

Pasi Salonen

## **Automaatiojärjestelmien IoT-mahdollisuudet**

IoT-mahdollisuuksien selvitys automaatiojärjestelmien harjoittelualustoihin

Opinnäytetyö

Kevät 2018

SeAMK Tekniikka

Automaatiotekniikan tutkinto-ohjelma

**SeAMK** 

SEINÄJOEN AMMATTIKORKEAKOULU  
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

## Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Tutkinto-ohjelma: Automaatiotekniikka

Suuntautumisvaihtoehto: Koneautomaatio

Tekijä: Pasi Salonen

Työn nimi: Automaatiojärjestelmien IoT-mahdollisuudet

Ohjaaja: Seppo Stenberg

Vuosi: 2018

Sivumäärä:33

Liitteiden lukumäärä:2

---

Opinnäytetyön toimeksiantajana oli Tampereella toimiva automaatioharjoitusalueita opetuskäyttöön kehittävä ja markkinoiva yritys JJJ-Automaatio Oy. Työn tavoitteena oli suunnitella IoT-prototyypilaitteisto liitettäväksi yhteen yrityksen valmistamaan automaatiojärjestelmän harjoitusalueeseen.

Opinnäytetyössä tutustuttiin merkittävimpiin IoT-käsitteeseen mielletäviin laitteistoihin, ohjelmistoihin ja palveluihin. Työssä tehtiin prototyyppi, jonka avulla testattiin kokonaisuutta siitä, miten nämä kaikki edellä mainitut asiat yhdistettäisiin konkreettisesti. Prototyyppiä testattiin käytännössä, sekä annettiin selvitys kohdeyritykselle prototyypin liittämistä harjoitusalueeseen. Työ kokonaisuudessaan toimi selvityksenä kyseisestä asiasta.

Työn painopiste ei ollut ohjelmakoodin laadinnassa tai tietyn sensorin perusteellisessa tutkimisessa, vaan siinä, millä tavalla laitteistoilla ja sovelluksilla IoT-harjoituksia voitaisiin tehdä, että harjoitusalueella käyttävä henkilö hahmottaisi IoT-käsitteen kokonaisuudessaan. IoT on hyvin laaja käsite, joka voidaan määrittää monella eri tavalla ja siksi työhön on lisätty muutamia käytännön esimerkkejä selvittämään, missä yhteydessä IoT-käsitteestä puhutaan tässä opinnäytetyössä.

Avainsanat: IoT, pilvipalvelut, automaatio, kehitysalustat

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

## Thesis abstract

Faculty: School of Technology

Degree programme: Automation Engineering

Specialisation: Machine Automation

Author: Pasi Salonen

Title of thesis: IoT possibilities in Automation Systems

Supervisor: Seppo Stenberg

Year: 2018      Number of pages:33      Number of appendices:2

---

The assignment for the thesis was given by JJJ-Automaatio Oy in Tampere. Their business is to design and produce automation practise platforms for educational purposes.

The main goal of the thesis was to design an IoT-prototype on the JJJ-Automaatio's existing automation platform. The thesis focused on the most important IoT-concept fields which are hardware, software and services. The thesis also introduced the production process of the IoT-prototype, explaining all the above-mentioned connections in practise.

This thesis did not focus on a specific programming code or a specific sensor specification, instead it aimed to research what kind of a hardware and software could be used to explain the IoT-concept with automation practice platforms. IoT is a massive concept, which can be defined in multiple ways and for this reason some practical examples of IoT were added to the thesis to clarify the context in which the term is used in this work.

Keywords: IoT, cloud services, automation, electronics development platform

## SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ.....	4
Kuva-, kuvio- ja taulukkoluettelo.....	5
Käytetyt termit ja lyhenteet .....	7
<b>1 JOHDANTO .....</b>	<b>8</b>
1.1 Työn tausta.....	8
1.2 Työn tavoite .....	8
1.3 Yritysesittely .....	9
<b>2 LAITTEISTOT, SOVELLUKSET JA PILVIPALVELUT .....</b>	<b>10</b>
2.1 Laitteisto.....	11
2.1.1 Arduino.....	13
2.1.2 HC-SR04-ultraäänisensori.....	13
2.1.3 ESP8266 .....	16
2.1.4 Siemens simatic IoT2000 -tuoteperhe.....	17
2.2 Sovellukset ja pilvipalvelut .....	18
2.2.1 Blynk.....	18
2.2.2 Oma Blynk-palvelin.....	20
2.2.3 LoRaWAN .....	21
<b>3 PROTOTYYPIN RAKENTAMINEN .....</b>	<b>23</b>
3.1 Harjoitusalue .....	23
3.2 Kytkeä .....	24
3.3 Prototyypin mekaaninen toteutus .....	25
3.4 Ohjelmakoodi käytännön testauksessa .....	27
3.4.1 Arduino IDE-ympäristön paikalliset testit.....	28
3.4.2 Blynk-pilviympäristön testit .....	28
<b>4 POHDINTA.....</b>	<b>30</b>
<b>LÄHTEET.....</b>	<b>31</b>
<b>LIITTEET .....</b>	<b>33</b>

## Kuva-, kuvio- ja taulukkoluetelo

Kuva 1. Kuvassa oikealla Arduino Uno rev.3, jonka päälle on liitetty Wiznet W5500 Ethernet -moduuli, sekä HC-SR04-ultraäänisensori vasemmalla. ....	12
Kuva 2. HC-SR04-ultraäänisensori, kuvasta näkee helposti moduulin kytkentäpinnit. ....	13
Kuva 3. Ultraäänisensorin toimintaperiaate. Siniset viivat kuvaavat sensorista lähtevää pulssia ja harmaat kohteesta kimmonnutta pulssia. ....	14
Kuva 4. Ultraäänipulssin sekvenssikaavio. ....	14
Kuva 5. ESP8266 Wi-Fi mikrokontrolleri. ....	16
Kuva 6. Blynk-sovelluksen topologia (Blynk 2018). ....	19
Kuva 7. Blynk-sovelluksen pienoisohjelmia (Blynk 2018). ....	20
Kuva 8. Raspberry Pi, joka on suosituin yhden piirin tietokone. ....	21
Kuva 9. LAI-harjoittelualusta, johon prototyyppi on tarkoitus asentaa jälkepäin (JJJ-Automaatio 2015). ....	23
Kuva 10. Arduino UNO -kehitysalusta, jonka mikrokontrolleria käytetään prototyyppin ohjaukseen (Fritzing [viitattu 8.5.2018]). ....	24
Kuva 11. HC-SR04 kytkentä Arduinoon liitettävään Ethernet-moduuliin. Kuvan moduuli mahdollistaa Arduinon kommunikoinnin ulkomaailmaan (Fritzing [viitattu 8.5.2018]). ....	25
Kuva 12. Testikäyttöön tarkoitettu pahvinen makasiini. ....	26
Kuva 13. Testikäyttöä ja koodin kalibrointia varten tehdyt pahviset makasiinipalikat. ....	27
Kuva 14. Arduino IDE -ohjelman monitorointi-ikkuna, jossa Arduinon koodi testattiin paikallisesti. ....	28

Kuva 15. Arduino IDE, rivi 15: Blynk-sovelluksen todennusavain. Rivi 49: Blynk-sovelluksen LCD-näytön virtuaalisen pinnan määrittäminen. ....	29
Kuva 16. Blynk-sovellus käytössä, LCD- ja videostream-widgetit. ....	29
Kuvio 1. Prototyypijärjestelmän topologia. ....	11
Taulukko 1. HC-SR04-ultraäänisensorin spesifikaatiot (Elec Freaks [viitattu 10.5.2018]). ....	15

## Käytetyt termit ja lyhenteet

<b>DIY</b>	Do It Yourself, tee se itse.
<b>GPIO</b>	General Purpose Input Output, signaaleiden lähettämisen ja vastaanottamisen mahdollistavat tulot ja lähdöt.
<b>IoT</b>	Internet of Things, esineiden tai asioiden internet, teollinen internet.
<b>IDE</b>	Integrated Development Environment, integroitu ohjelmointiympäristö.
<b>IP-osoite</b>	Internet Protocol, internetiin kytkettyjen verkkosovittimien yksilöintiin tarkoitettu osoite.
<b>LAN</b>	Local Area Network, lähiverkko, tietoliikenneverkko, jonka kautta erilaiset laitteen kommunikoivat keskenään.
<b>LoRaWAN</b>	Low Range Wide Area Network, erityisesti IoT-tarpeisiin suunniteltu verkko.
<b>LPWAN</b>	Low Power Wide Area Network, lisensoimattomia radiotaajuuksia käyttävä vähävirtainen laajalanueen radiostandardi.
<b>MCU</b>	Micro Controller Unit, mikro-ohjain.
<b>PLC</b>	Programming Logic Controller, ohjelmoitava logiikka.
<b>Widget</b>	Pieniohjelma, joka suorittaa yksittäisen tietojenkäsittelytehtävän.
<b>Wi-Fi</b>	Kaupallinen nimitys WLAN-verkosta.
<b>WLAN</b>	Wireless Local Area Network, langaton lähiverkko.

# 1 JOHDANTO

(Mooren laki), jonka mukaan transistoreiden lukumäärä mikropiireissä kaksinkertaistuu noin kahden vuoden välein, vaikuttaa osakseen teolliseen internetin kehitykseen. Vaikutukset heijastuvat useisiin IoT-käsitteen osa-alueisiin. Merkityksellisempiä alueita tässä kehityksessä on mm. antureiden hinnan romahdus, verkkoteknologioiden kehitys ja tiedon analytiikkaan sovellettavien järjestelmien kehitys ja hinta. (Collin & Saarelainen 2016, 35-36.)

