

Design and Implementation of Bluetooth Mesh Outdoor Lighting

Måns Hofvander



LUND
UNIVERSITY

Department of Automatic Control

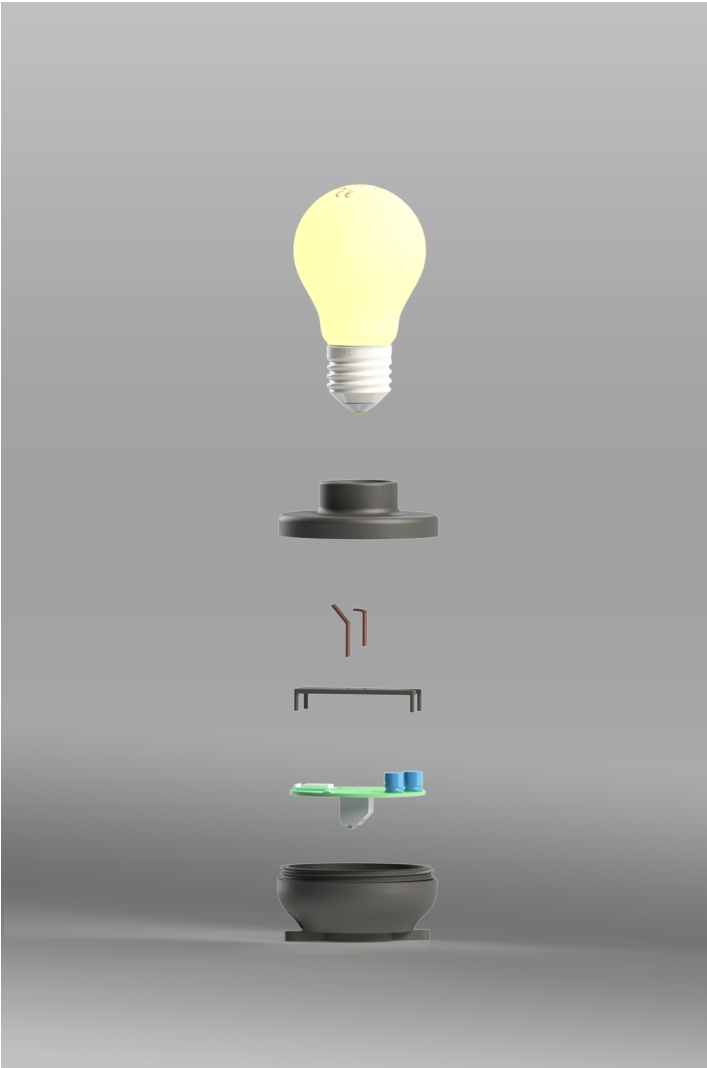
MSc Thesis
TFRT-6185
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2022 by Måns Hofvander. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2022

Abstract

In the days of IoT and smart homes Bluetooth is in constant development and with it the products that use Bluetooth. Bluetooth mesh is the latest feature and with it comes many opportunities of new products. This report describes the development of an outdoor light system which can be controlled from a website. The system is named Luminaire and consists of a custom built circuit board, an encapsulating casing and the software needed to run the Luminaire. Luminaire turns a light bulb on and off on demand but also after the rising and setting of the sun. The mesh functionality makes it possible for the smartphone or tablet to only reach a single unit and this unit will pass on the signal to the entire network. This makes it possible to control lights in a very large area. At the time of writing there is no such product on the market and Luminaire aims to fill that position. This projects aims not only to build this product but also acquire the knowledge needed for other mesh applications in the future. Under the management of Adevo Consulting the project has taken seven months and has been successful in every aspect so far. As a proof of concept the latest prototype is a great success and something to build on in the future for a product that aims to go on the market. There are issues with the high voltage circuit on the which means that at the time of writing the circuit board cannot provide the 230 VAC that the light bulb requires without external components. The complete system can be seen in figure 0.1.



Figur 0.1

Sammanfattning

I tiden då IoT och smarta hem är eftertraktat är Bluetooth i ständig utveckling, och med det även produkterna som använder Bluetooth. Bluetooth mesh är den senaste egenskapen och det skapar nya möjligheter för nya Bluetoothprodukter. Den här rapporten beskriver utvecklingen av ett system för utomhusbelysning som styrs från en hemsida. Det här systemet har namnet Luminaire och består av ett egenbyggt kretskort, ett inkapslande skal och mjukvaran som krävs för att styra systemet. Luminaire kan slå av och på en glödlampa på begäran men också styras efter solen upp och nedgång. Mesh-funktionaliteten gör att den styrande smarta enheten bara behöver nå en Luminaire enhet för att denna ska kunna skicka vidare meddelandet till resten av nätverket. Det gör det möjligt att styra belysningen över stora områden. I skrivande stund finns det ingen liknande produkt på marknaden och Luminaire avser att fylla den positionen. Det här projektet har inte bara som mål att bygga den här produkten utan också införskaffa vetskapen som krävs för att bygga andra mesh-produkter i framtiden. Under ledning av Adevo consulting har detta projekt pågått i sju månader och har varit framgångsrikt på varje plan hittills. Som ett bevis för att produkten är realistisk att konstruera har det här projektet varit en stor framgång. I skrivande stund har produkten fungerande lågspänningselektronik, ett meshnätverk och en fungerande hemsida.

Acknowledgements

I would like to start by thanking my industrial supervisors and managers Mattias Wallinius and Maja Arvehammar who are the creators of this project. They have been excellent managers and it is inspiring to work under such great people and engineers. Secondly I would like to thank my academic supervisor Karl-Erik Årzén and my examiner Martina Maggio for all their time and commitment. Lastly I would like to thank Patrik Larsson who, even though he cannot put his name on this paper, has been there every step of the way. Thank you all.

Innehåll

| | |
|---|-----------|
| 1. Introduction | 11 |
| 1.1 Background | 11 |
| 1.2 Purpose | 11 |
| 1.3 Project structure | 12 |
| 1.4 Limitations and Restrictions | 12 |
| 1.5 Disposition | 12 |
| 2. Theory | 14 |
| 2.1 Bluetooth | 14 |
| 2.2 PCB | 15 |
| 2.3 Solidworks | 17 |
| 2.4 Zephyr | 17 |
| 2.5 3D printing | 18 |
| 2.6 Power conversion | 18 |
| 2.7 Electromagnetic compatibility | 19 |
| 3. Methods | 20 |
| 3.1 Method | 20 |
| 4. System specifications/design | 22 |
| 4.1 System Specifications | 22 |
| 5. Mechanical Design | 25 |
| 5.1 Product development | 25 |
| 5.2 Contact metal | 30 |
| 6. Electronic Design | 34 |
| 6.1 Schematics | 34 |
| 6.2 Electronic design | 36 |
| 7. Software Design | 39 |
| 7.1 Embedded programming | 39 |
| 7.2 Software design requirements | 41 |

| | |
|---------------------------------------|-----------|
| 8. Results | 50 |
| 8.1 Electronic design | 50 |
| 8.2 Hardware design | 52 |
| 8.3 Software results | 56 |
| 8.4 Testing | 56 |
| 8.5 General project results | 57 |
| 9. Discussion | 58 |
| 9.1 Development process | 58 |
| 9.2 Final concept | 59 |
| 10. Conclusion and future work | 61 |
| 10.1 Future work | 61 |
| 10.2 Conclusion | 61 |
| Litteratur | 63 |
| A. | 65 |

1

Introduction

1.1 Background

Bluetooth is a technology in constant development which in turn leads to an evolution of the products using Bluetooth. One of the newer functions of Bluetooth is Bluetooth Low Energy (BLE), as the name suggests it offers a lower energy consumption by sending signals only when new information is available, in contrast to streaming communication. This is a good implementation for applications that monitors a system or rates such as a heart monitor or a thermometer. It is not as applicable for streaming of music which requires the steady stream of signals between sender and receptor. Together with BLE comes the opportunity to use another function, the so called mesh network of Bluetooth products. This means that many Bluetooth products can communicate with each other without having a connection to a smartphone or a tablet. With the mesh functionality only a single connection is required to send commands to the entire network of units and similarly it is possible to receive information from all units despite only being in range of one. At this moment in time mesh is mostly used to locate, position and measure distances with high accuracy in 3D space.

The idea behind this project is to control outdoor lighting for properties using this mesh technology. All lamps are controlled by one smart device which has access to the whole network through a website. The user will easily be able to connect and disconnect units from the network and make changes to the lighting settings, such as switch on time and energy saving.

1.2 Purpose

The purpose of this thesis is to develop the Bluetooth unit which controls the toggling of outdoor illumination. During an summer job earlier I have started writing the code to for this application and now there are new features to be added and many things to change and rewrite in a better way. During this project the goal is to design and construct a circuit board tailored to the dimensions and shape needed, a casing,

a connection between the e27 lamp and the circuit board and to write the code for both website and circuit board. At the time of writing there are no outdoor Bluetooth mesh lights on the market and therefore this product aims to fill that position.

1.3 Project structure

The project can be categorized into four subsections: electronic design, mechanical design, software adaptation and testing. Each of these development phases will follow the methodology of Ulrich and Eppinger [Eppinger, 2016] with several iterations of design for each subsections before reaching the final product. This follows the listed phases.

- Phase 0 - Product planning
The product development process commences and the mission statement is declared.
- Phase 1 - Concept development
In this phase the requirements of the product is discussed
- Phase 2 - System-level design
- Phase 3 - Detailed design
- Phase 4 - Testing and refinement
- Phase 5 - Production ramp-up

1.4 Limitations and Restrictions

There are a few limitations and assumptions which came in the project description from Adevo. The product should fit e27 standard light bulbs. It should be controlled from a website rather than an app. The circuit board design was to be round.

1.5 Disposition

In this master's thesis there is a total of ten chapters and here is a short summary of each one. Chapter 1 is an introduction to the the thesis and tells the background of the project. It also presents the purpose and the project structure Chapter 2 describes the theoretical knowledge needed to understand the report, the finished product and the development. Chapter 3 discusses the methods used in the product development. Chapter 4 lists the system specifications and the design requirements. Chapter 5 goes into detail of the concept generation of the hardware. Chapter 6 is a detailed description of the electronic design and each of its components. Chapter 7 discusses

the software in general as well as in certain details. Chapter 8 presents the results of the product at the time of presentation. Chapter 9 is the discussion of the methods chosen and it raises questions about the development process. Chapter 10 presents a conclusion and suggestions of future work.

