 8 raspberrypi tty1

raspberrypi login: pi
Password:
Linux raspberrypi 4.4.34-07+ #930 SMP Wed Nov 23 15:20:41 GMT 2016 armv7l

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ raspi-config_
```

Tämän jälkeen voidaan harjoitusta jatkaa asentamalla tarvittavat ohjelmistot käyttöjärjestelmään kohdan "LITE ohjelmistojen asennus." ohjeiden mukaan.

Liite 5. Raspbian Lite-version ohjelmistoasennukset

Lite-versio ei sisällä mitään ylimääräistä ja kaikki tarvittavat ohjelmistot joudutaan asentamaan manuaalisesti terminaalista (komentoriviltä). Lite-käyttöjärjestelmä käynnistyy suoraan terminaalinäkymään.

```

[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started Permit User Sessions.
        Starting Hold until boot process finishes up...
        Starting Terminate Plymouth Boot Screen...
[ OK ] Started LSB: Start NTP daemon.

Raspbian GNU/Linux 8 raspberrypi tty1

raspberrypi login: pi
Password:
Linux raspberrypi 4.4.34-v7+ #930 SMP Wed Nov 23 15:20:41 GMT 2016 armv7l

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ raspi-config_

```

Käyttöjärjestelmä kysyy ensimmäisenä käyttäjätunnuksia, jotka raspbianissa ovat:

”Login: pi” ja ”password: raspberry”

Tunnistautumisen jälkeen käyttöjärjestelmää voidaan muokata haluttuun muotoon.

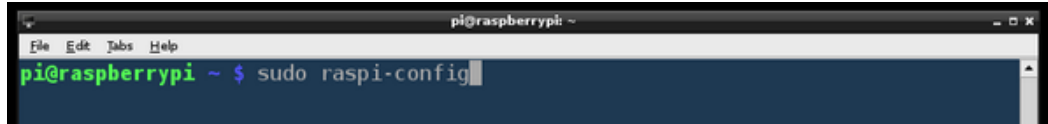
Seuraavaksi määritellään perusasetuksia ja asennetaan tarvittavat ohjelmistot.

(<https://www.raspberrypi.org/forums/viewtopic.php?t=133691>)

Ennen ohjelmistojen asennusta tarvitsee tehdä muutamat perusasetuksien muutokset.

Kirjoitetaan komentoriville

”Sudo raspi-config” ,



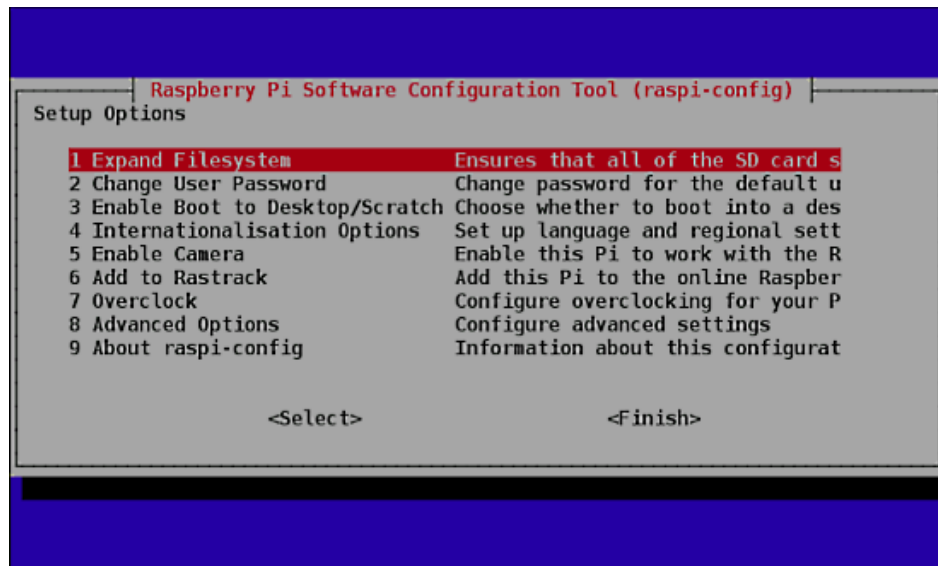
```
pi@raspberrypi ~ $ sudo raspi-config
```

HUOM: Jossain asennuksissa tämä ohjelma asentuu eriversiolla kuin esimerkeissä näytettävä. Tällöin tarvittavat valikot löytyvät kohdan ”Advanced Options” alta.

Valikkoja käytetään nuolinäppäimillä, välilyönillä, Enterillä ja TAB-näppäimellä. Valikoista voi hypätä ok painikkeelle TAB-näppäimellä.

Komennolla avautuu näkymä, jossa Raspberry Pi:n perusasetuksia voidaan muokata. Tavallisen käytön kannalta tärkeä asetus löytyy kohdasta 1 ”Expand Filesystem”. Koska jakeluversiot voidaan asentaa erikokoisille tallenteille, niihin on määritelty varattavaksi vain käynnistyksen kannalta tarvittava tila tallenteelta. ”Expand Filesystem” vapauttaa lopun tallenteella olevan tilan käyttöjärjestelmän käyttöön.

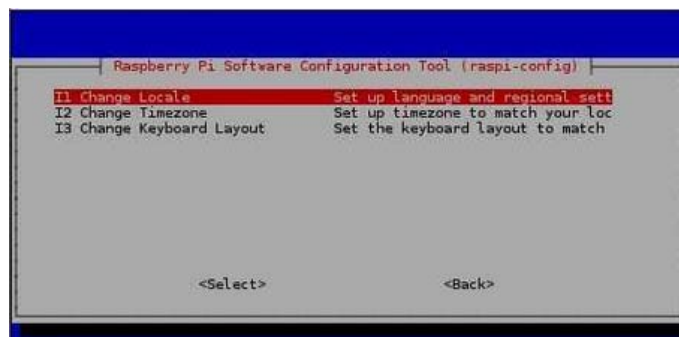
Valitaan näppäimistöllä valikko 1 ja painetaan ”Enter”. Hyväksytään ohjelman kysyessä varmistusta. Asennusohjelma tekee tarvittavat asetukset itsestään eikä muita komentoja tarvita.



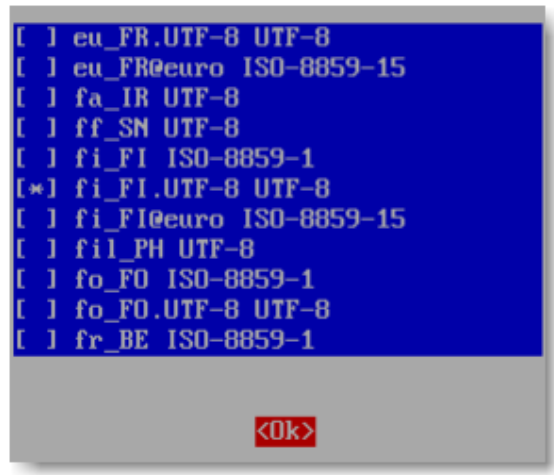
Omaan käyttöön asennettaessa tässä vaiheessa on hyvä myös määritellä käyttöjärjestelmälle uusi salasana valikon 2 "Change User Password" kautta. Harjoituksen sujuvuuden kannalta on kuitenkin helpompaa säilyttää järjestelmän oletustunnukset.

Vakio asetuksena käyttöjärjestelmä on määritelty englanninkieliseksi ja käyttämään sen mukaista näppäimistökarttaa. Monet erikoismerkit ovat tällöin eri näppäinten taakan kuin suomalaisessa näppäimistössä. Näitä asetuksia päästää vaihtamaan kohdasta "4 Internationalization Options"

Valitaan kohta 4 aktiiviseksi käyttäen näppäimistön nuolinäppäimiä ja painetaan "enter" Tällöin aukeaa seuraavan lainen valikko.

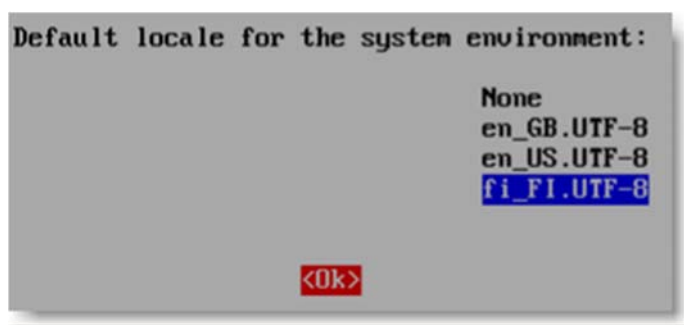


Valitaan ensimmäinen asetus, jolla voidaan säätää Raspberry käyttämään aluekohtaisia asetuksia. Paina "Enter" ja seuraava valikko aukeaa.



Hakasulkeiden sisällä oleva merkki kertoo kyseisen aluevalinnan olevan aktiivinen. Etshitään listalta suomenalueasetukset "fi_FI.utf8". Valinta tehdään tässä kohtaa painamalla halutun alueasetuksen kohdalla näppäimistöä "space". Tämän jälkeen voidaan painaa valikon alalaidan "<Ok>".

Seuraavassa valikossa valitaan käyttöjärjestelmä käyttämään oletusarvoisesti joitain asennettua alue-asetusta. Valitaan nuolinäppäimillä suomen alueasetukset aktiiviseksi ja painetaan taas "<Ok>".



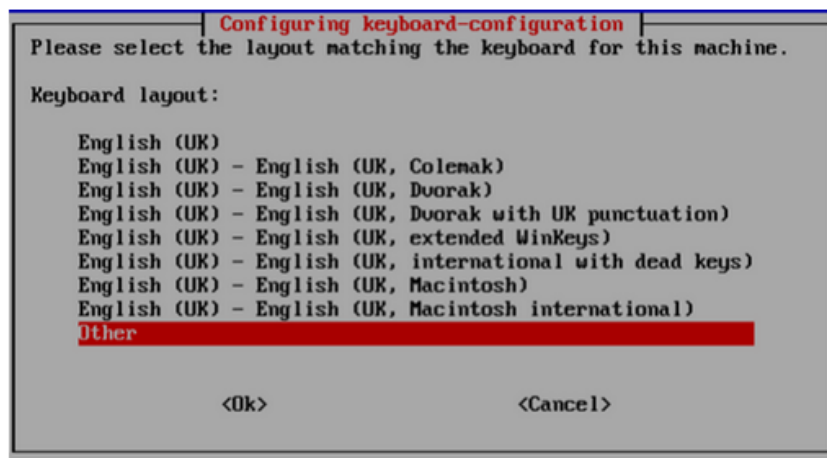
Tässä vaiheessa ohjelma palaa alueasetusten ylävalikkoon, josta valitaan seuraava kohta. Valikosta 2 "Change Timezone" valitaan suomen "helsinki" tunnuksilla oleva aikavyöhyke. Tämä löytyy kohdan "Europe" takaa.

Tee tarvittavat valinnat.

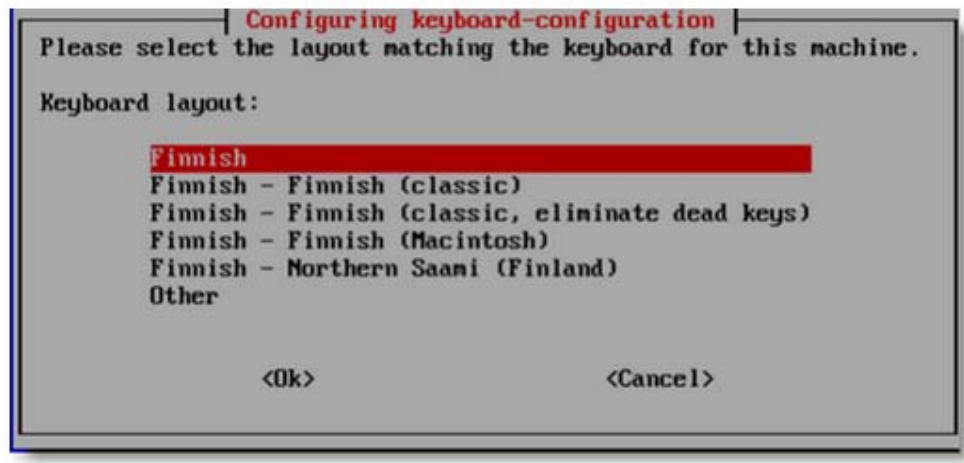
Asennus palaa taas aluevalikon ylävalikkoon, josta seuraavaksi asetetaan kohdasta 3 "Change Keyboard Layout" näppäimistökartta. Joillekin erikoisimmille näppäimistöille on suoraan oma kartta. Tavallisesti voidaan kuitenkin käyttää yleiskarttaa jolle vain määritellään suomenkieliset asetukset.

Valitaan listalta "Generic 105-key (Intl) PC" ja määritellään tälle suomenkielinen näppäimistökartta avautuvasta valikosta. Asennustiedostoista tähän latautuu pääliemäiseksi englanninkielinen lista.

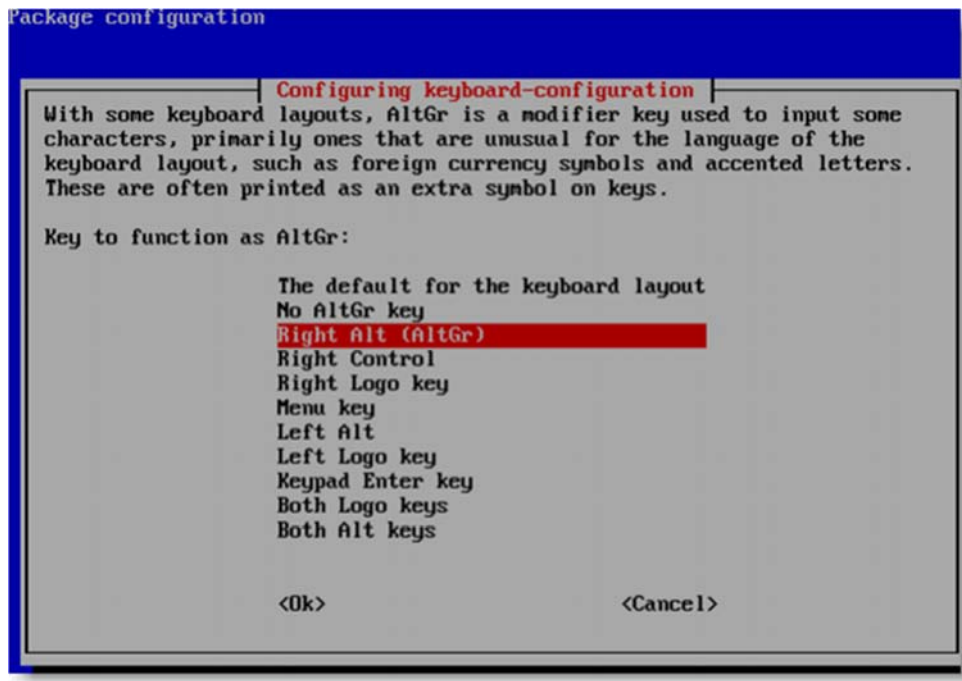
Valitaan listalta "Other" jolloin muun kieliset kartat saadaan näkyviin.



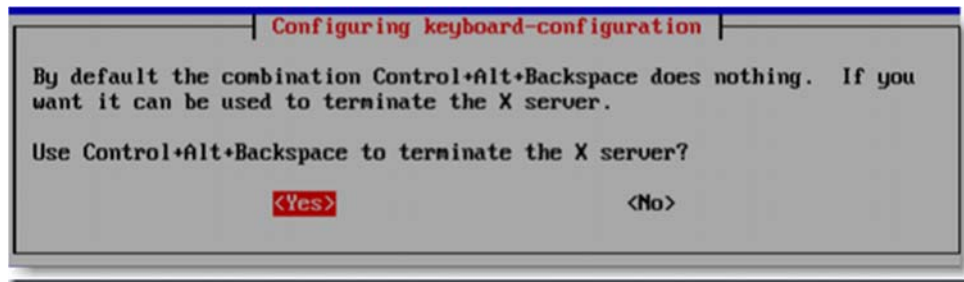
Valitaan listalta "Finnish" jolla päästään seuraavaan valikkoon. Valitaan käytettäväksi "Finnish - Finnish (Classic)" ja painetaan "<Ok>".



Tässä valikossa määritellään aktiiviseksi erikoismerkeille käytettävä "Right Alt (AltGr)"
Tämän näppäimen takaa saa esimerkiksi näppäimistöltä "@" merkin käyttöön.



Seuraavaksi asetetaan "CTRL+ALT+Del" näppäinyhdistelmä käynnistämään Raspberry Pi uudestaan suoraan. Esimerkiksi jos graafinenliittymä lopettaa vastaamisen, voidaan käyttöjärjestelmä käynnistää uudestaan tämän avulla. Valitse "<Yes>"



Raspberry 3 versiossa on sisäänrakennettu langatonvastaanotin. Tämä voidaan säätää tarkemmin kohdasta 4 "Change Wi-Fi Country" vastaavanlaisilla asetuksilla. Työssä käytetään vanhempaa Raspberry 2 versiota, joten tämän kohdan voi jättää välistä.

Kun halutut asetukset on tehty, valitaan alakulman painike "<Finish>" ja painetaan "enter".

Asennusohjelma varmistaa halutaanko sulkea ohjelma vai käynnistää käyttöjärjestelmä uudestaan. Koska on tehty käyttöjärjestelmän kannalta merkittäviä muutoksia, valitaan uudelleen käynnistys "Reboot".

Joissain tilanteissa asennus voi jättää uudelleenkäynnistyksen tekemättä ja palaa suoraan komentoriville. Tässä tilanteessa käynnistys voidaan suorittaa kirjoittamalla "sudo reboot" komentoriville ja painamalla "Enter"

Ennen työpöydän asennusta käyttöjärjestelmä on hyvä päivittää seuraavilla komennoilla. Aja seuraavat komennot järjestyksessä ja siirry sitten eteenpäin. Tarvittaessa voi hyväksyä lisätilan käytön ja varmistukset komentorivin ohjeiden mukaisesti kirjoittamalla "y"/"k" tai "n"/"e". Y=yes, n=no, k=kyllä, e-ei. Nämä mukautuvat automaattisesti käytettävän kielen mukaan.

Komento tarkistaa asennettujen ohjelmistopakettien versiovarmenteet ja lataa tarvittaessa uudemmat versiot järjestelmään.

```
sudo apt-get update
```

Komento käy läpi ladatut päivitystiedot ja asentaa tarvittavat päivityksen.

```
sudo apt-get upgrade
```

Komento käy yllämainittua käskyä tarkemmin läpi päivitykset ja asentaa tarvittaessa.

