Aalto-yliopisto, Sähkötekniikan korkeakoulu
ELEC-D0301 Protopaja
2024

Final Report

Project #1 Helvar
Seeing the Light

Date: 30.8.2024

Vasilina Toporova
Artem Kiarkianen
Ben Nguyem
Kirill Levo
Huy Vu

Information page

Students
Vasilina Toporova
Artem Kiarkianen
Ben Nguyem
Kirill Levo
Huy Vu

Project manager
Vasilina Toporova

Sponsoring Company
Helvar Oy Ab

Starting date
3.6.2024

Submitted date
30.8.2024

Tiivistelmä

"Seeing the Light" -projekti toteutettiin Aalto-yliopiston opiskelijoiden tiimin toimesta yhteistyössä Helvar Oy:n kanssa. Projektin tavoitteena oli kehittää prototyyppi anturi, joka pystyy mittaamaan tarkasti valaistusvoimakkuutta, määrittämään valon suuntaa ja erottamaan luonnollisen ja keinotekoisen toisistaan. Anturi suunniteltiin akkukäyttöiseksi, ja tiedonsiirto toteutettiin Bluetooth Low Energy (BLE) -mainostuksen avulla.

Projektin aikana tiimi eteni useiden keskeisten vaiheiden kautta, mukaan lukien suunnittelu, konseptointi, taitojen hankkiminen, vaatimusanalyysi, suunnittelu, prototypointi ja toteutus. Erityistä huomiota kiinnitettiin anturin tarkkuuden ja luotettavuuden varmistamiseen erilaisissa valaistusolosuhteissa. Lopputuote esiteltiin sekä Aalto-yliopiston demopäivässä että Helvarin toimistolla, missä se onnistuneesti esitteli kykyjään.

Projekti tarjosi myös arvokkaita oppimismahdollisuuksia tiimin jäsenille, mahdollistaen osaamisen kehittämisen esimerkiksi piirilevysuunnittelussa, anturiteknologiassa ja 3D-prototyypin luomisessa. Projektin tulokset vastasivat alkuperäisiä tavoitteita, ja prototyypin odotetaan edistävän Helvarin jatkuvaa tutkimus- ja kehitystyötä.

Abstract

The "Seeing the Light" project was undertaken by a team of Aalto University students in collaboration with Helvar Oy. The project's objective was to develop a prototype sensor capable of accurately measuring illuminance, determining light direction, and distinguishing between natural and artificial light sources. The sensor was designed to be battery-operated, with data transmission facilitated via Bluetooth Low Energy (BLE) advertisements.

Throughout the project, the team progressed through several key phases, including planning, conceptualization, skill acquisition, requirement analysis, design, prototyping, and implementation. A particular emphasis was placed on ensuring the sensor's accuracy and reliability across different lighting conditions. The final product was demonstrated both at the Aalto University demo day and at the Helvar office, where it successfully showcased its capabilities.

The project also provided valuable learning opportunities for the team members, allowing them to develop expertise in areas such as PCB design, sensor technology, and 3D prototyping. The outcomes of the project met the initial objectives, and the prototype is expected to contribute to Helvar's ongoing research and development efforts.

Table of Contents

1. Introduction

Lighting control systems are using light sensors to measure (estimate) how much light is on the surface below them. They are measuring light, which is reflecting to them. Light is generated by artificial lighting and daylight, which penetrates space.

This information is used to control luminosity. When sensors “see” more light compared to their set value, luminaires are dimmed down. If they see less light, luminaires light output is asked to increase. Feature is called “constant light” or “daylight harvesting” or DCL (daylight linked control).

Since these sensors are measuring reflected light, they need to be configured for each location they are used. Surface color (reflectance) influences how much light these sensors see. Unfortunately, reflectance also changes over the time. To be able to calibrate and auto calibrate sensors it would be good to know more about real lighting conditions at the surface. Hence, the project is about creating a device that can harvest information, particularly answer these three questions:

- How much light is there?
- What kind of light source (natural or artificial)?
- From which direction light comes to the surface?

2. Objective

The primary objective of the "Seeing the Light" project was to develop a functional prototype sensor for Helvar Oy that could accurately measure illuminance, determine the direction of incoming light, and differentiate between natural and artificial light sources. The sensor was to be designed as a compact, battery-operated device capable of transmitting data via Bluetooth Low Energy (BLE) advertisements. The project aimed to create a solution that could be integrated into Helvar's existing systems, contributing to their research and development initiatives. Additionally, the project provided an opportunity for the student team to enhance their technical skills and apply theoretical knowledge in a real-world context.

3. Technical report

3.1. 3D Designs

3.1.1. Earlier design

We discussed the possibility of creating a polyhedron that allows multiple sensors to mount on different surfaces, allowing measurement from different angles and identifying the light source direction. The model allows the mounting of up to 13 sensors. Notations for the result were discussed based on the vector angle, allowing the light source to be represented accurately.

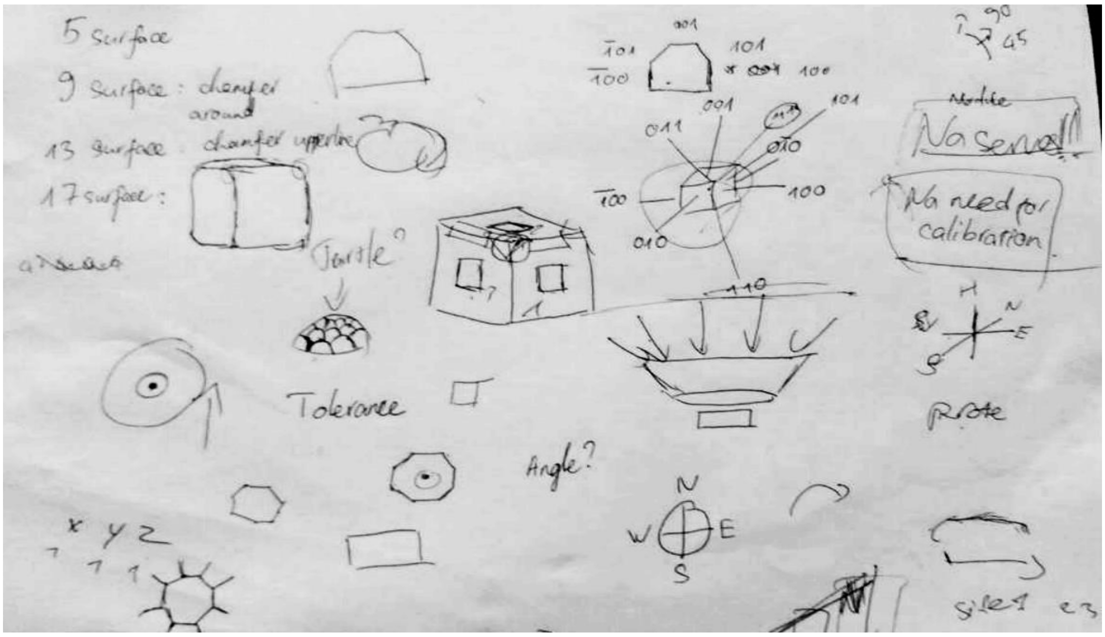


Figure 1. Sketches from the first week

The model was printed and presented to the assistant, which provided necessary feedback.

We also have created a model that has a transparent cover in order to protect the sensor, which has a frame that can slip the cover case on. However, this idea was scrapped as the transparent cover filters out some light intensity and light wave, which would provide inaccurate data and the process to recover data was unreliable and unnecessary.