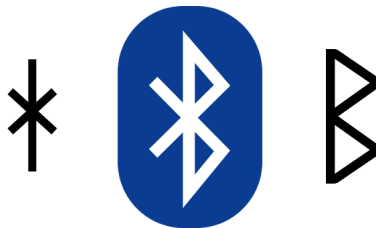
2

Theory

2.1 Bluetooth

Background

Bluetooth is a standard for short-range wireless communication using radio waves from 2.402 to 2.48 GHz. The development of the short link radio technology was started by Nils Rydbeck, CTO at Ericsson mobile in Lund in 1989. The initial purpose was to develop wireless headsets but when IBM approached Ericsson about integrating a mobile phone and a ThinkPad notebook the Bluetooth technology came in perfect hand to connect an Ericsson phone with an IBM computer. Both companies then agreed to make Bluetooth an open industry standard to allow maximum market access. Together with Intel, Nokia and Toshiba Ericsson and IBM formed the Bluetooth Special Interest Group (SIG) in 1998 to govern all standards related to Bluetooth. The SIG today has more than 35000 member companies in the areas of communication, computing and consumer electronics. The Bluetooth name and symbol comes from the Norwegian-Danish king Harald Bluetooth who managed to unite his kingdom in the 10th century, see figure 2.1



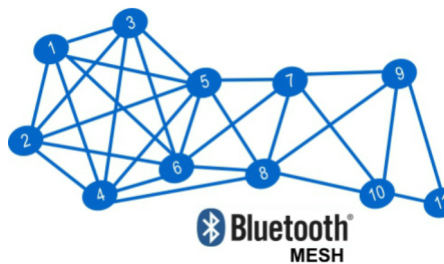
Figur 2.1 The Bluetooth symbol is composed of the nordic letters H and B for Harald Bluetooth

Low Energy

Bluetooth Low Energy (BLE) is an addition to Bluetooth classic, it was introduced in 2009 and developed by Nokia. BLE separates itself from classic, it uses the same 2.4GHz frequency but a simpler modulation system. A modern Bluetooth device has the ability to use either system or both. BLE is designed with small, low energy devices in mind and by sending and receiving shorter pieces of data (attributes) via a low energy link the device can sustain itself on a button cell battery for years.

Mesh

Bluetooth Mesh is a networking standard from 2017 which allows many-to-many communication based on BLE. The mesh functionality operates on a flood network principle which means that nodes relay the incoming message to more nodes. By this principle only one node needs to be in range of a Bluetooth device such as a smartphone or tablet in order to send a message to all nodes in the network, on the same principle a device can receive messages from all nodes despite only being in range of a single node. The maximum length of a message is 384 bytes when using Segmentation and Reassembly which means that the original message is divided into smaller segments of 11 bytes. See figure 2.2 for an illustration of the mesh network.



Figur 2.2 Graphic description of the mesh functionality

2.2 PCB

Overview

PCB stands for Printed Circuit Board. PCBs are used in almost every electronic product that exists today. The PCB works as a housing for electronic components by fitting these into holes in the boards or through conductive pads on the surface. It also removes the need for cables as components are connected through a conductive trace either on the surface of the board or through an internal layer.

Mechanical design

The internal structure of a PCB is alternating between conductive and insulating layers. The conductive layer is often copper and the insulating layer a substrate of a composite material such as epoxy. Boards can come in different thicknesses and layers, most commonly in 1,2 or 4 layers of copper with isolating layers in between. The advantage of a multi-layer board is a more compact design and the option to route traces through the internal copper layers rather than on the surface which saves space for components. This however makes the board harder to repair and debug if something were to go wrong. A multi-layer board is also more expensive to manufacture due to the added complexity in design. In order to route traces between layers small copper plated holes, called vias or plated through-holes, are used. Vias can either go all the way through the card or in a multi-layer card only from one layer to the next and not be seen from the outside. The traces are etched to divide the copper into separate lines and are insulated from each other by the substrate and the air. The surface of the board has a layer named silkscreen which is the text and instruction on the board, it can also show where components are to be mounted and the reference label of each component. Lastly the board is laminated to encapsulate the board and protect it from corrosion. [Keim, 2022]

KiCad

There are many existing Electronic Computer-Aided Design (ECAD) tools on the market today and they all have the same purpose, to make it easy and accessible to design PCBs and PCBAs. One of the most commonly used is KiCad [Rosso, 2022]. KiCad is a free, open source software supported by the development department at CERN. KiCad features a schematic editor, PCB layout, manufacturing output, simulations, calculations and creation of 3D models. It was founded in 1992 by Jean-Pierre Charras who at the time worked at the University of Technology in Grenoble. It started as a collection of the individual programs eeschema, PCBnew and a gerber viewer. Starting at 2013 developers from CERN were looking for a new and easier software to develop complex circuit boards. The choice fell on KiCad which showed most promise of the already existing alternatives. Since then developers working as a part of the open hardware initiative have been working to develop a free open source software with the aim to make it the industry standard in the future.

KiCad consists of a schematic handler, eeschema and a PCB-layout handler, PCBnew. In the schematic handler each component is represented by a symbol. Each symbol is then assigned a footprint. A footprint is the area which the component occupies on the circuit board. It will also contain the mounting of the component, it is either surface mounted or through hole mounted. This symbol must match with the footprint of the component for the PCB-layout handler to correctly make a grid of which pins or pads to connect with traces. Figure 2.3 shows the first version of footprints and traces of the circuit board.

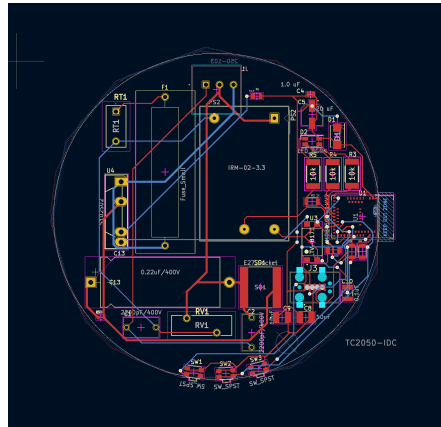


Figure 2.3 The first version of Luminaire in PCBnew view

2.3 Solidworks

As with ECAD there are many existing Computer-Aided Design (CAD) tools on the market. One tool used globally both in industry and in education is Solidworks [Solidworks, 2022]. Solidworks was founded in 1993 and is today used by millions of engineers and hundreds of thousands of companies. It is an easy and reliable tool for designing and developing mechanical designs first and foremost. There are however many additional add-ons for working with simulations, electrical circuits and more. When designing a component the user starts by drawing a 2-dimensional sketch and then choose from the tools available how to manipulate the sketch into a 3D model. When the model is satisfactory, different attributes such as material, weight and appearance can be assigned. Parts can also be evaluated for structural integrity and joined together with other parts in assembly mode. Models can then be saved in many formats for their intended use.

2.4 Zephyr

The Zephyr Project is a Linux Foundation project. It is an open source collaboration to build a small, scalable real-time operating system (RTOS) for resource-constrained devices. The aim is to reduce cost and accelerate time to market for billions of connected embedded devices. Zephyr is a good choice for small connected sensors, modems and wireless gateways. Zephyr supports more than 170 hardware devices which is more than any other RTOS. Zephyr offers many features and they are constantly growing in numbers. One of these is the devicetree support which

describes custom hardware and is used to create the application image. Another one is the Bluetooth Low Energy support which includes Bluetooth mesh.

2.5 3D printing

A 3D printed component is constructed from a 3D model designed in a CAD program. This CAD-file is inserted into software known as a slicer. A slicer is a software that calculates the movements and actions of a 3D printer in order to produce the 3D model. Many 3D printer manufacturers have their own slicers. The most common, and cheapest, way to print today is with plastic filament that is melted and then solidified in layers, starting from the bottom and working its way upwards. When using this method the geometry of the part to be printed is crucial for a successful print. There can be no large overhangs or geometry printed without support from beneath. Details may not be too small as the quality of the print depends on the accuracy of the printer and its specific settings. There are functions that can make support structure around complicated areas but they can often lead to new issues such as failing to print or not having the same surface finish when removed. The printer used in this project has been the Flashforge Inventor seen in figure 2.4.



Figur 2.4 The 3D printer Flashforge Inventor

2.6 Power conversion

In Swedish outlets there is single phase 230VAC at 50Hz. The microcontroller unit requires 3.3VDC. In order to convert the output from the outlets three steps need to be taken. Firstly the alternating current must be stepped down. This is done using a transformer. Then the alternating current is rectified using diodes. Lastly the now rectified but rippling voltage is smoothed out using capacitors. The capacitor stabilises the electron flow by charging and discharging in opposite to the voltage,

creating a smoother flow. If an even more stable voltage is required an additional voltage regulator can be added to make the voltage fully rectified. [Prof.Mad, 2022]

2.7 Electromagnetic compatibility

When working with electronics and circuit boards the electromagnetic interference needs to be taken into consideration. Interfering signals can disturb both equipment in surrounding areas as well as internal signals and sensors. Therefore there are guidelines for electromagnetic compatibility when designing a printed circuit board. One rule is that traces always take a 45 degree turn, a 90 degree turn can lead to reflection and EMI. By physically separating digital and analog signals as well as high speed traces from low speed traces it is possible to minimize the effect they have on each other. For the same reason traces are to be kept as short as possible to avoid cross talk. Many guidelines follow the same pattern when it comes to components, i.e., analog and digital components should be separated to avoid cross talk. High speed and switching components create a lot of disturbance, therefore these should be routed as short as possible, kept away from sensitive components and shielded with ground and filters. [limited, 2022]

3

Methods

3.1 Method

The development of the Luminaire follows the methodology by Ulrich and Eppinger [Eppinger, 2016] with several phases in the development process. The development is also agile with many iterations of the prototypes before arriving at a final product. This also leads to iterating through some of the phases described by Ulrich and Eppinger several times.

Agile development

The Agile development was introduced in 2001 by some of the leading figures in the software industry. One of the most important aspects of their new development strategy was the ability to divide the development into small increments of planning, design, development and testing before receiving feedback from customers and then continue with a new iteration of phases from planning to feedback. It is not only applicable to software and not all parts of the Agile Manifesto must be included in a development process for it to be considered Agile. Typical for agile processes is working with the kanban method where tasks are divided into smaller sub-tasks in a card-like form and then moved from the pile of waiting tasks to the finished pile. The purpose is to never get stuck with big tasks and to get the feeling of constant progress.[cprime, 2022]

Product design and development by Ulrich & Eppinger

According to Ulrich and Eppinger product design and development [Eppinger, 2016], the structure of the project follows the steps listed in chapter 1.4. These steps will be discussed more in detail in the following sections.

- Identifying customer needs
- Establishing sub-problems to satisfy customer needs
- Establishing concept screening criteria

- Weighting the criteria
- Developing sub-concepts for each sub-problem
- Evaluating each sub-concept based on the concept screening criteria
- Calculate the weighted score for each sub-concept
- Generate concepts from the sub-concepts
- Develop digital prototypes of the concepts and evaluating their advantages and disadvantages.

4

System specifications/design

4.1 System Specifications

The system specification of the product is a small compact device which is connected to standard 230 power and has a screw socket for a e27 light bulb in one end. The Bluetooth connection allows the user to turn the light on and off using a website from a smart device or a computer in range of a device in the network.

Identifying customer needs

In order to find corrects system specifications the customer needs are the first step. These needs are the results of discussion with the industrial supervisor as well as some basic functionality needs. Some needs are mechanically related, others are either electronically or software needs. There are also manufacturing needs such as need number 8, low manufacturing cost.