Tällä hetkellä eletään IoT-käsitteen muodostamaa neljättä teollisuuden vallankumouksen aikaa. Lähes kaikilla toimialoilla toimivilla yrityksillä on tarve selvittää tämän muutoksen mahdollistamia hyötyjä pysyäkseen kehityksen tuovan kasvun mukana. (Collin & Saarelainen 2016, 31.)

## 1.1 Työn tausta

JJJ-Automaatio halusi selvityksen siitä, miten voitaisiin lisätä IoT-tyyppisiä harjoituksia jo olemassa oleviin automaatioharjoitusalueisiin. Koska harjoitusalueita on suunnattu monille eri asteisille oppilaitoksille, prototyyppi tähän työhön tehtiin opiskelutarkoitukseen, ei niin, että tuote olisi valmis paketti teollisuuden käyttöön. Työhön valitaan Arduino Uno rev.3 -kehitysalusta, johon liitetään HC-SR04-ultraääni-sensori sekä Wiznet W5500 Ethernet -moduuli. Tämän prototyypin kaupallistamista pohditaan hieman teoriatasolla.

## 1.2 Työn tavoite

Tämän työn tavoitteena on tehdä toimiva ja helposti ymmärrettävissä oleva ratkaisu etävalvonta prototyypistä, johon on mahdollista lisätä opetusmielessä toteutettuja harjoituksia myöhemmin. Työssä pohditaan erilaisia ratkaisuja, mm. kuinka käytännön prototyyppi rakennetaan, mitkä ovat siihen valitut komponentit ja kuinka siitä saataisiin mielenkiintoinen kosketus erilaisiin tulevaisuuden tiedonsiirto- ja etäohjausjärjestelmiin. Lisäksi työssä selvitetään, miten tämän prototyypin kaupallistaminen olisi mahdollista annetuilla kriteereillä, ja onko jo olemassa tehdasvalmisteisia



IoT-järjestelmiä, joita voisi hyödyntää vanhempien opetusalustojen rinnalla niin sanottuina DIY-harjoituksina (Do It Yourself).

### **1.3 Yritysesittely**

JJJ-Automaatio Oy on Tampereella toimiva automaatioharjoituslaitteistoihin keskittyvä yritys. Yrityksen tuotteisiin kuuluu 12 erilaista harjoittelualustaa, joiden keskeinen osa-alue on oppimismateriaali. Näihin materiaaleihin kuuluu monipuoliset harjoitustehtävät erilaisista ohjauslaitteista, kytkennöistä ja käytännön sovelluksista. Uusimpana yrityksen harjoituslaitteistona voidaan mainita Suomessakin vauhdilla yleistynyt KNX-kotiautomaatiojärjestelmän harjoitusalue. (JJJ-Automaatio Oy 2018.)

JJJ-Automaatio Oy keskittyy pääosin koulutussektorille. Tämän vuoksi laitteistojen ja oppimateriaalin päivittäminen on jatkuva prosessi. Uusia tekniikoita valjastetaan teollisuuden käyttöön nopealla tahdilla ja siksi JJJ-Automaatio pyrkii etsimään ratkaisuja opetuskäyttöön tarkoitettujen harjoitusalueiden ja niiden oheismateriaalien uudistamiseen jatkuvasti.

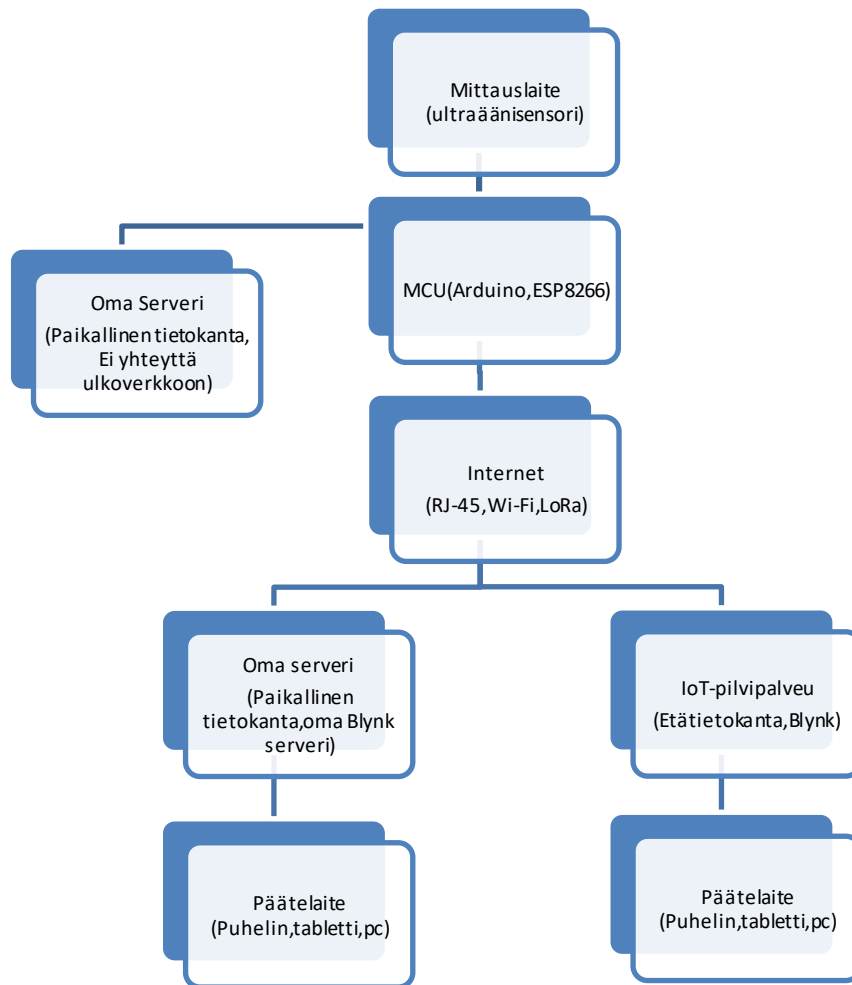
## 2 LAITTEISTOT, SOVELLUKSET JA PILVIPALVELUT

IoT (Internet of Things) on laaja käsitteeksi muodostunut termi, jota käytetään, kun yhdistetään fyysisiä arvoja mittaavia sensoreita internetiin. IoT muodostuu kolmesta pääalueesta: asioista, elektroniikasta ja yhdistettävyydestä (Sininenpolku.fi. 2016). Tässä työssä yhdistettävä osa on ultraäänisensori, jonka avulla valvotaan makasiinin täyttötilaa.

Seuraavan esimerkin avulla pyritään havainnollistamaan kyseistä asiaa.

*Esimerkkinä* toimii tavallinen sateenvarjo. Se otetaan mukaan ulos, kun on katsottu sää tiedotuksesta, että vettä sataa myöhemmin päivällä. Tämä on siis normaali tilanne. Mietitään, että sateenvarjoon lisätty ESP8266-mikrokontrolleri noutaisi tämän tiedon suoraan jostakin sää tietojen tarjoavasta internet palvelusta. Kun sadetta on luvassa, varjon kahvaan upotettu led-valo alkaisi vilkkua. Tällöin varjon omistaja tietää ottaa varjon mukaan ulos lähtiessään, vaikkei sää tietoja olisikaan katsonut muulla tavoin. (McEwen & Cassimally 2014, 10.)

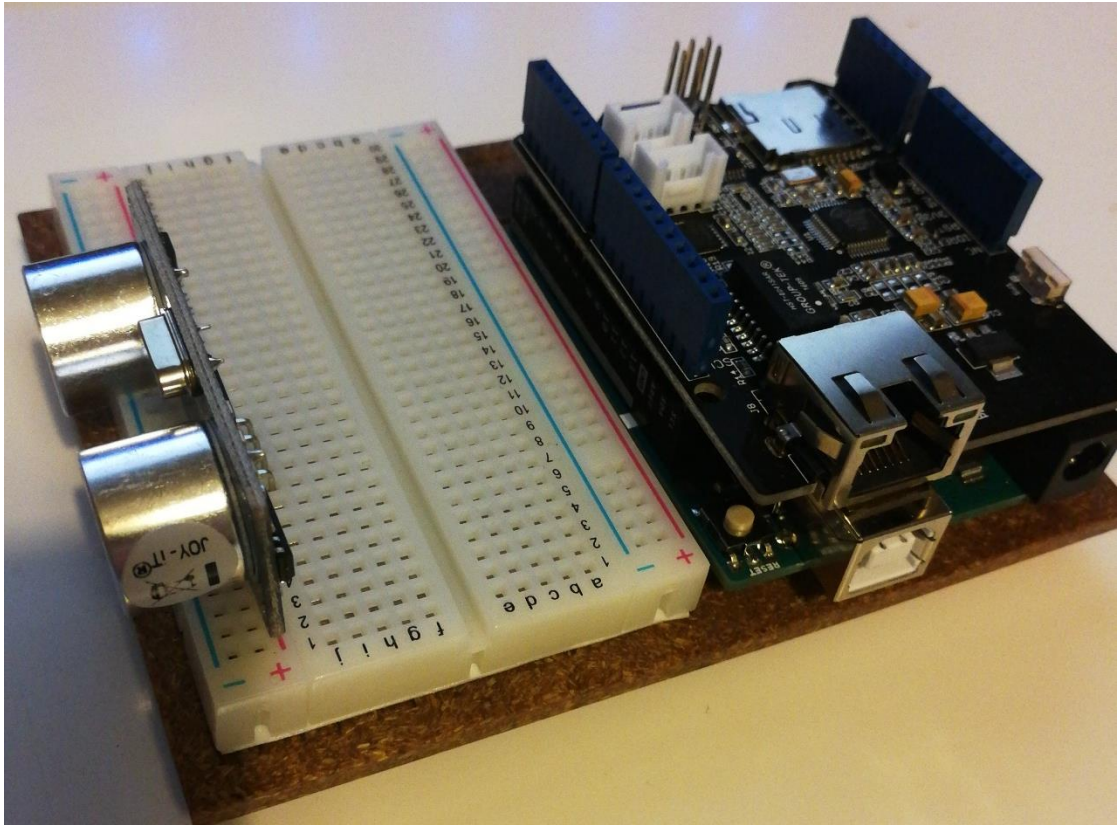
Edellä olevassa esimerkissä eloton asia (sateenvarjo) on tehty interaktiiviseksi. Varjo ei ainoastaan nouda sää tietoja internetistä, vaan se myös ilmaisee käyttäjälle tämän tiedon. Älykkään esineen, asian tai kokonaisuuden toteutus voidaan tehdä monella eri tavalla. Tämän työn prototyypin rakenteellisen toteutuksen näkee kuvista 1.