```
sudo apt-get dist-upgrade
```

Komento puhdistaa latauskansion päivityspaketit. Nämä veisivät turhaa tilaa käyttöjärjestelmästä.

```
sudo apt-get clean
```

Näiden asetusten jälkeen voidaan suorittaa tarvittavat ohjelmistoasennukset komentoriviltä.

Työpöytänäköymä (GUI) tarvitsee toimiakseen vähintään seuraavat neljä ohjelmistopakettia.

- Display Server.
- Desktop Environment
- Window Manager
- Login Manager

Kukin edellä mainituista paketeista pitää asentaa käsin komentorivin kautta. Uudelleen käynnistyksen jälkeen kirjaututaan jälleen Raspinin tunnuksilla ja saadaan komentorivi käyttöön.

Aloitetaan asentamalla Xorg Display Sever kirjoittamalla komentoriville seuraava komento.

```
sudo apt-get install --no-install-recommends xserver-xorg
```

Komento käyttää käyttöjärjestelmän sisäistä komentokirjastoa "apt" ja sitä tarkennetaan "-get" komennolla. Nämä ovat sisäisiä funktioita ja niihin voi perehtyä tarkemmin omatoimisesti. "install" käskee asentamaan. Lisämääritys "--no-install-recommends" rajaa asennuspaketin käyttämiä komentoja suorittamaan vain välttämättömimmät paketin komennot. Paketit sisältävät monesti myös suositeltuja lisäominaisuuksia. Tässä tapauksessa emme kuitenkaan tarvitse näitä. "Xserver-xorg" määrittelee käytettävän paketin.

Hyväksy komento painamalla "enter" ja anna ohjelman ajaa komennot loppuun.

Asennuksen jälkeen komentorivi palaa oletustilaan.

Asennetaan seuraavaksi käytettävä työpöytäympäristö: "Desktop Environment"

Harjoituksessa käytetään kevennettyä versiota Raspbianin omasta PIXEL GUI:sta.

Muitakin vaihtoehtoja on omien preferenssien suhteen. Aiheeseen voi perehtyä tarkemmin googlen avulla haulla "linux desktop environments" Asennusta voi halutessaan muokata toisella työpöytäliittymällä jos haluaa lisähaastetta tai on aikaisemmin käyttänyt linux ympäristöä. Avuksi voi käyttää seuraavaa linkkiä.

" <https://www.raspberrypi.org/forums/viewtopic.php?t=133691> "

Kirjoitetaan komentoriville ja painetaan "enter".

```
sudo apt-get install --no-install-recommends raspberrypi-ui-mods
lxterminal
```

Komennon lisäosapaketti "lxterminal" asentaa työpöydälle käytettäväksi komentoriviohjelman. Tätä ei ole kevennetyssä työpöydässä valmiina. Jos tätä ei asenna, komentoriviä ei voi avata työpöydältä vaan komentoriville pitää aina uudelleen käynnistää.

Annetaan asennuksen juosta ja kuitataan mahdolliset tarkennukset.

Käytettävä työpöytä "PIXEL" sisältää alla ohjeistetun ohjelmiston. Joten voit siirtyä suoraan tämän yli. Jos kuitenkin asennat jotain muuta työpöytäversiota, tulee tämäkin asentaa.

Jotta työpöytä saadaan toimimaan, tarvitaan vielä sisäänkirjautumiseen oma pakettinsa. "Login Manager" Tämä ohittaa Raspberry Pi'n käynnistyksen komentoriville, jolloin päästään suoraan työpöydälle.

Kirjoitetaan komentoriville ja painetaan "enter"

```
sudo apt-get install lightdm
```

LightDM on usein käytetty vaihtoehtoinen sisäänkirjautumisen hallintapaketti jakeluversioiden omille vastaaville toiminnoille.

Anna asennuksen ajaa ja hyväksy mahdolliset tarkennukset.

Näiden asennuksien jälkeen Raspberry Pi voidaan käynnistää uudelleen "reboot" komennolla. Käytä tarvittaessa sudo -etuliitettä.

Tämän jälkeen voidaan siirtyä asentamaan työpöydän kautta tarvittavia ohjelmistoja osiosta "Rabian-ohjelmistojen asennus: Domoticz, Arduino IDE, Web browser."

Liite 6. Rasbian-ohjelmistojen asennus

Lite-käyttöjärjestelmä ei sisällä mitään ohjelmistoja, jos ei niitä ole itse asentanut.

Tekemistä helpottamaan seuraavaksi asennetaan muutamat lisäohjelmat. Jos asensit täyden käyttöjärjestelmän voit siirtyä suoraan Domoticz-käyttöliittymän asennukseen kohdasta ”Domoticz valmistelut” eteenpäin.

Avaa työpöydän yläkulmasta komentorivi. Ikoni jossa mustalla pohjalla on merkit ”>”.



Epiphany on Raspberry Pi kehittäjien tekemä versio GNOME jakelun selaimesta ja toimii käyttöjärjestelmässä ilman ongelmia. Asennetaan komentoriviltä komennolla

```
sudo apt-get install epiphany-browser
```

Työpöydän valikkoihin ilmaantuu ikoni selaimelle.

Asennuksen jälkeen voidaan asentaa Arduino IDE, jolla Arduino-kehitysalustoja voidaan ohjelmoida. Jessie-jakeluversion vanhentunut toimiva paketti Arduino IDE:sta on versio 1.0.5 eikä tue koodissa käytettyä kirjastoja. Ohjelman kautta voidaan kuitenkin käyttää Serial Monitor-toimintoa. Tämän kautta voidaan tarkastella Raspberry Pi:lle tulevaa tietoa Arduino-ohjainkortilta.

Asennetaan komentoriviltä komennolla:

```
sudo apt-get install arduino
```

Domoticz valmistelut:

Jotta harjoituksen laitteisto saadaan keskustelemaan oikein Domoticzin vakio asennusta pitää hieman soveltaa.

KÄYTETTÄESSÄ RASPBERRY PI 3, Muutoin voi siirtyä pisteviivan jälkeisiin ohjeisiin.

Uusimmassa Raspberry Pi:ssa sisäänrakennettu bluetooth-ohjain häiritsee Domoticzin ja Z-Wave/Arduino-sillan välistä kommunikointia ja se joudutaan manuaalisesti säätämään pois käytöstä automaattisesti aina käynnistyksen yhteydessä.

Kirjoitetaan komentoriville komento:

```
sudo nano /boot/config.txt
```

Komennolla avataan teksti editori "Nano" ja suoraan tiedosto "config.txt" polusta "/boot/"

Lisätään tiedoston viimeiseksi teksti:

```
dtoverlay=pi3-disable-bt
```

Tällä määritellään käynnistysasetuksista bluetooth-ohjain pois käytöstä.

Poistutaan ohjelmasta komennolla "ctrl-X" ja hyväksytään muutokset.

Kirjoitetaan komentoriville.

```
sudo systemctl disable hciuart
```

Tällä vielä varmennetaan ettei bluetooth-ohjain osaa edes yhdistää sarjaportteihin.

Jatka katkoviivan jälkeisistä ohjeista.

Raspberry Pi'n sisäiset sarjaportti määrittelyt häiritsevät Domoticzin ja Z-Wave/Arduino-sillan välistä kommunikointia ja ne joudutaan manuaalisesti säätämään.

Kirjoitetaan komentoriville:

```
sudo nano /boot/cmdline.txt
```

Komennolla avataan teksti editori "Nano" ja suoraan tiedosto "cmdline.txt" polusta "/boot/"

Poistetaan tekstiriviltä kohta " console=serial0,115200 ", Muista jättää edeltävän ja seuraavan komennon väliin valilyönti.

Poistutaan ohjelmasta komennolla "ctrl-X" ja hyväksytään muutokset.

Kirjoitetaan komentoriville: Tällä määritellään Z-Wave linkin Baud nopeus.

```
sudo stty -F /dev/ttyAMA0 9600
```

Jotta asetukset tulevat voimaan pitää Raspberry Pi käynnistää uudestaan.

```
"sudo reboot"
```

Käynnistyksen jälkeen avaa komentorivi.

Yllä mainittujen komentojen jälkeen voidaan tehdä domoticzin puhdas asennus.

Domoticz-asennus:

Domoticz on tehty mahdollisimman käyttäjäystävälliseksi ja se voidaan asentaa yhdellä komennolla komentoriviltä:

```
sudo curl -L install.domoticz.com | bash
```

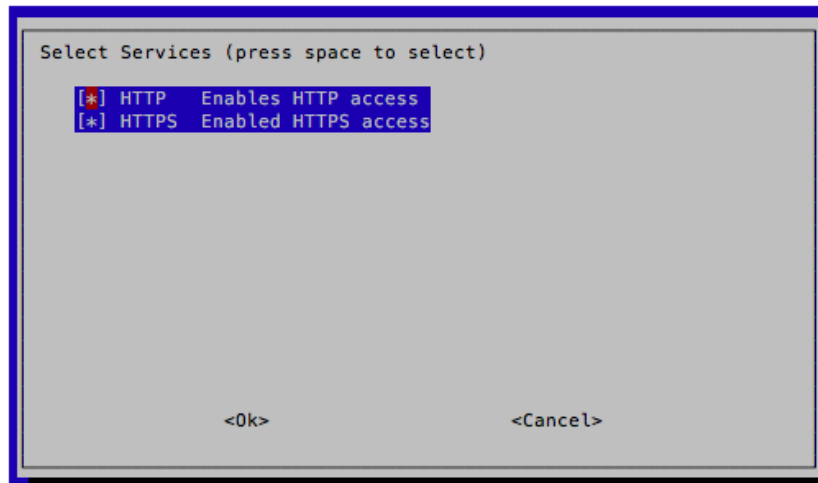
Asennusohjelma tekee tarvittavia päivitys ja tarkistustoimintoja, minkä jälkeen avautuu seuravanlainen ikkuna.



Harjoituksen kannalta ei ole merkittävää määritellä kiinteää ip-osiotetta. Tämä on kuitenkin hyvä pitää mielessä jos haluaa rakentaa vastaavanlaista kokoonpanoa muualle.

Hyväksytään kehoitukset painamalla "Enter"

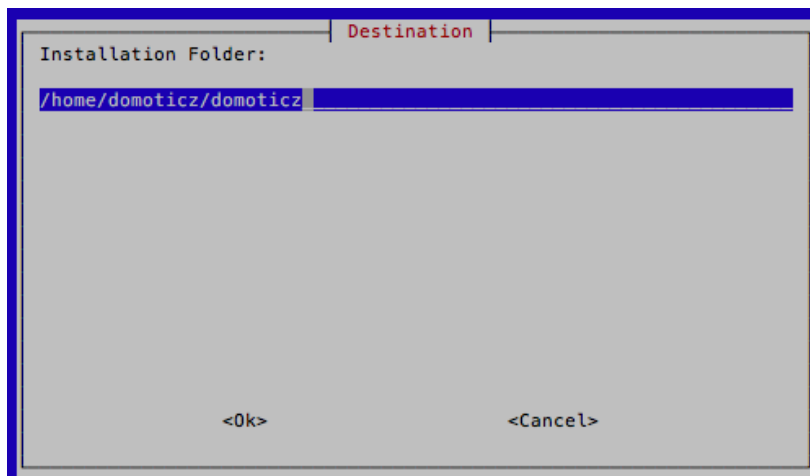
Seuraavassa valikossa valitaan käytettävät HTTP-määritykset.



Julkisen ja/tai salatun yhteyden salliminen merkillä "*" , välilyönnillä valinnat. TAB siirtää osoittimen alavalikkoon, josta valitaan "ok".

Asennus kysyy mitä portteja halutaan käyttää yllätehdyle yhteysvalinnoille. Nämä voidaan pitää vakiona http:8080 ja https:443. Nämä on kuitenkin hyvä selvittää käytännön asennuskohteissa.

Asennus kysyy vielä asennuskansion. tämä voidaan pitää myös harjoituksessa vakiona.



Asennus antaa tämän jälkeen yhteenvedon asetuksista. IP-osoite voi vaihdella ja se on hyvä ottaa tässä vaiheessa talteen. Kirjoita osoite paperille myöhempää käyttöä varten.

```
Installation Complete!  
  
Point your browser to either:  
  
HTTP:      192.168.1.90:8080  
HTTPS:     192.168.1.90:443  
  
Wiki:  https://www.domoticz.com/wiki  
Forum: https://www.domoticz.com/forum  
  
The install log is in /etc/domoticz.  
  
