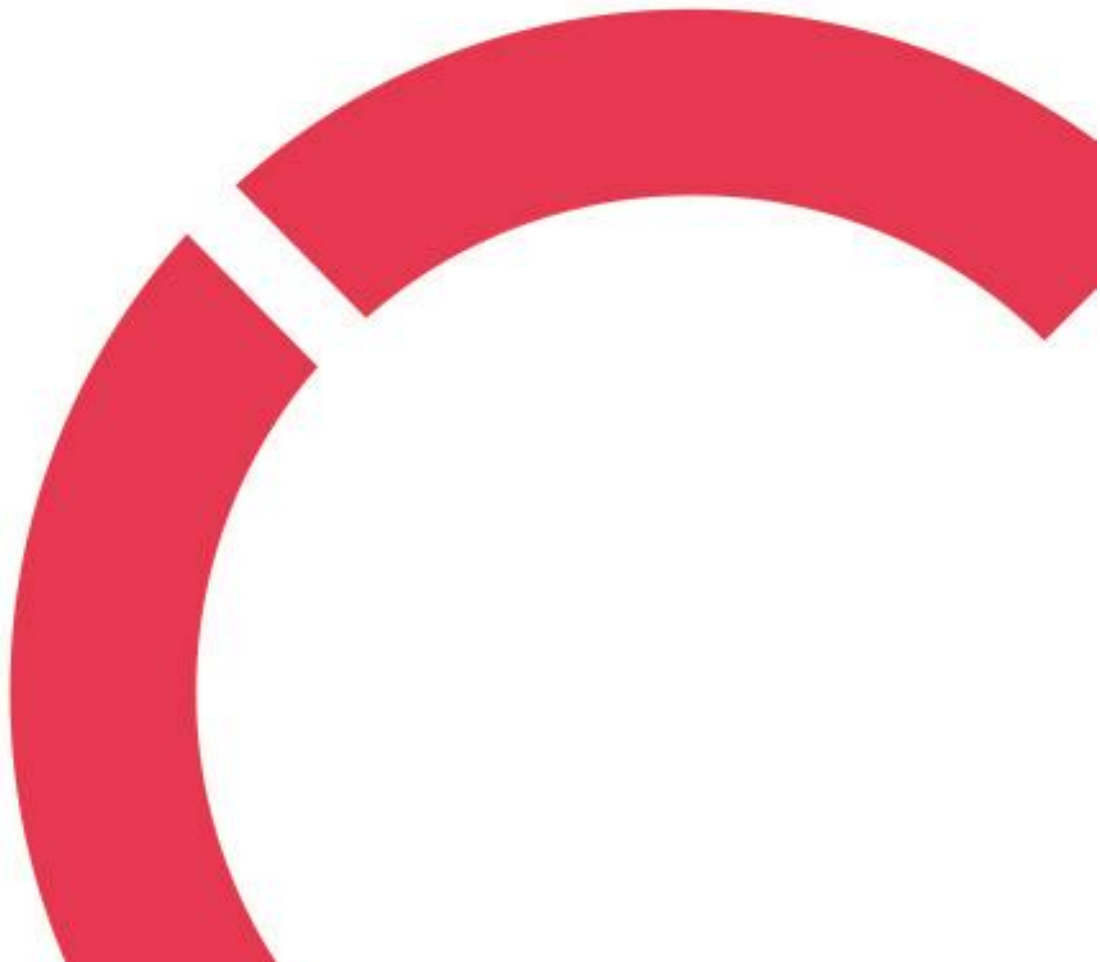


**Roni Muilu**

**MQTT-PROTOKOLLAN HYÖDYNTÄMINEN ARDUINOLAITTEIDEN P2P-VIESTINNÄSSÄ**

**Opinnäytetyö  
CENTRIA-AMMATTIKORKEAKOULU  
Tieto- ja viestintäteknikan koulutus  
Tammikuu 2023**



## TIIVISTELMÄ OPINNÄYTETYÖSTÄ

<b>Centria-ammattikorkeakoulu</b>	<b>Aika</b> Tammikuu 2023	<b>Tekijä/tekijät</b> Roni Muilu
<b>Koulutus</b> Tieto- ja viestintäteknikka		<input checked="" type="checkbox"/> AMK <input type="checkbox"/> YAMK
<b>Työn nimi</b> MQTT-PROTOKOLLAN HYÖDYNTÄMINEN ARDUINOLAITTEIDEN P2P-VIESTINNÄSSÄ		
<b>Työn ohjaaja</b> Jari Isohanni		<b>Sivumäärä</b> 26 + 2
<b>Työelämäohjaaja</b>		
<p>Tässä työssä esitellään keskeisiä asioita Arduino-kehitysalustoista, digitalisaatiosta sekä esineiden internetiin liittyvistä protokollista. Työssä kerrotaan myös lyhyesti digitalisaation mukanaan tuomista ilmiöistä ja niihin liittyvistä tietoturvaasteista sekä pohditaan, miksi on tärkeää huomioida tietoturvasuus erityisesti IoT-laitteiden kanssa.</p> <p>Työ aloitettiin tutustumalla erityyppisiin lähteisiin, jotka käsittelevät MQTT-protokollaa, esineiden internetiä, tietoturvaa sekä Arduino-kehitysalustaa. Lähteissä oli materiaalia laajasti ja eniten tietoa löytyi englannin kielellä. Työn lopullisiksi lähteiksi valittiin sellaisia, jotka keskittyvät eniten työn kannalta olennaisiin asioihin. Tässä työssä hyödynnettiin lähteinä monipuolisesti kirjoja, verkosta löytyvää materiaalia ja opinnäytetöitä.</p> <p>Työn käytännön osuudessa rakennetaan Arduino-kehitysalustoista laitteistokokonaisuus, jonka tavoitteena on mitata ympäristönsä lämpötila ja välittää tämä tieto MQTT-protokollaa hyödyntäen laitteelta toiselle. Toisen laitteen tavoite on reagoida ensimmäisen laitteen julkaisemaan lämpötilaan ja kytkeä tarvittaessa päälle lämmitin, jota tässä työssä kuvataan LED-diodilla. Käytännön osuus alkaa lähettävän ja vastaanottavan laitteiston kokoamisella, jossa yhdistetään Arduino-kehitysalustat sensoreihin ja muihin tarvittaviin osiin johtimien avulla. Tämän jälkeen asennetaan Arduino IDE-kehitysympäristöön tarvittavat lisäkirjastot ja kirjoitetaan molemmille Arduinolaitteille oma ohjelmakoodi, joka ohjaa niiden toimintaa. Lopputuloksena saatiin rakennettua kokonaisuus, jossa kaksi arduinolaitetta viestii keskenään.</p> <p>Työn loppuosassa esitellään käytännön osan tulokset ja analysoidaan niitä. Lisäksi pohditaan, millaisissa ympäristöissä käytännön osassa rakennettua laitetta voitaisiin hyödyntää ja kehittää edelleen.</p>		
<b>Asiasanat</b> Arduino, Internet of Things, MQTT, Sulautetut järjestelmät, Tietoliikenne, Tietotekniikka		

## ABSTRACT

<b>Centria University of Applied Sciences</b>	<b>Date</b> January 2023	<b>Author</b> Roni Muilu
<b>Degree programme</b> Bachelor of Engineering, Information technology		
<b>Name of thesis</b> Usage of MQTT-protocol in P2P-communication of Arduino devices		
<b>Centria supervisor</b> Jari Isohanni	<b>Pages</b> 26 + 2	
<b>Instructor representing commissioning institution or company</b>		
<p>The thesis introduces the reader basics of Arduino development boards, digitalization and common protocols of the Internet of Things (IoT). The thesis focuses mainly on MQTT protocol. IoT is very recent topic and this thesis also points out briefly the main security challenges that this new technology does face and also why it is important to take care of digital security, especially with IoT devices.</p> <p>The work was started by gathering recent information from books, internet pages and thesis about MQTT protocol, IoT and Arduino development boards. There was lots of information available, mostly in english. Only the most relevant sources from the perspective of this thesis were selected. Many different kinds of sources were used in this thesis.</p> <p>In the practical part of the thesis, device that is using Arduino development boards will be built. Its function is to measure temperature and send the information to other arduino, via internet broker server and using MQTT protocol. Other device subscribes this information and will eventually switch on heater which is simulated with LED-diode. In the beginning of the practical part of the thesis Arduino-development boards and their add-ons will be connected with jumper cables. Then needed libraries will be installed on Arduino IDE-software and programming source code will be written for each Arduino device. In the result of this there will be a system where two Arduino devices communicate each other.</p> <p>In the final part of the thesis, practical part results will be introduced and analysed. In addition it is thought what kind of places it would be useful to have that kind of device and how it can be improved further.</p>		
<b>Key words</b> Arduino, Embedded Systems, Internet of Things, MQTT, P2P, Information Technology		

## **KÄSITTEIDEN MÄÄRITTELY**

### **BROKER**

MQTT-protokollan viestien välittämisestä vastaava sovellus.

### **CLIENT**

Client (asiakas) on mikä tahansa laitteisto tai ohjelmisto, joka on yhteyksissä serverin tarjoamiin palveluihin.

### **GPIO**

(General Purpose Input/Output) Yleiskäyttöinen pientietokoneissa käytettävä liitäntäpinni joka toimii käyttötarkoituksesta riippuen joko saapuvaan tai lähtevään liikenteeseen.

### **IoT**

Internet of Things (esineiden internet) on ympäristö, jossa tavanomaiset laitteet toimivat verkossa ja tuottavat tai saavat sieltä tietoa.

### **IP-osoite**

Tulee sanoista Internet Protocol. Numeroista koostuva osoite jolla yksilöidään verkossa olevat laitteet.

### **Julkaisija/Tilaaja-periaate**

MQTT-protokollan tapa luokitella lähtevää ja saapuvaa viestiliikennettä.

### **Koekytkentälevy**

(Breadboard) on kytkentäpiirien testaamiseen suunniteltu levy. Levyssä on reikiä joihin pinneillä varustetut johdot ja lisälaitteet voidaan kytkeä.

### **MAC-osoite**

Mac-osoite on yksilöllinen koodi jonka valmistaja on määrittänyt tietyille verkkolaitteiden ryhmälle. Mac-osoite on lyhenne sanoista Media Access Control. Jokainen osoite on laitekohtainen.

### **MCU**

(Microcontroller Unit) Mikrokontrolleri, pienikokoinen tietokonelastu.

## **MQTT**

(Message Queue Telemetry Transport) on IBM:n kehittämä protokolla laitteiden väliseen viestintään.

## **P2P**

Hajautettu tietoverkkoarkkitehtuuri, jossa jokainen osallistuja toimii sekä asiakkaana että palvelimena muille jäsenille.

## **PUBSUB-arkkitehtuuri**

(Publish/Subscribe) on MQTT-protokollan tapa välittää tietoa. Tietoa voidaan julkaista ja tilata MQTT-Brokerilta.

## **QoS**

(engl. Quality of Service) on termi, jolla tarkoitetaan tietoliikenteen luokittelua ja priorisointia. MQTT käyttää QoS priorisointia.

## **Sketch**

Tarkoittaa ohjelmakoodia, joka käännetään konekieliseksi ja ladataan kehitysalustan muistiin.

## **SSH**

(Secure Shell) on salattuun etäyhteyteen käytettävä tietoliikenteen protokolla.

## **TCP/IP**

Transmission Control Protocol/Internet Protocol, tiedonsiirtoprotokolla.

## **TLS/SSL**

Transport Layer Security (TLS) on seuraaja vanhentuneelle Secure Sockets Layer (SSL) salausprotokollalle, jolla voidaan suojata Internet-sovellusten tietoliikenne IP-verkkojen yli.

## **WiFi/WLAN**

Langaton tietoliikenneteknologia.

## **ÄLYKOTI (Smart Home)**

Älykoti tarkoittaa kotia, jossa valaistusta, lämmitystä ja elektronisia laitteita voidaan hallinnoida etänä esimerkiksi tietokoneen, älypuhelimien tai älyrannekkeen avulla.

