

# **Valoa pilvestä –sensoriverkkoprojekti**

## **Arkkitehtuurikuvaus**

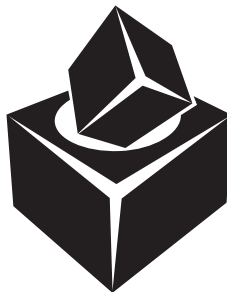
**Mika Ahonen**

**Heljä Ekman**

**Jukka Friman**

**Merika Peltola**

**Janne Uusitupa**



Versio: 2020–5–14

Julkinen

17. toukokuuta 2020

**Jyväskylän yliopisto**

**Tietotekniikan laitos**

**Jyväskylä**

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2020		
Tilaaaja	__.__.2020		
Ohjaaja	__.__.2020		

## Tietoa dokumentista

**Tekijät:**

- |                       |                              |
|-----------------------|------------------------------|
| • Mika Ahonen (MA)    | mika.k.ahonen@student.jyu.fi |
| • Heljä Ekman (HE)    | hememavu@student.jyu.fi      |
| • Jukka Friman (JF)   | jukkaf@gmail.com             |
| • Merika Peltola (MP) | merika.m.peltola@jyu.fi      |
| • Janne Uusitupa (JU) | janne.t.uusitupa@jyu.fi      |

**Dokumentin nimi:** Valoa pilvestä -projekti, Arkkitehtuurikuvaus

**Sivumäärä:** 6

**Tiedosto:** arkkitehtuuri.tex

**Tiivistelmä:** Dokumentti on osa Kokkolan yliopistokeskuksessa toteutettavaa sensoriverkkoprojektia. Sen tarkoitus on kuvata kehitettävän järjestelmän arkkitehtuuri kuvilla sekä sanallisesti.

**Avainsanat:** Azure, arkkitehtuuri, LoRa, NB-IoT, Scrumban, sensoriverkkoprojekti, XBee.

## Versiohistoria

Versio	Muutokset	Tekijät
2020–1–25	Dokumentin luonti, runko valmis.	HE
2020–3–30	Dokumentin täyttö ryhmän teksteillä	HE,MP,JU
2020–5–13	Kuvien lisäys, tekstien työstö	JF
2020–5–14	Dokumentin viimeistely, palautettava versio	HE
2020–5–17	Luottamuksellisuuden muutos	HE

## Tietoa projektista

Valoa pilvestä -projektissa kehitettiin jääkiekkokaukaloön automatisoitu valojen kytkentä. Projekti hyödynsi värinäsensoria ADXL362 kinnitettynä Dragino LoRa mini-piirilevyyn havaitsemaan liikettä kaukalossa. Viesti otetaan vastaan Microsoft Azure palvelussa, josta se edelleen välitetään XBee-NB-IoT-palikalke, joka kytkee kaukalolla valot päälle. Valot sammuvat tietyn ajan kuluessa, mikäli enempää liikettä ei havaita.

### Tekijät:

- |                       |   |
|-----------------------|---|
| • Mika Ahonen (MA)    | <code>mika.k.ahonen@student.jyu.fi</code> |
| • Heljä Ekman (HE)    | <code>hememavu@student.jyu.fi</code>      |
| • Jukka Friman (JF)   | <code>jukkaf@gmail.com</code>             |
| • Merika Peltola (MP) | <code>merika.m.peltola@jyu.fi</code>      |
| • Janne Uusitupa (JU) | <code>janne.t.uusitupa@jyu.fi</code>      |

### Tilaaaja:

- |                |                                    |
|----------------|------------------------------------|
| • Riku Immonen | <code>riku.j.immonen@jyu.fi</code> |
|----------------|------------------------------------|

### Ohjaajat:

- |                         |   |
|-------------------------|---|
| • Tuomo Härmanmaa       | <code>tuomo.harmanmaa@chydenius.fi</code> |
| • Veli-Matti Tornikoski | <code>veli.tornikoski@chydenius.fi</code> |



