UV-C LED Seed Disinfection

Mechanics Group

EE3L11: Bachelor Graduation Project

M. Mazurovs
D.O. Schat

Delft University of Technology

TU Delft

UV-C LED Seed Disinfection

Mechanics Group

by

M. Mazurovs
D.O. Schat

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Tuesday June 27, 2023 at 13:30 PM.

Authors:	M. Mazurovs	5050545
	D.O. Schat	5169801
Project duration:	April 24, 2023 – June 30, 2023	
Thesis committee:	Dr. ing. H. W. van Zeijl	TU Delft, supervisor
	Drs. Ir. L. Wymenga	TU Delft, supervising PHD candidate
	Prof. dr. ir. J. van Turnhout	TU Delft
	Dr. ing. I. E. Lager	TU Delft
	Dr. S. Izadkhast	TU Delft

with contribution from the entire UV-C disinfection group in Chapter 1 and 2.
Bsc. Electrical Engineering 2023 BAP group M:

E. Ergül	5334640
R.W.L. Imbens	5155940
L.C. Klootwijk	5155940
M. Mazurovs	5050545
D.O. Schat	5169801
E.H. van Weelderen	5315115

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface & acknowledgement

The research and development of advanced technologies in the field of disinfection and sterilization have been increasingly popular after the events of the Covid virus. With the ever-increasing population size and the need to improve sustainability, innovative solutions are required to optimize the growth and yield of the agricultural sector. This thesis presents a comprehensive proof of concept into the design and development of a UV-C LED-based seed disinfection device, aimed at improving the quality of seeds by removing pathogens.

The timespan to create this device was a total of 8 weeks, during which our team of Bachelor students, mobilized their efforts to overcome challenges presented by the budget and time limits and push the boundaries of seed disinfection technologies. Our goal was to use short-wave ultraviolet light (UV-C), and use the advantages of UV-C LED technology over traditional mercury UV lamps.

The first chapter gives some background information and a state-of-the-art analysis. The next chapters outline the design choices and modules selected for the seed treatment machine, where we present simulation results, optimization techniques like the use of a motor to turn around the seeds and design choices for the device. Our team, consisting of three subgroups, designed and implemented the necessary modules to ensure seamless operation between all modules.
The prototype implementation section highlights the iterative design process, from the initial impressions to the final design. Our results demonstrate the successful integration of various components and the achievement of our objectives. In the end, we critically examine the outcomes, highlighting the strengths and limitations of the UV-C LED-based seed disinfection device. Drawing from our designs and observations, we give conclusions and recommendations for further enhancements.

The seed disinfecting device was almost created entirely in-house. Leveraging our expertise in electronic engineering, we utilized KiCad, an open-source software, for designing Printed Circuit Boards (PCBs) that formed the electrical systems for this device. This allowed us to tailor the electronic modules precisely to our requirements. Additionally, we made use of Shapr3D, a powerful 3D modeling tool, in conjunction with an Apple Pencil, to design and refine the physical and mechanical structure of the machine. As we had never fabricated a PCB before and never used Shapr3D, we are pleased with the results.

Unfortunately, we could not test the device with complete integration of all modules on actual cabbage seed before the deadline for submitting this Bachelor's thesis, which was set 8 weeks after the beginning of the project. Though we do plan to perform these tests after the deadline and present the results in the final presentation, which takes place in week 10.

We express our sincere gratitude to our supervisors, dr. ing. H. W. van Zeijl, drs. ir. L. Wymenga and Prof. dr. ir. J. van Turnhout, who supported us throughout this project. Their invaluable expertise, criticism, and encouragement enabled us to overcome barriers and hopefully achieve meaningful results with this device. We had a great time working on this project with our team, consisting of E. Ergül, E.H. van Weelderen, L.C. Klootwijk, and R.W.L. Imbens, and also want to thank them for their incredible help. We want to specifically thank Rijk Zwaan for the additional funding of this project and their help with the planned PCR testing of the cabbage seed for pathogens. Finally, a special thanks is owed to M. Schumacher for providing us with the proper tools and for his hospitality at the Tellegen Hall. It is our hope that this thesis and the final device serve as a proper start for future research and development in the field of seed disinfecting technology.

M. Mazurovs
D.O. Schat
Delft, July 2023

Abstract

This thesis presents the design and development of a UV-C LED-based seed treatment machine aimed at enhancing seed quality by the extermination of pathogens. The research covers design choices, including a round irradiation pattern, consisting of two rings with three and nine LEDs for the inner and outer ring respectively, the use of a quartz plate as a holding plate for seeds for its high UV-C light permeability capabilities, the use of a vibration motor underneath the main operational stack for seed movement, and the use of Ethernet ports for power distribution and communication. The thesis discusses a comparative study between square and circular plate configurations, evaluating their performance using simulation results. Safety considerations were prioritized in the design, and appropriate precautions were implemented throughout the design process. The thesis also highlights the iterative design process for the mechanical system, discussing challenges encountered and improvements made to achieve a functional and robust prototype. Results demonstrate successful integration of components and achievement of objectives. The thesis concludes with discussions on the strengths, limitations, and future enhancements of the UV-C LED-based seed treatment machine. The research presented in this thesis provides valuable insights for further advancements in seed treatment technology, contributing to sustainable agricultural practices. Due to time constraints, conclusive results of testing on seed with this machine could not be obtained yet.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AC	Alternating current
DC	Direct current
op-amp	Operational Amplifier
LED	Light Emitting Diode
UVC & UV-C	Lowest range UV light (100 - 280 nm)
PoE	Power over Ethernet
PSU	Power Supply Unit
CU	Control Unit
MCU & MC	Motor Control Unit
LDS	LED Driving & Sensing module
PCB	Printed Circuit board
IC	Integrated-Circuit, usually referring to a chip or module
UART	Universal Asynchronous Receiver/Transmitter [1]
USB	Universal serial bus
I^2C	Inter-Integrated Circuit [2]
PWM	Pulse Width Modulation
V2020	Aluminium frame with V-slots on all sides with 20x20mm width & depth
OC_MC_FLAG	Motor Controller Overcurrent Flag
OC_PS_FLAG	Power Supply Unit Overcurrent Flag
RJ45	standard Ethernet connector
PoR	programme of requirements
MR	Mandatory requirements
ToR	trade-off requirements

Symbols

Symbol	Definition	Unit
V	Voltage	[V]
I	Current	[A]
P	Power	[W]
R	Resistance	[Ω]
C	Capacitance	[F]
L	Inductance	[H]
λ	Wavelength	[nm]

1 Introduction

Behind every product in a grocery store is a story to tell, from idea proposition to manufacturing and transportation. One thing that not many people know is that behind fruits and vegetables, it doesn't start with farmers planting seeds, it starts with seed breeders & distributors selling good quality seeds, disinfected from pathogens. The seed industry has been specializing in this sector for a long time, removing all kinds of pathogens, fungi, bacteria and viruses, from all sorts of plant species. With this, they establish the success of every year's harvests worldwide. By taking this precaution, fewer plants get infected after germination, thereby preventing whole acres of land from being infected with the pathogens and cutting the whole food production chain between farms and a customer's refrigerator. Consequently, it's very important in this market that all seed have a guarantee that they are disinfected. A lot of people, jobs and money thus depend on these first steps in the whole agriculture-meets-customer chain.

After being hosted by Rijk Zwaan, the fourth biggest seed distributor in the world, at their headquarters in De Lier, Netherlands, the current method of seed disinfection with its complications was explained. Currently, the seeds undergo a 2-step process: water basins and warm air ovens. First, the seeds are dipped in hot water baths. Because of the water, the pathogens are ”wake up” so to say, and killed because of the temperature of the water. However, the water also activates the seed for germination, which is unwanted yet. That's why the seeds then undergo a drying phase with hot air for quick water evaporation. This process has some drawbacks, the biggest of them being the dependency on water and energy consumption for heating both water and air. This makes the process cost-inefficient, unsustainable, time-consuming and commercially less interesting.

Thus, enterprises such as Rijk Zwaan are searching for more efficient and more sustainable alternative disinfection techniques and their potential for the seed industry. Specifically, Rijk Zwaan has made contact for help with the Technical University of Delft, which has research groups investigating the use of short-wavelength light, in other words, high energy ultraviolet light, for disinfection of pathogens. This is the bridge to the Electrical Engineering Graduation Project Proposal that led to this thesis.

1.1. Problem definition

Currently, the traditional disinfection technique with ultraviolet light is with mercury lamps which can emit short-wavelength light. Since mercury is a toxic metal and the lamps are quite power-consuming, the project's target was to investigate the seed disinfection problem with the use of light-emitting diodes (LEDs) which emit UV-C light (ultraviolet type C). The specific project proposal is to provide a proof of concept on the question of whether it's possible to disinfect seeds from pathogens using UV-C LEDs. In discussion with Rijk Zwaan, the specific disinfection testing would take place for cabbage seed (Brassica Oleracea) from the Alternaria fungus. This is a common-found fungus in the seed industry, but it had a high resistance and adeptness [3]. If the hypothesis is correct that this disinfection method is feasible, then this increases the chance of it being effective on other seeds and other pathogens too.
The execution of the graduation project team that took on this task was to create a prototype machine, which could prove the concept of UV-C seed disinfection of seeds, but for the scope of the project's

duration, only to be used in an experimental environment. In the end, a machine was created where cabbage seeds are placed on a quartz plate that is then put in a cylindrical container. In this container, underneath and on top of the quartz plate with seeds are frames with UV-C LEDs which disinfect the seeds. The user sets the desired exposure time, wavelength, and intensity and begins the disinfection. A motor with a decentralized weight is used to create vibration to turn the seed such that all sides are illuminated by the UV-C light. After this, the cabbage seeds should be disinfected. After the illumination, the project team could send the presumably disinfected seed back to Rijk Zwaan which could perform specific tests to find out whether the seeds are indeed disinfected and with which ratio.
For equal work distribution within the project group, which consists of six Electrical Engineering students, three subgroups were created that focused on their own module of the machine. One group was responsible for the LED modules, including the driving system and appropriate sensors for safety checking. The second group focused on the control unit and software of the system, including the user interface and the setting of parameters such as intensity. The third group worked on the power distribution and the mechanical side of the whole system, including the concept of seed vibrations with the use of a motor and the design of the machine. The development of this third module is what this thesis is about.

1.2. State-of-the-art analysis

Before going into more detail about the project, there is some vital theory regarding the matter that needs further analysis from previous research.

1.2.1. Short wave ultraviolet light, UV-C

Ultraviolet light is light that has a wavelength from 100 to 400 nm [4]. Far ultraviolet light, also known as UV-C, is part of this UV-light spectrum with wavelengths from 100 to 280 nm [5].

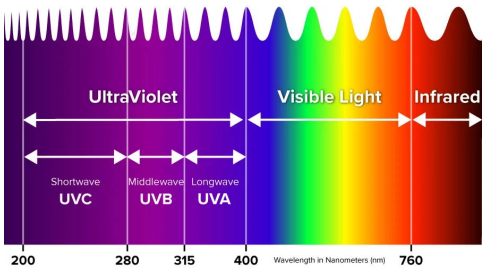


Figure 1.1: UV-C in the electromagnetic spectrum [4]

1.2.2. What can short-wave ultraviolet light do

The short-wave UV-C light can be used to disinfect objects. It is chemically capable of breaking and redirecting the bounds between A-T and G-C in DNA and RNA chains [6] can be seen in figure 1.2, in turn, inactivating bacteria, viruses, and even fungi [7]. Many pathogens have been tested, and UV irradiation has been a proven disinfection method [8]. Philips UV-C light sources have been proven to inactivate 99% of the SARS-CoV-2 virus in 6 seconds from a surface [9]. This small time window makes the technology interesting for industrial applications.

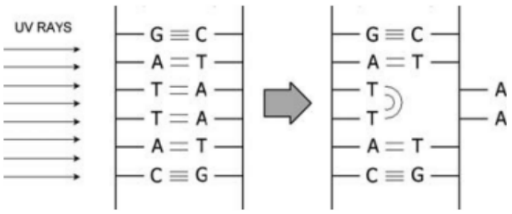


Figure 1.2: RNA & DNA inactivation by UV rays [6]

A study showed that for disinfecting some fungi in some specific circumstances, a higher UV-C dose

from 8 to 120 mJ/cm^2 is needed to achieve a 90% reduction [10]. In figure 1.3, the absorption and intensity of different parts of the ultraviolet spectrum on several building blocks of living organisms is given, namely DNA/RNA and protein strands, from different UV-C light sources.

