

Opinnäytetyö (AMK)

Elektroniikan koulutusohjelma

Tietoliikennejärjestelmät

2017

Patrick Lundelin

# ANTURITIE TOJEN ESITTÄMINEN JA TALLENTAMINEN KÄYTTÄEN ÄLYPUHELINSOVELLUKSIA SEKÄ LINUX -PALVELINTA

Patrick Lundelin

# ANTURITIETOJEN ESITTÄMINEN JA TALLENTAMINEN KÄYTTÄEN ÄLYPUHELINSOVELLUKSIA SEKÄ LINUX - PALVELINTA

Anturitietojen seuranta reaaliajassa on monesti tarpeellista paitsi tuotantoympäristöissä myös tuotteen testaus- ja prototyypivaiheessa. Antureina toimivat useasti erilaiset ympäristöolosuhteita tai laitteen toimintaa mittaavat sensorit. Esineiden internetin myötä melkein mikä tahansa laite voidaan liittää internetiin. Laitteesta riippuen, sille voidaan tarjota internetistä käsin palveluja tai laitetta voidaan monitoroida etäältä internet-yhteyden välityksellä.

Työssä esitettiin sovellusesimerkkien kautta kuinka älypuhelimelle rakennetaan sovellus, josta anturitietoja voidaan seurata reaaliajassa. Työssä tutustuttiin työkaluohjelmiin, joilla tarvittavan älypuhelinsovelluksen saa rakennettua ilman varsinaisia älypuhelimien sovelluskehitykseen tarkoitettuja ohjelmointikieliä. Yhteysmenetelmänä käytettiin älypuhelimien ja mikrokontrolleriin kytkettyjen anturien välillä sekä Bluetoothia että suoraan internetiin kytkettyä mikrokontrolleria, jolloin anturitietoa luetaan älypuhelimella internet-yhteyden välityksellä. Työssä otettiin käyttöön myös Linux-palvelin, jolloin anturitiedot saadaan näkyviin tietokoneen näytölle palvelimeen asennetun www-palvelimen kautta sekä tallennettua palvelimeen asennettuun tietokantaan. Linux-palvelimen osalta tutustuttiin Linux -käyttöjärjestelmään ja esimerkissä vaadittavien ohjelmien asentamiseen, myös palvelinohjelmointiin käyttäen pääasiassa PHP- ja Python-ohjelmointikieliä.

Työn lopputuloksena saatiin rakennetuksi järjestelmä, joka sisältää kaikkien työssä esitettyjen erillisten sovellusesimerkkien mukaiset ominaisuudet ja toiminnot. Työn tavoitteena oli rakentaa järjestelmä, jota olisi helppo kehittää eteenpäin tarvittavien vaatimusten ja ominaisuuksien mukaisesti. Saavutettujen tulosten perusteella voidaan todeta, että tavoite toteutui. Tulosten pohjalta pohdittiin, kuinka järjestelmää voisi kehittää jatkossa, kun tavoitteen mukainen toimiva perusta järjestelmän laajentamiselle oli saatu onnistuneesti rakennettua.

## ASIASANAT:

Anturi, Arduino, Bluetooth, Linux, PHP, Python, Raspberry Pi

Patrick Lundelin

# DISPLAYING AND STORING SENSOR DATA USING MOBILE PHONE APPLICATIONS AND A LINUX –BASED SERVER

Monitoring sensor data in real time is often necessary in production environments and when testing or prototyping a device. Sensors for measuring environmental conditions are often used for this purpose as well as sensors for measuring the operation of a device. Internet of Things makes it possible to connect almost any device to the Internet. Depending on the device, services can be offered to it from the Internet or the device can be remotely monitored over an Internet connection.

This thesis provides examples of how to build a mobile phone application for displaying sensor data in real time. Different web -based software development platforms were used in order to build the application without the need for actual mobile phone programming. The data from the sensors was sent from the microcontroller to the mobile phone via Bluetooth. A microcontroller connected directly to the Internet was also used for the same purpose. In the thesis, a Linux –based server system was also built with the capabilities of storing sensor data to a database and displaying the data on the computer screen through a locally installed web –server. In this example the purpose was to become acquainted with the Linux operating system and server side programming. PHP and Python were mainly used as the programming languages.

As a final result a working system was accomplished, including all the functionality from the examples shown in the thesis. The goal of the thesis was to build a system, which would be easy to expand and further develop. Based on the results, as a working foundation of the system was accomplished, the future improvements of the system are discussed.

## KEYWORDS:

Arduino, Bluetooth, Linux, PHP, Python, Raspberry Pi, Sensors

# SISÄLTÖ

<b>1 JOHDANTO</b>	<b>6</b>
<b>2 TIEDONSIIRTOTEKNIIKAT</b>	<b>7</b>
2.1 Bluetooth	7
2.2 LoRaWAN	8
2.3 Ethernet	8
2.4 Esineiden internet	8
<b>3 KEHITYSTYÖKALUT</b>	<b>10</b>
3.1 App Inventor-sovelluskehitysympäristö	10
3.2 Cayenne-sovellus- ja ohjelmointialusta	10
3.3 Arduino	11
3.3.1 Arduino Uno	11
3.3.2 Arduino Leonardo ETH	12
3.3.3 Arduino IDE	13
3.4 Raspberry Pi	13
3.5 Android Studio	14
3.6 Xamarin for Visual Studio	14
3.7 Komponentit ja anturit	15
3.8 VMware workstation	16
<b>4 SOVELLUKSET</b>	<b>18</b>
4.1 Kytkenät	18
4.2 Sovellusesimerkki 1	19
4.3 Sovellusesimerkki 2	25
4.4 Sovellusesimerkki 3	29
<b>5 POHDINTA</b>	<b>37</b>
<b>6 YHTEENVETO</b>	<b>39</b>
<b>LÄHTEET</b>	<b>40</b>

## KUVAT

Kuva 1. Arduino Uno.	12
Kuva 2. Arduino Leonardo ETH.	13
Kuva 3. Raspberry Pi 3 Model B.	14
Kuva 4. Velleman HC-05 Bluetooth-moduuli ja pinnijärjestys.	15
Kuva 5. Valovastus.	16
Kuva 6. Maxim Integrated DS18B20 lämpötila-anturi.	16
Kuva 7. Sovellusesimerkissä 1 käytetty kytkentä.	19
Kuva 8. Älypuhelimien näytön suunnittelunäkymä.	20
Kuva 9. Älypuhelinsovellus lohkorakenteena.	21
Kuva 10. Valmis älypuhelinsovellus.	21
Kuva 11. Kytkentä, johon on lisätty lämpötila-anturi.	22
Kuva 12. App Inventorin suunnittelunäkymä kahden anturin lukemista varten.	23
Kuva 13. Lohkorakenne kahden anturin lukemista varten.	24
Kuva 14. Kahden anturin reaaliaikaiset arvot älypuhelimien näytöllä.	24
Kuva 15. Cayennen tarvitsemat funktiokirjastot.	25
Kuva 16. Valmis kojelauta Cayenessa.	26
Kuva 17. Arduino Leonardo ETH:n verkkokortin valinta.	26
Kuva 18. Yhteystestaus Cayennen palvelimelle.	27
Kuva 19. DS18B20 lämpötila-anturin asetuksia Cayenessa.	27
Kuva 20. Lämpötila-anturin hälytysrajan asettaminen.	29
Kuva 21. Cayennen lähettämä sähköposti lämpötila-arvon ylittymisestä.	29
Kuva 22. Etäyhteys Raspberry Pihin.	30
Kuva 23. MySQL -tietokannan perustaminen.	32
Kuva 24. Valmis www-sivu.	33
Kuva 25. Linux-palvelimen lähettämiä sähköpostihälytyksiä.	33
Kuva 26. IFrame-tagin määrittäminen.	34
Kuva 27. Pythonin split -toiminto.	35
Kuva 28. Anturiarvojen tallennus tietokantaan.	35

## LIITTEET

Liite 1. Arduino -ohjelma.
Liite 2. HTML/JavaScript -ohjelma.
Liite 3. PHP -ohjelma.
Liite 4. Python -ohjelma.
Liite 5. Anturikytkentä.
Liite 6. Ohjelmien asennuskomennot.

# 1 JOHDANTO

Esineiden internet on tullut koko ajan lähemmäs tavallisenkin kuluttajan arkea. Kodinkoneista televisiot, digisovittimet sekä dvd- ja Blu-ray-soittimet ovat jo pitkän aikaa olleet suoraan yhteydessä internetiin. Internetin eri palveluista käsin kyseisiin laitteisiin tuotetaan sisältöä ja ohjelmistopäivityksiä. Suuntaus näyttää jatkuvan. Jääkaappeihinkin on saatavilla kameroita, joiden kuva on katsottavissa älypuhelimien asennetulla sovelluksella. Ruokakaupassa voidaan siis kurkistaa omaan jääkaappiin ja todeta mitä sieltä puuttuu. Esineiden internet mahdollistaa myös kuluttajan tai yrityksen tehdä omia sovellutuksia, joilla voidaan valvoa antureita ja sensoreita jotka mittaavat esimerkiksi ympäristön lämpötilaa tai jotakin muuta tärkeää tietoa ja lukea tätä tietoa internetin yli tietokoneelta tai miltä tahansa älylaitteelta missä ja milloin tahansa.

Tässä opinnäytetyössä käydään läpi esimerkkien kautta, kuinka mikrokontrolleriin kytkettyjä antureita luetaan ja kuinka luettu tieto saadaan päivittymään reaaliaikaisesti älypuhelimien ja tietokoneen näytölle. Työssä perehdytään erilaisten ohjelmistokehitysympäristöjen käyttöön sekä Linux-palvelimen asentamiseen ja palvelinohjelmointiin. Ohjelmointikielinä käytetään pääsääntöisesti C:tä, Pythonia ja PHP:tä. Tiedonsiirtomenetelminä käytetään Bluetoothia langattomaan yhteyteen sekä suoraan internetiin tai lähiverkkoon kytkettyjä laitteita.

Työssä esitetään kolme esimerkkiä, joista kahdessa ensimmäisessä käytetään ohjelmistokehitysympäristöä älypuhelimien ohjelmointiin. Nämä kehitysympäristöt ovat MIT App Inventor sekä myDevices Cayenne. Viimeisessä esimerkissä asennetaan koko järjestelmä alusta alkaen Linux-alustalle aloittaen tarvittavien sovellusten asentamisesta ja päättyen palvelimessa tarvittavien toimintojen ohjelmointiin. Esitettävät esimerkit voidaan ottaa kaikki käyttöön samalla kertaa ja samassa järjestelmässä, vaikka ne työssä käydään läpi erillisinä esimerkkeinä. Näin voidaan rakentaa järjestelmä, jolla saavutetaan kaikki sovellusesimerkeissä esitetyt toiminnallisuudet.

## 2 TIEDONSIIRTOTEKNIIKAT

Työssä käytetään tiedonsiirtotekniikoina Bluetoothia langattomaan tiedonsiirtoon sekä lähiverkkoon että internetiin kytkettyjä laitteita. LoRaWAN-tekniikka esitellään lyhyesti, koska käytetyistä sovelluskehitysympäristöistä vain Cayenne tukee LoRaWAN-laitteita. LoRaWAN-laitteita ei kuitenkaan ollut käytettävissä tätä työtä varten.

Bluetooth- ja LoRaWan-tekniikat ovat esimerkkejä lyhyen ja pitkän kantaman yhteystyypeistä, joita esineiden internetiin kytketyissä laitteissa useasti käytetään. Yhteystyyppejä on useita muitakin, mutta tässä esittelyssä käydään läpi tekniikat joita tässä opinnäytetyössä käytetään ja joille löytyy tuki työssä käytetyistä sovelluskehitysympäristöistä.

### 2.1 Bluetooth

Bluetooth on langattomaan tiedonsiirtoon kehitetty tekniikka. Bluetooth-tekniikan kehittäjä on Lundin yliopiston Sven Mattisson, joka vuonna 1995 Ericssonille siirtyessään sai tehtäväksi tutkia lyhyen kantaman radioyhteyksiä. Alkuperäisenä ideana oli saada tätä tekniikkaa hyödyntävät puhelimet kommunikoimaan keskenään. Yhteistyö Intelin kanssa alkoi vuonna 1997 jolloin todettiin, että tekniikkaa pystyisi laajentamaan myös muihin langatonta tiedonsiirtoa tarvitseviin laitteisiin. Kehitystyöhön kutsuttiin mukaan myös muita teknologiayrityksiä ja vuonna 1998 tekniikka julkaistiin nimellä Bluetooth. [1]

Bluetooth-tekniikka toimii UHF-aalloilla 2,4 – 2,485 GHz:n alueella. Tekniikka perustuu taajuushyppelyyn, jossa laitteen Bluetooth-piiri jakaa datan paketteihin, ja lähettää jokaisen paketin yhdellä Bluetoothille varatuista kanavista. Bluetooth-tiedonsiirron mahdollistamiseksi kommunikoivat laitteet tulee parittaa keskenään. [2]

Bluetooth versio 5 on tällä hetkellä uusin Bluetooth-versio. Se julkaistiin kesäkuussa 2016 ja siinä on uusia ominaisuuksia jotka on kehitetty nimenomaan IoT-tekniikkaa silmälläpitäen. [2]

Bluetooth-yhteyden kantama on tyypillisesti 10 – 100 m. [2]

## 2.2 LoRaWAN

LoRaWAN on LPWAN-spesifikaatioon perustuva langaton verkko, joka on tarkoitettu matalatehoisille paristokäyttöisille laitteille. LoRaWAN on ensisijaisesti suunnattu esineiden internetin tarpeisiin. LoRaWAN on suunniteltu pienellä datamäärällä liikennöiville laitteille, datasiirron kapasiteetin ollessa 0,3 kb/s – 50 kb/s. Arkkitehtuurillisesti LoRa-laitteet liikennöivät keskenään oletusyhdyskäytävän (gateway) kautta. Oletusyhdyskäytävät on kytketty verkkopalvelimiin IP-standardin mukaisesti ja päätelaitteet keskustelevat keskenään langattomasti oletusyhdyskäytävien välityksellä. [3, 4]

LoRaWAN toimii Euroopassa 868 MHz:n ja Pohjois-Amerikassa 915 MHz:n taajuualueella. Uusi LoRaWAN spesifikaatio versio 1.1 on kehitteillä ja on tulossa käyttöön näillä näkymin kuluvan vuoden puolen välin paikkeilla. LoRa tulee sanoista 'Long Range' joka tarkoittaa pitkää kantamaa. [3, 4]

Tässä opinnäytetyössä ei käytetä LoRaWAN-laitteita, mutta sovellusesimerkissä 2 esitellystä Cayenne sovelluskehitysympäristöstä löytyy jo tuki LoRaWAN-laitteille. Laitetu-ki on vielä betatestaus-vaiheessa.

