



Student's name and surname: Anita Dworzańska

ID: 165400

Undergraduate studies

Mode of study: Full-time studies

Field of study: Informatics

Profile: Computer Systems Architecture

Student's name and surname: Jan Komińczuk

ID: 160502

First cycle studies

Mode of study: Full-time studies

Field of study: Informatics

Profile: Teleinformatics

ENGINEERING DIPLOMA PROJECT

Title of project: Embedded system using Bluetooth Low Energy sensors for smart farming applications

Title of project (in Polish): System wbudowany oparty na sensorach Bluetooth Low Energy do zastosowań w rolnictwie precyzyjnym

Supervisor	Head of Department
<i>signature</i>	<i>signature</i>
dr inż. Łukasz Kulas	

Date of project submission to faculty office:



**DECLARATION regarding the diploma project titled:
Embedded system using Bluetooth Low Energy sensors for smart farming
applications**

First name and surname of student: Anita Dworzańska
Date and place of birth: 05.02.1997, Gdańsk
ID: 165400

Faculty: Faculty of Electronics, Telecommunications and Informatics
Field of study: informatics

Cycle of studies: undergraduate
Mode of study: Full-time studies

Aware of criminal liability for violations of the Act of 4th February 1994 on Copyright and Related Rights (Journal of Laws 2018, item 1191 with later amendments) and disciplinary actions set out in the Act of 20th July 2018 on the Law on Higher Education and Science (Journal of Laws 2018 item 1668 with later amendments),¹ as well as civil liability, I hereby declare that the submitted diploma project is my own work.

This diploma project has never before been the basis of an official procedure associated with the awarding of a professional title.

All the information contained in the above diploma project which is derived from written and electronic sources is documented in a list of relevant literature in accordance with art. 34 of the Copyright and Related Rights Act.

I confirm that this diploma project is identical to the attached electronic version.

Gdańsk,

.....
signature of the student

¹ The Act of 20th July 2018 on the Law on Higher Education and Science:

Art. 312, section 3. Should a student be suspected of committing an act referred to in Art. 287 section 2 items 1–5, the rector shall forthwith order an enquiry.

Art. 312, section 4. If the evidence collected during an enquiry confirms that the act referred to in section 5 has been committed, the rector shall suspend the procedure for the awarding of a professional title pending a judgement of the disciplinary committee and submit formal notification on suspicion of committing a criminal offence.



ABSTRACT

The main goal of this Bachelor of Engineering project titled *Embedded system using Bluetooth Low Energy sensors for smart farming applications* is to create a prototype of a system consistent with Agriculture 4.0 concept using Bluetooth Low Energy (BLE) technology. Developed solution shall be easy in implementation and its main functionality shall be periodic gathering of data from environmental sensors and displaying it on a dedicated web page.

A market research was performed at the beginning of the project. It focused on products already on the market that were similar to our goal. Knowledge about existing solutions was used to develop an innovative product idea and one that hopes to stand out among competition. The target system shall be a viable solution for small private greenhouses, small farms, as well as big agriculture companies.

The main focus of this work is the developed prototype. Multiple stages of development allow for verification of requirements set by product design and system models.

The prototype is developed using existing development platform solutions. An XDK (Cross Domain Development Kit) device manufactured by Bosch was used as a sensor board and a single-board computer - Raspberry Pi 3 B+ - as a hub. Both devices have a built-in BLE controller which was used in the project. The idea of the system is to use multiple sensor boards placed across a certain area to monitor environmental changes and deliver the measurements to a hub which analyzes and displays the data.

Successful development allowed for performing a field test to check basic functionalities and how the system behaves in different sensor board placement scenarios. Data gathered by sensor boards and then transferred to the hub is seen on a web page which proves the correct operation of the prototype and success of the project. Additionally gathered signal strength measurement in different scenarios allowed for noticing certain shortcomings of the prototype and form possible improvement for next iteration of the product prototype.

Keywords: Agriculture 4.0, smart farming, Bluetooth Low Energy, BLE, Internet of Things, IoT

OECD Fields of Science and Technology: Engineering and technology, Electrical engineering, electronic engineering, information engineering



Research leading to these results has received funding from the EU ECSEL Joint Undertaking under grant agreement no. 783221 (project AFarCloud) and from the National Centre for Research and Development on behalf of the Ministry of Science and Higher Education in Poland.



Gefördert durch:



aufgrund eines Beschlusses des Deutschen Bundestages

FUNDING PROVIDED BY THE FEDERAL MINISTRY FOR ECONOMIC AFFAIRS AND ENERGY

STRESZCZENIE

Celem pracy inżynierskiej pod tytułem *System wbudowany, oparty na sensorach Bluetooth Low Energy do zastosowań w rolnictwie precyzyjnym* jest stworzenie działającego prototypu systemu dla Rolnictwa 4.0 przy użyciu technologii Bluetooth Low Energy (BLE). Tworzony system ma być łatwy do zastosowania zarówno na bardzo małych, jak i rozległych przestrzeniach, a jego główną funkcjonalnością jest cykliczne zbieranie danych z sensorów i wyświetlanie ich w czytelny sposób na stronie internetowej.

Głównym przedmiotem rozważań pracy inżynierskiej jest tworzony prototyp. Podczas tworzenia wczesnej wersji systemu, twórcy mogą się upewnić, że użyte przez nich technologie umożliwiają zaimplementowanie wymagań przewidzianych podczas procesu kreowania modeli systemu, które przedstawiają ogólny zarys użyteczności oraz szczegółowy opis funkcjonalności prototypu.

Prototyp tworzony jest z użyciem kilku gotowych już elementów. Zostało użyte urządzenie XDK firmy Bosch jako płytki sensorów i Raspberry Pi jako jednostka zarządzająca. Oba urządzenia posiadają wbudowany moduł BLE. Prototyp, tak jak i docelowy produkt, składa się z wielu płytek sensorów rozłożonych na przestrzeni i jednostki głównej. W prototypie urządzenie centralne przemieszczane jest z użyciem robota, zapewniając mobilną funkcjonalność systemu.

Udane stworzenie prototypu systemu pozwoliło na przeprowadzenie testów w środowisku naturalnym, badając zachowanie układu przy różnym rozmieszczeniu sensorów i jednostki głównej względem siebie. Na stronie internetowej, widoczne są zebrane a następnie przeanalizowane przez jednostkę zarządzającą dane, co potwierdza funkcjonalność prototypu, której osiągnięcie było najważniejszym celem tworzonego systemu. Testom podlegały również poboczne cechy produktu, których analiza wskazała pewne ograniczenia wydajnościowe wymagające przyszłej poprawy.

Słowa kluczowe: Rolnictwo 4.0, rolnictwo precyzyjne, Bluetooth Low Energy, BLE, Internet rzeczy, IoT

Dziedzina nauki i techniki zgodna z OECD Nauki inżynierskie i techniczne, Elektrotechnika, elektronika i inżynieria informatyczna



Research leading to these results has received funding from the EU ECSEL Joint Undertaking under grant agreement no. 783221 (project aFarCloud) and from the National Centre for Research and Development on behalf of the Ministry of Science and Higher Education in Poland.



Gefördert durch:



aufgrund eines Beschlusses des Deutschen Bundestages

FUNDING PROVIDED BY THE FEDERAL MINISTRY FOR ECONOMIC AFFAIRS AND ENERGY

STRESZCZENIE

Celem pracy inżynierskiej pod tytułem *System wbudowany, oparty na sensorach Bluetooth Low Energy do zastosowań w rolnictwie precyzyjnym* jest stworzenie działającego prototypu systemu dla Rolnictwa 4.0 przy użyciu technologii Bluetooth Low Energy (BLE). Tworzony system ma być łatwy do zastosowania zarówno na bardzo małych, jak i rozległych przestrzeniach, co zapewnia szerokie grono odbiorców. Główną funkcjonalnością systemu jest cykliczne zbieranie danych z sensorów i wyświetlanie ich w czytelny sposób na stronie internetowej, tak by użytkownik miał do nich łatwy dostęp.

Na samym początku pracy nad produktem przeprowadzone było rozeznanie już istniejących na rynku systemów Agroturystyki 4.0, w celu jak najdokładniejszego wypełnienia luki produktowej podyktowanej zapotrzebowaniem zwiększonej precyzji w automatyzacji rolnictwa. Wiedza na temat istniejących rozwiązań ma ogromne znaczenie podczas projektowania produktu innowacyjnego i mającego wyróżniać się na tle innych systemów dzięki zawartym funkcjonalnościom. Docelowy produkt będzie znajdował swoje zastosowanie zarówno w prywatnych szklarniach, małych gospodarstwach rolnych jak i w dużych przedsiębiorstwach agrarnych. Łatwość jego obsługi dodatkowo poszerza grono odbiorców, którzy nie muszą posiadać specjalistycznej wiedzy by z systemu korzystać.

Najbardziej charakterystycznym, choć opcjonalnym elementem systemu jest dron, dzięki któremu ulokowana na nim jednostka zarządzająca (hub) jest w stanie szybko i sprawnie przemieszczać się pomiędzy płytkami sensorów nawet, gdy są one bardzo oddalone od siebie. Mimo, iż głównym podmiotem rozważań pracy inżynierskiej jest tworzony prototyp, to znajduje się w niej dokładny opis planowanego produktu docelowego. To na jego podstawie projektowany jest prototyp, którego stworzenie ma na celu upewnienie się, że zamodelowany system działa poprawnie i zgodnie z założeniami. Podczas tworzenia wczesnej wersji produktu, twórcy mogą się upewnić, że użyte przez nich technologie umożliwiają zaimplementowanie wymagań przewidzianych podczas procesu kreowania modeli systemu, które przedstawiają ogólny zarys użyteczności oraz szczegółowy opis funkcjonalności prototypu. Nie brakuje również diagramów aktywności ani sekwencji, które prezentują oficjalną charakterystykę produktu, a ponadto pełnią rolę dokumentacji, która w trakcie prac nad wytwarzanym systemem jest wskazówką dla programistów, ułatwiającą podejmowanie ostatecznych decyzji dotyczących implementacji oprogramowania.

Prototyp tworzony jest z użyciem kilku gotowych już elementów, które zostają przeprogramowane lub uzupełnione o dodatkowe funkcjonalności w celu dostosowania ich do działania w roli podkomponentu systemu. Płytką sensorów zostało urządzenie XDK firmy Bosch. Jest to nowy produkt z dziedziny Internetu Rzeczy w ofercie producenta, który cechuje przede wszystkim osiem wbudowanych sensorów i moduł BLE. Jednostką zarządzającą (hub) jest Raspberry Pi, który również posiada moduł BLE. Prototyp, tak jak i docelowy produkt, składa się z wielu płytek sensorów rozłożonych na przestrzeni, którą użytkownik chciałby monitorować i tylko jednej jednostki głównej, która jest sercem systemu. W prototypie urządzenie centralne przemieszczane jest z użyciem robota, który w wersji prototypowej zastępuje drona, zapewniając mobilną funkcjonalność systemu.

Udane stworzenie prototypu systemu pozwoliło na przeprowadzenie testów w środowisku naturalnym, nielaboratoryjnym, badając zachowanie układu przy różnym rozmieszczeniu sensorów

i jednostki głównej względem siebie w różnych odległościach. Na stronie internetowej, udostępnianej przez jednostkę zarządzającą, widoczne są zebrane a następnie przeanalizowane przez nią dane. Ich obecność potwierdza funkcjonalność prototypu, której osiągnięcie było najważniejszym celem tworzonego systemu. Testom podlegały również poboczne cechy produktu, których analiza wskazała pewne ograniczenia wydajnościowe wymagające poprawy dla uzyskania płynności działania.

Słowa kluczowe: Rolnictwo 4.0, rolnictwo precyzyjne, Bluetooth Low Energy, BLE, Internet rzeczy, IoT

Dziedzina nauki i techniki zgodna z OECD Nauki inżynierskie i techniczne, Elektrotechnika, elektronika i inżynieria informatyczna



Research leading to these results has received funding from the EU ECSEL Joint Undertaking under grant agreement no. 783221 (project AFarCloud) and from the National Centre for Research and Development on behalf of the Ministry of Science and Higher Education in Poland.



Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages

FUNDING PROVIDED BY THE FEDERAL MINISTRY
FOR ECONOMIC AFFAIRS AND ENERGY

CONTENTS

1. Introduction [AD]	17
1.1. Motivation	17
1.2. Preliminary Assumptions and the Purpose of the Work	17
1.3. Document Structure	18
2. Market Research [AD + JK]	19
2.1. Existing Products	19
2.1.1. Cesens [AD]	19
2.1.1.1. System overview	19
2.1.1.2. Sensors and functionalities	19
2.1.1.3. Communication	20
2.1.1.4. Configuration	20
2.1.2. OneSense [JK]	20
2.1.2.1. System overview	20
2.1.2.2. Sensors and functionalities	20
2.1.2.3. Communication	21
2.1.2.4. Configuration	21
2.1.3. SmartCrop System [AD]	21
2.1.3.1. System overview	21
2.1.3.2. Sensors and functionalities	21
2.1.3.3. Communication	22
2.1.3.4. Configuration	22
2.2. Product Comparison [JK]	22
3. System Architecture [AD + JK]	25
3.1. Proposed System Idea	25
3.1.1. System overview [AD]	25
3.1.2. Sensors and functionalities [JK]	26
3.1.3. Communication and data presentation [AD]	27
3.1.4. Configuration [AD]	27
3.1.5. Comparison [JK]	28
3.1.6. Bill of materials [JK]	29
3.2. System Model [AD + JK]	29
3.2.1. Use case diagram [AD]	29
3.2.1.1. Formal description of use cases	32
3.2.2. Activity diagram [AD]	36
3.2.2.1. Initialization of a hub	36
3.2.2.2. Initialization of a sensor board	37
3.2.2.3. Gathering data process	37
3.2.2.4. Exchanging data between hub and sensor board	38
3.2.2.5. Providing data on website	39

3.2.3. Sequence diagram [JK]	39
3.2.3.1. Hub	39
3.2.3.1.1. Initialization of a hub	39
3.2.3.1.2. Display website	40
3.2.3.1.3. Connect to sensor board	41
3.2.3.1.4. Receive data and store	41
3.2.3.1.5. Save data to storage	42
3.2.3.2. Sensor Board	42
3.2.3.2.1. Configure	42
3.2.3.2.2. Read sensor values	42
3.2.3.2.3. Save data	43
3.2.3.2.4. Send data via BLE	43
4. System Prototype Development [AD + JK]	45
4.1. System Overview [AD]	45
4.1.1. Initial threats	46
4.1.2. Restrictions	46
4.2. Hardware Components [JK]	46
4.2.1. Bosch XDK	46
4.2.1.1. Specification	47
4.2.1.2. Sensors and functionalities	47
4.2.1.3. Configuration	48
4.2.2. Raspberry Pi [JK]	48
4.2.2.1. Specification	48
4.2.3. Data flow	48
4.3. Software Components [JK]	49
4.3.1. Data presentation	50
4.3.2. Bosch XDK	50
4.3.2.1. Architecture	50
4.3.3. Data formats	51
4.3.4. Tinydb	52
4.3.5. Flask	52
4.4. Communication [JK]	53
4.4.1. Bluetooth Low Energy (BLE)	53
4.4.1.1. Bluetooth	53
4.4.1.2. Differences	53
5. System Tests and Measurements [AD + JK]	55
5.1. Functional Test [JK]	55
5.1.1. Testing process	55
5.1.2. Results	56
5.2. Measurements in Operational Environment [JK]	56
5.2.1. Test environment [AD]	58
5.2.2. Results [AD]	59
5.2.3. Summary [JK]	59