**TIIVISTELMÄ**  
**ABSTRACT**  
**KÄSITTEIDEN MÄÄRITTELY**  
**SISÄLLYS**

<b>1 JOHDANTO</b> .....	<b>1</b>
<b>2 TAUSTATIETOA</b> .....	<b>2</b>
2.1 Digitalisaatio ja IoT.....	2
2.2 Arduino-alusta.....	3
2.3 Internet of Things (IoT).....	6
2.3.1 Yleistä.....	6
2.3.2 Protokollat.....	7
2.3.3 Tietoturva ja riskit.....	13
<b>3 MQTT-VIESTIN LÄHETYS JA VASTAANOTTO ARDUINOLLA</b> .....	<b>16</b>
3.1 Käytännön työn tavoite.....	16
3.2 Komponentit ja kytkennät.....	16
3.3 Ohjelmointi ja laitteiden konfigurointi.....	18
3.4 Viestin lähetys ja vastaanotto.....	20
<b>4 TULOKSET</b> .....	<b>24</b>
<b>5 POHDINTA</b> .....	<b>26</b>
<b>LÄHTEET</b> .....	<b>27</b>
<b>LIITTEET</b>	
<b>KUVIOT</b>	
KUVIO 1. IoT-ekosysteemin 5 pääkomponenttia.....	2
KUVIO 2. HTTP:n toiminta kun webiselain lataa sivua.....	8
KUVIO 3. Tiedon välittäminen MQTT:ssä.....	9
KUVIO 4. Esimerkki topic-tunnisteiden käytöstä.....	10
KUVIO 5. WiFi-verkon ja mesh-verkon eroavaisuudet.....	13
<b>KUVAT</b>	
KUVA 1. Arduinon Uno- ja Nano-kehitysalustojen klooneja.....	4
KUVA 2. Internet of Thingsin haasteita.....	15
KUVA 3. Käytännön osan laitteiden kytkennät.....	17
KUVA 4. Brokerin IP-osoitteen selvittäminen.....	19
KUVA 5. MQTT Explorerin käyttöliittymä ja sen asetukset.....	20
KUVA 6. Arduino IDE-ohjelma.....	21
KUVA 7. Sarjaporttimonitori ja lähettävän laitteen liikennettä.....	22
KUVA 8. Sarjaporttimonitorin liikennettä tilaavan ESP32-laitteen puolelta.....	23
KUVA 9. Lähetetyt tiedot Mosquitton testibrokerilla.....	24
<b>TAULUKOT</b>	
TAULUKKO 1. Yleisimpien Arduinon kehitysalustojen teknisiä tietoja.....	5
TAULUKKO 2. QoS-luokitus ja sen vaikutus viestintään.....	10

TAULUKKO 3. Connect-datakehys.....	11
TAULUKKO 4. Connack-datakehysten sisältö.....	11
TAULUKKO 5. Connack-paketin palautuskoodin arvot. ....	12
TAULUKKO 6. Subscribe ja suback-pakettien sisältö .....	12
TAULUKKO 7. Publish- ja puback-pakettien sisältö .....	12

## 1 JOHDANTO

Opinnäytetyön tavoitteena on selventää lukijalle, mitä ovat Arduino-kehitysalustat, MQTT-protokolla ja esineiden internet (IoT).

Tietoperustassa esitellään lyhyesti Arduino-mikrokontrollerien keskeisimpiä ominaisuuksia, kuten pinnit ja liitännät. Lisäksi esitellään ja vertaillaan Internet of Thingsin (IoT) kannalta keskeisimpiä internetin protokollia ja niiden toimintaa ja tarkastellaan näiden hyödyntämismahdollisuuksia.

Opinnäytetyön käytännön osassa rakennetaan kahdesta Arduino-kehitysalustasta ja niiden lisälaitteista koostuva laitekokonaisuus. Laitekokonaisuudessa toinen kehitysalusta julkaisee tiedon MQTT-protokollaa hyödyntäen ja broker-välittäjäohjelmiston välityksellä toisen alustan käytettäväksi. Toinen kehitysalusta säätelee toimintaansa saamansa tiedon mukaan.

Tavoitteena on tuottaa Arduino-kehitysalustan ja siihen kytkettävien lisälaitteiden avulla sellainen kytkentä, jolla voidaan lähettää langattomasti tietoa verkon kautta toiseen laitteeseen. Tiedon välittämiseen käytetään tässä opinnäytetyössä MQTT-protokollaa.

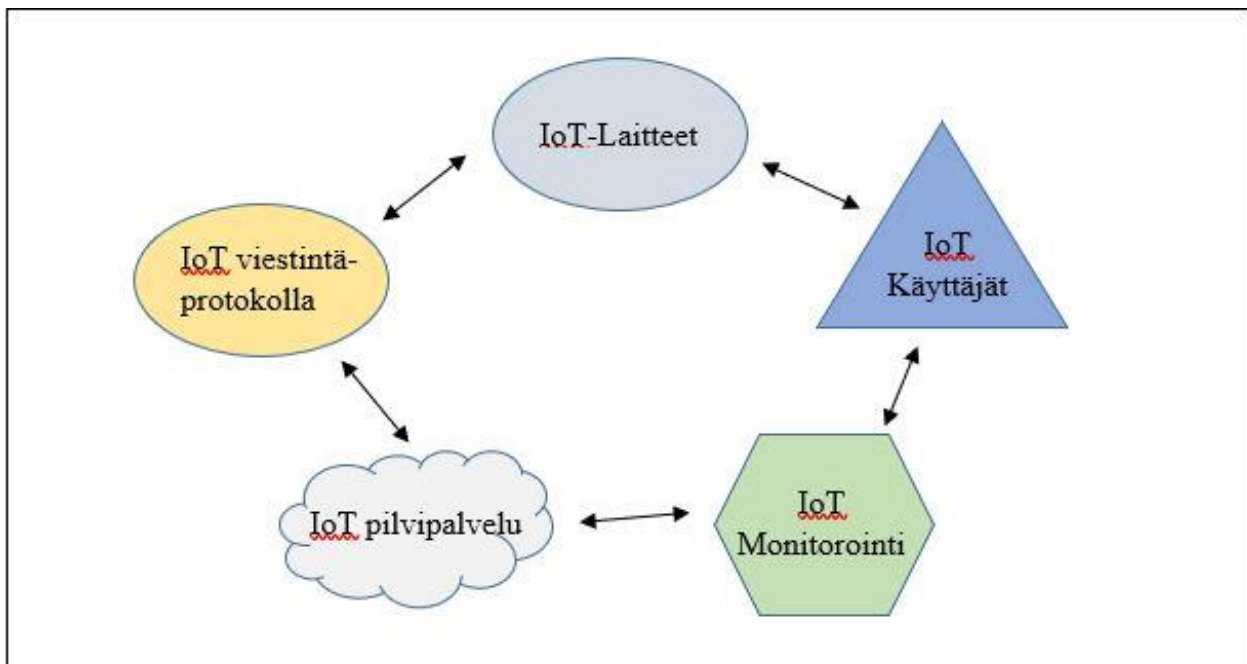


## 2 TAUSTATIETOA

### 2.1 Digitalisaatio ja IoT

Digitalisaatio tarkoittaa perinteisten toimintojen siirtämistä verkkoihin. Tämä kehitys on jatkunut jo 1970-luvulta alkaen ja jatkuu kiihtyvällä tahdilla edelleen. Nykyään lähes kaikenlaiseen toimintaan liittyy verkot. Jos laite on mahdollista liittää verkkoon tai ohjata sitä verkon välityksellä, yleensä näin tehdään. (Moilanen 2022.)

Internet of Things (IoT) on ilmiö, joka on syntynyt digitalisaation seurauksena. Yleensä ajatellaan, että internet on oma erillinen ympäristönsä ja koskee vain tietokoneita tai älypuhelimia. Vaikka verkot ovat näkymättömiä, ne vaikuttavat entistä enemmän myös fyysisen ja näkyvän maailman toimintoihin. (Moilanen 2022.). Verkkoon liitettyjen erityyppisten laitteiden määrä kasvaa. On yhä yleisempää, että verkkojen välityksellä on myös mahdollista ohjata fyysisiä laitteita ja järjestelmiä. (Moilanen 2022.). IoT-ekosysteemi (KUVIO 1) koostuu viidestä tärkeästä pääkomponentista: laitteet, viestintäprotokolla, pilvipalvelu, monitorointi sekä käyttäjät. (Eshghi 2023.).



KUVIO 1. IoT-ekosysteemin 5 pääkomponenttia. Mukailten Eshghi 2023.

IoT-laitteilla on monia käyttötarkoituksia kotitalouksille. Käyttötarkoitukset ovat pääasiassa arjen helpottaminen, turvallisuus sekä erilaisten asioiden monitorointi vaikkapa energiatehokkuuden vuoksi. Ympäristön ajattelu ja kuluissa säästäminen ovat järkevää ajattelua. Uusissa kodinkoneissa on hyvin usein mukana verkkotoiminnallisuuksia. Älykäs jääkaappi saattaa lähettää reseptiehdotuksia kaapissa olevien raaka-aineiden mukaan tai pesukone valitsee itse ohjelman ja pitää käyttäjän ajan tasalla tapahtumista. (Älykäs jääkaappi tunnistaa jo puheen 4.9.2018.)

Teollisuudessa IoT-laitteilla voidaan automatisoida prosesseja, valvoa tuotantolinjan turvallisuuteen tai tuotteiden laatuun liittyviä asioita antureiden avulla. Verkon välityksellä voidaan myös päivittää laitteiden ohjelmistoja sekä seurata tarkasti prosessien kulkua ja valmistusmateriaalien kulutusta. IoT:llä on tärkeä rooli tuotannon automatisoinnissa. Uuden teknologian avulla laitteet voivat kommunikoida keskenään ja ohjelmistojen kanssa. (Esineiden internetin hyödyt valmistavassa teollisuudessa 2023.)

Statistan arvioiden mukaan IoT-laitteita oli vuonna 2021 yli 10 miljardia ja laitteiden määrä kasvaa huimaa vauhtia. Statista arvioi laitteiden määrän ylittävän 20 miljardin rajan vuoteen 2030 mennessä. (Statista 2021.).

## **2.2 Arduino-alusta**

Arduino on helppoon prototyyppien kehittämiseen tarkoitettu kehitysalusta, joka perustuu 8-bittiseen Atmel AVR mikrokontrolleriin tai 32-bittiseen Atmel ARM mikrokontrolleriin. (Karppila 2014; arduino a.). Mikrokontrolleri on lastun muodossa oleva pienoiskokoinen tietokone.



KUVA 1. Arduinon Uno- ja Nano-kehitysalustojen klooneja. Kuvassa Elegoon valmistama Uno ja Geekcreitin valmistama Nano.

Arduinoa on saatavilla montaa eri mallia, joiden keskeisin ero on fyysinen koko sekä lähtöjen ja muiden määrä. Yleisimmistä Arduinon alustoista (esimerkiksi Uno ja Mega) on saatavilla toisten piirilevyvalmistajien tekemiä klooneja, jotka vastaavat toiminnaltaan alkuperäistä Arduinon tekemää alustaa. Useat alustat sisältävät 5 voltin lineaarisen jännitteensäätimen ja 16 MHz:n kideoskillaattorin. (Doukas 2012, 67.). Arduinon lähdeistä ulospäin saatava jännite on tyypillisesti 5 V mutta niissä on myös 3,3V lähtö, koska osa lisälaitteista toimii pienemmällä jännitteellä. Tarvitsemansa käyttöjännitteen Arduino saa yleensä USB-johdosta, erillisestä virtalähteestä tai 9 V patterista. Useimpien Arduino-kehitysalustojen toimintajännite on viisi voltia, mutta jotkin kehitysalustoista toimivat 3,3 voltin jännitteellä. (Karpila 2014, 8.).

TAULUKKO 1. Yleisimpien Arduinon kehitysalustojen teknisiä tietoja. Mukailten arduino c.

Arduino-kehitysalusta	Proessori	Muisti	Digitaalisia lähtöjä	Analogisia lähtöjä
Arduino Uno	16 MHz ATmega328	2KB SRAM 32KB flash	14	6 sisään, 0 ulos
Arduino Due	84MHz AT91SAM3X8E	96KB SRAM 512KB flash	54	12 sisään, 2 ulos
Arduino Mega	16 MHz ATmega2560	8KB SRAM 256KB flash	54	16 sisään, 0 ulos
Arduino Leonardo	16 MHz ATmega32u4	2.5KB SRAM 32KB flash	20	12 sisään, 0 ulos
Arduino UNO WiFi	16 MHz ATmega328	2KB SRAM 32KB flash	20	6 sisään, 0 ulos
Arduino UNO WiFi Rev. 2	16 MHz ATmega4809	6KB SRAM 48KB flash	14	6 sisään, 0 ulos

Arduino-kehitysalustassa on prosessori, muistia ja useita ohjelmoitavia lähtöjä ja tuloja lisälaitteiden liittämistä varten. Arduino-kehitysalustaa ohjelmoidaan kytkemällä se USB-kaapelilla tai langattomalla yhteydellä tietokoneeseen. Tähän tarvitaan yleensä erillinen Arduino IDE-sovellus, johon kirjoitetaan ohjelmakoodi. Ohjelma kääntää ohjelmakoodin ja tallentaa mikrokontrollerin muistiin. Koska laitteita ja yhteystyyppejä on olemassa lukematon määrä, usein joudutaan lataamaan Arduino-sovellukseen lisäkirjastoja, jotta laitteiden ohjelmointi olisi sujuvampaa. Lisäkirjasto on kokoelma lähdekoodia, joka tekee lisälaitteiden kytkemisestä Arduinoon helppoa. (Arduino d.).

Arduino-ohjelmisto perustuu avoimeen lähdekoodiin, ja sen ohjelmistoa voidaan laajentaa C++-kirjastoilla. Arduino-piirien piirustukset ovat avoimesti saatavilla, joten piirilevyjen valmistajat voivat tehdä omia laajennettuja ja paranneltuja versioita Arduinosta. Toisin kuin monet muut mikrokontrollerit, Arduino tukee Windowsin lisäksi myös MacOS- ja Linux-käyttöjärjestelmiä. (Arduino a.)