Tabell 4.1 Customer needs hardware

| No. | Need | Comment | Imp. |
|-----|--------------------------------------|------------------------------|------|
| 1 | The LUM fits in a lampholder | | 5 |
| 2 | The LUM is water- and dust resistant | | 4 |
| 3 | The LUM is easy to install | | 4 |
| 4 | The LUM is easy to assemble | | 3 |
| 5 | The LUM is safe | The Electronics are isolated | 4 |
| 6 | The LUM has a user interface | | 2 |
| 7 | The LUM does not use excessive power | | 2 |
| 8 | The LUM has low manufacturing cost | | 3 |
| 9 | The LUM outputs the right power | | 3 |
| 10 | The LUM is durable | | 2 |

Design of electronics

Design of electronics is the first phase of the development since the mechanical design must take into consideration the dimensions of the electronics. The items listed are the key elements of the product electronics. The list is the original list put together with the industrial supervisor with regards to the planned functionality of the Luminaire and based on the customer needs.

- Nordic RF 58240 processor
- Bluetooth Low Energy antenna
- Interface pads for programming
- Push button for reset and configuration
- Rectifier 240V AC / 5V DC
- Voltage regulators
- Oscillator
- Power relay
- Necessary discrete components like resistors and capacitors
- Connectors for 240 VAC,
- Connectors for light bulbs

Mechanical design

The system specification for the mechanical design is to encapsulate the electrical design in a way that it protects vital parts from outside interference as well as allows both manufacturers and users to interact with it in an easy way. This is to be done using Solidworks and then 3D print a prototype casing to the existing electronics board. Since the mechanical design is dependent on the electronics for both size, form and functionality it is difficult to specify further details and constraints in an early stage of the development.

Software

The software specification included both the embedded programming on the devices as well as the cloud service for the website.

Tabell 4.2 Customer needs software/user interface

| No. | Need | Comment | Imp. |
|-----|--------------------------------------|---------|------|
| 1 | No app that needs updates | | 5 |
| 2 | Simple to connect | | 4 |
| 3 | Available to all Bluetooth devices | | 4 |
| 4 | Can set the switch on and off time | | 3 |
| 5 | Turns on and off with sunrise/sunset | | 3 |

With the customer needs from the table the target specifications are set as following.

- Bluetooth Mesh functionality
- Discoverable in chrome websites
- Geo-location services
- Energy saving option
- Clock synchronization
- Manual time setting for switching on and off.

5

Mechanical Design

5.1 Product development

The mechanical design includes the casing of the electronics and the separate contact metal which links the circuit board to the light bulb. Here the full product development following the methodology of Ulrich & Eppinger is presented. In table 5.1 the metrics have been decided for comparison between following solutions and concepts.

Tabell 5.1 Concept metrics

| No. | Need number | Metrics | Weight | Unit | Max. | Min. | Prefered |
|-----|-------------|---------------------|--------|-----------------|------|------|----------|
| 1 | | Cost | | SEK | | | |
| 2 | | Height | | mm | | | |
| 3 | | Base area | | mm ² | | | |
| 4 | | Max voltage / Power | | V/W | | | |
| 5 | | Assembly time | | s | | | |

Concept Screening

Sub-problems

The problem of the design and its functions is then divided into sub-problems. Each sub-problem can now have several different solutions that can be ranked and compared.

Tabell 5.2 Sub-problems

| Sub-problems | Comment |
|-------------------------|---------|
| Opening mechanism | |
| Placement of light bulb | |
| PCB mounting | |
| Tin mounting | |
| Lamp housing mounting | |

On the following criteria the generated solutions were ranked and compared to each other.

Concept Screening Criteria

Tabell 5.3 Concept screening criteria

| No. | Criteria | Comment |
|-----|--------------------|--------------------------|
| 1 | Manufacturability | |
| 2 | Durability | |
| 3 | Assembly | |
| 4 | Size | Height and width |
| 5 | Design complexity | E.g. number of parts |
| 6 | Prototyping | E.g. 3D-printability |
| 7 | Environment proof | |
| 8 | Manufacturing cost | Rough estimation on cost |

Criteria Weights

The weighting of the criteria has been made with prototyping as the main focus but at the same time trying to look ahead into the future and possible manufacturing. From the industrial supervisor came instructions that the ease of assembly when manufactured should be taken into careful consideration.

Tabell 5.4 Criteria weights

| Criteria | Points | Percentage of total points |
|--------------------|--------|----------------------------|
| Manufacturability | 2 | 10,00% |
| Durability | 2 | 10,00% |
| Assembly | 4 | 20,00% |
| Size | 3 | 15,00% |
| Design complexity | 2 | 10,00% |
| Prototyping | 5 | 25,00% |
| Enviornment proof | 1 | 5,00% |
| Manufacturing Cost | 1 | 5,00% |
| Sum | 20 | 100,00% |

Concept generation

The sub-problem solutions were generated in brainstorming sessions with complete freedom to try new ideas. However for the first prototype to be assembled into an existing lamp housing the screw in version was mandatory. Sketches were made in notebooks, on whiteboard and some existed only in our heads. Inspiration was also found in existing material such as indoor lamps, similar products and outdoor lighting systems.

Concept Evaluation

From the concept generation the three most promising solutions entered into the concept evaluation. Here each solution was given a score on every criteria. The score was then weighted and summed.

Tabell 5.5 Concept evaluation

| Sub problems | Concept | Manufacturability | | Durability | | Assembly | | Size | | Design complexity | | Prototyping | | Environment proof | | Manufacturing Cost | |
|-------------------------|---------------------|-------------------|-----|------------|-----|----------|-----|-------|------|-------------------|-----|-------------|------|-------------------|------|--------------------|------|
| | | Score | WS. | Score | WS. | Score | WS. | Score | WS. | Score | WS. | Score | WS. | Score | WS. | Score | WS. |
| Opening mechanism | Screw | 5 | 0,5 | 4 | 0,4 | 3 | 0,6 | 3 | 0,45 | 4 | 0,4 | 4 | 1 | 4 | 0,2 | 2 | 0,1 |
| | Hinge | 2 | 0,2 | 2 | 0,2 | 5 | 1 | 3 | 0,45 | 2 | 0,2 | 2 | 0,5 | 2 | 0,1 | 3 | 0,15 |
| | Box with click | 3 | 0,3 | 1 | 0,1 | 4 | 0,8 | 3 | 0,45 | 3 | 0,3 | 5 | 1,25 | 3 | 0,15 | 3 | 0,15 |
| Placement of light bulb | High, center | 4 | 0,4 | 3 | 0,3 | 4 | 0,8 | 1 | 0,15 | 3 | 0,3 | 3 | 0,75 | 3 | 0,15 | 3 | 0,15 |
| | Low, center | 4 | 0,4 | 4 | 0,4 | 4 | 0,8 | 5 | 0,75 | 3 | 0,3 | 3 | 0,75 | 3 | 0,15 | 4 | 0,2 |
| | Low, off center | 2 | 0,2 | 3 | 0,3 | 2 | 0,4 | 3 | 0,45 | 1 | 0,1 | 2 | 0,5 | 3 | 0,15 | 1 | 0,05 |
| PCB mounting | Clamped | 3 | 0,3 | 4 | 0,4 | 3 | 0,6 | 3 | 0,45 | 2 | 0,2 | 2 | 0,5 | 3 | 0,15 | 4 | 0,2 |
| | Pegs | 3 | 0,3 | 2 | 0,2 | 4 | 0,8 | 3 | 0,45 | 2 | 0,2 | 3 | 0,75 | 3 | 0,15 | 4 | 0,2 |
| | Resting | 5 | 0,5 | 2 | 0,2 | 2 | 0,4 | 3 | 0,45 | 4 | 0,4 | 5 | 1,25 | 3 | 0,15 | 5 | 0,25 |
| Tin mounting | Rotatation mounting | 3 | 0,3 | 2 | 0,2 | 3 | 0,6 | 3 | 0,45 | 2 | 0,2 | 4 | 1 | 3 | 0,15 | 3 | 0,15 |
| | Through hole | 5 | 0,5 | 5 | 0,5 | 4 | 0,8 | 3 | 0,45 | 5 | 0,5 | 4 | 1 | 3 | 0,15 | 5 | 0,25 |
| | Click version | 1 | 0,1 | 1 | 0,1 | 3 | 0,6 | 2 | 0,3 | 1 | 0,1 | 1 | 0,25 | 3 | 0,15 | 1 | 0,05 |
| Lamp housing mounting | Screw in | 3 | 0,3 | 3 | 0,3 | 3 | 0,6 | 3 | 0,45 | 3 | 0,3 | 3 | 0,75 | 3 | 0,15 | 3 | 0,15 |

Tabell 5.6 Sup-problem Weighted Scores

| Sub problems | Sub-concepts | Total weighted score |
|-------------------------|---------------------|----------------------|
| Opening mechanism | Screw | 3,65 |
| | Hinge | 2,8 |
| | Box with click | 3,5 |
| Placement of light bulb | High, center | 3 |
| | Low, center | 3,75 |
| | Low, off center | 2,15 |
| | Mid, center | 3,6 |
| PCB mounting | Clamped | 2,8 |
| | Pegs | 3,05 |
| | Resting | 3,1 |
| Tin mounting | Rotatation mounting | 3,05 |
| | Through hole | 4,15 |
| | Click version | 1,65 |
| | None | 3,85 |
| Lamp housing mounting | Screw in | 3 |

Concept combinations

From the sub-concepts provided four full solutions were created, called concept 1-4. They all apply different combinations of the sub-concepts to find the best fit of solutions.

Tabell 5.7 Concept combinations

| Concept | Trivial Name | Opening mechanism | Light bulb | PCB mounting | Tin mounting | House mounting |
|-----------|------------------|-------------------|--------------|--------------|---------------|----------------|
| Concept 1 | Top-score | Screw | Low, center | Resting | Through hole | Screw In |
| Concept 2 | Our choice, v1.1 | Screw | High, center | Pegs | Through hole | Screw In |
| Concept 3 | Box | Box with click | Low, center | Pegs | Through hole | Screw In |
| Concept 4 | Hinge | Hinge | Low, center | Clamped | TH/Clamp Hole | Screw In |

Tabell 5.8 Concept Advantages & Disadvantages

| Concept combination | Trivial Name | Advantages | Disadvantages |
|---------------------|-------------------|---|---|
| Concept 1 | Top-score | <ul style="list-style-type: none"> *No tin mouting *Lower design *Few parts *Easy assembly *Easy access to buttons | <ul style="list-style-type: none"> *Leaves the PCB exposed *Questionable tin mounting durability |
| Concept 2 | Three part design | <ul style="list-style-type: none"> *Multiple assembly options. *Tin mouting support | <ul style="list-style-type: none"> *3 parts *More complex *Tall design *Tall tins |
| Concept 3 | Box | <ul style="list-style-type: none"> *Few parts *Medium height design *Easy assembly *Tin mouting support | <ul style="list-style-type: none"> *Buttons hard to reach *Very reliant on click-mechanism *Reliant on tolerances |
| Concept 4 | Hinge | <ul style="list-style-type: none"> *Very easy assembly *Few parts *Medium hieght design *Tin mounting support | <ul style="list-style-type: none"> *Poor protection against environment *Reliant on tolerances *Reliant on locking mechanism *Reliant on hinges |

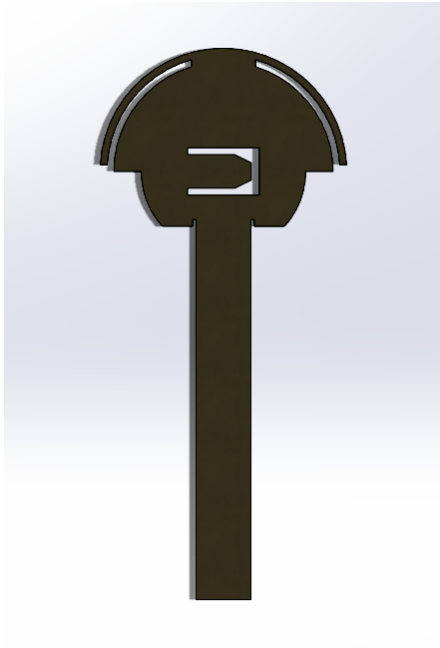
5.2 Contact metal

Depending on the chosen design for the casing the contact metal that links the circuit board and light bulb needs to be designed to fit. This will have to be an original custom design and to be manufactured separately since they must be a conductive material. The most important aspect is that it is easy to assemble. Some aspects have to be developed together with the casing, such as the connection to the light bulb.