6. Summary [AD + JK]	61
6.1. Goals Achieved [AD]	61
6.2. The Biggest Challenges [AD]	61
6.3. Future Plans [AD]	61
6.4. Personal View on the Topic	62
6.4.1. Anita Dworżańska	62
6.4.2. Jan Komińczyk	62
References	65

LIST OF FIGURES

3.1	System Overview	26
3.2	Use Case Diagram of the Proposed System	30
3.3	Activity Diagram - Initialization of a Hub	36
3.4	Activity Diagram - Initialization of a Sensor Board	37
3.5	Activity Diagram - Gathering Data Process	37
3.6	Activity Diagram - Exchanging Data Between Hub and Sensor Board	38
3.7	Activity Diagram - Providing Data on Website	39
3.8	Sequence Diagram - Initialization of Hub	40
3.9	Sequence Diagram - Display Website	40
3.10	Sequence Diagram - Connect to Sensor Board	41
3.11	Sequence Diagram - Receive Data and Store	41
3.12	Sequence Diagram - Save Data to Storage	42
3.13	Sequence Diagram - Configure	42
3.14	Sequence Diagram - Read Sensor Values	42
3.15	Sequence Diagram - Save Data	43
3.16	Sequence Diagram - Send Data via BLE	43
4.1	Diagram of Prototype System Functionalities	45
4.2	Bosch XDK110	47
4.3	Raspberry Pi 3 Model B+	48
4.4	Diagram of Data Flow in the System	49
4.5	Diagram of Software Blocks in the System	49
4.6	Sensor Board - system architecture	50
4.7	BLE Message Data Format	51
4.8	Hub Database Model	52
5.1	Functional Test - test set	55
5.2	System in Action	56
5.3	Test Cases - diagrams	57
5.4	Test Set-up: Raspberry Pi	58
5.5	Test Set-up: Bosch XDK	58
5.6	Average BLE Signal Strength	59



LIST OF TABLES

2.1	Market Research - Product Comparison	23
3.1	System Architecture - Product Comparison	28
3.2	Sensor Board - Bill of Materials	29
3.3	Hub - Bill of Materials	29
5.1	Acceptable Signal Strengths	59

ABBREVIATIONS

API Application Program Interface. 20, 21, 50, 51

BLE Bluetooth Low Energy. 1, 3, 5, 8, 11, 25, 27, 28, 31, 33–35, 38, 43, 45–51, 53, 55, 56, 59, 61–63

IDE Integrated Development Environment. 61

IoT Internet of Things. 17, 18, 26, 46, 53, 62

RSSI Received Signal Strength Indicator. 56, 59

SDK Software Development Kit. 47, 49, 50, 61, 62

UML Unified Modeling Language. 18, 29, 61

XDK Cross Domain Development Kit. 1, 3, 5, 8, 46, 47, 49–51, 53, 55, 56, 58, 59, 61, 62

1. INTRODUCTION [AD]

1.1. Motivation

One of the most futuristic and at the same time very interesting technology field is the Internet of Things (IoT). It brings a lot of opportunities to optimize everyday tasks, simplify complex procedures and, finally, increase perceived convenience of our lives. What we are passionate about the most is that to create an IoT device is relatively simple and requires only basic knowledge of electronics and programming, thus enabling development ranging from an elementary to the most sophisticated system.

Aggregate Farming in the Cloud (AFarCloud) is a research and development project (R&D) on an agriculture topic. It aims to create a distributed platform for autonomous farming integrated with management software, both of which will support monitoring mechanisms enabling the decision-making process to be much easier than ever before. The project consortium popularizes the agriculture trend by creating multi-functional robots able to communicate with each other or with other devices and applications. Ideas and main challenges of the AFarCloud are impressive and intriguing. The proposed problem solving approach encouraged us to follow them in the agriculture product manufacturing and contribute to the Agriculture 4.0 [1] community by bringing to life our own vision of a smart product.

Agriculture 4.0 is not yet very popular in Poland. It is the future of farming achieved through use of IoT and big data concepts which together deliver a substantial amount of functionalities to the client [2]. Since the beginning of this century agriculture started becoming smarter, using automated procedures for improving farming precision and focusing all operation power on development of future benefits. Farming fields, with the abundance of measurable variables could generate massive amounts of data every day and everything that needs to be done is to gather and analyze the data to benefit the market. Nowadays devices exceed humans in some tasks enabling the people to achieve more with the same or even smaller level of involvement; for example alerting about draught or pest invasion. Through smart data use farmers can better understand their growing practices and know which actions will generate the greatest business value [3]. Agriculture 4.0 is a term describing a smart and efficient industry that transforms big data and new technologies into benefits for the whole supply chain.

A big vision of possibilities encouraged us to create our own system guided by a trend of Agriculture 4.0. Without any experience with system design, only with passion for programming and electronics we developed a system, from a model to a ready-to-use prototype providing all of basic concepts we planned which were determined as needed in order to be competitive with already existing products on the market.

1.2. Preliminary Assumptions and the Purpose of the Work

The goal of our Bachelor of Engineering Thesis is to model and implement a system, based on inexpensive IoT devices, as an Agriculture 4.0 solution. The system could be used to increase efficiency in agriculture use cases. Primary features of the product are gathering real life data

from installed sensors with the use of an autonomous robot and presenting them to a user in a convenient and intuitive way. We took the challenge of creating something we have never done before and did not have any experience with. Because of the system complexity the target product will be presented only as a future idea but to prove effectiveness of our solution a prototype was built. It includes basic functionalities, which can be upgraded in future version of the product, but are enough to compose a working system. Beyond the established technical acceptance criteria the most important part of the project to us is to gain adequate experience in creating IoT systems, from the market research and system design to a performing product.

1.3. Document Structure

The following document contains information about process of Agriculture 4.0 product creation, additionally presenting our point of view, problems we encountered, challenging situations and experience we gained.

Following this introduction the second section of the document focuses on market research to determine validity of our preliminary assumptions and find points of focus for our product idea. Understanding market trends and the needs of farming industry is necessary to appropriately adjust parameters and functionalities of the created product. The section presents the idea of the final product, its main features and a predicted bill of materials.

The third section is devoted to the prototype, starting with detailed description of its system model, showing business and functional view on the project. UML (Unified Modeling Language) diagrams are included to introduce a general idea of specific implementation plans of the designed system. We also describe the hardware and software components, presenting various system features and used technologies.

Going further, we present how the prototype behaves in reality. Including photos from field tests we demonstrate the results of our final prototype version. Additionally we show outcome of a field test we performed and ideas for further development and testing.

The fifth section concludes all work which was needed to plan, implement and test the created system. Detailed description containing issues we encountered when developing the prototype of the planned product. At the end each author of the project summarizes their own point of view on the subject, lists their input to the project, challenges and gained experience.

Initials [AD](Anita Dworżańska) and [JK](Jan Komińczyk) were placed next to section titles to show which one of the authors worked on that paragraph.

2. MARKET RESEARCH [AD + JK]

In this chapter we present 3 existing products designed for use in smart agriculture and our own idea for a system. Each device's description consists of an overview, a description of its functionalities, a way of communication with other devices and configuration options. At the end we compare existing products with ours and present a simple bill of materials.

2.1. Existing Products

2.1.1. Cesens [AD]

2.1.1.1. System overview

Cesens Mini [4] is an agro-climatic information system for decision making support in agriculture. The main part is a sensing node, able to have a maximum of 4 simultaneously connected sensors chosen from 11 different sensors types, whose main functionality is data collection from the field, especially wineries. Small solar panel on the top of the device helps keep the battery life longer (up to 20 years), which minimizes the maintenance requirements.

Cesens system is made of two elements, the agro-climatic stations and the Cesens Cloud Platform. The stations gather data and send them to the platform, which processes them and returns analyzed information such as possible risk of plant infection or need of watering the field.

Product weighs about 1kg and measures 62 x 90 x 120 mm. Additionally, device has a connector for an external antenna.

2.1.1.2. Sensors and functionalities

The device has been designed to support up to 11 different sensors like:

- humidity sensor
- pressure sensor
- temperature sensor
- pluviometer
- dew point sensor
- anemometer
- water flow / consumption sensor
- tensometer or light sensor.

Using cloud computing power to analyze data and processes it through interactive graphs, comparing data between stations and those of previous years, the system is able to monitor multiple parameters such as:

- soil water potential
- soil volumetric water content
- leaf wetness, temperature

- relative humidity and atmospheric pressure
- soil temperature
- dendrometry
- wind speed and direction
- rainfall
- solar radiation.

Additionally, stations generate their own energy which makes the system less dependent on user.

2.1.1.3. Communication

The system uses Sigfox ultra-narrowband network for uplink and downlink communication, which reduces energy consumption while sending data remotely. Collected data is sent automatically to the Cesens Cloud Platform.

User can easily check gathered information via a dedicated application or webpage, which shows various data stored on the Cesens Cloud Platform, such as: temperature, humidity, daily rainfall, water usage. The application also contains a notification system informing the user about emergencies, i.e. rainfall, strong wind, very high or low temperature.

2.1.1.4. Configuration

Manufacturer does not provide any information about the initial configuration process of the Cesens Mini system. The manufacturer provides an API which can be used to integrate agroclimatic information through various users' applications.

2.1.2. OneSense [JK]

2.1.2.1. System overview

Product created by Evvos, OneSense Environment and Agriculture [5] is a compact, battery-powered wireless sensor data transmitter. Product weighs 200g and measures 115 x 65 x 40 mm. It is fully weatherproof.

2.1.2.2. Sensors and functionalities

OneSense supports several interfaces and sensors, and is capable of acquiring and processing data from multiple sensors. The manufacturer lists different types of supported sensors:

- soil moisture sensor
- soil temperature sensor
- air temperature sensor
- rain gauge
- relative humidity sensor
- wind speed sensor

- water meter
- soil volumetric water content sensor
- soil electrical conductivity sensor
- leaf wetness sensor
- IR thermometer
- feels like temperature sensor
- dew point sensor
- pulse counter
- flow meter
- pyranometer.

Additionally, OneSense allows a connection of an ON/OFF state or door OPEN/CLOSED circuit to control various other systems based on current environmental data.

2.1.2.3. Communication

Collected sensor data is transmitted wirelessly through the Sigfox network to a cloud service for further processing. User is able to access the data on the cloud, as well as make use of provided API to integrate the sensor information in their own applications. The cloud service can be set up to relay the data so that the user's application is able to receive the data instead of having to download it from Evvos' platform.

2.1.2.4. Configuration

The manufacturer claims the device's firmware can be flashed with use case specific firmware. Evvos cloud interface API is provided so sensor data can be easily integrated in the user's own application or portal.

2.1.3. SmartCrop System [AD]

2.1.3.1. System overview

Smartfield Base Station is a product created for collecting a variety of data from other systems in the company's lineup. That includes SmartCrop sensor [6] and SmartWeather [7]. The base station sends the gathered data to CropInsight [8] - a cloud based application - for storage and further analysis.

2.1.3.2. Sensors and functionalities

Smartfield Base Station has three types of sensors, it can measure:

- ambient temperature
- relative humidity
- rainfall amount.

This station is the base for the SmartCrop, which is a system intended for gathering plants' parameters, based on infrared technology. Measuring canopy (aboveground portion of a plant) temperature by infrared thermometer is the main feature, which provides precise information about plant's health. The system reads data every minute and sends an average report every 15 minutes using radio.

SmartCrop System is made of SmartCrop Sensors; each sensor is a device used to perform infrared measurements of temperature. Creators of the system allow for connecting up to 4 sensors at once. Non obligatory, but providing continuous plant's canopy temperature collection is a component called SmartWeather. This stand-alone tool gathers a variety of weather condition data points and summarizes plant environment parameters providing analyzed information. It can measure different parameters collected, at the end, by the Smartfield Base Station. It measures parameters as below:

- wind speed
- wind direction
- solar radiation
- barometric pressure in conjunction with the ambient temperature
- relative humidity
- rainfall.

2.1.3.3. Communication

Receiving crop data is possible via emails, text alerts or CropInsight account, to which the stand-alone components send logs to. CropInsight is an online web service used for presenting gathered data on readable charts, generating reports and changing settings, as well as creating alerts for any kind of measurement.

2.1.3.4. Configuration

Manufacturer provides information about the number of supported sensors, which depends on the configuration option, but does not explain the details of this process.

2.2. Product Comparison [JK]

There is a substantial amount of smart agriculture products currently on the market. They focus on supporting different types of agriculture and most of them feature similar qualities that fall into the Agriculture 4.0 paradigm. We chose three products which we consider to be a good representation of the range of features currently available solutions are able to provide.

Cesens and OneSense are ready-to-use, all-in-one products which incorporate non-removable onboard sensors connected to the main module. It is a very good solution for small-scale cultivation where only few monitoring devices are needed or in cases where a client wants a ready-to-use solution and does not want the hassle of buying many different modules which then create a whole system. The downside to such an approach is the limited amount of sensors which can be connected at one time and the fact that such systems are not very scalable. That can, in turn, raise costs when there is a need of expansion. When there is a need to gather a lot of data from many

different sensors a product like Smartfield can be used. It is able to interface with many different sensors on one agricultural area. It is easy to notice that every product uses more than three sensors and most often two of them are a thermometer and a moisture sensor.

Products like Smartfield, which consist of multiple modules are complex, which makes them expensive and difficult to use, but provide a standardized in Agriculture 4.0 solution of distributed system. Products, such as Smartfield, always implement a hub - a device which controls all the modules and collects data (gathered earlier from modules' sensors). Every product provides a way to access the gathered information, in most cases by means of a cloud-based web application which stores the data, analyzes it and shows statistics. The hub sends raw gathered data to the cloud; in some cases it also pre-analyzes it.

All smart agriculture products are focused on power-efficiency and long, uninterrupted operation. In the majority of cases they are used in remote locations, that is why they employ a battery and occasionally a solar panel to charge the battery throughout the day in order to lengthen the unattended operation time.