Lisälaitteet voivat olla lähes mitä tahansa: esimerkiksi antureita, kytkimiä, releitä, näyttöjä tai radiotaajuuslaitteita kuten Wifi. Mikrokontrollerit ovat yksinkertainen ja helposti lähestyttävä tapa tehdä omia laitteita, ja nykyään monet tekniikkaharrastajat käyttävät niitä esimerkiksi kodin laitteiden ja toimintojen automatisointiin ja ohjaamiseen.

Arduino on suunniteltu virtapiirien ja omien laitteiden kehittämiseen. Arduino-sarjoissa tulee usein mukana koekytkentälevy, jonka reikiin komponentit voidaan painaa. Koska lisälaitteissa ja johtojen päissä on pinnit tai reiät, voidaan komponentit yhdistää ja virtapiirejä testata helposti ilman tinaamista. (Arduino b.)

## 2.3 Internet of Things (IoT)

### 2.3.1 Yleistä

Internet of Things tarkoittaa sitä, että erilaiset laitteet ovat saaneet kykyä vastaanottaa ja tuottaa tietoa verkossa. Näin esimerkiksi kodin laitteiden tilaa voi seurata matkalla ja valvoa ja ohjata kodin turvallisuutta silloin, kun ei ole paikalla. Yksinkertaisimmillaan IoT-laitteet ovat langattomia tai langallisia antureita, jotka mittaavat jotakin suuretta ja välittävät tämän tiedon edelleen jalostettavaksi. (Isohanni, Heikkilä, Malkamo & Himanka 2022.).

Tyypillisesti IoT-laitteita varten on olemassa oma gateway-reititin kutakin protokollaa varten. Reititin voi olla fyysinen laite tai ohjelmisto, joka toimii yhteyspisteenä pilvipalveluille, älylaitteille ja antureille. (Posey 2022.). Reitittimeen voidaan kytkeä esimerkiksi erilaisia sähköpistokkeita, antureita, valaistukseen liittyviä komponentteja ja muita IoT-laitteita. Reititin on yleensä yhdistetty kodin sisäverkkoon, jonka kautta saadaan yhteys internetiin. Jos käytetään useamman eri protokollan laitteita, jokaiselle protokollalle tarvitaan oma gateway-reititin. Vaihtoehtoisesti väliin voidaan laittaa laitteita, jotka muuntavat protokollia yhteensopiviksi. Kaikki IoT-laitteiden ja pilvipalvelujen välinen data kulkee reitittimen kautta. (Posey 2022.).

Esineiden internetiin liittyvät olennaisesti P2P-verkot. Ne ovat verkkoja, jotka toimivat ilman keskitettyä hallintaa ja ovat skaalautuvia. Skaalautuvuus tarkoittaa sitä, että P2P-protokolla toimii sellaisenaan, vaikka verkkoon tulisi uusia jäseniä. (P2P-järjestelmät) Myös MQTT on suunniteltu siten, että verkot ovat samaan tapaan skaalautuvia.

## 2.3.2 Protokollat

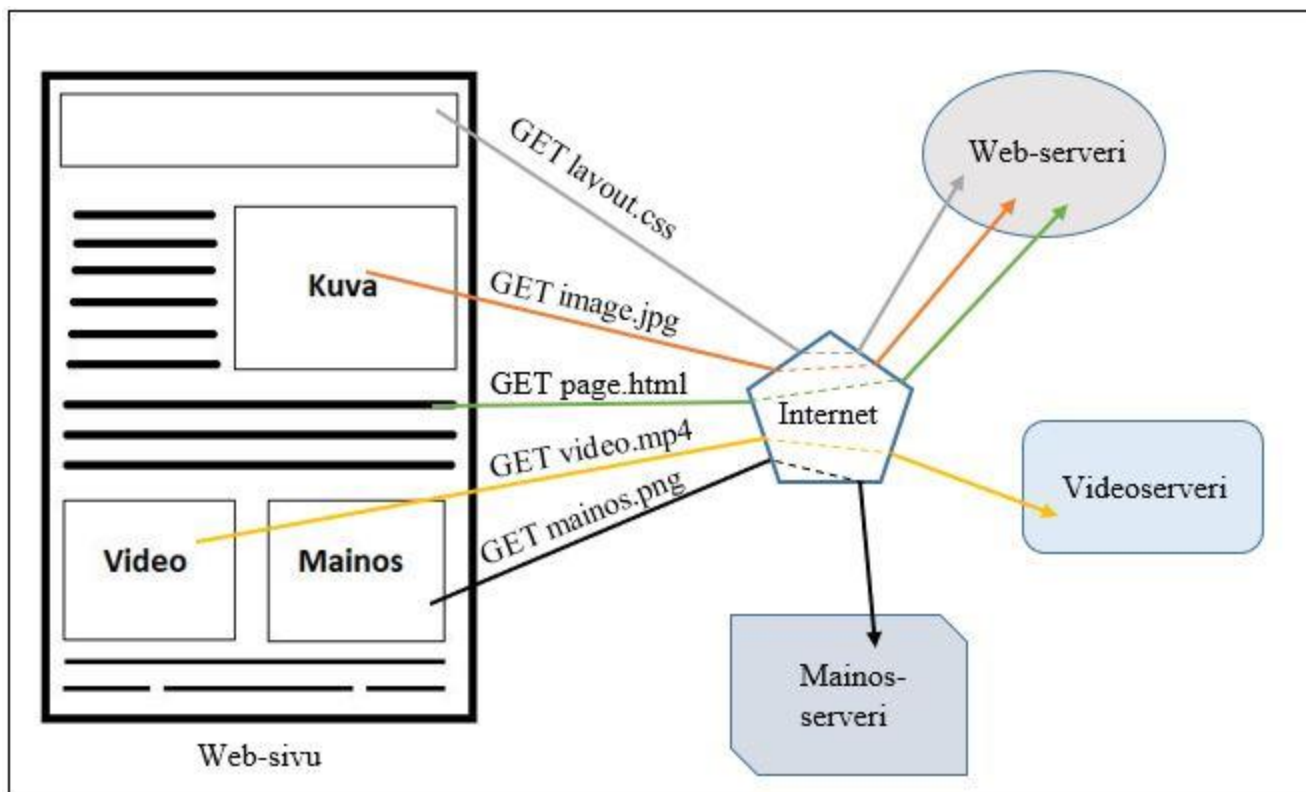
Protokollat ovat sääntöjä ja käytäntöjä, joita käytetään tiettyihin erityistarkoituksiin. Jotta laitteet ja sovellukset ymmärtävät toisiaan, on niiden käytettävä samoja ennalta sovittuja periaatteita viestinnässä. Internetissä käytetään useita eri protokollia, joista kerron tarkemmin alempana.

Esineiden internet on suhteellisen tuore ilmiö. IoT-älylaitteiden valmistajia ja protokollia on lukuisia ja tästä johtuen järjestelmien yhteensopivuudessa on vielä ollut haasteita. Ongelman ratkaisemiseksi on luotu Matter 1.0 standardi, joka takaa sen, että laitteet toimivat yhteneväisellä tavalla riippumatta siitä kenen valmistama laite on. Suuret valmistajat, kuten Amazon, Apple, Google ja Samsung, ovat lukuisien pienempien valmistajien lisäksi lähteneet tukemaan tätä standardia. (Korhonen 2022.). Tähän mennessä ainakin 280 yritystä on liittynyt mukaan kehittämään yhdessä Matter 1.0 standardia (Matter Arrives Bringing A More Interoperable, Simple And Secure Internet Of Things to Life 2022.). Tällä hetkellä on jo saatavilla Matter 1.0 standardia tukevia älypistorasioita ja valaisimia, mutta tulevaisuudessa standardin kerrotaan laajenevan kattamaan myös monimutkaisempia älylaitteita, kuten esimerkiksi kodinkoneita (Korhonen 2022.).

### 2.3.2.1 *Http/https*

HTTP (Hypertext transfer protocol) on yleisin webissä käytettävistä protokollista. HTTP on protokolla, joka on perusta kaikelle tiedonvälittämiselle webissä. HTTP:n on alun perin kehittänyt Tim Berners-Lee tiimeineen vuosina 1989-1991. Vuosien varrella HTTP:tä on paranneltu lukuisia kertoja, jotta se on saatu tehokkaammaksi ja pystynyt vastaamaan nopeaan kehitykseen. (Evolution of HTTP.)

HTTP:n arkkitehtuurissa liikenne kulkee suoraan asiakkaalta (client) serverille. Asiakkaan päässä useimmiten web-selain lähettää pyyntöjä (engl. request) serverille, ja sivut ladataan selaimeen elementti kerrallaan GET-pyyntön avulla (An overview of HTTP.)



KUVIO 2. HTTP:n toiminta kun webselain lataa sivua. Mukaillen An overview of HTTP.

HTTPS (Hyper Text Transfer Protocol Secure) on HTTP:n salattu ja turvallisempi yhteystyyppi. HTTPS tekee yhteydestä turvallisemman luomalla lyhyen aikaa kestäviä istuntokohtaisia ja sertifioituja avaimia tiedonsiirtoihin käyttäjän ja serverin välillä. Liikenteen salauksen purkamiseen ja sen selkokielelle muuntamiseen tulee olla hallussa oikea istuntokohtainen avain (What is the difference between HTTP and HTTPS.)

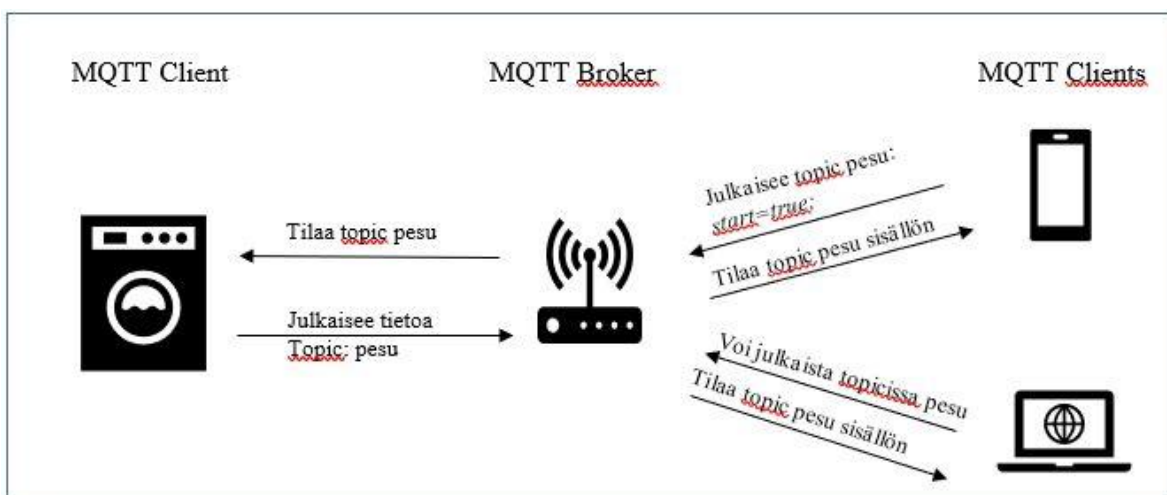
### 2.3.2.2 Mqtt

MQTT:n tapaa välittää verkkoliikennettä kutsutaan PubSub-arkkitehtuuriksi. PubSub-arkkitehtuurissa ei luoda pyyntöjä ja vastauksia asiakkaan ja serverin välillä kuten HTTP:ssä, vaan toiminta on tapahtumapohjaista ja perustuu aiheiden (topic) julkaisemiseen ja tilaamiseen (engl. publish and subscribe). PubSub-arkkitehtuurissa ei ole koskaan suoraa yhteyttä julkaisijan ja tilaajan välillä, joten viestintä on joustavaa niin tilan, ajan kuin synkronoinnin näkökulmasta. Laitteiden ei tarvitse tietää toistensa ip-osoitteita ja portteja tai edes olla käynnissä samaan aikaan. (HiveMQ 2020, 7; Manandhar 2017, 7.).

MQTT-yhteys muodostetaan ainoastaan clientin (tilaajan) ja brokerin (julkaisijan) välillä, mikä mahdollistaa uusien laitteiden liittämisen verkkoon helposti. Näin yhteyttä ei tarvitse luoda jokaisen tilaavan ja julkaisevan laitteen välille erikseen, vaan yhdellä katkaisijalla voidaan helposti ohjata niin monta laitetta kuin tarvitsee. Pub-Sub-arkkitehtuurin selkeä etu on myös erityyppisten laitteiden yhteensopivuus ja järjestelmän skaalautuvuus. MQTT-verkko kykenee skaalautumaan jopa miljooniin yhdistettyihin laitteisiin. (The Messaging and Data Exchange Protocol of the IoT.)

