Aalto-yliopisto, Sähkötekniikan korkeakoulu

ELEC-D0301 Protopaja

2023

<u>Loppuraportti</u>

Project #04 Helvar Oy Ab

MpW sensor (Multipurpose wireless sensor)

Date: 7.9.2023

Tuomas Maijala

Ilmari Oivanen

Iiro Naulapää

Muhammad Qaisar Nadeem

Information page

Students

Tuomas Maijala

Ilmari Oivanen

Iiro Naulapää

Muhammad Qaisar Nadeem

Project manager

Tuomas Maijala

Sponsoring Company

Helvar Oy Ab

Starting date

1.6.2023

Submitted date

7.9.2023

Tiivistelmä

Hankkeemme tarkoituksena oli tehdä prototyyppi monikäyttöisestä langattomasta anturista, jota voitaisiin käyttää nykyaikaisissa toimistoympäristöissä tai missä tahansa muussa sisätilassa keräämään ympäristötietoja ihmisten käyttäytymisestä. Helvar ei ollut lainkaan tiukka tuotteellemme asetettujen vaatimusten suhteen, mutta lämpötilan ja ilmankosteuden seurantaa ei tarvittu, koska muut laitteet hoitivat ne jo valmiiksi. Meidän oli itse selvitettävä, mitkä ominaisuudet sisällytettäisiin tuotteeseen, mikä oli jo pieni haaste itsessään. Harkittuamme taitojamme ja käytettävissä olevaa aikaa päätimme keskittyä liikkeen havaitsemiseen, koska se näytti olevan tärkein näkökohta sisätilojen valvonnassa. Myöhemmin lisäsimme myös komponentin, jonka avulla pystyimme havaitsemaan, milloin ovi on auki tai kiinni.

Laite seuraa toimintaa sisätilassa ja havaitsee esimerkiksi oven sulkeutumisen, tuolin liikkeen tai kahvinkeittimen käytön. MpW-anturi sisältää 3-akselisen kiihtyvyysanturin, jota käytetään erilaisten asioiden liikkeen ja tilan seuraamiseen. Laitteessamme on eri käyttötilanteita varten erilaisia tiloja, ja käyttäjä voi vaihtaa tilojen välillä napilla ja RGB LEDillä, joka näyttää käytössä olevan tilan. Laite käyttää Bluetooth Low Energyä tietojen lähettämiseen, kun laite havaitsee riittävästi liikettä. Anturin herkkyys riippuu käytössä olevasta tilasta.

Odotettu käyttäjä on henkilö, joka haluaa kerätä tietoja sisätiloista seuratakseen niiden ominaisuuksia ja käyttöä. Teimme tuotteesta helppokäyttöisen, jotta jopa henkilö, jolla ei ole teknistä taustaa, voi käyttää sitä. Vaikka laitteemme yksinkertaisuudessa on etunsa, se myös rajoittaa käyttötapauksia ja tuotteemme laajennettavuutta. Muiden sisäänrakennettujen antureiden puuttuminen voi olla haitallista, eikä käyttäjä voi helposti lisätä uusia antureita. Tuote ei myöskään sovellu ulkokäyttöön.

Abstract

Our project was about making a prototype of a multipurpose wireless sensor node that could be used in modern office environments, or in any other indoor space, to collect environmental data on human behavior. Helvar wasn’t strict at all with the requirements for our product, but monitoring temperature and air humidity was not needed as other devices already took care of that. We had to figure out ourselves which features would be included, which was already a small challenge in itself. After consideration of our skills and available time, we decided to focus on sensing movement, since that seemed to be the most important aspect of monitoring indoor spaces. Later we also added a component which allowed us to sense when the door is open or closed.

The device tracks actions in an indoor space and detects door closings, chair movement or coffee machine usage for example. The MpW sensor includes a 3-axis accelerometer sensor which is used to track the movement and state of various things. There are different modes to match the different use cases of our device, and the user can switch between the modes by using a button and an RGB LED which will show the mode in use. The device uses Bluetooth Low Energy to send data when the device detects enough movement. The sensitivity of the sensor depends on the mode in use.

The expected user is a person who wants to collect data from indoors, to monitor its aspects and usage. We made the product easy to use so that even a person that does not have a tech related background can use it with ease. Although the simplicity of our device has its benefits, it also restricts the use cases and extendibility of our product. The lack of other built-in sensors can be detrimental and new sensors cannot be easily added by the user. The product is also not for outside use.