Despite the large amount of products found on the market information about them is scarce or hard to find. Manufacturers do not always include dimensions and weight of their products in the description; or how to install them. Available documentation is most often just brochures and the way the communication with a potential customer is carried out suggests that the products are targeted at big companies or corporations, and not agriculturalists with know-how. Moreover, researched products are directed at large fields or plantations and in a lot of cases it would not be feasible to use them in small-scale greenhouses, orchards or gardens. It was found that there is an untapped market for small solutions directed at individuals who want to monitor smaller agricultural areas or in-door environments. Table 2.1 shows a comparison of the products.

Table 2.1: Market research - product comparison

Product Name	Cesens	OneSense	SmartCrop
Maximum number of connected sensors in one sensor board	4	3	N/A
Number of types of sensors	11	17	>5
Can only use built-in sensors	no	no	yes
Whole system in one device	yes	yes	no
System uses hub device	no	no	yes
Connection with cloud	yes	yes	yes
Communication with cloud	Sigfox	Sigfox	cellular network
Analyzed data available in cloud	yes	yes	yes
Waterproof	yes	yes	yes
Power source	solar panel	battery	solar panel and a battery backup

3. SYSTEM ARCHITECTURE [AD + JK]

3.1. Proposed System Idea

3.1.1. System overview [AD]

After comparing different products, noticing their advantages as well as flaws we have devised a product which will fill a gap of small-scale sensor solutions on the market, will be low cost and easy to configure for every technical or non-technical user.

At the heart of the system there is a hub located on a robot or a drone (Figure 3.1), which enables the system to work even in vast and mountainous areas. Movement of the hub across the working area ensures that the measurement results are stored locally or sent directly to the operating center. Usage of a drone in the system is optional and the default variant is based on a static hub located in the middle of measurement area.

The hub creates an 802.11 based wireless network. Users are able to connect to the hub using devices with built-in WiFi and display dedicated systems' website on it. Hub also provides an option to connect to an existing wireless network and offer its services in it instead. Those must be configured by editing a configuration file on a microSD card inserted into the hub (this option is recommended for technical users only or could be provided to the client by the manufacturer on a special demand). Hub provides a web-based interface with various types of information. It presents raw gathered data, analyzed data and also suggestions about present or future plant's needs. Data is presented in various ways to adjust to user preferences. The elementary view consists of time dependent graphs of sensor values and heat maps of all sensor types visualizing data changing across the measured area.

The second end of the system's workflow is represented by many sensor boards scattered throughout the area over which user wants to have environment monitoring. Sensor boards store data from different built-in sensors in their own memory (microSD card). Each sensor type has individual period of measurement which can be configured by a user via the website. Stored data is sent to the hub using Bluetooth Low Energy protocol (Figure 3.1), thanks to a built-in BLE module in both endpoint devices (hub and sensor board). Hub periodically gathers data from sensor boards and processes it. In each data collection cycle the hub initiates a connection with a board, receives data and stores it.

The main target group of the system is users who possess agriculture areas, from small greenhouses to wide and spatial crop fields. Non-stationary hub is able to work on non-planar and trees covered areas without limits of arable field size. This functionality expands the group of system recipients, from private hobbyists or home gardeners to huge industrial crop farming companies.



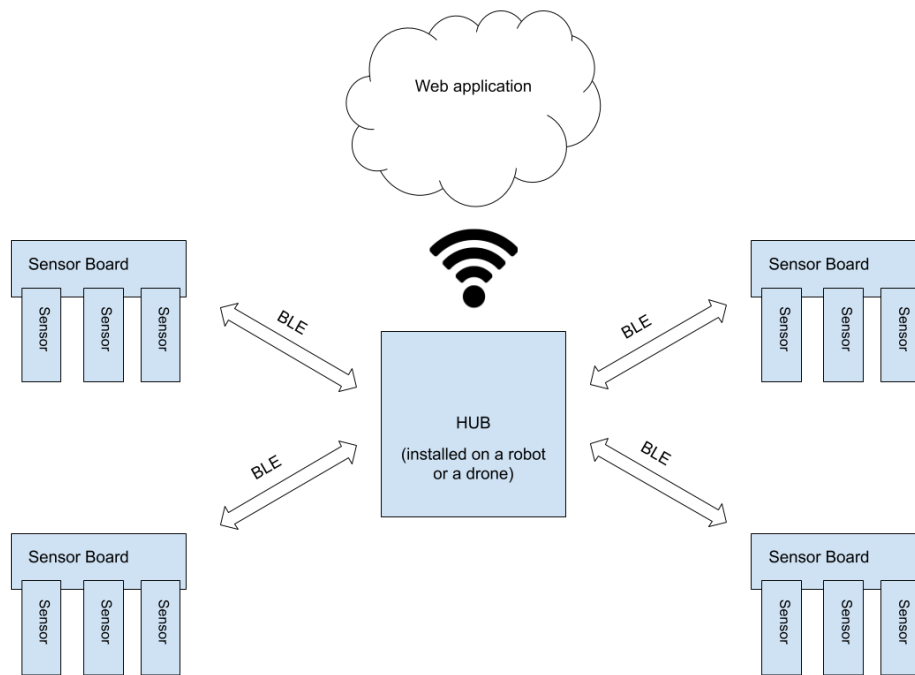


Figure 3.1: System Overview

3.1.2. Sensors and functionalities [JK]

The target product's market defines main functions it should possess to be classified into Agriculture 4.0, including being waterproof, energy efficient and supporting real-life operations. Each separate part of the product must be water resilient to ensure high quality working standard even in adverse climatic conditions.

Sensor board is powered by a solar panel, but in case of no sunlight, device is also equipped with a built-in button cell battery, which should stay serviceable for about 3 years of continuous operation. As it is common with IoT devices, sensor boards periodically move to sleep mode which provides power saving and prolonging the time the device works without charging or replacing the battery.

The next crucial function is GPS locating unit included in every sensor board. Although it is not one of the main features, geographical position can be used by the hub during data analysis and further increase the precision of information about every small part of the measured area. Giving more realistic data about the arable land via WiFi while being far away from the real farmland is a big profit for every user who wants to keep an eye on their plants at any given time.

The final product allows connecting up to 256 sensor boards to the hub. Each sensor board is able to interface with 4 sensors of different kinds:

- soil moisture sensor
- soil temperature sensor
- air temperature sensor
- relative humidity sensor

- wind speed sensor
- water meter
- soil volumetric water content sensor
- soil electrical conductivity sensor
- IR thermometer
- dew point sensor.

Sensor boards are configured to use specific sensors with different frequency of sensor data gathering. It is possible to use them with additional features:

- logging max values
- logging min values
- informing about plant health
- suggesting when to fertilize
- sending alarms over SMS
- displaying area temperature map.

3.1.3. Communication and data presentation [AD]

Communication between the sensor boards and the hub is Bluetooth Low Energy (BLE) based. It is one of the most popular protocols for smart farming applications and was introduced with the Bluetooth 4.0 standard [9]. Connection takes only a few milliseconds and after it is established the devices stay in sleep mode whenever they can which reduces power consumption.

The hub and each of the sensor boards have a built-in BLE module responsible for exchanging data between devices. Sensor boards periodically wake from deep sleep in order to gather data from its sensors and if they have been woken up by the hub to communicate. During communication the sensor board sends the collected data to the hub and receives configuration settings if they have been changed since last data transfer. The communication is periodically initialized by the hub, which stores collected data from each sensor board. Frequency of data gathering can be changed and depends on the number of sensor boards and the distance between them.

The hub creates a wireless, 802.11 based network for ease of access to its web interface and acts as an access point. User is able to connect to this network using their own device with WiFi functionality. After that they are able to open a dedicated web page which presents all data gathered by the hub. Grouped data is displayed in several sections, which contain charts of different sensor types' values and the time of measurement, as well as a heat map illustrating changes of data values across the analysed area. The displayed data is gathered from the sensor boards and then calculated and analyzed by the hub, such as averages values or expected future weather conditions. In addition, the hub processes all data to propose to the user the best practices to achieve the goals specified by that user in order to ensure the most optimized plant care. Status of plant's health is also presented on the website.

3.1.4. Configuration [AD]

There are several different ways to configure the system. The most convenient is changing configuration settings via the website, which provides appropriate forms to allow the user to cho-

ose types of data which should be displayed on the web page, change display names and identification numbers of sensor boards and change the measurement periods of sensors (each sensor type has its own data gathering frequency). Hub saves the configuration settings from the web forms in a database. To spread the information throughout the system the hub sends a configuration file to the sensor board which configuration has been changed by the user.

There is a microSD card in every sensor board which stores configuration settings alongside gathered data. The second method of configuration is a directed transfer of the configuration file to the memory card. Because dedicated file must be written in specific format this mode is only for advanced users. By default the microSD card is configured at the system production stage so user does not have to configure the sensor board themselves, moreover it is optional and mostly for hobbyists.

3.1.5. Comparison [JK]

Table 3.1 shows a comparison of existing solutions and our solution.

Table 3.1: System architecture - product comparison

Product Name	Cesens	OneSense	SmartCrop	Our idea
Maximum number of connected sensors in one sensor board	4	3	N/A	4
Number of types of sensors	11	17	>5	10
Can only use built-in sensors	no	no	yes	yes
System uses a hub device	no	no	yes	yes
Communication: sensor board with a hub	N/A	N/A	radio	BLE
Communication: sensor board with a cloud/web application	Sigfox	Sigfox	cellular network	BLE
Analyzed data	yes	yes	yes	yes
Device provides web-based interface	no	no	no	yes
Weatherproof	yes	yes	yes	yes
Power source	solar panel	battery	solar panel and a battery backup	battery/solar panel

3.1.6. Bill of materials [JK]

Price of a full system is important to a potential buyer. When designing a product we have to balance features and cost of materials. The system has to be available to our wide consumer target group. What follows is our prognosis of the cost of the whole system. In our idea the Hub can be attached to a weatherproof, long distance drone which would additionally cost an upward of \$3000. Tables 3.2 and 3.3 show a generic bill of materials for the system.

Table 3.2: Sensor board - bill of materials

Name	No. of items	Cost [in \$]
Circuit board \w components	1	10
Sensor	4	1
Micro SD card	1	1
Weatherproof enclosure	1	5
Solar panel	1	1
Rechargeable battery	1	0.5
Total		18.5

Table 3.3: Hub - bill of materials

Name	No. of items	Cost [in \$]
Single board computer	1	25
Weatherproof enclosure	1	2
Battery	1	1.5
Total		28.5

3.2. System Model [AD + JK]

Unified Modeling Language (UML) is used in most projects to design the product before its implementation. During architecture stage UML models are repeatedly examined and modified to achieve the most suitable implementation schema for the planned system.

Before an implementation stage of the prototype we created a use case diagram of the whole prototype system, an activity diagram specifying system task flow and detailed descriptions of each use case.

3.2.1. Use case diagram [AD]

Use case diagram shows various actions a user can take and those which are performed by the system. Figure 3.2 presents such diagram. Some of the use cases can be executed by different actors but the action can be different from their perspective.

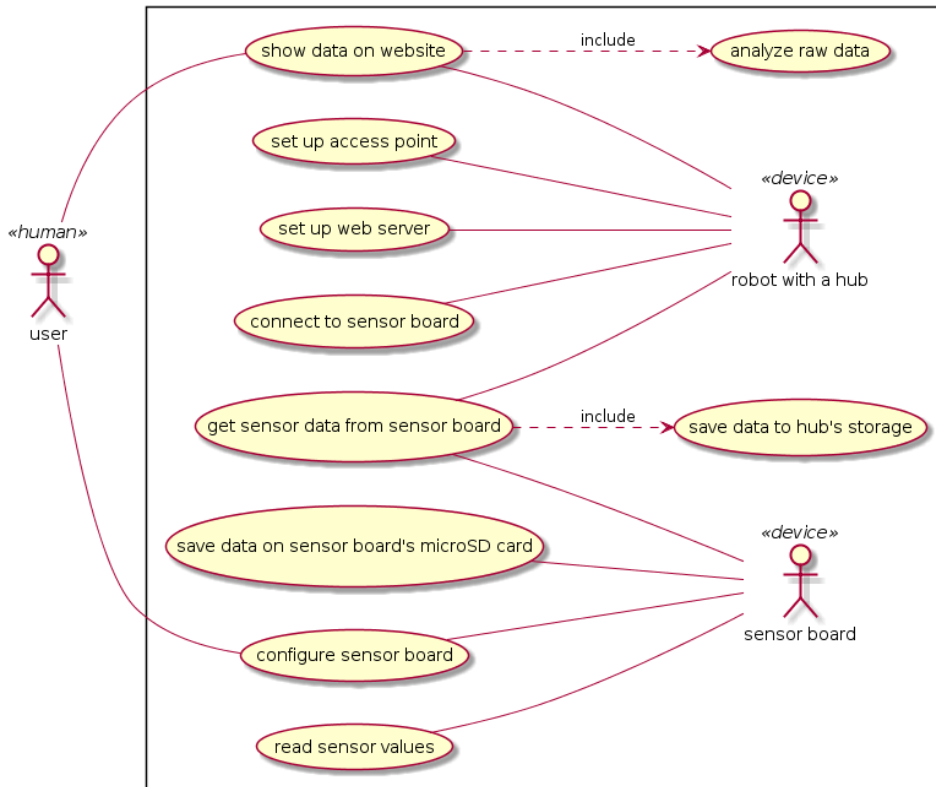


Figure 3.2: Use case diagram of the proposed system

The paragraph below shortly describes each use case presented on Figure 3.2 providing a description and actions taken by actors related to that use case. Procedure of each use case illustrates a real life interaction.

Use Case: show data on website

Description: User wants to display gathered data. Web service displays data received from sensors as well as analyzed data.

Procedure: S1: User connects to hub's wireless network using his own personal computing device.
 S2: User opens dedicated website in browser.
 S3: Web service reads data from storage.
 S4: Web service fills template with data.
 S5: Web service presents the data.

Relation: Includes "analyze data" use case.

Use Case: analyze raw data

Description: Web service analyzes part of the data received from sensor boards to displays it with the rest of the data on the website.

Procedure: S1: Hub reads data from memory.
 S2: Web service analyzes data, e.g calculates an average values.

Relation: Is included by "show data" use case.

Use Case: set up access point

Description: Configure and start wireless access point.

Procedure: S1: Hub starts an access point daemon.

S2: Hub starts a dhcp server.

Relation: none

Use Case: set up web server

Description: Start web server running on hub, accessible on wireless network.

Procedure: S1: Hub starts an http server.

Relation: none

Use Case: connect to sensor board

Description: Connect to sensor board using BLE.

Procedure: S1: Hub looks for available sensor board via BLE.

S2: Hub connects to sensor board via BLE, if there is a connection error hub tries to reconnect.

Relation: none

Use Case: get sensor data from sensor board

Description: Hub receives stored by sensor board data via BLE.