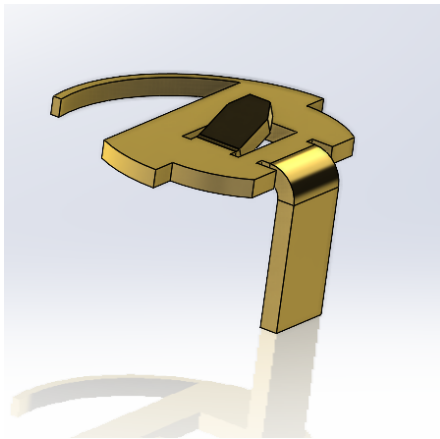
Connection to light bulb

The connection to the light bulb is very important since there must be a link between the circuit board and the light bulb. One contact sheet needs to be touching the active part of the bulb and the other the passive. The passive part of the light bulb is the conductive threading and the active part is the conductive metal plate at the bottom. Different designs were considered in early stages when there was a considerable distance between the bulb and the board due to components in the way. Figure 5.1 and 5.2 are two designs of contact metal that were planned to be easily assembled by just clicking them into position. During the development of the iterations of both components and casing the distance between board and light bulb could be

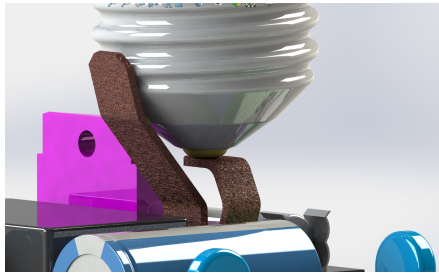
decreased which allowed for simpler designs. The 3D model in figure 5.3 was used to evaluate how much space the contact metal needed on the circuit board.



Figur 5.1 A spring locked version of contact metal



Figur 5.2 A mechanically locked, spring fastened version.



Figur 5.3 An early visualization of the contact metal

Material

A conductive material is a necessity and the two options available were either copper or brass. For the prototype stage the price was not a factor but rather the springiness of the material and the deformation under pressure. When pushed down in contact with the bulb the contact metal needs to bend but then spring back when released. What was found was that brass does not have the same elasticity as copper and therefore copper was chosen.

Thickness

A too thin contact metal will deform too easily and a too thick will not flex at all, it will also be hard to manufacture. A batch of 1mm copper sheets were laser cut as can be seen in figure 5.4. However, the price was high and the elasticity was not satisfactory. 0.5mm showed better elasticity under the forces that could be expected and at that thickness it can be cut with a pair of scissors.



Figur 5.4 An early design of the 1mm active contact metal which has been laser cut and then bent by hand.

Footprint

The footprint needed to be small enough to fit on the board but big enough so the contact sheets could be assembled by hand and still support the weight and forces upon them. Again several versions were tried including one large hole to be soldered, a flat contact pad and a tripod solution. None were without its issues but the tripod solution showed the most promise to be soldered by hand.

6

Electronic Design

6.1 Schematics

To start the electronic design an example schedule was provided from the industrial supervisors with what might need to be included in the final schematic. The schematic contained the symbols of the target specification of the electronics. The first task was to decide what systems or components were either redundant or missing. The full electric schedule can be found in appendix A.2.

In KiCad the symbol for each component is important in order for the program to assign the correct pins in the PCB layout mode. Therefore the first step when adding a component to the schematic is to assign it the correct symbol. Many component symbols already exist in KiCad or can be modified easily to fit. However some symbols need to be custom designed in order to have the correct pin layout or numbering.

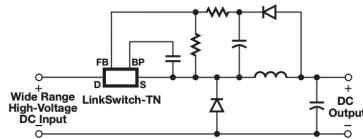
Component selection

All of the components below have been chosen with the same criteria in mind. Firstly, they must be kept in stock at the retailer. The lead times for many components are very long due to semi-conductor shortage, global supply-chain issues and the Covid-19 pandemic. Secondly the components should be as small as possible but not to an unmotivated price. There must therefore exist a balance between price and size. See appendix A.1 for the full component list.

Microcontroller unit. The MCU was the first part chosen and it is the component the project revolves around. The Nordic nRF52840 system on chip solution has GPIO, SPI, UART, I2C, I2S, PMD, PWM, ADC, NFC and USB interfaces for connecting peripherals and sensors. However, this project does not make use of all of these functions. The general-purpose input/output pin registry (GPIO) is the most important for the functioning circuit board. The assigned GPIO pins control every input and output from button to light bulb with their assigned function. The nRF52840 also has Bluetooth Low Energy and Mesh compatibility which is deve-

loped specifically for the purpose of simple use in IoT networks and smart home technology.

Power conversion. The first task was to find components that could meet the system specifications for the power conversion, from 230V alternating current to rectified 3.3V which is what the nRF52840 requires. For this a circuit using an LNK306D was implemented. This is a Off-Line IC switcher that can be used in different configurations depending on the input and desired output. Figure 6.1 shows a standard configuration of the off-line switch converter which is used to convert high voltage into low voltage.



Figur 6.1 A general configuration of the Off-Line switch converter

Solid state relay. The circuit board is powered with 230 VAC, this is needed to power the light bulb. The solid state relay acts as guard for when the high current is allowed to reach the light bulb. The relay is in turn controlled by a GPIO pin on the microcontroller that lets a small current activate the relay and turn on the light bulb.

Fuse. An electric fuse is a component to protect from too high voltage. When the voltage through the fuse is higher than the fuse rating the electronic conductor inside the fuse breaks and no longer conducts. If it breaks it is possible to swap this component out in order to make the board work again.

Diodes. Two types of diodes are used, the most commonly known is the light emitting diode (LED). There are two LEDs on the board, one is a RGB diode and used an interface and is programmable. The other LED is an indicator that the 230VAC is conducting to the light bulb. The second type of diode is the rectifier which allows for the current to pass in one direction but not the other in order to rectify the alternating current.

Capacitors. The capacitors are in the circuit to protect delicate electronics from disturbances. The capacitor stores electric charge and then discharges smoothly to avoid sudden drops in voltage. This protects the microcontroller and the other components and provide them with a consistent current and a smooth voltage. [WatElectronics, 2022a]

Resistors. Resistors are used to limit the current flow and divide the voltage in order to deliver the exact current and voltage a component requires. The chosen resistors are all fixed resistors with a stable resistance value. They can either be

surface mounted or axial, and in consideration to size they are all surface mounted. [WatElectronics, 2022b]

Inductors. Inductors are coils of wire that can store a charge in the form of a magnetic field while the coil has power. On its own it can function as a low pass filter and together with capacitors and resistors more specific filters can be devised. [Burris, 2022]

Optocoupler. An optocoupler is used to isolate high voltage from low voltage and therefore protect parts of a circuit. In this case a signal from GPIO pin 15 on the microcontroller allows a current to pass through the optocoupler and this in turn activates the relay and turns on the light bulb.

Terminal. The terminal connects the two power cables into the back of the circuit board. The two criteria the terminal needs to fulfill are, small size and the possibility of connecting the cables into the terminal by hand. This terminal is spring loaded and very small which makes it ideal for this product.

6.2 Electronic design

Board layout

With the board layout certain things need to be taken into consideration. The goal is a functioning circuit board that complies with EMC rules. It should not interfere with other systems and it should not be disturbed by other systems or itself. Electromagnetic emission can come from both electrical discharges as well as switching devices and the traces in the board can act as radio antennas. For this reason the traces need to be kept short and also at a distance from surrounding components and traces. To comply with EMC the 230VAC needs to be separated from the 3.3VDC traces as much as possible to avoid interference. At the same time certain components need to be in fixed positions, such as the footprint for the connectors to the light bulb. These have to be centered in order to be easily assembled. For accessibility the LEDs and the buttons need to be close to the edge. The Nrf52840 must not be located directly over ground therefore it also needs to be on the perimeter. The interface pad for programming is placed close to the MCU and close to the edge for accessibility and for a short trace layout. When the largest and most critical components have been placed smaller components such as resistors and capacitors are fitted to be as shortly routed as possible. In order to not intersect traces most of the horizontal traces have been drawn on the front and the vertical traces have been drawn on the back. In figure 6.2 it is possible to see the finished KiCad drawing where the red colour signifies the traces and footprints on the front. The blue traces are drawn on the back of the board. Figures 6.3 and 6.4 show the routing and the footprints of the finished board.

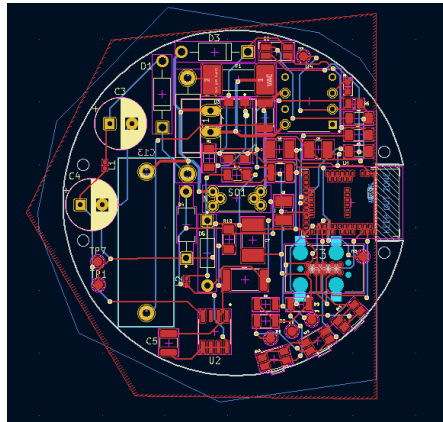


Figure 6.2 A KiCad view of the layout for the traces and footprints, red fields mark the front and blue lines are on the back

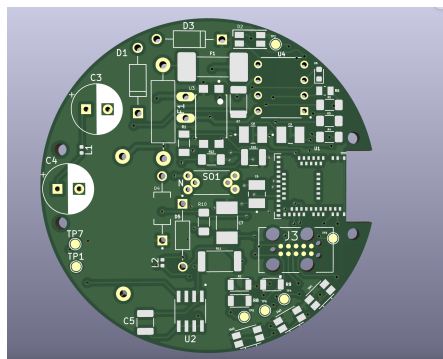
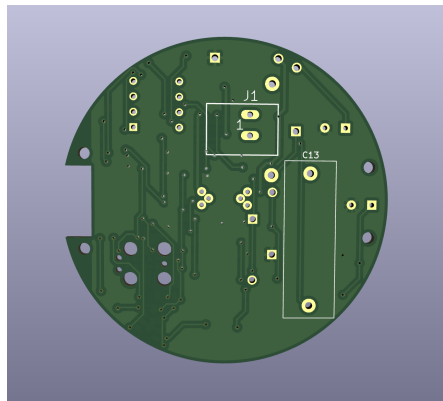


Figure 6.3 Frontal view of the traces and footprints of the board,



Figur 6.4 View from the back of the traces and footprints of the board

7

Software Design

7.1 Embedded programming

The embedded development of this product has used Nordic Semiconductors' nRF Connect software development kit. This integrates the Zephyr Real Time Operating System. When using Nordic semiconductors' software development tool it is possible to get access to many existing applications. This project was based on an existing application that uses Bluetooth mesh functionality and from there this embedded code has been further developed. The language is C and the IDE used is Visual Studio Code. The code has been stored on Github in Adevos' repository.