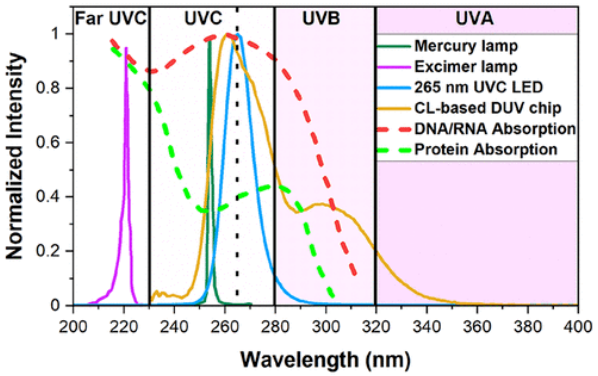


Figure 1.3: Spectrum of absorbance and intensity of ultraviolet light [11]

1.2.3. UV-C Mercury lamps versus UV-C LEDs

As mentioned before, UV-C irradiation is widely used for disinfection already. Most of these use cases utilize UV-lamps with mercury. Using new UV-C LEDs can improve upon this existing technology because they would be safer to use, but also because they create less heat, so they are more sustainable, and emit more concisely the desired wavelengths [12]. Moreover, mercury lamps require time and energy to warm up, which isn't process efficient. Also in terms of design mercury lamps are not favourable, since they are much bigger and bulkier than small LEDs [3].

1.2.4. UV-C disinfection & Alternaria

As stated before, the target pathogen to disinfect is the fungus Alternaria. UV-C disinfection has already been proven by the use of a mercury UV-C lamp at a distance of 30 cm [13]. The tests were done during the span of 12, 24, 36 and 48 seconds and the article shows a dose of 0.25kJ/m^2 should be enough to control the growth of Alternaria, which corresponded to an exposure time of 12 seconds. It even suggests a larger dose increases susceptibility to diseases. This already shows promising results for the UV-C LED disinfection device.
Another study also shows inhibition of Alternaria after UV-C treatment by each of the dose settings of 1, 5 and 10 kJ/m^2. These tests were done from 10 cm, also at a wavelength of 254 nm [14].

1.2.5. Safety considerations

Just like the other parts of the ultraviolet spectrum, the short wavelength UV-C can also be harmful to humans. Therefore, working with this technology should be done with safety in mind. UV-C irradiance can be harmful to the eye and skin [15].

1.2.6. Testing for success rate

There are several ways in which it can be tested whether the plant seeds that have undergone a UV-C disinfection process are indeed disinfected. These tests were explained to the project's team by Rijk Zwaan [3].
The most reliable, but also most time-consuming of all the tests is planting the seeds. After some weeks of planting, the seed will have germinated and grown. Via observation, it can be stated whether it's infected or not. Another significantly reliable test method is the use of PCR tests. By running a lot of cycles, this process measures the presence of amplified labelled fluorescing parts of DNA or RNA. Thus, the more specific parts of DNA/RNA are still on the seed, labelled the more of the pathogen is still present on the seed [3].
The problem with these tests is usually, however, that during the iteration process, it also amplifies DNA of already dead organisms that were on the seeds [16]. In this context it should be less of a problem since in theory, the UV-C light could still destroy these DNA and RNA strands which then should not be detected. This is the test primarily used to test the prototype's functionality.

2 General

2.1. Programme of requirements

The programme of requirements (PoR) should act as a the main leading factor in the design and development of the device. There are certain functionalities that the device must have and some functionalities that are there to make the final product more appealing to the end users. These can be distinguished as *mandatory requirements (MR)* and *trade-off requirements (ToR)*, respectively. The PoR also acts as a method of assessing the performance of the project. It must be noted that the PoR has been constructed for an experimental environment and must be reviewed when upgrading to a commercial environment..

Requirements that fall under the label of mandatory requirements are the constraints of the design. Whether the design is a success or not is based on these requirements primarily. Therefore, one should always aim to satisfy each input of the MR.

Trade-off requirements are requirements that the system or sub-module do not necessarily have to satisfy to assess the performance of the project. However, some of these requirements might make it more appealing. The requirements shall experience a trade-off, mainly consisting of time versus project progress gain, thus how much time will it take to satisfy these requirements and how does this result in coming closer to successfully finishing the project [17].

Furthermore, the requirements are divided into two different system levels. First, the PoR for the device as a whole is presented and these are general requirements which define what the device has to do. These are also listed in the thesis reports of the other two subgroups [18] [19]. Secondly, a PoR for the Power Supply Unit and a PoR for the Motor Control Unit are created, wherein the requirements for the sub-modules that are described in this thesis will be shown in their respective chapters.

2.2. System requirements

First, the system requirements are set. These hold for the entire project. These requirements are either set by the project proposers or the BAP-group.

2.2.1. Mandatory requirements

Number	Requirement
MR.1	The system must make use of UV-C LEDs
MR.2	The system must disinfect Alternaria on cabbage seed
MR.3	The system must irradiate all seeds via an uniform radiation pattern.
MR3.1	The seeds must be irradiated on all sides
MR3.2	The irradiation intensity must be uniform on all locations
MR.4	All modules of the system must be powered from the same voltage source
MR.5	All modules of the system must be able to communicate with the Control Unit
MR.6	The system must have appropriate error detection
MR.7	The system must only be enabled when there is no error detected
MR.7.1	The system must turn off when the UV-C LEDs go beyond their optimum operating temperature
MR.7.2	The system must turn off when the ambient temperature of the seeds puts the germination chance of the seeds at risk
MR.7.3	The system must turn off when the ozone concentration in the radiation enclosure exceeds the allowed maximum concentration of 0.1 ppm $\left[200\,\mathrm{g/m^3}\right]$) [20]
MR.7.4	The system must turn off when the total current draw exceeds that of the lowest maximum current rating of any component that experiences that exact amount of current flow
MR.7.5	The system must turn off when one of the modules experiences a current draw exceeding that of the lowest maximum current rating of any component in that path
MR.8	The system must be externally controllable
MR.8.1	The radiation parameters must be adjustable
MR.8.1.1	The intensity of present wavelengths in the radiation pattern must be adjustable
MR.8.1.2	The duration of radiation must be adjustable
MR.8.1.3	The intensity of the radiation must be adjustable
MR.8.2	The system must be able to be turned off with a single switch
MR.9	The state of the system must be monitored at all times
MR.9.1	The temperature of the UV-C LEDs must be monitored
MR.9.2	The temperature of the seeds must be monitored
MR.9.3	The ozone levels in the radiation enclosure must be monitored
MR.9.4	The current drawn by the system must be monitored
MR.9.5	The current drawn by each module must be monitored

Table 2.1: The mandatory requirements of the system

2.2.2. Trade-off requirements

Number	Requirement
TR.1	The system will be in a closed casing to prevent UV leakage
TR.2	The system will have an easy-access mechanism for the adding/removing the seeds
TR.2.1	The system will have an automated seed transportation mechanism
TR.3	The system will be able to inactive more pathogens than Alternaria
TR.4	The system will be able to disinfect more seeds than cabbage seed
TR.5	Independent error detection will be implemented for finding which module triggered the system error
TR.6	The system can only turn on when the enclosure is completely sealed or closed

Table 2.2: The trade-off requirements of the system

2.3. PSU and MCU Requirements

The requirements per module that are created and discussed in this thesis, namely the Power Supply Unit and Motor Control Unit, can be found in the assigned chapters for the modules, namely chapter 4.1 and 5.1 respectively.

2.4. General design choices

2.4.1. Modules

The device will consist of a couple of modules, which are separate PCBs. Each module has its own function. This report will cover the power distribution and the motor controller. The other modules will get a brief introduction without going into detail:

The first component is the control unit, it is responsible for communication, error checking, data collection, and parameter configuration such as time, intensity, and wavelength.

Next up are the two LED drivers. It was decided that radiation from the bottom and the top was ideal and useful for seed disinfection looking at the project's requirement to irradiate the seeds as uniform as possible. Therefore two identical PCBs are needed. In short, these will consist of a stack of PCBs. These PCBs will be responsible for irradiation (LED PCB), sensing (middle PCB) and power conversion for the LEDs.

As stated before, this thesis covers the motor controller, power supply, LED configuration simulations and overall 3D design of the device. The motor controller is chosen to be used because of the general design choice MR3.1. By the use of a motor with a decentralized weight the seed can be vibrated, which will result in slight movements and turning of the seed and therefore the seed will be more uniformly illuminated.

2.4.2. Power distribution for all modules

The question that now arises is, what is the best way to connect all the modules to each other for power, but also for signal communication? Instead of implementing power and several signal cables, which would result in chaos by having so many cables, the decision was made to make use of Ethernet cables. These cables are perfect in this case since you don't need necessarily to use them for distributing internet, you can use them for personal signals and at the same time also use them for power distribution. The Ethernet cables consist of eight wires. In the case of communication between the PSU and the other modules, one of the eight wires is reserved for the ground cable and another one for the positive voltage, the power supply.

Not only does the implementation of Ethernet cables tidy up the system, but it also offers more stability to the system and a lower chance of modules disconnecting as a consequence of too much vibration from the motor or of an unforeseen kick to the machine. This certainty is given by the functionality of Ethernet cables to click into the Ethernet ports on all modules making them difficult to be unintentionally disconnected.

Using the Ethernet cable for power distribution is similar to the well-known Power-over-Ethernet protocol, which as the name suggests, also distributes power over Ethernet cables. This technology has also an advantage of being wide-spread in many use-cases, making it easy and cheap to acquire, which in turn gives an advantage in terms of cost and manufacturing.

2.4.3. Signal communication

Generally, all modules, except for the power supply, communicate with the control unit through I^2C, which is capable of having multiple modules on the same communication bus that can send and receive information. Only some hard-wired error flags will be used on the Ethernet ports in case the software does not function properly or fails.

2.4.4. Wavelength decision

As seen in figure 1.3, different parts of what pathogens are made of have different optimum wavelengths for them to be destroyed. In order to test what works the best, or whether a combination is even more efficient, the design choice was made to make use of four different wavelengths:

- 255 nm - Due to the combination of availability and that this wavelength covers the peak at which DNA/RNA absorbs UV-C light [11], and thus gets destroyed
- 275 and 285 nm - Due to availability of LEDs with these wavelengths, the combination could lead to a combined peak at 280 nm. This is needed for the best protein absorption of the pathogens [11]. This way, not only DNA/RNA gets affected by this machine, but also the building blocks of the pathogens.
- 395 nm - inspired by ongoing research about the awakening of pathogens from their hibernation state with the use of UV-A light [21], it was decided to apply this to our experiments too. In theory, with this pre-treatment, the pathogens would be weaker and have more chance of really being destroyed by UV-C light.

In the end, only three LED types are needed. Since the UV-A light would only be needed independent from the UV-C treatment, a LED type package was found that either illuminates 285nm or 395nm. The next step would be to decide on the layout and number of these LEDs. This will be discussed in chapter 3.2.2.

3

Simulation

As stated in the program of requirements of the seed UV-C disinfector, all the seeds on the plate should be just as much irradiated with UV-C light. For example, the seed in the centre should be just as much irradiated as the seed at the sides, since it's essential that there's a guarantee that all seeds are disinfected enough.

3.1. Square plate

As described in the general design choices, section 2.4, it was chosen during the design process of the prototype to work with a circular plate instead of a square plate where upon the seeds are laying. At first thought, from the view of process efficiency, this is illogical. A square plate of the same width has more area than a circular plate, so more seeds could be exposed to UV-C light at the same time. However, in terms of uniform irradiance, it seems that it's much less efficient to use a square plate than a circular plate. This was the conclusion when the simulation method of chapter 3.2.2 was adapted to be able to place LEDs in a layout of our preference on a square plate. The input parameters are the same as for a circular plate, as will be explained in the next section 3.2.2, however, now, instead of the number of rings and LEDs per ring, the function accepts the number of rows and LEDs per row. The full code can be found in appendix A.1.1. This code is an edited version, applicable for this project's context, of the irradiance simulation script written by R.P.M. Bakker and M.J.H. Brouwers [22]. Since both code implementations for square and circular plates are similar, explaining the code twice is unnecessary. In the sections below, only different layouts on a square plate are shown with observations and conclusions obtained from the plots. Each figure describes the next step of finding the optimum LED layout which should be the perfect balance between the number of LEDs, applicability, and irradiance uniformity.

