

Adafruit Circuit Playground Express

Langaton yhteys ja tiedonsiirto



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeenlinnan korkeakoulukeskus, Tietojenkäsittelyn koulutusohjelma

kevät, 2019

Ari Oksanen

Tietojenkäsittelyn koulutusohjelma
Hämeenlinnan korkeakoulukeskus

Tekijä	Ari Oksanen	Vuosi 2020
Työn nimi	Adafruit Circuit Playground Langaton yhteys ja tiedonsiirto	
Työn ohjaaja/t	Tommi Saksa, Lasse Seppänen	

TIIVISTELMÄ

Opinnäytetyön tavoitteena on toteuttaa ekosysteemi IoT-laitteen ja mobiililaitteen välisen langattoman yhteyden, tiedonsiirron sekä laitteen ohjauksen tarpeisiin. Työssä tutkitaan mahdollisuuksia toteuttaa mobiilisovellus, jolla voidaan ohjata Adafruit Circuit Playground -kehitysalustan sensoreita Bluetooth-yhteyden kautta. Kehitysalustalta on myös tarkoituksena saada siirrettyä sensoreiden tuottamaa dataa sovellukseen. Tutkimustyön tuloksena tuotetaan ohjeistus vastaavan systeemin rakentamiseksi.

Ekosysteemi tulee koostumaan Adafruit Circuit Playground -kehitysalustasta, Adafruit Bluefruit LE UART Friend -Bluetooth-moduulista ja Android-sovelluksesta. Kehitysalustan ohjelmointi toteutetaan Arduino-kielellä ja Android-sovelluksen kehittämiseen käytetään joko Microsoft Xamarin- tai Android Studio-ohjelmistokehystä. Järjestelmä testataan Android-puhelimella.

Systeemin toteutus onnistui suunnitelman mukaisesti. Adafruit Playground Express -kehitysalustan ja mobiililaitteen välinen yhteys saatiin toimimaan langattomasti. Keskeiset tutkimusongelmat saatiin näin ollen ratkaistua.

Avainsanat Puettava teknologia, langaton tiedonsiirto, Bluetooth Low Energy, Arduino, Android Studio.

Sivut 26 sivua, joista liitteitä 0 sivua

Degree Programme in Business Information Technology
 Hämeenlinna University Centre

Author	Ari Oksanen	Year 2020
Subject	Adafruit Circuit Playground Wireless connection and data transfer	
Supervisors	Tommi Saksa, Lasse Seppänen	

ABSTRACT

The purpose of this thesis is to implement an ecosystem for the wireless connection, data transfer and device control between the IoT device and the mobile device. The work explores the possibilities of implementing a mobile application that can control the sensors of the Adafruit Circuit Playground development platform via Bluetooth. The development platform is also intended to transfer the data generated by the sensors to the application. As a result of the research, instructions for building a corresponding system are produced.

The ecosystem will be made up of the Adafruit Circuit Playground, Adafruit Bluefruit LE UART Friend, and the Android app. The framing platform is programmed in Arduino, and the Microsoft Xamarin or Android Studio software framework is used to develop the Android application. The system is tested using Android mobile phone.

The implementation of the system according to plan was successful. The wireless connection between Adafruit Playground Express development board and the mobile device is working wirelessly, hence the main research questions have been answered.

Keywords Wearable technologies, wireless communication, Arduino, Android Studio

Pages 26 pages including appendices 0 pages

SANASTO

AT-komento	(Attention) Hayes commands, alun perin modeemien ohjaukseen kehitetty komentokieli
Bluetooth	lyhyen kantaman langaton tiedonsiirtotekniikka
CTS	Clear to Send, sarjaliikenneyhteydessä käytetty datan vastaanotto valmiusilmoitus
CPX	Lyhenne sanoista Circuit Playground Express
GitHub	Git-versionhallinta käyttävien ohjelmakehitysprojektien sivusto
IDE	integrated development environment, integroitu ohjelmointiympäristö
IoT	Internet of Things, suomeksi esineiden internet
MIT-lisenssi	Massachusetts Institute of Technologyssä (MIT) kehitetty vapaa ohjelmistolisenssi
MockFlow	rautalankamallien suunnitteluun tarkoitettu pilvipalvelu
RTS	Request to Send, sarjaliikenneyhteydessä käytetty datan lähetys pyyntö
UART	Universal Asynchronous Receiver Transmitter, protokolla rinnakkaismuotoisen tiedon muuntamiseksi sarjamuotoiseksi
XML	Extensible Markup Language, tiedon säilyttämiseen ja siirtämiseen kehitetty merkintäkieli

SISÄLLYS

SANASTO.....	3
1 JOHDANTO.....	5
2 INTERNET OF THINGS (IOT)	6
2.1 Laitteita ja käyttökohteita	6
2.2 Puettava tekniikka.....	7
2.3 Kehitysalustat ja -työkalut.....	9
2.4 Sensorit ja anturit.....	10
2.5 Tietoturva.....	10
3 KEHITYSTYÖKALUT JA TEKNIikka	11
3.1 Arduino.....	11
3.2 Adafruit.....	11
3.2.1 Circuit Playground Express	12
3.2.2 Adafruit Bluefruit LE UART Friend	12
3.2.3 Adafruit-kehitysalustan ohjelmointi.....	13
3.2.4 Arduino IDE.....	13
3.3 Android.....	14
3.4 Android Studio	15
3.5 Xamarin	15
4 ANDROID- JA ARDUINO-SOVELLUKSIEN TOTEUTUS	17
4.1 Android-sovelluksen ohjelmointi	18
4.2 Arduino-sovelluksen ohjelmointi	19
5 LANGATON IOT-EKOSYSTEEMI	21
5.1 Järjestelmän kytkentä	21
5.2 Järjestelmän testaus.....	23
6 YHTEENVETO	26
LÄHTEET.....	27

1 JOHDANTO

Opinnäytetyössä käsitellään vaatteisiin ommeltavan elektroniikan testialustaa ja sen tuottaman datan muuntamista ymmärrettävään muotoon. Testialusta sisältää sensoreita, jotka mittaavat erilaisia ympäristön suureita, kuten lämpötilaa, liikettä ja valoisuutta. Sensorien tuottama data voidaan tallentaa alustan sisäiseen muistiin ja lähettää alustaan liitettävän Bluetooth-moduulin välityksellä esimerkiksi mobiililaitteeseen. Laitteen liitettävyyden ja tiedonkeruun ominaisuuksien takia laitetta kutsutaan IoT eli Internet of Things -laitteeksi ja koska laite voidaan ommella vaatteisiin, se kuuluu myös kategoriaan puettava teknologia.

Puettava teknologia mahdollistaa IoT-laitteiden kuljettamisen mukana saumattomasti. Puettavan teknologian laitteille on ominaista, että ne ovat pienikokoisia ja ne kulkevat mukana mahdollisimman huomaamattomasti. Laitteet voivat olla erillisiä, esimerkiksi kehoon kiinnitettyjä tai vaatteisiin integroituja. Laitteita käytetään keräämään tietoa käyttäjästä ja ympäristöstä.

Tiedon langattomaan siirtämiseen liittyy monia kysymyksiä. Näitä ovat muun muassa langattoman yhteyden toteutus ja toimintavarmuus sekä laitteiden tietoturva. Tässä työssä keskitytään selvittämään langattoman yhteyden toteuttamista ja toimintavarmuutta. Tietoturvaa tullaan käsittelemään teoriatasolla.

Opinnäytetyön tavoitteena on toteuttaa ekosysteemi IoT-laitteen ja mobiililaitteen välisen langattoman yhteyden, tiedonsiirron sekä laitteen ohjauksen tarpeisiin. Työn toimeksiantaja on Hämeen Ammattikorkeakoulun Smart Design -työryhmä. IoT-laitteeksi on valikoitunut Adafruit Circuit Playground -kehitysalusta. Opinnäytetyö sisältää teoriaosuuden, jossa käsitellään yleisesti IoT-laitteita ja puettavaa teknologiaa ja tietoturvaa, sekä yksityiskohtaisemmin työhön vaadittavia kehitystyökaluja ja tekniikoita. Käytännön osuudessa tarkastellaan järjestelmän kokoonpanoa ja käydään läpi sovelluksen suunnittelu- ja ohjelmointiprosessit. Opinnäytetyön aihe valikoitui kiinnostuksesta puettavaan teknologiaan ja ohjelmointiin. Työssä yhdistyvät molemmat aihealueet käytännön tasolla. Aiheen valitsemiseen vaikutti myös pitkä työkokemus elektroniikan alalta. Työkokemuksesta ja omasta työtilasta oli hyötyä myös systeemin kokoamisessa, johon vaadittiin muun muassa juotostaitoja ja -välineistöä.

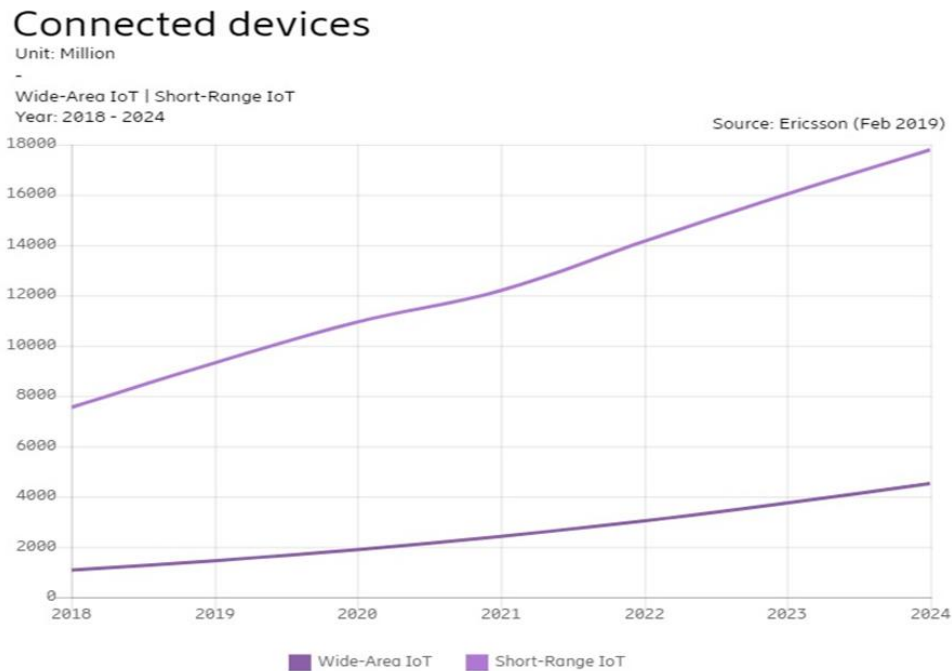
Opinnäytetyön keskeiset tutkimuskysymykset ovat:

- Miten toteutetaan langaton yhteys laitteiden välillä ja datan siirto kehitysalustasta mobiililaitteeseen?
- Miten toteutetaan sensorien ohjaus langattomasti mobiililaitteesta?