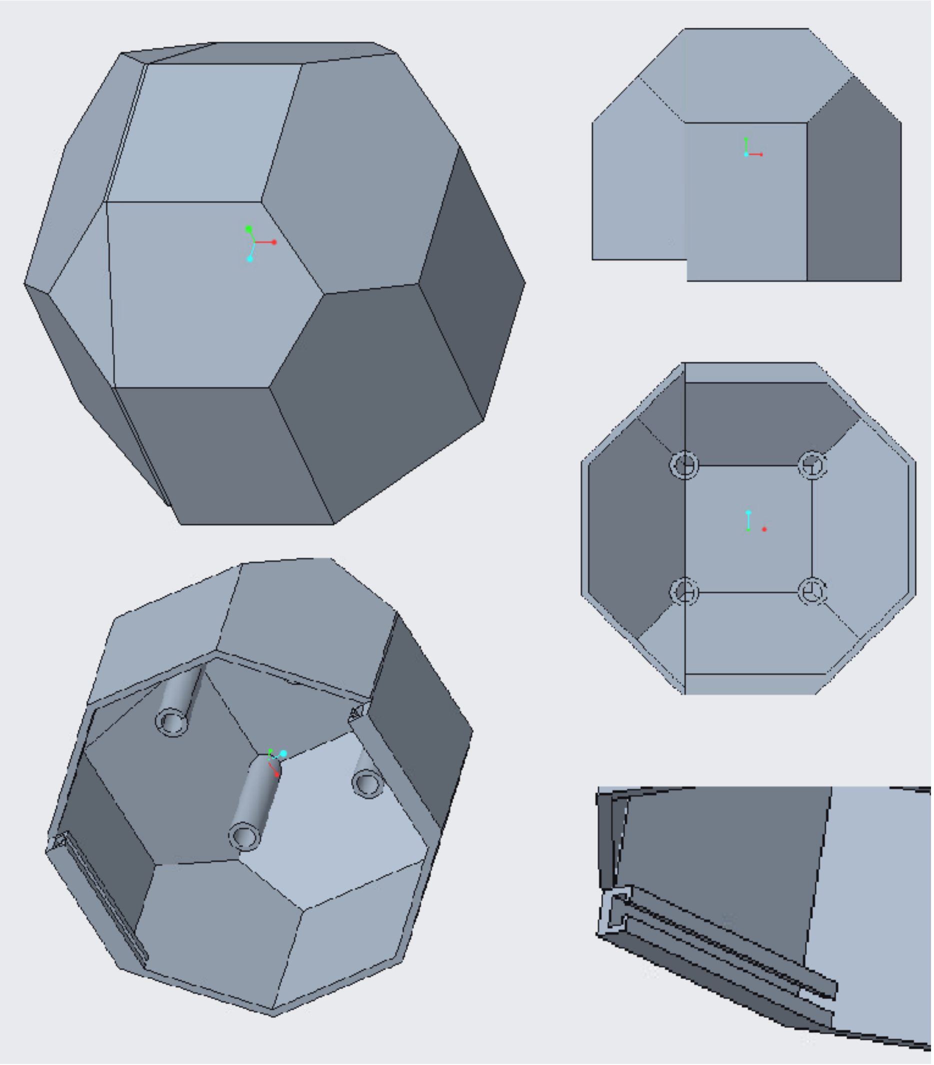


Figure 2. Original design of the case

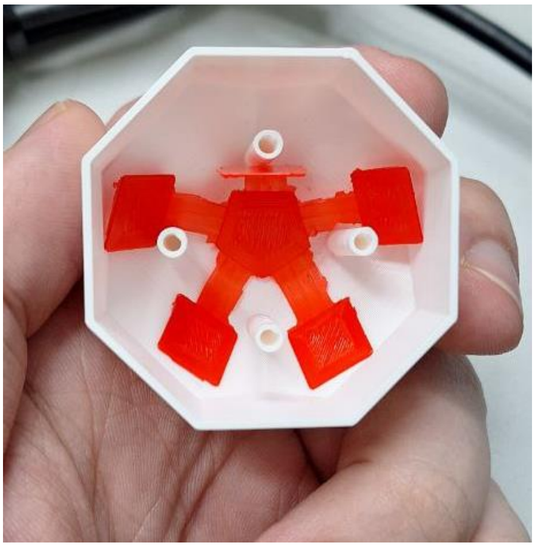


Figure 3. The mock flexible circuit board inside the prototype model

The flexible printed circuit board (PCB) was also agreed to be used during this time, and we managed to create a 3D model (Figure 3) of it to explore how it would fit inside the model.

3.1.2. Second iteration & prototyping

The final product was agreed to have 4-5 sensors, which would provide the necessary information without overloading the main microcontroller and have a reasonable cost. As the model can provide such information, it was changed slightly to be more aesthetic pleasing and cover other use cases.

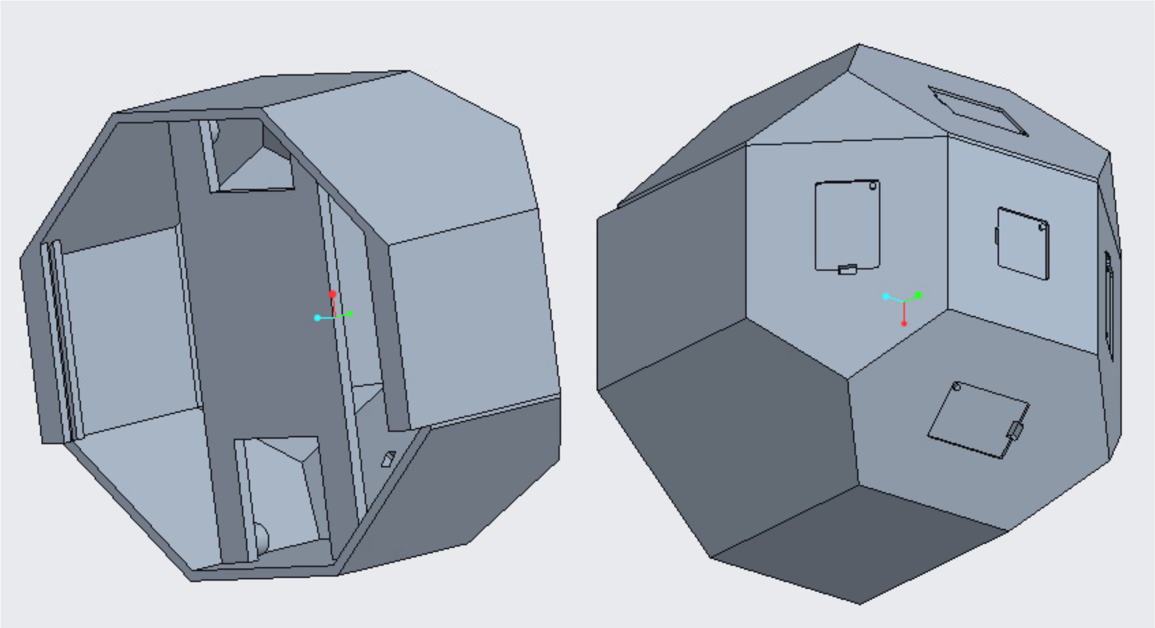


Figure 4. Second design, the model has a placeholder for the battery chamber, which was then considered to be printed with the cover case.

It was necessary for the code to be tested during this time in order to verify the assumption and to check the PCB. Using the previous model, we created a mounting pad for the sensors we have at the lab, which was successfully calculated.

3.1.3. Final design

The final design was made to be compatible with the final version of the flexible PCB.The final version contains 3 parts: the cover case, the battery chamber and the closing panel of the battery chamber.

The cover case (Figure 5) contained slots that were specifically designed to mount the PCB. It is also made that the holes that have the sensor poke out have chamfer, so that the sensors have a good area to capture the data for the light.

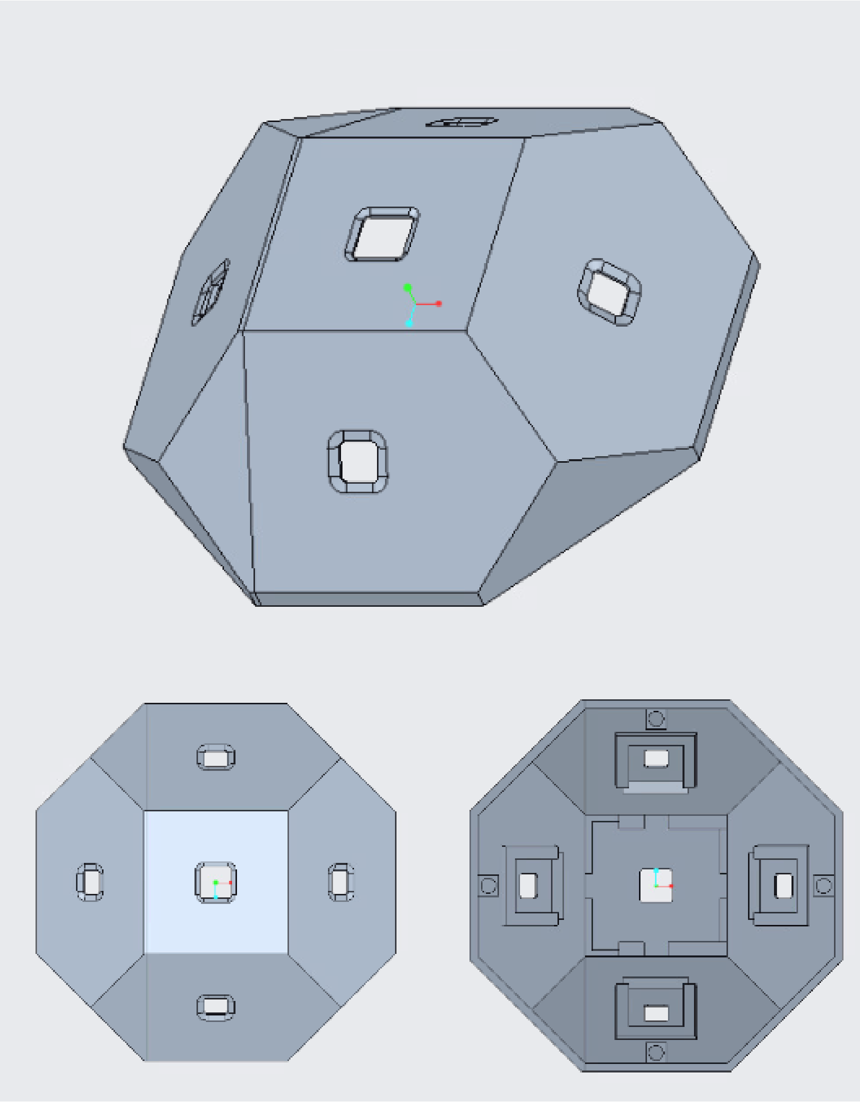


Figure 5. The cover case, tilted, top and bottom view. The flexible PCB slot can be seen from the bottom view, allow the component to be able to slide in

The battery chamber (Figure 6) was designed to quickly change the battery without having to interact with other electronic components. The battery chamber has a hole to connect to the flexible PCB, and also holes that the battery holder can be locked into. **The closing panel** (Figure 7) is connected to the chamber via a turning mechanism, which is easy to use.