3.1.1. Diagonal

First, a layout with diagonally placed LEDs was tried. This is thought to be convenient since the LEDs with other wavelengths could then be placed the same way but with a horizontal offset in each row. Since this is a simulation in the first phases of the design, reflection isn't taken into account yet. This way, the influence of the form could better be accentuated. The resulting irradiance for one wavelength is shown here below in figure 3.1.

An interesting observation here is that the corners get irradiated much less than the centre and that in the centre a circle emerges where the irradiance within the circle is evenly distributed.

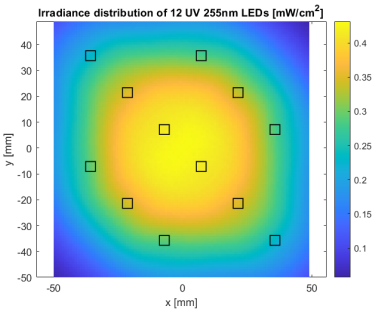


Figure 3.1: Relative irradiance distribution of 12 255 nm LEDs on a square plate, diagonally placed

3.1.2. Straight

Now, a layout with LEDs on straight lines was tried with different combinations.

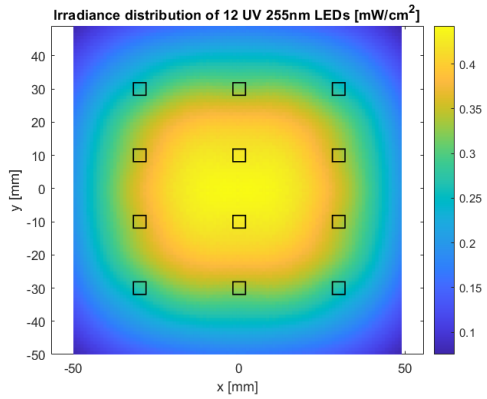


Figure 3.2: Relative irradiance distribution of 12 255 nm LEDs on a square plate, with straight lines and 4x3 as layout

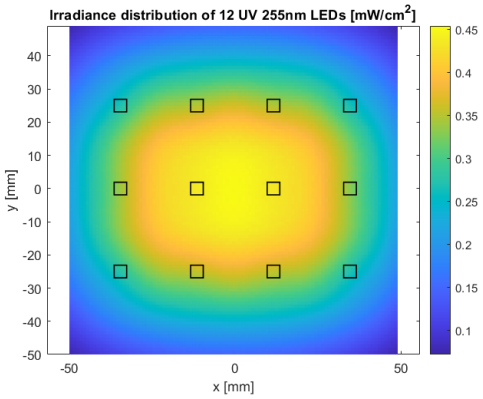


Figure 3.3: Relative irradiance distribution of 12 255 nm LEDs on a square plate, with straight lines and 3x4 as layout

Again, a concentration in the centre is present. The next step was to try to move the outer LED sides more to the side. First to the left and right side, and then also combined with stretching to the bottom and upside.

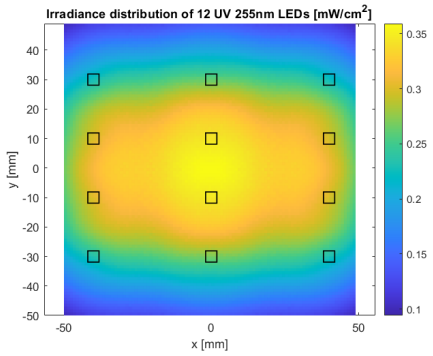


Figure 3.4: Relative irradiance distribution of 12 255 nm LEDs on a square plate, with straight lines and 4x3 as layout, stretched to the left and right side

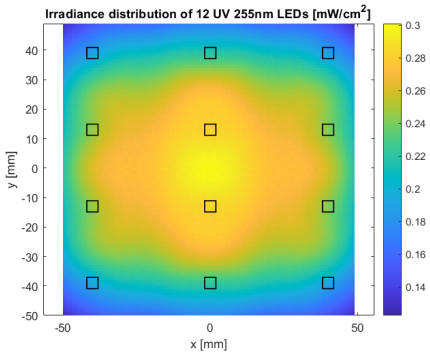


Figure 3.5: Relative irradiance distribution of 12 255 nm LEDs on a square plate, with straight lines and 4x3 as layout, stretched to all sides

The one that is stretched to all four corners already looked a little bit better, however still not very satisfactory. Also, the maximum intensity dropped $0.1mW/cm^2$. Besides, this had to be combined with twelve LEDs of the 275 nm wavelength and another twelve of 285 nm/395 nm. To roughly make some space in each row for the other LEDs while having the same distribution for all four wavelengths, the simulation was done with the application of an offset in each row. The result was as such:

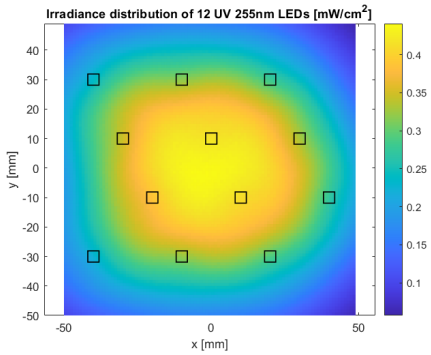


Figure 3.6: Relative irradiance distribution of 12 255 nm LEDs on a square plate, with straight lines with offset and 4x3 as layout

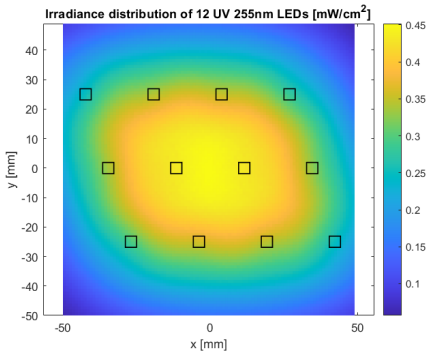


Figure 3.7: Relative irradiance distribution of 12 255 nm LEDs on a square plate, with straight lines with offset and 3x4 as layout

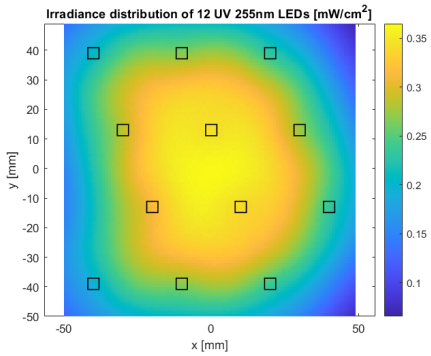


Figure 3.8: Relative irradiance distribution of 12 255 nm LEDs on a square plate, with straight lines with offset and 4x3 as layout, stretched to the corners

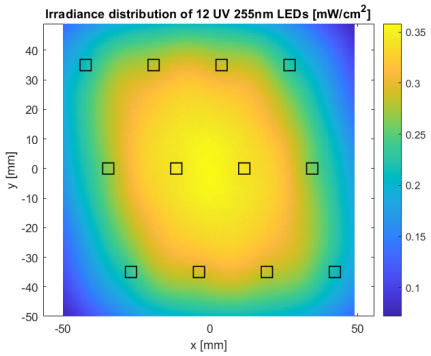


Figure 3.9: Relative irradiance distribution of 12 255 nm LEDs on a square plate, with straight lines with offset and 3x4 as layout, stretched to the corners

As to be seen in figure 3.9, two of the corners are already better lit than for example in figure 3.2. However, the other two are still in the dark. In conclusion, a few layouts were tried to analyse the irradiance it gives on the plate. In the end, all layouts confirmed our hypothesis that a square plate is inefficient for use. Moreover, this is only the conclusion from the irradiance perspective. When viewing this problem from a mechanical perspective, a square plate would not satisfy one of the system's requirements, namely the solution to uniformly irradiate all sides of a seed on all places of the plate. As described in chapter 2.4, in order to not only expose the bottom and top of the seeds by illuminating from underneath and above but also the sides, a motor is installed to vibrate the seeds, which then will uniformly rotate the seed. In the case of a square plate, seeds would presumably move to the corners of the plate, resulting in amassments of seeds there. This is unfavorable since that would mean that the seeds that are buried don't get exposed to UV-C light, and thus don't get disinfected. On top of that, as could be seen from simulations, the corners have always had a lower irradiance.

So the combination of all the negative impacts of a square plate implementation outweighed the advantage of it having a bigger capacity for disinfecting seed, so simulations for a circular plate were tried to test whether a circular plate indeed does a better job at this.

3.2. Circular plate

3.2.1. Seed layout

As said before, PCR tests are the most reliable test with short duration Rijk Zwaan can run to test the disinfection rate of the seeds. Specifically, the tests they perform have a capacity for testing a thousand seeds or twice a batch of 500 [3]. Thus, once it was established that a circular plate would be used, it was time to decide on the diameter. It had to be a size that would be easy to pull out of the machine, but also big enough that enough seed could be disinfected and sent for testing at once.

When simulating how many seeds could fit on a circular plate with a diameter of 10 centimeters, a total of 1949 seeds was the result, see figure 3.10. However, this is an ideal case where the seeds are perfectly laying next to each other and are all perfect spheres of 2 mm.
The problem, however is that it would be too time-consuming to place the seeds all ideally next to each other. To compensate for the irregularity of seeds and the non-ideal placing of the seeds, a margin of 0.75 mm per seed is chosen. The resulting layout would be as given in figure 3.11 with a total number of seeds being 1028, which is perfect for the batch needed for PCR testing.

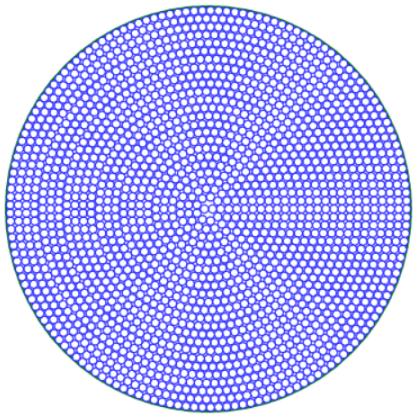


Figure 3.10: Seed distribution on a circular plate with 10cm diameter assuming 2 mm seed thickness

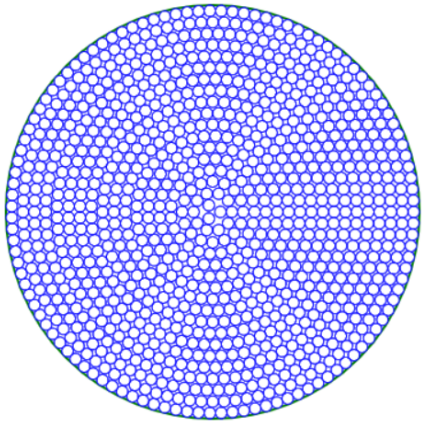


Figure 3.11: Seed distribution on a circular plate with 10cm diameter assuming 2,75 mm seed thickness

3.2.2. LED layout

A plate full of 1024 seeds is prepared, now it's time to establish the dose of UV-C light to which the seeds are exposed. This is mainly determined by two of the following factors: the exposure time and the received irradiance, expressed in [W/m^2].

Exposure time

The minimum exposure time is determined by the minimal time needed for pathogens to be destroyed. This is more elaborated in the state-of-the-art analysis 1.2.4

Irradiance

The dose is, next to the time that the seed gets exposed, also dependent on the irradiance of the spot on the whole plate where it's the lowest. It's not possible to base the dose on the mean value of the irradiance of the whole plate since then a risk arises that the seeds with the lowest irradiance don't get irradiated enough. Thus, for a defined minimal dose and as low as possible exposure time for disinfection, since the less this is, the more efficient the overall work process is, this lowest irradiated point must be as high as possible. Since a higher irradiance in another spot is useless, it was decided to focus on finding an irradiance that's as much as possible evenly distributed across the surface of the plate.

To break down the physics behind the irradiance of a light source in short, one should first realise a few things:

- The further away an irradiated spot is from the light source, the lesser the intensity
- The closer a spot is to the perpendicular bisector of the light source, in other words, the lesser the angle between this bisector and the distance light passes from source to destination, the bigger the intensity. This angle is also known as the zenith angle [22]. Thus, a spot that is exactly perpendicular to the light source, will have the biggest intensity of all spots on the plate.