Kuvio 1. Prototyypijärjestelmän topologia.

## 2.1 Laitteisto

Saatavilla olevia MCU (Micro Controller Unit) sisältäviä kehitysalustoja on markkinoilla useita. Tähän työhön valittiin Arduino Uno Rev.3 -kehitysalusta, johon liitettiin HC-SR04-ultraäänisensori. Sensori mittaa makasiinin täyttöasteen, tämä tieto lähetetään Arduinon kautta myöhempää prosessointia varten. Tiedonsiirron Arduinosta internetiin/LAN (Local Area Network) -verkkoon hoidetaan erillisellä Wiznet W5500 Ethernet -moduulilla, joka kytketään RJ45-kaapelilla LAN/WAN-tukiasemaan (kuva 1).



Kuva 1. Kuvassa oikealla Arduino Uno rev.3, jonka päälle on liitetty Wiznet W5500 Ethernet -moduuli, sekä HC-SR04-ultraäänisensori vasemmalla.

Koska työn testausosuuden pystyi suorittamaan monilla erilaisilla ratkaisulla, oli tarpeellista selvittää myös vaihtoehtoisia mahdollisuuksia. Seuraavaksi on selvitetty erilaisten mikrokontrollerien soveltuvuutta tähän työhön ja merkittäviä IoT-aiheeseen liittyviä ratkaisuja yleisesti, mukaan lukien työn prototyyppiosuudessa käytetyt sensorit, kehitysalustat ja pilvipalvelu.

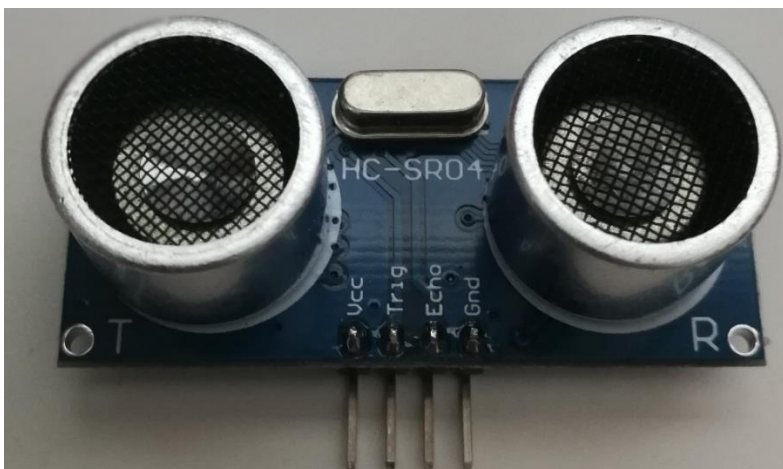
### 2.1.1 Arduino

Arduino kehitysalustan tuotekehitys on alkanut jo 2000-luvun alkupuolella Design Institute of Ireassa. Ensimmäinen Arduino kehitysalusta julkaistiin 2005 ja sen tarkoituksena oli tarjota oppilaille edullinen tapa tehdä mikrokontrolleriohjauksia konkreettisesti. Nykyisin Arduino ja sen eri tuoteperheen avoimiin ohjelmistoihin ja laitteistoihin perustuvat ratkaisut ovat käytössä maailmanlaajuisesti harrastelijoiden, opiskelijoiden, suunnittelijoiden ja yritysten osana tehdä ideoista totta. (Arduino 2018.)

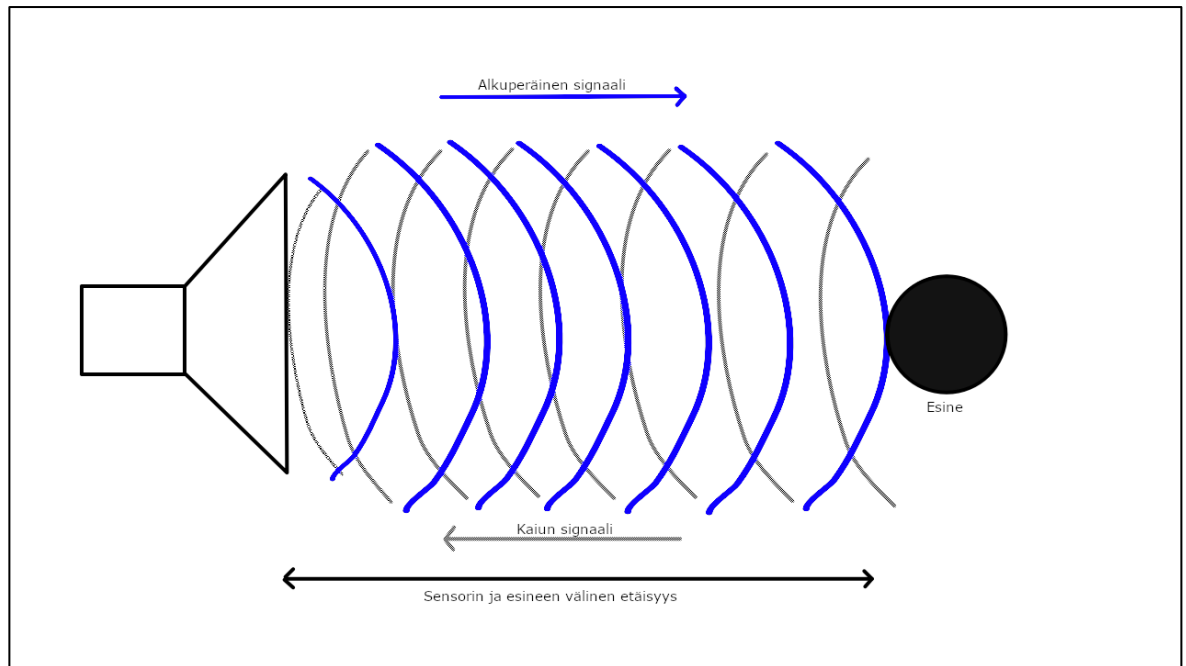
Tähän työhön kehitysalustaksi valittiin Arduino Uno rev.3, koska edullinen hinta, laajennettavuus, sekä toimintajännite sopivat erinomaisesti toteutettavaan prototyyppi-laitteistoon. Arduino on helppo tapa päästä sisälle elektroniikan komponentteihin, sensoreihin, ohjausjärjestelmiin, ohjelmointiin ja data-analytiikkaan. (Arduino 2018.) Tämä opinnäytetyö keskittyy vain pieneen osaan siitä, mitä kaikkea nykyaikaisilla kehitysalustoilla pystyisi rakentamaan.

### 2.1.2 HC-SR04-ultraäänisensori

HC-SR04-ultraäänisensorin (kuva 2) toiminta perustuu ultraääni-impulssin matkajan mittaamiseen. Tästä kuluneesta ajasta voidaan määrittellä sensorin ja pulssin kimmottaneen kohteen välinen etäisyys.



Kuva 2. HC-SR04-ultraäänisensori, kuvasta näkee helposti moduulin kytkentäpinnit.



Kuva 3. Ultraäänisensorin toimintaperiaate. Siniset viivat kuvaavat sensorista lähtevää pulssia ja harmaat kohteesta kimmonnutta pulssia.



Kuva 4. Ultraäänipulssin sekvenssikaavio.

Kuvissa 3 ja 4 havainnollistetaan ultraäänisensorin toimintaa. Sensorin input-pinniin johdetaan  $10\ \mu\text{s}$  kestävä TTL-signaali (kuva 4, trig-pinni), joka aktivoi sensorin lähettämään kahdeksan  $40\ \text{kHz}$ :n pulssin sarjan (kuva 4, sensorin lähettämät pulssit). Tämä pulssisarja heijastuu kohteesta takaisin aikavälillä  $150\ \mu\text{s}$  ja  $25\ \text{ms}$ , joista voidaan laskea kohteen etäisyys (kuva 4, echo-pinni).

Koska on tiedossa, kuinka kauan äänellä kestää liikkua tietty matka ilmassa (343 m/s 20 °C), niin ajasta voidaan laskea kohteen etäisyys seuraavalla kaavalla:

$$L = CT \tag{1}$$

jossa L on etäisyys kohteesta, C on äänen nopeus ilmassa, T on aika. (Elecfreaks [viitattu 10.5.2018].)