2 INTERNET OF THINGS (IOT)

Esineiden internet (engl. Internet of Things, lyhyemmin IoT) muodostuu internetiin kytketyistä laitteista, joiden avulla on mahdollista mitata ja havainnoida ympäristöä sekä välittää tätä tietoa analysoitavaksi. Tämän lisäksi laitteita ja sensoreita voidaan yleensä myös ohjata internetin kautta. IoT-laitteiden määrän ennustetaan lisääntyvän eksponentiaalisesti tulevina vuosina (Vermesan, O., & Friess, 2013).

IoT-tekniikamarkkinoiden on yksi nopeimmin kasvavista tekniikamarkkinoista. Ruotsalainen telekommunikaatio- ja informaatiotekniikayritys Ericsson on arvioinut tammikuun 2019 raportissaan, että IoT-matkapuhelinverkkojen määrä olisi 4,1 miljardia vuonna 2024. Laajapeittoisten (wide-area) IoT-laitteiden määrän arvioidaan kasvavan 4,5 miljardiin ja lyhyen kantaman (short-range) laitteiden määrän 17,8 miljardiin (Ericsson, 2019).



Kuva 1. Mobiiliverkkoon yhdistetyt IoT-laitteet 2018-2024 (Ericsson, 2019)

2.1 Laitteita ja käyttökohteita

Esineiden internetiä voidaan soveltaa monilla eri tekniikan ja teollisuuden aloilla. Esimerkiksi energia-alalla sähköyhtiöt ovat asentaneet kuluttajien sähkökaappeihin etäluettavia mittareita, joilla kuluttaja saa lähes reaaliaikaista tietoa kulutuksesta ja energiayhtiö tietoa laskutukseen sekä analytiikkaan. Logistiikassa esineiden internetiä voidaan hyödyntää esimerkiksi ruokatavaran mitatessa ympäristönsä lämpötilaa jakeluketjussa ja hälyttää mikäli lämpötila poikkeaa sallituista rajoista.

Kulutuselektronikassa IoT-laitteita käytetään esimerkiksi mittaamaan ja analysoimaan kehon lämpötilaa, sydämen sykettä, sijaintia, nopeutta, askelmääriä. Näistä esimerkkinä urheilukellot ja aktiivisuusrannekkeet. Koti on suhteellisen helppo muuttaa älykodiksi vaihtamalla pistorasiat ja kytkimet etäohjattaviksi ja lisäämällä lähiverkkoon laitteen, jolla etäohjaus voidaan toteuttaa. Kodeissa IoT-laitteet antavat mahdollisuuden ohjata toimintoja automaattisesti ja etäyhteydellä. Esimerkiksi lämmitystä, valaistusta, ilmanvaihtoa ja sähkön kulutusta voidaan ohjata ja seurata internetiin kytketyn laitteen avulla. Lisäksi taloon kytketyt sensorit voivat tuottaa tietoa rakennuksen kunnosta ja ilmoittaa poikkeavista arvoista. Terveysthuollossa esineiden internet tuo uusia mahdollisuuksia muun muassa vanhusten palveluihin. Ikääntyvät ihmiset voivat asua kotonaan pidempään, kun heidän hyvinvointiaan voidaan seurata etäyhteyksien avulla. Etänä luettavat verensokeri-, sekä muita lääkintää vaativia arvoja mittaavat sensorit, parantavat myös muiden terveydenhuoltopalveluita säännöllisesti käyttävien elämänlaatua. Muita IoT-tekniikoista hyötyviä aloja ovat esimerkiksi kiinteistöhuolto, rakennusteollisuus, jätehuolto ja kunnossapito. Eräs suurista aloista on myös autoteollisuus, jossa IoT-laitteet jo näkyvät erilaisten hätäpalvelujen, sijainti-, etädiagnostiikka- ja turvatoimintojen myötä. Tulevaisuudessa IoT-laitteiden mahdollistama monitorointi on tärkeässä roolissa autonomisissa kuljetusjärjestelmissä. (Vermesan & Friess, 2015)

2.2 Puettava tekniikka

Puettavalla tekniikalla käsitetään laitteita, joita voidaan pitää keholla tai kehon läheisyydessä. Ne voivat sijaita esimerkiksi kehoon kiinnitettyinä (urheilukello, sykevyö) tai vaatteissa (ommeltavat laitteet, älyvaatteet). Kuten muidenkaan IoT-laitteiden, ei puettavan tekniikan laitteen ole välttämätöntä olla yhteydessä internetiin vaan se voi toimia esimerkiksi Bluetooth tai LoRa (Long Range) -yhteyttä hyödyntäen (Godfrey ym., 2018).

Suomessa puettavaa teknologiaa ja älyvaatteita kehitti vuonna 2001 perustettu Clothing+ -yritys. Yritys myytiin vuonna 2015 yhdysvaltalaiselle Jabil Circuit -pörssiyritykselle. Yrityksen Kankaanpäässä toiminut yksikkö suljettiin vuonna 2018. Tällä hetkellä Suomessa toimii useita puettavaan teknologiaan keskittyvää yritystä ja tahoja ("Puettavan teknologian tarina päättyy | Satakunnan Kansa", n.d.).

Kuopiolainen Myontec tutkii ja kehittää lihasaktivaatiota mittaavia sensoreita ja älyvaatteita. Vaatteiden avulla voidaan mitata työn kuormittavuutta ja sitä kautta parantaa työn laatua ja optimoida työmäärää. Älyvaatteiden käytöllä pyritään vähentämään loukkaantumisia ja näistä johtuvia sairauspoissaoloja. Yritys kehittää ja valmistaa myös urheilijoille ja valmentajille suunnattuja älyvaatteita ja sovelluksia, joilla mitataan lihasten toimintaa ja kuntoa ("Wearable Technology | Myontec", n.d.).



Kuva 2. Myontec Mbody 3 -älyvaate urheilijoille

Aalto-yliopisto ja Helsingin Lastensairaala toimivat yhteistyössä tutkiesaan mahdollisuuksia tekstiilelektrodien hyödyntämisestä keskosvauvojen neurobiologisten sairauksien hoidossa ("Keskosten älyvaatteet ja 6 muuta suomalaista älytekstiiliuutuutta - Fab", n.d.).

Lastenvaatteita valmistava Reima, kehitti yhdessä muun muassa urheilukelloja valmistavan Suunnon kanssa, lapsille suunnatun aktiivisuusrannekkeen. Rannekkeen avulla lapsi ja vanhemmat voivat seurata lapsen liikkumista. Rannekkeen mittaamat tiedot välittyvät erilliseen sovellukseen, jossa ne muunnetaan energiaksi, jonka avulla voi edetä sovelluksen pelimaailmassa. Tavoitteena on kannustaa lasta liikkumaan aktiivisesti pelaamisen avulla ("ReimaGO | Reima", n.d.).

Suunto on kehittänyt Movesensen, avoimen kehitysympäristön liiketunnistussovelluksille. Sitä voidaan käyttää monenlaiseen liikkeen tuottavan datan seuraamiseen ja analysointiin. Sen avulla voidaan seurata muun muassa urheilua, terveyttä ja laitteita. Movesensen verkkosivuilta voi ladata kehitystyökalut ja verkkokaupasta voi tilata sensoreita ja tarvikkeita oman sovelluksen kehittämiseksi ("Movesense - open wearable device platform for motion and biometrics", n.d.).



Kuva 3. Suunto Movesense-sensori

2.3 Kehitysalustat ja -työkalut

Puettavan tekniikan testaamiseen ja prototyyppien valmistamiseen on saatavilla erilaisia kehitysalustoja. Valmistajia ovat muun muassa Arduino, Adafruit Electronics ja Sparkfun Electronics. Adafruit ja Sparkfun ovat aloittaneet tekemällä omia mikrokontrollerikortteja Arduino-korttien pohjalta ja suunnittelemalla yhdessä Arduino-yhtiön kanssa. Nykyään esimerkiksi Adafruit-yhtiön korteista suurin osa on heidän oman suunnittelunsa tulosta. Kortit sisältävät yleensä minimissään mikrokontrollerin, ohjelmointiliittimen, virtaliittimen ja ohjelmoitavia liittimiä ulkoisia sensoreita tai muita laitteita varten.

Taulukossa 1. on vertailua kahden puettavan teknologian kehitysalustan, Arduino Lilypad ja Adafruit Circuit Playground Express välillä.

Taulukko 1. Lilypadin ja CPX:n vertailutaulukko.

BOARD	LILYPAD ARDUINO USB	ADAFRUIT CIRCUIT PLAYGROUND EXPRESS
Microcontroller	ATmega32u4	ATSAMD21 ARM Cortex M0
Operating Voltage	3.3V	3.3V
Input Voltage	3.8V-5V	1.6 - 3.8
Digital I/O Pins	9	8
PWM Channels	4	8
Analog Input Channels	4	8
DC Current per I/O Pin	40 mA	20mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by	256 KB (ATSAMD21), 2MB external Flash Memory
SRAM	2.5 KB (ATmega32u4)	32 KB
EEPROM	1 KB (ATmega32u4)	no Eeprom, 2MB external Flash Memory
Clock Speed	8 MHz	48 MHz

2.4 Sensorit ja anturit

Kehitysalustoihin liitettäviä sensoreita ja antureita on kymmeniä erilaisia. Käytännössä lähes kaikki ympäristöä havainnoivat, dataa tuottavat sensorit ja anturit voidaan liittää IoT-laitteeseen.

Sensori- ja anturi-nimeä käytetään usein arkikielessä ja tuotetussa tekstissäkin sekaisin. Opinnäytetyön kirjoittajan käsityksen mukaan sensorit eroavat antureista lähinnä siinä, että sensorit sisältävät yleensä lisäkomponentteja anturilta saatavan mittaustiedon käsittelyyn ja tuottamiseen. Lisäksi sensori voi sisältää useampia antureita.