Procedure: S1: Sensor board blocks gathering data process.

S2: Hub sends a 'start' command to a sensor board over BLE.

S3: Sensor reads data from microSD card.

S4: Sensor board sends data to the hub.

S5: Hub receives data.

S6: Hub disconnects after all data has been received.

S7: Sensor board deletes stored data.

S8: Sensor board resumes gathering data process.

Relation: Includes "save data to storage" use case.

Use Case: save data to hub's storage

Description: Save data received from sensor board.

Procedure: S1: Hub saves data.

Relation: Is included by "receive data" use case.

Use Case: save data on sensor board's microSD card

Description: Save gathered data to storage.

Procedure: S1: Sensor board saves read data to a microSD card.

Relation: none

Use Case: configure sensor board

Description: User wants to change default configuration parameters stored on sensor board's microSD card, such as name of sensor board or different gathering intervals for sensor types. Sensor board reads configuration from storage and applies it.

Procedure: S1: User takes a microSD card from the sensor board.

S2: User edits configuration file stored on a microSD card.
 S3: User saves the configuration file.
 S4: Sensor board reads configuration data from a microSD card on boot up.
 S5: Sensor board applies configuration values to own parameters.

Relation: none

Use Case: read sensor values

Description: Gather data from built-in sensors.

Procedure: S1: Sensor board reads values of built-in sensors.
 S2: Sensor board stores gathered data on a microSD card.

Relation: none

3.2.1.1. Formal description of use cases

Formal descriptions providing detailed information about each use case of Figure 3.2. Thorough analysis presented below is essential for following this stage creation of activity diagrams, which shows more specified information of system functionalities.

Use Case: set up access point

Actors: robot with hub
 Abstract: Configure and set up wireless access point running on hub.
 Initiating event: Hub powered on.
 Structure: none
 The initial conditions: Hub is powered on. Access point is not configured yet.
 Full description: Hub provides a wireless access point to let a user connect to it and access system website.
 Exceptional situations: none
 Final conditions: Access point is configured and running.
 Comment: Access point should be protected with a password.

Use Case: set up web server

Actors: robot with hub
 Abstract: Run configured web server with website providing sensor data from the system.
 Initiating event: Hub powered on.
 Structure: none
 The initial conditions: Access point is running.
 Full description: Hub starts a web server. User is able to access the website after connecting to hub's wireless access point. The website shows data, as well as analyzed data, from the whole system.
 Exceptional situations: none
 Final conditions: Web server is running.
 Comment: Website should be accessible on a known address or a domain translation should be set up so user can easily remember website's address.

Use Case: **connect to sensor board**

Actors: robot with hub

Abstract: Hub connects to a sensor board using BLE.

Initiating event: Hub sees an available BLE device it hadn't contacted yet.

Structure: none

The initial conditions: Hub is scanning for active BLE devices.

Full description: Hub searches for active sensor boards and connects to them. After receiving the data it stores information about the sensor board in order to not connect to it again throughout the data gathering procedure.

Exceptional situations: Connection was interrupted: hub tries to connect to sensor board one more time.

Final conditions: Connections between hub and sensor board is established.

Comment: When hub tries to reconnect with sensor board, it should be limited to three retries. Hub should store sensor board's MAC address after receiving data from it.

Use Case: **receive data**

Actors: robot with hub

Abstract: Hub receives data from a sensor board.

Initiating event: Hub connects to a sensor board using BLE and sends a 'start' command to the sensor board.

Structure: (include) save data to storage

The initial conditions: Hub is connected to a sensor board.

Full description: When hub is connected to a sensor board via BLE, it receives proper frame with data from sensor board. When hub is sure that all data has been received it finishes BLE connection with sensor board.

Exceptional situations: Connection was interrupted during sending the data: hub tries to connect to sensor board one more time.

Final conditions: Data from sensor board has been received.

Comment: Data will be sent over multiple frames.

Use Case: **save data to storage**

Actors: robot with hub

Abstract: Hub saves data received from a sensor board to its storage.

Initiating event: Hub finishes collecting data from sensor board.

Structure: (extend) save data to storage

The initial conditions: Connection between hub and sensor board is established. Hub sent a command to sensor board and sensor board begins sending data.

Full description: Hub processes data, add timestamps to each data point, saves unsaved data in proper format in database.

Exceptional situations: Memory for raw data is full: hub deletes the oldest data.

Final conditions: Data is saved in memory of hub.

Comment: Hub saves the received data into memory when connected to the sensor board.

Use Case: **show data**

Actors: robot with hub

Abstract: Show data received from sensor boards and data analyzed by hub.

Initiating event: User opens system's webpage.

Structure: (include) analyze data

The initial conditions: Web server is running. User is connected to hub's wireless access point and is accessing the webpage.

Full description: Website displays data gathered from sensor boards, as well as analyzed data. Website provides charts etc. for user's ease of comprehension.

Exceptional situations: There is no gathered data yet: webpage displays appropriate statement.

Final conditions: Data is displayed on the webpage.

Comment: Analyzed, displayed data should contain diagrams, charts, tables with data for clarity of information.

Use Case: **analyze data**

Actors: robot with hub

Abstract: Prepare information easy to understand for a regular user.

Initiating event: User requests a website view of data.

Structure: (extend) show data

The initial conditions: The data is not analyzed at all or not analyzed in the way indicated by the configuration settings.

Full description: Website analyzes data (e.g. calculates average values, creates graphs and charts) to fill a website template.

Exceptional situations: There is no data to analyze: hub does not analyze any data.

Final conditions: Data is analyzed and returned to website.

Comment: Only a part of data will be analyzed for the website. Analyzed data will not be stored.

Use Case: **configure**

Actors: sensor board

Abstract: Sensor board configures few of own parameters based on data from SD card.

Initiating event: Sensor board started.

Structure: none

The initial conditions: Sensor board's name and gathered period of every sensor type have default values.

Full description: Sensor board reads configuration data from a microSD card and applies configuration values to own parameters, like BLE device id or period for gathering sensor values for every type of sensor.

Exceptional situations: There is no SD card inserted into sensor board: sensor board stops working and flashes red diode.

Final conditions: Sensor board's internal parameters are set or left as defaults.

Comment: The most important is to provide unique sensor board's name, which allows the hub to recognize all devices while connecting via BLE. When there is no value for gathering period sensor board uses a default value.

Use Case: read sensor values

Actors: sensor board
Abstract: Sensor board reads built-in sensor values.
Initiating event: Built-in timer reached for a given sensor type.
Structure: none
The initial conditions: Sensor board is running and is configured.
Full description: Sensor board starts reading sensor values when built-in timer reaches a set value. Next sensor board restarts the timer after finishing collecting data.
Exceptional situations: none
Final conditions: Data from a sensor is gathered.
Comment: Sensor board blocks availability of BLE connection for hub as long as gathering data process is running.

Use Case: save data

Actors: sensor board
Abstract: Sensor board saves data after it read sensor values.
Initiating event: Sensor board finishes collecting data from sensors.
Structure: none
The initial conditions: Sensor board collected data from sensors, but has not saved it yet.
Full description: Sensor board saves unsaved data in proper format in storage.
Exceptional situations: Memory is full: sensor board overwrites the oldest measurement.
Final conditions: Data is saved.
Comment: The risk of overfilling the microSD card is minimized by sensor board deleting gathered data after it is sent to the hub. This applies only if the hub periodically connects to the sensor board to receive its data.

Use Case: send data via BLE

Actors: sensor board
Abstract: Sensor board sends sensor values to hub via BLE.
Initiating event: Sensor board receives a 'start' command from hub.
Structure: none
The initial conditions: Hub connects to sensor board via BLE.
Full description: Sensor board reads data from microSD card and send it to a hub. After hub disconnects sensor board deletes data from memory and unlock gathering data (making gathering available again).
Exceptional situations: Connection was interrupted: Sensor board cancels the process.
Final conditions: Data is sent.
Comment: Communication begins only after the sensor board receives a 'start' command.

3.2.2. Activity diagram [AD]

Activity diagrams are essential for showing detailed and consistent structure of system operation. An activity diagram consists of sets of small actions forming together a whole procedure performed by the system in order to accomplish a certain use case. Presented diagrams (Figures 3.3-3.7) are a low level representation of actual system functionalities which makes them helpful for developers during implementation.

3.2.2.1. Initialization of a hub

Use cases: "Set up access point"
"Set up web server"

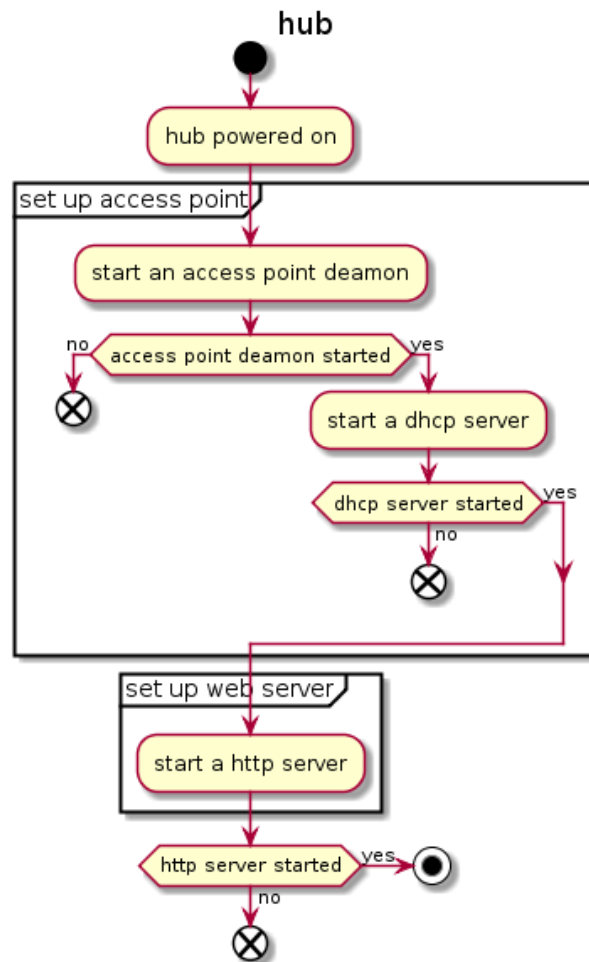


Figure 3.3: Activity Diagram - Initialization of a Hub

3.2.2.2. Initialization of a sensor board

Use cases: "Configure"

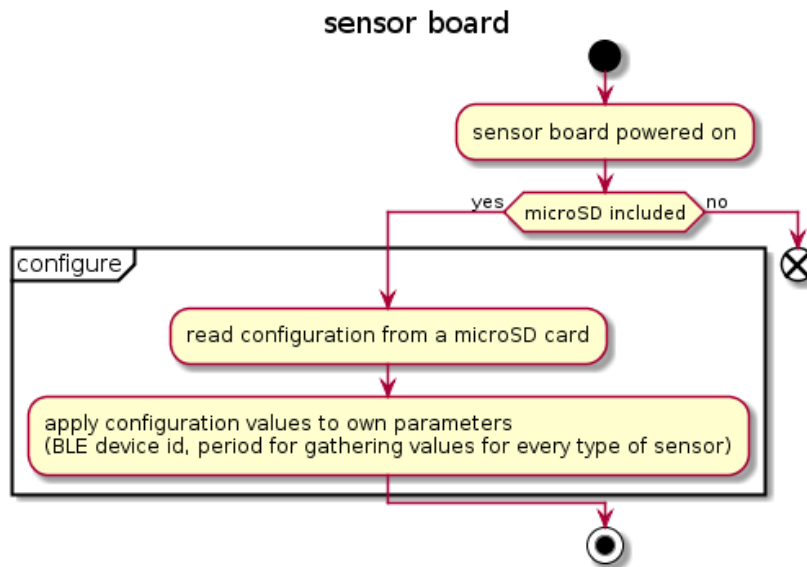


Figure 3.4: Activity Diagram - Initialization of a Sensor Board

3.2.2.3. Gathering data process

Use cases: "Read sensor values"
"Save data"

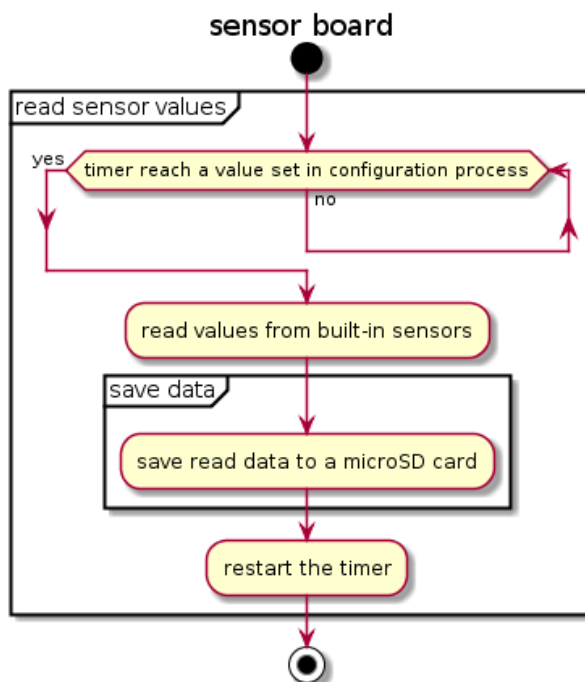


Figure 3.5: Activity Diagram - Gathering Data Process

3.2.2.4. Exchanging data between hub and sensor board

Use cases: "Connect to sensor board"

"Receive data"

"Save data to storage"

"Send data via BLE"