Arduinon virallinen käyttöjännite on 5 voltia, joka on yleisesti käytettävissä olevien kehitysalustoihin suunnattujen sensoreiden käyttöjännite. Tämä mahdollisti ultraäänisensorin suoran kytkennän Arduinoon, eikä väliin tarvita erillistä logiikkatason jännitteensäädintä (taulukko 1).

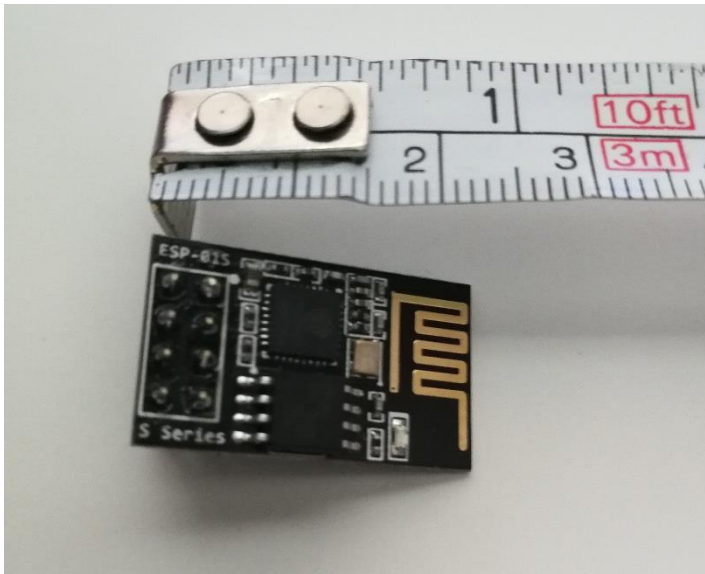
Taulukko 1. HC-SR04-ultraäänisensorin spesifikaatiot (Elecfreaks [viitattu 10.5.2018]).

Käyttöjännite	DC 5 V
Käyttövirta	15 mA
Havaintoetäisyys	2-400 cm
Tarkkuus	±0.3 cm
Havaintokulma	15°

Kuten taulukosta 1 voidaan todeta, sensorin tarvitsema käyttövirta on todella pieni muihin ominaisuuksiin nähden. Näiden arvojen vuoksi se soveltuu hyvin esimerkiksi täysin mobiileihin projekteihin, joissa tarvitaan esineen tunnistamisen lisäksi etäisyystietoa.

### 2.1.3 ESP8266

ESP8266 on vuonna 2014 kehitetty maailmanlaajuisesti tunnettu Wi-Fi-mikrokontrolleri. ESP8266-tuotteen kehitti Espressif Systems -niminen yritys, jonka pääkonttori sijaitsee Shanghaissa Kiinassa. ESP8266 on vähän virtaa tarvitseva, edullinen, pienikokoinen ja hyvän tieturvan omaava langattoman ratkaisun mahdollistava mikrokontrolleri WLAN-piiri (kuva 5). (Espressif 2018a.)



Kuva 5. ESP8266 Wi-Fi mikrokontrolleri.

Näiden ominaisuuksien vuoksi voidaan sanoa, että ESP8266-kontrollerin markkinoille tuleminen vuoksi langattomat järjestelmät lähtivät räjähdysmäiseen kasvuun kuluttajapuolella, ja osaksi tämän vuoksi IoT-käsite on saanut paljon julkisuutta lähivuosina. ESP8266-mikrokontrolleri on suosittu DIY-harrastelijoiden parissa, jotka haluavat kytkeä laitteensa internetiin tai muuten ohjata ja/tai monitoroida laitteitaan langattomasti. (Espressif 2018b.)

Tämän työn prototyypissä ei käytetä ESP8266-mikrokontrolleria, koska Arduinon suora liitettävyyden Siemens IoT2000 -perheen laitteistoihin mahdollistaa prototyypin liittämisen PLC-ohjattuihin automaatioharjoituslaitteisiin ja näin ollen mahdollistaa tuotteen kehittämisen työn tilanne yrityksen haluamaan suuntaan. Prototyyppi olisi kuitenkin ollut mahdollista tehdä toimivaksi pelkällä ESP8266-kontrollerilla, johon on liitetty HC-SR04-ultraäänisensori.



#### 2.1.4 Siemens simatic IoT2000 -tuoteperhe

Työn yhtenä tarkoituksena oli tehdä esiselvitys mahdollisesta kehitysalustojen liittämistä PLC-ohjauksen rinnalle, sekä olisiko se mahdollista tehdä rikkomatta harjoitusalojen tehdaskomponenttiperiaatetta. Näihin kriteereihin vastaava laitteisto löytyi Siemens Simatic IoT2000- tuoteperheestä.

Simatic IoT2040 ominaisuudet ovat:

- Intel Quark -prosessori 1 GB RAM, 2 Ethernet-porttia, 2 x RS232/485
- Syöttöjännite 9-36 VDC
- Yocto linux -tuki
- Laajennettavissa **Arduino**- ja miniPCIe-korteilla
- Ohjelmointikielet C/C++
- Kompakti koko ja DIN-kiskoasennus
- Simatic-laatu takaa kestävyttä, luetettavuutta ja pitkää ikää. (Siemens 2018.)

Tämän työn prototyypiosiossa ei puututa käytännön tasolla edellä mainittuun laitteistoon, mutta voidaan todeta prototyypin yhteensopivuus suoraan esimerkiksi IoT2040-tuotteeseen. Lisäksi useat harjoitusalojen ohjaukset ja anturit voidaan kytkeä tähän laitteeseen erillisellä IO-moduulilla. Tällaisen ratkaisun avulla säilytetäisiin harjoitusalojen teollinen rakenne, joka oli yksi JJJ-Automaation pyytämä selvityksen kohde ja tuotteen kaupallistamisen kriteeri.

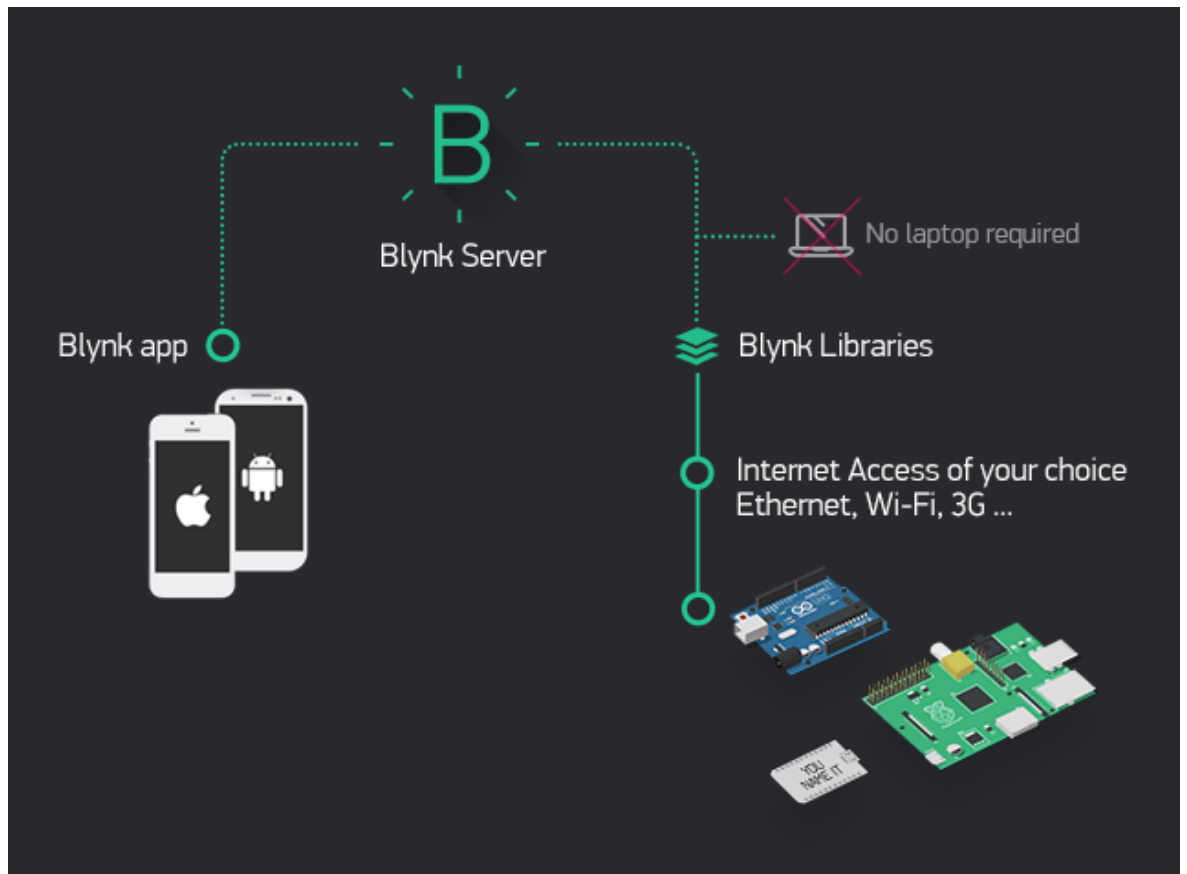
## 2.2 Sovellukset ja pilvipalvelut

Erialaisten pilvipalveluiden tarjonta on todella kirjavaa. Palveluita on mahdollista hankkia juuri tietyn tyyppisiin tarpeisiin, pienistä sovelluksista suuriin palvelukokonaisuuksiin. IoT-pilvipalvelun tarkoitus on poistaa oman fyysisen palvelinlaitteiston tarve ja saada toimilaitte yhdistettyä suoraan internetiin jotakin kommunikointitapaa käyttäen esimerkiksi Wi-Fi. Pilvipalvelun etuna on, että siihen yhdistettyjä laitteita voidaan monitoroida tai ohjata mistä tahansa internetin välityksellä. Lisäksi pilvipalveluun voidaan varastoida dataa pitkiltä aikaväleiltä, jolloin saadaan luotua esimerkiksi graafinen diagrammi tarkasteltavaksi. (Collin & Saarelainen 2016, 156.)