Table of Contents

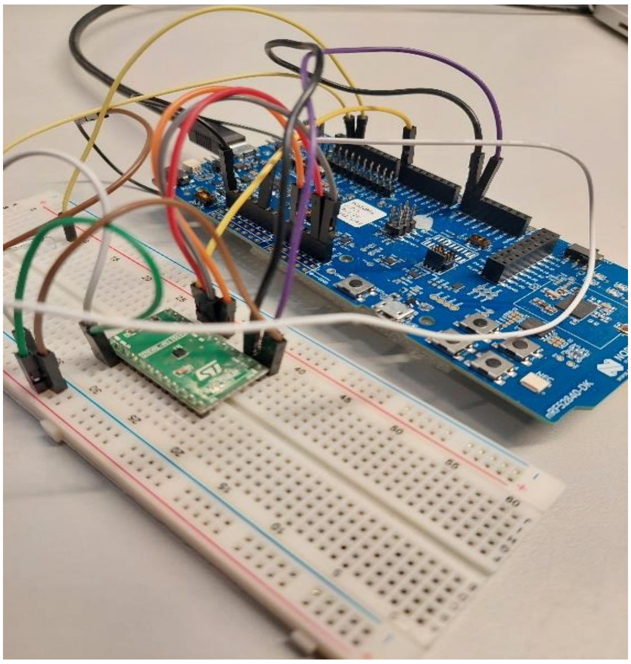
Tiivistelmä 3
Abstract 4
Table of Contents 5
1. Introduction 6
2. Objective 6
3. Results: PCB design 7
   3.1. Overall board design 7
   3.2. Components 9
4. Results: Software 10
   4.1. Development 10
   4.2. Code 11
5. Results: Bluetooth 12
6. Results: Enclosure 13
7. Reflection of the Project 14
   7.1. Reaching objective 14
   7.2. Timetable 15
   7.3. Risk analysis 15
8. Discussion and Conclusions 16
   8.1. Tuomas 16
   8.2. Ilmari 17
   8.3. Iiro 17
   8.4. Qaisar 17
   8.5. Conclusions 18
List of Appendixes 19
References 20

1. Introduction

In modern office environments, there is a growing need to collect reliable and accurate information about various aspects of the workspace. Multipurpose sensors play a good role in fulfilling this requirement by gathering data on different environmental parameters and user behavior. These sensors are designed to be versatile, cost-effective, and easy to install, making them ideal for monitoring office, and other indoor spaces efficiently. The big picture is to create a large network of sensors to garner a comprehensive understanding of the usage of objects and activities happening within space.

2. Objective

The goal of the project was to make a multipurpose sensor node for gathering ground truth data from trivial-sounding things, such as the usage of chairs, doors or whether a coffee pot has finished making coffee. Helvar required a prototype to be around the size of a match box and have low power consumption. The sensor would have the option to switch between different modes depending on what events the user would want to detect. When an event happened, the sensor would send a Bluetooth advertisement to Helvar's office's Bluetooth mesh. For example, if the sensor detected a chair moving, it would send an advert titled "Chair" to the mesh.



Picture of our first implementation: a development kit with the accelerometer adapter board connected via wires.

3. Results: PCB design

The PCB needed to be as small as possible so the sensor would not stick out too much and could be easily attached to even small surfaces. To ensure this, we needed to choose our components wisely and design the board circuitry efficiently. To achieve the small size, it was important to focus on energy efficiency, because bigger batteries would naturally require more space. We also felt that energy efficiency was important because the device would probably be in the same place for a long time without anyone interacting with it, so having it constantly run out of batteries would be annoying for the user.