2.5 Tietoturva

Verkkoon liitettävien laitteiden turvallisuus on yksi IoT-laitteiden suurimmista ongelmista. Monien IoT-laitteiden ohjelmistot eivät ole päivitettävissä ja pahimmissa tapauksissa käyttäjätunnukset ja salasana eivät ole vaihdettavissa. Tämä lisää riskiä, mikäli laitteen ohjelmistosta paljastuu tietoturva-aukkoja, joita hyödyntämällä hyökkääjä voi ottaa laitteen haltuunsa. Monissa IoT-laitteissa on tehtaalla asetettu käyttäjätunnus ja salasana, joka on sama kaikissa sarjan laitteissa. Tällöin laitteen käyttäjän vastuulla on vaihtaa tunnukset ja mikäli käyttäjä ei perehdy tuotteen ominaisuuksiin ja ohjeistukseen riittävästi, jää tämä usein valitettavasti tekemättä. ("Dyn Analysis Summary Of Friday October 21 Attack | Dyn Blog", n.d.).

Syyskuussa 2016 hyökkääjät kaappasivat Mirai-haittaohjelman avulla IoT-laitteita, joissa oli oletusarvoiset käyttäjätunnukset ja salasanat. IoT-laitteista muodostettiin bottiverkko, joka teki palvelunestohyökkäyksiä häiriten muun muassa Twitterin ja Spotifyn toimintaa.

Marraskuussa 2016 tapahtuneessa palvelunestohyökkäyksessä verkkoon liitetty kiinteistöautomaation järjestelmä kaatui Lappeenrannassa ja kahden kerrostalon lämmönjakelu katkesi. Järjestelmän kaatuminen oli todennäköisesti seurausta IoT-laitteisiin kohdistuvasta hyökkäyksestä. Järjestelmää valvoi vain yksinkertainen palomuuuri. Tapahtuneen jälkeen palomuuuri uusittiin asianmukaiseksi. ("Lappeenrannasta varoitettava esimerkki – 'Ei kannata tuudittautua siihen, että isoilta vahingoilta on vielä välttytty' | Tekniikkatalous", n.d.)

3 KEHITYSTYÖKALUT JA TEKNIikka

Järjestelmän sovelluksen kehitystyökaluksi valikoitui Android Studio. Verailujen ja testauksien jälkeen todettiin, että Android Studio -kehitysympäristölle on saatavilla valmis pohja Android-sovelluksen toteuttamiseksi. Xamarin-ohjelmistolla testatut pohjat olivat joko vanhentuneita, tai eivät olleet tarkoituksen mukaisia tätä työtä ajatellen. Koska opinnäytetyön tekijällä ei ollut aiempaa kokemusta Circuit Playground -alustan ja Bluetooth-moduulin ohjelmoinnista sekä työhön käytettävän ajan määrä huomioiden olisi ollut haastavaa aloittaa sovelluksen kehitys täysin puhtaalta pöydältä.

Työssä tutkittiin kumpi ohjelmointikielistä, CircuitPython vai Arduino, soveltuisi paremmin mikrokontrollerin ohjelmoimiseen kyseisessä projektissa. Selvisi, että CircuitPython kirjastot ovat vielä sen verran kehitysvaiheessa, että projektiin sopivaa Bluetooth-kirjastoa ei työtä tehdessä ollut. Arduino valikoitui myös sen laiteläheisyyden vuoksi ja auttoi näin ollen paremmin ymmärtämään systeemin toimintaa. Microsoft MakeCode vaikutti taas visuaalisuutensa ja lohkoihin perustuvan logiikkansa puolesta turhan yksinkertaiselta projektin tutkimuskysymyksiä ajatellen.

3.1 Arduino

Arduino on suunnittelu- ja testausalusta, jonka ohjelmat perustuvat avoimeen lähdekoodiin. Sen ytimenä on piirilevylle sijoitettu mikrokontrolleri, jonka ympärille on suunniteltu erilaisia lisämoduuleja (shield). Moduuli on yleensä joko sensori tai se on tarkoitettu yhteyden muodostamiseen muiden laitteiden välille. Moduuleja voidaan kytkeä useampia ja ne yhdistetään joko liittimen tai kytkentäjohtojen avulla Arduino-kehitysalustaan. Arduino-kehitysalusta ohjelmoidaan Wiring-ohjelmointikielestä kehitetyllä Arduino-kielellä. Ohjelmointiin käytetään Processing-projektiin perustuvaa Arduino IDE -ohjelmistoa ("Arduino - Introduction", n.d.).

Arduinon alkuvaiheet alkoivat Italiassa Interaction Design Institute -yliopistossa vuonna 2005. Arduinon kehitys alkoi tarpeesta saada oppilaille helpokäyttöinen alusta elektroniikkaprojekteihin. Vuonna 2001 oli kehitetty projekti nimeltä Processing, joka pyrki madaltamaan ohjelmoinnin aloittamisen kynnystä. Tämä toteutettiin eräänlaisen digitaalisen luonnoskirjan avulla, jolla luodaan pienellä ohjelmointimäärällä visuaalisia ja graafisia multimediateoksia. Vuonna 2003 Hernando Barragán alkoi suunnitella Processing-projektin pohjalta Arduinon edeltäjää, Wiring-nimistä mikrokontrollerikorttia (Nussey, 2013).

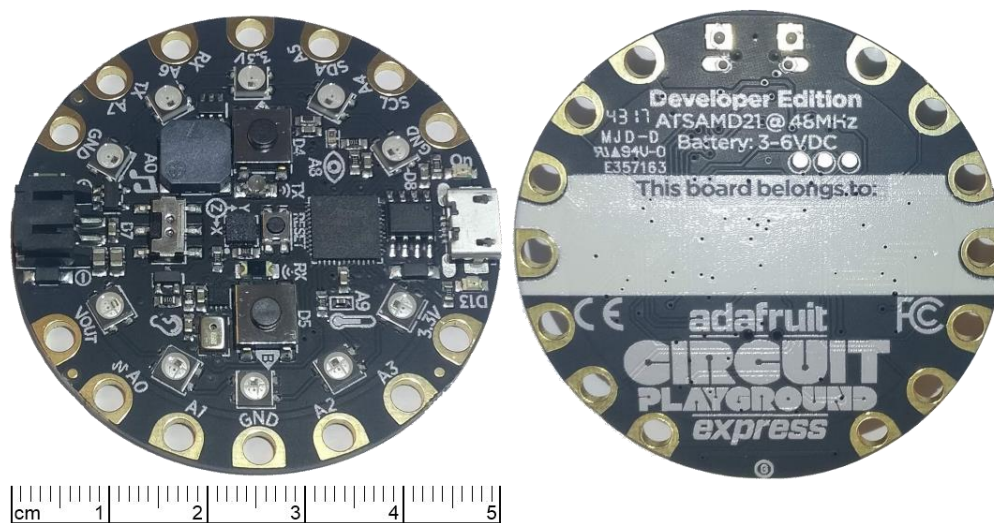
3.2 Adafruit

Massachusetts Institute of Technology eli MIT-korkeakoulun insinööri Limor "Ladyada" Fried perusti Adafruitin vuonna 2005. Hänen tavoitteenaan

oli suunnitella valmiita rakennussarjoja ja luoda verkko-opetusalustan kaiken tasoisille elektroniikkaharrastajille. Adafruit valmistaa omat mikrokontrollerikorttinsa Arduinon pohjalta ja ohjelmointiin käytetään Arduino-ohjelmointikieltä. Uudempia kortteja voidaan ohjelmoida myös Microsoft MakeCode- ja CircuitPython-kielillä. Adafruit Industries -yhtiöllä on tänä päivänä yli sata työntekijää ja 4600m²:n tehdas New Yorkissa. Yhtiön liikevaihto oli vuonna 2016 noin 45 miljoonaa USD. (”Inside Adafruit, the Open-Source Manufacturing Maker Champions”, n.d.)

3.2.1 Circuit Playground Express

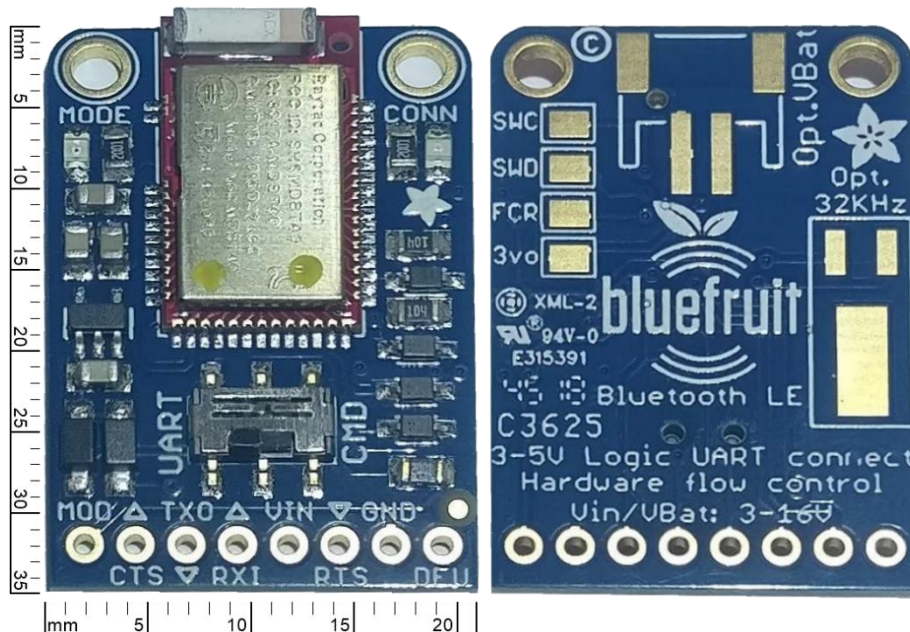
Adafruit Circuit Playground Express on kehitysalusta, jolla voidaan tehdä puettavan teknologian prototyyppinä. Circuit Playground -kehitysalustaan on integroitu liiketunnistin sekä lämpötila-, valoisuus- ja äänisensorit. Lisäksi alustassa on kymmenen NeoPixels-lediä, minikaiutin vahvistimella, kaksi mikrokytkintä, liukukytkin sekä infrapunalähetin- ja vastaanotin. Laitteessa on USB-portti ohjelmointia ja virheiden etsintää varten. Ulkopuolista kommunikointia varten laitteessa on sarjaportti. (”Overview | Adafruit Circuit Playground Express | Adafruit Learning System”, n.d.)



Kuva 4. Adafruit Circuit Playground Express -kehitysalusta.