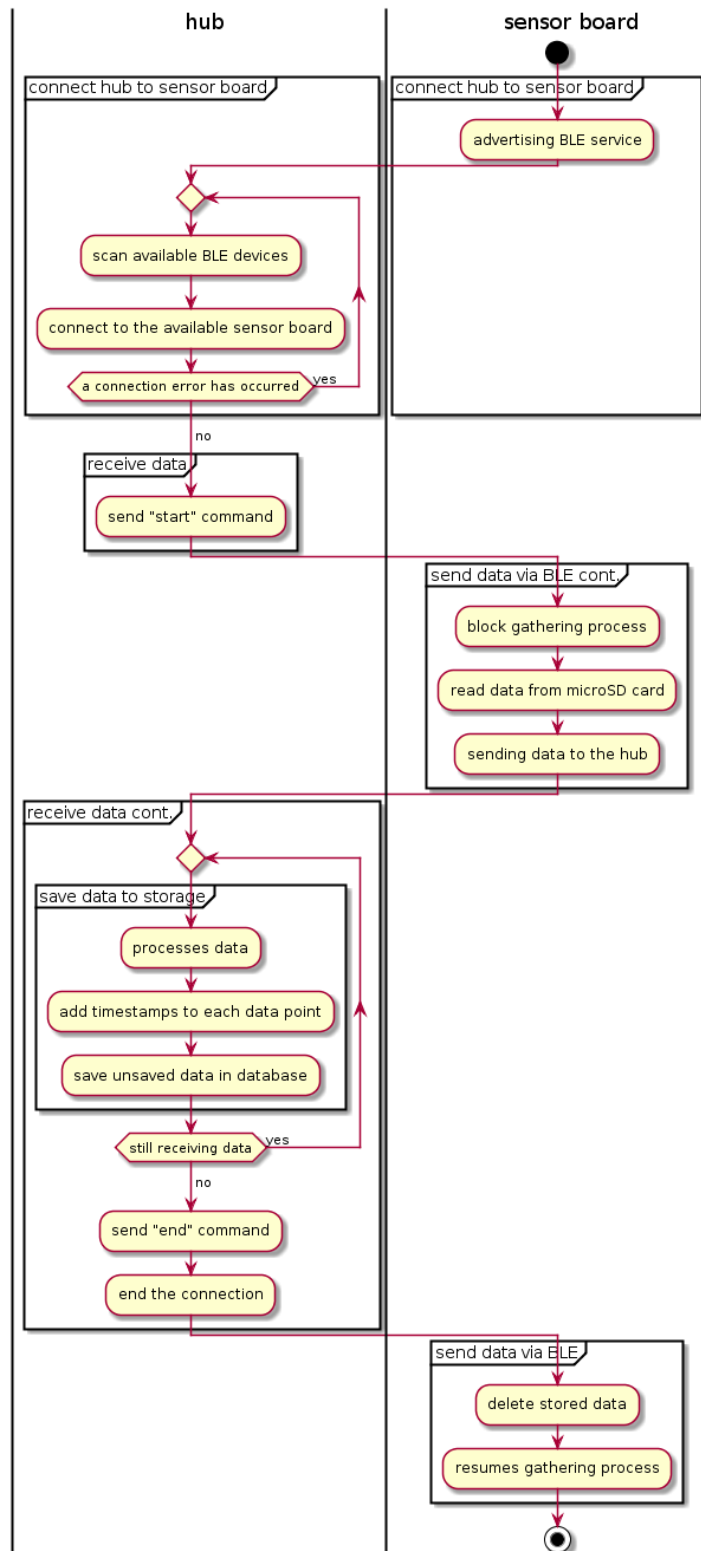


Figure 3.6: Activity Diagram - Exchanging Data Between Hub and Sensor Board

3.2.2.5. Providing data on website

Use cases: "Show data"
"Analyze data"

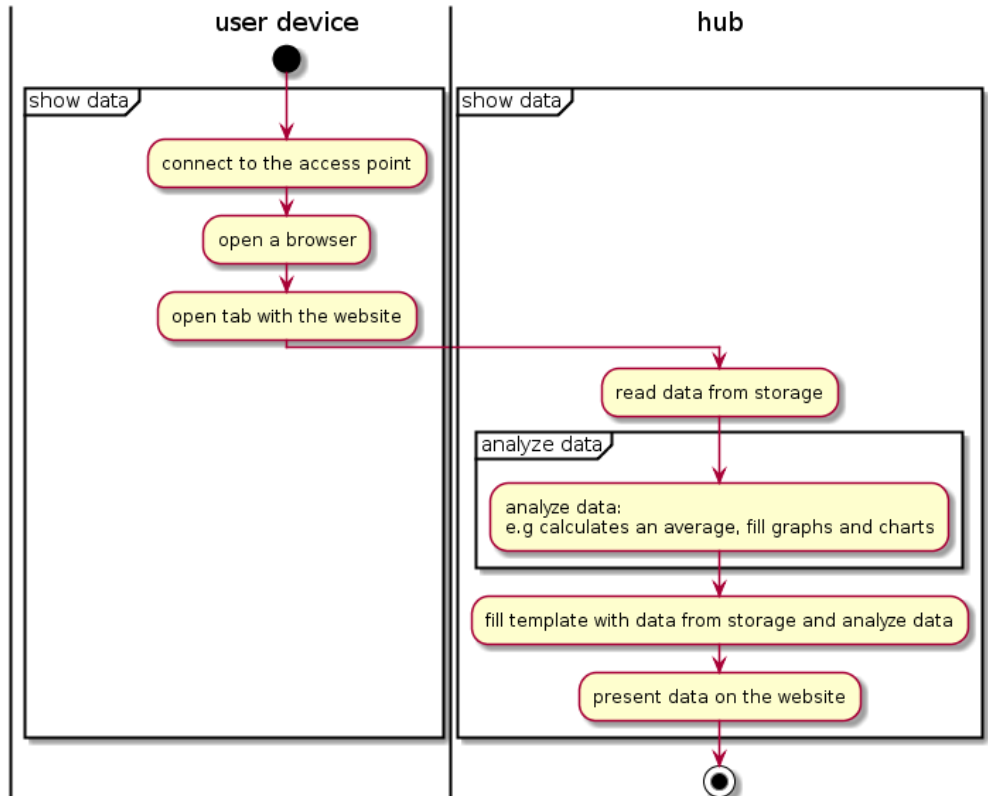


Figure 3.7: Activity Diagram - Providing Data on Website

3.2.3. Sequence diagram [JK]

Sequence diagrams are similar to activity diagrams but bring more detailed information on data flow and the sequence of events. Figures 3.8-3.16 provide schemas of system's workflow, showing concrete data and information exchanging process. Presented diagrams clearly specify the interactions between the real system subcomponents, facilitating the implementation process for the developers.

3.2.3.1. Hub

3.2.3.1.1. Initialization of a hub

Use cases: "Set up access point"
"Set up web server"

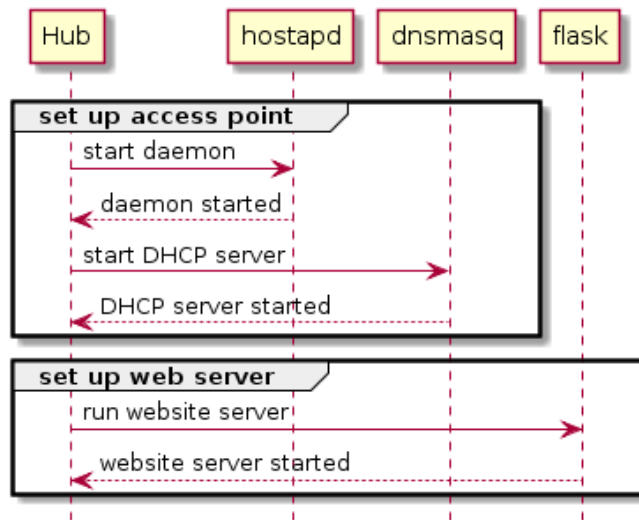


Figure 3.8: Sequence Diagram - Initialization of Hub

3.2.3.1.2. Display website

Use cases: "Show Data"
"Analyze data"

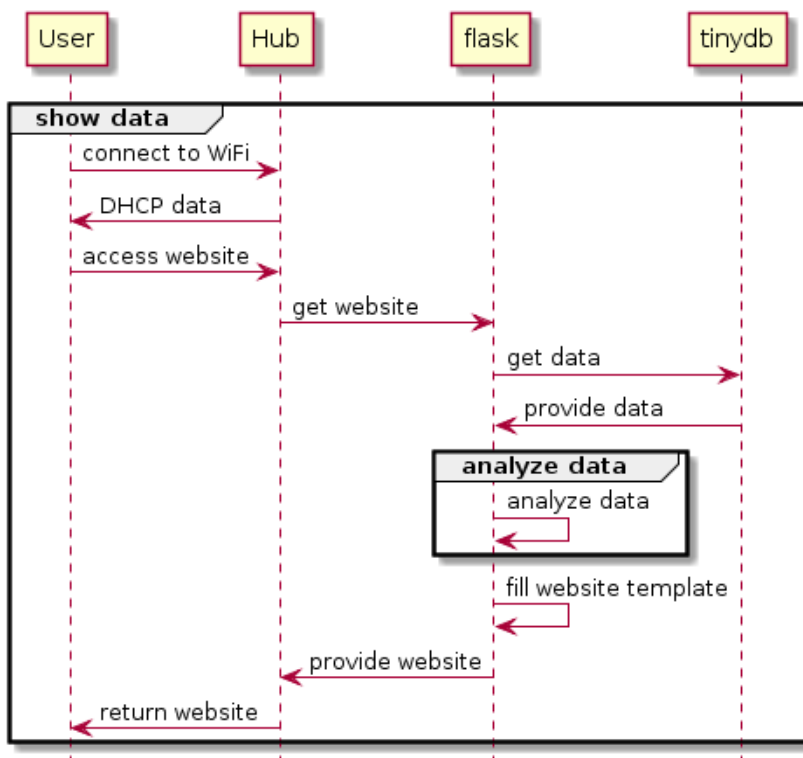


Figure 3.9: Sequence Diagram - Display Website

3.2.3.1.3. Connect to sensor board

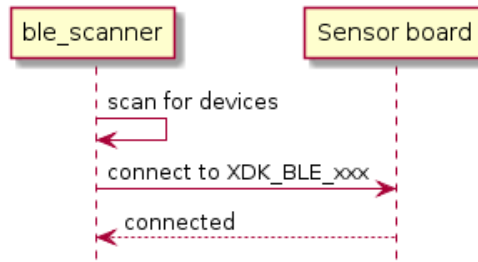


Figure 3.10: Sequence Diagram - Connect to Sensor Board

3.2.3.1.4. Receive data and store

Use cases: "Receive data"
"Save data to storage"

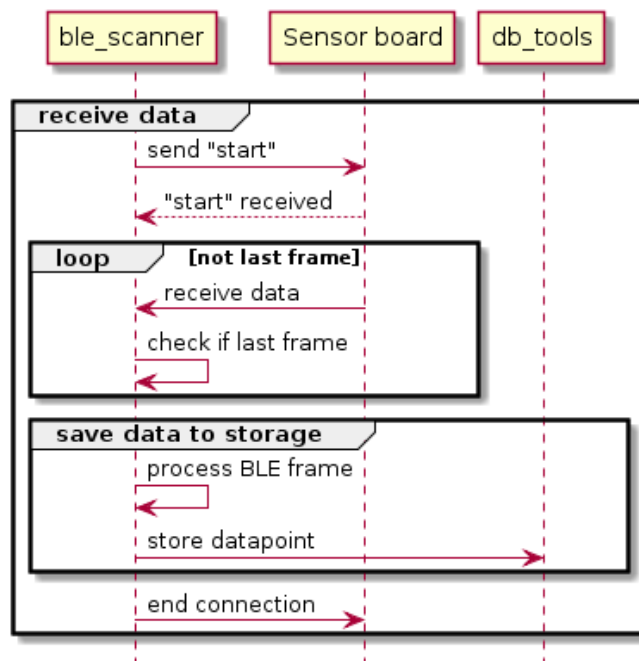


Figure 3.11: Sequence Diagram - Receive Data and Store

3.2.3.1.5. Save data to storage

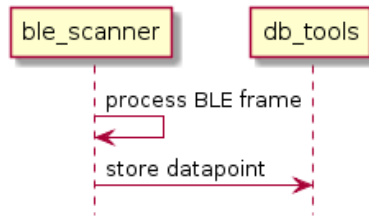


Figure 3.12: Sequence Diagram - Save Data to Storage

3.2.3.2. Sensor Board

3.2.3.2.1. Configure

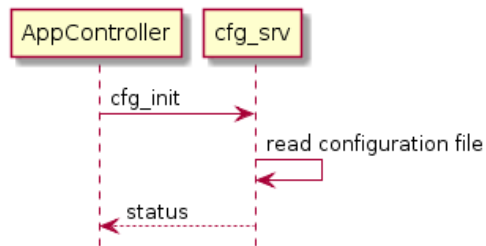


Figure 3.13: Sequence Diagram - Configure

3.2.3.2.2. Read sensor values

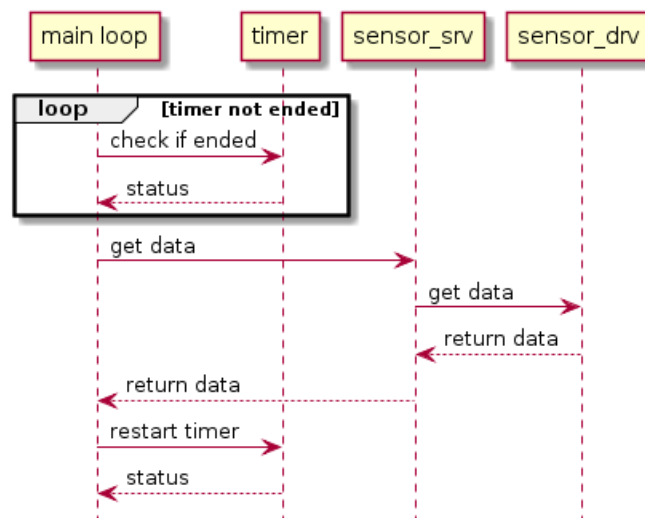


Figure 3.14: Sequence Diagram - Read Sensor Values

3.2.3.2.3. Save data

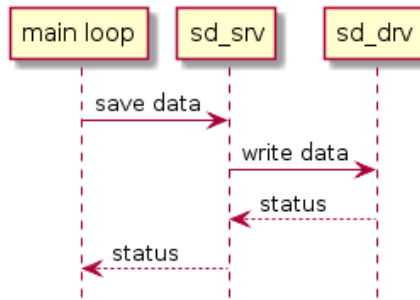


Figure 3.15: Sequence Diagram - Save Data

3.2.3.2.4. Send data via BLE

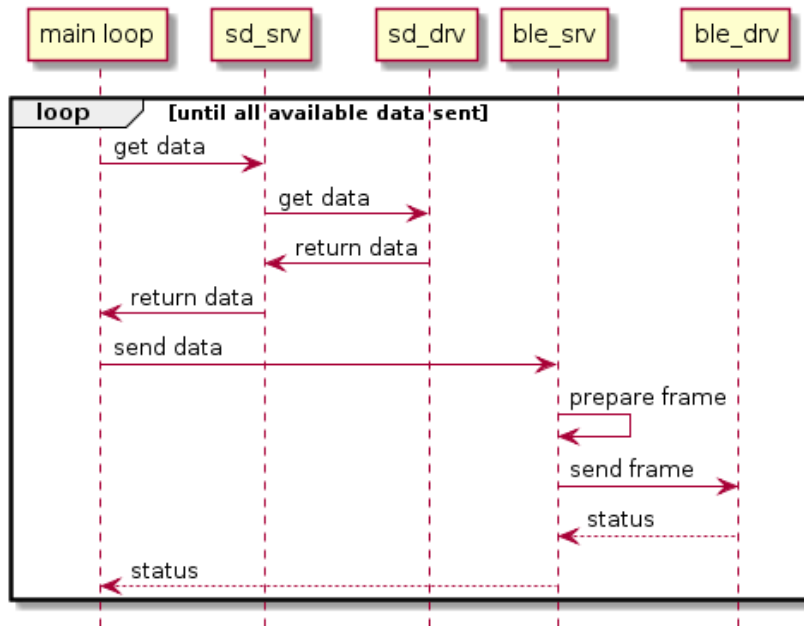


Figure 3.16: Send Data via BLE

4. SYSTEM PROTOTYPE DEVELOPMENT [AD + JK]

4.1. System Overview [AD]

Before the full implementation of the target product, a prototype was created to ensure that every chosen technology solution is adequate for a final system.

