

# Home Assistantin käyttö Raspberry Pi:llä

Case: Älykodit



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus  
syksy, 2023

Nuutti Jussila

Tietojenkäsittelyn koulutus

Tiivistelmä

Tekijä Nuutti Jussila

Vuosi 2023

Työn nimi Home Assistantin käyttö Raspberry Pi:llä  
Case: Älykoodit

Ohjaaja Ismo Turve

---

## TIIVISTELMÄ

Tässä toiminnallisessa opinnäytetyössä perehdyttiin esineiden internetiin, älykoteihin ja niissä yleisimmin käytössä oleviin protokolleihin. Lisäksi kerrottiin yleisesti älykotien ja valittujen protokollien tietoturva. Tutkimusaineistoa kerättiin kirjallisuudesta ja internetlähteistä. Paikallinen kahden älylaitteen älykotiratkaisu toteutettiin asentamalla Home Assistant -ohjelmisto Raspberry Pi -minitietokoneeseen. Älylaitteena käytettiin Shelly-älypistorasiaa ja WiZ-älylamppua, sekä toteutettiin niillä neljä eri automaatiota. Shellyn valinnan taustalla oli sen mahdollistavat energiasäästöt. WiZ:in avulla demonstroitiin YAML-merkintäkieltä.

Opinnäytetyön teoriaosuudessa käytiin läpi esineiden internetiä käsitteenä. Seuraavaksi avattiin älykoteja, niiden ohjauskeskuksia, sekä perehdyttiin syvällisemmin kolmeen kirjoitushetkellä älykodeissa yleisimmin käytössä oleviin protokolleihin. Tämän jälkeen käytiin läpi ZigBee-, MQTT- ja CoAP-protokollien toimintaa ja tietoturvaa. Lopuksi esiteltiin toteutuksessa käytetyt laitteet ja ohjelmistot.

Tutkimuksessa havaittiin esineiden internetin olevan käsitteenä hyvin laaja, joten aiheen rajaus oli haastavaa. Oma älykotiratkaisu saatiin toteutettua ja automaatioita olisi voinut helposti tehdä enemmänkin. Alustan käyttöön perehdyttyä huomattiin Home Assistantin mahdollistavan kattavan ja helposti muokattavissa olevan älykotiratkaisun suhteellisen pienellä vaivalla. Monimutkaisemmat automaatiot vaativat YAML-merkintäkielen opettelua.

Avainsanat Esineiden internet, Home Assistant, Raspberry Pi, kotiautomaatio, älykoti

Sivut 85 sivua ja liitteitä 4 sivua

Degree Programme in Business Information Technology      Abstract  
Author      Nuutti Jussila      Year 2023  
Subject      Using Home Assistant with Raspberry Pi  
                 Case: Smart homes  
Supervisor      Ismo Turve

---

## ABSTRACT

This practical thesis explored the Internet of Things, smart homes, the commonly used protocols in them and the security issues surrounding these all. The research data was gathered from literature and internet sources. Two device smart home solution was accomplished by installing the Home Assistant home automation software to a Raspberry Pi. The utilized smart devices were a Shelly smart plug and a WiZ smart bulb. They were used to create four different automations. The smart plug was chosen for its possibilities in lowering energy costs. The smart bulb is used to demonstrate the YAML markup language.

The research part of this thesis focused on the Internet of Things as a concept. Smart homes, its control hubs, and at the time of writing, three commonly used protocols were examined. The mechanics and security of ZigBee, MQTT and CoAP protocols were explained. The protocols were chosen for their compatibility with Home Assistant. Lastly, the devices and software used in the solution were showcased.

The Internet of Things as a concept, is vast. It was challenging to limit the writing topics. The smart home solution was achieved, and more automations could have easily been added. After becoming familiar with Home Assistant, it was clear that it can provide a comprehensible, low-effort, and easily customizable smart home solution. For more complicated automations knowledge of the YAML markup language is required.

Keywords      Home Assistant, Internet of Things, Raspberry Pi, home automation, smart home  
Pages      85 pages and appendices 4 pages

## Sanasto

A/D-muunnin	Analog-to-Digital converter. Elektroniikan laite, joka muuntaa jatkuvan analogisen signaalin arvoja digitaalisiksi lukuarvoiksi.
AES	Advanced Encryption Standard. Tietotekniikassa käytettävä lohkosalausmenetelmä.
API	Application Programming Interface. Ohjelmointirajapinta.
ARPANET	Advanced Research Projects Agency Network. 1960-luvulla suunniteltu Yhdysvaltalainen pakettikytkentäinen tietokoneverkko, josta kehittyi Internet.
DNS	Domain Name System. Internetin nimipalvelujärjestelmä, joka muuntaa verkkotunnuksia IP-osoitteiksi.
DTLS	Datagram Transport Layer Security. Tietoliikenneprotokolla, joka tarjoaa suojan UDP-sovelluksille.
Docker	Avoimen lähdekoodin virtualisointialusta, jonka avulla ohjelmakoodi ja sen tarvitsemat kirjastot voidaan paketoita virtualisoituun konttiin.
GPS	Global Positioning System. Yhdysvaltain puolustusministeriön kehittämä ja rahoittama satelliittipaikannusjärjestelmä.
HTTP	Hypertext Transfer Protocol. Selaimien ja WWW-palvelimien käyttämä protokolla tiedonsiirtoon.
I/O-moduuli	Input/Output module. Laite, jolla ohjelmoitava logiikka liitetään muihin laitteisiin, erityisesti automaatiojärjestelmissä käytettäviin mittalaitteisiin.
IP	Internet Protocol. Protokolla, joka huolehtii IP-pakettien toimittamisesta perille pakettikytkentäisessä tietokoneverkossa.
IoT	Internet of Things. Esineiden internet.
JSON	JavaScript Object Notation. Ihmisille helposti luettava ja kirjoitettava tiedostomuoto tiedonvälitykseen ja tallennukseen.
OSI	Open Systems Interconnection Reference Model. Malli, joka kuvaa tiedonsiirtoprotokollien yhdistelmän seitsemässä kerroksessa.
PAN	Personal Area Network. Lyhyen kantaman verkkoratkaisu.
RFID	Radio Frequency Identification. Radiotaajuinen etätunnistus.

TCP	Transmission Control Protocol. Tietoliikenneprotokolla tietokoneiden väliseen luotettavaan tiedonsiirtoon.
TCP/IP	TCP/IP-viitemalli. Internetin arkkitehtuurin kuvaamisessa käytetty tietoliikenneverkkojen viitemalli.
TLS/SSL	Transport Layer Security / Secure Socket Layer. Salausprotokolla, jolla voidaan suojata Internet-sovellusten tietoliikenne IP-verkkojen yli.
UDP	User Datagram Protocol. Ns. yhteydetön tietoliikenneprotokolla, joka ei vaadi yhteyttä laitteiden välille, mutta mahdollistaa tiedon siirron.
UTF-8	Muuttuvapituinen merkkikoodausstandardi, jota käytetään sähköisessä viestinnässä.
WAN	Wide Area Network. Pitkän kantaman verkkoratkaisu.
YAML	Yet Another Markup Language. Merkintäkieli, jota käytetään usein konfiguraatiodostoissa, mutta sitä voi käyttää myös muunlaiseen tietojen talletukseen.

## Sisällys

1	Johdanto .....	1
2	Esineiden internet (IoT) .....	2
2.1	Lyhyt historia .....	2
2.2	Arkkitehtuuri .....	3
2.3	Sensorit ja laitteet .....	5
3	Älykoodit .....	7
3.1	Ohjauskeskukset .....	7
3.2	Tietoturva .....	8
4	Laitteiden kommunikaatio älykodeissa .....	10
4.1	ZigBee .....	11
4.1.1	Toiminta .....	11
4.1.2	Laitteiden yhteensopivuus .....	14
4.1.3	Tietoturva .....	14
4.2	MQTT (Message Queuing Telemetry Transport) .....	16
4.2.1	Toiminta .....	16
4.2.2	Yhteyden muodostaminen .....	17
4.2.3	Julkaiseminen .....	18
4.2.4	Tilaaminen ja jokerimerkit .....	19
4.2.5	Palvelun laatutaso .....	20
4.2.6	Tietoturva .....	22
4.3	CoAP (Constrained Application Protocol) .....	23
4.3.1	Toiminta .....	23
4.3.2	Viestit .....	24
4.3.3	Tietoturva .....	26
5	Käytetyt ohjelmistot ja laitteet .....	27
5.1	Home Assistant .....	27
5.2	Raspberry Pi 3 Model B+ .....	27
5.3	Shelly Smart Plug .....	28
5.4	WiZ LED A60 full color -lamppu .....	29
6	Älykodin rakentaminen .....	30
6.1	Home Assistantin asennus .....	30
6.1.1	Balena Etcher .....	30
6.1.2	Valmistelu .....	33

6.2	Älylaitteiden lisääminen.....	35
6.2.1	Shelly Smart Plug.....	35
6.2.2	Android puhelin.....	41
6.2.1	Airam Wi-Fi Smart Bulb.....	45
6.2.2	WiZ-älylamppu.....	45
6.3	Automaatioiden luonti.....	52
6.3.1	Shelly pois päältä kodista lähtiessä.....	52
6.3.2	Shelly pesukoneen tilailmoitukset puhelimeen.....	56
6.3.3	WiZ tilan tallentaja.....	63
6.3.4	WiZ välähdys WhatsApp-viestin saapuessa.....	67
7	Johtopäätökset ja pohdinta.....	78
8	Yhteenveto.....	80
	Lähteet.....	81

## Kuvat ja YAML-merkintäkoodit

Kuva 1.	Nelikerroksinen IoT-järjestelmän arkkitehtuuri (altexsoft, 2022).....	4
Kuva 2.	IoT-laitteen rakennusosia (Premier Farnell Ltd, 2017).....	6
Kuva 3.	Protokollat ja protokollapinot OSI- ja TCP/IP-mallin eri kerroksissa (Guinard & Trifa, 2016, s. 114). .....	10
Kuva 4.	Laitteen internet yhteys gatewayn avulla (Guinard & Trifa, 2016, s. 129). .....	11
Kuva 5.	ZigBee mesh-verkko (STMicroelectronics, 2022). .....	12
Kuva 6.	Sovelluksen linkitysavain luottamuskeskukselta (Silicon Laboratories, 2022, s. 8). .....	15
Kuva 7.	MQTT julkaisija-tilaaja-malli (MQTT, 2022). .....	16
Kuva 8.	Eri laatutasojen kommunikaatio (Zhou, 2023). .....	21
Kuva 9.	Julkaisun laatutason alentaminen tilauksen laatutasolle (Zhou, 2023). .....	22
Kuva 10.	CoAP asiakas-palvelin-malli (Beshkov, 2023). .....	24
Kuva 11.	CoAP-pyyntöihin vastaaminen (Azzola, 2018).....	26
Kuva 12.	Raspberry Pi 3 Model B+ (Raspberry Pi, ei pvm.).....	28
Kuva 13.	Shelly Smart Wi-Fi-älypistorasia (Shelly, ei pvm.-b). .....	29
Kuva 14.	WiZ LED A60 full color -lamppu (WiZ, ei pvm.).....	29
Kuva 15.	Balena Etcher ohjelmiston asetukset. ....	31

Kuva 16. Balena Etcher virhe linkin kautta kirjoittamisessa.....	31
Kuva 17. Balena Etcher onnistunut kirjoittaminen. ....	32
Kuva 18. Raspberry Pi mikroSD -kortilla kytkettynä virtaan ja reitittimeen. ....	32
Kuva 19. Home Assistantin valmistelu selaimessa. ....	33
Kuva 20. Home Assistantin alkunäkymä asennuksen jälkeen. ....	34
Kuva 21. File Editor -lisäosa asennettuna.....	34
Kuva 22. Shelly tukiasemana puhelimen Wi-Fi-asetuksissa. ....	35
Kuva 23. Shellyn lisääminen kotiverkkoon käyttöliittymästä.....	36
Kuva 24. Shelly kirjautumisvaatimuksen asettaminen käyttöliittymästä. ....	36
Kuva 25. Shelly ColoT-asetus. ....	37
Kuva 26. Shelly integraatio tunnistettuna HA:ssa. ....	38
Kuva 27. Shellyn lisäämisen kysely. ....	38
Kuva 28. Shelly lisätty onnistuneesti. ....	38
Kuva 29. Shellyn uudelleenintegroinnin kirjautumistiedot. ....	39
Kuva 30. Shellyn uudelleenintegrointi onnistuneesti.....	39
Kuva 31. Shelly ”Laitteet & palvelut” -listalla.....	40
Kuva 32. Shellyn uudelleen nimeäminen. ....	41
Kuva 33. HA-instanssin yhdistäminen sovellukseen.....	41
Kuva 34. HA-instanssiin kirjautuminen sovelluksessa. ....	42
Kuva 35. HA-instanssin kaksivaiheinen tunnistautuminen sovelluksessa.....	42
Kuva 36. Laitteen nimeäminen ja seurannan salliminen HA sovelluksessa. ....	43
Kuva 37. HA-instanssin ilmoitusten päälle kytkeminen sovelluksessa. ....	44
Kuva 38. HA-instanssin koontinäyttö sovelluksessa.....	44
Kuva 39. WiZ Connected -sovellus ensiasennuksen jälkeinen näkymä. ....	45
Kuva 40. WiZ Connected -sovellus löydetty laite. ....	46
Kuva 41. WiZ Connected -sovellus löydetyn laitteen tyyppi. ....	46
Kuva 42. WiZ Connected -sovellus laitteen huonevalinta. ....	47
Kuva 43. WiZ Connected -sovellus kotiverkon tiedot. ....	47
Kuva 44. WiZ Connected -sovellus laitteen onnistunut lisäys kotiverkkoon.....	48
Kuva 45. WiZ Connected -sovellus lisätty laite lopputulos.....	49
Kuva 46. HA WiZ-lampun integraation lisäys. ....	50
Kuva 47. HA WiZ-lampun IP-osoitteen automaattinen etsintä.....	50



Kuva 48. HA WiZ-lamppu löydetty. ....	51
Kuva 49. HA WiZ-lamppu onnistunut integraatio huonevalinta. ....	51
Kuva 50. HA WiZ-lampun uudelleennimeäminen. ....	52
Kuva 51. Kodin paikallaoloa seuraava integraatio. ....	53
Kuva 52. HA-instanssin uudelleenkäynnistys. ....	53
Kuva 53. Shellyn automaattisammutuksen laukaisuehto. ....	54
Kuva 54. Shellyn automaattisammutuksen toiminto. ....	54
Kuva 55. Shellyn automaattisammutuksen tallennus. ....	55
Kuva 56. Shellyn automaattisammutus automaatiolistassa. ....	55
Kuva 57. Shellyn automaattisammutuksen jäljityspainike. ....	56
Kuva 58. Shellyn automaattisammutuksen jäljitysnäkymä. ....	56
Kuva 59. Shellyn pesukoneautomaation energiankulutusarvot. ....	57
Kuva 60. Shellyn pesukoneautomaation avustaja. ....	58
Kuva 61. Shellyn pesukoneautomaation laukaisuehto. ....	59
Kuva 62. Shellyn pesukoneautomaation ehto. ....	60
Kuva 63. Shellyn pesukoneautomaation toiminnot. ....	61
Kuva 64. Shellyn pesukoneautomaation syklin loppuminen. ....	62
Kuva 65. Shellyn pesukoneautomaation pesukoneen pois päältä kytkeminen. ....	62
Kuva 66. Shellyn pesukoneautomaation puhelinilmoitukset. ....	63
Kuva 67. WiZ-lampun tilatallentaja-automaation laukaisuehdot. ....	64
Kuva 68. WiZ-lampun tilatallentaja-automaation ehdot. ....	64
Kuva 69. WiZ-lampun tilatallentaja-automaation toiminto. ....	65
Kuva 70. WiZ-lampun tilanpalauttaja-automaation laukaisuehto, ....	65
Kuva 71. WiZ-lampun tilanpalauttaja-automaation toiminto. ....	66
Kuva 72. WiZ-lampun tilatallentaja- ja tilanpalauttaja-automaatiot listattuna. ....	66
Kuva 73. WiZ-lampun WhatsApp-automaation ilmoitusten sallittujen lista. ....	67
Kuva 74. WiZ-lampun WhatsApp-automaation ilmoitusanturi päälle kytkettynä. ....	68
Kuva 75. WiZ-lampun WhatsApp-automaation laukaisusanan avustaja. ....	69
Kuva 76. WiZ-lampun WhatsApp-automaation laukaisusanan asetus. ....	69
Kuva 77. WiZ-lampun WhatsApp-automaation tiedostonimien asetus. ....	70
Kuva 78. WiZ-lampun WhatsApp-automaation "sensor.yaml" -tiedoston luonti. ....	70
Kuva 79. WiZ-lampun WhatsApp-automaation viestin tallennusintegraatio. ....	71

Kuva 80. WiZ-lampun WhatsApp-automaation laukaisusanan tarkastajaintegraatio....	71
Kuva 81. WiZ-lampun WhatsApp-automaatioon luodut integraatiot listattuna. ....	72
Kuva 82. WiZ-lampun WhatsApp-automaation laukaisuehto.....	72
Kuva 83. WiZ-lampun WhatsApp-automaation ehdot.....	73
Kuva 84. WiZ-lampun WhatsApp-automaation ensimmäinen toiminto.....	74
Kuva 85. WiZ-lampun WhatsApp-automaation toinen-toiminto.....	74
Kuva 86. WiZ-lampun WhatsApp-automaation "Repeat" -toiminnon ensimmäinen toiminto. .....	75
Kuva 87. WiZ-lampun WhatsApp-automaation "Repeat" -toiminnon toinen toiminto.	75
Kuva 88. WiZ-lampun WhatsApp-automaation kolmas toiminto. ....	76
Kuva 89. WiZ-lampun WhatsApp-automaation tallennus. ....	76
Kuva 90. WiZ-lampun WhatsApp-automaatio automaatiolistassa. ....	76
Kuva 91. WiZ-lampun WhatsApp-automaation jäljitys. ....	77
Kuva 92. WiZ-lampun WhatsApp-automaation lokitiedot.....	77
YAML-merkintäkoodi 1. WiZ tilatallentaja-automaation toinen ehto. ....	64
YAML-merkintäkoodi 2. Sensori- ja binäärisensorienteettien tiedostojen merkkkaus..	69

## **Liitteet**

Liite 1	Aineistonhallintasuunnitelma
Liite 2	Paikallaolointegraation YAML-koodi
Liite 3	WhatsApp viestin tallennusintegraation YAML-koodi
Liite 4	WhatsApp laukaisusanan tarkistusintegraation YAML-koodi

## 1 Johdanto

Opinnäytetyön tarkoituksena on perehtyä älykoteihin, niissä käytettyihin protokolliin, miten esineiden internet on osa tätä kokonaisuutta, sekä käytännön osuuden laitteisiin ja ohjelmistoihin. Tietoturva ja sen toteutumista käydään läpi älykotien ja protokollien osalta. Paikallis- ja pilviratkaisuiden eroista kerrotaan ja miten paikalliskäytännöt ovat yleensä pilviratkaisuja turvallisempia, vaikka jokseenkin vaikeimpia toteuttaa. Aihetta on rajattu protokollien osalta kolmeen yleisimmin älykotiratkaisussa käytössä olevaan. Ne käydään läpi yleisen toiminnan ja ominaisuuksien tasolla.

Tavoite on luoda paikallinen älykotiratkaisu käyttäen sen ohjauskeskuksena Raspberry Pi -minitietokoneeseen asennettua Home Assistant -ohjelmistoa. Ohjainkeskukset yhdistävät älykotien laitteet yhden käyttöliittymän alle, jolloin irrallisia sovelluksia ei tarvita. Älylaitteina toimivat Shelly Smart Plug -älypistorasia ja WiZ-älylamppu. Shelly valittiin sen mahdollistavien energiasäästöjen ja -seurannan takia. Shellyyn luotiin kaksi eri automaatiota, joiden avulla voidaan saada energiasäästöjä ja turvallisuutta aikaan. Myös WiZ:iin luotiin kaksi eri automaatiota. WiZ:in automaatioissa demonstroidaan YAML-merkintäkieltä omien integraatioiden luomisessa. Alun perin WiZ:in tilalla piti olla Airam-älylamppu, mutta se rajattiin lopuksi pois, sillä sen integraatio HA:n oli poistettu. Ratkaisun käytännön asennukset ja muut asetukset käydään läpi tarkalla tasolla, jotta se olisi muidenkin toteutettavissa. Lopuksi pohditaan Home Assistantin soveltuvuutta paikalliseksi älykotiratkaisuksi, sekä Airamin mahdollista integraatoratkaisua tulevaisuutta ajatellen.

Opinnäytetyössä pyritään vastaamaan seuraaviin tutkimuskysymyksiin:

- Miten Home Assistant asennetaan Raspberry Pi:hin?
- Miten Home Assistant keskustelee IoT-laitteiden kanssa?
- Miten IoT-laite saadaan toimimaan Home Assistantissa?
- Miten Home Assistant eroaa muista ohjaimista?

## 2 Esineiden internet (IoT)

Esineiden internet on nykypäivänä käytössä yleisesti ja siihen on vaikea olla törmäämättä. Vuonna 2022 esineiden internetiin oli yhdistettynä noin 13,1 miljardia eri laitetta ja määrän ennustetaan nousevan yli 15 miljardiin vuonna 2023 (Vailshery, 2022). IoT-laitteita löytyykin monilta elämän eri osa-alueilta yksittäisistä lamputa ja kahvinkeitinistä rakennuksen vedenkulutuksen seurannasta, aina kokonaisen kaupungin ilmanlaadun seurantaan asti (Vadakkanmarveetil, 2022). IoT-laitteita on tuhansia, mutta pääasiassa ne jakautuvat kahteen kategoriaan: kuluttajien ja teollisuuden laitteisiin. Kuluttajien laitteet ovat yleispäteviä ja käytettävissä moniin eri tarkoituksiin. Teolliset laitteet ovat enemmän tiettyihin tarkoituksiin ja tehdasmaisiin ympäristöihin rakennettuja. (Guinard & Trifa, 2016, s. 84)

Ei ole selviä rajoja, mitä IoT tarkalleen on tai ei ole ja siten se onkin haastava selittää yhdellä lauseella. Suuntaa antavana määritelmänä IoT on **järjestelmä** fyysisiä laitteita, joita pystytään monitoroimaan, kontrolloimaan, automatisoimaan ja joiden kanssa ollaan vuorovaikutuksessa elektronisten välineiden avulla. Laitteet kommunikoivat monien verkkoprotokollien välityksellä päämääränään internet. (Guinard & Trifa, 2016, ss. 4–5)

Itse fyysisen laitteen pitää täyttää pääsääntöisesti kaksi tekijää, jotta se luokitellaan osaksi esineiden internetiä: sillä on IP-osoite, ja se lähettää **omatoimisesti** itsestään jatkuvasti dataa järjestelmiin talletettavaksi ja analysoitavaksi (Collin & Saarelainen, 2016, ss. 30–31). On tärkeätä kuitenkin huomioida, että esimerkiksi älypuhelin tai tietokonetta ei yleisesti tulkita IoT-laitteiksi, vaikkakin niillä on IP-osoite ja ne lähettävät dataa eri järjestelmiin. IoT-laitteiksi tulkitaan enemmänkin laitteet, joiden ei normaalisti oleteta olevan yhteydessä internetiin, keräävän tai lähettävän dataa. Esimerkiksi hehkulamppu tai pesukone. (Ranger, 2020)

### 2.1 Lyhyt historia

Ensimmäinen tunnettu IoT-laite, vaikkei sitä silloin tällä termillä kutsuttukaan, oli neljän Carnegie Mellon -yliopiston oppilaan tekemä laite 1980-luvulla. Laite piti silmällä Coca-Cola-automaatin juomavaraston tilannetta monitoroimalla sen merkkivaloja. Valot vilkkuivat

oston yhteydessä ja pysyivät päällä, kun automaatti on tyhjä. Kaapeli vedettiin laitteesta koulun ARPANET-terminaaliin, jolloin automaatin tilannetta voitiin tarkkailla tietokoneelta sille kirjoitetun ohjelman avulla. Yksi oppilaista sai tämän idean kävellessään turhia matkoja tyhjälle automaatille. (Teicher, 2018)

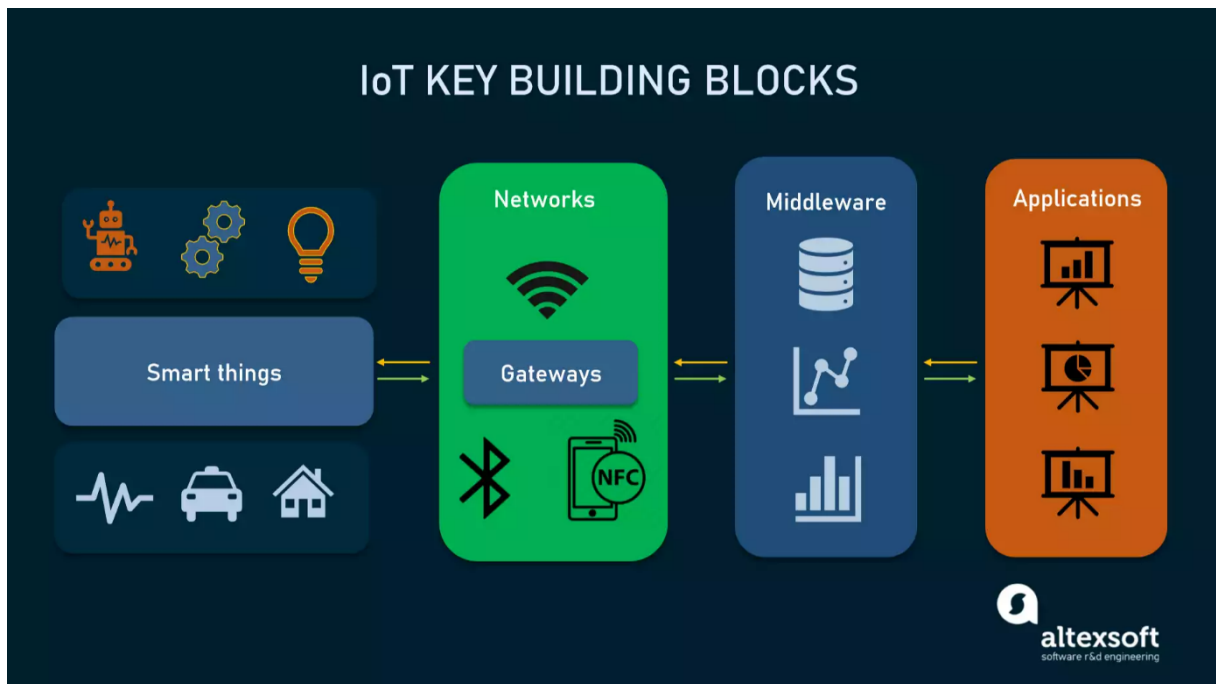
Termi Internet of Things keksittiin vuonna 1999, jolloin Kevin Ashton yritti saada esihenkilöitensä innostumaan toimitusketjun tuotteiden jäljityksestä RFID-sirujen avulla. Ashton käytti sen aikaista muotisanaa ”internet” saadakseen esihenkilöidensä huomion. Näin sana ”internet” on pysynyt termissä. Vaikka Ashton saikin muutaman esihenkilön innostumaan ideasta, ei IoT:hen kiinnitetty sen suuremmin huomiota kuin vasta vuonna 2010. US:n teknologiajättien, kuten Googlen ja Amazonin, panostuksen ansiosta IoT on saanut paljon jatkuvaa huomiota ja kiinnostusta viime vuosina. (Christou, 2019; Lueth, 2014)

## 2.2 Arkkitehtuuri

IoT-järjestelmän aikaansaamiseksi pitää olla ymmärrys sen arkkitehtuurista. Arkkitehtuurilla tarkoitetaan kaikkia komponentteja ja osa-alueita, joita järjestelmä pitää sisällään: laitteistot, ohjelmistot, protokollat, verkkoyhteydet ja kaikki muu tarvittava. Arkkitehtuurin tarkoituksena on selventää ja kategorisoida IoT-järjestelmää kokonaisuutena, ja näin auttaa käyttäjien päätöksentekoa, sekä omien projektien rakentamista ja ylläpitämistä. (InterviewBit, 2022)

Arkkitehtuuri kuvataan prosessina, jossa data kulkee eri kerroksien välillä. Jokaisella kerroksella on oma toimintansa ja tarkoituksensa. Arkkitehtuurin kerroksien määrä riippuu laajalti projektista itsestään, eikä yhtäpitävää jokaiseen projektiin sopivaa arkkitehtuuria olekaan tarjolla. Kuvassa 1 hahmotettu nelikerroksinen arkkitehtuuriratkaisu on kuitenkin olemassa, johon IoT-järjestelmän perusidea sisällytetään. (Satyabrata, 2020)

Kuva 1. Nelikerroksinen IoT-järjestelmän arkkitehtuuri (altexsoft, 2022).



Nelikerroksisen arkkitehtuurin kerrokset ovat:

1. Havainnointi (*eng. Perception*)

Havainnointikerros sisältää kaikki sensorit, aktuaattorit ja laitteet, jotka keräävät ja tuottavat käytettävän datan ollessaan vuorovaikutuksessa ympäristönsä kanssa. (NCS, 2022)

2. Verkko tai kommunikaatio (*eng. Network or Transport*)

Kaikki havainnointikerroksen data kulkee tämän kerroksen läpi. Verkkokerroksen tehtävänä on muuntaa data digitaaliseen luettavaan muotoon, jotta se voidaan lähettää eteenpäin kommunikaatioprotokollien avulla ylemmille kerroksille. Eli käytännössä verkkokerros yhdistää fyysiset laitteet verkkoon, internetiin ja muihin laitteisiin. Lisäksi verkkokerros salaa ja varmistaa datan turvallisuuden. (NCS, 2022)

3. Prosessointi tai väliohjelmisto (*eng. Processing or Middleware*)

Datan talletus, prosessointi, analysointi, tulkitseminen ja edelleen saatavaksi tekeminen on prosessointikerroksen toimintoja. Prosessointikerros tekee tulkitun datan mukaan päätöksiä, suodatuksia ja yhdistämiä. (Satyabrata, 2020)

4. Applikaatio (*eng. Application*)

Applikaatiokerros vaatii vuorovaikutuksen ihmisen kanssa. Kerros sisältää palvelut, joilla prosessoitua dataa voidaan hyödyntää esimerkiksi käyttöliittymän tai

mobiilisovelluksen avulla, eli käytännössä kaikki palvelut, jotka voivat lopullista dataa jotenkin hyödyntää. Tästä kerroksesta dataa voidaan myös lähettää takaisinpäin havainnointikerrokselle. (O'Shields, 2022)

### 2.3 Sensorit ja laitteet

Esineiden internetiä ei olisi ilman dataa, eikä dataa ilman sensoreita. Sensorit ovat fyysisiä komponentteja, jotka havaitsevat, mittaavat ja reagoivat ympäristössään tapahtuviin ilmiöihin. Ilmiö voi olla lämpötila, paine, ääni, valoisuus, liike, kosteus tai muu ympäristössä tapahtuva ilmiö. Dataa voidaan myös kerätä itse laitteesta, johon sensori on asennettu. (Collin & Saarelainen, 2016, s. 155)

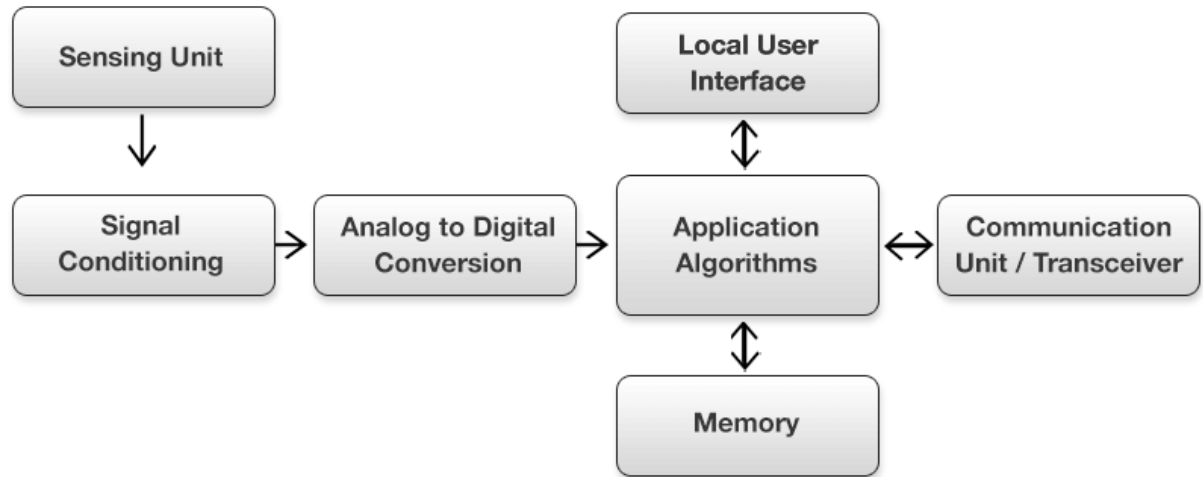
Suurin osa sensoreista ovat joko passiivisia tai aktiivisia:

- Passiiviset sensorit pelkästään vastaanottavat ympäristöstä havaitsemansa datan. Esimerkiksi lämpötilan.
- Aktiiviset sensorit lähettävät ensin oman signaalin, jonka jälkeen ne havaitsevat ympäristössä tapahtuvan vastauksen siihen. Esimerkiksi infrapunalla toimiva etäisyysanturi.