Pilvipalvelun valinnassa on tärkeää huomioida tietoturvallisuus, toimintavarmuus, hinta, sijainti ja yhdistettävyyt. Nämä asiat tulevat todella tärkeiksi, jotta palvelusta saadaan juuri omaan tarpeeseen sopiva kokonaisuus. (Collin & Saarelainen 2016, 156.)

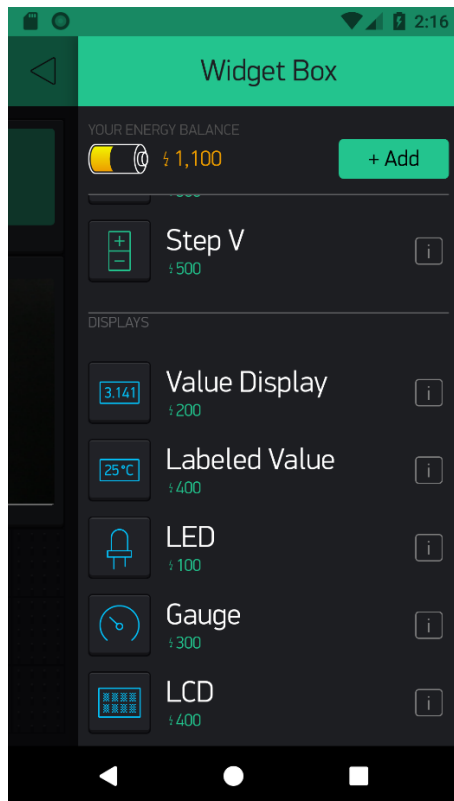
### 2.2.1 Blynk

Blynk on sovellus, jonka avulla saadaan helposti yhdistettyä Arduino tai joku muu samantapainen kehitysalusta Blynkin omaan pilvipalveluun tai paikalliseen Blynk-palvelimeen. Blynk toimii internetin välityksellä, ja siksi laitteiston, jota halutaan ohjata tai valvoa täytyy olla yhdistetty jollakin tapaa internetiin. Kun laitteistoon on määritelty jokin tapa kommunikoida Blynk-sovelluksen kanssa, siinä ei tarvita välissä muita laitteita, vaan esimerkiksi pelkkä Android-puhelin/tabletti riittää (kuva 4). (Blynk 2018.)



Kuva 6. Blynk-sovelluksen topologia (Blynk 2018).

Blynk-sovellus on helppo käyttää ja ilmainen normaalikäyttäjälle. Sen avulla voidaan ohjata/valvoa suoraan Arduinin GPIO-pinnejä, jolloin varsinaista paikallista koodia ei tarvita. Koodi on Blynk-sovelluksessa sisällä ja ohjauksiin tarvittava koodaus tapahtuu erilaisia pienoishjelmia lisäämällä (kuva 7). Tämä tapa soveltuu hyvin pienten järjestelmien rakentamiseen, mutta monimutkaisempien kokonaisuuksien tuottaminen vaatii usein myös ulkopuolista koodausta. (Blynk 2018.)



Kuva 7. Blynk-sovelluksen pienoisohjelmat (Blynk 2018).

### 2.2.2 Oma Blynk-palvelin

Oman Blynk-palvelimen toimintaperiaate on sama kuin normaalissa Blynk-sovelluksessa. Erona on se, että käyttäjä asentaa itselleen paikallisen palvelimen, joka toimii lähiverkossa tai internetin välityksellä. Etuina tässä on kolmannelle osapuolelle lähetettävien tietojen poistuminen. Oman palvelimen voi myös eristää ulkoverkosta täysin ja käyttää sitä LAN (Local Area Network) -ympäristössä. Tällöin saavutetaan korkeampi tietoturva, mutta menetetään ulkomaailman yhdistettävyyden tuomat mahdollisuudet. (Blynk 2018.)

Oman palvelimen ylläpitämiseen tarvitaan myös laitteisto, joka on useissa tapauksissa vaikea asentaa toimintaan. Laitteiston täytyy olla myös päällä kaiken aikaa, mikä lisää sähkönkulutusta. Näiden seikkojen vuoksi oman palvelimen ylläpitäminen on usein kannattamatonta, varsinkin pienemmissä järjestelmissä. (Wired 2018.)

Oman palvelimen voi toteuttaa useilla eri laitteistoilla. Yksi suosituimmista laitteistoratkaisuista IoT-palvelimeksi on harrastelijoiden keskuudessa noussut Raspberry Pi

(kuva 8). Raspberry Pi on yhden piirin tietokone, johon voi suoraan ladata valmistajan kotisivuilta Rasbian Os -käyttöjärjestelmän. Raspberry Pi -alustaan tarkoitettu Rasbian pohjautuu Debian-käyttöjärjestelmästä. (Opensource [viitattu 12.5.2018].)



Kuva 8. Raspberry Pi, joka on suosituin yhden piirin tietokone.

Raspberry Pi on Blynk-palvelinsovelluksen kanssa täysin yhteensopiva. (Blynk 2018). Tarvittavat ohjelmistot ja ohjeistukset ovat suoraan saatavilla Blynkin kotisivuilta. Opinnäytetyön aiheeseen ei varsinaisesti kuulunut oman palvelimen rakentaminen, vaan ainoastaan mahdollisuuksien tutkiminen vaihtoehtoisista ratkaisuista.

### 2.2.3 LoRaWAN

LoRaWAN-verkko on suunniteltu erityisesti esineiden internetiä varten. Verkon käyttö vaatii erityisen vähän virtaa ja sen kantama on parempi kuin esimerkiksi kilpailevalla WLAN-verkolla. LoRa-radioteknologiaan perustuva verkon tiedonsiirtokapasiteetti on kuitenkin suhteellisen pieni, joten se soveltuukin parhaiten pienten datamäärien siirtämiseen esimerkiksi anturin mittausarvoon. (Collin & Saarelainen, 139.)

Suomessa valtakunnallisesta LoRaWAN-verkon rakentamisesta vastaa Digita. Digitan LoRaWAN-verkko perustuu LPWAN (Low Power Wide Area Network) -verkoteknologiaan, jonka kehityksestä vastaa LoRa Alliance -järjestö. Digitan LoRaWAN-verkko kattaa koko Suomen ja lisäksi se on saatavilla useiden oppilaitosten käyttöön ilmaiseksi tuotekehitystä ja opiskelua varten. (Digita 2018.)

Tämän työn prototyypissä ei käytetä LoRa-radioteknologiaan perustuvaa verkkoa, vaan todetaan, että se pystyttäisiin toteuttamaan sillä. Seuraavaksi on esitelty teoreettinen esimerkki asiaan liittyen.

Oletetaan, että harjoituslaitteistoa pitäisi siirtää eri koulutuksista toisiin. Tällöin mukana pitäisi olla sopivat välineet alustan kytkemiseksi ulkomaailmaan tai vaihtoehtoisesti ohjelmakoodin muokkaaminen kyseisen tilan laitteistoihin sopivaksi. LoRaWAN-verkon avulla laitteistosta saataisiin täysin mobiili kaikkialla verkon peittoalueella olevissa kohteissa. Tämä tarkoittaa sitä, että tarvittava yhdyskäytävä internetiin muodostuu Digitan asentamien LoRa-vastaanottimien kautta ja tiedonsiirto sen avulla haluttuun pilvipalveluun.

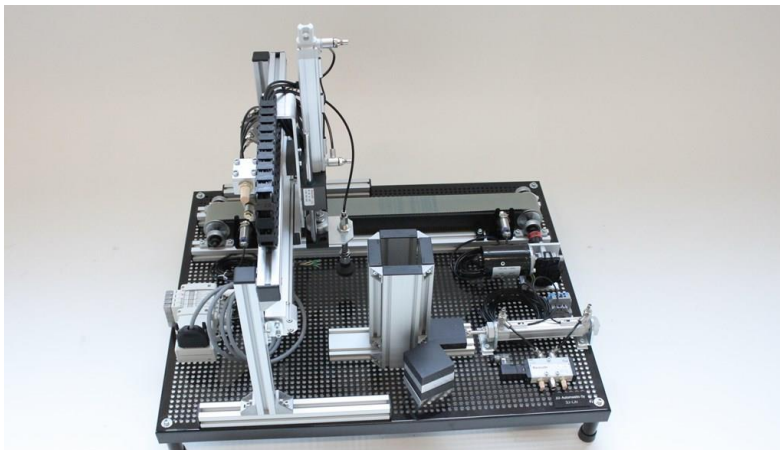
### 3 PROTOTYYPIN RAKENTAMINEN

Prototyypin avulla voidaan konkreettisesti osoittaa, että kehitettävä idea toimii käytännössä eikä se jää vain suunnitteluasteelle. Prototyypin tarkoitus ei ole tehdä suoraan valmista tuotetta, vaan auttaa rakentamaan prototyyppi osio kerallaan. (Karvinen & Karvinen 2010, 15-16.)