3.2.2 Adafruit Bluefruit LE UART Friend

Koska Circuit Playground -kehitysalusta ei sisällä Bluetooth yhteyttä, tarvitaan lisäksi Bluefruit LE UART Friend. Bluefruit LE UART Friend on Bluetooth low energy (BLE) -moduuli, jolla saadaan luotua langaton yhteys Android-puhelimen ja Circuit Playground - kehitysalustan välille.



Kuva 5. Adafruit Bluefruit BLE -Bluetooth-moduuli.

3.2.3 Adafruit-kehitysalustan ohjelmointi

Adafruit Circuit Playground Express -kehitysalustan käyttö on helppo aloittaa, vaikka kokemusta ohjelmoinnista ei olisi juuri lainkaan. Adafruit tarjoaa kolme eri ohjelmointitapaa alustalle. Helpoin tapa aloittaa ohjelmointi on Microsoft MakeCode for Adafruit -kehitystyökalulla. MakeCode on ilmainen, avoimen lähdekoodin kehitysalusta, joka perustuu koodilohkojen käyttämiseen. Kehitysalustaa on myös mahdollista ohjelmoida Python-kielillä. CircuitPython on Adafruitin MicroPython-ohjelmointikielystä jalostama kieli, joka on tarkoitettu mikrokontrollereiden ohjelmointiin. (Adafruit, 2019)

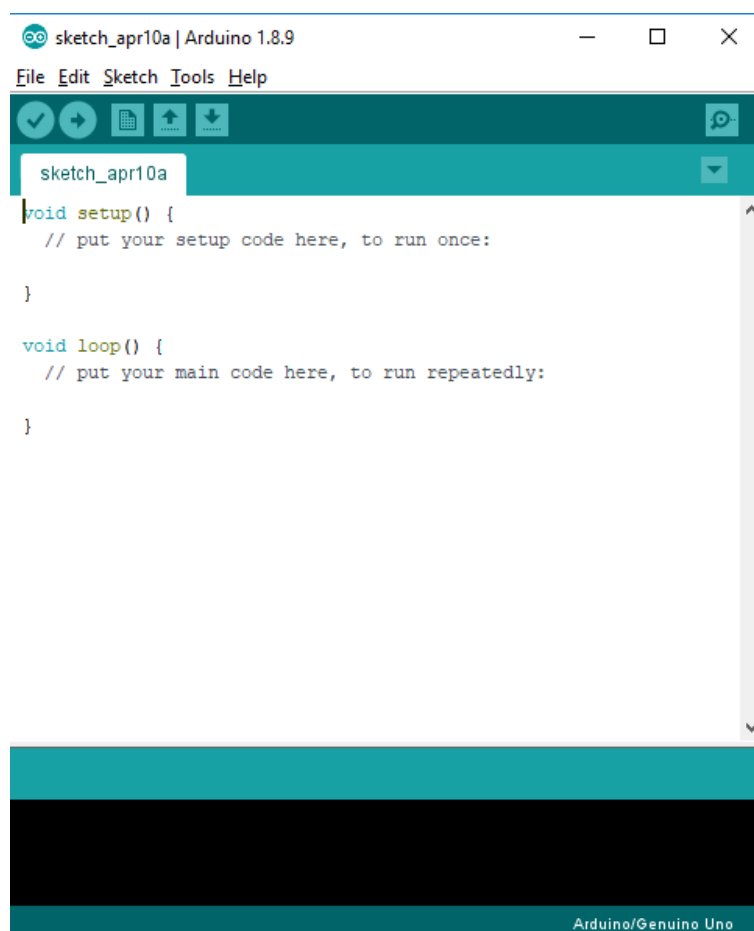
CircuitPython-kielen suosio vaikuttaa tällä hetkellä olevan nousussa ja Adafruit julkaisee lähes päivittäin siihen liittyviä videoita YouTube-kanavallaan.

3.2.4 Arduino IDE

Jotta Arduino-kehitysympäristö toimisi on sen mikrokontrollerille tallennettava ohjelma, joka huolehtii mm. kunkin I/O-nastan tilasta. Arduino IDE -ohjelmointiympäristö on tarkoitettu Arduinon ja vastaavien kehitysalustojen ohjelmakoodin kehittämiseen. Arduino-ohjelmointikieli perustuu Wiring-kieleen, molempien pohjana on C/C++-kielellä kirjoitetut kirjastot. Sekä Wiring- että Arduino-ohjelmistot (IDE) perustuvat Processing-ohjelmistokehykseen, joka on ohjelmoitu Javalla. Arduino kehitysalusta (ja muut yhteensopivat alustat) kytketään USB-portin kautta tietokoneeseen ja Arduino IDE -ohjelmointiympäristöön. Tarvittava ohjelmakoodi

kirjoitetaan Arduino IDE -ohjelmointiympäristössä ja ladataan kehitysalustan mikrokontrollerille. Mikrokontrollerille on valmiiksi asennettu bootloader-ohjelmisto, joka huolehtii koodin kirjoittamisesta Flash-muistiin. Ilman bootloader-ohjelmistoa mikrokontrollerin ohjelmointi pitäisi tehdä ulkoisella ohjelmointilaitteella. ("Arduino - Environment", n.d.)

Arduino IDE on ulkonäöltään ja toiminnoiltaan hyvin pelkistetty ohjelmisto. Siihen sisältyy melko rajoitetusti virheenjäljitysominaisuuksia. Debuggausta voidaan suorittaa lähinnä sarjamonitorin avulla. Editointi ei ole kovin intuitiivista, eikä se sisällä monista kehitysympäristöistä tuttuja ominaisuuksia kuten automaattista koodin täydennystä tai refaktorointia.



Kuva 6. Arduino IDE.

3.3 Android

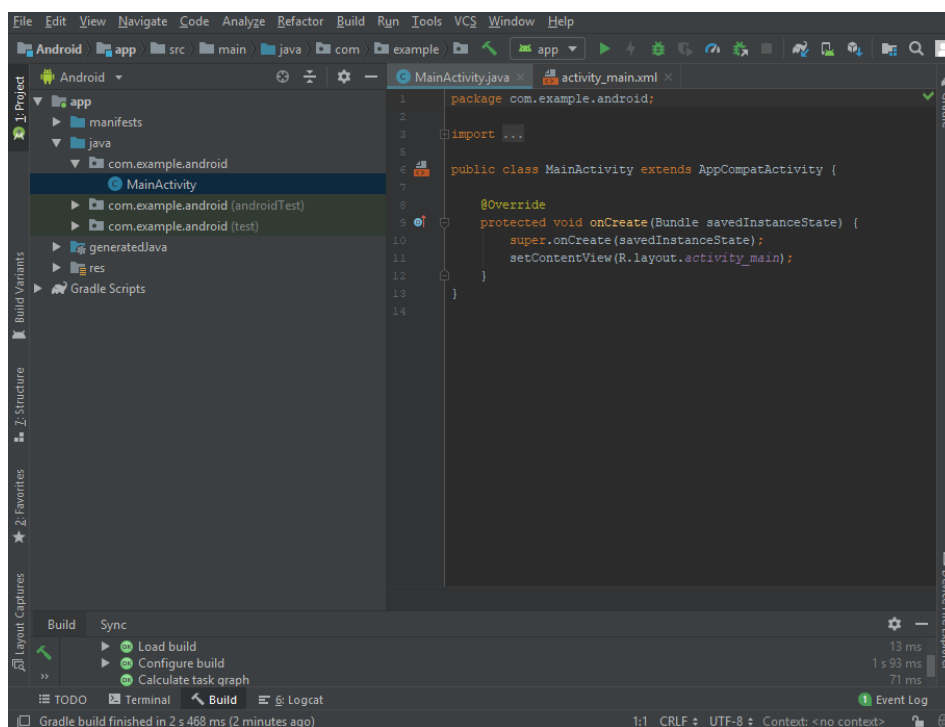
Vuonna 2005 Google oli suuntaamassa mobiilimarkkinoille ja osti Linux -käyttöjärjestelmään pohjautuvan Android-mobiilikäyttöjärjestelmän. Suurin osa Android-käyttöjärjestelmän koodista on julkaistu Apache-lisensillä, mahdollistaen käyttöjärjestelmän avoimen kehityksen. Android-käyttöjärjestelmä on saatavilla useille alustoille. Eri alustoille on saatavilla näille optimoidut järjestelmät kuten Wear OS, Android TV, Android for Cars ja Android Things. Android-sovellukset toimivat myös Chrome OS -laitteilla

kuten Chromebook-tietokone. Android-käyttöjärjestelmän uusin sovellus-versio on toukokuussa 2019 on koodinimeltään Pie ja versionumeroltaan 9. ("A Brief History of Google's Android, 11 Years Since Its Inception", n.d.)

3.4 Android Studio

Google julkaisi vuonna 2007 ennakkoversion Android kehityspakista (SDK). Vuonna 2009 julkaistiin ADT- eli Android Developer Tools -lisäosa Eclipse ohjelmointiympäristölle. Android Studio julkistettiin vuonna 2013 ja siitä tuli Googlen Android-käyttöjärjestelmän virallinen ohjelmointiympäristö. Android Studio -kehitysympäristössä ohjelmoidaan Java-kielillä (Yener & Dundar, 2016)

Android Studio erosi käyttötuntumaltaan esimerkiksi Microsoft Visual Studio -ohjelmasta ja ohjelman toimintojen opettelemiseen meni aluksi aikaa. Logiikka on kuitenkin pitkälti samanlainen, lähinnä Visual Studio -ohjelmasta tutut näppäinkomennot olivat erilaisia. Kätevä toiminto oli esimerkiksi puuttuvien luokkaviittauksien lisääminen "import"-listaan automaattisesti kirjoitettaessa koodia.