3.1. Overall board design

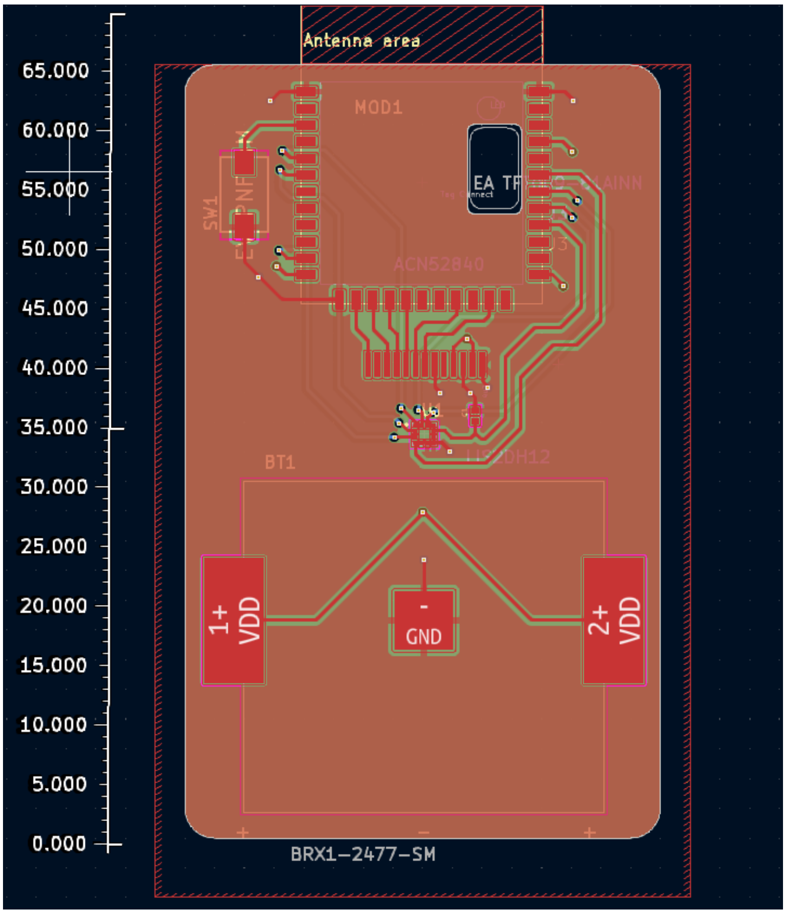
During the project, we successfully developed two distinct board designs. While each design had its own advantages and disadvantages in terms of aesthetics and functionality, we ultimately had to abandon the first design due to specific issues and time constraints.

First design:

The first design had more of a user-friendly approach. The key feature and major difference component-wise was the lcd display, which was supposed to show the user which mode the device was on, using drawn images or simple text. To save power, we were thinking of having the display be on for only a few seconds when choosing modes, so the power consumption would be minimized when the device was left alone.

The PCB had 4 layers, which in all honesty was unnecessary but this was the designer's first ever PCB, so he didn't have the best understanding when it came to basic PCB design. 4-layer board made the power and ground connections easier, because they both had a dedicated layer.

This board design had a couple issues: The lcd display's backlight pins were directly connected to the VDD, which meant that they would always be on when the battery was in. This would always consume way too much unnecessary power. There were a couple possible fixes for this, but we also had issues with the display code, so we just had to accept that a simpler design would be safer because the deadline was slowly approaching. Another thing was the location of the battery. On this design, the battery holder was on the front side just like all the other components. We realized that we could make the PCB much smaller by having the battery holder on the backside.



