UV-C Sterilizer

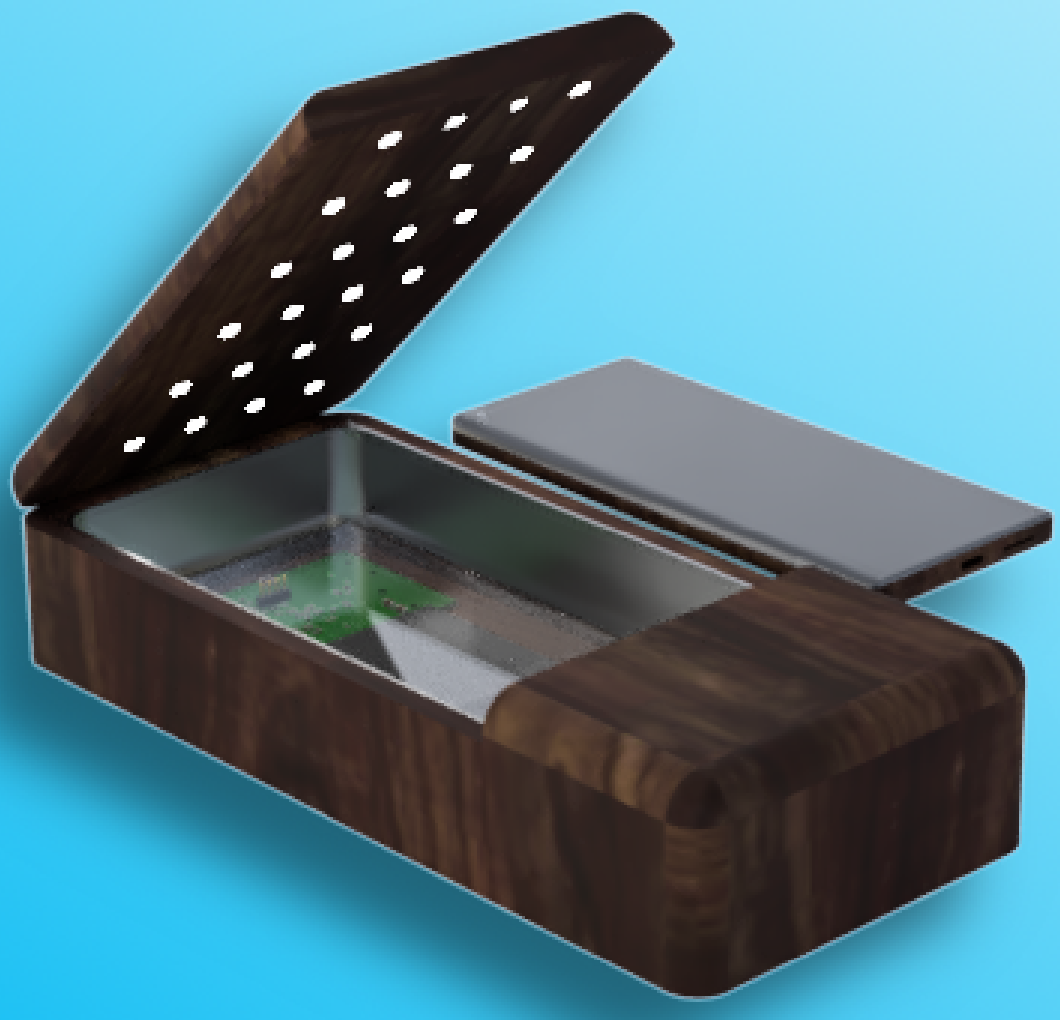
for the Wireless Powerlizer

C.K.O. De Jonghe
G.W. Lagerweij

Bachelor Thesis

June 29, 2021

Delft University of Technology



TU Delft

UV-C Sterilizer

for the Wireless Powerlizer

by

C.K.O. De Jonghe
G.W. Lagerweij

to obtain the degree of Bachelor of Science
at the Delft University of Technology.

Authors:	C.K.O. De Jonghe	4937023
	G.W. Lagerweij	4837703
Project duration:	April 19, 2021 – July 2, 2021	
Supervisors:	Dr. J. Dong,	
	G. Yu.	
Defense:	June 29, 2021 11:00 – 12:30	
Thesis committee:	Prof.dr. N. Llombart Juan,	TU Delft
	Dr. J. Dong,	TU Delft
	Dr. T.B. Soeiro.	TU Delft

TU Delft

Abstract

The effectiveness of ultraviolet (UV) irradiation for disinfection has been well-known for more than a century. Recent advances in light-emitting diode (LED) technology have brought UV LEDs to the consumer market. These offer many advantages over traditional gas-discharge lamps, allowing UV radiation to be generated with higher reliability in a much smaller form factor. In this thesis, the design of a portable battery-powered sterilizer based on UV LEDs is described. The sterilizer is intended for the sterilization of personal items such as mobile phones, face masks, and keys.

The design of the sterilizer consisted of two parts: the LED array and the LED driver. For their design, a model-based approach was taken to ensure high performance and low cost. For the LED array, a radiometric model was developed and applied in an optimization procedure. The design of the LED driver was based on an analytical loss model in MATLAB and LTspice simulations.

The UV LED array provides a radiant power of 113 mW to the disinfection area, resulting in an irradiation dose of 195 $\mu W/cm^2$. This allows for more than 99.9 % disinfection in less than 15 min. The LED driver is a boost converter operating at 750 kHz in the discontinuous conduction mode. This converter supplies the array with a nominal input power of 4.3 W at a simulated efficiency of 91 %.

The design of the LED driver is verified with a prototype. Measurements show efficiencies around 83 % at 4.1 W output power. Incorporating several improvements over the prototype, efficiencies between 85–87 % can be expected for the LED driver in the sterilizer.

Preface

To finalize their bachelor's degree, all Electrical Engineering students of the Delft University of Technology have to complete a final graduation project, officially called the EE3L11 Bachelor Graduation Project Electrical Engineering.

The goal of our specific project is to design the Wireless Powerlizer, which is a combination of a power bank, a wireless charging unit, and a UV-C sterilizer. Each of these three subjects was worked on by a different subgroup. This thesis will focus on the UV-C Sterilizer.

The Sterilizer will be able to disinfect common personal items like phones, keys, and face masks. This way, it can help prevent the transmission of many different pathogens, like for example SARS-CoV-2. To design this Sterilizer, we had to use our technical knowledge to create a UV-C LED array and LED driver. We also had to use our creativity and problem-solving skills to overcome all challenges we faced the past few months.

We want to thank our supervisors Dr. Jianning Dong and Guangyao Yu. We also want to thank Dr. Thiago Batista Soeiro, Calvin Riekerk, and Wenli Shi. They were always ready to answer any questions we had and to provide the necessary guidance to successfully finish this project.

We also want to express our gratitude towards Dr. ing. Bart Roodenburg and the rest of the DC Systems, Energy Conversion & Storage group, for providing us with materials and measurement equipment, and for giving us advice on building and testing our prototype.

Of course, we also want to thank our group members, Jeroen van Ammers, Hsukang Chen, Brecht Hurkmans, and Floris van der Kolk, without whom we could have never completed this project.

And finally, we want to thank our friends and families for supporting us through the past three years of our studies.

Charlotte De Jonghe
Gijs Lagerweij
Delft, June 2021

Contents

Abstract i

Preface ii

1 Introduction 1
1.1 Project Objective . . . 1
1.2 State-of-the-art Analysis . . . 2
1.3 Thesis Outline . . . 2

2 Program of Requirements 4
2.1 System Requirements . . . 4
2.2 Sterilizer Requirements . . . 5
2.3 Key Performance Indicators . . . 5

3 Ultraviolet Germicidal Irradiation 6
3.1 Ultraviolet Radiation . . . 6
3.2 Inactivation Mechanisms . . . 7
3.3 Disinfection Dose . . . 7
3.4 Safety . . . 9

4 UV LED Array 10
4.1 Radiometry . . . 10
4.2 UV LED Characteristics . . . 11
4.3 LED Selection . . . 12
4.4 LED Placement . . . 12
4.5 Optimization . . . 14
4.6 Thermal Performance . . . 16
4.7 Conclusion . . . 17

5 LED Driver 18
5.1 LED Configuration . . . 18
5.2 Topology . . . 19
5.3 Modeling . . . 19
5.4 Control . . . 23
5.5 Design. . . . 23
5.6 Simulation. . . . 31
5.7 Thermal Performance . . . 32
5.8 Conclusion . . . 33

6 Controller 34
6.1 Microcontroller Selection . . . 34
6.2 Functional Design . . . 35
6.3 Conclusion . . . 36

7 Prototype 37
7.1 Scope . . . 37
7.2 Design. . . . 38
7.3 Verification . . . 38
7.4 Conclusion . . . 42

8 Conclusion and Recommendations 43
8.1 Conclusion . . . 43
8.2 Recommendations for Future Work . . . 43

A Definitions and Abbreviations **45**
A.1 List of Physical Constants . . . 45
A.2 List of Symbols . . . 45
A.3 List of Abbreviations . . . 46

B Derivations **47**
B.1 LED Array Irradiance Reflections . . . 47
B.2 Boost Converter . . . 48
B.3 Prototype Measurement Uncertainty . . . 50

C Figures **51**
C.1 LED Array . . . 51
C.2 LED Configuration . . . 51
C.3 LED Driver . . . 52

D Schematics **54**

E Test Procedure **57**
E.1 Functional . . . 57
E.2 Performance . . . 58

F Listings **59**
F.1 LED Array Simulation . . . 59
F.2 LED Driver Model MATLAB . . . 70
F.3 LED Driver Model LTspice . . . 76

Bibliography **78**

1 Introduction

Due to the current COVID-19 pandemic, there has been an increasing interest in personal hygiene and demand for disinfection methods and devices. People started to wash their hands more and disinfect commonly touched items (such as mobile phones, laptops, etc.). While UV-C disinfection has been around for quite some time, it is a new and upcoming method to easily inactivate pathogens on small and portable devices.

UV-C radiation can inactivate bacteria, viruses, and other pathogens by damaging their genetic material. When the UV sources (e.g. LEDs) are arranged in an array, they can disinfect larger surfaces in a short amount of time. This LED array needs to be powered using an LED driver, usually consisting of a DC/DC converter to regulate the voltage and current provided to the array and a control system to manage the converter. Some of the challenges for building a portable UV-C disinfection device are caused by the low efficiency of UV-C LEDs.

1.1. Project Objective

The aim of this project is to develop a power bank that can charge a mobile phone wirelessly and sterilize personal belongings such as keys, mobile phones, or face masks using UV-C light. The final product will be a portable box called the Wireless Powerlizer which integrates a power bank and its management circuit, a wireless charging coil and its control circuit, and a UV-C generator for sterilization. The system-level architecture is presented in Figure 1.1.

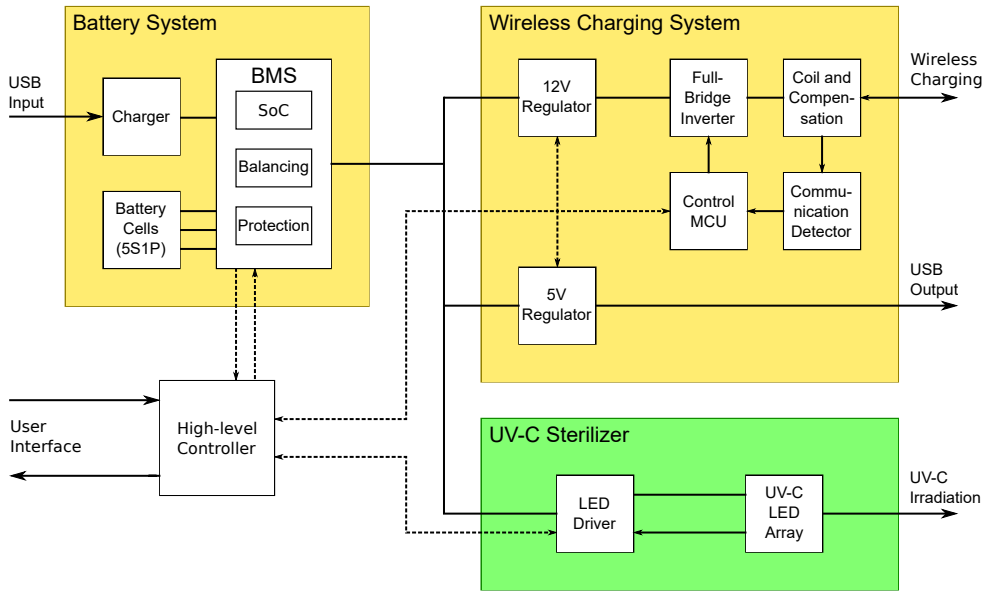
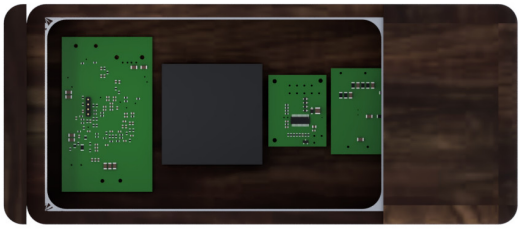


Figure 1.1: System-level Architecture

In Figure 1.2, a 3D model of the Wireless Powerlizer is presented. This model was designed as part of the prototype. Figure 1.2a shows the Powerlizer with an opened lid and a phone next to it for size comparison. In Figure 1.2b, the bottom of the disinfection area was removed to show the prototype PCBs and wireless charging coil. The battery is placed in the compartment on the right.

(a) Side View



(b) Top View; The bottom of the disinfection area is removed to show the internal components.

Figure 1.2: 3D Model of the Wireless Powerlizer

The focus of this thesis will be the design and verification of the UV-C sterilizer subsystem, while the battery system and wireless charging system are the subjects of two other theses [1, 2]. In this thesis, an LED array and LED driver will be designed while trying to obtain the highest efficiency and smallest size possible. The functionality of the high-level controller pertaining to the UV-C Sterilizer will also be designed.

1.2. State-of-the-art Analysis

Ultraviolet (UV) irradiation has been known to be effective against microorganisms for more than 140 years. It was first discovered in 1878 by Arthur Downes and Thomas P. Blunt [3]. Since then, UV irradiation has been used effectively to disinfect waste- and drinking water, sealed rooms, forced air systems, and biosafety cabinets [4].

Historically, UV radiation was generated using a variety of methods, most notably: mercury-based lamps operating at low vapor pressure (253.7 nm) and pulsed xenon lamps (230 nm) [4]. More recently, light-emitting diodes (LEDs) operating in the UV spectrum (between 255–280 nm) have been developed as an alternative to the gas-discharge lamps. The UV LEDs offer many benefits over the traditional lamps: they are mercury-free and offer higher efficiency, longer lifetime, smaller footprint, and more constant light intensity. Additionally, the frequency of the emitted UV radiation can be carefully controlled [5, 6].

Besides the advances in UV disinfection technology, personal sterilization has become more important than ever since the COVID-19 pandemic. There is a high demand for compact, safe, and efficient sterilization equipment. UV-C LEDs have proven to be an essential technology for realizing this equipment in both medical and consumer applications [7, 8].

While the uses and effectiveness of UV-C irradiation for disinfection have been well-described in literature, this mostly pertains to large installations. Portable disinfection devices based on UV-C LEDs are a relatively new phenomenon, and the design challenges are not well-known. In this thesis, the design process and challenges of such a portable device will be described in detail.

1.3. Thesis Outline

In Chapter 2, the requirements of the Wireless Powerlizer and the UV-C Sterilizer subsystem are presented. The theory behind ultraviolet germicidal irradiation (UVGI) is discussed in Chapter 3. The required irradiation dose is determined based on a literature review of common microbes on mobile phones.

In Chapter 4, the design of the LED array is presented. The design is based on a radiometric model of the irradiance in the disinfection area. An optimization procedure is developed to find the best LED

placement. In Chapter 5, the design of the LED driver is discussed. First, a suitable DC/DC converter topology is selected and a power loss model is developed. This loss model is then used to support the design decisions. The high-level controller for fault monitoring and the user interface is designed in Chapter 6.

A prototype is developed in Chapter 7 to test and verify the design of the Sterilizer subsystem. The test procedure and measurement results are presented and evaluated. Finally, the results of this thesis are discussed in Chapter 8. Conclusions are drawn from the discussion, and recommendations for future work are given.

In Appendix A, a list of symbols and abbreviations used throughout the thesis is included.

2

Program of Requirements

In this chapter, the requirements of the Wireless Powerlizer system and the UV-C Sterilizer subsystem are discussed.

2.1. System Requirements

The requirements for the entire Wireless Powerlizer system are derived from the project objective (see Section 1.1), and can be divided into several categories. The functional requirements specify what functionality the system must provide.

[**SF.1**] The Powerlizer shall be able to provide two full (0–100 %) wireless phone charges to a phone with average battery capacity (roughly 3500 mAh). This will determine the minimum required battery capacity, as well as the wireless charging time.

[**SF.2**] The Powerlizer shall have a USB-C port for charging the internal battery. An official USB protocol shall be supported [1].

[**SF.3**] The Powerlizer shall be able to disinfect an area of 10 cm × 17.5 cm. These dimensions are chosen such that most modern smartphones will fit in the disinfection area. Modern smartphone dimensions typically range between 65–80 mm width, 145–170 mm height, and 7.5–10 mm thickness.

[**SF.3.1**] The disinfection should inactivate the most common microbes found on mobile phones.

[**SF.3.2**] The disinfection should inactivate at least 99 % and preferably even 99.9 % of these microbes. This corresponds to a 2 or 3 log reduction, respectively.

[**SF.4**] The state of charge of the power bank shall be displayed on an LED indicator. These LEDs may also be used for other user interface purposes (such as displaying disinfection status, wireless charging status, etc.)

Taking into account the portable nature of the Wireless Powerlizer, the allowable operating conditions, aspects of the enclosure, and product dimensions must be considered.

[**SO.1**] The Powerlizer shall be able to operate in ambient temperatures between 0–45 °C. The allowable temperature range is limited by the operating temperature of the battery cells [1].

[**SO.2**] The Powerlizer enclosure shall be

[**SO.2.1**] Capable of withstanding ambient temperatures up to 60 °C.

[**SO.2.2**] Splash resistant.

[**SO.3**] The Powerlizer enclosure shall be at most 12.5 cm × 27.5 cm × 6 cm.

Safety is another important aspect since the Powerlizer will be used in close proximity to humans.

[**SS.1**] The Powerlizer shall follow the relevant IEC, ISO, and national safety standards. The wireless charging system must comply with EMI and electromagnetic field exposure regulations.

[**SS.2**] The disinfection area shall not be accessible while the disinfection LEDs are turned on. If the area is opened, disinfection must stop as soon as possible.

[**SS.3**] The voltages used in the Powerlizer shall not exceed 42 V [9].

2.2. Sterilizer Requirements

Similarly, functional requirements can be derived for the Sterilizer subsystem based on the project objective and specifications of the other subsystems.

[**F.1**] The Sterilizer shall be able to inactivate of the species of bacteria and viruses most commonly found on mobile phones and face masks. The inactivation shall be at least 99 %, but preferably 99.9 % or higher.

[**F.2**] The disinfection time shall be less than 15 min.

[**F.3**] The disinfection area shall be able to accommodate mobile phones, face masks, and keys. The minimum dimensions are 10 cm × 17.5 cm × 3 cm.

[**F.4**] The input voltage range of the Sterilizer shall be at least 12.5–21 V. This is defined by the battery management system [1].

[**F.5**] The Sterilizer shall be compatible with the large high-frequency magnetic fields generated by the wireless charging system [2].

Since the Sterilizer uses UV-C radiation, the device must be safe to use. This yields the following safety requirements:

[**S.1**] No UV radiation may escape from the disinfection area. If the disinfection area is opened during disinfection, the UV source should be shut off within 10 µs. The exposure dose limit during a workday (8 h) is 3 mJ/cm^2 [10].

[**S.2**] The device shall be designed to comply with relevant safety standards. This will be discussed in more detail in Section 3.4.

Finally, there are some specifications which will require a trade-off to be made.

[**T.1**] The irradiation of the disinfection area should be as uniform as possible.

[**T.2**] The UV LEDs and electronics should generate as little heat as possible to avoid needing a heat sink or fan for cooling.

[**T.3**] The switching frequency of any switching converters should be higher than 200 kHz to minimize electromagnetic noise radiated by the wireless charger in the communication band (100–200 kHz) [2].

[**T.4**] Either the top side or both sides of the disinfection area should be exposed to UV-C irradiation. The bottom side may not be accessible due to the wireless charging system.

2.3. Key Performance Indicators

The most important aspects of these requirements can be condensed into the key performance indicators (KPI).

[**KPI.1**] Dose of UV delivered to the disinfection area.

[**KPI.2**] Efficiency and size of the LED driver.

3

Ultraviolet Germicidal Irradiation

Many methods exist to inactivate microbes, with varying effectiveness and required inactivation time. Darnell et al. [11] compare several methods to inactivate coronaviruses (specifically SARS-CoV-1): ultraviolet (UV) light at 254 nm, heat treatment of 65 °C or greater, alkaline ($\mathrm{pH} > 12$) or acidic ($\mathrm{pH} < 3$) conditions, formalin and glutaraldehyde treatments. This study shows that UV irradiation (and UV-C in particular) is the most effective method to inactivate coronaviruses. It is also the fastest method, typically requiring less than 15 min. The same result is found in many other articles concerning the inactivation of microbes.

In this chapter the different aspects of using ultraviolet (UV) radiation to inactivate microbes are discussed. First, UV radiation is defined, and its properties concerning germicidal irradiation are presented. Second, the mechanisms by which UV radiation inactivates microbes are discussed. Finally, the safety risks and regulations pertaining to UV radiation, as defined by various articles and IEC/ISO standards, are explained.

3.1. Ultraviolet Radiation

Ultraviolet (UV) radiation is the part of the electromagnetic spectrum with wavelengths between 10–400 nm. The ISO 21348 standard [12] subdivides this range into several smaller ranges, which are presented in Table 3.1.

Table 3.1: Ultraviolet Spectrum [12]

Type	Wavelength
Vacuum UV	10–200 nm
UV-C	100–280 nm
UV-B	280–315 nm
UV-A	315–400 nm

UV-C and UV-B radiation are most frequently used for the inactivation of microbes and germs. Wavelengths in this range are particularly damaging to cells because they are absorbed by proteins, RNA, and DNA. When UV radiation is used for the inactivation of microorganisms, i.e., to destroy their ability to reproduce by causing damage to their genetic material, it is also called Ultraviolet Germicidal Irradiation (UVGI). [4]

UV-A radiation is the radiation typically received from the sun and is far less damaging than UV-C or UV-B. From UV-B to vacuum UV, the absorption by the atmosphere increases substantially.

The radiation dose D [J/m^2] is related to the irradiance E_e [W/m^2] and the radiant flux Φ_e [W] as shown in (3.1).

$$D = E_e \cdot t = \frac{\Phi_e}{A} \cdot t \tag{3.1}$$

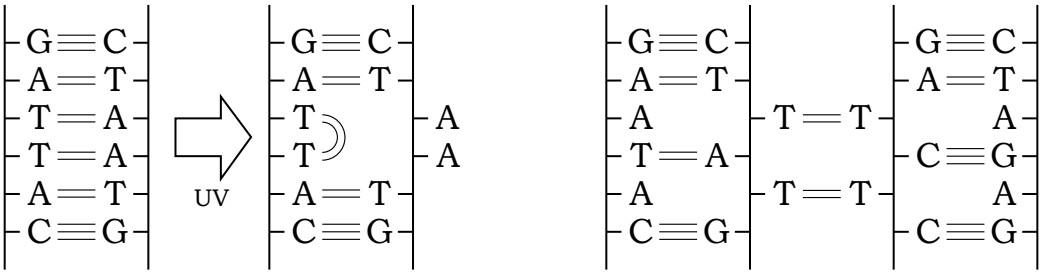
Here, A [m^2] is the surface area, and t [s] is the exposure time.

3.2. Inactivation Mechanisms

The inactivation mechanisms of UVGI are related to the effect UV radiation has on the genetic material of the microbes. This genetic material is either ribonucleic acid (RNA) or deoxyribonucleic acid (DNA) in viruses or both in bacteria. DNA and RNA are large macromolecules consisting of nucleotides. Each nucleotide has three parts: (deoxy)ribose, phosphate, and one of four bases: thymine (T), adenine (A), cytosine (C), or guanine (G). In RNA, thymine is replaced by uracil (U).

Dimerization One of the two main inactivation mechanisms is dimerization. When thymine/uracil bases are next to each other in the DNA or RNA strands this pair is called a doublet. The dimerization of doublets by UV radiation can lead to the inactivation of the DNA or RNA, with the result that the cell may be unable to reproduce effectively. Secondary damage is caused by the dimerization of cytosine [4]. An example of thymine dimerization is presented in Figure 3.1a.

Cross-linking The second mechanism is cross-linking, in which two non-adjacent thymine/uracil bases bond together. Cross-linking may also cause the inactivation of a part of the DNA or RNA and can occur between the nucleotides and proteins in the capsid of viruses [4]. An example of thymine cross-linking is presented in Figure 3.1b.



(a) Dimerization (adapted from Kowalski [4])

(b) Cross-linking

Figure 3.1: Inactivation Mechanisms

Optimal Wavelength Because the optimal wavelength for absorption by DNA/RNA (and particularly the thymine nucleotides) is around 260–280 nm, UV-C is far more effective than UV-B for disinfection purposes: the effectiveness of UV-B is roughly 15 % that of UV-C [4, 13].

Environmental Parameters The disinfection environment has some impact on the effectiveness of the disinfection. Increased relative humidity decreases the decay rate for bacteria but is barely noticeable for viruses. Near 100 %RH, the log reduction doses for bacteria tend towards those in water. The ambient temperature has little effect on the effectiveness of UV germicidal irradiation, but it may cause heat damage or freezing. [4]

3.3. Disinfection Dose

The number of microbes inactivated during disinfection is usually expressed on a logarithmic scale: the log reduction can be calculated as shown in (3.2).

$$\text{log reduction} = \log_{10} \frac{N_0}{N} \tag{3.2}$$

Where N_0 and N are the number of microbes before and after disinfection, respectively. For example, a 3 log reduction corresponds to 99.9 % inactivation.

The log reduction dose D_{90} [mJ/cm^2] indicates the irradiation dose required to obtain 1 log reduction. It may vary with orders of magnitude between different microbes and different media (i.e., on a surface, in a fluid, in air).

3.3.1. Common Bacteria

Following from requirement [**F.1**], the Sterilizer must be able to provide at least a 2 log reduction of the most common microbes. The bacteria most commonly found on mobile phones and the corresponding log reduction doses are presented in Table 3.2. Much research has been done on the transmission of bacteria via mobile phones because of their ubiquity in recent years [14–18].

The log reduction doses are found from relevant papers and books [4, 19]. Where possible, the value for bacteria on a surface was taken, as this is the most representative for the current application. If this is not available, the value for air or water is taken. In case multiple values from different experiments are listed, the highest value is given.

Table 3.2: Bacteria Commonly Found on Mobile Phones [4, 14–19]

Species	Rel. Freq.	Pathogenicity	D_{90} [mJ/cm^2]
Staphylococcus epidermidis	Very common	Commensal	2.9[b]
Staphylococcus aureus	Common	OP[a]	6.6
Micrococcus spp.	Common	Commensal, OP	10.0
Escherichia coli	Less common	Some species	5.5
Enterococcus faecalis	Less common	OP	9.0
Klebsiella pneumoniae	Less common	OP	6.8[c]
Actinetobacter spp.	Less common	OP	1.8
Bacillus spp.	Rare	Some species	74.3
Pseudomonas aeruginosa	Rare	OP	5.5

[a] Opportunistic pathogen
[b] In air
[c] In water

Almost all of these bacteria are pathogenic in some capacity. The risk of an infection depends on the strength of the immune system of an individual.

3.3.2. Common Viruses

Next, the epidemic viruses that can be transmitted via mobile phones are discussed [20–22] and their log reduction doses are presented in Table 3.3.

The log reduction values were obtained from relevant papers and books [4, 23]. Since coronaviruses do not differ much structurally, SARS-CoV-1 and SARS-CoV-2 will be assumed to have the same log reduction dose.

Table 3.3: Epidemic Viruses Transmitted via Mobile Phones [4, 20–23]

Species	Rel. Freq.	D_{90} [mJ/cm^2]
Respiratory Syncytial Virus	Very common	20.0
SARS-CoV-2	Very common	10.6
Influenza A	Very common	1.9[a]
Influenza B	Common	1.9[a]
Measles	Common	2.2[b]
Influenza C	Less common	1.9[a]
Rotavirus	Less common	20.0[b]
Human Metapneumovirus	Less common	20.0[c]
SARS-CoV-1	Rare	10.6

[a] In air
[b] In water
[c] Because of structural similarity to RSV (hOPV)

There are four types of influenza viruses; Influenza A, B, C, and D. Influenza A causes most seasonal epidemics and it has the potential to cause pandemics. Influenza B only sometimes causes seasonal epidemics. Milder illnesses are caused by Influenza C, while Influenza D does not appear to infect humans. [21]

All viruses in Table 3.3 are pathogenic, the severity of an infection depends on the type of virus and the immune system of the infected person. Generally, the Respiratory Syncytial Virus (RSV) causes common colds, with only minor symptoms. Influenza A and B cause more severe flu infections, while Influenza C causes a milder illness.

Most other viruses like herpes, hepatitis, HIV, etc. can not be transmitted through surface contamination, and will thus not be considered for this application.