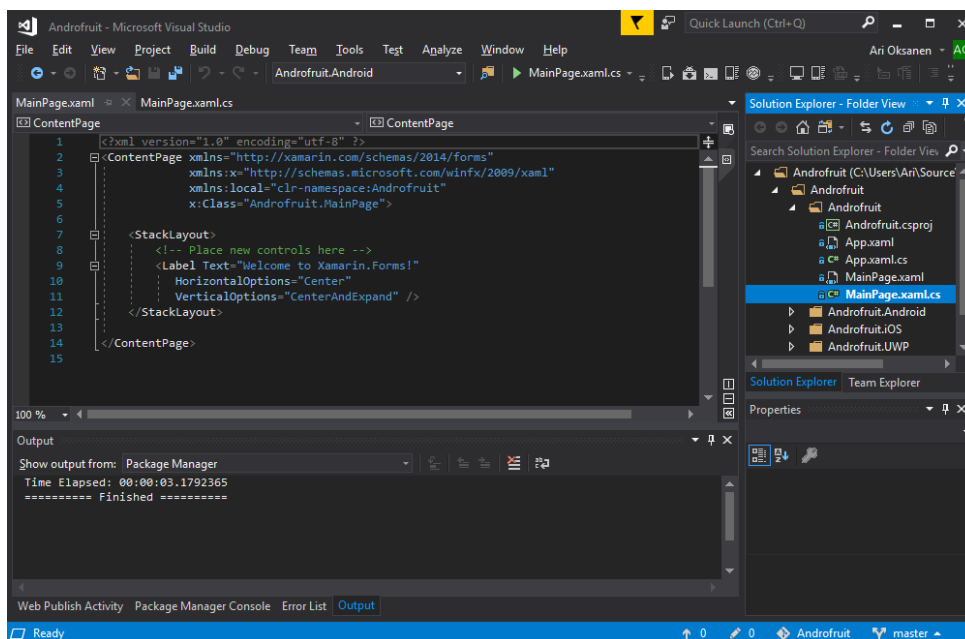


Kuva 7. Näkymä Android Studio -kehitysympäristön käyttöliittymästä.

3.5 Xamarin

Vuonna 2001 Miquel de Icazan ja Nat Friedman vaihtoivat Helix Code -yhtiönsä nimen Ximianiksi ja julkistivat kehittämänsä Mono-projektin, joka tarkoituksena oli tuoda C#-ohjelmointikieli ja .NET ohjelmistokehitys Linuxille ja muille käyttöjärjestelmille. Novell osti Ximianin vuonna 2003 ja

huhtikuussa 2011 se siirtyi Attachmate-yhtiön omistukseen. Attachmate ilmoitti kaupan jälkeen irtisanovansa suuren osan entisistä Novell-työntekijöistä ja Mono-projektin tulevaisuus oli vaakalaudalla. Saman vuoden toukokuussa Miquel de Icaza ilmoitti, että Monoa tulee kehittämään ja tukemaan juuri perustettu yritys Xamarin. Hetken Mono-projektin tulevaisuutta varjosti vielä se seikka, että Attachmate kehitti Monoa kaupallisena tuotteena ja Xamarinin kehittämästä Mono-projektista tulisi näin ollen suora kilpailija. Heinäkuussa 2011 Novell, Attachmentin tytäryhtiönä ja Xamarinin ilmoittivat Novellin myöntäneen Xamarinille pysyvän lisenssin Monolle, MonoTouchille ja Mono Androidille. (Li, 2019)

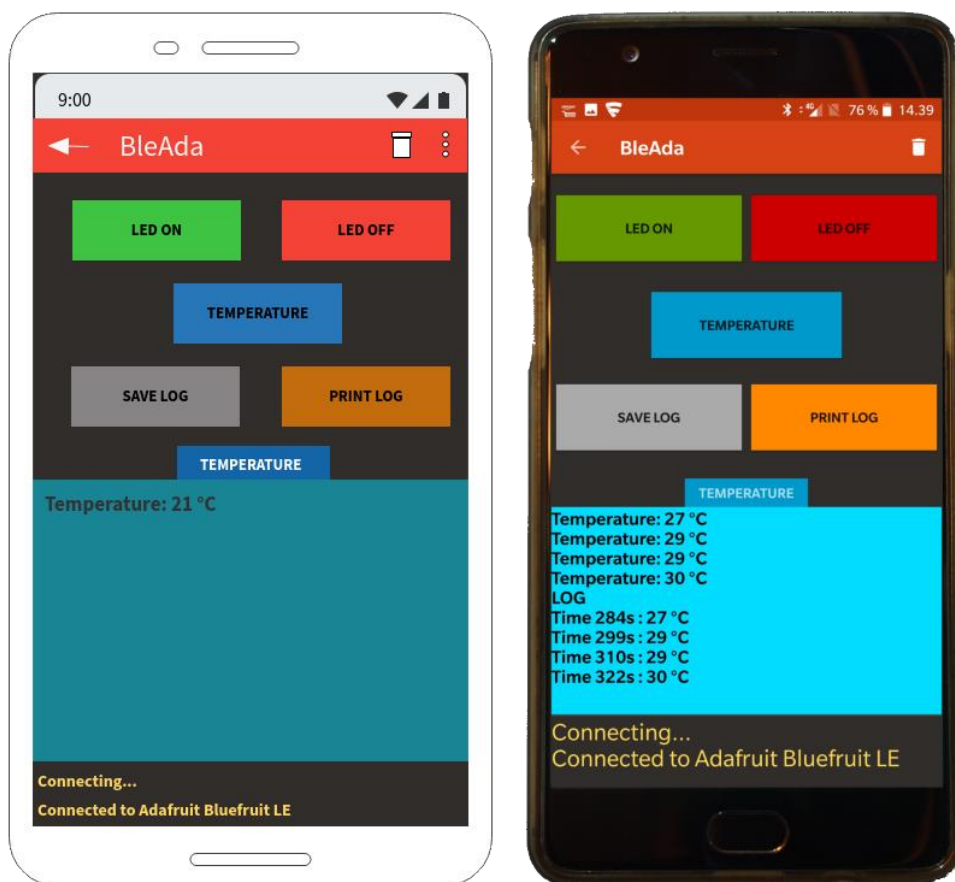


Kuva 8. Esimerkki Visual Studio 2017 Xamarin-projektista.

4 ANDROID- JA ARDUINO-SOVELLUKSIEN TOTEUTUS

Kehittämistyön tavoitteena on rakentaa ja ohjelmoida puettavan teknologian laite tallentamaan dataa sensoreilta mikrokontrollerille. Laite siirtää tiedon laitteen muistista langattomasti (Bluetooth) mobiililaitteelle. Lisäksi laitteen sensoreja on tarkoitus ohjata mobiililaitteella. Tiedon siirtoon ja sensorien ohjaukseen tarvittava sovellus on tarkoitus ohjelmoida Xamarin-ohjelmistokehyksellä ja C#-kielellä tai vaihtoehtoisesti Android Studio -kehitysohjelmistolla ja Java-kielellä. Itse laitteen mikrokontrollerin ohjelmisto ohjelmoidaan Arduino IDE -ohjelmistoympäristöllä ja C/C++-kieleen perustuvalla Arduino-kielellä. Ekosysteemi on skaalattavissa toimimaan Bluetooth Low Energy -moduulia käyttävien kehitysalustojen kanssa. Opinäytetyössä ei testattu toimivuutta eri valmistajien alustojen kanssa, mutta pienillä muutoksilla se on toteutettavissa Arduino-pohjaisten alustojen kanssa.

Suunnittelun aluksi luonnosteltiin Android-sovelluksen käyttöliittymä MockFlow -online palvelun Wireframe-editorilla (Kuva 9.). Sovelluksessa on viisi painiketta käskyjen lähettämiseksi Circuit Playground -alustalle ja tekstikenttä lämpötilatiedon vastaanottamiseksi ja lämpötila lokin tulostamiseksi.

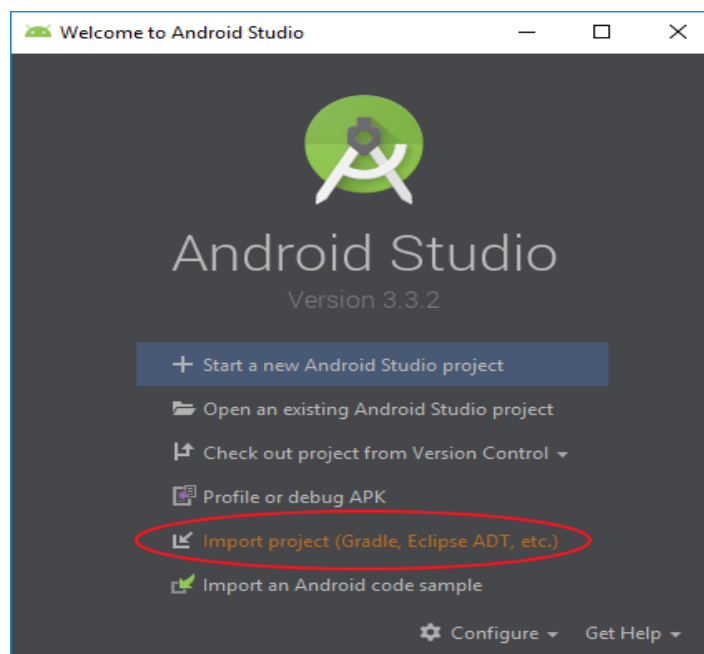


Kuva 9. Vasemmalla Wireframe-editorilla tehty luonnos. Oikealla valmis Android-sovellus.

4.1 Android-sovelluksen ohjelmointi

Rajallisen aikaikkunan vuoksi tässä työssä hyödynnettiin valmista ohjelmakoodia, joka mahdollistaa yhteyden muodostamisen Android-puhelimen ja Bluefruit Bluetooth-moduulin välille. Ohjelmakoodi on saatavilla osoitteessa <https://github.com/kai-morich/SimpleBluetoothLeTerminal/>. Koodi on lisensoitu MIT-lisenssillä, joka antaa käyttäjälle oikeudet vapaasti muokata, kopioida ja käyttää koodia omassa projektissaan. Koodia muokattiin tähän projektiin sopivaksi muun muassa lisäämällä painikkeita ja muuttamalla sarjadataan lähetykseen liittyviä toimintoja.

Ohjelman kehitys aloitettiin lataamalla ohjelmakoodi GitHub-sivustolta ja tuomalla se Android Studio -ohjelmaan aloitusruudussa olevan Import project -toiminnon avulla (Kuva 10.) Tuleva Android-ohjelma nimettiin samalla BleAda-nimiseksi, joka kuvaa paremmin käyttötarkoitusta.



Kuva 10. Android Studio: projektin tuominen ohjelmaan.

Ohjelmaan lisättiin kaksi painiketta, joilla ohjataan Circuit Playground -alustan led-valoja. Lisäksi lisättiin tekstikenttä ja kolmas painike, jota painamalla, ohjelma noutaa tekstikenttään kehitysalustalla sijaitsevan lämpöanturin ilmoittaman lämpötilan tekstikenttään. Jotta ohjelmalle saataisiin halutut painikkeet ja tekstikenttä, piti muuttaa fragment_terminal.xml-tiedostoa, jossa määritellään kehitysalustaa ohjaavan aktiviteetin ulkoasua kuvaavat elementit. Tiedostoa voidaan muuttaa muokkaamalla joko manuaalisesti editori-ikkunan Text-välilehdellä tai visuaalisesti Design-välilehdellä.