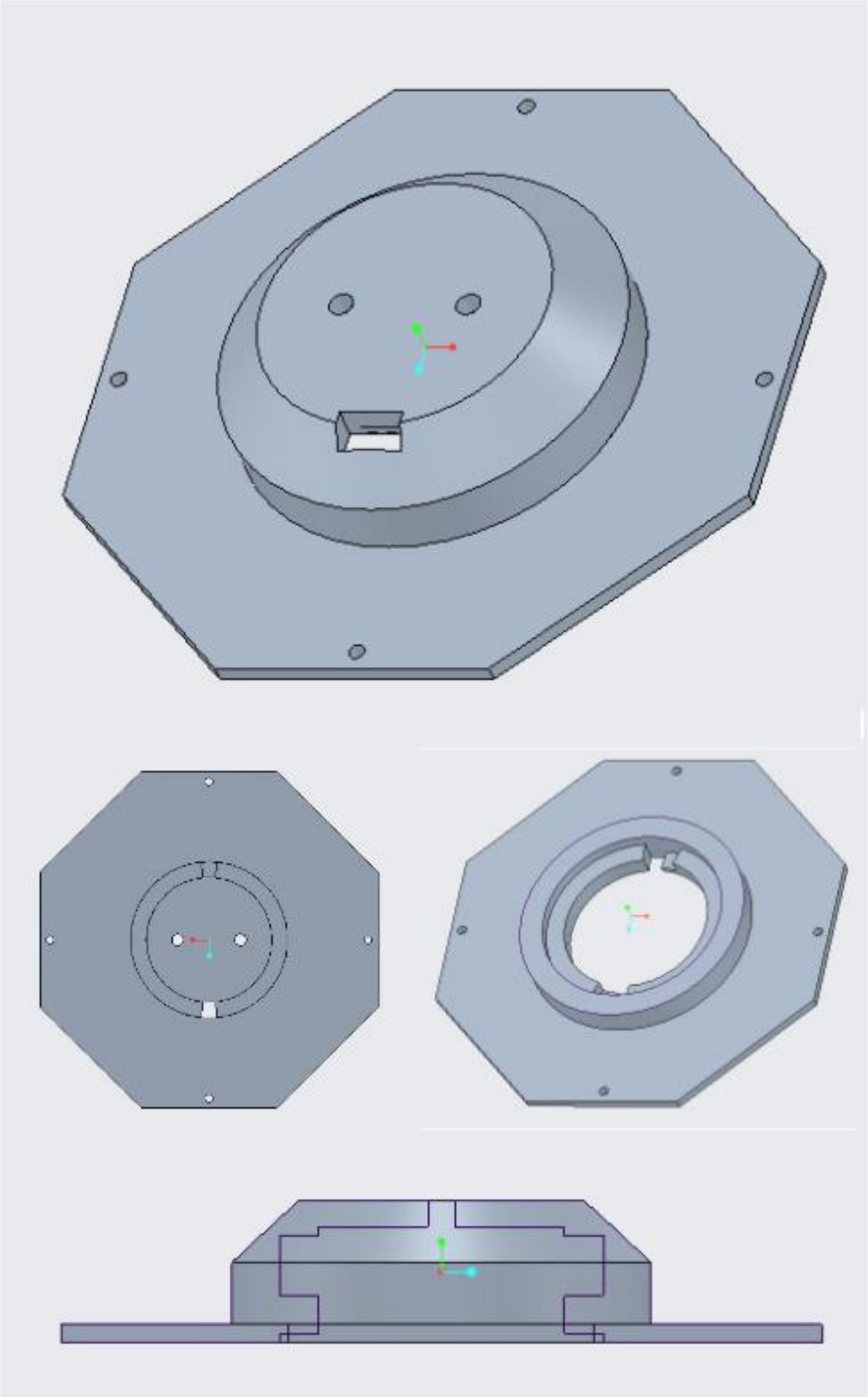


Figure 6. The battery chamber, top and tilted view; tilted and side cross section view

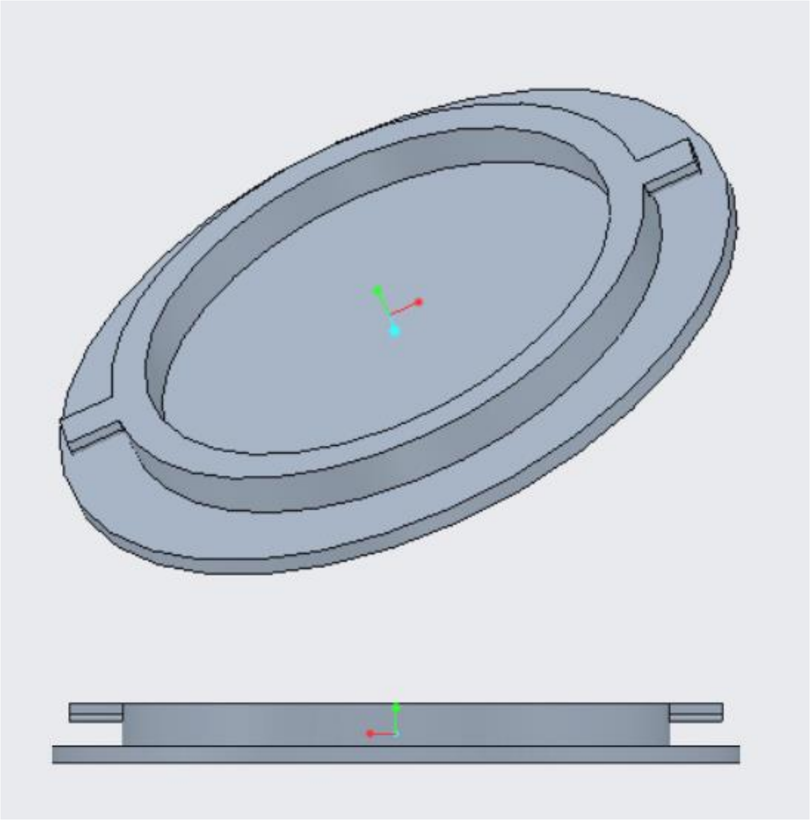


Figure 7. The closing plate, tilted and side view

The final model (Figure 8) was printed using an SLS machine inside Vilho Paja, then was cleaned to remove the particles and polished. It is then sprayed with a primer to create a good surface for usage/painting. The flexible PCB managed to fit in nicely according to the planned schematic.

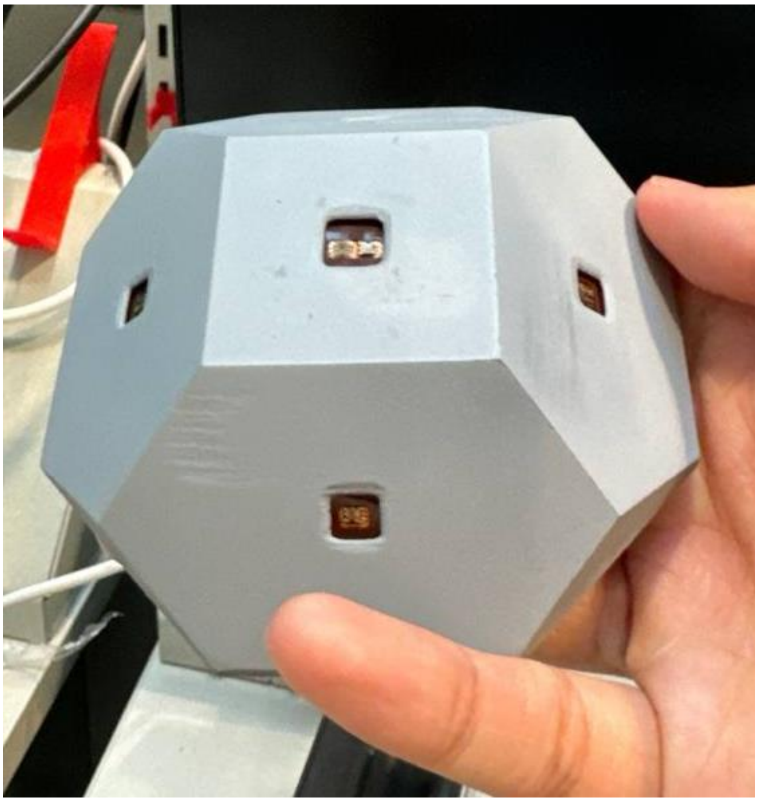


Figure 8. Final product

3.2. Circuit Board

3.2.1. First prototype

Prior to embarking on the circuit board design, it was essential to thoroughly assess the specific requirements and characteristics the board needed to fulfill. This involved estimating the necessary capabilities and identifying any specialized functionalities integral to its performance. Given that the casing design was not strictly defined, the board design had to be adjusted at certain points to accommodate these changes. Our team conducted extensive research on board design tools, learning how they work and how to create a circuit board from scratch. We ultimately chose KiCad software due to its free availability and its use in more complex projects within some companies.

First of all, depending on the objectives set, the device must capture light from five different directions and the light sensors must be faced in the appropriate way. We had multiple options to accomplish this:

- Make 5 separate PCBs and connect them with basic/flat wires
- Make rigid-flex PCB
- Make totally flexible board (FCB)

Additionally, there is an extra mandatory objective:

- Having all used components available at the manufacturers (JLCPCB) stock

Every option was thought out. We carefully evaluated the advantages and disadvantages, and subsequently took the initial step toward developing the first potential prototype by selecting option number one, with the separate boards and wires that can be soldered to connect boards together. This decision was based on the idea that it would provide greater flexibility in prototyping and facilitate easier identification of errors.

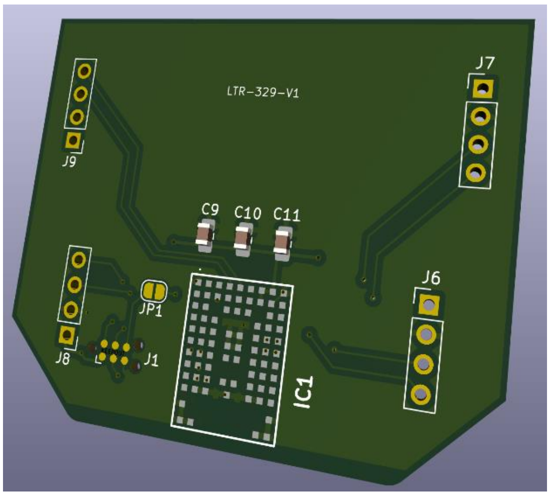


Figure 9. The control-master board

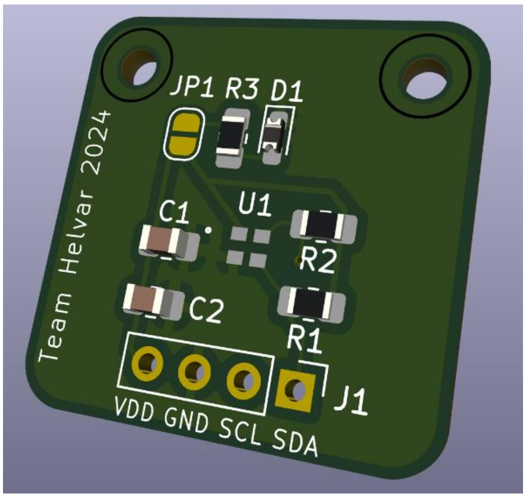


Figure 10. The light sensor board, of which four units are used in total.

Figure 9 and Figure 10 illustrate the components used in the prototyping design.

The controller board features a NORA-B126 module with an integrated nRF5340 microcontroller. The board measures 41.25 mm by 50 mm and manages four sensor boards, each measuring 21.5 mm by 21.5 mm, via the I2C interface. The module collects sensor data, performs computations, and transmits the results through the built-in Bluetooth Low Energy (BLE) functionality.