MQTT-protokollassa sekä sisältöä tuottavaa tahoja että sen tilaajaa kutsutaan samalla nimellä *client*. Clientin toiminta voi rajoittua vain aiheiden tilaamiseen tai julkaisemiseen, mutta on tärkeää huomata, että client voi myös tehdä molempia. Palvelua tai ohjelmistoa, joka välittää sisältöä clientien välillä kutsutaan nimellä *broker*. On olemassa broker-ohjelmistoja, jotka ladataan ja asennetaan tietokoneeseen ja brokereita jotka toimivat pilvipalveluna. Laajojen kokonaisuuksien luomiseen brokereita voidaan sillata, jolloin ne toimivat yhdessä. MQTT:ssä broker on serverin kaltainen, mutta toisin kuin servereissä, brokerit pystyvät lähettämään clienteleille tietoa koska tahansa. Brokerit tavallisesti välittävät tiedon mutta eivät tallenna sitä (Cope 2021.)

Broker-ohjelmistot voidaan jakaa kaupallisiin ja avoimen lähdekoodin ohjelmistoihin. Tunnetuimmat kaupalliset ohjelmistot ovat HiveMQ ja CloudMQTT. (Siltanen, 2019.) HiveMQ:sta on olemassa myös ilmainen kevennetty versio. Avoimen lähdekoodin ohjelmistoista yleisin on Mosquitto, joka on Eclipse-projektin kehittämä. Mosquittoilla on verkossa oma broker, jolla voidaan helposti testata omia IoT-järjestelmiä ja harjoitella MQTT:n käyttöä. Tämän opinnäytetyön käytännön osuudessa käytän edellä mainittua *test.mosquitto.org*-brokeria.



KUVIO 3. Tiedon välittäminen MQTT:ssä. Mukailten HiveMQ 2020.



MQTT:ssä sisältö luokitellaan käsitteellä *topic* (*aihe*). Topic on vähän samankaltainen asia kuin hashtagit sosiaalisessa mediassa. Näillä topic-tunnisteilla julkaisija kertoo sisältönsä liittyvän tiettyyn aiheeseen. Vastaavasti clientit voivat löytää ja tilata sisällön, joka on merkitty tiettyyn aiheeseen kuuluvaksi. On määritelty, että aiheiden tulee olla UTF-8-tyyppisiä merkkijonoja, niissä saa olla välilyöntejä ja isot ja pienet kirjaimet huomioidaan erikseen. On kuitenkin sellainen poikkeus, että \$SYS-alkuisia aiheita on mahdollista vain tilata mutta niihin ei voi julkaista. (Siltanen 2019, 5; Cope 2018.).

```

/
/koti
koti/olohuone/televisio
koti/keittio/valaistus

```

KUVIO 4. Esimerkki topic-tunnisteiden käytöstä. Mukailten Cope 2018.

MQTT:ssä on kaksi erityistä yhteyksiin liittyvää ominaisuutta. Ne ovat viimeinen tahto (Last will) ja QoS (Quality of Service, palvelun laatutaso). Viimeinen tahto määrittää sen, millaisia aiheita ja arvoja välittäjän halutaan julkaisevan, jos yhteys katkeaa äkillisesti. Esimerkiksi jos kyse on uunin etäohjauksesta, niin yhteyden katketessa last will voisi olla lämmitystehon arvojen laskeminen minimiin, jotta vältetään tulipaloriski. QoS:n avulla voidaan erotella viestit niiden tärkeyden mukaan kolmeen eri tasoon ja näin saadaan tärkeät tiedot varmasti perille. (Siltanen 2019.)

TAULUKKO 2. QoS-luokitus ja sen vaikutus viestintään

QoS-luokitus	Toiminta
1	Lähetetään enintään kerran
2	Lähetetään vähintään kerran
3	Lähetetään täsmälleen kerran

MQTT-yhteyteen liittyviä peruskomentotyyppisiä ovat, CONNECT, CONNACK, DISCONNECT, PINGREQ ja PINGRESP. Jokaiselle paketille on oma tietokehys. (Siltanen 2019.)

Yhteys muodostetaan siten, että client lähettää brokerille connect-pyyntön ja broker vastaa tähän pyyntöön omalla connack-pyyntöllään.

Connect-paketissa (TAULUKKO 3) on oltava vähintään clientID ja cleanSession-määritys. CleanSession määrittää sen, ladataanko clientin ja välittäjän aiempi viestihistoria, kun yhteys muodostetaan. Käyttäjänimeä tai salasanaa ei tarvitse antaa, mutta niillä voidaan parantaa turvallisuutta. KeepAlive-tieto liittyy yhteyden laatuun. Se on määritys, kuinka monta sekuntia välittäjä ja asiakas saavat olla erossa toisistaan ennen kuin yhteys tulkitaan katkenneeksi. (HiveMQ 2020, 13; Siltanen 2019, 7)

TAULUKKO 3. Connect-datakehys

CONNECT	
<u>clientID</u>	""client1""
<u>cleanSession</u>	TRUE
<u>username</u>	""user""
<u>password</u>	""secretpass123""
<u>keepAlive</u>	60
<u>lastWilltopic</u>	""/user/lastwill""
<u>lastWillQos</u>	2
<u>lastWillMessage</u>	""yhteys katkennut ""
<u>lastWillRetain</u>	FALSE

Connack-pyyntö (TAULUKKO 4) sisältää palautuskoodin (returnCode) arvona tiedon siitä, onnistuiko yhteyden muodostaminen vai ei. Yhteys on hyväksytty, jos palautuskoodin arvo on 0 (HiveMQ 2020, 14; Siltanen 2019, 7.) Kaikki connack-paketin palautuskoodin arvot löytyvät taulukosta 5.

TAULUKKO 4. Connack-datakehysten sisältö

CONNACK	
<u>sessionPresent</u>	TRUE
<u>returnCode</u>	0

## TAULUKKO 5. Connack-paketin palautuskoodin arvot

Connack-paketin palautuskoodin arvot

0	Yhteys hyväksytty
1	Yhteys hylätty, väärä protokollaversio
2	Yhteys hylätty, väärä tunniste
3	Yhteys hylätty, palvelin ei käytettävissä
4	Yhteys hylätty, käyttäjätunnus tai salasana ei kelpaa
5	Yhteys hylätty, ei valtuutusta

Onnistuneen yhteyden muodostamisen jälkeen brokerin tehtävä on ylläpitää yhteyttä, kunnes client pyytää yhteyden katkaisemista tai yhteys katkeaa muutoin.

Julkaisuun ja tilaukseen liittyviä komentoja ovat SUBSCRIBE, SUBACK, PUBLISH ja PUBACK. Alla olevissa taulukoissa 6 ja 7 esitetään näiden pakettien sisältö.

## TAULUKKO 6. Subscribe ja suback-pakettien sisältö

SUBSCRIBE		SUBACK	
<u>packetID</u>	2763	<u>packetID</u>	2763
<u>QoS1</u>	0	<u>returnCode1</u>	0
<u>topic1</u>	”autotalli/lampotila”	<u>returnCode2</u>	2
<u>QoS2</u>	1	....	....
<u>topic2</u>	”olohuone/lampotila”		
...	...		

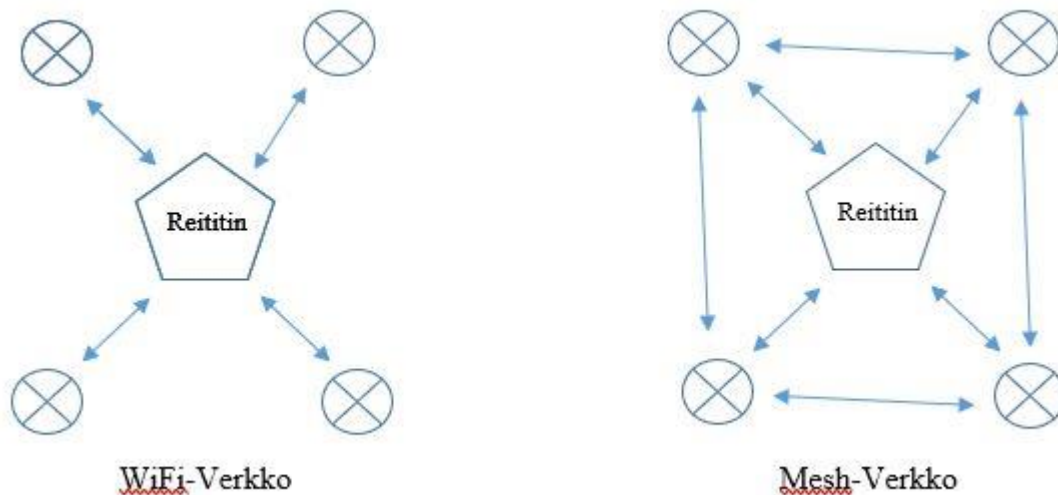
## TAULUKKO 7. Publish- ja puback-pakettien sisältö

PUBLISH		PUBACK	
<u>packetID</u>	2763	<u>packetID</u>	2763
<u>topicName</u>	”autotalli/lampotila”		
<u>payload</u>	”lampotila: 18.1”		
<u>QoS</u>	1		
<u>retainFlag</u>	FALSE		
<u>dupFlag</u>	FALSE		

### 2.3.2.3 Zigbee 3.0

Zigbee 3.0 on langaton protokolla, jota käytetään hyvin yleisesti varsinkin kodin IoT-laitteissa. Zigbeeen etu on alhainen virrankulutus ja edulliset valmistuskustannukset. Koska virrankulutus halutaan Zigbee-laitteissa pitää alhaisena, on lähetysetäisyys noin 10 - 100 metriä riippuen esteistä ja ympäristöstä. Zigbee toimii 2.4 GHz:n, 900 MHz:n ja 868 MHz:n taajuuksilla. Ainakin Ikeassa ja Lidlissä myytävissä Smart Home-aloituspakkauksissa ja laitteissa käytetään Zigbee 3.0-radiotekniikkaa.

Zigbee-verkon liikenne on suojattu symmetrisellä 128-bittisellä salausavaimella. Joillakin Zigbee-laitteista on kyky luoda mesh-verkkoja. Toisin kuin normaalissa verkossa, jossa laitteet on yksitellen kytketty reitittimeen, mesh-verkossa laitteet voivat kopioida tietoja ja välittää ne toiselle laitteelle. Mesh-verkon etu on sen suurempi kyky sietää yhteyshäiriöitä ja pidempiä etäisyyksiä.



KUVIO 5. WiFi-verkon ja mesh-verkon eroavaisuudet. Mukaillen Mikä on Zigbee-tekniikka, miksi se on tärkeä älykotille ja mitkä Xiaomi-tuotteet sitä tukevat 2022.

### 2.3.3 Tietoturva ja riskit

Yhä useammat laitteet niin kotona, laitoksissa ja yrityksissä on kytketty verkkoon, ja tämä tuo uudenlaisia riskejä, jotka on osattava ottaa huomioon. IoT-järjestelmiä suunniteltaessa tietoturva on yksi tärkeimmistä lähtökohdista.

Monet älylaitteet helpottavat elämäämme ja viihdyttävät meitä. Sitä ei tule ensimmäisenä ajatelleeksi, että ainakin osalla laitteiden valmistajista voi olla epärehellisiä tarkoituksia. On aina mahdollista, että esimerkiksi Kiinassa tuotettujen edullisten laitteiden mukana tulee mahdollisuus vakoilla ihmisten koteja ja yksityiselämää. Tuotteisiin voidaan jo valmistusvaiheessa jättää aukkoja, joilla valmistaja saa haltuunsa tietoja. Laitteisiin voidaan myös tarkoituksella jättää haavoittuvuuksia, joita jokin muu taho voi hyödyntää omiin tarkoituksiinsa. Koska tekijät ja suunnittelijat ovat ihmisiä, on myös olemassa se mahdollisuus, että tuotteisiin jää haavoittuvuuksia ja vikoja myös vahingossa. Ihmisen toiminnasta voi aiheutua tietoturvariskejä myös käyttäjän puolella ja yksi iso riski on, ettei laitteiden oletussalasanoja osata tai välitetä vaihtaa tietoturvallisempiin. (Cimpanu, 2019.)

Riippuu laitteessa olevista antureista, kuinka paljon laite voi aiheuttaa harmia, mutta teoriassa kaikkia laitteen antureita ja keräämiä tietoja voidaan käyttää hyväksi. On ollut jo pitkään puhetta siitä, että erilaiset palvelut (muun muassa Facebook) myyvät käyttäjästä kerättyjä tietoja mainostajille ja kolmansille osapuolille. Nämä tiedot ovat suurien summien arvoista kauppatavaraa. Kosola toteaa YLE:n verkkosivuilla julkaistussa artikkelissaan, että käyttäjistä muodostetaan profiileja tietojen perusteella, ja jopa useammalla laitteella kerätyt tiedot on mahdollista yhdistää samaan käyttäjään (Kosola 2016.) Palvelu kerää oman palvelunsa sisällä tehdyn toiminnan lisäksi käytännössä myös kaiken muun tiedon, johon se käsiksi pääsee: vierailut muut verkkosivut, hakukonehaut, puhelu- ja tekstiviestitietoja, yhteystiedot ja jopa sijaintihistorian. Miksipä myös Internet of Things-laitteiden valmistajat eivät keräisi ja myisi eteenpäin käyttäjistään tietoa samalla tavoin?