Ohjelmaan lisättiin Design-välilehdellä kolme painiketta (button) ja tekstikenttä (textview) sekä poistettiin kaksi ylimääräistä elementtiä.

```

<Button
    android:id="@+id/tempButton"
    style="@style/Widget.AppCompat.Button.Colored"
    android:layout_width="wrap_content"
    android:layout_height="70dp"
    android:layout_margin="5dp"
    android:layout_weight="1"
    android:background="@android:color/
    holo_blue_dark"
    android:text="@string/
    temperature_text" />

```

Ohjelmakoodi 1. Esimerkki painikkeen (button) koodista XML-tiedostossa.

Tämän jälkeen aktiviteetin toiminnallisuudesta vastaava TerminalFragment.java -tiedostoa muutettiin, jotta painikkeille saatiin vastaavat toiminnot. XML-tiedostossa on painikkeelle määritetty tunniste (id), johon viit- taamalla painikkeeseen päästään käsiksi Java-koodista. Kuvan 8. esimer- kissä olevan painikkeen tunnisterivi on "android:id="@id/tempButton". Painikkeeseen päästään Java-koodista käsiksi findViewById -metodilla. Ta- pahtumankäsittelijämetodi OnClickListener() rekisteröidään painikkeeseen ja "send"-funktion kautta lähetetään kehitysalustalle käsky "tempe- rature". Kehitysalustan koodissa kyseinen käsky tarkoittaa lämpötila-antu- rin arvon lähettämistä sarjaportin ja Bluetooth-moduulin kautta puheli- melle.

```

View tempButton = view.findViewById(R.id.tempButton);
tempButton.setOnClickListener(v -> send("tempera-
ture"));
return view;

```

Ohjelmakoodi 2. Esimerkki painikkeen toimintokoodista.

Android-sovellusta suunniteltaessa on otettava huomioon, mitä käyttöoi- keuksia sovellus tarvitsee Android-käyttöjärjestelmältä toimiakseen oi- kein. Tässä tapauksessa tarvittiin oikeudet käyttää Bluetooth-toimintoa ja karkeaa sijaintia. Oikeudet määritellään AndroidManifest.xml tiedostossa uses-permissions -elementissä.

```

<uses-permission            android:name="android.permis-
    sion.BLUETOOTH" />
<uses-permission            android:name="android.permis-
    sion.BLUETOOTH_ADMIN" />
<uses-permission            android:name="android.permis-
    sion.ACCESS_COARSE_LOCATION"/>

```

Ohjelmakoodi 3. Esimerkki käyttöoikeuksien määrittelystä.

4.2 Arduino-sovelluksen ohjelmointi

Adafruit Circuit Playground Express -kehitysalustan toimintojen ohjaami- seen käytettävä sovellus kehitettiin Arduino IDE -ohjelmointiympäristössä.

Pohjana käytettiin Adafruitin nRF51822 mikropiireillä varustetuille Bluefruit LE moduuleille tarkoitettua yhteyssovellusta. Sovellus on MIT lisenssin alainen, joten muokkaaminen ja käyttäminen projekteissa on mahdollista, sillä ehdolla, että lisenssin kuvaus säilytetään lähdekoodissa mukana. Ohjelma hoitaa yhteydenpidon ja sarjakomentojen välittämisen Adafruit Circuit Playground Express -kehitysalustan, Bluefruit Bluetooth-moduulin ja Android-sovelluksen välillä. Ohjelma hyödyntää BluefruitConfig.h kirjastoa bluetooth-asetuksissa ja FlashStorage.h kirjastoa lämpötila- ja aikatie-tojen tallentamiseksi Circuit Playground Expressin sisäiseen Flash-muistiin.

```
int count = 1;
int number_of_elements = 0;
String CurTemp = ("Temperature: ");
String LineEnd1 = (" \xC2");
String LineEnd2 = ("\xB0");
String LineEnd3 = ("C");
String elapsed;
typedef struct {
    int temperature[BUFSIZE+1];
    int elapsedTime[BUFSIZE+1];
} Temperature;

// Reserve a portion of flash memory to store an "int"
variable
// and call it "my_flash_store".
FlashStorage(my_flash_store, Temperature);
Temperature temperature_log;
unsigned long StartTime = millis();
```

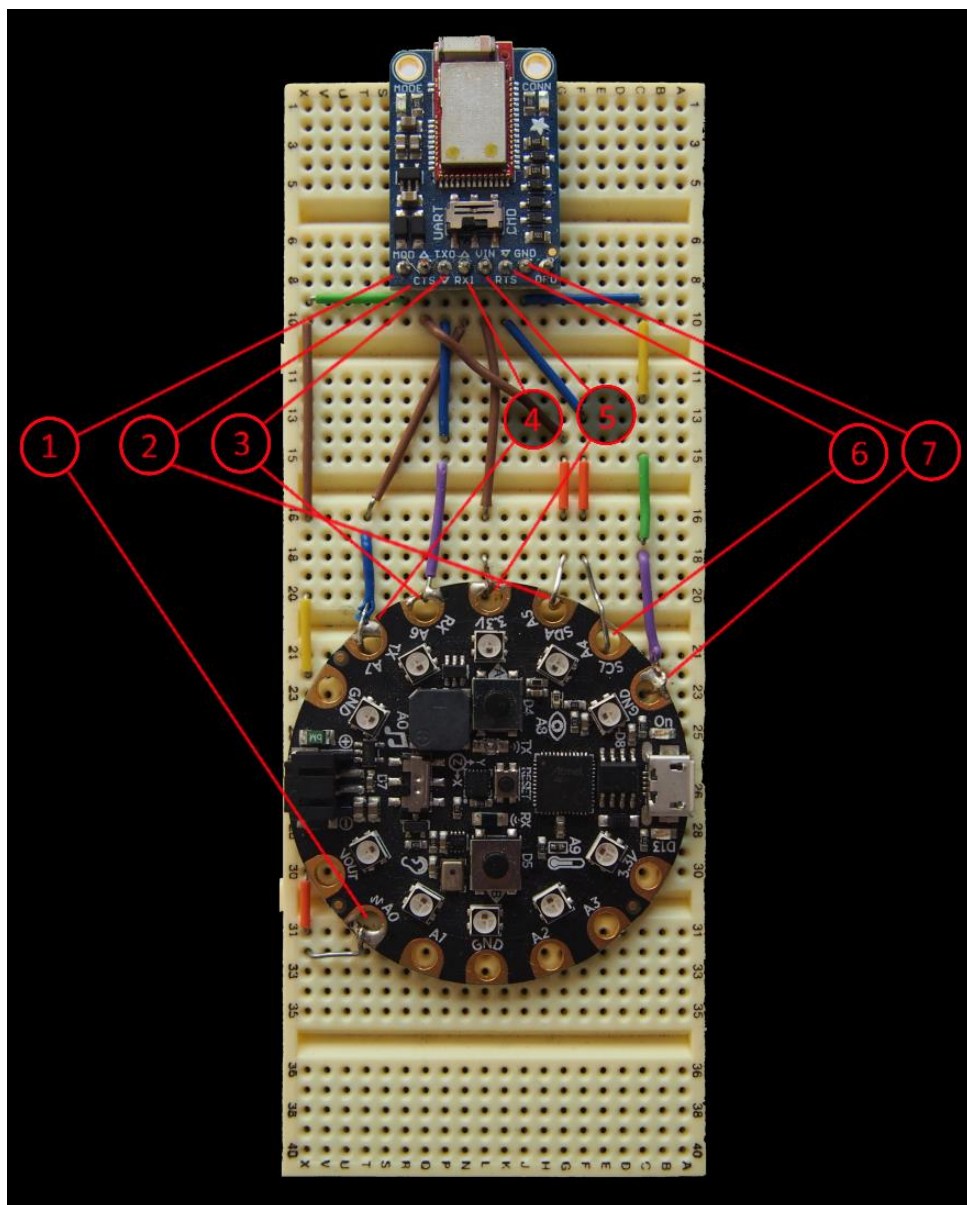
Ohjelmakoodi 4. Esimerkki muuttujien alustuksesta Arduino-koodissa.

5 LANGATON IOT-EKOSYSTEEMI

Ekosysteemi koostuu Adafruit Circuit Playground Express -kehitysalustasta, Adafruit Bluefruit LE UART Friend -Bluetooth moduulista ja Android-sovelluksesta. Android-sovelluksella otetaan yhteys Bluetooth-moduuliin. Tämä mahdollistaa sarjaliikenneyhteyden Android-sovelluksen ja Circuit Playground Express -alustan välille. Android-sovelluksesta voidaan yhteyden muodostamisen jälkeen ohjata alustan toimintoja ja vastaanottaa dataa.

5.1 Järjestelmän kytkentä

Järjestelmä kytketään koekytkentälevylle (Kuva 11.) taulukon (Taulukko 2.) mukaisesti:



Kuva 11. Järjestelmä kytkettynä koekytkentälevylle.

Taulukko 2. Järjestelmän liitinten kytkentä.

Kuvassa numero	Adafruit Circuit Playground Express -liitin	Bluefruit LE UART Friend -liitin
1	A0 / D12	MOD
2	A5 / SDA	CTS
3	A6 / RX	TXO
4	A7 / TX	RXI
5	3.3V	VIN
6	A4 / SCL	RTS
7	GND	GND

Seuraavassa tarkastellaan lähemmin Bluefruit-moduulin liittimien funktioita.

MOD-liittimen avulla valitaan missä tilassa UART-palvelua käytetään. UART-palvelulla on kaksi tilaa Data ja Command. Data-tilassa UART-yhteys toimii ”läpinäkyvänä” käytävänä Bluetooth-moduulin ja Android-sovelluksen välillä. Command-tilassa moduulin konfigurointiin käytetään modeemien ohjauksesta tunnettuja AT-komentoja. MOD-liitin kytketään Circuit Playground Express -alustan (jatkossa lyhennettynä CPX) digitaaliseen liitäntään D12.