3.3.3. Conclusion

As indicated in requirement [**F.1**], the Sterilizer should inactivate at least 99 % of common microorganisms, corresponding to a 2 log reduction. From Tables 3.2 and 3.3 it is found that an irradiation dose of at least 148.6 mJ/cm^2 is needed (which is two times the D_{90} of *Bacillus* spp.). Since all other log reduction doses are much lower, this will guarantee a more than 99.9 % reduction of all other bacteria and viruses listed in the tables.

A possibility is to allow the user of the Sterilizer to choose how strong they want the disinfection to be. The user would have the option to choose the maximal dose of 150 mJ/cm^2 or they could opt for a lower dose of for example around 40 mJ/cm^2, which would be sufficient for all other listed microorganisms, to limit the energy drawn from the battery for a single disinfection cycle and to allow a shorter disinfection time.

3.4. Safety

Since this product is intended for the consumer market, extra attention must be paid to the safety aspects of using UV disinfection. In this section, the risks and regulations regarding UV disinfection are discussed.

3.4.1. Risks

UV radiation is known to cause a multitude of skin conditions and diseases including cancer, inflammation, and others. UV-C radiation has very high energy and a short wavelength, which makes it unable to penetrate deep into the skin layers. Still, UV-C exposure can cause eye and skin damage, and in the long-term even premature aging and skin cancer. [24]

Therefore, it is very important to make sure that the user of the Sterilizer is not exposed to UV-C. Some kind of protection is thus needed to ensure that whenever the user opens the Sterilizer, the disinfection process is stopped and no UV-C will radiate out.

3.4.2. Regulations

There are multiple standards and regulations concerning UV-C disinfection. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) have created regulations that products need to adhere to. The minimum safety requirements of technological equipment and UV-C lamps can be found in the IEC 62368 [9] and ISO 15858 [25] standards. Additionally, the IEC 62471 [26] standard defines several regulations for the photobiological safety of lamp systems.

If the Sterilizer would be used in a medical setting to, for example, disinfect healthcare worker's phones, the ISO 11137 [27] standard may also apply. Because of restricted availability, these standards have not been investigated in detail. When the Sterilizer is to be introduced on the market, the requirements following from these standards need to be incorporated.

4

UV LED Array

In this chapter, the modeling and design of the UV-C LED array are discussed. This array will illuminate the disinfection area with UV-C radiation. It must be capable of delivering the desired dose (150 mJ/cm^2, as derived in Chapter 3) in at most 15 min.

First, basic radiometry and UV LED characteristics are discussed. Second, different UV-C LEDs are compared, and a selection is made for this design. Finally, the LED placement is optimized based on a radiometric model of the LED irradiance on the disinfection surface.

4.1. Radiometry

Radiometry is a set of techniques for characterizing the distribution of power in space. To calculate the power delivered to the disinfection area and determine the required LED characteristics, some radiometric theory will be used [28, 29].

The two radiometric quantities that will be used in this chapter are the radiant intensity I_e [W/sr] and the irradiance E_e [W/m^2]. Their formal definitions are given in (4.1) and (4.2)

$$I_e = \frac{\partial \Phi_e}{\partial \Omega} \tag{4.1}$$

$$E_e = \frac{\partial \Phi_e}{\partial A} \tag{4.2}$$

Here, Φ_e [W] is the radiant flux, Ω [sr] is the solid angle, and A [m^2] is the surface area. Combining these two equations, the irradiance can be written in terms of the radiant intensity. This formulation is useful, because E_e can be determined from the radiant intensity of the source.

$$E_e = I_e \frac{\partial \Omega}{\partial A} = I_e \frac{\cos\theta}{r^2} \tag{4.3}$$

Equation (4.4) is obtained by writing the radiant intensity of the source in terms of its radiation pattern $I_{rel}(\theta, \phi)$, total radiant flux Φ_0, and the solid angle Ω.

$$I_e(\theta, \phi) = I_{e,0} I_{rel}(\theta, \phi) = \frac{\Phi_0 I_{rel}(\theta, \phi)}{\int I_{rel}(\theta, \phi) d\Omega} \tag{4.4}$$

Combining (4.3) and (4.4) gives an expression for the irradiance in terms of the source properties.

$$E_e = \frac{\Phi_0 I_{rel}(\theta, \phi)}{r^2 \int I_{rel}(\theta, \phi) d\Omega} \cos(\theta) \tag{4.5}$$

The formula in (4.5) will be used to calculate the irradiance on the disinfection area due to a UV LED. This can be extended to multiple LEDs by applying the superposition principle.

4.2. UV LED Characteristics

UV-C LEDs are increasingly being used for air, surface, and water disinfection instead of traditional mercury-vapor-based lamps. They offer clear benefits over the traditional UV lamps: The LEDs are mercury-free, have a compact footprint, and allow for instant on/off operation and unlimited cycling [5, 6]. The compact footprint makes UV-C LEDs ideal for portable applications.

Electrical Characteristics UV-C LEDs behave similarly to normal LEDs or diodes, with an exponential current-voltage relationship. Typically, the forward voltage of a UV-C LED (6.0–8.0 V) is high compared to normal color LEDs (2.0–3.5 V). The forward voltage of an LED is proportional to the frequency of the emitted light and can be calculated using (4.6) [30]. Usually, the forward voltage is somewhat higher than this due to voltage drops over contacts and heterobarriers. For UV LEDs this is a significant contributor to the forward voltage due to their quantum well heterostructure [31].

$$V_F = \frac{E_g}{q} = \frac{hc}{\lambda q} \tag{4.6}$$

UV LEDs are not made to be operated in reverse bias, and doing so may destroy them in some cases.

Efficiency The efficiency of UV LEDs is determined by several factors: the internal quantum efficiency (IQE), which is the fraction of the diode current that produces luminescence; the light extraction efficiency (LEE), which is the ratio of the number of emitted photons to the number of generated photons; and the electrical efficiency (EE), which describes the excess energy applied to the device to overcome voltage drops across contacts and heterobarriers. The external quantum efficiency (EQE) and wall plug efficiency (WPE) are given by (4.7) and (4.8). [32]

$$\mathrm{EQE} = \frac{N_{photons}}{N_{electrons}} = \frac{P_{out}}{\hbar\omega}\frac{q}{I} = \mathrm{IQE} \times \mathrm{LEE} \tag{4.7}$$

$$\mathrm{WPE} = \frac{P_{out}}{P_{in}} = \frac{P_{out}}{V \cdot I} = \mathrm{EQE} \times \mathrm{EE} \tag{4.8}$$

Typical efficiencies up to 5 % may be achieved with Gen III devices [32]. This number can be improved by using new technologies, potentially reaching more than 20 % [33].

Radiation Pattern The most important optical parameters of an LED are its radiation pattern $I(\theta, \phi)$, the corresponding beam width Θ [°], and the radiant flux Φ_0 [W]. The beam width is the angle over which the relative radiant intensity is larger than 50 %, as illustrated in Figure 4.1 for a Lambertian emitter ($I_{rel}(\theta, \phi) = \cos\theta$).

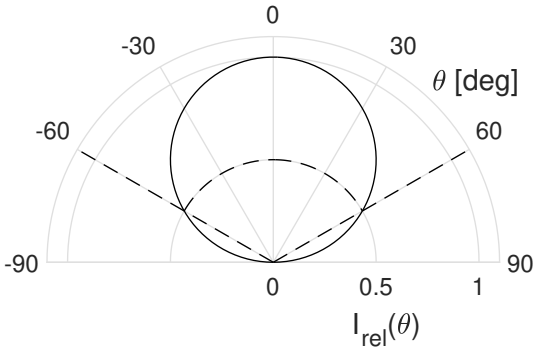


Figure 4.1: Radiation Pattern of a Lambertian Emitter

Most UV-C LEDs with square windows behave like Lambertian emitters. Their radiant intensity does not vary along the ϕ axis and they have a wide beam width around 120°. See Table 4.1 in the next paragraph.

4.3. LED Selection

Based on the characteristics discussed above, a set of UV-C LEDs available on the market are compared in Table 4.1. These LEDs are intended for UVGI applications and have a wavelength in the range of optimum absorption (260–280 nm, see Section 3.2). The most differentiating characteristics are the output power Φ_e and the wall plug efficiency WPE.

Table 4.1: UV-C LED Comparison Table

Type	Φ_e [mW]	Θ [°]	λ [nm]	P [mW]	V_F [V]	WPE [%]	Price [€/pc]
QBHP684E-UV265	2.5	120	265	260	6.0	0.96	11.42
QBHP684E-UV265N	10	120	265	1000	6.0	1.00	22.44
SU CULBN1.VC	4.7	120	275	180	6.0	2.61	6.69
SU CULDN1.VC	42	120	275	1995	5.7	2.11	41.79
UVR270-SC12	3	120	270	400	8.0	0.75	7.60
UVR270-4F12	48	120	270	5760	32.0	0.83	49.10
VLMU35CB20-275-120	10	120	275	710	7.1	1.41	6.72
VLMU35CL20-275-120	2.5	120	275	136	6.8	1.84	4.65
VLMU35CT20-275-120	19	120	275	1080	7.2	1.76	10.98

Since this application is intended to be portable, efficiency is an important aspect to ensure sufficient run-time on a single battery charge. Hence, an LED with a high efficiency should be chosen. To ensure that the entire disinfection area is uniformly illuminated, it is likely not beneficial to have few LEDs with large Φ_e. This may concentrate the irradiance in a small area under the LED. Hence, the Φ_e of the LED should not be too large. Due to the low efficiency, a high Φ_e will also mean high losses in a single LED. This is undesirable since the heat dissipation should be kept as low as possible (trade-off [**T.2**]).

Based on these two criteria, the SU CULBN1.VC LED was chosen. From its parameters, it can be derived that $\mathrm{EQE} = 3.47\,\%$ and $\mathrm{EE} = 75.2\,\%$. The radiation pattern of the LED was extracted from the datasheet of the manufacturer [34] and is shown in Figure 4.2.

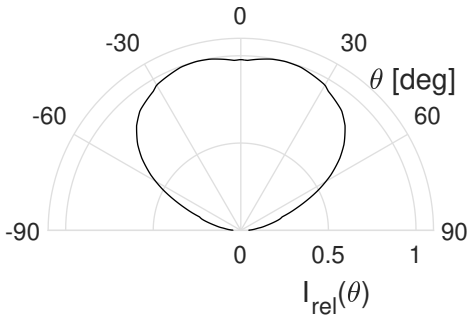


Figure 4.2: SU CULBN1.VC Radiation Pattern

4.4. LED Placement

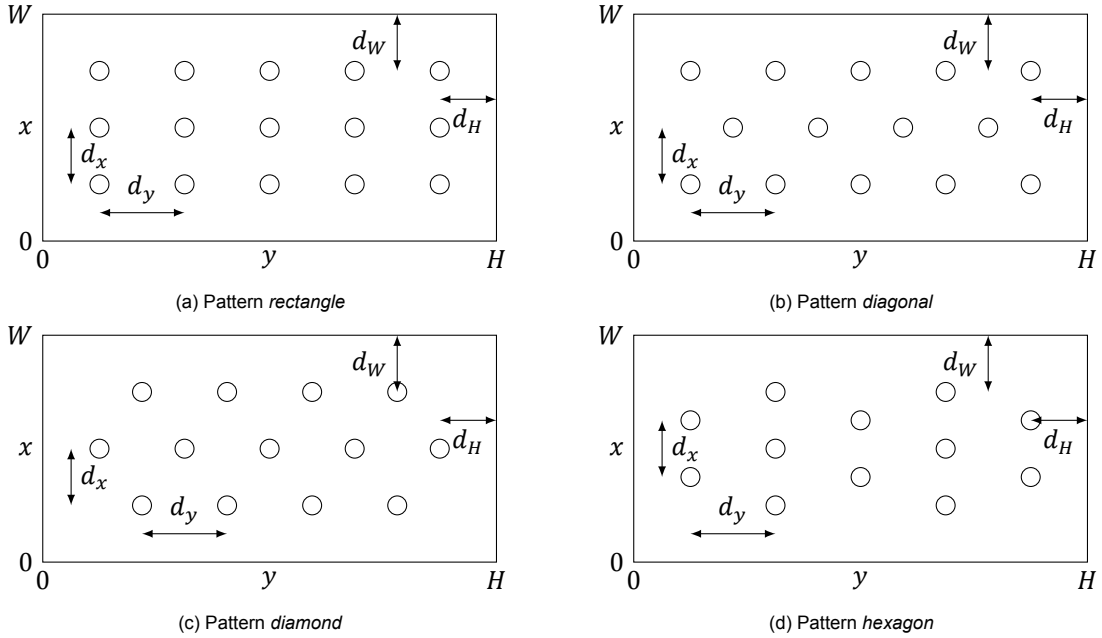
After the LED selection, the placement pattern needs to be determined. First, a decision needs to be made whether the Sterilizer should disinfect both sides of an object, or only the top side (trade-off [**T.4**]).

While it would be useful if the Sterilizer is able to disinfect both sides of an object, this is not practical since the wireless charger is located on the bottom side of the Powerlizer. The wireless charger requires the phone to be very close to the coil to achieve good coupling [2], while the Sterilizer requires some distance between the LEDs and the object to be disinfected. The LEDs will thus only be placed on the top side, in the lid, of the Powerlizer.

The distance between the LEDs and the object to disinfect is assumed to be 2 cm. This value is derived from the maximum thickness of a phone (1 cm) and the height of the disinfection area (3 cm).

4.4.1. LED Placement Patterns

Four LED placement patterns are compared: the *rectangle*, *diagonal*, *diamond*, and *hexagon* patterns. These patterns are easily adjustable, making them very suitable for an optimization simulation. They are presented in Figure 4.3. The LED locations are marked with a circle, and the variables d_W and d_H represent the distance of the pattern to the wall in the x and y directions respectively.



(a) Pattern *rectangle*

(b) Pattern *diagonal*

(c) Pattern *diamond*

(d) Pattern *hexagon*

Figure 4.3: LED Placement Patterns (with $N_{col} = 3$, $N_{row} = 5$)

4.4.2. Irradiance Calculation

To calculate the irradiance on the disinfection area, the area is divided into a number of grid elements (x, y, z). The irradiance can then be calculated by writing (4.5), the final result of Section 4.1, in matrix form and summing the irradiance obtained for each LED. The LEDs are numbered $n = 1, \ldots, N$, and each LED has a position $(x_{s,n}, y_{s,n}, z_{s,n})$.

$$\mathbf{E}_e = \frac{\Phi_e}{\Omega} \sum_{n=1}^{N} \frac{I_{rel}(\boldsymbol{\theta}_n)}{\mathbf{r}_n^{\circ 2}} \cos \boldsymbol{\theta}_n \tag{4.9}$$

Here $\mathbf{r}_n^{\circ\alpha}$ denotes the Hadamard power (r_{ij}^{α}) and N is the number of LEDs. The matrices $\mathbf{r}_n$ and $\boldsymbol{\theta}_n$ contain the distances and angles of each grid point to the LED with index n.

$$\mathbf{r}_n = \sqrt[\circ 2]{(\mathbf{x} - x_{s,n})^{\circ 2} + (\mathbf{y} - y_{s,n})^{\circ 2} + (\mathbf{z} - z_{s,n})^{\circ 2}} \tag{4.10}$$

$$\boldsymbol{\theta}_n = \arccos\left((\mathbf{z} - z_{s,n}) \oslash \mathbf{r}_n\right) \tag{4.11}$$

Here, $\mathbf{A} \oslash \mathbf{B}$ denotes the Hadamard division (A_{ij}/B_{ij}).

To judge the quality of a particular LED arrangement, two performance metrics have been defined which can be calculated from the irradiance $\mathbf{E}_e$: the overdose O [-] and the minimum irradiance E_{min} [mW/cm^2]. They are defined in (4.12) and (4.13).

$$O = \frac{\max \mathbf{E}_e}{\min \mathbf{E}_e} \tag{4.12}$$

$$E_{min} = \min \mathbf{E}_e \tag{4.13}$$

The MATLAB implementation of these equations is included in Appendix F.1.4.

4.4.3. Reflections

Since the LEDs are used inside a closed box, its walls will cause reflections of the UV-C radiation. These reflections also need to be taken into account, as they quite heavily influence the radiation at each point. Not taking these reflections into account when the sides of the box are made of a highly reflective material, will result in a lot of power being effectively lost in the walls, while this would be reflected in practice.

Table 4.2 shows the reflection coefficient Γ [%] of different materials. Even though e-PTFE has the highest reflectivity, its reflections are diffusive. Aluminum, on the other hand, creates specular reflections and has a high commercial availability [35]. Therefore, a UV-Enhanced Aluminum coating with a reflection coefficient of 90 % was chosen for the inner walls of the disinfection area [36].

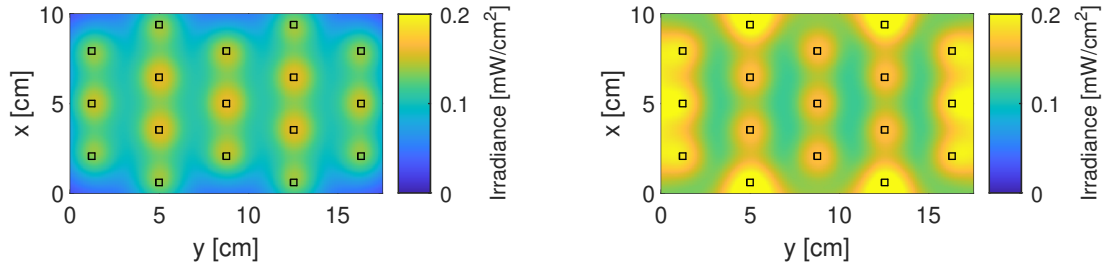
Table 4.2: Reflection Coefficients of Potential Wall Materials [35–37]

Material	Γ [%]
Expanded Polytetrafluoroethylene (e-PTFE)	95–99
UV-Enhanced Aluminum	90
Aluminum	88
Polished Sheet Aluminum	75
Stainless Steel	20–28

For indirect reflections, the distance traveled by the UV-C wave is much longer than for direct reflections, thus their contribution to the irradiation will be much lower. Since these indirect reflections make the simulation more complex and their influence on the irradiation pattern is rather low, only the direct reflections are simulated.

The angle of incidence θ and the distance traveled r_{tot} are derived in Appendix B.1. Using these, combined with (4.9), the irradiance due to a reflection can be found. This is added to the MATLAB implementation by means of matrix equations as presented in Appendix F.1.3.

In Figure 4.4, a simulation is shown for two different values of the wall reflectivity. This shows the importance of including the reflectivity of the walls for the obtained irradiance.



(a) Wall Reflectivity = 0 %

(b) Wall Reflectivity = 90 %

Figure 4.4: Irradiance Simulation of the *Hexagon* Pattern ($N_{col} = 4$, $N_{row} = 5$)

4.5. Optimization

In the previous section, a method to calculate the irradiance distribution of an arbitrary LED arrangement has been presented. In this section, an optimization procedure to find the best arrangement will be discussed.

The optimal arrangement depends on the optimization criteria and constraints. From the requirements, it can be derived that the ideal layout has minimal N_{led} (lower cost and power dissipation), minimal O (higher efficiency), and maximum E_{min} (lower disinfection time).

4.5.1. Pattern Optimization

The first step to optimizing the LED array is to find the optimal parameters for each possible pattern configuration (i.e., pattern type and number of rows and columns). The minimum irradiance E_{min} should be maximized to ensure that the sterilization time requirement [**F.2**] can be met. This can be formulated as a multi-variable optimization problem with variables $\boldsymbol{x}$, static inputs $\boldsymbol{p}$, and objective function $f(\boldsymbol{x}, \boldsymbol{p})$ [38]. Note that in this case, $f(\boldsymbol{x}, \boldsymbol{p})$ must be maximized.

$$\boldsymbol{x} = [d_W, d_H] \tag{4.14}$$

$$\boldsymbol{p} = [\text{pattern}, N_{row}, N_{col}, \Gamma] \tag{4.15}$$

$$f(\boldsymbol{x}, \boldsymbol{p}) = E_{min} = \min \mathbf{E}_e \tag{4.16}$$

This optimization procedure is implemented in MATLAB using the `fminsearch` function for multi-variable optimization. The code is included in Appendix F.1.5.

4.5.2. LED Array Optimization

As mentioned before, three objective functions need to be optimized: minimize N_{led}, minimize O, and maximize E_{min}. Because of the discrete nature of the optimization variables, the following optimization procedure is developed:

1. Generate feasible LED placements (pattern, N_{row}, N_{col}).
2. Minimize E_{min} for each placement as shown in Section 4.5.1.
3. Calculate N_{led} and O for each solution.
4. Choose the solution with the optimal combination $[N_{led}, E_{min}, O]$.

The last step may require a trade-off to be made between the different objectives.

The procedure is applied to the optimization of the LED array placement using the following allowed layout parameters. The number of rows and columns was restricted to these values to avoid obtaining solutions with a very large or very small number of LEDs, since these will not satisfy the requirements anyway.

$$\begin{aligned} \text{pattern} &= \{\text{rectangle}, \text{hexagon}, \text{diagonal}, \text{diamond}\} \\ N_{col} &= \{3, 4, 5\} \\ N_{row} &= \{4, 5, 6\} \end{aligned}$$

Constraints can be placed on the minimum irradiance and overdose by requirements [**F.2**] and [**T.1**].

$$O \leq 1.75 \qquad E_{min} \geq \frac{150\,\text{mJ/cm}^2}{15\,\text{min}} = 167\,\mu\text{W/cm}^2$$

Discarding the results that do not meet these requirements, Figure 4.5 is obtained. Each dot represents a valid solution. Its color indicates the pattern that is used to generate that solution. The unconstrained results are presented in Figure C.1 in the Appendices.

From the figure it can be seen that the rectangle pattern performs the best in all objectives. Several of the best results are presented numerically in Table 4.3.

Table 4.3: Best Layout Configurations

Configuration			N_{led}	O	E_{min}
Pattern	N_{col}	N_{row}	[-]	[-]	[μW/cm^2]
Rectangle	4	6	24	1.25	195
Rectangle	5	6	30	1.20	246

A trade-off must be made between the number of LEDs and the achievable overdose O and minimum irradiance E_{min} to find the best LED placement. Since both configurations meet the minimum requirements, the configuration with the lowest number of LEDs is chosen to minimize cost and power dissipation: the rectangle pattern with $N_{col} = 4$ and $N_{row} = 6$.

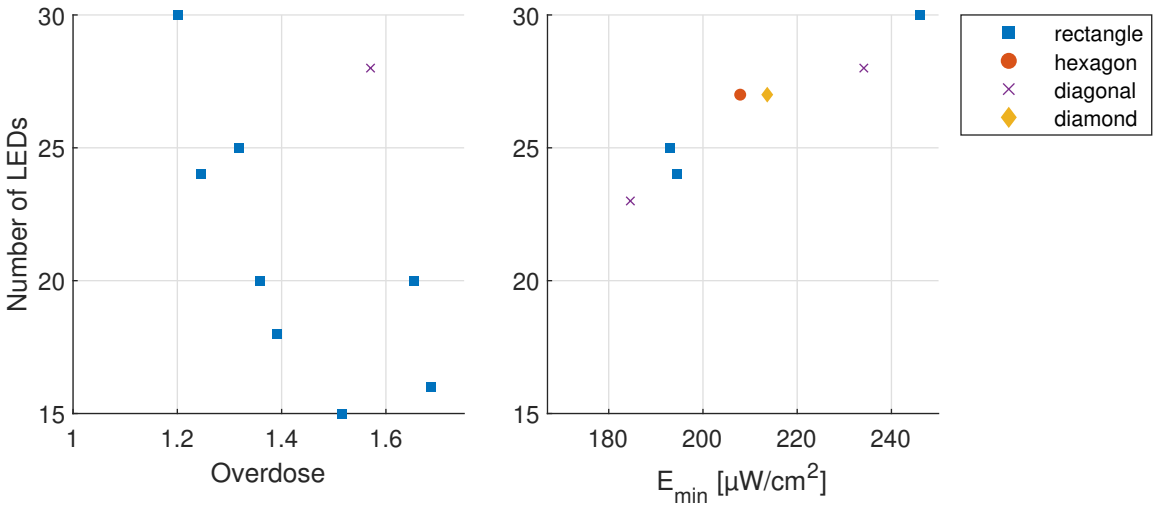


Figure 4.5: Layout Optimization Results (Constrained)

4.6. Thermal Performance

The absolute maximum operating temperature range of the selected SU CULBN1.VC LED is −40 to 60 °C, with a maximum junction temperature of 85 °C [34, 39].

From the specified power dissipation $P = 180\,\text{mW}$ and the thermal resistance from the junction to the solder point $R_{\theta JS} = 25\,\text{K/W}$, the temperature rise can be calculated using (4.17).

$$\Delta T_{JS} = R_{\theta JS} P \approx 4.5\,\text{K} \tag{4.17}$$

This temperature rise, however, is with respect to the PCB. If the heat energy cannot be extracted from the PCB, the PCB temperature will also rise, resulting in a higher junction temperature than expected based on this simple calculation. Additionally, because the LEDs will be quite closely spaced on the PCB, their thermal coupling might be significant. This may create an additional temperature rise at the LED junction.

A first approximation can be made by simulating the heat transfer through a simplified model of the Sterilizer (using FEMM [40] for example). The expected mechanical stack-up is given in Table 4.4. This can give an indication of the temperature rise to be expected, but cannot be taken as a final value. To be certain, thermal tests must be performed on a prototype board.

Table 4.4: Stack-up for Thermal Calculations

Material	k_z [W/(m K)]	c_p [J/(kg K)]	t [mm]	A [mm^2]
LED				13
Copper	401	380	0.035	17 500
FR4	0.3	1100	1.57	17 500
Copper	401	380	0.035	17 500
Plastic	0.35	1500	1.0	17 500

The convection coefficient h of the plastic surface can be taken to be $15\,\text{W/(m}^2\,\text{K)}$ for natural convection and an air velocity of 0 m/s [41]. The power dissipation of each LED can be calculated using the efficiency η of the selected LED and (4.18).

$$P_{LED} = (1-\eta)P = 97.4\,\% \cdot 180\,\text{mW} = 175.3\,\text{mW} \tag{4.18}$$

The thermal simulation is performed in FEMM 4.2 [40] with a Lua script to construct the geometry. The script is included in Appendix F.1.8. The simulation shows that a temperature rise around 22.5 K

is to be expected for the outer LED strings, and 23 K for the inner LED strings. Assuming a maximum operating temperature of 45 °C means that the junction temperature of the LEDs will be at most 68 °C.

This temperature rise can be attributed to two major contributors: The thermal resistance of the LED ($R_{\theta JS} = 25\,\text{K/W}$) and the resistance presented by convection to ambient, which can be calculated using (4.19).

$$R_{\theta PA} = \frac{1}{hA} \approx 3.8\,\text{K/W} \implies \Delta T_{PA} \approx 16\,\text{K} \tag{4.19}$$

The thermal simulation has shown that the junction temperature of the LEDs is below the specified maximum of 85 °C. Even if an error of 20 % was made due to the simplicity of the simulation, the junction temperature is still acceptable.

4.7. Conclusion

In this chapter, the modeling and design of the UV-C LED array of the sterilizer have been discussed. The SU CULBN1.VC LED from Osram Opto Semiconductors has been selected as the most suitable UV-C LED because of its relatively high efficiency.

Next, the irradiance on the disinfection area was simulated using radiometry theory, and the LED layout was optimized to find a solution with the minimum number of LEDs, maximum uniformity, and maximum irradiance. The result was a rectangular pattern with four columns and six rows ($d_W = 14\,\text{mm}$, $d_H = 13.5\,\text{mm}$). The irradiance of this configuration is presented in Figure 4.6.

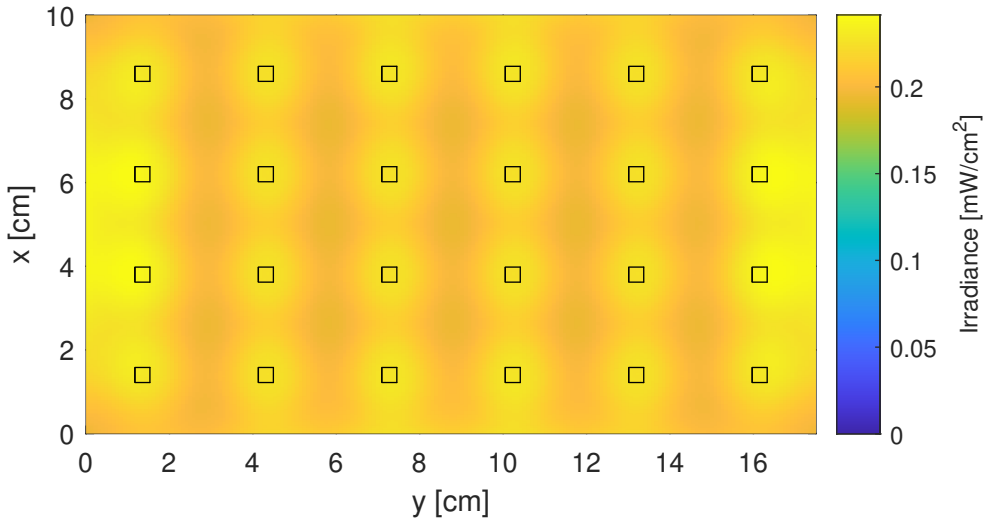


Figure 4.6: Irradiance Simulation of the Chosen LED Array Layout

Finally, a simple thermal simulation of the LED array was performed to ensure that the LEDs will not heat up too much. It was found that the LED temperature rise with respect to ambient was approximately 23 K. This resulted in a maximum junction temperature of 68 °C, which is lower than the specified maximum. When the product design and in particular its enclosure and internal layout are finished, a new thermal simulation should be performed (using e.g., COMSOL Multiphysics), as well as thermal measurements on a prototype board to ensure that the temperature rise is acceptable.