Vaikka valmistaja ei keräisi tietoja, on olemassa myös muita riskejä. IoT-laitteet ovat usein edullisia, joten niiden tietoturva ei ole välttämättä parasta mahdollista. Kun nämä laitteet kuitenkin usein kytetään kotiverkkoon, niistä voi tulla heikko lenkki, jonka kautta vihamielinen taho voi tunkeutua myös muihin kotiverkkoon liitettyihin laitteisiin. Koska samaan verkkoon kytketyt laitteet pystyvät havaitsemaan toisensa, suositellaan käyttämään erillistä verkkoa, johon IoT-laitteet kytetään (Cervantes, 2022.) Kun IoT-laitteet ovat erillisissä verkoissa, haavoittuvuus IoT-laitteessa ei anna tilaisuutta päästä käsiksi esimerkiksi puhelimiin, tabletteihin tai tietokoneisiin. Yleisimpiä tapoja eristää IoT-laitteet omaan verkkoon on käyttää kahta eri reititintä tai jakaa laitteet virtuaalisiin verkkoihin käyttäen reitittimen VLAN-väylää. (Cimpanu, 2019.) Sisäverkko voidaan suojata myös kolmannen osapuolen laitteen ja tietoturvaohjelmiston avulla. Tällainen on esimerkiksi F-Secure Sense reititin ja sovellus, jotka on suunniteltu erityisesti älykkään kodin laitteita varten ja tietoturvaa ajatellen.

### Internet of Thingsin (IoT) haasteita

- Mitä asioita laitteiden tai henkilöiden tarvitsee tietää?
- Todistaminen ja näyttö on vaikeaa (Milloin laite siirtyi minun omistukseeni?)
- Milloin yhteydet muutetaan? (Mitä jos myyn auton tai vaihdan asuntoa?)

KUVA 2. Internet of Thingsin haasteita. Mukailen Smith 2017.

Yksi riski IoT-laitteissa on tilanne, jossa laite tai kiinteistö, joka sisältää laitteet, myydään eteenpäin. (Smith 2017, 110.) Yhteyksien tiedot voivat jäädä edellisen omistajan laitteiden muistiin ja mahdollistaa väärinkäytön. Esimerkiksi Netflix-tunnuksiin voi olla liitettynä maksukortin tiedot, ja älytelevision uusi omistaja saa tietoonsa mitä edellinen on katsonut, voi käyttää palvelua ilmaiseksi ja saattaa pystyä tekemään ostoksia sovelluksessa.

## 3 MQTT-VIESTIN LÄHETYS JA VASTAANOTTO ARDUINOLLA

### 3.1 Käytännön työn tavoite

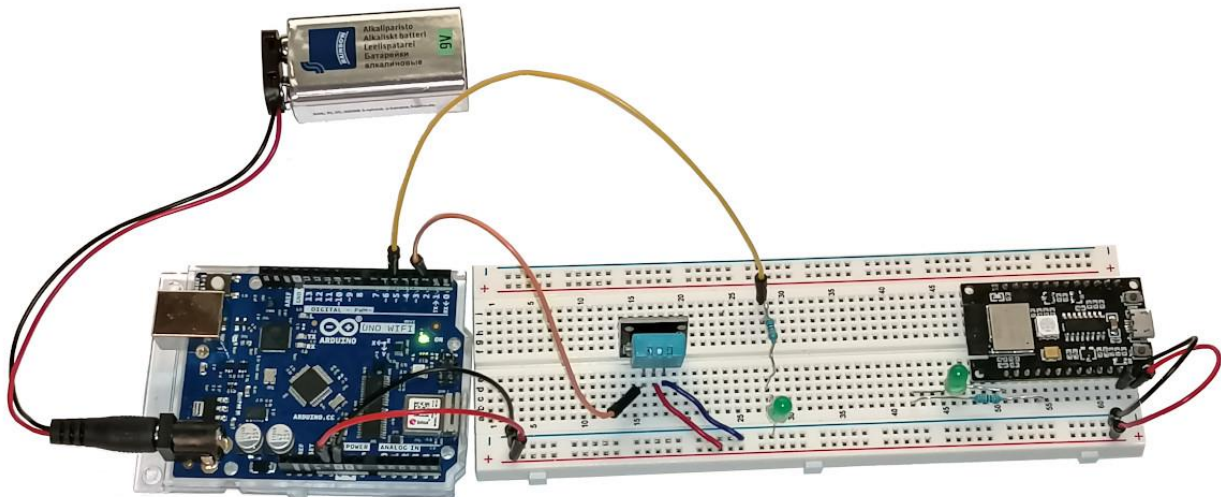
Tavoitteena tässä käytännön osuudessa on rakentaa ja valmistaa laite, joka mittaa ympäristönsä lämpötilan ja välittää tiedon MQTT-protokollaa hyödyntäen toiselle laitteelle, joka kytkee lämmittimen päälle, mikäli annettu lämpötila on alle määritetyn lukeman.

Laitteisto muodostaa yhteyden internetiin Robonet-nimisen verkon kautta. Sen jälkeen julkaiseva kehitysalusta julkaisee tiedon internetissä sijaitsevalle broker-serverille, josta tilaava kehitysalusta saa sen. Tilaava kehitysalusta asettaa lämmittimen päälle kun annettu lämpötila menee määriteltyä astelukua alemmas. Laitteiston toimivuus tarkistetaan lopuksi MQTT Explorer ohjelmalla, joka näyttää reaaliajassa aiheisiin julkaistut tiedot. Ohjelman avulla lämpötilaa ja lämmittimen tilaa voidaan myös seurata vaikkapa mobiililaitteilla ja tarvittaessa ohjata laitteistoa manuaalisesti.

Tämäntyyppistä kytkentää voidaan hyödyntää esimerkiksi silloin kun halutaan pitää talosta tietty osa tai huone viileänä ilman että lämpötila pääsee kuitenkaan laskemaan niin alas että aiheutuu rakenteellisia vaurioita talolle. Koska työssä keskeistä on MQTT-protokollan käyttäminen ja viestintä, kytkentä pidetään yksinkertaisena ja lämmitintä kuvataan LED-diodilla. Ledin syttyminen kuvastaa sitä, että lämmitin on toiminnassa. Käytännössä lämmitintä ohjattaisiin releen avulla, joten LED:n saamaa virtaa voidaan käsitellä releen saamana ohjausvirtana. Releellä voidaan pienemmän ohjausvirran avulla ohjata suurempia jännitteitä ja sähkövirtoja, joita esimerkiksi lämmitin yleensä tarvitsee toimiakseen.

### 3.2 Komponentit ja kytkennät

Käytännön osuutta varten käytettiin Arduino Uno Wifi Rev. 2 kehitysalustaa ja vastaanottavana laitteena käytettiin ESP-C3 kehitysalustaa (KUVA 3).



KUVA 3. Käytännön osan laitteiden kytkennät. Vasemmalla Arduino Uno WiFi ja koekytkentälevyn oikeassa reunassa ESP32-C3.

Sekä Arduino Uno WiFi että ESP32-C3 saavat virtansa 9 V paristosta. Arduino Unon liitännöistä olen tuonut 5 V jännitteen ja maadoituksen koekytkentäalustan virtakiskoon. Siitä saavat käyttöjännitteensä ja maadoituksensa DHT11 kosteus/lämpöanturi ja ESP32-C3.

DHT11-anturin signaali (keskimmäinen pinni) on kytketty Arduino Uno WiFi:n digitaaliseen pinniin nro 2. Digitaaliseen pinniin nro 5 on kytketty vihreä LED-diode, jonka syttyminen ilmaisee lämpötilan olevan alle määritetyn 23 celsiusastetta. Virtalähteen ja LED:in väliin tarvitaan aina vastus rajoittamaan LED:in saamaa virtaa. Tässä työssä kummankin LED:in edessä on metallikalvotyypinen vastus, jonka resistanssi on 330 Ohm.

ESP-kehitysalustan ulostulopinniin GPIO2 olen kytkenyt 330 Ohm metallikalvovastuksen ja vihreän LED:in kuvaamaan lämmittimen toimintaa. Vihreä LED on maadoitettu ESP32:n maadoituspinniin.



### 3.3 Ohjelmointi ja laitteiden konfigurointi

Kehitysalustojen ohjelmointiin käytettiin USB-kaapelia ja Arduino IDE-ohjelmointiympäristöä, johon ladattiin Arduino Uno WiFi Rev.2:ta varten lisäkirjastot *Arduino megaAVR Boards*, *ArduinoMqtt-Client* ja *WiFiNINA*. Vastaanottavaa ESP32-laitetta varten ohjelmointiympäristöön ladattiin lisäkirjastot *PubSubClient* ja *WiFi*.

Ohjelmakoodin lataaminen Arduinon kehitysalustalle tapahtuu Windowsissa niin että tarkistetaan laitehallinnasta COM-sarjaportti, johon Arduino on kytketty ja valitaan vastaava portti Arduino IDE-sovelluksesta. Tämän jälkeen painetaan ikkunan vasemmassa ylänurkassa sijaitsevaa oikealle päin osoittavaa nuolipainiketta, jolloin ohjelmakoodi käännetään konekieliseen muotoon ja ladataan kehitysalustalle.

Kummallekin kehitysalustalle on oma ohjelmakoodinsa. Ohjelmakoodien määrytykset yhteyksiä varten ovat luonnollisesti samat, mutta ne eroavat toisistaan roolien mukaan. Arduino Uno WiFi Rev. 2 toimii julkaisevana laitteena ja välittää brokerille lämpötila-arvon ja lämmittimen tilan. ESP32-C3 vastaanottaa nämä tiedot brokerilta ja käynnistää lämmittimen jos lämmittimen tila on määritelty lähetävässä laitteessa asentoon *on*. Koska julkaisevassa ja vastaanottavassa kehitysalustassa käytetään eri lisäkirjastoja, niiden tapa määrittää yhteyden parametrit on hieman erilainen. Julkaisevassa laitteessa brokerin osoite on kirjoitettu muotoon `test.mosquitto.org`. Tilaavassa laitteessa annetaan vastaavaan kohtaan parametriksi MQTT-brokerin IP-osoite. Koska Mosquitton testibrokerin IP-osoite ei ollut tiedossa, selvitettiin se `iplocation.net`-verkkosivuston avulla ja kirjoitettiin saatu osoite vastaanottavan laitteen yhteyden määrytyksiin (KUVA 4).

You've entered a domain name. We've found an IP address from the domain name you've entered. Your translated IP address is **91.121.93.94**

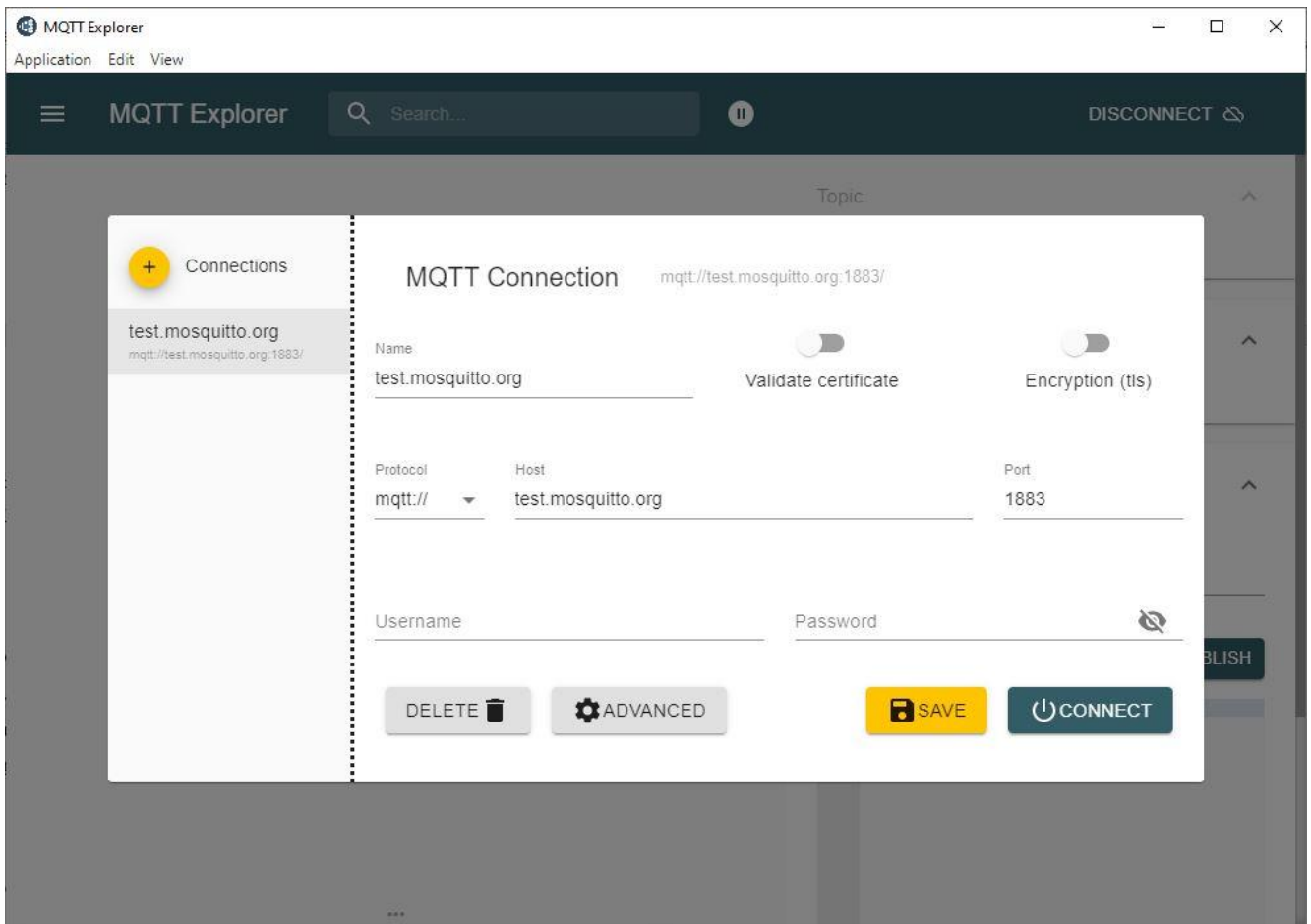
### Geolocation data from IP2Location (Product: DB6, 2023-1-1)

 <b>DOMAIN NAME:</b> test.mosquitto.org	 <b>ISP:</b> OVH SAS
 <b>COUNTRY:</b> France 	 <b>ORGANIZATION:</b> Not available
 <b>REGION:</b> Hauts-de-France	 <b>LATITUDE:</b> 50.6942
 <b>CITY:</b> Roubaix	 <b>LONGITUDE:</b> 3.1746

KUVA 4. Brokerin IP-osoitteen selvittäminen.