## 2.3 Ethernet

Esineiden internetiin kytketyt laitteet voivat toimia myös ilman langatonta yhteyttä, jolloin ne on kytketty suoraan internetiin lähiverkon kautta langallisesti. Tyypillisesti laite on kytketty lähiverkkoreitittimeen joko suoraan tai kytkimen kautta mikäli internetiin liitettäviä laitteita on useita. Internet-operaattorin verkon kautta laite pääsee internetiin kuten aikaisempien esimerkkien kohdalla. Tyypillinen toteutus on sellainen, missä laite itsessään, esimerkiksi mikrokontrolleri, on kytketty internetiin lähiverkon kautta. Kannettavat laitteet, esimerkiksi tietokoneet ja älypuhelimet, kommunikoivat sitten langattomasti mikrokontrollerin kanssa. [5, 6]

## 2.4 Esineiden internet

Esineiden internet (Internet of Things) koostuu erilaisista internetiin kytketyistä laitteista. Tyypillisiä esimerkkejä ovat kodinkoneet, ympäristö-olosuhteiden mittausta suoritta-



vat laitteet, lääketieteelliset laitteet potilaan kotona ja esimerkiksi etäohjattavat liikenteen opastaulut ja valvontakamerat. [6]

Esineiden internet on tarinan mukaan saanut alkunsa jo vuonna 1982. Carnegie Mellon -yliopiston ohjelmoijat liittivät tällöin yliopistossa sijainneen Coca Cola -automaatin yliopiston verkkoon ja tekivät palvelimelle ohjelman joka kertoi kuinka kauan kukin pullo oli ollut automaatissa. Tämän perustella voitiin päätellä oliko automaatissa ollut juotava jo riittävän kylmää. [7]

Esineiden internetin konsepti yleistyi vuonna 1999 RFID-tekniikan myötä. RFID-tekniikkaa käytetään muun muassa ihmisten ja ajoneuvojen tunnistamiseen. Inventointitarkoituksiin alettiin käyttää esineiden 'täggäystä' (engl. tagging) esimerkiksi viivakoodien ja QR -koodien muodossa. [6]

Gartnerin mukaan vuoteen 2020 mennessä esineiden internetiin liitettyjä laitteita olisi 20,8 miljardia. Toisen tutkimuksen, ABI Researchin, mukaan laitteita olisi vuoden 2020 loppuun mennessä jo 30 miljardia. [8, 9]

## 3 KEHITYSTYÖKALUT

### 3.1 App Inventor-sovelluskehitysympäristö

MIT –yliopiston App Inventor sovelluskehitysympäristö on avoimeen lähdekoodiin perustuva Android-sovellusten ohjelmointityökalu. App Inventor on lähtöisin Googlen Mark Friedmanin ja MIT:n professori Hal Abelsonin ideasta kehittää yksinkertainen ja helposti omaksuttava ohjelmointityökalu Android-sovellusten kehittämiseen. App Inventor toimii www-palveluna jota käytetään internet-selaimen kautta. Kehitystyökalun avulla on mahdollista toteuttaa mobiilisovelluksia ilman varsinaista ohjelmointiosaamista. Kehitysympäristön käyttö perustuu valmiiden ohjelmalohkojen sijoitteluun vedä-ja-pudota menetelmällä haluttujen toimintojen aikaansaamiseksi. App Inventorin avulla saadaan toteutettua mobiilisovelluksia moneen eri tarpeeseen. Näitä voivat olla muun muassa erilaiset testi- ja prototyypisovellukset tai räätälöidyt sovellukset tiettyyn tarpeeseen. App Inventor on ilmainen ja vapaasti kaikkien käytettävissä. Tässä työssä App Inventorilla luodaan älypuhelinsovellus, jonka avulla luetaan langattomasti Bluetooth-yhteyden välityksellä anturien arvoja. [10]

### 3.2 Cayenne-sovellus- ja ohjelmointialusta

Cayenne on myDevices –yhtiön kehittämä sovellus- ja ohjelmointialusta erilaisille IoT-ratkaisuille. Cayennen toiminta perustuu niin sanottuun kojelautaan (dashboard) jonka kautta hallitaan esimerkiksi mikrokontrolleriin kytkettyjä antureita ja oheislaitteita. Cayennen kojelauta voi sisältää esimerkiksi mittareita ja kytkimiä antureiden arvojen näyttämistä ja oheislaitteiden etäohjausta varten. Kojelauta on käyttäjän muokattavissa. Käytön rajoituksena on se, että hallittavien antureiden ja oheislaitteiden on löydettävä Cayennen omista tietokannoista. Oheislaitetietokanta on kuitenkin jo nyt melko laaja ja sinne lisätään uusia laitteita jatkuvasti. [11]

Kuten App Inventor, Cayennekin toimii www-palveluna ja käyttö tapahtuu internet-selaimella. Cayenne mahdollistaa myös oman ohjelmointikoodin lisäämisen lisäominaisuuksien saavuttamiseksi. [12]

Tässä työssä Cayennea käytetään suoraan internetiin liitetyn mikrokontrollerin ja siihen kytkettyjen antureiden arvojen lukemiseen Cayennen kojelaudan (dashboard) kautta. Kojelautasovellus toimii sekä internet-selaimella että älypuhelimessa.

Cayenne on myös ilmainen.

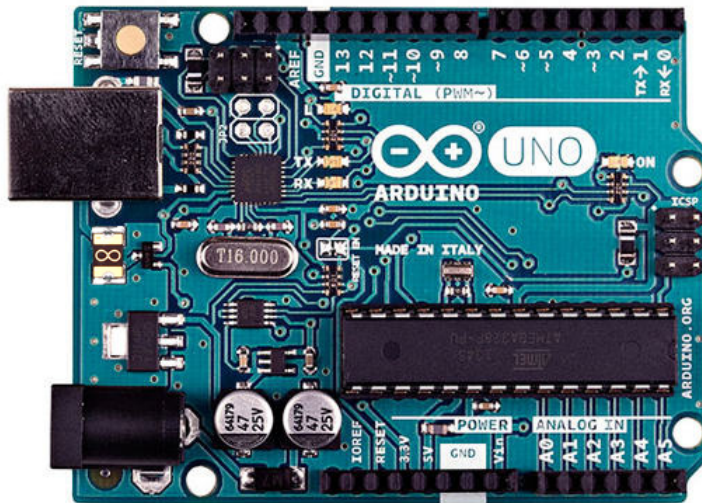
### 3.3 Arduino

Arduino on avoimeen lähdekoodiin perustuva kehitysalusta ja mikrokontrolleri. Arduinossa on liittimiä sekä analogisten että digitaalisten oheislaitteiden kytkemistä varten. Arduinon omassa, Arduino IDE, kehitysympäristössä voidaan Arduinon keskusyksikköä ohjelmoida haluttujen toimintojen aikaansaamiseksi. Arduinon ohjelmointikielen perustana on C-kieli jota on laajennettu Arduinon omilla funktiokirjastoilla. Arduino – tuoteperheeseen kuuluu lukuisia eri tuotteita käyttötarpeen mukaan. [13]

Arduino -mikrokontrolleria käytetään tässä työssä antureiden tietojen luentaan, tietojen muokkaukseen sekä tietojen välittämiseen eteenpäin tietokoneelle.

#### 3.3.1 Arduino Uno

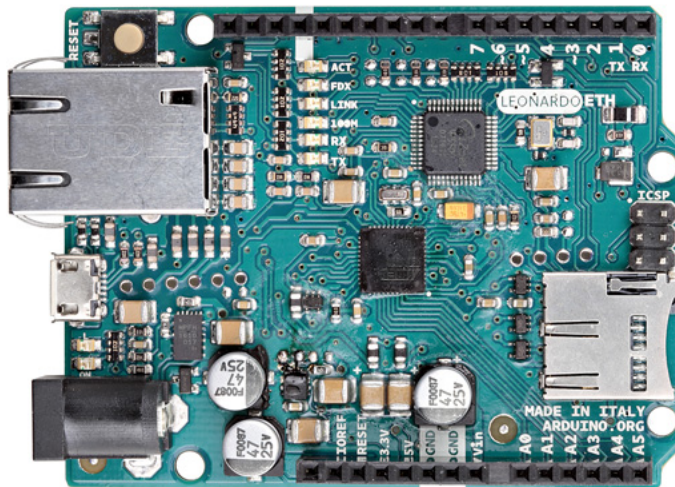
Arduino Uno on ATmega328P 16 MHz:n prosessoriin perustuva mikrokontrolleri (kuva 1). Siinä on 6 analogista sisääntuloporttia ja 14 digitaalista I/O –porttia, joista kuutta voidaan käyttää PWM -ulosmena. Arduino Uno on suunnattu aloittelijoiden käyttöön, mutta on useimmissa tapauksissa kuitenkin varsin riittävä mikäli siihen kytkettävien oheislaitteiden ja anturien lukumäärä ei ole kovin suuri. Myös sisäisen muistin vähäinen määrä asettaa rajoituksia ohjelmien pituudelle. Sisäistä muistia Arduino Unossa on 32 kt. [14]



Kuva 1. Arduino Uno. [15]

### 3.3.2 Arduino Leonardo ETH

Arduino Leonardo ja sen ETH –malli (kuva 2) on suunniteltu erityisesti IoT-käyttöön. Mikrokontrollerissa on mallimerkinnän mukaisesti Ethernet-liitäntä valmiina mikä esimerkiksi Arduino Uno on saatavilla vain laajennuskorttina. Arduino Leonardo ETH perustuu 16 MHz:n ATmega32u4 prosessoriin. Siinä on 20 digitaalista I/O –liitäntää, joista seitsemää voidaan käyttää PWM –ulosmena ja 12:ta analogisena sisääntulona. Myös Arduino Leonardo ETH:ssa on sisäistä muistia 32 kt:a. [16]



Kuva 2. Arduino Leonardo ETH. [17]

### 3.3.3 Arduino IDE

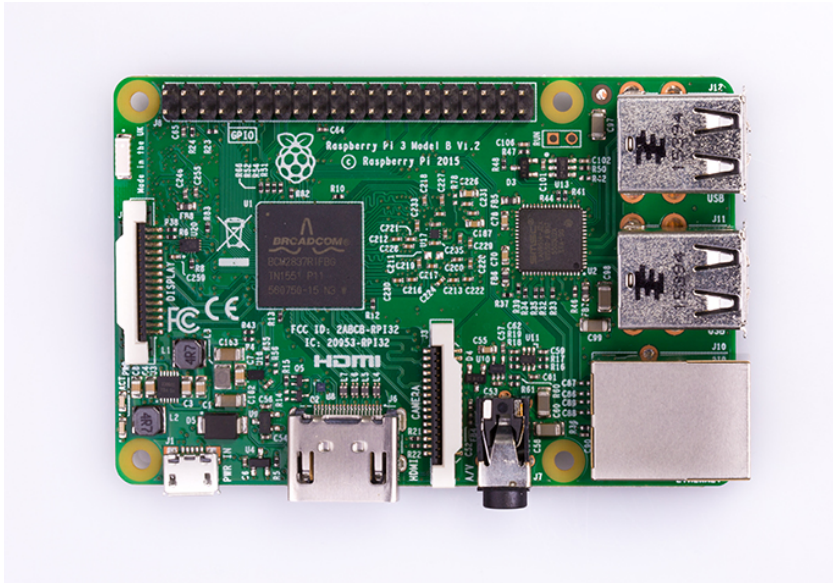
Avoimeen lähdekoodiin perustuva kehitysympäristö, Arduino IDE, on kaikkien Arduino -tuotteiden ohjelmointiin kehitetty sovellus. Kehitysympäristössä kirjoitetaan ohjelmakoodi jonka mikrokontrollerin halutaan suorittavan. Valmis koodi ladataan sitten mikrokontrollerille suorittamista varten. Vaikka Arduinolle voidaan kirjoittaa koodia millä tahansa ohjelmointikielillä joka pystytään kääntämään Arduinon prosessorin ymmärtämään konekielimuotoon, tapahtuu ohjelmointi Arduinon omassa kehitysympäristössä C- tai C++ -kielillä. Arduino IDE sisältää lisäksi kirjastofunktioita jotka tarjoavat runsaasti I/O-toimintoja erilaisten Arduinon kytkettävien komponenttien ja oheislaitteiden ohjelmointia ja hallintaa varten. [18]

### 3.4 Raspberry Pi

Raspberry Pi on tehokas, yhdelle piirilevyllä rakennettu, vain kämmenen kokoinen tietokone moneen eri käyttötarkoitukseen. Raspberry Piä on julkaistu jo useita eri versioita. Ensimmäinen versio, Raspberry Pi 1 Model B, julkaistiin helmikuussa 2012. Seuraava merkittävä versio, Raspberry Pi 2, esiteltiin helmikuussa vuonna 2015. Tällä hetkellä uusin versio Raspberry Piä on Raspberry Pi 3 Model B (kuva 3) joka julkaistiin helmikuussa 2016. Raspberry Pi 3:ssa on 1,2 Ghz:n neliytiminen 64-bittinen prosessor-

ri. Keskusmuistia siinä on 1 Gt:a. Laitteessa on myös WiFi- ja Bluetooth-yhteysmahdollisuus. Liittimiä on runsaasti, muun muassa 4 USB –porttia, HDMI –portti, Ethernet –liitäntä ja 40 GPIO –pinniä, MicroSD –muistikorttipaikka sekä audio tai kuu-  
lokelitääntä. [19, 20, 21]

Raspberry Pi toimii tässä työssä Linux-palvelimena.