It can be concluded that the LED array satisfies functional requirements [**F.1**], [**F.2**], and [**F.3**]: It provides an irradiation dose of at least 195 µW/cm^2, allowing for 99 % disinfection in at most 13 min. Additionally, the configuration has high uniformity ($O = 1.25$), satisfying trade-off [**T.1**]. For trade-off [**T.4**], it was chosen to only disinfect the top side of an object in the disinfection area due to the wireless charging system on the bottom side.

5

LED Driver

In this chapter, the modeling and design of the LED driver are presented. The LED driver will supply the UV-C LED array (Chapter 4) with suitable voltage and current. It must be capable of delivering up to 4.7 W to the LEDs. The design of the LED driver will focus on making it small and efficient.

First, the electrical LED configuration is discussed. Then a suitable DC/DC converter topology is selected and modeled in MATLAB. The MATLAB model is used to design the converter. Finally, the converter is simulated in LTspice and the thermal performance is evaluated.

5.1. LED Configuration

In the previous chapter, the LED placement was determined. For the design of the LED driver, the electrical LED configuration (i.e., the number of parallel strings and LEDs per string) must be determined. This will have a large impact on the selection of the converter topology.

Since the Powerlizer has to be as compact as possible the converter should use a simple topology (such as buck or boost), instead of more complex topologies such as flyback, SEPIC, or Ćuk. The batteries used in the Powerlizer provide a voltage ranging from 12.5–21 V to the Sterilizer [1]. The maximum allowable voltage of the Sterilizer is 42 V, as stated in requirement [**SS.3**]. The forward voltage of a single LED is between 5.5–6.5 V, and the nominal forward current of each LED is 30 mA [34].

Figure C.2 shows all LED configurations possible with 24 LEDs. To avoid creating many parallel LED strings for balancing purposes, a boost converter is preferred to convert the input voltage of the LED driver up. There are two possible configurations when using a boost converter, 4 parallel strings of 6 LEDs or 6 parallel strings of 4 LEDs. The first configuration is preferable since this will result in fewer parallel strings. Another advantage of this configuration is that it avoids very low duty cycles of the switch.

A configuration with 4 parallel strings of 6 LEDs in series each was chosen. Looking at the voltage and current requirements of the LEDs, this results in a necessary output current of the LED driver of 120 mA at an output voltage of 33–39 V.

5.1.1. Balancing

When paralleling strings of LEDs, care must be taken to ensure that each string conducts approximately the same current. Due to the variation in forward voltage, the current in the strings might become unbalanced, making the irradiation inconsistent and potentially causing thermal runaway [42]. There are multiple ways to balance the current through the LEDs, some of the main methods will be discussed below.

- Using a resistor in series with each string. When the current through a string goes up, the voltage over the resistor will rise. This in turn decreases the current through the string again. This method causes a lot of losses and does not offer very good balancing.
- Using inductors for the balancing, where the value of the inductor determines the average current through the LED string. This approach combines the DC/DC converter and balancing circuit, but requires an inductor per LED string. [43]

- Using balancing transformers, in combination with an AC conversion, where 2^N LED strings are connected using $2^N - 1$ transformers with a turns ratio of 1:1. This offers superior balancing (less than 1 % current imbalance). [44]
- Using transistors to create a current mirror that only allows a certain current to flow through each LED string. Any mismatch in forward voltage results in different voltages across the transistors. This is called active balancing [45].

Each balancing method has its advantages and its limitations. The best method depends on the LED power, allowable LED current imbalance, and complexity. Most low and medium-power LED driver ICs implement LED string balancing using the current mirror approach. This is a good compromise between complexity, current matching performance, and power dissipation.

5.2. Topology

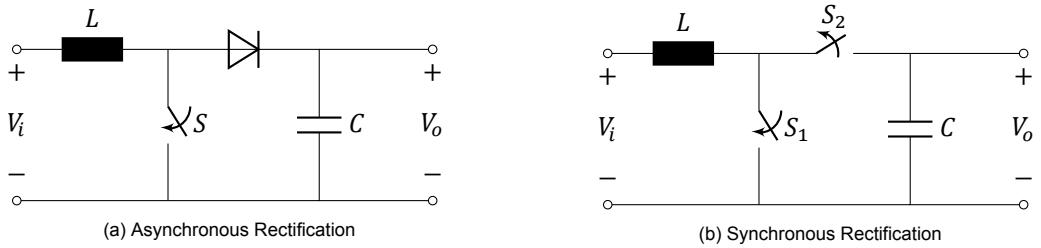
In principle, any DC/DC converter topology can be used to drive LEDs. Because LEDs behave like constant voltage loads, a current source type converter is preferred [46]. This will ensure that the desired current flows through the LED and the desired radiant flux is produced, while the voltage is variable and determined by the LED.

To determine which topology should be used, the input and output requirements of the LED driver are listed in Table 5.1.

Table 5.1: LED Driver Input and Output Requirements

Parameter	Value
Input voltage	12.5–21 V
Output voltage	33–39 V
Output current (nominal)	120 mA
Control type	Constant current
Galvanic isolation	Not necessary
Power flow	Unidirectional

Since the output voltage range is higher than the input voltage range, a step-up converter is needed. The simplest step-up converter topology meeting the requirements in Table 5.1 is the boost topology shown in Figure 5.1a. The switch is driven using pulse-width modulation (PWM), controlling the on- and off-time.



(a) Asynchronous Rectification

(b) Synchronous Rectification

Figure 5.1: Boost Converter Topology

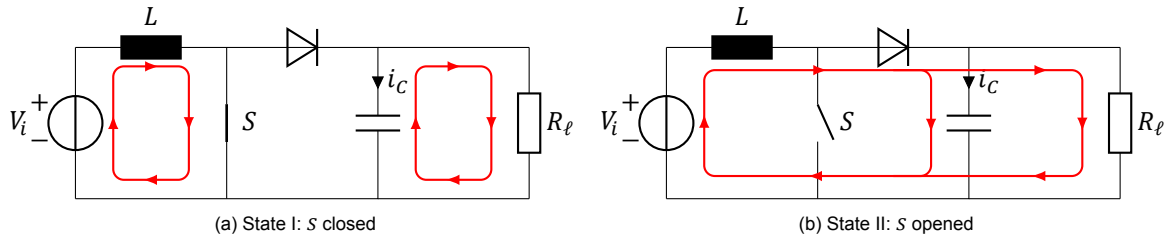
The diode can also be replaced by a switch acting as a synchronous rectifier (Figure 5.1b). This may improve converter performance since the conduction losses of the switch are usually much smaller than the conduction losses of the diode [47]. This is discussed in more detail in Section 5.5.1.

5.3. Modeling

In this section, several models of the boost converter will be discussed. These models will be used as a theoretical basis for the design decisions made in Section 5.5.

5.3.1. Steady-State Model CCM

The boost converter has two states in continuous conduction mode (CCM), which are shown in Figure 5.2. In state I, the switch S is closed and the inductor current rises. In state II the switch is opened and the energy of the inductor is transferred to the output capacitor and the load.



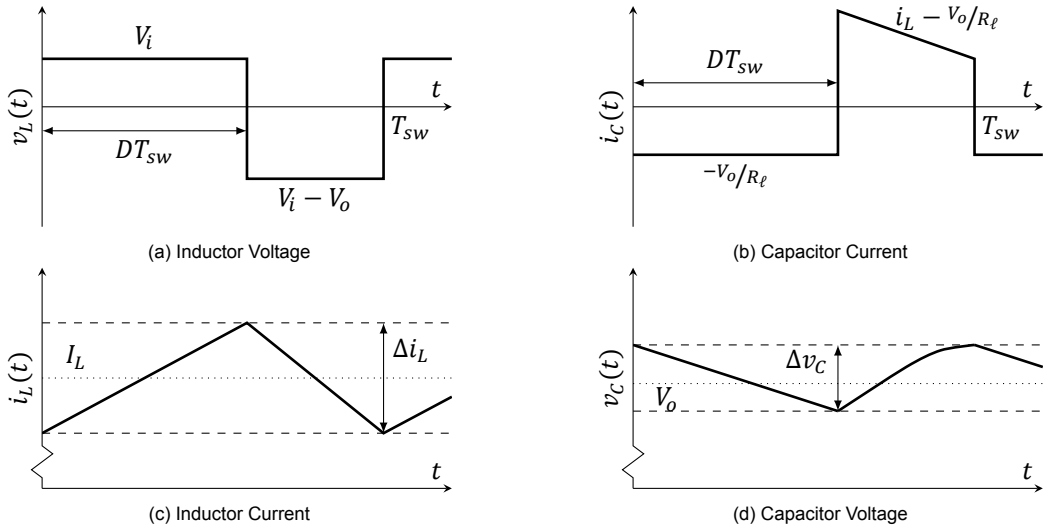
(a) State I: S closed (b) State II: S opened

Figure 5.2: Boost Converter States

The switching frequency is f_{sw} [Hz], and the duty cycle D [%] determines the on- and off-time of the switch: $t_{on} = DT_{sw}$, $t_{off} = (1-D)T_{sw}$. In steady-state, the inductor current and capacitor voltage must obey (5.1) [48].

$$\begin{cases} i_L(kT_{sw}) = i_L\left((k+1)T_{sw}\right) \\ v_C(kT_{sw}) = v_C\left((k+1)T_{sw}\right) \end{cases} \qquad k \in \mathbb{Z} \tag{5.1}$$

The inductor current and capacitor voltage waveforms can be derived (using the small ripple approximation) from the schematics presented in Figure 5.2. The result is shown in Figure 5.3.



(a) Inductor Voltage (b) Capacitor Current

(c) Inductor Current (d) Capacitor Voltage

Figure 5.3: Boost Converter CCM Waveforms

From Figure 5.3a, the input-output voltage relation can be found. The steady-state condition dictates that the total volt-seconds applied to the inductor during a switching cycle must be zero [48]. From this, the conversion factor $M(D)$ can be derived.

$$0 = \int_0^{T_{sw}} v_L(t)dt = V_i DT_{sw} + (V_i - V_o)(1-D)T_{sw} \tag{5.2}$$

$$\frac{V_o}{V_i} = M(D) = \frac{1}{1-D} \tag{5.3}$$

$$\frac{I_o}{I_i} = \frac{1}{M(D)} = 1 - D \tag{5.4}$$

The current ripple Δi_L can also be derived from this figure by integrating over the on-time.

$$\Delta i_L = \frac{1}{L}\int_0^{DT_{sw}} v_L(t)dt = \frac{V_i D}{L f_{sw}} \tag{5.5}$$

Similarly, the voltage ripple Δv_C can be derived from Figure 5.3b by integrating over the on-time.

$$\Delta v_C = \frac{1}{C}\int_0^{DT_{sw}} i_C(t)dt = \frac{I_o D}{C f_{sw}} \tag{5.6}$$

This model can be used to derive requirements for parameters of the converter (such as switching frequency and duty cycle) and component values. However, it is an ideal model and does not include non-idealities such as the series resistance of the inductor and capacitor, on-state resistance of the switch, or forward voltage of the diode.

5.3.2. Steady-State Model DCM

Discontinuous conduction mode (DCM) occurs when the inductor is so small that it completely discharges before the switching cycle is over. The duty cycle δ of the diode is no longer $1 - D$. Neither the switch nor the diode will conduct for a period $(1 - D - \delta)T_{sw}$. In practice, the inductor current is not zero in this period, but oscillates around zero with a frequency determined by the inductor and the parasitic capacitance on the switch node ($\omega = 1/\sqrt{LC_{par}}$). The switch node voltage also oscillates during this period [49]. This is not drawn in the figures.

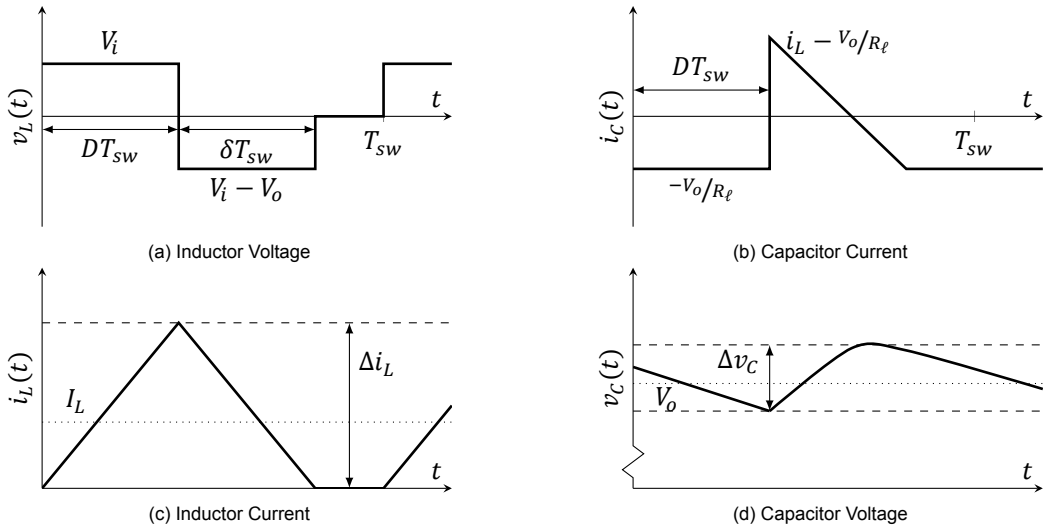


Figure 5.4: Boost Converter DCM Waveforms

In DCM, the voltage gain does not only depend on the duty cycle, but also on the load current. The converter goes into DCM when $K < K_{crit}$, with K and K_{crit} defined in (5.7) and (5.8) [48].

$$K = \frac{2Lf_{sw}I_o}{V_o} \tag{5.7}$$

$$K_{crit} = D(1-D)^2 \tag{5.8}$$

By again applying the steady-state condition and inductor volt-second balance, the conversion ratio in DCM can be derived. Note that this does indeed depend on the load through the factor K. The derivation is presented in Appendix B.2.2.

$$\frac{V_o}{V_i} = M(D,K) = \frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{4D^2}{K}} \qquad \text{if} \quad K < K_{crit}(D) \tag{5.9}$$

5.3.3. Converter Losses

In this section, a simple analysis of the converter losses is presented based on the book by Ericson, et al. [48]. A very detailed analysis is outside the scope of this thesis and is also not necessary or interesting because of the low power of the converter (<5 W). The model will be used to support the design decisions and component selection.

Switch The switch is implemented as a MOSFET in most low and medium-power applications. The losses generated by the MOSFET can be divided into two categories: conduction losses and switching losses. The conduction losses can be modeled by considering the on-state behaviour of the MOSFET. It acts approximately as a resistor $R_{S,on}$, and hence the switch conduction losses can be calculated by (5.10), where $I_{S,rms}$ is the RMS current through the switch.

$$P_{S,cond} = R_{S,on} I_{S,rms}^2 \tag{5.10}$$

The switching losses consist of several components: finite turn-on and turn-off time, reverse recovery of the rectifier diode, and the charging/discharging process of the output capacitance.

$$P_{S,sw} = P_{S,sw,on} + P_{S,sw,off} + P_{S,rr} + P_{S,Co} \tag{5.11}$$

The turn-on and turn-off losses can be estimated by assuming the current to be constant, and the voltage a linearly decreasing or increasing function of time. The following approximations are found [50].

$$P_{S,sw,on} = \frac{1}{2} V_{S,on} i_S(t = t_{on}) t_{S,on} f_{sw} \tag{5.12}$$

$$P_{S,sw,off} = \frac{1}{2} V_{S,off} i_S(t = t_{off}) t_{S,off} f_{sw} \tag{5.13}$$

$$P_{S,rr} = \left(V_{S,on} i_S(t = t_{on}) t_{rr} + V_{S,on} Q_{rr} \right) f_{sw} \tag{5.14}$$

$$P_{S,Co} = \frac{1}{2} \left(C_{S,o} V_{S,on}^2 + C_{D,o} V_D(t = t_{on})^2 \right) f_{sw} \tag{5.15}$$

Here, $V_{S,on}$ and $V_{S,off}$ are the voltages across the switch at turn-on and turn-off respectively. The switching times $t_{S,on}$ and $t_{S,off}$, and output capacitance $C_{S,o}$ are properties of the switch. The reverse recovery time t_{rr}, charge Q_{rr}, and output capacitance $C_{D,o}$ are properties of the rectifier diode.

Diode In the conducting state, a diode can be modeled as a voltage source (representing its forward voltage drop) and a resistor in series [51]. The resulting conduction losses can be calculated by (5.16).

$$P_{D,cond} = R_{D,on} I_{D,rms}^2 + V_F I_{D,avg} \tag{5.16}$$

Usually, Schottky diodes or ultrafast diodes are used as rectifiers in converters. Schottky diodes have almost no reverse recovery and are often used in high-frequency designs for this reason. However, they do have a significant junction capacitance which may be up to 5 times that of a comparable p-n junction diode [51]. This causes losses in the switch which are described by (5.15).

Inductor Inductors exhibit two types of power loss: copper loss, originating from the resistance of the wire, and core loss, due to hysteresis and eddy currents in the magnetic core. The copper loss can be modeled as a resistance R_L in series with the ideal lossless inductor.

$$P_{L,Cu} = R_L I_{L,rms}^2 \tag{5.17}$$

The core losses can be approximated using the Steinmetz equation (5.18) [48].

$$P_{fe} = K_{fe0} f^{\alpha} (\Delta B)^{\beta} A_c \ell_m \tag{5.18}$$

The parameters K_{fe0}, α, and β are determined by fitting (5.18) to the datasheet core loss graph, and $A_c \ell_m$ is the volume of the core. The core losses can be modeled by a frequency-dependent resistor in parallel with the inductor [52].

MATLAB The equations presented above are implemented in MATLAB, along with complementary functions for calculating the RMS and average component currents (derivation is presented in Appendix B.2). This model can be used to quickly simulate component losses and efficiency for different converter operating points and component choices. Comparing this model with corresponding LTspice simulations shows that it matches these to within a few tens of milliwatts (around 5 %). The MATLAB code is included in Appendix F.2.

5.4. Control

The output voltage of the boost converter can be controlled by varying the duty cycle, this is called voltage-mode control or direct duty ratio control.

Another possibility is current-mode control, in which the PWM signal is generated based on a measured current signal (such as the inductor or switch current). Several methods for current-mode control exist: peak current-mode, valley current-mode, and average current-mode control [48]. These will be briefly discussed to introduce their properties, but a detailed analysis is outside the scope of this thesis. The control IC selected in Section 5.5 will implement the control loop.

These control methods only influence the PWM generation and are usually the inner loop. An outer control loop is necessary to control the reference for the inner loop. This outer loop will provide the actual output regulation (constant voltage or constant current).

5.4.1. Peak Current-Mode Control

In peak current-mode control (PCMC), the switch is driven by a PWM signal generated based on the inductor or switch current waveform. At the beginning of each switching cycle, the switch is turned on. When the current reaches a reference peak value, the switch is turned off.

PCMC makes use of the current sensor information to obtain simpler converter dynamics. The switch is also protected against excessive current by limiting the switch current to a peak value. The loop compensation in current-mode control is also much simpler than in voltage-mode control. However, PCMC is very susceptible to noise in the measured and reference currents.

Subharmonic Oscillation The PCM controller is unstable whenever the steady-state duty cycle is greater than 0.5. This can be shown by considering how a small perturbation influences the inductor current waveform. Ericson, et al. show that a perturbation $\hat{i}_L(0)$ becomes (5.19) [48].

$$\hat{i}_L(nT_{sw}) = \hat{i}_L(0)\left(-\frac{D}{1-D}\right)^n \qquad n \in \mathbb{N} \tag{5.19}$$

This only decays to zero for $n \to \infty$ if $D < 0.5$. This can be resolved by adding an artificial compensating ramp to the sensed current waveform. This is called slope compensation [48, 53].

This problem does not appear in discontinuous conduction mode, because disturbances introduced in one cycle will no longer influence the next cycle: the inductor current will be zero for a part of the switching cycle, causing the current to start rising from a well-defined point.

Valley Current-Mode Control Valley current-mode control is very similar to peak current control, but it turns the switch on when the sensed current reaches a minimum level. It has almost identical properties and suffers from subharmonic oscillations for $D < 0.5$ if no slope compensation is applied.

5.4.2. Average Current-Mode Control

In average current-mode control (ACMC) an average converter current (such as the output current) is measured. The PWM control signal is generated based on the error between the measured and reference current.

ACMC has several advantages over PCMC: First, direct control over the average current is required in some applications such as battery chargers and LED drivers. Furthermore, low-pass filtering associated with the current sensing reduces sensitivity to noise and switching disturbances. It can achieve stable operation without slope compensation for any duty cycle [54].

One disadvantage of ACMC is that there is no limitation for the peak switch current, but only for the average current. As a result, additional circuitry is necessary to obtain cycle-by-cycle protection against excessive peak currents during transients [48].

5.5. Design

In the following paragraphs, the design of the boost converter will be described. First, several design considerations are discussed, after which the converter is designed and components are selected using the proposed converter loss model.

5.5.1. Design Considerations

According to requirement [**KPI.2**], the LED driver should be small and efficient. In the design of the boost converter, several trade-offs must be made to achieve this. In the following paragraphs, the trade-offs and how they will impact the design will be discussed.

Conduction and Switching Losses As derived in Section 5.3.3, the conduction and switching losses of the semiconductor devices depend heavily on their electrical characteristics. While selecting components, these parameters must be kept in mind to obtain low losses.

There is often a trade-off between conduction and switching losses. The switching frequency must also be taken into account when considering this trade-off.

Switching Frequency Higher switching frequency allows for smaller passive components and lower passive losses, as well as a lower inductor value to obtain the same ripple than a lower switching frequency. However, it does generate higher switching losses in the semiconductor devices. The critical frequency f_{crit} [Hz] is the switching frequency for which the dynamic (frequency-dependent) losses are equal to the static losses. It is a rough upper limit on the switching frequency of a practical converter. Beyond the critical frequency, the efficiency will decrease rapidly.

Besides using a fixed switching frequency, it is also possible to use a variable switching frequency. This makes it possible to achieve high efficiency over a wide load range but makes it harder to design a good input filter than for a fixed frequency.

To reduce EMI, the switching frequency is sometimes varied slightly around a fixed center frequency. This is called spread spectrum or random pulse-width modulation [55].

Conduction Mode The choice between continuous and discontinuous conduction mode depends on many variables. In DCM, the inductor can often be much smaller than in CCM due to the higher allowable ripple. Additionally, CCM will exhibit sub-harmonic oscillation when using peak current control without slope compensation if $D > 0.5$ and loop compensation is more difficult [53]. Because of the larger ripple and discontinuous inductor current, EMI might be a concern when using DCM.

The semiconductor switching losses are usually reduced by using DCM since the turn-on happens at zero current (ZCS) and reverse recovery losses are reduced. The turn-off losses might be increased slightly by the increased turn-off current, but these are usually much smaller than the turn-on losses.

Synchronous Rectification For synchronous rectification, the rectifier diode is replaced by a synchronously driven switch, as shown in Figure 5.1b. This has various advantages and disadvantages.

The main advantage of synchronous rectification is the reduction of the conduction losses. The diode, even if it is a Schottky diode with a relatively low V_F, has a significant voltage drop. This voltage drop is eliminated by using synchronous rectification. Furthermore, thermal performance may be improved and PCB size reduced [47].

However, there is also a significant cost associated with using synchronous rectification. At high frequencies, the losses in the switch (due to dead-time conduction and reverse recovery of the body diode) may become larger than the achieved reduction in conduction losses. Furthermore, the control logic will be more complex [56].

5.5.2. Operating Point

From the input and output requirements presented in Table 5.1, the possible operating points of the converter can be derived. From this, specifications such as average value and ripple can be established for the inductor current and output voltage.

The two conduction modes CCM and DCM will both be considered for the implementation. A trade-off between the two solutions will be made in Section 5.5.6.

CCM Over the full input and output voltage range, the duty cycle in CCM will be $0.36 \leq D \leq 0.68$. This can be calculated from (5.3). The average inductor current I_{av} can be calculated from the input and output power. Neglecting any losses in the converter the input and output power must be equal.

$$I_{av} = \frac{I_o V_o}{V_i} \tag{5.20}$$

It follows that the average inductor current is between 190–375 mA. Typically, a ripple between 20–40 % is chosen to limit radiated EMI, switching losses, and inductor core losses. The output voltage ripple due to the ESR of the output capacitor may also be a concern [57]. In this case, a ripple of 130 mA will be deemed acceptable based on several simulations.

The output voltage ripple should be small to limit the variation of the current (and heat dissipation) in the LEDs. However, since this is not a lighting application, small variations in radiant flux are acceptable. A voltage ripple of at most 50 mV should be achievable using reasonably sized components.

DCM In discontinuous conduction mode, the duty cycle can be calculated from (5.9). Over the full input and output voltage range, the duty cycle in DCM will be $0.22 \leq D \leq 0.55$ at nominal load (I_o = 120 mA). To ensure DCM operation, the current ripple should be larger than 200 % or 750 mA. The output voltage ripple in DCM should also be less than 50 mV.

5.5.3. Control Circuit

First, an LED driver integrated circuit will be selected because this will determine what is possible and required for the rest of the converter design. In Table 5.2, several suitable driver ICs are presented. The listed output currents are per LED string, each driver IC supports four strings.

Table 5.2: LED Driver IC Comparison Table

Type	V_{in} [V]	V_{out} [V]	I_{out} [mA]	f_{sw} [kHz]	Switch	Note
LP8551	2.7–22	<40	50	156–1250	Internal (N-FET)	
LT3599	3.1–30	<44	120	200–2100	Internal (npn)	LED T_j derating
TPS61194	4.5–40	<45	100	300–2200	Internal (N-FET)	Spread-spectrum f_{sw}
TPS92390	3–48	<48	200	100–2200	External	Spread-spectrum f_{sw}

The listed ICs feature integrated four-channel LED string balancing using the current mirror approach (see Section 5.1.1). Additionally, most LED driver ICs have an integrated switch. These features are preferred since they will reduce component count and make the circuit smaller.

The LT3599 [58] features analog dimming control which can be used to program a derating curve for maximum LED current versus temperature. This feature could be useful to prevent the LED temperature from rising too high in case the device cannot be cooled adequately. Additionally, it does not require an external low-voltage supply. Hence, the LT3599 is chosen as the LED driver IC.

5.5.4. Inductor

By rewriting the equation for current ripple in CCM, equation (5.21) is obtained. Filling in the worst-case (largest) value for $V_i D$ and the inductor current ripple, a minimum Lf_{sw} product of 75 is found.

$$Lf_{sw} \geq \frac{V_i D}{\Delta i_L} = 75 \tag{5.21}$$

In DCM, the inductance can be much smaller because a larger current ripple is allowed. This condition can be derived from the boundary between CCM and DCM as described by (5.7) and (5.8).

$$Lf_{sw} \leq \frac{V_o D}{2I_o}(1-D)^2 = \frac{V_i D}{2I_{av}} = 11.4 \tag{5.22}$$

This allows the inductor and the DC series resistance to be much smaller in DCM. However, because of the large current ripple the core losses will increase.

Finally, the inductor saturation current rating must be chosen such that the inductor does not saturate during normal operation. This would reduce the effective inductance, potentially creating a short circuit, and cause a lot of core losses. Typically, the saturation current is chosen 30 % above the maximum inductor current.

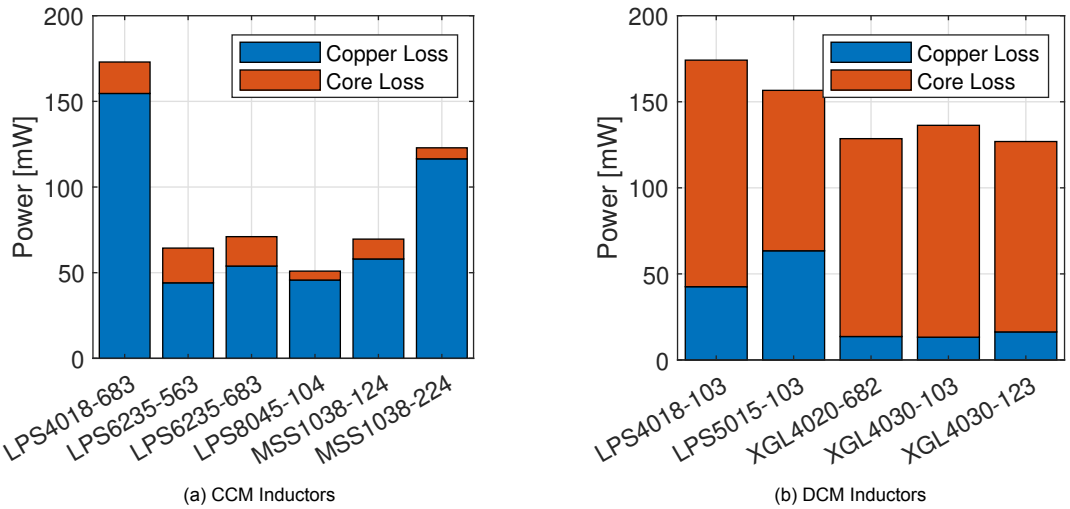
$$I_{L,max} = I_{av} + \frac{1}{2}\Delta i_L = \begin{cases} 440\,\text{mA} & \text{CCM} \\ \geq 750\,\text{mA} & \text{DCM} \end{cases} \tag{5.23}$$

In Table 5.3, a selection of inductors for the CCM and DCM operating points is presented. For each inductor, the DC series resistance, size, and the f_{sw} required to reach the desired operating point (the ripple requirement for CCM, boundary conduction mode for DCM) are listed.