(Trivedi, 2020)

Sensorit ovat osa IoT-laitekokonaisuutta (kuva 2), joka on käytännössä pieni tietokone omalla mikroprosessorilla, ohjausyksiköllä / datanmuuntimella, virtalähteellä (yleensä patteri), muistilla ja verkkoyhteyden mahdollistavalla komponentilla. Mitattu data ei aina ole suoraan digitaalisessa elektroniikalle luettavassa muodossa. Muunnos on tarpeen, jotta data olisi käsiteltävissä ja näin hyödynnettävissä. Tämä tapahtuu laitteen ohjausyksikön, erillisen A/D-muuntimen tai I/O-moduulin avulla. Verkkoyhteys mahdollistaa kerätyn datan lähettämisen eteenpäin. Laitteilla on myös oma muisti datan talletukseen. Laitteen omilla algoritmeilla ja ohjelmistoilla tapahtuvan datan esikäsittelyn avulla voidaan vähentää turhan tietoliikenteen määrää. Esimerkiksi laite voidaan asettaa lähettämään vain dataa, joka on lähellä mennä yli asetettujen rajojen. Turhaa liikennettä vältetään, sillä dataa ei lähetetä koko ajan. Tämä vähentää myös energiankulutusta. (Collin & Saarelainen, 2016, ss. 152, 155; Farnell element14, 2017)

Kuva 2. IoT-laitteen rakennusosa (Premier Farnell Ltd, 2017).





### 3 Älykodit

Älykoti on pähkinänkuoressa koti, joka on varustettu IoT-laitteilla. Kotiautomaatio tarkoittaa sen sijaan näiden laitteiden automaatiota. Automaatiolla laitteisiin voidaan ohjelmoida erilaisia toimintoja, jotka käynnistyvät niille asetettujen parametrien, muiden laitteiden havaintojen tai pyyntöjen perusteella. Esimerkkinä valot syttyvät päälle, kun astut huoneeseen ja sammuvat lähtiessäsi, termostaatti voi käynnistää ilmastointilaitteen ennen kotiin tuloa (tai muuna haluttuna aikana), etuovi lukittuu vasta, kun kaikki perheenjäsenet ovat kotona, uuni sammuu automaattisesti, jos se unohtuu päälle tai liikeseensori lähettää puhelimeen hälytyksen sen huomattua liikettä asunnossa, jonka pitäisi muiden sensoreiden mukaan olla tyhjä. Energiasäästöjä saadaan aikaiseksi laitteilla, jotka monitoroivat energiankulutusta ja antavat käskyjä muille laitteille sen mukaan. Tällaiset arkea helpottavat, turvaa tuovat ja säästöjä aikaansaavat asiat ovat älykodin päätarkoituksia. Laitteet ovat yhteydessä internetiin, joten niiden etäohjaus on myös mahdollista puhelimen, äänikomentojen tai muunlaisten käyttöliittymien avulla. (Vigderman & Turner, 2022)

#### 3.1 Ohjauskeskukset

Laitteita löytyy monien eri valmistajien kehittämistä ja niihin yleensä sisältyy oma sovellus, jonka kautta laitetta voidaan kontrolloida. Älykoti ei kuitenkaan ole käytännöllinen, jos jokaista laitetta tarvitsee kontrolloida ja konfiguroida erillisen sovelluksen kautta.

Kontrollonin keskittämiseen tarvitaan älykodin ohjauskeskus. Ohjauskeskus kerää älykodin laitteet yhden käyttöliittymän alle, jossa niitä voidaan helposti konfiguroida ja kontrolloida. (Elias, 2021)

Ohjauskeskuksia löytyy valmistajariippuvaisia ja -riippumattomia. Riippuvaisiin keskuksiin voidaan yhdistää vain saman valmistajan laitteita ja riippumattomiin useamman eri valmistajan laitteita. Ohjauskeskuksia löytyy myös yhtä tai useampaa protokollaa tukevia. Esimerkkinä Samsung SmartThings tukee Wi-Fi-, Z-Wave- ja ZigBee-protokollia. (Bell, 2022)

Ohjauskeskukset tuovat mukanaan myös automaatiot. Erilaisilla automaatioilla voidaan luoda "kohtauksia" (*eng. scene*), joilla usea laite voidaan ohjelmoida tekemään tietty asia tiettyä ajankohtana, muun laitteen datan tai ajastimen mukaan. Kohtauksia voidaan myös

kontrolloida manuaalisesti käyttöliittymän kautta. Esimerkiksi elokuvien katseluun luodulla kohtauksella voidaan valot himmentää ja sälekaihtimet sulkea samalla komennolla. (Bell, 2022)

Ohjauskeskukset toimivat pilvessä, paikallisesti tai näiden yhdistelmänä. Pilvitoiminnalliset ohjauskeskukset ovat yleensä helpompi asentaa ja laitteiden yhdistäminen on vaivattomampaa, mutta niissä on myös huonot puolensa. Yksityisyys voi olla ongelma, sillä kaikki data kulkee valmistajan kautta. Valmistajayhtiö voi myydä dataa eteenpäin kolmansille osapuolille tai heidän järjestelmiinsä voi tapahtua tietomurto, jolloin data joutuu väärin käsiin. Pilviratkaisut tuovat riippuvuuden valmistajayhtiön järjestelmiin ja internetyhteyteen, jolloin niiden ollessa pois käytöstä syystä tai toisesta, myös älykodin laitteet ovat pois käytöstä. Paikallisissa ratkaisuissa kontrolli on kokonaan oma, jolloin riippuvuutta valmistajayhtiön järjestelmiin tai internetyhteyteen ei tarvita. Yksityisyys on riippuvainen tietoturvaratkaisuista, mutta data pysyy oman verkon sisällä. Paikalliset ratkaisut ovat yleensä vaikeampi ottaa käyttöön ja vaativat enemmän teknistä osaamista. (Ovens, 2020)

### **3.2 Tietoturva**

Älykodin laitteet tuovat mukanaan arkea helpottavia mukavuuksia, mutta myös uusia tietoturvariskejä, jotka pitää ottaa huomioon kotiverkon turvaamiseksi. Älykodin laitteisiin voi kohdistua useita erilaisia hyökkäyksiä, joiden onnistuttua yksityisyys vaarantuu, laitteet voidaan ottaa haltuun tai pahimmassa tapauksessa ne voivat aiheuttaa hengenvaarallisia tilanteita. Esimerkkinä olohuoneessa olevasta turvakamerasta voi myös hyökkääjä monitoroida kodin toimia ja suunnitella murron ajankohtaan, jolloin koti on tyhjiillään. (Kaspersky, 2021)

Rajoitetun laitteiston ja prosessointitehokkuuden takia älykodin laitteet eivät kykene samantasoiseen tietoturvaan, kuten tietokone, joten ne ovat helpompi murtaa. Älykodin laitteiden käytössä olevia protokollia on monia ja niissä omat tietoturvatimensa, joten sama laite käyttäen eri protokollaa voi olla haavoittuvaisempi, kuin toinen. Laitteiden valmistuksessa käytettävät yleiset peruskomponentit tai koodikirjastot voivat haavoittuvuuden esiin tullessa vaikuttaa samanaikaisesti miljooniin laitteisiin. Laittevalmistajat monesti haluavat tuotteensa nopeasti markkinoille ja sivuuttavat

tietoturvan kokonaan. Näiden takia hyökkääjät kohdentavatkin yleisesti älykodin laitteita. (Fortinet, ei pvm.)

Konkreettinen esimerkki hyökkäyksestä on vuonna 2016 maailmalla esiintynyt Mirai-bottiverkko. Bottiverkko koostui yli 100 000 IoT-laitteesta, joita käytettiin palvelunestohyökkäyksiin. Palvelunestohyökkäyksen tarkoituksena on estää jonkin verkkosivuston käyttö, kohdentamalla siihen suuria määriä tietoliikennettä, jotta se ei voisi enää palvella oikeita asiakkaita. Laitteet kaapattiin bottiverkkoon skannaamalla internetiä julkisista IoT-laitteista ja yhdistämällä niihin käyttämällä tunnettuja oletuskäyttäjänimiä ja -salasanoja. (Malwarebytes, ei pvm.)

Älykodin turvaamiseksi, ennen laitteiden ostoa on tärkeä tutkia niiden ominaisuuksia ja tarkistaa laitevalmistajien luotettavuus ja tietosuojakäytännöt, varsinkin, jos laitteet yhdistävät pilveen. Laitteiden käyttötarkoitusta ja tarpeellisuutta tulee miettiä ennen niiden lisäämistä omiin ratkaisuihin. Laitteiden oletuskäyttäjätunnukset ja -salasanat on hyvä vaihtaa vahvoihin, sekä aika-ajoin kokonaan uusiin. Laitteiden käyttämättömät portit ovat tietoturvan kannalta tärkeä sulkea. Päivitykset on syytä pitää ajan tasalla ja automaattitarkistukset päällä. Tietoturvaa lisää laitteiden verkotus omiin vierasverkkoihin, jolloin ne ovat eristettyinä kodin muista laitteista. (Kaspersky, 2021)

## 4 Laitteiden kommunikaatio älykodeissa

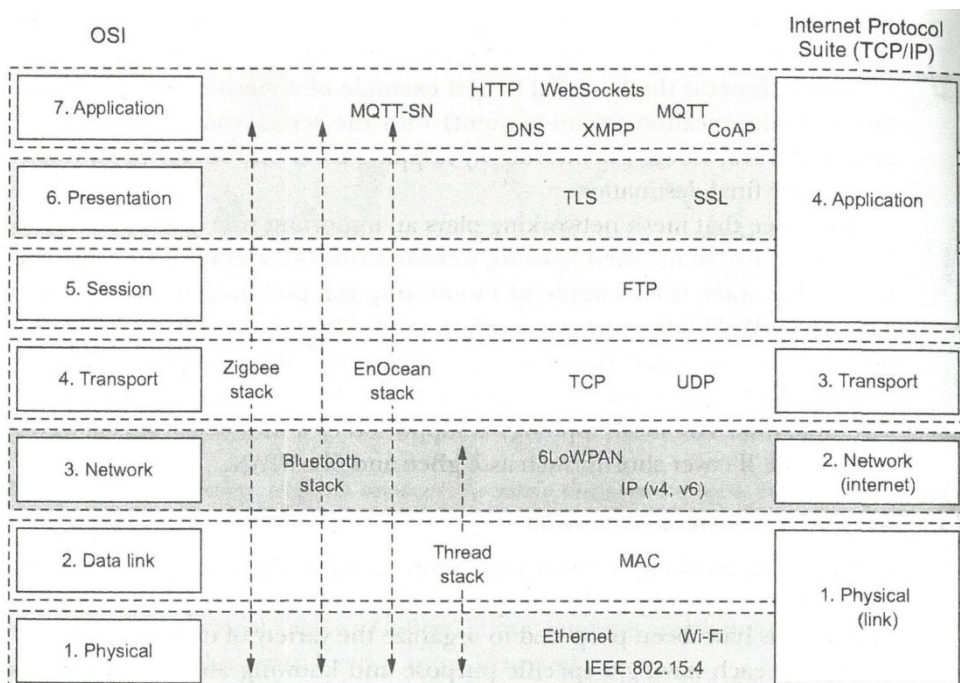
Datan kerääminen on tärkeä vaihe IoT:ssa, mutta datasta ei saada kovinkaan paljoa hyötyä, jollei sitä pystytä lähettämään verkon muille laitteille, sovelluksille tai palveluille.

IoT-laitteiden kommunikaatio tapahtuu erilaisten protokollien ja standardien avulla.

Protokollat voidaan jakaa niiden toiminnan perusteella pääasiassa kahteen kategoriaan:

data- ja verkkoprotokolliin. Protokollien toiminta voidaan sijoittaa kuvan 3 mukaisesti OSI- tai TCP/IP-mallin eri kerroksiin, joka auttaa hahmottamaan kuinka ne toimivat yhteydessä toisiinsa. Jokaisesta kerroksesta voidaan ottaa käyttöön yhteensopivia protokollia, näin muodostaen protokollapinon, jonka avulla laitteen data saadaan siitä sen päämäärään ja aina käytettävään sovellukseen asti. Kuvassa 3 myös nähdään, että protokollat kuten Zigbee ja Thread, muodostavat itsessään jo täysin tai osittain käytettävän protokollapinon. (Guinard & Trifa, 2016, ss. 113–115)

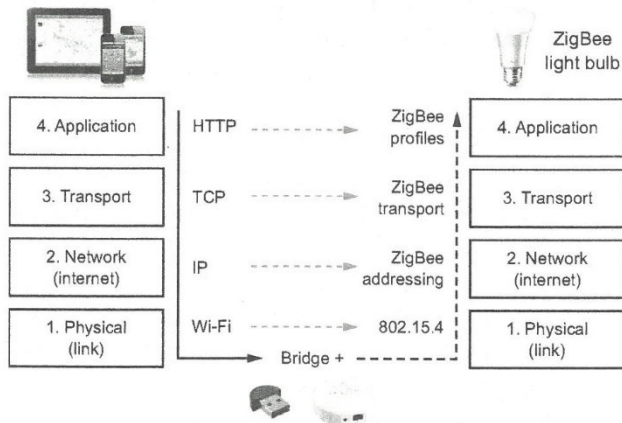
Kuva 3. Protokollat ja protokollapinot OSI- ja TCP/IP-mallin eri kerroksissa (Guinard & Trifa, 2016, s. 114).



Verkkoprotokollan tarkoitus on muodostaa verkkoyhteys eri laitteiden välille ja lopulta internetiin. Protokollan mukaan laite voi muodostaa yhteyden internetiin suoraan tai

gateway-laitteen (*yhdyskäytävän*) avulla. Gateway-laitteet mahdollistavat internet yhteyden protokollille, jotka eivät perustu IP:hen, muuntamalla ne internetissä ymmärrettävään muotoon (kuva 4). (Microsoft Azure, ei pvm.)

Kuva 4. Laitteen internet yhteys gatewayn avulla (Guinard & Trifa, 2016, s. 129).



Verkkoprotokollaa valittaessa tärkeinä ominaisuuksina ovat niiden kantama, virrankulutus ja datansiirtonopeus. Pitkän kantaman WAN-verkkoteknologiat, kuten mobiiliverkkojen 4G tai 5G, omaavat erittäin nopean datansiirron, mutta kuluttavat paljon virtaa. Lyhyen kantaman PAN-verkkoteknologiat, kuten Bluetooth, sen sijaan siirtävät dataa hitaammin, mutta erittäin minimaalisella virrankulutuksella. Pieni virrankulutus on IoT:ssa hyvin tärkeä ominaisuus, sillä laitteet ovat yleensä patterilla toimivia. (Collin & Saarelainen, 2016, ss. 163–166)

Dataprotokollan tarkoitus on sen sijaan mahdollistaa laitteidenvälisen kommunikaation tässä verkossa, eli tarjota yhteinen kieli, jolla laitteet puhuvat. (Microsoft Azure, ei pvm.)

## 4.1 ZigBee

ZigBee on IEEE 802.15.4-radiostandardiin perustuva avoin langaton protokolla, jonka kehitys on painottunut alhaiseen virrankulutukseen ja hintaan, näin sopien mainiosti koti- ja teollisuusautomaation, sekä terveydenhuollon IoT-ratkaisuihin. (Felch, 2021)

### 4.1.1 Toiminta

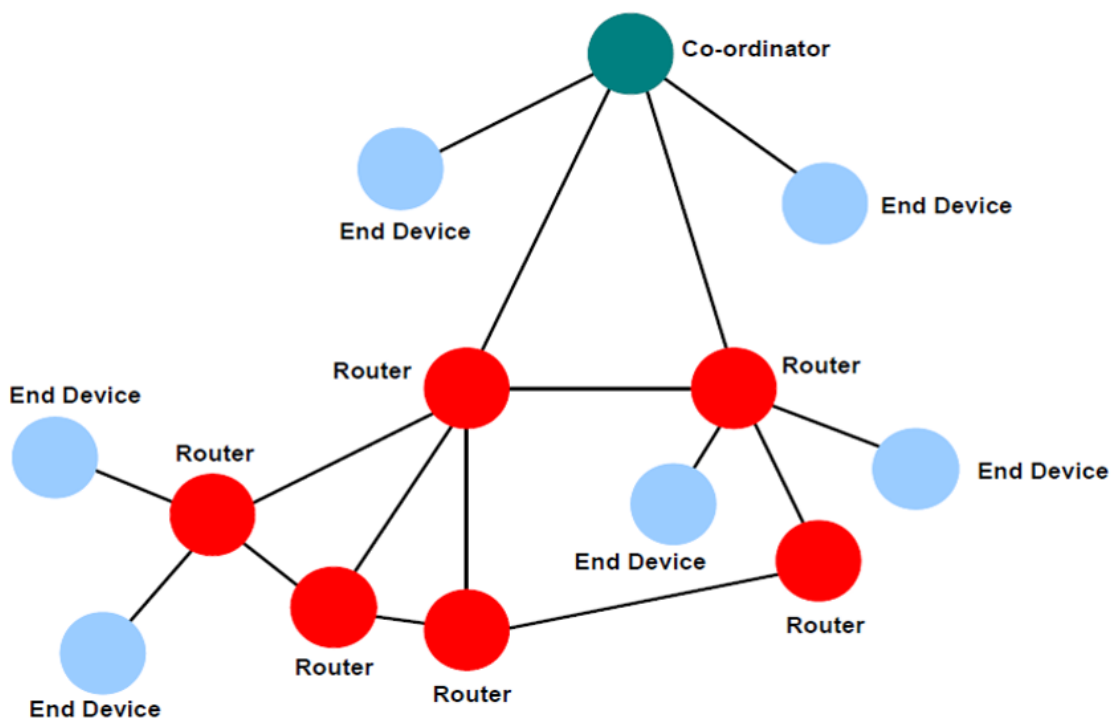
Alhaisen virrankulutuksen omaavana ZigBeen kommunikaatio tapahtuu hyvin vähäisellä tiedonsiirtonopeudella, noin 20–250 kbps. Tämä ei kuitenkaan ole varsinainen ongelma,

vaan voi tuoda protokollan kommunikaatioon toimintavarmuutta, sillä pakettien pienen koon ansiosta niitä harvoin tarvitsee yrittää lähettää uudelleen, eli ne menevät kerrasta perille. ZigBeeen kantama on noin 10–100 metriin, joten se kuuluu PAN-verkkoihin. ZigBee käyttää samaa 2.4GHz taajuutta, kuin Wi-Fi, joten häiriötä voi tapahtua. (Felch, 2021)

ZigBee-laitteet pystyvät sisäänrakennetun ominaisuutensa ansiosta olemaan suurimman osan ajasta lepotilassa, näin vähentäen virrankulutusta entisestään. Laite herää aika-ajoin kommunikoidaan, jonka jälkeen se kytkeytyy takaisin lepotilaan. Tämän ja muiden virrankulutusta alentavien ominaisuuksien ansiosta patterillinen ZigBee-laite voikin olla toimintakykyinen jopa vuoden. (Felch, 2021)

Yksi ZigBeeen suurista eduista on sen mahdollistama mesh- tai vertaisverkkotopologia. Kuten kuvassa 5 nähdään, tämän avulla tietoliikenne voi kulkea usean laitteen kautta päämääräänsä ja näin myös pidentäen kantavuutta huomattavasti. Tietoliikenteen ei tarvitse kulkea yhden tietyn laitteen kautta päämääräänsä. Verkko on myös itsestään paraneva, eli se osaa reitittää liikenteen uudelleen, jos jokin laite verkossa kaatuu. (Collin & Saarelainen, 2016, s. 174)

Kuva 5. ZigBee mesh-verkko (STMicroelectronics, 2022).



ZigBee-verkko koostuu kolmesta eri laitetypistä:

1. ZigBee keskusyksikkö (*eng. Coordinator*)

Keskusyksikkö on erillinen ostettava laite, joita ZigBee-verkossa on aina vain yksi. Keskusyksikön tehtävänä on ensin muodostaa verkko yhdistämällä siihen laitteet, autentikoimalla ne ja antamalla niille osoitteet. Keskusyksikkö tekee tämän myös laitteille, jotka yhdistyvät verkkoon myöhemmin. Verkon luotua keskusyksikkö toimii reitittimen tavoin ja näin myös vaatii kokoaikaisen päällä olemisen. Keskusyksikkö luosillan kotiverkon ja ZigBee-verkkojen välille, yhdistäen laitteet internetiin. ZigBee ei ole IP-pohjainen protokolla, joten internetyhteyden toimimiseksi keskusyksikön tehtävänä on myös muuntaa se IP-muotoon. Tämän avulla muut verkon laitteet, kuten puhelimet ja Wi-Fi-reitittimet, voivat ymmärtää ZigBeetä. (Felch, 2021)

2. ZigBee reititin (*eng. Router*)

Reititin on laite, joka toimii verkossa välikätenä. Se välittää tietoliikennettä eteenpäin kohti päämäärää, näin laajentaen verkon kantamaa. Reititin pystyy myös keskusyksikön avulla yhdistämään laitteita verkkoon. Reitittimet tallentavat niihin yhdistyneiden päätelaitteille tarkoitetun datan ja lähettävät sen niille vasta, kun päätelaitteet sitä pyytävät. Tämän takia reitittimen täytyykin olla koko ajan päällä, joten patterilliset laitteet eivät ole sopivin vaihtoehto reititinlaitteeksi. Hyvä esimerkki sopivasta reititinlaitteesta on älypistorasia, sillä sen voi kiinnittää suoraan verkkovirtaan. Vaikkakin reitittimet lisäävät verkon kantavuutta, eivät ne ole pakollisia. (Felch, 2021)

3. ZigBee päätelaite (*eng. End device*)

Päätelaitteet ovat kerrallaan yhteydessä yhteen reitittimeen, eivätkä ne voi välittää liikennettä suoraan toisille päätelaitteille. Päätelaitteet ovat yleensä patterillisia, ja näin suurimman osan ajasta lepotilassa. Päätelaitteet heräävät aika-ajoin tarkistamaan niille mahdollisen kuljetuksessa olevan datan reitittimeltään. (Felch, 2021)

#### 4.1.2 Laitteiden yhteensopivuus

ZigBee Alliance (nykyään Connectivity Standards Alliance, CSA) on protokollan kehitykseen ja sertifiointiin perustettu liittouma, johon osallistuu noin 400 eri yritystä, kuten Samsung, Philips, Google, Apple, Amazon ja IKEA. Liittouma ylläpitää ja julkaisee eri ZigBeen versioita, joiden tarkoituksena on taata eri laitevalmistajien laitteiden yhteensopivuus. Eli jos laite on varustettu ZigBeen logolla, voidaan olla varma, että se on yhteensopiva muiden ZigBee laitteiden kanssa. (Qin, 2023)

Ennen nykyistä ZigBee Alliancen kehittämää 3.0-protokollaversiota, laitteet käyttivät erilaisia profiileja, jotka olivat tiettyihin käyttötarkoituksiin kehitettyjä protokollan versioita. Tietyn profiilin laitteet olivat yhteensopivia keskenään, mutta eivät eri profiilien laitteiden kanssa. Kotiautomaatiossa käytetään pääasiassa kahta profiilia: ZigBee Light Link (ZLL) ja Zigbee Home Automation (ZHA). ZigBee 3.0 yhdistää nämä profiilit saman version alle, jotta yhteensopivuus olisi varmempi. Laitteiden yhteensopivuus kannattaa siis varmistaa, sillä jotkin laitteet käyttävät vielä vanhempia yhden profiilin versioita. (Basumallick, 2022)

#### 4.1.3 Tietoturva

ZigBee turvaa laitteiden välisen kommunikaation 128 bittisellä AES-salauksella. Salaus tapahtuu verkko- ja sovellustasolla käyttäen erilaisia salausavaimia, joista kaikki ovat 128-bittisiä. (Silicon Laboratories, 2022, s. 3)

Verkkotasolla käytetään koko verkon tiedossa olevaa verkkoavainta, jota käytetään siinä tapahtuvan kommunikaation salaukseen ja purkamiseen. Datan eheys tarkistetaan jokaisella reitittimien välillä tapahtuvassa ”loikkauksessa” (*eng. hop-by-hop*). Jokaisella laitteella on oma laskuri, joka pitää laskua omista ja naapureidensa loikkausmääristä. Tämä estää mahdollisen datan injektioita verkkoon. Jos data kulkee verkossa ilman verkkoavainta tai laskuria, tiputetaan se pois. (Silicon Laboratories, 2022, s. 3)

Verkon autentikaatiossa voidaan käyttää niin sanottuja luottamuskeskuksia (*eng. trust center*). Yleensä luottamuskeskus on sama laite, kuin ZigBee keskusyksikkö, mutta se voi olla myös oma laitteensa. Kaikki uudet laitteet, jotka haluavat verkkoon, yhdistetään ensin vain väliaikaisesti, kunnes luottamuskeskus päättää sallitaanko ne verkkoon. Yhdistävä laite voi



kommunikoida vain autentikaatioon liittyvää dataa ennen verkkoon pääsemistä.

Luottamuskeskus voi päättää sallia yhteyden, estää sen tai olla huomioimatta sitä. Salliessa laitteelle lähetetään verkkoavain, muissa tapauksissa se tiputetaan verkosta.

Luottamuskeskus myös luo ja lähettää aika-ajoittain uuden verkkoavaimen verkon laitteille.

Tämä luo turvaa verkkoavaimen päätyessä väärin käsiin, sillä sen käyttöaika on lyhyt.

(Silicon Laboratories, 2022, ss. 4, 7)

Sovellustasolla käytetään kahden laitteen välistä linkitysavainta (*eng. link key*), joka on vain näiden kahden laitteen tiedossa. Linkitysavaimia on kahden tapaisia:

- **Luottamuskeskuksen linkitysavain** (*eng. Trust center link key*)

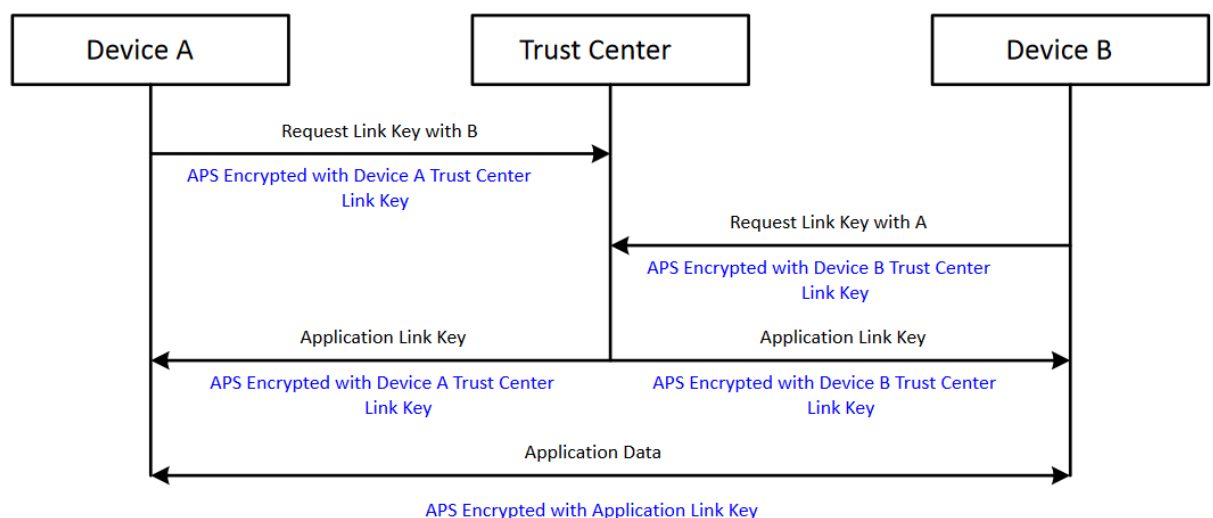
Tätä avainta käytetään laitteen ja luottamuskeskuksen välisessä kommunikaatiossa. Jokaisella verkon laitteella on oma luottamuskeskuksen jakama avain.

- **Sovelluksen linkitysavain** (*eng. Application link key*)

Tätä avainta käytetään kahden laitteen välisessä kommunikaatiossa (paitsi luottamuskeskuksen kanssa). Kommunikoivat laitteet voivat sopia linkitysavaimen keskenään tai kuvan 6 mukaisesti pyytää sitä luottamuskeskukselta.

(Silicon Laboratories, 2022, ss. 6–8)

Kuva 6. Sovelluksen linkitysavain luottamuskeskukselta (Silicon Laboratories, 2022, s. 8).



## 4.2 MQTT (Message Queuing Telemetry Transport)

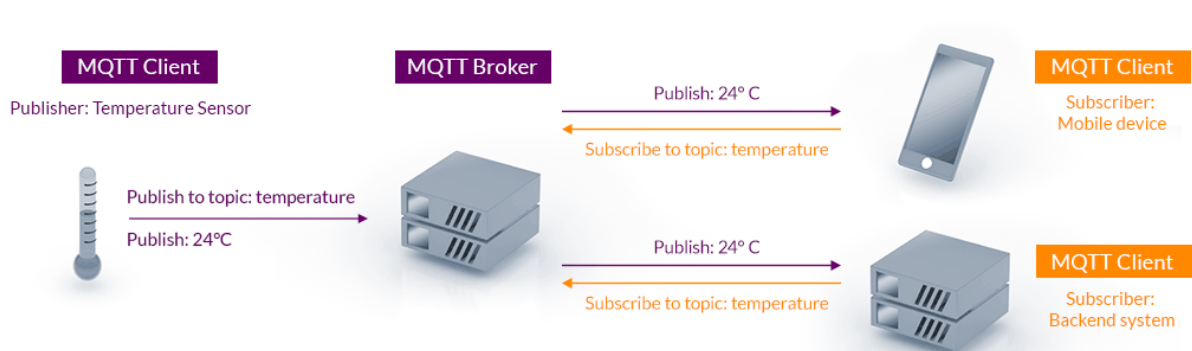
MQTT on IoT:ssa käytetty TCP/IP protokollapinon päälle rakennettu avoin dataprotokolla. MQTT on resurssivaatimuksiltaan kevyt, tehokas ja pienen tiedonsiirtonopeuden omaava protokolla, joka on kehitetty toimimaan luotettavasti verkoissa, joissa yhteys ei välttämättä ole parasta luokkaa. Pienen virrankulutuksen omaavana protokolla sopii erinomaisesti myös älykoteihin. (Catchpoint, ei pvm.)

### 4.2.1 Toiminta

MQTT:n toiminta perustuu kuvan 7 mukaiseen julkaisija-tilaaja-malliin, jossa laitteet eristetään toisistaan yhteisen välittäjälaitteen avulla. Mallin laitteet eivät ole tietoisia muusta kuin välittäjästä. Tämä on yksi mallin vahvuuksia, sillä laitteiden ei tarvitse pitää kirjaa kuin välittäjän osoitteesta, eikä lähettäjä- ja vastaanottajalaitteiden tarvitse kommunikoida samaan aikaan tai odottaa toisiaan. Mallin laitteita kutsutaan asiakkaisiksi ja niillä on kaksi roolia: julkaisija ja tilaaja. Roolin mukaan, asiakkaat voivat julkaista itsestään aiheiden alle organisoitua dataa tai/ja tilata dataa omaan käyttöön haluamistansa aiheista. (HiveMQ, ei pvm., osa 2)

Asiakslaitteet eivät suoraan kommunikoi toistensa kanssa, vaan kaikki tietoliikenne kulkee erillisen MQTT-välittäjän kautta. Välittäjän tehtävänä on yhdistää ja autentikoida asiakaslaitteet verkkoon, pitää kirjaa yhteyksistä ja varmistaa niiden salaus, ottaa vastaan julkaisuja ja tilauksia, välittää aiheiden viestit tilaajille ja säilyttää viestejä mahdollista uudelleenlähetystä varten. (Catchpoint, ei pvm.)

Kuva 7. MQTT julkaisija-tilaaja-malli (MQTT, 2022).



MQTT sessioissa käytetään monia erilaisia viestejä. Näistä yleisimmät ja tärkeimmät ymmärtää ovat: yhteyden muodostamisessa -, julkaisussa - ja tilaamisessa käytetyt viestit. (Catchpoint, ei pvm.)