Kuva 3. Raspberry Pi 3 Model B. [20]

### 3.5 Android Studio

Android Studio on Googlen kehittämä virallinen Android-älylaitteiden ohjelmistokehitykseen tarkoitettu kehitysympäristö. Vuoden 2016 toukokuusta lähtien Android Studio on ollut täysin ilmainen Apache License 2.0 alaisuudessa. Ohjelmien kirjoittaminen Android Studiossa tapahtuu Java-ohjelmointikielellä. Kehitysympäristön käyttö ja täysi hyödyntäminen vaatii siis ohjelmointiosaamista Java-ohjelmointikielellä vaikka kehitysympäristö tarjoaa myös ohjattuja toimintoja esimerkiksi graafisten käyttöliittymien rakentamiseen. [22, 23]

### 3.6 Xamarin for Visual Studio

Xamarin on nykyisin Microsoftin omistuksessa oleva ohjelmistoyhtiö. Alun perin Xamarin Studio-nimisenä julkaistu kehitysympäristö on tarkoitettu mobiililaitteiden ohjelmisto-

kehitykseen. Nykyään Xamarin Studio on integroitu osaksi Microsoftin Visual Studio kehitysympäristöä ja nimi on muuttunut Xamarin for Visual Studioksi. Ohjelmointi tapahtuu C#-ohjelmointikielellä ja myös Xamarin tarjoaa monia työkaluja graafisten käyttöliittymien rakentamiseen. Vaikka Xamarin on ilmainen lisäosa Microsoft Visual Studioon, on Microsoft Visual Studio maksullinen kehitysympäristö. Microsoft Visual Studios-ta on saatavilla myös ilmainen, ominaisuuksiltaan rajoitettu versio, johon Xamarin for Visual Studion voi asentaa. Kuten Android Studion, myös Xamarin for Visual Studion käyttö vaatii ohjelmointiosaamista ja –kokemusta älypuhelinlustralle eivätkä tästä syystä valikoituneet käyttöön tässä työssä. [24]

### 3.7 Komponentit ja anturit

Sovellusesimerkkien kytkennöissä käytetään erilaisia komponentteja ja antureita toiminnan havainnollistamiseksi. Kytkennät itsessään ovat yksinkertaisia, koska työn tarkoituksena on esitellä menetelmiä ja ohjelmistoja, joilla saadaan rakennettua antureiden arvojen tulkitsemiseen tarvittavat käyttöliittymät älypuhelimelle ja tietokoneelle.

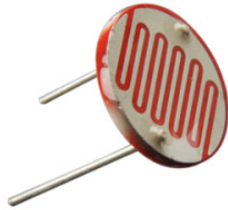
Langattomaan tiedonsiirtoon älylaitteen ja mikrokontrollerin välille valittiin Velleman HC-05 Bluetooth-moduuli (kuva 4). Bluetooth-moduulista on käytössä RX, TX, GND ja VCC –pinnit. RX ja TX ovat datan vastaanottoa ja lähetystä varten. GND on maadoituspinni ja VCC on moduulin käyttöjännitettä varten. HC-05 toimii 5 V käyttöjännitteellä joka saadaan suoraan Arduino -mikrokontrollerista. Arduino näkee HC-05 Bluetooth-moduulin sarjaporttina, joten tiedonsiirto moduulin välityksellä tapahtuu Arduinon sarjaportin ohjauskomentoja käyttämällä. [25]



Kuva 4. Velleman HC-05 Bluetooth-moduuli ja pinnijärjestys. [25]

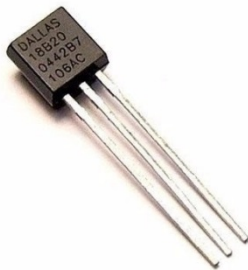
Sovellusesimerkissä 1 käytetään anturina LDR- eli valovastusta (kuva 5). Valovastus reagoi valoon siten, että valon voimakkuuden kasvaessa vastuksen resistanssi piene-

nee. Valovastukset ovat käyttökelpoisia muun muassa ohjattaessa valaistusta tai haluttaessa muusta syystä monitoroida ympäristön valaistusolosuhteita.



Kuva 5. Valovastus. [26]

Sovellusesimerkin 1 jälkimmäiseen kytkentään lisätään mukaan lämpötila-anturi. Anturi on Maxim Integratedin valmistama digitaalinen anturi mallia DS18B20 (kuva 6). Anturin mittaustarkkuus on  $\pm 0,5$  °C lämpötilojen -10 °C ja +85 °C välillä ja toimintarajat -55 °C – +125 °C. Anturin toimintajännite on 3,3 – 5 V. Kytkennässä anturi tarvitsee suojaksi 4,7 k $\Omega$ :n vastuksen. [27, 28]



Kuva 6. Maxim Integrated DS18B20 lämpötila-anturi. [29]

### 3.8 VMware workstation

Sovellusesimerkissä 3 käytetään virtuaalikoneeksi asennettua Debian Linux 8.7.1-käyttöjärjestelmää toimintojen ja ohjelmakoodien testaamiseen ennen järjestelmän siirtoa Raspberry Pi -alustalle. Tähän tarkoitukseen käytettiin VMware workstation virtualisointiohjelmaa, jolla voidaan virtualisoida lähes kaikki saatavilla olevat käyttöjärjestelmät. Nimensä mukaisesti VMware workstation on tarkoitettu lähinnä käyttöjärjestelmien työasemaversioiden virtualisointiin ja on maksullinen ohjelma. Virtualisointiohjelmaa löytyy myös maksuttomina versioina. Esimerkiksi Oraclen omistama VirtualBox on



hyväksi todettu virtualisointiohjelma. Virtuaalikone voidaan myös tallentaa sellaisenaan toimivaksi käyttöjärjestelmäksi. VMwarelta löytyy ilmainen VMware Player, jolla tallennettu virtuaalikone voidaan avata ja käyttää aivan kuten tietokoneen kiintolevyllä asennettua käyttöjärjestelmää. Tämä on hyvin kätevä ominaisuus esimerkiksi haluttaessa kokeilla eri käyttöjärjestelmiä. Internetissä toimiva [www.osboxes.org](http://www.osboxes.org) tarjoaa valmiiksi asennettuja virtuaalikoneita joita voi ajaa esimerkiksi edellä mainitulla VMware Player-ohjelmalla. [30, 31, 32]

## 4 SOVELLUKSET

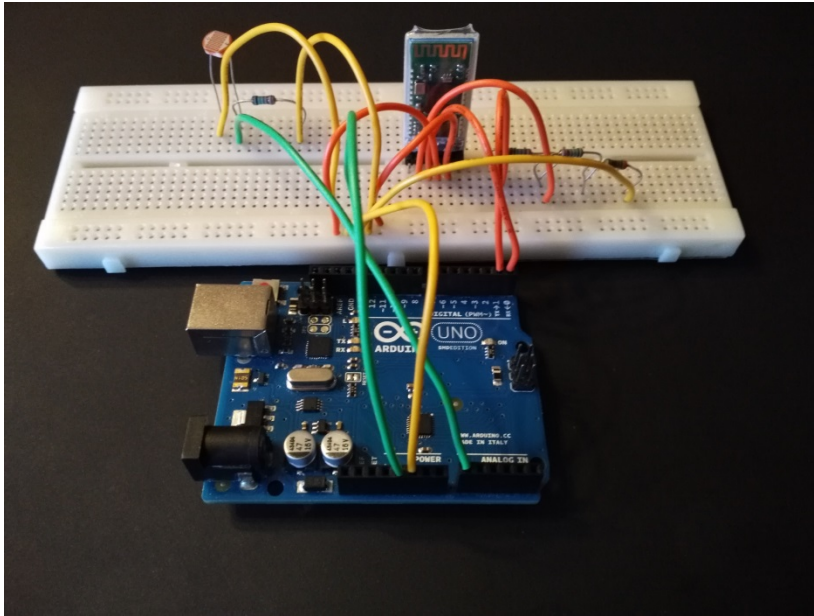
Tässä luvussa esitetään kolme sovellusesimerkkiä. Ensimmäinen ja toinen esimerkki perustuvat pääosin kappaleessa kaksi esitettyjen internet-pohjaisten kehitystyökalujen käyttöön. Molemmissa esimerkeissä tehdään Android-alustalle käyttöliittymä, josta voidaan reaaliaikaisesti seurata mikrokontrolleriin kytketyn anturin arvoja. Jälkimmäisessä esimerkissä tehdään myös internet-selaimella toimiva käyttöliittymä anturin arvojen näyttämiseksi. Käytetty kytkentä on melko yksinkertainen, koska ajatuksena on esittää kuinka mobiilialustaa voidaan käyttää tietojen reaaliaikaiseen esittämiseen ja toisaalta, kuinka selaimella toimivassa versiossa saadaan tietoa esitettyä graafisessa muodossa esimerkiksi mittareiden avulla.

Kolmannessa esimerkissä rakennetaan Linux -alustalla toimiva palvelinjärjestelmä, jossa datan esittämisen lisäksi perehdytään myös tiedon tallentamiseen tietokantaan, hälytysten lähettämiseen sähköpostilla ennalta määrättyjen sääntöjen pohjalta, internet-pohjaisen käyttöliittymän rakentamiseen sekä tarvittavien ohjelmien kirjoittamiseen eri ohjelmointikielillä.

Tässä luvussa sovellusesimerkit esitetään itsenäisinä kokonaisuuksina, mutta ajatus on, että esimerkkien mukaiset sovellukset voi myös yhdistää. Näin ollen on mahdollista rakentaa järjestelmä, jossa kaikki kolme esimerkkiä toimivat yhdessä ja muodostavat yhtenäisen kokonaisuuden.

### 4.1 Kytkennät

Ensimmäisessä ja toisessa sovellusesimerkissä käytetään kytkentää, jossa luetaan valovastuksen (LDR) arvoa erilaisissa valaistusolosuhteissa. Kytkennässä on mukana myös Bluetooth-moduuli, jonka avulla tietoa siirretään langattomasti Bluetooth-yhteyden välityksellä älypuhelimeen. Kytkentä on tehty koekytkentäalustalle ja mikrokontrollerina käytetään Arduino Unoa (kuva 7). Liitteessä 5 on esitetty samassa kytkentäkaaviossa molemmat sovellusesimerkissä 1 käytetyt kytkennät.



Kuva 7. Sovellusesimerkissä 1 käytetty kytkentä.

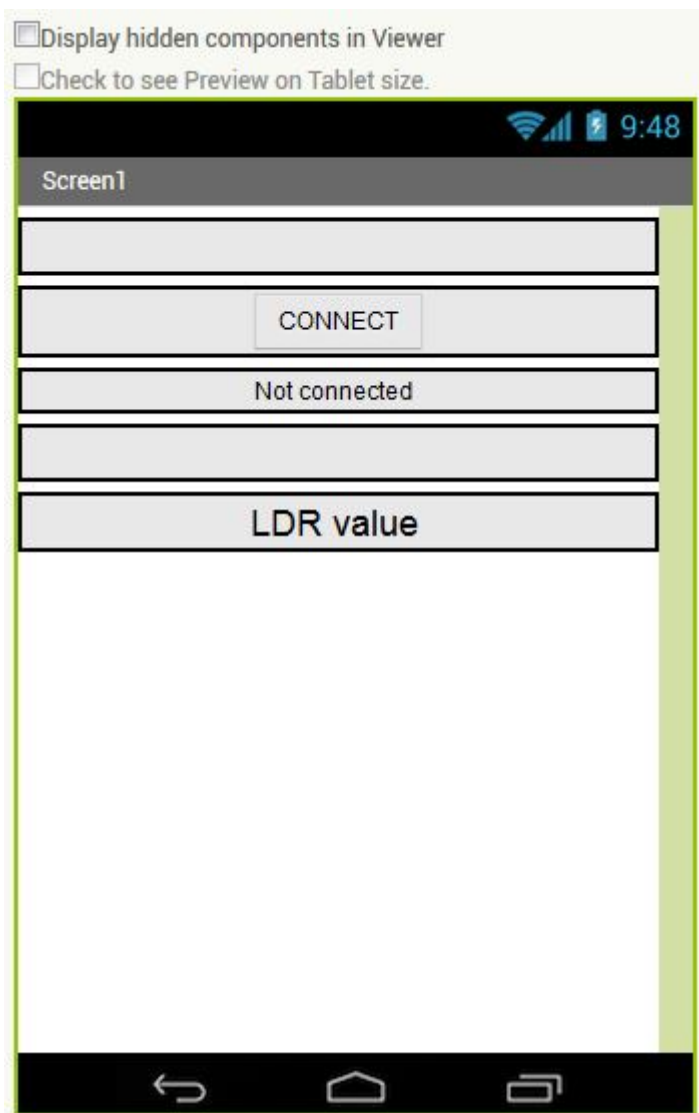
#### 4.2 Sovellusesimerkki 1

Sovellusesimerkissä 1 luetaan kuvan 7 mukaisesta kytkennästä LDR -vastuksen arvo yhden sekunnin välein ja siirretään tieto langattomasti Bluetooth-yhteydellä älypuheliin. LDR -vastuksen arvo päivittyy älypuhelimien näytölle reaaliajassa.

Koodiesimerkki 1. LDR -vastuksen arvon lukeminen Arduino IDE:ssä.