In order to keep in mind the first behaviour, the panel with LEDs should be hung as close to the plate with seeds as possible. However, it should also be hung high enough so that this plate with seeds could

easily be taken out, and so that there is enough space for the seeds to vibrate enough, as is discussed in general design choices 2.4.1. Moreover, the higher the LED is, the bigger the part of the plate that it is able to irradiate is, but with lesser intensity. This is observed experimentally with the simulation, as shown in the figures below. From left to right the irradiance is simulated with LEDs hanging 10mm, 30mm, and 50 mm above the plate.

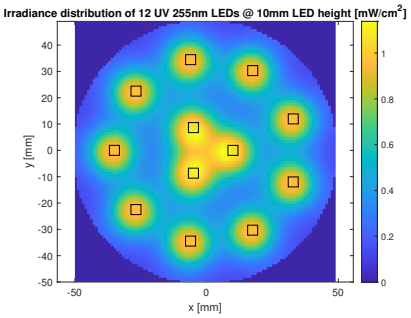


Figure 3.12: Intensity distribution of 12 255 nm UV-C LEDs hanging 10mm above a circular plate

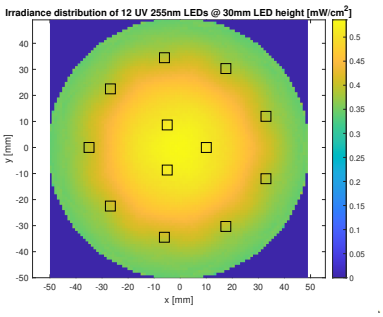


Figure 3.13: Intensity distribution of 12 255 nm UV-C LEDs hanging 30mm above a circular plate

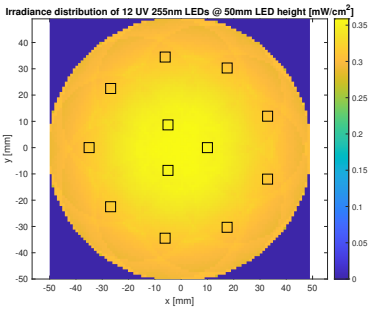


Figure 3.14: Intensity distribution of 12 255 nm UV-C LEDs hanging 50mm above a circular plate

With this all in mind, it was decided during the design process to set a height of 30 mm. In simulations, this was also reasonable, since the cutback in uniformity is compensated with the higher intensity.

The second behaviour that was mentioned is dependent on the radiation pattern of a LED. As was described in the general design choices 2.4.4, three LEDs are going to be used for the four wavelengths that are needed for disinfection. For two of the three specific UV-C LEDs that were chosen for this design, radiation patterns could be extracted from the datasheets for simulation use. The third radiation pattern was nonexistent in the datasheet and therefore chosen to test in the real design, however, this pattern is expected to be similar to all the radiation patterns of UV-C LEDs. The patterns that were acquired are visualised in a polar plot with the normalised intensity on the radius axis and the angle between the LED's trajectory and its perpendicular bisector on the angle axis, one for the 255 nm wavelength LED in figure 3.15 and one for the 275 nm wavelength LEDs in figure 3.16. These patterns were transformed into Matlab value matrices with steps of 1 degree. This conversion code can be found in the report of this project's first subgroup in the appendix [18].

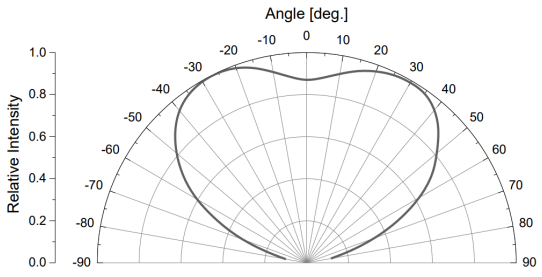


Figure 3.15: Spatial distribution of a 255nm wavelength LED, with the normalised intensity I expressed against the zenith angle θ [23]

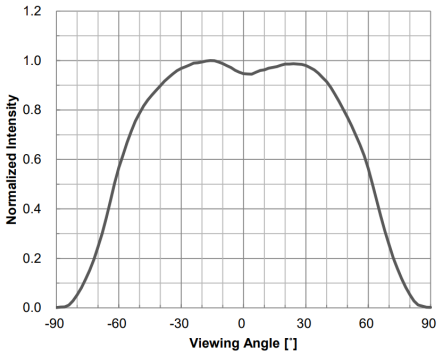


Figure 3.16: Spatial distribution of a 275nm wavelength LED, with the normalised intensity I expressed against the zenith angle θ [24]

Mathematically, the irradiance is given by the incident radiant flux per unit area of a surface, where the surface is still defined as a line perpendicular to the LED's trajectory, as is illustrated by dA in figure 3.17. The formula states as given in equation 3.1 [22]. This can be rewritten to the ratio of the radiant intensity and the squared radius of the LEDs corona, ergo distance between the LED and the spot that

is calculated [25] [26].

$$E = \frac{\delta\Phi}{\delta A} = \frac{I}{r^2} \tag{3.1}$$

However, this is the absolute intensity distribution, and from the datasheets only the normalised intensity distribution is available. From theory [27], the absolute intensity distribution can be calculated by using the LED flux value Φ_0 provided by the manufacturer and the normalised intensity distribution I_{norm}, as given in equation 3.2:

$$I(\theta,\phi) = \left[\frac{\Phi_0}{\int_{4\pi} I(\theta,\phi)_{norm} d\Omega}\right] I(\theta,\phi)_{norm} \tag{3.2}$$

Rewriting equation 3.1 [22] with the formula to calculate the absolute intensity distribution 3.2 and the area defined by the angle and the position of the spot with respect to the LED, the following expression is established:

$$E(r,\theta,\phi) = \frac{\Phi_0 I(\theta,\phi)}{r^2 \int I(\theta,\phi) d\Omega} \tag{3.3}$$

Since our illuminated spot isn't a tilted area as shown in the illustration in figure 3.17, but a spot on a horizontal plate, it's clear that the real surface is the calculated surface divided by the cosine of the zenith angle.

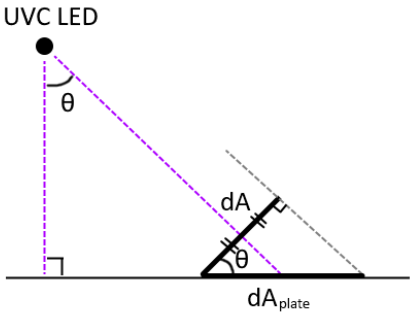


Figure 3.17: Schematic representation of a LED's light trajectory and its illumination on a projected area A_{plate} and zenith angle θ [22]

Implementing the area of the plate into equation 3.3 leads to:

$$E(r,\theta,\phi) = \frac{\Phi_0 I(\theta,\phi)\cos\theta}{r^2 \int I(\theta,\phi) d\Omega} \tag{3.4}$$

Now that it's established how to calculate the irradiance of each spot on the plate that is illuminated by one LED, it's time to simulate this effect in MatLab and in combination with more LEDs. To simulate the irradiance of the plate where multiple LEDs are present, you have to add up the irradiance of each LED that illuminates that spot, for every spot on the plate.
Fortunately, there is already existing MatLab code for this kind of purpose, as first was created by M. Brouwers and R. Bokhorst in [22]. This code was changed to apply to our requirements and included in appendix A.1. In the following section, the main function will be mainly explained, and also what the adaptations are for our implementation.

LED layout for proportionate irradiance

Some of the parameters for simulation were already predefined by design choices, so these were not changed during researching what the best LED layout would be. These non-changeable parameters, based on the requirements, are as follows:

- $h_{led} = 30mm$

  - The height of the LEDs with respect to the plate
- $\vec{\theta}_{tilt} = 0°$
  - The tilt angle of the LEDs in each ring, this is in our implementation zero everywhere, since the positive impact of this feature isn't applicable here
- $r_{filter} = 50mm$
  - Radius of the filter, which in the context of the seed UV-C disinfector is the radius of the plate with seeds, which is 50 mm
- LED size = 3.8 mm for both the 255 nm and 275 mm LEDs
- Grid dimension = 100; This determines the resolution of simulated irradiance
- Grid length = 100; This determines the size of the grid, and thus also the resolution of simulated irradiance
- Reflection coefficient = 0.85, in the case of aluminium. This is the cheapest, most easily accessible and thus most useful solution for UV-C light

The rest of the input parameters are changed with each simulation to analyse what the best layout would be. They are defined as follows:

- n_{ring} - number of rings in the layout
- $\vec{n}_{LED}$ - a vector in which each element defines how many LEDs there are in each ring
- $\vec{r}_{LED}$ - a vector in which each element defines what the radius is of each ring with respect to the centre
- Radiation pattern - a matrix variable with the spatial distribution of the specific LED which is to be simulated
- Φ_{LED} - Radiant flux of the LED which is to be simulated, which is 3.5 mW for 255 nm and 11.5 mW for 275 nm
- Reflection - a boolean variable to simulate with or without reflection
- Reflection resolution - the number of degrees which will be the steps in which the program calculates the reflection, influences the resolution
- Create a plot - a boolean variable to create a plot
- Plot all LEDs - a boolean variable to plot the place of LEDs of all other wavelengths too with a rotation respective to the current LED layout

The whole implementation of the code can be found in appendix A.1.2.

3.2.3. Circular plate simulations

From previous research on different topics, but with the same implementation of UV-C light in circular form [22], a similar layout was first tested for simulation. This consisted of three rings and a centre LED which has a place on a very small circle so that the centre LEDs of the other three wavelengths could fit on the LED plate too.

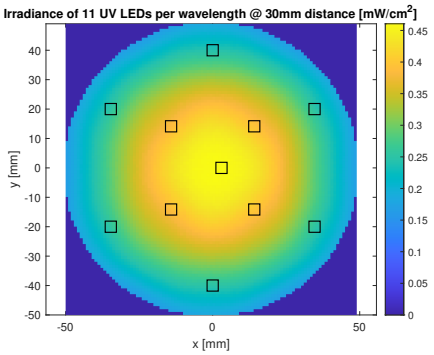


Figure 3.18: Intensity distribution of 11 255nm UV-C LEDs on a circular plate, with 3 rings with respective radii [3,20,35] cm and [1,4,6] LEDs per ring

Soon it was clear that it works already better than for a square plate, since the lowest intensity point is around $0.2mW/cm^2$, while for the square plate, this was $0.1mW/cm^2$ or lower as in figure 3.9. However, this could clearly still be improved. In order to still stay on roughly the same amount of LEDs, due to the high cost of UV-C LEDs but also the small space on the PCB, it was decided to increase the number of LEDs in the outer ring and reduce the number of LEDs in the inner rings.

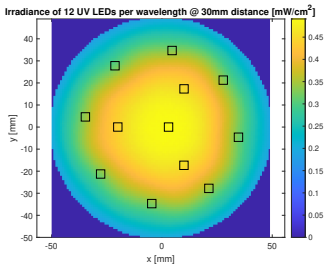


Figure 3.19: Intensity distribution of 12 255 nm UV-C LEDs on a circular plate, with 3 rings with respective radii [3,20,35] cm and [1,3,8] LEDs per ring

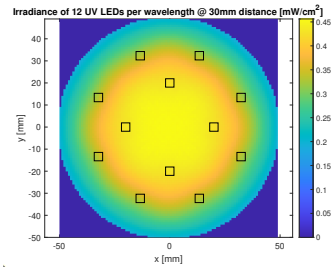


Figure 3.20: Intensity distribution of 12 255 nm UV-C LEDs on a circular plate, with 2 rings with respective radii [20,35] cm and [4,8] LEDs per ring

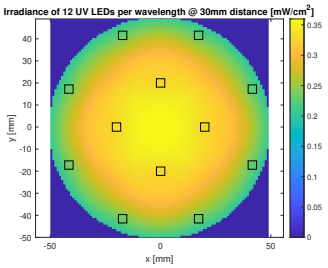


Figure 3.21: Intensity distribution of 12 255 nm UV-C LEDs on a circular plate, with 2 rings with respective radii [20,45] cm and [4,8] LEDs per ring

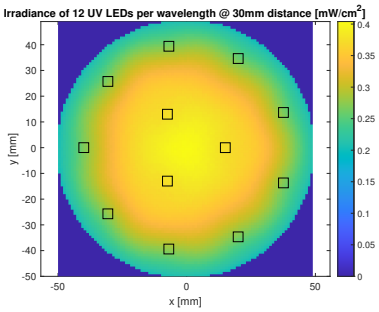


Figure 3.22: Intensity distribution of 12 255 nm UV-C LEDs on a circular plate, with 2 rings with respective radii [15,40] cm and [3,9] LEDs per ring

After doing simulations with various layouts, the layout in figure 3.21 seemed to have the best uniform intensity distribution, however, the LEDs were too close to the side which would have given complications with the PCB design and potentially with heating. In the end, the layout from figure 3.22 with two rings, three LEDs in the inner ring and 9 in the outer ring, with radii of 15 cm and 40 cm, was the end decision for implementation. The uniformity was still not perfect, however, that is without taking into account the reflection of the sides and the reflection of the lights from the bottom plate on the upper reflection plate. Implementing a thin aluminium plate on the inner side of the cylindrical disinfection chamber would result in UV-C light reflecting from the sides with a reflection coefficient ranging from 0.65 from [28] and [29] to 0.85 [30]. Taking the average of the values found in these papers, the reflection coefficient of 0.75 was implemented in the simulation. This resulted in the uniform intensity distribution as shown in in figure 3.25.