## Sisältö

<b>1</b>	<b>Suunnittelulähtökohdat</b>	<b>1</b>
1.1	Anturi . . . . .	1
1.2	Aktuaattori . . . . .	2
1.3	Backend . . . . .	2
<b>2</b>	<b>Järjestelmien osien tarkemmat kuvaukset</b>	<b>3</b>
2.1	Anturi . . . . .	3
2.1.1	Digita . . . . .	4
2.2	Aktuaattori . . . . .	4
2.3	Backend . . . . .	5

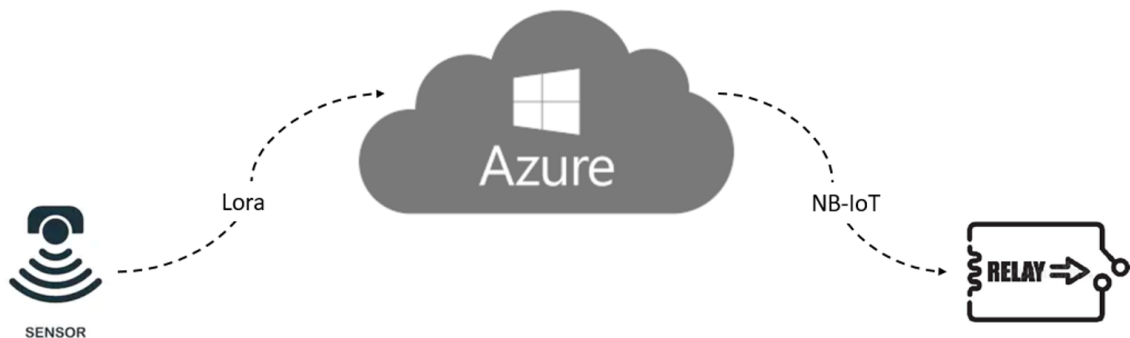
## Liitteet





# 1 Suunnittelulähtökohdat

Järjestelmän arkkitehtuuri on suunniteltu siten, että se mahdollistaa yksittäisten osajärjestelmän tai komponentin vaihtamisen ilman, että koko järjestelmää tarvitsee rakentaa uudelleen. Arkkitehtuurin lähtökohtana on ollut tukea jo olemassa olevia JKL-Open -projektin puitteissa toteutettuja sovelluksia ja ratkaisuja esimerkiksi backendin ja anturitoteutusten osalta. Projektissa toteutettu järjestelmä koostuu kolmesta osajärjestelmästä ja muodostaa kuvan 1.1 mukaisen kokonaisuuden.



Kuva 1.1: Järjestelmän arkkitehtuuri.

## 1.1 Anturi

Liikkeen tunnistamiseen voidaan käyttää erilaisia teknologioita. Tässä järjestelmässä liikettä havainnoidaan kiihtyvyytenä, mutta myös muut vaihtoehdot ovat mahdollisia liittää järjestelmään tai korvata nykyinen teknologia kokonaan muulla. Sensoreiden suunnittelussa on lähestytty siten, että jokainen sensori toimii itsenäisesti ja toimittaa tietonsa eteenpäin.

Sensoreiden tehtävänä on havainnoida liikettä jääkiekkokaukalossa ja mahdollisesti sen ympäristössä sekä lähettää tiedot taustajärjestelmään eli backendille. Sensoreiden tuottamaa dataa voidaan myös hyödyntää tarvittaessa muuhun kuin tämän projektin puitteissa suunniteltuun käyttöön. Sensorit eivät itse tee päätöksiä valojen ohjauksesta tai muista toimenpiteistä, joita niiden tuottamalla tiedolla tehdään.

Tiedonsiirto sensoreilta backendiin voidaan toteuttaa yleisimmillä IoT-alustojen tukemilla, pitkän kantaman teknologioilla. Tässä projektissa käytettiin LoRa- ja NB-IoT-teknologioita.

## 1.2 Aktuaattori

Aktuaattorin tehtävä on ohjata valoja saamansa syötteen mukaisesti. Järjestelmän äly sijaitsee backendissä, jonne yksi tai useampi sensori toimittaa tietoa kentän käytöstä, ja tämän tiedon perusteella aktuaattorille annetaan backendistä käsky sytyttää valot.