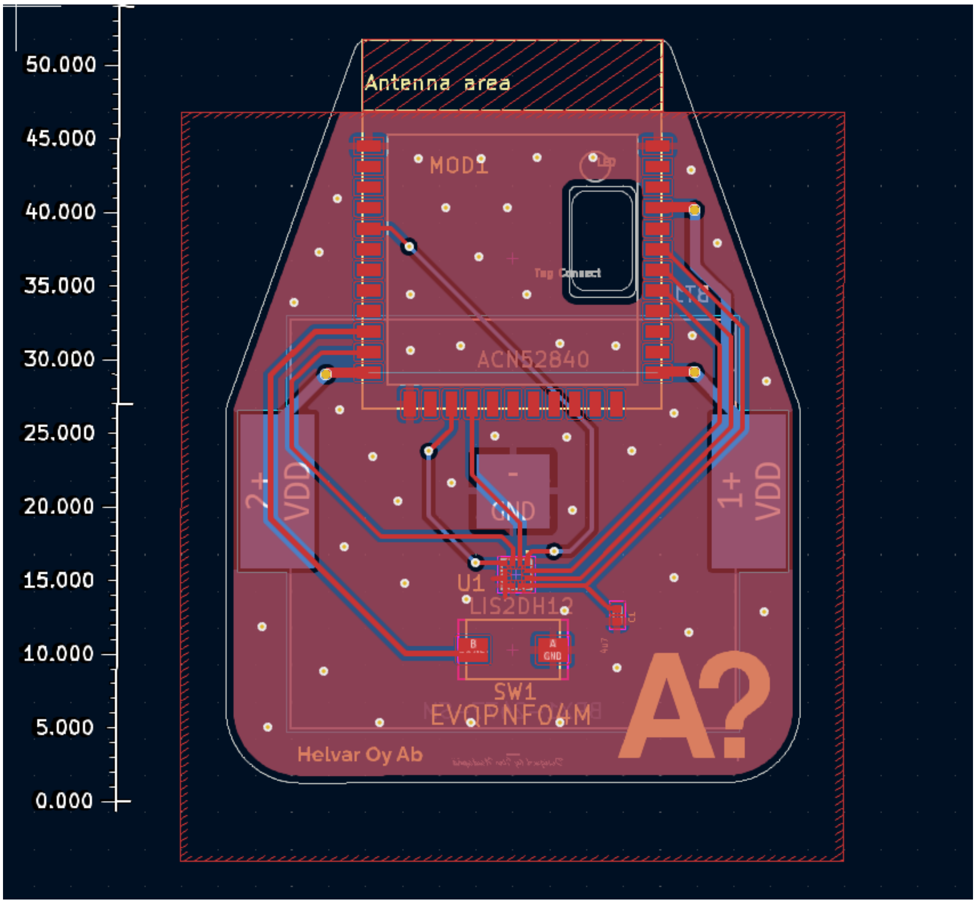
Picture of the first PCB-design

2nd / Final design

This design ended up being way simpler than the first one, with the display removed. The first design was more user friendly, but we think that with devices like these, low power consumption is a priority. This design is also more compact, which increases its places of usage.

This was a 2-layer PCB, because our designer had learnt from his mistakes. With this design we also put the battery holder on the backside, which was the biggest reason for the size improvement. This naturally made the device a little thicker, but in terms of shape, it was better than having an unnecessary long design.

Because we scrapped the display, we went with a lot simpler mode indicator: The Bluetooth module's built-in RGB LED. In other words, different LED colors indicate different modes. This means that the user must know the color-mode combinations beforehand, but that isn't a big issue.



Picture of the final PCB-design

3.2. Components

Our objective was to identify components that were both compact and energy-efficient, while maintaining simplicity and ease of programming. To collect motion data, we opted for the LIS2DH12 accelerometer[1] due to its exceptionally small form factor and widespread utilization in compact designs. We decided to use the specific sensor because of its readily available documentation that ensures a straightforward integration process into our prototype.

For the Bluetooth part we decided to use the Aconno Acn52840 BLE module[2] which seemed very beginner friendly and compact. The module featured the powerful nrf52840 chip by Nordic Semiconductors[3], which is also a popular choice for wide range of applications involving Bluetooth. By choosing a module instead of designing our own antenna we saved time and space while also ensuring that the prototype would have an efficiently working antenna. This module also had a 6-pin Tag-Connect port, which could be used for programming the micro-controller. This port was located on the backside of the module, so we had to design a hole in the location of that port in the PCB. In our case, we programmed the chip by using Aconno's cable[4], which had the Tag-Connect 6-pin connector with a USB-connector that could be plugged into the computer. This way the chip could be flashed using the drag and drop method.

For the energy source we chose a CR2477 coin cell battery[5], because in terms of voltage and overall energy consumption it seems good and would make the prototype last a long time. If we wanted to make the prototype even smaller, there was an option to use even smaller coin cell batteries, but we deemed that unnecessary because the size/energy balance of the CR2477 was good enough for us.

Our display of choice ended up being EA TFT009-81AINN[6] which was a very small LCD display module with flexible mounting. This display seemed very good for our prototype because of its thin, small size, which would make it not affect the overall size of the PCB much. The flexible mounting made it easy to fold it nicely on top of other components without sacrificing a lot of space.

4. Results: Software

4.1. *Development*

We used the nRF Connect SDK to build the software for our product [7]. The SDK integrates the Zephyr Project, which is a real-time operating system (RTOS) optimized for resource-constrained devices [8]. The SDK also has a free extension for VS Code so that became our IDE of choice. With these specialized tools, usage of the C programming language was basically necessary which meant the development process was quite challenging at first without prior experience. Luckily for us, the documentation of the SDK and Zephyr was detailed and helped us a lot during the project [9].