Design with Zephyr OS

Zephyr OS offers many kernel services and a number of services for development:

- Multi-threading services for different kinds of threads such as cooperative, priority-based, non-preemptive, and preemptive threads. These have optional round robin time-slicing. This includes POSIX pthreads with compatible API support.
- Interrupt services for compile-time registration of interrupt handlers.
- Memory allocation services for an allocation and freeing of fixed and variable-size memory blocks.
- Synchronization for threads with binary semaphores, counting semaphores, and mutex semaphores.
- Inter-thread data passing in message queues and byte streams.
- Power management services such as tickless idle.
- It is highly configurable and modular for flexibility. This allows an application to only use the capabilities it needs, and to specify their quantity and size.

- Cross architecture supports many boards with different CPU architectures and developer tools.
- Memory protection and it also implements thread isolation with thread-level memory protection on x86, ARC, and ARM architectures, userspace and memory domains.
- For memory constrained devices, Zephyr can create a monolithic image that gets loaded and executed on a system's hardware. Both the application code and kernel code execute in a single shared address space.
- Compile-time resource definition allows system resources to be defined at compile-time, which reduces code size and increases performance for resource-limited systems.
- Devicetree support with the use of devicetree to describe hardware. Information from devicetree is used to create the application image.
- Native networking stack supporting multiple protocols. Networking support is fully featured and optimized. OpenThread support (on Nordic chipsets) is also provided. This is a mesh network designed to securely and reliably connect hundreds of products around the home.
- Bluetooth Low Energy 5.0 support Bluetooth 5.0 compliant (ESR10) and Bluetooth Low Energy Controller support (LE Link Layer). Includes Bluetooth mesh and a Bluetooth qualification-ready Bluetooth controller.
- Virtual file system interface for memory constrained applications, and file system enhancements for logging and configuration.
- Powerful multi-backend logging framework with support for log filtering, object dumping, panic mode and multiple backends.
- User friendly and full- featured Shell interface. There is a multi-instance shell subsystem with user-friendly features such as auto-completion, wildcards, coloring, and history. It also supports static commands and dynamic sub-commands.
- Settings on non-volatile storage. The settings subsystem gives modules a way to store persistent per-device configuration and runtime state. Settings items are stored as key-value pair strings.
- Non-volatile storage (NVS). NVS allows storage of binary blobs, strings, integers, longs, and any combination of these.
- Native POSIX port. This supports running Zephyr as a Linux application with support for various subsystems and networking.

7.2 Software design requirements

Here are the final software design requirements listed and in the following sections the implementation of each requirement will be described.

- Turn on and off light bulb in relation to the position of the sun.
- Know its global coordinates.
- Know the local time.
- Toggle light from a smart device.
- Connect to a secure network.
- Save set values.
- Save set configurations after power loss without effecting performance.
- A cheap solution with a memory constraint device.
- I/O needs to fulfill user requirements through push button and LEDs.

I/O

For handling I/O there is a generic layer that from the device tree configuration, at compile time, allows the run time to find and configure the device I/O. It gives them descriptive names that can be used to access individual I/Os. The generic layer also handles the initialization of the I/O and registers call back functions for the interrupt handling of individual I/O or groups of I/Os. Figures 7.1 and 7.2 below show two examples. The use of this generic code will allow the application programmer to focus on two areas:

- Initialization functions and registering of callback functions for interrupt handling.
- Reading and writing to individual I/Os.

```

int dk_buttons_init(button_handler_t button_handler)
{
    int err;

    button_handler_cb = button_handler;

    gpio_flags_t flags = (IS_ENABLED(CONFIG_DK_LIBRARY_INVERT_BUTTONS) ?
                          GPIO_PULL_UP : GPIO_PULL_DOWN);

    if (IS_ENABLED(CONFIG_DK_LIBRARY_DYNAMIC_BUTTON_HANDLERS)) {
        k_mutex_init(&button_handler_mut);
    }

    for (size_t i = 0; i < ARRAY_SIZE(button_pins); i++) {
        button_devs[i] = device_get_binding(button_pins[i].port);
        if (!button_devs[i]) {
            LOG_ERR("Cannot bind gpio device");
            return -ENODEV;
        }

        err = gpio_pin_configure(button_devs[i], button_pins[i].number,
                                GPIO_INPUT | flags);

        if (err) {
            LOG_ERR("Cannot configure button gpio");
            return err;
        }
    }

    uint32_t pin_mask = 0;

    for (size_t i = 0; i < ARRAY_SIZE(button_pins); i++) {
        /* Module starts in scanning mode and will switch to
         * callback mode if no button is pressed.
         */
        err = gpio_pin_interrupt_configure(button_devs[i],
                                           button_pins[i].number, GPIO_INT_DISABLE);
        if (err) {
            LOG_ERR("Cannot disable callbacks ()");
            return err;
        }

        pin_mask |= BIT(button_pins[i].number);
    }

    gpio_init_callback(&gpio_cb, button_pressed, pin_mask);

    for (size_t i = 0; i < ARRAY_SIZE(button_pins); i++) {
        err = gpio_add_callback(button_devs[i], &gpio_cb);
        if (err) {
            LOG_ERR("Cannot add callback");
            return err;
        }
    }

    k_work_init_delayable(&buttons_scan, buttons_scan_fn);

    state = STATE_SCANNING;

    k_work_schedule(&buttons_scan, K_NO_WAIT);

    dk_read_buttons(NULL, NULL);

    atomic_set(&my_buttons, (atomic_val_t)get_buttons());

    return 0;
}

```

Figur 7.1 The button-init function

```

void dk_read_buttons(uint32_t *button_state , uint32_t *has_changed)
{
    static uint32_t last_state;
    uint32_t current_state = atomic_get(&my_buttons);

    if (button_state != NULL) {
        *button_state = current_state;
    }

    if (has_changed != NULL) {
        *has_changed = (current_state ^ last_state);
    }

    last_state = current_state;
}

```

Figur 7.2 The read-button function

BLE

There are three layers that make up the BLE protocol stack. The first layer is Host which sits right below the application. According to the Zephyr documentation [Zephyr, 2022a] it is comprised of multiple network and transport protocols which enables communication with peer devices. The controller layer implements the Link Layer. This is a real-time protocol that provides standard communication together with the Radio Hardware layer. The Link Layer guarantees the delivery of data and handles all the Link Layer control procedures. The Radio Hardware layer implements the needed baseband functional blocks which allow the Link Layer firmware to pass and receive messages in the 2.4GHz spectrum. The Bluetooth message passing is handled by a single thread. Nordic RF has it's own layers for coding and using the NRF MCU Bluetooth stack. It fully supports functionality as Bluetooth mesh provisioning, standard mesh models and custom mesh models. The NRF C framework has a lot of convenient C macros enabling and setting up mesh models where the application programmer only provides the callback functions for handling received messages and their data. It also have function for each models that allow the application programmer to send specific messages that is a part of a mesh model. In this application mesh models for Generic ON/OFF, Time server, Location server has been implemented with the NRF framework. There is also a custom Adevo mesh model that can transfer data for the application. Figure 7.3 show the model handler .c file where Bluetooth models are implemented.

Kapitel 7. Software Design

```
/*
 * Copyright (c) 2019 Nordic Semiconductor ASA
 *
 * SPDX-License-Identifier: LicenseRef-Nordic-5-Clause
 */

#include <zephyr/bluetooth/bluetooth.h>
#include <bluetooth/mesh/models.h>
#include <bluetooth/mesh/time.h>
#include <bluetooth/mesh/time_srv.h>
#include <bluetooth/mesh/time.h>
#include <math.h>
#include <bluetooth/mesh/time_cli.h>

#include "model_handler.h"
#include "time_srv_model.h"
#include "onoff_srv_model.h"
#include "health_srv_model.h"
#include "location_srv_model.h"
#include "flash_handler.h"
#include "dk_buttons_and_leds.h"

static const struct bt_mesh_onoff_srv_handlers onoff_handlers = {
    .set = luminaire_set,
    .get = luminaire_status,
};

struct bt_mesh_onoff_srv onoff_server = BT_MESH_ONOFF_SRV_INIT(&onoff_handlers);

static void led_set(struct bt_mesh_onoff_srv *srv, struct bt_mesh_msg_ctx *ctx,
                   const struct bt_mesh_onoff_set *set,
                   struct bt_mesh_onoff_status *rsp);

static void led_get(struct bt_mesh_onoff_srv *srv, struct bt_mesh_msg_ctx *ctx,
                   struct bt_mesh_onoff_status *rsp);

struct led_ctx
{
    struct bt_mesh_onoff_srv srv;
    struct k_work_delayable work;
    uint32_t remaining;
    bool value;
};

static struct led_ctx led_ctx[] = {
#ifdef DT_NODE_EXISTS(DT_ALIAS(led0))
    { .srv = BT_MESH_ONOFF_SRV_INIT(&onoff_handlers) },
#endif
#ifdef DT_NODE_EXISTS(DT_ALIAS(led1))
    { .srv = BT_MESH_ONOFF_SRV_INIT(&onoff_handlers) },
#endif
#ifdef DT_NODE_EXISTS(DT_ALIAS(led2))
    { .srv = BT_MESH_ONOFF_SRV_INIT(&onoff_handlers) },
#endif
#ifdef DT_NODE_EXISTS(DT_ALIAS(led3))
    { .srv = BT_MESH_ONOFF_SRV_INIT(&onoff_handlers) },
#endif
};

static void led_transition_start(struct led_ctx *led)
{
    int led_idx = led - &led_ctx[0];

    /* As long as the transition is in progress, the onoff
     * state is "on":
     */
    dk_set_led(led_idx, true);
    k_work_reschedule(&led->work, K_MSEC(led->remaining));
    led->remaining = 0;
}
```

```

static void led_status(struct led_ctx *led, struct bt_mesh_onoff_status *status)
{
    /* Do not include delay in the remaining time. */
    status->remaining_time = led->remaining ? led->remaining :

    k_ticks_to_ms_ceil32(k_work_delayable_remaining_get(&led->work));
    status->target_on_off = led->value;
    /* As long as the transition is in progress, the onoff state is "on": */
    status->present_on_off = led->value || status->remaining_time;
}

static void led_set(struct bt_mesh_onoff_srv *srv, struct bt_mesh_msg_ctx *ctx,
                   const struct bt_mesh_onoff_set *set,
                   struct bt_mesh_onoff_status *rsp)
{
    struct led_ctx *led = CONTAINER_OF(srv, struct led_ctx, srv);
    int led_idx = led - &led_ctx[0];

    if (set->on_off == led->value)
    {
        goto respond;
    }

    led->value = set->on_off;
    if (!bt_mesh_model_transition_time(set->transition))
    {
        led->remaining = 0;
        dk_set_led(led_idx, set->on_off);
        goto respond;
    }

    led->remaining = set->transition->time;

    if (set->transition->delay)
    {
        k_work_reschedule(&led->work, K_MSEC(set->transition->delay));
    }
    else
    {
        led_transition_start(led);
    }

respond:
    if (rsp)
    {
        led_status(led, rsp);
    }
}

static void led_get(struct bt_mesh_onoff_srv *srv, struct bt_mesh_msg_ctx *ctx,
                   struct bt_mesh_onoff_status *rsp)
{
    struct led_ctx *led = CONTAINER_OF(srv, struct led_ctx, srv);

    led_status(led, rsp);
}

static void led_work(struct k_work *work)
{
    struct led_ctx *led = CONTAINER_OF(work, struct led_ctx, work.work);
    int led_idx = led - &led_ctx[0];

    if (led->remaining)
    {
        led_transition_start(led);
    }
    else
    {
        dk_set_led(led_idx, led->value);

        /* Publish the new value at the end of the transition */
        struct bt_mesh_onoff_status status;

        led_status(led, &status);
        bt_mesh_onoff_srv_pub(&led->srv, NULL, &status);
    }
}