#### 4.2.2 Yhteyden muodostaminen

Yhteyden muodostamisessa asiakaslaite lähettää välittäjälle ensin CONNECT-viestin. Välittäjä vastaa laitteelle CONNACK-viestillä, jossa kerrotaan yhteyden hyväksymisestä tai hylkäämisestä. CONNECT-viestin pitää sisältää tietyt asiat, jotta välittäjä hyväksyy ja yhdistää laitteen verkkoon: (HiveMQ, ei pvm., osa 3)

- **Tunniste (eng. ClientID)**  
Laitteen uniikki nimi, jota välittäjä käyttää laitteiden tunnistamiseen ja yhteyksien ylläpitämiseen verkossa. Tunnisteen voi valita itse tai antaa laitteen generoida satunnaisen. Tunnisteet kannattaa valita niin, että ne helpottavat laitteiden ylläpitämistä. Tieto on pakollinen. (HiveMQ, ei pvm., osa 3)
- **Puhdas sessio (eng. Clean Session)**  
Muuttuja, jonka avulla välittäjä tietää laitteen session laadun. Muuttujan ollessa epätosi, sessio on pysyvä. Pysyvässä sessiossa välittäjä säilyttää laitteen tilaukset ja palvelun laatutason 1 tai 2 omaavat viestit mahdollista uudelleenlähetystä varten. Muuttujan ollessa tosi, sessio on ei pysyvä, eli puhdas. Puhtaassa sessiossa välittäjä ei säilytä mitään ja tyhjentää muistin mahdollisen edellisen pysyvän session tiedoista. Tieto on pakollinen. (HiveMQ, ei pvm., osa 3)
- **Käyttäjänimi ja salasana**  
Välittäjä voi vaatia yhteyden muodostamiseksi käyttäjänimen ja salasanan. Tieto on valinnainen, mutta käyttäjänimen ja salasanan vaatimista suositellaan tietoturvan lisäämiseksi. (HiveMQ, ei pvm., osa 3)
- **Viimeinen tahto (eng. Last Will and Testament, LWT)**  
Viesti, jonka välittäjä lähettää sen tilaajille, jos laitteen yhteys menetetään. Viesti lähetetään vain, kun yhteys menetetään odottamattomalla tavalla, eli tapauksissa,

joissa yhteyden katkaisu ei ole hallittu. Viimeisen tahdon viesti on samanlainen, kuin tavallisen julkaisun viesti. Tieto on valinnainen. (HiveMQ, ei pvm., osa 3)

- Pidä elossa (*eng. Keep Alive*)

Aikamäärä, jonka rajoissa laitteen ja välittäjän pitää lähettää viestejä toisilleen.

Laitteen tehtävänä on pitää yhteys välittäjään elossa aikarajan lähestyessä, lähettämällä sille PINGREQ-viesti. Välittäjä vastaa laitteelle PINGRESP-viestillä.

Aikarajan ylittyessä yhteys katkaistaan. Yhteys pysyy elossa ilman PINGREQ-viestiäkin, sillä ajastin alkaa alusta kaiken välittäjän kanssa käydyin kommunikation toimesta. Tieto on pakollinen. (HiveMQ, ei pvm., osa 3)

#### 4.2.3 Julkaiseminen

Julkaisussa asiakaslaite lähettää välittäjälle PUBLISH-viestin, jota välittäjä käyttää lähettääkseen viestin edelleen tilaajille. Välittäjä vastaa laitteelle palvelun laatutason mukaan PUBACK-, PUBREC- ja PUBCOMP-viesteillä ja lähettää sen aiheen tilaajille. PUBLISH-viestin pitää sisältää tietyt asiat: (HiveMQ, ei pvm., osa 4)

- Paketin tunniste (*eng. Packet Identifier*)

Numerollinen tunniste, jota käytetään viestin yksilöimiseen laitteen ja välittäjän välillä. Tunniste luodaan automaattisesti laitteen tai välittäjän toimesta. Tunnistetta käytetään hyväksi vain viesteissä, joiden palvelun laatutaso on 1 tai 2. Laatutason 0 viesteissä paketin tunniste on aina 0. (HiveMQ, ei pvm., osa 4)

- Aihe (*eng. Topic*)

Aihe on UTF-8 merkistöä käyttävä merkkijono, joka auttaa välittäjää ohjaamaan viestin sen tilaajille. Aiheet voidaan jakaa kauttaviivan (/) avulla ala-aiheisiin, esimerkiksi "koti/olohuone/kattolamppu", joka vastaa kattolamppua kodin olohuoneessa. Aiheiden pituus pitää olla vähintään yksi merkki, ja ne ovat merkkikokoriippuvaisia, eli aihe "koti/olohuone" ei ole sama kuin "Koti/olohuone". Välittäjä hyväksyy kaikki aiheet ilman sen erillistä luomista, eli aihe voidaan viestiin laittaa, vaikkei sitä olisikaan vielä olemassa. Aiheet kannattaa valita niin, että ne helpottavat laitteiden ylläpitoa. (HiveMQ, ei pvm., osa 5)

- **Palvelun laatutaso** (*eng. Quality of Service, QoS*)  
Laatutasoja on kolme: 0, 1 ja 2. Palvelun laatutaso, määrittää kuinka luotettavasti viesti toimitetaan vastaanottajalle. (HiveMQ, ei pvm., osa 4)
- **Säilytyslippu** (*eng. Retain flag*)  
Muuttuja, joka kertoo välittäjälle pitääkö viesti säilyttää. Välittäjä säilyttää aina vain yhden, eli viimeisimmän, viestin per aihe. Säilytetyn viestin ideana on pitää tilaajat aina ajan tasalla aiheen viimeisimmästä viestistä. Välittäjä lähettää säilytetyn viestin aiheen uusille tilaajille. Tilaajat voivat kytkeytyä verkosta pois ja saada viimeisimmän viestin takaisin tullessaan. Säilytyslippua on hyvä käyttää esimerkiksi tilanteissa, joissa sensorin tila muuttuu harvoin. Tilaajien ei tarvitse odottaa aiheeseen uutta julkaisua, vaan saavat viimeisimmän tiedon heti. (HiveMQ, ei pvm., osa 8)
- **Tietosisältö** (*eng. Payload*)  
Itse viestin sisältö, joka voi olla lämpötila, kosteus, päällä/pois -tieto tai mikä vain muu laitteen tuottama data. Sisältö muotoillaan ja lähetetään laitteen haluamalla tavalla, esimerkiksi binääri-, teksti- tai JSON-muodossa. (HiveMQ, ei pvm., osa 4)
- **DUP-lippu** (*eng. DUP flag*)  
Muuttuja, joka kertoo viestin olevan duplikaatti, eli se on lähetetty uudestaan. Viesti voidaan lähettää duplikaattina, jos vastaanottaja ei tiedosta sen perille saapumista. Muuttujaa käytetään hyväksi vain viesteissä, joiden palvelun laatutaso on 1 tai 2. (HiveMQ, ei pvm., osa 4)

#### 4.2.4 Tilaaminen ja jokerimerkit

Aiheen tilaamiseen asiakaslaite lähettää välittäjälle SUBSCRIBE-viestin. Välittäjä vastaa laitteelle SUBACK-viestillä, jossa kerrotaan tilauksen onnistuminen tai epäonnistuminen. SUBSCRIBE-viestin pitää sisältää tietyt asiat: (HiveMQ, ei pvm., osa 4)

- **Paketin tunniste** (*eng. Packet Identifier*)  
Numerollinen tunniste, jota käytetään viestin yksilöimiseen laitteen ja välittäjän välillä. Tunniste luodaan automaattisesti laitteen tai välittäjän toimesta. (HiveMQ, ei pvm., osa 4)

- Lista tilauksista

Yhdessä SUBSCRIBE-viestissä voidaan tilata useampi aihe kerralla. Tilauksessa kerrotaan haluttu aihe ja palvelun laatutaso, jolla aiheen viestit halutaan vastaanottaa. Jos tilauksessa tilataan vahingossa sama aihe useampaan kertaan, välittäjä hyväksyy niistä suurimman laatutason omaavan. (HiveMQ, ei pvm., osa 4)

Aiheeseen voidaan tilauksissa sisällyttää ns. jokerimerkkejä (*eng. Wildcards*), joiden avulla yhteen aiheeseen voidaan sisällyttää useampi aihetaso kerrallaan. Jokerimerkkejä on kahden tyyppisiä: (HiveMQ, ei pvm., osa 5)

- Plus (+)

Voidaan korvata yksi aihetaso, esimerkiksi "koti/alakerta/+/lämpötila". Tällä tilataan kaikki kodin alakerran lämpötilaa viestittävät aiheet. Esimerkki toimii vain aiheissa, jotka alkavat "koti/alakerta" ja loppuvat "lämpötila". Edessä, välissä tai lopussa ei saa olla mitään erilaista. Eli aiheita "koti/yläkerta/lämpötila" tai "koti/alakerta/kosteus/" ei tilattaisi. (HiveMQ, ei pvm., osa 5)

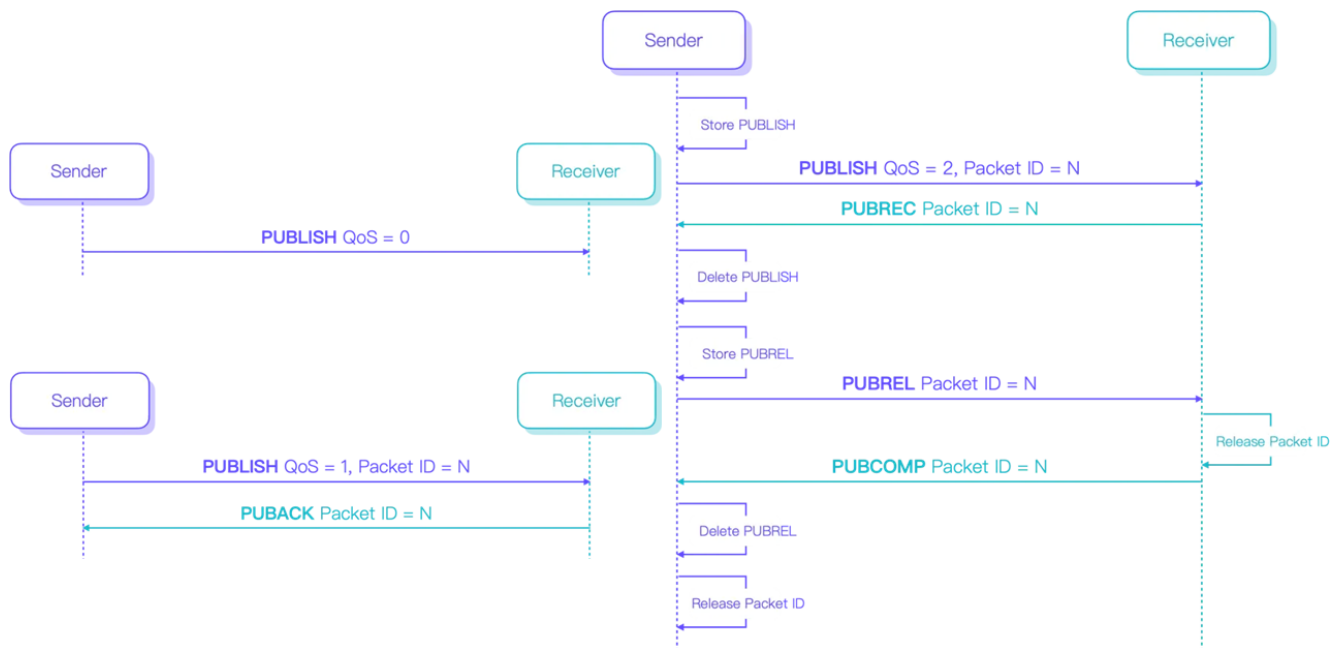
- Risuaita (#)

Voidaan korvata monta aihetasoa, esimerkiksi "koti/alakerta/#". Tällä tilataan kodin alakerran **kaikki** aiheet. Toimiakseen jokerimerkki pitää sijoittaa aiheen loppuun ja sitä pitää edeltää kauttaviiva. (HiveMQ, ei pvm., osa 5)

#### 4.2.5 Palvelun laatutaso

Palvelun laatutasoa käytetään viestien toimittamisen takaamiseen. Laatutasolla voidaan kontrolloida viestin tärkeyttä erottaen kriittiset viestit tavallisista. Kuten kuvassa 8 nähdään, tasoja on kolme, joiden toimituksen luotettavuus nousee, mitä isompi taso on kyseessä. Luotettavuuden noustessa, myös tietoliikenne kasvaa, sillä viestin toimittamiseen tarvitaan enemmän kommunikaatiota puolin ja toisin. Tästä syystä tasojen nopeuksissa on eroja. Viestin kulku on julkaisija-välittäjä-tilaaja. (Zhou, 2023)

Kuva 8. Eri laatutasojen kommunikaatio (Zhou, 2023).



- **Laatutaso 0**

Kuvassa 8 vasemmalla ylhäällä olevassa tasossa viesti lähetetään enintään kerran, sitä ei tallenneta uudelleenlähetystä varten, eikä sen vastaanottajalta odoteta kuittausta. Viesti ei välttämättä mene perille. Taso sopii tilanteisiin, joissa viestejä tulee nopealla aikavälillä, eikä jokaista päivitystä tarvita toiminnan takaamiseksi. (Zhou, 2023)
- **Laatutaso 1**

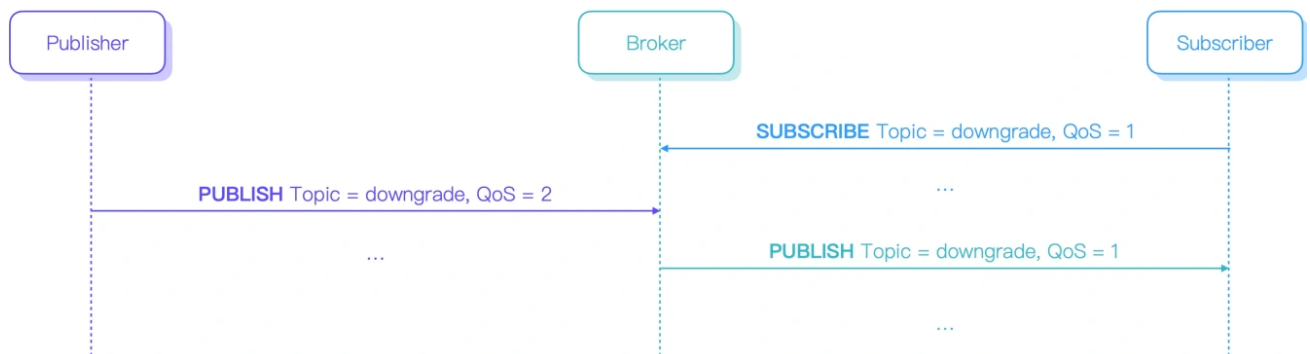
Kuvassa 8 vasemmalla alhaalla olevassa tasossa viesti lähetetään vähintään kerran ja sen vastaanottajalta odotetaan kuittaus. Viesti tallennetaan muistiin, kunnes sen vastaanotosta saadaan kuittaus. Viesti voidaan vastaanottaa mahdollisesti useammin kuin kerran, sillä se lähetetään uudelleen, jos kuittausta ei saada syystä tai toisesta. Taso sopii tilanteisiin, joissa viestin vastaanotto halutaan varmistaa ja vastaanottaja kykenee käsittelemään mahdolliset duplikaatit. (Zhou, 2023)
- **Laatutaso 2**

Kuvassa 8 oikealla olevassa tasossa viesti lähetetään ainoastaan kerran ja sen kulkua seurataan neliosaisen kuittauksen avulla. Viesti tallennetaan ensimmäisen kuittauksen ajaksi, jonka jälkeen se poistetaan, eikä sitä voida lähettää uudestaan.

Tästä eteenpäin viestin kulkua seurataan PUBREC-, PUBREL- ja PUBCOMP-paketeilla. Paketin tunnistetta ja erillisiä seurantapaketteja käytetään duplikaattien välttämiseksi. Taso sopii tilanteisiin, joissa viestin vastaanotto on kriittistä vain ja ainoastaan kerran, vastaanotto on taattu ja hitaampi kommunikaatio ei haittaa toimintaa. (Zhou, 2023)

Laatutasoa käytetään aiheiden julkaisemisessa ja tilaamisessa, jolloin voi tulla vastaan tilanne, jossa näiden laatutasot eroavat toisistaan. Laatutasoa voidaan pienentää, jotta tilaajan taso vastaisi haluamaansa, mutta ei suurentaa. Julkaisijan tason 1 viesti voidaan välittää tilaajalle tasolla 0, mutta ei tasolla 2. Kuvan 9 esimerkissä julkaisun tason 2 viesti välitetään tilaajalle tasolla 1, koska laatutaso on pienempi, se voidaan välittää tilaajan haluamalla tasolla. Muuten viestit välitetään tilaajille julkaisijan laatutason mukaan. (Zhou, 2023)

Kuva 9. Julkaisun laatutason alentaminen tilauksen laatutasolle (Zhou, 2023).



#### 4.2.6 Tietoturva

MQTT:ssä välittäjän ja asiakaslaitteen yhteys voidaan autentikoida käyttäjänimen ja salasanan avulla. Lisäksi asiakaslaite lähettää välittäjälle uniikin tunnisteensa, jolloin välittäjä tarkistaa pääsytiedot ja vertaa ovatko ne asetettu tunnisteeseen omaavalle asiakaslaitteelle. Lopuksi välittäjä hyväksyy tai hylkää yhteyden. (HiveMQ, 2015, osa 3)

Välittäjä voi rajata asiakaslaitteiden julkaisu- ja tilausoikeuksia tiettyihin aiheisiin. Rajaukseen käytetään erilaisia käyttöoikeustiedostoja, joihin laitteiden oikeudet tallennetaan. Oikeuksiin



sisältyy sallitut aiheet, toiminnot (julkaisu ja tilaaminen) ja palvelun laatu- taso. (HiveMQ, 2015, osa 4)

MQTT:ssä viestien sisältö ei ole suoraan salattu, joten viestit kulkevat verkossa luettavana tekstinä. Viestit voidaan salata käyttämällä TLS/SSL-salausprotokollaa. Protokollan toiminta perustuu julkisiin ja yksityisiin avainpareihin, joita käytetään viestien salaukseen ja purkamiseen. Välittäjä- ja asiakaslaite sopivat yhteisestä salausavaimesta, jolloin niiden väliset viestit salataan ennen lähettämistä. Välittäjät ja asiakaslaitteet voivat käyttää X.509-varmenteita, joilla laitteet voivat todistaa ”henkilöllisyytensä”. Varmenteilla laitteet voivat olla varmoja, että ne kommunikoivat oikeiden laitteiden kanssa. (HiveMQ, 2015, osa 5–6)

Asiakaslaitteet voivat lisäksi salata viestiensä tietosisällön, jolloin vain lähettävä – ja päätelaite osaavat lukea sen. Viestin muu sisältö, kuten aihe ja palvelun laatu- taso, pysyy samana, jotta välittäjä osaa reitittää sen oikein. Salaus tapahtuu julkisten ja yksityisten avaimien avulla. (HiveMQ, 2015, osa 8)

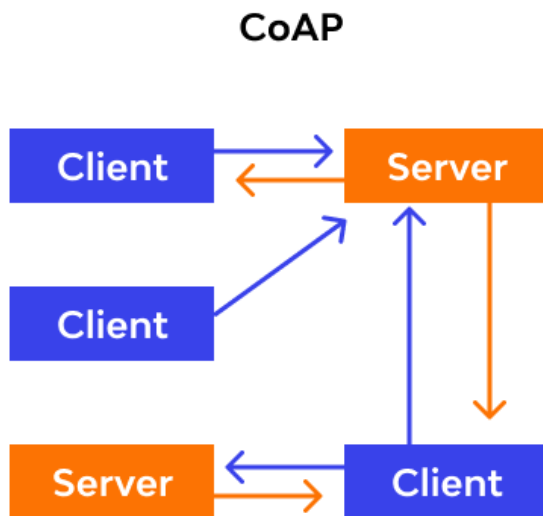
### **4.3 CoAP (Constrained Application Protocol)**

CoAP on IoT:ssa käytetty UDP-protokollaa hyödyntävä avoin dataprotokolla, joka on suunniteltu erityisesti toimimaan resurssirajoitteisissa laitteissa, näin soveltuen hyvin myös IoT-laitteisiin. CoAP:in käyttämä UDP-protokolla on TCP-protokollaa kevyempi, mutta ei mahdollista yhtä luotettavaa tiedonsiirtoa. (Beshkov, 2023)

#### **4.3.1 Toiminta**

CoAP-protokollan toiminta perustuu kuvan 10 mukaiseen asiakas-palvelin-malliin. Mallissa asiakaslaite lähettää CoAP-pyyntönsä palvelinlaitteelle, johon palvelinlaite vastaa CoAP-vastauksella. (Beshkov, 2023)

Kuva 10. CoAP asiakas-palvelin-malli (Beshkov, 2023).



CoAP-protokollan kommunikatio tapahtuu HTTP:n kaltaisella pyyntö-vastaus-mallilla, jossa asiakaslaitteet lähettävät pyyntöjä ja saavat vastauksia palvelinlaitteilta. Erona HTTP-pyyntöihin CoAP-pyyntöt toimivat asynkronisesti, eli laite voi lähettää ja vastaanottaa viestejä samanaikaisesti ilman vastauksen odottamista. (Azzola, 2018)

#### 4.3.2 Viestit

CoAP-viestien toiminta jaetaan kahteen tasoon, jossa yksi taso hoitaa UDP-protokollan vaatimukset ja toinen itse viestisisällön, eli pyynnön tai vastauksen. Jokainen viesti sisältää ID:n, jonka avulla se voidaan tunnistaa tietoliikenteessä. ID:llä voidaan välttää myös mahdollinen viestien monistuminen. (Azzola, 2018)

UDP-tason viestejä on neljää eri tyyppiä:

- CON-viesti (*eng. Confirmable*)

Luotettava viesti, jonka vastaanottajan tarvitsee vahvistaa vastaanotetuksi. Lähettäjä odottaa vastaanottajalta ACK-viestin, jolla sen vastaanotto vahvistetaan. Viesti lähetetään aina uudelleen, kunnes vastaanottaja kuittaa sen ACK-viestillä. CON- viestejä on hyvä käyttää tilanteissa, joissa lähetetään kriittisempää dataa ja tietoliikenteen määrän lisääntyminen ei haittaa toimintaa. (Azzola, 2018)

- NON-viesti (*eng. Non-Confirmable*)  
Epäluotettava viesti, jonka vastaanottajan ei tarvitse vahvistaa vastaanotetuksi. Lähettäjä ei odota ACK-viestiä, eikä sitä lähetetä uudelleen. NON-viestejä on hyvä käyttää tilanteissa, joissa jokaisen viestin vastaanotto ei ole tarpeellista ja nopeus on tärkeää. (Azzola, 2018)
- ACK-viesti (*eng. Acknowledgement*)  
Vastaus CON-viestiin, jolla sen vastaanotto vahvistetaan lähettäjälle. ACK-viestiin sisällytetään sama ID, kuin alkuperäisessä CON-viestissä. (Azzola, 2018)
- RST-viesti (*eng. Reset*)  
Viestiä käytetään tilanteissa, joissa aikaisemmin lähetetty viesti on epäkelpo tai sitä ei pystytä syystä tai toisesta käsittelemään. RST-viestin vastaanotettua lähettäjän on nollattava yhteys. RST-viestiin sisällytetään virhetilanteen aiheuttavan viestin ID, jolloin lähettäjä tietää, mikä viesti virheeseen johti. (Azzola, 2018)

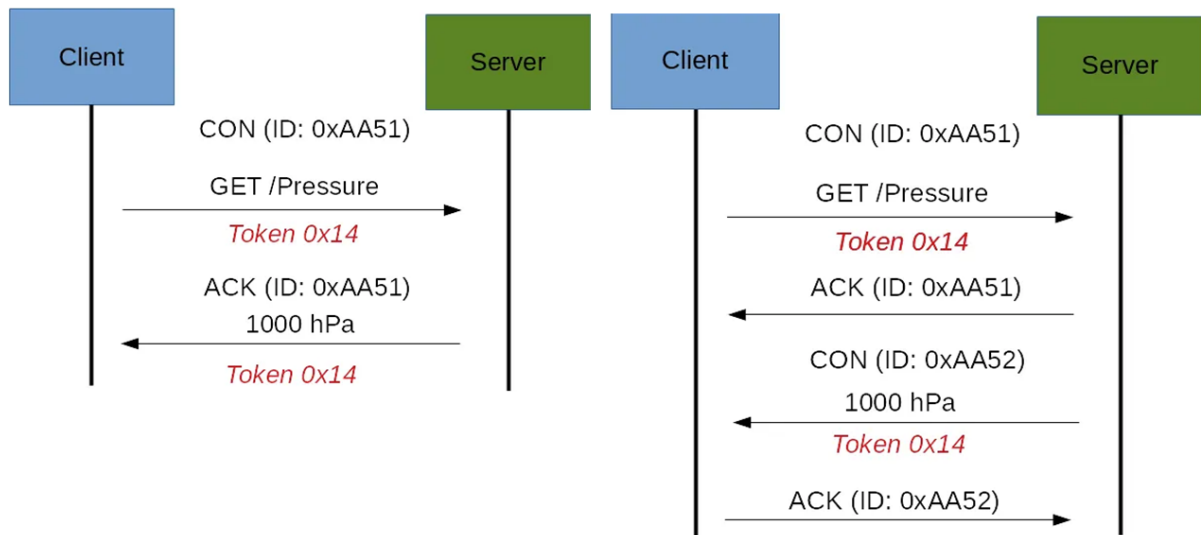
Pyyntö- ja vastausviestit ovat osa CoAP-viestin kokonaisuutta, joka sisällytetään CON- tai NON-viesteihin. Pyyntö- ja vastausviesteihin sisältyy HTTP:n kaltaisia koodeja, joilla pyynnön menetelmä tai vastauksen tila tunnistetaan. Menetelmiä ovat GET (haku), POST (tallennus), PUT (muokkaus) ja DELETE (poisto). Vastauksen tilakoodeja ovat esimerkiksi 2.xx alkuiset onnistumisviesteille, 4.xx asiakaspuolen virheviesteille ja 5.xx palvelinpuolen virheviesteille. (Jakhar, 2020)

Viesteihin sisällytetään lisäksi uniikki tunnus (*eng. token*), joka toimii samanlailla kuin ID, mutta sitä käytetään vastauksen tunnistamiseen pyynnöissä. Tunnus on tarpeellinen, sillä jotkin pyyntöviestit, kuten GET, voivat vaatia useamman kuin yhden vastauksen. Tunnuksella tunnistetaan viestit, jotka kuuluvat samaan vastausketjuun. (Azzola, 2018)

Palvelinlaite vastaa pyyntöön ja sisällyttää vastauksen CON- tai ACK-viestiin, riippuen voiko se vastata heti vai vasta myöhemmin. Kuvassa 11 vasemmalla palvelinlaite vastaa heti, jolloin vastaus sisällytetään suoraan ACK-viestiin, eli pyyntöön vastataan ja se vahvistetaan samassa viestissä. Kuvassa 11 oikealla palvelinlaite ei voi syystä tai toisesta vastata pyyntöön heti. Se lähettää ensin tyhjän ACK-viestin ja pystyessään vastaamaan uuden CON-viestin.

Uuteen CON-viestiin odotetaan myös vastaanottoviesti. Jos pyyntö lähetetään NON-viestinä, siihen vastataan NON-viestillä. (Azzola, 2018)

Kuva 11. CoAP-pyyntöihin vastaaminen (Azzola, 2018).



#### 4.3.3 Tietoturva

CoAP voi käyttää DTLS-salausprotokollaa viestien salaukseen, joka on avaimiin ja varmenteisiin perustuva protokolla. Laitteet voivat käyttää eri turvallisuustasoja salauksessa, jotka ovat NoSec, PreSharedKey, RawPublicKey ja Certificate. Turvallisuustasoissa eroja on siinä, käytetäänkö viesteissä salausta ollenkaan, jaettuja avaimia, asymmetrisia avaimia ilman varmenteita vai X.509-varmenteiden kanssa. Avaimilla viestit salataan ja varmenteilla todistetaan laitteiden "henkilöllisyys". (Jakhar, 2020)

Erona MQTT:hen CoAP ei itsessään tue autentikaatioon tai autorisaatioon liittyviä mekanismeja, mutta ne voidaan toteuttaa muilla tavoilla. Eli CoAP ei pysty suoraan käyttämään esimerkiksi käyttäjänimeä ja salasanaa laitteiden tunnistamiseen tai käyttöoikeuslistoja laitteiden oikeuksien rajaamiseen. (Jakhar, 2020)

## 5 Käytetyt ohjelmistot ja laitteet

### 5.1 Home Assistant

Home Assistant (HA) on avoimen lähdekoodin älykodin ohjauskeskuksen ohjelmisto, joka on kirjoitettu Pythonilla. Ohjelmistona HA toimii useilla eri alustoilla, kuten Raspberry Pi, Windows, Linux ja Docker, joten se voidaan asentaa omien käyttötarkoitusten mukaan mihin vain sen toimintoja mahdollistavaan laitteeseen. HA:n käyttöjärjestelmäversio voidaan myös asentaa virtuaalikoneelle. Se tukee monia eri protokollia, kuten ZigBee, Z-Wave ja Bluetooth, joilla laitteita voidaan integroida järjestelmään. Tällä hetkellä HA tukee suoraan yli 2000 eri laitteen integraation. (Colman, 2022)

HA on suunniteltu paikalliseksi hallintaratkaisuksi, eli se toimii kodin verkossa ilman pilvipalveluita tai internetyhteyttä. Tämä auttaa luomaan tietoturvallisen ja yksityisen ratkaisun, sillä tietoja voidaan hallita ilman niiden lähetystä ulkopuolisille palveluille. Vaikka HA on pääasiassa paikallinen ratkaisu, sitä voidaan käyttää internetyhteyden välityksellä Nabu Casa -pilvipalvelun tai erilaisten DNS-palveluiden avulla. Nabu Casa on HA:n oma maksullinen tilauspohjainen pilvipalvelu, joka mahdollistaa etäkäytön ja HA:n yhdistämisen muihin ohjauskeskuksiin, kuten Amazon Alexaan. HA suosittelee Duck DNS -palvelun käyttöä, jonka avulla HA:an voidaan konfiguroida oma etäohjaus. (Colman, 2022)

HA:n käyttöliittymä on selainpohjainen ja hyvin muokattavissa. Se voidaan jakaa omien käyttötarkoitusten mukaan eri osioihin ja näkymiin. Käyttöliittymässä on myös saatavilla karttanäkymä, jossa laitteiden sijainti eri huoneissa voidaan visualisoida. Laitteita voidaan myös ohjata suoraan tästä karttanäkymästä. (Colman, 2022)

### 5.2 Raspberry Pi 3 Model B+

Raspberry Pi on yhden piirilevyn tietokone, jonka tarkoituksena on mahdollistaa halpa ja helppo tapa oppia ja testata tietotekniikkaa. Raspberry Pi toimii samanlaisella kuin tietokone, mutta se on pienempi ja sen suorituskyky on rajallisempi. Raspberry Pi käyttää siihen optimoitua Debian-jakeluun perustuvaa Linux-käyttöjärjestelmän versiota nimeltään Raspberry Pi OS, mutta siihen voidaan asentaa myös muita Linux-käyttöjärjestelmiä.

Raspberry Pi:stä on monia eri malleja eri ominaisuuksilla ja liitännöillä. Joihinkin malleihin kuuluvat GPIO-pinnit. GPIO-pinnit ovat fyysisiä liittimiä, joilla voidaan luoda yhteys erilaisiin elektroniisiin laitteisiin, sekä komponentteihin ja kontrolloida niitä. (Opensource, ei pvm.)

Raspberry Pi 3 malli B+ (kuva 12) sisältää 64-bittisen Broadcom ARMv8 -neliydinprosessorin 1,4 GHz kellotaajuudella, 1 GB keskusmuistin, Micro-SD-korttipaikan, 4 USB 2.0 -porttia, HDMI-portin, 300 Mbit Ethernet LAN -portin, Wi-Fi-ac, Bluetooth 4.2, Micro-USB-portti virtajohdolle, 3,5 mm yhdistetty -liitin äänelle/kuvasignaalille ja 40-pinninen GPIO-liitin. (Gigantti, ei pvm.)

Kuva 12. Raspberry Pi 3 Model B+ (Raspberry Pi, ei pvm.).



### 5.3 Shelly Smart Plug

Shelly Smart Plug (kuva 13) on älypistorasia, jonka avulla siihen kytkettyjä sähkölaitteita voidaan kytkeä päälle tai pois automaattisesti. Älypistorasia pystyy myös mittaamaan energiankulutusta siihen kytketyistä sähkölaitteista. Älypistorasia toimii liittämällä se kodin Wi-Fi-verkkoon ja ohjaamalla sitä puhelimeen saatavan sovelluksen kautta. Erillistä ohjauskeskusta ei tarvita, mutta se on yhteensopiva Google Assistant tai Amazon Alexa - ohjauskeskusten kanssa. (Shelly, ei pvm.-b)

Shelly käyttää oletuksena CoAP-protokollaan perustuvaa ColoT-protokollaa kommunikointiin, mutta tukee myös MQTT-protokollaa. ColoT-protokolla on Shellyn luoma, jossa kaikki viestit ovat JSON-muodossa. (Shelly, ei pvm.-a)

Kuva 13. Shelly Smart Wi-Fi-älypistorasia (Shelly, ei pvm.-b).



#### 5.4 WiZ LED A60 full color -lamppu

WiZ LED A60 full color -lamppu (kuva 14) on WiZ-tuoteperheeseen kuuluva led-lamppu, joka sopii kodin yleisvalaistukseen. Lampun värit vaihtelevat valkoisen eri sävyistä viileästä lämpimään, sekä 16 miljoonaan muun värin välillä. Lamppu yhdistetään kodin Wi-Fi-verkkoon ja sitä voidaan kontrolloida WiZ Connected -sovelluksella. Erillistä tukiasemaa ei tarvita, mutta se voidaan liittää myös Google Assistant, Amazon Alexa tai Siri Shortcut ääniohjaukseen. Sovelluksen avulla lampun väriä voidaan säätää manuaalisesti, asettaa ajastuksiin perustuvia automatisointeja, sekä automatisoida se vaihtumaan liikkeen perusteella (vaatii kahden tai useamman lampun). (WiZ, ei pvm.)

Kuva 14. WiZ LED A60 full color -lamppu (WiZ, ei pvm.).