The light sensor setup includes four sensors instead of the originally planned five due to the nRF5340 microcontroller's limitation of supporting only four I2C buses simultaneously. All sensors have identical fixed addresses that cannot be configured, necessitating the placement of each sensor on a separate I2C bus.

The I2C interface was selected for this application due to its efficient communication capabilities and low pin requirement. I2C enables synchronous communication between all sensors and the microcontroller, allowing coordinated data transfer and ensuring reliable operation across multiple devices. Its two-wire configuration (SDA and SCL) minimizes the number of required connections, reducing circuit complexity and making I2C an optimal choice for managing multiple sensors within embedded systems, particularly when simplicity and reliable communication are essential.

Component overview:

- **NORA-B126**

  The NORA-B126 is a compact, low-power wireless module optimized for IoT applications like remote light sensors. It supports Bluetooth Low Energy (BLE) for reliable, long-range communication and features robust security measures to protect data. Its small size and efficient power consumption make it ideal for battery-operated, space-constrained devices.

The nRF5340 microcontroller at the heart of the NORA-B126 is a dual-core SoC featuring two ARM Cortex-M33 processors. This architecture allows one core to handle complex tasks while the other focuses on energy-efficient BLE communication. With 512 KB RAM, 1 MB Flash, and low power usage, the nRF5340 provides the perfect balance of performance and efficiency for demanding IoT applications. This made us stop on this microcontroller, because it suited our initial objectives.

- **LTR-329**

  The LTR329 is a highly sensitive, low-power ambient light sensor designed for precise measurement of ambient light levels in various environments. It features dual photodiodes that capture both visible and infrared light, providing accurate light sensing with 16-bit resolution. The sensor outputs data through an I2C interface, making it easy to integrate into various systems. With a wide dynamic range capable of measuring up to 64,000 lux, the LTR329 is ideal for applications like automatic display brightness adjustment in smartphones, tablets, and other portable devices. Its compact package size of 2.0 x 2.0 x 0.65 mm allows it to fit into space-constrained designs, while its ultra-low power consumption ensures it can be used effectively in battery-operated devices without significantly impacting battery life.

- **TC2030-IDC-NL**

  The TC2030-IDC-NL is a compact, spring-loaded programming and debugging connector designed for use with embedded systems, such as the nRF5340 microcontroller. This "No Legs" connector eliminates the need for a traditional PCB header by providing a direct, reliable contact through its small footprint of approximately 0.03 square inches, making it ideal for space-constrained designs.

  Key benefits of the TC2030-IDC-NL include its ease of use, robust mechanical connection, and cost efficiency by reducing the Bill of Materials (BOM) complexity. It supports both SWD (Serial Wire Debug) and JTAG interfaces, ensuring compatibility with standard ARM Cortex-M debugging tools. Additionally, its design minimizes signal integrity issues and is capable of withstanding multiple connection cycles, making it suitable for both development and production environments. Because this connector suited projects needs that much, we used it to the very end.

Following the development of this version, we began work on a subsequent iteration of the sensor. In this new version, we opted for flexible wiring, which ultimately led to the adoption of a flexible circuit board (FCB). We also considered a second option: using rigid-flex circuit boards with flat wires interconnecting them. This idea was appealing, and we envisioned a design similar to that in Figure 2, with mounting holes and a smaller, more compact controller board (Figure 1), including a coin battery holder attached to its backside for easy access. However, this design required the inclusion of wide connectors on each side of the controller board, as well as the procurement of suitable flat wires. Additionally, all necessary components needed to be available in the manufacturer's (JLCPCB) inventory. This eventually led to the decision to use a flexible circuit board, as option three, outlined at the beginning of this paragraph.

3.2.2. Further designing

Next, we considered the use of a FCB for our project. This decision required extensive research to understand the principles and capabilities of FCBs, especially since none of us had prior experience designing even a basic PCB. Our research provided valuable insights into the unique properties of FCBs, including their ability to achieve extremely thin profiles – often thinner than a human hair in the absence of any stiffener.

We also found that the inherent flexibility of the circuit board itself helped to resolve wiring issues, as the FCB could be easily bent and routed in tight spaces. This flexibility proved particularly useful in addressing a design oversight involving the placement of the battery socket, which was partially missing from the original design. Toward the end of the process, we decided to create a separate connector to make the battery socket external to the board, allowing for easier user access. Final design of the board you can see below (Figure 11). The figure include all important components, except NORA-B126 module.

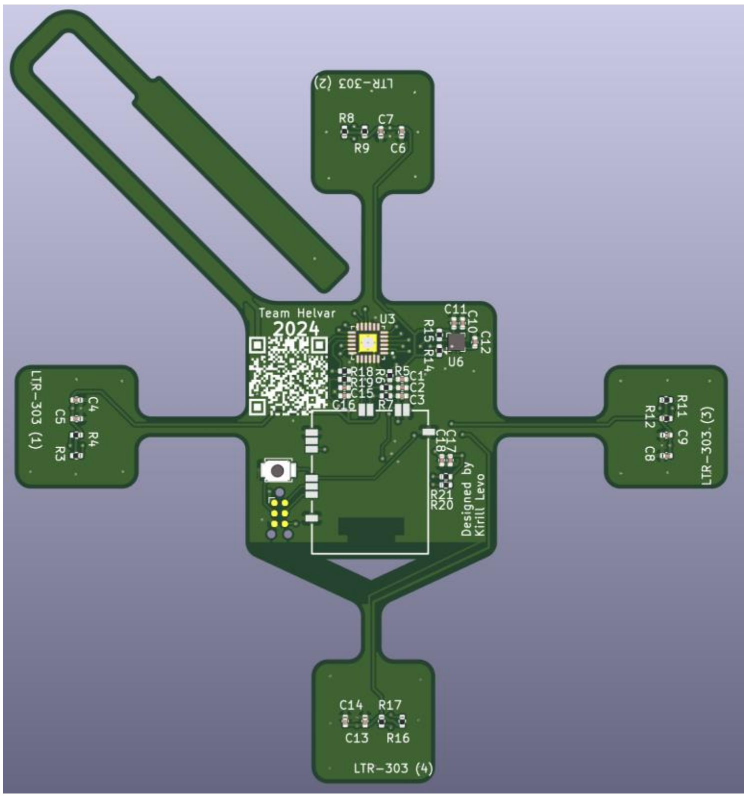


Figure 11. The front side of the FCB

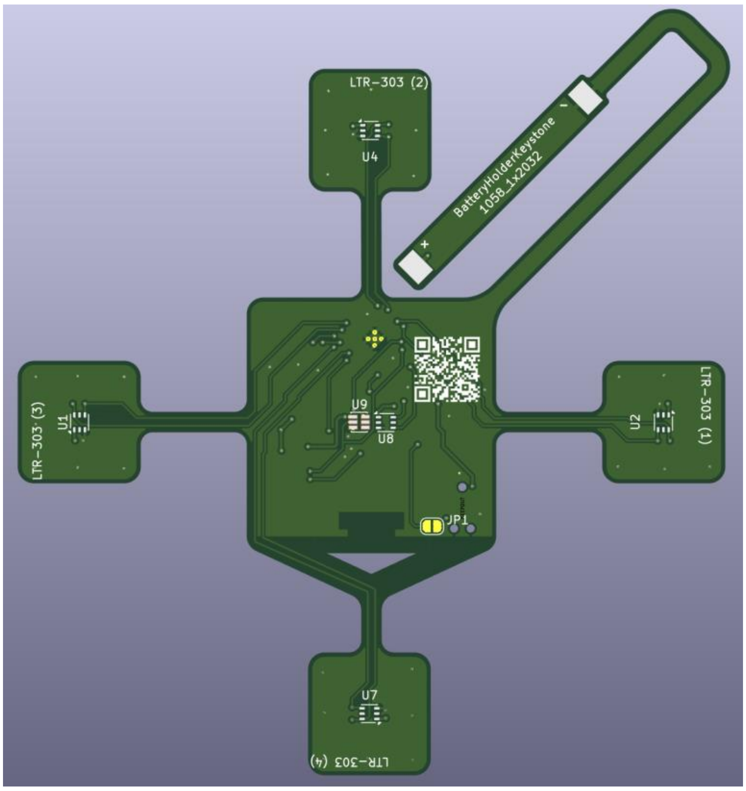


Figure 12. The back side of the FCB

The FCB design shares some similarities with the initial prototype board; however, there are significant changes in both components and structural layout. A notable addition is the battery socket trail, which allows the battery socket to be soldered directly onto the board and aligned with the battery chamber, providing easy access to the battery. This also means that the FCB has more space for passive components.

Components are now assembled on both sides of the board. Light sensors are positioned exclusively on the back side, while all other components are placed on the front. This arrangement ensures that the passive components do not scratch the casing and prevents any gaps between the circuit board and the casing, ensuring that light from the environment is accurately emitted on the sensor's surface, making calculations more accurate.

Component overview:

- **LTR303 (5)**

  The LTR303 is a high-performance ambient light sensor designed to provide accurate light measurements in a variety of lighting conditions. This sensor features a dual-channel configuration, measuring both visible and infrared light, allowing for precise ambient light detection. The LTR303 is equipped with an I2C interface for seamless integration into electronic systems, making it suitable for use in mobile devices, displays, and lighting control applications. Its low power consumption and high sensitivity contribute to efficient operation in space-constrained designs.

  The LTR329 ambient light sensor was identified as offering a more suitable implementation for this project due to its enhanced sensitivity and advanced features. However, due to limited availability in the manufacturer's inventory, the LTR303 was selected as an alternative. While the LTR303 provides reliable performance and meets the project's requirements, the LTR329's additional capabilities could have potentially offered improved accuracy and flexibility in the sensor's application. Despite this, the LTR303 remains a viable and effective solution for the project's needs.

- **TCS34725FN**

  The TCS34725FN is a compact RGB color sensor with an integrated IR filter, designed for accurate color detection and ambient light sensing. It features a high-sensitivity photodiode array with 4 channels (red, green, blue, and clear) and provides digital output via I2C, making it easy to integrate into various applications. The sensor excels in delivering precise color measurements with low power consumption, ideal for mobile devices and consumer electronics. Its programmable gain and integration time offer flexibility, while the integrated IR filter ensures reliable performance in varying lighting conditions.