Ohjelmakoodin pohjana käytettiin Arduino Docsin sivulta Sending Data over MQTT-sivulta löytyvää esimerkkikoodia, jota muokattiin tähän työhön sopivaksi. Koodissa on määritelty lämmittimen tilan muuttuvan asennosta off asentoon on, kun lämpötila on alle 23 celsiusastetta. Lämpötilan noustessa jälleen 23 astetta suuremmaksi lämmittimen tila muuttuu vastaavasti asentoon off. Lämpötilan raja-arvoksi valitsin 23 astetta koska testiympäristön lämpötila on noin 22 astetta. Näin on helppoa tarvittaessa nostaa lämpötilaa esimerkiksi puhaltamalla lämmintä ilmaa anturiin.

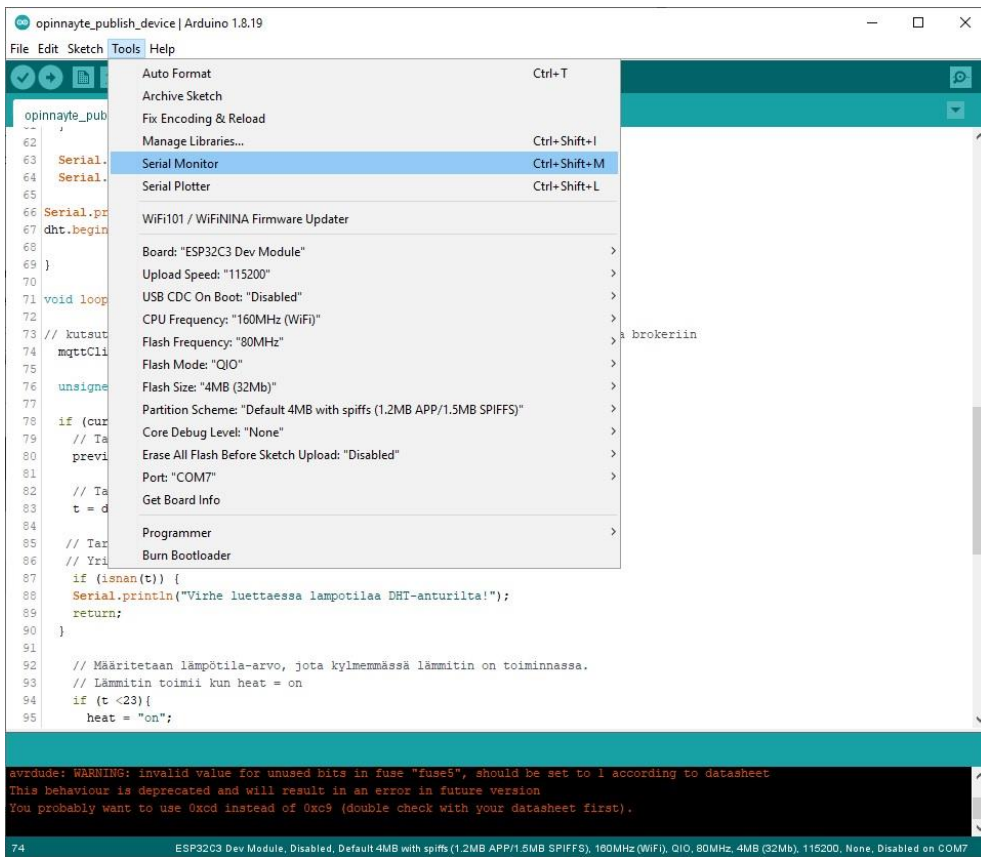
Laitteet konfiguroitiin niin, että ne ottavat Robonet-nimisen Wifi-verkon kautta yhteyden Ranskassa sijaitsevalle brokerille test.mosquitto.org ja liikenne tapahtuu portin 1883 kautta. Portti 1883 on MQTT:ssä oletusportti salaamattomalle liikenteelle. Tietokoneelle ladattiin myös MQTT Explorer-ohjelma, jolla voi julkaista ja katsoa testibrokerin tapahtumia (KUVA 4).



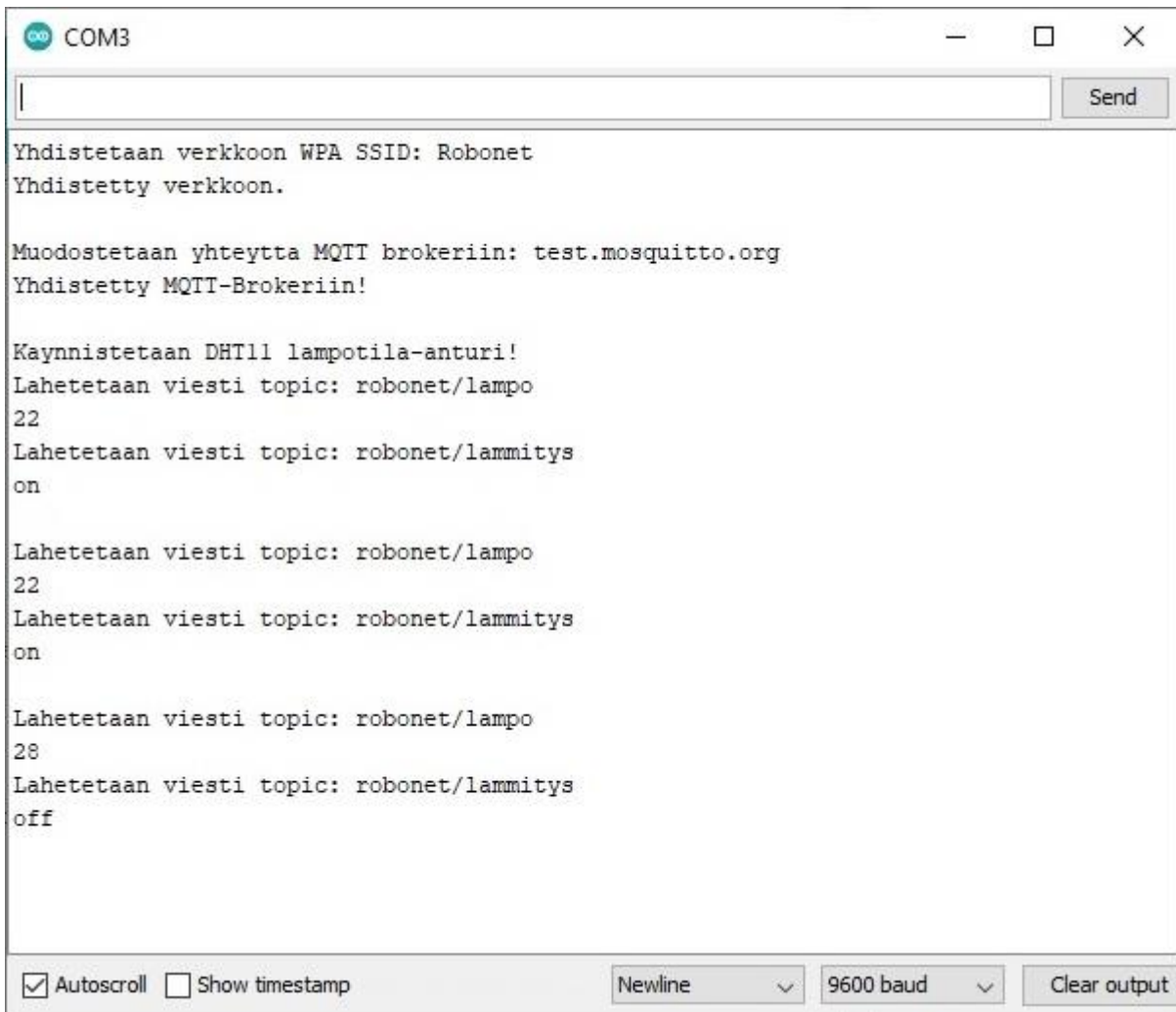
KUVA 5. MQTT Explorerin käyttöliittymä ja sen asetukset.

### 3.4 Viestin lähetys ja vastaanotto

Kun ohjelmakoodi (Liite 1) on käännetty konekielelle ja ladattu Arduino Uno WiFi Rev 2:n muistiin, ohjelmaa suorittaminen alkaa automaattisesti. Tämän jälkeen avataan Arduino-ohjelmointiympäristön tools-valikosta Serial Monitor (KUVA 5). Jotta Serial Monitor toimii oikein ja alkaa näyttää tietoja sarjaporttiliikenteestä, on oltava valittuna oikea COM-sarjaportti ja nopeus. Tällä kertaa lähettävän laitteen sarjaporttina on COM3 ja valittuna on nopeus 9600 baud.



KUVA 6. Arduino IDE-ohjelma. Serial Monitor löytyy Tools-valikon alta.



```
COM3
| Send
Yhdistetaan verkkoon WPA SSID: Robonet
Yhdistetty verkkoon.

Muodostetaan yhteyttä MQTT brokeriin: test.mosquitto.org
Yhdistetty MQTT-Brokeriin!

Kaynnistetaan DHT11 lämpötila-anturi!
Lahetetaan viesti topic: robonet/lampo
22
Lahetetaan viesti topic: robonet/lammitys
on

Lahetetaan viesti topic: robonet/lampo
22
Lahetetaan viesti topic: robonet/lammitys
on

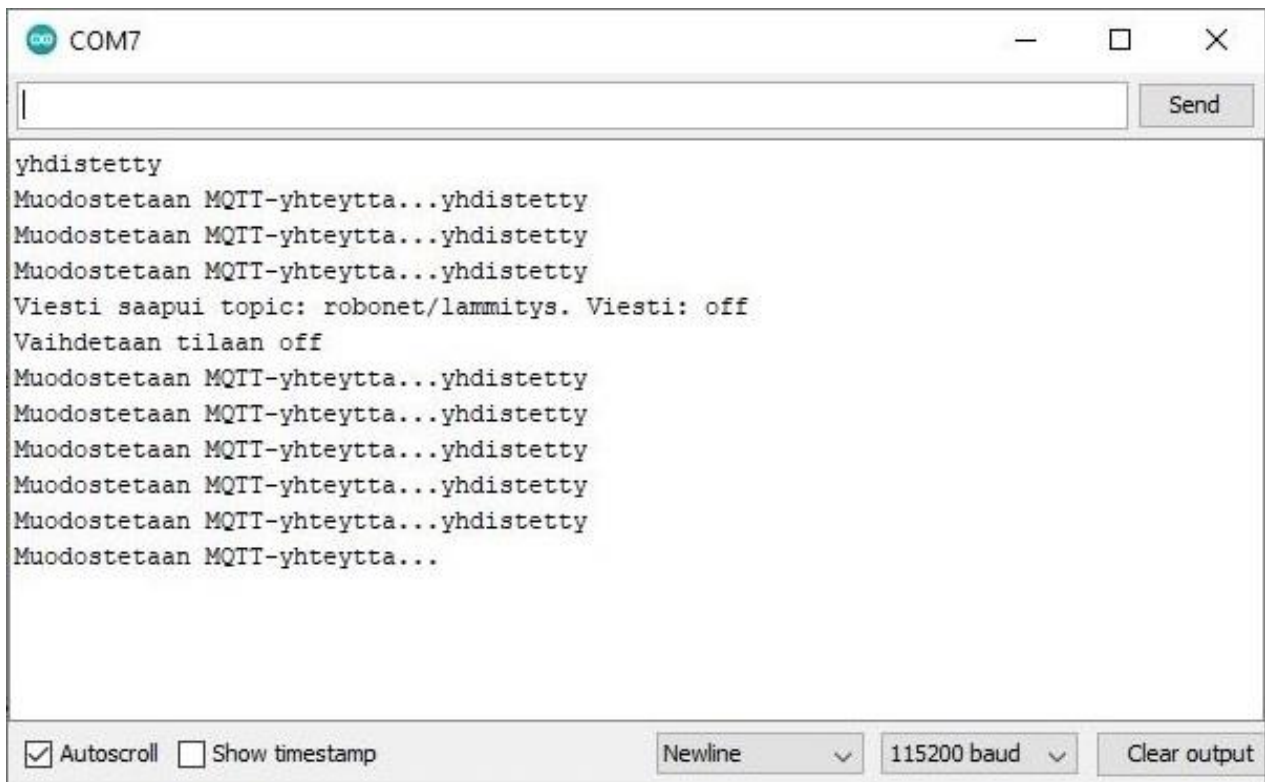
Lahetetaan viesti topic: robonet/lampo
28
Lahetetaan viesti topic: robonet/lammitys
off

 Autoscroll  Show timestamp
Newline 9600 baud Clear output
```

KUVA 7. Sarjaporttimonitori ja lähettävän laitteen liikennettä.