## 6 Älykodin rakentaminen

Älykodin rakentaminen aloitetaan Home Assistantin käyttöjärjestelmän asentamisella Pi:hin. HA:n asentamisen jälkeen eri laitteet, sekä älypuhelin, integroidaan toimimaan HA:n sisällä. Tämän jälkeen niihin voidaan rakentaa eri automaatioita.

### 6.1 Home Assistantin asennus

Home Assistantin asennus vaatii mikroSD-kortin formatoimisen ja HA:n käyttöjärjestelmän kirjoittamisen siihen. Formatointi ja kirjoittaminen onnistuu Balena Etcher -ohjelmistolla. Valmisteluosiossa muutetaan HA:n asetuksia ja asennetaan siihen tarvittava File Editor -lisäosa.

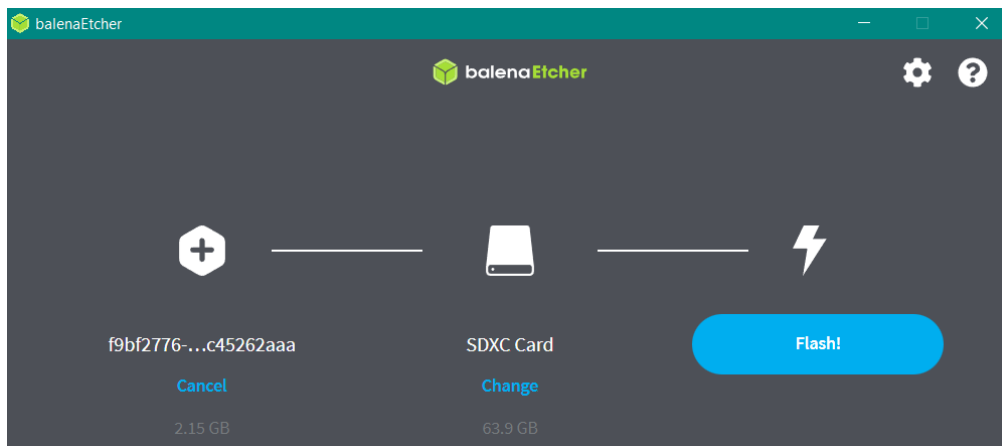
#### 6.1.1 Balena Etcher

Asennusta varten tarvitaan mikroSD-kortti, johon Home Assistantin käyttöjärjestelmä asennetaan. Liitetään mikroSD-kortti tietokoneeseen muistikortinlukijan avulla ja ladataan, sekä asennetaan Balena Etcher ohjelmisto. Balena Etcheriä käytetään HA:n käyttöjärjestelmätiedoston kirjoittamiseen mikroSD-korttiin.

Balena Etcher -ohjelmistoon täytetään tiedot kuvan 15 mukaisesti HA:n käyttöjärjestelmätiedoston kirjoittamista varten. Vasemmalla kuvassa 15 valitaan käyttöjärjestelmätiedosto. Tiedosto haetaan suoraan HA:n Raspberry Pi 3 GitHub-linkin kautta. Keskellä kuvassa 15 valitaan muistikortinlukijan avulla yhdistetty mikroSD-kortti kirjoittamispaikaksi. Lopuksi kirjoittaminen aloitetaan painamalla kuvassa 15 oikealla näkyvää *"Flash!"* -painiketta.

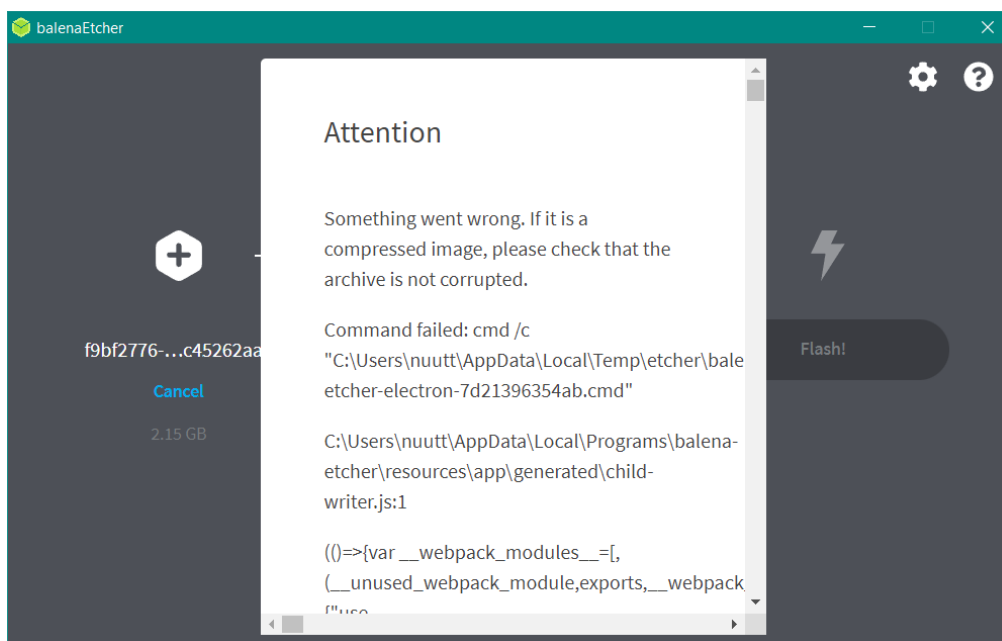


Kuva 15. Balena Etcher ohjelmiston asetukset.



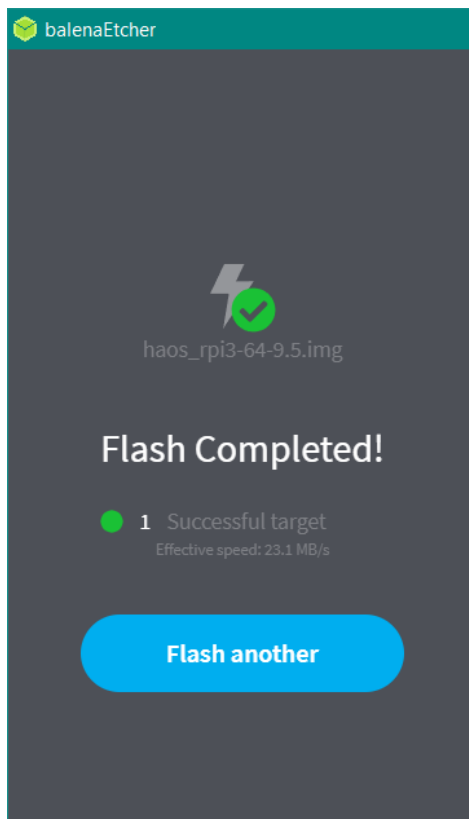
GitHub-linkin kautta kirjoittaminen tuotti kuvan 16 mukaisen virheen, joka korjattiin lataamalla tiedosto saman linkin takaa tietokoneelle ja käyttäen sitä kirjoittamistiedostona.

Kuva 16. Balena Etcher virhe linkin kautta kirjoittamisessa.



Kirjoittamisen onnistuttua saadaan siitä kuvan 17 mukainen viesti. Tämän jälkeen muistikortinlukija voidaan poistaa tietokoneesta ja microSD-kortti siirtää Pi:hin. Liitetään Pi:hin virtajohto ja kytketään se reitittimeen Ethernet-kaapelilla (kuva 18).

Kuva 17. Balena Etcher onnistunut kirjoittaminen.



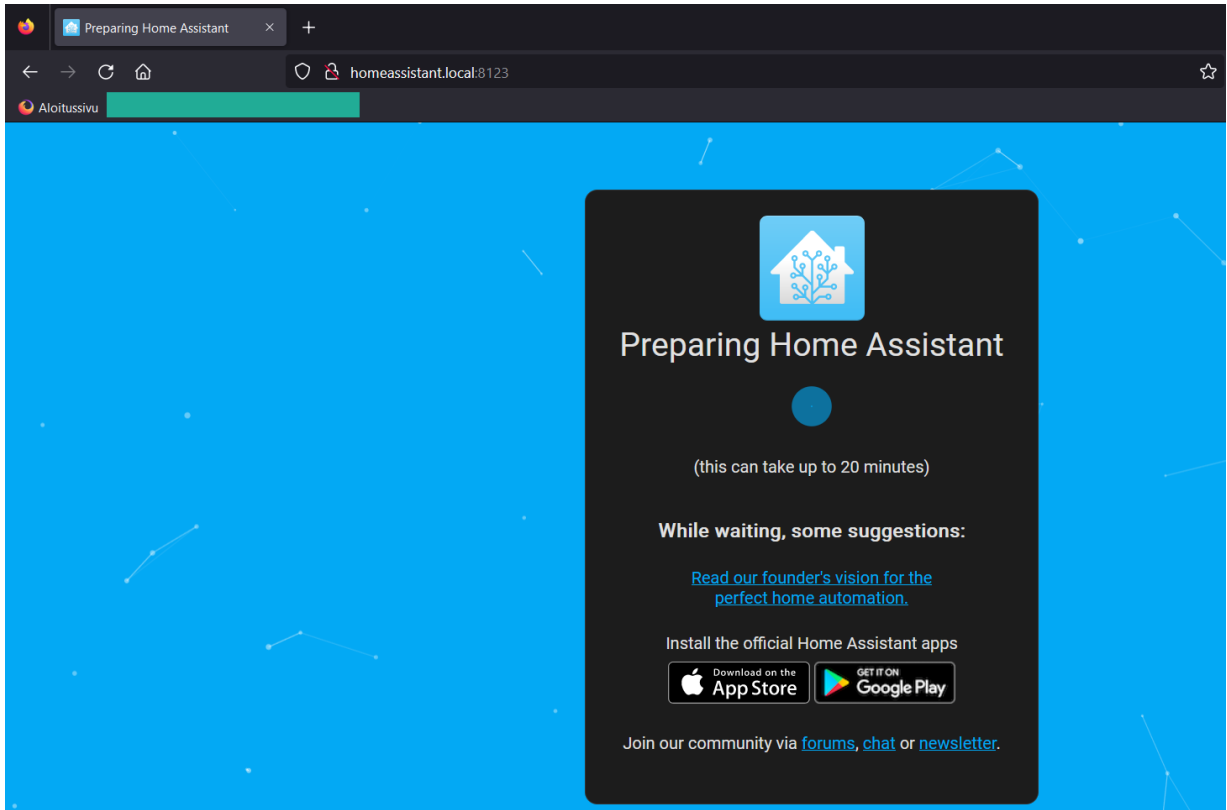
Kuva 18. Raspberry Pi mikroSD -kortilla kytkettynä virtaan ja reitittimeen.



## 6.1.2 Valmistelu

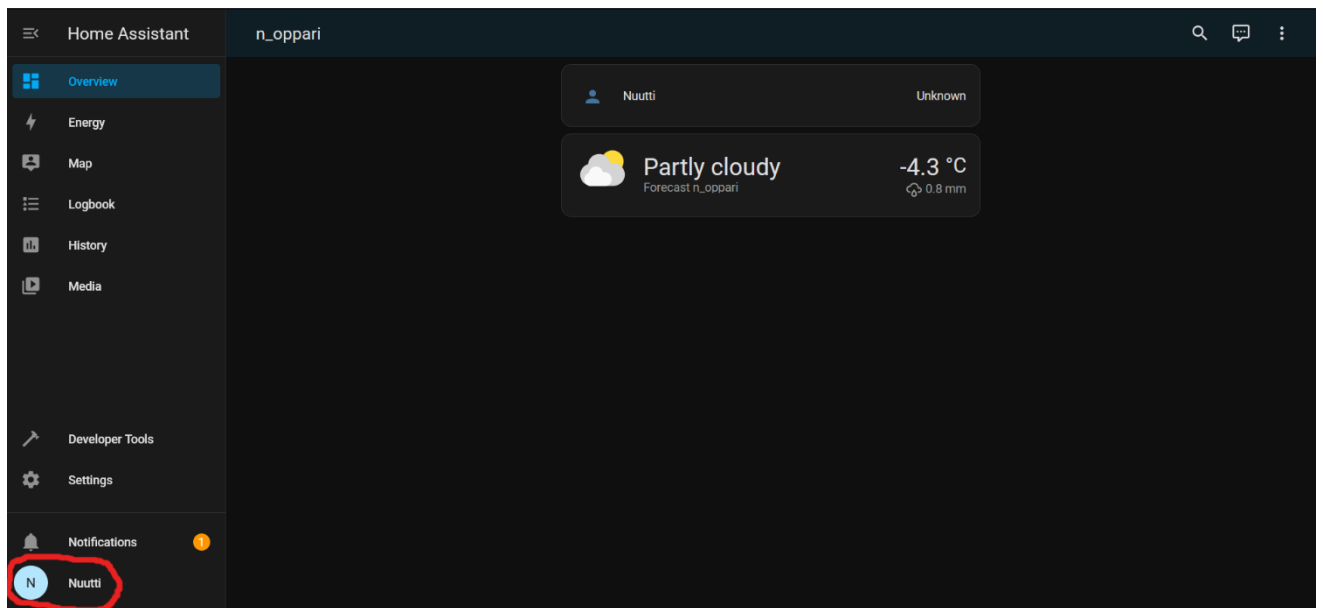
Muutaman minuutin kuluttua HA:an saadaan yhteys selaimen avulla osoitteesta <http://homeassistant.local:8123/>. Kuvassa 19 nähdään osoitteen antama viesti selaimessa, jossa kerrotaan HA:n valmistelusta. Valmisteluun voi mennä jopa 20 minuuttia.

Kuva 19. Home Assistantin valmistelu selaimessa.



Valmistelun jälkeen HA vaatii käyttäjän luomisen, joka toimii HA:n ylläpitäjänä. Luominen vaatii nimen, käyttäjänimen ja salasanan. Seuraavaksi älykodille annetaan nimi, halutessaan sijaintitiedot, joita käytetään säätietojen näyttämiseksi ja aurinkoon perustuvien automaatioiden toteutuksessa, maa, kieli, aikavyöhyke, korkeus merenpinnasta, yksikköjärjestelmä ja valuutta. Halutessaan käyttäjätietoja voidaan jakaa Home Assistantille. Viimeiseksi HA tunnistaa mahdollisesti Wi-Fiin jo liitetyt laiteintegraatiot ja kysyy niiden määrittämisestä. Lopuksi päästiin Home Assistantin alkunäkymään (kuva 20).

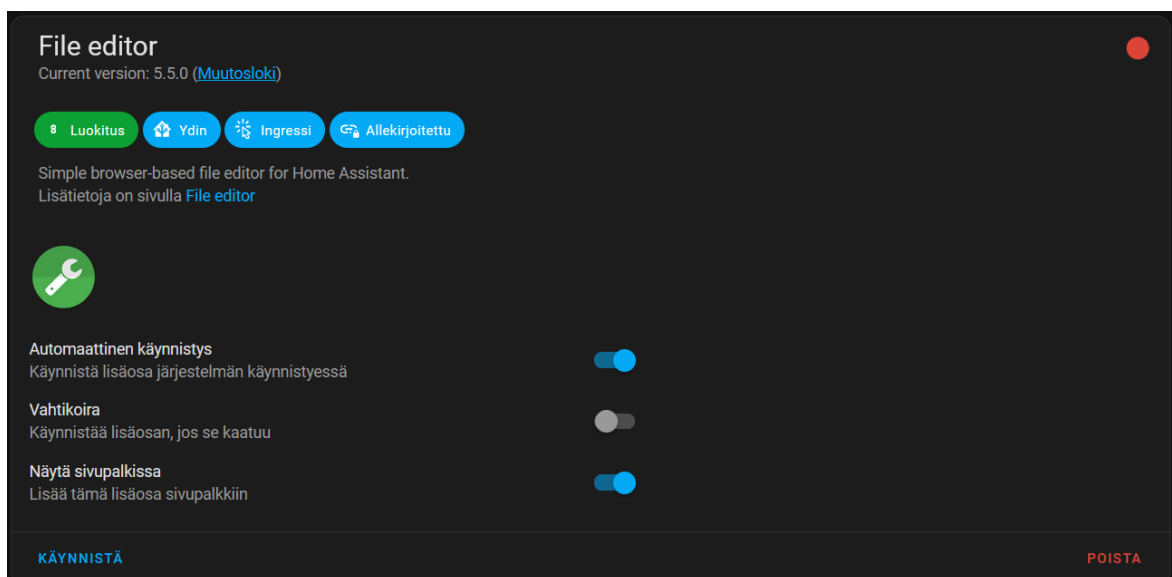
Kuva 20. Home Assistantin alkunäkymä asennuksen jälkeen.



Kuvassa 20 alavasemmalla punaisella ympyröidyn profiilin alta voidaan muokata asetuksia. Muokataan ajan formaatti, viikon alkamispäivä, teema ja tietoturvan lisäämiseksi kytketään kaksisuuntainen tunnistus päälle käyttäen Google Authenticator -puhelinsovellusta.

Asennetaan File Editor -lisäosa (kuva 21), jotta HA:n konfiguraatitiedostoja voidaan muokata selaimen kautta. Lisäosat löytyvät asetuksien alta. Otetaan lisäosa näkyviin sivupalkissa klikkaamalla "Näytä sivupalkissa" -painiketta.

Kuva 21. File Editor -lisäosa asennettuna.



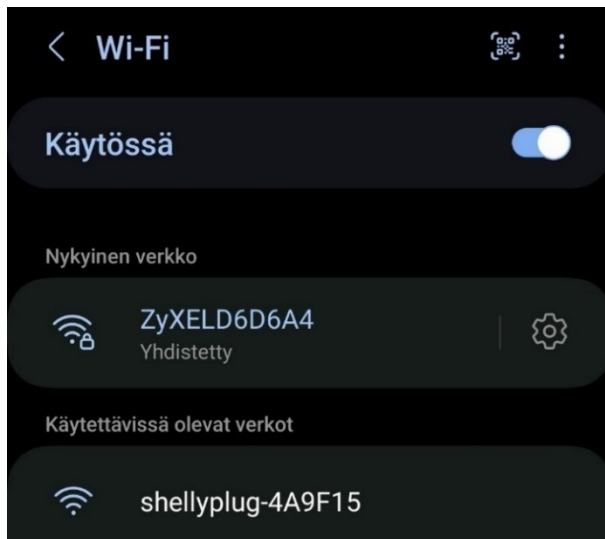
## 6.2 Äylaitteiden lisääminen

Äylaitteiden lisääminen HA:n vaatii ensin niiden lisäämisen kotiverkkoon. Shelly voidaan yhdistää ilman erillistä sovellusta, mutta WiZ vaatii oman sovelluksen. HA tunnistaa suurimman osan laitteista automaattisesti, sen jälkeen, kun ne ovat samassa verkossa HA:n kanssa.

### 6.2.1 Shelly Smart Plug

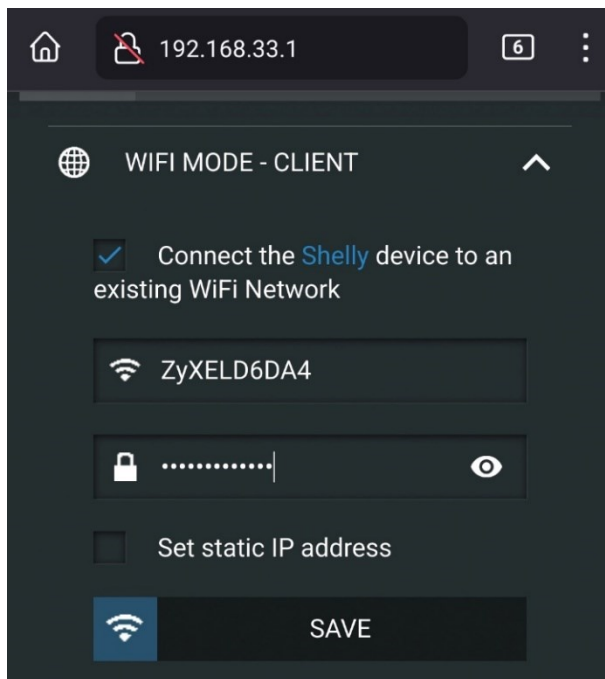
Shelly asennetaan pistorasiaan. Shellyssä olevaa vesipisaran muotoista painiketta pidetään noin 10 sekuntia pohjassa, jolloin se palautuu oletusasetuksilleen ja toimii Wi-Fi tukiasemana. Tämän jälkeen Shelly löytyy puhelimen Wi-Fi-asetuksista (kuva 22). Shellyn luomaan tukiasemaan yhdistetään ja sen käyttöliittymä avataan selaimella osoitteesta *192.168.33.1*.

Kuva 22. Shelly tukiasemana puhelimen Wi-Fi-asetuksissa.



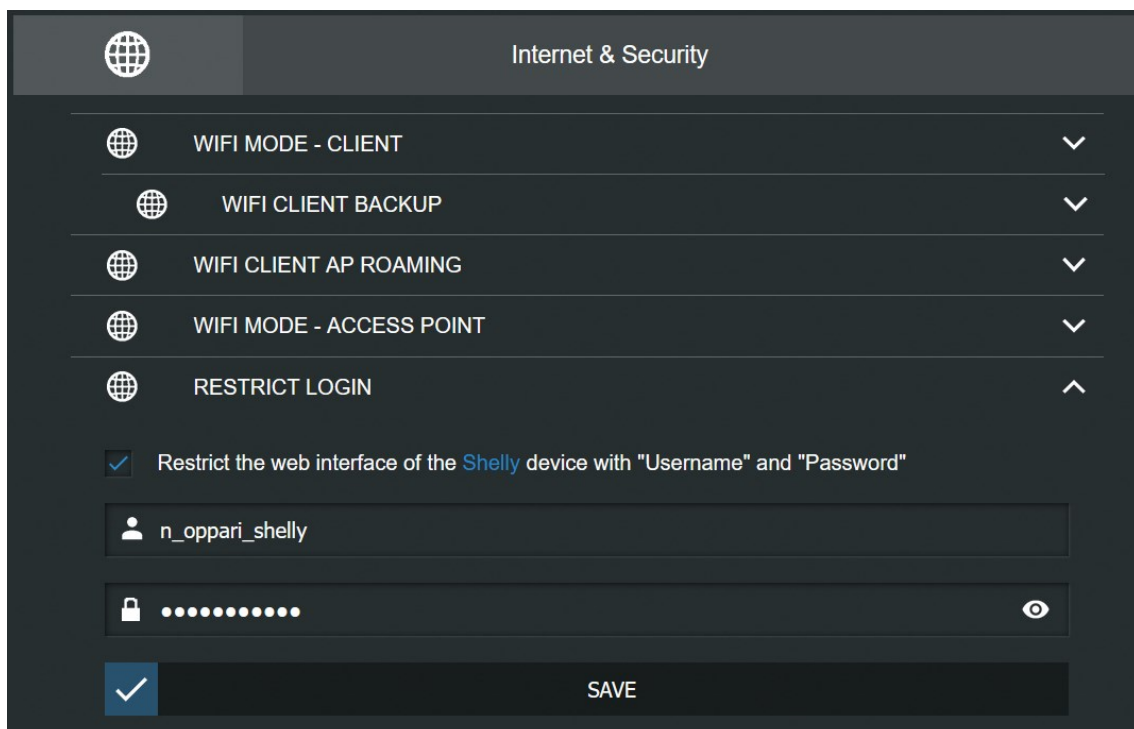
Kuvan 23 mukaisesti, selaimelle avatun käyttöliittymän asetukseen syötetään Wi-Fi-verkon SSID ja salasana, jolloin Shelly yrittää yhdistää verkkoon. Shellyn yhdistettyä onnistuneesti kotiverkkoon, sen uusi IP-osoite tarkistetaan reitittimen asetuksista, jolloin sen käyttöliittymä voidaan taas avata selaimella kyseisestä osoitteesta.

Kuva 23. Shellyn lisääminen kotiverkkoon käyttöliittymästä.



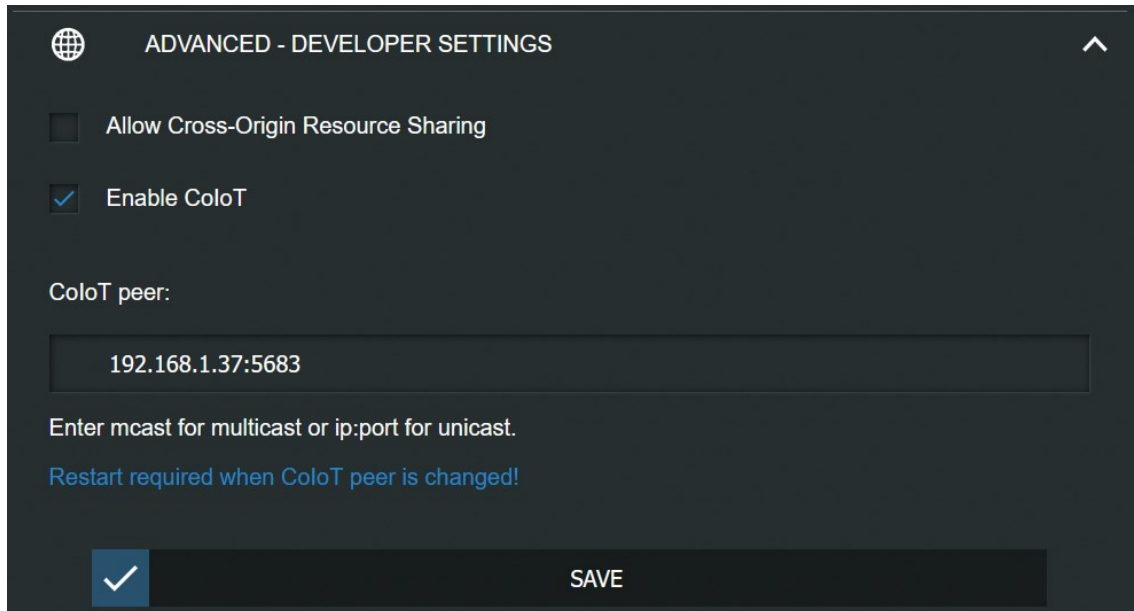
Tietoturvan lisäämiseksi asetetaan "Internet & Security" -välilehden "RESTRICT LOGIN" -kohdasta kuvan 24 mukaisesti käyttöliittymälle kirjautumisvaatimus käyttäjänimen ja salasanan avulla.

Kuva 24. Shelly kirjautumisvaatimuksen asettaminen käyttöliittymästä.



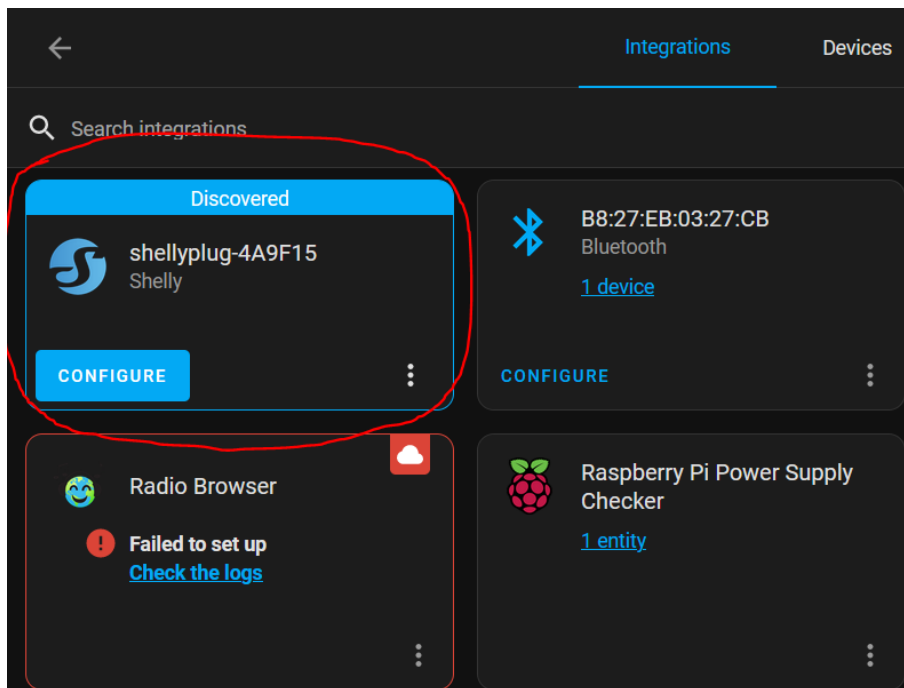
Shellyn integraation toimimiseksi HA:ssa kytketään kuvan 25 mukaisesti "Internet & Security" -välilehden kohdasta "ADVANCED – DEVELOPER SETTINGS" ColoT-asetus päälle. "ColoT peer" -kohtaan annetaan HA:n IP-osoite ja portiksi 5683.

Kuva 25. Shelly ColoT-asetus.

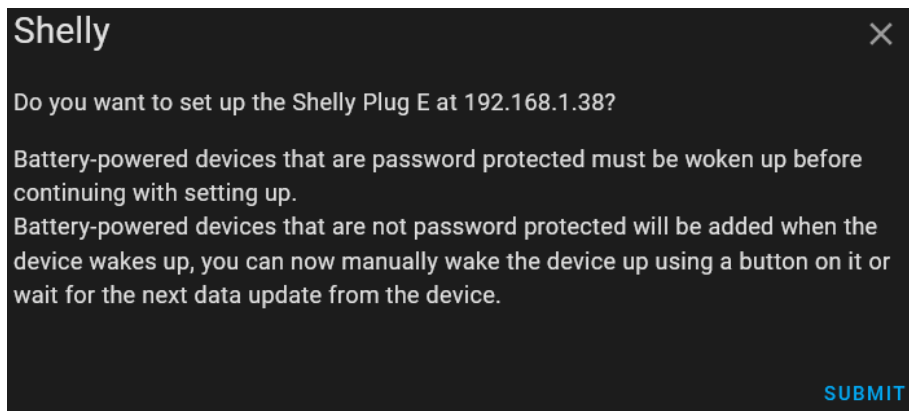


HA:n puolella navigoidaan sivupalkista kohtaan "Asetukset" ja valitaan "Laitteet & palvelut". Kuvan 26 mukaisesti Shelly tunnistetaan automaattisesti. Painetaan "CONFIGURE" -painiketta, jonka jälkeen HA kysyy, halutaanko Shelly lisätä (kuva 27). Painetaan "SUBMIT" -painiketta ja HA ilmoittaa lisäyksen onnistuneen kuvan 28 mukaisesti.

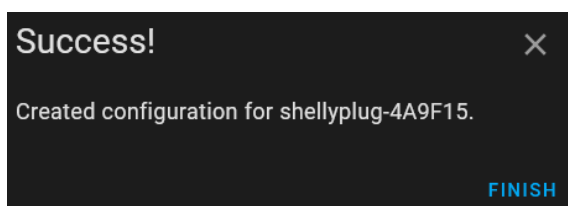
Kuva 26. Shelly integraatio tunnistettuna HA:ssa.



Kuva 27. Shellyn lisäämisen kysely.



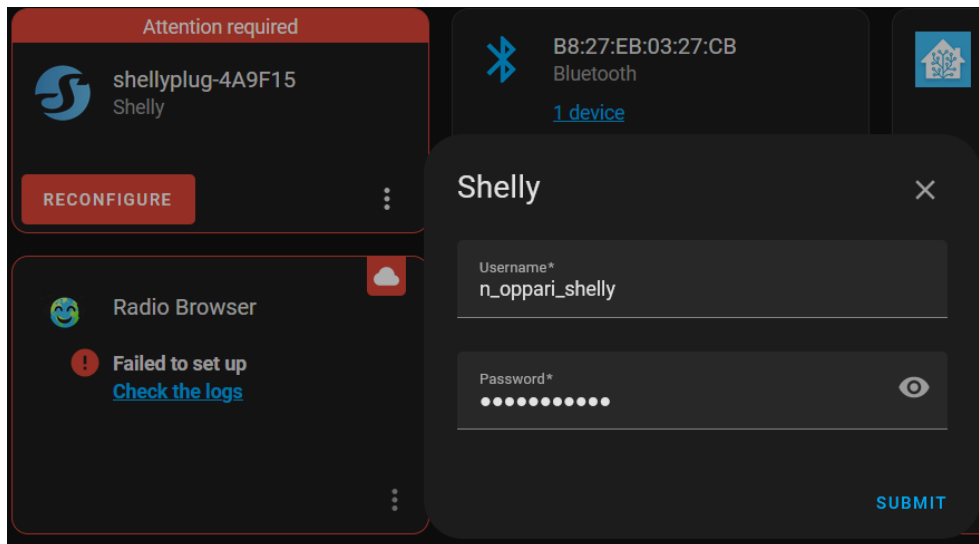
Kuva 28. Shelly lisätty onnistuneesti.