  The predecessor of this sensor in our project was the AS7341, an 11-channel sensor. However, it operated only 6 channels simultaneously in our implementation. We encountered difficulties in obtaining accurate information from the AS7341, and the development of code based on its data proved inadequate. After conducting further research, we identified alternatives and chose the TCS34725FN. Although it offers fewer channels, it has considerably eased the development process for our team.

- **TCA9548ARGER**

  The TCA9548ARGER is an I2C multiplexer that enables the connection of multiple I2C devices to a single I2C bus, effectively expanding the bus capacity and addressing limitations in systems with numerous peripherals. This device features 8 selectable downstream channels, allowing independent communication with each connected I2C device, thus preventing address conflicts and simplifying bus management.

  The ARGER package was specifically chosen for this project to facilitate easier trace management during PCB design, ensuring optimal routing and minimizing signal interference. This compact package contributes to a more streamlined and efficient circuit layout, essential for maintaining signal integrity in complex designs.

- **LIS3MDL**

  The LIS3MDL is a high-performance 3-axis magnetometer designed for precise magnetic field measurements, commonly used in applications requiring direction sensing, such as electronic compasses. In this project, the LIS3MDL's primary task was to determine the absolute angle for the light sensor, functioning similarly to a traditional compass by providing a reference point, specifically identifying the location of true north. This reference was intended to support the calculation of light direction through azimuth and polar angle measurements.

  However, the LIS3MDL's accuracy in this application proved to be insufficient, with only about 15% accuracy due to its susceptibility to interference from nearby objects. This limitation rendered it ineffective for reliable direction sensing, compromising its intended purpose. Consequently, it was determined that the LIS3MDL could be removed from the design, as its contribution to the overall system was negligible and did not justify its continued use.

In this component list, the NORA-B126 module is not included, not due to prior mention, but because we were required to design an adapter board for it and subsequently solder all components together. The NORA-B126 module features more than 80 pins, but in this design, we utilized only 12 of them, with 4 serving as ground pins.

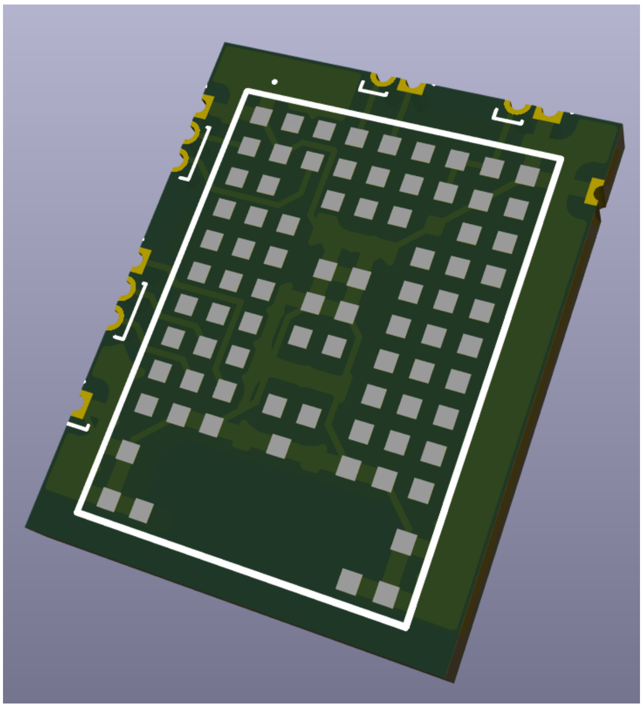


Figure 13. The NORA-B126 adapter board with dimensions of 16,75mm x 13,25mm

Upon further consideration of the board's fragile nature due to its thin profile, we decided to reinforce specific sections containing components by adding a polyimide stiffener. This reinforcement increases the thickness in targeted areas, reducing flexure and ensuring a flatter surface. KiCad, however, does not provide a built-in solution for defining such customizations or distinguishing this design from a standard PCB. Fortunately, the JLCPCB manufacturer offers an option to include a stiffener layer on flexible circuit boards (FCBs). This was achieved by creating a custom layer in the Gerber files and clearly labeling it with its purpose, material specifications, and thickness.

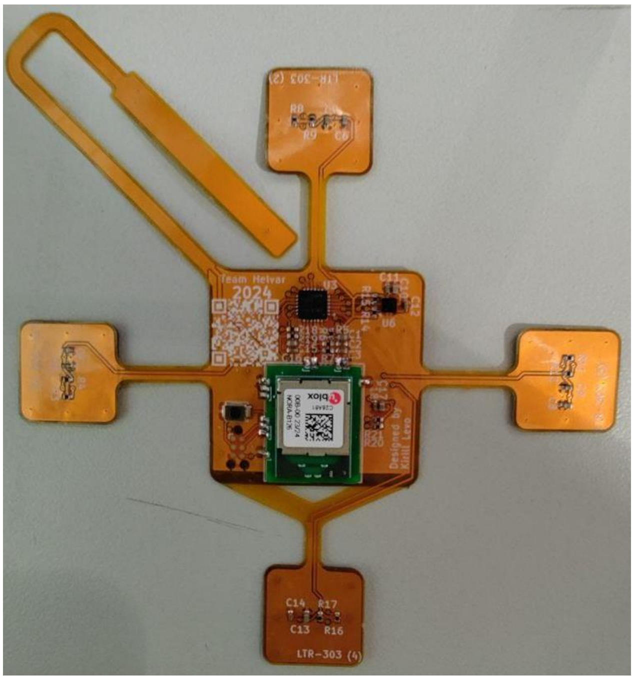


Figure 14. The top side of the final FCB

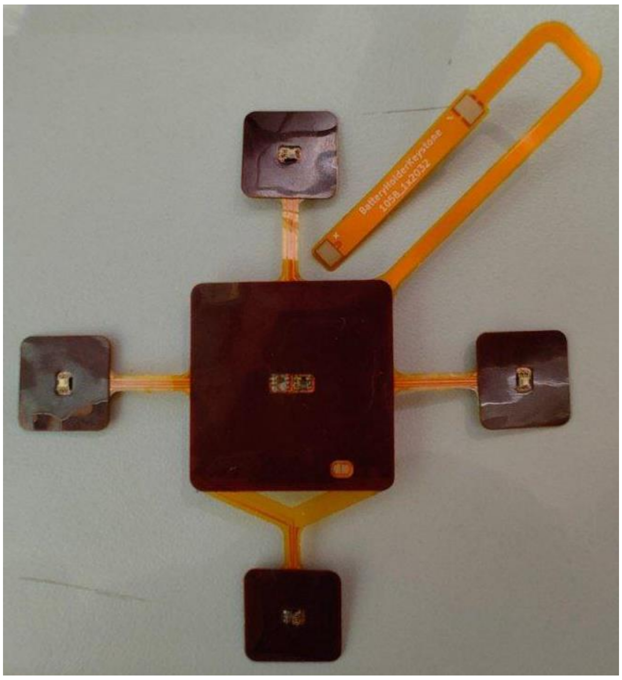


Figure 15. The bottom side of the final FCB

The final FCB incorporates a reset button, in addition to the previously outlined specifications, to address potential malfunctions. A small gap is present beneath the NORA-B126 module, which is essential for the proper functioning of the integrated RF antenna. This gap is specifically designed to prevent interference between the ground plane and the Bluetooth advertisement signal. Consequently, the ground plane copper layer is intentionally omitted in this area around the gap.

3.3. Code

3.3.1. Software Requirements

The software requirements for the "Seeing the Light" project are defined by the need to manage sensor data acquisition, processing, and Bluetooth Low Energy (BLE) communication on an embedded platform. The key requirements include the following:

Development Environment

- IDE: The code was developed using Visual Studio Code (VSCode) with the nRF Connect extension, which provides the necessary tools for working with Nordic Semiconductor's nRF series of microcontrollers.
- Toolchain: The Zephyr Real-Time Operating System (RTOS) was used as the development platform, providing drivers, kernel services, and Bluetooth stack.
- Version Control: Git was used for version control, ensuring proper code management, collaboration, and traceability.

Core Functionalities

- Bluetooth Low Energy (BLE) Communication:
    - The software must enable BLE for broadcasting sensor data using advertisements.
    - Configuration of advertisement packets, including manufacturer-specific data fields.
    - BLE initialization, management of advertisement data updates, and start/stop control.

- Sensor Data Acquisition:
    - Interface with multiple I2C sensors (LTR329 light sensors and TCS34725 RGB sensor).
    - Initialize and configure sensors for data acquisition, including setting gain, integration times, and operational modes.
    - Periodic reading of sensor data for continuous monitoring.

- Data Processing and Analysis:
    - Calculation of lux values from raw sensor data using predefined algorithms.
    - Calculation of color temperature (CCT) and chromaticity based on RGB sensor data.
    - Direction detection using lux values from multiple light sensors, with calculations for azimuth and polar angles.
    - Determination of light source type (natural or artificial) based on processed sensor data.

- Security and Data Integrity:
    - Optional data encryption using AES-128 encryption with CCM mode for BLE advertisement data.
    - Generation of encryption keys, nonces, and authentication tags for secure transmission.

Hardware Interface

- I2C Communication:
    - The software must interface with I2C devices (light sensors) to perform read and write operations.
    - The configuration and selection of the correct I2C channels using a multiplexer are essential to managing multiple sensors.
- GPIO Control and Power Management:
    - Control GPIOs for sensor power management if necessary.
    - Implement low-power modes in the Zephyr RTOS to extend battery life.

Timing and Scheduling

- RTOS Integration:
    - The software is designed to run under Zephyr RTOS, making use of its kernel services for task scheduling, sleep management, and timing.
    - Periodic tasks, such as sensor readings and BLE advertisement updates, are managed using the kernel's timing services.

Debugging and Testing

- Logging: Integration of logging facilities using Zephyr's printk() function to output debug information, including sensor readings, calculated values, and error messages.
- Testing:
    - The device should be tested under different lighting conditions to validate the sensor readings and processing algorithms.
    - Simulation of sensor data and edge cases to ensure robust operation under various scenarios.
- BLE central device:
    - We have built an extra device to catch the advertising packages sent by the sensor. This device will automatically parse, decrypt (if encryption is enabled), and display the data via a serial monitor. The Github link for the device is listed in this document’s appendix.