Table 5.3: Inductor Comparison Table

Type	L [µH]	I_{rms} [A]	I_{sat} [A]	R_{DC} [mΩ]	Size [mm]	f_{sw} [kHz]
CCM						
LPS4018-683	68	0.56	0.46	950	3.9 × 3.9 × 1.7	1100
LPS6235-563	56	1.10	1.10	280	6.0 × 6.0 × 3.5	1340
LPS6235-683	68	1.00	0.96	345	6.0 × 6.0 × 3.5	1100
LPS8045-104	100	1.10	1.10	300	8.0 × 8.0 × 4.5	750
MSS1038-124	120	1.50	1.50	380	10.2 × 10.0 × 3.8	625
MSS1038-224	220	0.97	0.99	380	10.2 × 10.0 × 3.8	340
DCM						
LPS4018-103	10	1.30	1.3	200	3.9 × 3.9 × 1.7	1140
LPS5015-103	10	0.95	1.3	300	4.8 × 4.8 × 1.4	1140
XGL4020-682	6.8	4.20	3.4	63.6	4.0 × 4.0 × 2.1	1680
XGL4030-103	10	3.90	3.1	63	4.0 × 4.0 × 3.1	1140
XGL4030-123	12	3.40	2.7	78.5	4.0 × 4.0 × 3.1	950

The inductor losses were simulated in the worst-case operating point with the loss model presented in Section 5.3.3. The results are presented in Figure 5.5. The distribution of copper and core losses is roughly as expected for CCM and DCM.



(a) CCM Inductors

(b) DCM Inductors

Figure 5.5: Inductor Loss Comparison

A selection cannot be made based on just these results, as the choice for DCM or CCM also has an impact on the semiconductor losses. These will be discussed first, and the best design concept (combination of the inductor, diode, conduction mode, and switching frequency) will be chosen in Section 5.5.6. Two design concepts are considered with the best inductor choices: CCM with the LPS8045-104, and DCM with the XGL4030-123.

5.5.5. Semiconductors

Since the LT3599 control IC has an integrated switch and it does not support synchronous rectification, only the rectifier diode must be selected. As discussed above, this is preferably a Schottky diode.

The peak diode current and reverse voltage can be calculated from the current and voltage waveforms. As a rule of thumb, the breakdown voltage rating should be chosen 30 % higher than the peak reverse voltage. On the other hand, the peak current rating of Schottky diodes is usually much higher than the average rating, so this is not much of a concern [57].

$$I_{D,pk} = I_{av} + \frac{1}{2}\Delta i_L = 435\,\text{mA} \implies I_F \geq 435\,\text{mA} \tag{5.24}$$

$$V_{DR,pk} = V_o + \frac{1}{2}\Delta v_o = 40\,\text{V} \implies V_R \geq 52\,\text{V} \tag{5.25}$$

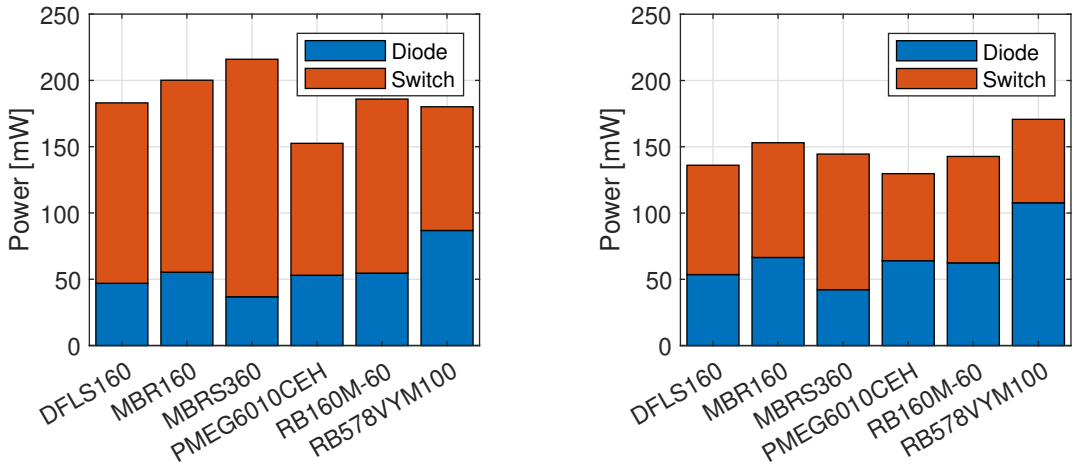
Finally, the forward voltage and junction capacitance should be as small as possible to minimize the conduction and switching losses respectively.

Several suitable diodes and their most important properties (maximum reverse voltage, continuous forward current, forward voltage drop, and junction capacitance) are presented in Table 5.4.

Table 5.4: Rectifier Diode Comparison Table

Type	V_R [V]	I_F [A]	V_F [mV]	C_d [pF]
DFLS160	60	1	500	67
MBR160	60	1	550	140
MBRS360	60	3	630	200
PMEG6010CEH	60	1	570	60
RB160M-60	60	1	490	120
RB578VYM100	100	0.7	850	50

In Figure 5.6, the simulated semiconductor losses are shown. The switch losses are estimated from the LT3599 datasheet [58] and incorporate the loss due to the charging and discharging of the diode junction capacitance. The PMEG6010CEH was chosen because it has the best performance over the operating range in CCM and DCM.



(a) With CCM Inductor

(b) With DCM Inductor

Figure 5.6: Semiconductor Losses (at f_{sw} = 750 kHz)

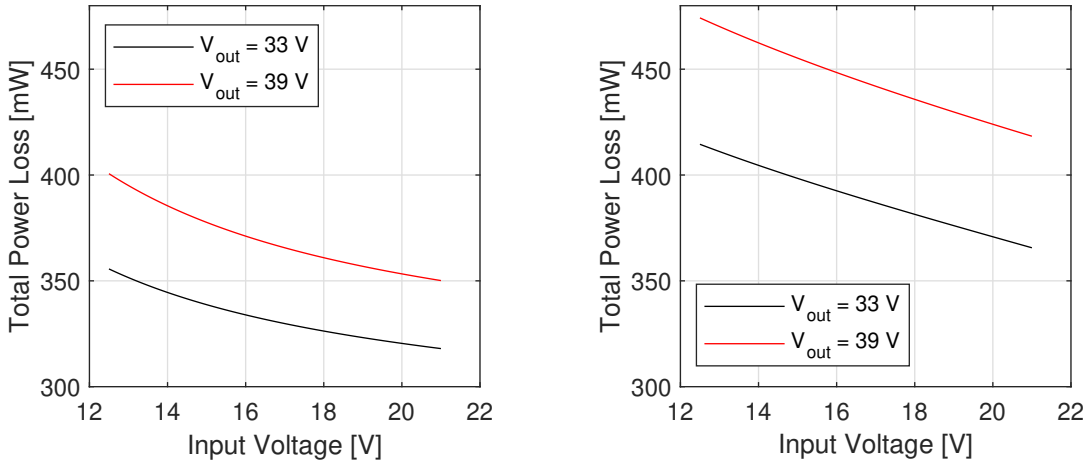
5.5.6. Concept Selection

Based on the design and simulation results of the previous section, two design concepts can be identified: (i) Continuous conduction mode with the LPS8045-104 inductor and PMEG6010CEH diode, and (ii) Discontinuous conduction mode with the XGL4030-123 inductor and PMEG6010CEH diode.

To allow for small passive components (most notably the input and output capacitors), the switching frequency should be as high as possible without causing too much switching losses. A trade-off was

made between these two factors, and a switching frequency of 750 kHz was chosen. Simulation figures of the losses as a function of switching frequency are presented in Appendix C.3.

Simulating the losses of these two configurations at $f_{sw} = 750$ kHz over the input and output voltage range shows that the CCM configuration has about 10 % smaller losses (Figure 5.7).



(a) Continuous Conduction Mode

(b) Discontinuous Conduction Mode

Figure 5.7: Converter Loss vs. Input Voltage

In Table 5.5, the two design concepts are compared. The total power loss, converter efficiency, and inductor volume $\mathcal{V}_L$ are listed for both concepts.

Table 5.5: Design Concept Comparison Table

Concept	P_{loss} [mW]	η [%]	$\mathcal{V}_L$ [mm^3]
CCM (i)	318–401	91.6–92.9	288
DCM (ii)	365–474	90.3–91.6	49.6

Considering requirement [**KPI.2**], the LED driver should be as efficient and as small as possible. From the table, it can be seen that the DCM configuration has slightly lower efficiency than the CCM configuration, but it also has a much smaller inductor. Besides this, DCM has several other advantages: DCM is inherently stable under peak current control, loop compensation is easier, and the losses in the IC are lower since the losses in the switch are lower (Figure 5.6). Hence, the DCM configuration was chosen for this design.

5.5.7. Passive Components

In this section, the passive components will be designed and selected. Most of these components are required to configure the various settings of the LT3599 driver IC (e.g., LED current, switching frequency, etc.). The equations and constants used in this section are taken from the datasheet [58]. Component names refer to the components in the schematic in Appendix D.

Output Capacitor The output capacitor must be selected to keep the output voltage ripple low enough. The voltage ripple consists of two components: the charging and discharging of the capacitor (derived in Appendix B.2.5) and the voltage drop over the ESR.

$$C_6 \geq \frac{(\Delta i_L - I_o)^2 I_o}{\Delta v_o \Delta i_L} \frac{L}{V_i D} = 2.35\,\mu\text{F} \tag{5.26}$$

$$\Delta v_{C,ESR} = R_C i_{L,pk} \tag{5.27}$$

The ESR should be as low as possible to limit the voltage ripple $\Delta v_{C,ESR}$ and power losses in the capacitor. With modern high-value multi-layer ceramic capacitors (MLCC), ESRs in the range of 1–10 mΩ are achievable.

More concerning are the various sources of capacitance loss in MLCC capacitors, most notably temperature and DC bias. To limit the temperature factor, the capacitor should have at least an X5R dielectric (±15 % variation over −55–85 °C). The influence of DC bias is always present in high-value MLCC capacitors [59], but may be reduced by choosing a part with a higher voltage rating or paralleling parts with lower capacitance.

Based on these considerations, the TDK C3225X7R1N106K250AC was chosen as output capacitor. It has a nominal capacitance of 10 µF, $\mathrm{ESR} = 1.8\,\mathrm{m\Omega}$ at 750 kHz and an X7R dielectric. The worst-case capacitance will be 2.6–3.0 µF at the nominal operating voltage (36 V), giving a worst-case output voltage ripple of 46.5 mV.

Input Filter The input filter must supply the high-frequency inductor ripple current to prevent this HF current from flowing through the potentially long traces/wires to the battery. If this happens, there might be resonance or interference with other subsystems (such as the wireless charging system).

Because of the relatively low power and ripple, it is not necessary to make a complete EMI filter at the converter input. A single capacitor for high-frequency decoupling is enough. Although the board parasitics (such as stray inductance in the source wires) may affect the input filter, simulations have shown that the influence is minimal for reasonable stray inductance values (approx. 150 nH) [60].

The TDK C2012X7S1E106K125AC was chosen as the input capacitor. Again, taking into account temperature and DC bias, this capacitor has a worst-case capacitance of approximately 2.3 µF at the maximum DC operating voltage. LTspice shows that this gives an input voltage ripple of <65 mV.

LED Current Setting The nominal LED current is 30 mA [34]. To set this value, the control IC needs an external resistor R_{ISET} whose value can be calculated with (5.28).

$$R_{\mathrm{ISET}} = \frac{1330}{I_{LED}} = 44.2\,\mathrm{k\Omega} \tag{5.28}$$

Switching Frequency Setting The switching frequency is set by the resistor R_{T}. The value of this resistor can be determined from Figure 9 of the datasheet [58].

$$R_{\mathrm{T}} = 77.5\,\mathrm{k\Omega} \qquad \text{for} \quad f_{sw} = 750\,\mathrm{kHz} \tag{5.29}$$

Over-voltage Protection The nominal output voltage is 36 V, but the output voltage may increase to 39 V depending on the forward voltage of the LEDs. To protect the driver IC and LEDs, the over-voltage limit should be set 10 % higher than the maximum operating voltage.

$$V_{\mathrm{OUT(MAX)}} = 1.223\,\mathrm{V} \cdot \left(1 + \frac{R_8}{R_9}\right) \tag{5.30}$$

The required resistor values are $R_8 = 1\,\mathrm{M\Omega}$, $R_9 = 30\,\mathrm{k\Omega}$ and, which gives $V_{\mathrm{OUT(MAX)}} = 42\,\mathrm{V}$.

Under-voltage Lockout The under-voltage lockout (UVLO) should be set to the lowest allowable input voltage for the LED driver. The control IC defines separate thresholds for turn-on and turn-off.

$$V_{\mathrm{S(OFF)}} = 1.4\,\mathrm{V} \cdot \left(1 + \frac{R_6}{R_7}\right) \tag{5.31}$$

$$V_{\mathrm{S(ON)}} = V_{\mathrm{S(OFF)}} + 4\,\mathrm{\mu A} \cdot R_6 \tag{5.32}$$

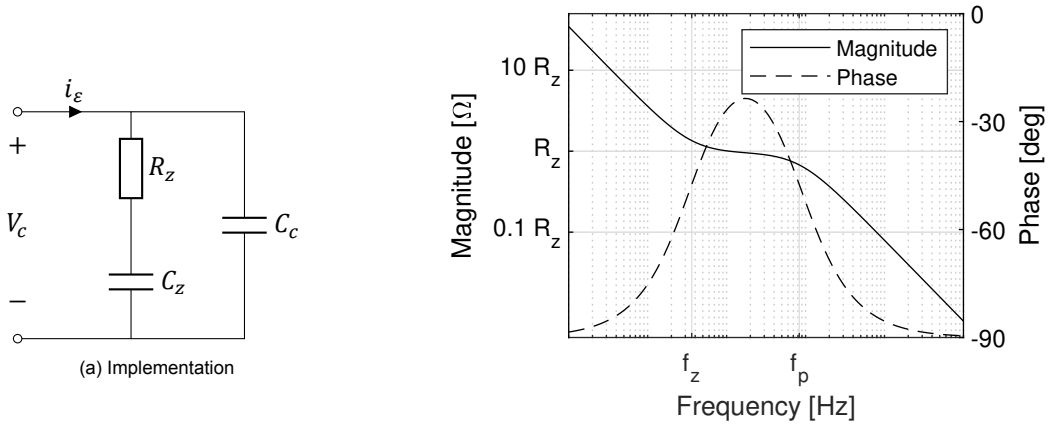
For nominal operation, the input voltage should be higher than 12.5 V. Hence, the turn-on threshold should be 12.5 V. The turn-off threshold must be lower (e.g., 12 V). The required resistance values are $R_6 = 200\,\mathrm{k\Omega}$ and $R_7 = 27.4\,\mathrm{k\Omega}$.

Soft Start The soft-start function of the LT3599 limits the inrush current and output voltage overshoot during startup. An external capacitor C_{SS} determines the switch current ramp up. The capacitor is charged from 0.3 V to V_{REF} with $I_{\mathrm{SS}} \approx 11\,\mu\mathrm{A}$. A capacitor of 47 nF is recommended, giving a ramp-up time of 3 ms.

$$C_{\mathrm{SS}} = \frac{I_{\mathrm{SS}} \Delta t}{\Delta V_{\mathrm{SS}}} = 47\,\mathrm{nF} \tag{5.33}$$

Compensation Network The compensation network is required to ensure the loop stability of the LT3599. However, the datasheet does not give any equations to calculate the required component values. By analyzing the block diagram of the IC, it can be found that the LT3599 internally generates two error (current) signals from the LED current and the over-voltage protection in the outer control loop. These two error signals are summed and flow into the compensation network, which converts the current into a voltage that is then used as reference for the peak current control.

The suggested compensation network is shown in Figure 5.8. The network is a Type-II compensator (lag compensator) and acts as a PI controller (R_z, C_z) with an extra integrator C_c to attenuate the high-frequency switching noise. The Type-II compensator is used to improve the phase response in the mid-frequency range [46, 48, 61].



(a) Implementation

(b) Impedance Frequency Response

Figure 5.8: LT3599 Compensation Network

Not enough information is given in the datasheet to calculate the required component values, so appropriate values are derived through simulation, starting from the default values suggested in the datasheet. The final values are $C_c = 50\,\mathrm{pF}$, $C_z = 2.2\,\mathrm{nF}$, $R_z = 25\,\mathrm{k\Omega}$.

LED Current Derating The LT3599 features LED current derating through the CTRL pin. This can be implemented by a resistive divider using a thermistor next to the LEDs. The voltage on the CTRL pin will drop as the temperature rises (NTC thermistor), reducing the LED current.

Thermal Protection The driver IC is the primary source of losses in this circuit as it performs the LED string balancing and has an integrated switch. The IC has built-in programmable junction temperature protection.

$$V_{\mathrm{TSET}} = V_{\mathrm{REF}} \frac{R_{12}}{R_{12} + R_{11}} \tag{5.34}$$

The datasheet recommends $R_{11} = 53.6\,\mathrm{k\Omega}$ and $R_{12} = 80.6\,\mathrm{k\Omega}$ to set the temperature protection to 100 °C. This will be sufficient for this application.

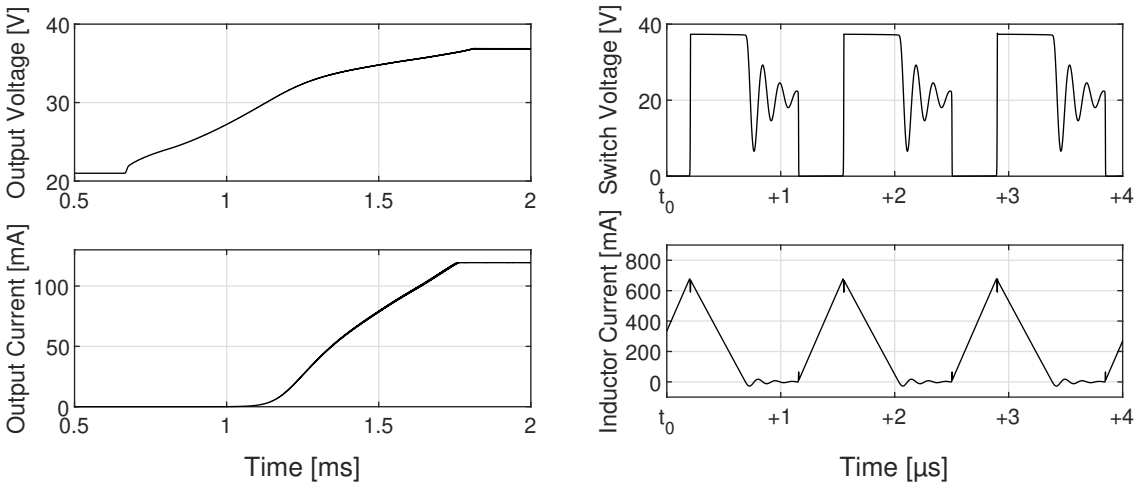
Fault Detection The LT3599 driver IC has two open-collector outputs to indicate a short or open-circuited LED string. A pull-up resistor of 100 kΩ should be used.

5.6. Simulation

The designed LED driver was simulated using LTspice. The schematic is presented in Appendix D, and the simulation netlist is included in Appendix F.3. The component models and subcircuits are obtained from their manufacturers.

5.6.1. Waveforms

In Figure 5.9a, the output voltage and current during startup phase of the LED driver are presented. This shows that the output is well regulated during startup and only goes up to the set current of 120 mA.



(a) Startup

(b) Steady State

Figure 5.9: LTspice Simulation of the Converter ($V_i = 21\,\mathrm{V}$)

In Figure 5.9b, the switching waveforms during steady-state are shown (t_0 is chosen somewhere during steady-state operation). These match the expected waveforms presented in Figure 5.4. Some numerical simulation results are presented in Table 5.6 for the minimum and maximum input voltage. These results show that the converter complies with the specifications.

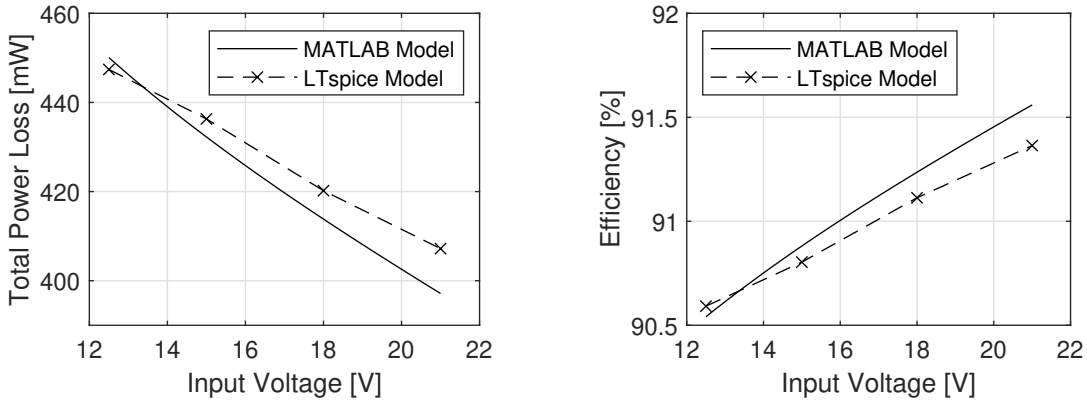
Table 5.6: Numerical Simulation Results

Parameter	Value $V_i = 12.5\,\mathrm{V}$	Value $V_i = 21\,\mathrm{V}$
Average inductor current	372 mA	219 mA
Inductor current ripple	830 mA	675 mA
Output voltage ripple	45 mV	42 mV
Efficiency	90.6 %	91.3 %

5.6.2. Efficiency

Next, the power loss and efficiency over the input voltage range are calculated from several LTspice simulations. The result is plotted together with the prediction of the MATLAB model in Figure 5.10.

The MATLAB model makes a relative error in power loss less than 2.5 % when compared with the LTspice simulation. This is an acceptable error since the MATLAB model is based on idealized current and voltage waveforms and simplified component models.



(a) Power Loss

(b) Efficiency

Figure 5.10: Performance over Input Voltage Range (V_o = 36 V)

5.6.3. Shutdown

The LT3599 IC has the ability to shut down if the $\overline{\text{SHDN}}$/UVLO pin is pulled low. This functionality can be used to shut off the LEDs when the Powerlizer is opened, as explained in Section 6.2.2.

The shutdown functionality was simulated using LTSpice. Because of the fast shutdown times, the parasitic inductance in the LED current path might generate large voltages on the input channel pins. This was also measured for a reasonable estimate L_{par} = 300 nH [60]. There are also anti-parallel diodes for each LED string to prevent the pin voltage from rising too high.

The simulation waveforms are presented in Figure C.5. From the figure, it can be seen that even though the voltage over the LEDs remains constant, due to the balancing current mirror being turned off, the LED current becomes zero. The shutdown time is 1.41 µs for the both simulated input voltages (12.5 V and 21 V). The voltage over each LED pin will rise to the output voltage. The simulated voltage drop over the parasitic inductance in each LED string is 1.3 V. This will not create any issues.

This also presents another advantage of using DCM operation. Because the inductor is small, it stores much less energy than an inductor suitable for CCM operation. This allows shutdown to happen very fast and with negligible voltage rise at the output.

5.7. Thermal Performance

Based on the LTspice simulation, the thermal performance of the LED driver is evaluated. In Table 5.7, the simulated (worst-case) power loss, thermal resistance (junction to ambient), and corresponding temperature rise are listed for each component in the power path.

Table 5.7: Thermal Performance

Component	P_{loss} [mW]	$R_{\theta JA}$ [K/W]	ΔT_{JA} [K]
Diode	57	330	18.8
Inductor	166	44	7.3
Driver IC	256	28	7.2

The dissipation in the input and output capacitors is sufficiently low (<1 mW) due to their low ESR and ripple current. No thermal resistance is given, but the expected temperature rise can be found in their datasheet. This is less than 0.05 K because of the relatively low current ripple.

Note that the dissipation in the driver IC can become much larger if the LED strings have very different forward voltages. It will ensure the same current flows through each string by dissipating the forward voltage difference in the internal current mirror. If the LEDs are mismatched across the entire output voltage range (33–39 V), the dissipation in the driver might become as large as 650 mW, leading to ΔT_{JA} = 18 K. This is still acceptable but should be avoided to maintain good efficiency.

5.8. Conclusion

In this chapter, the modeling and design of the LED driver of the sterilizer have been discussed. First, the electrical configuration of the UV-C LED array was chosen to be four parallel strings of six LEDs each, this is the optimal configuration from a balancing and driving perspective.

Next, the boost DC/DC converter topology was chosen based on the input and output requirements. An analytical loss model of this topology was developed in MATLAB. Based on this model, the converter was designed and components were selected. The MATLAB model predicts an efficiency between 90.3–91.6 % over the complete input-output operating range.

Finally, the designed circuit was simulated in LTspice to verify that all parameters meet the specifications and the thermal performance of the converter is evaluated. The simulations in LTspice show that this driver meets the requirements and can be used in the Powerlizer.

The LED driver satisfies functional requirement [**F.4**]. The efficiency of the driver is around 91 %, which is very high for low-power converters. This was an important design goal following from trade-off [**T.2**]. Since the switching frequency was chosen to be 750 kHz it also satisfies trade-off [**T.3**].

6 Controller

In this chapter, the conceptual design of a high-level controller for the Wireless Powerlizer will be presented. This controller will provide high-level monitoring, protection, and user interface functionality.

First, a microcontroller unit (MCU) is selected according to the requirements of each subsystem. The high-level controller will be implemented on this MCU. Finally, the functionality of the high-level controller is designed on a conceptual level. A complete software implementation is not the main focus of this thesis.

6.1. Microcontroller Selection

For the selection of the microcontroller unit, the requirements of each of the subsystems must be considered. The MCU will need several inputs for buttons, status monitoring, and communication; as well as outputs to control the various subsystems. These requirements are summarized in Table 6.1.

The battery management IC [1] can communicate with an external MCU using the SMBus protocol. If no MCU is available with SMBus support, it is also possible to use the I^2C serial protocol or a software implementation of the SMBus protocol.

Table 6.1: MCU Requirements

Requirement	BMS [1]	UVC	WCS [2]
Inputs	2	4	2
Outputs	1	3	3
Timers		1× (15 min) or RTC	
ADC	1× 8 bit		
Serial	SMBus or I^2C		I^2C

In Table 6.2, a selection of suitable microcontrollers is presented. Many different configurations and packages are available for each of these MCU types. These MCUs accept supply voltage up to 3.6 V.

Table 6.2: MCU Comparison Table

Type	Current [µA/MHz]	GPIO	Features
LPC11U6x	125	34–80	6 timers, 12 bit ADC, 2× I^2C
STM32F0x0	650	15–55	11 timers, 12 bit ADC, 2× I^2C (SMBus)
STM32L0x2	88	27–51	9 timers, 12 bit ADC, 2× I^2C (SMBus)
STM32L5x2	106	36–115	16 timers, 4× 12 bit ADC, 4× I^2C (SMBus), USB-PD

Based on the requirements, each of these MCUs would be acceptable. To make the system design simpler, the STM32L5x2 was chosen. This MCU incorporates a USB-PD controller, eliminating the

need for an extra controller IC for the battery system USB interface [1]. Additionally, it has low power consumption and several low-power run and standby settings.

6.2. Functional Design

In this section, the design of the high-level controller functionality is presented. This consists mainly of two aspects: safety and the user interface. Finally, the functionality is summarized in a state diagram.

6.2.1. Fault Protection

The LED driver IC has two active-low fault reporting pins $\overline{\text{OPENLED}}$ and $\overline{\text{SHORTLED}}$. These pins should be monitored by the controller. If a fault is detected, the controller should immediately stop the disinfection using the shutdown function of the driver IC, and the fault can be reported to the user (see Section 6.2.3).

If the disinfection were continued after a reported fault, this may compromise the effectiveness of the disinfection. There is no risk of further damage to the LED array by continuing since the LED driver controls the current in each string separately and will switch off faulty strings. However, the operating point of the converter may shift significantly.

6.2.2. Safety Shutdown

Due to the safety concerns related to UV-C radiation (see Chapter 3), the users' exposure should be limited as much as possible. According to safety requirement [**S.1**], the UV-C LEDs should be completely turned off at most 10 µs after the lid of the Powerlizer is opened.

There are multiple ways to detect whether the lid is open or not. A few of these options are using (i) a pressure sensor, (ii) a magnetic sensor, or (iii) a switch. These are all common methods in various products (such as laptops, etc.). The exact detector will have to be chosen by looking at all advantages and disadvantages of each method, however, this will not be done for the conceptual design.

The LED driver shutdown mechanism that was simulated in Section 5.6.3 shows that the LEDs can be completely turned off in less than 1.5 µs. This mechanism can be used for the safety shutdown. When the controller detects the opening of the lid, it can pull the $\overline{\text{SHDN}}$/UVLO pin of the LT3599 driver IC to ground, causing the LEDs to turn off. This mechanism also ensures that when the Powerlizer lid is opened, the UV-C disinfection can not start.

6.2.3. User Interface

The user should be able to interact with the Sterilizer in a manner that is fast and easy to understand. The most fundamental operation is starting and stopping a disinfection cycle. This may be implemented using a single button. Switching between the long and short disinfection (as described in Chapter 3) can either be implemented through an extra sliding switch or by pressing the start button for a short or long period of time. To avoid accidentally stopping a disinfection cycle, the required button press duration should be long (e.g. $\geq$1 s).

To indicate whether a disinfection cycle is currently being performed, an LED should be present next to the start button. This LED can also be used to report fault status if a multi-color LED (e.g., blue for disinfection, red for fault) is used.