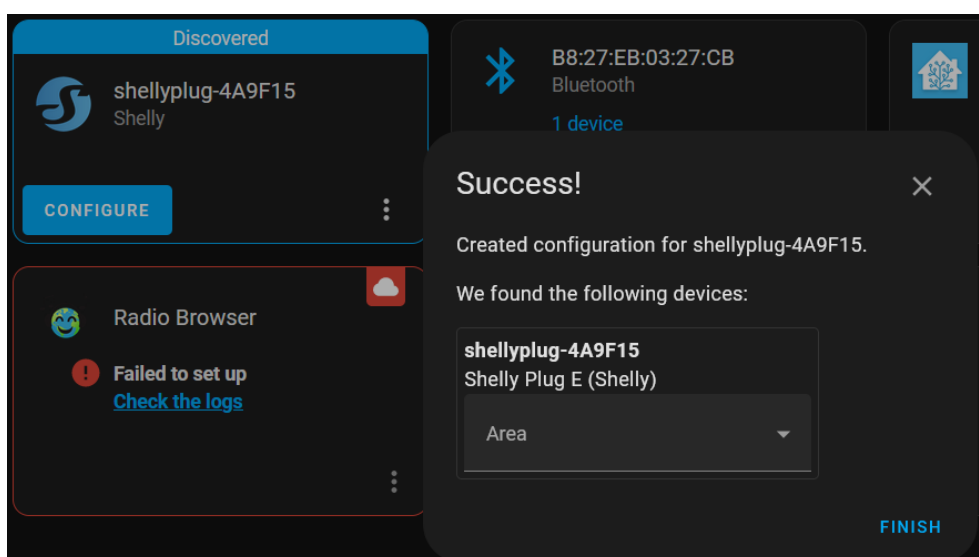
Vaikka HA ilmoitti lisäyksen onnistuneen näin ei kuitenkaan ollut. Painettiin kuvassa 29 näkyvää "RECONFIGURE" -nappia, jonka jälkeen HA kysyi Shellyyn aikaisemmin asetettuja kirjautumistietoja.

Kuva 29. Shellyn uudelleenintegroinnin kirjautumistiedot.



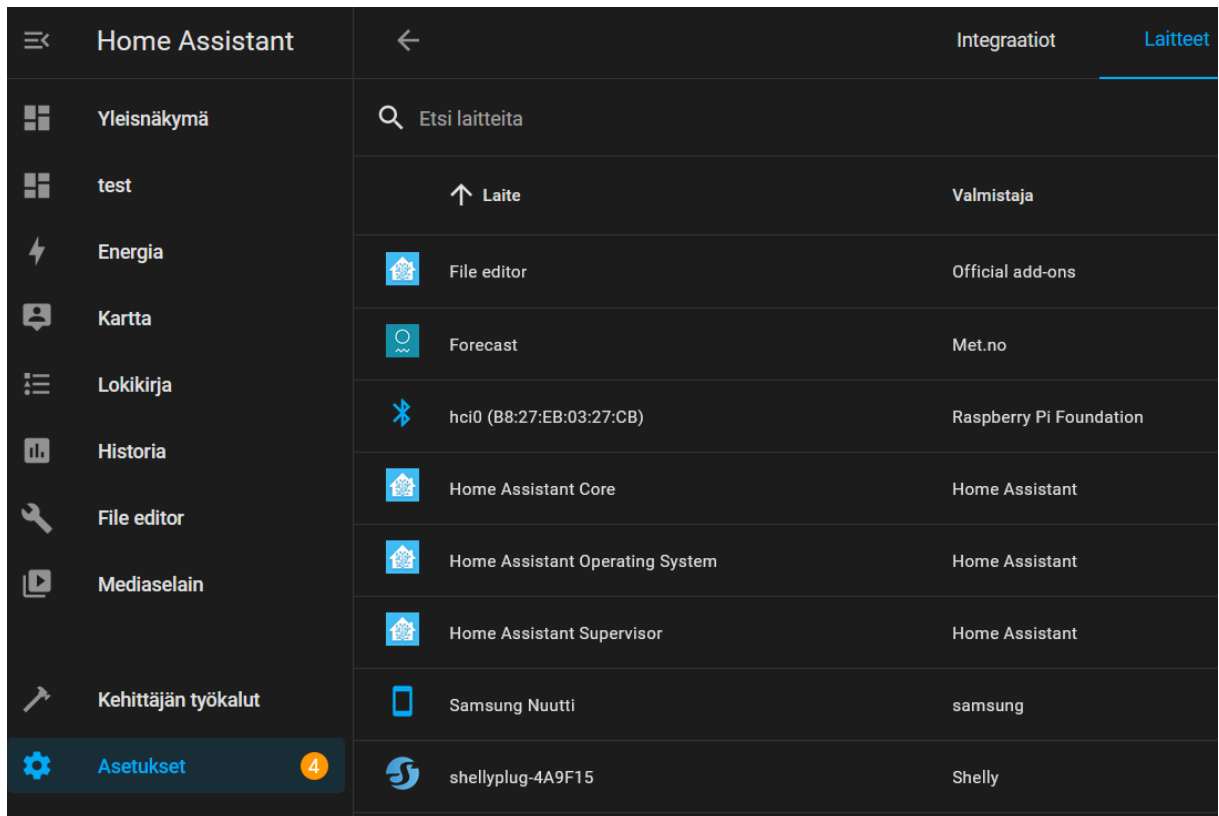
Annettiin kirjautumistiedot ja Shelly lisättiin onnistuneesti. HA kysyy aluetta, mihin se haluttan lisätä (kuva 30).

Kuva 30. Shellyn uudelleenintegrointi onnistuneesti.



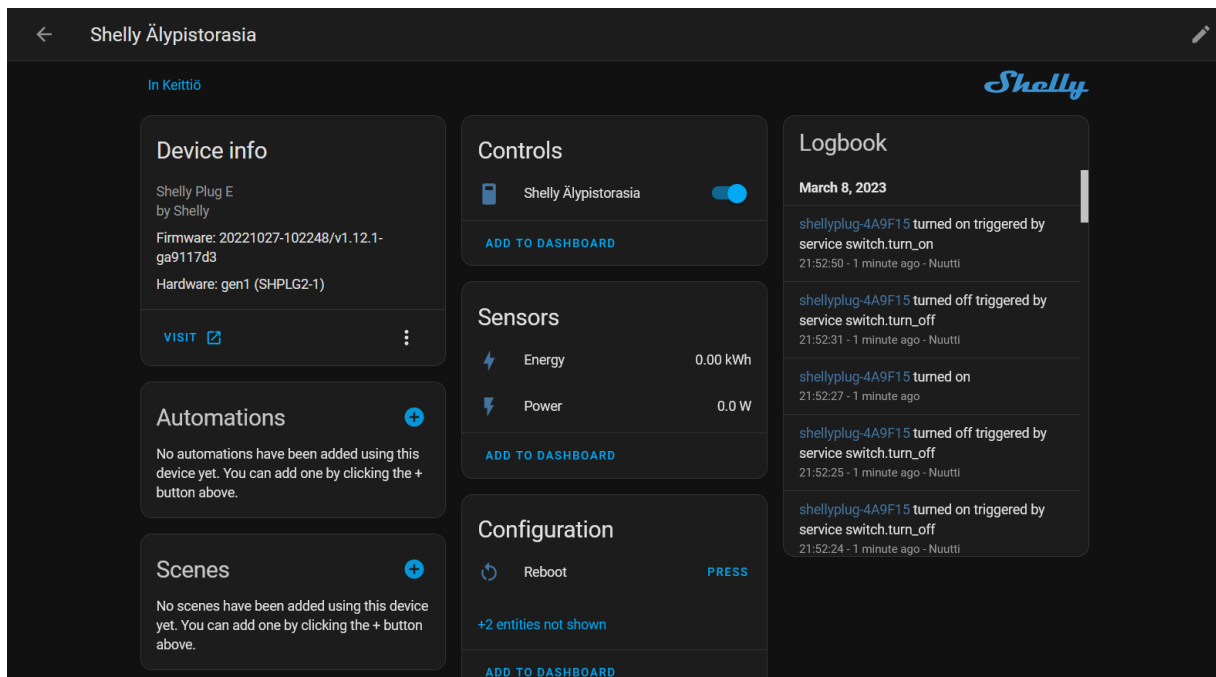
Lopuksi käydään nimeämässä Shelly uudelleen navigoimalla sivupalkista kohtaan ”Asetukset” ja valitsemalla ”Laitteet & palvelut”. Avautuneesta sivusta valitaan ”Laitteet”-välilehti ja listasta Shelly (kuva 31).

Kuva 31. Shelly ”Laitteet & palvelut” -listalla.



Kuvan 32 mukaisesti avautuneesta sivusta nähdään Shellyn kaikki entiteetit ja niiden arvot, kontrollit ja loki. Shellylle annetaan uusi nimi ”Shelly Älypistorasia” painamalla oikeassa ylänurkassa olevaa kynäkuvaketta.

Kuva 32. Shellyn uudelleen nimeäminen.

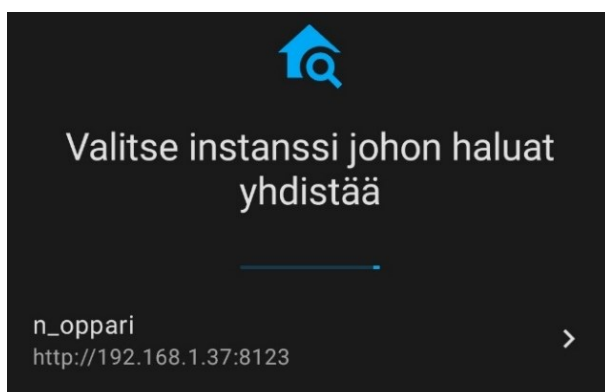


## 6.2.2 Android puhelin

Home Assistant Companion -sovellus ladataan puhelimelle, jotta voidaan saada ilmoituksia puhelimeen ja mahdollistetaan esimerkiksi sijaintiin perustuvien automatisointien luonti.

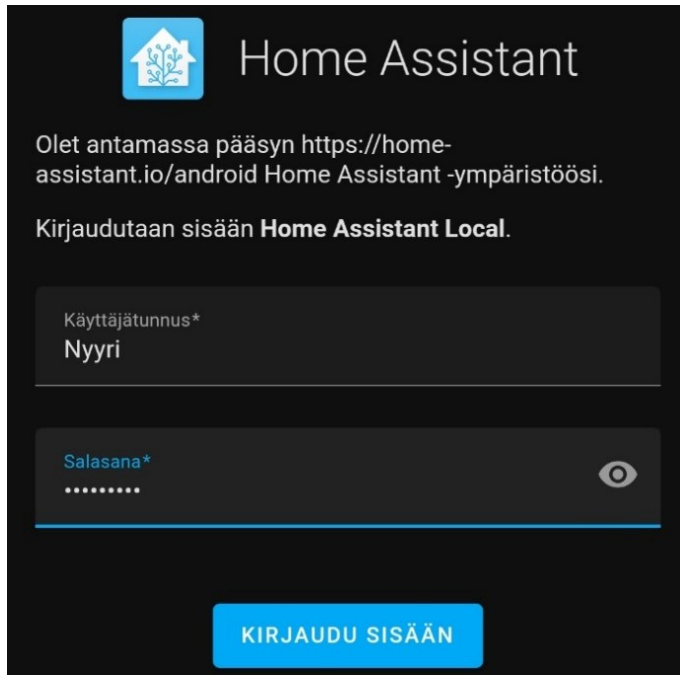
Sovellus voidaan ladata Google Play Storesta. Ensiavauksella (kuva 33) sovellus kysyy HA-instanssin, johon halutaan yhdistää. Puhelimen ollessa HA:n kanssa samassa Wi-Fi:ssä, sovellus löytää HA-instanssin automaattisesti. HA-instanssin URL-osoite voidaan syöttää myös manuaalisesti.

Kuva 33. HA-instanssin yhdistäminen sovellukseen.

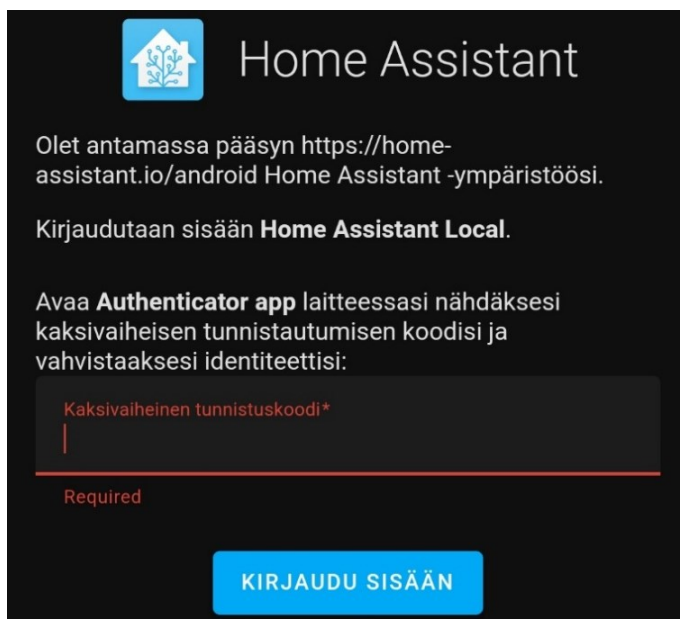


Instanssin valittua sovellus kysyy kirjautumistietoja. Kuvan 34 mukaisesti instanssiin voidaan kirjautua HA:an aikaisemmin luoduilla tunnuksilla. Oikeiden kirjautumistietojen annettua aikaisemmin asetetun kaksivaiheisen tunnistautumisen tunnuskoodi vaaditaan (kuva 35).

Kuva 34. HA-instanssiin kirjautuminen sovelluksessa.

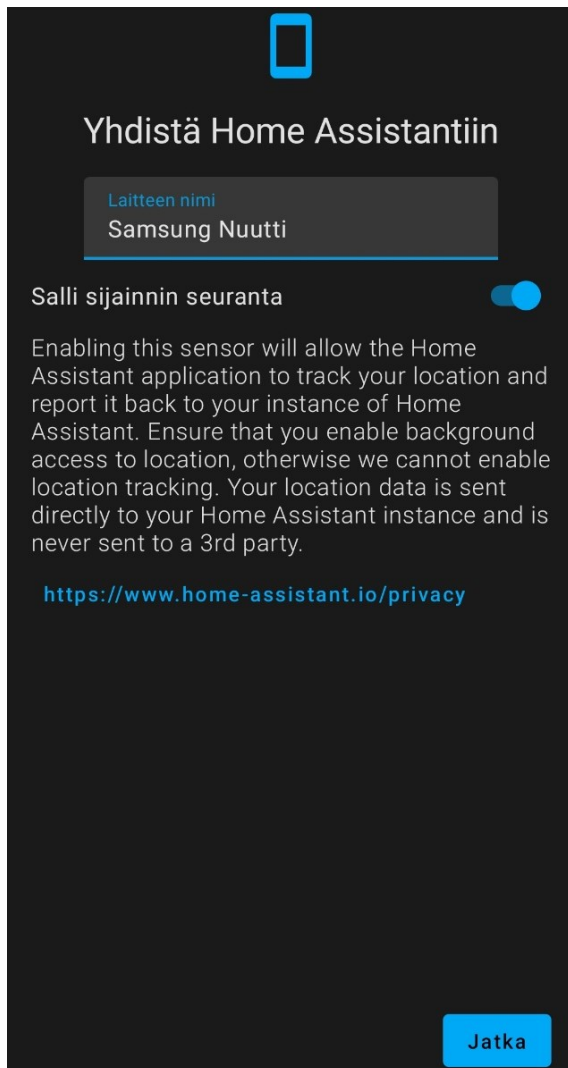


Kuva 35. HA-instanssin kaksivaiheinen tunnistautuminen sovelluksessa.



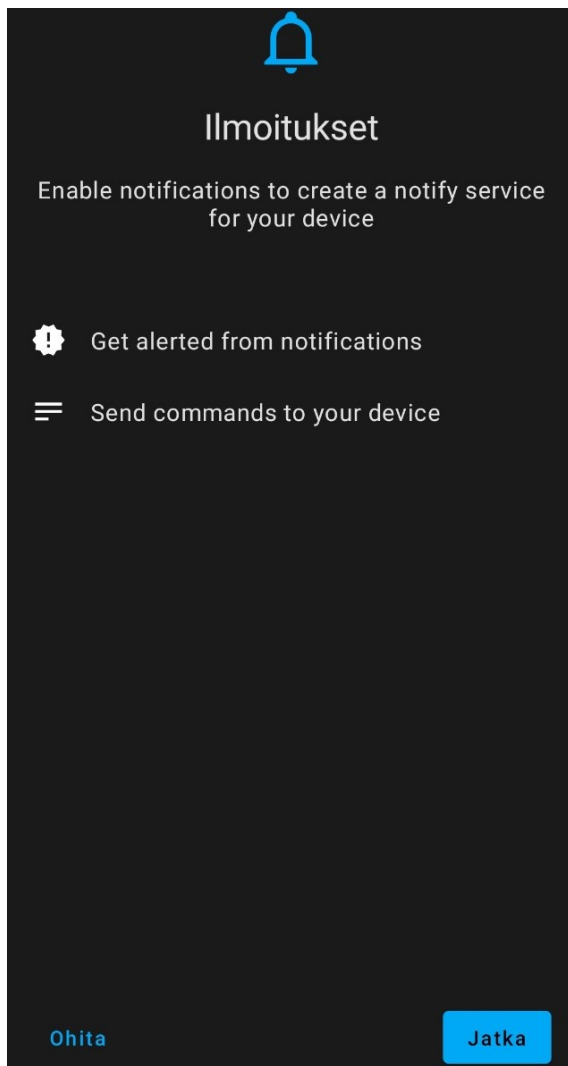
Kirjaututtua sisään, puhelimelle voidaan asettaa kuvan 36 mukaisesti tunnistettava nimi ja halutessa sallia sen sijainnin seuranta GPS:n avulla. Seuranta voidaan käyttää erilaisten automaatioiden luontiin.

Kuva 36. Laitteen nimeäminen ja seurannan salliminen HA sovelluksessa.

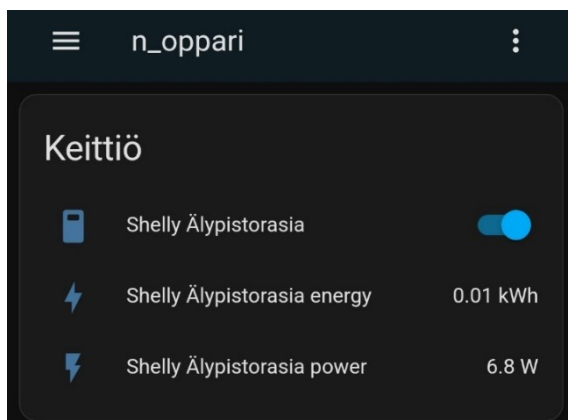


Laitteasetuksien jälkeen sovellus pyytää ilmoitusten päälle kytkemistä (kuva 37). Ilmoitukset kytketään päälle, jotta laitteet voivat lähettää puhelimeen komentoja, sekä tietoa omasta tai automatisaation tilasta. Lopuksi nähdään sovelluksen koontinäyttö (kuva 38).

Kuva 37. HA-instanssin ilmoitusten päälle kytkeminen sovelluksessa.



Kuva 38. HA-instanssin koontinäyttö sovelluksessa.



### 6.2.1 Airam Wi-Fi Smart Bulb

Airamin virallinen HA-integraatio on poistettu käytöstä, joten sitä ei saatu toimimaan. Laite saatiin toimimaan vain Airamin omassa pilvipohjaisessa sovelluksessa.

### 6.2.2 WiZ-älylamppu

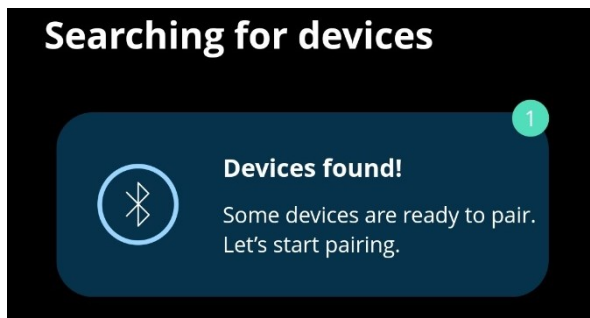
WiZ-lamppu asennetaan kiinni valaisimeen ja se kytketään valokatkaisijasta päälle. WiZ Connected -sovellus ladataan Google Play Storesta puhelimelle, jotta WiZ-lamppu saadaan kytkettyä kotiverkkoon. Ensiavauksella seurataan sovelluksen ohjeita. Hyväksytään käyttöehdot ja tietosuojailmoitus, ohitetaan käyttäjän luonti, luodaan paikallinen profiili, luodaan uusi koti, lisätään kotiin huoneita ja hyväksytään sovelluksen vaatimat käyttöoikeudet. Ensiasennuksen jälkeen päädytään kuvan 39 mukaiseen näkymään, jossa painetaan oikealla ylhäällä sijaitsevaa plusikonia.

Kuva 39. WiZ Connected -sovellus ensiasennuksen jälkeinen näkymä.

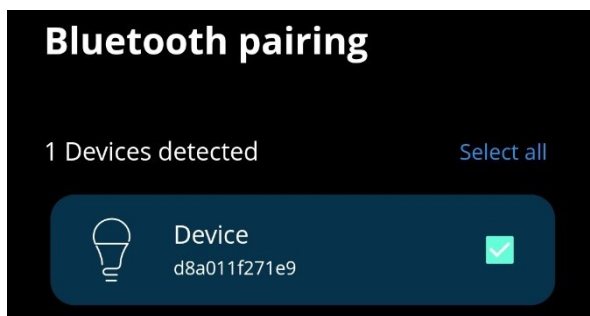


Sovellus alkaa etsimään lähellä olevia laitteita Bluetoothin avulla ja ilmoittaa löytäneensä uuden laitteen (kuva 40). Painetaan ilmoitusta, jolloin sovellus näyttää löydetyn laitteen (kuva 41). Jatketaan painamalla "Start".

Kuva 40. WiZ Connected -sovellus löydetty laite.



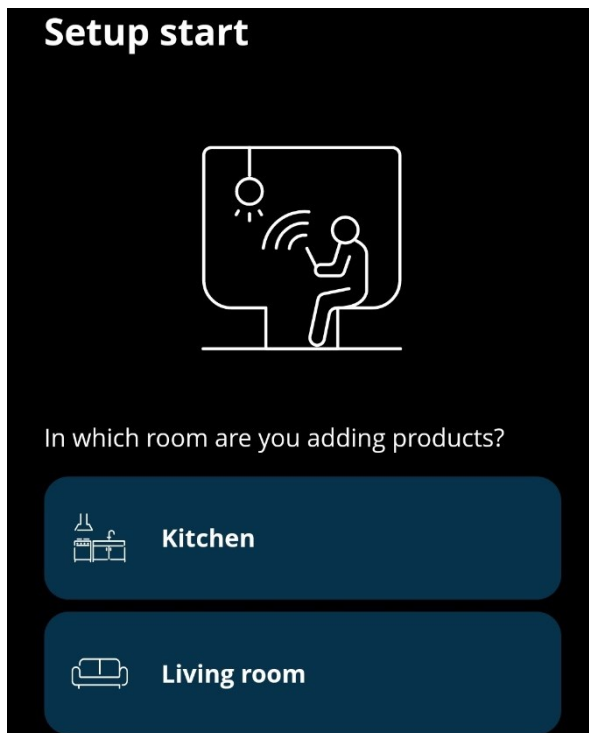
Kuva 41. WiZ Connected -sovellus löydetyt laitteen tyyppi.



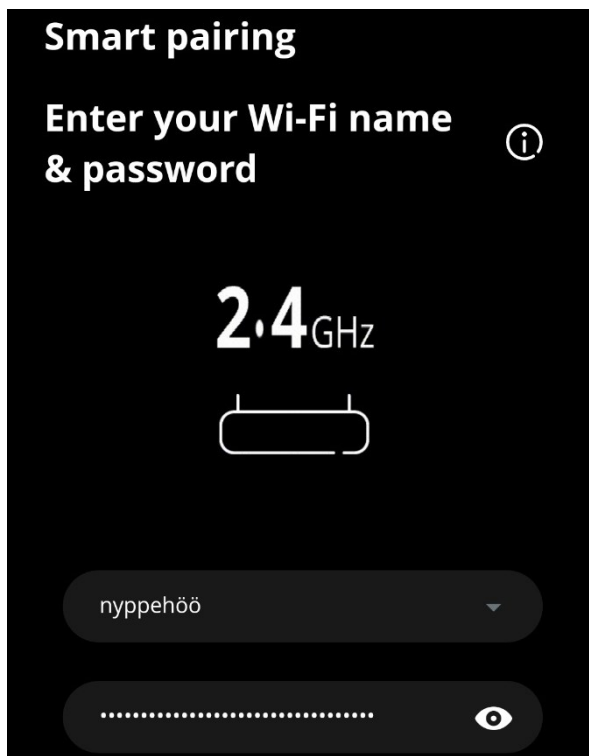
Sovellus kysyy huonetta, johon laite lisätään (kuva 42). Valitaan "Living room", jonka jälkeen sovellus kysyy kotiverkon nimeä ja salasanaa (kuva 43). Jatketaan painamalla "Start".



Kuva 42. WiZ Connected -sovellus laitteen huonevalinta.

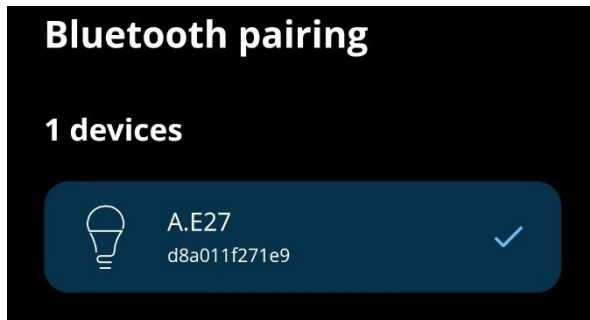


Kuva 43. WiZ Connected -sovellus kotiverkon tiedot.



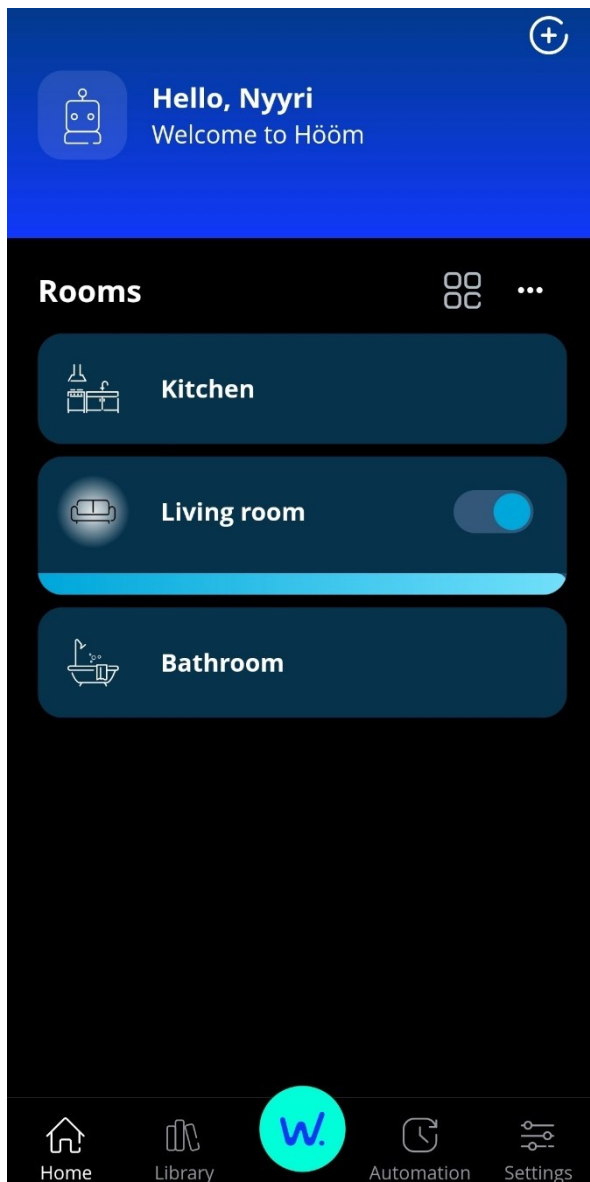
WiZ-lamppu lisätään onnistuneesti kotiverkkoon (kuva 44). Jatketaan painamalla "Done!".

Kuva 44. WiZ Connected -sovellus laitteen onnistunut lisäys kotiverkkoon.



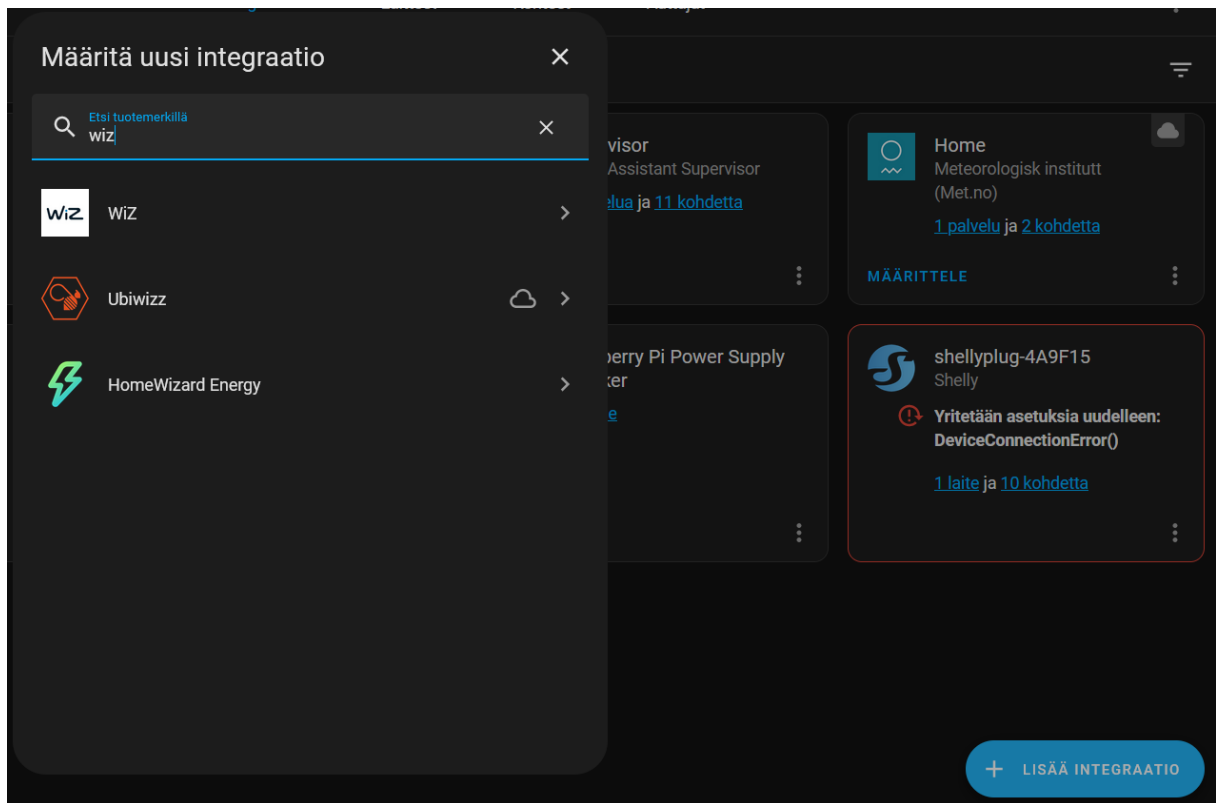
Lampulle valitaan sovelluksen sisäinen ikoni, nimi, katkaisijan funktio ja himmennyssefekt. Valintojen jälkeen päädytään kuvan 45 mukaiseen lopputulokseen.

Kuva 45. WiZ Connected -sovellus lisätty laite lopputulos.



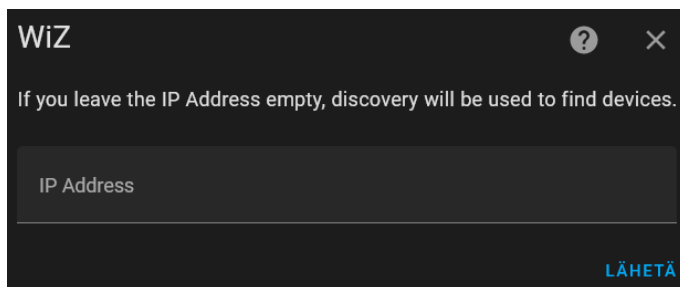
HA:n puolella navigoidaan sivupalkista kohtaan "Asetukset" ja valitaan "Laitteet & palvelut". Painetaan oikealla alhaalla olevaa "LISÄÄ INTEGRAATIO" -painiketta ja etsitään tuotemerkkiä "wiz" (kuva 46).

Kuva 46. HA WiZ-lampun integraation lisäys.



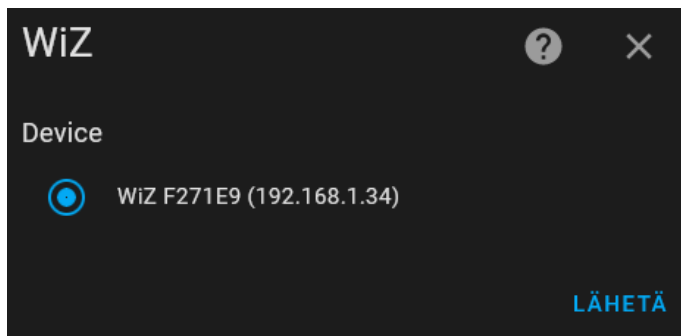
Valitaan "WiZ", jolloin HA kysyy kuvan 47 mukaisesti kotiverkossa olevan WiZ-lampun IP-osoitetta. Jätetään kenttä tyhjäksi, jolloin HA etsii laitteita automaattisesti.

Kuva 47. HA WiZ-lampun IP-osoitteen automaattinen etsintä.