These software requirements ensure that the project meets its goals of accurate light measurement, data processing, and communication, while also providing a reliable and maintainable codebase.

3.3.2. Code Architecture

The code for the "Seeing the Light" project is organized into several key modules, each responsible for specific functionalities such as sensor management, data processing, Bluetooth Low Energy (BLE) communication, and encryption. The architecture ensures that the system is modular, maintainable, and scalable.

Main Application Module

- Responsibilities:
    - Acts as the entry point of the application, initializing the system and starting Bluetooth services.
    - Manages the main loop for continuous operation, including sensor data collection and BLE communication.
- Functions:

    - main() Function:
        - Initializes the Bluetooth subsystem using bt_enable().
        - Starts BLE advertising with bt_le_adv_start().
        - Configures and initializes I2C devices and sensors.
        - Enters a loop where sensor data is periodically collected, processed, and transmitted via BLE.
    - Main Loop:
        - Continuously reads sensor data from multiple I2C-connected sensors.
        - Processes the raw sensor data (e.g., converting to lux, calculating angles).
        - Updates BLE advertisement data with the latest sensor readings.
        - Manages the timing of sensor readings and BLE updates using k_msleep().

Sensor Management Module

- Responsibilities:
    - Manages the initialization, configuration, and data acquisition of all sensors.
    - Ensures correct communication with sensors via I2C channels.
- Functions:
    - configure_ltr329_sensor() Function:
        - Configures the LTR329 light sensor for data acquisition, setting gain and integration time.
    - select_mux_channel() Function:
        - Selects the appropriate I2C channel via a multiplexer to communicate with specific sensors.
    - read_ltr329_sensor_data() :
        - Reads lux data from the LTR329 sensors through I2C.
    - RGB Sensor Configuration:
        - Sends configuration commands to the TCS34725 RGB sensor over I2C to enable the sensor, set the integration time, and configure the gain.

Data Processing Module

- Responsibilities:
    - Transforms raw sensor data into meaningful metrics like lux, CCT, and light direction.
    - Implements algorithms to analyze and process the data.
- Functions:
    - calculate_lux_ltr329() :
        - Converts raw sensor readings from the LTR329 sensor into lux values using specific algorithms.
    - cct_calculation() :
        - Calculates the correlated color temperature (CCT) based on RGB sensor data by mapping RGB values to XYZ color space and applying McCamy's formula (STMicroelectronics, 2021).
    - calculate_angles() :
        - Computes the azimuth and polar angles using lux values from multiple light sensors to determine the light source's direction.
        - The code computes two key angles — **azimuth** and **polar** — based on differences in light intensity values, represented as vectors in a **3D space**. It first calculates the difference between two vectors, derived

from the light intensity (lux) readings. The **azimuth angle** is then calculated as the angle in the XY-plane between this difference vector and the positive X-axis, ensuring it falls within the 0 to 360-degree range. The **polar angle** measures the tilt of the vector from the horizontal XY-plane, calculated using the Z-component of the vector and the projection of its XY components. Both angles are determined using the **$atan^2$ function**, which ensures accurate angle calculations, and the results are converted from radians to degrees. Additionally, the average light intensity is computed by taking the mean of **all provided lux values**, giving an overall measure of light intensity.

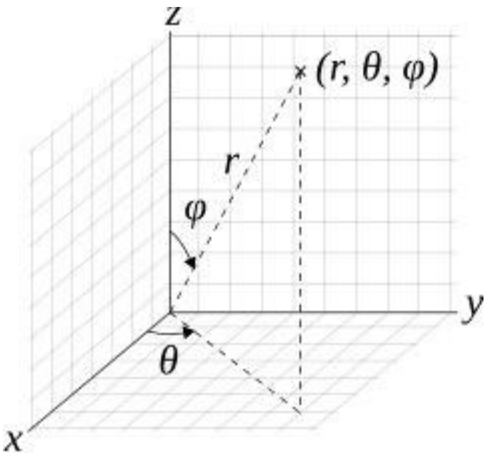


Figure 16. A spherical coordinate system. This is a coordinate system for three-dimensional space where the position of a given point in space is specified by three real numbers: the radial distance r along the radial line connecting the point to the fixed point

- calculate_average_lux() :
    - Calculates the average lux across multiple sensors to assess the overall illumination level.
- Determine the light source:

    - The process at first needs to normalize the **RED**, **GREEN**, and **BLUE** color values to bring them into a common scale. This normalization ensures that the values are in a **comparable range** (0 to 256), which allows for meaningful comparisons between the different colors. After normalization, **the absolute differences** between each pair of color values (RED-GREEN, GREEN-BLUE, and BLUE-RED) are calculated. These differences provide a measure of how much each color component varies from the others, indicating the **relative balance of colors**.

    - To determine if the light source is natural, specific conditions are checked based on the calculated color differences. The light is considered natural if the **difference** between green and blue is small, suggesting these colors are **closely balanced**, and the difference between blue and red is larger, indicating a clear distinction from blue to red. Additionally, the difference between the red-green and blue-red values should be **relatively small**, implying a balanced distribution of all three colors. These checks help identify whether the color proportions match those typically associated with **natural daylight.**

BLE Communication Module

- Responsibilities:
    - Manages BLE initialization, advertisement, and secure data transmission.
    - Ensures that sensor data is transmitted correctly and securely via BLE.
- Functions:
    - bt_enable() :
        - Initializes the Bluetooth subsystem.
    - bt_le_adv_start() :
        - Starts BLE advertising using the advertisement and scan response data arrays.
    - update_advertisement_data() :
        - Updates the BLE advertisement packet with the latest sensor data, including azimuth, polar angle, average lux, and whether the light source is natural or artificial.
    - generate_encrypted_data() :
        - Encrypts the sensor data using AES-128 encryption in CCM mode before it is transmitted via BLE.

Debugging and Logging Module

- Responsibilities:
    - Provides real-time feedback during development and testing.
    - Logs critical information to assist in debugging and validating system performance.
- Functions:
    - printk() :
        - Used throughout the code to output debug information, such as sensor readings, calculated values, and error messages.

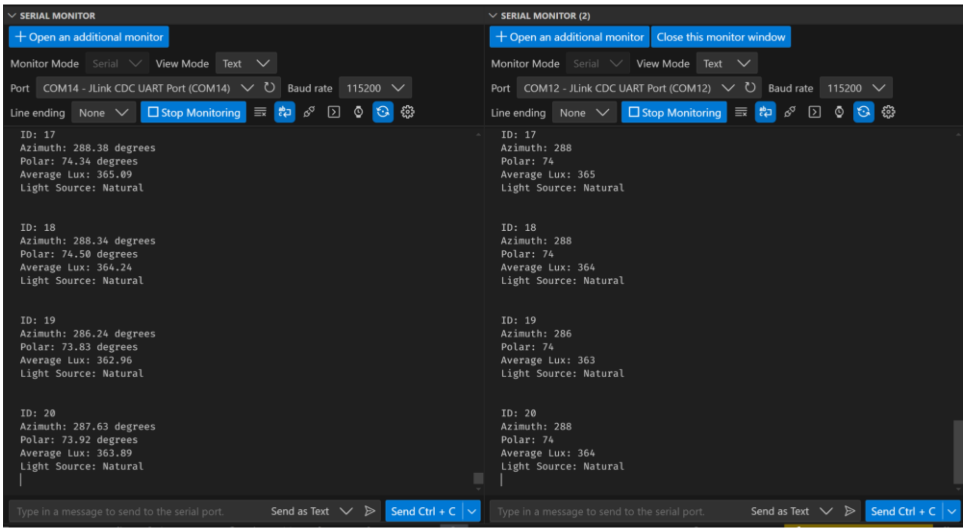
Code Flow Overview

- System Initialization:
    - Initializes the Bluetooth subsystem and configures the I2C sensors with necessary parameters (gain, integration time).
- Data Acquisition:
    - Iterates in the main loop, selecting the appropriate I2C channels, and reads data from connected light sensors.
    - Collects and processes RGB sensor data.
- Data Processing:
    - Processes sensor data to calculate lux levels, CCT, and the direction of the light source.
    - Determines whether the light source is natural or artificial based on processed data.
- BLE Communication:
    - Formats and transmits sensor data via BLE advertisements.
    - Applies data encryption if enabled for secure communication.
- Logging and Debugging:
    - Logs sensor readings, computed values, and any errors to assist in debugging and performance monitoring.

3.3.3. *Integration with Hardware*

The integration of software with hardware was a crucial aspect of the "Seeing the Light" project, aiming to ensure that the developed code could effectively interact with the physical components, such as sensors and the microcontroller. However, the project faced significant challenges in this area, particularly with the transition from the initial prototype using the nRF52 microcontroller to the final product intended to use a flexible PCB with the nRF53 microcontroller.

Successful Integration with nRF52 Prototype



Monitor Mode: Serial
View Mode: Text
Port: COM14 - JLink CDC UART Port (COM14)
Baud rate: 115200
Line ending: None

```
ID: 17
Azimuth: 288.38 degrees
Polar: 74.34 degrees
Average Lux: 365.09
Light Source: Natural


ID: 18
Azimuth: 288.34 degrees
Polar: 74.50 degrees
Average Lux: 364.24
Light Source: Natural


ID: 19
Azimuth: 286.24 degrees
Polar: 73.83 degrees
Average Lux: 362.96
Light Source: Natural


ID: 20
Azimuth: 287.63 degrees
Polar: 73.92 degrees
Average Lux: 363.89
Light Source: Natural
```

Monitor Mode: Serial
View Mode: Text
Port: COM12 - JLink CDC UART Port (COM12)
Baud rate: 115200
Line ending: None

```
ID: 17
Azimuth: 288
Polar: 74
Average Lux: 365
Light Source: Natural


ID: 18
Azimuth: 288
Polar: 74
Average Lux: 364
Light Source: Natural


ID: 19
Azimuth: 286
Polar: 74
Average Lux: 363
Light Source: Natural


ID: 20
Azimuth: 288
Polar: 74
Average Lux: 364
Light Source: Natural
```

Figure 17. The two serial monitors for our two devices. The serial monitor on the left is displaying the data collected and transmitted by the main sensor. While the one on the right is displaying the data received by the observer.