CTS-liitin on tarkoitettu sarjaliikenteen vuonohjaukseen. Vuonohjauksen periaatteena on ruuhkatilanteiden syntymisen estäminen ja purkaa ne, mikäli niitä syntyy. CTS-liitäntän kautta CPX-alusta ilmoittaa, milloin sarjadataa voi lähettää Bluetooth-moduulilta TXO-liitäntän kautta CPX-alustalle. Mikäli käytetään ohjelmajohjaista sarjaliikenneyhteyttä, vuonohjaus tarvitaan. Vuonohjaus mahdollistaa sarjaliikenteen nopeamman ja varmemman toiminnan. CTS-liitin kytketään CPX:n digitaalisen liitäntään SDA.

TXO-liitintä käytetään Bluetooth-moduulilta kehitysalustalle kulkevan sarjamuotoisen datan lähettämiseen. Tämä liitin kytketään CPX:n sarjadataa vastaanottavaan RX-liittimeen.

RXI-liitintä käytetään Bluetooth-moduulilta kehitysalustalle kulkevan sarjamuotoisen datan vastaanottamiseen. Tämä liitin kytketään CPX:n sarjadataa lähettävään TX-liittimeen.

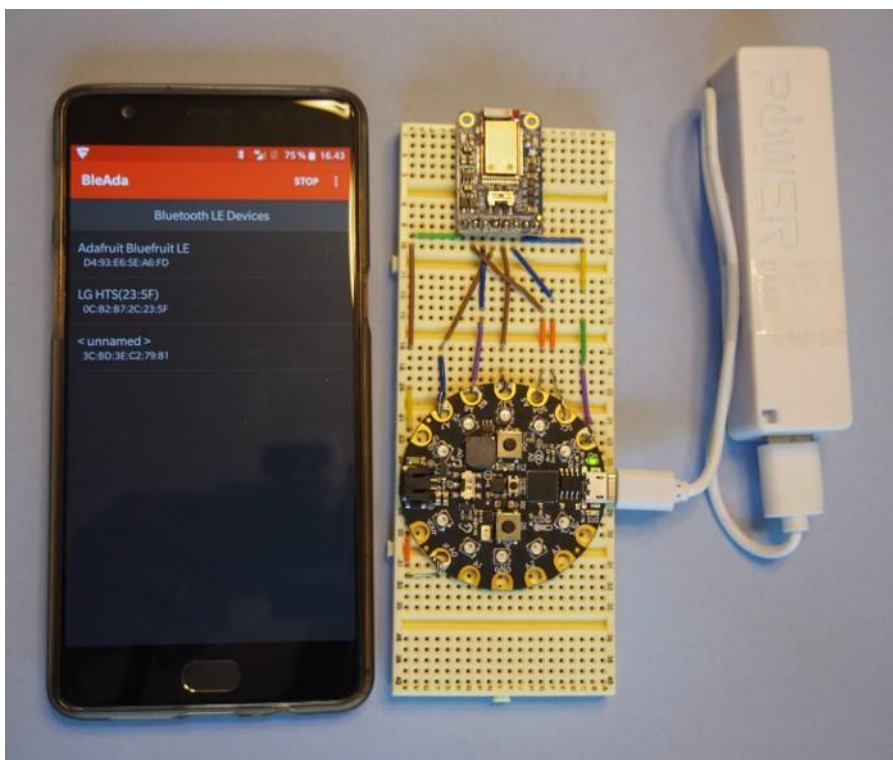
VIN-liittimen kautta Bluefruit-moduulille syötetään 3.3-16V käyttöjännite. RXI-liitintä käytetään mikrokontrollerilta Bluetooth-moduulille kulkevan sarjamuotoisen datan lähettämiseen. Tämä liitin kytketään CPX:n sarjadataa lähettävään TX-liittimeen.

RTS-liitin on tarkoitettu sarjaliikenteen vuonohjaukseen. RTS-liitännän kautta Bluetooth-moduuli ilmoittaa, milloin sarjadataa voi lähettää CPX-alustalta RXI-liitännän kautta Bluetooth-moduulille. RTS-liitin kytketään CPX:n digitaalisen liitäntään SCL.

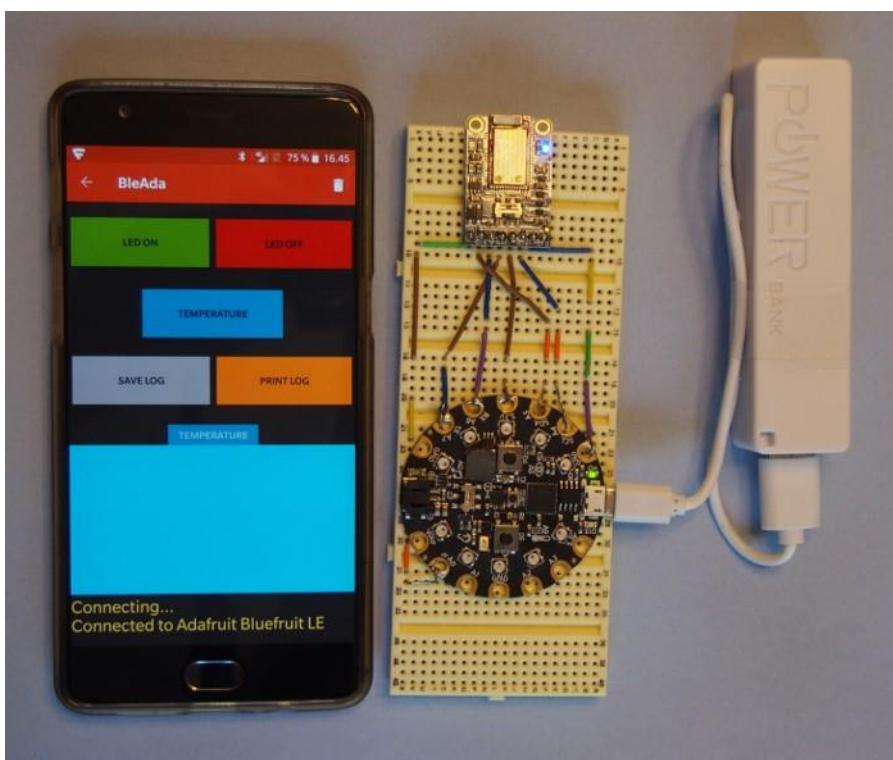
GND-liitin kytketään CPX-alustan nollapotentiaaliin eli maadoitusliittimeen GND.

5.2 Järjestelmän testaus

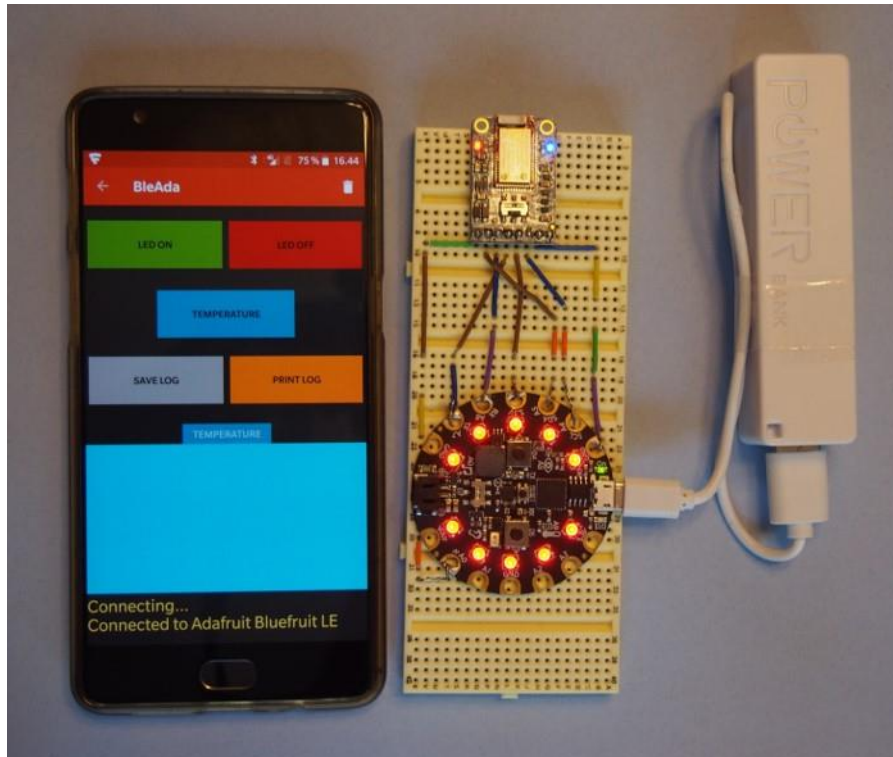
Circuit Playground Express -kehitysalusta liitettiin tietokoneeseen USB-kaapelilla. Arduino IDE -ohjelmalla tehty ohjelma (sketch) ladattiin Circuit Playground -alustan mikrokontrollerille Upload-painikkeella. Tämän jälkeen käynnistettiin sarjamonitorointi valikosta Tools -> Serial Monitor. Sarjamonitorista näkee Android-sovelluksen lähettämät komennot Circuit Playground -alustalle ja vastaavasti Circuit Playground -alustan lähettämät tiedot Android-sovellukselle. Tämä helpottaa vian hakua mahdollisissa viikatilanteissa. Android-puhelin liitettiin tietokoneeseen USB-kaapelilla. Android-puhelimen asetuksista kytkettiin Kehittäjäasetukset-valikko näkyviin ja valikosta USB-vianetsintä toiminto päälle. Android Studio -ohjelmasta puhelimelle tehty sovellus ajettiin valitsemalla valikosta Run -> Run 'app' toiminto. Tämän jälkeen avautui Select Deployment Target -ikkuna, josta valittiin USB-kaapelilla kytketty puhelin. Sovellus asentui puhelimelle ja puhelimesta kytkettiin Bluetooth ja sijainti toiminnot päälle. Sovelluksesta valittiin Scan-toiminto ja tulevalta Bluetooth-laitteiden listalta valittiin Adafruit Bluefruit LE -laite. Sovelluksesta testattiin tämän jälkeen lämpötilan nouto Circuit Playground -alustalta sekä alustan ledien sytyttäminen ja sammuttaminen. Bluetooth-yhteys muodostui nopeasti ja pysyi melko vakaasti päällä. Muutaman kerran yhteys katkesi itseksensä, mutta ilman kattavampia testauksia on mahdotonta sanoa, johtuiko se virheestä koodissa vai ulkoisista häiriötekijöistä, esimerkiksi kantoalueella olevasta muusta langattomasta liikenteestä. Jotta systeemin käyttökelpoisuus puettavan teknologian tarpeisiin tuli todettua, testattiin toiminta myös täysin langattomana. Kehitysalustaan liitettiin 2000mAh:n USB-varavirtalähde ja alusta sekä Android-puhelin irrotettiin tietokoneyhteydestä. Aiemman testauksen yhteydessä Android Studio asensi sovelluksen puhelimeen ja testaaminen voitiin toistaa täysin langattomasti. Kun Bluetooth-yhteys oli kytketty päälle, löydetyt yhteydet listautuivat puhelimen näytölle (Kuva 12). Listalta valittiin Adafruit Bluefruit LE -yhteys ja Android-sovellus muodosti yhteyden kehitysalustaan (Kuva 13). Android-sovelluksella testattiin kehitysalustan NeoPixels-ledien sytytys ja sammutus onnistuneesti (Kuva 14). Lämpötilan nouto kehitysalustalta sekä lämpötilan ja ajan listaus sovelluksen tulostusnäkyvään toimivat suunnitellusti (Kuva 15). Bluetooth-yhteyden kantavuus testattiin karkeasti viemällä puhelin kauemmaksi kehitysalustasta ja mittaamalla matka, jolloin yhteys katkesi. Matkaksi mitattiin 9,20 metriä suljetussa tilassa, kolmen puuseinän ollessa välissä.