Zephyr was also useful in other ways. It included updated drivers for hundreds of popular devices and multiple sample applications. The drivers especially were a huge help as we weren't skilled enough to write our own, but we also couldn't have progressed as fast if we hadn't learned the basics from the sample code.

4.2. Code

The goal for the software side was to control and communicate information between the different hardware pieces and process the data from these parts accordingly. The first part of the software is to define each GPIO-pin for each component. In our case these components are the button, RGB LED, reed switch, accelerometer, and the Bluetooth module. The different components and the way they communicate with the chip also needed configuration. For example, the accelerometer uses Serial Peripheral Interface (SPI) to input and output data, so the maximum supported frequency had to be defined as well.

```
&spi0 {
    compatible = "nordic,nrf-spi";
    status = "okay";
    pinctrl-0 = <&spi0_default>;
    pinctrl-1 = <&spi0_sleep>;
    pinctrl-names = "default", "sleep";
    cs-gpios = <&gpio0 27 GPIO_ACTIVE_LOW>;
    accelerometer: lis2dh@0 {
        compatible = "st,lis2dh";
        reg = <0>;
        irq-gpios = <&gpio0 15 GPIO_ACTIVE_HIGH>, <&gpio0 17 GPIO_ACTIVE_HIGH>;
        spi-max-frequency = <10000000>;
        label = "lis2dh";
    };
};
```

Picture of the SPI and accelerometer configuration

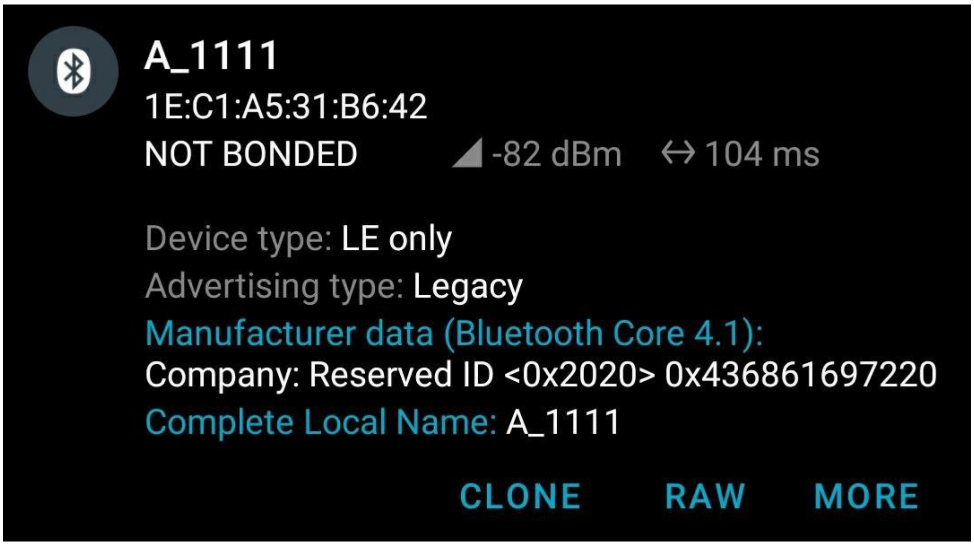
After this we initialize the button, LEDs, sensor drivers and our Bluetooth communication. The next part of our code is the logic part of our code. Here we make it, so a button press switches the LED color and the sensor mode. The LED colors for the modes are red for chairs, blue for door and green for coffee. By combining these colors, it is also possible to add more modes in the future.

These modes' main purpose is to set a different threshold for each application. For example, the coffee mode, which purpose is to detect the small vibrations of the coffee machine when coffee is brewing, has high thresholds on the x and y axis and a low threshold on the z axis. This is to minimize possible accidental event detections from moving the device and instead focus on the small changes in the z axis aka the downward position. We found out through testing that vibrations only really change the values in this axis. The code compares the previous sensor readings to the new ones and if the subtraction exceeds the threshold, it counts it as an event. In the case of the "door" mode, an event is counted when the device is no longer detecting a magnet nearby. While no magnet is being detected it will send these adverts until the mode is changed or the door is closed again. The company requested this function as they want keep track of changes in air quality and how airflow affects it. The modes can also be used for different applications than their set mode names. For example, coffee mode can be used to detect many devices that vibrate when they are on, like a microwave. The chair mode is good for detecting simple movements of any objects, so for example the usage of cabinets.