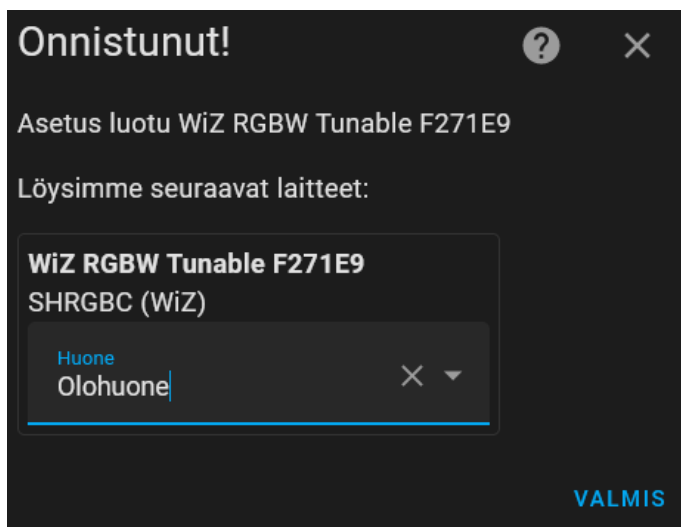


WiZ löydetään. Jatketaan painamalla "LÄHETÄ" -painiketta (kuva 48). Asennus onnistuu ja WiZ-integraatiolle valitaan kuvan 49 mukaisesti huone, johon se lisätään.

Kuva 48. HA WiZ-lamppu löydetty.

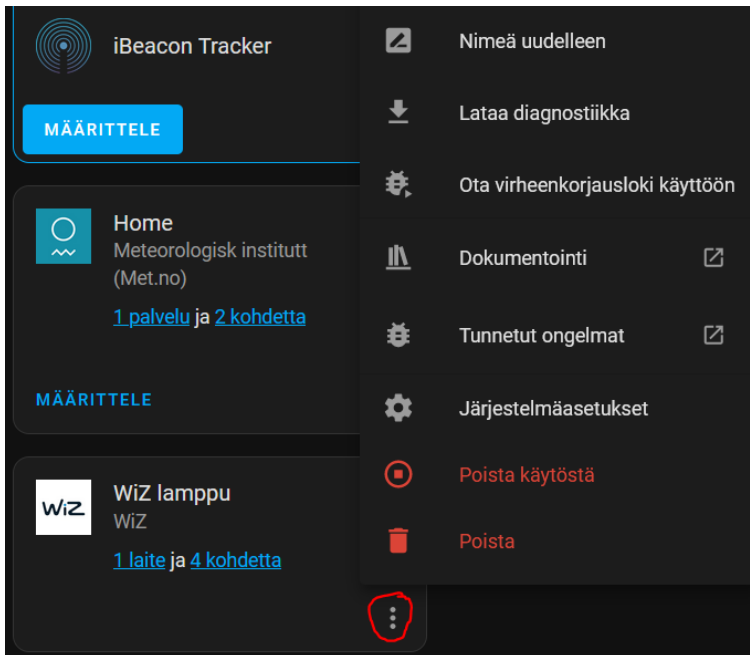


Kuva 49. HA WiZ-lamppu onnistunut integraatio huonevalinta.



Lopuksi integraation nimi vaihdetaan painamalla kuvassa 50 ympyröityä kolmetta pistettä, valitsemalla "Nimeä uudelleen", antamalla uusi nimi ja painamalla OK.

Kuva 50. HA WiZ-lampun uudelleennimeäminen.



### 6.3 Automaatioiden luonti

Luodaan neljä eri automaatiota. Kaksi automaatiota luodaan Shellyn ja kaksi WiZ-lampun avulla. Automaatioihin luodaan omia sensori-integraatioita YAML-merkintäkielen avulla, sekä käytetään hyväksi älypuhelimien antamia tietoja Home Assistant Companion -sovelluksen kautta.

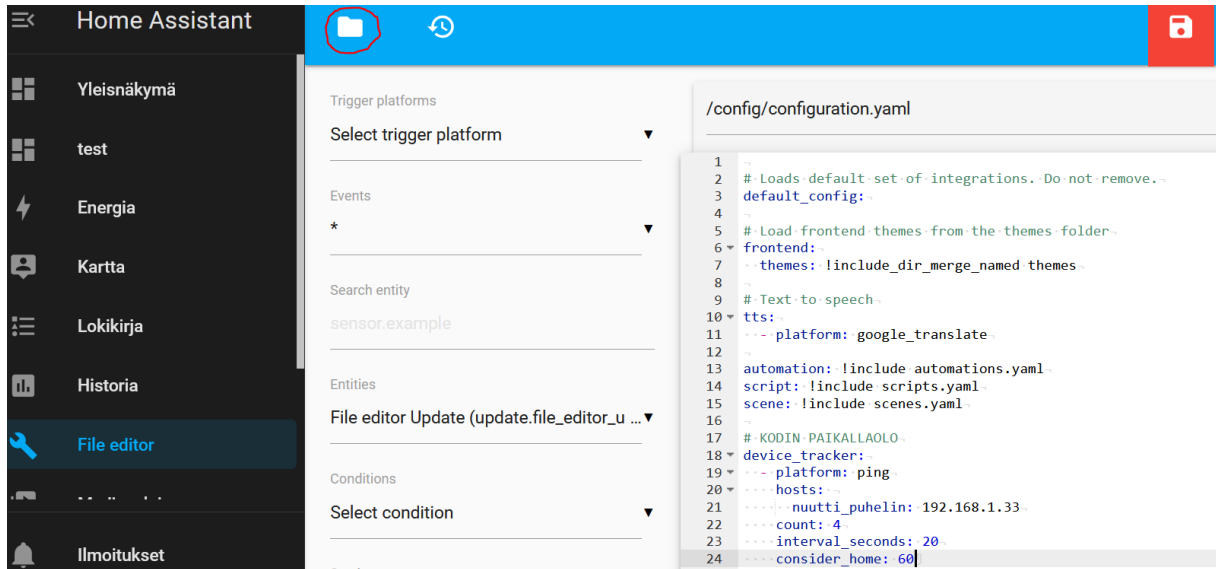
#### 6.3.1 Shelly pois päältä kodista lähtiessä

Luodaan automaatio, joka sammuttaa Shellyn ja siihen kytketyn laitteen. Sammutus tapahtuu kodin ollessa tyhjillään. Kodin paikallaoloa seurataan integraatiolla, joka toimii seuraamalla puhelimen yhteyttä kotiverkkoon lähettämällä sille ping-viestejä aika-ajoin. HA lähettää puhelimelle ping-viestin ja sen mennessä perille oletetaan käyttäjän olevan kotona, muuten poissa.

Luodaan paikallaolon integraatio avaamalla aikaisemmin asennettu File editor - tiedostonhallinta sivupalkista. Etsitään kuvassa 51 punaisella ympyröidyn kansio kuvakkeen alta "configurations.yaml" -tiedosto ja kirjoitetaan siihen liitteen 2 mukainen YAML-

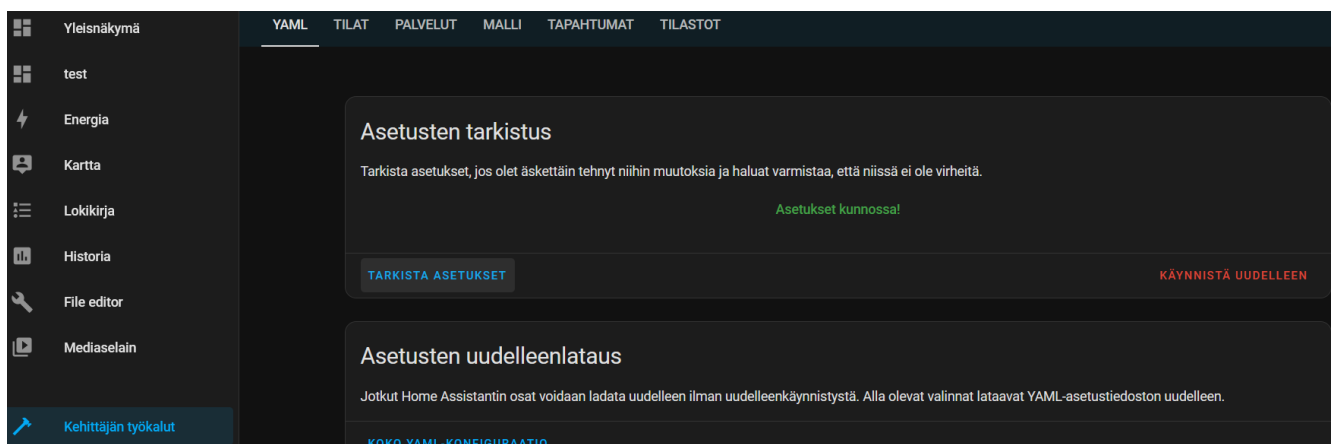
merkintäkoodi. Lopuksi tiedosto tallennetaan painamalla punaisella taustalla olevaa kuvaketta.

Kuva 51. Kodin paikallaoloa seuraava integraatio.



Navigoidaan sivupalkissa olevalle ”Kehittäjän työkalut” -välilehdelle ja tarkistetaan YAML-tiedosto virheiden varalta painamalla kuvassa 52 näkyvää ”Tarkista asetukset” -painiketta. Asetukset ovat kunnossa, joten HA-instanssi voidaan käynnistää uudelleen painamalla ”Käynnistä uudelleen” -painiketta. Tämä on hyvä tehdä aina YAML-tiedostoja muokattaessa.

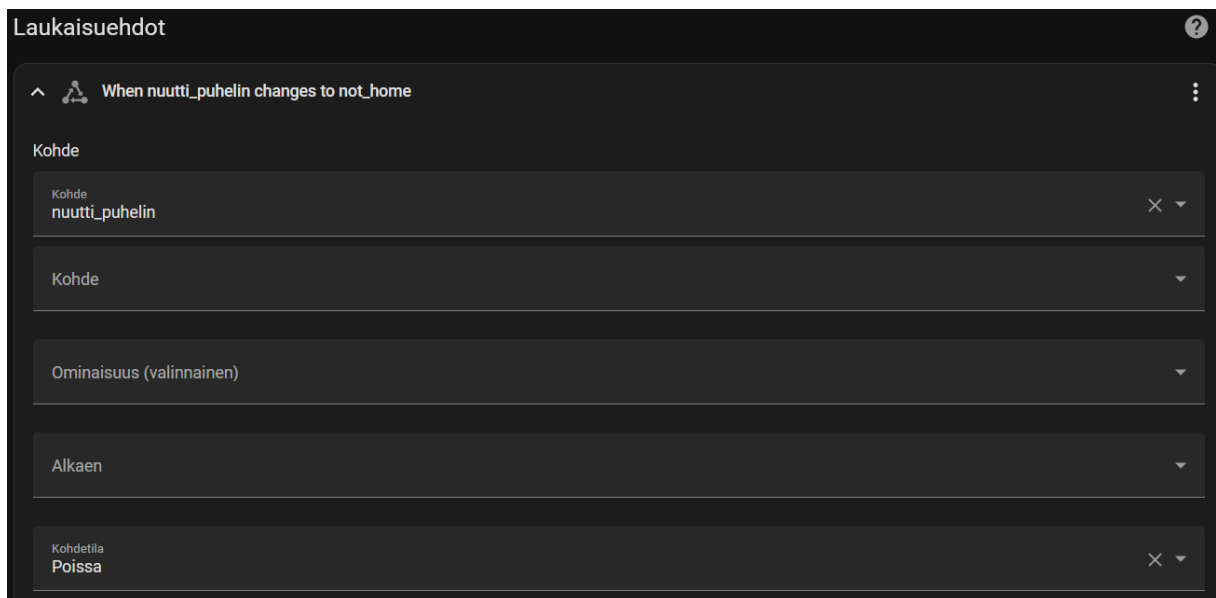
Kuva 52. HA-instanssin uudelleenkäynnistys.



Luodaan uusi automaatio navigoimalla sivupalkista kohtaan ”Asetukset” ja valitaan ”Automaatiot & tilanteet”. Avautuvasta sivusta painetaan ”LISÄÄ AUTOMAATIO” -painiketta ja valitaan ”Luo uusi automaatio”.

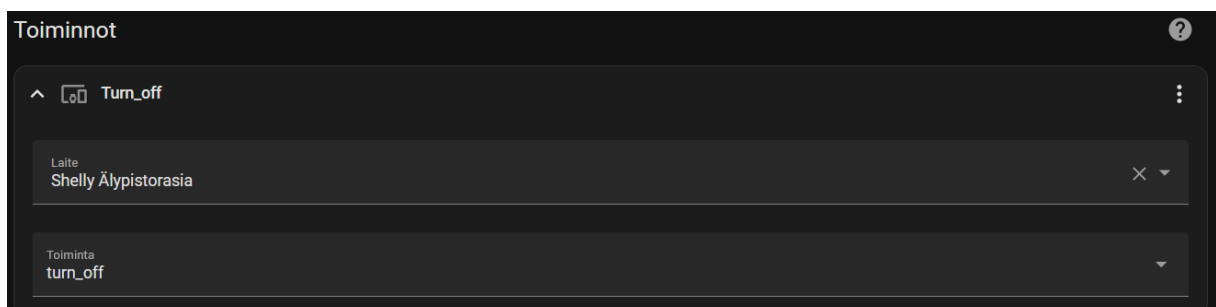
Uuteen automaatioon lisätään laukaisuehto valitsemalla ”LAUKAISUEHTO” -painikkeen alta avautuvasta listasta ”Tila”. Laukaisuehtoon annetaan kohteeksi ”nuutti\_puhelin”, joka on aikaisemmin tehty laitteen paikallaolon seuraaja. Kohdetilaksi annetaan ”Poissa”, jolloin päädytään kuvan 53 mukaiseen lopputulokseen.

Kuva 53. Shellyn automaattisammutuksen laukaisuehto.



Lisätään toiminto valitsemalla ”LISÄÄ TOIMINTO” -painikkeen alta avautuvasta listasta ”Laitte”. Laitteeksi annetaan ”Shelly älypistorasia” ja toiminnaksi ”turn\_off”, jolloin päädytään kuvan 54 mukaiseen lopputulokseen.

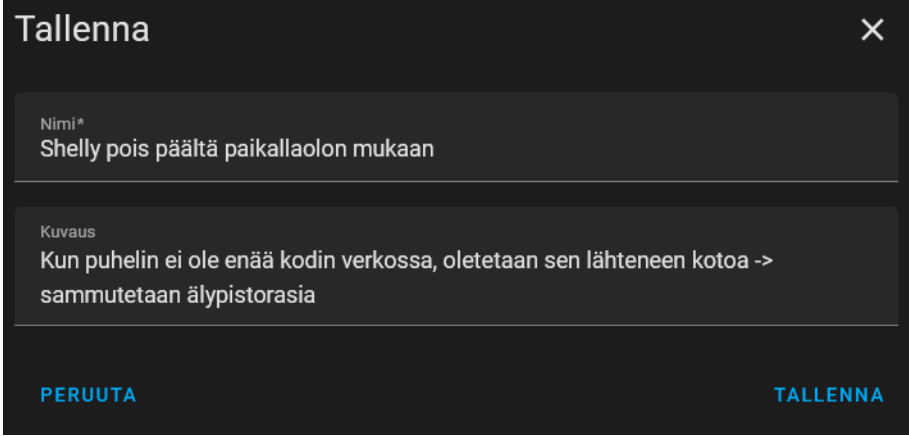
Kuva 54. Shellyn automaattisammutuksen toiminto.



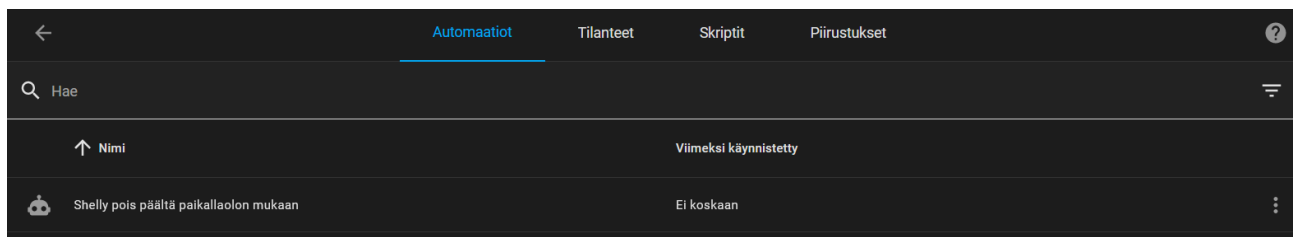


Painetaan oikealla alhaalla sijaitsevaa ”TALLENNA” -painiketta. Avautuvaan ikkunaan automaatiolle annetaan nimi ja kuvaus (kuva 55). Luotu automaatio näkyy kuvan 56 mukaisesti automaatiolistassa.

Kuva 55. Shellyn automaattisammutuksen tallennus.

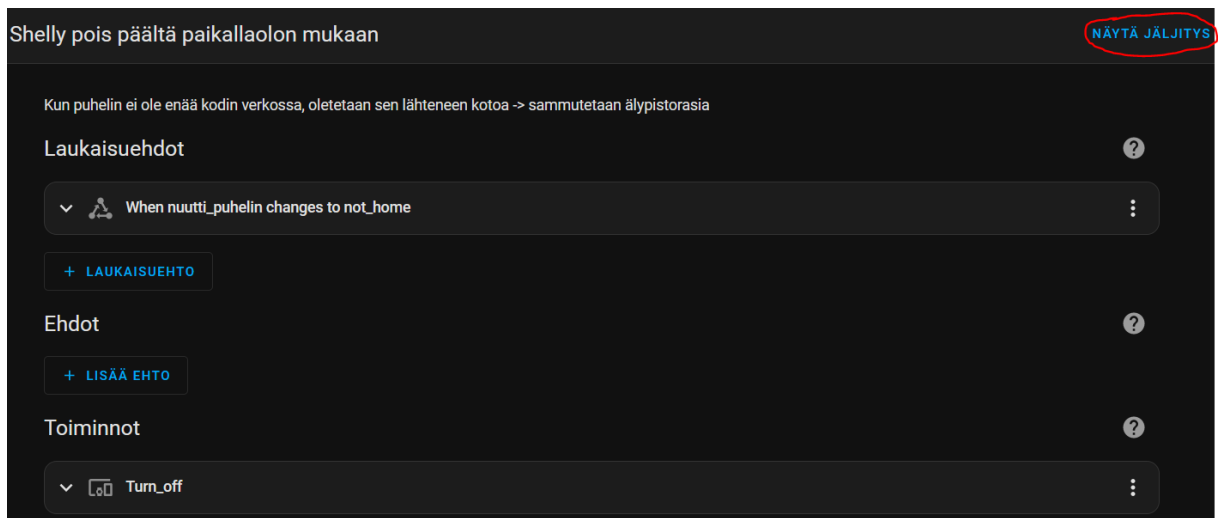


Kuva 56. Shellyn automaattisammutus automaatiolistassa.



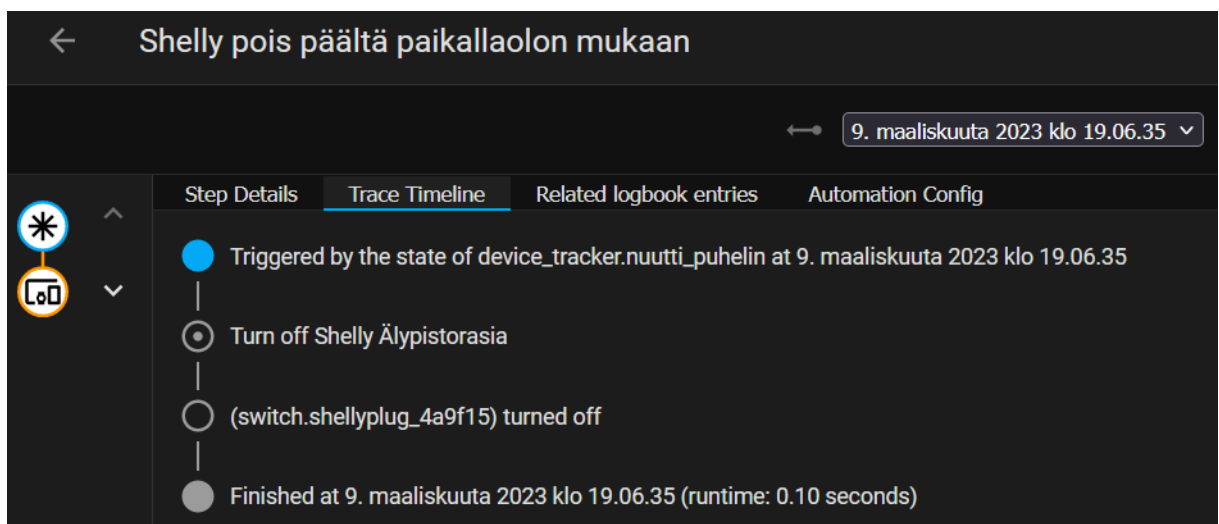
Testataan automaatiota kytkemällä Shelly pistorasiaan, kahvinkeitin Shellyyn ja puhelin kotiverkkoon. Tämän jälkeen laitetaan Shelly ja kahvinkeitin päälle, sekä kytketään puhelin pois kotiverkosta. Noin 10 sekunnin jälkeen Shelly kytketty pois päältä ja sen mukana kahvinkeitin. Automaation lokeja voidaan lukea avaamalla automaatio kuvassa 56 näkyvästä automaatiolistasta ja painamalla avautuvan sivun kuvassa 57 ympyröityä ”Näytä jäljitys” -painiketta.

Kuva 57. Shellyn automaattisammutuksen jäljituspainike.



Avautuvassa sivussa (kuva 58) nähdään milloin ja miten automaatio on käynnistynyt, sekä miten se on edistynyt. Tehty testi onnistui ja Shelly kytkeytyi pois päältä puhelimen lähdettyä kotiverkosta.

Kuva 58. Shellyn automaattisammutuksen jäljitysnäkyvä.



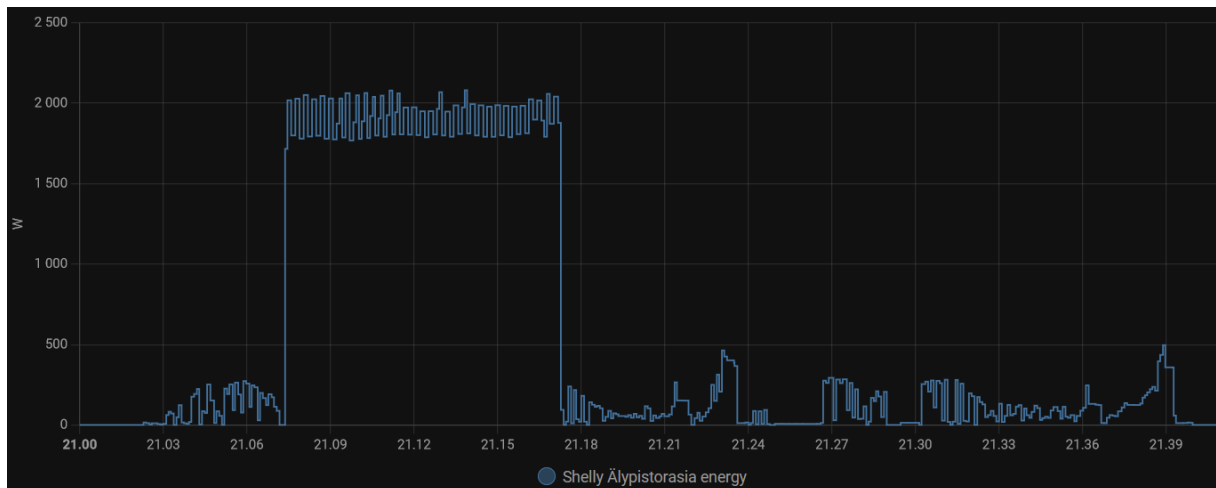
### 6.3.2 Shelly pesukoneen tilailmoitukset puhelimeen

Luodaan kolme automaatiota, jotka seuraavat pesukoneen tilaa ja lähettävät siitä ilmoituksia puhelimeen. Pesukoneen tilaa seurataan käyttämällä Shellysta saatuja

energiankulutusarvoja. Näistä arvoista saadaan ilmoituksia, kun pesukoneen sykli on käynnissä, sykli loppuu ja pesukone on pois päältä.

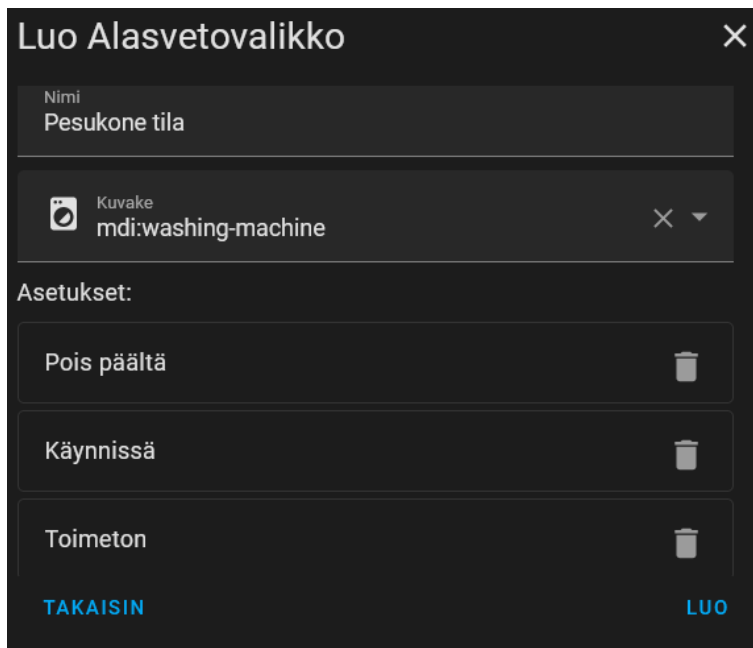
Ensin Shelly kytketään pistorasiaan ja pesukone Shellyyn. Pesukoneella ajetaan yksi sykli, jotta saadaan automaatiossa raja-arvoina käytetyt energiankulutusarvot (kuva 59).

Kuva 59. Shellyn pesukoneautomaation energiankulutusarvot.



Luodaan automaatiossa käytettävä avustaja pesukoneelle. Pesukone ei ole HA:ssa itse laitteena, mutta avustajalla sen tilaa voidaan simuloida näennäisesti. Navigoidaan sivupalkista kohtaan "Asetukset" ja valitaan "Laitteet & palvelut". Avautuneesta sivusta valitaan "Auttajat" -välilehti ja painetaan "LUO AVUSTIN" -painiketta. Avautuvasta valikosta valitaan "Alasvetovalikko". Avustajalle annetaan kuvan 60 mukaisesti nimi, kuvake ja kolme eri vaihtoehtoa.

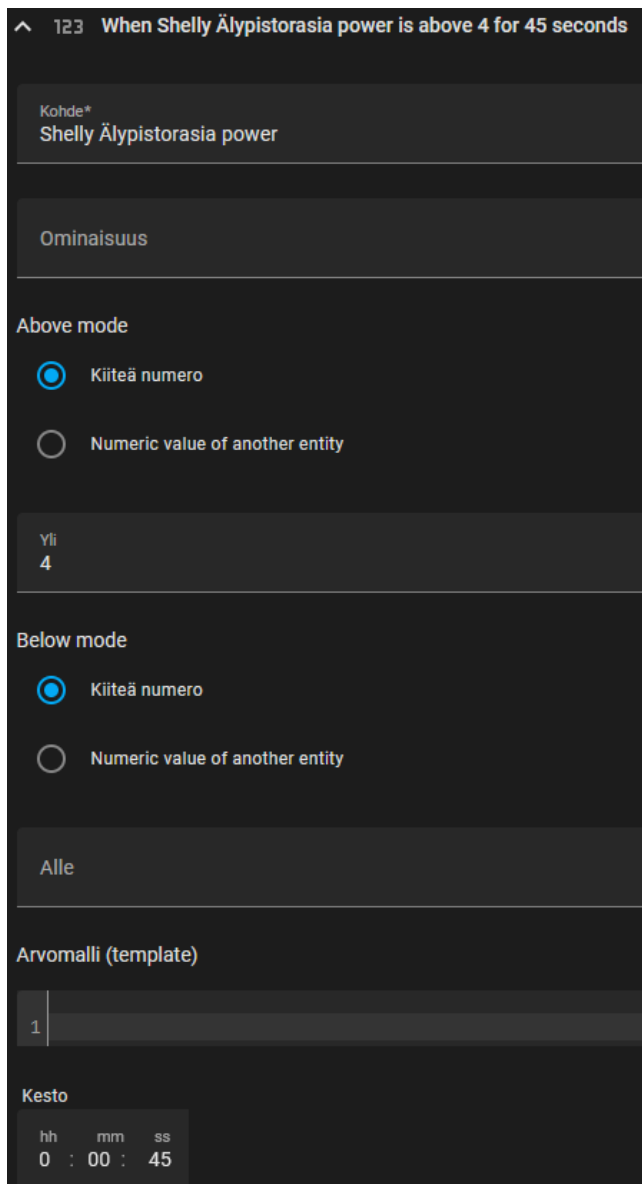
Kuva 60. Shellyn pesukoneautomaation avustaja.



Luodaan uusi automaatio navigoimalla sivupalkista kohtaan "Asetukset" ja valitaan "Automaatiot & tilanteet". Avautuvasta sivusta painetaan "LISÄÄ AUTOMAATIO" -painiketta ja valitaan "Luo uusi automaatio".

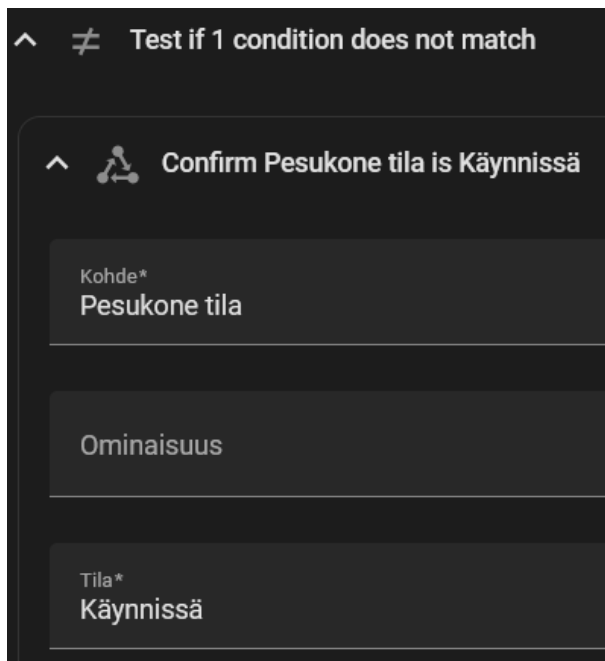
Automaatioon lisätään laukaisuehto valitsemalla "LAUKAISUEHTO" -painikkeen alta avautuvasta listasta "Numeerinen". Kohteeksi annetaan "Shelly Älypistorasia power". "Above mode" -kohtaan valitaan "Kiinteä numero" ja sen arvoksi annetaan 4. Kestoksi asetetaan 45 sekuntia, jolloin päädytään kuvan 61 mukaiseen lopputulokseen.

Kuva 61. Shellyn pesukoneautomaation laukaisuehto.



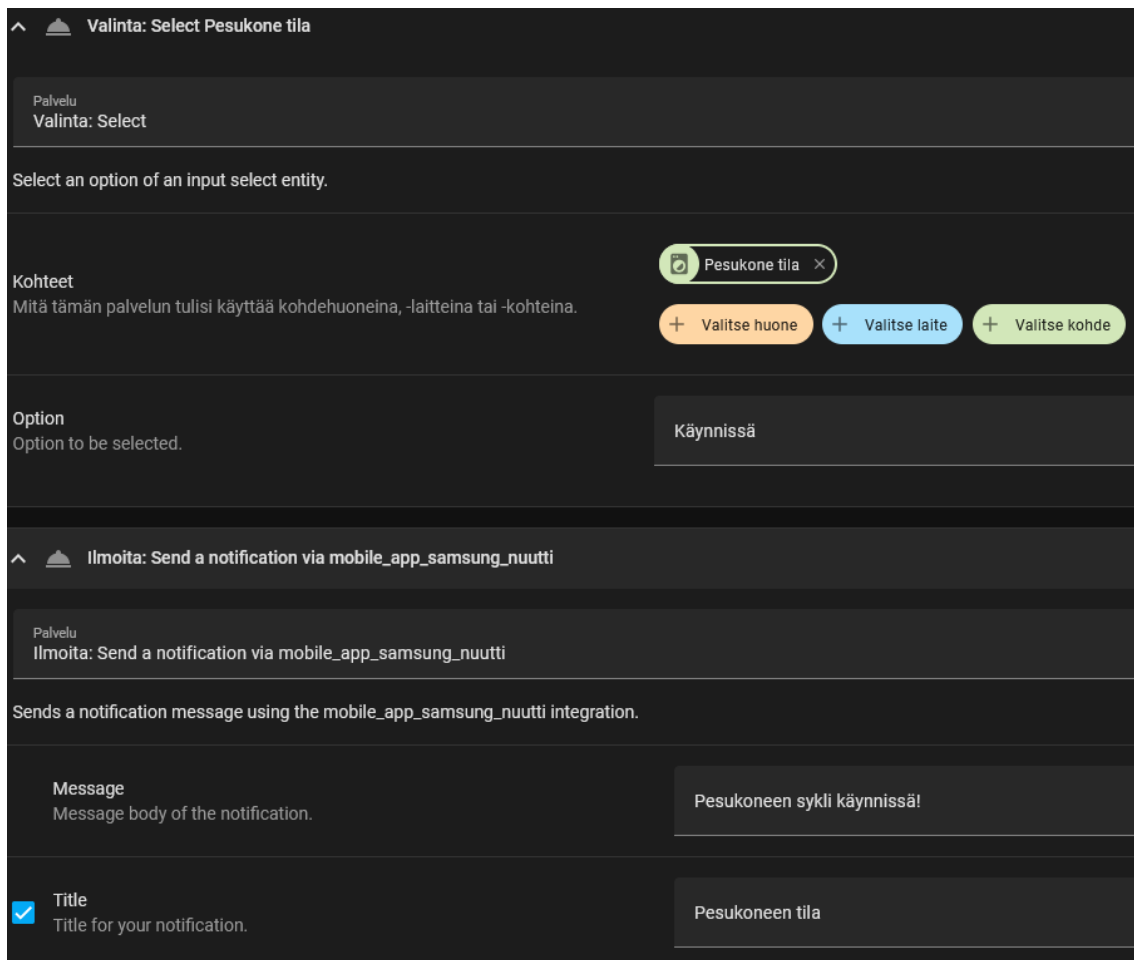
Lisätään ehto valitsemalla "LISÄÄ EHTO" -painikkeen alta avautuvasta listasta "Ei". Kohteeksi annetaan aikaisemmin luotu avustaja "Pesukone tila" ja tilaksi "Käynnissä", jolloin päädytään kuvan 62 mukaiseen lopputulokseen.