```

Kapitel 7. Software Design

```
/* Set up a repeating delayed work to blink the DK's LEDs when attention is
 * requested.
 */
static struct k_work_delayable attention_blink_work;
static bool attention;

    basicstyle=\tiny, %or \small or \footnotesize etc.
]

static void attention_blink(struct k_work *work)
{
    static int idx;
    const uint8_t pattern[] = {
# if DT_NODE_EXISTS(DT_ALIAS(led0))
        BIT(0),
# endif
# if DT_NODE_EXISTS(DT_ALIAS(led1))
        BIT(1),
# endif
# if DT_NODE_EXISTS(DT_ALIAS(led2))
        BIT(2),
# endif
# if DT_NODE_EXISTS(DT_ALIAS(led3))
        BIT(3),
# endif
    };

    if (attention)
    {
        dk_set_leds(pattern[idx++ % ARRAY_SIZE(pattern)]);
        k_work_reschedule(&attention_blink_work, K_MSEC(30));
    }
    else
    {
        dk_set_leds(DK_NO_LEDS_MSK);
    }
}

static const struct bt_mesh_health_srv_cb health_srv_cb = {
    .attn_on = attention_on,
    .attn_off = attention_off,
};

static struct bt_mesh_health_srv health_srv = {
    .cb = &health_srv_cb,
};

BT_MESH_HEALTH_PUB_DEFINE(health_pub, 0);

// static struct bt_mesh_time_srv time_srv = BT_MESH_TIME_SRV_INIT(time_update_cb);

static const struct bt_mesh_loc_srv_handlers loc_handlers = {
    .global_get = global_location_get,
    .global_set = global_location_set,
    .local_get = local_location_get,
    .local_set = local_location_set};

static struct bt_mesh_loc_srv location_srv = BT_MESH_LOC_SRV_INIT(&loc_handlers);

static struct bt_mesh_elem elements[] = {
    BT_MESH_ELEM(
        1, BT_MESH_MODEL_LIST(BT_MESH_MODEL_CFG_SRV,
            BT_MESH_MODEL_HEALTH_SRV(&health_srv, &health_pub),
            BT_MESH_MODEL_TIME_SRV(&time_srv),
            BT_MESH_MODEL_LOC_SRV(&location_srv),
            BT_MESH_MODEL_ONOFF_SRV(&onoff_server)),
        BT_MESH_MODEL_NONE),
};

static const struct bt_mesh_comp comp = {
    .cid = CONFIG_BT_COMPANY_ID,
    .elem = elements,
    .elem_count = ARRAY_SIZE(elements),
};
```

```
// void time_srv_init(void);  
const struct bt_mesh_comp *model_handler_init(void)  
{  
    init_health_svr();  
    time_srv_init();  
    k_work_init_delayable(&attention_blink_work, attention_blink);  
  
    for (int i = 0; i < ARRAY_SIZE(led_ctx); ++i)  
    {  
        k_work_init_delayable(&led_ctx[i].work, led_work);  
    }  
  
    init_flash_handler();  
  
    return &comp;  
}
```

Figur 7.3 The model handler .c file

Non volatile storage

In order to store configuration parameters consistently the non volatile storage is used. It is the most minimal file system in Zephyr and it stores key pair values.

Main task

The main task of the system is to handle the switching on and off of the light bulb by constantly checking for the relevant conditions. The requirement leads to that only one thread needs to be used and that is the application main thread. To minimize the CPU load an implementation is done using a mailbox with message passing [Zephyr, 2022b]. A mailbox is a kernel object which provides message queue capabilities. The mailbox allows threads to send and receive messages synchronously or asynchronously. As seen in figure 7.4 the check of the conditions are rescheduled with 10 seconds. The message is passed with the use of mailbox to the working queue thread.

```
void init_k_work()
{
    printk("Enter init_k_work \n");

    k_work_init_delayable(&time_tracking_work, time_tracking);
    int rc = k_work_reschedule(&time_tracking_work, K_MSEC(10000));
    time_tracking(&time_tracking_work);
    printk("rc: \t%d\n", rc);
}
```

Figur 7.4 Example of a button interrupt configuration

For turning the light bulb on/off manually this is done via the Bluetooth Mesh Generic ON/OFF model. This is driven by the NRF BLE framework and the thread being used there.

Sun service

An important part of the functionality is the sun service file. The idea is that the Luminaire node knows its location as well as the time and date. Based on this it can use the sun service functions and input the date and location as parameters. Sun service then calculates at what time the sun will rise and set at this particular location. Sun service then compares the current time with sunset and sunrise and if the sun is not in the sky it sets a flag to switch on the light bulb.

Code structure

The goal of the code structure has been to keep each functionality in a separate file for easy handling and readability. Every model has its own .c and .h file and only the initiating function and functions that must be reached by other .c files are put in

the header file. Furthermore there is a file to only handle the read and write of the non-volatile storage. This handles the usage of the flash memory to remember the coordinates when the unit loses power or restarts. The model handler inputs all the existing models into an element and the model handler is then initiated in the main file.

User interface programming

The Luminaire is controlled from a Google chrome website with Bluetooth functionality. On top of this a mesh stack has been built. At the time of writing communication between the board and the website requires an app for provisioning. Once the connection is established it is possible to send messages from the website, i.e., the client side, and receive them on the board, i.e., the server side.

Time message

When a time message is sent it consists of several parts. Firstly the address to which board the message will be sent must be stated, it could also be to all boards in a mesh network. This consists of two bytes. The payload of the message then consists of the opcode which signals which model this message is sent to, this consists of two bytes. In the same payload the time follows as ten bytes. The first five bytes are the tai-time, which is seconds past since January 1st 1970. Then comes one byte of sub-seconds which is manually set to zero, the same goes for the next byte which signals the uncertainty of the message. The next two bytes are the difference between the tai time and the utc time which at the moment of writing is 37 seconds. The last two bytes are the time zone offset in 15 minute increments, which in Sweden in summer time is +2 hours so the value becomes 8. These bytes are then converted to values that can be compared to the rising and setting of the sun.

Location message

A location message is similar to a time message but the opcode specific for this model. The first four bytes in the payload consists of the latitude and the four following are the longitude. The last two are the altitude but they are currently not used. When writing and looking at the messages hexadecimal numbers are used for readability, two hexadecimals can hold the same value as eight bits which is between 0 and 255 or 0 and 0xFF.

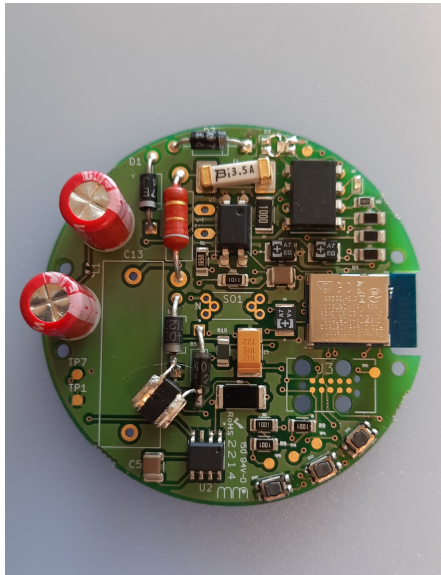
8

Results

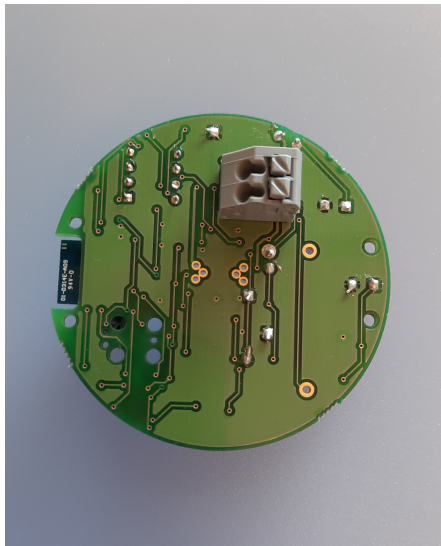
8.1 Electronic design

The circuit board was manufactured according to the KiCad drawings made and the manufacturing went without problems. Then the circuit boards were given to the electronic consultants Cervitrol in Lund where most of the components were attached. Even though checks of components and circuit boards had been completed before the purchasing of components and assembly several errors were found at this stage. Firstly the buttons which were planned to be 90 degrees turned out from the board for easy access did not fit the footprints on the board, instead new buttons that did fit the footprint needed to be purchased. However these are clicked on from the top of the circuit board instead which makes them inaccessible from the outside of the casing. Secondly the RGB LED was a common cathode instead of a common anode which meant that a new RGB LED needed to be bought, but no common anode fitted the footprint well enough which meant that the constructor was forced to make a homemade soldering solution. The third major issue was the inductors, there had been a mix up of which component to use so the component bought was no inductor at all. This meant that the footprint was too small to fit an inductor. In order to solve this issue one inductor had to be left out completely and another could be soldered on upside down and bypassing the footprint completely. Furthermore it was decided that the large capacitor c13 was not critical and could therefore be left out completely, making more space for the inductor. The finished product can be seen in figures 8.1, 8.2 and 8.3.