In addition to the mode algorithms, we implemented some minor functionalities to improve our product. One of those was a timer which turned the LEDs off after a few seconds to keep the power consumption low.

5. Results: Bluetooth

The goal here was to send an advertisement whenever a sensor event happens via Bluetooth to our company's Bluetooth mesh, where they would process it further. We are using Bluetooth Low Energy with Bluetooth 4 for this device. The advertisement is non-connectable. Whenever a sensor event happens, our source code sends out a Bluetooth advertisement. In the case of the "coffee" mode, the code waits for several consecutive events to happen before sending out an advertisement. This is to simplify the meaning of the advert and to not overload the device receiving these adverts. The Bluetooth advertisement data sent has the mode that the device is on and the name of the sensor. The name is used to identify different MPW-sensors from each other. The mode is sent via the Manufacturer Specific Data (type 0xFF) and is displayed with bytes on the receiving end. The Bluetooth advertisement will stop after three seconds, and a new sensor event must happen for the device to send another advertisement.



Picture of the advertisement data on the receiving end (the bytes spell out the mode; in this case “Chair”)

6. Results: Enclosure

During the course made enclosures for both designs. Our designers had some experience in 3d-modelling with Fusion 360, but this was his first time designing enclosures so it took some trial and error. With the case design’s we wanted to make sure that the PCB wouldn’t move around too much when shaking the product. This could mess up with the sensor measurements.

1st Design’s Enclosure

This enclosure was kind of bulky but came out nicely. The user would slide the PCB inside the bottom part and then snap the top part on top, locking the PCB inside. The top part also had a spot for the display and a hole for the button. The measurements were quite good for a first design, because the PCB was tightly shut inside and couldn’t move around. However, the lid was very tightly attached to the bottom part and was quite difficult to remove. This was something we aimed to fix in our next design.

2nd Design’s Enclosure

This design was more compact, partly because the PCB was smaller, but also because we made the walls thinner. The lid was also easier to remove, but still tight enough so it wouldn’t accidentally fall off. The lid had holes for the button and LED, so that the user could see the mode colors and press the button to switch modes. The designs shape raised some questions because it wasn’t symmetrical. The reason for this shape was the idea to make the product resemble an “A” shape, as in Aalto. Unfortunately, we didn’t have time to make this dream into reality, so we were left with this oddly shaped box.

We also made a cap for the button, to make the product look and feel nicer. The way this button cap works is that the user puts it on top of the button, and then closes the lid. The button cap’s bottom surface is larger than the hole for the button, so the lid holds it in place. The cap sticks out of the hole so the user can still easily press the button.

7. Reflection of the Project

7.1. Reaching objective

Our lack of experience in programming and working with Bluetooth technology was a significant hurdle. We invested a large sum of our time and effort into learning the basics of C and Bluetooth advertising. Many of the problems we faced we had to trial and error because information and help about our issues was scarce, since we were using a lot of brand-new software and even hardware. Our biggest help was the Nordic Semiconductors own tutorials and forum answers, which we had to adapt to our circumstances.

Our most detrimental challenge came on the last weeks, when we had trouble connecting our small sensor to our PCB. Also because of this we were unable to properly test our code and see if it was working as we wanted it to. We were running out of sensors, usable PCBs, and time. However, through some soldering heroics and last-minute software modifications we managed to get the whole device working as intended in time for the demo.

The basic goals we set in our project plan were to detect door closing, coffee machine and chair usage. The device was to send out data using Bluetooth. With some last-minute decisions, we managed to achieve these goals. We also planned on being able to switch between modes by using switches and/or a display. This part we couldn’t get to work in time and as planned. We ended up using a simple button and discarded the display from our device. This was due to us running out of time and some unfortunate errors. With better time management, prioritization and more decisive decision making, we could have added more to the project, while also reducing our stress. This was an event that all of us learned a lot from. In the end, we managed to make our device compact size, easy to use and energy efficient, which were some of the additional goals we had set.