- I2C Communication Setup:
  - For the initial prototype using the nRF52 microcontroller, we successfully integrated the software with the hardware, particularly in managing the I2C communication with multiple sensors, including the LTR329 light sensors and the TCS34725 RGB sensor.
  - The software was designed to initialize the I2C bus and configure the sensors at startup, using functions like configure_ltr329_sensor() and configure_rgb_sensor() to set the appropriate parameters such as gain and integration time.

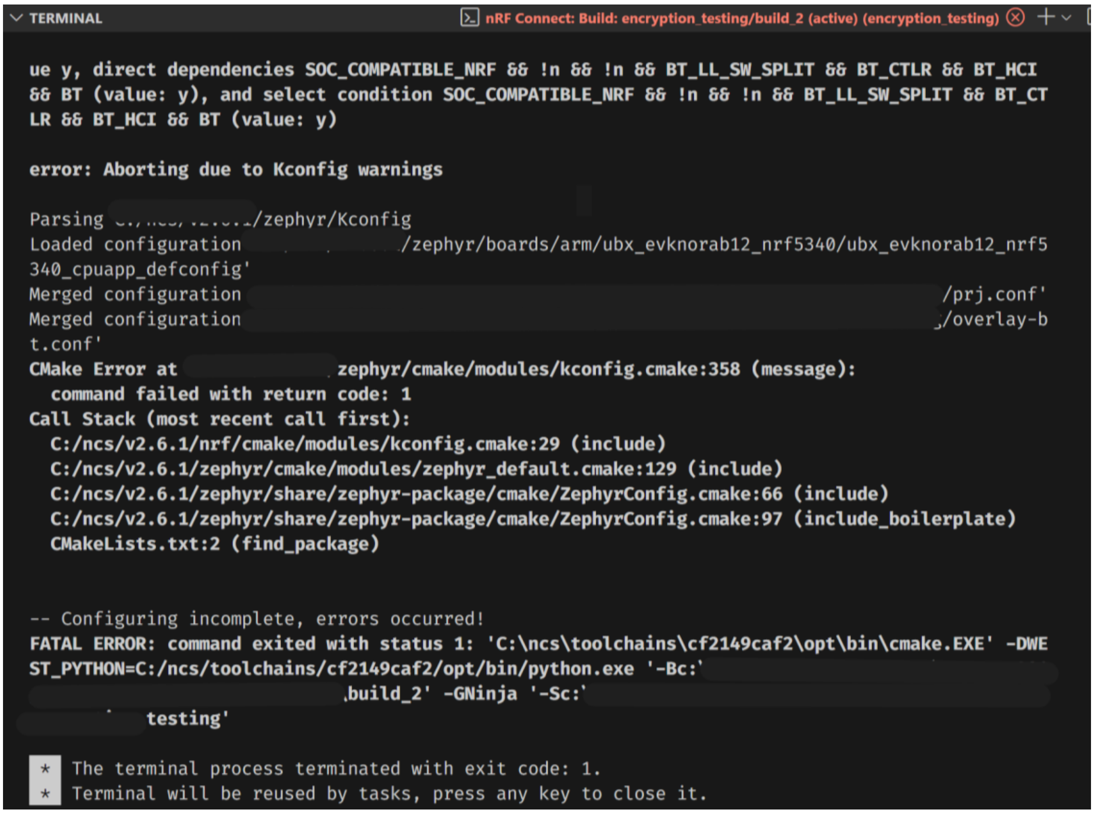
- Multiplexer Control:
  - The use of an I2C multiplexer was effectively managed through the select_mux_channel() function, allowing the software to communicate with multiple sensors on the same I2C bus by selecting the correct channel.
  - This setup worked well with the nRF52-based prototype, ensuring reliable data acquisition from all connected sensors.

- Data Acquisition and Processing:
  - Data from the sensors was successfully read and processed in real-time. The functions read_ltr329_sensor_data() and read_rgb_sensor_data() were used to retrieve raw data, which was then processed by the software to calculate lux levels, CCT, and light direction.

- This integration allowed the prototype to perform as expected, accurately reflecting the environmental conditions through the processed data.

- BLE Communication:
    - The processed data was transmitted via Bluetooth Low Energy (BLE) using the nRF52 microcontroller. The update_advertisement_data() function handled the formatting and transmission of sensor data, which was securely encrypted using AES-128 before being broadcasted.

Challenges with nRF53 Integration



```
ue y, direct dependencies SOC_COMPATIBLE_NRF && !n && !n && BT_LL_SW_SPLIT && BT_CTLR && BT_HCI
&& BT (value: y), and select condition SOC_COMPATIBLE_NRF && !n && !n && BT_LL_SW_SPLIT && BT_CT
LR && BT_HCI && BT (value: y)

error: Aborting due to Kconfig warnings

Parsing [redacted]/zephyr/Kconfig
Loaded configuration [redacted]/zephyr/boards/arm/ubx_evknorab12_nrf5340/ubx_evknorab12_nrf5
340_cpuapp_defconfig'
Merged configuration [redacted]/prj.conf'
Merged configuration [redacted]/overlay-b
t.conf'
CMake Error at [redacted]zephyr/cmake/modules/kconfig.cmake:358 (message):
  command failed with return code: 1
Call Stack (most recent call first):
  C:/ncs/v2.6.1/nrf/cmake/modules/kconfig.cmake:29 (include)
  C:/ncs/v2.6.1/zephyr/cmake/modules/zephyr_default.cmake:129 (include)
  C:/ncs/v2.6.1/zephyr/share/zephyr-package/cmake/ZephyrConfig.cmake:66 (include)
  C:/ncs/v2.6.1/zephyr/share/zephyr-package/cmake/ZephyrConfig.cmake:97 (include_boilerplate)
  CMakeLists.txt:2 (find_package)


-- Configuring incomplete, errors occurred!
FATAL ERROR: command exited with status 1: 'C:\ncs\toolchains\cf2149caf2\opt\bin\cmake.EXE' -DWE
ST_PYTHON=C:/ncs/toolchains/cf2149caf2/opt/bin/python.exe '-Bc:'[redacted]
[redacted]build_2' -GNinja '-Sc:'[redacted]
[redacted]testing'

 *  The terminal process terminated with exit code: 1.
 *  Terminal will be reused by tasks, press any key to close it.
```

Figure 18. The terminal returns error as we are trying to build the program for nRF53

- Transition to Flexible PCB with nRF53:
    - The final product was intended to use a flexible PCB featuring the nRF53 microcontroller. However, we encountered significant difficulties in flashing the program onto the nRF53 device. The nRF53 microcontroller introduces a new dual-core architecture, which necessitates a substantial restructuring of the code, especially in managing the interaction between the cores and adapting to the new development environment.

- Issues with Dual-Core Architecture:
    - The nRF53's architecture, which includes both an application core and a network core, posed challenges in porting the existing code designed for the single-core nRF52. The communication between these cores and the distribution of tasks across them required a different approach, which was not fully supported by our existing code structure. As a result, we were unable to successfully integrate the software with the nRF53 hardware. The inability to

flash the program onto the nRF53 device meant that we could not test or validate the final product as intended.

Future Considerations:

The challenges encountered with the nRF53 highlight the need for a revised development approach when dealing with more complex microcontroller architectures. Moving forward, the code will need to be restructured to take full advantage of the nRF53’s dual-core design.This will involve splitting tasks between the application and network cores, managing inter-core communication, and possibly rewriting significant portions of the software to be compatible with the new architecture.

Summary

The integration of software with hardware in the "Seeing the Light" project was partially successful. While the initial integration with the nRF52-based prototype was achieved, allowing for successful data acquisition, processing, and BLE communication, the transition to the final product using the nRF53 microcontroller presented substantial challenges. The dual-core architecture of the nRF53 requires a different approach to software design, which we were unable to implement within the project’s timeframe. As a result, the final product could not be fully realized, underscoring the complexity of integrating advanced microcontroller architectures into existing software frameworks.

4. *Reflection of the Project*

4.1. Reaching objective

The primary objective of the "Seeing the Light" project was to develop a sensor capable of measuring illuminance, determining light direction, and differentiating between natural and artificial light sources. The project successfully met these objectives, as the final prototype demonstrated the ability to perform all required functions accurately and reliably. The sensor was able to transmit data via Bluetooth Low Energy (BLE) advertisements, as specified in the project plan. The completion of these objectives was the result of thorough planning, dedicated teamwork, and effective problem-solving throughout the project's phases.

In addition to the technical goals, the project provided significant learning opportunities for all team members. Each member was able to focus on their personal goals, such as PCB design, 3D modeling, and coding, and applied these skills to contribute to the overall success of the project. Although there were challenges, such as integrating hardware (the customized PCB with nrf53) and software components, these were addressed through collaboration and persistence, leading to a successful project outcome.

4.2. Timetable

The project adhered closely to the timetable outlined in the project plan, with most milestones reached within the scheduled time frame. However, there was a notable delay in the shipment of the flexible PCB, which significantly compressed the final stages of the project. As a result, the team had only one day to test the PCB, solder components, and build and flash the code. Despite this time constraint, the team managed to complete these critical tasks successfully, although it required intense effort and coordination. This delay highlighted the importance of contingency planning and the ability to adapt quickly to unforeseen challenges.

4.3. Risk Analysis

In the "Seeing the Light" project plan, we identified several potential risks at the outset, along with strategies for mitigating these risks. As the project progressed, we were able to overcome or avoid many of these risks, thanks to careful planning, teamwork, and effective problem-solving. However, a few significant risks did materialize, impacting the project, particularly in the final stages. Below is an analysis of the risks encountered and their impact on the project.

1. Delays in Acquiring Necessary Components

Severity: High

Outcome: This risk had a significant impact on the project. The FCB crucial to the final product's development was delayed and arrived just one day before the final presentation due to issues with the shipping company. This delay severely limited our ability to integrate and test the final product, leading to cascading effects on other aspects of the project.