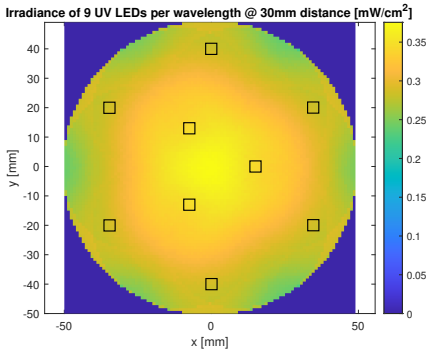


Figure 3.23: Intensity distribution of 9 255 nm UV-C LEDs on a circular plate, with 2 rings with respective radii [15,40] cm and [3,6] LEDs per ring, and a reflection coefficient of 0.75

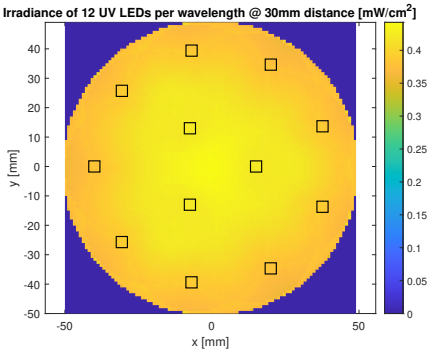


Figure 3.24: Intensity distribution of 12 255 nm UV-C LEDs on a circular plate, with 2 rings with respective radii [15,40] cm and [3,9] LEDs per ring, and a reflection coefficient of 0.75

Choosing 9 LEDs for the outer ring, however, wasn't only trial-and-error. The final layout should keep in mind that two other LED types, with three other different wavelengths, should also fit into the PCB with LEDs. Choosing a number of LEDs in the outer ring that is a multiple of the number of LEDs in the inner ring gives the layout the advantage of proportionate rotation. In other words, when rotating both rings at the same time around the origin, the LEDS in neither of the rings should overlap. By assigning three LEDs to the inner ring, since less wouldn't be uniform anymore and more isn't necessary, that leaves the outer ring to be six LEDs as in figure 3.23 or nine LEDs as in figure 3.25 since more than that wouldn't fit. In the end, nine LEDs were chosen for a more uniform distribution.
By plotting the other 24 LEDs, twelve per LED type, in the same graph, the best rotation for each LED type with respect to the first LED type could be established. The rotation should be multiple of a third of the angle difference between LEDs in the outer ring since then they will be as far as possible from each other. This means a multiple of $\frac{1}{3} \cdot \frac{360°}{9} = 13.333°$. However, the LEDs in the inner ring should also have enough space between them, thus this adds other criteria to the problem definition. The second LED type should be rotated k times 13.333°and the third LED type should be rotated l times 13.333°, with k and l being large enough so that the LEDs in the inner ring don't overlap. With k being 2, the LEDs already don't overlap, so with l being either 4 or 7, the two other LED types are rotated 26.667°and respectively 53.333°or 93.333°. For the latter, both choices are exactly the same, so it was arbitrarily decided to go for 26.667°and 93.333°

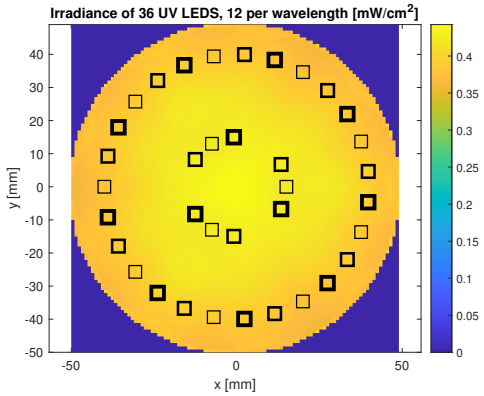


Figure 3.25: Intensity distribution of 36 UV-C LEDs on a circular plate, with 2 rings with respective radii [15,40] cm and [3,9] LEDs per ring, and a reflection coefficient of 0.75

Another beautiful result of this circular LED layout is that the intensity distribution is the same when you rotate the layout since it's a circle. This gives this implementation another advantage over the layouts on a square plate since those rotations were all different from each other.

3.3. Optimization

With the requirements that are specific to the project, the optimization script from Bakker and Brouwer's research [22] was changed and ran to search for the optimized radii of the rings, with a fixed number of LEDs per ring, since this is important for rotation. This tweaked optimization script can be found in appendix A.1.3. In short, the script iterates through every possible radii for the inner and outer ring between the set boundaries for a number of rings, LEDs per ring and for the minimal distance between the rings and between outer ring and border. After each iteration, the script calculates the lowest and highest present intensity on the plate. Then it calculates the ratio of these extremes for uniformity observation, and also the ratio of number of LEDs and the lowest intensity measured. The settings with the best ratio of the irradiation intensity extremes are saved each time if there's a better ratio with the new iteration. A part of the output of this optimization script is shown in appendix B.

From this process the result was that the best parameters with the set boundaries are 18mm for the inner ring with 3 LEDs and 45mm for the outer ring with 9 LEDs.

4

Power Distribution

As described in the introduction, the machine consists of different modules with different functions. Logically, all the modules need the power to operate and perform functions that it is designed for, for example rotating a motor for the Motor Controller or turning on the LEDs to expose the plant seeds to UV-C light. All these modules shouldn't be simply connected to the power socket, since there would be a voltage across the system that is too high and overall many other functions would be impossible to implement, many of them being for safety. Resulting from this, it is decided to buy a power supply that converts the voltage of 230V AC from the socket to 24V DC and to make a Power Supply Unit, a self-designed PCB for power distribution and useful functions that are explained in section 4.1.

4.1. Requirements

With the decision being established on what the communication and way of power distribution would be, the next step is to set the requirements and what functions the PSU should be able to do. These requirements are established as follows:

- Distribution of power to four different modules: Control Unit, Motor Controller and two LED Drivers
- Shutdown of power supply to the motor controller and the two LED driving modules at a specified digital signal received from the Control Unit
- Conversion of 24V to 5V and 5V to 3.3V for internal use (5V for the relay and op-amps, 3.3V for the comparator circuit)
- Over-current detection and communication of this to the Control Unit
- Low-pass filter for blocking high-frequency current from the LED Driver

The requirements that require more explanation will be explained in detail in the subsections below.

4.1.1. Power distribution and shutdown

As explained in the introduction, the system uses Ethernet connections for communication between all the modules. Two of the eight possible wires within an Ethernet cable are reserved for providing power to all the modules. In addition to this, the ground of the modules is also connected to the ground of this module.

At some point in using the machine, a user could turn on the machine, but not activate the LEDs and motor, because of a failure in the system, like an overload current, or another reason to abruptly stop the power distribution. In this case, there should go power to the Control Unit, which includes a screen for setting user preferences, but not go to the Motor Controller and LED Drivers. Since there still is going current to the Control Unit, different settings can be chosen for the motor or LEDs and/or the system can be debugged. In both cases, it's clear that the power to Control Unit and the other modules should be separated. Moreover, the power to the other modules should be able to be turned off whenever the Control Unit says so, for example via an analogue active high signal. This means

that whenever the voltage that is across a specified pin coming from the Control Unit is high, the PSU distributes power to the other modules, and whenever the voltage across the pin is low, in other words: close to zero, it should stop distributing this power. How this could be designed is explained in section 4.2.

4.1.2. Over-current protection

In the PSU, the load of the whole system is connected since it distributes power to all four modules, consisting of the Control Unit, Motor Controller and two LED Drivers. Whenever the system uses too much power, a bigger current goes through the system due to Ohm's Law:

$$P = I \cdot V = I^2 \cdot R \tag{4.1}$$

This increase in power consumption can be, for example, caused by a malfunctioning motor or LEDs with an internal short circuit. When such a situation happens, the resistance decreases, more current flows and therefore more power is used. This leads to an increase in heat dissipation and raises the chances of the destruction of components and maybe even total failure. This can also damage the 230V to 24V AC to DC power supply. In conclusion, to protect the system, an over-current detector is crucial to be designed. Moreover, not only needs the PCB to detect it, but it should also be able to communicate this to the control unit as a digital signal via a specified pin. The control unit should in turn process this and send a signal back through another pin to disable the distribution of power to the other modules. It's important to do this disabling of power distribution when there's an over-current because the control unit is able to store all the specified settings and measured values from sensors at the moment of load overload in an SD card. As a result, the user can analyse this data from the SD card and make changes to the system so that the overload doesn't happen again. The design of this over-current protection is in detail explained in section 4.2.

4.1.3. Low-pass filter

As described in the report about the LED driving modules [18], the boost converter has a fast switching behaviour. Thus, the current going to these modules fluctuates with a high frequency too. Their module has an over-current trigger system, however, with these current fluctuations caused by the boost converters, the overcurrent warning can get triggered oftentimes because of the sudden, short-lasting, high current at on-time of the boost converters. In order to prevent this over-current and thus protect the system, a low-pass filter is needed to only pass the low frequencies. This way, the over-current protection won't be triggered and the boost converters don't step up the voltage too fast, but gradually. Another useful application of the RC filter is when looking at the PWM signal components coming in from the modules aside from the DC components. These come from their ATMEGA components which are set to operate at 980Hz. Because all the PWM signals come together in this PCB, they compensate each other a bit, but there isn't a guarantee it's a perfect compensation that would result in a constant high voltage. In order to make it an approximately straight line voltage level nonetheless, only low frequencies should be passed [31] with the use of an RC filter [32]. In figure 4.1, a plot of the PWM signal is shown where a fast-Fourier transform has been applied to show the frequency peaks. This is to show that in order to only leave the DC component and filter out the high frequencies, a low-pass filter has to be applied. The result of this technique is shown in figure 4.2. The resulting voltage rises slowly during the on-time and discharges slowly during the off-time. Because this happens much slower than the original PWM signal, the resulting average voltage is around half of the high voltage, which is $3.3V/2 = 1.65V$. In the long run, this results in fairly constant voltage as shown in figure 4.3.

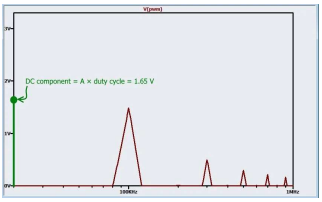


Figure 4.1: Fast Fourier Transform of a PWM signal with a spike at the carrier frequency, its harmonics and the DC component [32]

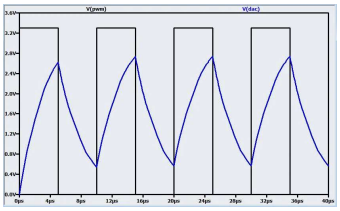


Figure 4.2: Transient response of a PWM signal in combination with the resulting PWM signal after RC filter, zoomed in [32]

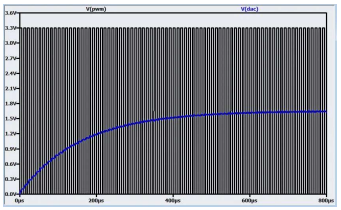


Figure 4.3: Transient response of a PWM signal in combination with the resulting PWM signal after RC filter, in the long run [32]

4.2. Design process

By combining all these requirements, some careful considerations should be made. In terms of safety, it's very important that the power distribution always goes as planned. Of course, this applies for example also to turn on a LED, however, if that doesn't go as planned, in the worst-case scenario, the LED gets destroyed and a new one should be placed. However, when speaking about the power distribution, in the worst-case scenario the whole system gets overloaded and all sensitive components in the whole system get destroyed. Before focusing on the design choices of specific sections on the PSU, an overall schematic can be found in appendix C. In section 4.3, the resulting PCB design and 3D model are shown. Both the schematic and resulting PCB design and 3D model can also be found on a bigger scale respectively in appendix C, D and E.