Tässä luvussa käydään läpi opinnäytetyön toiminnallista osuutta. Työ alkoi suunnittelulla, miten tavoitteeseen päästäisiin, mitkä olisivat työhön tarvittavat laitteistot, mikä pilvipalvelu valittaisiin ja minkälaista ohjelmointia työhön tarvitaan. Kaikki tämä suunniteltiin siten, että työn tilannut kohdeyritys voisi mahdollisesti hyödyntää kyseistä prototyyppiä ja siihen liittyvää selvitystä tulevaisuudessa. Lisäksi prototyyppiä varten tehtiin pahiset korvikkeet harjoitusalueen käytettävistä osista eli makasiinista ja makasiinissa olevista kappaleista. Näiden avulla testattiin laitteistoa ennen varsinaista sijoituspaikkaa.

#### 3.1 Harjoitusalue

Harjoitusalueena (kuva 9) toimii JJJ-Automaation suunnittelema avoimen logiikan ratkaisu, johon voi kytkeä minkä tahansa ohjauksen oman mieltymyksensä mukaan. Tässä prototyypissä ei kytketä mitään logiikkaa ohjaamaan toimilaitteita, vaan liitetään Arduino Uno -kehitysalustalle rakennettu ultraäänisensori mittaamaan kuljettimen makasiinin täyttöastetta. Tämän prototyyppilaitteiston avulla pyritään havainnollistamaan mitä IoT-käsite tarkoittaa ja mitä hyötyä sillä voidaan saavuttaa.

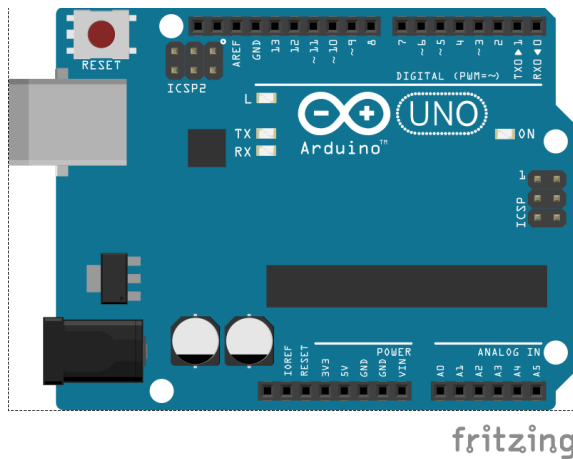


Kuva 9. LAI-harjoittelualusta, johon prototyyppi on tarkoitus asentaa jälkepäin (JJJ-Automaatio 2015).

### 3.2 KytKentä

Ennen varsinaisia johdotuksia, Arduino Uno -kehitysalustaan (kuva 10) liitettiin Wiznet W5500 Ethernet -moduuli (kuva 11). Moduuli tulee Arduino-alustan GPIO-pinnien päälle. On huomioitava, että moduuli itsessään käyttää joitakin Arduinon GPIO-pinnejä. Näiden pinnien käyttöä tulee välttää sensoreiden liittämisessä, jotta vältetään mahdollisilta häiriöiltä. Seuraavassa kytkennässä on varmistettu tarvittavien pinnien käytettävyys valmistajan tuotespesifikaatiosta.

Prototyyppilaitteiston kytkentä tehtiin ensiksi koekytkentälevylle. Koekytkentälevylle voidaan liittää erilaisia laitteistoja ilman juottamista. Tämän kytkennän avulla varmistettiin komponenttien toimivuus ennen niiden sijoittamista prototyypin aihioihin.

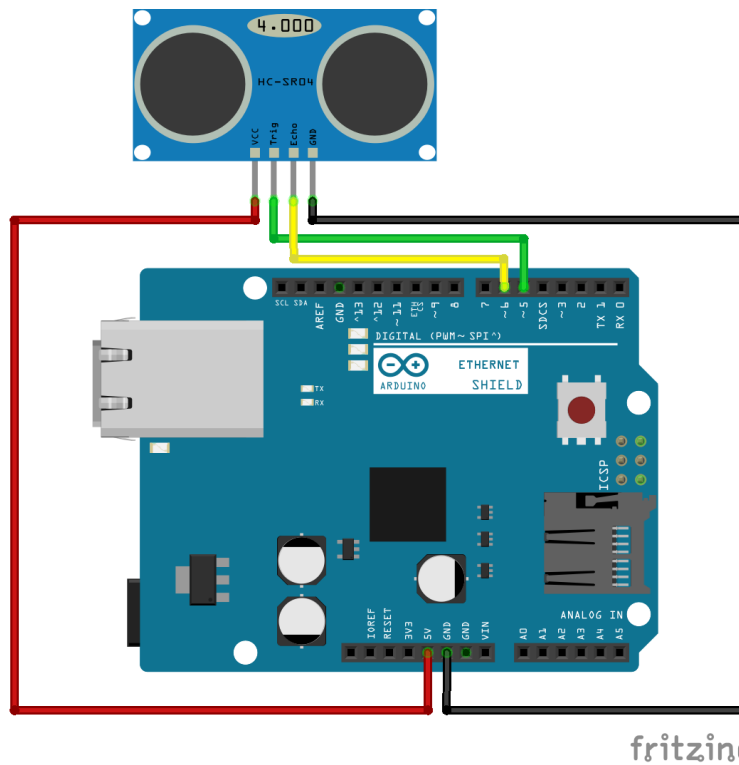


Kuva 10. Arduino UNO -kehitysalusta, jonka mikrokontrolleria käytetään prototyypin ohjaukseen (Fritzing [viitattu 8.5.2018]).

Kytkennässä musta johdin on GND (maa) eli negatiivinen johdin. Arduinossa on kaksi GND-pinniä, ei ole väliä kumpaan näistä pinneistä ultraäänisensorin GND kytketään (kuva 11). Punainen johdin on positiivinen virtajohdin ja se yhdistetään ultraäänisensorin VCC-pinnin, sekä Arduinon +5V-pinnin välille. Ultraäänisensori tarvitsee lisäksi kaksi datansiirtojohdinta. Kuvassa 11 vihreä johdin on Arduinon lähdeksi määritetyn 5-pinnin sekä ultraäänisensorin trig-pinnin välillä. Keltainen johdin on puolestaan Arduinon sisääntuloksi määritetyn 6-pinnin sekä ultraäänisensori echo-pinnin välillä. Tämän kytkennän avulla saadaan HC-SR04-ultraäänisensori liitettyä Arduino Uno -kehitysalustaan (kuva 11). Huomioitavaa on, että edellä kerrottu



kytkentä tehdään konkreettisesti W5500-moduuliin, koska se on liitetty Arduinon ”laajennukseksi”.



Kuva 11. HC-SR04 kytkentä Arduinoon liitettävään Ethernet-moduuliin. Kuvan moduuli mahdollistaa Arduinon kommunikoinnin ulkomaailmaan (Fritzing [viitattu 8.5.2018]).

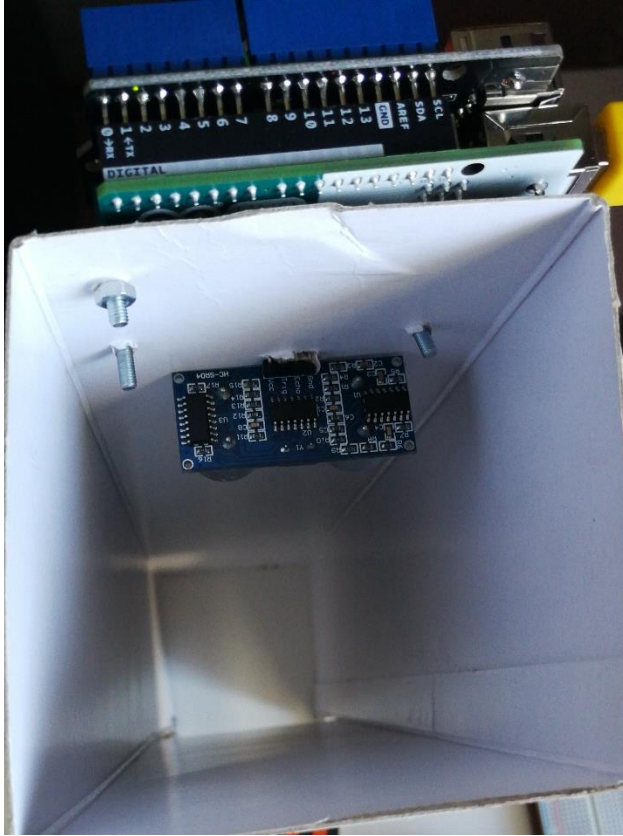
Kytkeäntöjen jälkeen yhdistettiin Arduino tietokoneeseen USB-kaapelin avulla, jota käytetään ohjelmakoodin lataamiseen, sekä käyttövirran syöttämiseen Arduinoon.

### 3.3 Prototyypin mekaaninen toteutus

Koska harjoituslusta (kuva 9) ei ollut fyysisesti saatavilla työtä tehdessä, käytettiin toteutuksessa vaihtoehtoisia välineitä korvaamaan lopullisia laitteita. Nämä korvaavat ratkaisut olivat tärkeitä, koska prototyypilaitteistoa täytyi pystyä testaamaan eri kehitysvaiheissa.

Makasiini rakennettiin pahvista JJJ-Automaation LAI-harjoituslustalla olevaa alumiinista makasiinia vastaavaksi (kuva 12). Makasiinin yhteneväiset mitat lopullisen sijoituskohteen vuoksi mahdollistivat prototyypin kalibroinnin ennalta. Tätä menetel-

mää käyttäessä täytyy ottaa huomioon, että pienikin ero anturin sijoittamisessa vaikuttaa sen tuottamaan lähtöarvoon. Edellä mainittu poikkeavuus on kuitenkin helposti korjattavissa ohjelmakoodia muokkaamalla.