<Ok>
```

Domoticz-käyttöliittymän toimivuus voidaan kokeilla kirjoittamalla Raspberry Pi'n selaimen osoitekenttään annettu http ip-osoite.

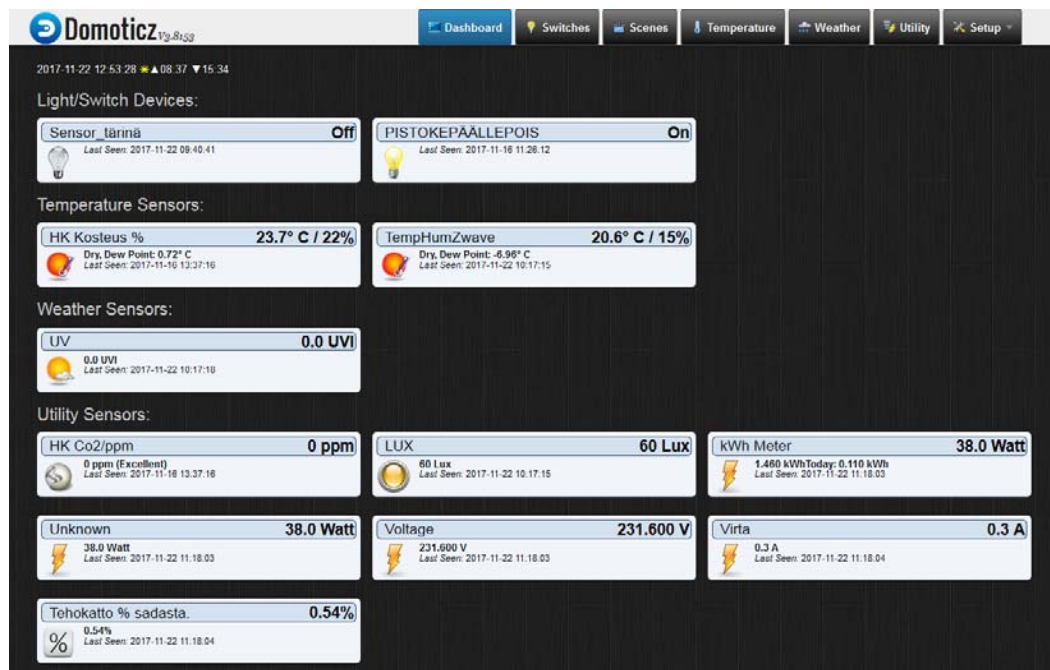
Voit siirtyä Domoticz-käyttöliittymän ohjeisiin.

Liite 7. Domoticz-käyttöliittymä: Z-Waven asennus

Jos käytät Domoticzia Raspberry Pi'n kosketusnäytöltä, anna ohjelmalle aikaa ja vältä turhia napinpainalluksia. Raspberry Pi'n grafiikkaohjain ei ole maailman nopein.

Z-Wave laitteiden käyttö ei tarvitse erillisiä toimenpiteitä Domoticz-käyttöliittymän asentamisen jälkeen. Tarvittavat taustaprosessit ovat kiinteänä asennuspaketissa. Z-Wave-järjestelmän oma käyttöliittymää ei suositella asennettavaksi yhtä aikaa Domoticz-käyttöliittymän kanssa. Z-wave-järjestelmän omaan liittymään voi tutustua osoitteessa. ” <http://razberry.z-wave.me/index.php?id=5> ”

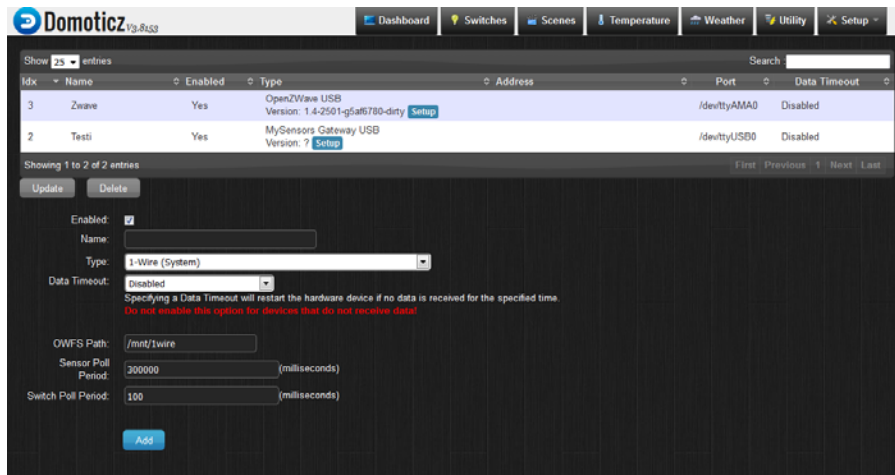
Seuraavassa kuvassa näkyy esiasennetus Domoticz-liittymän etusivu, johon on asennettuna eri järjestelmien antureita.



Järjestelmät otetaan käyttöön Setup välilehden Hardware valikosta.



Tästä aukeaa kuvan mukainen asetusvalikko. Laiterivillä näkyy esimerkki toimivasta laiteyhteydestä järjestelmiin. Ylin OpenZWave USB on Razberry2 kortin kautta toimiva Z-wave ja MySensors Gateway USB on Arduino-sillan kautta toimiva Modbus-väylä. Ensi käynnistyksellä näitä ei kuitenkaan ole vielä asennettu.



Aloitetaan liittämällä Z-Wave-järjestelmä seuraavan kuvan mukaisesti valitsemalla ”OpenZwave USB” aktiiviseksi .

Enabled:

Name:

Type: 1-Wire (System)

Data Timeout: Neft Easy HTTP server over LAN interface

OWFS Path: Nest Thermostat/Protect

Sensor Poll Period: Netatmo

Switch Poll Period: Onkyo AV Receiver (LAN)

OpenTherm Gateway USB

OpenTherm Gateway with LAN interface

OpenZWave USB

P1 Smart Meter USB

P1 Smart Meter with LAN interface

PanasonicTV

Philips Hue Bridge

PiFace - Raspberry Pi IO expansion board

Plugwise Anna Thermostat via LAN interface

PVOutput (Input)

Rainforest RAVen USB

Raspberry's GPIO port

Razberry Z-Wave via LAN interface (Deprecated)

Rego 6XX USB/serial interface

Relay-Net 8 channel LAN Relay and binary Input module

”Enabled” määrittlee liitettävän järjestelmän aktiiviseksi.

”Name” tähän voit antaa järjestelmälle haluamasi nimen. Esimerkiksi ”Z-wave”

”Type” kohdan alavetovalikosta löytyy kaikki Domoticzin tukemat järjestelmät ja listalta valitaan ”OpenZWave USB”.

Seuraavaksi ilmestyvät valikot riippuvat valitusta järjestelmästä. OpenZWave valinta antaa alla olevat asetukset.

Enabled:

Name:

Type: OpenTherm Gateway USB

Data Timeout: Disabled

Specifying a Data Timeout will restart the hardware device if no data is received for the specified time.
Do not enable this option for devices that do not receive data!

Serial Port: /dev/serial0

Add

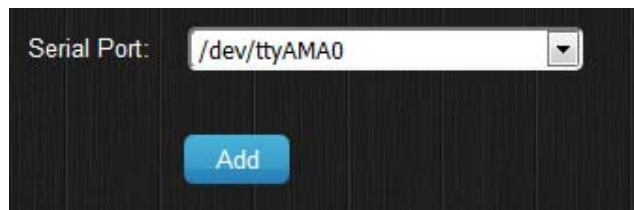
Tässä kohtaa Data Timeout kohdan voi jättää toimeettomaksi. Tällä säädetään Domoticz uudelleen käynnistämään asennettu liityntäjärjestelmä uudelleen, jos siltä ei saada informaatiota annetussa ajassa.

”Serial Port” valikosta valitaan käytettä laiteportti. Raspberry Pi’lla porttien nimet muuttuvat sen mukaan mihin laitteet on liitetty. Razberry2-ohjain on liitettyä GPIO-pineihin jolloin sen portti määrittytty ttyAMA*.

*0 = laite 1, *1= laite 2 jne.

USB-porttiin liitetyt laitteet saavat ttyUSB*-määrittelyksen.

Valitaan kuvan mukaisesti /dev/ttyAMA0 ja painetaan ”Add”



Määritelty järjestelmä ilmestyy nyt käyttöliittymän listalle. Aluksi kohta Version voi näyttää kysymysmerkkiä. Tähän kuitenkin ilmestyy laiteyhteyden varmistuttua käytössä oleva ohjain versio. Samoin version vierestä löytyvä Setup-nappi voi näyttää punaista ennen kuin yhteys varmennettu.

Seuraavaksi painetaan mainittua Setup-nappia. Tämän kautta päästää määrittelemään kyseisen järjestelmän käyttämiä laitteita.

Huom! Molempien laitteiden ohjekirjat löytyvät harjoitukseen liittyvästä tehtäväkansiosista. Niistä löytyvät mm. tehdaspalautuksen ohjeet, jos laitteet joutuvat ongelmalaan eivätkä automaattisesti palauta itseään.

Kuvassa valmiiksi asennetut Z-Wave anturit.



Z-Wave on suljettuverkko, johon laitteet pitää varmentaa. Tämä tapahtuu käyttämällä yläkulmasta löytyvän valikon Include node -valintaa ja kytkemällä Z-Wave anturista Inclusion Mode -aktiiviseksi. Laitekohtaiset ohjekirjat kertovat kunkin käytettävän laitteen tavat toteuttaa varmennus.

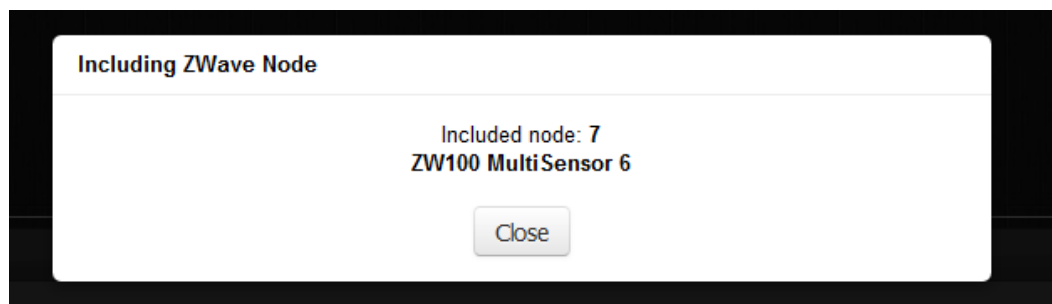
Tehdään liittäminen mahdolliseksi aktivoimalla liityntä Include Node -napilla.



Painetaan Aeotecin Multisensorin takana olevaa aktivointi nappia kuvan kohdasta.



Laitteen etupuolella olevaan lediin pitäisi syttyä vihreä vilkkuva LED-valo. Laite on tällöin liityntätilassa. Jos liittäminen onnistuu, Domoticzin käyttöliittymään tulee seuraava ilmoitus.



Node-numero annetaan isäntä-ohjaimen toimesta. Alle tulee näkymään laitteen perustiedot.

Philion pistokemittari liitetään samalla tekniikalla.

Domoticzin hallinnasta "Include node"

Vastaavasti painetaan pistokkeen etupuolelta löytyvää nappia kolmesti 2 sekunnin sisällä sen ollessa kytkettynä seinään.

Pistokemittari menee myös automaattisesti kytkentä tilaan, jos sille ei ole määriteltyä isäntäverkkoa, ja se kytketään pitorasiaan. Painonappi vilkuttaa n. 0.5 sekunnin syklissä punaista lediä ollessaan liityntätilassa.

Onnistunut liittäminen antaa ylemmän kuvan mukaisen ilmoituksen.

Jos laitteet halutaan irroittaa järjestelmästä, tehdään vastaavat toimenpiteet mutta valitaan Domoticzin-ohjaimesta "Exclude Node"-valinta.

Asennetut laitteet löytyvät nyt Domoticzin käytettävänä käyttöliittymän valikoista.

Valitaan Domoticzin setup-valikosta "Devices" osio.



Z-Wave laitteilla voi olla useita toimintoja ja mittauksia. Nämä tulevat näkyviin Devices-osion alle.

 A screenshot of the Domoticz 'Used' devices page. It shows a table with 19 entries. The table has columns for Idx, Hardware, ID, Unit, Name, Type, SubType, and a numerical value. The devices listed include a kWh Meter, an Unknown device, an Alarm Type, an Alarm Level, a Sensor, a Burglar, an Alarm Type (Burglar 7), a Lux sensor, a UV sensor, and a Temperature/Humidity sensor.

| Idx | Hardware | ID | Unit | Name | Type | SubType | |
|-----|----------|----------|------|------------------------------|-----------------|--------------|-----------------|
| 7 | Zwave3 | 00000501 | 1 | kWh Meter | General | kWh | 1.550 kWh |
| 8 | Zwave3 | 0000501 | 2 | Unknown | Usage | Electric | 34.6 Watt |
| 12 | Zwave3 | 00000733 | 1 | Alarm Type | Light/Switch | Switch | Off |
| 13 | Zwave3 | 00000732 | 1 | Alarm Level | Light/Switch | Switch | Off |
| 14 | Zwave3 | 00000700 | 1 | Sensor | Light/Switch | Switch | On |
| 15 | Zwave3 | 00000735 | 1 | Burglar | Light/Switch | Switch | On |
| 16 | Zwave3 | 00000735 | 7 | Alarm Type: Burglar 7 (0x07) | General | Alarm | Event: 0x08 (8) |
| 17 | Zwave3 | 0000703 | 255 | Unknown | Lux | Lux | 38 Lux |
| 18 | Zwave3 | 071B | 0 | UV | UV | UVN128,UV138 | 0.0 UVI |
| 19 | Zwave3 | 0701 | 0 | TempHum | Temp + Humidity | WTGR800 | 21.1 C, 18 % |

Vasemmassalaidassa eri ikoneilla määritellään pisteen merkki ja toiminta. Esimerkiksi lampun kuva tarkoittaa valoa tai kytkintä ja ikonin väri kertoo suoraan onko kyseinen piste aktiivinen vai ei.



Välilehdeltä selviää kaikki mahdollinen laitteista ja mittauksista saatava tieto.

| Idx | Hardware | ID | Unit | Name | Type | SubType | |
|-----|----------|----------|------|------------------------------|-----------------|--------------|-----------------|
| 7 | Zwave3 | 00000501 | 1 | kWh Meter | General | kWh | 1.550 kWh |
| 8 | Zwave3 | 0000501 | 2 | Unknown | Usage | Electric | 34.6 Watt |
| 12 | Zwave3 | 00000733 | 1 | Alarm Type | Light/Switch | Switch | Off |
| 13 | Zwave3 | 00000732 | 1 | Alarm Level | Light/Switch | Switch | Off |
| 14 | Zwave3 | 00000700 | 1 | Sensor | Light/Switch | Switch | On |
| 15 | Zwave3 | 00000735 | 1 | Burglar | Light/Switch | Switch | On |
| 16 | Zwave3 | 00000735 | 7 | Alarm Type: Burglar 7 (0x07) | General | Alarm | Event: 0x08 (8) |
| 17 | Zwave3 | 00000703 | 255 | Unknown | Lux | Lux | 38 Lux |
| 18 | Zwave3 | 071B | 0 | UV | UV | UVN128,UV138 | 0.0 UVI |
| 19 | Zwave3 | 0701 | 0 | TempHum | Temp + Humidity | WTGR800 | 21.1 C, 18 % |

Tästä on ensi katsomalla vaikea erottaa mikä on minkäkin laitteen mittapiste. Tämä selviää kiinnittämällä huomiota ID-sarakkeen tietoihin. Ensimmäinen numero kertoo laitteelle annetun Node ID:n. Liitettäessä Multisensoria se sai ID:n 7 joten kaikki tähän laitteeseen liittyvät pisteet alkavat numerolla 7. Vastaavasti pistokemittari on saanut kuvan esimerkissä ID:n 5 ja sen pisteet alkavat numerolla 5.

Alla olevasta kuvasta selviää laiteyhteyden tietoja. Jotkin laitejärjestelmät voivat ilmoittaa langattoman viestinnän vahvuuden antenni-ikonin alla. Z-Wave laitteille tyypillinen käyttövirrantasoo näkyy vihreän paristo-ikonin alta. Esimerkin Multisensor on

varustettu kahdella paristolla, joista toinen on käytössä ja toinen varalla. Paristojen tila näkyy laitevalikossa ja auttaa ennakoimaan paristojen vaihtoa.

| | | Search : | | | |
|---|-----|----------|---|-----------|---------------------|
| | | | | Last Seen | |
| - | - | ← | 🔋 | 📄 | 2017-11-22 14:43:05 |
| - | - | ← | 🔋 | 📄 | 2017-11-22 14:43:05 |
| - | 100 | ← | 🔋 | 📄 | 2017-11-22 14:34:16 |
| - | 100 | ← | 🔋 | 📄 | 2017-11-22 14:34:16 |
| - | 100 | → | 🔋 | 📄 | 2017-11-22 14:34:16 |
| - | 100 | → | 🔋 | 📄 | 2017-11-22 14:34:16 |

Kuvan seuraavat kolme ikonia käsittelee laitteen hallintaa. Vihreä nuoli kertoo, että laite on liitettävissä Domoticzin käyttöön ja sininen väri kertoo laitteen jo olevan aktiivinen, jolloin nuoli poistaa sen aktiivivalikoista.

Jos listalla on toimimattomia pisteitä, ne voidaan poistaa valitsemalla vasemman reunan ”täpällä” ja painamalla roskakorin kuvaa. Esimerkiksi jos jokin laite lopettaa vastaanotusta ja se kytketään uudelleen järjestelmään ja saa uuden ID:n. Tällöin vanhat mittaukset voivat jäädä vielä listoilla.

Etsi listasta seuraavat pisteet ja tee ne vihreällä nuolella aktiiviseksi. Nimeä pisteet kysyttäessä helpommin tunnistettavaan muotoon. Esim. ”Pistoke on/off” ja ”Valoisuus”. Näitä pisteitä tarvitaan myöhemmässä vaiheessa tapahtumaohjelmointi esimerkissä.

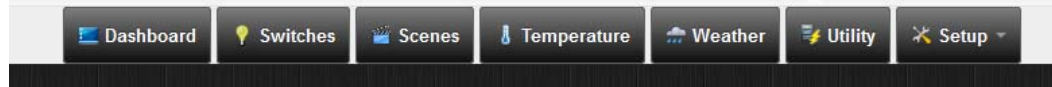
Multisensor:

Type: Lux,

Pistokemittari:

Name: Switch

Etsi aktivoituneet pisteet Domoticzin valikoista ja tarkista, että aktivointi onnistui.

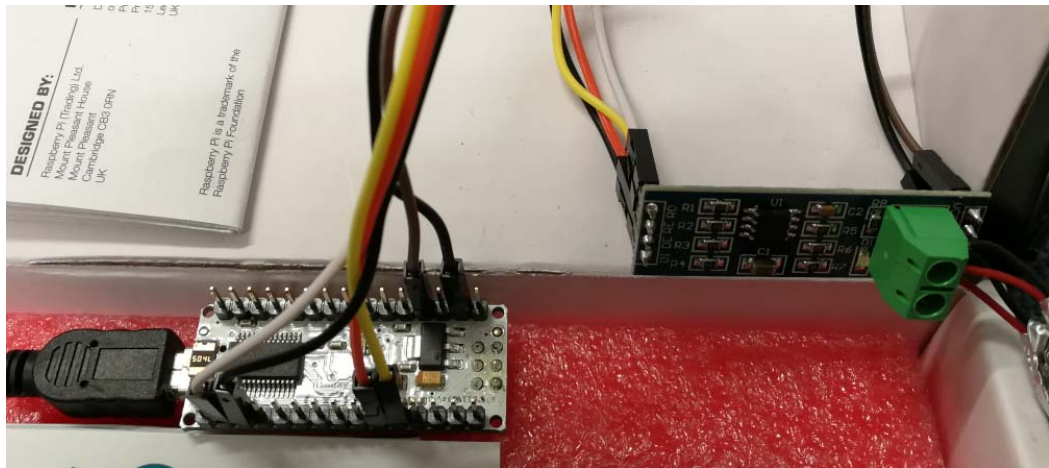


Nyt voit siirtyä yhdistämään Arduino-Modbus-sillan Domoticz järjestelmään tai tekemään Domoticzin tapahtumaohjelmointiharjoitusta. Molemmille on omat ohjeistuksensa. "Arduino Modbus asennus" ja "Domoticz tapahtumaohjelmointi"

Liite 8. Arduino-Modbus-sillan kytkennät ja ohjelmointi

Arduino Nano-kehitysalustan fyysiset kytkennät on tehty valmiiksi. Kytkennät on kuitenkin hyvä tarkistaa ennen aloittamista, ettei tapahdu mitään odottamatonta.

Arduino Modbus-silta koostuu kuvan mukaisesta kytkennästä.



Nanosta adapterille kytketään pinnit seuraavasti:

Nano: D3 – MAX: DE

Nano: D2 – MAX: RE

Nano: D11 – MAX: R0

Nano: D10 – MAX: DI

Nano: 5 V – MAX: VCC

Nano: GND – MAX: GND

Adapterin liittimen A ja B paikat kytketään Modbus-väylään niin, että punainen väyläjohdin on A ja musta B.

Väylälaitteet liittyvät väyläkaapeliin vastaavasti punainen A ja Musta B. Maata ei tarvitse kytkeä harjoituksessa. Poikkeuksena EnOcean-adapteri, jonka merkinnöissä D1+ vastaa punaista A-johdinta ja D1- mustaa B-johdinta

Väyläkaapeli päätetään 120 ohm vastuksella kuvan mukaisesti



!!!! TARKISTA ETTEI VIRTALÄHDE OLE JÄNNITTEELLINEN!!!!

Väylälaitteille tuodaan erillinen 24v käyttöjännite. Punainen +24v ja musta GND. Tarkista virtalähteeltä menevät johdot.

Väylä laitteiden osoitteet: HK instruments paneeli 1, sääasema 8

Laitteiden tulee keskustella nopeudella 9600 baud.

Käytössä ei ole pariteetti bittejä.

Asetukset voi tarkistaa seuraavasti.

EnOcean adapteri: Kytke vipukytkimet kuvan avustuksella annettuihin arvoihin.

Table 4a: DIP Switches - Modbus Address

| Modbus Address | Switch 1 | Switch 2 | Switch 3 | Switch 4 |
|----------------|----------|----------|----------|----------|
| 1 | OFF | OFF | OFF | OFF |
| 2 | ON | OFF | OFF | OFF |
| 3 | OFF | ON | OFF | OFF |
| 4 | ON | ON | OFF | OFF |
| 5 | OFF | OFF | ON | OFF |
| 6 | ON | OFF | ON | OFF |
| 7 | OFF | ON | ON | OFF |
| 8 | ON | ON | ON | OFF |
| 9 | OFF | OFF | OFF | ON |
| 10 | ON | OFF | OFF | ON |
| 11 | OFF | ON | OFF | ON |
| 12 | ON | ON | OFF | ON |
| 13 | OFF | OFF | ON | ON |
| 14 | ON | OFF | ON | ON |
| 15 | OFF | ON | ON | ON |
| 16 | ON | ON | ON | ON |

Table 4b: DIP Switches - Baud Rate

| Baud Rate | Switch 5 | Switch 6 |
|-----------|----------|----------|
| 2400 | OFF | OFF |
| 4800 | ON | OFF |
| 9600 | OFF | ON |
| 19200 | ON | ON |

Table 4c: DIP Switches - Parity

| Parity | Switch 7 | Switch 8 |
|--------|----------|----------|
| None | OFF | OFF |
| Even | ON | OFF |
| Odd | OFF | ON |
| None | ON | ON |

Please note if DIP Switches are altered while power is on then no change will take effect until power is re-applied.

!!! NYT VOIT KYTKEÄ VÄYLÄN VIRTA LÄHTEEN PÄÄLLE !!!

HK Instruments: Paneelissa on kosketusnäyttö. Pidä sormea näytöllä sen aikaa kunnes asetusvalikko aukeaa. Valitse valikosta "output" – "Modbus" ja aseta oikeat arvot jos ne poikkeavat.

Kytcentöjen ja asetusten tarkistuksen jälkeen voidaan siirtyä Arduino-Modbus-sillan ohjelmoimiseen.

Asennus suoritetaan Arduino IDE-ohjelmistolla. Version tulee olla 1.6.5 tai uudempi muutoin käytettävät kirjastot eivät toimi.

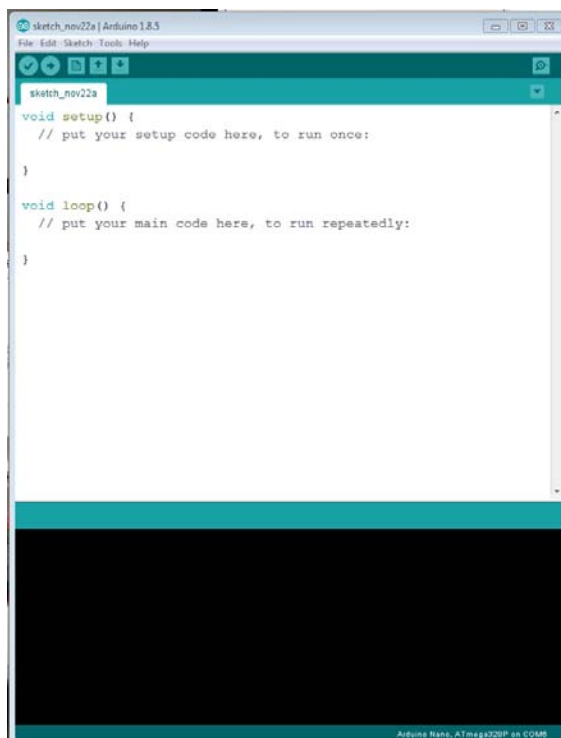
Harjoitukseen liittyvästä harjoituskansiosta löytyy tarvittaessa Arduino IDE-asennuspaketti versiolle 1.8.5, jos ohjelmistoa ei ole asennettu tai se on vanhentunut.

HUOM! Harjoitukseen liittyvässä Raspberry Pi käyttöjärjestelmässä Arduino IDE versio on 1.0.5 eikä sitä voi päivittää. Arduinon ohjelmointi tulee suorittaa PC:llä. Tarkista kuitenkin ohjaajalta onko Raspberry Pi'n asennusohjeet päivitetty uudemmalle Stretch-jakeluversiolle. Tällöin siihen voidaan asentaa uudempi Arduino IDE-ohjelmisto ja ohjelmoinnin voi suorittaa sen kautta.

Asenna Arduino IDE-ohjelmisto tarvittaessa tai avaa se kuvan mukaisesta ikonista työpöydältä tai Windowsin start-valikon kautta.

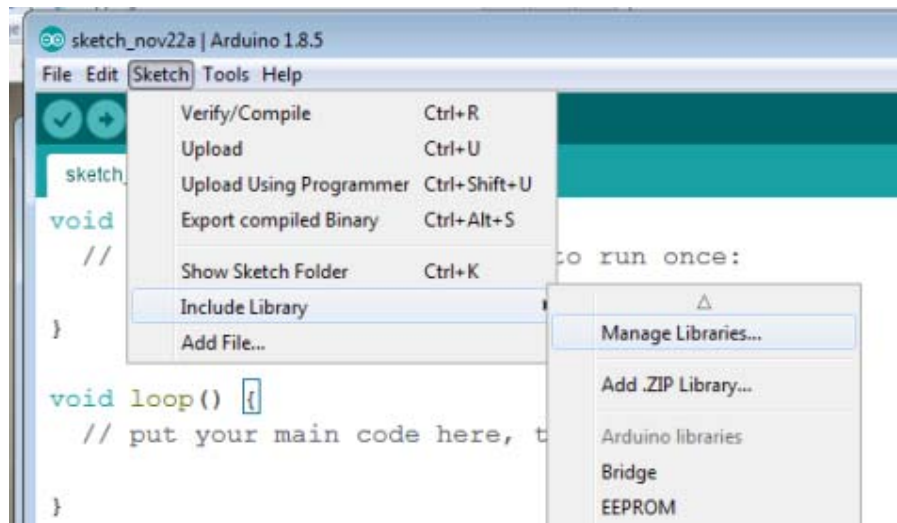


Ohjelma aukeaa kuvan mukaiseen näkymään.

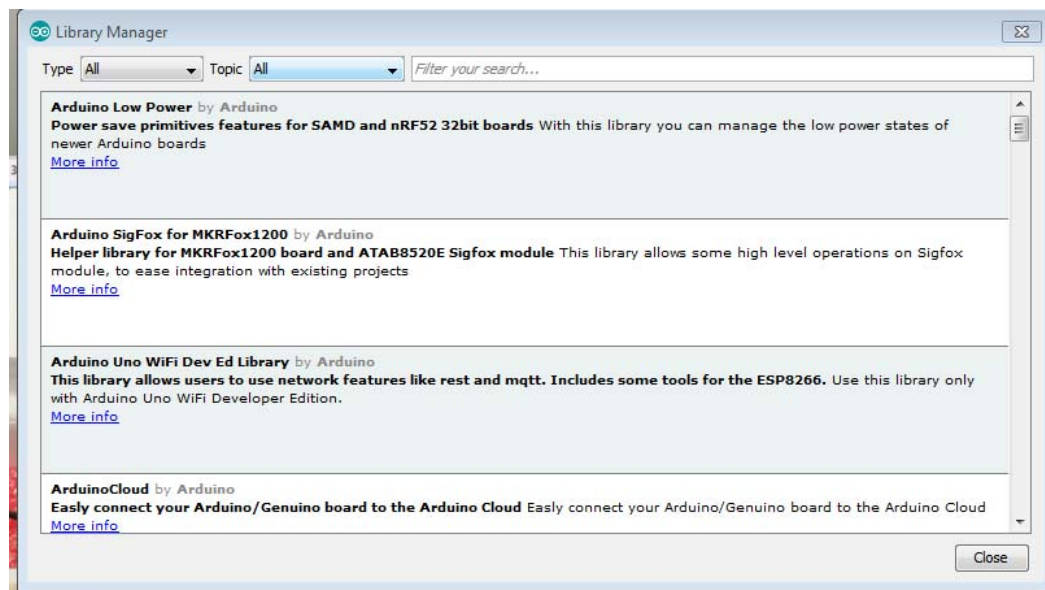


Ensimmäiseksi tarkistetaan löytyykö käytettävät kirjastot valmiiksi tietokoneelta tai tarvitseeko niitä päivittää. Kirjastot tulee asentaa jos niitä ei ole valmiiksi.

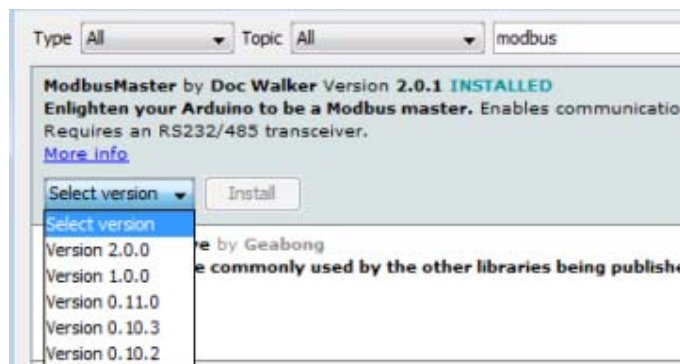
Avataan ohjelman ylävalikosta välilehti Sketch – Include library – Manage Libraries.



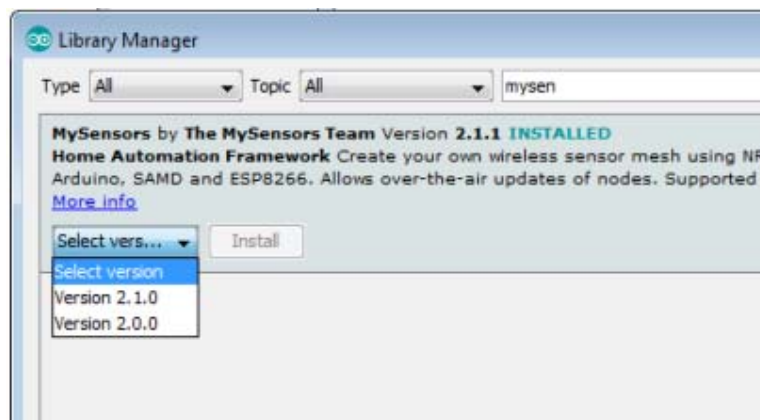
Tästä aukeaa Library Manager:



Kirjoitetaan haku kenttään "ModbusMaster". Hakukone ilmoittaa jos kirjasto on asennettu. Ohjelma on kirjoitettu versiolla 2.0.1. Jos versio on eri, se tulee päivittää vastaavaksi. Vaihtoehtoisesti koodin voi yrittää päivittää uudemmalle versiolle toimivaksi perehtymällä kirjaston tekijän muutosmerkintöihin.

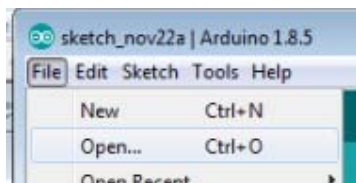


Tarkistetaan MySensors-kirjasto vastaavasti. Ohjelmakoodi on kirjoitettu versiolla 2.1.1.

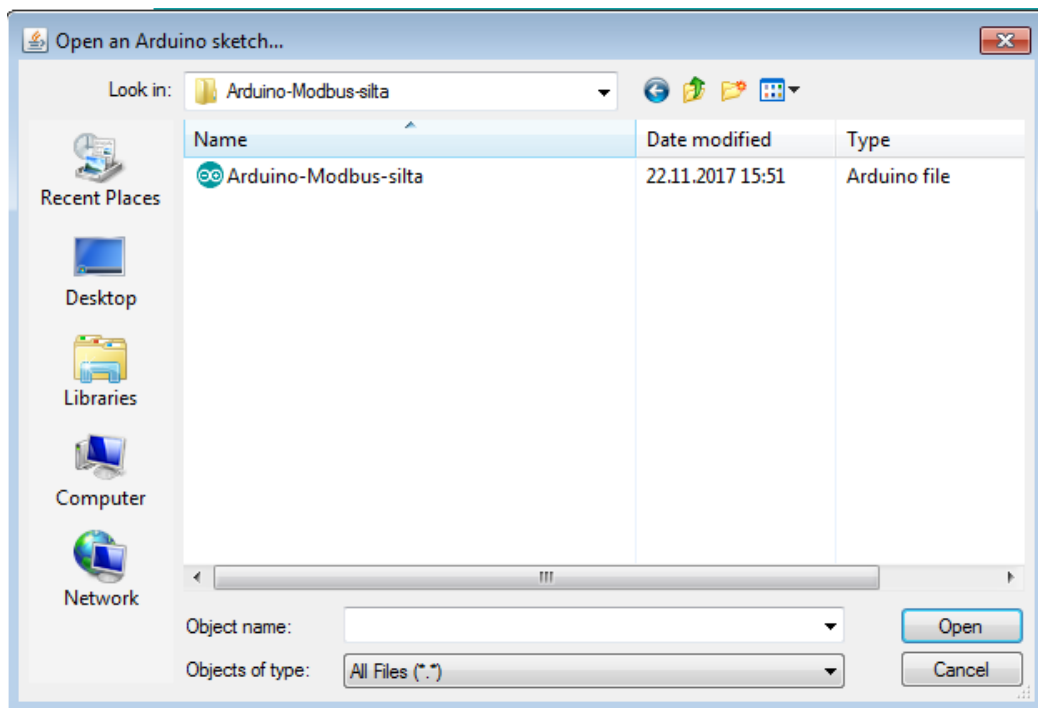


Kun kirjastot on tarkistettu tai asennettu valitsemalla versio ja painamalla "install", pitää ohjelmointi ympäristö käynnistää uudelleen sulkemalla pääikkuna ja avaamalla ohjelma uudelleen. Ohjelma tarkistaa käynnistyksessä kirjastot ennen kuin niitä voidaan käyttää.

Valitaan ohjelman auettua ylävalikosta "File" - "Open..."



Etsitään tietokoneelta harjoituskansio ja sieltä käytettävä malli. Avaa tiedosto.



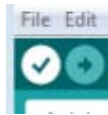
Ohjelmakoodi on myös tulostettuna tämän harjoituksen liitteenä. Käy ohjelmakoodi läpi ja tutustu sen osiohin. Osa alueet on kommentoitu ja niissä selitetään kunkin osion tehtävät.

Ohjelmakoodi koostuu kolmesta kirjastosta. Edellä asennetuista ohjelmistossa ja valmiiksi asennetusta SoftwareSerial-sarjaporttiemulattorista.

Ennen Arduinon ohjelmointia se pitää kytkeä käytettävän tietokoneen USB-porttiin sopivalla johdolla. Tarkista yhteys Arduino IDE:n Tools-valikosta. Ohjelman tulisi tunnistaa käytettävä alusta, sen prosessori ja yhdistetty USB-portin linkki COM*. Tarkista että ne vastaavat "Arduino Nano, ATmega328P, COM*" (portin voi tarkistaa tietokoneen Device Managerista kohdasta Ports. Sieltä löytyvä USB Serial on käytettävä portti.)

Käynnistä Tools-valikosta Serial Monitor -toiminto, kun oikea portti on aktivoitu. Tällä voidaan seurata Arduinolta tietokoneelle tulevaa viestintää. Valitse aukeavan ikkunan alalaidasta 115200 baudia. Arduinolla pyörii ohjelma jos ikkunaan alkaa tulostumaan tekstiä.

Valitse ohjelmakoodin ikkuna aktiiviseksi ja paina kuvan nappia.



Tällä IDE-ohjelmisto käy ohjelmakoodin läpi ja ilmoittaa jos ongelmia havaitaan.

Paina tämän jälkeen viereistä nuoli-ikonia jolloin ohjelmakoodi ladataan Arduinolle.

Tähän menee hetki joten voit avata Serial Monitorin taas aktiiviseksi. Siihen pitäisi tulostua kuvan mukainen ohjelman alustus.