6.2.4. State Diagram

The operation of the controller described above can be divided into distinct states with well-defined transitions. The functionality is summarized in the state diagram of Figure 6.1.

In the Off and Fault states, the $\overline{\text{SHDN}}$/UVLO pin of the driver IC should be pulled low to turn off the LED driver. This pin should only be allowed above the threshold in the Run state, turning on the driver.

After a fault, the device should not return to the Off state. Faults are only triggered for catastrophic events (shorted or open-circuited LED strings) which may reduce the effectiveness of a disinfection cycle and will cause the LED driver to go to the Fault state immediately after startup. To still allow the user to have some control, the device can return to the Off state after a reset.

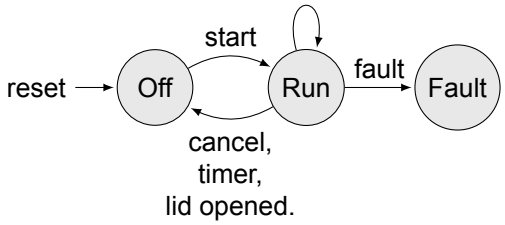


Figure 6.1: Controller State Diagram

6.3. Conclusion

In this chapter, the conceptual design of the high-level controller for the Sterilizer subsystem has been presented. The following functionality is implemented in the controller: (i) the user interface (starting and stopping a disinfection cycle), (ii) fault protection and reporting, and (iii) the safety shutdown, which prevents UV-C radiation escaping from the disinfection area when the lid of the Powerlizer is opened. The other functionality of the controller can be found in the theses concerning the other two subsystems [1, 2].

Using the controller, the Sterilizer design satisfies safety requirement [**S.1**] by shutting off the UV-C LEDs in less than 1.5 µs. It also implements the user interface elements required by functional requirement [**SF.4**].

7

Prototype

In this chapter, the design and verification of the prototype are discussed. This prototype is developed to demonstrate the functionality of the design presented in this thesis.

First, the scope of the prototype is defined by considering time, budget, and safety constraints. Then, the design is presented. Finally, the functionality of the prototype is verified according to the presented test procedure.

7.1. Scope

The prototype will be used to test and verify the functionality of the UV-C Sterilizer. Due to budget, time, and safety constraints, not all parts of the design will be incorporated into the physical prototype. In this section, the scope of the prototype and the included functionality will be discussed.

UV-C LED Array The UV-C LED array will not be built and tested for the prototype, because of the following reasons:

- Special personal protective equipment (such as gloves, glasses, etc.) would be needed to safely perform the tests and prevent exposure to UV-C radiation.
- It is also quite difficult to test the UV-C output. Because performance metrics like uniformity, irradiation dose, and disinfection level can not be measured, it is not possible to verify that the requirements are met.
- UV-C LEDs are very expensive and the full LED array requires 24 LEDs.

Therefore, it was decided not to include the UV-C LED array in the prototype and to use an array of regular LEDs with similar voltage and current capabilities.

LED Driver For the LED driver, most functions can be tested. The input and output voltage and current will be measured. This way, the input and output power can be determined and the efficiency of the driver can be calculated. Testing the LED temperature protection will be difficult without the LED array, therefore a potentiometer will be used to create an analog dimming control. The balancing properties of the LED Driver IC will also be tested. This can be done by removing an LED from one of the parallel strings to simulate what would happen if the forward voltage of an LED string changes.

Controller Since the top-level controller was only designed conceptually, it will not be tested in the prototype. The control signals (shutdown, fault detection, etc.) will be controlled and monitored manually.

3D Model To aid in the integration of the three subsystems of the Wireless Powerlizer, a three-dimensional model is created in Autodesk Fusion 360. This will also be useful for the visual representation of the system during the presentation of the business plan for this project.

7.2. Design

In this section, the design of the prototype will be presented. This consists mainly of two parts: the prototype PCB containing the LED driver and the 3D model of the complete Wireless Powerlizer.

7.2.1. PCB

A prototype PCB for the LED driver was designed. This approach was chosen because the most important components (LED driver IC, inductor, and diode) were only available as SMD parts. The schematic and board layout were made in KiCad and are presented in Appendix D. A 3D render is shown in Figure 7.1.

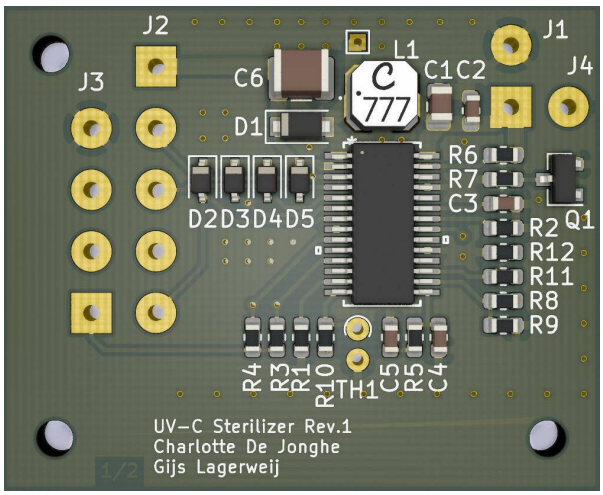


Figure 7.1: Prototype PCB

Because the chosen inductor (XGL4030-123) was not available at the time of ordering the components, the LPS4018-103 was chosen for the prototype. This will make it difficult to evaluate performance (efficiency, thermal performance, etc.), but will still allow for verification of the functionality of the LED driver.

As mentioned above, the LED temperature protection cannot be tested without the proper UV-C LED array. To be able to verify this functionality, the thermistor is replaced by an empty through-hole footprint. This will first be tested with a potentiometer to control LED brightness. Later, a thermistor may be soldered in place of the potentiometer.

LED Array To test the LED driver prototype a suitable load is necessary. For this purpose, an LED array with normal color LEDs is built, as explained in Section 7.1. This LED array must match the electrical properties of the designed UV-C LED array.

For this purpose, an 11×4 array of green LEDs (Würth Elektronik 151053GS03000) was built. These LEDs have a forward voltage of 3.4 V at a forward current of 30 mA. The four LED strings will thus have a nominal voltage of 37.4 V. This is sufficient to simulate the UV-C LED array.

7.2.2. 3D Model

The 3D model was made by combining all components from all three subsystems of the Wireless Powerlizer. For the UV-C Sterilizer, this meant the PCB and the LED array. Two renders of the complete 3D model can be found in Figure 1.2. The final dimensions of the Wireless Powerlizer were derived from the model, resulting in a box of 11.4 cm $\times$ 25.5 cm $\times$ 5.2 cm.

7.3. Verification

In this section, the operation of the prototype as described in Section 7.1 will be verified. The verification will be done according to the test procedure presented in Appendix E. The test setup is show in Figure 7.2.

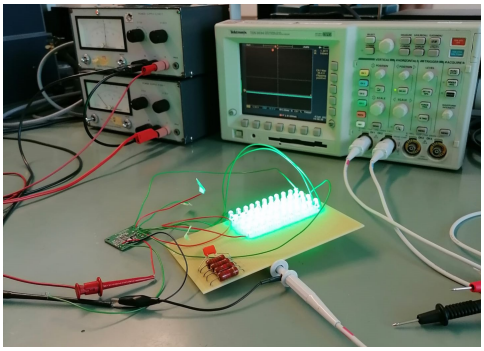
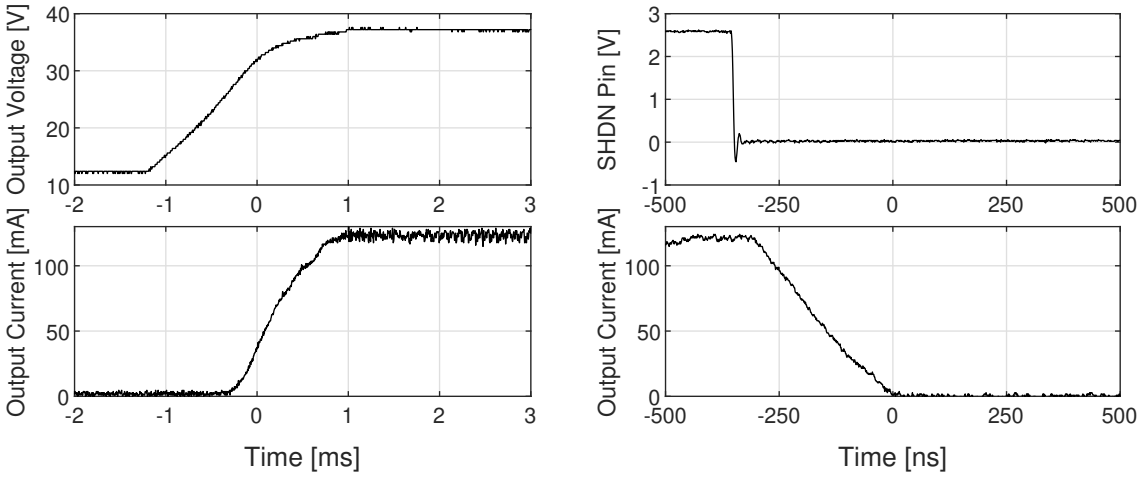


Figure 7.2: Test Setup with Working Prototype

7.3.1. Functional

First, the functionality of the driver was verified and several important waveforms were captured with the oscilloscope.

Startup and Shutdown The converter startup and shutdown function according to the design. Transient measurements are presented in Figure 7.3. The soft-start functionality can be seen clearly in Figure 7.3a. It takes approximately 2 ms, which is close enough to the design value of 3 ms. The output voltage and current are well-regulated and there is no overshoot or oscillation during startup.



(a) Startup (V_i = 12.5 V)

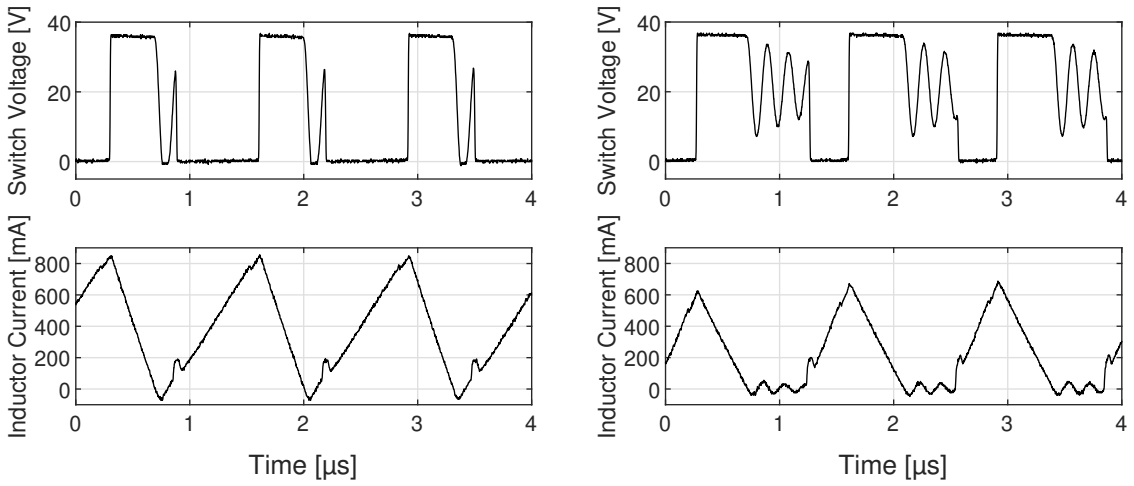
(b) Shutdown (V_i = 21 V)

Figure 7.3: Prototype Transient Measurements

Shutdown (in Figure 7.3b) does happen much faster than what was simulated: the current reaches zero in approximately 370 ns after the shutdown command is given. This is most likely due to a difference between the LTspice model of the LT3599 and the real integrated circuit.

Switching Waveforms The switch-node voltage and inductor current were measured for $V_i = 12.5$ V and 21 V. The waveforms are presented in Figure 7.4. It can be seen that the converter stays in DCM over the complete input voltage range.

The measured waveforms match the LTspice simulations (Section 5.6) quite well. The current ripple is as expected for both operating points. The switching frequency is approximately 760 kHz. From the frequency of the oscillation during the period of zero inductor current, it can be derived that the parasitic capacitance on the switch-node is approximately $C_{par} \approx 87$ pF. This matches the values expected from the switch and diode quite well ($C_{S,o} \approx 50$ pF, and $C_{D,o} \approx 20$ pF).



(a) V_i = 12.5 V (b) V_i = 21 V

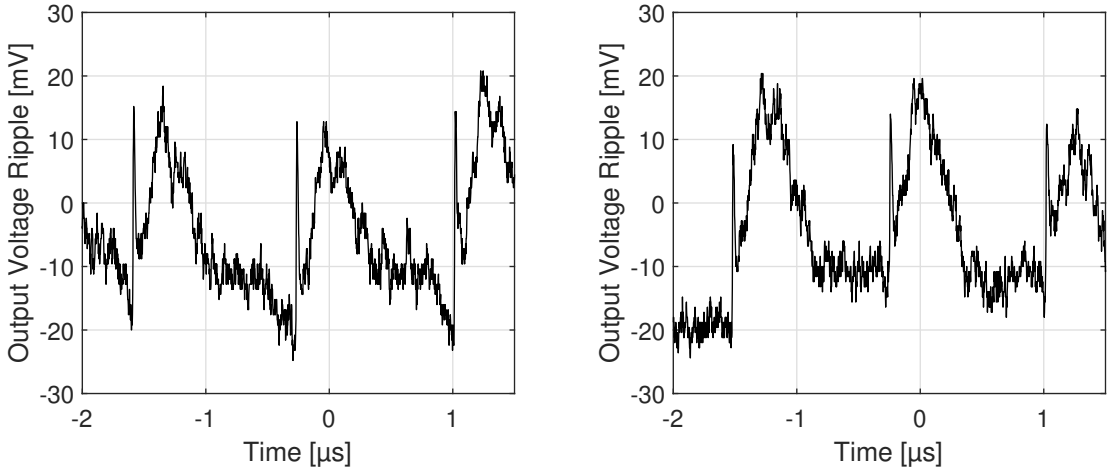
Figure 7.4: Switching Waveform Measurements

Analog Dimming Using an external potentiometer, the LED current can be regulated between 0.6 mA (0 %) and 30 mA (100 %). The LED current does not become zero completely, but this result is adequate for the intended application (temperature derating).

7.3.2. Performance

Finally, several parameters related to the performance of the converter were measured: voltage ripple, efficiency, and thermal performance.

Output Ripple The output voltage ripple was measured to be less than the target value of 50 mV over the complete input voltage range. For $V_i = 12.5\,\text{V}$, the measured ripple was 43.2 mV_{pp}. For $V_i = 21\,\text{V}$, it was 38.4 mV_{pp}. The voltage ripple waveforms for the two measured input voltages are presented in Figure 7.5.

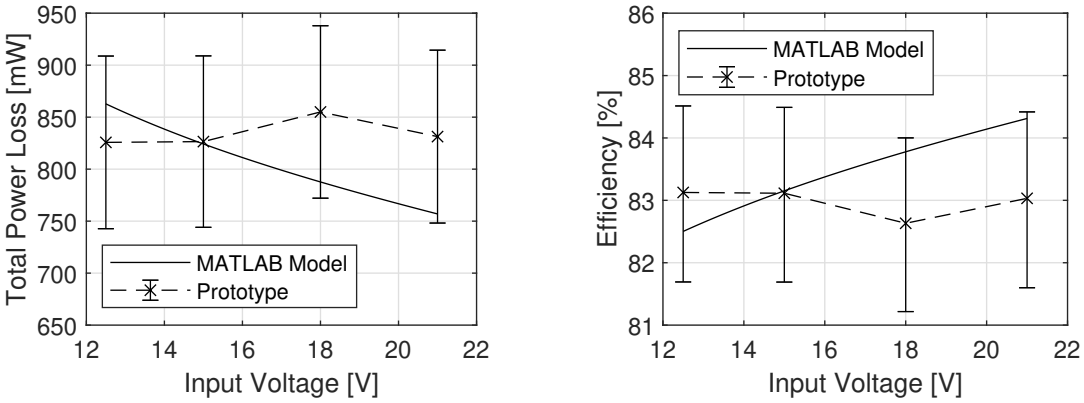


(a) Ripple (V_i = 12.5 V) (b) Ripple (V_i = 21 V)

Figure 7.5: Output Voltage Ripple Measurements

The shape of the voltage waveform matches the waveform presented in Chapter 5. There is an additional voltage spike at the lowest point of the ripple waveform. This is most likely interference due to the switch switching off the peak inductor current at that time. At turn-on the current is approximately zero, hence there is no spike there.

Efficiency The results of the power loss and efficiency measurements on the prototype are presented in Figure 7.6. The error bars indicate the uncertainty of the measurement introduced by the multimeters (Voltcraft VC276TRMS and Elro M970). The uncertainty calculations are derived in Appendix B.3.



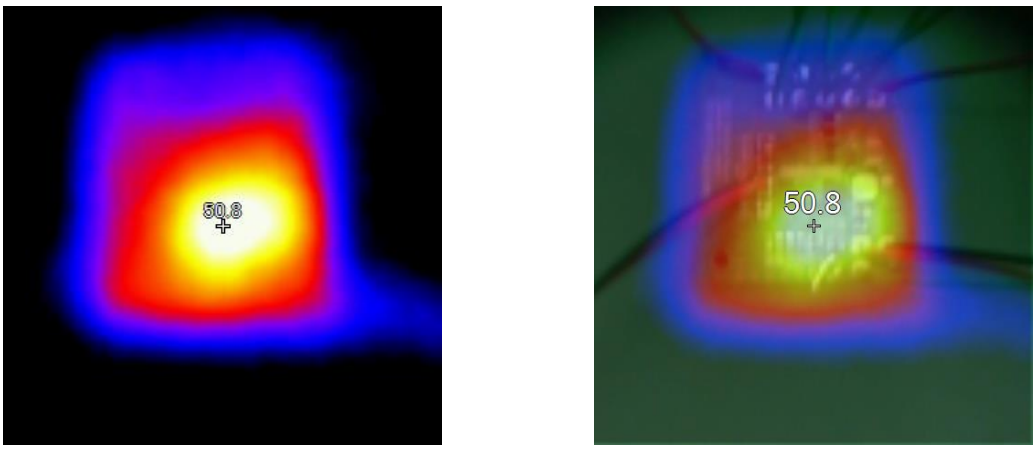
(a) Power Loss (b) Efficiency

Figure 7.6: Efficiency Measurements (Error bars indicate measurement uncertainty)

These losses are much larger than initially simulated in LTspice. This is in part because a different inductor had to be used for the prototype. However, the most significant difference is most likely caused by the dissipation in the control IC. This is not specified in detail in the datasheet and is probably dependent on the input voltage (hence causing the slope in the power loss to disappear). The MATLAB model was updated to include an additional dissipation estimate from the datasheet and LTspice simulations (approx. 280 mW for analog circuits and current sources) for the figures.

The efficiency can still be improved somewhat from the value measured here. The inductor used for the prototype was a replacement (LPS4018-103 instead of XGL4030-123) with 50–100 mW higher losses. If the correct inductor is used, an efficiency around 85–87 % can be expected.

Thermal Performance The thermal performance of the converter was verified using a Fluke VT04 thermal camera. Although this device does not offer very good spatial resolution, the global temperature of the board could be measured (Figure 7.7). The prototype was first run for 5 min and verified to have reached thermal steady state.



(a) Infrared Data (b) Photograph with IR Overlay

Figure 7.7: Thermal Performance Measurement (V_i = 21 V)

The measured temperature rise is approximately 25.8 K (ambient temperature $T_A = 25\,°\mathrm{C}$) and seems to be mostly concentrated around the control IC and inductor. Given the losses in these com-

ponents, the temperature rise is reasonable. No significant difference in temperature was measured for different input voltages.

7.4. Conclusion

In this chapter, the design and verification of the prototype are presented. A prototype PCB of the LED driver is designed, as well as an LED array based on normal color LEDs to act as a load. Additionally, a 3D model of the complete Wireless Powerlizer system is made.

The prototype is tested according to the procedure presented in Appendix E. These tests have shown that all functionality of the driver works as expected. The performance measurements have shown that the efficiency of the driver is approximately 83 %, which is lower than expected. The most likely cause of this is additional losses in the control IC which were not accounted for in the design. These are not described in detail in the datasheet. However, the thermal performance is still acceptable.

8

Conclusion and Recommendations

In this thesis, the design of a portable UV-C sterilizer for personal items such as mobile phones, keys, and face masks has been presented. The design consists mainly of a UV-C LED array and an LED driver.

8.1. Conclusion

A radiometric model of the irradiance by the LED array was developed and applied in an optimization procedure to find the configuration with the lowest cost and heat dissipation, sufficient irradiation to meet the specifications, and the highest efficiency. This resulted in an LED array of 24 UV-C LEDs arranged in a 4×6 rectangular pattern.

The LED driver was designed based on a simple analytic loss model of the boost converter topology. With this model, the LED driver could be designed to be as small and efficient as possible. Finally, a DCM boost converter operating at 750 kHz was designed. The converter efficiency was estimated to be 91 % at the nominal output power of 4.3 W. The volume of the LED driver was approximately 20 mm $\times$ 25 mm $\times$ 5 mm.

Finally, a prototype of the Sterilizer subsystem was designed and built. With this prototype, the functionality and performance of the designed LED driver were verified. All designed functionality worked as expected, and the prototype showed an efficiency of 83 % with some potential for improvement.

The designed Sterilizer complies with the requirements presented in Chapter 2. The compliance is evaluated at the end of each chapter and summarized here:

The design complies with all safety and functional requirements, as far as these could be tested on the prototype or ensured by design. Because the LED array was not built for the prototype, requirement [**F.5**] could not be evaluated. From the magnetic field simulations in [2], the expectation is that the LED array will be able to function properly if the loop created by the supply wires is small. In Chapter 2, two key performance indicators were introduced: (i) The dose delivered to the disinfection area, and (ii) the size and efficiency of the LED driver. Adequate results have been obtained for both KPIs.

The system-level requirements have mostly been satisfied by the subsystems as far as could be designed and tested. The design presented in this complies with system requirements relevant to the UV-C Sterilizer system (functional requirements [**SF.3**], [**SF.4**], and [**SO.1**]; and safety requirements [**S.1**] and [**SS.3**]). Electromagnetic compatibility (EMC) with the other subsystems, and conducted and radiated EMI have not been validated.

8.2. Recommendations for Future Work

Although a large part of the design of the Sterilizer has been presented and verified in this thesis, several things must still be done before it can be accepted as an actual product. These recommendations are briefly presented and discussed in the following paragraphs.

LED Array In future work, the LED array should be built and properly tested. The UV-C output performance should be analyzed by determining the irradiation dose, uniformity, and disinfection level. The thermal performance of the UV-C LEDs should also be verified.

System Integration The top-level controller should be further developed and tested, together with the two other parts of the Wireless Powerlizer: the Battery Management System and the Wireless Charging unit. When the full Wireless Powerlizer is assembled, the electromagnetic compatibility of the Sterilizer with the Wireless Charging unit should be tested.

Certification If the intention is to bring the Wireless Powerlizer to the market, the relevant safety standards and regulations for the CE marking (for selling on the European market) should be investigated. This design complies with these standards as best as possible, but because of the limited availability of these standards, this could not be done in detail.

UV-C LED Technology UV-C LEDs are an upcoming technology, meaning that many improvements are expected in the near future. While their efficiency is quite low at this point in time, more efficient LEDs have already been designed and will soon be available on the market. The driver designed in this project can easily deliver 5 W continuously at high efficiency, however, the LEDs dissipate a lot of this power ($\eta_{LED} \approx 2.6\,\%$). When more efficient LEDs are used, fewer LEDs can be used to complete a full disinfection cycle in less time.

A

Definitions and Abbreviations

A.1. List of Physical Constants

Symbol	Units	Description	Value
c	[m/s]	Speed of light	299 792 458 m/s
h	[J/Hz]	Planck constant	$6.626\,070\,15 \times 10^{-34}$ J/Hz
$\hbar$	[J/Hz]	Reduced Planck constant	$1.054\,571\,817 \times 10^{-34}$ J/Hz
q	[C]	Elementary charge	$1.602\,176\,634 \times 10^{-19}$ C

A.2. List of Symbols

Note that these symbols may be accompanied by superscripts or subscripts which are not indicated here.

Symbol	Units	Description
A	[m^2]	Area
B	[T]	Magnetic flux density
C	[F]	Capacitance
D	[-], [%]	Duty cycle
D	[J/m^2]	Irradiation dose
D_{90}	[mJ/cm^2]	Log reduction dose
E_e	[W/m^2]	Irradiance
E_g	[eV]	Band gap energy
f	[Hz]	Frequency
f_{sw}	[Hz]	Switching frequency
I	[A]	Current
I_e	[W/sr]	Radiant intensity
I_F	[A]	Forward current
$I_{rel}(\theta, \phi)$	[-]	Relative radiant intensity
K	[-]	Measure for the tendency to operate in DCM [48]
L	[H]	Inductance
$M(D, K)$	[-]	Converter conversion factor
O	[-]	Overdose
P	[W]	Power
Q	[C]	Electric charge
r	[m]	Distance between two points
R	[Ω]	Resistance
R_θ	[K/W]	Thermal resistance
t	[s]	Time
T	[°C], [K]	Temperature
T_{sw}	[s]	Switching period

Symbol	Units	Description
V	[V]	Voltage
V_F	[V]	Forward voltage
$\boldsymbol{x}$	[-]	Optimization variable(s)
Γ	[-], [%]	Reflection coefficient
δ	[-], [%]	Duty cycle (diode)
Δi_L	[A]	Inductor current ripple
Δv_C	[V]	Capacitor voltage ripple
η	[-], [%]	Efficiency
θ	[°], [rad]	Elevation angle
Θ	[°]	Beam divergence angle
λ	[m]	Wavelength
σ	[-]	Standard deviation
Φ_e	[W]	Radiant flux
ω	[rad/s]	Angular frequency
Ω	[sr]	Solid angle

A.3. List of Abbreviations

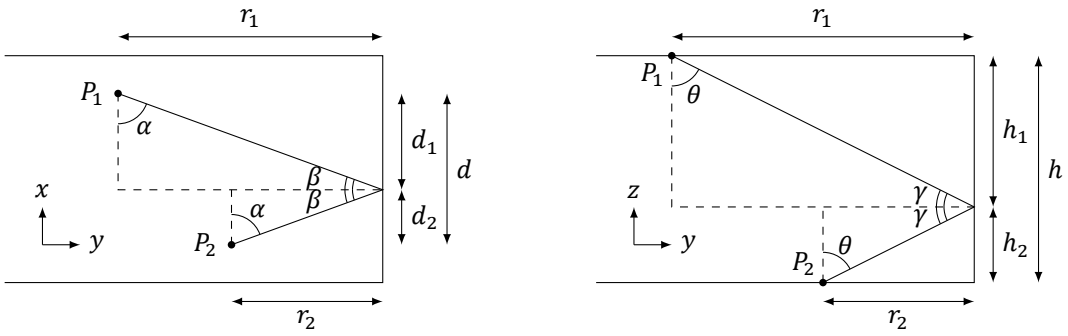
Abbreviation	Description
ACMC	Average current-mode control
BCM	Boundary conduction mode (boundary between CCM and DCM)
BMS	Battery management system
CCM	Continuous conduction mode
COVID-19	Coronavirus disease 2019 (SARS-CoV-2)
DC	Direct current
DCM	Discontinuous conduction mode
DNA	Deoxyribonucleic acid
EMC	Electromagnetic compatibility
EMI	Electromagnetic interference
EQE	External quantum efficiency
ESR	Equivalent series resistance
HF	High frequency
hMPV	Human Metapneumovirus
hOPV	Human Orthopneumovirus (RSV)
IC	Integrated circuit
IQE	Internal quantum efficiency
LED	Light-emitting diode
LEE	Light extraction efficiency
MCU	Microcontroller unit
MLCC	Multi-layer ceramic capacitor
MOSFET	Metal–oxide–semiconductor field-effect transistor
PCB	Printed circuit board
PCMC	Peak current-mode control
PID	Proportional–integral–derivative [controller]
PWM	Pulse-width modulation
RMS	Root mean square
RNA	Ribonucleic acid
RSV	Respiratory Syncytial Virus
SMBus	System Management Bus
spp.	Multiple species (Lat. *species pluralis*)
USB	Universal Serial Bus
UV	Ultraviolet
UVGI	Ultraviolet germicidal irradiation
WPE	Wall plug efficiency
ZCS	Zero current switching

B

Derivations

B.1. LED Array Irradiance Reflections

To calculate the reflections some basic trigonometry is needed. The setup is presented in Figure B.1. It shows the box-shaped disinfection area from two sides (in the xy and yz planes). The points P_1 and P_2 represent the source and receiver, respectively.