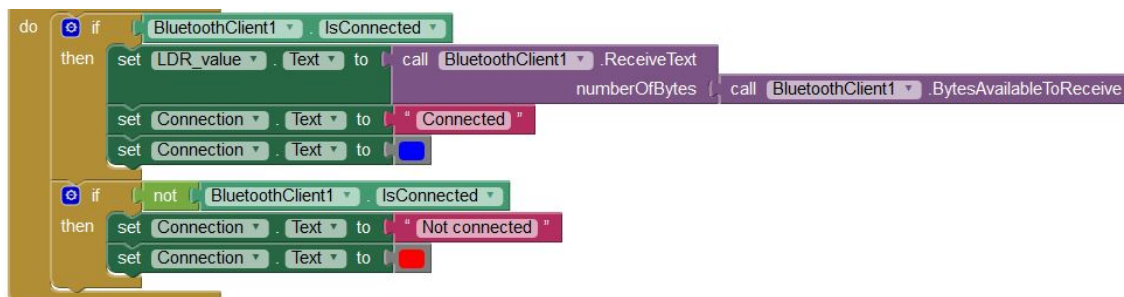
```
int LDR_value = analogRead(LDR_pin);
Serial.println(LDR_value);
delay(1000);
```

Koodiesimerkissä 1 luetaan LDR -vastuksen kulloinkin arvo muuttujaan LDR\_value joka heti tämän jälkeen lähetetään Bluetooth-moduulille Serial.println -komennolla. Jotta älypuhelimella voidaan vastaanottaa Bluetooth-yhteydellä saapuva tieto, täytyy älypuhelin ensin parittaa Arduino -mikrokontrolleriin kytketyn Bluetooth-moduulin kanssa. Älypuhelimien paritukseen ja datan esittämiseen tarvittavat toiminnot toteutetaan molemmat App Inventor kehitysympäristössä (kuva 8).



Kuva 8. Älypuhelimien näytön suunnittelunäkymä.

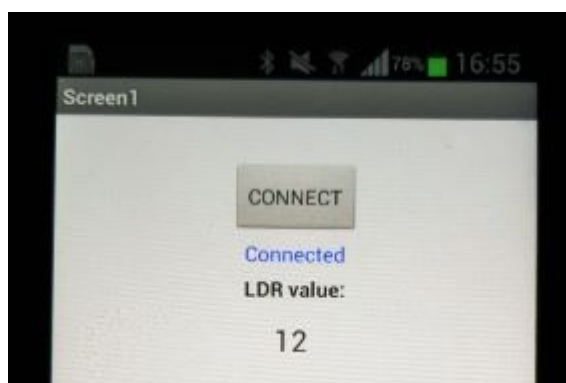
Kuvassa 8 on lisätty App Inventorin suunnittelunäkymässä 'CONNECT' -painike Bluetooth-laitteiden paritusta varten. LDR -vastuksen arvon näyttämistä varten on lisätty tekstikenttä 'LDR value'. Varsinaiset toiminnot määritetään App Inventorin lohkonäkymässä. Lohkonäkymän käyttö perustuu toimintojen ohjelmoimiseen raahaamalla valmiita ohjelmalohkoja ja ohjelmien ohjausrakenteita haluttuun järjestykseen lohkonäkymässä (kuva 9). Lohkorakenteiden käyttö edellyttää jonkin verran ymmärrystä ohjelmoinnista jotta halutut toiminnot saadaan toteutettua.



Kuva 9. Älypuhelinsovellus lohkorakenteena.

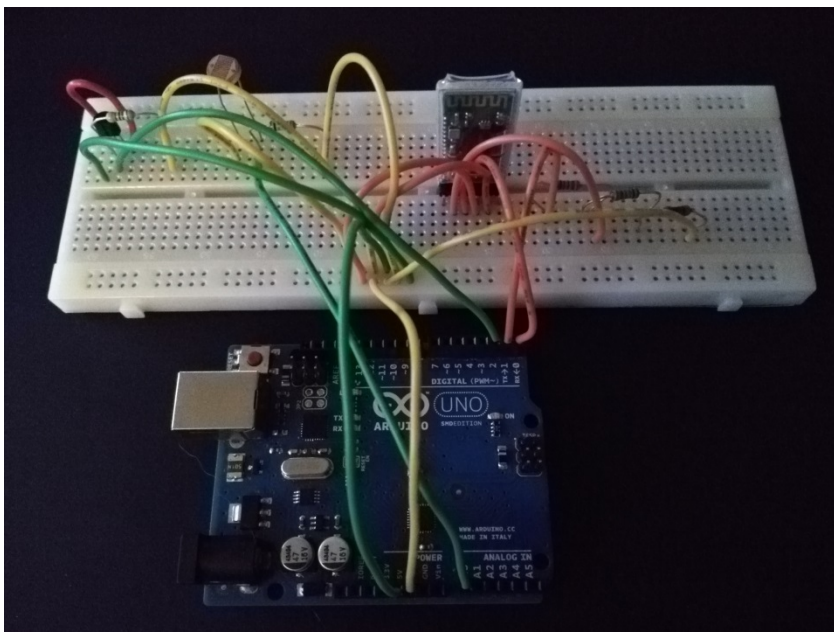
Kuvan 9 mukaisessa ohjelmalohkossa suoritetaan jatkuvaa do –silmukkarakennetta. Mikäli laitteet ovat paritettuna ja Bluetooth-yhteys aktiivinen, näytetään LDR -vastuksen arvo reaaliaikaisena älypuhelimien näytöllä ja teksti 'Connected' näkyy sinisellä värillä. Muussa tapauksessa älypuhelimien näytöllä lukee punaisella 'Not connected'.

Lopuksi App Inventorilla luotu sovellus (kuva 10) pitää kääntää älypuhelimien Android-käyttöjärjestelmää varten. Tämä tapahtuu 'Build' –valikon kohdassa 'App (save .apk to my computer)'. Android-alustalla toimiva .apk -sovellus pitää lopuksi kopioida ja asentaa puhelimeen.



Kuva 10. Valmis älypuhelinsovellus.

Samaan tapaan kuin tässä esimerkissä yhden anturin osalta, voidaan mikrokontrolleriin kytkeä useampi anturi ja vastaavasti esittää useamman anturin arvot samalla kertaa älypuhelimien näytöllä. Kuvan 7 mukaiseen kytkentään lisätään nyt lämpötila-anturi ja rakennetaan uusi sovellus App Inventorilla, jolla saadaan luettua samanaikaisesti kahden anturin arvoja (kuva 11).



Kuva 11. KytKentä, johon on lisätty lämpötila-anturi.

Lämpötila-anturin lisääminen kytkentään vaatii muutoksia Arduino -mikrokontrollerin ohjelmakoodiin. Kytkenässä käytetty lämpötila-anturi vaatii kirjastojen OneWire ja DallasTemperature sisällyttämisen Arduinon kirjastoluetteloon. Kyseiset kirjastot sisältävät funktioita lämpötila-anturin ohjelmoimiseen sekä anturin arvojen tulostamiseen. [33, 34]

Koodiesimerkki 2. LDR-vastuksen ja lämpötila-anturin arvojen lukeminen Arduino IDE:ssä.

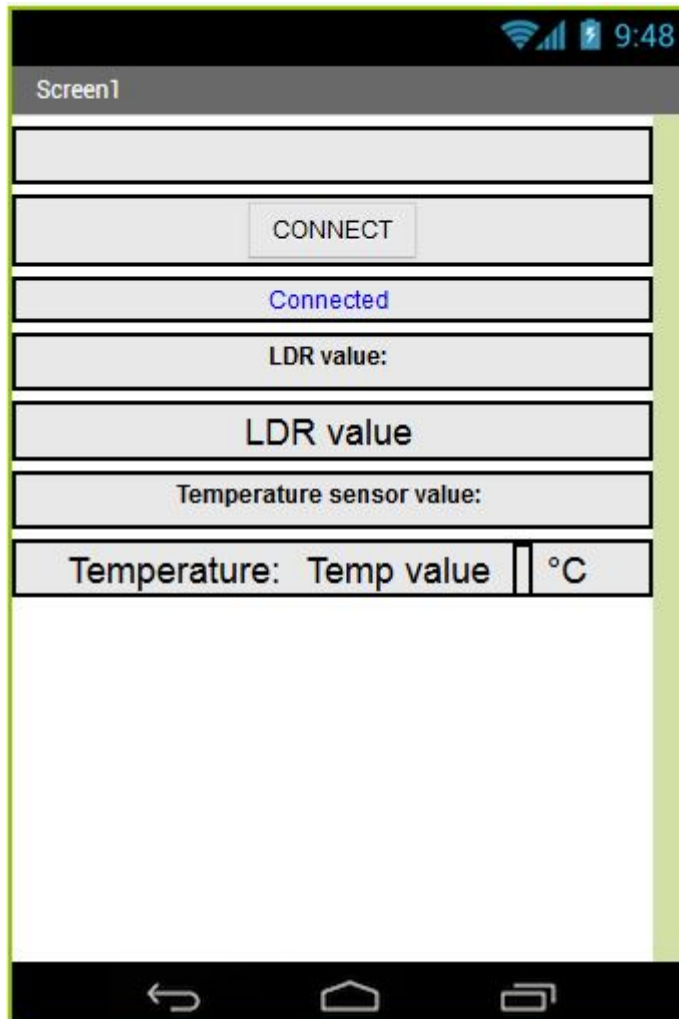
```
int LDR_value = analogRead(LDR_pin);
sensors.requestTemperatures();

Serial.print(LDR_value);
Serial.print(",");
Serial.println(sensors.getTempCByIndex(0));
```

Koodiesimerkissä 2 luetaan LDR -vastuksen arvo muuttujaan LDR\_value sekä kutsutaan requestTemperatures -funktioita, jolla saadaan lämpötila-anturin kulloinenkin arvo tulostettua Arduinon sarjaporttiin. Funktiolla getTempCByIndex saadaan lämpötila tulostettua suoraan Celsius-asteina. Koska nyt luetaan arvoja kahdelta eri anturilta, jotka halutaan samanaikaisesti lähettää älypuhelinsovellukselle, joudutaan käyttämään erotinmerkkiä arvojen välissä. Erotinmerkin avulla älypuhelinsovellus osaa pilkkoa sille

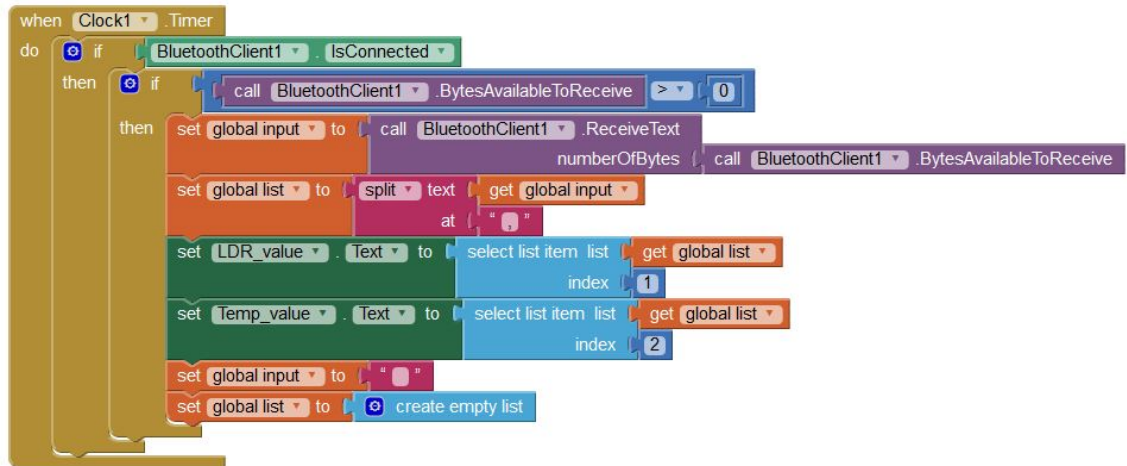
lähetetystä merkkijonosta LDR -vastuksen ja lämpötila-anturin arvot omiin tekstikenttiinsä.

App Inventor –suunnittelunäkymässä lisätään lämpötila-anturin arvolle uusi tekstikenttä 'Temp value' (kuva 12).

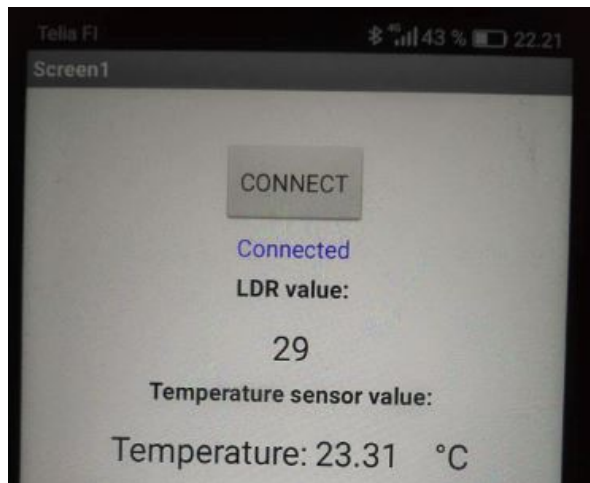


Kuva 12. App Inventorin suunnittelunäkymä kahden anturin lukemista varten.

App Inventorin lohkonäkymässä toteutetaan Arduinolta lähetyn, anturiarvoja sisältävän, merkkijonon pilkkominen. Lohkonäkymässä toteutetaan ensin yhteystarkistus. Mikäli Bluetooth-yhteys on avattu ja Arduinolta saapuu anturiarvoja siirrytään ohjelmalohkossa osioon, jossa saapuva merkkijono pilkotaan erotinmerkin kohdalta ja vastaavat merkkijonon puolikkaat tulostetaan omiin tekstikenttiinsä älypuhelimien näytöllä (kuvat 13 ja 14).



Kuva 13. Lohkorakenne kahden anturin lukemista varten.



Kuva 14. Kahden anturin reaaliaikaiset arvot älypuhelimien näytöllä.