4.2.1. Power distribution

As described in the general design choices 2.4.2, it was decided by the whole group to use a similar strategy to Power-over-Ethernet connections for, among other things, power distribution to the four modules. Since the connection with the Control Unit is a bit more complex than the connection with other modules since this connection requires two-way communication, it was decided to put the Ethernet port for Control Unit on the other side. A simple schematic of the four ports is shown here below.

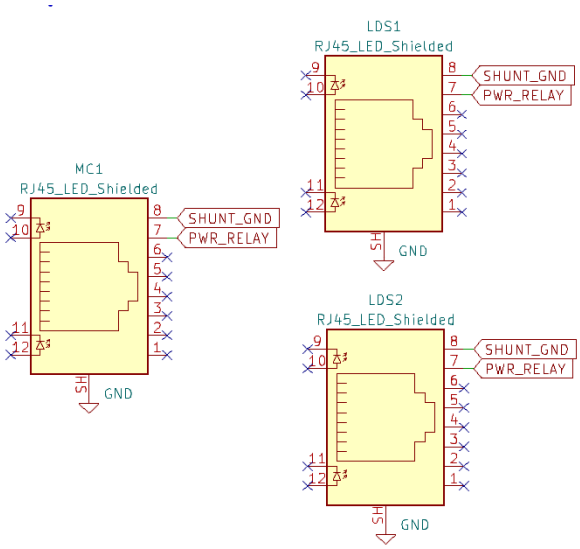


Figure 4.4: Schematic representation of the output Ethernet ports of the PSU

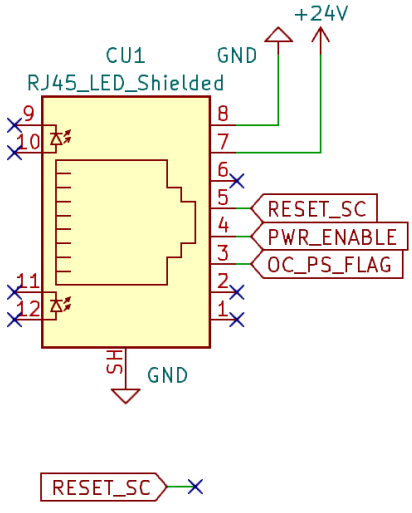


Figure 4.5: Schematic representation of the Ethernet connection between the PSU and the CU

For the Ethernet ports that are connected to the Motor Controller module (MCU) and the two LED Driving & Sensing modules 4.4, only two pins are used. One is connected between the ground of the modules and the over-current protection of the PSU 4.2.4, defined as the SHUNT_GND flag. The other

pin is not directly connected to 24V but to the Relay module 4.2.2, indicated by the PWR_RELAY flag. The other pins are not disconnected, since no communication is needed between the PSU and the MCU and LDS.

For the Ethernet port that connects to the CU 4.5, six pins are used. Two of them are connected to the 24V and ground connections of the PSU, namely numbers 7 and 8 respectively. Then there are two pins that are reserved for the two-way analogue communication between the PSU and the CU with the use of error flags, namely pin numbers 3 and 4.
Pin number 3, with the flag name OC_PS_FLAG, is used by the over-current detection module by setting the pin either low or high voltage across it, dependent on whether there's an overload or not, more about that in subsection section 4.2.4.
Pin number 4, indicated by the flag name PWR_ENABLE, is used by the relay module. At either low or high voltage across this pin, the relay module reacts accordingly, as stated in subsection 4.2.2.
In the end, pin number 5 is used by the control unit to send a reset signal, however, this was not needed for this implementation, so this remains disconnected.

4.2.2. Relay module

The most important function of the Power Supply is to distribute the power to the modules and stop this distribution when commanded by the Control Unit. This is done by using an N-channel power MOSFET, a relay and a corresponding circuit to regulate this.
For debugging and user-friendliness it was chosen to include two LEDs: one LED that indicates whether there's power in the module, indicated in figure figure 4.6 by D2, and one LED that indicates the status of the relay, whether it's open or closed, indicated by D3. Both require a resistor put in beforehand to compensate for the voltage drop across the LED.
Now comes the circuit for distribution at play consisting of a MOSFET, relay, flywheel diode and a resistor to control the inrush current from the PWR_ENABLE flag. The latter value is chosen as such that with a high voltage of 5V and a usual supplied current from a controller of 10mA, a resistance of 500Ω is needed. In order to minimise the difference in components for ordering and easiness of soldering, the same resistance as for R6 and R7 is chosen.
As explained in [33], when the MOSFET will turn off, the coil within the relay will freewheel, resulting in a rising drain voltage, in other words, a reverse voltage is induced. Placing a diode clamps this voltage and thus protects the MOSFET and the circuit where the flag comes from, the microcontroller from the Control Unit. Next, a MOSFET is connected between the flag and the relay to assure that the relay driver is in the off-state when the flag is low, by creating an open circuit. And vice-versa, make sure the relay is in its on-state when the flag is high [34]. In this case, the circuit, including the relay's coil and status LED, closes, and a switch within the relay gets magnetically attracted to the coil. This results in the 24V source being connected to the PWR_RELAY flag that delivers power to the MCU and LDS modules.

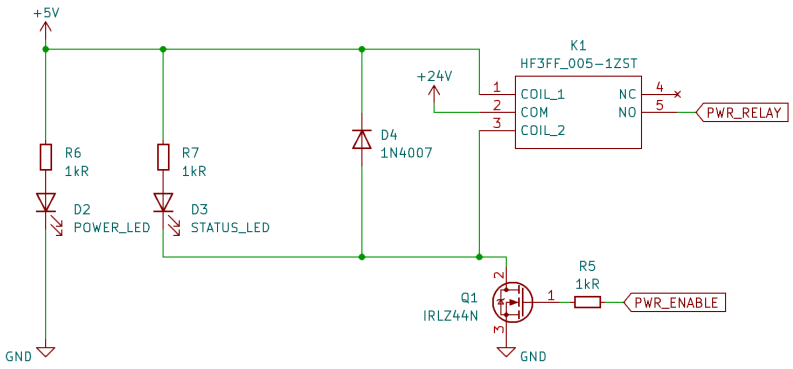


Figure 4.6: Schematic of the relay module

4.2.3. Voltage conversions

Conversion of 24V to 5V

For some components within the PSU, a voltage of 5 Volt is needed to be operational. This is applicable for the relay module section 4.2.2 and the operational amplifier section 4.2.4. To convert 24V to 5V, the K7805-1000R3L DC-DC converter was chosen. From this component's datasheet [35], the circuit shown in figure figure 4.7 is created with capacitance values of 10μF and 22μF for C2 and C3.

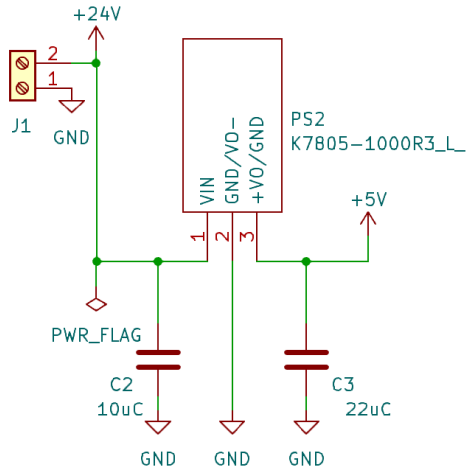


Figure 4.7: Schematic of the conversion from 24V to 5V

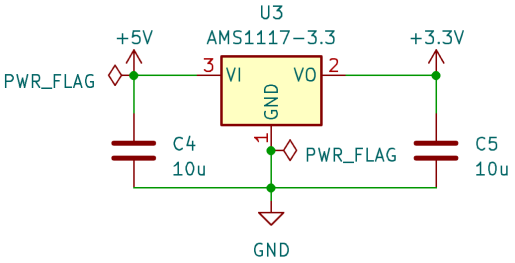


Figure 4.8: Schematic of the conversion from 5V to 3.3V

Conversion of 5V to 3.3V

For the comparator circuit as described in section section 4.2.4, a reference voltage of 3.3V is needed. This is chosen to be done with the AMS1117 series 5V to 3.3V Linear Regulator with two capacitors, C4 and C5, of 10μ F [36]. This corresponding circuit is shown in figure 4.8.

4.2.4. Over-current protection

The total load that is connected to the PSU comes from the four modules connected to it. For the control unit, it was chosen not to include this in the over-current protection, since there aren't any big risks of over-current in that system. For calculating whether the system doesn't require too much power from the power supply, the consumption by CU does need to be calculated, as is done in table 4.2.4. The total power consumption is 122.16W, as can be seen from the calculation in table 4.2.4. This is well under the power rating of the 230V to 24V Ac to DC power supply, which is 180W. For the other three modules, the two LDS modules and the MCU, the total power consumption during normal functionality should first be calculated, and then what it would be if too much current is going through the system.

Module	Calculation	Total power consumption
CU	$24V \cdot 140mA$	3.36W
LDS 1[1]	$3 \cdot \frac{(60V)^2}{500\Omega} + 60V \cdot (3 \cdot 200mA)$	57.6W
LDS 2 [1]	$3 \cdot \frac{(60V)^2}{500\Omega} + 60V \cdot (3 \cdot 200mA)$	57.6W
MCU	$12V \cdot 300mA$	3.6W
Total	$3.36W + 57.6W + 57.6W + 3.6W$	122.16W

During standard operation, the load on the system with which calculations will be made for over-current protection is 118.8W. Since 24V is delivered to the load, a current of $I = \frac{P}{U} = \frac{118.8W}{24V} = 4.95A$ goes back from the modules to the protection circuit, specifically, to the shunt resistor. The shunt resistor is placed between the parallel Ethernet ports of the three modules and the ground. The voltage over

[1] This is the summation of the discharge and peak current for all three different drivers for the three LED types, for more details refer to [18]

the shunt resistor is used to measure the over-current. The voltage over the shunt resistor can be calculated from the calculated current going to the shunt resistor via Ohm's law as shown below. The value of the shunt resistor is chosen to be 0.1Ω since this is common for low-current applications.

$$V_{shunt} = I_{load} \cdot R_{shunt} = 4.95A \cdot 0.1\Omega = 0.495V$$

A non-inverting amplifier circuit should amplify this voltage so that it's high enough for the comparator circuit. In this circuit the amplified voltage is compared with 3.3V. When the input voltage is higher than the reference voltage, then the output voltage is $+V_{cc}$, in other words around 5V - the driving voltage of the op-amp. This should happen when there's over-current, since then there's a higher current than normally, and thus a higher voltage than the threshold voltage set to be compared with the reference voltage. Vice versa, if the input voltage is lower than the reference, the output voltage is low, around 0V. This is the case during standard operation, since the current is then lower than the set limit. This output is then connected to a pin on the Ethernet port which is connected to the CU which reads this flag.
In order to have a margin from which moment the over-current flag becomes low, it was chosen to activate this when the power consumption is 25-30% higher than at standard operation (118.8W), namely 150W. This results in doing the same calculations as before, but now with 150W. The resulting calculated voltage over the shunt resistor is:

$$V_{shunt} = I_{load} \cdot R_{shunt} = \frac{150W}{24V} \cdot 0.1\Omega = 0.625V$$

This voltage should be amplified to 3.3V with the non-inverting amplifier, as this should be the threshold for the overcurrent. The gain should be:

$$A_v = \frac{V_{out}}{V_{in}} = \frac{3.3\ [V]}{0.625\ [V]} = 5.28$$

The gain of the non-inverting amplifier can be computed via the standard formula [37] and the resistor variable names as shown in figure 4.9:

$$\begin{aligned} V_{out} &= (1 + \tfrac{R_3}{RV1})V_{in} \\ A_v &= \tfrac{V_{out}}{V_{in}} = 1 + \tfrac{R_3}{RV1} = 5.28 \\ \tfrac{R_3}{RV1} &= 4.28 \end{aligned}$$

For the fixed valued resistor R3 a value of $56k\Omega$ was chosen, arbitrarily but also so it's high enough to implement the correct gain. With the variable resistor RV1 the value could then be later be tweaked to be perfectly aligned with the system. From calculations, the value should be around $RV1 = \frac{56k\Omega}{4.28} \approx 13k\Omega$. This is between the rated values of the 50kΩ potentiometer, which is a variable resistor.