7.2. Timetable

The workload given to the Work Packages was surprisingly accurate in most cases. On the first half of the Work Packages, what was different was the division of the hours within the Work Packages. Our expectations were different with the amount of time that we went into studying each subject, since many of them were completely new to us. The real division of hours was that more time than expected went into learning, but less time went into doing the task in a Work Package. On the latter half some of the estimated times required were different. On WP6 (system integration) we ended up spending quite a lot more time than we estimated. We also should have prioritized this part more as it is essential to get a working product and fixing mistakes in this part can take weeks because you might need to order new parts. Work Package 7s time estimation was quite accurate. On WP8 we ended up spending less time than expected since our code worked pretty well already in the first tests and also because we didn't have much time at that point. On WP9 we overall spent about the same amount of time, but we spent more time on documentation and less on the presentation than expected.

7.3. Risk analysis

In the initial Project plan, we identified and listed several foreseen risks. However, during the project, it's common for new risks to emerge, and some of the previously identified risks can also materialize. Let's review what happened:

Foreseen Risks:

The project involved working with programming languages, and as newcomers to programming, technical challenges were expected. We faced difficulty in sending specific sensor data as an event over Bluetooth advertisement at the start. This could be due to complexities in the Bluetooth protocol or issues with data formatting.

The risk of delays in receiving components and PCBs was foreseen, as this can always impact the project timeline.

New Risks:

- A new risk was the challenge of attaching a display to the board successfully. This caused compatibility issues and difficulties in interfacing with the display module.
- Another risk was to solder the sensor successfully on the PCB. As the sensor in too tiny itself and the connections underneath were barely visible.

Realized Risks and Risk Management:

We faced difficulties in certain aspects of programming and working with the new technologies. To manage this, we were actively learning, seeking assistance from relevant resources, and conducting research to improve programming skills.

As expected, there are delays in receiving parts. We had to focus on tasks that did not depend on the delayed components.

Risk Management Strategies:

To address the challenges faced during the project and mitigate the risks, we adopted several strategies:

- We were actively learning and seeking guidance from online resources and tutorials.
- We were dedicating time to troubleshooting and debugging the code, sensors, and Bluetooth communication to identify and resolve issues.
- Regular communication and collaboration with project assistants were essential to share insights, brainstorm solutions, and support each other throughout the project.

By utilizing available resources and keeping the project objectives in mind, we managed to make the sensor compatible with the board, successfully connected it via I2C and SPI, and managed to send beacons over Bluetooth.

8. Discussion and Conclusions

Overall, we all learned about prototyping and working in a group of people with varied skillsets to reach our goals. We all learned a good deal about teamwork, communication and what is required to complete a project. However, since we divided the work between us based on our skills and interests, everyone ended up focusing on at least slightly different things thus also making everyone’s learning experience different from others.

8.1. Tuomas

Overall, I learned a lot of new things during this project since, as a first-year student, most of the things were completely new to me beforehand. The subject I dove most deep into was Bluetooth and how it works in collaboration with sensors data and electronics in general. As a project manager I also learned how to keep up good communication between the company and the project team. The project though me the importance of time management and thorough planning of a project. I also came to learn the importance of dividing work between individuals. I learned how to collect and process raw sensor data and in general code in C using the nRF Connect SDK along with VS Code to communicate with a PCB. Along with all this I also learned some basics in circuitry and how to configurate a PCB in a software.

8.2. Ilmari

As a first-year computer science student, I had prior experience in programming, which was useful to some degree. The problem was that I had mostly used Scala and a bit of Python beforehand, so programming in C was new and the first thing to learn. There's a lot of differences between those languages and I will try to learn C better at some point in my studies. For this project, only the fundamentals were needed as most of the missing knowledge was about electronics and how to build software for them using somewhat specialized tools. Specifically, I learned how to work with the nRF Connect SDK, Zephyr Project RTOS and some physical tools such as a development kit and a programmer to program and flash the nrf52840 chip.

8.3. Iiro

For me, this project was a very eye-opening experience when it comes to PCB design. As a first-year student this was my first-time doing circuit design and it taught me a lot about the general practices when it comes to designing PCBs. Overall, it was great getting to know how KiCad works, and I hope I can apply this knowledge in future courses.

In general, working as a group like this and making decisions together taught a lot about the importance of communication and workload sharing. Making tough decisions, such as deciding to scrap the first design and going with a simpler one was irritating, but at the same time a great learning experience and I'm glad that we had the willpower to give up on something that we may not have gotten to work.