App Inventorilla saa nopeasti rakennettua sovelluksia älypuhelimista varten. Kehitysympäristön käyttö ja sovelluksien luominen vaatii kuitenkin vähintään ohjelmoinnin perusasioiden ymmärtämistä. Sovelluksien rakentaminen lohkonäkyssä vastaa ohjelmien kirjoittamista perinteisellä tavalla alusta alkaen, joten ohjelmoinnin perus- ja ohjausrakenteet pitää vähintäänkin tuntea. App Inventorin suurimmat hyödyt tulevat esille sovellusten rakentamisen nopeudessa sekä kirjoitusvirheiden välttämiseksi, mikä usein on ongelma ohjelmia perinteisellä tavalla kirjoitettaessa. App Inventorilla on kattavat ohjeet ja aktiivinen käyttäjäfoorumi josta löytyy apua useimpiin ongelmatilanteisiin. [35]



### 4.3 Sovellusesimerkki 2

Toisessa sovellusesimerkissä käytetään kuvan 11 mukaista kytkentää, josta luetaan LDR-vastuksen sekä lämpötila-anturin arvoja Cayenne -sovellusalustan avulla. Älypuhelimeen asennetaan Cayenne -mobiilisovellus, jonka avulla antureiden arvot saadaan näkyviin myös puhelimesta. Cayennen etuna on, että erillistä mobiilialustaohjelmointia ei tarvitse tehdä lainkaan. Kun tarvittavat anturit on määritetty sovellusalustassa ja lisätty sen kojelautanäkymään, on anturien arvot nähtävillä suoraan Cayenne-mobiilisovelluksessa ilman erillisiä toimenpiteitä. Cayenne -sovellusalustassa voidaan myös määrittää esimerkiksi liipaisut (trigger) suoraan kojelautanäkymässä. Esimerkkinä sähköposti- tai tekstiviestihälytys kun tietty anturin arvo ylittyy tai alittuu. Cayenne -mobiilisovellus löytyy Googlen Play -kaupasta nimellä Cayenne. [11]

Tässä sovellusesimerkissä otettiin käyttöön Arduino Leonardo ETH -mikrokontrolleri. Tämä mikrokontrolleri saadaan liitettyä suoraan verkkoon siinä olevan Ethernet-liitännän kautta. Kuvan 11 kytkennän anturit liitetään Arduino Leonardo ETH:ssa samoihin pinneihin kuin aikaisemmassa esimerkissä. LDR -vastus analogiseen pinniin A0 ja lämpötilasensori digitaaliseen pinniin 2. Arduino IDE:ssä on muistettava muuttaa ohjelmoitava mikrokontrolleri vastaamaan nyt käytössä olevaa, koska aikaisemman esimerkin jäljiltä käytössä on vielä Arduino Uno. Tämä tapahtuu Arduino IDE:n valikosta 'Tools' ja sen alavalikosta 'Boards'. Leonardo ETH -mikrokontrolleriin ladattava ohjelmakoodi poikkeaa myös edellisestä esimerkistä. Nyt pitää ottaa käyttöön funktiokirjastot mikrokontrollerin verkkokorttia sekä Cayenne -sovellusalustaa varten (kuva 15).

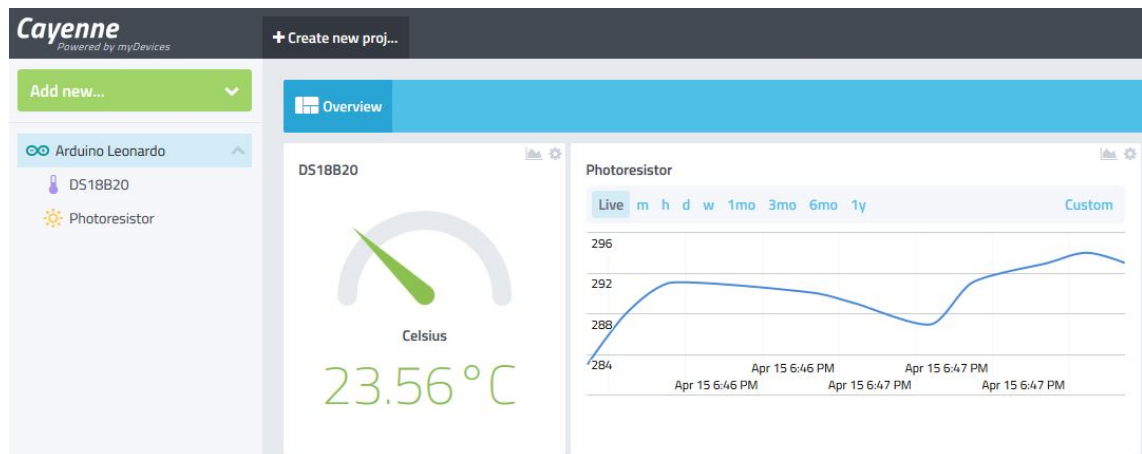
```
#include <CayenneEthernetW5500.h>
#include <Ethernet2.h>
#include <SPI.h>
```

Kuva 15. Cayennen tarvitsemat funktiokirjastot.

Kuvan 15 ensimmäisellä koodirivillä sisällytetään ohjelmakoodiin funktiokirjastot joilla Arduino Leonardo ETH yhdistetään Cayennen palvelimiin. W5500 on mikrokontrollerin verkko-ohjainpiiri. Toisella rivillä lisätään funktiokirjastot Leonardon verkkokortin ohjaukseen ja IP-osoitteen määrittämistä varten. Mikäli IP-osoitetta ei erikseen määritetä ja käytössä on DHCP-palvelin osaa Leonardo ETH hakea itselleen automaattisesti IP-osoitteen. Viimeisellä rivillä lisätään SPI.h -funktiokirjasto mikrokontrollerin sarjaliikenteen hallintaa varten. [11, 36]

Cayennen käyttöä varten tarvitaan käyttäjätili, jonka voi käydä luomassa ilmaiseksi osoitteessa: <https://cayenne.mydevices.com/cayenne/signup>

Tunnusten luonnin jälkeen kirjaututaan Cayenne -sovellusalustaan (kuva 16).



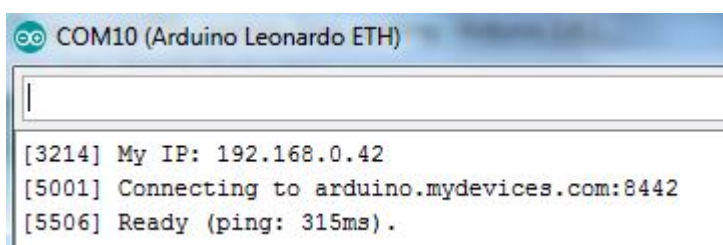
Kuva 16. Valmis kojelauta Cayenessa.

Kuvassa 16 on luotu Cayennen kojelautanäkymään mittari lämpötilasensorin arvojen sekä graafi LDR-vastuksen arvojen näyttämistä varten. Kojelaudan rakentaminen aloitetaan lisäämällä Cayenneen laite, johon anturit on kytketty. Laite voi olla Arduino -mikrokontrolleri, Raspberry Pi tai LoRa -laite. Tässä esimerkissä lisätään laitteeksi edellä mainittu Arduino Leonardo ETH. Laitteen lisääminen tapahtuu Cayennen aloitusnäkökulmasta kohdasta 'Add new...'. Avautuvan valikon 'Device/Widget' -kohdasta valitaan käytössä oleva Arduino -mikrokontrolleri ja tapa, jolla Arduino on yhdistetty tietokoneeseen tai verkkoon. Arduino Leonardo ETH on tässä tapauksessa kytketty suoraan verkkoon, joten listasta valitaan 'Ethernet Shield W5500' (kuva 17). Samaisesta ikkunasta otetaan talteen myös autentikointia varten tarvittava 'auth token', joka liitetään Arduinolle ladattavaan ohjelmakoodiin. Autentikointitunnus on yksilöllinen tunnus jokaista Cayenne -palveluun liitettävää laitetta varten.



Kuva 17. Arduino Leonardo ETH:n verkkokortin valinta.

'Sketch' –painikkeen takaa löytyy valmis pohja Arduinoon ladattavalle ohjelmakoodille jota pitää kuitenkin täydentää kaikkien tarvittavien ominaisuuksien saamiseksi käyttöön. Edellä mainitut #include –funktiokirjastot liitetään osaksi Arduinon ohjelmakoodia. Tämän lisäksi saatetaan tarvita muitakin lisäyksiä riippuen sovelluksesta ja halutuista ominaisuuksista. Tässä esimerkissä lisätään koodiesimerkki 2:sen mukaiset koodirivit anturitiedon sarjaporttiin kirjoittamista varten, jotta sovellusesimerkki 3:n mukainen Linux-järjestelmä saadaan samanaikaisesti käyttöön ja toimimaan tämän esimerkin mukaisen järjestelmän rinnalle. Kun ohjelmakoodi on ladattu Arduinoon, tarkastetaan Arduino IDE:n Serial Monitorista että yhteys Cayennen palvelimeen saadaan muodostettua. Kun yhteys toimii (kuva 18), voidaan siirtyä Cayennen kojelautanäkymään lisäämään halutut mittarit monitorointia varten.



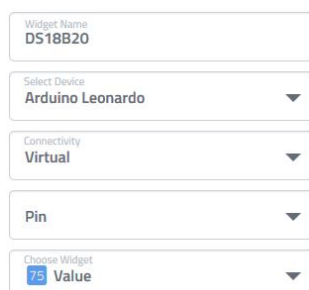
```

COM10 (Arduino Leonardo ETH)
[3214] My IP: 192.168.0.42
[5001] Connecting to arduino.mydevices.com:8442
[5506] Ready (ping: 315ms).

```

Kuva 18. Yhteystestaus Cayennen palvelimelle.

Kojelautanäkymässä lisätään ensin mittari lämpötila-anturin arvon näyttämistä varten. Mittarin lisääminen tapahtuu samaa kautta kuin edellä Leonardo ETH:n lisääminen laitteeksi Cayenneen. Cayennen anturikirjastosta löytyy valmiina kytkennässä käytössä oleva DS18B20 -lämpötila-anturi. Lämpötila-anturin lisääminen tehdään 'Sensors' –valikon kohdassa 'Temperature' josta valitaan haluttu anturi (kuva 19). Mikäli Cayennen kirjastosta ei löydy käytössä olevaa anturia, voidaan käyttää Cayennen 'Custom Widgets' –valikkoa jonka kautta useimpien kirjastosta puuttuvien anturien lisääminen onnistuu.



Widget Name  
DS18B20

Select Device  
Arduino Leonardo

Connectivity  
Virtual

Pin

Choose Widget  
Value

Kuva 19. DS18B20 -lämpötila-anturin asetuksia Cayenessa.

Kuvassa 19 määritetään asetukset lämpötila-anturille. Tärkeimmät ovat 'Connectivity' ja 'Pin' –asetukset. DS18B20 kytketään virtuaalisen pinnin avulla Cayenneen. Kohtaan 'Pin' valitaan arvo joka täsmää Arduinon ohjelmakoodissa määritetyn globaalin muuttujan VIRTUAL\_PIN kanssa. Tarvittavien asetusten määrittämisen jälkeen ladataan 'Sketch File' –painikkeen takaa löytyvä ohjelmakoodi Arduinoon. Asetuksissa määritetään lisäksi, että lämpötila-anturin arvo halutaan näkyvän mittarina Cayennen kojelautanäkymässä. Kohtaan 'Choose Widget' valitaan 'Gauge'. Painikkeella 'Add Sensor' lisätään luotu mittari kojelautanäkymään.

LDR -vastuksen arvon näyttämistä varten lisätään kojelautaan graafi, 'Graph'. Itse anturin lisääminen tapahtuu samalla tavalla kuin edellä lämpötila-anturin kohdalla. Nyt valitaan 'Sensors' –valikosta 'Photoresistor', valovastus. LDR-vastus on kytketty Arduinossa analogiseen A0 –pinniin, joten 'Connectivity' –kohtaan valitaan 'Analog' ja kohtaan 'Pin' valitaan arvo 'A0'. Sketch –tiedostoa ei tarvitse tässä tapauksessa ladata Arduinoon, vaan tarvittavat muutokset Arduinon ohjelmakoodiin tehdään käsin. Ensin määritetään muuttuja LDR\_pin, joka kertoo mihin pinniin LDR -vastus on Arduinossa kytketty.

```
int LDR_pin = A0;
```

Arduinon void loop() –rakenteeseen kirjoitetaan koodirivit LDR -vastuksen arvon lukemiseksi A0 –pinnistä, sekä koodirivit jolla sekä LDR -vastuksen että lämpötila-anturin arvot kirjoitetaan Arduinon sarjaporttiin. Sarjaporttiin kirjoitusta tarvitaan, koska sovellusesimerkissä 3 esitelty Linux-palvelin halutaan käyttöön samanaikaisesti Cayenne-sovellusesimerkin rinnalle. Linux-palvelin lukee Arduinon sarjaporttiin kirjoittamat tiedot ja toimii kuten sovellusesimerkissä 3 on esitetty.

```
int LDR_value = analogRead(LDR_pin);
Serial.print(LDR_value);
Serial.print(",");
Serial.println(sensors.getTempCByIndex(0));
```

Lopuksi lisätään lämpötila-anturille liipaisu (trigger) mikäli tietty lämpötila-arvo ylittyy. Liipaisu lisätään 'Add new...' –kohdasta 'Trigger'. Kuvassa 20 lämpötilarajaksi asetetaan 24,5 °C jonka ylityttyä lähetään sähköpostilla varoitus. Liipaisuja voidaan lisätä haluttaessa useampikin, samalle anturille tai jokaiselle anturille erikseen. Varoitus on myös mahdollista lähettää tekstiviestinä mutta tätä vaihtoehtoa ei tässä esimerkissä lähdetty kokeilemaan.

Kuva 20. Lämpötila-anturin hälytysrajan asettaminen.

'Add custom recipient' –kohdassa määritetään keille sähköpostia lähetetään mikäli lämpötila, tässä tapauksessa, nousee yli 24,5 °C:n. Sähköpostin vastaanottajia voi olla useampia. Kuvassa 21 on Cayennen lähettämä sähköpostivaroitus lämpötilarajan ylitymisestä.

### Device Notification

## Arduino Leonardo DS18B20

has reached the threshold temperature

**24.5°C**

Kuva 21. Cayennen lähettämä sähköposti lämpötila-arvon ylittymisestä.