Our prototype consists of two sensor boards which measure and store environmental data in a remote location, a hub used to gather data from sensor boards and display it on a web page available on a wireless network the hub provides and a robot provided by Gdańsk University of Technology which travels near the sensor boards to gather data from them. Hub's mobility ensures real time system operation, which results in being able to store the most recent data. Figure 4.1 shows a diagram of the system, its physical components and functionalities (excluding the robot which we did not develop).

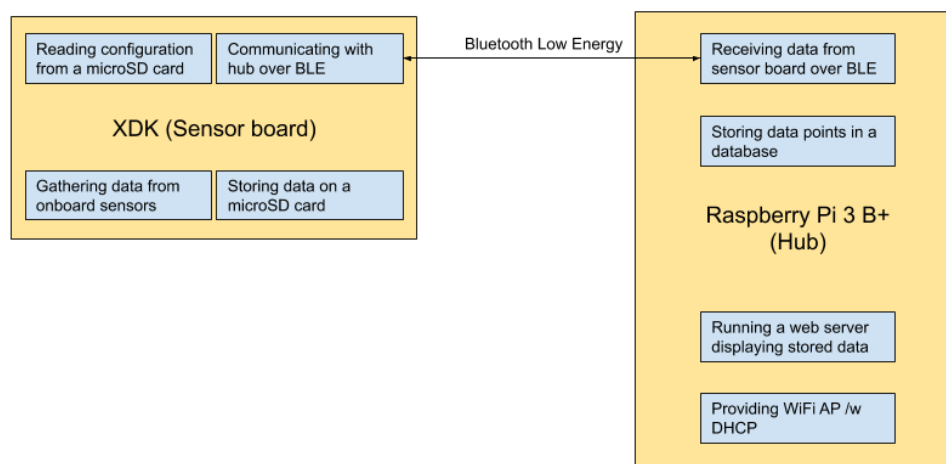


Figure 4.1: System functionality diagram

To grant access to the hosted website the hub creates an 802.11 based wireless network. Using devices with built-in WiFi users are able to connect to hub to then access a dedicated system's website. The name and the password of the wireless network are saved in a configuration file stored on a microSD card inserted into the hub. Raw data of every sensor type gathered from the field are presented on a website. Information are grouped into tables and consist: time, value and sensor board identification number of every measurement.

Many sensor boards scattered throughout the measurement area store data from different built-in sensors in their own memory (microSD card). Each sensor type has individual period of measurement, which are hard programmed in one minute due to convenient system's test proceeding. Stored data is sent to the hub using Bluetooth Low Energy protocol, thanks to built-in BLE module in both endpoint devices (hub and sensor board). In each data collection cycle the hub periodically gathers data from sensor boards: initiates a connection with a board, receives data and stores it.

4.1.1. Initial threats

The system's complexity forced us to decide which of the components will be developed by ourselves and which will be integrated from ready-to-use solutions. While discussing possible scenarios we worked out many essential issues, determining future system functionalities and technologies used.

Creating the sensor board by ourselves was the first goal we wanted to achieve but the time constraint was an issue. Additionally, creating the sensor board from the basics brings a lot of risks and possible problems, from construction of the board to its implementation. In the end we borrowed Bosch Cross Domain Development Kit (XDK) boards from Gdańsk Technical University. Bosch XDK is a ready-to-use development platform which makes it impossible to modify the device to achieve certain functionalities. It is sufficient though, as a base of the prototype sensor board.

Bluetooth Low Energy communication between a hub and a sensor board was to be implemented using a Nordick nrf52840 board with a built-in Bluetooth module. Nrf52840 was supposed to be connected via serial with the hub, acting as an external Bluetooth component. Nordick does not provide easy to use examples and we decided that implementing BLE and serial communication would take too much time and work. Extensive documentation including a lot of specific details suggested that changes need to complete such implementation would outgrow the complexity of the problem. To avoid further problems with BLE communication between a hub and a sensor board a different solution was used in the project. In the prototype a single-board computer plays a role of the hub and because of an included Bluetooth module it was not necessary to look for other solutions. The module is well documented with a variety of open source libraries providing Bluetooth functionality.

4.1.2. Restrictions

The main restriction in the prototype is the XDK hardware construction which bring many software limitations. We are not able to easily implement additional features of the board. Additionally, Bosch did not implement all functionalities we would expect from a microcontroller (an example being different power modes - it is not possible for the board to go to sleep in order to conserve power). The limitations did not stop us from creating the prototype, however it would block the implementation of the final product.

4.2. Hardware Components [JK]

The sensor boards were based on a Bosch XDK programmable sensor device while the hub was developed on a single computer board - Raspberry Pi 3 Model B+. We do not provide information about the robot as it was outside of scope of our project.

4.2.1. Bosch XDK

Cross Domain Development Kit [10], abbreviated as XDK (Figure 4.2), is a product developed by Bosch with Internet of Things (IoT) projects in mind. The manufacturer advertises the product as a professional prototyping platform with the ability to employ a project based on it either as

a short-term proof-of-concept or as a node for long-term projects. The kit consists of a small form factor, battery powered device with built-in sensors and connectivity, as well as a Software Development Kit.

We used the XDK to create a sensor board for our project.



Figure 4.2: Bosch XDK110

Source: xdk.bosch-connectivity.com/hardware (Nov 27, 2019)

4.2.1.1. Specification

The XDK device is based around a 32-bit ARM microcontroller with Bluetooth Low Energy and WiFi capability. It has a microSD slot which enables data storage and a rechargeable battery which allows for remote operation. The microcontroller runs a compiled project created using a Software Development Kit available from Bosch. The firmware is based on a FreeRTOS operating system and allows for concurrently running tasks which makes creating advanced systems much easier.

4.2.1.2. Sensors and functionalities

Our prototype has limited features compared to the full product we have devised and offers only basic functionalities. This is because of limited time to finish this project and prototyping is a vital part of product development.

The sensor board is powered by a lithium-ion (Li-Ion) rechargeable battery (included in the XDK) which lasts between two weeks and several months, depending on workload. As of the making of the project Bosch has not implemented additional power states on the XDK board so we were not able to utilize sleep modes in order to conserve battery life. Considering the workload and manufacturer's description we assume the sensor board would be able to stay working for more than two weeks but no testing has been done to determine actual battery life in our use case.

The prototype system is able to handle up to 256 sensor boards. This limitation is caused by the data format of packets exchanged between a sensor board and a hub. It could be expanded in the final product.

We chose to utilize 4 of XDK's built-in sensors in the prototype:

- humidity sensor
- pressure sensor
- temperature sensor
- light sensor

Sensor board gathers data from each sensor and stores it on a microSD card. The storage holds a configuration file as well which can be used to set different intervals of data gathering for each sensor type.

4.2.1.3. Configuration

Only the sensor board of the prototype can be configured. This can be done by editing a configuration file stored on a microSD card present in every sensor board. The device uses the memory card to store configuration settings and all gathered data. The user is able to change an identification number of a sensor board (used to identify which sensor board specific data comes from and present that information to user using hub's web page) and measurement periods of each sensor type (number of seconds to wait before storing data from a sensor).

4.2.2. Raspberry Pi [JK]

Raspberry Pi is a single board computer developed by the Raspberry Pi Foundation [11]. The foundation's goal is to provide low-cost, high-performance computers to people who use it to learn and solve problems. It is an inexpensive platform that is useful in prototyping. The computer's ARM processor is able to run operating systems that are familiar to PC users which enables cross-platform development and allows for easy skill transfer. In our prototype we used the third generation model - Raspberry Pi 3 Model B+ (Figure 4.3). Its wireless network capability and Bluetooth Low Energy support was used to develop the hub of our system.

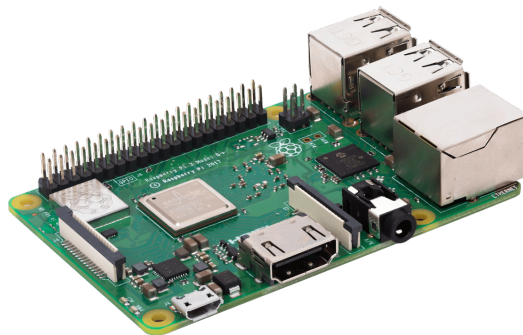


Figure 4.3: Raspberry Pi 3 Model B+

Source: market.samm.com/raspberry-pi-3-b-plus-en (Nov 27, 2019)

4.2.2.1. Specification

Raspberry Pi 3 Model B+ is a single board computer based on a 64-bit quad-core ARM processor with 1GB of RAM. It supports 2.4GHz and 5GHz WiFi, as well as Bluetooth 4.2. Additionally it offers an HDMI port, a Gigabit Ethernet port and 4 USB 2.0 ports which were used to configure the computer and develop the hub's software.

4.2.3. Data flow

Figure 4.4 shows how data flows through the system. It begins with sensor board reading data from its on-board sensors and storing it on a microSD card. When the hub connects to the

sensor board and sends a start command each stored data point is fitted into a data structure with a format shared by the sensor board and the hub. After that the sensor board sends the message over BLE and the hub receives it, reads the data structure and stores the received data point in a database. When an user wants to access the website hosted on the hub the data in the database is accessed, analyzed and displayed to the user.

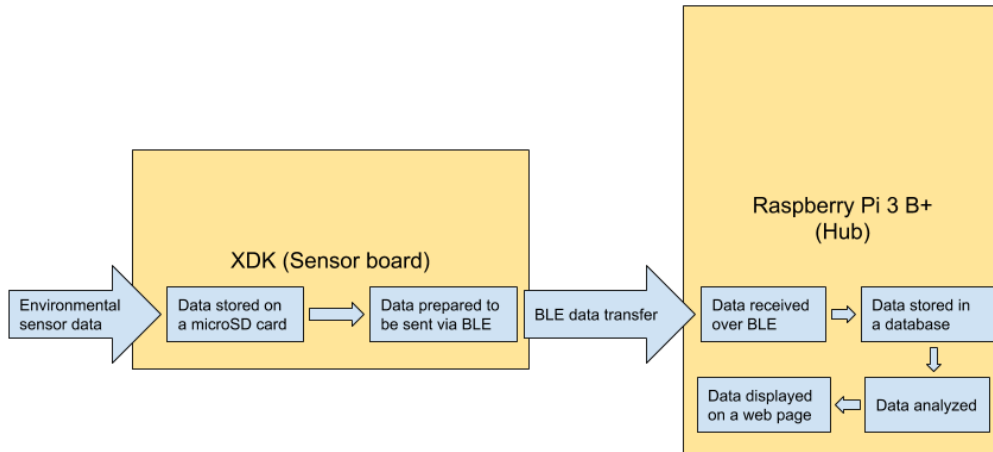


Figure 4.4: Data flow diagram

4.3. Software Components [JK]

Sensor boards were programmed using Bosch's Software Development Kit which comes with the XDK and is written in C programming language. The hub runs an Arch operating system and makes use of packages available in official operating system repositories to create a wireless access point and start a DHCP server to allow for an easy connection. The webpage provided by the hub is based on Flask framework which is written in Python programming language. We created additional Python scripts to handle Bluetooth Low Energy communication and database storage. Figure 4.5 shows a software block diagram of the system.

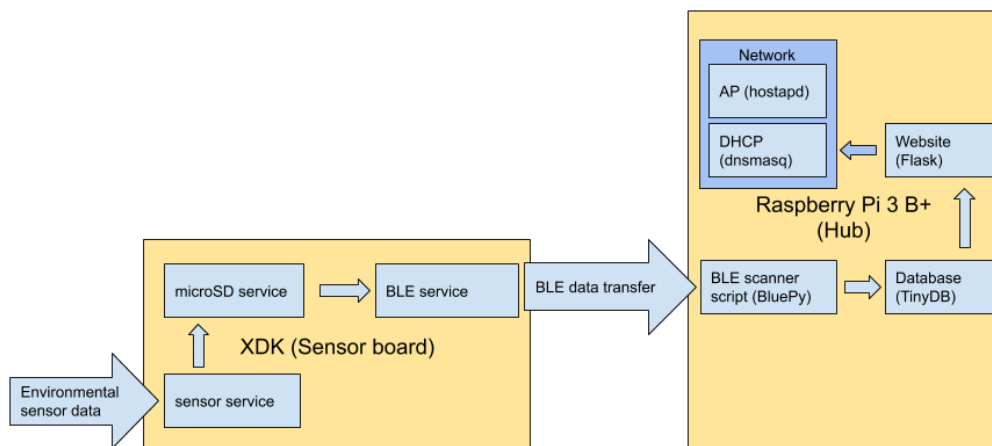


Figure 4.5: Software block diagram

4.3.1. Data presentation

The Raspberry Pi acts as an access point creating a wireless, 802.11 based network (commonly named WiFi) for ease of access to its web interface. Using a device with WiFi functionality users are able to connect to the network which gives them access to a dedicated web page which presents all data gathered by the sensor boards that is stored on the hub. Displayed data is grouped into tables, one for every sensor type, containing a timestamp of the data point, measured value and a sensor board identification number.

4.3.2. Bosch XDK

The Cross Domain Development Kit comes with a Software Development Kit which was used to develop the sensor board's firmware.

4.3.2.1. Architecture

We chose to implement an organizational-like architecture. This hierarchy of control ensures code clarity and helps with simplicity.

Figure 4.6 shows the architecture with lines signifying flow of control. Main and AppController are modules specific to the XDK sdk (Software Development Kit) and the rest was developed by us. Sensors is a module responsible for running two concurrent tasks - BLE connection handling and sensor data gathering. Lower in the hierarchy there are services. They provide a high-level API (Application Program Interface) and include business logic to handle different scenarios. In total there are 4 services:

- configuration service (cfg_srv)
- microSD service (sd_srv)
- Bluetooth Low Energy service (ble_srv)
- sensor service (sensor_srv)

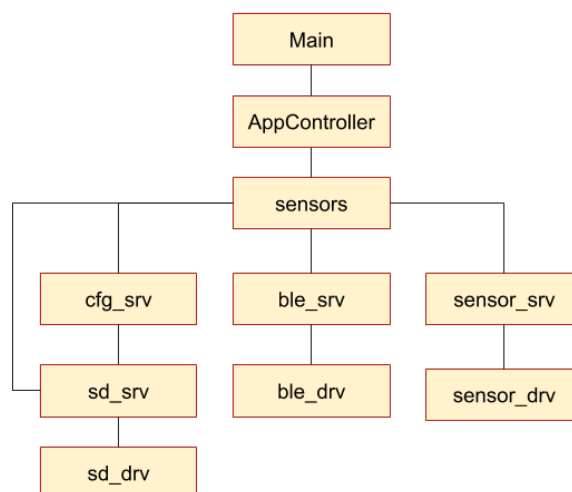


Figure 4.6: Sensor board system architecture

Each service has a corresponding driver. Only a driver directly communicates with hardware (in our case uses low level XDK API) and each service uses only its own driver.