Käsky valojen kytkemisestä toimitetaan MQTT-viesteinä, mitä varten aktuaattorin tulee sisältää MQTT-asiakas ja tarvittavat verkkotoiminnallisuudet. Aktuaattorin tulee kuitata vastaanottamansa MQTT-viestit lähettäjälle, jotta backendiin saadaan tieto siitä, saiko aktuaattori käskyn perille vai yritetäänkö lähetystä uudelleen.

## 1.3 Backend

Järjestelmän äly sijaitsee backendissä. Se vastaanottaa sensoreilta tulevat viestit, tallentaa viestit tietokantaan myöhempää käyttöä varten sekä komentaa aktuaattoria sytyttämään valot. Backendin tulee myös sisältää mahdollisuus älykkyyden ohjelmointiin esimerkiksi useamman sensorin yhteisvaikutuksen huomioimiseksi takaisinkytkennän toteuttamisessa.

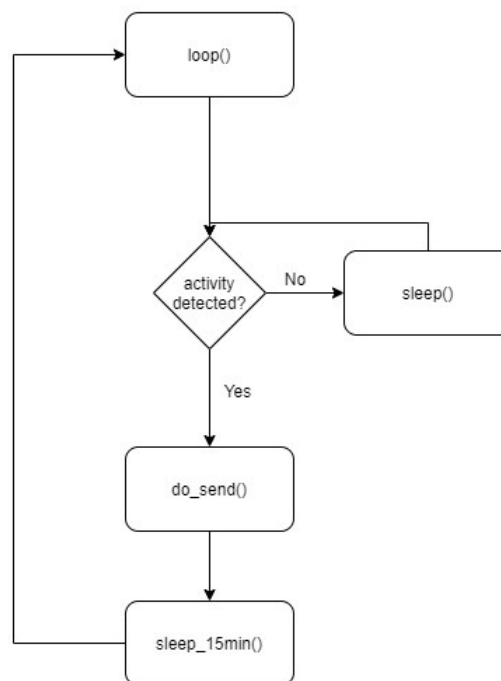
Backend tukee erilaisia liityntäteknologioita niin sensoreiden kuin aktuaattorin teknologiariippumattomuuden toteuttamiseksi. Tietoa tuottavia sensoreita tai aktuaattoreita voidaan liittää useita eri teknologioilla. Näin mahdollistetaan erilaisten sensoreiden ja aktuaattoreiden liittäminen järjestelmään. Tuettuja teknologioita ja protokollia ovat mm. LoRa, NB-IoT, HTTP/REST ja MQTT.

## 2 Järjestelmien osien tarkemmat kuvaukset

### 2.1 Anturi

Sensoriksi valittiin projektin aikana värinäsensori, joita voidaan liittää järjestelmään yksi tai useampia. Laite koostuu Draginon LoRa mini dev -laudasta ja ADXL362-värähtelyanturista. Dragino sisältää ATmega328P-mikrokontrollerin ja LoRa-radion, jonka mahdollistaa LoRaWAN-viestien lähetyksen. Mikropiiri on sama kuin Arduino Unossa, joten ohjelmointiin voidaan hyödyntää Arduinon IDE:tä ja kirjastoja.

ADXL362 on MEMS-teknologian kolmenakselinen kiihtyvyysanturi. Projektin järjestelmässä LoRaWAN-viestin payload koostuu anturin x-,y- ja z-akselien arvoista, anturin sisäisestä lämpötilasta sekä pariston varaustilasta. Näitä arvoja ei tämän projektin puitteissa käytetä, mutta myöhemmin akseliarvoja voitaisiin käyttää värähtelyn voimakkuuden laskemiseen. Lämpötilaa voidaan käyttää anturin kalibrointiin, sillä akseliarvot saattavat vaihdella lämpötilan mukaisesti. Pariston varaustilasta voidaan luoda hälytys, kun paristot on vaihdettava.



Kuva 2.1: Anturin toimintalogiikka.