(a) In the xy plane

(b) In the yz plane

Figure B.1: Derivation of LED Array Reflections

Equations (B.1) and (B.2) can be derived from Figure B.1a using trigonometric identities.

$$\tan\beta = \frac{d_1}{r_1} = \frac{d_2}{r_2} \tag{B.1}$$

$$d = d_1 + d_2 \tag{B.2}$$

Using these two equations, the following formulas for d_1 and d_2 can be found:

$$d_1 = \frac{r_1}{r_1 + r_2} d \tag{B.3}$$

$$d_2 = \frac{r_2}{r_1 + r_2} d \tag{B.4}$$

Using the same derivation, the values for h_1 and h_2 are found from Figure B.1b as shown in (B.5) and (B.6).

$$h_1 = \frac{r_1}{r_1 + r_2} h \tag{B.5}$$

$$h_2 = \frac{r_2}{r_1 + r_2} h \tag{B.6}$$

The angle of incidence, θ is found according to (B.7) using Figure B.1b together with (B.5).

$$\tan\theta = \frac{r_1}{h_1} = \frac{r_1 + r_2}{h} \tag{B.7}$$

Finally, the total distance the UV wave has traveled, r_{tot}, is found by adding both the distance from the LED to the reflection point and the distance from the reflection point to the observation point, as shown in (B.8).

$$r_{\text{tot}} = \sqrt{{d_1}^2 + {r_1}^2 + {h_1}^2} + \sqrt{{d_2}^2 + {r_2}^2 + {h_2}^2} \tag{B.8}$$

B.2. Boost Converter

B.2.1. Discontinuous Conduction Mode Condition

In DCM, the current ripple must be larger than twice the average inductor current.

$$\Delta i_L > 2I_{av} \tag{B.9}$$

Filling in the definitions of Δi_L and I_{av} yields the following.

$$\frac{V_i D T_{sw}}{L} > \frac{2I_o}{1-D} \tag{B.10}$$

This can be written in terms of $K = {}^{2L}/_{RT_{sw}}$, which is a measure of the tendency of a converter to operate in the discontinuous conduction mode.

$$\frac{2LV_iI_o}{V_iDT_{sw}(1-D)V_o} < 1 - D \tag{B.11}$$

$$\frac{2LI_o}{V_oT_{sw}} < D(1-D)^2 \tag{B.12}$$

The critical factor is the value of K at the boundary between CCM and DCM: $K_{crit} = D(1-D)^2$.

B.2.2. Discontinuous Conduction Mode Conversion Ratio

The conversion ratio can be be derived from the inductor volt-second balance. Here, D is the duty cycle of the switch, and δ is the duty cycle of the diode.

$$\frac{V_i}{L}DT_{sw} + \frac{V_i - V_o}{L}\delta T_{sw} = 0 \implies \frac{V_o}{V_i} = \frac{D+\delta}{\delta} \tag{B.13}$$

However, δ is still unknown. To find it, the output current can be related to the diode current waveform.

$$I_o = \langle i_D \rangle = \frac{1}{T_{sw}}\int_0^{T_{sw}} i_D(t)dt = \frac{\delta}{2}\frac{V_iDT_{sw}}{L} \tag{B.14}$$

Combining these two results, the following second-order polynomial in V_o is obtained.

$$V_o^2 - V_oV_i - \frac{V_i^2D^2}{K} = 0 \tag{B.15}$$

The conversion ratio follows from the quadratic formula. Only the positive solution is considered, since (B.13) must be positive for $D, \delta > 0$.

$$\frac{V_o}{V_i} = \frac{1 + \sqrt{1 + \frac{4D^2}{K}}}{2} \tag{B.16}$$

B.2.3. Continuous Conduction Mode RMS Currents

Using the definition of the root-mean-square (RMS), the RMS currents through the components can be calculated from the switch and diode current waveforms. Here I_{min} is defined as $I_{min} = I_{av} - \frac{1}{2}\Delta i_L$.

$$I_{S,rms} = \sqrt{\left(I_{min}^2 + I_{min}\Delta i_L + \frac{1}{3}\Delta i_L^2\right)D} \tag{B.17}$$

$$I_{D,rms} = \sqrt{\left(I_{min}^2 + I_{min}\Delta i_L + \frac{1}{3}\Delta i_L^2\right)(1-D)} \tag{B.18}$$

The RMS inductor current is calculated by combining the switch and diode current.

$$I_{L,rms} = \sqrt{I_{S,rms}^2 + I_{D,rms}^2} = \sqrt{I_{min}^2 + I_{min}\Delta i_L + \frac{1}{3}\Delta i_L^2} \tag{B.19}$$

Finally, the average diode current is the converter output current.

$$I_{D,avg} = I_{L,avg}(1-D) = I_o \tag{B.20}$$

B.2.4. Discontinuous Conduction Mode RMS Currents

For DCM, the RMS currents can be calculated in the same as above. Because $I_{min} = 0$ in DCM (per definition), the RMS currents become as follows.

$$I_{S,rms} = \sqrt{\frac{D}{3}}\Delta i_L \tag{B.21}$$

$$I_{D,rms} = \sqrt{\frac{\delta}{3}}\Delta i_L \tag{B.22}$$

$$I_{L,rms} = \sqrt{\frac{D+\delta}{3}}\Delta i_L \tag{B.23}$$

The average diode current must still be equal to the converter output current to ensure the steady-state conditions for the output capacitor holds.

$$I_{D,avg} = \frac{1}{2}\Delta i_L \delta = I_o \tag{B.24}$$

B.2.5. Discontinuous Conduction Mode Voltage Ripple

From the steady-state condition for the capacitor it can be derived that the integral of the capacitor current over one cycle must be zero.

$$\int_0^{T_{sw}} i_C(t)dt = 0 \tag{B.25}$$

The voltage ripple can be calculated from either the positive or the negative area of the capacitor current waveform, since these two must be equal. The area gives the charge difference ΔQ.

$$\Delta v_o = \frac{\Delta Q}{C} = \frac{1}{2C}(\Delta i_L - I_o)t_2 = \frac{(\Delta i_L - I_o)^2\delta}{2C\Delta i_L f_{sw}} \tag{B.26}$$

Here, t_2 is the time for which the waveform is positive. Filling in the definition of δ, the following equation for the output voltage ripple is found.

$$\Delta v_o = \frac{(\Delta i_L - I_o)^2}{C}\frac{I_o L}{\Delta i_L V_i D} \tag{B.27}$$

This shows that the voltage ripple in DCM can be very different than what is predicted by the equations for CCM. Differences up to 40 % have been calculated for the different operating points of the LED driver.

B.3. Prototype Measurement Uncertainty

Because of the low power involved, it is desirable to evaluate the uncertainty in the efficiency measurements of the prototype. For these measurements, the input voltage and current are measured with two multimeters. The uncertainty of the multimeters is derived from the datasheet and given in Table B.1.

Table B.1: Multimeter Accuracy

Measurement	Range	Resolution	Uncertainty
Voltage	60.00 V	0.01 V	$\pm(0.9\,\% + 0.04\,\mathrm{V})$
Current	600.0 mA	0.1 mA	$\pm(1.0\,\% + 0.7\,\mathrm{mA})$

Using the uncertainties from the table, the standard deviation of the voltage and current measurements (σ_V and σ_I) can be calculated. The power can then be calculated according to $P = VI$, which will have an uncertainty σ_P dependent on σ_V and σ_I. The dependency is given in (B.28) [62].

$$\sigma_P \approx VI\sqrt{\left(\frac{\sigma_V}{V}\right)^2 + \left(\frac{\sigma_I}{I}\right)^2 + 2\frac{\sigma_{VI}}{VI}} \tag{B.28}$$

To calculate the uncertainty of the efficiency, the uncertainty of the input and output power (σ_{P_i} and σ_{P_o}) as calculated above must be used.

$$\sigma_\eta \approx \frac{P_o}{P_i}\sqrt{\left(\frac{\sigma_{P_o}}{P_o}\right)^2 + \left(\frac{\sigma_{P_i}}{P_i}\right)^2 - 2\frac{\sigma_{P_oP_i}}{P_oP_i}} \tag{B.29}$$

For these measurements, it will be assumed that the measurements are independent, and hence the covariances σ_{VI} and $\sigma_{P_oP_i}$ are zero.

C

Figures

In this appendix, several figures that were not included in the main body text are presented with some accompanying explanatory text.

C.1. LED Array

In Figure C.1, the full result of the LED array layout optimization is presented. This figure shows several interesting trends: (i) The minimum irradiance is linearly dependent on the number of LEDs and does not seem to depend much on the LED pattern. (ii) The overdose is also linearly dependent on the number of LEDs, but the *rectangle* pattern performs much better than the other patterns.

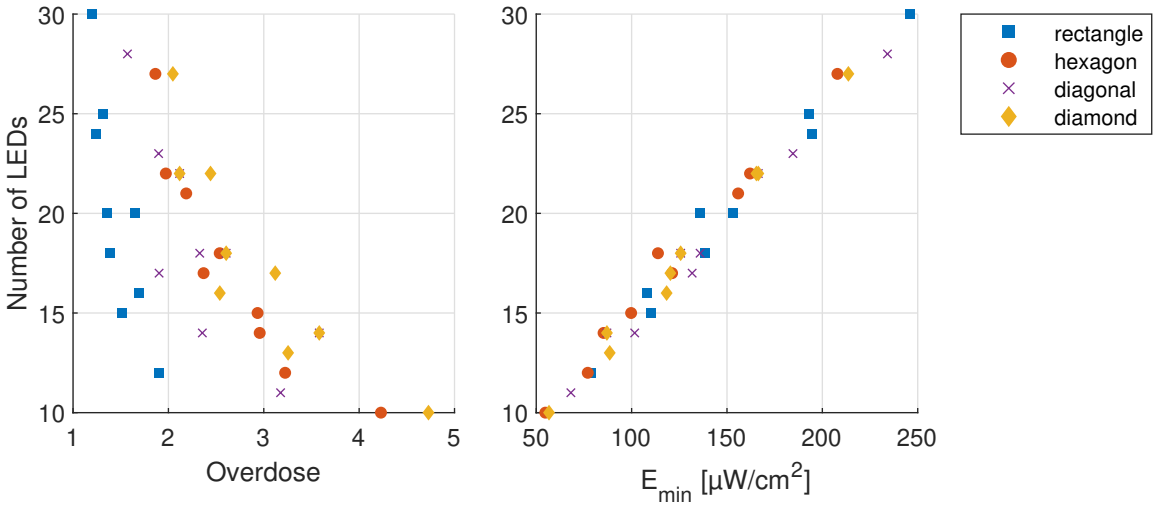


Figure C.1: Layout Optimization Results

C.2. LED Configuration

In Figure C.2, the possible electrical configurations (i.e., the number of parallel strings and series LEDs per string) are presented. The total forward voltage per string is plotted on the vertical axis, with the error bars indicating the range of possible values.

The red areas indicate that these solutions would not comply with the specifications or are otherwise undesirable: (i) The voltages in the Powerlizer should be <42 V according to [**SS.3**]. (ii) It is undesirable to have a voltage close to the battery voltage (12.5–21 V) as this necessitates a more complex converter topology.

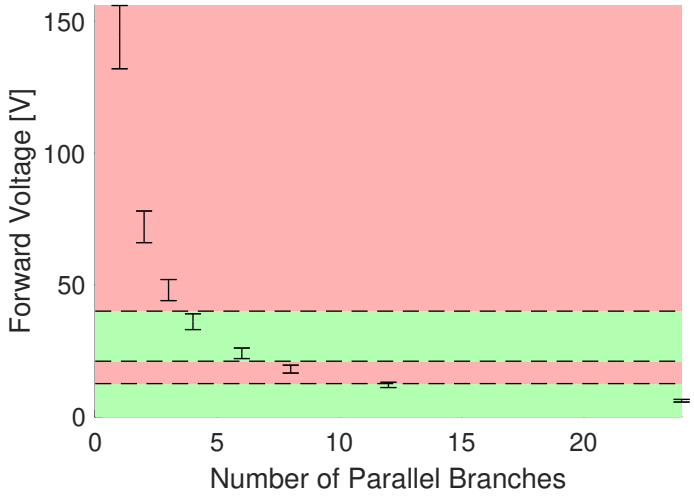


Figure C.2: Possible Electrical Configurations with 24 LEDs

C.3. LED Driver

In Figures C.3 and C.4, the power loss of the designed boost converter is plotted versus the switching frequency. These plots are used to make a trade-off between switching frequency and power loss.

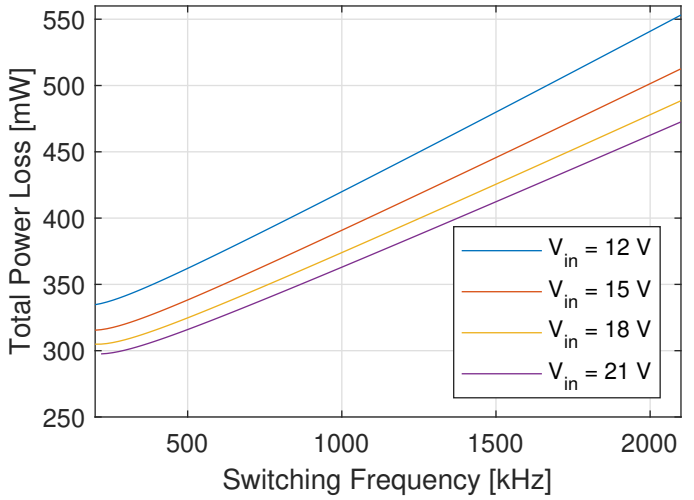


Figure C.3: Total Converter Power Loss vs Switching Frequency (CCM Configuration)

Note that the switching frequency in DCM cannot be too high, otherwise converter operation will revert to CCM, causing a large increase in losses (since there is no ZCS anymore). These operating points are not shown in Figure C.4.

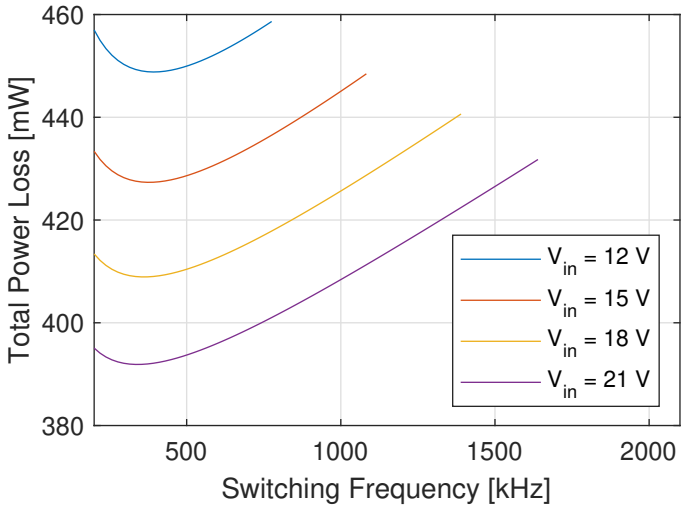


Figure C.4: Total Converter Power Loss vs Switching Frequency (DCM Configuration)

C.3.1. Shutdown

In Figure C.5, a transient simulation of the driver shutdown functionality is shown. The simulation is performed at $V_i = 21\,\text{V}$ and with 300 nH of parasitic inductance in each LED string.

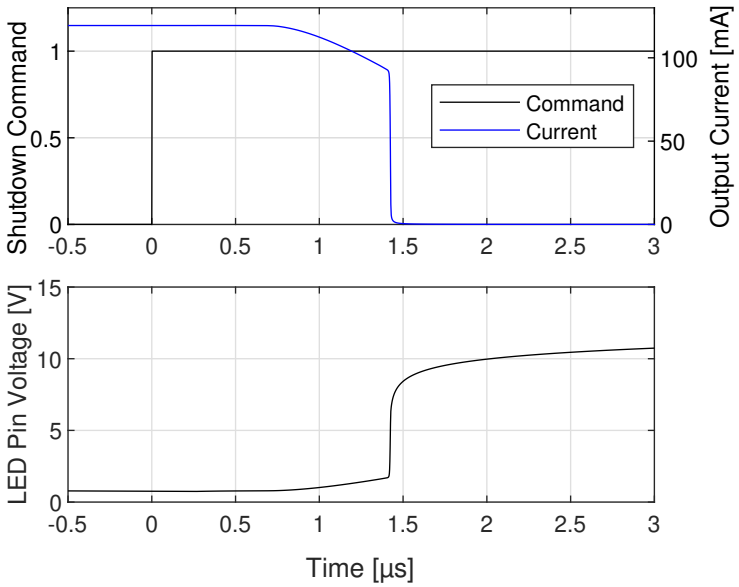
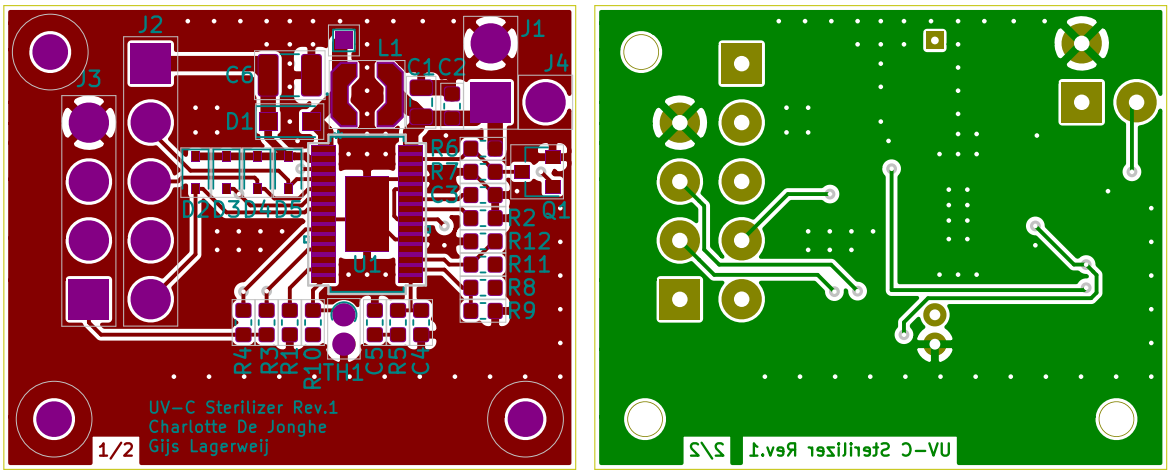


Figure C.5: Transient Simulation of LED Driver Shutdown ($L_\sigma = 300\,\text{nH}$ per LED string)

D

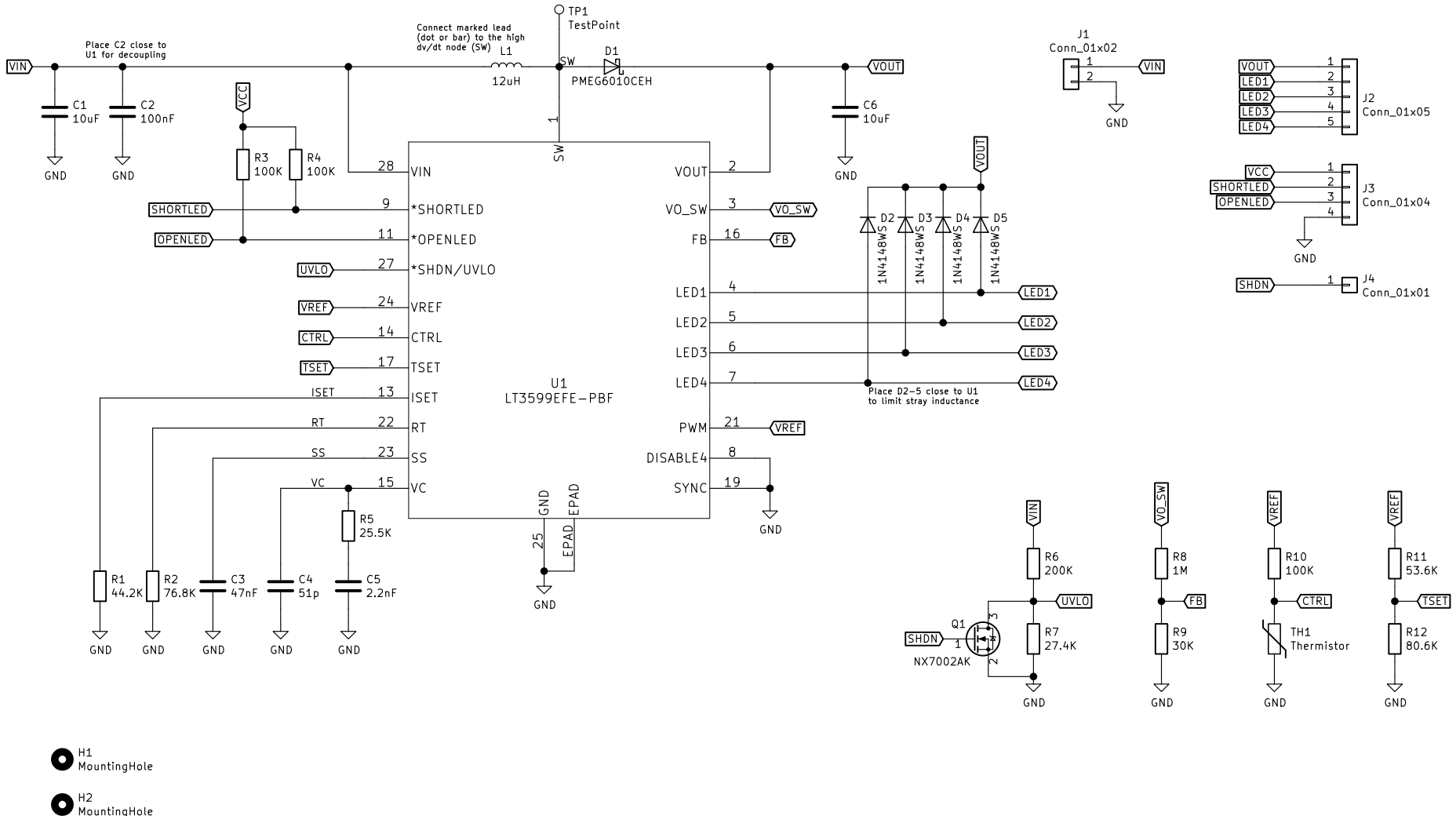
Schematics



(a) Front

(b) Back

Figure D.1: Prototype PCB Layout

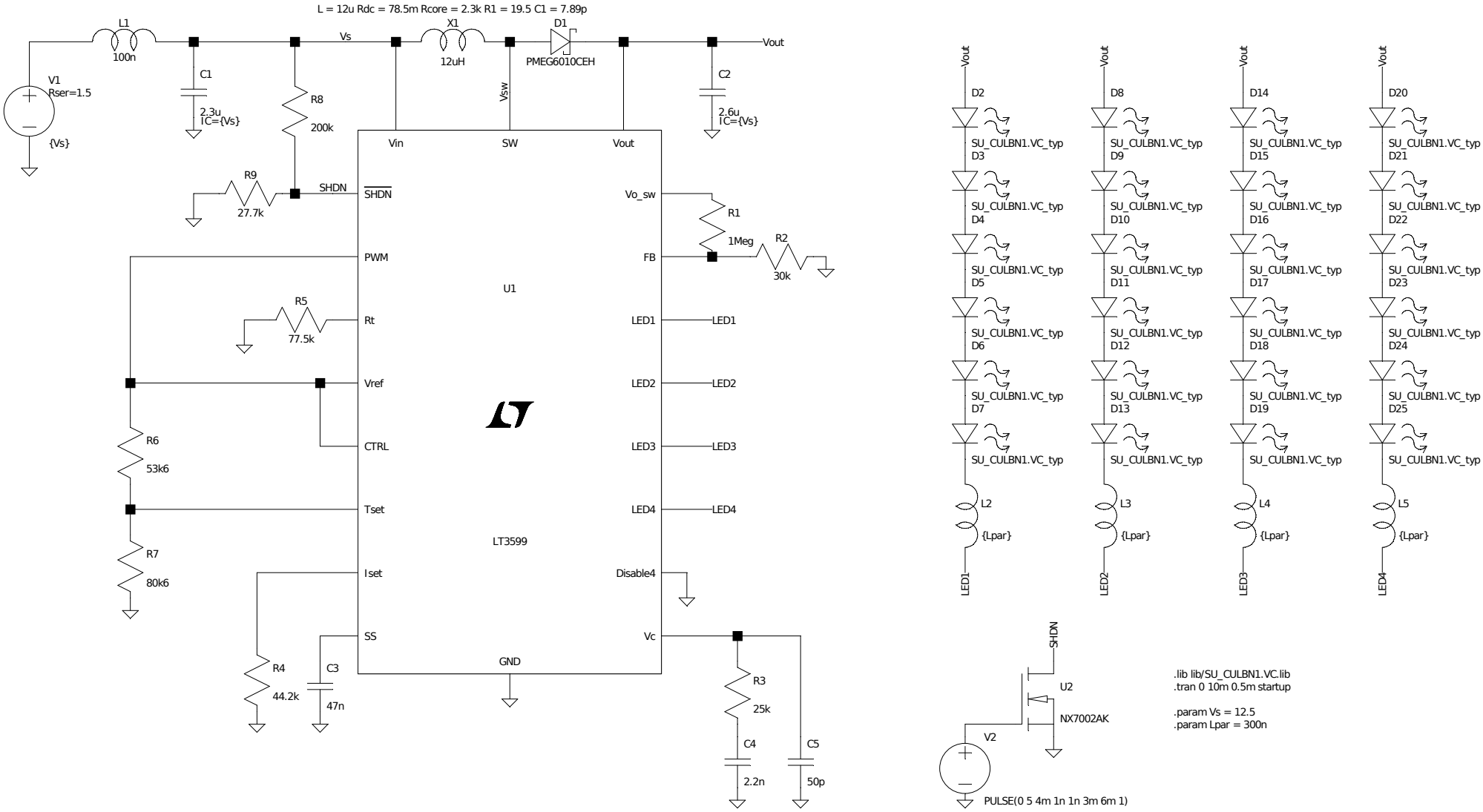


TU Delft

Sheet: /
File: sterilizer_proto_Rev2.sch

Wireless Powerlizer: UV-C Sterilizer Prototype

Size: A4	Date: 2021-06-09	Rev: 2
KiCad E.D.A. kicad (5.1.4)-1		Id: 1/1


L = 12u Rdc = 78.5m Rcore = 2.3k R1 = 19.5 C1 = 7.89p
L1
100n
Vs
X1
12uH
D1
PMEG6010CEH
Vout
V1
Rser=1.5
{Vs}
C1
2.3u
IC={Vs}
R8
200k
Vsw
C2
2.6u
IC={Vs}
Vin
SW
Vout
R9
27.7k
SHDN
SHDN
Vo_sw
R1
1Meg
R2
30k
PWM
FB
U1
R5
77.5k
Rt
LED1
LED1
Vref
LED2
LED2
R6
53k6
CTRL
LED3
LED3
Tset
LED4
LED4
LT3599
R7
80k6
Iset
Disable4
SS
Vc
GND
R4
44.2k
C3
47n
R3
25k
C4
2.2n
C5
50p
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D21
D22
D23
D24
D25
SU_CULBN1.VC_typ
L2
L3
L4
L5
{Lpar}
LED1
LED2
LED3
LED4
SHDN
U2
NX7002AK
V2
PULSE(0 5 4m 1n 1n 3m 6m 1)
.lib lib/SU_CULBN1.VC.lib
.tran 0 10m 0.5m startup
.param Vs = 12.5
.param Lpar = 300n

E

Test Procedure

In this chapter, the test procedure for the Sterilizer prototype, as discussed in Chapter 7, will be presented. First, the functional tests should be performed to verify that the prototype functions as expected. Finally, the performance measurements should give insight into the quality of the designed LED driver. Reference designators refer to the schematic presented in Appendix D

E.1. Functional

E.1.1. Startup

Purpose	Verify the startup behaviour of the LED driver.
Requirements	The LED driver will start up when the input voltage is $\geq$12 V and the SHDN signal (J4) is low. During the startup transient, the output voltage and current should not overshoot the final value.
Procedure	1) Connect the load to J2. 2) Apply a voltage $\geq$3 V to J4. 3) Connect the voltage supply to J1 and set it to 12.5–21 V. 4) Turn on the LED driver by setting the voltage on J4 to 0 V and measure the output voltage and current on J2.

E.1.2. Inductor Current Waveform

Purpose	The inductor current waveform should be measured to ensure that the converter behaves as designed and simulated.
Requirements	None
Procedure	1) Start the LED driver as described in Procedure E.1.1. 2) Measure the input current with a current clamp around the supply wire. If no input capacitance is present, this will equal the inductor current.

E.1.3. Switch Voltage Waveform

Purpose	The switch voltage waveforms should be measured to ensure that the converter behaves as designed and simulated.
Requirements	None
Procedure	1) Start the LED driver as described in Procedure E.1.1. 2) Measure the switch voltage at the switch-node test point (TP1).

E.1.4. Shutdown

Purpose	Verify the shutdown behaviour of the LED driver.
Requirements	The driver shall turn off the LEDs within 10 µs if the SHDN signal is asserted.
Procedure	1) Start the LED driver as described in Procedure E.1.1. 2) Increase the voltage on J4 to $\geq$3 V. 3) Measure the output current on J2. 4) Measure the voltage rise on the LED channels (LED1-4).

E.1.5. Analog Dimming

Purpose	Verify the analog dimming capability of the LED driver IC for use in the LED temperature protection.
Requirements	The LED current should be adjustable through a variable resistor. In the final product, this resistor shall be a thermistor measuring the LED case temperature.
Procedure	1) Connect a 500 kΩ potentiometer to TH1 and set it to the maximum resistance. 2) Start the LED driver as described in Procedure E.1.1. 3) Vary the potentiometer from its maximum to its minimum value and verify that the load current drops to almost zero at the minimum.

E.2. Performance

E.2.1. Voltage Ripple

Purpose	Measure the input and output voltage ripple of the LED driver.
Requirements	The output voltage ripple shall be $<$50 mV.
Procedure	1) Start the LED driver as described in Procedure E.1.1. 2) Measure the input voltage ripple on J1. 3) Measure the output voltage ripple on J2.

E.2.2. Efficiency

Purpose	Measure the efficiency of the LED driver.
Requirements	None
Procedure	1) Start the LED driver as described in Procedure E.1.1. 2) Measure the input voltage and current on J1. To ensure that the DC input voltage and current are measured, a large input capacitor can be added. 3) Measure the output voltage and current on J2.