Kuva 12. Testikäyttöön tarkoitettu pahvinen makasiini.

Lisäksi prototyyppiin tehtiin pahviset kappaleet (kuva 13), jotka vastaavat LAI-harjoituslustan kappaleiden fyysisiä mittoja (kuva 9). Materiaalilla ei ollut tässä tilanteessa väliä, koska lopullisessa työssä kappaleiden koostumusta ei tarkastella makasiinissa, vaan se suoritetaan myöhemmin harjoituslustan linjastolla olevien antureiden avulla.



Kuva 13. Testikäyttöä ja koodin kalibrointia varten tehdyt pahviset makasiinipalikat.

### 3.4 Ohjelmakoodi käytännön testauksessa

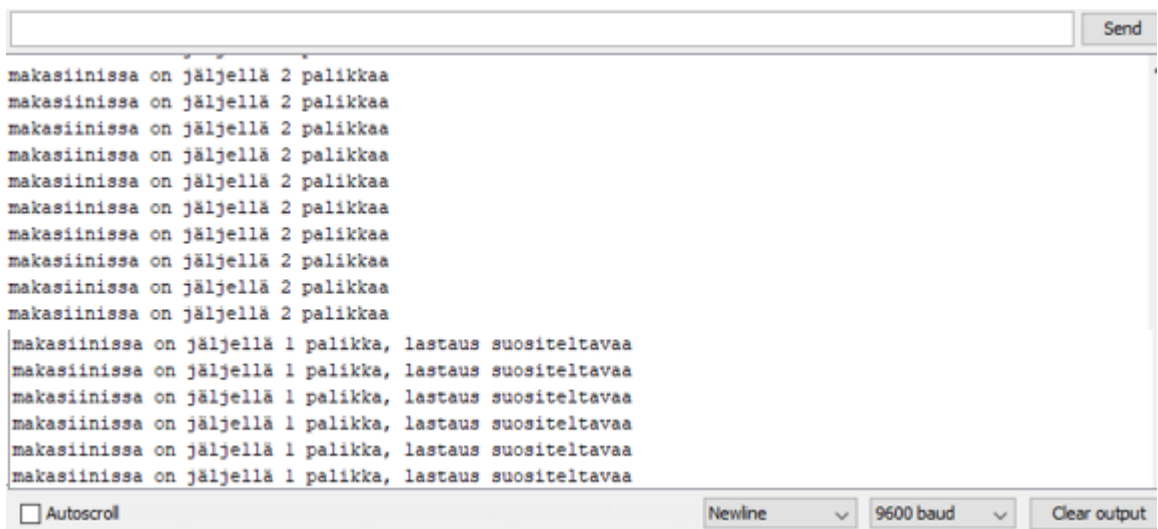
Prototyypin rakentamisessa on tärkeää, että ohjelmakoodi testataan heti ensimmäisen toiminnon valmistuttua. Tällä menetelmällä voidaan varmistua kehitysympäristön toimivuudesta. Sen avulla kehittäjä tietää, että mahdolliset myöhemmin esiintyvät viat ovat seurausta ohjelmakoodin virheestä, eikä esimerkiksi viallisesta komponentista tai tiedonsiirtoyhteydestä. (Karvinen & Karvinen 2010, 16.)

Käytännön testaus toteutettiin kahdessa osassa: paikalliset testaukset ja pilvitestatukset. Paikallinen testaus suoritettiin, koska haluttiin varmistua sovelluksen toimivuudesta ennen tietojen kierrättämistä pilvipalvelun kautta.

### 3.4.1 Arduino IDE-ympäristön paikalliset testit

Paikalliset testaukset olivat helppo toteuttaa Arduino IDE-ympäristöön kuuluvan sarjamonitorin kautta (kuva 14). Tämän avulla pystyttiin havainnollistamaan nopeasti mahdolliset virheet ohjelmakoodissa ja korjaamaan ne ennen pilviympäristöttestauksia.

Testaus suoritettiin manuaalisesti lisäämällä ja poistamalla kappaleita makasiinista. Kun haluttu sensoriarvon tarkkuus tällä menetelmällä oli saavutettu, siirryttiin pilviympäristön kehittämiseen.



The screenshot shows the Arduino IDE serial monitor window. The output text is as follows:

```

makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 2 palikkaa
makasiinissa on jäljellä 1 palikka, lastaus suositeltavaa
makasiinissa on jäljellä 1 palikka, lastaus suositeltavaa
makasiinissa on jäljellä 1 palikka, lastaus suositeltavaa
makasiinissa on jäljellä 1 palikka, lastaus suositeltavaa
makasiinissa on jäljellä 1 palikka, lastaus suositeltavaa
makasiinissa on jäljellä 1 palikka, lastaus suositeltavaa

```

At the bottom of the window, there are controls: an unchecked 'Autoscroll' checkbox, a 'Newline' dropdown menu, a '9600 baud' dropdown menu, and a 'Clear output' button.

Kuva 14. Arduino IDE -ohjelman monitorointi-ikkuna, jossa Arduinon koodi testattiin paikallisesti.

### 3.4.2 Blynk-pilviympäristön testit

Prototyypin pilvitestaamiseen valittiin Blynk-sovellus ja sen tarjoama oma pilvipalvelu. Nämä molemmat ovat ilmaisia, mutta kaupalliseen käyttöön vaaditaan palvelun ostaminen.

Ennen pilviympäristöttestauksia Arduino Uno liitettiin ulkoverkkoon RJ-45-kaapelilla. Tämän jälkeen Blynk-sovellukseen luotiin uusi projekti nimeltä opnt-ultraääni. Projektin sisälle pystyttiin lisäämään haluttuja pienoishohjelmia, joita tässä prototyypissä olivat LCD- ja videostream-widgetit (kuva 14). Videokuva sovellukseen toteu-



## 4 POHDINTA

Tämän työn avulla hankittu tieto ja käytännön kokemus erilaisista ohjaus- ja etähallintasovelluksista avasi yleisesti teollisuudessa käytettävissä olevien PLC-ratkaisujen toimintaa paljon. Esimerkiksi teollisuuteen tarkoitettu Siemens simatic loT2000-tuoteperhe ja Digitan rakentama LoRaWAN -verkko olivat mielenkiintoisia vaihtoehtoja.

Työn edetessä eteen tuli ongelmia siitä, miten saataisiin ohjelmakoodi optimoitua tulevaan harjoitusalueeseen ilman kyseistä harjoitusaluetta. Tämä vuoksi jouduttiin tekemään vastaava laitteisto pahvisena. Pahvisen laitteiston rakentaminen lisäsi aiheeseen paneutumista entisestään ja sai aikaan mielenkiinnon heräämisen käytännön prototyypin rakentamisesta ja siihen liittyvien ongelmien selvittämisestä.

Vaikka prototyypin suorittama toiminto voidaan tehdä kokonaisuudessaan esimerkiksi kalliilla PLC-laitteistolla, tässä työssä tuodaan esiin vaihtoehtoinen ratkaisu. Kehitysalustojen ja niihin tarkoitettujen sensoreiden edullisuuden vuoksi ovat ne loistava mahdollisuus opiskelijoille tutustua asiaan tarkemmin esimerkiksi kotona ja jatkaa itsensä kehittämistä koulun ulkopuolella.

IoT-käsitteeseen liitettäviä sovelluksia, komponentteja, sensoreita ja palveluita on tarjolla paljon. Tämän vuoksi on lähes varmaa, että jokainen löytää omaan työhönsä sopivan ratkaisun. Ilmaiseksi löytyvää ohjelmamateriaalia löytyy internetistä runsaasti, puhuvasta nallesta oman sähköauton rakentamiseen asti.

Työn aiheen valintaan vaikutti oma kiinnostuksen kohde kehitysalustoihin, prototyypin kehittämiseen ja yleisesti ottaen IoT-käsitteen tutkimiseen. Aiheesta on varmasti apua tulevaisuudessa. Uskon, että IoT-käsitteeseen perehtyneestä automaatioasiantuntijasta, joka ymmärtää PLC-ohjelmoinnista ja mikrokontrollereihin käytettävästä ohjelmakoodista, on tarvetta yrityselämässä.