Kuva 62. Shellyn pesukoneautomaation ehto.



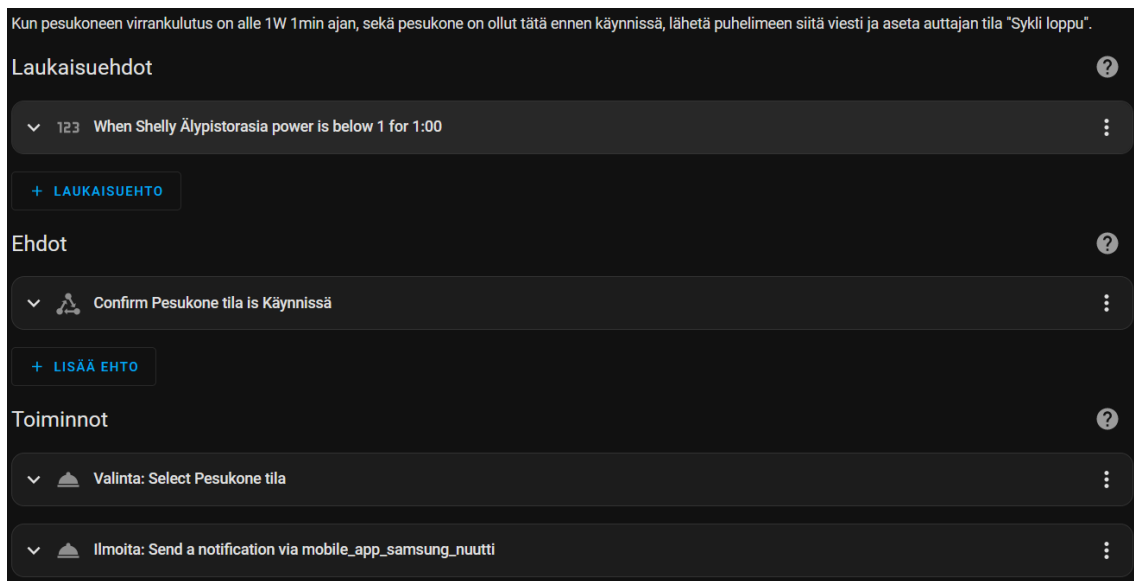
Lisätään kaksi toimintoa valitsemalla "LISÄÄ TOIMINTO" -painikkeen alta avautuvasta listasta "Kutsu palvelua". Ensimmäiseen toimintoon annetaan palveluksi "Valinta: Select" ja kohteeksi "Pesukone tila". "Option" -kohtaan kirjoitetaan "Käynnissä". Toiseen toimintoon annetaan palveluksi "Ilmoita: Send a notification via mobile\_app\_samsung\_nuutti" ja "Message" -kohtaan kirjoitetaan "Pesukoneen sykli käynnissä!". "Title" -valintaruutu rastitetaan ja siihen kirjoitetaan "Pesukoneen tila", jolloin päädytään kuvan 63 mukaiseen lopputulokseen.

Kuva 63. Shellyn pesukoneautomaation toiminnot.



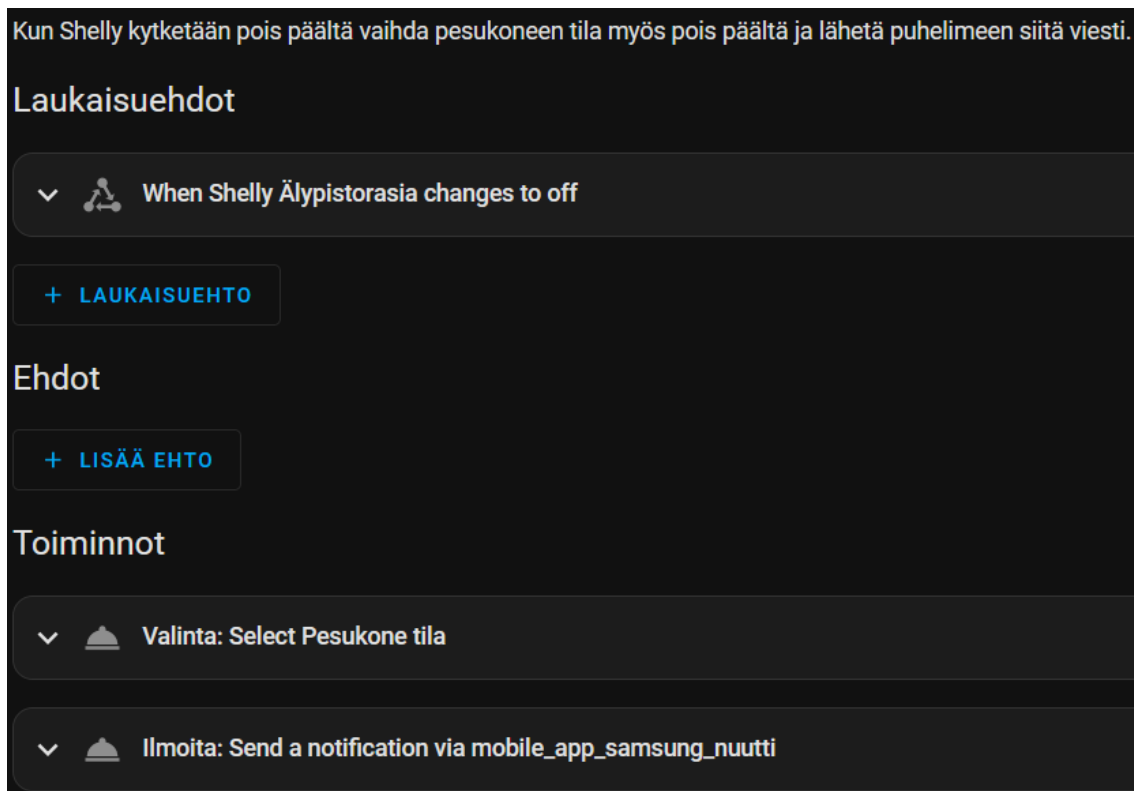
Syklin loppumisen ja pesukoneen pois päältä kytkemisen automaatiot luodaan samalla logiikalla. Syklin loppumisen automaatiolle (kuva 64) laukaisuehdoksi annetaan energiankulutus alle 1W minuutin ajan, ehdoksi pesukoneen tilan täytyy olla käynnissä ja toiminnoiksi pesukoneen tilan vaihtuminen "Sykli loppu" ja ilmoituksen lähettäminen puhelimeen.

Kuva 64. Shellyn pesukoneautomaation syklin loppuminen.



Pesukoneen pois päältä kytkemisen automaatiolle (kuva 65) laukaisuehdoksi annetaan Shellyn sammutus ja toiminnoiksi pesukoneen tilan vaihtuminen "Sykli loppu" ja ilmoituksen lähettäminen puhelimeen.

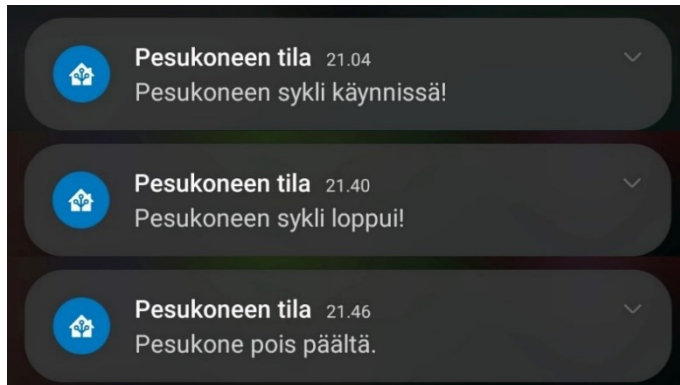
Kuva 65. Shellyn pesukoneautomaation pesukoneen pois päältä kytkeminen.





Testataan automaatiota kytkemällä Shelly pistorasiaan ja pesukone Shellyyn. Tämän jälkeen Shelly, pesukone ja pesukoneen sykli laitetaan päälle. Syklin aikana puhelimeen saadaan sen eri tiloista kuvan 66 mukaiset kolme ilmoitusta.

Kuva 66. Shellyn pesukoneautomaation puhelinilmoitukset.



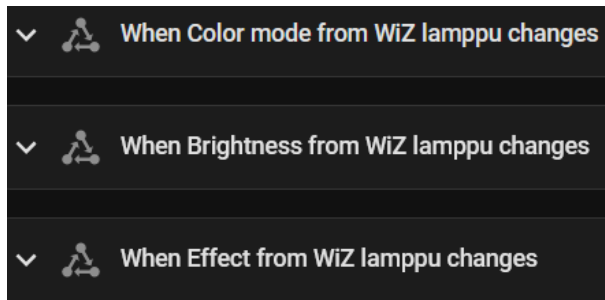
### 6.3.3 WiZ tilan tallentaja

Luodaan kaksi automaatiota, jotka tallentavat WiZ-lampun tilan (väri, kirkkaus ja efekti) sen vaihduttua ja palauttavat tallennetun tilan lampun kytkeydyttyä takaisin päälle katkaisijan kautta. Lamppu ei osaa itse palautua viimeksi asetettuun tilaan, joten se kytkeytyy päälle oletus tila-asetuksilla.

Luodaan ensimmäinen automaatio navigoimalla sivupalkista kohtaan "Asetukset" ja valitaan "Automaatiot & tilanteet". Avautuvasta sivusta painetaan "LISÄÄ AUTOMAATIO" -painiketta ja valitaan "Luo uusi automaatio".

Ensimmäiseen automaatioon lisätään kolme laukaisuehtoa valitsemalla "LAUKAISUEHTO" painikkeen alta avautuvasta listasta "Tila". Jokaisen laukaisuehdon kohteeksi valitaan "WiZ lamppu". Ensimmäisen laukaisuehdon "Ominaisuus (valinnainen)" -kohtaan valitaan "Color mode", toisen "Brightness" ja kolmannen "Effect", jolloin päädytään kuvan 67 mukaiseen lopputulokseen.

Kuva 67. WiZ-lampun tilatallentaja-automaation laukaisuehdot.

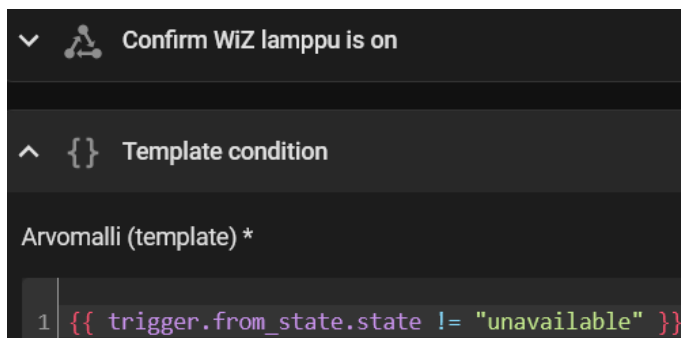


Lisätään kaksi ehtoa valitsemalla "LISÄÄ EHTO" -painikkeen alta avautuvasta listasta "Tila". Ensimmäiseen ehtoon annetaan kohteeksi "WiZ lamppu" ja tilaksi "on". Toiseen ehtoon listasta valitaan "Malli (template)". "Arvomalli (template)" -kenttään kirjoitetaan YAML-merkintäkoodi 1 mukainen koodi, jolloin päädytään kuvan 68 mukaiseen lopputulokseen.

YAML-merkintäkoodi 1. WiZ tilatallentaja-automaation toinen ehto.

```
{{ trigger.from_state.state != "unavailable" }}
```

Kuva 68. WiZ-lampun tilatallentaja-automaation ehdot.



Lisätään yksi toiminto valitsemalla "LISÄÄ TOIMINTO" -painikkeen alta avautuvasta listasta "Kutsu palvelua". Ensimmäiseen toimintoon annetaan palveluksi "Skene: Create" ja kirjoitetaan "Scene entity ID" -kohtaan "wiz\_state". Rastitetaan "Snapshot entities" -valintaruutu ja kirjoitetaan siihen "light.wiz\_rgbw\_tunable\_f271e9", jolloin päädytään kuvan 69 mukaiseen tulokseen.

Kuva 69. WiZ-lampun tilatallentaja-automaation toiminto.

Skene: Create

Palvelu  
Skene: Create

Creates a new scene.

Scene entity ID  
The entity\_id of the new scene.

wiz\_state

Entities state  
The entities to control with the scene.

1

Snapshot entities  
The entities of which a snapshot is to be taken

1 light.wiz\_rgbw\_tunable\_f271e9  
2

Painetaan oikealla alhaalla sijaitsevaa "TALLENNA" -painiketta. Avautuvaan ikkunaan automaatiolle annetaan nimi ja kuvaus.

Luodaan toinen automaatio, jolle lisätään yksi laukaisuehto valitsemalla "LAUKAISUEHTO" painikkeen alta avautuvasta listasta "Tila". Kohteeksi valitaan "WiZ lamppu". "Alkaen" -kohtaan valitaan "unavailable" ja kohdetilaksi "on", jolloin päädytään kuvan 70 mukaiseen lopputulokseen.

Kuva 70. WiZ-lampun tilanpalauttaja-automaation laukaisuehto,

When WiZ lamppu changes from unavailable to on

Kohde

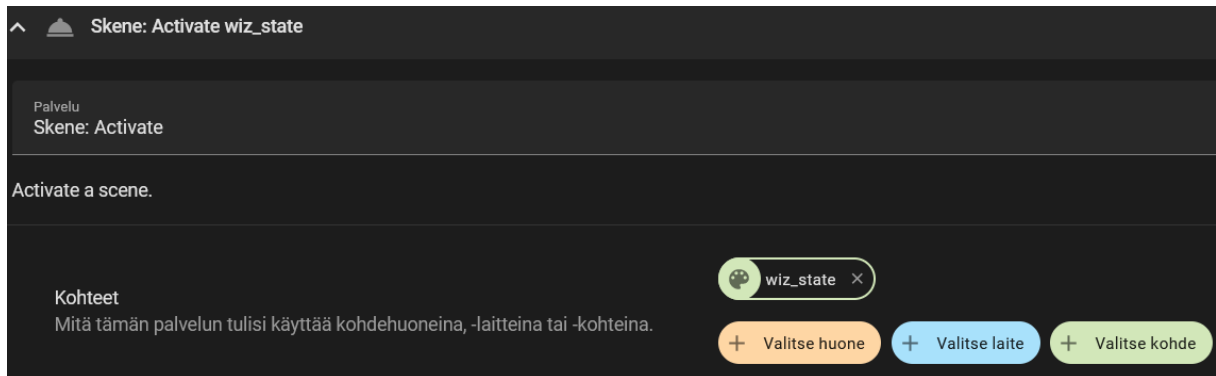
Kohde  
WiZ lamppu

Alkaen  
unavailable

Kohdetila  
on

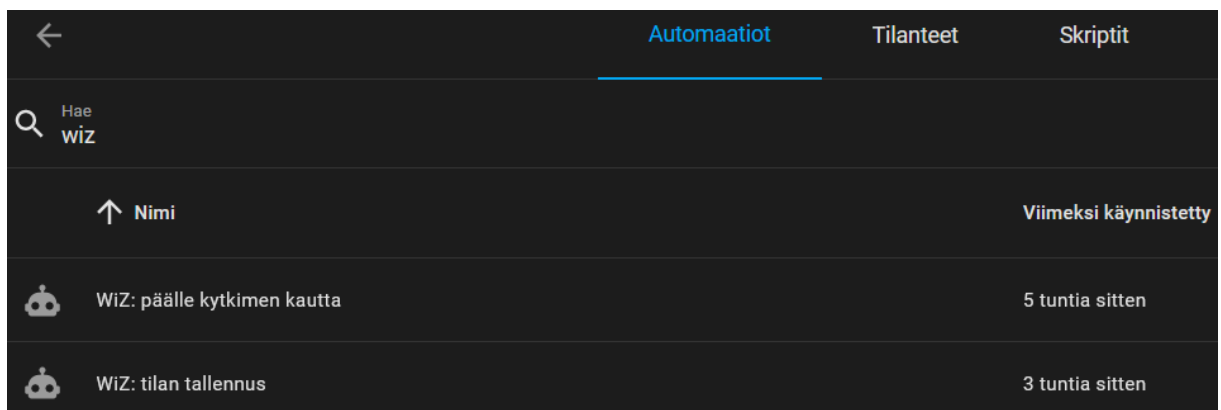
Ehtoja ei lisätä. Lisätään yksi toiminto valitsemalla ”LISÄÄ TOIMINTO” -painikkeen alta avautuvasta listasta ”Kutsu palvelua”. Palveluksi annetaan ”Skene: Activate” ja kohteeksi ”+ Valitse kohde” -kohdasta aikaisemmin luotu tilanne ”wiz\_state”, jolloin päädytään kuvan 71 mukaiseen tulokseen.

Kuva 71. WiZ-lampun tilanpalauttaja-automaation toiminto.



Painetaan oikealla alhaalla sijaitsevaa ”TALLENNNA” -painiketta. Avautuneeseen ikkunaan automaatiolle annetaan nimi ja kuvaus. Luodut automaatiot näkyvät kuvan 72 mukaisesti automaatiolistassa.

Kuva 72. WiZ-lampun tilatallentaja- ja tilanpalauttaja-automaatiot listattuna.



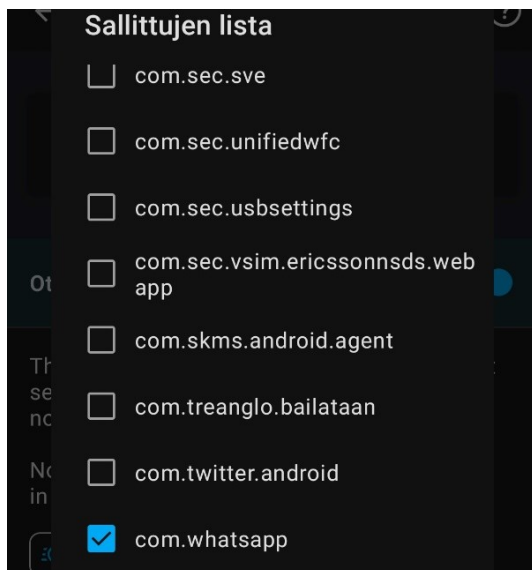
Testataan automaatiota vaihtamalla WiZ-lampun asetuksia, jonka jälkeen se kytketään valokatkaisijasta pois päältä. Tilan vaihtuessa ”unavailable” -moodiin lamppu kytketään takaisin päälle, jolloin se valaisee oletusasetuksillaan. Noin kymmenen sekunnin kuluttua lampun asetukset muuttuvat siihen, mitä ne olivat ennen poiskytkemistä.

### 6.3.4 WiZ välähdys WhatsApp-viestin saapuessa

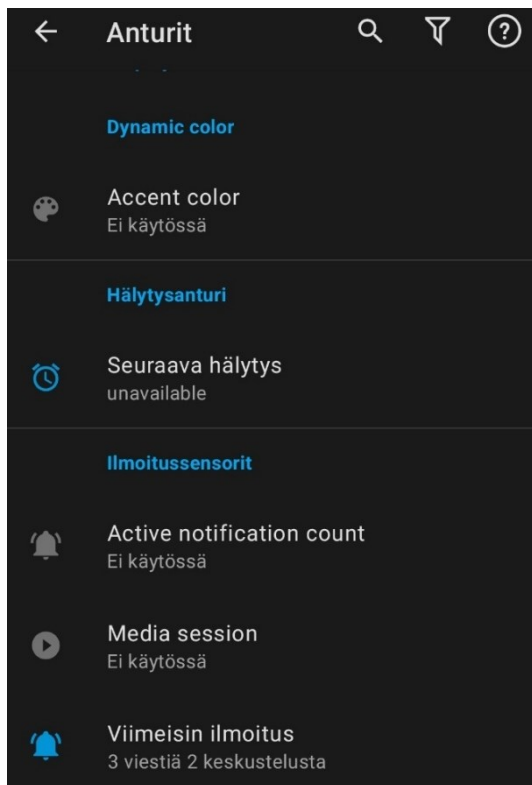
Luodaan automaatio, joka väläyttää WiZ-lamppua 2 kertaa ja palauttaa sen alkuperäiseen tilaansa. Automaatio laukaistaan, kun WhatsApp-sovellukseen saadaan viesti, joka sisältää sanan ”!tärkeä!”.

Otetaan käyttöön kaksi sensoria puhelimeen ladatun Home Assistant Companion -sovelluksen kautta. Navigoidaan sivupalkista kohtaan ”Asetukset” ja valitaan ”Kumppanisovellus”. Avautuvasta sivusta valitaan ”Hallitse antureita” ja etsitään kohta ”Ilmoitussensorit”, jonka alta valitaan ”Viimeisin ilmoitus”. Avautuvasta sivusta anturi kytketään päälle. Valitaan kohdan ”Sensoriasetukset” alta ”Sallittujen lista” ja rastitetaan sieltä päälle kuvan 73 mukainen ”WhatsApp (com.whatsapp)”. Anturi kytkeytyy päälle ja näyttää vain WhatsApp-sovelluksen ilmoitukset (kuva 74).

Kuva 73. WiZ-lampun WhatsApp-automaation ilmoitusten sallittujen lista.



Kuva 74. WiZ-lampun WhatsApp-automaation ilmoitusanturi päälle kytkettynä.



Luodaan automaatiossa käytettävän avustajan määrittämä laukaisusana. Navigoidaan sivupalkista kohtaan "Asetukset" ja valitaan "Laitteet & palvelut". Avautuvasta sivusta valitaan "Auttajat" -välilehti ja painetaan "LUO AVUSTIN" -painiketta. Avautuvasta valikosta valitaan "Teksti". Avustajalle annetaan kuvan 75 mukaisesti nimi, kuvake, vähimmäis- ja enimmäispituus.

Kuva 75. WiZ-lampun WhatsApp-automaation laukaisusanan avustaja.

Avataan luotu auttaja ja asetetaan sille kuvan 76 mukainen laukaisusana.

Kuva 76. WiZ-lampun WhatsApp-automaation laukaisusanan asetus.

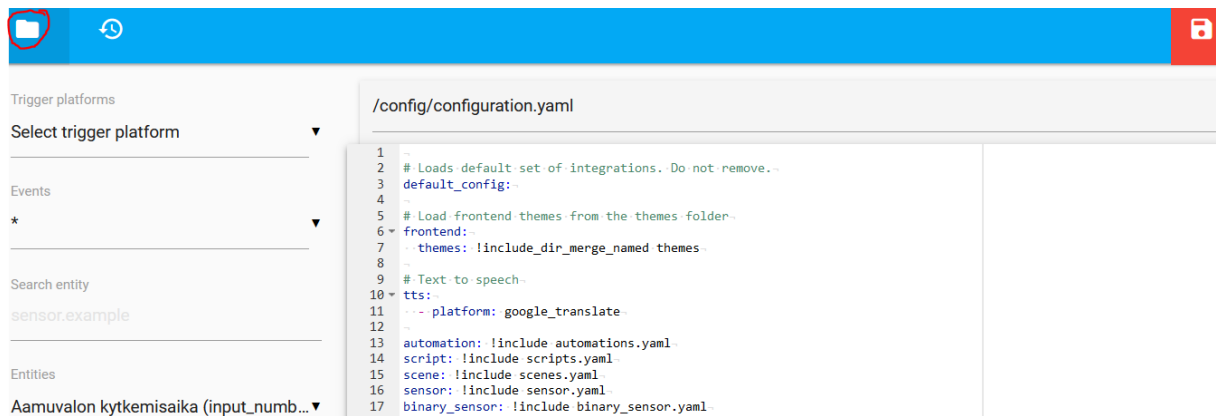


Avataan File editor -tiedostonhallinta sivupalkista. Etsitään kuvassa 77 punaisella ympyröidyn kansiokuvakkeen alta "configuration.yaml" -tiedosto ja kirjoitetaan siihen rivit 16-17 YAML-merkintäkoodi 2 mukaisesti. Rivit kertovat HA:lle, että kaikki luodut *sensor*- ja *binary\_sensor*-entiteetit löytyvät annetun nimisestä tiedostoista. Näin integraatioita voidaan helposti kategorisoida. Lopuksi tiedosto tallennetaan painamalla punaisella taustalla olevaa kuvaketta.

YAML-merkintäkoodi 2. Sensori- ja binäärisensorientiteettien tiedostojen merkkkaus.

```
sensor: !include sensor.yaml
binary_sensor: !include binary_sensor.yaml
```

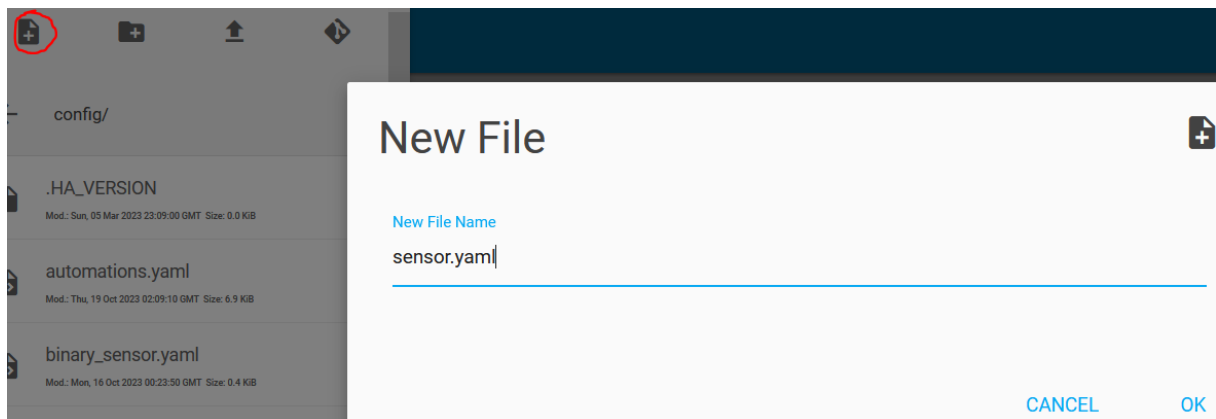
Kuva 77. WiZ-lampun WhatsApp-automaaation tiedostonimien asetus.



Luodaan kaksi uutta tiedostoa painamalla samaa kansiokuvaketta kuin kuvassa 77.

Avautuvasta valikosta painetaan kuvassa 78 ympyröityä tiedostokuvaketta, annetaan tiedostolle nimi "sensor.yaml" ja painetaan OK. Sama toistetaan ja toiselle tiedostolle annetaan nimi "binary\_sensor.yaml".

Kuva 78. WiZ-lampun WhatsApp-automaaation "sensor.yaml" -tiedoston luonti.

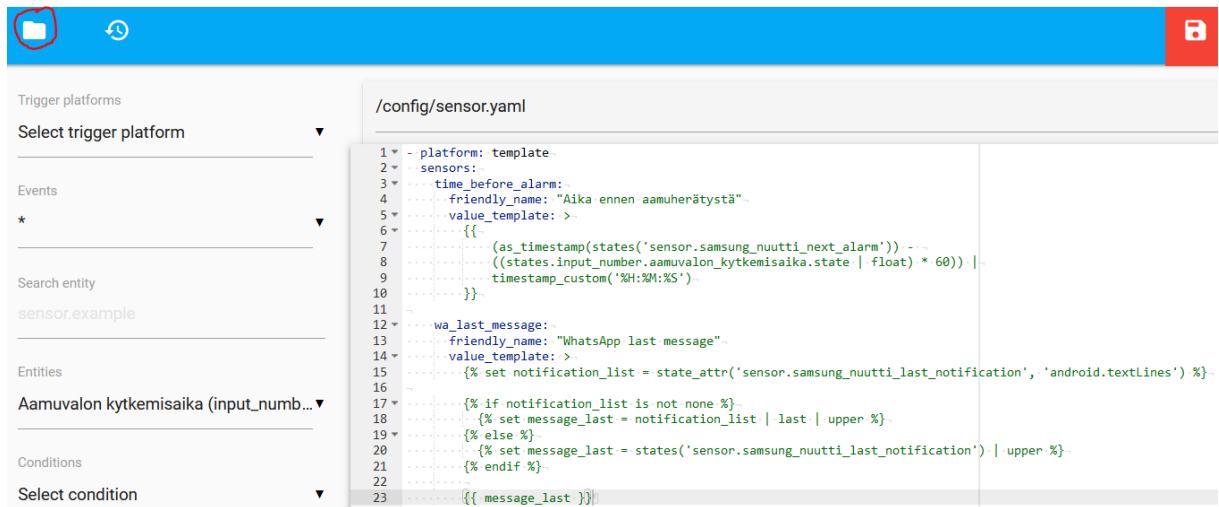


Luodaan integraatio, joka ottaa puhelimesta WhatsApp-sovelluksen viimeisimmän viesti-ilmoituksen talteen. Ilmoituksesta otetaan edelleen talteen viestisisältö ja se muutetaan isokirjaimiseksi, jotta sen kirjoitusasu ei haittaa automaatioiden toimintaa.

Etsitään kuvassa 77 punaisella ympyröidyn kansiokuvakkeen alta "sensor.yaml" -tiedosto ja kirjoitetaan kuvassa 79 näkyville riveille 12-23 liitteen 3 mukainen integraatio YAML-merkintäkoodilla. Lopuksi tiedosto tallennetaan painamalla punaisella taustalla olevaa kuvaketta.



Kuva 79. WiZ-lampun WhatsApp-automaation viestin tallennusintegraatio.



Luodaan integraatio, joka tarkistaa, löytyykö aikaisemmin luodun viestin tallennusintegraation tekstistä auttajaan asetettua laukaisusanaa. Löytäessä palauttaa tosi, muuten epätosi.

Etsitään kuvassa 79 punaisella ympyröidyn kansiokuvakkeen alta "binary\_sensor.yaml" -tiedosto ja kirjoitetaan kuvassa 80 näkyville riveille 1-12 liitteen 4 mukainen integraatio YAML-merkintäkoodilla. Lopuksi tiedosto tallennetaan painamalla punaisella taustalla olevaa kuvaketta.

Kuva 80. WiZ-lampun WhatsApp-automaation laukaisusanan tarkastajaintegraatio.



Luodut integraatiot löytyvät kuvan 81 mukaisesti sivupalkin "Kehittäjän työkalut" -sivun "Tilat" -välilehden alta.

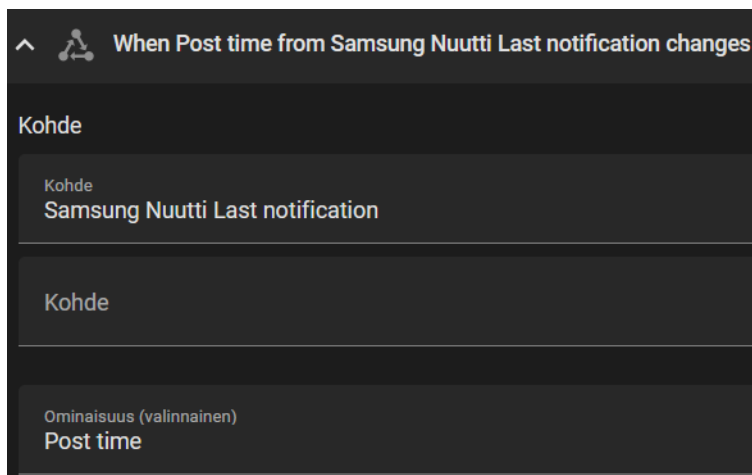
Kuva 81. WiZ-lampun WhatsApp-automaatioon luodut integraatiot listattuna.

Kohde	Tila	Määritteet <input checked="" type="checkbox"/>
<input type="text" value="Suodata kohteita messa"/>	<input type="text" value="Suodata tiloja"/>	<input type="text" value="Suodata määritteitä"/>
<a href="#">binary_sensor.wa_last_message_contains_trigger_w</a> <a href="#">ord</a> WhatsApp last message contains trigger word	off	friendly_name: WhatsApp last message contains trigger word
<a href="#">sensor.wa_last_message</a> WhatsApp last message	JEE JEE	friendly_name: WhatsApp last message

Luodaan uusi automaatio navigoimalla sivupalkista kohtaan ”Asetukset” ja valitaan ”Automaatiot & tilanteet”. Avautuneesta sivusta painetaan ”LISÄÄ AUTOMAATIO” -painiketta ja valitaan ”Luo uusi automaatio”.