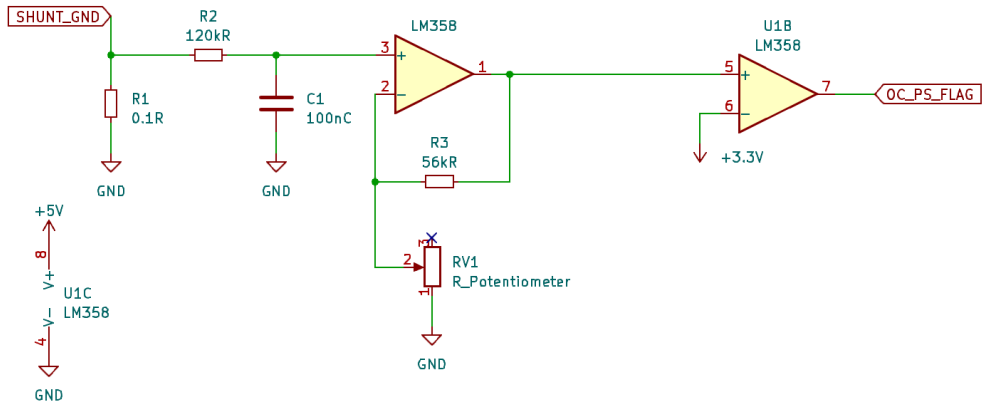


Figure 4.9: Schematic of the over-current detection

4.2.5. Low-pass filter

As was said before, this filter is here to filter out the high frequencies, in other words, only passes low frequencies. This is done by taking the output of an RC circuit off the capacitor. Hereby the cutoff frequency is obtained by setting the magnitude equal to $\frac{1}{\sqrt{2}}$, resulting in the equation hereunder [37]. With the value of the resistor and capacitor shown in figure 4.10, a cutoff frequency of 83.3rad/s, or 13.3Hz and a time constant $\tau = R \cdot C = 120k\Omega \cdot 100nF = 0.012s$ are obtained. This frequency is much lower than the PWM signal that is used in the system, by the motor controller and LED Drivers, namely 980Hz.

$$H(\omega_c) = \frac{1}{\sqrt{1+\omega_c^2 R^2 C^2}} = \frac{1}{\sqrt{2}}$$

$$\omega_c = \frac{1}{RC} = \frac{1}{120k\Omega \cdot 100nF} = 83.333rad/s = 13.26Hz$$

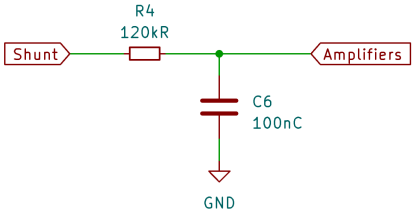


Figure 4.10: Schematic of the low-pass filter

After calculation, this was tested using LTspice and the correct values for the PWM frequency and RC filter. In figure 4.11, the behaviour of the resulting output voltage is shown in detail between a few PWM signal peaks. Because the RC filter has such a smaller frequency than the PWM signal, the voltage fluctuates only approximately 0.3V. In the long run, it looks like a constant voltage, as shown in figure 4.12. In practice the same would happen if only one PWM signal were to be received, however, as described before, several components come in, so they already compensate each other. This would only mean that the fluctuation decreases even more, resulting in a smooth constant voltage, which is the whole function of the low-pass filter.

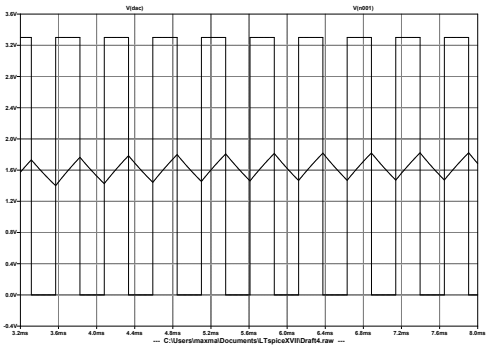


Figure 4.11: Transient response of the resulting PWM signal in combination with the resulting PWM signal after RC filter, zoomed in

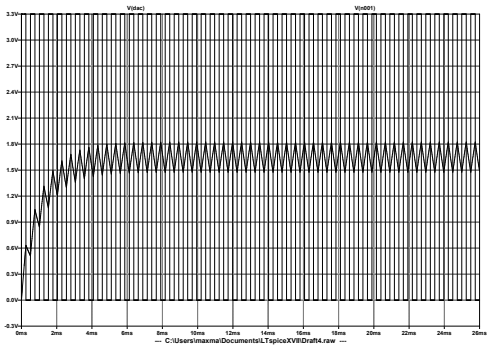


Figure 4.12: Transient response of the resulting PWM signal in combination with the resulting PWM signal after RC filter, in the long run

4.3. Result

4.3.1. PCB design

Combining these schematics, a PCB design was made. Since it's a relatively simple design, it was chosen to make a two-layered PCB. The bottom copper layer is reserved for the ground. The final design can be seen in figure 4.13 and in a bigger view in appendix D. Figure 4.14 gives the 3D model of this PCB design. Here also a bigger version can be found in appendix E.

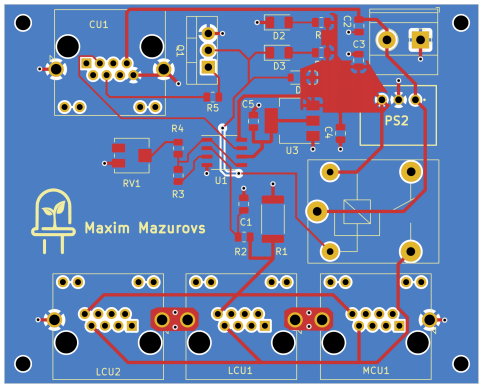


Figure 4.13: PCB design of the PSU

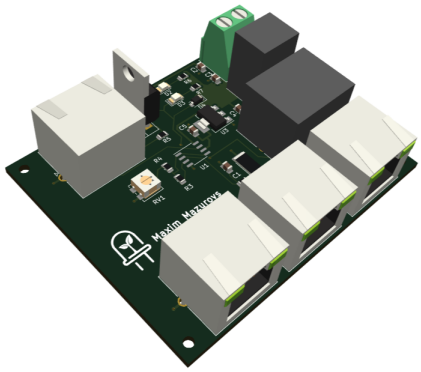
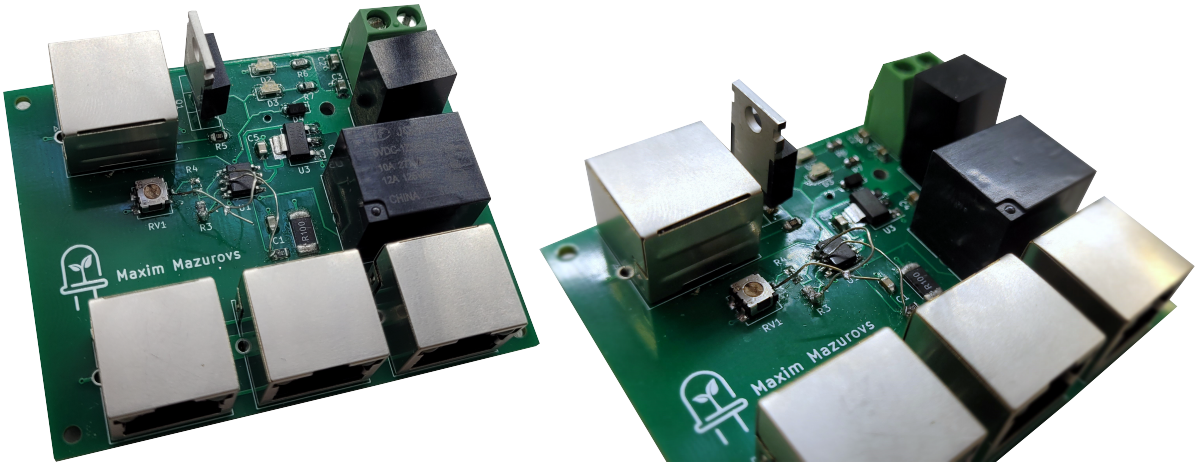


Figure 4.14: 3D model of the PSU

4.3.2. PCB assembly

The soldered PCB is shown in figure 4.15b. In chapter 7.1.2, the conclusions of the design are listed and what should be improved in a future version of the design.



(a) Top view

(b) Side view

Figure 4.15: Final Power Supply Unit PCB

5

Motor Controller

5.1. Requirements

- Communication with control unit through i^{2}c
- Able to power a DC motor and change its speed according to control unit input
- Over-current protection for the DC motor in case of failure (hardware & software)
- Easy expansion to add components in hindsight
- Programmable
- Can be mounted to the bottom plate

5.2. Design process

Some boundaries to work within have been set by the overall device design. All subgroups agreed upon the following standards in order to be able to connect all PCBs together and function without the hassle:

- Each module is equipped with at least two Ethernet (RJ45) connectors, one for power and one for communication to the main control unit.
- The communication protocol between micro-controllers is I^{2}C.
- All PCBs will be powered with 24 volts DC.

5.2.1. Main design choices

Voltage conversion

The PCB will make use of the following voltages, conversion for these will be done onboard:

- 3.3 V (Used by comparator)
- 5.0 V (Used by microcontroller & relays)
- 12.0 V (Used by motor)

The power supply will supply all PCBs with 24 V over the Ethernet connector. Therefore, there will be a need for some power conversions. The first step is 24V to 12V conversion. The 12 Volts should be able to power the DC motor, which should allow for roughly 12 Watts i.e. 12V at 1A (explained later in 5.2.3). This is achieved by the K7812-1000R3 according to its datasheet [38], this power converter is able to deliver these demands. The circuit is given in the figure in appendix C.6
The 5.0 V is needed by the microcontroller, it does not demand any significant power according to its datasheet [39]. Conversion is done by the circuit given in the figure in appendix C.5. It makes use of the AMS11175.0 IC [36]. Notice the microcontroller should be programmable, therefore a USB port is added in the design. In order to make sure these two power sources do not interfere with each other, a switching circuit is created by an op-amp that compares the power with VBUS, which is the voltage from the USB connector. This circuit is also provided with reverse polarity protection shown in the figure, squared in red. Notice this part needs some attention in the result in section 5.3.

The 3.3 V is used by some comparators in the design. The final and full design is given in appendix C. The 3.3 V is conveniently done by the FT232R UART USB interface IC, which has a 5.0 V input voltage. Its datasheet is also given in [40].

Micro-controller

As the motor controller should be programmable, ready for expansion and needs a PWM output for motor speed control, a microcontroller such as the ATmega328P-A [39] was a convenient choice because it can fulfil all of the mentioned requirements and wishes. It features analogue and digital pins like the well-known Arduino development boards, which can be used as inputs or outputs after programming. Because of this expansion will be relatively simple. The microcontroller is programmable by USB through the UART interface, which is connected according to the specifications.

Connections

As stated before, the motor controller makes use of two RJ45 connectors for power and communication. Their pin-outs are determined beforehand. The pin-outs for PSU and CU connections can be found in the appendix figures C.3 & C.4 respectively.

5.2.2. Expansion

As the motor controller uses a microcontroller (the ATmega328P-A), it is possible to add expansion easily by adding components in hindsight if necessary. These can even be controlled by the main control unit through the $i^2$2 protocol. This way, components can be added to improve the device. Some of the components that can be added are temperature sensors, a door sensor, and humidity sensors but also simple LEDs or relays.

Two relays and four LEDs are already implemented on the PCB for fast access if necessary. The two Relays are connected to pins D11 & D12 and the LEDs are connected to D14 through D17. Their connection is displayed in figure 5.1 and 5.2. The relays include two LEDs indicating power and status. The power LEDs are useful to determine if the PCB is connected to power and the status LEDs indicate whether the relays are on or off.