E.2.3. Thermal Performance

Purpose	Measure the temperature rise of several critical components.
Requirements	None
Procedure	1) Start the LED driver as described in Procedure E.1.1. 2) Measure the temperature of the following components using the thermal camera or similar device. a) Driver IC (U1) b) Inductor (L1) c) Schottky diode (D1)

F

Listings

The code presented in this chapter is available at https://gitlab.com/gwlagerweij/powerlizer.

F.1. LED Array Simulation

F.1.1. LED Position Generation

```
% F_GEN_LED_POSITIONS Generate LED position vector
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   29/04/2021
%
%  [pos, N] = f_gen_led_positions(Ncol, Nrow, W, H, dW, dH, method);
%
% Inputs:
% ------------------------------------------------------------------------
% | Name   | Description                          | Unit  | Size  | Type   |
% ========================================================================
% | Ncol   | Number of columns                    | [-]   | 1     | Double |
% ------------------------------------------------------------------------
% | Nrow   | Number of rows                       | [-]   | 1     | Double |
% ------------------------------------------------------------------------
% | W      | Width                                | [m]   | 1     | Double |
% ------------------------------------------------------------------------
% | H      | Height                               | [m]   | 1     | Double |
% ------------------------------------------------------------------------
% | dW     | Distance from edge (along x-axis)    | [m]   | 1     | Double |
% ------------------------------------------------------------------------
% | dH     | Distance from edge (along y-axis)    | [m]   | 1     | Double |
% ------------------------------------------------------------------------
% | method | Generation method                    | [-]   | 1 x M | Char   |
% |        |  rectangle, hexagon,                 |       |       |        |
% |        |  diagonal, diamond                   |       |       |        |
% ------------------------------------------------------------------------
%
% Outputs:
% ------------------------------------------------------------------------
% | Name   | Description                          | Unit  | Size  | Type   |
% ========================================================================
% | pos    | Generated positions                  | [m]   | 3 x N | Double |
% ------------------------------------------------------------------------
% | N      | Number of generated positions        | [-]   | 1     | Double |
```

```
35 % ------------------------------------------------------------------------
36 function [pos, N] = f_gen_led_positions(Ncol, Nrow, W, H, dW, dH, method)
37     % Select correct generation function based on the selected method
38     if(strcmpi(method, 'rectangle'))
39         [pos, N] = f_gen_led_pos_rectangle(Ncol, Nrow, W, H, dW, dH);
40     elseif(strcmpi(method, 'diagonal'))
41         [pos, N] = f_gen_led_pos_diagonal(Ncol, Nrow, W, H, dW, dH);
42     elseif(strcmpi(method, 'diamond'))
43         [pos, N] = f_gen_led_pos_diamond(Ncol, Nrow, W, H, dW, dH);
44     elseif(strcmpi(method, 'hexagon'))
45         [pos, N] = f_gen_led_pos_hexagon(Ncol, Nrow, W, H, dW, dH);
46     else
47         warning('Unknown generation method: %s.\nReturning empty position
               vector.', method);
48         pos = [];
49         N   = 0;
50     end
51 end
52
53 % Generate positions in a rectangle pattern
54 %   *   *   *
55 %   *   *   *
56 %   *   *   *
57 function [pos, N] = f_gen_led_pos_rectangle(Ncol, Nrow, W, H, dW, dH)
58     N = Ncol * Nrow;
59     pos = nan(3, N);
60     for i = 1:Ncol
61         for j = 1:Nrow
62             pos(1, (i - 1) * Nrow + j) = (i - 1) * (W - 2 * dW) / (Ncol -
                   1) + dW;
63             pos(2, (i - 1) * Nrow + j) = (j - 1) * (H - 2 * dH) / (Nrow -
                   1) + dH;
64             pos(3, (i - 1) * Nrow + j) = 0;
65         end
66     end
67 end
68
69 % Generate positions in a diagonal pattern
70 %   *   *   *
71 %     *   *
72 %   *   *   *
73 function [pos, N] = f_gen_led_pos_diagonal(Ncol, Nrow, W, H, dW, dH)
74     pos = [];
75     for i = 1:Ncol
76         if(mod(i, 2) == 1) % Generate odd columns like 'rectangle'
77             for j = 1:Nrow
78                 x0 = (i - 1) * (W - 2 * dW) / (Ncol - 1) + dW;
79                 y0 = (j - 1) * (H - 2 * dH) / (Nrow - 1) + dH;
80                 z0 = 0;
81                 pos = [pos, [x0; y0; z0]];
82             end
83         else                % Generate even columns with an offset
84             for j = 1:Nrow - 1
85                 x0 = (i - 1) * (W - 2 * dW) / (Ncol - 1) + dW;
86                 y0 = (j - 0.5) * (H - 2 * dH) / (Nrow - 1) + dH;
87                 z0 = 0;
```

```
                pos = [pos, [x0; y0; z0]];
            end
        end
    end
    N = size(pos, 2);
end

% Generate positions in a diamond pattern
%     *   *   *
%   *   *   *   *
%     *   *   *
function [pos, N] = f_gen_led_pos_diamond(Ncol, Nrow, W, H, dW, dH)
    pos = [];
    for i = 1:Ncol
        if(mod(i, 2) == 0) % Generate even columns like 'rectangle'
            for j = 1:Nrow
                x0 = (i - 1) * (W - 2 * dW) / (Ncol - 1) + dW;
                y0 = (j - 1) * (H - 2 * dH) / (Nrow - 1) + dH;
                z0 = 0;
                pos = [pos, [x0; y0; z0]];
            end
        else                % Generate odd columns with an offset
            for j = 1:Nrow - 1
                x0 = (i - 1) * (W - 2 * dW) / (Ncol - 1) + dW;
                y0 = (j - 0.5) * (H - 2 * dH) / (Nrow - 1) + dH;
                z0 = 0;
                pos = [pos, [x0; y0; z0]];
            end
        end
    end
    N = size(pos, 2);
end

% Generate positions in a hexagon pattern
%     *   *   *
%   *   *   *   *
%     *   *   *
%   *   *   *   *
%     *   *   *
function [pos, N] = f_gen_led_pos_hexagon(Ncol, Nrow, W, H, dW, dH)
    pos = [];
    for j = 1:Nrow
        if(mod(j, 2) == 0) % Generate odd rows like 'rectangle'
            for i = 1:Ncol
                x0 = (i - 1) * (W - 2 * dW) / (Ncol - 1) + dW;
                y0 = (j - 1) * (H - 2 * dH) / (Nrow - 1) + dH;
                z0 = 0;
                pos = [pos, [x0; y0; z0]];
            end
        else                % Generate even rows with an offset
            for i = 1:Ncol - 1
                x0 = (i - 0.5) * (W - 2 * dW) / (Ncol - 1) + dW;
                y0 = (j - 1) * (H - 2 * dH) / (Nrow - 1) + dH;
                z0 = 0;
                pos = [pos, [x0; y0; z0]];
            end
```

```
        end
    end
    N = size(pos, 2);
end
```

F.1.2. LED Radiation Pattern

```
% F_GEN_RAD_PATTERN Generate LED radiation pattern
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   01/05/2021
%
%  [Irel, Phi_e, Omega] = f_gen_rad_pattern(led_name);
%
% Inputs:
% ------------------------------------------------------------------------
% | Name     | Description                       | Unit  | Size  | Type   |
% ========================================================================
% | led_name | Name of the LED                   | [-]   | 1 x N | Char   |
% ------------------------------------------------------------------------
%
% Outputs:
% ------------------------------------------------------------------------
% | Name     | Description                       | Unit  | Size  | Type   |
% ========================================================================
% | Irel     | Relative radiant intensity        | [-]   | 1     | Handle |
% ------------------------------------------------------------------------
% | Phi_e    | Radiant flux                      | [W]   | 1     | Double |
% ------------------------------------------------------------------------
% | Omega    | Solid angle                       | [sr]  | 1     | Double |
% ------------------------------------------------------------------------
function [Irel, Phi_e, Omega] = f_gen_rad_pattern(led_name)
    % Load LED parameters
    load(sprintf('leds/%s.mat', led_name), 'Irel_pat', 'theta_pat', 'Phi_e
        ');

    % Create handle to calculate relative radiant intensity
    Irel = @(theta) (f_calc_irel(Irel_pat, theta_pat, theta));

    % Calculate solid angle
    theta = linspace(-pi / 2, pi / 2, 1000);
    Omega = 2 * pi * trapz(theta, Irel(theta) .* cos(theta));
end

% F_CALC_IREL Function to be used as handle
%  Calculates relative radiant intensity at angle theta
%   from a set of data points extracted from the datasheet.
function Irel = f_calc_irel(Irel_pat, theta_pat, theta)
    % Interpolate datasheet values
    Irel = interp1(theta_pat, Irel_pat, abs(theta));

    % If no value is available for Irel at the given theta, return zero
    % This can occur if theta is outside the range of theta_pat
    Irel(isnan(Irel)) = 0;
end
```

F.1.3. Reflection Calculation

```
% F_CALC_REFLECTION Calculate angle and travel distance of reflection
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   30/04/2021
%
%  [theta, r] = f_calc_reflection(pos1, pos2, wall, coord);
%
% Inputs:
% ------------------------------------------------------------------------
% | Name   | Description                  | Unit   | Size      | Type   |
% ========================================================================
% | pos1   | Source position              | [m]    | 1 x 3     | Double |
% ------------------------------------------------------------------------
% | pos2   | Target position              | [m]    | N x M x 3 | Double |
% ------------------------------------------------------------------------
% | wall   | Wall type ('x' or 'y')       | [-]    | 1         | Char   |
% ------------------------------------------------------------------------
% | coord  | Wall coordinate              | [m]    | 1         | Double |
% ------------------------------------------------------------------------
%
% Outputs:
% ------------------------------------------------------------------------
% | Name   | Description                  | Unit   | Size      | Type   |
% ========================================================================
% | theta  | Incidence angle              | [rad]  | N x M     | Double |
% ------------------------------------------------------------------------
% | r      | Travel distance              | [m]    | N x M     | Double |
% ------------------------------------------------------------------------
function [theta, r] = f_calc_reflection(pos1, pos2, wall, coord)
    % Calculate distance to wall perpendicular to wall axis
    if(strcmpi(wall, 'x'))
        r1 = abs(pos1(2) - coord);
        r2 = abs(pos2(:, :, 2) - coord);
        dx = abs(pos1(1) - pos2(:, :, 1));
    elseif(strcmpi(wall, 'y'))
        r1 = abs(pos1(1) - coord);
        r2 = abs(pos2(:, :, 1) - coord);
        dx = abs(pos1(2) - pos2(:, :, 2));
    else
        error('Invalid wall: %s', wall);
    end

    % Calculate distance along wall axis and z-axis
    dz = abs(pos1(3) - pos2(:, :, 3));

    % Solve for unknown distances
    r0 = r1 + r2;
    d1 = dx .* r1 ./ r0;
    d2 = dx .* r2 ./ r0;
    h1 = dz .* r1 ./ r0;
    h2 = dz .* r2 ./ r0;

    % Calculate incidence angle and total travel distance
    theta = pi / 2 - atan(dz ./ r0);
    r = sqrt(d1.^2 + h1.^2 + r1.^2) + sqrt(d2.^2 + h2.^2 + r2.^2);
end
```

F.1.4. Irradiance Calculation

```
% F_CALC_IRRADIANCE Calculate irradiance on a rectangular domain
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   30/04/2021
%
%  [x, y, E, O] = f_calc_irradiance(setup);
%
% Inputs:
% ------------------------------------------------------------------------
% | Name   | Description                       | Unit    | Size  | Type   |
% ========================================================================
% | setup  | Structure describing the problem | [-]     | 1     | Struct |
% ------------------------------------------------------------------------
%
% Outputs:
% ------------------------------------------------------------------------
% | Name   | Description                       | Unit    | Size  | Type   |
% ========================================================================
% | x      | x coordinate array                | [m]     | 1 x N | Double |
% ------------------------------------------------------------------------
% | y      | y coordinate array                | [m]     | 1 x N | Double |
% ------------------------------------------------------------------------
% | E      | Irradiance                        | [W/m^2] | N x N | Double |
% ------------------------------------------------------------------------
% | O      | Overdose                          | [-]     | 1     | Double |
% ------------------------------------------------------------------------
function [x, y, E, O] = f_calc_irradiance(setup)
    % Grid elements
    x = linspace(0, setup.W, setup.Nel);
    y = linspace(0, setup.H, setup.Nel)';
    z = setup.z;

    % Element position matrix
    el_pos(:,:,1) = repmat(x, size(y));
    el_pos(:,:,2) = repmat(y, size(x));
    el_pos(:,:,3) = z * ones(length(y), length(x));
    src_pos       = setup.src_pos;

    % Wall definition
    wall = [{'x', 0}; {'x', setup.H}; {'y', 0}; {'y', setup.W}];

    % Calculate irradiance
    E = zeros(setup.Nel);
    for i = 1:setup.N
        src = src_pos(:, i);

        % Direct incidence
        r = sqrt((x - src(1)).^2 + (y - src(2)).^2 + (z - src(3)).^2);
        theta = acos((z - src(3)) ./ r);
        E = E + setup.Phi_e * setup.Irel(theta) .* cos(theta) ./ (r.^2 *
            setup.Omega);

        % First reflection
```

```
        for j = 1:size(wall, 1)
            [theta, r] = f_calc_reflection(src, el_pos, wall{j, 1}, wall{j
                , 2});
            E = E + setup.Gamma * setup.Phi_e * setup.Irel(theta) .* cos(
                theta) ./ (r.^2 * setup.Omega);
        end
    end

    % Calculate overdose
    O = max(max(E)) / min(min(E));
end
```

F.1.5. Pattern Optimization

```
% F_OPTIMIZE_PATTERN Optimize LED pattern for a given Ncol, Nrow
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   04/05/2021
%
%  [dW, dH, Emin] = f_optimize_pattern(setup, Ncol, Nrow, pat);        (1)
%  [dW, dH, Emin] = f_optimize_pattern(setup, Ncol, Nrow, pat, plot);  (2)
%
% Inputs:
% ------------------------------------------------------------------------
% | Name    | Description                          | Unit  | Size  | Type   |
% ========================================================================
% | setup   | Structure describing the problem     | [-]   | 1     | Struct |
% ------------------------------------------------------------------------
% | Ncol    | Number of columns                    | [-]   | 1     | Double |
% ------------------------------------------------------------------------
% | Nrow    | Number of rows                       | [-]   | 1     | Double |
% ------------------------------------------------------------------------
% | pat     | Pattern name                         | [-]   | 1 x N | Char   |
% ------------------------------------------------------------------------
% | plot    | Show optimization plot               | [-]   | 1     | Bool   |
% ------------------------------------------------------------------------
%
% Outputs:
% ------------------------------------------------------------------------
% | Name    | Description                          | Unit  | Size  | Type   |
% ========================================================================
% | dW      | Distance from wall (x)               | [m]   | 1     | Double |
% ------------------------------------------------------------------------
% | dH      | Distance from wall (y)               | [m]   | 1     | Double |
% ------------------------------------------------------------------------
% | Emin    | Maximized irradiance                 | [-]   | 1     | Double |
% ------------------------------------------------------------------------
function [dW, dH, Emin] = f_optimize_pattern(setup, Ncol, Nrow, pat, plot)
    % Show debug information if plot == 1
    if(nargin == 5 && plot == 1)
        options = optimset('Display', 'iter', 'PlotFcns', @optimplotfval);
    else
        options = optimset('Display', 'off');
    end

    % Multi-variable optimization with fminsearch
    f = @(x) (objfunc(x, setup, Ncol, Nrow, pat));
```

```
    [x, fval] = fminsearch(f, [1.5e-2, 1.5e-2], options);

    % Optimization results
    dW   = x(1); dH = x(2);
    Emin = -fval;
end

% OBJFUNC Objective function to minimize
function [v] = objfunc(x, setup, Ncol, Nrow, pattern)
    % Generate pattern with optimization variables x = [dW, dH]
    [setup.src_pos, setup.N] = f_gen_led_positions(Ncol, Nrow, setup.W,
        setup.H, x(1), x(2), pattern);

    % Calculate irradiance for the generated pattern
    %  and return the overdose
    [~, ~, E, ~] = f_calc_irradiance(setup);
    v = -min(min(E));
end
```

F.1.6. Simulation Script

```
% IRRADIANCE Simulate and plot irradiance
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   30/04/2021
%
%% Parameters
setup.W   = 10e-2;          % Width of disinfection area
setup.H   = 17.5e-2;        % Height of disinfection area
setup.z   = 2e-2;           % Distance of LEDs to disinfection area
setup.Nel = 200;            % Number of grid elements

setup.Gamma = 0.9;          % Wall reflectivity

led_name = 'SUCULBN1_V2'; % Name of the LED
pattern  = 'rectangle';   % LED placement pattern

Ncol = 4;                   % Number of columns of LEDs
Nrow = 6;                   % Number of rows of LEDs
dW   = 1.4e-2;              % x distance from wall
dH   = 1.35e-2;             % y distance from wall

%% Calculate irradiance
[setup.src_pos, setup.N] = f_gen_led_positions(Ncol, Nrow, setup.W, setup.
    H, dW, dH, pattern);
[setup.Irel, setup.Phi_e, setup.Omega] = f_gen_rad_pattern(led_name);
[x, y, E, O] = f_calc_irradiance(setup);
Emin = min(min(E)) * 0.1;

%% Plot
figure(1); clf

% Plot irradiance
s = surface(y * 1e2, x * 1e2, E' * 1e-1);
s.EdgeAlpha = 0;
%contourf(y * 1e2, x * 1e2, E' * 1e-1);
xlabel('y [cm]')
```

```
ylabel('x [cm]')

% Scale colour [0, Emax]
c = gca;
c.CLim = [0, max(max(E)) * 0.1];

cb = colorbar;
cb.Label.String = 'Irradiance [mW/cm^2]';

% Draw LEDs
hold on
Wled = 0.36; % Width [cm]
Hled = 0.36; % Height [cm]
for i = 1:length(setup.src_pos)
    x_ = setup.src_pos(1, i) * 1e2;
    y_ = setup.src_pos(2, i) * 1e2;
    plot3([y_ - 0.5 * Hled, y_ - 0.5 * Hled, y_ + 0.5 * Hled, y_ + 0.5 * Hled, y_ - 0.5 * Hled, y_ - 0.5 * Hled],...
          [x_ - 0.5 * Wled, x_ + 0.5 * Wled, x_ + 0.5 * Wled, x_ - 0.5 * Wled, x_ - 0.5 * Wled, x_ + 0.5 * Wled],...
          [5, 5, 5, 5, 5, 5], 'k');
end
axis equal
xlim([0, setup.H] * 1e2)
ylim([0, setup.W] * 1e2)

box on
```

F.1.7. Optimization Script

```
% OPTIMIZE_LAYOUT Optimize LED array layout
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   04/05/2021
%
%% Parameters
setup.W   = 10e-2;        % Width of disinfection area
setup.H   = 17.5e-2;      % Height of disinfection area
setup.z   = 2e-2;         %
setup.Nel = 100;          % Number of grid elements

setup.Gamma = 0.9;        % Wall reflectivity

led_name = 'SUCULBN1_V2'; % Name of the LED

%%
pattern = {'rectangle', 'hexagon', 'diagonal', 'diamond'};
Ncol    = [3, 4, 5];
Nrow    = [4, 5, 6];

%% Generate LED radiation pattern
[setup.Irel, setup.Phi_e, setup.Omega] = f_gen_rad_pattern(led_name);

%% Optimize solutions
data = [];
data_idx = [];
for i = 1:length(pattern)
```

```
    fprintf('Starting pattern %s\n', pattern{i});
    for j = 1:length(Ncol)
        for k = 1:length(Nrow)
            [dW, dH, Emin] = f_optimize_pattern(setup, Ncol(j), Nrow(k),
               pattern{i});

            data = [data; {pattern(i)}, Ncol(j), Nrow(k), Emin, dW, dH];
            data_idx = [data_idx; i];
        end
    end
end

%% Put data in table
headers = {'pattern', 'Ncol', 'Nrow', 'Emin', 'dW', 'dH'};
T = cell2table(data);
T.Properties.VariableNames = headers;
```

F.1.8. FEMM Thermal Simulation

```
-- Parameters
W = 10;               -- Width [cm]
H = 17.5;             -- Length [cm]

Wled = 0.36;          -- Width of LED [cm]
Hled = 0.18;          -- Height of LED [cm]
Rled = 24;            -- LED thermal resistance [K/W]
Ta   = 45 + 273.13;   -- Ambient temperature [K]

 -- Calculate effective k for LED [W/(m K)]
kled = Hled / (Rled * Wled * H) * 100 * 6;

 -- Calculate effective LED power [W]
eta = 2.6;            -- Efficiency [%]
P   = 0.18;           -- Total power [W]
led_power = (100 - eta) / 100 * P * 6;

-- Problem definition
newdocument(2);
hi_probdef("centimeters", "planar", 1e-8, H, 30);

Nlayers   = 4;
thickness = {35e-4, 0.157, 35e-4, 0.1};
width     = {W, W, W, W};
mat       = {"Copper", "FR4", "Copper", "Teflon"};
cond      = {nil, nil, nil, nil};

-- Material definition
hi_addmaterial("Copper", 401, 401, 0, 3.41);
hi_addtkpoint("Copper", 273, 401)
hi_addtkpoint("Copper", 473, 389)
hi_addtkpoint("Copper", 673, 378)
hi_addtkpoint("Copper", 873, 366)
hi_addtkpoint("Copper", 1073, 352)
hi_addtkpoint("Copper", 1273, 336)

hi_addmaterial("FR4", 0.9, 0.3, 0, 0);
```

```
hi_addmaterial("LED", kled, kled, 0, 0);
hi_addmaterial("Teflon", 0.35, 0.35, 0, 1.8);

hi_addconductorprop("LED1", 0, led_power, 0);
hi_addconductorprop("LED2", 0, led_power, 0);
hi_addconductorprop("LED3", 0, led_power, 0);
hi_addconductorprop("LED4", 0, led_power, 0);
hi_addboundprop("convection", 2, 0, 0, Ta, 15, 0);

-- Draw the geometry
curY = 0;
for i = 1, Nlayers do
        W = width[i];
        H = thickness[i];
        M = mat[i];
        C = cond[i];
        nextY = curY - H;

        hi_addnode(-W/2, curY);
        hi_addnode(W/2, curY);
        hi_addnode(-W/2, nextY);
        hi_addnode(W/2, nextY);

        hi_addsegment(-W/2, curY, W/2, curY);
        hi_addsegment(W/2, curY, W/2, nextY);
        hi_addsegment(W/2, nextY, -W/2, nextY);
        hi_addsegment(-W/2, nextY, -W/2, curY);

        hi_addblocklabel(0, (curY + nextY) / 2);
        hi_selectlabel(0, (curY + nextY) / 2);
        hi_setblockprop(M, 1, 0, 0);

        if(C ~= nil) then
                hi_selectsegment(-W/2, curY, W/2, curY);
                hi_setsegmentprop("", 0, 1, 0, 0, C);
        end

        curY = nextY;
end

hi_selectsegment(-W/2, curY, W/2, curY);
hi_setsegmentprop("convection", 0, 1, 0, 0, "");

-- Draw LEDs
dW = 1.4; -- Distance from border
leds = {"LED1", "LED2", "LED3", "LED4"};
for i = 1, 4 do
        x = -W/2 + dW + (i - 1) * (W - 2 * dW) / 3

        hi_addnode(x - Wled / 2, 0);
        hi_addnode(x + Wled / 2, 0);
        hi_addnode(x - Wled / 2, Hled);
        hi_addnode(x + Wled / 2, Hled);

        hi_addsegment(x - Wled / 2, 0, x + Wled / 2, 0)
        hi_addsegment(x + Wled / 2, 0, x + Wled / 2, Hled)
```

```
        hi_addsegment(x - Wled / 2, Hled, x + Wled / 2, Hled)
        hi_addsegment(x - Wled / 2, 0, x - Wled / 2, Hled)

        hi_addblocklabel(x, Hled / 2);
        hi_selectlabel(x, Hled / 2);
        hi_setblockprop("LED", 1, 0, 0);

        hi_selectsegment(x - Wled / 2, Hled, x + Wled / 2, Hled);
        hi_setsegmentprop("", 0, 1, 0, 0, leds[i]);
end

-- Set up view
hi_clearselected();
hi_zoomnatural();

-- Save and analyze
hi_saveas("led_heat.feh");
hi_analyze();

-- Load solution
hi_loadsolution();
```

F.2. LED Driver Model MATLAB

F.2.1. Operating Point Calculation

```
% F_CALC_OP Calculate boost converter operating point
%   from input and output power, inductor value and switching frequency.
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   13/05/2021
%
%  [Iav, Iripp, D, d, DCM] = f_calc_op(Vi, Vo, Io, L, fsw, eta);
%
% Inputs:
% ---------------------------------------------------------------------
% | Name   | Description                       | Unit    | Size  | Type   |
% =====================================================================
% | Vi     | Input voltage                     | [V]     | N x M | Double |
% ---------------------------------------------------------------------
% | Vo     | Output voltage                    | [V]     | N x M | Double |
% ---------------------------------------------------------------------
% | Io     | Output current                    | [A]     | N x M | Double |
% ---------------------------------------------------------------------
% | L      | Inductance                        | [H]     | N x M | Double |
% ---------------------------------------------------------------------
% | fsw    | Switching frequency               | [Hz]    | N x M | Double |
% ---------------------------------------------------------------------
% | eta    | Efficiency estimate               | [-]     | 1     | Double |
% ---------------------------------------------------------------------
%
% Outputs:
% ---------------------------------------------------------------------
% | Name   | Description                       | Unit    | Size  | Type   |
% =====================================================================
% | Iav    | Average inductor current          | [A]     | N x M | Double |
% ---------------------------------------------------------------------
```

```
% | Iripp   | Inductor current ripple          | [A]     | N x M | Double |
% ------------------------------------------------------------------------
% | D       | Switch duty cycle                | [-]     | N x M | Double |
% ------------------------------------------------------------------------
% | d       | Diode duty cycle                 | [-]     | N x M | Double |
% ------------------------------------------------------------------------
% | DCM     | Operation in DCM mode            | [-]     | N x M | Logic  |
% ------------------------------------------------------------------------
function [Iav, Iripp, D, d, DCM] = f_calc_op(Vi, Vo, Io, L, fsw, eta)
    D   = 1 - eta .* Vi ./ Vo;              % CCM duty cycle
    Iav = Io ./ (1 - D);                    % Average inductor current

    K     = 2 * L .* fsw ./ (Vo ./ Io);     % K factor
    Kcrit = D .* (1 - D).^2;                % Critical K for boost
        converter

    DCM = K < Kcrit;                        % Check if operation in DCM
    D   = D .* (1 - DCM)...                 % Duty cycle
        + sqrt(K ./ eta .* (Vo ./ Vi - 1) .* Vo ./ Vi) .* DCM;
    Iripp = Vi .* D ./ (L .* fsw);          % Ripple current
    d     = (1 - D) .* (1 - DCM)...         % Diode duty cycle
          + Iripp .* L .* fsw ./ (Vo - Vi) .* DCM;
end
```

F.2.2. RMS Current Calculation

```
% F_CALC_RMS Calculate RMS and average component currents
%   from operating point information.
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   13/05/2021
%
%  [Isrms, Idrms, Ilrms, Idavg] = f_calc_rms(Iav, Iripp, D, d, mode);
%
% Inputs:
% ------------------------------------------------------------------------
% | Name    | Description                    | Unit    | Size  | Type   |
% ========================================================================
% | Iav     | Average inductor current       | [A]     | N x M | Double |
% ------------------------------------------------------------------------
% | Iripp   | Inductor current ripple        | [A]     | N x M | Double |
% ------------------------------------------------------------------------
% | D       | Switch duty cycle              | [-]     | N x M | Double |
% ------------------------------------------------------------------------
% | d       | Diode duty cycle               | [-]     | N x M | Double |
% ------------------------------------------------------------------------
% | mode    | Operation in DCM mode          | [-]     | N x M | Logic  |
% ------------------------------------------------------------------------
%
% Outputs:
% ------------------------------------------------------------------------
% | Name    | Description                    | Unit    | Size  | Type   |
% ========================================================================
% | Isrms   | RMS switch current             | [A]     | N x M | Double |
% ------------------------------------------------------------------------
% | Idrms   | RMS diode current              | [A]     | N x M | Double |
% ------------------------------------------------------------------------
```

```
% | Ilrms  | RMS inductor current              | [A]     | N x M | Double |
% ------------------------------------------------------------------------
% | Idavg  | Average diode current             | [A]     | N x M | Double |
% ------------------------------------------------------------------------
function [Isrms, Idrms, Ilrms, Idavg] = f_calc_rms(Iav, Iripp, D, d, mode)
    % Continuous conduction mode
    Imin      = Iav - 0.5 * Iripp;
    Isrms_ccm = sqrt((Imin.^2 + Imin.*Iripp + Iripp.^2 / 3) .* D);
    Idrms_ccm = sqrt((Imin.^2 + Imin.*Iripp + Iripp.^2 / 3) .* (1 - D));
    Idavg_ccm = Iav .* (1 - D);

    % Discontinuous conduction mode
    Isrms_dcm = sqrt(D / 3) .* Iripp;
    Idrms_dcm = sqrt(d / 3) .* Iripp;
    Idavg_dcm = 0.5 * Iripp .* d;

    % Select correct mode
    Isrms = Isrms_ccm .* (1 - mode) + Isrms_dcm .* mode;
    Idrms = Idrms_ccm .* (1 - mode) + Idrms_dcm .* mode;
    Idavg = Idavg_ccm .* (1 - mode) + Idavg_dcm .* mode;
    Ilrms = sqrt(Idrms.^2 + Isrms.^2);
end
```