Automaatioon lisätään laukaisuehto valitsemalla ”LAUKAISUEHTO” -painikkeen alta avautuvasta listasta ”Tila”. Laukaisuehtoon annetaan kohteeksi ”Samsung Nuutti Last notification”. ”Ominaisuus (valinnainen)” -kohtaan valitaan ”Post time”, jolloin päädytään kuvan 82 mukaiseen lopputulokseen.

Kuva 82. WiZ-lampun WhatsApp-automaation laukaisuehto.



Lisätään kaksi ehtoa valitsemalla ”LISÄÄ EHTO” -painikkeen alta avautuvasta listasta ”Laite”. Ensimmäiseen ehtoon annetaan laitteeksi ”WiZ lamppu” ja ehdoksi ”WiZ lamppu is on”.

Toiseen ehtoon listasta valitaan ”Tila”. Kohteeksi annetaan aikaisemmin luotu integraatio ”WhatsApp last message contains trigger word” ja tilaksi ”On”, jolloin päädytään kuvan 83 mukaiseen lopputulokseen.

Kuva 83. WiZ-lampun WhatsApp-automaation ehdot.

**Ehdot**

^ WiZ lamppu is on

Laite  
WiZ lamppu

Ehto  
WiZ lamppu is on

Kesto

hh mm ss  
0 : 00 : 00

^ Confirm WhatsApp last message contains trigger word is on

Kohde\*  
WhatsApp last message contains trigger word

Ominaisuus

Tila\*  
on

Lisätään kolme toimintoa valitsemalla ”LISÄÄ TOIMINTO” -painikkeen alta avautuvasta listasta ”Kutsu palvelua”. Ensimmäiseen toimintoon annetaan palveluksi ”Skene: Create” ja kirjoitetaan ”Scene entity ID” -kohtaan ”wiz\_alkutila”. Rastitetaan ”Snapshot entities” -valintaruutu ja kirjoitetaan siihen ”light.wiz\_rgbw\_tunable\_f271e9”, jolloin päädytään kuvan 84 mukaiseen tulokseen.

Kuva 84. WiZ-lampun WhatsApp-automaation ensimmäinen toiminto.

Skene: Create

Palvelu  
Skene: Create

Creates a new scene.

Scene entity ID  
The entity\_id of the new scene. wiz\_alkutila

Entities state  
The entities to control with the scene. 1

Snapshot entities  
The entities of which a snapshot is to be taken 1 light.wiz\_rgbw\_tunable\_f271e9

Toiseen toimintoon valitaan "Repeat", toistotyyppiä "Laske" ja toistokerroiksi viisi. "Repeat"-toiminnolle lisätään kaksi omaa toimintoa valitsemalla "LISÄÄ TOIMINTO" (kuva 85).

Kuva 85. WiZ-lampun WhatsApp-automaation toinen-toiminto.

Repeat an action 5 times

Toistotyyppi  
Laske

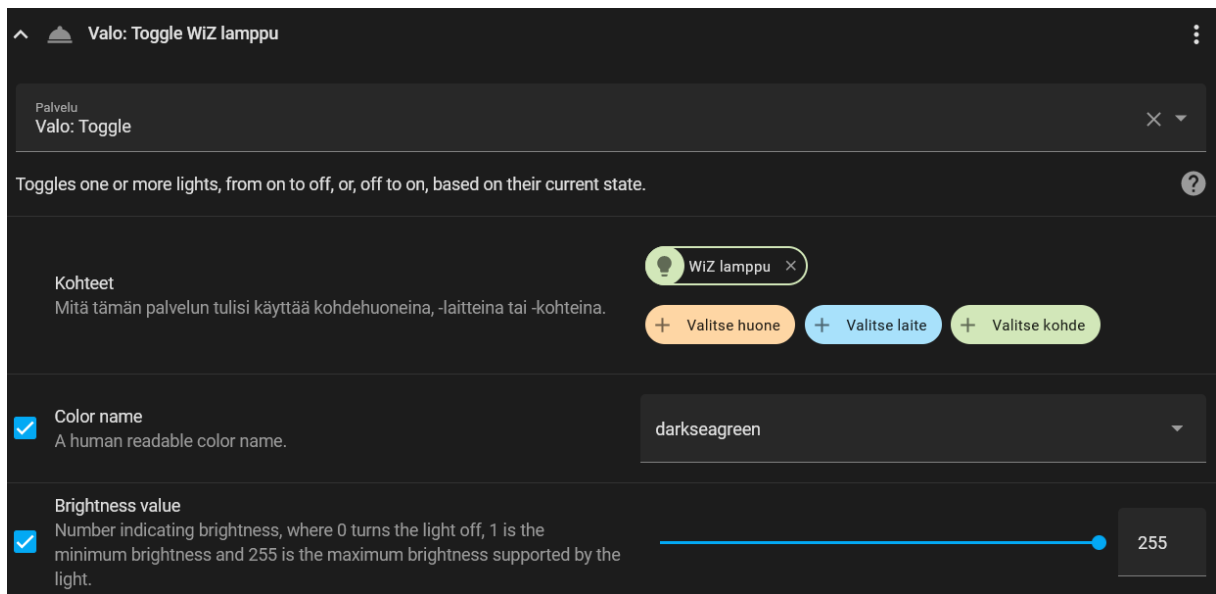
Laske  
5

Toiminnot:

+ LISÄÄ TOIMINTO

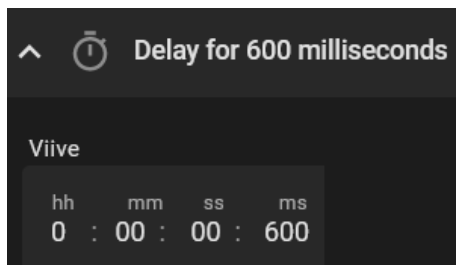
Ensimmäiseen "Repeat"-toimintoon valitaan "Kutsu palvelua". Palveluksi annetaan "Valo: Toggle" ja kohteeksi "+ Valitse kohde" -kohdasta "WiZ lamppu". Rastitetaan "Color name" -valintaruutu ja annetaan väriksi "darkseagreen". Rastitetaan "Brightness value" -valintaruutu ja annetaan arvoksi 255, jolloin päädytään kuvan 86 mukaiseen tulokseen.

Kuva 86. WiZ-lampun WhatsApp-automaation "Repeat" -toiminnon ensimmäinen toiminto.



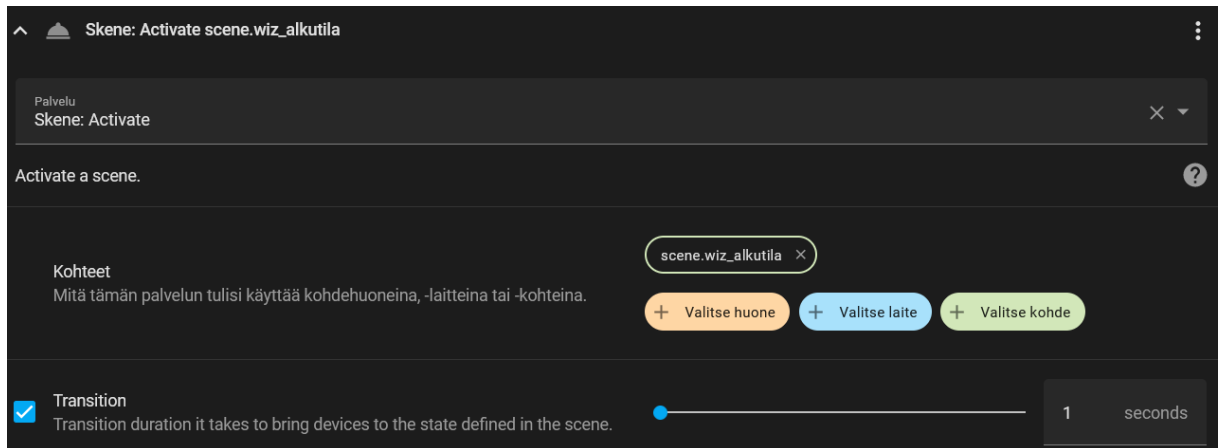
Toiseen "Repeat" -toimintoon valitaan "Viive". Arvoksi annetaan 600ms, jolloin päädytään kuvan 87 mukaiseen tulokseen.

Kuva 87. WiZ-lampun WhatsApp-automaation "Repeat" -toiminnon toinen toiminto.



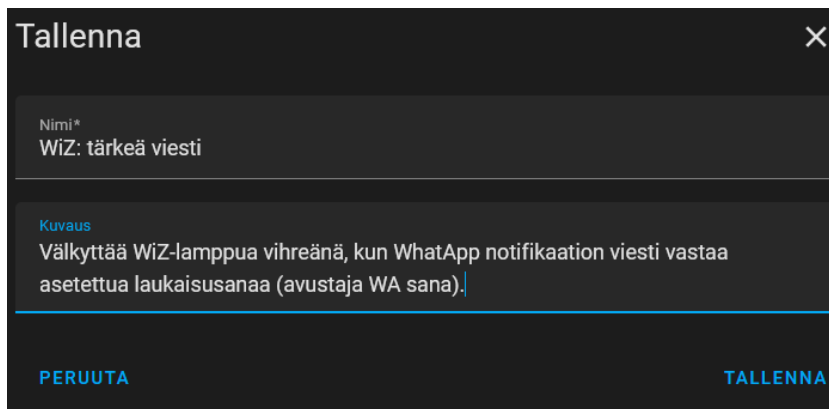
Kolmanteen toimintoon valitaan "Kutsu palvelua". Palveluksi annetaan "Skene: Activate" ja kohteeksi "+ Valitse kohde" -kohdasta aiemmin luotu tilanne "wiz\_alkutila". Rastitetaan "Transition" -valintaruutu ja annetaan arvoksi 1 sekunti, jolloin päädytään kuvan 88 mukaiseen tulokseen.

Kuva 88. WiZ-lampun WhatsApp-automaation kolmas toiminto.

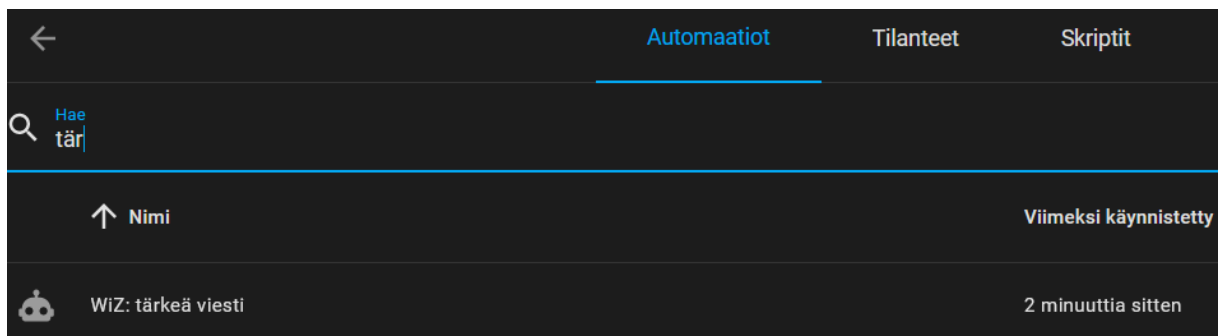


Painetaan oikealla alhaalla sijaitsevaa "TALLENNA" -painiketta. Avautuvaan ikkunaan automaatiolle annetaan nimi ja kuvaus (kuva 89). Luotu automaatio näkyy kuvan 90 mukaisesti automaatiolistassa.

Kuva 89. WiZ-lampun WhatsApp-automaation tallennus.

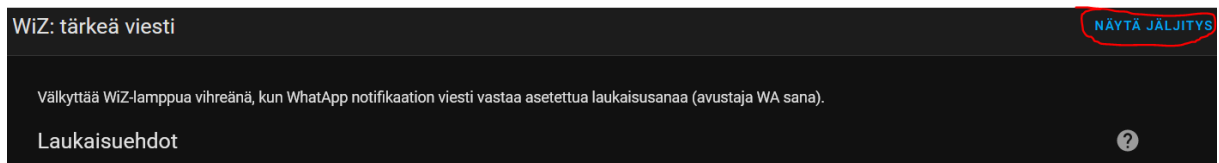


Kuva 90. WiZ-lampun WhatsApp-automaatio automaatiolistassa.



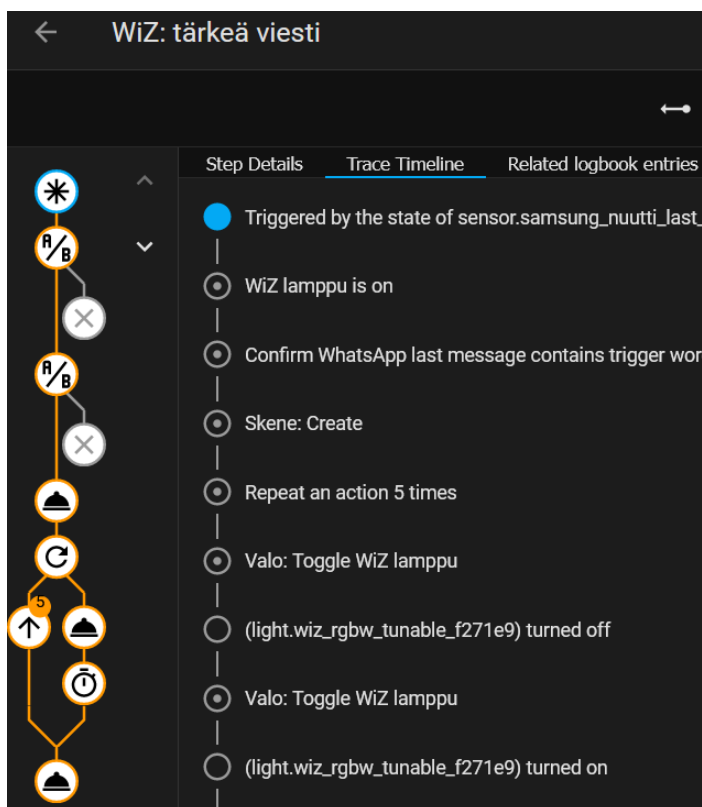
Testataan automaatiota pyytämällä jotakuta lähettämään viesti WhatsApp-sovellukseen, joka sisältää sanan laukaisusanan "tärkeä!". WhatsApp sovelluksesta poistutaan, jotta ilmoitus viestistä saapuu puhelimen ilmoituksiin. Ilmoituksen saavuttua WiZ-lamppu välkähtää 2 kertaa ja palautuu alkuperäiseen tilaansa. Automaation lokeja voidaan lukea avaamalla automaatio kuvassa 90 näkyvästä automaatiolistasta ja painamalla avautuneen sivun kuvassa 91 ympyröityä "Näytä jäljitys" -painiketta.

Kuva 91. WiZ-lampun WhatsApp-automaation jäljitys.



Avautuvassa sivussa (kuva 92) nähdään milloin ja miten automaatio käynnistyi, sekä miten se edistyi. Tehty testi onnistui: WiZ-lamppu välkähti 2 kertaa ja palautui alkuperäiseen tilaansa.

Kuva 92. WiZ-lampun WhatsApp-automaation lokitiedot.



## 7 Johtopäätökset ja pohdinta

Tavoitteena oli saada käsitys esineiden internetistä, älykoodista ja niissä käytössä olevista protokollista. Käytännössä tarkoitus oli saada HA toimimaan älykodin ohjauskeskuksena Raspberry Pi:ssä, liittää siihen älylaite ja luoda siihen muutama automaatio. Heti aloittaessa huomasin, että aihetta piti rajata huomattavasti, sillä tietoa löytyi paljon. Esineiden internetillä on monta eri käyttötarkoitusta, ja tieto oli jakautunut käyttötarkoituksen mukaan erilaisiin näkökulmiin. Tämä toi tiedonhakuun haasteita. Käytössä olevia protokollia on kymmeniä erilaisia, joista valitsin tutkittavaksi vain kolme. Jokaisesta työssä läpikäydyistä aiheista saisi mielestäni helposti aikaiseksi oman opinnäytetyön.

Esineiden internet tai älykodit eivät olleet itselleni kauhean tuttu aihe teknisesti, enkä Raspberry Pi:tä ollut ennen käyttänyt, joten opittavaa oli paljon. Esineiden internetin arkkitehtuurin, protokollien ja niiden toiminnan ymmärtämiseen meni selvästi eniten aikaa. HA:n asennuksesta ja käytöstä löytyi kattavat ohjeet netistä, joita lukemalla oma ratkaisu saatiin hyvin aikaiseksi. YAML-merkintäkoodin rakenteiden oppimisessa meni aikaa, enkä kaikkia HA:n automaatiomahdollisuuksia vielä oppinutkaan. Automaatioiden luomisessa on aika lailla vain mielikuvitus rajana. Shelly valittui älylaitteeksi energiasäästöjen mielessä, ja tämän se kyllä mahdollistaakin. Sen asennus oli helppo ja siihen liitettyjen laitteiden energiankulutusta voidaan seurata HA:n käyttöliittymästä vaivattomasti. WiZ:in käyttöönotto oli yhtä helppo, kuin Shellynkin. WiZ:iin otin aiheeksi luoda monimutkaisempia automaatioita käyttäen YAML-merkintäkoodia. HA:ssa on kätevä YAML-merkintäkoodin testauskäyttöliittymä, jossa pystyy nopeasti kokeilemaan omia ideoita. Paikallisten älykotiratkaisujen edut saavat minut välttämään jatkossa pilviratkaisuja täysin. Tutkimuskysymyksiin saatiin mielestäni vastattua.

Opinnäytetyön myötä itselleni on jäänyt kiinnostusta jatkaa ja laajentaa omaa älykotiratkaisua. Raspberry Pi:ssä käytetty microSD-kortti voitaisiin vaihtaa ulkoiseen SSD-kovalevyyn. HA:n foorumeita tutkiessa selvisi, että microSD-kortit eivät sovellu HA:n pyörittämiseen ja menevät helposti rikki.

Airam Wi-Fi-älylampun integrointi epäonnistui, sillä siitä on poistettu HA-integraatio. Integraatio HA:han voitaisiin mahdollisesti kumminkin toteuttaa käyttäen Tasmota-



ohjelmistoa, jonka avulla IoT-laitteiden laiteohjelmisto voidaan korvata Tasmotan laiteohjelmistolla. Tasmotalla on olemassa oleva integraatio HA:ssa, joten todennäköisesti Airamien laiteohjelmiston korvattua se saataisiin toimimaan jollain tasolla HA:ssa. Tasmotan käyttö olisi myös hyvä aihe opinnäytetyölle tulevaisuutta ajatellen. Suosittelemme pitämään myös silmällä Thread- ja Matter-protokollia, jotka ovat uusia kehitteillä olevia älykoteja varten suunniteltuja protokollia. Thread ja Matter lupaavat täyden yhteensopivuuden eri laitteiden välille.

## 8 Yhteenveto

Suunnitelmana oli tutkia esineiden internetiä, älykoteja, sekä niissä käytettyjä protokollia. Esineiden internetistä saatiin hyvä käsitys, miten järjestelmät ja laitteet luokitellaan osaksi sitä. Esineiden internet yllätti laajuudellaan. Jotkin käyttämistäni lähteistä tutki esineiden internetiä teollisuusjärjestelmien osalta (IIoT, Industrial Internet of Things), joten välillä oli vaikeaa hahmottaa, miten asioita saisi peilattua älykoteihin sopivaksi. Teollisuuden esineiden internet on aivan oma maailmansa älykotien esineiden internetistä. Lisäksi haasteita loi esineiden internetistä käytetyt erilaiset termit, kuten Internet of Everything (IoE) ja Web of Things (WoT), jotka myös käsittelevät esineiden internetiä eri näkökulmista.

Älykotien tuomat laajat mahdollisuudet ja hyödyt saatiin esille, mutta niissä piilee myös paljon erilaisia tietoturvariskejä. Kannattaa siis miettiä, mikä on laitteiden tuoma hyöty verrattuna riskeihin. Esineiden internetiä ei olla kehitetty keskitetysti, jonka takia protokollia on hyvin monenlaisia eri ominaisuuksilla, vahvuuksilla ja heikkouksilla. Kolme käsittelyyn otetut protokollat vaihtuivat opinnäytetyötä kirjoittaessa moneen otteeseen, mutta valituilla protokollilla oli yhteistä se, että ne kaikki ovat yhteensopivia HA:n kanssa. Protokollat ovat suunniteltu pieni energiankulutus mielessä, sillä suurin osa esineiden internetin laitteista ovat patterillisia.

Raspberry Pi:tä en ollut ennen käyttänyt, mutta olin kyllä tutkinut sitä ja sen toiminnallisuuksia. HA:n asennus siihen oli helppoa, eikä siihen kulunut kauaa aikaa. HA vakuutti laajoilla integraatiomahdollisuuksilla. Kirjoitushetkellä se mahdollistaakin yli 2000 eri laitteen suoran integraation. Shellyn integraatio toteutui vain parilla klikkauksella ja automaatioiden luonti tapahtui nopeasti käyttöliittymän avulla. WiZ:iä käyttämällä saatiin luotua haastavampia integraatioita YAML-merkintäkoodilla. YAML on tarpeellinen vain erikoisimpiin integraatioihin, kuten WhatsAppin viestiseuraajaan. Toteutuksessa saatiin aikaan se, mitä lähdettiin tavoittelemaan, eli oma paikallinen älykotiratkaisu saatiin luotua HA sen ohjauskeskuksena. Paikallisena älykotiratkaisuna HA toimii mahtavasti. Netistä löytyvät kattavat ohjeet sen käyttöönotosta ja eri toiminnoista. Shellylla seurattiin siihen kiinnitettyjen laitteiden energiankulutusta ja luotiin näiden perusteella säästöjä aikaansaavia automaatioita. WiZ:illä sen sijaan leikittiin enemmän luovuudella. Älykotiratkaisu saatiin dokumentoitua niin, että samanlainen ratkaisu voidaan saada aikaan sitä seuraamalla.

## Lähteet

altexsoft. (2022). *IoT Architecture Building Blocks*.

<https://content.altexsoft.com/media/2020/08/iot-architecture-building-blocks.png.webp>

Azzola, F. (2018, marraskuuta 8). *CoAP Protocol: Step-by-Step Guide - DZone*. Dzone.Com.

<https://dzone.com/articles/coap-protocol-step-by-step-guide>

Basumallick, C. (2022, kesäkuuta 22). Zigbee vs. Z-Wave: Understanding 10 Key Differences.

*Spiceworks*. <https://www.spiceworks.com/tech/iot/articles/zigbee-vs-z-wave/>

Bell, D. (2022, syyskuuta 17). *Choosing A Smart Home Controller*. Vesternet.

<https://www.vesternet.com/en-eu/pages/choosing-a-smart-home-controller>

Beshkov, M. (2023, helmikuuta 20). *CoAP Protocol: What is Meaning, Architecture and*

*Function*. <https://www.wallarm.com/what/coap-protocol-definition>

Catchpoint. (ei pvm.). *The Guide to MQTT Broker*. Noudettu 19. helmikuuta 2023, osoitteesta

<https://www.catchpoint.com/network-admin-guide/mqtt-broker>

Christou, L. (2019, helmikuuta 27). History of IoT: From idea to an industry approaching \$1tn.

*Verdict*. <https://www.verdict.co.uk/history-of-iot/>

Collin, J., & Saarelainen, A. (2016). *Teollinen internet*. Alma Talent.

Colman, L. A. (2022, elokuuta 2). *What is Home Assistant in 2022?* Home Assistant Guide.

<https://home-assistant-guide.com/guide/what-is-home-assistant-and-what-can-it-do/>

Elias, R. (2021, tammikuuta 20). *What is a Smart Home Controller?* Strategic Home Media.

<https://www.strategichomemedia.com/blog/what-is-a-smart-home-controller-2021>

- Farnell element14. (2017, elokuuta 29). *Smart sensors – overview and latest technology* | Farnell France. <https://fr.farnell.com/en-FR/smart-sensors-overview-and-latest-technology>
- Felch, R. (2021, elokuuta 27). Understanding Zigbee and Wireless Mesh Networking. *Black Hills Information Security*. <https://www.blackhillsinfosec.com/understanding-zigbee-and-wireless-mesh-networking/>
- Fortinet. (ei pvm.). *Top IoT Device Vulnerabilities: How To Secure IoT Devices*. Fortinet. Noudettu 1. maaliskuuta 2023, osoitteesta <https://www.fortinet.com/resources/cyberglossary/iot-device-vulnerabilities>
- Gigantti. (ei pvm.). *Raspberry Pi 3 Model B+ yhden piirilevyn tietokone—Gigantti verkkokauppa*. Noudettu 2. maaliskuuta 2023, osoitteesta <https://www.gigantti.fi/product/tietokoneet-ja-toimistotarvikkeet/tietokoneet/yhden-piirilevyn-tietokoneet/raspberry-pi-3-model-b-yhden-piirilevyn-tietokone/RASPI3MBPLU16>
- Guinard, D. D., & Trifa, V. M. (2016). *Building the web of things: With examples in Node.js and Raspberry Pi*. Manning.
- HiveMQ. (ei pvm.). *MQTT Essentials—All Core Concepts explained*. Noudettu 1. maaliskuuta 2023, osoitteesta <https://www.hivemq.com/mqtt-essentials/>
- HiveMQ. (2015, kesäkuuta 22). *MQTT Security Fundamentals—Wrapping up the complete series*. <https://www.hivemq.com/blog/mqtt-security-fundamentals-wrap-up/>
- InterviewBit. (2022, kesäkuuta 3). *IoT Architecture—Detailed Explanation*. InterviewBit. <https://www.interviewbit.com/blog/iot-architecture/>
- Jakhar, A. (2020, heinäkuuta 27). *IoT Security—Part 11 (Introduction To CoAP Protocol And Security)*. Payatu. <https://payatu.com/masterclass/iot-security-part-11-introduction-to-coap-protocol-and-security/>

Kaspersky. (2021, tammikuuta 13). *How safe are smart homes?* Www.Kaspersky.Com.

<https://www.kaspersky.com/resource-center/threats/how-safe-is-your-smart-home>

Lueth, K. L. (2014, marraskuuta 19). *Why it is called Internet of Things: Definition, history,*

*disambiguation.* <https://iot-analytics.com/internet-of-things-definition/>

Malwarebytes. (ei pvm.). *What was the Mirai botnet.* Malwarebytes. Noudettu 1.

maaliskuuta 2023, osoitteesta <https://www.malwarebytes.com/what-was-the-mirai-botnet>

Microsoft Azure. (ei pvm.). *IoT Technologies and Protocols | Microsoft Azure.* Noudettu 7.

helmikuuta 2023, osoitteesta <https://azure.microsoft.com/en-us/solutions/iot/iot-technology-protocols/>

MQTT. (2022). *MQTT - The Standard for IoT Messaging.* <https://mqtt.org/>

NCS. (2022, elokuuta 9). *What is IoT Architecture? | NCS SG.* <https://www.ncs.co/en-sg/knowledge-centre/iot-architecture/>

Opensource. (ei pvm.). *What is a Raspberry Pi? | Opensource.com.* Noudettu 2. maaliskuuta

2023, osoitteesta <https://opensource.com/resources/raspberry-pi>

O'Shields, J. (2022, elokuuta 25). *IoT Architecture and Its Key Layers in Examples.* JatApp.

<https://jatapp.co/blog/iot-architecture/>

Ovens, S. (2020, marraskuuta 18). *Cloud control vs local control: What to choose for your*

*home automation | Opensource.com.* <https://opensource.com/article/20/11/cloud-vs-local-home-automation>

Premier Farnell Ltd. (2017). *Smart sensor building blocks.*

[https://fr.farnell.com/wcsstore/ExtendedSitesCatalogAssetStore/cms/asset/images/common/campaign/internet\\_of\\_things/smart-sensor-building-blocks.png](https://fr.farnell.com/wcsstore/ExtendedSitesCatalogAssetStore/cms/asset/images/common/campaign/internet_of_things/smart-sensor-building-blocks.png)

Qin, B. (2023, tammikuuta 26). *What is the Zigbee Alliance.* *Dusuniot.*

<https://www.dusuniot.com/blog/what-is-the-zigbee-alliance/>

- Ranger, S. (2020, helmikuuta 3). *What is the IoT? Everything you need to know about the Internet of Things right now*. ZDNET. <https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/>
- Raspberry Pi. (ei pvm.). *Buy a Raspberry Pi 3 Model B+*. Raspberry Pi. Noudettu 2. maaliskuuta 2023, osoitteesta <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>
- Satyabrata, J. (2020, kesäkuuta 12). *Architecture of Internet of Things (IoT)*. *GeeksforGeeks*. <https://www.geeksforgeeks.org/architecture-of-internet-of-things-iot/>
- Shelly. (ei pvm.-a). *Shelly Family Overview – API Reference*. Noudettu 2. maaliskuuta 2023, osoitteesta <https://shelly-api-docs.shelly.cloud/gen1/#shelly-family-overview>
- Shelly. (ei pvm.-b). *Shelly Plug—Automatically monitor and control lighting, heating and any other connected electrical appliance at home*. Shelly Cloud. Noudettu 2. maaliskuuta 2023, osoitteesta <https://www.shelly.cloud/en/products/shop/1xplug>
- Silicon Laboratories. (2022). *AN1233: Zigbee Security*. <https://www.silabs.com/documents/public/application-notes/an1233-zigbee-security.pdf>
- STMicroelectronics. (2022). *Connectivity Figure*. [https://wiki.st.com/stm32mcu/nsfr\\_img\\_auth.php/0/06/Connectivity\\_Figure2.png](https://wiki.st.com/stm32mcu/nsfr_img_auth.php/0/06/Connectivity_Figure2.png)
- Teicher, J. (2018, helmikuuta 7). *The little-known story of the first IoT device*. Industrious. <https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/>
- Trivedi, T. (2020, toukokuuta 26). *Internet Of Things (IoT) Sensors And Connectivity*. <https://www.c-sharpcorner.com/article/internet-of-things-iot-sensors-and-connectivity/>
- Vadakkanmarveetil, J. (2022, kesäkuuta 6). *12 Major Applications of IoT You Should Know | Jigsaw Academy*. <https://www.jigsawacademy.com/top-uses-of-iot/>

Vailshery, L. (2022, marraskuuta 22). *IoT connected devices worldwide 2019-2030*. Statista.

<https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>

Vigderman, A., & Turner, G. (2022, toukokuuta 12). What Is Home Automation and How

Does It Work? *Security.Org*. <https://www.security.org/home-automation/>

WiZ. (ei pvm.). *MODERNI LAMPPU Lamppu 8,8 W (vastaa 60 W) E27*. WiZ. Noudettu 19.

lokakuuta 2023, osoitteesta [https://www.wizconnected.com/fi-fi/p/moderni-](https://www.wizconnected.com/fi-fi/p/moderni-lamppu-lamppu-8-8-w--vastaa-60-w--e27/8720169072299)

[lamppu-lamppu-8-8-w--vastaa-60-w--e27/8720169072299](https://www.wizconnected.com/fi-fi/p/moderni-lamppu-lamppu-8-8-w--vastaa-60-w--e27/8720169072299)

Zhou, Z. (2023, tammikuuta 12). *Introduction to MQTT QoS 0, 1, 2*. *Www.Emqx.Com*.

<https://www.emqx.com/en/blog/introduction-to-mqtt-qos>

## **Liite 1: Aineistonhallintasuunnitelma**

Opinnäytetyössä ei käsitellä henkilötietoja tai muita luottamuksellisia tai salassa pidettäviä tietoja.

Kehitysprojektin aikana pidetään päiväkirjaa, johon kerätään teknistä tietoa projektista. Tämä tieto analysoidaan opinnäytetyötä varten. Päiväkirjaa säilytetään tekijän tietokoneen C-aseamalla, ja siitä tehdään säännöllisesti varmuuskopioita ulkoiselle kovalevyllä, sekä kahdelle eri muistitikulle.

Päiväkirjaa säilytetään C-aseamalla, ulkoisella kovalevyllä ja kahdella muistitikulla ainakin vuoden verran opinnäytetyön valmistumisesta.

Valmiin projektin onnistumisesta kerätään tietoa ottamalla kuvakaappauksia eri asennuksien vaiheista, sekä ohjelmistoista. Niiden säilytys ja varmuuskopiointi tapahtuu samalla tavalla kuin päiväkirjan kanssa.



**Liite 2: Paikallaolointegraation YAML-koodi**

```
# KODIN PAIKALLAOLO
device_tracker:
  - platform: ping
    hosts:
      nuutti_puhelin: 192.168.1.33
    count: 4
    interval_seconds: 20
    consider_home: 60
```

**Liite 3. WhatsApp viestin tallennusintegraation YAML-koodi**

```
wa_last_message:
  friendly_name: "WhatsApp last message"
  value_template: >
    {% set notification_list =
state_attr('sensor.samsung_nuutti_last_notification', 'android.textLines')
%}

    {% if notification_list is not none %}
      {% set message_last = notification_list | last | upper %}
    {% else %}
      {% set message_last =
states('sensor.samsung_nuutti_last_notification') | upper %}
    {% endif %}

    {{ message_last }}
```

**Liite 4. WhatsApp laukaisusanan tarkistusintegraation YAML-koodi**

```
- platform: template
  sensors:
    wa_last_message_contains_trigger_word:
      friendly_name: "WhatsApp last message contains trigger word"
      value_template: >
        {% set test_for_message = states.input_text.wa_sana.state | upper
%}

        {% if test_for_message in states('sensor.wa_last_message') %}
          true
        {% else %}
          false
        {% endif %}
```