Serial Monitorin klikkaaminen avaa uuden ikkunan, johon alkaa tulla tietoa (KUVA 6). Ensimmäiseksi yhdistetään WiFi-verkkoon ja seuraavaksi MQTT-brokeriin. Sitten lämpötila-anturi alkaa julkaista tietoja 8 sekunnin välein aiheisiin robonet/lampo ja robonet/lammitys. Lämpötila alussa on 22 astetta ja lämmitys on päällä. Puhalsin lämmintä ilmaa lämpötilasensoriin, jolloin lämpötila nousi 28 asteeseen ja lämmitys otettiin pois päältä, koska lämpötila ylitti määritellyn 23 asteen rajan.

Seuraavaksi tarkastellaan MQTT-liikennettä vastaanottavan ESP32-C3-laitteen puolelta. Vastaanottavan laitteen ohjelmakoodi on liitteessä 2.

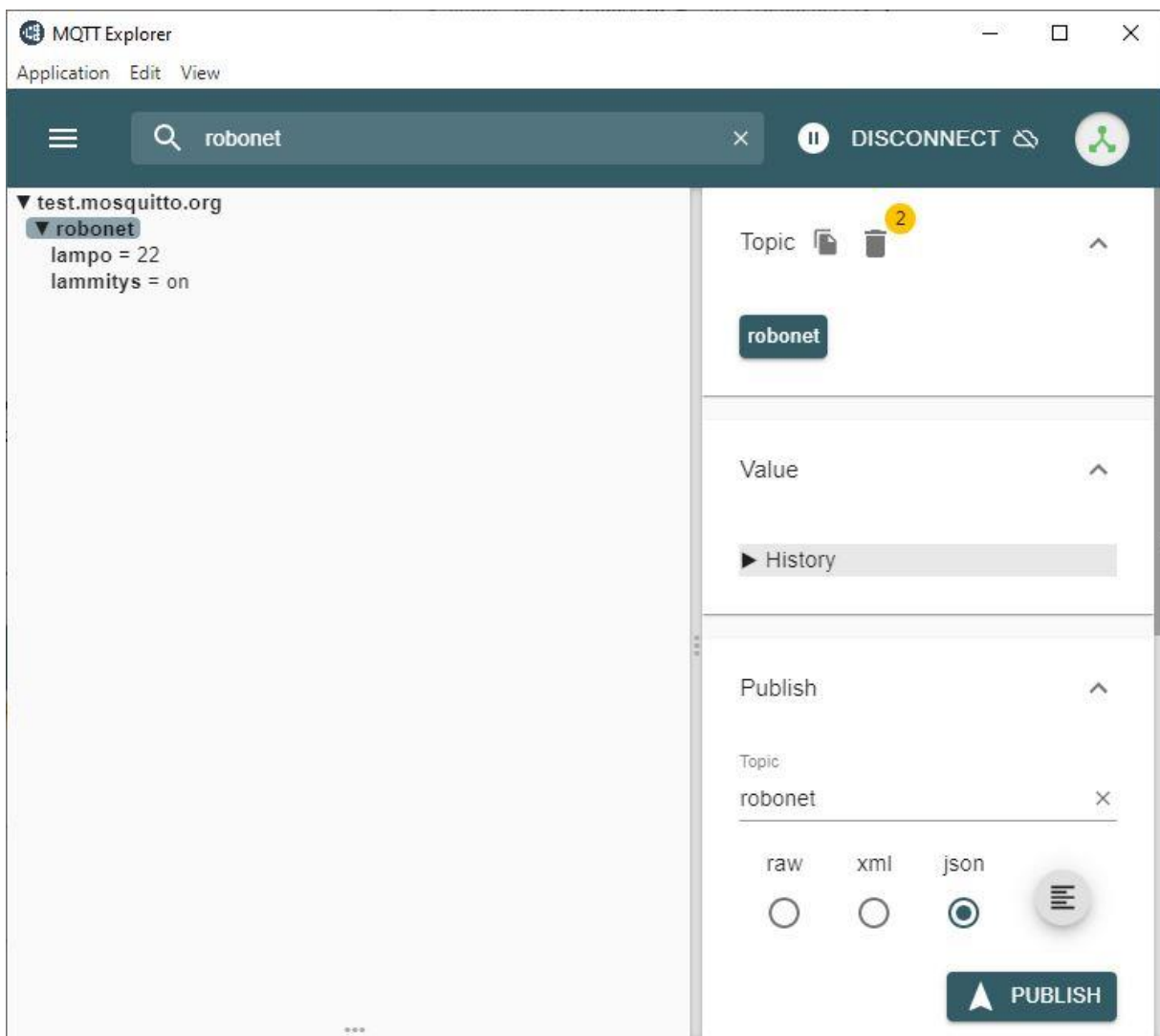


KUVA 8. Sarjaporttimonitorin liikennettä tilaavan ESP32-laitteen puolelta.

Vastaanottavan laitteen sarjaportti on COM7 ja nopeus 115200 baud. Vastaanottava laite muodostaa ensin yhteyden samaan WiFi-verkkoon ja MQTT-brokeriin kuin lähettävä laite. Sen jälkeen laite tilaa aiheen robonet/lammitys. Kun lähettävä laite muuttaa aiheen tilaa asentojen on ja off välillä, vastaanottava laite saa siitä tiedon ja kytkee lämmittimen vastaavasti päälle tai pois päältä (KUVA 7).

## 4 TULOKSET

Laitteiden välinen tiedonsiirto toimii hienosti. Kumpikin laite sai muodostettua yhteyden Robonet-verkkoon, jonka kautta laitteet muodostivat yhteyden Mosquitton testibrokeriin. MQTT Explorerin avulla huomattiin, että julkaiseva laite julkaisee sujuvasti ja säännöllisesti tietoa määrättyihin aiheisiin ja lämpötilasensorin saamat arvot vaikuttavat siihen mikä arvo lämmittimelle annetaan (KUVA 8).



KUVA 9. Lähetetyt tiedot Mosquitton testibrokerilla.

Lämmittimen tilaa ilmaisevat ledit syttyvät hieman peräjälkeen, mutta vastaanottavan laitteen ledi palaa joitakin sekunteja lähettävän laitteen ledin sammumisen jälkeen. Laitteen toimivuutta tämä pieni viive ei kuitenkaan merkittävästi muuta.



## 5 POHDINTA

Sähköä säästäville ja etäyhteydellä toimiville laitteille on yhä enemmän kysyntää. Sellaisenaan tai vähäisellä muokkaamisella kokonaisuutta voisi käyttää vaikkapa tilojen tai veden lämpimänä pitämiseen. Julkaisevan laitteen ohjelmakoodissa (Liite 1) on määritelty lämpötilan julkaisu tiheys 8 sekuntiin, joka on varmasti riittävä lämmitystarkoitukseen. Kuitenkin useimmissa käyttötarkoituksissa päivitysväliksi ja mittausten väliseksi ajaksi voisi määritellä pidemmän ajan, jolloin laitteesta saadaan energiatehokkaampi ilman että toiminnallisuus kärsii.

Sähkön tehokkaampaan säästämiseen kokonaisuutta voisi käyttää esimerkiksi siten, että tuo Nordpoolista pörssisähkön hintatiedot seuraavalle vuorokaudelle ja asettaa paljon sähköä kuluttavat laitteet toimimaan silloin kun sähkön hinta on edullisempaa. Laitteita voisi myös luokitella niiden virrankulutuksen mukaan ja määritellä vaikkapa vesivaraajalle tai lämmittimelle lämpötilarajan, jonka alle se ei saa pudota, vaikka pörssisähkön hinta olisikin korkea.

MQTT:ssä olisi järkevää käyttää salausta ja mikäli laitekokonaisuutta kehittäisi eteenpäin, voisi openssl-ohjelmistolla luoda avaimet ja sertifikaatit, joilla liikenne voidaan salata. Laitekokonaisuuteen olisi myös helppoa lisätä etäohjausmahdollisuus siten, että vastaanottavan laitteen lähdekoodiin lisätään mahdollisuus kytkeä lämmityslaite manuaaliseen hallintaan. Tämän voisi toteuttaa helposti asettamalla lämmittimen käynnistymisen ehtoihin uuden aiheen, jonka tilan voi esimerkiksi MQTT Explorerin tai vastaavan älypuhelinsovelluksen avulla vaihtaa.

Työtä tehdessä huomattiin, että Arduino-kehitysalustojen ja MQTT:n avulla voidaan helposti rakentaa itse laitteita erilaisiin käyttötarkoituksiin. Lisäkirjastoja voi tehdä itse ja niitä on saatavilla monipuolisesti, joten on myös helppoa sovittaa erilaisia laitteita toimimaan Arduino-kehitysalustan kanssa. Asetetut tavoitteet, jotka tälle työlle oli määritelty, saavutettiin.

## LÄHTEET

- An overview of HTTP. MDN web docs. Saatavilla: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>. Päivitetty 22.11.2022. Viitattu 8.11.2022.
- Arduino a. What is Arduino? Saatavilla: <http://www.arduino.cc/en/Guide/Introduction>. Viitattu: 8.11.2022.
- Arduino b. Components. Saatavilla: <http://playground.arduino.cc/Main/Components>. Viitattu: 8.11.2022.
- Arduino c. What are the Different Types of Arduino Boards. Elprocus. Saatavilla: <https://www.elprocus.com/different-types-of-arduino-boards/>. Viitattu 8.11.2022.
- Arduino d. Installing Libraries. Arduino Docs. Saatavilla: <https://docs.arduino.cc/software/ide-v1/tutorials/installing-libraries#what-are-libraries>. Viitattu 8.11.2022.
- Cervantes, R. 2022. Why Your Remote Team Should Put IoT Devices on a Separate Wi-Fi Network. Csolutionssit. Saatavilla: <https://csolutionssit.com/iot-devices-separate-wi-fi-network/>. Viitattu 12.11.2022.
- Cimpanu, C. 2019. FBI recommends that you keep your IoT devices on a separate network. Zdnet. Saatavilla: <https://www.zdnet.com/article/fbi-recommends-that-you-keep-your-iot-devices-on-a-separate-network/>. Viitattu 8.11.2022.
- Cope, S. 2018. Understanding MQTT Topics. Verkkosivu. Saatavilla: <http://www.steves-internet-guide.com/understanding-mqtt-topics/>. Päivitetty 11.2.2022. Viitattu 8.11.2022.
- Cope, S. 2021. How MQTT Works - Beginners Guide. Verkkosivu. Saatavilla: <http://www.steves-internet-guide.com/mqtt-works/>. Päivitetty 12.2.2021. Viitattu 8.11.2022.
- Doukas, C. 2012. Building Internet of Things with the Arduino: V1.1. North Charleston, S.C.: CreateSpace. Viitattu 8.11.2022.
- Eshghi, B 2023. A 2023 Guide to IoT Ecosystem: 5 Components & 3 Challenges. AI Multiple. Saatavilla: <https://research.aimultiple.com/iot-ecosystem/>. Päivitetty 11.1.2023. Viitattu 16.1.2023.
- Esineiden internetin hyödyt valmistavassa teollisuudessa. Empirica. Saatavilla: <https://www.empirica.fi/esineiden-internetin-hyodyt-teollisuudessa/>. Viitattu 16.1.2023.
- Evolution of HTTP. MDN Web Docs. Saatavilla: [https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics\\_of\\_HTTP/Evolution\\_of\\_HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP). Päivitetty 5.1.2023 Viitattu 8.12.2022.
- HiveMQ 2020. MQTT & MQTT 5 Essentials. A comprehensive overview of MQTT facts and features for beginners and experts alike. Pdf-dokumentti. Saatavilla: <https://www.hivemq.com/downloads/hivemq-ebook-mqtt-essentials.pdf>. Viitattu 8.11.2022.