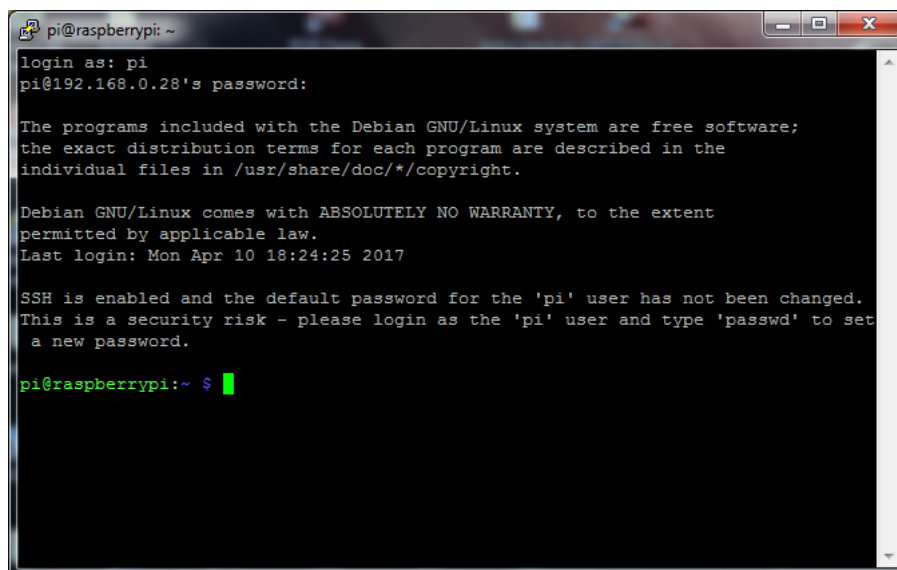
Liitteessä 1 on esitetty Arduino -mikrokontrollerissa toimiva ohjelma kokonaisuudessaan. [33, 34, 36]

#### 4.4 Sovellusesimerkki 3

Kolmannessa sovellusesimerkissä rakennetaan Linux-pohjainen palvelinjärjestelmä anturitiedon vastaanottamiseen ja tallentamiseen. Antureiden tiedot näytetään graafina, joka toimii paikallisessa www-palvelimessa sekä internet-pohjaisessa plot.ly –

palvelussa. Tiedot tallennetaan palvelimessa MySQL –tietokantaan. Palvelin lähettää myös hälytyksen sähköpostilla mikäli tietty anturin arvo ylittyy. [5]

Esimerkissä käytetään Raspberry Pille asennettua Raspbian Jessie -käyttöjärjestelmää. Raspbian Jessie pohjautuu Debian Linux -käyttöjärjestelmään, joten virtuaalikoneessa tehdyt ohjelmistoasennukset ja testaukset toimivat sellaisenaan myös Raspbian Jessie -käyttöjärjestelmässä. Raspbian Jessie -käyttöjärjestelmä asennetaan muistikortille ja liitetään tämän jälkeen Raspberry Pihin. Tarvittavat ohjelmat ja asennuspaketit asennetaan Raspberry Pihin etäyhteydellä, koska siihen ei kytkeä lainkaan hiirtä, näppäimistöä tai näyttöä. Tarvitaan ainoastaan verkkokaapeli ja virransyöttö. Windows:iin asennetulta PuTTY SSH –etähallintaohjelmalta muodostetaan yhteys Raspberry Pihin (kuva 22). Etäyhteyden auettua voidaan Raspberry Pitä hallita aivan kuten käytettäisiin tietokoneen omaa näppäimistöä. Komentokehote-ikkunassa syötetään tarvittavien ohjelmistojen ja moduulien asennuskomennot jotka on esitetty liitteessä 6. [37]



```
pi@rasberrypi: ~  
login as: pi  
pi@192.168.0.28's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Apr 10 18:24:25 2017  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
pi@rasberrypi:~ $
```

Kuva 22. Etäyhteys Raspberry Pihin.

Linux-palvelimeen pitää asentaa www-palvelin- ja tietokantapalvelinohjelmistot sekä esimerkiksi käytetyt ohjelmointikielet. Lisäksi pitää asentaa ohjelma graafien piirtoa varten sekä erilaisia moduuleita, joilla asennetut ohjelmat saadaan keskustelemaan keskenään. [5, 38, 39, 40, 41]

## **Apache**

Apache on The Apache Software Foundationin ylläpitämä avoimeen lähdekoodiin perustuva www-palvelinohjelmisto Linux- ja Windows -käyttöjärjestelmille. Apachen avulla voidaan ylläpitää omaa www-palvelinta lähiverkossa tai liittää www-palvelin internetiin jolloin palvelimella sijaitsevat www-sivut ovat saatavilla koko internetin laajuudella. [38]

## **MySQL**

MySQL on nykyisin Oracle Corporationin omistuksessa oleva tietokantaohjelmisto. MySQL Server on tietokanta-alusta, jolle esimerkiksi Linux-palvelimelle asennettuna pystyy asentamaan useammankin tietokannan eri käyttötarkoituksia varten. Kuten useimmat Linux-sovellukset, myös MySQL perustuu avoimeen lähdekoodiin. [39]

## **PHP**

PHP on ohjelmointikieli, joka on erityisesti tarkoitettu www-sovelluksien kehittämiseen. Asentamalla niin sanotun LAMP-palvelimen, joka on lyhenne sanoista Linux-Apache-MySQL-PHP, saadaan aikaan kokonaisuus, jossa kyseiset ohjelmistot saadaan toimimaan saumattomasti keskenään ja PHP:llä ohjelmoitua siihen haluttuja toimintoja ja lisäominaisuuksia. [40]

## **Python**

Python on yleisohjelmointikieli jota voidaan käyttää monenlaisiin eri käyttötarkoituksiin. Tässä esimerkissä Pythonia käytetään sarjaportille saapuvan tiedon lukemiseen ja jatkokäsittelyyn Raspberry Pilla. [41]

## **Plotly**

Plotly on graafien ja kaavioiden piirtämiseen sekä tiedon analysointiin tarkoitettu ohjelma. Plot.ly on internetissä toimiva palvelu, jossa Plotlyllä piirretyt graafit ja kaaviot ovat nähtävissä millä tahansa laitteella jossa on internet-yhteys. Tässä esimerkissä käytetään Plot.ly:n ilmaisversiota joka on ominaisuuksiltaan riittävä anturitietojen näyttämiseen ja tulkintaan. [42]

## Moduulit

Edellä mainittujen ohjelmien lisäksi asennetaan järjestelmään myös moduuleita, joilla saadaan asennettuihin ohjelmiin lisäominaisuuksia sekä ohjelmille yhteisiä tiedonsiirto-rajapintoja.

Tässä esimerkissä käytetään kuvan 11 mukaista kahden anturin kytkentää. Raspberry Pin sarjaportista luetaan siihen Arduinolta saapuvien anturien arvoja. Arduino -mikrokontrollerin ohjelmakoodi on sama kuin sovellusesimerkki 1:n jälkimmäisessä sovelluksessa. Esimerkin mukaisen järjestelmän rakentaminen aloitetaan asentamalla liitteen 6 mukaiset ohjelmat ja moduulit Linux-palvelimelle. Tarvittavien ohjelmien ja moduulien asennusten jälkeen luodaan palvelimelle tietokanta anturiarvojen tallennusta varten (kuva 23).

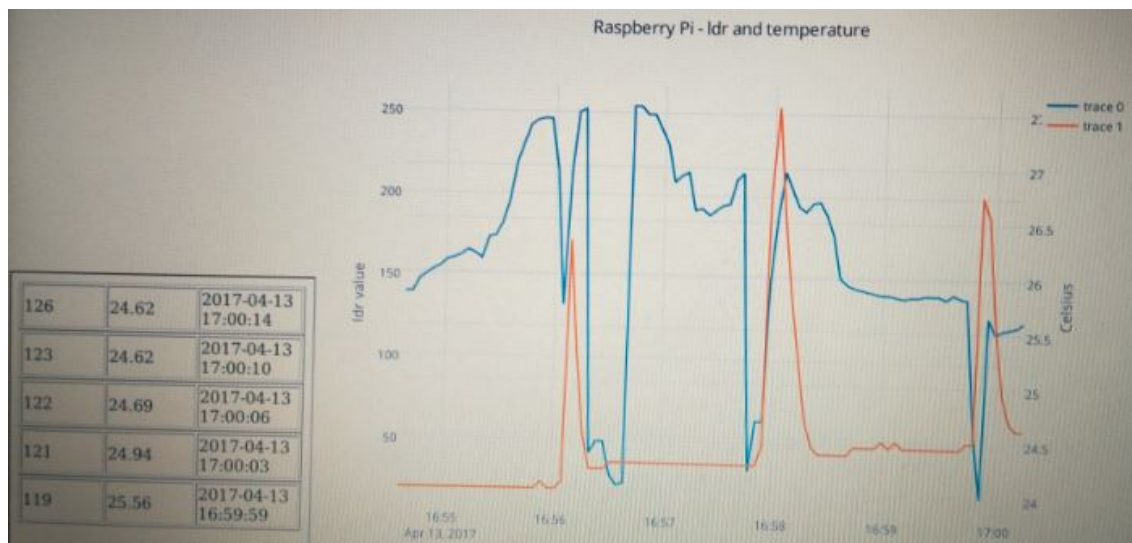
```
mysql> CREATE DATABASE sensordata;  
mysql> USE sensordata;  
mysql> CREATE TABLE sensormonitor (ldr FLOAT, temp FLOAT, timest  
TIMESTAMP);
```

Kuva 23. MySQL -tietokannan perustaminen.

Rivillä CREATE TABLE luodaan tietokantaan taulu nimeltä 'sensordata', jossa on kolme kenttää – ldr, temp ja timest sekä määritetään näiden tietotyypit. Tietotyyppi TIMESTAMP tallentaa jokaisen tietueen tallennuksen yhteydessä aikaleiman jona hetkenä anturien arvot luettiin. [43]

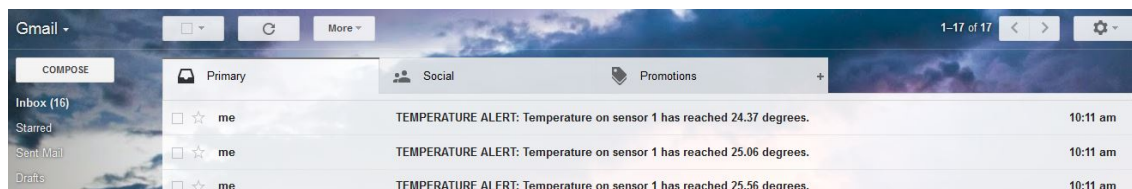
Valmiin sovelluksen halutaan näyttävän www-sivulla molempien antureiden arvot samassa graafissa eri väreillä sekä taulukkona LDR- ja lämpötila-antureiden arvot laskevassa järjestyksessä (kuva 24). Näiden toimintojen toteuttamiseen tarvitaan palvelimessa toimivia ohjelmia.





Kuva 24. Valmis www-sivu.

Sovelluksen halutaan myös lähettävän sähköpostilla varoitus, mikäli etukäteen asetettu lämpötilaraja ylittyy. Tätä varten luotiin gmail:iin sähköpostitili jonne sovelluksen sähköpostihälytykset lähetetään. Kuvassa 25 Linux-palvelimelta saapuneita sähköpostihälytyksiä asetetun lämpötilarajan ylittymisestä.



Kuva 25. Linux-palvelimen lähettämiä sähköpostihälytyksiä.

Edellä mainittujen toimintojen ja ominaisuuksien toteuttamiseksi luodaan kolme palvelimella toimivaa ohjelmaa. Ohjelmat toteutetaan eri ohjelmointikielillä. Index.html –tiedosto toteutetaan html- ja JavaScript-kielillä. Html ei varsinaisesti ole ohjelmointikieli, mutta JavaScript jolla luodaan toiminallisuutta www-sivulle, voidaan sellaiseksi jo luokitella. JavaScript-koodi kirjoitetaan html-koodin joukkoon index.html –tiedostossa. Lisäksi luodaan Python ja PHP-kielillä varsinaiset järjestelmän toimintaa ohjaavat ohjelmat. Kuvassa 24 näkyvät taulukko ja graafi halutaan päivittyvän itsenäisesti ja toisistaan riippumatta ilman, että tarvitsee päivittää koko www-sivua kerralla. Tämä ominaisuus toteutetaan html-koodissa olevalla iframe-tagilla. Iframea käytetään kun halutaan upottaa tiedosto nykyisen tiedoston, index.html, sisään. Iframeja tarvitaan kaksi kappa-

letta. Ensimmäinen näyttää antureiden arvot taulukkona ruudun vasemmassa reunassa. Toinen piirtää antureiden arvot graafina näytölle (kuva 26).

```
<iframe id="query" src="query.php" width="200" height="300"/>
</iframe>
<iframe id="temp" width="800" height="600" frameborder="0"
seamless="seamless" scrolling="no" src="https://plot.ly/
~patrick.lundelin/1/.embed?width=800&height=600"></iframe>
```

Kuva 26. IFrame-tagin määrittäminen.

Kuvan 26 koodissa luodaan kaksi iframe-tagia, query ja temp, sekä määritetään niiden koot. Ensimmäisessä iframe -määrittäksessä kutsutaan ohjelmaa query.php jolla ladataan anturien reaaliaikaiset arvot sisältävä taulukko näytön vasempaan reunaan. Jälkimmäisessä iframe -määrittäksessä ladetaan plot.ly:n pilvipalvelusta omaan käyttäjätiliin määritetty graafi näytölle piirtämistä varten.

Query.php on PHP-kielellä toteutettu ohjelma, jolla haetaan tiedot index.html -sivulla näkyvään taulukkoon suoraan tietokannasta. Tiedosto sisältää tietojen noutamiseen tarvittavan SQL-kyselyn sekä html-muotoilutagit, joilla tiedot sisältävä taulukko tulostetaan näytölle.