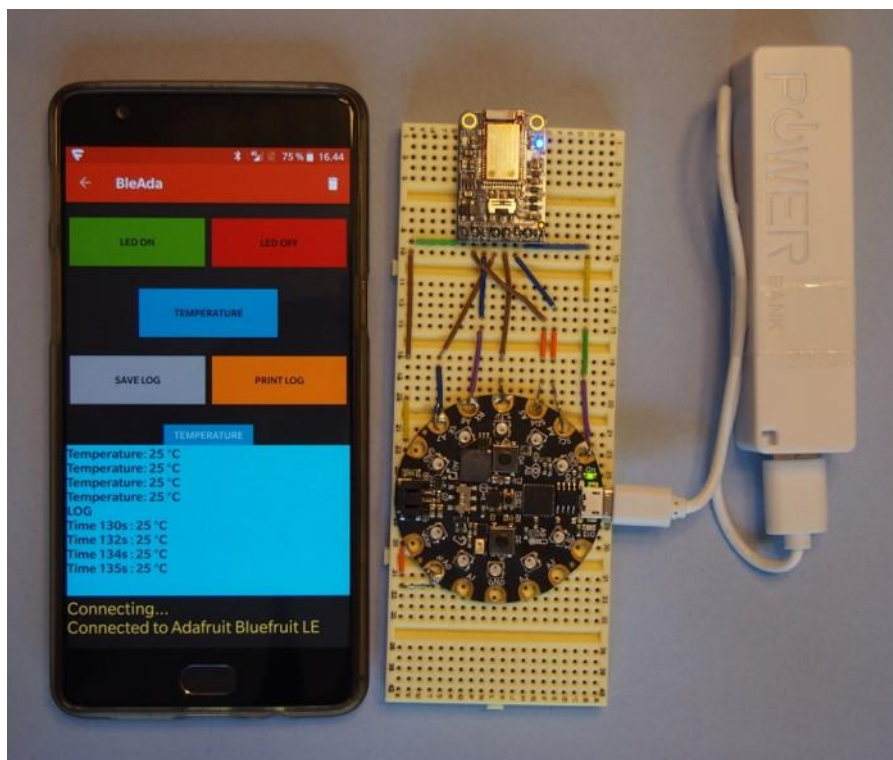
Kuva 12. Bluetooth-yhteyksien skannaus ja listaus.



Kuva 13. Sovellus käynnistetty, Bluetooth-yhteys luotu.



Kuva 14. NeoPixels-ledien sytytys.



Kuva 15. Lämpötilan nouto, tallennus lokiin ja tulostus.

6 YHTEENVETO

Opinnäytetyö vastasi kumpaankin tutkimuskysymykseen onnistuneesti. Langaton yhteys laitteiden välillä ja datan siirto kehitysalustasta mobiililaitteeseen saatiin toteutettua ja dokumentoitua. Lisäksi saatiin toteutettua datan tallennus paikallisesti, kehitysalustan Flash-muistiin. Myös sensorien ohjaus langattomasti mobiililaitteesta saatiin toteutettua ja dokumentoitua. Työtä tehdessä tuli vastaan monia asioita, kuten Arduino-kieli sekä Android Studio -ohjelmointiympäristö, joihin piti perehtyä syvällisemmin ja näin ollen opittua paljon uutta asiaa mobiili- ja IoT-sovellusten kehittämistä.

Opinnäytetyössä tuli vastaan joitakin ongelmia, jotka kuitenkin saatiin melko hyvin ratkaistua. Suurimpia haasteita oli Android-järjestelmän versioiden moninaisuus. Ohjelmoitaessa tuli eteen tilanteita, joissa Android Studio -ohjelmiston kääntäjä ilmoitti virheistä. Useimmiten virheet johtuivat ohjelmakoodissa käytettyjen kirjastojen ja kääntäjälle määritellyn Android-version yhteensopimattomuudesta. Myös ohjelmointikirjastojen ja ohjelmointikielien jatkuvasti kehittyessä, tulee eteen tilanteita, joissa metodeista tai luokista tulee vanhentuneita (deprecated) ja niiden käyttöä ei suositella. Koska Circuit Playground Express -kehitysalustaan on integroitu useita sensoreita ja toimintoja se vaikutti varteenotettavalta vaihtoehdolta ideoiden testaamiseen. Alustalle on saatavilla useita lisäosia, joiden avulla voi testata ja suunnitella esimerkiksi yhteyksien ottamista laitteeseen. Alusta ei ainakaan kaikilta osin sovi kriittisempään käyttöön, koska monien anturien kuten lämpötila-anturin tuottama data on melko epätarkkaa. Mikäli on tarve saada tarkempia mittaustietoja ympäristöstä, on käytettävä ulkoisia sensoreita. Lisäksi Adafruit Bluefruit LE UART Friend -Bluetooth moduulin RTS- ja CTS-liittimet on otettava käyttöön, sarjaliikenneyhteyden vakauden parantamiseksi.

Jatkossa järjestelmää voisi kehittää toteuttamalla tiedon tallennuksen Flash-muistille. Lisäksi voisi tehdä käyttöjärjestelmäriippumattoman sovelluksen, jolla järjestelmää voisi käyttää myös muilta alustoilta, kuten Windows ja Linux. Järjestelmän sensoreita voisi hyödyntää laajemmin ja testata Circuit Playground Express -alustan sopivuutta esimerkiksi kaatumisen havainnointiin. Järjestelmän komponentteja tilattaessa, ei puuttavan teknologian laitteeksi suunniteltua Flora Wearable Bluefruit LE Bluetooth-moduulia ollut saatavilla, joten oli tilattava Bluefruit LE UART Friend -moduuli. Flora Bluetooth -moduulin kanssa voisi tutkia lisää puuttavan teknologian mahdollisuuksia.

LÄHTEET

A Brief History of Google's Android, 11 Years Since Its Inception. (n.d.).
Noudettu 15.5.2019, osoitteesta
<https://www.digitaltrends.com/mobile/android-version-history/>

Arduino - Environment. (n.d.). Noudettu 5.5.2019, osoitteesta
<https://www.arduino.cc/en/Guide/Environment>

Arduino - Introduction. (n.d.). Noudettu 5.5.2019, osoitteesta
<https://www.arduino.cc/en/Guide/Introduction>

Dyn Analysis Summary Of Friday October 21 Attack | Dyn Blog. (n.d.).
Noudettu 14.5.2019, osoitteesta <https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/>

Ericsson. (2019). Ericsson Mobility Report. *Ericsson*, (November), 36.
Noudettu 9.4.2019, osoitteesta
<https://www.ericsson.com/assets/local/mobility-report/documents/2019/ericsson-mobility-report-world-economic-forum.pdf>

Godfrey, A., Hetherington, V., Shum, H., Bonato, P., Lovell, N. H., & Stuart, S. (2018). From A to Z: Wearable technology explained. *Maturitas*, 113, 40–47. Noudettu 24.7.2019, osoitteesta
<https://doi.org/10.1016/j.maturitas.2018.04.012>

Inside Adafruit, the Open-Source Manufacturing Maker Champions. (n.d.). Noudettu 21.4.2019, osoitteesta
<https://makezine.com/2017/06/13/open-source-ideals-engineering-genius-helm-adafruit-maker-revolution-manufacturing/>

Keskosten älyvaatteet ja 6 muuta suomalaista älytekstiiliuutuutta - Fab. (n.d.). Noudettu 20.8.2019, osoitteesta
<https://www.fablehti.fi/keskosten-alyvaatteet-ja-muut-alytekstiiliuutuudet/>

Lappeenrannasta varoittava esimerkki – ”Ei kannata tuudittautua siihen, että isoilta vahingoilta on vielä välttytty” | Tekniikkatalous. (n.d.).
Noudettu 14.5.2019, osoitteesta
<https://www.tekniikkatalous.fi/uutiset/lappeenrannasta-varoittava-esimerkki-ei-kannata-tuudittautua-siihen-etta-isoilta-vahingoilta-on-viela-valtytty/c8d10bd6-4066-35f2-a540-71d51da4b704>

Movesense - open wearable device platform for motion and biometrics. (n.d.). Noudettu 19.8.2019, osoitteesta <https://www.movesense.com/>

Nussey, J. (2013). *Arduino for Dummies*. John Wiley & Sons, Incorporated. Noudettu 15.4.2019, osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/detail.action?docID=1183913>

Overview | Adafruit Circuit Playground Express | Adafruit Learning System. (n.d.). Noudettu 5.5.2019, osoitteesta <https://learn.adafruit.com/adafruit-circuit-playground-express/>

Puettavan teknologian tarina päättyy | Satakunnan Kansa. (n.d.). Noudettu 20.8.2019, osoitteesta <https://www.satakunnankansa.fi/satakunta/puettavan-teknologian-tarina-paattyy-kankaanpaassa-201118264>

ReimaGO | Reima. (n.d.). Noudettu 20.8.2019, osoitteesta <https://www.reima.com/fi/reimago>

Vermesan, O., & Friess, P. (2012). *Internet of Things : Converging Technologies for Smart Environments*. Noudettu 4.4.2019 osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi>

Vermesan, O., & Friess, P. (2015). Building the hyperconnected society : Internet of things research and innovation value chains, ecosystems and markets. Noudettu 21.7.2019 osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi>

Wearable Technology | Myontec. (n.d.). Noudettu 20.8.2019, osoitteesta <https://www.myontec.com/>