```
COM6
0;255;3;0;9;MCO:BGN:INIT GW,CP=R-NGA--,VER=2.1.1
0;255;3;0;14;Gateway startup complete.
0;255;0;0;17;2.1.1
0;255;3;0;11;DOMOTICZ Harjoitus, RakLa
0;255;3;0;12;1.0
0;1;0;0;22;HK Co2-ppm
0;2;0;0;6;HK temperature
0;6;0;0;7;HK Kosteus %
0;3;0;0;6;Vantage Sääasema Ulko
0;12;0;0;7;Vantage rH% ulko
0;5;0;0;8;Vantage Barometri hPa
0;255;3;0;9;MCO:BGN:STP
0;255;3;0;9;MCO:REG:NOT NEEDED
0;255;3;0;9;MCO:BGN:INIT OK,TSP=NA
ModbusSlave 1
0;2;1;0;0;23.30
0;1;1;0;37;496
0;6;1;0;1;12.50
ModbusSlave 8
0;5;1;0;4;1014.20
0;3;1;0;0;19.80
0;12;1;0;1;19.00
ModbusSlave 1
0;2;1;0;0;23.30
0;1;1;0;37;495
0;6;1;0;1;12.50
ModbusSlave 8
0;5;1;0;4;1014.20
0;3;1;0;0;19.80
0;12;1;0;1;19.00
Autoscroll Newline
```

Jos alustus sisältää jotain muuta ohjelmansuorittamisessa on ongelmia.

Arduinossa on nyt toimiva tarkoituksen mukainen ohjelma. Tällä varmistetaan seuraan kohdan onnistuminen.

Ohjelmakoodissa on kommentoituna osio jolla väylään kirjoitetaan tietoa. Harjoituksessa esimerkkinä käytetään HK Instruments paneelin analogisen lähdön skaalausta eli mikä mittausarvo vastaa analogisen lähdön ala ja mikä ylärajaa.

Voit valita koodin ohjeistuksen avulla minkä mittauksen haluat vaikuttavan lähtöön ja millaisella mittausvälillä lähtö skaalataan. Apuna voi käyttää seuraavaan kuvaa ohjelmakoodin lisäksi. Käytä muita kuin äärirajoja.

Function code 06 - Write single register

| Register | Parameter description | Data Type | Value | Range |
|----------|----------------------------|-----------|------------|---------------------------------|
| 4x0001 | Parameter for P-controller | 16 bit | 0..3 | 0-CO2, 1-rH, 2-TE, 3-MAX, 4-Off |
| 4x0002 | CO2 high limit | 16 bit | 500...2000 | 500...2000 ppm |
| 4x0003 | CO2 low limit | 16 bit | 0...1900 | 0...1900 ppm |
| 4x0004 | rH high limit | 16 bit | 100...1000 | 10.0...100.0 % |
| 4x0005 | rH low limit | 16 bit | 0...900 | 0.0...90.0 % |
| 4x0006 | TE high limit | 16 bit | 50...500 | 5.0...50.0 °C |
| 4x0007 | TE low limit | 16 bit | 0...450 | 0...45.0 °C |
| 4x0008 | Parameter for relay | 16 bit | 0..3 | 0-CO2, 1-rH, 2-TE, (3-Off) |
| 4x0009 | CO2 relay on | 16 bit | 500...1950 | 500...1950 ppm |
| 4x0010 | CO2 relay off | 16 bit | 450...1900 | 450...1900 ppm |
| 4x0011 | rH relay on | 16 bit | 15...990 | 1.5...99.0 % |
| 4x0012 | rH relay off | 16 bit | 10...985 | 1.0...98.5 % |
| 4x0013 | TE relay on | 16 bit | 15...490 | 1.5...49.0 °C |
| 4x0014 | TE relay off | 16 bit | 10...485 | 1.0...48.5 °C |

Tehtyäsi valinnat voit ladata ohjelmakoodin arduinolle. Avaa taas Serial Monitori ja odota sarjaliikenteen käynnistyminen. Onnistunut alustus antaa seuraavan viestin.

Huomaa kohdat "Result write ok: 0" jos joissakin kolmesta kohdasta on muuta kuin 0 yhteydessä on ongelmia.