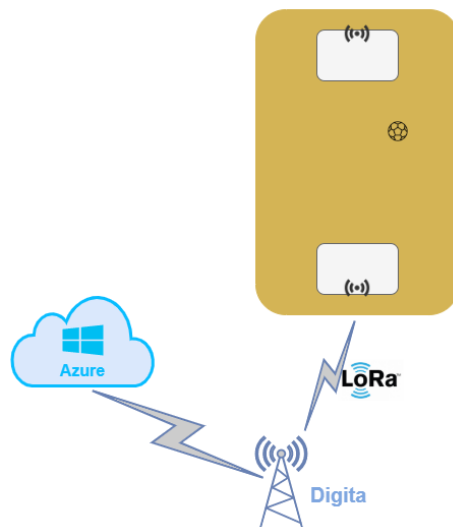
LoRaWAN-viestit lähetetään Digitan hallinnoimaan verkkoon. Mikropiiri kykenee antamaan myös tietoa akun varaustilasta, joten akun vaihtoon osataan varautua jo

etukäteen. Mikropiirin koodissa käytetään LoRaWAN-viestin lähettämiseen Imic-kirjastoa, jota käytetään usein Arduino pohjaisissa LoRa-ratkaisuissa.

### 2.1.1 Digita

Digita on Suomessa LoRa-verkoston ylläpitävä yritys, jonka antennit sijaitsevat ympäri Suomea. Digita toimii kansainvälisen The Thing Networkin (TTN) alaisena ja sen käyttämä portaali vastaa TTN:n portaalia. Järjestelmä ei siten ole sijainnistaan riippuvainen.

Sensorin lähettämä viesti otetaan vastaan Digitan antennilla. Jokainen laite rekisteröidään Digitan verkkoon, jotta oikeat viestit saadaan perille. Tämän jälkeen sensorin lähettämää LoRaWAN-viestejä voidaan katsoa Digitan palvelusta. Siellä näkyvät laitteesta perus tunnistetiedot, viestin lähetysaika, viestin sisältö sekä muita tietoja. Tiedot ohjataan erillisen profiilin kautta järjestelmän backendinä toimivaan Microsoft Azureen käyttäen hyödyksi HTTP POST -metodia ja Digitan järjestelmän omaa uudelleenohjausta.



Kuva 2.2: Anturi-osajärjestelmän yhteydet.

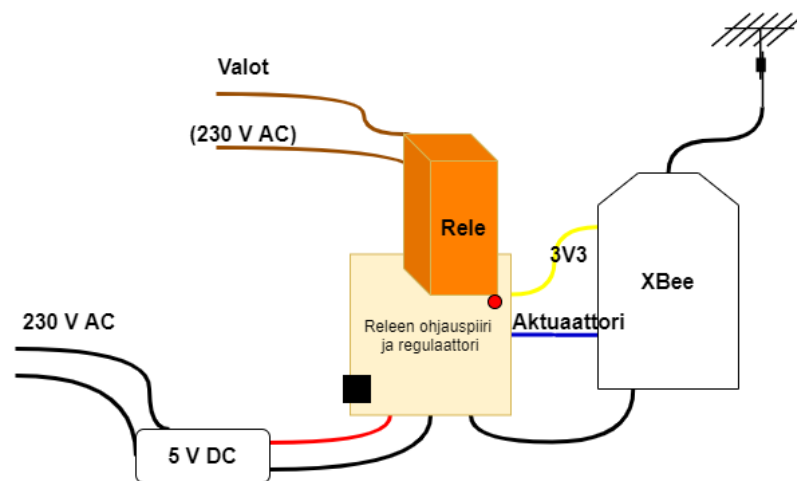
## 2.2 Aktuaattori

Projektissa toteutettu aktuaattori koostuu XBee-radiomoduulista (NB-IoT/LTE-M) ja releestä. Releen ja XBeen välissä on releenohjauspiiri, jossa on lisäksi regulaattori.

Aktuaattorikokonaisuus on kytkettynä verkkovirtaan.