Graafin piirtoa varten tarvitaan Pythonilla kirjoitettu koodi, joka myös sisältää useimmat tässä esimerkissä tarvittavat toiminnot. Koodi lukee ensiksi Raspberry Pin sarjaporttiin Arduinolta saapuvat anturitiedot. Tämän jälkeen koodissa avataan tietokantayhteys MySQL -tietokantaan sekä kirjaudutaan sisälle plot.ly -pilvipalveluun omilla käyttäjätunnuksilla. Koodissa määritetään myös niin sanotut 'stream token' -tunnukset, joita tarvitaan yksi kutakin piirrettyä graafia kohti. Tässä esimerkissä piirretään graafit kahdelta eri anturilta, joten stream tokeneita tarvitaan kaksi kappaletta. Plot.ly -pilvipalvelussa käydään luomassa tarvittava määrä stream tokeneita ja liitetään ne osaksi ohjelmakoodia. Seuraavaksi ohjelmakoodissa määritetään graafien piirtoon ja ulkoasuun sekä muotoiluun liittyvät ominaisuudet, kuten graafin tyyppi ja x- sekä y - akselien nimet. Koodin lopussa on while -silmukka, jolla toteutetaan anturien arvojen luennan reaaliaikainen osuus. Anturit lähettävät tietoa Raspberry Pille jatkuvasti joten myös ohjelmakoodin on reagoitava jokaiseen Arduinolta saapuvaan tietoon erikseen. Koska Arduinolta saapuvat anturitiedot tulevat tekstirivinä, jossa erotinmerkinä eri anturien arvojen välillä käytetään pilkkua, on tiedot Raspberry Pille saapuessaan ero-

tettava toisistaan Python-ohjelmakoodissa. Tähän käytetään Pythonin split –toimintoa (kuva 27). [44]

```
val = ser.readline()
datasplit = val.split(",")
ldr = datasplit [0]
temp = datasplit [1]
```

Kuva 27. Pythonin split -toiminto.

Split –toiminnon jälkeen anturien arvot saadaan tallennettua omiin muuttujiinsa. LDR -vastuksen arvo muuttuun ldr ja lämpötilavastuksen arvo muuttuun temp. Näin saadaan kunkin anturin arvoja käsiteltävä erikseen.

Anturien arvot tallennetaan seuraavaksi tietokantaan. Python-koodissa on tietokantayhteyden avaamiseen ja tietojen tallentamiseen tarvittavat koodirivit. Kunkin arvon sijaan omassa kentässään voidaan tietokantaan kohdistaa kyselyitä, jotka kohdistuvat vain tietyn anturin arvoon tai lajitella tietoja tietyn anturin arvojen mukaisesti. Ohjelmakoodin while –silmukassa on myös if –ehtorakenne joka suoritetaan, mikäli tietty lämpötilaraja ylittyy. Ehtorakenteeseen on määritetty toiminto, joka lähettää sähköpostilla hälytyksen mikäli anturin mittaama lämpötila 24 °C ylittyy. Kuvassa 28 on anturiarvojen tietokantaan tallennukseen käytetty SQL-kielinen komento.

```
logg = "INSERT INTO sensormonitor (ldr, temp) VALUES (%s, %s)"
cur.execute(logg, (ldr, temp))
db.commit()
```

Kuva 28. Anturiarvojen tallennus tietokantaan.

Liitteissä 2, 3 ja 4 on esitetty kaikki Linux-palvelimella toimivat ohjelmat.

Liitteen 2 mukainen ohjelma käyttää JavaScript-kieltä kahden www-sivulle määritetyn iframen päivitykseen toisistaan riippumatta. Mallia toiminnallisuuden saavuttamiseksi haettiin Stack Overflow:n JavaScript –aiheiselta foorumilta, koska JavaScript kielenä ei ollut entuudestaan tuttua. Foorumilta löytyikin apua ongelmaan. Löytynyt ohjelmakoodi toimi lähes sellaisenaan, pienin muutoksin ja lisäyksin, tässä esimerkissä esitetyssä järjestelmässä. [45]

Liitteen 3 ohjelmassa muodostetaan tietokantayhteys MySQL –tietokantaan PHP-koodin avulla. Koodin lopussa on html -muotoilukomennot tietokannasta noudettujen

tietojen www-sivulle kirjoittamista varten. PHP-kielessä on valmiina tuki MySQL – tietokantayhteyksiä ja -toimintoja varten. PHP:n itse ylläpitämältä käyttöohjesivulta löytyy ohjeet tietokantayhteyksien ja –toimintojen ohjelmoimiseksi PHP-kielellä. [46]

Liitteen 4 ohjelmassa luetaan Raspberry Pin sarjaporttiin saapuvat anturitiedot, avataan tietokantayhteys anturitietojen tietokantaan tallentamista varten sekä määritetään plotly -ohjelman edellyttämät muotoilukomennot graafin piirtoa varten. Ohjelma on luotu Python-kielellä. Plotly –graafien muotoilukomentoihin löytyi apua 'How to Build Software' -sivustolta sekä plot.ly:n omilta käyttöohjesivuilta. Ohjelman lopussa on toiminto, joka lähettää sähköpostivaroituksen mikäli asetettu lämpötila-anturin arvo ylittyy. Ohjelmakoodia on plotly –graafien muotoilukomentojen osalta muokattu tähän esimerkkiin sopivaksi edellä mainittujen www-sivujen esimerkkien perusteella. Koodin loppuosaa varten on tutkittu Python –dokumentaatiota tarvittavien toiminnallisuuksien saavuttamiseksi. [44, 47, 48, 49]

## 5 POHDINTA

Opinnäytetyön tarkoituksena oli esittää erilaisia kehitysympäristöjä mikrokontrolleriin kytkettyjen antureiden tietojen esittämiseen älypuhelimien sekä tietokoneen näytöllä. Mukaan otettiin myös esimerkki oman Linux -palvelimen pystyttämiseksi anturitietojen näyttämiseksi plot.ly -pilvipalvelussa, omalla www-sivulla sekä tietokantaan tallentamista varten. Esimerkkien yhteisenä päämääränä oli osoittaa, kuinka myös vähemmän ohjelmointiin perehtynyt pystyy rakentamaan mikrokontrolleriin kytkettyjen antureiden testi- ja monitorointiympäristöjä.

Työssä käytettiin verraten yksinkertaisia kytkentöjä, koska pääpaino oli kokonaisuuden saamisessa toimintakuntoon. Työssä käytettyjen esimerkkien pohjalta järjestelmää on helppo laajentaa ja käyttää erilaisia, itse kuhunkin käyttötarkoitukseen kulloinkin sopivia antureita. Ohjelmoinnin osuutta ei voitu kokonaan jättää pois, koska mikrokontrolleria ja siihen kytkettyjä antureita ohjataan ohjelmakoodilla. Toisaalta, myös Linux -palvelimessa tarvitaan järjestelmän toimintoja ohjaavia ohjelmia. Ohjelmointikielinä käytettiin tarpeellisia, mutta myös helpommin omaksuttavia ohjelmointikieliä kuten esimerkiksi PHP ja Python. Älypuhelinlustralle ei tässä työssä ohjelmoitu itse mitään. Lähemmäksi älypuhelinohjelmointia päästiin App Inventor -sovelluskehitysympäristössä, jossa älypuhelinsovellus rakennettiin ohjelmalohkoista itse. Tämä vaatii onnistuakseen vähintään ohjelmoinnin alkeiden ymmärtämistä.

Työn edetessä tuli vastaan useitakin haasteita. Useimmat näistä haasteista liittyivät ohjelmakoodin kirjoittamiseen, sillä oma ohjelmointitausta ei ollut kovinkaan vahva. Kiinnostus mikrokontrolleri- ja palvelinohjelmointiin sekä aktiiviset käyttäjäfoorumit auttoivat kuitenkin asiassa eteenpäin, unohtamatta joitakin yrityksiä ja erehdyksiä – menetelmällä ratkaistuja ongelmia.

Työn jatkokehitystä voisi olla nyt kirjoitettujen ohjelmakoodien optimointi. Optimointi voisi pitää sisällään ohjelmakoodien saattamisen kompaktimpaan muotoon tai esimerkiksi nopeutta vaativien toimintojen kirjoittamisen jollakin toisella, tehokkaammalla ohjelmointikielillä, esimerkiksi C-kielellä. Tässä työssä haluttiin esittää Arduino -mikrokontrollerin käyttöä, mutta antureita on myös mahdollista liittää suoraan Raspberry Piin, jolloin järjestelmä yksinkertaistuu ja kustannukset pienenevät tarvittavien laitteiden vähentymisen myötä. Toisaalta, mikäli järjestelmän ohjaamiseen ei tarvita tietokonetta, voidaan mikrokontrolleri liittää suoraan internetiin kuten sovellusesimerkissä 2

on tehty. Lopputuotteen kannalta olisi myös tarpeellista koteloida laitteet ja anturit sekä suunnitella kytkentä piirilevyille nyt käytetyn koekytkentälevyn asemasta.

Tässä työssä ei tarvittu varsinaista älypuhelinohjelmointia, mutta pääsääntöisesti C#- ja Java-kielillä tapahtuva ohjelmistokehitys älypuhelimille voi olla tarpeellista koko järjestelmän tuotantoympäristöön siirtämistä varten. Tähän voi olla syynä jonkin tarpeellisen toiminnon tai ominaisuuden puuttuminen tässä työssä käytetyistä kehitysympäristöistä tai esimerkiksi tietoturvasyyt. Älypuhelinsovellusten, kuten myös muidenkin sovellusten, asentaminen laitteeseen näkemättä lähdekoodia voi olla tietoturvariski. Yrityksillä on usein tiukat tietoturvamääräykset sallittujen ohjelmistotoimittajien ja asennettavien ohjelmien suhteen.

## 6 YHTEENVETO

Opinnäytetyössä esiteltiin kolmen eri esimerkin kautta mikrokontrolleriin kytkettyjen antureiden arvojen reaaliaikaista näyttämistä älypuhelimella ja tietokoneen näytöllä. Kahdessa ensimmäisessä esimerkissä käytettiin ilmaisia sovelluskehitysympäristöjä jotka ovat vapaasti kaikkien käytettävissä. App Inventorilla tehtiin sovellus joka asennetaan älypuhelimeen. Cayenne -sovelluskehitysympäristö on internetissä toimiva palvelu, jolla tehdyt sovellukset ovat käytettävissä reaaliaikaisesti millä tahansa laitteella. Viimeisessä esimerkissä rakennettiin alusta pitäen Linux -palvelinympäristö joka mahdollistaa tietojen tallennuksen palvelimella sijaitsevaan tietokantaan sekä anturitietojen näyttämisen palvelimeen asennetun www-palvelimen ja plot.ly -pilvipalvelun avulla.

Työssä käytettiin kahta erilaista Arduino -mikrokontrolleria sekä Raspberry Pi -tietokonetta. Kaikkien ominaisuuksien ja toimintojen toteuttamiseksi Linux -palvelimella tutustuttiin syvällisemmin PHP ja Python ohjelmointikieliin sekä Arduinon C-kieliseen ohjelmointiin.

Sovellusesimerkkien toiminta tuli testattua kutakin esimerkkiä rakennettaessa ja tarvittavia ohjelmakoodeja kirjoitettaessa ja testatessa. Testit ja kokeilut suoritettiin pääosin virtuaalikoneympäristössä. Esimerkkien mukaiset ohjelmat toimivat lopulta kaikki etukäteen asetettujen tavoitteiden mukaisesti.

## LÄHTEET

- [1] The history of Bluetooth. [www-dokumentti].  
Saatavilla: <http://www.ericssonhistory.com/changing-the-world/Anecdotes/The-history-of-Bluetooth/> (Luettu: 17.4.2017)
- [2] Wikipedia: Bluetooth. [www-dokumentti].  
Saatavilla: <https://en.wikipedia.org/wiki/Bluetooth> (Luettu: 17.4.2017)
- [3] LoRa Alliance™ Technology. [www-dokumentti].  
Saatavilla: <https://www.lora-alliance.org/What-Is-LoRa/Technology> (Luettu: 16.4.2017)
- [4] LoRaWAN101 – A technical introduction. [www-dokumentti]. Saatavilla:  
[http://portal.lora\\_alliance.org/DesktopModules/Inventures\\_Document/FileDownload.aspx?ContentID=1643](http://portal.lora_alliance.org/DesktopModules/Inventures_Document/FileDownload.aspx?ContentID=1643) (Luettu: 16.4.2017)
- [5] Negus, C., Bresnahan, C., Linux Bible, 8th edition, 2012
- [6] Wikipedia: Internet of things. [www-dokumentti].  
Saatavilla: [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things) (Luettu: 16.4.2017)
- [7] From 1982 Coca-Cola vending machine to latest trend. [www-dokumentti].  
Saatavilla: <https://realbusiness.co.uk/tech-and-innovation/2015/07/15/from-1982-coca-cola-vending-machine-to-latest-trend-what-the-internet-of-things-means-for-business/> (Luettu 17.4.2017)
- [8] Gartner. [www-dokumentti].  
Saatavilla: <http://www.gartner.com/newsroom/id/3165317> (Luettu: 17.4.2017)
- [9] ABIresearch. [www-dokumentti].  
Saatavilla: <https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne/> (Luettu: 17.4.2017)
- [10] MIT App Inventor. [www-dokumentti].  
Saatavilla: <http://AppInventor.mit.edu/explore/about-us.html> (Luettu: 8.4.2017)
- [11] Cayenne features. [www-dokumentti].  
Saatavilla: <https://mydevices.com/cayenne/features/> (Luettu: 13.4.2017)
- [12] Cayenne custom code. [www-dokumentti].  
Saatavilla: <https://mydevices.com/cayenne/features/custom-code/> (Luettu: 13.4.2017)
- [13] Arduino. [www-dokumentti]. Saatavilla: <https://www.arduino.cc/> (Luettu: 9.4.2017)
- [14] Arduino Uno. [www-dokumentti].  
Saatavilla: <https://www.arduino.cc/en/Main/ArduinoBoardUno> (Luettu: 7.4.2017)
- [15] Arduino Uno. [www-dokumentti].  
Saatavilla: <http://www.arduino.org/products/boards/arduino-uno> (Luettu: 18.4.2017)
- [16] Arduino Leonardo ETH. [www-dokumentti].  
Saatavilla: <https://www.arduino.cc/en/Main/ArduinoBoardLeonardoEth> (Luettu: 9.4.2017)