Impact: The late arrival of the FCB directly contributed to the difficulties we faced in integrating the software with the nRF53 microcontroller. With only a single day available, there was insufficient time to adapt the code to the new dual-core architecture of the nRF53, resulting in an incomplete final product. This also meant that we could not fully assess the battery life in the final configuration, leaving this aspect partially untested.

2. Integration Issues Between Hardware and Software

Severity: Medium

Outcome: This risk materialized during the final integration phase, particularly with the transition from the nRF52 to the nRF53 microcontroller. The nRF53's dual-core architecture introduced complexities that our existing code structure could not accommodate without significant rework. Unfortunately, due to the delayed arrival of the FCB, we were unable to make the necessary adjustments in time.

Impact: The inability to successfully integrate the software with the nRF53 hardware meant that we could not achieve the intended functionality of the final product. While the nRF52-based prototype worked well, the final product could not be fully tested or demonstrated, impacting the overall success of the project.

3. Insufficient Battery Life

Severity: Medium

Outcome: Although battery life was identified as a potential risk, it did not become a critical issue during the project. The nRF52-based prototype achieved a battery life of approximately 14 hours, which is sufficient for Helvar’s usage scenarios. However, without being able to test the final product with the nRF53 microcontroller, the true impact on battery life remains unknown.

Impact: While the current battery life is adequate, there is potential for improvement, especially in the context of the nRF53’s architecture. Future work will need to focus on optimizing power consumption to extend battery life further.

14.14mA average

31.47mA max

10:00m time

8.49C charge

Figure 19. Power Kit II's measurements for our device, the battery test was run for 10 minutes, with the sensor operates at an interval of 30 seconds

4.Inaccurate Measurements Due to Sensor Issues

Severity: High

Outcome: This risk was effectively mitigated through thorough testing and calibration of sensors. The LTR329 and TCS34725 sensors performed reliably, and we did not encounter significant accuracy issues during the prototype phase. On the designed FCB, we use a different version of light sensor, LTR303, but these sensors have no significant difference except for a slight change in battery consumption.

Impact: No significant impact; the measurements obtained were accurate and met the project's requirements.

5. Lack of Expertise in New Technologies

Severity: Medium

Outcome: The team dedicated time to learning and practicing with the new technologies involved in the project, including the nRF series microcontrollers and BLE communication. This proactive approach mitigated the risk, allowing the team to handle most technical challenges effectively.

Impact: The team's enhanced skills contributed to the successful development of the nRF52-based prototype. However, the nrf53's new core architecture posed too much of a challenge in such limited time.

6. Data Transmission Failures

Severity: Medium

Outcome: We did not experience significant data transmission failures. The Bluetooth functions were reliable, and extensive testing ensured that the data was transmitted effectively within the required range. There were a few obstacles with the encryption of data, but all of them were fixed by the end.

Impact: No significant impact; the BLE communication was stable and met the project requirements.

7. Limited Availability of Team Members

Severity: Medium

Outcome: This risk was managed through careful planning and scheduling. Regular progress check-ins ensured that tasks were completed on time, despite the varying availability of team members.

Impact: No significant impact; the project timeline was adhered to, with most tasks completed as scheduled.

8. Unexpected Technical Challenges

Severity: High

Outcome: Apart from the integration issues with the nRF53, the team was able to navigate most technical challenges successfully. Buffer time allocated in the project schedule allowed for troubleshooting and problem-solving, minimizing disruptions.

Impact: The major technical challenge was the integration with the nRF53, which could not be resolved within the project's timeframe.

9. Misalignment with Sponsor's Expectations

Severity: High

Outcome: Regular communication with Helvar ensured that the project remained aligned with their expectations. Any potential concerns were addressed promptly, and feedback was incorporated throughout the project.

Impact: No significant impact; the project remained in line with Helvar's goals and requirements.

10. Incomplete Documentation

Severity: Low

Outcome: Documentation was prioritized throughout the project, with specific team members assigned to this task. Regular reviews ensured that the documentation was comprehensive and up-to-date.

Impact: No significant impact; the project documentation was completed successfully.

Summary

The "Seeing the Light" project encountered several risks, with the most significant being delays in acquiring the flexible PCB and the challenges related to integrating the software with the nRF53 microcontroller. While the initial prototype using the nRF52 was successful, the transition to the nRF53 was hampered by the delayed component delivery, limiting our ability to adapt the code to the new architecture. Despite these challenges, the project made significant progress, with many risks successfully mitigated, resulting in a functional prototype that met most of the project's objectives. Future efforts will need to focus on resolving the integration issues with the nRF53 and further optimizing the system for battery life and performance.

5. Discussion and Conclusions

As we bring the "Seeing the Light" project to a close, we find ourselves reflecting on a journey that has been both challenging and rewarding. There is a sense of relief that the project is complete, mixed with pride in what we have achieved, and a small measure of regret that the final product did not entirely match the vision we set out with. Over the past few months, this project has tested us in many ways, pushing us to grow not only as engineers but also as a team.

One of the most significant lessons we learned was the importance of effective collaboration. Each of us brought our own unique skills and perspectives to the table, and it was through working together that we were able to take an abstract idea and turn it into a functioning prototype. However, the road was not always smooth. We often found that our differing ideas and approaches led to communication challenges. Whether it was trying to align our technical decisions within the team, or ensuring that our progress and concerns were clearly communicated to our teaching assistants and the Helvar representatives, there were moments when miscommunication slowed us down. These experiences taught us the value of clear, open dialogue and the importance of listening to and understanding each other’s viewpoints. It is a lesson that will be invaluable in any future collaborative work we undertake.

From a technical standpoint, adapting to new technologies was one of the most demanding aspects of the project. Working with the nRF52 microcontroller allowed us to successfully integrate our software with the hardware, and seeing our prototype come to life was incredibly satisfying. However, as with any ambitious project, we faced our share of challenges. We had hoped to incorporate the nRF53 microcontroller into our final product, but the complexity of its dual-core architecture required more time and resources than we had available. While this was a setback, it underscored the importance of being flexible and open to learning new things, even when they do not go as planned.

The delays in receiving the flexible PCB were another major challenge we encountered. Despite our best efforts to plan and anticipate potential issues, the PCB’s arrival just one day before our final presentation left us with limited time to test and integrate it into our final product. This delay had a significant impact, particularly on our ability to fully explore and optimize the device’s battery life. While our prototype’s battery life of around 14 hours is sufficient for Helvar’s needs, we recognize that there is room for improvement. This experience taught us valuable lessons about managing time under pressure and the importance of being prepared for unexpected delays.

Despite these challenges, we are incredibly proud of what we have accomplished. The project was not just about coding or working with sensors and microcontrollers; it was also about learning to function as a cohesive team, improving our communication skills, and adapting to the curveballs that inevitably arise in any complex project. We have grown not only in our technical abilities but also in our ability to collaborate effectively and support one another through difficult times.

On a more personal note, this project has been an opportunity to form lasting relationships and build a sense of camaraderie that we will carry with us long after our time at Aalto university. We shared late nights and long hours troubleshooting together, and those experiences have brought us closer as a team. Working on a real-world problem, tasked by a company like Helvar, provided us with insights and experiences that go beyond what we could have learned in the classroom.

In conclusion, while the "Seeing the Light" project may not have ended exactly as we envisioned, it has been a success in many important ways. We have gained invaluable experience, developed our skills, and created something tangible that we can all be proud of. There are certainly things we could have done differently, and we will carry those lessons with us into future projects. However, we also recognize that the journey itself—full of learning, growth, and collaboration—has been one of the most rewarding aspects of our time working together. We leave this project with not only a deeper understanding of our field but also with great memories and a strong sense of accomplishment.

6. List of Appendixes

Github link:
https://github.com/ArtemKIA/Helvar-Seeing-the-Light

TCS34725 RGB sensor :
https://www.alldatasheet.com/datasheet-pdf/view/894928/AMSCO/TCS34725.html

AS7341 10-Channel RGB:
https://www.mouser.com/catalog/specsheets/AMS_03152019_AS7341_DS000504_1-00.pdf

LTR-303ALS-01 light sensor: https://www.mouser.com/datasheet/2/239/Lite-On_LTR-303ALS-01_DS_ver%201.1-1175269.pdf

LTR-329ALS-01 light sensor:
https://www.mouser.com/datasheet/2/239/Lite-On_LTR-329ALS-01%20DS_ver1.1-348647.pdf

Multiplexer TCA9548A:
https://www.ti.com/lit/ds/symlink/tca9548a.pdf

Magnetometer LIS3MDL:
https://www.st.com/resource/en/datasheet/lis3mdl.pdf

NRF5340 DEVKIT:
https://www.nordicsemi.com/-/media/Software-and-other-downloads/Product-Briefs/nRF5340-DK-PB-10.pdf

NORA-B126-00B module: https://content.u-blox.com/sites/default/files/NORA-B1_DataSheet_UBX-20027119.pdf

KiCad PCB design from scratch most useful guide:
https://youtube.com/playlist?list=PLn6004q9oeqGl91KifK6xHGuqvXGb374G&si=WmeNgo9hDkIueGG_

Bluetooth low energy Advertising:
https://devzone.nordicsemi.com/guides/short-range-guides/b/bluetooth-low-energy/posts/ble-advertising-a-beginners-tutorial

Nordic Semiconductor technical documentation:
https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk5.v15.0.0%2Ftwi_sensor_example.html

Zephyr documentation:
https://docs.zephyrproject.org/latest/hardware/peripherals/i2c.html

Zephyr programing guides: https://blog.golioth.io/how-to-use-zephyr-shell-for-interactive-prototyping-with-i2c-sensors/

Texas Precision labs light sensors: https://www.ti.com/video/series/precision-labs/ti-precision-labs-ambient-light-sensors.html

Basics of light sensors:
https://www.vishay.com/docs/84154/appnotesensors.pdf

Tinycrypt cryptography library:
https://github.com/intel/tinycrypt

BLE central device for this sensor:
https://github.com/hdvflss/Helvar_Central_BLE

References

STMicroelectronics. (2021, July). *CCT computation using VD6283 - Application note.* Retrieved from STMicroelectronics: https://www.st.com/resource/en/application_note/an5638-how-correlated-color-temperature-is-calculated-by-vd6283-stmicroelectronics.pdf