4.3.3. Data formats

Communication between a sensor board and a hub required defining a data format for the Bluetooth Low Energy messages. Figure 4.7 shows a 16-byte message consisting of 5 fields:

- "is last frame- value signifying if the message/frame is last and the hub should end the connection after processing it
- "xdk id- sensor board identification number
- "sensor type- enumerated value signifying a sensor type for that data point
- "data- sensor value
- "timestamp- number of milliseconds since sensor board's boot at the time of taking the measurement

The unused space on byte number 4 is made by a compiler. Code representation of the frame is a structure that cannot be packed because XDK's processor does not support unaligned memory addresses. The representation could be changed or the fields could be moved to mitigate this but for the purpose of the prototype we decided to accept this.

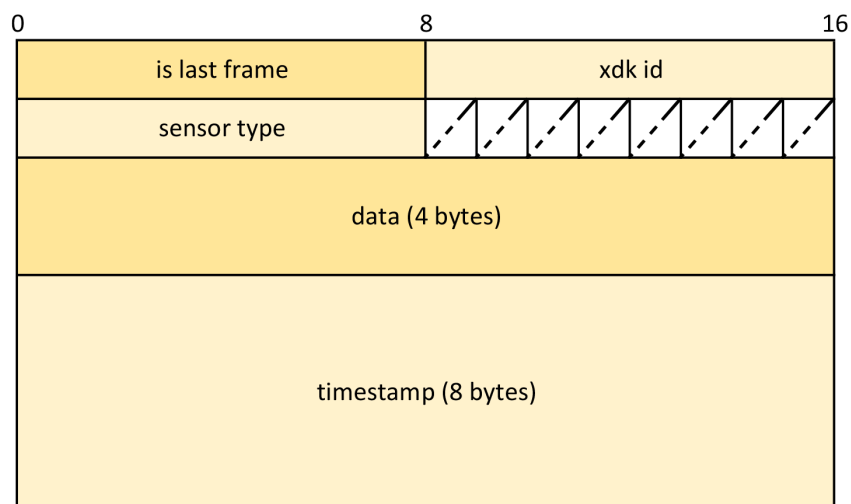


Figure 4.7: BLE message data format

Bluetooth Low Energy allows for sending up to 20 bytes at one time. It is possible then to add additional fields in the future if they are needed.

It is worth noting that the timestamp recorded by the sensor board is a value representing the number of milliseconds since that sensor board's boot at the time of measurement. When all data has been sent by a sensor board it sends a last frame to the hub which contains no data but a timestamp of actual number of milliseconds since boot. With that the hub can calculate the time differences of each data point and create a new timestamp with correct date and hour of sensor measurement. We decided to do it this way because synchronizing the XDK devices with actual time would be too difficult.

$$datetime = current_datetime - (boot_milliseconds_now - boot_milliseconds_at_measurement)$$

When the hub receives data from a sensor board it buffers it, waiting for the last message. After that the hub processes all messages, calculates correct timestamps and stores the data in a database. Figure 4.8 shows a database model used in our project. It consists of just one table with 4 fields:

- "sensor- enumerated value signifying a sensor type for that data point
- "timestamp- date and hour of sensor measurement
- "xdk_id- sensor board identification number
- "data- sensor value

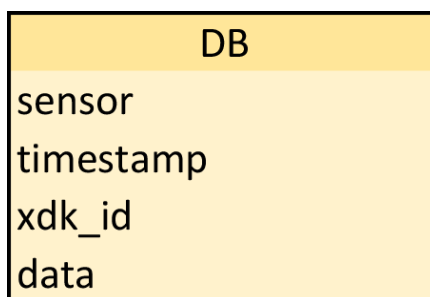


Figure 4.8: Database model

The database is very simple and is used just as a means of storage and to allow for querying the data.

4.3.4. Tinydb

Our project did not need an advanced database so we chose a minimalistic one. Tinydb is a Python library implementing a simple database. It is lightweight, document oriented and does not have additional dependencies. The database is stored on the hub as a JSON (JavaScript Object Notation) formatted file.

A script was created with functions for accessing the database and analyzing data therein. It is used by hub's web page template.

4.3.5. Flask

Similarly to the database our project did not need an advanced web page and http server. We chose Flask - a micro web framework written in Python. It allows for very quick web application development.

A web page was created consisting of tables, one for every sensor type, containing a timestamp of the data point, measured value and a sensor board identification number. The data is taken from the database using a helper script.



4.4. Communication [JK]

Hub communicates with sensor boards via Bluetooth Low Energy protocol, this same occurs in the final product. BLE is the one of the most often used protocols in Agriculture 4.0 and it requires only few milliseconds to make a connection to exchange data.

The Raspberry Pi and each of the XDKs have a built-in BLE module responsible for exchanging data between devices. Sensor board is advertising the owning BLE service and periodically gather data from its sensors. The communication is initialized by the Raspberry Pi after user clicked dedicated button on a website, which launch data collection from sensor boards. If the hub is in the board's range it communicate with a XDK to exchange data. During communication the sensor board sends the collected data to the Raspberry Pi, which saves them into own memory. The XDK delete stored data after send it to the hub.

4.4.1. Bluetooth Low Energy (BLE)

Bluetooth Low Energy is a variation of the Bluetooth standard designed for low power consumption. It was introduced as a part of the Bluetooth 4.0 specification [9]. BLE was developed with wearable devices, proximity sensors and smart home appliances in mind. It finds its use widely in Internet of Things (IoT) implementations [12].

In our project we used Bluetooth Low Energy to send sensor data stored on sensor boards to a hub. Low power consumption of this protocol ensures longer battery life for the sensor boards which is important in remote, automated systems such as ours.

4.4.1.1. Bluetooth

Bluetooth is a wireless technology standard developed by Bluetooth Special Interest Group. At this moment the newest specification, presented in 2019, defines Bluetooth 5.1.

Bluetooth operates in the 2.4GHz frequency band. Data transfer utilizes packets exchanged through one of 79 designated channels (each of 1MHz width) [13]. Additionally to being a packet-based protocol Bluetooth uses a master/slave architecture. One master is able to communicate with up to 7 slaves. Packet exchange is based on master's clock. Different number of clock ticks define slots. Master transmits packets in even slots and receives in odd slots. The slave, on the contrary, transmits in odd slots and receives in even ones. Packets can be 1, 3 or 5 slots long.

4.4.1.2. Differences

The main difference, and at the same time primary advantage, of BLE compared to Bluetooth is lower power consumption. Devices implementing BLE are able to last multiple years on a single coin-cell battery. Unlike normal Bluetooth devices those using the Low Energy variant remain in sleep mode constantly except for when a connection is made. The initial connection can be timed in single milliseconds while a Bluetooth enabled device would take close to a hundred milliseconds [14]. This is one of the factors determining battery life. Bluetooth Low Energy uses the same frequency band as classic Bluetooth but uses a wider set of channels - 40 2MHz channels instead of 79 1Mhz. Both variants use a master/slave architecture where packet exchange is based on master's clock but the methods of utilizing slots are different [15].



5. SYSTEM TESTS AND MEASUREMENTS [AD + JK]

Towards the end of our project we performed a field test to determine if all of the prototype functionalities have been delivered. Additionally we tested signal strength of sensor boards as perceived by the hub. More testing needs to be done in order to find shortcomings and better shape further path of development of the system.

5.1. Functional Test [JK]

As a part of the project a functional test was done. We wanted to check and demonstrate that our prototype provides its planned functionalities. The test took place in front of Faculty of Electronics, Telecommunications and Informatics on Gdańsk University of Technology grounds. As can be seen in Figure 5.1 the test set included two XDK boards (sensor boards) propped on wooden platforms, a Raspberry Pi 3 model B+ (hub, seen in the picture at the top of a robot - Figure 5.1a), a tablet (Figure 5.1b) used to connect to a wireless network provided by the hub, and a robot developed in the before mentioned faculty.



(a) robot and XDKs during test



(b) gathering data view on system website

Figure 5.1: Functional Test - test set

5.1.1. Testing process

The sensor boards were powered on and let gather data for a minute. The hub was then powered on and a script used to connect with sensor boards in the area of hub was ran. The script correctly found two of the sensor boards and proceeded to connect sequentially to each one and receive data from them. After the script informed all data has been received the script was terminated and a website hosted on the hub was opened on the tablet. Figure 5.2 shows what a sensor board and a BLE communication script reports while running.

```

INFO | XDK DEVICE 1: sd_srv: Init!
INFO | XDK DEVICE 1: sd_drv: Mounting SD card
INFO | XDK DEVICE 1: sd_drv: Removing old bak file
INFO | XDK DEVICE 1: sd_drv: Backing up current data file
INFO | XDK DEVICE 1: cfg_srv: Init!
INFO | XDK DEVICE 1: cfg_srv: Set pressure sensor delay to 10
INFO | XDK DEVICE 1: cfg_srv: Set humidity sensor delay to 10
INFO | XDK DEVICE 1: cfg_srv: Set temperature sensor delay to 10
INFO | XDK DEVICE 1: cfg_srv: Set ambient light sensor delay to 5
INFO | XDK DEVICE 1: cfg_srv: Configuration loaded!
INFO | XDK DEVICE 1: sensor_srv: Init!
INFO | XDK DEVICE 1: ble_srv: Init!
INFO | XDK DEVICE 1: BleEventCallBack : BLE powered ON successfully
INFO | XDK DEVICE 1: BleEventCallBack : Device Wake up succeeded
INFO | XDK DEVICE 1: Sensor 0 timer struck!
INFO | XDK DEVICE 1: Sensor 1 timer struck!
INFO | XDK DEVICE 1: Sensor 2 timer struck!

```

(a) Log from running sensor board

```

Discovered a new device: fc:d6:bd:10:3c:4f XDK_BLE_0
Connecting to XDK_BLE_0
Starting gathering data...
Finished gathering data from XDK_BLE_0

```

(b) Log from running BLE communication script

Figure 5.2: System in Action

5.1.2. Results

Data received from the sensor boards and displayed on the website was deemed correct by us; the environmental variables seemed adequate with weather conditions at the time of the test.

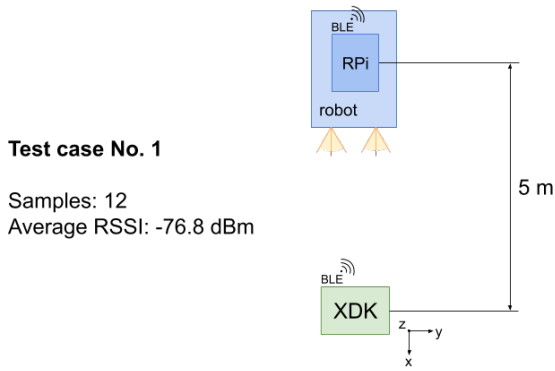
However during the test we encountered problems with BLE communication between the hub and the sensor boards. The hub often would not connect to a sensor board or the connection would fail while receiving data. We think the issue may have been caused by 3 things: seemingly crowded frequency bandwidth - there were multiple other Bluetooth devices and multiple wireless access points in the area, the script used to communicate between the hub and a sensor board makes use of a library which cannot handle problems with communication caused by crowded radio space or weak signal, and finally we think the XDK boards used as sensor boards use too little transmission power on their Bluetooth antennas.

Despite communication problems the test was successful and all previously planned system functionalities were working.

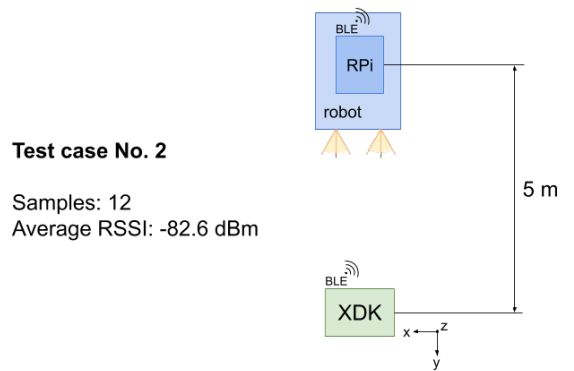
5.2. Measurements in Operational Environment [JK]

Additionally to a functional test we tested received signal power from the sensor board, commonly referred to as RSSI (Received Signal Strength Indicator (RSSI)). RSSI is a relative index and signal strength is a measurement value in dBm (decibel-milliwatt). The higher the dBm value, the stronger the signal. This helped us determine possible causes of communication issues we encountered while testing functionality of the system. The test was carried out in six scenarios as shown in Figure 5.3.