- [17] Arduino Leonardo ETH. [www-dokumentti].  
Saatavilla: <http://www.arduino.org/products/boards/arduino-leonardo-eth> (Luettu: 9.4.2017)
- [18] Wikipedia: Arduino software development. [www-dokumentti].  
Saatavilla: [https://en.wikipedia.org/wiki/Arduino#Software\\_development](https://en.wikipedia.org/wiki/Arduino#Software_development) (Luettu: 9.4.2017)
- [19] Raspberry Pi. [www-dokumentti].  
Saatavilla: <https://www.raspberrypi.org/learning/hardware-guide/requirements/> (Luettu: 8.4.2017)
- [20] Raspberry Pi 3 Model B. [www-dokumentti].  
Saatavilla: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (Luettu: 8.4.2017)
- [21] Wikipedia: Raspberry Pi. [www-dokumentti].  
Saatavilla: [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi) (Luettu: 8.4.2017)
- [22] Wikipedia: Android Studio. [www-dokumentti].  
Saatavilla: [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio) (Luettu: 9.4.2017)
- [23] Meet Android Studio. [www-dokumentti].  
Saatavilla: <https://developer.android.com/studio/intro/index.html> (Luettu: 9.4.2017)
- [24] Wikipedia: Xamarin. [www-dokumentti].  
Saatavilla: <https://en.wikipedia.org/wiki/Xamarin> (Luettu: 9.4.2017)
- [25] Bluetooth HC-05 transmission module. [www-dokumentti].  
Saatavilla: <https://www.velleman.eu/products/view/?id=435518> (Luettu: 9.4.2017)
- [26] LDR. [www-dokumentti].  
Saatavilla: <http://projectshopbd.com/product/ldr/> (Luettu: 17.4.2017)
- [27] DS18B20. [www-dokumentti].  
Saatavilla: <https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html> (Luettu: 9.4.2017)
- [28] DS18B20. [datalehti].  
Saatavilla: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> (Luettu: 9.4.2017)
- [29] DS18B20. [www-dokumentti].  
Saatavilla: <http://articulo.mercadolibre.com.ar/MLA-620686115-sensor-digital-temperatura-ds18b20-18b20-1-wire-dallas- JM> (Luettu: 18.4.2017)
- [30] Debian. [www-dokumentti]. Saatavilla: <http://www.debian.org/> (Luettu: 13.4.2017)
- [31] Virtualbox. [www-dokumentti]. Saatavilla: <https://www.virtualbox.org/> (Luettu: 10.4.2017)
- [32] VMware workstation. [www-dokumentti].  
Saatavilla: <http://www.vmware.com/products/workstation.html> (Luettu: 10.4.2017)
- [33] Arduino temperature control library. [www-dokumentti].  
Saatavilla: <https://github.com/milesburton/Arduino-Temperature-Control-Library> (Luettu: 13.4.2017)

- [34] OneWire library. [www-dokumentti].  
Saatavilla: [https://www.pjrc.com/teensy/td\\_libs\\_OneWire.html](https://www.pjrc.com/teensy/td_libs_OneWire.html) (Luettu: 10.4.2017)
- [35] MIT App Inventor support forum. [www-dokumentti].  
Saatavilla: <https://groups.google.com/forum/#!forum/mitAppInventortest> (Luettu: 11.4.2017)
- [36] SPI library. [www-dokumentti].  
Saatavilla: <https://www.arduino.cc/en/Reference/SPI> (Luettu: 15.4.2017)
- [37] Raspbian. [www-dokumentti].  
Saatavilla: <https://www.raspberrypi.org/downloads/raspbian/> (Luettu: 10.4.2017)
- [38] Apache – http server project. [www-dokumentti].  
Saatavilla: <https://httpd.apache.org/> (Luettu: 10.4.2017)
- [39] MySQL. [www-dokumentti]. Saatavilla: <https://www.mysql.com/> (Luettu: 13.4.2017)
- [40] PHP. [www-dokumentti]. Saatavilla: <http://www.php.net/> (Luettu: 11.4.2017)
- [41] Python. [www-dokumentti].  
Saatavilla: <https://www.python.org/> (Luettu: 12.4.2017)
- [42] Plotly. [www-dokumentti]. Saatavilla: <https://plot.ly/> (Luettu: 12.4.2017)
- [43] MySQL 5.7 reference manual. [www-dokumentti].  
Saatavilla: <https://dev.mysql.com/doc/refman/5.7/en/> (Luettu: 13.4.2017)
- [44] Python documentation – common string operations. [www-dokumentti].  
Saatavilla: <https://docs.python.org/2/library/string.html> (Luettu: 13.4.2017)
- [45] JavaScript, iframe reload. [www-dokumentti]. Saatavilla:  
<http://stackoverflow.com/questions/29947502/javascript-reloading-iframe> (Luettu: 10.3.2017)
- [46] PHP, mysql\_connect. [www-dokumentti]. Saatavilla:  
<http://php.net/manual/en/function.mysql-connect.php> (Luettu: 10.3.2017)
- [47] How to create two y-axes streaming plotly. [www-dokumentti]. Saatavilla:  
<http://www.howtobuildsoftware.com/index.php/how-do/bj3F/python-plotly-how-to-create-two-y-axes-streaming-plotly> (Luettu: 11.3.2017)
- [48] Line and scatter plots in Python. [www-dokumentti]. Saatavilla:  
<https://plot.ly/python/line-and-scatter-plots-tutorial/> (Luettu: 11.3.2017)
- [49] How to send emails with Gmail using Python. [www-dokumentti]. Saatavilla:  
<http://stackabuse.com/how-to-send-emails-with-gmail-using-python/> (Luettu: 12.3.2017)

```
#define CAYENNE_PRINT Serial
#define VIRTUAL_PIN V1
#include <OneWire.h>
#include <DallasTemperature.h>
#include <CayenneEthernetW5500.h>
#include <Ethernet2.h>
#include <SPI.h>

const int tmpPin = 2;
int LDR_pin = A0;

OneWire oneWire(tmpPin);
DallasTemperature sensors(&oneWire);

char token[] = "*****";

void setup()
{
  Serial.begin(9600);
  Cayenne.begin(token);
  sensors.begin();
}

void loop()
{
  Cayenne.run();
}
CAYENNE_OUT(VIRTUAL_PIN)
{
  sensors.requestTemperatures();
  Cayenne.celsiusWrite(VIRTUAL_PIN, sensors.getTempCByIndex(0));
  int LDR_value = analogRead(LDR_pin);
  Serial.print(LDR_value);
  Serial.print(",");
  Serial.println(sensors.getTempCByIndex(0));
}
```

```
<head>
<title>IFrame Reload</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<script type="text/javascript"><!--

var reloadInterval = 3000;

function init()
{
  setTimeout('reload()', reloadInterval);
}
interval
function reload()
{
  var iframe = document.getElementById('query');
  if (!iframe) return false;
  iframe.src = iframe.src;
  setTimeout('reload()', reloadInterval);
}
window.onload = init;
--></script>

</head>

<body>

<iframe id="query" src="query.php" width="300" height="300"/></iframe>
<iframe id="temp" width="800" height="600" frameborder="0" seamless="seamless"
scrolling="no" src="https://plot.ly/~p*****.l*****/1/.embed?width=800&height=
600"></iframe>

</body>
</html>
```

```
<?php

$dbh = mysql_connect(localhost, root, root) or die ("Unable to connect to
MySQL");

$connection = mysql_select_db(sensordata, $dbh) or die ("database error");

$result = @mysql_query ("SELECT * FROM sensormonitor ORDER BY timest DESC LIMIT
5");

while ($row = mysql_fetch_row ($result))
{
echo "
<TABLE BORDER=1>
<TR>
<TD WIDTH=100>$row[0]</TD>
<TD WIDTH=100>$row[1]</TD>
<TD WIDTH=100>$row[2]</TD>
</TR>
</TABLE>
";
}

?>
```

```

#!/usr/bin/python
#!/usr/bin/env python
import serial
import smtplib
import MySQLdb as mdb
import subprocess
import re
import sys
import time
import datetime
import plotly.plotly as py
from plotly.graph_objs import Scatter, Layout, Figure, Data, Stream, YAxis

ser = serial.Serial('/dev/ttyACM0', 9600)

db = mdb.connect(host='localhost',user='*****',passwd='*****',db='sensordata');

cur = db.cursor()

username          = 'p*****.l*****'
api_key           = '*****'
stream_token_ldr  = '*****'
stream_token_temp = '*****'

py.sign_in(username, api_key)

trace_ldr = Scatter
(
    x=[],
    y=[],
    stream=Stream(token=stream_token_ldr),
    yaxis='y'
)
trace_temp = Scatter
(
    x=[],
    y=[],
    stream=Stream(token=stream_token_temp),
    yaxis='y2'
)
layout = Layout
(
    title='Raspberry Pi - ldr and temperature',
    yaxis=YAxis
    (
        title='ldr'
    ),
    yaxis2=YAxis
    (
        title='Celsius',
        side='right',
        overlaying="y"
    )
)

```

```
data = Data([trace_ldr, trace_temp])
fig = Figure(data=data, layout=layout)

print py.plot(fig, filename='Raspberry Pi - ldr and temperature')

stream_ldr = py.Stream(stream_token_ldr)
stream_ldr.open()

stream_temp = py.Stream(stream_token_temp)
stream_temp.open()

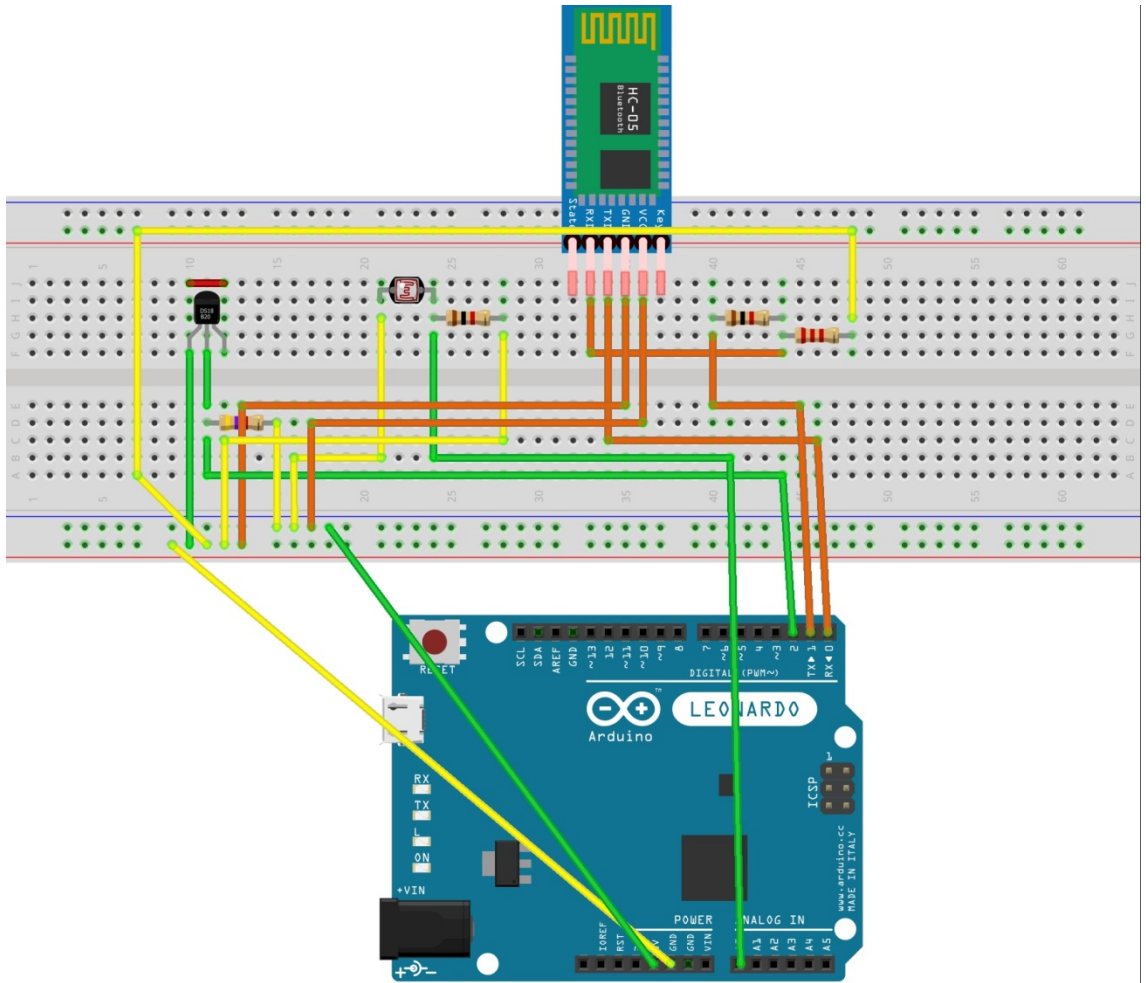
while(True):

    val = ser.readline()
    datasplit = val.split(",")
    ldr = datasplit [0]
    temp = datasplit [1]
    logg = "INSERT INTO sensormonitor (ldr, temp) VALUES (%s, %s)"
    cur.execute(logg, (ldr, temp))
    db.commit()

if float(temp) >= 24.5:
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login("*****@gmail.com", "password")
    msg = "Subject: TEMPERATURE ALERT: Temperature on sensor 1 has reached %s
degrees." % temp
    server.sendmail("*****@gmail.com", "*****@gmail.com", msg)
    server.quit()

now = datetime.datetime.now()
stream_ldr.write({'x': now, 'y': ldr })
stream_temp.write({'x': now, 'y': temp })

time.sleep(0.25)
```





```
sudo apt-get install python-dev

wget https://github.com/pypa/setuptools/raw/bootstrap/ez\_setup.py -O
- | sudo python

sudo easy_install -U distribute

sudo apt-get install python-pip

sudo pip install rpi.gpio

sudo pip install plotly

sudo apt-get install apache2

sudo apt-get install mysql-server

sudo apt-get install php5 php5-common php5-cli php5-mysql
libapache2-mod-php5

sudo apt-get install python-mysqldb

sudo pip install pyserial
```