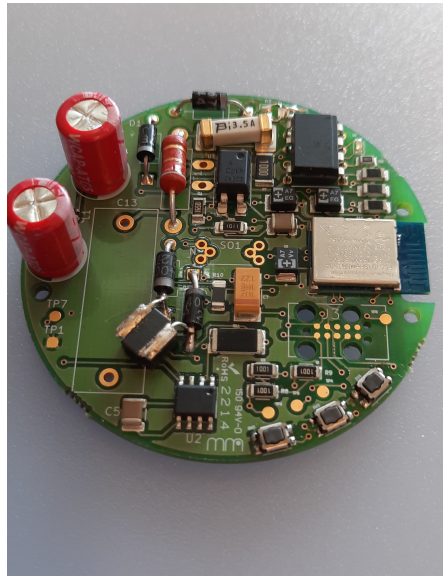
Despite all this the low power components work as intended when run with 3.3 volts and the Bluetooth functionality is up and running as well as the LEDs and the buttons. However there seems to be some kind of routing error in the high voltage part of the circuit board. This means that when connected to 230VAC the voltage is not stepped down and rectified as planned and this damages components and short circuits on the board. What this routing error is caused by is not clear but the result is that the board cannot work as intended without a separate AC/DC converter.



Figur 8.1 The assembled circuit board as seen from the top



Figur 8.2 The assembled circuit board as seen from the bottom.



Figur 8.3 The assembled circuit board as seen up close

8.2 Hardware design

The design of the casing has followed the changes of the circuit board closely and when the large capacitor c13 was no longer needed the casing could be remodeled to become lower. This in turn meant that the light bulb could come closer to the circuit board and the contact metal needed only to extend a very short distance from the board. On the board there are four mounting holes in the event that it needed to be locked into position on pegs. These are not used in this design of casing, however, they presented an opportunity to instead attach a small plastic part which now is named a bridge. The bridge keeps the contact copper from bending out of position and touch to avoid a short circuit. It also acts as a stop for the light bulb so it cannot come in contact with the board itself. Figure 8.4 shows what it looks like on the inside of the casing.



Figure 8.4 The opened casing with the bridge in place

From the product development process it can be seen that this is the concept that scored highest in the evaluation, which was named concept 1. The board rests on a ledge on the inside of the casing. In the bottom there is hole which allows the cables to have access to the terminal, as can be seen in figure 8.5.

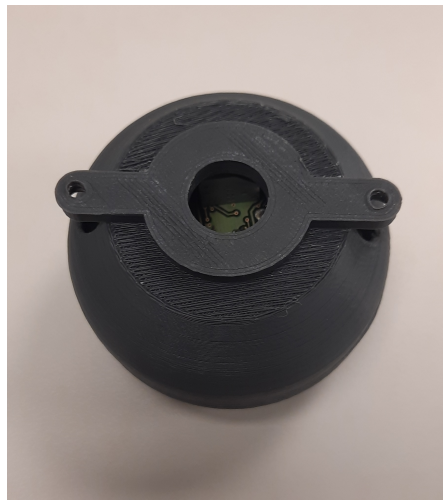
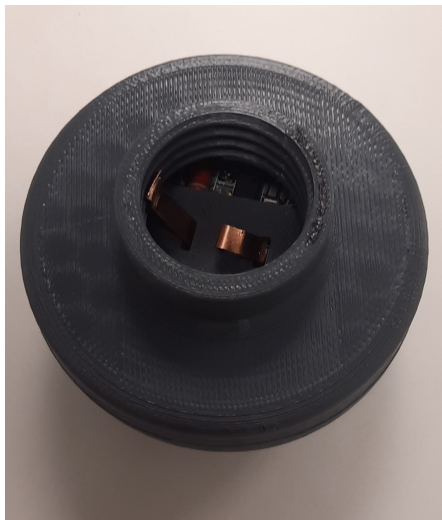


Figure 8.5 The casing seen from the bottom

The tin mounting is as described a through hole solution and the light bulb is positioned low in the center which can be seen in figures 8.6 and 8.9. The upper and lower parts of the casing are both threaded with M20 2,5mm thread which is a standard thread size. It is then opened and closed by turning. The entire casing is then screwed to the lamp housing with 3mm screws as seen in figures 8.9 and 8.8.



Figur 8.6 A general configuration of the Off-Line switch converter



Figur 8.7 A general configuration of the Off-Line switch converter



Figur 8.8 The casing and the light bulb is mounted to the lamp housing



Figur 8.9 The casing is mounted to the lamp house

8.3 Software results

At the time of writing many of the goals for software functionality have been fulfilled. From the website it is possible to connect to the card through Bluetooth and send it commands. The original goal to control lights with the rising and setting of the sun has been fulfilled. From the website it is possible to send a time and location to switch on LEDs. On the server side the message is received and handled through the models to blink a LED. The buttons and LEDs work even though they have not been assigned to a useful functionality yet. There is much more to be done with the website and the cloud service but they are in development stages.

8.4 Testing

Originally five units were constructed. However, one was destroyed the first day during testing when it for a moment received 5 volts instead of 3.3. Otherwise the card has been tested for the functionality of the low power components which ha-

ve proved successful. From the client, which is the website, it is possible to send messages containing time and location and receive them on the server side and light up the LEDs. For most of the software testing a development kit, nrf52840 dk, with the same microcontroller and similar functionality as the custom card has been used. When connecting 230VAC to the circuit board there are critical issues. The fuse used to protect the circuit from high voltage explodes which means there are routing issues with the board. This also damages other components such as the LNK306D which needs to be replaced in the tested card.

8.5 General project results

The overall results of this project have been a great success. There are issues with the high voltage side of the circuit board and it might not be possible to light up the light bulb using only the designed card at the moment. However, many parts of the project have been successful. Most important is the knowledge gained and the proof that this can be a successful product. If this version of the board does not work exactly as intended then the process of making a new design will be much more quick and efficient.

9

Discussion

9.1 Development process

This project has been ongoing for seven months now and there have been times when the project has progressed very rapidly and there have been times when it has been to almost a standstill. I am personally very happy it was possible to extend this project beyond the original 20 weeks so that I could follow it until it was a finished product. The original plan was for me to undertake this masters' thesis together with my friend and fellow mechatronics' student Patrik Larsson. However, due to unforeseen events he was not allowed to start his thesis work. Nevertheless he has stood by this project and me all this time and without him this project would not have come as far as it has. We have worked together closely but with much independence from our supervisors at times. There has been a strong will from our supervisor that we learn by doing and to find our own ways of solving problems and structuring our work. This has generally been very appreciated, but there is no doubt that the progress has been much quicker during times of assistance from our managers, and some obstacles would not have been overcome without them.

It was said that this was to be an agile working process using kanban workflows and standup meetings, and it has been so. The standups have been consistent during all of the project. The kanban workflow has been varying depending on how close support we have gotten from the supervisors. When Patrik and I worked more independently we had difficulties in dividing tasks and setting strict time goals for our work, then the kanban board functioned more as notepad of active and passive tasks. During closer supervision the kanban flow has functioned as it was meant to with stricter time limits and division of task, however at those time we received the tasks from our managers rather than wrote them ourselves. It should, however, be taken into consideration that Bluetooth mesh communication is cutting edge technology and we are still inexperienced students. Nevertheless we have learnt a lot that can we can take with us to future projects. The iteration work of agile has come naturally with new version of circuit boards, hardware and software all forcing another to be better adapted or adjusted.

9.2 Final concept

The final concept presented here is hopefully far from being the last and final concept. There is much potential in this product and it may very well become a product on the market in the future. But before that there are many issues with the current version that need to be corrected for the next one. This project started with an electric schedule with suggestions of components and routing. That schedule would look very different if a new circuit board was made. During the course of the project we have learned many things and many of the original components have been exchanged for smaller or other components with different functionality. It is absolutely possible to make the circuit board even smaller. The circuit board itself needs to be redesigned from an EMC standpoint, the power conversion switching is most likely outputting disturbing frequencies that do not fulfill the existing requirement. I have learnt many things about the contact metal design and there needs to be a much easier way to manufacture these for future versions, also the footprint needs to be designed in a way that it can be soldered on automatically instead of by hand as it has been done now. But all things considered we have learnt an enormous amount about product development and we are happy with the finished product even though it does not function exactly as we hoped. Unfortunately that is unexpected from a first prototype. In hindsight the routing and electric schedule should have been triple checked with the electronic consultants from Cervitrol. This would have saved us several weeks in waiting for new components as well as not needing the quick fix reliability issues that we now might have. The process moved a bit too fast and communication could have been better.

10

Conclusion and future work

10.1 Future work

There is a lot of work ahead which I hope to be a part of. Even if the 230VAC would have worked a new board would have to be designed anyway. With this new board the lessons learnt from this project need to be applied. The board can be smaller, to accomplish this some components can be removed completely and some can be swapped to smaller alternatives. The board layout needs to be designed with the new EMC knowledge we have and that means putting many of the switching components closer to each other. Of course the footprints that were incorrect need to be corrected as well. Then this product has to be seen from a mass manufacturing point of view, i.e., which components are too expensive to buy in a large number and how should the contact metal be manufactured. If this product is to be an outdoor product then it needs to be environment resistant which means a new casing and adding materials such as rubber. Then there is the question of how this should be designed, prototyped and then manufactured.

10.2 Conclusion

To conclude this masters' thesis it has been a great success as a proof of concept and a first prototype. There is a bright future for IoT home appliances in general and Bluetooth mesh in particular. This means that even if this project would not lead any further we have acquired the knowledge to develop other products with mesh capability in the future. As stated before mesh technology is still very new and few guides exist on how to use this. This means that with this knowledge we can become attractive to future employers. This has been a very extensive project with three different sections, electronics, mechanics and programming. It has been very educational to try all three and as a mechatronics' student it was exactly what I hoped for. I personally found the programming to be the hardest but it is also cutting edge and where I learnt the most.