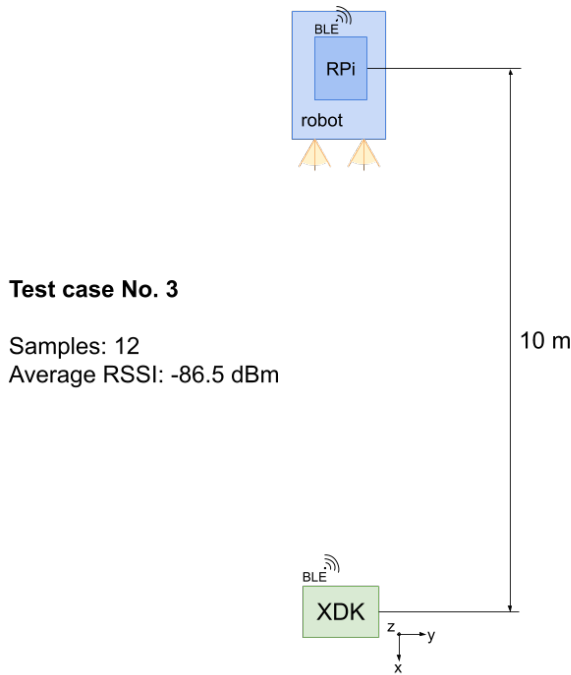




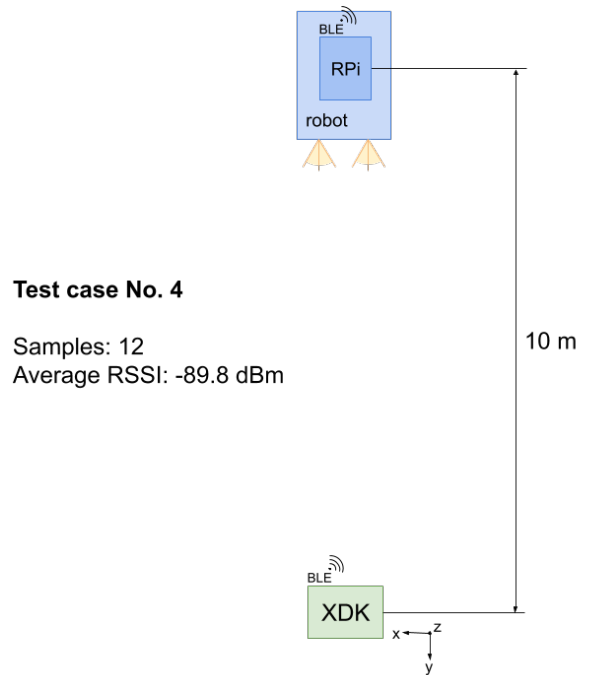
(a) Test case No. 1



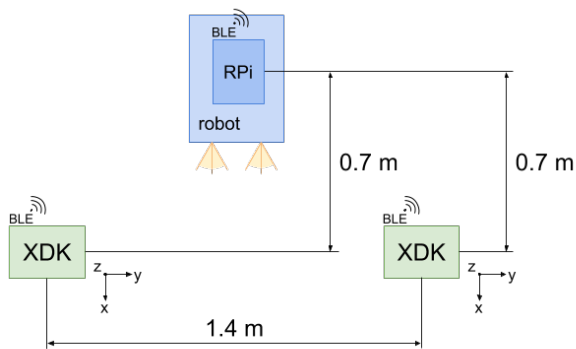
(b) Test case No. 2



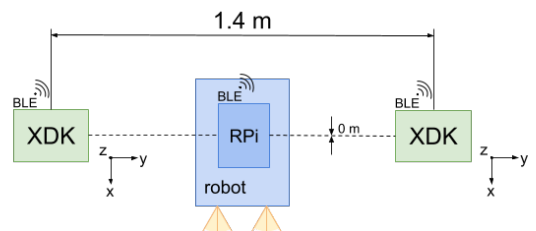
(c) Test case No. 3



(d) Test case No. 4



(e) Test case No. 5



(f) Test case No. 6

Figure 5.3: Test Cases - diagrams

5.2.1. Test environment [AD]

The test took place in front of the building of Faculty of Electronics, Telecommunications and Informatics, on Gdańsk University of Technology grounds. In all cases the XDK boards were placed on wooden platforms, 50cm from the ground (Figure 5.5), while the Raspberry Pi was placed on top of a robot, 20cm from ground (Figure 5.4). There were multiple wireless networks and Bluetooth devices present in the area.

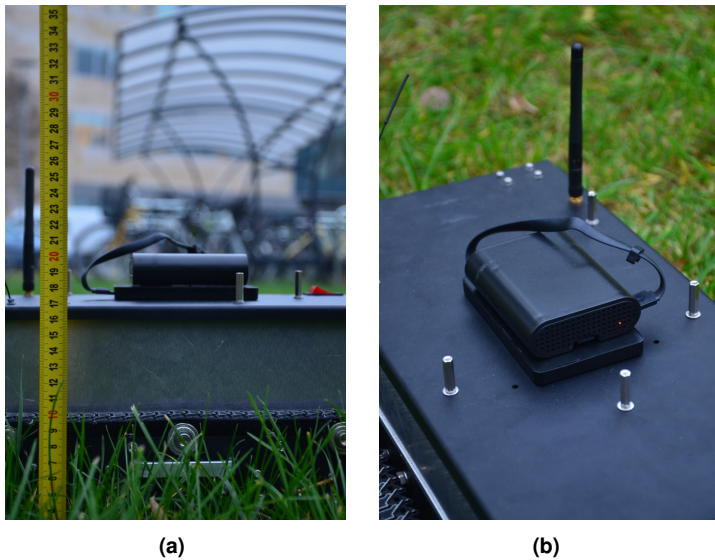


Figure 5.4: Test set-up: Raspberry Pi on robot

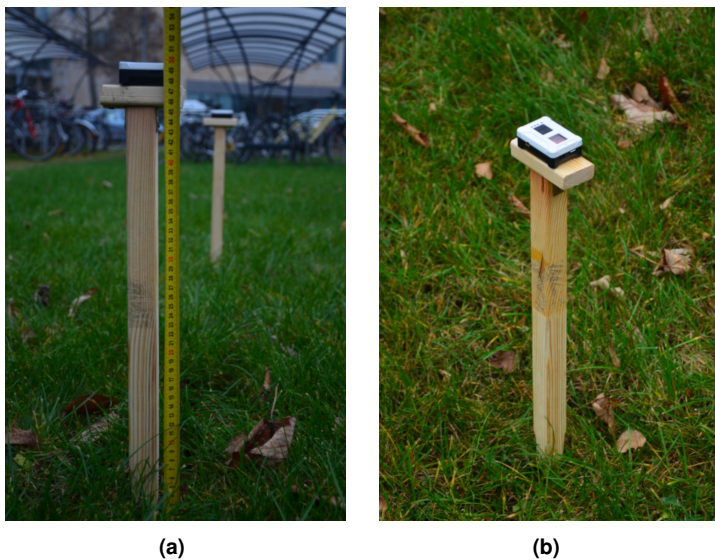


Figure 5.5: Test set-up: Bosch XDK located on a stake

5.2.2. Results [AD]

Each different test set configuration (Figure 5.3) has different average RSSI value, which is presented in Figure 5.6. The higher the average value is, the stronger the BLE signal is. Generalized rating of various signal strengths is shown in Table 5.1.

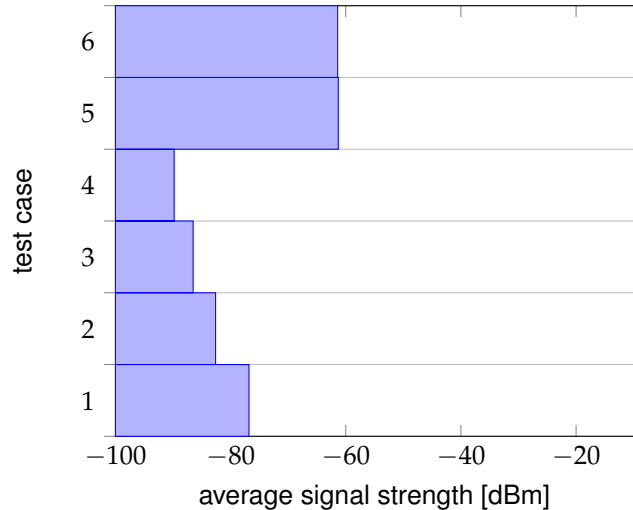


Figure 5.6: Average BLE signal strength measured through different test cases.

Table 5.1: Acceptable Signal Strengths

Source: www.speedcheck.org/wiki/rssi/ (Nov 27, 2019)

Signal Strength (dBm)	Rating
-30 dBm	Amazing
-67 dBm	Very Good
-70 dBm	Okay
-80 dBm	Not Good
-90 dBm	Unusable

5.2.3. Summary [JK]

Signal strength tests allowed for determining shortcomings of the prototype system. We think the XDK boards use too little transmission power on their Bluetooth antennas but we could not find information about that in the manufacturer's documentation. The hub's antenna has a sensitivity of -94dBm so according to our results it should still be able to work correctly in our tests. However it is very likely that the library used while developing Bluetooth Low Energy communication script is unable to handle those conditions and the script would need to be refactored with those problems in mind. Further testing should include influence of different orientations of a sensor board on signal strength, ability of the system to communicate with many more sensor boards and BLE message throughput to name a few.

6. SUMMARY [AD + JK]

6.1. Goals Achieved [AD]

The prototype was completed and it works as it was modeled, which is the most important achievement in this project. It was the first product we created from an idea to a real field test of a prototype, which was a great learning experience. Despite the time pressure and various challenges when creating a full version of the prototype success is visible in the gained experience. We familiarized ourselves with various types of cooperation, different methodologies of team work and standards of product development, including correct order of stages we had to go through.

6.2. The Biggest Challenges [AD]

Creating the presented system was the first time for us to work on such a large project. We never before went through the whole path of designing a product, from a beginning with a market research, further intensive planning and development. It takes a lot of time to analyze the market and conclude which part of the system should be designed by ourselves and how it should perform. Implementation process was not difficult for us as we already have certain experience in programming, but combining it with product design was new to us.

Bosch XDK was used in the system prototype as a sensor board. It was the first time we worked with a device development kit in such a way. It required getting familiar with the XDK-Workbench (Integrated Development Environment (IDE) for XDK) and XDK's SDK (Software Development Kit for Bosch XDK). Structure of XDK's SDK is based on hierarchy specified for the Bosch XDK device which is closely related to the software architecture of XDK. As a result we learned a lot about XDK software structures and available services responsible for BLE connection, sensor data gathering or data storing, which allowed us to implement suitable software functionalities to develop the prototype.

6.3. Future Plans [AD]

We already had basic knowledge of Unified Modeling Language (UML) modelling but it was the first time we used it in a real use case. Lack of experience in this field brought us a lot of difficulties, extending the time of UML models preparation. After much effort we began to understand how to think about the system and its components or functionalities to designed brief documentation of the UMLs and useful diagrams for future implementation process.

Bosch XDK is a relatively new product which is the reason for some of its shortcomings. At the beginning we planned to periodically move the sensor board to sleep in order to conserve power, unfortunately Bosch XDK does not enable different sleep modes. Before the implementation of a final product a new subcomponent would need to be found or an own solution created.

We planned to use nRF52840 (Nordic dongle) as the BLE communication module used by the hub. Nordic provides several detailed examples of BLE usage but documentation did not explain enough about BLE services' properties which forced a change in chosen solution to let the hub

connect to the sensor board. In the end we decide to use a Raspberry Pi single-board computer with a built-in BLE module. High availability of BLE examples and various libraries allowed for quick and easy BLE implementation. In the next iterations of the system a different solution could be chosen to move the BLE implementation away from the hub.

6.4. Personal View on the Topic

6.4.1. Anita Dworzańska

The possibility to create an Internet of Things (IoT) system was exciting to me as I am passionate about IoT solutions. Additionally, designing a whole product was interesting because of the complexity and various types of methods used throughout the final release of the system. It let me experience product development process, during which I had to play the role of a product owner, developer and an architect. The most noticeable part was to have an opportunity to manage in various stages, which is not a popular practice in development where everyone has their own narrow area of responsibility.

Most aspects of prototype creation were discussed between Jan and me, although it was my responsibility to be sure that the system is properly designed from the user side and it provides all of the basic functionalities of the final product. From the technical side I configured sensors in Bosch XDK and an HTTP server on the Hub, creating also a web page template. I was tasked with the implementation of BLE communication using an nRF dongle module but after much time spent on research without any progress and because of the complexity of Nordic implementation the decision was to move BLE functionality directly to Raspberry Pi which Jan enabled eventually. Beyond the strongly technical tasks I created the business and functional models, and activity diagrams, constantly making sure that the whole product model is consistent and it is not going beyond the scope of prototype capabilities. To enable convenient writing of the Bachelor Thesis I sourced a LaTeX template and adjusted it to our needs which simplified the document's creation process. I also watched over the project documentation, making sure that all essential data is included.

My cooperation with Jan was very productive, I feel we are a team and both are truly engaged which makes me convinced that together we are able to complete everything we plan to. We were discussing every single part of the created system where each of us had their own innovative ideas. This together built the impressive whole.

6.4.2. Jan Komińczyk

The project was quite exciting to me. It was an advanced project and up until that point I had experience only with small personal projects, not a whole product design. I was also excited about the embedded aspect of the project as I am interested in electronics and programming alike. I, just as Anita, had to accept few compromises; our thesis supervisor had a certain vision for the project but I think it was correct and shaped up our project to be a more coherent product rather than a basic programming project.

Throughout the project I focused more on technical aspects as that is what interests me the most. I developed system architecture for the sensor board (additional abstraction layer on top of XDK's SDK), implemented microSD service and driver, BLE service and driver, configuration



service and driver as well as main tasks (BLE handling and sensor data gathering). I also configured our hub by installing and configuring wireless access point management and a DHCP server. I wrote a script to handle database access used by a web server and a script to handle BLE communication on the hub. As to more high-level work I created sequence diagrams, wrote mostly technical sections of the Thesis and worked on language correctness in the work.

I think the subject of our project was an interesting one. Agriculture 4.0 is, as some sources write, the future of farming technology and working towards something that can help in the future is rewarding.

The project was a very good learning experience. I enjoyed working alongside Anita. I think we complemented each other.

REFERENCES

- [1] *What is Agriculture 4.0?* proagrica.com Proagrica, n.d. Web. 21 Nov. 2019.
- [2] Shoab Farooq, Muhammad, Shamyra Riaz, Adnan Abid, Kamran Abid, and Muhammad Azhar Naeem. *A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming* ieeexplore.ieee.org IEEE, 25 Oct. 2019. Web. 30 Nov. 2019.
- [3] Ayaz, Muhammad, Mohammad Ammad-Uddin, Zubair Sharif, Ali Mansour, and El-Hadi M. Aggoune. *Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk* ieeexplore.ieee.org IEEE, 01 Aug. 2019. Web. 30 Nov. 2019.
- [4] *Cesens Mini* partners.sigfox.com Sigfox, n.d. Web. 13 Aug. 2019.
- [5] *OneSense Agriculture* evvos.com Evvos, n.d. Web. 13 Aug. 2019.
- [6] *SmartCrop System* smartfield.com Smartfield, n.d. Web. 13 Aug. 2019.
- [7] *SmartWeather* smartfield.com Smartfield, n.d. Web. 13 Aug. 2019.
- [8] *CropInsight* smartfield.com Smartfield, n.d. Web. 13 Aug. 2019.
- [9] Bluetooth Special Interest Group. *Bluetooth Specification Version 4.0* bluetooth.com Bluetooth SIG, 30 Jun. 2010. Web. 24 Nov. 2019.
- [10] *Cross Domain Development Kit* bosch-connectivity.com Bosch, n.d. Web. 20 Nov. 2019.
- [11] *Raspberry Pi Foundation - About Us* raspberrypi.org Raspberry Pi Foundation, n.d. Web. 23 Nov. 2019.
- [12] *BLE Definition* techterms.com TechTerms, 8 Mar. 2019 Web 23 Nov. 2019.
- [13] *Bluetooth* en.wikipedia.org Wikipedia, 28 Nov. 2019. Web. 24 Nov. 2019.
- [14] *Bluetooth Vs. Bluetooth Low Energy: What's The Difference?* link-labs.com LinkLabs, 1 Nov. 2015. Web. 10 Nov. 2019.
- [15] *Bluetooth Low Energy* en.wikipedia.org Wikipedia, 21 Nov. 2019. Web. 24 Nov. 2019.