```

COM6
|
|
0;255;3;0;9;MCO:BGN:INIT GW,CP=R-NGA--,VER=2.1.1
0;255;3;0;14;Gateway startup complete.
0;255;0;0;17;2.1.1
0;255;3;0;11;DOMOTICZ Harjoitus, RakLa
0;255;3;0;12;1.0
0;1;0;0;22;HK Co2-ppm
0;2;0;0;6;HK temperature
0;6;0;0;7;HK Kosteus %
0;3;0;0;6;Vantage Sääasema Ulko
0;12;0;0;7;Vantage rH% ulko
0;5;0;0;8;Vantage Barometri hPa
0;255;3;0;9;MCO:BGN:STP
Result write ok:
0
Result write ok:
0
Result write ok:
0
0;255;3;0;9;MCO:REG:NOT NEEDED
0;255;3;0;9;MCO:BGN:INIT OK,TSP=NA
ModbusSlave 1
0;2;1;0;0;23.40
0;1;1;0;37;487
0;6;1;0;1;12.50
ModbusSlave 8
0;5;1;0;4;1014.20
0;3;1;0;0;19.80
0;12;1;0;1;19.00

```

Tarkista vielä mittapaneelin asetuksista kohdasta "outputs" – "regulator", että halutut arvot ovat kirjattuna muistiin. Kohdasta "Parameter" aktiivisena pitäisi olla kirjoittamasi mittaus. Valikon "P-band" kohdissa high ja low limits asettamasi arvot.

Tämän jälkeen voit kommentoida uudestaan resultsetup-muuttujan ja koko kirjoitusosion. Aja uudelleen kommentoitu ohjelma Arduinolle ja tarkista Serial Monitorilla, että väyläsilta toimii ongelmitta.

Käy seuraavaksi alla olevia ohjeita apuna käyttäen mitä kaikkea sarjaliikenteestä selviää ja vertaa liikenteen viestejä itse ohjelmakoodiin.

Sarjaporttiliikenteessä Esimerkiksi HK instrumentin laitemittaukset Co2 ja Temperature alustetaan mittalaitteiksi seuraavasti

```
0;1;0;0;22;HK Co2-ppm
0;2;0;0;6;HK temperature
```

Tästä selviää mittalaitteen ID:t "1" ja "2" sekä käytettävä mittauslaatu "22" ja "6".

Alustuksen jälkeen vastaavilta mittauksilta saatu tieto tuodaan sarjaliikenteessä Domaticille seuraavasti:

```
ModbusSlave 1
0;2;1;0;0;23.40
0;1;1;0;37;487
```

";2;" ja ";1;" kertovat mittapisteen ID:n, seuraavat ";1;" ja ";1;" ovat käskyjä asettaa mitta-arvot. ";0;" ja ";37;" kertovat mittauksen "V_*" laadun. ";23.4" ja ";487" arvoille asetettavat mittaukset laitteilta.

Voit verrata muita sarjaliikenteen viestejä käyttäen apuna seuraavaa MySensors-sarjaliikenne vinkkiä. Tarkemmin aiheesta hakemalla selaimella ”Mysensors serial format”

Message Structure

The serial protocol used between the Gateway and the Controller is a simple semicolon separated list of values. The last part of each "command" is the payload. All commands ends with

node-id ; child-sensor-id ; command ; ack ; type ; payload \n

You can play around with the debug output from your gateway or nodes using our [online log parser](#).

| Message Part | Comment |
|------------------------|---|
| node-id | The unique id of the node that sends or should receive the message (address) |
| child-sensor-id | Each node can have several sensors attached. This is the child-sensor-id that uniquely identifies one attached sensor |
| command | Type of command sent - See table below |
| ack | The ack parameter has the following meaning: Outgoing: 0 = unacknowledged message, 1 = request ack from destination node, Incoming: 0 = normal message, 1 = |
| type | Depending on command, the type field has different meaning. See tables below |
| payload | The payload holds the message coming in from sensors or instruction going out to actuators. |

The maximum payload size is 25 bytes! The NRF24L01+ has a maximum of 32 bytes. The MySensors library (version 2.0) uses 7 bytes for the message header.

command

| Type | Value | Comment |
|--------------|-------|--|
| presentation | 0 | Sent by a node when they present attached sensors. This is usually done in the presentation() function which runs at startup. |
| set | 1 | This message is sent from or to a sensor when a sensor value should be updated |
| req | 2 | Requests a variable value (usually from an actuator destined for controller). |
| internal | 3 | This is a special internal message. See table below for the details |
| stream | 4 | Used for OTA firmware updates |

presentation

When a presentation message is sent from a sensor, **type** can one the following:

The payload of presentation message will be set to the library version (node device) or an optional description for the sensors.

| Type | Value | Comment | Variables |
|----------|-------|------------------------------|--|
| S_DOOR | 0 | Door and window sensors | V_TRIPPED, V_ARMED |
| S_MOTION | 1 | Motion sensors | V_TRIPPED, V_ARMED |
| S_SMOKE | 2 | Smoke sensor | V_TRIPPED, V_ARMED |
| S_BINARY | 3 | Binary device (on/off) | V_STATUS, V_WATT |
| S_DIMMER | 4 | Dimmable device of some kind | V_STATUS (on/off), V_PERCENTAGE (dimmer level 0-100), V_WATT |
| S_COVER | 5 | Window covers or shades | V_UP, V_DOWN, V_STOP, V_PERCENTAGE |
| S_TEMP | 6 | Temperature sensor | V_TEMP, V_ID |

Kun olet saanut käsityksen miten ohjelmakoodi toimii siirry seuraavaan osioon.

Liite 9. Domoticz-käyttöliittymä: Arduino-modbus-sillan asennus

Jos käytät Domoticzia Raspberry Pi'n kosketusnäytöltä, anna ohjelmalle aikaa ja vältä turhia napinpainalluksia. Raspberry Pi'n grafiikkaohjain ei ole maailman nopein.

Arduino-Modbus-silta on nyt ohjelmoitu ja testattu toimivaksi. Tämän jälkeen silta voidaan kytkeä Raspberry Pi'n USB-porttiin ja asentaa Domoticz-käyttöliittymään.

Aloita liittämällä Arduinolta USB-johto Raspberry Pi'n vapaaseen USB-porttiin.

Ota tämän jälkeen yhteys Domoticz-käyttöliittymään verkkoselaimella, joko verkon yli IP-osoitteella ja portilla tai vastaavasti paikallisesti Raspberry Pi'stä.

Siirrytään Domoticz-käyttöliittymän Setup valikon Hardware lehdelle.



Lisätään kuvan mukaisesti MySensor gateway, jonka läpi Modbus toimii.

Showing 1 to 1 of 1 entries

Update Delete

Enabled:

Name:

Type:

Data Timeout:
 Specifying a Data Timeout will restart the hardware device if no data is received for the specified time.
 Do not enable this option for devices that do not receive data!

Serial Port:

Baudrate:

Add

Listaan pitäisi ilmestyä kuvan mukainen merkintä

Domoticz v3.8.153

Dashboard Switches Scenes Temperature Weather

Show 25 entries Search

| Idx | Name | Enabled | Type | Address | Port |
|-----|----------------|---------|---|---------|--------------|
| 4 | Arduino-Modbus | Yes | MySensors Gateway USB Version: ? Setup | | /dev/ttyUSB0 |

”Version ?” päivittyy kun järjestelmät saavat yhteydet varmistettua. Paina riviltä si-
nistä Setup-ikonia, jolloin kyseistä järjestelmää päästään säätämään.

Napista aukeaa kuvan mukainen näkymä. Alempi lista avautuu automaattisesti, kun
valitset ylemmän listan aktiiviseksi (aktiivinen rivi on aina tummemmalla värillä) Kir-
joita listojen välissä olevaan Name-kenttään Nodelle nimi jolla se Domoticsissa tun-
nistetaan, ja paina vieressä olevaa update-nappia.

The screenshot shows the Domoticz web interface. At the top, there are navigation tabs: Dashboard, Switches, Scenes, Temperature, Weather, Utility, and Setup. Below the navigation, the device name 'Arduino-Modbus' is displayed. The main content is divided into two sections: 'Nodes' and 'Children'.

Nodes Section:

- Shows 25 entries.
- Table with columns: NodeID, Name, Sketch Name, Version, Children, Last Seen.
- One entry is visible: NodeID 0, Name Unknown, Sketch Name DOMOTICZ Harjoitus, RakLa, Version 1.0, Children 7, Last Seen 2017-11-23 09:59:12.
- Buttons: Update, Delete, Refresh.
- Form: Name: Unknown

Children Section:

- Shows 25 entries.
- Table with columns: ChildID, Type, Name, Values, Ack, Ack Timeout, Last Seen.
- Seven entries are visible:

| ChildID | Type | Name | Values | Ack | Ack Timeout | Last Seen |
|---------|----------------|-----------------------|-------------------------|-------|-------------|---------------------|
| 1 | S_AIR_QUALITY | HK Co2-ppm | #1. V_LEVEL (477) | true | 1200 | 2017-11-23 09:59:27 |
| 2 | S_TEMP | HK temperature | #1. V_TEMP (22.9) | true | 1200 | 2017-11-23 09:59:27 |
| 3 | S_TEMP | Vantage Sääasema Ulko | #1. V_TEMP (19.2) | true | 1200 | 2017-11-23 09:59:29 |
| 5 | S_BARO | Vantage Barometri hPa | #1. V_PRESSURE (1010.1) | true | 1200 | 2017-11-23 09:59:29 |
| 6 | S_HUM | HK Kosteus % | #1. V_HUM (13) | true | 1200 | 2017-11-23 09:59:27 |
| 12 | S_HUM | Vantage rH% ulko | #1. V_HUM (20) | true | 1200 | 2017-11-23 09:59:29 |
| 255 | S_ARDUINO_NODE | 2.1.1 | | false | 1200 | 2017-11-23 09:58:07 |

Buttons: Update, Delete, Refresh.

MySensors-järjestelmässä Node-tarkoittaa etälaitetta johon on kiinnitettyä mittalaitteita. Uusimmassa ohjelmaversiossa Node voi olla myös Gateway-yhteys Domoticzille. Tämän takia tässä ei näy erillistä Gateway-ohjainta.

Erilliset mittaukset näytetään alemmassa listauksessa Children-laitepisteinä. Nämä voivat olla mittauksia, painonappeja yms.

Arduino-ohjelmakoodissa määriteltyjen käskyjen mukaan määräytyvät mittapisteen ja niiden käyttämät mittalaadut ja mittapisteelle annettu nimi näkyvät Children-listassa. S_***** -kertoo Domoticzille mittalaitteen tyyppin (S=sensor) ja V_*** kertoo Domoticzille millaista tietoa laitteesta saadaan ("V=variable"). Näillä tiedoilla Domoticz osaa antaa käyttöliittymässä näytettäville pisteille oikeanlaiset tyyppitykset.

Väylästä saatavat mittaukset näkyvät suluissa Values-kentän muuttuja listauksessa.

| ChildID | Type | Name | Values |
|---------|---------------|-----------------------|-------------------------|
| 1 | S_AIR_QUALITY | HK Co2-ppm | #1. V_LEVEL (477) |
| 2 | S_TEMP | HK temperature | #1. V_TEMP (22.9) |
| 3 | S_TEMP | Vantage Sääasema Ulko | #1. V_TEMP (19.2) |
| 5 | S_BARO | Vantage Barometri hPa | #1. V_PRESSURE (1010.1) |

Sarjaporttiliikenteessä esimerkiksi HK instrumentin laitteen Co2 ja Temperature alustetaan mittalaitteiksi seuraavasti

```
0;1;0;0;22;HK Co2-ppm
0;2;0;0;6;HK temperature
```

Tästä selviää mittalaitteen ID "1" ja "2" sekä käytettävä mittalaatu "22" ja "6".

Alustuksen jälkeen vastaavilta mittauksilta saatu tieto tuodaan sarjaliikenteessä Domoticzille seuraavasti:

```
ModbusSlave 1
0;2;1;0;0;23.40
0;1;1;0;37;487
```

;2; ja ;1; kertovat taas mittapisteen ID:n, seuraavat ;1; ja ;1; ovat käskyjä asettaa mit-taarvot. ;0; ja ;37; kertovat mittauksen V_* laadun. ";23.4" ja ";487" arvoille asetetta-vat mittaukset laitteilta.

Siirry seuraavaksi Domoticz Setup-valikon Devices lehdelle

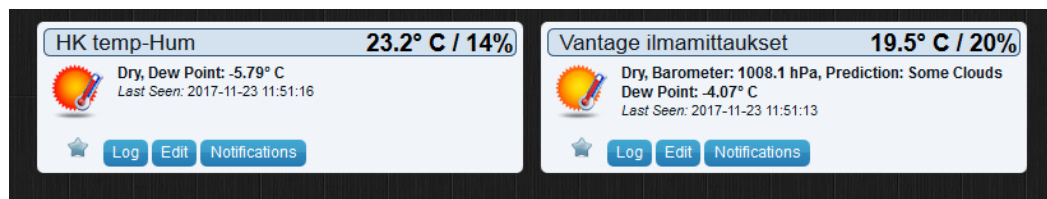


Modbus-väylän laitteet pitäisi näkyä kuvaa vastaavalla tavalla:

| Idx | Hardware | ID | Unit | Name | Type | SubType | |
|-----|----------------|------|------|-----------------------|------------------------|-----------------|--------------------------|
| 21 | Arduino-Modbus | 0000 | 1 | HK Co2-ppm | Air Quality | Voltcraft CO-20 | 883 ppm |
| 22 | Arduino-Modbus | 000A | 0 | HK temperature | Temp + Humidity | WTGR800 | 23.0 C, 14 % |
| 23 | Arduino-Modbus | 000F | 15 | Vantage Barometri hPa | Temp + Humidity + Baro | Weather Station | 19.5 C, 20 %, 1008.1 hPa |

Domotizcin sisäisistä ohjauksista johtuen tietyt tyyliset mittaukset linkittyvät yhteen vaikka ne Hardware-osioissa näkyvät liitetyn järjestelmän osalta irrallisina. HK Instruments paneelin lämpötila ja kosteus ovat yhdessä sekä sääaseman tilamittaukset. Myös Z-Wave Multisensorin lämpötila ja kosteus toimivat vastaavalla tavalla. Tämä vaikuttaa vain siihen miten mittapisteen näytetään Domoticzin-käyttöliittymässä. Tapahtumaohjelmoinnissa yksittäiset mittaukset voidaan käsitellä erillään.

Lisää Arduino-Modbus-laitteet oikealta näkyvällä vihreällä nuolella ja nimeä mittapisteen haluamallasi tavalla tunnistettavasti. Löydät nyt mittaukset Ylälaidan välilehden Temperatures alta.



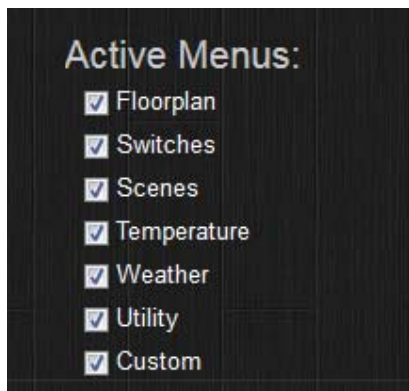
Domoticz käynnistyy aina sen Dashboard välilehdelle. Välilehdelle voi lisätä mittapistettä aktivoimalla mittapisteen omilta välilehdiltä mittauskohtaisesti pienellä tähtikonilla.



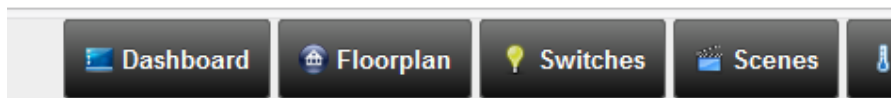
Tarkista valitsemasi mittaukset Dashboard-sivulta.

Domoticzilla voidaan luoda myös pohjakuva-kartta laitepisteistä. Avaa Setup-valikon Settings-valinta. Etsi alalaidasta kuvan kohta ja aktivoi kohta Floorplan.

HUOM! Käytettäessä Raspberry Pi'n kosketusnäytöltä ikonissa ei välttämättä näy harkasta, vaan ruutuvaihtaa väriä. Tummempi värisävy tarkoittaa valintaa ja vaalea avointa.



Tämän jälkeen paina ylälaidasta "apply settings" Välikehtivalikkoon pitäisi nyt ilmes-tyä Floorplan-sivu.



Avaa Setup-valikosta seuraavat -> More Options->Plan->Room Plans.

Tässä valikossa voidaan luoda huoneita joihin voidaan määritellä eri laitepisteitä.

The screenshot shows a web interface with two main sections. The top section is a table with columns 'Idx', 'Name', and 'Order'. It contains two entries: '2 Huone1' and '1 \$Hidden Devices'. Below the table are 'Edit' and 'Delete' buttons. The bottom section is titled 'Devices' and includes a search bar, a 'Delete' button, and a dropdown menu for 'Device' with the value '[Arduino-Modbus] HK Co2-ppm (Air Quality/Volcraft CO-20)' and an 'Add' button.

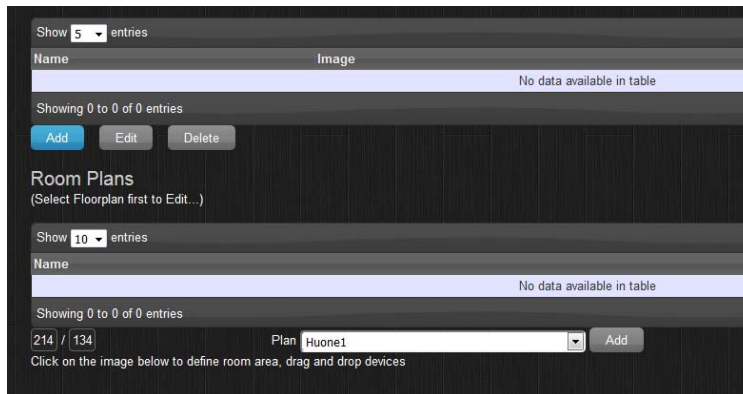
Paina yläkulman Add Plan -nappia ja anna huoneelle nimi. Aktivoi tämän jälkeen huone listasta ja lisää siihen alalaidan alavetovalikosta laitepisteitä.

Mitta/laitepisteet ilmaantuvat Devices listalle:

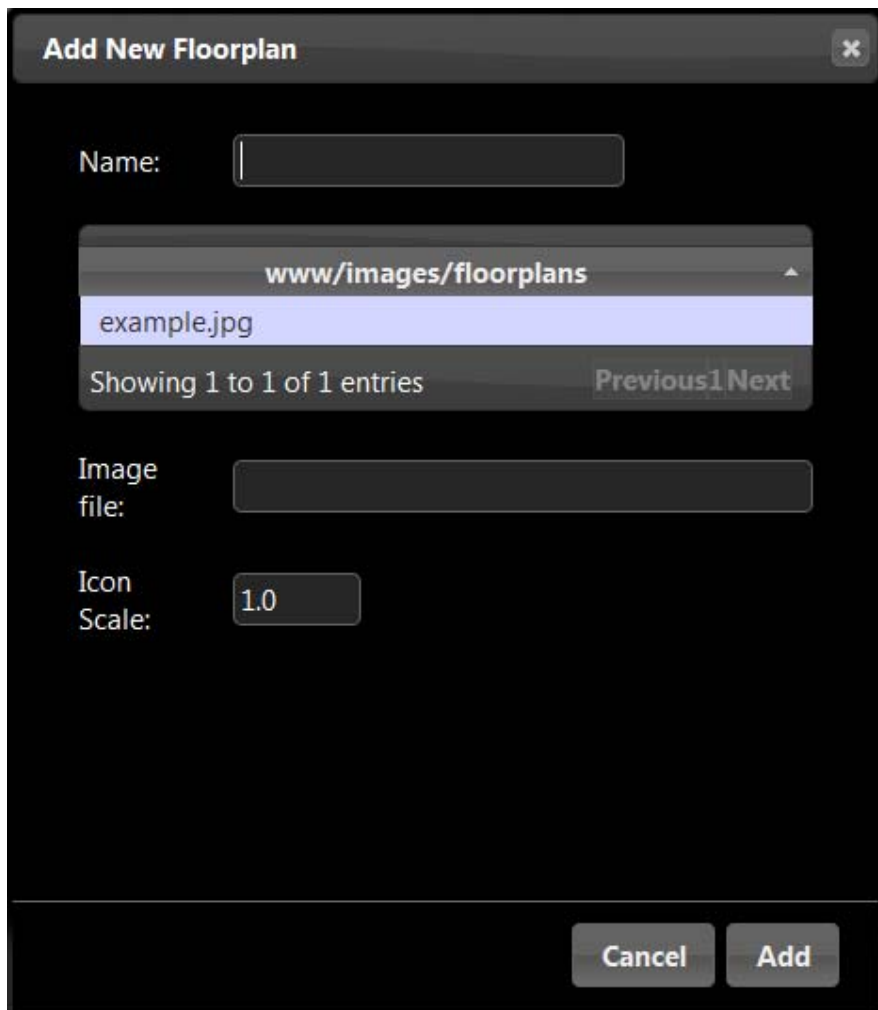
The screenshot shows the 'Devices' section of the web interface. It features a table with columns 'Idx', 'Name', and 'Order'. There are five entries: '7 kWh Meter', '17 Valoisuus', '8 Unknown', '22 HK temp-Hum', and '21 HK Co2-ppm'. Each entry has a green up arrow and a red down arrow in the 'Order' column. Below the table are 'Delete' and 'Clear' buttons. At the bottom, there is a dropdown menu for 'Device' with the value '[Arduino-Modbus] HK Co2-ppm (Air Quality/Volcraft CO-20)' and an 'Add' button.

Varmista että huoneesi on ylävalikossa ensimmäisenä käyttämällä nuoli-ikoneita.

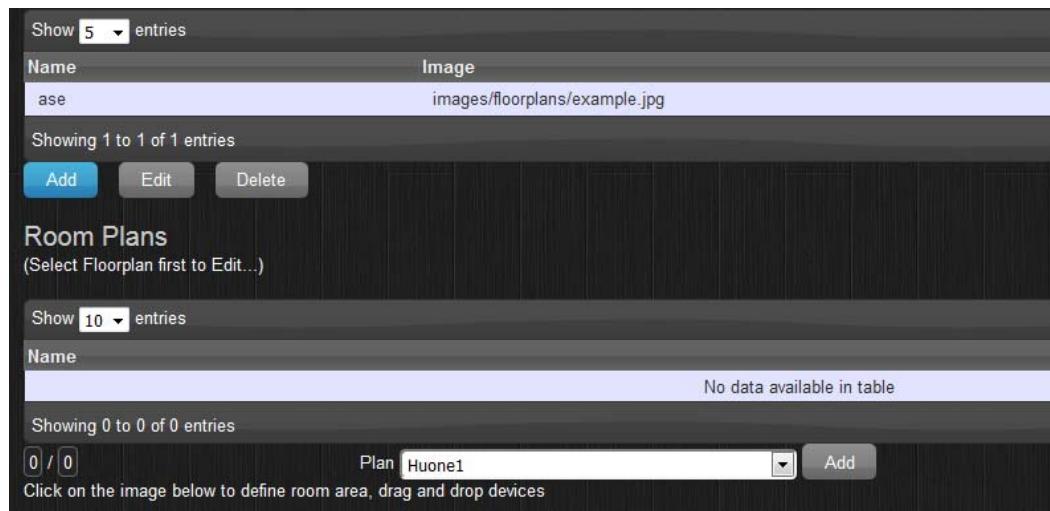
Avaa Setup valikon kautta More Options-Plans-Floorplan ja varmista, että alalaidan alasettovalikosta on valittuna tekemäsi huone.



Paina Add ja anna pohjakuvalle nimi ja valitse Examples.jpg aktiiviseksi. Paina tämän jälkeen alalaidan Add.



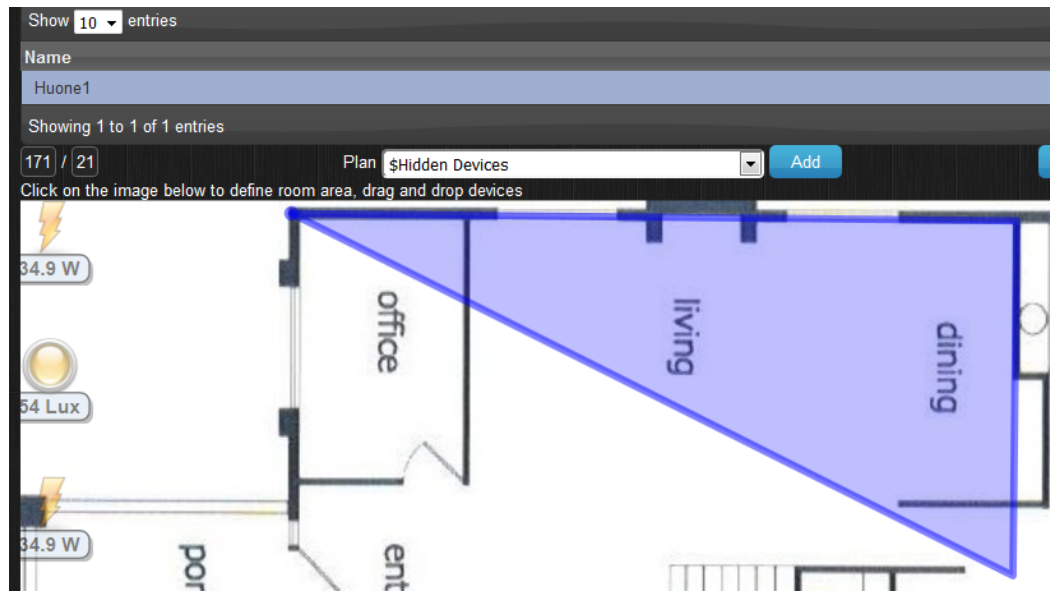
Pohjakuva pitäisi nyt näkyä listalla:



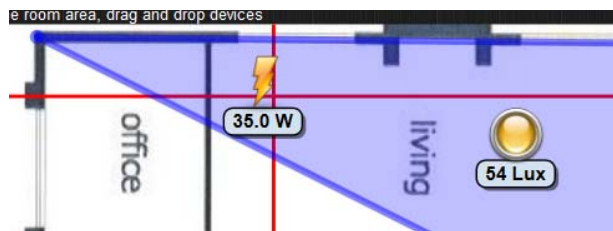
Aktivoi pohjakuva jolloin näytölle ilmaantuu kuva ja Add-nappi aktivoituu. Paina nappista jolloin huonekohtaiset laitteet tulevat listalle



Kuvan päälle ilmestyy punainen ristikko jolla kuvasta voidaan määritellä laitteille rajattu käyttöalue huonekohtaisesti. Piirrä ristikolla sopiva alue siirtämällä osoitin haluamasi alueen kulmaan. Paina hiirestä kerran ja siirrä kohdistin seuraavaan kulmaan ja paina hiirestä kerran. Toista kunnes haluamasi alue on valittuna.



Siirrä tämän jälkeen vasemmanlaidan laitepisteet alueelle vetämällä ne osoittimella haluttuun kohtaan.



Kun laitteet on siirretty paikalleen paina Update-nappia kuvan ylälaidasta jolloin suunnitelma päivitetään Floorplan-välilehdelle.

Omia pohjakuvia voi ladata Domoticzin asennuskansioon:

</home/pi/domoticz/www/images>

Tutustu Floorplan-välilehden sisältöön painamalla eri mittapisteitä ja niistä aukeavia sivuja.

Mihin näitä tietoja voitaisiin käyttää?

Siirry seuraavaksi tapahtumaohjelmointi-ohjeisiin.

Liite 10. Domoticz-käyttöliittymä: Blockly-tapahtumaohjelmointi

Domoticz-käyttöliittymän kautta voidaan tehdä monenlaisia tapahtumaohjelmointeja usealla eri ohjelmointikielillä. Domoticz-tukee myös monia muita ohjelmointikieliä mutta niillä tehdyt ohjelmat pitää kirjoittaa erikseen ja tallentaa käsin Domoticzin kansioihin. Harjoituksessa perehdytään käyttöliittymän kautta tehtäviin ohjelmointeihin: Blockly, LUA-script ja Python.

Tapahtumaohjelmoinnit suoritetaan Domoticzin Setup-valikon kohdasta "Events"

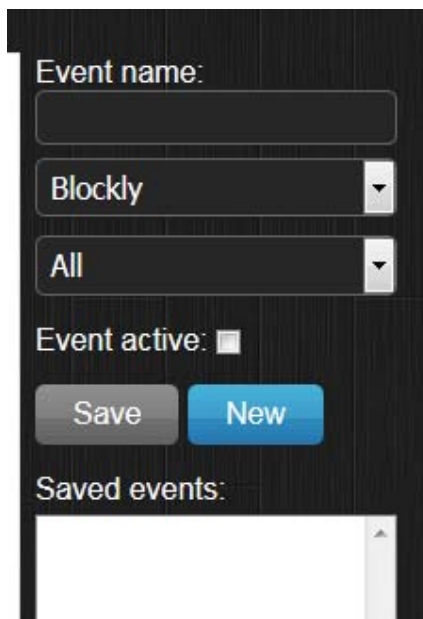


Tästä aukeaa ohjelmointi portaali:



Ohjelmoidaan yksinkertainen tapahtuma esimerkki käyttäen Domoticz-käyttöliittymän ohjelmointiportaalia.

Kirjoita ohjelmointi-ikkunan oikeanlaidan valikkoon tapahtumalle kuvaava nimi kohtaan "Event Name". Jätä event active kohta toistaiseksi tyhjäksi ettei ohjelma vahingossa aja keskeneräisenä. Alasvetovalikoista voidaan valita käytettävä ohjelmointikieli ja tarkoittaa ohjelman määritelmää, jolla sen löytää myöhemmin. Harjoituksessa kohtaan voi jättää All -valinta. . Blockly valinnalla ohjelma ei anna tallentaa ennen kuin ensimmäinen elementti on asetettu ohjelmointipöydälle. Tallenna ohjelman myöhemmässä vaiheessa.

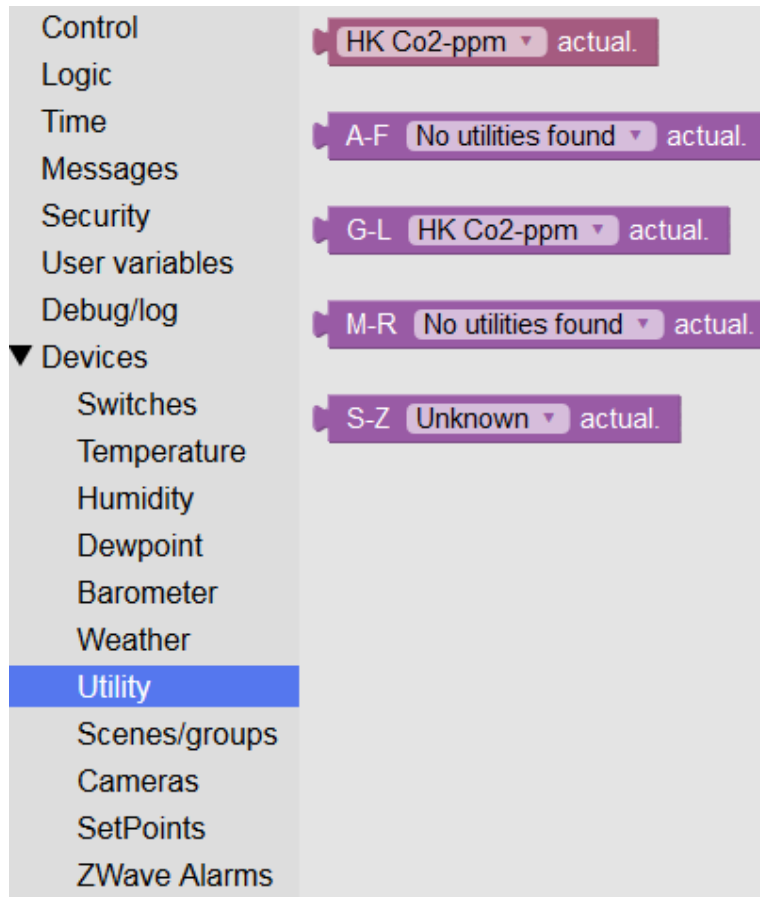


The image shows a dark-themed mobile application interface for configuring an event. It features the following elements:

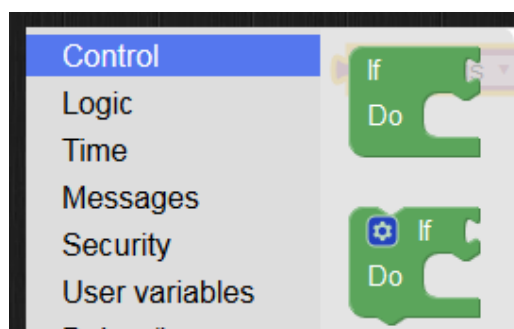
- Event name:** A text input field.
- Blockly:** A dropdown menu with a downward arrow.
- All:** A dropdown menu with a downward arrow.
- Event active:** A checkbox that is currently unchecked.
- Save:** A grey button.
- New:** A blue button.
- Saved events:** A list view area at the bottom, currently empty.

Ohjelmointi-ikkunan vasemmasta laidasta löytyy Blockly-elementit, joita käytetään ohjelman luomiseen.

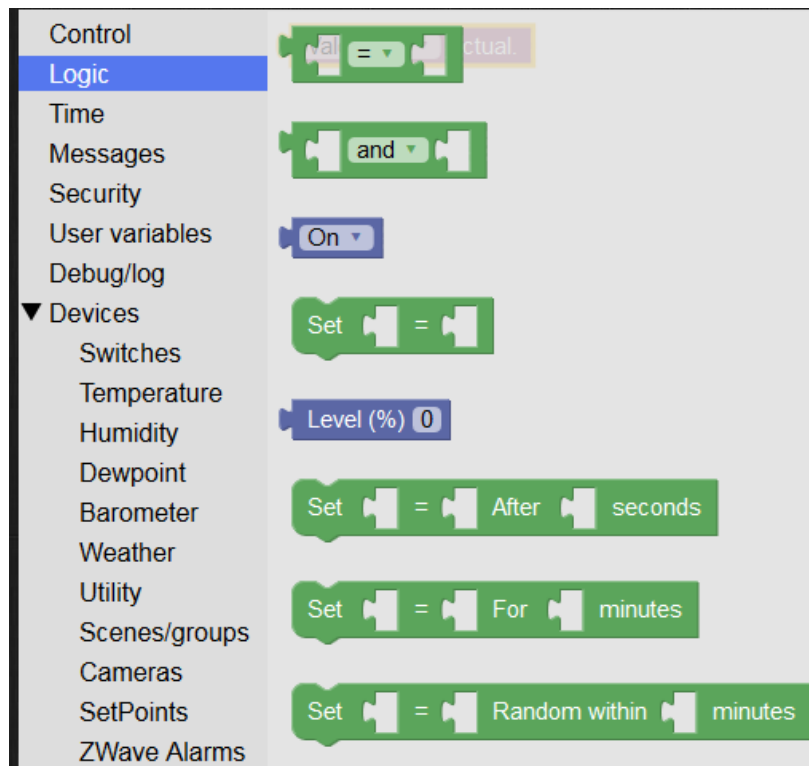
Devices: Tästä löytyvät Domoticziin liitetyt mitta- ja laitepisteet. Esim. Utilityn alta tekniset mittaukset, sähkönkulutus, CO2-pitoisuus, valoisuus-Lux jne.



Control: Perinteiset ohjelmoinnissa käytetyt if/do elementit.

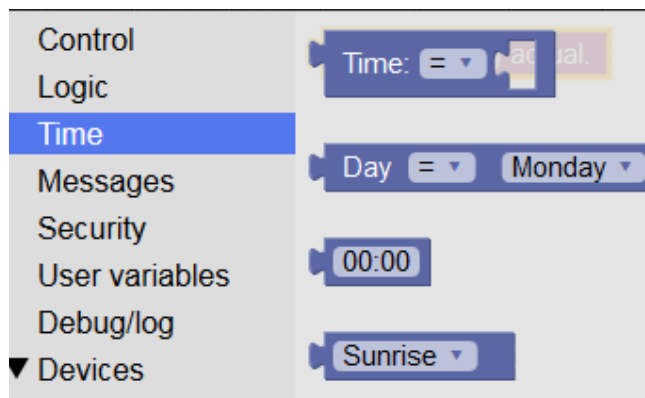


Logic: Loogiset ehdot kuten



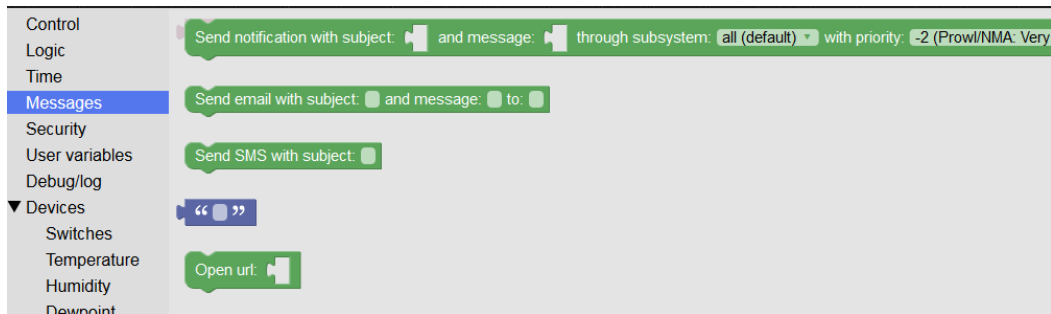
The screenshot shows a logic editor interface with a sidebar on the left and a workspace on the right. The sidebar lists categories: Control, Logic (highlighted), Time, Messages, Security, User variables, Debug/log, Devices (expanded), Switches, Temperature, Humidity, Dewpoint, Barometer, Weather, Utility, Scenes/groups, Cameras, SetPoints, and ZWave Alarms. The workspace contains several logic blocks: a green 'Set' block with a dropdown menu and a text input field containing 'actual.', a green 'and' block, a blue 'On' block, a green 'Set' block with an equals sign, a blue 'Level (%)' block with a text input field containing '0', a green 'Set' block with an equals sign, a dropdown menu containing 'After', and a text input field containing 'seconds', a green 'Set' block with an equals sign, a dropdown menu containing 'For', and a text input field containing 'minutes', and a green 'Set' block with an equals sign, a dropdown menu containing 'Random within', and a text input field containing 'minutes'.

Time: Käyttöliittymän aikaelementit aikasidonnaisten ehtojen suorittamiseen.

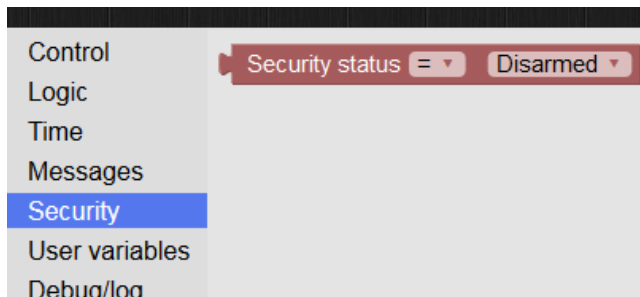


The screenshot shows a logic editor interface with a sidebar on the left and a workspace on the right. The sidebar lists categories: Control, Logic, Time (highlighted), Messages, Security, User variables, Debug/log, and Devices (expanded). The workspace contains several time-related logic blocks: a blue 'Time:' block with a dropdown menu and a text input field containing 'actual.', a blue 'Day' block with a dropdown menu containing 'Monday', a blue '00:00' block, and a blue 'Sunrise' block with a dropdown menu.

Messages: Näillä komendoilla voidaan ohjelmaan liittää ilmoituksia tapahtumista paikalliseen verkkoon, sähköpostiin, toiseen liitettyyn järjestelmään jne.

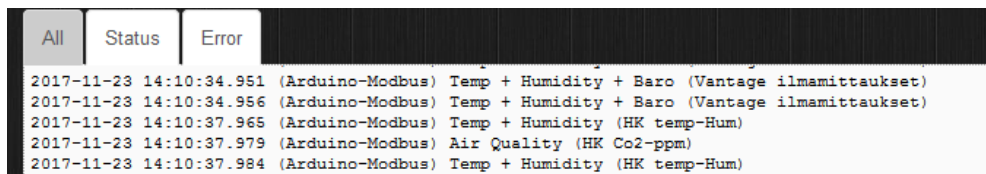
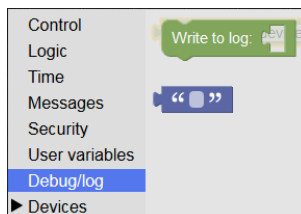


Security: Domoticziin rakennettu murtohälytintoiminto ja sen komennot.



User Variables. käyttöliittymään voidaan Hardware osiosta luoda sisäisiä Dummy-pisteitä joita voidaan käsitellä tämän valikon-kautta. Harjoituksessa näitä ei käytetä.

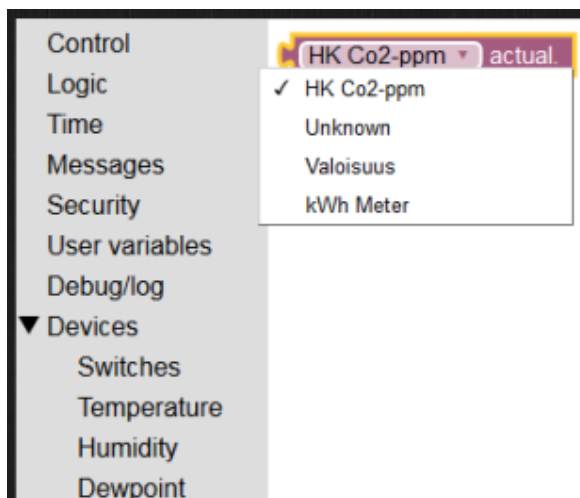
Debug/Log: Ohjelmaan voidaan lisätä ehtoja joiden seurauksena Domoticzin sisäiseen logiin kirjoitetaan viesti.



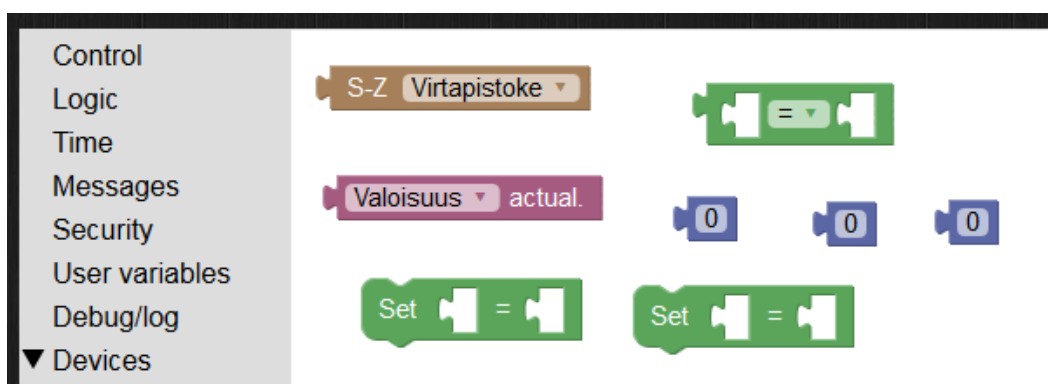
Kirjoitetaan Blocklylla yksinkertainen ohjelma, joka ohjaa Z-Wave etäpistoketta valaistuksen mukaan. Esimerkillä voidaan kotona varmistaa, ettei kahvinkeitin jää päälle, kun valot suljetaan asunnosta poistuessa.

Elementeistä saadaan kuvaukset, kun hiirtä kellutetaan sen päällä hetken aikaa.

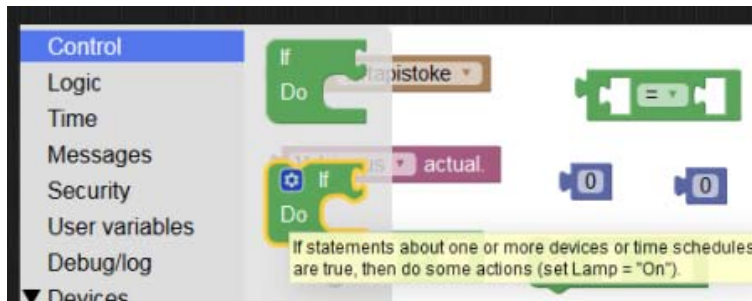
Aloitetaan ohjelmointi-ikkunan kohdasta "Devices -> Utility". Valitse ylin elementti painamalla sitä. Elementti aktivoituu automaattisesti ohjelmointipöydälle. Muuta elementin alavetovalikosta mittapiste, josta saadaan Valoisuus-tieto (LUX). Mittapisteen nimi voi olla kuvasta poikkeava. Nimi on se, jonka sille Domoticzin laiteasetuksissa annoit.



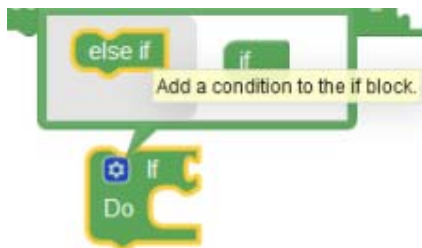
Etsi seuraavan kuvan mukaiset elementit valikoista:



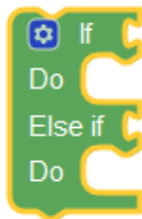
Kun olet löytänyt tarvittavat elementit pitää niille asettaa ehtoja. Valitse Control-valikosta kuvan mukainen elementti.



Paina ohjelmointi pöydällä elementin ratasta:



Siirrä "Else If" elementti oikean puoleiseen tilaan "if"-ehdon alle. Onnistunut asetus lisää ohjelmointipöydälle kuvan mukaiset ehdot.

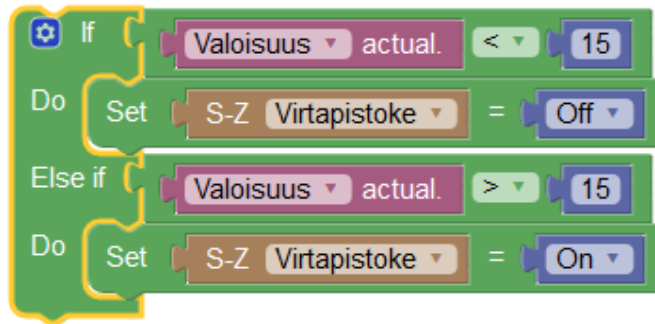


Paina tämän jälkeen uudestaan rattaasta ja asetus-valikko sulkeutuu.

Tallenna ohjelma tässä välissä oikeanlaidan save-napista.

Mieti miten voisit järjestää pöydällä olevat elementit niin, että "Virtapistoke" sulkeutuu, kun valaistus laskee tietyn rajan alle ja kytkeytyy taikaisin päälle kun valaistus lisääntyy tietyn rajan yli. Seuraavalla sivulla on valmis asettelu, jos tarvitset apua. Tehävä myös jatkuu tämän jälkeen.

Oikea elementtiasettelu annettuun tehtävään:



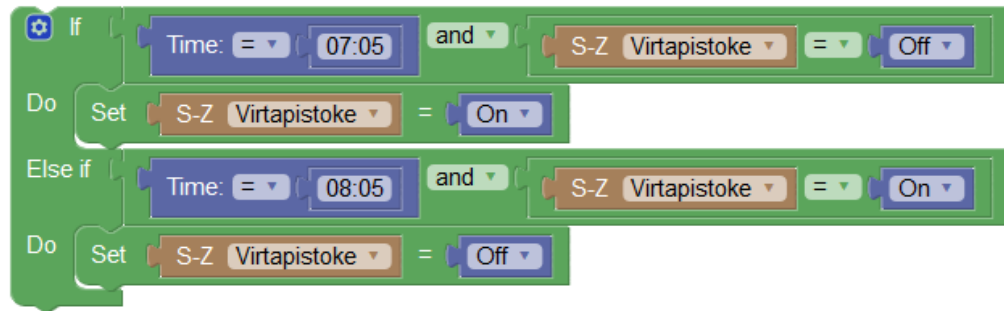
KahvinkeitTIMEN naksuttelu päälle ja pois pelkän valaistuksen mukaan voi aiheuttaa ongelmatilanteita. Lisätään ohjelmaan ehto, jolla keittimen käyttö aikaa rajataan.

Muokkaa ohjelmaa niin, että saat aamukahvit valmiiksi ja keitin sammuu esimerkiksi yhden tunnin päästä aloituksesta. Tähän on useampi oikea vaihtoehto. Malli toteutus on seuraavalla sivulla. Mallissa on lisäehtoina kytkimen asennontarkistus. Jos olet jo laittanut keittimen ennen aikarajaa päälle, ei ohjelma yritä sitä turhaan kytkeä toistamiseen päälle.

Voit testata toimiiko ohjelma säätämällä aikarajoja nykyhetkeen minuutin päähän ja katsomalla vaihtaako pistoke tilaansa halutusti.

Mitä muita ehtoja olisi hyvä käyttää tällaisessa tapauksessa?

Aikarajattu tapahtumaesimerkki: Aamukahvit.



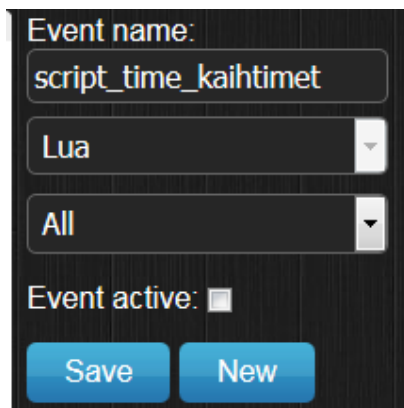
Voit halutessasi ohjelmoida muita Blockly-tapahtumia, lopettaa harjoituksen hyväksytysti tai jatkaa muiden ohjelmointikielien esimerkkeihin.

Liite 11. Domoticz-käyttöliittymä: LUA-tapahtumaohjelmointi

LUA-ohjelmointi muistuttaa perinteistä ohjelmointia. LUA-scripteja voidaan kuitenkin käyttää automaattisesti. Scriptit keskustelevat aktiivisesti Domoticzin tietokannan kanssa ja scriptit suoritetaan jos määritellyt ehdot toteutuvat. Scriptit voidaan jätellä toimimaan 5min tai esim. useiden päivien syklillä jolloin tarpeettomia tietokanta kyselyitä ei suoriteta.

Harjoituksessa käydään esimerkki ohjelman luomisen kautta läpi eri elementtejä, joilla scriptejä pystytään tekemään.

Avataan Domoticzin ”setup-more options-events”-valikko ja asetetaan oikealta laidalta käytettävä ohjelmointikieli ja ohjelman nimeksi ”script_time_kaihtimet”. Jätetään Event active tyhjäksi, jolloin keskeneräistä ohjelmaa ei yritetä automaattisesti ajaa.



The screenshot shows a configuration window for an event in Domoticz. It features a text input field for the event name containing 'script_time_kaihtimet', a dropdown menu for the script language set to 'Lua', another dropdown menu for the event type set to 'All', and an unchecked checkbox for 'Event active'. At the bottom, there are two blue buttons labeled 'Save' and 'New'.

Lua-valinta avaa ohjelmointi-ikkunaan valmiin Domoticzin ohjeistuksen.