- Isohanni, J. & Heikkilä, M. & Malkamo, V. & Himanka, M. 2022. IoT Next Phase. Allied ICT Finland. PDF-dokumentti. Saatavilla: [https://www.alliedict.fi/wp-content/uploads/2022/02/AIF\\_IoT\\_Next\\_Phase.pdf](https://www.alliedict.fi/wp-content/uploads/2022/02/AIF_IoT_Next_Phase.pdf). Viitattu 16.1.2023.
- Karppila, A. 2014. Arduino-pohjainen laite liikkeen ja lämpötilan monitorointiin. Helsinki: Haaga-Helia ammattikorkeakoulu. Opinnäytetyö. Saatavilla: <https://urn.fi/URN:NBN:fi:amk-2014111315627>. Viitattu 8.11.2022.
- Korhonen, S. 2022. Älykotilaitteiden yhteispeli paranee: Matter 1.0 julkaistiin. Mikrobitti. Julkaistu 5.10.2022. Saatavilla: <https://www.mikrobitti.fi/uutiset/alykotilaitteiden-yhteispeli-paranee-matter-10-julkaistiin/ccc509e8-5092-452d-bc9c-254a4fb58246>. Viitattu 8.11.2022.
- Kosola, L. 2016. Facebook säilöö jopa puhelutietoja – näin meistä rakennetaan profiileja mainostajille. Yle. Saatavilla: <https://yle.fi/aihe/artikkeli/2016/10/11/facebook-sailoo-jopa-puhelutietoja-nain-meista-rakennetaan-profiileja>. Viitattu 08.12.2022.
- Manandhar, S. 2017. MQTT based Communication in IoT. Tampere: Tampereen teknillinen yliopisto. Diplomityö. Saatavilla: <https://urn.fi/URN:NBN:fi:tty-201711222224>. Viitattu 2.12.2022.
- Matter Arrives Bringing A More Interoperable, Simple And Secure Internet Of Things to Life. Connectivity Standards Alliance. Julkaistu 4.10.2022 Saatavilla: <https://csa-iot.org/newsroom/matter-arrives/>. Viitattu 8.11.2022.
- Mikä on Zigbee-tekniikka, miksi se on tärkeä älykotille ja mitkä Xiaomi-tuotteet sitä tukevat? Xiaomi-planet. Saatavilla: <https://fi.xiaomiplanet.sk/zigbee-tekniikka/>. Julkaistu 22.11.2022. Viitattu 9.12.2022.
- Moilanen, P. 2022. Kansalaisen Kyberturvallisuus lv. 22-23. Jyväskylä: Jyväskylän yliopisto. Verkko-kurssi. Saatavilla: <https://onlinecourses.jyu.fi/course/view.php?id=33&section=2#tabs-tree-start/>. Viitattu 8.11.2022.
- P2P-järjestelmät. Tampere: Tampereen yliopisto. Verkkokurssi. Saatavilla: [https://plus.tuni.fi/comp.sec.100/fall-2021/m12\\_distributed\\_systems/p2p\\_gen/?hl=fi](https://plus.tuni.fi/comp.sec.100/fall-2021/m12_distributed_systems/p2p_gen/?hl=fi). Viitattu 12.12.2022.
- Posey, B. 2022. IoT Gateway. Techtarget. Saatavilla: <https://www.techtarget.com/iotagenda/definition/IoT-gateway>. Päivitetty maaliskuussa 2022. Viitattu 8.11.2022.
- Siltanen, J. 2019. MQTT-protokollan soveltuvuus IoT-järjestelmiin. Tampere: Tampereen yliopisto. Kandidaatintutkielma. Saatavilla: <https://urn.fi/URN:NBN:fi:tuni-201912166912>. Viitattu 8.11.2022.
- Smith, S. 2017. The Internet of Risky Things. Trusting the Devices That Surround Us. Sebastopol CA, USA: O' Reilly media Inc. Viitattu 16.12.2022.
- Statista 2021. Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030. Saatavilla: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>. Viitattu 16.1.2023.
- The Messaging and Data Exchange Protocol of the IoT. HiveMQ. Verkkosivu. Saatavilla: <https://www.hivemq.com/mqtt/mqtt-protocol/>. Viitattu 6.11.2022.

What is the difference between HTTP and HTTPS? Brightedge. Saatavilla: <https://www.brightedge.com/glossary/what-is-https>. Viitattu 8.12.2022.

Älykäs jääkaappi tunnistaa jo puheen. TM Rakennusmaailma. Lehtiartikkeli. Julkaistu 4.9.2018. Saatavilla: <https://rakennusmaailma.fi/alykas-jaakaappi-tunnistaa-jo-puheen/>. Viitattu 8.11.2022.

## Liite 1: Julkaisevan laitteen ohjelmakoodi

```
opinnayte_publish_device
1 #include "DHT.h"
2 #include <SPI.h>
3 #include <ArduinoMqttClient.h>
4 #include <WiFiNINA.h>
5
6 #define DHTPIN 2 // Digitaalinen pinni johon DHT-anturi on kytketty
7 #define DHTTYPE DHT11 // Anturin tyyppi on DHT 11
8 #define HEATERPIN 5 // Digitaalinen pinni johon lämmittimen toiminnan LED-ilmaisim on kytketty
9
10 DHT dht(DHTPIN, DHTTYPE);
11
12 // WiFi-verkon SSID (nimi)
13 char ssid[] = "Robonet";
14 // WiFi-verkon salasana, käytä tietoturvasalaisempia salasanoja kuin tämä!
15 char pass[] = "salainensanal23";
16
17 WiFiClient wifiClient;
18 MqttClient mqttClient(wifiClient);
19
20 //Brokerina käytetään serveriä test.mosquitto.org
21 //Määritetään portti ja topicien tiedot
22 const char broker[] = "test.mosquitto.org";
23 int port = 1883; // MQTT oletusportti on 1883
24 const char topic[] = "robonet/lampo";
25 const char topic2[] = "robonet/lammitys";
26
27 //Asetetaan päivitysväli ms (millisekunteina)
28 //Määritetään lämpötilan ja lämmittimen tilan päivitystiheydeksi 8 sekuntia.
29 const long interval = 8000;
30 unsigned long previousMillis = 0;
31
32 //Määritellään tarvittavat muuttujat
33 int count = 0;
34 int t;
35 String heat;
36
37 void setup() {
38
39 //Aloitetaan sarjaporttiyhteys
40 Serial.begin(9600);
41 while (!Serial) {
42 ; // Odotetaan sarjaportin käynnistymistä
43 }
44
45 // Yhdistetään WiFi-verkkoon:
46 Serial.print("Yhdistetään verkkoon WPA SSID: ");
47 Serial.println(ssid);
48 while (WiFi.begin(ssid, pass) != WL_CONNECTED) {
49 // Jos epäonnistui, yritetään uudestaan 5 sekunnin kuluttua
```

```
50 Serial.print(".");
51 delay(5000);
52 }
53
54 Serial.println("Yhdistetty verkkoon.");
55 Serial.println();
56 Serial.print("Muodostetaan yhteyttä MQTT brokeriin: ");
57 Serial.println(broker);
58
59 if (!mqttClient.connect(broker, port)) {
60     Serial.print("MQTT yhteysvirhe! Virhekoodi = ");
61     Serial.println(mqttClient.connectError());
62
63     while (1);
64 }
65
66 Serial.println("Yhdistetty MQTT-Brokeriin!");
67 Serial.println();
68
69 Serial.println("Käynnistetään DHT11 lämpötila-anturi!");
70 dht.begin(); //Yhteyden muodostamisen jälkeen käynnistä DHT-anturi.
71
72 }
73
74 void loop() {
75
76 // kutsutaan poll()-funktioita säännöllisesti jotta vältetään yhteyden katkeamista brokeriin
77 mqttClient.poll();
78
79 unsigned long currentMillis = millis();
80
81 if (currentMillis - previousMillis >= interval) {
82     // Tallenna viimeisin aika jolloin viesti lähetettiin
83     previousMillis = currentMillis;
84
85     // Tallenna DHT-anturin antama Celsius-lämpötila muuttujaan t
86     t = dht.readTemperature();
87
88     // Tarkista tapahtuiko lämpötilan lukemisessa virheitä
89     // Yritys uudestaan jos tuli virhe
90     if (isnan(t)) {
91         Serial.println("Virhe luettaessa lämpötilaa DHT-anturilta!");
92         return;
93     }
94
95     // Määritetään lämpötila-arvo, jota kylmemmässä lämmitin on toiminnassa.
96     // Lämmitin toimii kun heat = on
97     if (t < 23){
98         heat = "on";
```

```
99     }
100     else
101         heat = "off";
102
103     //Tulosta serial monitoriin lämpötila ja ilmoitus julkaisusta
104     Serial.print("Lahetetaan viesti topic: ");
105     Serial.println(topic);
106     Serial.println(t);
107     Serial.print(F(" C"));
108
109     //Tulosta serial monitoriin lämmityksen tilanne ja ilmoitus julkaisusta
110     Serial.print("Lahetetaan viesti topic: ");
111     Serial.println(topic2);
112     Serial.println(heat);
113
114     // Lähetä viesti, print-käskeyä voidaan käyttää viestin sisällön muokkaamiseen
115     mqttClient.beginMessage(topic);
116     mqttClient.print(t);
117     mqttClient.endMessage();
118
119     mqttClient.beginMessage(topic2);
120     mqttClient.print(heat);
121     mqttClient.endMessage();
122
123     Serial.println();
124 }
125
126 //Jos lämpötila on alle määritetyn, sytytetään merkiksi LED joka on kytketty pinniin 5.
127 //Jos lämpötila on määritetyn 23 astetta tai enemmän, sammutetaan LED.<
128 if (heat=="on") {
129     digitalWrite(5, 220);
130 }
131 else
132 digitalWrite(5, 0);
133
134 }
```

## Liite 2: Tilaavan laitteen ohjelmakoodi

```
opinnayte_subscribe_device
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 // WiFi-verkon nimi ja salasana
5 const char* ssid = "Robonet";
6 const char* password = "salainensanal23";
7
8 // MQTT-brokerin (test.mosquitto.org) IP-osoite tulee tähän
9 const char* mqtt_server = "91.121.93.94";
10
11 WiFiClient espClient;
12 PubSubClient client(espClient);
13 long lastMsg = 0;
14 char msg[50];
15 int value = 0;
16
17 // Määritetään lämmittimen ulostulopinni
18 const int lammPin = 2;
19
20
21 void setup() {
22   Serial.begin(115200);
23   // Oletusasetukset
24
25   setup_wifi();
26   client.setServer(mqtt_server, 1883); // MQTT-liikenteen portti on 1883
27   client.setCallback(callback);
28
29   //Määritetään lämmittimen pinni ulostuloksi
30   pinMode(lammPin, OUTPUT);
31 }
32
33 void setup_wifi() {
34   delay(10);
35   // Aloitetaan muodostamalla yhteys WiFi-verkkoon
36   Serial.println();
37   Serial.print("Yhdistetaan ");
38   Serial.println(ssid);
39
40   WiFi.begin(ssid, password);
41
42   while (WiFi.status() != WL_CONNECTED) {
43     delay(500);
44     Serial.print(".");
45   }
46
47   Serial.println("");
48   Serial.println("WiFi yhdistetty");
49   Serial.println("IP-osoite: ");
```



```
50 Serial.println(WiFi.localIP());
51 }
52
53 void callback(char* topic, byte* message, unsigned int length) {
54   Serial.print("Viesti saapui topic: ");
55   Serial.print(topic);
56   Serial.print(". Viesti: ");
57   String messageTemp;
58
59   for (int i = 0; i < length; i++) {
60     Serial.print((char)message[i]);
61     messageTemp += (char)message[i];
62   }
63   Serial.println();
64
65
66   // Kun topic robonet/lammitys saa uuden viestin
67   // Käynnistää lämmittimen kun viesti on "on" ja sammuttaa sen kun viesti on "off"
68   if (String(topic) == "robonet/lammitys") {
69     Serial.print("Vaihdetaan tilaan ");
70     if(messageTemp == "on"){
71       Serial.println("on");
72       digitalWrite(lammPin, HIGH);
73     }
74     else if(messageTemp == "off"){
75       Serial.println("off");
76       digitalWrite(lammPin, LOW);
77     }
78   }
79 }
80
81 void reconnect() {
82   // Silmukka yhdistämiseen asti
83   while (!client.connected()) {
84     Serial.print("Muodostetaan MQTT-yhteyttä...");
85     // Yhdistetään
86     if (client.connect("ESP8266Client")) {
87       Serial.println("yhdistetty");
88       // Tilataan topic
89       client.subscribe("robonet/lammitys");
90     } else {
91       Serial.print("virhe, rc=");
92       Serial.print(client.state());
93       Serial.println(" try again in 5 seconds");
94       // Odottaa 5 sekuntia ennen kuin yrittää uudelleen muodostaa yhteyttä
95       delay(5000);
96     }
97   }
98 }
```

```
99
100 void loop() {
101   if (!client.connected()) {
102     reconnect();
103   }
104   client.loop();
105
106   long now = millis();
107   if (now - lastMsg > 5000) {
108     lastMsg = now;
109
110   }
111 }
```

---