Litteratur

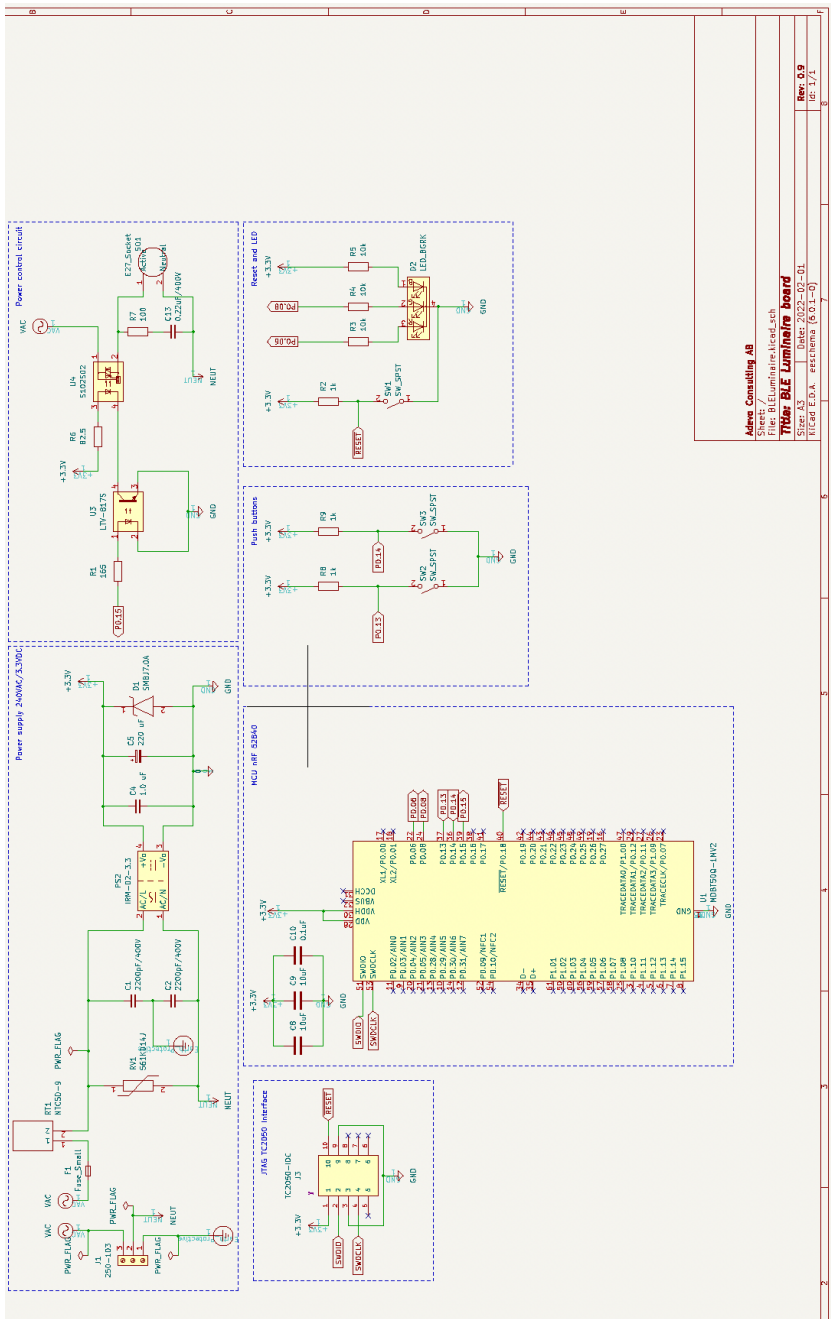
- Burris, M. (2022). URL: <https://www.lifewire.com/applications-of-inductors-818816> (hämtad 2022-07-25).
- cprime (2022). URL: <https://www.cprime.com/resources/what-is-agile-what-is-scrum/> (hämtad 2022-05-09).
- Eppinger, U. (2016). URL: <https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/202-Product-Design-and-Development-Karl-T.-Ulrich-Steven-D.-Eppinger-Edisi-6-2015.pdf> (hämtad 2022-05-19).
- Keim, R. (2022). URL: <https://www.allaboutcircuits.com/technical-articles/what-is-a-printed-circuit-board-pcb/> (hämtad 2022-04-13).
- limited, M. circuits (2022). URL: <https://www.mclpcb.com/blog/pcb-electromagnetic-issues/> (hämtad 2022-05-17).
- Prof.Mad (2022). URL: https://www.youtube.com/watch?v=JXJaRXPwjQ&ab_channel=Prof.Mad (hämtad 2022-05-15).
- Rosso, A. D. (2022). URL: <https://home.cern/news/news/computing/kicad-software-gets-cern-treatment> (hämtad 2022-04-13).
- Solidworks (2022). URL: https://www.solidworks.com/sw/docs/Corp_FactSheet_2012Q2.pdf (hämtad 2022-05-09).
- WatElectronics (2022a). URL: <https://www.watelectronics.com/mcq/capacitors/> (hämtad 2022-05-13).
- WatElectronics (2022b). URL: <https://www.watelectronics.com/types-of-resistors/> (hämtad 2022-05-13).
- Zephyr (2022a). URL: <https://docs.zephyrproject.org/latest/hardware/peripherals/gpio.html#overview> (hämtad 2022-08-09).
- Zephyr (2022b). URL: https://docs.zephyrproject.org/3.0.0/reference/kernel/data_passing/mailboxes.html (hämtad 2022-08-09).



A.1 Component list

| Part | Terminal | Part | Description | Height x Length x Width | Mounting | Mfr. No. |
|----------------------|---------------|---------|--|-------------------------|----------|---------------------------|
| Fuse | J1 | F1 | 3.5A, 250VDC, 250MAC | 3.1x10.1x0.1 | HMT | Z50-102 SF-30T1L3871-2 |
| 0.1uF | C5, C10 | C5, C10 | 50VDC ceramic 10% (XBR) | 7x3.2x2.5 | SMT | C1210C104K5HACAUTO |
| 0.22uF 400VAC | C13 | C13 | 400VAC E300DC 10% (PFP) | 16.5x26.5x6.5 | HMT | MKPH1J0320F00KSSD |
| 4.7uF 490V | C3, C4 | C3, C4 | 400 VDC electrolytic 20% | 89mm H11.5mm | HMT | 8002317408 |
| 100uF 630V | C3, C9 | C3, C9 | Tantalum Capacitors - Polymer 29 VDC 10uF 20% ESR 100mOhm | 19.0x9.0x9 | SMT | 25TCC104EB |
| 100uF 160V | C7 | C7 | Tantalum Capacitors - Solid SMT 160V 100uF 20% ESR 100ms | 2.6x3.4x3.8 | SMT | 24H100R001AT |
| 1mH | L1, L2 | L1, L2 | Fixed Inductors 0402 1800mH 100MHz 500mA AEC-Q203 30% | 0.6x1.1x3 | SMT | MOF1035A01102ATD25 |
| Diode | D5 | D5 | Rectifiers 1A, Vr 600V, RT 2000ms, Vf 1.1V | 2.7x5.2x2.7 | HMT | 1M405GP |
| Diode | D4 | D4 | Rectifiers V600V 101A | 2.7x5.2x2.7 | HMT | UF4005 |
| 3 LEDs | D2 | D2 | Rectifiers 1.0Amp 1000-Volts Trim Leads, Vf 1.1, F 5uA | 2.7x5.2x2.7 | HMT | WS-S283ATRGB |
| LED | D6 | D6 | Standard LEDs - SMD 0.6x0.3mm BLUE, If=5mA, Vf 2.9V | 0.9x1.6x0.8 | SMT | 150060CS84000 |
| 8.2 Ohm | RF1 | RF1 | Meat Film Resistors - Through Hole PR02 5% 8A 802 1W | 03.3L1.0 | HMT | FRG00502038BLA100 |
| 100 Ohm | R7 | R7 | Thin Film Resistors - Through Hole 100 Ohm 1% 250mW | 0.8x0.2x0.5 | SMT | CM20101F5X1000ELF |
| 165 Ohm | R8 | R8 | Thin Film Resistors - SMD Beige Power 2010 100R 1% 1W TC100 | 0.8x0.2x0.5 | SMT | CM20101F5X1000ELF |
| 1k | R2, R8, R9 | R1 | Thin Film Resistors - SMT 1650m 1% 250mW | 0.6x3.2x1.6 | SMT | CR1206-FX-1650ELF |
| 1.1k | R12 | R2 | Thin Film Resistors - SMD 3.5W 1600ms 1% 100ppm Ultra Hip Chip | 0.6x3x1.6 | SMT | CHP1206-FX-1001ELF |
| 1.88k | R11 | R12 | Thin Film Resistors - SMD 1.0W 1.1K 0m 0.5% 120P 25ppm | 0.5x3.2x1.6 | SMT | HR3216P-1101-D1-1 |
| 2k | R10 | R11 | Thin Film Resistors - SMD 1 Watt 1.87KOhms 1% 100ppm | 0.5x6.3x3.25 | SMT | PCV192312187PREG |
| | R3, R4, R5 | R10 | Thin Film Resistors - SMD 1W 200m 1% 20ppm AEC-Q200 | 0.8x1.6x1.2 | SMT | PPFA1206200185T11 |
| MCU | U1 | U1 | Thin Film Resistors - SMD 0.3W 1000m 2% AEC-Q200 | 0.6x3.16x1.6 | SMT | CRNA12064P1000RGEF |
| Off-Line Switcher IC | U2 | U1 | | | SMT | M09T50C1MV2 |
| Optocoupler | U3 | U2 | | | SMT | UK-S300D |
| State state Relay | U4 | U3 | Solid State Relays - PCB Mount SSR 8-pin DIP | 3.5x4.6x6.5 | HMT | LTV-8175 |
| Switches | SMT, SMT, SWS | U4 | | | HMT | PS3MFP2NSZH |
| | | | | | SMT | 436331C4552 |

A.2 Electric schedule



Adve Consulting AB

Sheet 1 of 1
TRIPLE DIMENSION BOARD
 (COPPER) 2022-07-01
 Rev: 0.9
 ID: 1/1

| | | |
|--|---|--------------------------|
| Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden | <i>Document name</i> MASTER'S THESIS | |
| | <i>Date of issue</i> August 2022 | |
| | <i>Document Number</i> TFRT-6185 | |
| <i>Author(s)</i> Måns Hofvander | <i>Supervisor</i> Mattias Wallinius, Adevo Consulting AB Maja Arvehammar, Adevo Consulting AB Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden Martina Maggio, Dept. of Automatic Control, Lund University, Sweden (examiner) | |
| <i>Title and subtitle</i> Design and Implementation of Bluetooth Mesh Outdoor Lighting | | |
| <i>Abstract</i> <p>In the days of IoT and smart homes Bluetooth is in constant development and with it the products that use Bluetooth. Bluetooth mesh is the latest feature and with it comes many opportunities of new products. This report describes the development of an outdoor light system which can be controlled from a website. The system is named Luminaire and consists of a custom built circuit board, an encapsulating casing and the software needed to run the Luminaire. Luminaire turns a light bulb on and off on demand but also after the rising and setting of the sun. The mesh functionality makes it possible for the smartphone or tablet to only reach a single unit and this unit will pass on the signal to the entire network. This makes it possible to control lights in a very large area. At the time of writing there is no such product on the market and Luminaire aims to fill that position. This projects aims not only to build this product but also acquire the knowledge needed for other mesh applications in the future. Under the management of Adevo Consulting the project has taken seven months and has been successful in every aspect so far. As a proof of concept the latest prototype is a great success and something to build on in the future for a product that aims to go on the market. There are issues with the high voltage circuit on the which means that at the time of writing the circuit board cannot provide the 230 VAC that the light bulb requires without external components. The complete system can be seen in figure 0.1.</p> | | |
| <i>Keywords</i> | | |
| <i>Classification system and/or index terms (if any)</i> | | |
| <i>Supplementary bibliographical information</i> | | |
| <i>ISSN and key title</i> 0280-5316 | | <i>ISBN</i> |
| <i>Language</i> English | <i>Number of pages</i> 1-66 | <i>Recipient's notes</i> |
| <i>Security classification</i> | | |

<http://www.control.lth.se/publications/>