## LÄHTEET

- Arduino. 2018. About Us. [Verkkosivu]. Arduino. [Viitattu 8.5.2018]. Saatavissa: <https://www.arduino.cc/en/Main/AboutUs>
- Blynk. Ei päiväystä. How Blynk Works. [Verkkosivu]. Blynk. [Viitattu 10.5.2018]. Saatavissa: <http://docs.blynk.cc/#intro-how-blynk-works>
- Collin, J. & Saarelainen, A. 2016. Teollinen internet. Helsinki: Talentum.
- Digita. Ei päiväystä. Mikä on Iorawan. [Verkkosivu]. Digita Oy. [Viitattu 4.5.2018]. Saatavissa: [https://www.digita.fi/yrityksille/iot/mika\\_on\\_iorawan](https://www.digita.fi/yrityksille/iot/mika_on_iorawan)
- ElecFreaks. Ei päiväystä. HC-SR04 User Guide. [Verkkosivu]. ElecFreaks. [Viitattu 10.5.2018]. Saatavissa: [https://elec Freaks.com/estore/download/EF03085-HC-SR04\\_Ultrasonic\\_Module\\_User\\_Guide.pdf](https://elec Freaks.com/estore/download/EF03085-HC-SR04_Ultrasonic_Module_User_Guide.pdf)
- Espressif. 2018a. Esp8266ex overview. [Verkkosivu]. Espressif Systems. [Viitattu 8.5.2018]. Saatavissa: <https://www.espressif.com/en/products/hardware/esp8266ex/overview>
- Espressif. 2018b. Espressif Achieves the 100-Million Target for IoT Chip Shipments. [Verkkosivu]. Espressif Systems. [Viitattu 9.5.2018]. Saatavissa: [https://www.espressif.com/en/media\\_overview/news/espressif-achieves-100-million-target-iot-chip-shipments](https://www.espressif.com/en/media_overview/news/espressif-achieves-100-million-target-iot-chip-shipments)
- Fritzing. Ei päiväystä. Fritzing home. [Verkkosivu]. Fritzing. [Viitattu 8.5.2018]. Saatavissa: <http://fritzing.org/home/>
- JJJ-Automaatio. 2015. 3J-LAI kuljetin- / portaalilaitteisto. [Verkkolähde]. JJJ-Automaatio Oy. [Viitattu 10.5.2018]. Saatavissa: <http://www.jjj-automaatio.fi/fi/3j-lai/>
- Karvinen, T. & Karvinen, K. 2010. Sulautetut: Opi rakentamaan robotteja ja muita sulautettuja järjestelmiä. Helsinki: Readme.fi.
- McEwen, A. & Cassimally, H. 2014. Designing the internet of things. Chichester: Wiley.
- Opensource. Ei päiväystä. Raspberry-pi. [Verkkosivu]. Opensource.com. [Viitattu 12.5.2018]. Saatavissa: <https://opensource.com/resources/raspberry-pi>
- Siemens. 2018. Industrial IoT. [Verkkosivu]. Siemens AG. [Viitattu 12.5.2018]. Saatavissa: <https://w3.siemens.com/mcms/pc-based-automation/en/industrial-iot/pages/default.aspx>

Sininenpolku.fi. 2016. Internet of Things (IoT) eli Esineiden Internet: Mistä on kyse? [Blogikirjoitus]. [Viitattu 7.5.2018]. Saatavissa: <https://www.sininenpolku.fi/fi/2016/06/01/internet-of-things-iot-eli-esineiden-internet-mista-on-kyse/>

Wired. 2018. Set up a home server. [Blogikirjoitus]. [Viitattu 9.5.2018]. Saatavissa: [https://www.wired.com/2010/02/set\\_up\\_a\\_home\\_server/](https://www.wired.com/2010/02/set_up_a_home_server/)



## LIITTEET

Liite 1. Paikallisen testauksen Arduino IDE-ohjelmakoodi.

Liite 2. Prototyypin lopullinen ohjelmakoodi.

## LIITE 1 OPNT.ino

Paikallisen testauksen Arduino IDE-ohjelmakoodi.

```

OPNT §
1  /*
2  Ultraäänianturi pinnit - arduino pinnit:
3  VCC: +5VDC
4  Trig : Trigger (INPUT) - Pin5
5  Echo: Echo (OUTPUT) - Pin 6
6  GND: GND
7  */
8
9  int trigPin = 5;    //Trig - vihreä kaapeli
10 int echoPin = 6;   //Echo - keltainen kaapeli
11 long duration, cm;
12
13 void setup() {
14     //Serial Port begin
15     Serial.begin (9600);
16     //Määritetään arduinon tulot/lähdöt
17     pinMode(trigPin, OUTPUT);
18     pinMode(echoPin, INPUT);
19 }
20
21 void loop()
22 {
23     // Sensorille annetaan 10uS high pulssi trigger pinniin.
24     digitalWrite(trigPin, LOW);
25     delayMicroseconds(2);
26     digitalWrite(trigPin, HIGH);
27     delayMicroseconds(10);
28     digitalWrite(trigPin, LOW);
29
30     // luetaan kuinka kauan High pulssilla kesti kimmota takaisin echo sisääntuloon
31     pinMode(echoPin, INPUT);
32     duration = pulseIn(echoPin, HIGH);
33
34     // Muunnetaan kulunut pulssin lähtö-paluu aika matkaksi.
35     // Kulunut aika jaetaan kahdella koska ääni kulkee kohteeseen ja kohteesta takaisin
36     //
37     cm = (duration/2) / 29.1;
38
39     if (cm <= 30 && cm >= 28) {
40         Serial.print("makasiinissa on jäljellä 1 palikka, lataus suositeltavaa");
41         Serial.println();
42     }
43     else if (cm <= 27 && cm >= 26) {
44         Serial.print("makasiinissa on jäljellä 2 palikkaa");
45         Serial.println();
46     }
47     else if (cm <= 25 && cm >= 23) {

```

```
47▢ else if (cm <= 25 && cm >= 23) {
48   Serial.print("makasiinissa on jäljellä 3 palikkaa");
49   Serial.println();
50 }
51▢ else if (cm <= 22 && cm >= 20) {
52   Serial.print("makasiinissa on jäljellä 4 palikkaa");
53   Serial.println();
54 }
55▢ else if (cm <= 19 && cm >= 17) {
56   Serial.print("makasiinissa on jäljellä 5 palikkaa");
57   Serial.println();
58 }
59▢ else if (cm <= 16 && cm >= 14) {
60   Serial.print("makasiinissa on jäljellä 6 palikkaa");
61   Serial.println();
62 }
63▢ else if (cm <= 13 && cm >= 11) {
64   Serial.print("makasiini on tynä");
65   Serial.println();
66 }
67 else
68▢ {
69   Serial.print("Makasiini tyhjä");
70   Serial.println();
71 }
72 //poista kommentointi seuraavasta kolmesta rivistä kalibroinnin helpottamiseksi.
73 //Serial.print(cm);
74 //Serial.print("cm");
75 //Serial.println();
76
77   delay(50);
78 }
```

---

## LIITE 2 OPNT-blynk.ine

## Prototyypin lopullinen ohjelmakoodi.

```

OPNT-blynk$
1 □ /*
2   Ultraäänianturi pinnit - arduino pinnit:
3   VCC: +5VDC
4   Trig : Trigger (INPUT) - Pin5
5   Echo: Echo (OUTPUT) - Pin 6
6   GND: GND
7   */
8   #define BLYNK_PRINT Serial
9
10
11  #include <SPI.h>
12  #include <Ethernet2.h>
13  #include <BlynkSimpleEthernet2.h>
14
15  char auth[] = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX";
16
17  int trigPin = 5;    //Trig - vihreä kaapeli
18  int echoPin = 6;   //Echo - keltainen kaapeli
19  long duration, cm;
20
21 □ void setup() {
22   //Serial Port begin
23   Serial.begin (9600);
24   Blynk.begin(auth);
25   //Määritetään arduinon tulot/lähdöt
26   pinMode(trigPin, OUTPUT);
27   pinMode(echoPin, INPUT);
28 }
29
30 void loop()
31 □ {
32   // Sensorille annetaan 10uS high pulssi trigger pinniin.
33   digitalWrite(trigPin, LOW);
34   delayMicroseconds(2);
35   digitalWrite(trigPin, HIGH);
36   delayMicroseconds(10);
37   digitalWrite(trigPin, LOW);
38
39   // luetaan kuinka kauan High pulssilla kesti kimmota takaisin echo sisääntuloon
40   pinMode(echoPin, INPUT);
41   duration = pulseIn(echoPin, HIGH);
42
43   // Muunnetaan kulunut pulssin lähtö-paluu aika matkaksi.
44   // Kulunut aika jaetaan kahdella koska ääni kulkee kohteeseen ja kohteesta takaisin
45   //
46   cm = (duration/2) / 29.1;
47
48

```

```
49 WidgetLCD lcd(V1); //Virtuaalisen pinnan määrittäminen Blynk sovellusta varten.
50
51 if (cm <= 30 && cm >= 28) {
52     Serial.print("makasiinissa on jäljellä 1 palikka, lastaus suositeltavaa");
53     lcd.print(0, 0, "makasiinissa 1 palikka"); //Makasiinin tilatiedon tulostus virtuaaliseen LCD-näyttöön.
54     Serial.println();
55 }
56 else if (cm <= 27 && cm >= 26) {
57     Serial.print("makasiinissa on jäljellä 2 palikkaa");
58     lcd.print(0, 0, "makasiinissa 2 palikkaa");
59     Serial.println();
60 }
61 else if (cm <= 25 && cm >= 23) {
62     Serial.print("makasiinissa on jäljellä 3 palikkaa");
63     lcd.print(0, 0, "makasiinissa 3 palikkaa");
64     Serial.println();
65 }
66 else if (cm <= 22 && cm >= 20) {
67     Serial.print("makasiinissa on jäljellä 4 palikkaa");
68     lcd.print(0, 0, "makasiinissa 4 palikkaa");
69     Serial.println();
70 }
71 else if (cm <= 19 && cm >= 18) {
72     Serial.print("makasiinissa on jäljellä 5 palikkaa");
73     lcd.print(0, 0, "makasiinissa 5 palikkaa");
74     Serial.println();
75 }
76 else if (cm <= 17 && cm >= 15) {
77     Serial.print("makasiinissa on jäljellä 6 palikkaa");
78     lcd.print(0, 0, "makasiinissa 6 palikkaa");
79     Serial.println();
80 }
81 else if (cm <= 14 && cm >= 11) {
82     Serial.print("makasiini on täynnä");
83     lcd.print(0, 0, "makasiini on täynnä");
84     Serial.println();
85 }
86 else
87 {
88     Serial.print("Makasiini tyhjä");
89     lcd.print(0, 0, "makasiinissa 0 palikkaa");
90     Serial.println();
91 }
92
93
94 delay(50);
95 Blynk.run();
96 }
```