```

1  --
2  -- Domoticz passes information to scripts through a number of global tables
3  --
4  -- otherdevices, otherdevices_lastupdate and otherdevices_svalues are arrays for all devices:
5  -- otherdevices['yourotherdevicename'] = "On"
6  -- otherdevices_lastupdate['yourotherdevicename'] = "2015-12-27 14:26:40"
7  -- otherdevices_svalues['yourotherthermometer'] = string of svalues
8  --
9  -- uservariables and uservariables_lastupdate are arrays for all user variables:
10 -- uservariables['yourvariablename'] = "Test Value"
11 -- uservariables_lastupdate['yourvariablename'] = "2015-12-27 11:19:22"
12 --
13 -- other useful details are contained in the timeofday table
14 -- timeofday['Nighttime'] = true or false
15 -- timeofday['SunriseInMinutes'] = number
16 -- timeofday['Daytime'] = true or false
17 -- timeofday['SunsetInMinutes'] = number
18 -- globalvariables['Security'] = 'Disarmed', 'Armed Home' or 'Armed Away'
19 --
20 -- To see examples of commands see: http://www.domoticz.com/wiki/LUA\_commands#General
21 -- To get a list of available values see: http://www.domoticz.com/wiki/LUA\_commands#function\_to\_dump
22 --
23 -- Based on your logic, fill the commandArray with device commands. Device name is case sensitive.
24 --
25 commandArray = {}
26

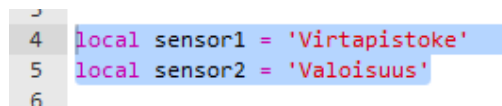
```

Tyhjennä ohjelmointikenttä ja aloitetaan tyhjältä pöydältä.

Määritellään ohjelmassa käytettäviä paikallismuuttujia.

```
local sensor1 = 'Virtapistoke'
```

```
local sensor2 = 'Valoisuus'
```



```

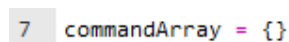
3
4 local sensor1 = 'Virtapistoke'
5 local sensor2 = 'Valoisuus'
6

```

Tehtävässä käytetään valomäärämittausta ja virtapistokkeen kytkintä. Muuttujat nimetään näiden laitepisteiden mukaan tarkista asettamasi nimet Domoticzin laiteasetuksista.

LUA-koodi aloitetaan:

```
commandArray = {}
```



```

7 commandArray = {}

```

Määritellään commandArray = {} alle seuraavat vakiot:

ReportHour=08

ReportMinute=00

ReportHour_2=20

ReportMinute_2=00

```
9 ReportHour=08
10 ReportMinute=00
11
12 ReportHour_2=20
13 ReportMinute_2=00
14
```

Näillä vakiolla hallitaan ohjelman käyttöä vertaamalla käyttöliittymän kellonaikaa annettuihin muuttujiin.

Kirjoitetaan aika muuttuja, jonka arvo haetaan käyttöjärjestelmästä:

```
time = os.date("*t")
```

```
time = os.date("*t")
```

Määritellään seuraavaksi aikaehto, jonka sisällä suoritetaan toiminto:

```
if time.hour == ReportHour and time.min == ReportMinute then
```

```
    if (otherdevices[sensor2] > '20') then
```

```
        commandArray[sensor1] = 'On'
```

```
    end
```

```
end
```

```
time = os.date("!*t")
if time.hour == ReportHour and time.min == ReportMinute then

    if (otherdevices[sensor2] > '20') then

        commandArray[sensor1] = 'On'

    end

end
```

Tässä verrataan käyttöjärjestelmän ilmoittamaa aikaa aikaisemmin määriteltyihin ehtoihin "08" ja "00". Tapahtuma aloitetaan jos järjestelmän ilmoittama aika vastaa klo 08.00.

Toiminnoille on myös annettu lisäehto liittyen anturiin "sensor2".

Komento `otherdevices[sensor2]` hake Domoticzin tietokannasta Valoisuus-mittauksen arvon. Sitä verrataan annettuun ehtoon "on pienempi kuin 20". Seuraava komento suoritetaan vain jos molemmat ehdot täyttyvät.

Komennolla `commandArray[sensor1]` käsketään virtapistoke päälle.

"End" lopettaa kunkin vertailun.

Esimerkillä virtapistoke on päällä ikuisesti. Määritellään seuraavaksi sille sulkemiskomento.

Kopioi ja liitä koodi jatkoksi ja muokkaa se alla olevan mukaisesti.