F.2.3. Loss Calculation

```
% F_CALC_LOSS Calculate power loss of the converter
%   Uses either an efficiency estimate or will iteratively calculate loss
%   until it converges to an efficiency.
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   14/05/2021
%
%  [Ploss, Pd, Ps, PLcond, PLcore, Pssw, mode]
%    = f_calc_loss(Vi, Vo, Io, fsw, inductor, diode, switch_);          (1)
%  [Ploss, Pd, Ps, PLcond, PLcore, Pssw, mode]
%    = f_calc_loss(Vi, Vo, Io, fsw, inductor, diode, switch_, eta);     (2)
%
% Inputs:
% ------------------------------------------------------------------------
% | Name     | Description                    | Unit    | Size  | Type   |
% ========================================================================
% | Vi       | Input voltage                  | [V]     | N x M | Double |
% ------------------------------------------------------------------------
% | Vo       | Output voltage                 | [V]     | N x M | Double |
% ------------------------------------------------------------------------
% | Io       | Output current                 | [A]     | N x M | Double |
% ------------------------------------------------------------------------
% | fsw      | Switching frequency            | [Hz]    | N x M | Double |
% ------------------------------------------------------------------------
% | inductor | Inductor name                  | [-]     |       | Char   |
% ------------------------------------------------------------------------
% | diode    | Diode name                     | [-]     |       | Char   |
% ------------------------------------------------------------------------
% | switch_  | Switch name                    | [-]     |       | Char   |
% ------------------------------------------------------------------------
% | eta      | Efficiency estimate            | [-]     | 1     | Double |
% ------------------------------------------------------------------------
```

```
%
% Outputs:
% ------------------------------------------------------------------------
% | Name     | Description                    | Unit    | Size  | Type   |
% ========================================================================
% | Ploss    | Total power loss               | [W]     | N x M | Double |
% ------------------------------------------------------------------------
% | Pd       | Power loss in the diode        | [W]     | N x M | Double |
% ------------------------------------------------------------------------
% | Ps       | Power loss in the switch       | [W]     | N x M | Double |
% ------------------------------------------------------------------------
% | PLcond   | Inductor copper loss           | [W]     | N x M | Double |
% ------------------------------------------------------------------------
% | PLcore   | Inductor core loss             | [W]     | N x M | Double |
% ------------------------------------------------------------------------
% | Pssw     | Switching loss in the switch   | [W]     | N x M | Double |
% ------------------------------------------------------------------------
% | mode     | Converter operating mode       | [-]     | N x M | Logic  |
% ------------------------------------------------------------------------
function [Ploss, Pd, Ps, PLcond, PLcore, Pssw, mode] = f_calc_loss(Vi, Vo,
    Io, fsw, inductor, diode, switch_, eta)
    if(nargin == 7) % Calculate iteratively
        Niter = 0;
        eta  = 1;          % Initial efficiency estimate
        while(1)
            [Ploss, Pd, Ps, PLcond, PLcore, Pssw, mode] = f_calc_loss(Vi,
               Vo, Io, fsw, inductor, diode, switch_, eta);

            Po = Vo .* Io;
            Pi = Po + Ploss;

            eta_  = eta;      % Save previous estimate
            eta   = Po ./ Pi; % Calculate efficiency from losses
            Niter = Niter + 1;
            if(all(all(abs(eta - eta_) < 0.1/100)) || Niter > 10)
                break;
            end
        end

    elseif(nargin == 8) % Calculate using efficiency estimate
        [Ploss, Pd, Ps, PLcond, PLcore, Pssw, mode] = calc_loss(Vi, Vo, Io
           , fsw, inductor, diode, switch_, eta);
    else
        error('Incorrect number of input arguments')
    end
end

function [Ploss, Pd, Ps, PLcond, PLcore, Pssw, mode] = calc_loss(Vi, Vo,
   Io, fsw, inductor, diode, switch_, eta)
    %% Load component data
    [Rson, tson, tsoff, Cso] = f_load_switch(switch_);
    [Rdon, Vdsat, Cdo]       = f_load_diode(diode);
    [L, RLdc, Rcore0, Rac0]        = f_load_inductor(inductor);
    RLcore = Rcore0 * sqrt(fsw);    % Frequency-dependent core loss model
    Rac    = Rac0 * sqrt(fsw);

```

```
    %% Calculate operating point & rms currents
    [Iav, Iripp, D, d, mode]     = f_calc_op(Vi, Vo, Io, L, fsw, eta);
    [Isrms, Idrms, Ilrms, Idavg] = f_calc_rms(Iav, Iripp, D, d, mode);

    %% Conduction losses
    Pdcond = Rdon * Idrms.^2 + Vdsat .* Idavg;
    Pscond = Rson * Isrms.^2;

    PLcond = (RLdc + Rac) .* Ilrms.^2;
    PLcore = (Vi.^2 .* D + (Vi - Vo).^2 .* d) ./ RLcore;

    %% Switching losses
    Vsoff = Vo;                  % Switch turn-off voltage
    Vson  = Vo .* (1 - mode)... % Switch turn-on voltage CCM
            + Vi .* mode;        % Switch turn-on voltage DCM
    Isoff = Iav + 0.5 * Iripp;  % Switch turn-off current
    Ison  = (Iav - 0.5 * Iripp) .* (1 - mode)... % Switch turn-on current CCM
            + (0) .* mode;                       % Switch turn-on current DCM

    % Reverse recovery not modeled: Schottky diodes do not have this
    % Gate charging loss not modeled: no parameters available from datasheet
    Pssw_ton  = 0.5 * Vson .* Ison .* tson .* fsw;
    Pssw_toff = 0.5 * Vsoff .* Isoff .* tsoff .* fsw;
    Pssw_co = 0.5 * (Cdo + Cso) * Vo.^2 .* fsw .* (1 - mode)...
           + (0.5 * (Cdo * (Vo - Vi).^2 + Cso * Vi.^2) .* fsw) .* mode;
    Pssw = Pssw_ton + Pssw_toff + Pssw_co;

    %% IC losses
    Pic = 0.0 * Io;     % Balancing

    %% Total losses
    Ploss = Pdcond + Pscond + PLcond + PLcore + Pic + Pssw;
    Ps = Pscond + Pssw;
    Pd = Pdcond;
end
```

F.2.4. Component Database

```
% F_LOAD_DIODE Load diode model from component database
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   14/05/2021
%
%  [Ron, Vsat, Co] = f_load_diode(name);
%
% Inputs:
% ------------------------------------------------------------------------
% | Name   | Description                       | Unit    | Size  | Type   |
% ========================================================================
% | name   | Component name                    | [-]     | 1 x N | Char   |
% ------------------------------------------------------------------------
%
% Outputs:
% ------------------------------------------------------------------------
```

```
% | Name   | Description                        | Unit    | Size  | Type   |
% ============================================================================
% | Ron    | On-state resistance                | [Ohm]   | 1     | Double |
% ----------------------------------------------------------------------------
% | Vsat   | On-state saturation voltage        | [V]     | 1     | Double |
% ----------------------------------------------------------------------------
% | Co     | Output capacitance                 | [F]     | 1     | Double |
% ----------------------------------------------------------------------------
function [Ron, Vsat, Co] = f_load_diode(name)
    data = load(sprintf('components/diodes/%s.mat', name));
    Ron  = data.Ron;
    Vsat = data.Vsat;
    Co   = data.Co;
end
```

```
% F_LOAD_INDUCTOR Load inductor model from component database
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   14/05/2021
%
%  [L, Rdc, Rcore0, Rac0] = f_load_inductor(name)
%
% Inputs:
% ----------------------------------------------------------------------------
% | Name   | Description                        | Unit    | Size  | Type   |
% ============================================================================
% | name   | Component name                     | [-]     | 1 x N | Char   |
% ----------------------------------------------------------------------------
%
% Outputs:
% ----------------------------------------------------------------------------
% | Name   | Description                      | Unit       | Size  | Type   |
% ============================================================================
% | L      | Inductance                       | [H]        | 1     | Double |
% ----------------------------------------------------------------------------
% | Rdc    | DC winding resistance            | [Ohm]      | 1     | Double |
% ----------------------------------------------------------------------------
% | Rcore0 | Core loss model coefficient      | [Ohm/rtHz] | 1     | Double |
% ----------------------------------------------------------------------------
% | Rac0   | AC resistance model coeff        | [Ohm/rtHz] | 1     | Double |
% ----------------------------------------------------------------------------
function [L, Rdc, Rcore0, Rac0] = f_load_inductor(name)
    data = load(sprintf('components/inductors/%s.mat', name));
    L      = data.L;
    Rdc    = data.Rdc;
    Rcore0 = data.Rcore0;
    Rac0   = data.Rac0;
end
```

```
% F_LOAD_SWITCH Load switch model from component database
%  Author: Gijs Lagerweij, Charlotte De Jonghe
%  Date:   19/05/2021
%
%  [Rson, tson, tsoff, Cso] = f_load_switch(name)
%
% Inputs:
% ----------------------------------------------------------------------------
```

```
% | Name    | Description                          | Unit     | Size   | Type    |
% ================================================================================
% | name    | Component name                       | [-]      | 1 x N  | Char    |
% --------------------------------------------------------------------------------
%
% Outputs:
% --------------------------------------------------------------------------------
% | Name    | Description                          | Unit     | Size   | Type    |
% ================================================================================
% | Rson    | On-state resistance                  | [Ohm]    | 1      | Double  |
% --------------------------------------------------------------------------------
% | tson    | Turn-on time                         | [s]      | 1      | Double  |
% --------------------------------------------------------------------------------
% | tsoff   | Turn-off time                        | [s]      | 1      | Double  |
% --------------------------------------------------------------------------------
% | Cso     | Output capacitance                   | [F]      | 1      | Double  |
% --------------------------------------------------------------------------------
function [Rson, tson, tsoff, Cso] = f_load_switch(name)
    data = load(sprintf('components/switches/%s.mat', name));
    Rson  = data.Rson;
    tson  = data.tson;
    tsoff = data.tsoff;
    Cso   = data.Cso;
end
```

F.3. LED Driver Model LTspice

```
* D:\Users\gijsw\Documents\TU\projects\BAP\ltspice\LT3599_driver.asc
XU1 LED1 LED2 LED3 LED4 0 MP_01 MP_02 N031 N011 N033 N012 N030 MP_03 MP_04
    N011 N017 N032 N011 MP_05 SHDN Vs Vsw Vout N006 0 LT3599
R1 N006 N012 1Meg
R2 0 N012 30k
C3 N032 0 47n
R4 N031 0 44.2k
R5 N017 0 77.5k
R6 N011 N030 53k6
R7 N030 0 80k6
R8 Vs SHDN 200k
R9 SHDN 0 27.7k
V1 N001 0 {Vs} Rser=1.5
D2 Vout N002 SU_CULBN1.VC_typ
D3 N002 N007 SU_CULBN1.VC_typ
D4 N007 N013 SU_CULBN1.VC_typ
D5 N013 N018 SU_CULBN1.VC_typ
D6 N018 N022 SU_CULBN1.VC_typ
D7 N022 N026 SU_CULBN1.VC_typ
D8 Vout N003 SU_CULBN1.VC_typ
D9 N003 N008 SU_CULBN1.VC_typ
D10 N008 N014 SU_CULBN1.VC_typ
D11 N014 N019 SU_CULBN1.VC_typ
D12 N019 N023 SU_CULBN1.VC_typ
D13 N023 N027 SU_CULBN1.VC_typ
D14 Vout N004 SU_CULBN1.VC_typ
D15 N004 N009 SU_CULBN1.VC_typ
D16 N009 N015 SU_CULBN1.VC_typ
```

```
D17 N015 N020 SU_CULBN1.VC_typ
D18 N020 N024 SU_CULBN1.VC_typ
D19 N024 N028 SU_CULBN1.VC_typ
D20 Vout N005 SU_CULBN1.VC_typ
D21 N005 N010 SU_CULBN1.VC_typ
D22 N010 N016 SU_CULBN1.VC_typ
D23 N016 N021 SU_CULBN1.VC_typ
D24 N021 N025 SU_CULBN1.VC_typ
D25 N025 N029 SU_CULBN1.VC_typ
C2 Vout 0 2.6u V=75 Rser=1.85m IC={Vs}
XD1 Vsw Vout PMEG6010CEH
C1 Vs 0 100n V=25 Rser=0 IC={Vs}
R3 N033 N035 25k
C5 N033 0 50p
C4 N035 0 2.2n
XX1 Vs Vsw inductor_model params: L = 12u Rdc = 78.5m Rcore = 2.3k R1 = 19
    .5 C1 = 7.89p
L1 N001 Vs 100n Rser=10m
V2 N034 0 PULSE(0 5 4m 1n 1n 3m 6m 1)
L2 N026 LED1 {Lpar}
L3 N027 LED2 {Lpar}
L4 N028 LED3 {Lpar}
L5 N029 LED4 {Lpar}
XU2 SHDN N034 0 NX7002AK

* block symbol definitions
.subckt inductor_model 1 2
L1 N001 2 {L}
R1 N001 1 {Rdc}
R2 2 N001 {Rcore}
C1 N002 N001 {C1}
R3 2 N002 {R1}
.ends inductor_model

.model D D
.lib standard.dio
.lib lib\SU_CULBN1.VC.lib
.lib lib\NX7002AK.lib
.lib lib\PMEG6010CEH.lib
.lib LT3599.sub

.param Vs = 21
.param Lpar = 300n
.tran 0 10m 0.5m startup

.backanno
.end
```

Bibliography

[1] Jeroen van Ammers and Hsukang Chen. Battery Management System for the Wireless Powerlizer, 2021. Bachelor Thesis, Delft University of Technology.

[2] Brecht Hurkmans and Floris van der Kolk. Wireless Charging System for the Wireless Powerlizer, 2021. Bachelor Thesis, Delft University of Technology.

[3] Arthur Downes, Thomas P. Blunt, and John Marshall. On the Influence of Light upon Protoplasm. *Proceedings of the Royal Society of London*, 28(190-195):199–212, January 1879. doi:10.1098/rspl.1878.0109.

[4] Wladyslaw Kowalski. *Ultraviolet Germicidal Irradiation Handbook*. Springer, July 2009. ISBN 978-3-642-01998-2. doi:10.1007/978-3-642-01999-9.

[5] Oliver Lawal, Jim Cosman, Jennifer Pagan, et al. UV-C LED devices and systems: current and future state. *IUVA News*, 20(1):22–28, 2018.

[6] Yoshihiko Muramoto, Masahiro Kimura, and Suguru Nouda. Development and future of ultraviolet light-emitting diodes: UV-LED will replace the UV lamp. *Semiconductor Science and Technology*, 29(8):084004, June 2014. doi:10.1088/0268-1242/29/8/084004.

[7] Roy Bakker and Marcel Brouwers. Smart Personal Protective Equipment: UVGI, 2020. Bachelor Thesis, Delft University of Technology. uuid:0aef244b-a657-404e-bac3-fbd7dcac87a7.

[8] R. Santhosh and S. Yadav. Low Cost Multipurpose UV-C Sterilizer box for protection against COVID'19. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pages 1495–1498, 2021. doi:10.1109/ICAIS50930.2021.9395752.

[9] IEC TC/SC 108. Audio/video, information and communication technology equipment – Part 1: Safety requirements. IEC 62368-1, International Electrotechnical Commission, 2018. URL https://webstore.iec.ch/publication/63964.

[10] Global Lighting Association. Position Statement on Germicidal UV-C Irradiation, May 2020. [Online]. Available: https://www.globallightingassociation.org/images/files/publications/GLA_UV-C_Safety_Position_Statement.pdf. [Accessed: May 5, 2021].

[11] Miriam E.R. Darnell, Kanta Subbarao, Stephen M. Feinstone, and Deborah R. Taylor. Inactivation of the coronavirus that induces severe acute respiratory syndrome, SARS-CoV. *Journal of Virological Methods*, 121(1):85–91, 2004. ISSN 0166-0934. doi:10.1016/j.jviromet.2004.06.006.

[12] ISO/TC 20/SC 14. Space environment (natural and artificial) – Process for determining solar irradiances. ISO 21348, International Organization for Standardization, 2007. Available: https://www.iso.org/standard/39911.html.

[13] Mohamed O. Elasri and Robert V. Miller. Study of the Response of a Biofilm Bacterial Community to UV Radiation. *Applied and Environmental Microbiology*, 65(5):2025–2031, 1999. ISSN 0099-2240. doi:10.1128/AEM.65.5.2025-2031.1999.

[14] Matthew Olsen, Mariana Campos, Anna Lohning, Peter Jones, John Legget, Alexandra Bannach-Brown, Simon McKirdy, Rashed Alghafri, and Lotti Tajouri. Mobile phones represent a pathway for microbial transmission: A scoping review. *Travel Medicine and Infectious Disease*, 35:101704, 2020. ISSN 1477-8939. doi:10.1016/j.tmaid.2020.101704.

[15] Siiri Kõljalg, Rando Mändar, Tiina Sõber, Tiiu Rööp, and Reet Mändar. High level bacterial contamination of secondary school students' mobile phones. *GERMS*, 7:73–77, 2017. doi:10.18683/germs.2017.1111.

[16] Kabir O. Akinyemi, Audu D. Atapu, Olabisi O. Adetona, and Akitoye O. Coker. The potential role of mobile phones in the spread of bacterial infections. *The journal of infection in developing countries*, 3(8), 2009. ISSN 628-632. doi:10.3855/jidc.556.

[17] Sanchi Malhotra, Jordan Wlodarczyk, Christopher Kuo, Catherine Ngo, Marisa Glucoft, Ivan Sumulong, Michael A. Smit, and Jeffrey M. Bender. Shining a light on the pathogenicity of health care providers' mobile phones: Use of a novel ultraviolet-C wave disinfection device. *American Journal of Infection Control*, 48(11):1370–1374, 2020. ISSN 0196-6553. doi:10.1016/j.ajic.2020.05.040.

[18] Radhika Rana, Sunanda Joshi, Sucheta Lakhani, Mandeep Kaur, and Pragnesh Patel. Cell Phones – Homes for Microbes! *International Journal of Biological and Medical Research*, 4: 3403–3406, 2013. ISSN 0976:6685.

[19] J. Koivunen and H. Heinonen-Tanski. Inactivation of enteric microorganisms with chemical disinfectants, UV irradiation and combined chemical/UV treatments. *Water Research*, 39(8):1519–1526, 2005. ISSN 0043-1354. doi:10.1016/j.watres.2005.01.021.

[20] S. Pillet, P. Berthelot, A. Gagneux-Brunon, O. Mory, C. Gay, A. Viallon, F. Lucht, B. Pozzetto, and E. Botelho-Nevers. Contamination of healthcare workers' mobile phones by epidemic viruses. *Clinical Microbiology and Infection*, 22(5):456.e1–456.e6, 2016. ISSN 1198-743X. doi:10.1016/j.cmi.2015.12.008.

[21] Shannon Johnson. Influenza A vs. B: What to know. Medical News Today, 2020. [Online]. Available: https://www.medicalnewstoday.com/articles/327397. [Accessed: April 26, 2021].

[22] Human Metapneumovirus (HMPV) Clinical Features. Centers for Disease Control and Prevention, 2019. [Online]. Available: https://www.cdc.gov/surveillance/nrevss/hmpv/clinical.html. [Accessed: April 28, 2021].

[23] M. Heßling, K. Hönes, P. Vatter, and C. Lingenfelder. Ultraviolet irradiation doses for coronavirus inactivation – review and analysis of coronavirus photoinactivation studies. *GMS Hygiene and Infection Control*, May 2020. doi:10.3205/dgkh000343.

[24] Kevin Kahn. Is UVC Safe? Klaran University by Crystal IS, 2021. [Online]. Available: https://www.klaran.com/is-uvc-safe. [Accessed: April 28, 2021].

[25] ISO/TC 142. UV-C Devices – Safety information – Permissible human exposure. ISO 15858, International Organization for Standardization, 2016. URL https://www.iso.org/standard/55553.html.

[26] IEC TC 76. Photobiological safety of lamps and lamp systems. IEC 62471, International Electrotechnical Commission, 2006. URL https://webstore.iec.ch/publication/7076.

[27] ISO/TC 198. Sterilization of health care products – Radiation. ISO 11137, International Organization for Standardization, 2020. URL https://www.iso.org/committee/54576/x/catalogue/p/1/u/0/w/0/d/0.

[28] Ataollah Kheyrandish, Fariborz Taghipour, and Madjid Mohseni. UV-LED radiation modeling and its applications in UV dose determination for water treatment. *Journal of Photochemistry and Photobiology A: Chemistry*, 352:113–121, 2018. ISSN 1010-6030. doi:10.1016/j.jphotochem.2017.10.047.

[29] Radu Bogdan Dragomir, Radu Dragomir, and Brânduşa Pantelimon. Irradiance model and simulation of a lighting LED system. *University Polytehnica of Bucharest Scientific Bulletin*, 76(4), 2014. ISSN 2286-3540.

[30] Donald A. Neamen. Optical Devices, In *Semiconductor Physics and Devices: Basic Principles*, pages 618–669. McGraw-Hill Education, 4 edition, 2012. ISBN 978-007-108902-9.

[31] M. Kneissl, T. Kolbe, C. Chua, V. Kueller, N. Lobo, J. Stellmach, A. Knauer, H. Rodriguez, S. Einfeldt, Z. Yang, N. M. Johnson, and M. Weyers. Advances in group III-nitride-based deep UV light-emitting diode technology. *Semiconductor Science and Technology*, 26(1):014036, dec 2010. doi:10.1088/0268-1242/26/1/014036.

[32] Max Shatalov, Wenhong Sun, Rakesh Jain, Alex Lunev, Xuhong Hu, Alex Dobrinsky, Yuri Bilenko, Jinwei Yang, Gregory A Garrett, Lee E Rodak, Michael Wraback, Michael Shur, and Remis Gaska. High power AlGaN ultraviolet light emitters. *Semiconductor Science and Technology*, 29(8):084007, June 2014. doi:10.1088/0268-1242/29/8/084007.

[33] Shuai Wang, Hanling Long, Yi Zhang, Qian Chen, Jiangnan Dai, Shuang Zhang, Jingwen Chen, Renli Liang, Linlin Xu, Feng Wu, Zi-Hui Zhang, Haiding Sun, Changqing Chen, and Yihua Gao. Monolithic integration of deep ultraviolet LED with a multiplicative photoelectric converter. *Nano Energy*, 66:104181, 2019. ISSN 2211-2855. doi:10.1016/j.nanoen.2019.104181.

[34] *SU CULBN1.VC: AlGaN UV Emitter*. Osram Opto Semiconductors, May 2021. Version 1.5. [Online]. Available: https://dammedia.osram.info/media/resource/hires/osram-dam-17966691/SU%20CULBN1.VC_EN.pdf. [Accessed: May 10, 2021].

[35] T. Quill, S. Weiss, C. Hirschler, V. Pankadzh, G. DiBattista, M. Arthur, and J. Chen. Ultraviolet Reflectance of Microporous PTFE. Porex Corporation, 2020. [Online]. Available: https://www.porex.com/wp-content/uploads/2020/04/Ultraviolet-Reflectance-of-Microporous-PTFE.pdf. [Accessed: May 7, 2021].

[36] UV-Enhanced Aluminum Mirrors. Thorlabs, Inc., 2021. [Online]. Available: https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=12393. [Accessed: May 7, 2021].

[37] Rajul Randive. AN011 – Using UV Reflective Materials to Maximize Disinfection. Klaran University by Crystal IS, 2016. [Online]. Available: https://www.klaran.com/using-uv-reflective-materials-to-maximize-disinfection. [Accessed: May 7, 2021].

[38] Jorge Nocedal and Stephen J. Wright. Fundamentals of Optimization, In *Numerical Optimization*, pages 1–9. Springer-Verlag New York, 2 edition, 2006. ISBN 978-0387-30303-1. doi:10.1007/978-0-387-40065-5.

[39] *AN052 – Thermal management of light sources based on SMT LEDs*. Osram Opto Semiconductors, August 2018. [Online]. Available: https://dammedia.osram.info/media/resource/hires/osram-dam-2496744/Thermal%20management%20of%20light%20sources%20based%20on%20SMT%20LEDs.pdf. [Accessed: May 8, 2021].

[40] David Meeker. Finite Element Method Magnetics (FEMM) 4.2, February 2018. [Online]. Available: https://www.femm.info. [Accessed: May 8, 2021].

[41] *AND9596/D – A Quick PCB Thermal Calculation for Power Electronic Devices with Exposed Pad Packages*. ON Semi, October 2017. Revision 1. [Online]. Available: https://www.onsemi.com/pub/Collateral/AND9596-D.PDF. [Accessed: May 8, 2021].

[42] *AN1656 – Design Challenges of Switching LED Drivers*. Texas Instruments, May 2013. [Online]. Available: https://www.ti.com/lit/pdf/snva253. [Accessed: May 22, 2021].

[43] Ruihong Zhang and Henry Chung. Paralleled LED Strings: An Overview of Current-Balancing Techniques. *Industrial Electronics Magazine, IEEE*, 9:17–23, 06 2015. doi:10.1109/MIE.2014.2354681.

[44] K. I. Hwu and Sheng-Chien Chou. A Simple Current-Balancing Converter for LED Lighting. In *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, pages 587–590, 2009. doi:10.1109/APEC.2009.4802717.

[45] Sinan Li and S. Y. Ron Hui. Self-Configurable Current-Mirror Circuit With Short-Circuit and Open-Circuit Fault Tolerance for Balancing Parallel Light-Emitting Diode (LED) String Currents. *IEEE Transactions on Power Electronics*, 29(10):5498–5507, 2014. doi:10.1109/TPEL.2013.2291392.

[46] Steve Winder. *Power Supplies for LED Driving*. Newnes, 2017. ISBN 978-0-08-100925-3. doi:10.1016/C2015-0-04539-7.

[47] Robert Selders. Synchronous rectification in high–performance power converter design. Texas Instruments, 2016. [Online]. Available: https://www.ti.com/lit/pdf/snva595. [Accessed: May 16, 2021].

[48] Robert W. Ericson and Dragan Maksimović. *Fundamentals of Power Electronics*. Springer International Publishing, 3 edition, 2020. ISBN 978-3-030-43881-4. doi:10.1007/978-3-030-43881-4.

[49] Marcin Walczak. Impact of inductor current ringing in DCM on output voltage of DC-DC buck power converters. *Archives of Electrical Engineering*, 66:313–323, 06 2017. doi:10.1515/aee-2017-0023.

[50] Ron Lenk. *Practical Design of Power Supplies*. Wiley-IEEE Press, July 2005. ISBN 978-0-471-75045-1. doi:10.1109/9780470545621.

[51] Ned Mohan, Tore M. Undeland, and William P. Robbins. *Power Electronics: Converters, Applications, and Design*. Wiley, 3 edition, 2003. ISBN 978-0-471-22693-2.

[52] Ridley Engineering. Magnetics Core Loss Webinar, November 2020. [Online]. Available: https://ridleyengineering.com/videos-e/307-magnetics-core-loss-webinar.html. [Accessed: May 13, 2021].

[53] Robert Sheehan. Understanding and Applying Current-Mode Control Theory. Power Electronics Technology Exhibition and Conference, October 2007. [Online]. Available: https://www.ti.com/lit/pdf/snva555. [Accessed: May 17, 2021].

[54] Chunxiao Sun and B. Lehman. Discussions on Control Loop Design in Average Current Mode Control. In *Conference Record of the 2000 IEEE Industry Applications Conference. Thirty-Fifth IAS Annual Meeting and World Conference on Industrial Applications of Electrical Energy (Cat. No.00CH37129)*, volume 4, pages 2411–2417, 2000. doi:10.1109/IAS.2000.883161.

[55] D. Grahame Holmes and Thomas A. Lipo. Continuing Developments in Modulation, In *Pulse Width Modulation for Power Converters: Principles and Practice*, pages 585–621. Wiley-IEEE Press, 2003. ISBN 978-0-471-20814-3. doi:10.1109/9780470546284.ch14.

[56] Bing Lu. Control and Design Challenges for Synchronous Rectifiers. Texas Instruments Power Supply Design Seminar, 2018. [Online]. Available: https://www.ti.com/seclit/ml/slup378/slup378.pdf. [Accessed: May 16, 2021].

[57] *Understanding Boost Power Stages in Switchmode Power Supplies*. Texas Instruments, March 1999. [Online]. Available: https://www.ti.com/lit/pdf/slva061. [Accessed: May 16, 2021].

[58] *LT3599 4-String 120mA LED Driver*. Analog Devices, February 2009. Revision F. [Online]. Available: https://www.analog.com/en/products/lt3599.html. [Accessed: May 14, 2021].

[59] Andrew Skelly and Mark D. Waugh. Understanding DC Bias Characteristics in High-Capacitance MLCCs. Ceramic Industry, October 2009. [Online]. Available: http://www.ceramicindustry.com/articles/90304-understanding-dc-bias-characteristics-in-high-capacitance-mlccs/. [Accessed: May 18, 2021].

[60] *High Speed PCB Layout Techniques – High Speed Analog Design and Application Seminar*. Texas Instruments, August 2005. [Online]. Available: https://www.ti.com/lit/pdf/slyp173. [Accessed: June 2, 2021].

[61] S.W. Lee. Practical Feedback Loop Analysis for Current-Mode Boost Converter. Texas Instruments, March 2014. [Online]. Available: https://www.ti.com/lit/pdf/slva636. [Accessed: May 27, 2021].

[62] H. H. Ku. Notes on the use of propagation of error formulas. *Journal of Research of the National Bureau of Standards. Section C: Engineering and Instrumentation*, 70C(4):263–273, October 1966. ISSN 0022-4316. doi:10.6028/jres.070C.025.