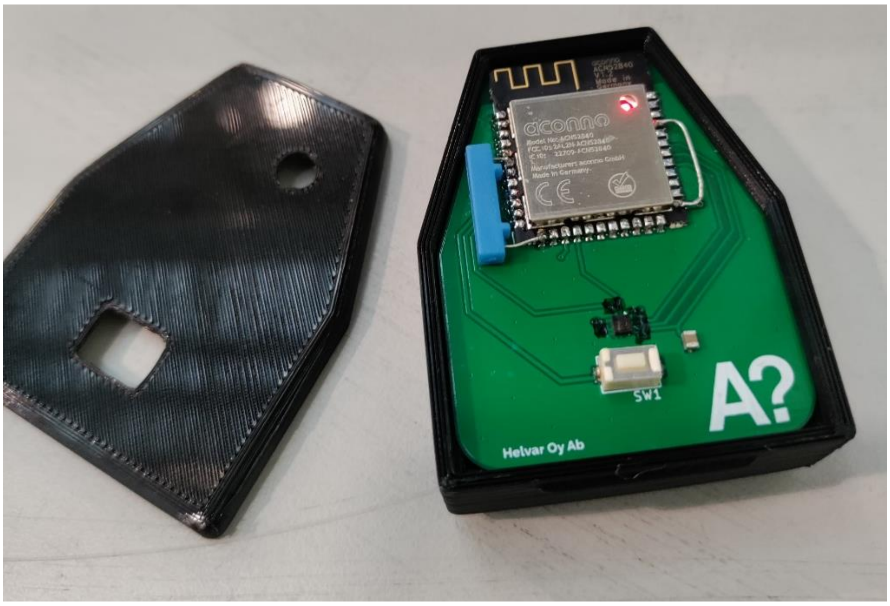
8.4. Qaisar

As an information technology student, I initially assumed the project would be easy, but when I started programming the microcontroller, I realized it was harder than I expected. However, I learned a lot during the project, like working with datasheets, using different ways to communicate between devices like I2C, SPI, and UART, and understanding how Bluetooth works, including Bluetooth BLE. I also got familiar with some programming software like Visual Studio Code, Seger Embedded Systems, and Arduino IDE. As I have good experience in electronics work, I enjoyed the PCB assembly and use of different new tools. Like I have never used an oven to solder the extremely small components on a PCB. We faced a great challenge to solder the sensor as it was extremely tiny itself and the connections were barely visible but with a lot of effort, we managed to successfully solder it. Overall, it was a great experience, and I hope I can use what I learned in future courses.

8.5. Conclusions

As a team with no prior experience in prototyping, the process was quite hard, especially since we also needed to learn a lot of new skills related to programming and electronics. During the course it became very clear to us that because of our inexperience, the biggest difficulties were the ones we could not predict. The assistants warned us about some of the possible troubles we would face but unfortunately, we didn't understand or listen to them enough. We definitely could have asked for help from the assistants more often. Also, in addition to the assistants, the people at Helvar were supportive and nice to work with. Without their help, the product would have been worse, so big thanks to them.

By the end of this course, we managed to achieve results that satisfied us. However, we still decided to try improving the prototype to give Helvar a more polished product and to work out some of the smaller problems that we didn't manage to solve in time. The experience overall has been valuable, and we look forward to applying our knowledge in the future.



Picture of the finished product on the demo day

List of Appendixes

- Software: https://version.aalto.fi/gitlab/oivanei1/mpwsensor-testi
- Product poster: HelvarProject_poster
- Demo video: Link
- KiCad design files:
  - Design 1
  - Design 2
- Gerber files:
  - Design 1
  - Design 2
- 3D-printed enclosure:
  - Design 1
  - Design 2

References

[1] https://www.st.com/en/mems-and-sensors/lis2dh12.html

[2] https://aconno.com/products/acn52840/

[3] https://www.nordicsemi.com/products/nrf52840

[4] https://aconno.com/products/aconno-acnprog/

[5] https://ruuvi.com/fi/tuotteet/cr2477t-paristo/

[6] https://www.mouser.fi/ProductDetail/Display-Visions/EA-TFT009-81AINN?qs=Wj%2FVkw3K%252BMAVh2P%252BL0Z6dg%3D%3D

[7] https://www.nordicsemi.com/Products/Development-software/nrf-connect-sdk

[8] https://www.zephyrproject.org/learn-about/

[9] https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/index.html