Aktuaattori kytkeytyy NB-IoT-teknologialla Microsoft Azuressa olevaan backendiin MQTT-protokollalla. Aktuaattori kuittaa saamansa MQTT-viestin lähettäjälle, jotta backendiin saadaan tieto viestin vastaanottamisesta.

Rele sytyttää valot myös järjestelmän ulkopuolisen painonapin avulla. Releenä on kellorele, joka vastaanotettuaan pulssin sytyttää valot ja asettaa 30 minuutin ajastimen valojen sammumiselle. Valoja ei siis ole mahdollista sammuttaa järjestelmän avulla, eikä sammumisesta tai järjestelmän ulkopuolisesta painonapista myöskään saada tietoa backendiin tai tietokantaan.



Kuva 2.3: Aktuaattorin arkkitehtuuri.

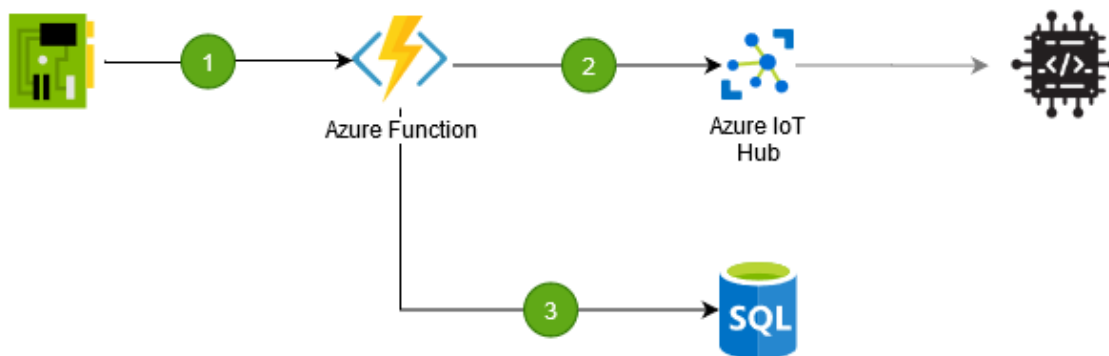
## 2.3 Backend

JKL-Open -projektissa on käytetty backendinä Microsoft Azure -ympäristöä, joten tämänkin järjestelmän backendinä päätettiin hyödyntää kyseistä alustaa. Azure tarjoaa IoT-järjestelmien tarpeeseen kattavan valikoiman tarpeellisia palveluita, joista voi koota tarvitsemansa kokonaisuuden.

Backendin varsinainen logiikka on toteutettu Azure Functions serverless -funktiolla. Funktio ottaa vastaan sensoreilta tulevan datan, tallentaa tiedot tietokantaan ja lähettää tietojen perusteella käynnistyssignaalin aktuaattorille. Funktio käynnistyy HTTP triggerillä kun Digitan backendin kautta saadaan HTTP POST -kutsu Function Appiin. Funktio hoitaa sensoreiden autorisoinnin vastaanottamansa JSON-vies-

tin sisällön perusteella. Digitalta tulevat kutsut tulevat suoraan Function Appin julkiseen internetrajapintaan, eikä siinä käytetä IOT Hubia lainkaan. Sensorien lähettämä data tallennetaan Azure SQL -tietokantaan.

Aktuaattorin kanssa kommunikointiin käytetään Azure IOT Hubia ja sen MQTT-toteutusta. Valojen ohjaukseen käytetään Direct Method tyyppistä metodia, joka toteuttaa pyyntö-vastaus-interaktion pilven ja laitteen välillä niin, että kutsuun saadaan välitön vastaus onnistumisesta tai epäonnistumisesta. Tämä metodi ei sisällä MQTT-aiheen jonotustoimintoa, jolloin mahdollisten yhteysongelmien tms. takia ei jonoon jääneitä viestejä pureta yhteyden palattua ja mahdollisesti turhaan sytytetä valoja.



1. HTTP Trigger käynnistää Funktion
2. Funktio lähettää IOT HUB:n avulla MQTT-viestin XBeelle
3. Sensorin lähettämä data tallennetaan tietokantaan

Kuva 2.4: Backendin arkkitehtuuri.