```

if time.hour == ReportHour_2 and time.min == ReportMinute_2 then
    if (otherdevices[sensor1] == 'On' and otherdevices[sensor2] < '10') then
        commandArray['Virtapistoke'] = 'Off'
    end
end

return commandArray

```

Kokonaisuudessaan ilman muuttujamäärittelyitä koodin tulisi nyt näyttää tältä:

```

time = os.date("!*t")
if time.hour == ReportHour and time.min == ReportMinute then
    if (otherdevices[sensor2] > '20') then
        commandArray[sensor1] = 'On'
    end
end
if time.hour == ReportHour_2 and time.min == ReportMinute_2 then
    if (otherdevices[sensor1] == 'On' and otherdevices[sensor2] < '10') then
        commandArray['Virtapistoke'] = 'Off'
    end
end
return commandArray

```

Huomaa lisätä koodin loppuun "return commandArray"

Millaisia ehtoja sulkemiselle on annettu?

Mitä eroa merkinnöillä "==" ja "=" on?

Miten laitteita voidaan käsitellä LUA-scriptissä?

Mitä muita käyttöjä tällaiselle ohjelmalle voisi tehdä saatavilla olevista laitteista?

Voit testata koodin toiminnollisuutta käytännössä tekemällä sen aktiiviseksi oikeanlaidan Event active-valinnalla ja tallentamalla koodin Save-napilla. Vaihda kuitenkin aikavertailuun käytettyjä muuttujia nykyhetkeen sopiviksi. Esimerkissä käytetään hiilioksidimittausta, joten voit puhaltaa hetken kevyesti mittapaneelin akkoihin. Hiilioksidipitoisuus tasaantuu huoneen yleistasolle, kun sen jättää hetkeksi rauhaan.

Lisää LUA-scripteihin voi tutustua Domoticzin wiki-sivulla. Voit halutessasi kokeilla muokata ohjelmaa tai lopettaa harjoituksen.

Liite 12. Arduino-Modbus-silta: ohjelmakoodi kokonaisuudessaan

Alla esitelty ohjelmakoodi on lopullinen harjoituksissa käytettävä esimerkkikoodi.

```
*/  
  
// Mysensors määritelmät: Voidaan ottaa käyttöön ja pois lisäämällä/poistamalla "//"  
rivin edestä.  
  
// Enable debug prints to serial monitor  
  
// Vianselvitys määrittely sarjaporttiin  
  
#define MY_DEBUG  
  
  
// Enable and select radio type attached -- -- valitaan jos käytetään radioyhteyttä.  
  
//#define MY_RADIO_NRF24  
  
//#define MY_RADIO_RFM69  
  
  
// Set LOW transmit power level as default, if you have an amplified NRF-module and  
// power your radio separately with a good regulator you can turn up PA level.  
  
  
// Radiolinkin käyttöteho, kotioiloissa voidaan käyttää matalaa lähetystasoa.  
  
// Pitkillä matkoilla voidaan käyttää erillisä vahvistettua antenniyksikköä ja tehoa  
nostaa.  
  
  
//#define MY_RF24_PA_LEVEL RF24_PA_LOW  
  
  
// Enable serial gateway - - Määritellään käytettäväksi sarjaporttia Raspberry Pi isän-  
nän ja Arduino-Modbus sillan välillä.
```

```
// Viestit lähetetään laitetason sarjaporttiin (Hardware serial, Arduinossa pinnit rX ja tX, sekä USB)
```

```
#define MY_GATEWAY_SERIAL
```

```
// Lisätään ohjelmakoodissa käytettävät kirjastot.
```

```
// Mysensors kirjasto.
```

```
#include <MySensors.h>
```

```
// Ohjelmallisen sarjaportin kirjasto.
```

```
#include <SoftwareSerial.h>
```

```
// ModbusMaster kirjasto.
```

```
#include <ModbusMaster.h>
```

```
/*!
```

Määritellään MySensor kirjastolle käytettäviä muuttujia.

Vakio määritelmä CHILD_ID_Muuttuja -> esim. CHILD_ID_TEMP 1 lämpötila tunnuksella 1.

Muokattu muotoon MBslave*_ID_' jossa "*" on Modbus orjalaitteen osoitetunnus ja "" mittautystyyppi.

```
*/
```

```
#define MBslave1_ID_AIQ 1
```

```
#define MBslave1_ID_TEMP 10
```

```
#define MBslave1_ID_HUM 6
```

```
//#define MBslave8_ID_TEMPIN 1
```

```
//#define MBslave8_ID_HUMIN 11
```

```
#define MBslave8_ID_TEMPOUT 15

#define MBslave8_ID_HUMOUT 22

#define MBslave8_ID_BARO 30

/*!

MAX485-Väyläadapterin käyttämättä pinnit Arduinokortilta. Käytetään adapterin
hallintaan.

*/

#define MAX485_DE 3 // Arduinossa fyysinen pinni D3

#define MAX485_RE_NEG 2 // Arduinossa fyysinen pinni D2

// Määritellään ohjelmallisen sarjaliikenteen käyttämät pinnit. Käytetään Modbus-
väyläadapterin viestintä pinneinä.

#define rxPin 11 // Vastaanotto väylästä, Arduinossa D11

#define txPin 10 // Lähetys väylään, Arduinossa D10

/*!

Määritellään Luokka nimitykset. Näiden avulla ohjelmassa voidaan kutsua tapaus-
kohtaisesti kirjaston sisäisiä komentoja.

MBslave1; Luo objektin jolla voidaan ohjelmassa käsitellä väylän orjalaitetta 1

MBslave8; Luo objektin jolla voidaan ohjelmassa käsitellä väylän orjalaitetta 1

Uudet Modbus anturit listään vastaavasti. Numero kertoo orjalaitteen väyläosoit-
teen.
```

Serial1; Luodaan ohjelmallinen sarjaportti ja sille käytettävät pinnit.

```
*/
```

```
ModbusMaster MBslave1;
```

```
ModbusMaster MBslave8;
```

```
SoftwareSerial Serial1 = SoftwareSerial(rxPin, txPin);
```

```
/*!
```

ModbusMaster kirjaston viestinlähetykseen liittyvä funktio, jota voidaan kutsua ohjelmassa.

Määritellään väyläadapterille väyläänlähetyksi.

```
*/
```

```
void preTransmission()
```

```
{
```

```
digitalWrite(MAX485_RE_NEG, 1);
```

```
digitalWrite(MAX485_DE, 1);
```

```
}
```

```
/*!
```

ModbusMaster kirjaston viestinlähetykseen liittyvä funktio, jota voidaan kutsua ohjelmassa.

Määritellään väyläadapterille väylästä vastaanotto aktiiviseksi.

```
*/
```

```
void postTransmission()
```

```
{
```

```
digitalWrite(MAX485_RE_NEG, 0);
```

```
digitalWrite(MAX485_DE, 0);
```

```

}

/#!

    Määritellään Mysensors kirjaston mukaiset viestirakenteet.

    Kullekin mittaukselle ja orjalaitteelle omansa.

    Sulkujen sisälle määritellään muuttujien tyyppi kullekin mittaukselle kirjaston mu-
kaisesti.

*/

MyMessage msgMBslave1_CO2(MBslave1_ID_AIQ, V_LEVEL);

MyMessage msgMBslave1_CO2prefix(MBslave1_ID_AIQ, V_UNIT_PREFIX);

MyMessage msgMBslave1_Hum(MBslave1_ID_HUM, V_HUM);

MyMessage msgMBslave1_Temp(MBslave1_ID_TEMP, V_TEMP);

// Esimerkkivaraus ohjelmanlaajentamiselle

//MyMessage msgMBslave8_TempIN(MBslave8_ID_TEMPIN, V_TEMP);

MyMessage msgMBslave8_TempOUT(MBslave8_ID_TEMPOUT, V_TEMP);

//MyMessage msgMBslave8_HumIN(MBslave8_ID_HUMIN, V_HUM);

MyMessage msgMBslave8_HumOUT(MBslave8_ID_HUMOUT, V_HUM);

MyMessage msgMBslave8_Baro(MBslave8_ID_BARO, V_PRESSURE);

/#!

    Setup osiossa tehdään ohjelmallisia valmisteluita ennen syklisenohjelman suori-
tusta. Komennot suoritetaan vain kerran.

*/

void setup()

{

```

```
//Voidaan käyttää väylään kirjoittamisen varmennuksena.  
  
//uint8_t resultsetup;  
  
// Tarkennetaan ohjelmallisen sarjaportin pinnien määritelmiä asettamalla toinen  
sisääntuloksi ja toinen lähettäväksi.  
  
pinMode(rxPin, INPUT);  
  
pinMode(txPin, OUTPUT);  
  
// Määritellään ohjelmallisen sarjaportin nopeus Baudeina. Määräytyy Modbus-  
väylä asetusten mukaan.  
  
Serial1.begin(9600);  
  
Serial.begin(115200); // USB_Portti PC:LLE. Nopeus voidaan valita joko 38400 tai  
115200 baud. // Määräytyy Mysensor asetusten mukaan.  
  
// MAX485-väyläadapterille annettujen pinnien muoto, laitetaan molemmissa lähte-  
väksi. Adapteri ei lähetä näihin takaisin mitään.  
  
pinMode(MAX485_RE_NEG, OUTPUT);  
  
pinMode(MAX485_DE, OUTPUT);  
  
// Asettetaan adapterin pinnien tila nolaksi.  
  
digitalWrite(MAX485_RE_NEG, 0);  
  
digitalWrite(MAX485_DE, 0);  
  
// Modbus slave ID, Määritellään väyläviestinnät kullekin orjalaitteelle.
```

```
// Tässä tapauksessa molemmat käyttävät ohjelmallista sarjaporttia 1 aikaisemmin  
määritetyllä nimellä Serial1.
```

```
// Jos Arduino kortilla on useampi portti joissa kussakin oma väylänsä voidaan tässä  
kohtaa yksilöidä käytettävät väylät
```

```
// Esimerkiksi Arduino Megassa on 4kpl laitetason väyliä Serial1,2 ja 3. Serial0 on  
eristetty muista USB-liitäntään.
```

```
MBslave1.begin(1, Serial1); //HK isntruments mittapaneeli
```

```
MBslave8.begin(8, Serial1); //Vantage sääasema-adapteri
```

```
// Callbacks allow us to configure the RS485 transceiver correctly
```

```
// Määritetään viestinlähetykseen käytettäviä komentoja, joita käytetään adapterin  
viestiohjaukseen.
```

```
MBslave1.preTransmission(preTransmission);
```

```
MBslave1.postTransmission(postTransmission);
```

```
MBslave8.preTransmission(preTransmission);
```

```
MBslave8.postTransmission(postTransmission);
```

```
delay(1500); // Tämä viive on tärkeä, jotta kaikki sarjaporttiliikenne saadaan tasaan-  
tumaan ennen ohjelmasykliä.
```

```
/*
```

```
 //Esimerkkivaraus väyläänkirjoittamiseen. Poista "/*" ylemmältä riviltä ja "/*" en-  
nen resultsetup muuttujaa.
```

```
 // Kommentoi kenoviivoilla lähetyksion viimeiseltä riviltä "*/"
```

```
 //HK INSTRUMENT PANEELI OSOITTEESSA 1 ILMANKOSTEUDEN SÄÄTÖRAJAT  
ANALOGISELLE LÄHDÖLLE.
```

```
// Ajetaan kerran käynnistyksessä. onnituneen kirjoituksen jälkeen voidaan kom-  
mentoida uudestaan ja
```

```
// uudelleen ohjelmoida Arduinolle.
```

```
// Vähentää ohjelman muistinkäyttöä ja vaikuttaa toimintaa!!!
```

```
resultsetup = MBslave1.writeSingleRegister(0, lowWord(1)); // HK Instruments  
4x0001, P-parametri, 0=CO2, 1= rH%, 2=TE
```

```
if (resultsetup == MBslave1.ku8MBSuccess)
```

```
{
```

```
Serial.println("Result write ok:");
```

```
Serial.println(resultsetup);
```

```
}
```

```
else if (resultsetup != MBslave1.ku8MBSuccess)
```

```
{
```

```
Serial.println("Result write fail:");
```

```
Serial.println(resultsetup);
```

```
}
```

```
delay(1500);
```

```
resultsetup = MBslave1.writeSingleRegister(3, lowWord(1000)); // HK  
Instruments 4x0004, rH% high limit 10-100% (100-1000)
```

```
if (resultsetup == MBslave1.ku8MBSuccess)
```

```
{
```

```
Serial.println("Result write ok:");
```

```
Serial.println(resultsetup);
```



```

}

else if (resultsetup != MBslave1.ku8MBSuccess)

{

Serial.println("Result write fail:");

Serial.println(resultsetup);

}

delay(1500);

    resultsetup = MBslave1.writeSingleRegister(4, lowWord(0)); // HK Instruments
4x0005, rH% low limit 0-90% (0-900)

if (resultsetup == MBslave1.ku8MBSuccess)

{

Serial.println("Result write ok:");

Serial.println(resultsetup);

}

else if (resultsetup != MBslave1.ku8MBSuccess)

{

Serial.println("Result write fail:");

Serial.println(resultsetup);

}

    delay(1500);

*/

/#!

```

MySensors kirjasto luo määritetyistä mittauksista virtuaaliset laitteet, jotka esitel-
lään käyttöliittymälle

sen tunnistamassa muodossa. Näin tehtäessä jokaisen laitteen mittaukselle ei tarvise käsin koodata

tunnistusformaattia vaan kirjasto luo kaikki automaattisesti.

```
*/  
}  
void presentation()  
{  
    // Present locally attached sensors  
    // Lähetetään käyttöliittymälle tietoa sovelluksesta  
    sendSketchInfo("DOMOTICZ Harjoitus, RakLab", "1.0");  
  
    // Register all sensors to gateway (they will be created as child devices)  
    // Määritellään ja rekisteröidään annetut mittapisteet käyttöliittymään ja isäntäohjaimen.  
    // Tässä tapauksessa Arduino toimii sekä mittapisteenä(node) ja isäntänä(controller) samanaikaisesti.  
    // Yksittäiset mittapisteen mittaukset (child).  
    // Perinteisessä Mysensors sovelluksessa mittapisteenä ja isäntänä toimisi erillinen Arduino-ohjain.  
    // Isäntä pitää muistissa seuraavat määritelmä jolloin mittapisteen asetukset voidaan palauttaa  
    // isännältä esimerkiksi sähkökatkon jälkeen. Tässä kohtaa S_* määrittää mittalaitteen tyyppin käyttöliittymälle.  
    // V_* määrittelee mitattavan suureen tyyppin. S_TEMP käyttää muuttujaa V_TEMP jne. Ilmoituksessa voidaan myös
```

```
// lähettää isännälle ja käyttöliittymälle tietoja mittalaitteesta/pisteestä lisäämällä  
määrityksen perään , "viesti"
```

```
present(MBslave1_ID_HUM, S_HUM, "HK Kosteus %");
```

```
present(MBslave1_ID_AIQ, S_AIR_QUALITY, "HK Co2-ppm");
```

```
present(MBslave1_ID_TEMP, S_TEMP, "HK temperature");
```

```
present(MBslave8_ID_TEMPOUT, S_TEMP, "Vantage Sääasema Ulko");
```

```
//present(MBslave8_ID_TEMPIN, S_TEMP, "Vantage Sääasema Sisä");
```

```
//present(MBslave8_ID_HUMIN, S_HUM, "Vantage rH% sisä");
```

```
present(MBslave8_ID_BARO, S_BARO, "Vantage Barometri hPa");
```

```
present(MBslave8_ID_HUMOUT, S_HUM, "Vantage rH% ulko");
```

```
}
```

```
// Pääohjelman käynnistys.
```

```
void loop()
```

```
{
```

```
    uint8_t result; // Modbus orjalaite 1:n viestin välitysvarmistus summa, käytetään  
    varmistamaan viestin oikeellisuus
```

```
    // Kirjasto muodostaa viesteistä automaattisesti tarkistussumman ja tämä tallenne-  
    taan yllämainittuihin muuttujiin.
```

```
    // Samalla Väylään lähetetään luku kyselyt komennoilla *.readInputRegisters(mo-  
    nesko rekisteri, monta siitä eteenpäin)
```

```
    // Rekisteri määritelmät löytyvät laitevalmistajan ohjeistuksista.
```

```
// Vastaukset talletetaan ModbusMasterin puskurimuistiin(".*Responsebuffer(paikka)") . myöhempää käyttöä varten.

delay(2000); // Kuten yllä tärkeä tasausviive. Syytä käyttää aina ennen väylään lukua tai kirjoitusta.

// Luetaan HK instrument mittapaneelin Input rekisteri paikasta 0 seuraavat 6 eteenpäin.

result = MBSlave1.readInputRegisters(0, 6);

// Seuraavalla ehdolla määritellään tehtäväksi kometoja jos orjalaitteen MBSlave1 viestintäsumma on hyväksytty.

// Kirjaston tausta funktiot palauttavat 0 kun kaikki onnistuu.

if (result == MBSlave1.ku8MBSuccess)

{

    Serial.println("ModbusSlave 1");

    // lähetetään MySensors kirjaston mukaisella rakenteella mittaustietoja isäntälaitteelle ja sen kautta käyttöliittymälle.

    send(msgMBSlave1_Temp.set(MBSlave1.getResponseBuffer(0x03) / 10.0f, 2));

    // "set(float value, uint8_t decimals)"; rakenne jolla kirjasto olettaa saavansa float tyyppisen muuttujan.

    // tärkein viimeinen decimaalien määrä ",2", /10.0f tällä mittatieto jaetaan 10 kahden desimaalin tarkuuteen.

    // Laitekohtaiset ohjeistukset aina valmistajien ohjeistuksista."
```

```
// HK:n paneeli antaa lämpötilan välillä 0-500 jolloin mittaus pitää jakaa 10:llä jolloin esimerkiksi lämpötilan
```

```
// viesti arvo 245 lähtee käyttöliittymään muodossa 24.50.
```

```
send(msgMBslave1_CO2.set(MBslave1.getResponseBuffer(0x01)));
```

```
// Hiilidioksidimittaus on välillä 0-2000 muodossa 0-2000ppm,
```

```
// mittarvo on kokonaisluku eikä tarvitse erillisiä muuntoja.
```

```
send(msgMBslave1_Hum.set(MBslave1.getResponseBuffer(0x02) / 10.0f, 2));
```

```
//set(float value, uint8_t decimals); vrt. Lämpötilanmittaus ylempänä.
```

```
}
```

```
delay(2000);
```

```
result = MBslave8.readHoldingRegisters(3, 30);
```

```
if (result == MBslave8.ku8MBSuccess)
```

```
{
```

```
Serial.println("ModbusSlave 8");
```

```
send(msgMBslave8_Baro.set(MBslave8.getResponseBuffer(0) * 0.1f, 2));
```

```
//send(msgMBslave8_TempIN.set(MBslave8.getResponseBuffer(1) * 0.1f, 2));
```

```
//send(msgMBslave8_HumIN.set(MBslave8.getResponseBuffer(2), 2));
```

```
send(msgMBslave8_TempOUT.set(MBslave8.getResponseBuffer(3) * 0.1f, 2));
```

```
send(msgMBslave8_HumOUT.set(MBslave8.getResponseBuffer(15), 2));
```

```
}
```

```
delay(1000); // loppuviive auttaa tasaamaan kaiken uutta sykliä varten.
```

```
}
```