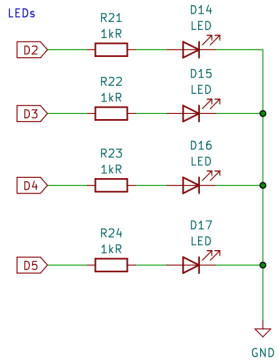


Figure 5.1: LEDs schematic

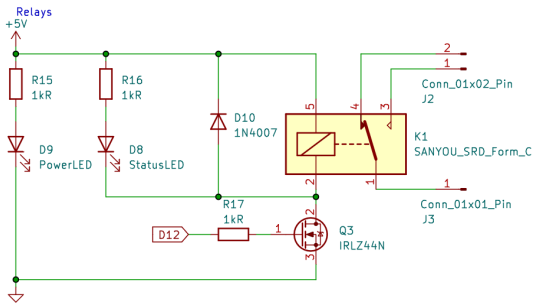


Figure 5.2: Relay schematic

5.2.3. Motor over-current protection

The selected motor is the RS360-Y DC motor [41]. This motor will be sufficient for its goal, which is to create a vibration that turns the seeds in the tray. The motor will be controlled through a 12V pulse width modulated current. This motor has a maximum current of 0.121 A when no load is given. When stalled, the motor speed is zero and the current is 1.493 A, following from the datasheet. This can generate heat and therefore can be harmful to the device in case of failure. Because of this, over-current protection is designed in hardware, as well as software in case one of the other fails. The protection circuit should activate when the current drawn is about 1 A. This current should never be drawn by the motor in this use case since there is no 'big' load on the motor.

The circuit given in figure 5.4 is the hardware over-current protection. It measures the voltage over the 0.1 Ω shunt resistor on the ground side. When the voltage is 0.1 V, the current is 1 A following from Ohm's law: $V = I \cdot R$. This is forwarded to the non-inverting input of an LM358 op-amp through a Low-pass filter to filter out some frequency components from the DC motor. In practice, the shunt resistor was measured to be 0.18 Ω, this resulted in a 0.18 V potential for a current of 1 A. This should then be amplified so it corresponds to a 3.3 V potential at the output. The amplification (A_v) follows the non-inverting op-amp formula according to [42].

$$A_v = 1 + \frac{R25}{(R26 + RV1)}$$

A potentiometer is added in order to be able to change the amplification later if necessary. When the potentiometer is set to 0, the gain $A_v \approx 18$, when set to 50$k\Omega$ it corresponds to a gain of $A_v \approx 2$, meaning the error flag's sensitivity can be set and customized, if necessary, the resistors can even be desoldered and repicked all together. The signal is amplified so the output can be connected to the input of a comparator (Also on the LM358 op-amp IC) that compares the voltage to the constant 3.3 V generated by the microcontroller. When the output voltage from the first op-amp is more than 3.3 V, The output of the comparator is set high. This flag can then be sent to the Control unit that can make a decision based upon this "error flag". It's even possible to feed this flag to a relay in order to make sure the motor is turned off immediately. Though, this will result in zero current pulling the error flag low and re-activating the motor. So this would need some extra design steps.

As stated before, the over-current protection should also be able to be handled through software. That's why the output of the first amplifier is also sent to the analogue pin "A0" of the microcontroller, which can process the information further. This way the microcontroller can also turn off the DC motor through software in order to prevent total failure. This is done by the lines code from 72 to 88 in the motor-controller code given in A.2. The micro-controller's analogue input pin "A0" ranges from 0 to 1023 [43], so the codes converts this range into voltage and then also current. This information is then sent to the Control unit.

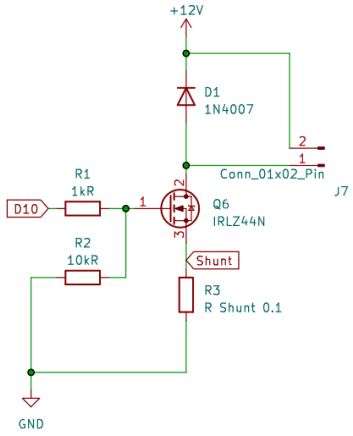


Figure 5.3: Motor control circuit schematic

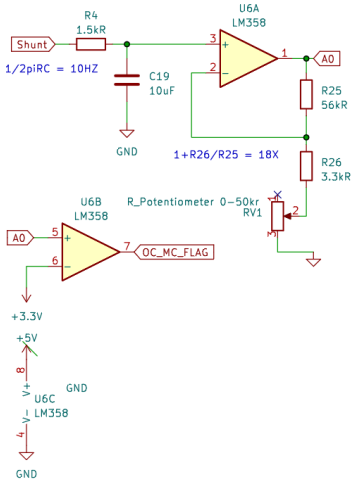


Figure 5.4: Current measurement circuit

5.2.4. Motor control

The circuit for driving a motor with PWM using the ATmega328P-A is given in figure 5.3. The motor will be connected the rightmost pins referred to as "J7". It uses an IRLZ44N MOSFET, more than capable of handling the maximum current of 1A, together with a flyback diode for motor protection and a shunt resistor which is used for current measurement, the microcontroller can generate a PWM signal to control the motor speed by the use of a MOSFET. The MOSFET serves as a switch, enabling or disabling the motor's power supply on the PWM signal, which mimics changing the mean power to the motor and in turn changing its speed, done by changing the duty cycle of the PWM signal. A flyback diode is connected in reverse-biased across the motor terminals to prevent damage from the back electromotive

force when the motor is turned off which is needed because the motor will still spin a bit longer when turned off.

The shunt resistor, placed in series with the motor's negative supply line, can give some sort of current measurement by measuring the voltage drop over a known resistance which is proportional to the current. The flyback diode connects the motor's positive terminal to the motor's negative terminal. Detailed circuit diagrams can be found in appendix C.2.

5.2.5. Communication

As stated before, communication between the motor controller and control unit is needed. This is done through I^2C. The lines SCL_MC and SDA_MC are the communication lines on pins 1 and 2 to the CU on the Ethernet port as seen in figure C.4. The communication code is given in appendix A.2. This code follows a communication protocol explained in detail in the control unit subgroup [19]. In lines 45 through 60, the information received is read and pointed to the correct handler function for each event such as setting motor speed.

5.3. Result

A 3D model of the PCB is given in figure 5.5. the expansion pins and relays are visible in 5.6. More information about the final design results can be found in the conclusion and recommendations for this module in sections 7.1.3 and 7.2.3 respectively. After some small changes in the connections, the motor control module could be successfully programmed and the motor could be controlled.

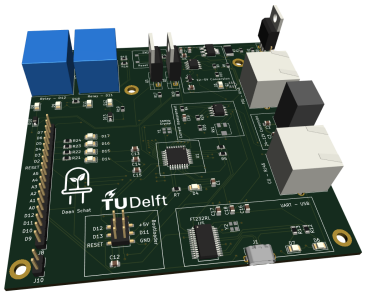


Figure 5.5: Motor control model - top

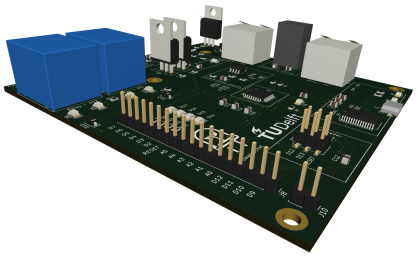


Figure 5.6: Motor control model - expansion

6

Prototype Implementation

6.1. Requirements

The UV-C disinfecting device can only be made with a proper plan of approach. Therefore a brief list of requirements is drafted for the main concerns in designing the device, some of these were previously determined by the general agreements, such as the seed movement:

- Should house all components securely
- Should not allow UV-C light to pass through to the user
- Seed housing should allow for slight movement/vibration
- Should be easy to operate
- Should be safe for the user

6.2. Design process

This part will go further into detail about the actual prototype design and considerations that go with it for some of the main iterations of the device.

6.2.1. V1, a first impression

This model, referred to as V1 in the project, gives a first impression of what a UV-C disinfecting device could look like. It sketches the basic idea of materials to use and LED placement without going too much into detail.

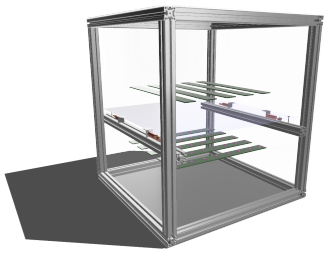


Figure 6.1: First iteration of the LED UV-C disinfecting device - V1

The one component that was selected that made it through every iteration of the device was framing: V2020 aluminium framing was selected for its strength and easy configurability. At this point in time, the basic idea of having to move/vibrate the seeds for even irradiation was already thought of and added to the design. This was done by placing vibration motors at the edges of the seed plate with decentralized weights on their axes. a figure of V1 can be found in 6.1.

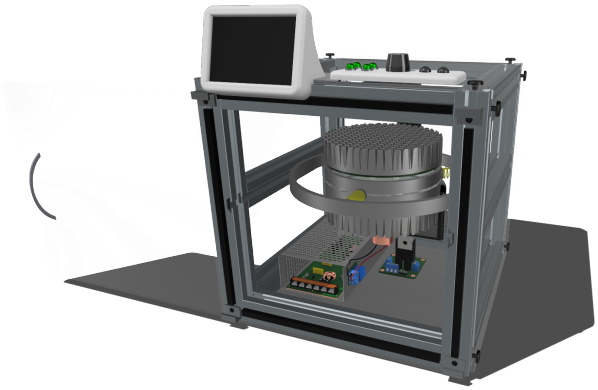


Figure 6.4: second main iteration of the UV-C disinfecting device

6.2.2. Quartz

The first iteration of the prototype still has many flaws, the casing is fully enclosed the PCBs are floating and not yet final and many components need to be selected. The first and one of the most important of which, is the seed plate. The main considerations for this are the ability to pass through UV-C light and price.

Not many, preferably transparent, materials let UV-C light pass through as shown in figure 6.2. But, one material does stand out, which is quartz. In figure 6.2, quartz has a transmission of about 90 to 95 % at wavelengths larger than 200 nm, which should be more than sufficient as a seed plate.

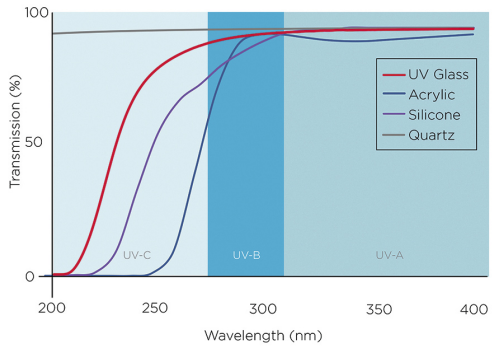


Figure 6.2: transmittance of UV glass, acrylic, silicone and quartz for different wavelengths of ultraviolet light [44]

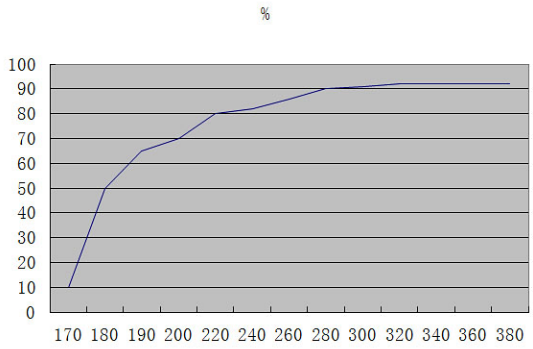


Figure 6.3: transmittance of quartz for different wavelengths of light from with wavelength on the X-axis, and transmission on the Y-axis [45]

As the general requirements and agreements in the machine suggested, a circular design was selected. Therefore a circular quartz plate was selected of 2 mm thickness for rigidity. This is implemented in V2 of the design, explained in section 6.2.3. The manufacturer's specification of the transmission of UV-C light is shown in figure 6.3. It does show a steep decline in transmission for wavelengths lower than 200. But the performance should be good enough as the transmission for 220 nm wavelengths is equal to or greater than about 80%.

6.2.3. V2, A more realistic approach

At this point in time, the first main components such as the power supply was modelled and a couple of PCBs got a better indication of size. Also, more components needed have been selected such as the previously mentioned quartz plate, but also the plexiglass casing to block any outgoing UV-C light [46]. The transmittance of Plexiglas (PMMA) against different wavelengths is also shown in figure 6.5. From this, the conclusion can be drawn that Plexiglas is safe to use against UV-C irradiation since it blocks any light from around 380 nm and downwards. Next to this, the PCBs are placed as a "stack",

as seen in figure 6.4 with aluminium reflectors on the inside, such that already most of the light stays inside the stack. Notice a see-through material was selected instead of total-light-blocking materials. This is because debugging and building the device is easier this way.

V2 also adds more realistic placement and mounting of hardware. The UV-C emitting LEDs are mounted on a circular PCB added to the previously mentioned stack with cooling on the top and bottom. The quartz is placed in the middle of an aluminium reflector, such that most of the UV-C light is reflected back to the inside of the stack [47]. The reflectance of aluminum is given in figure 6.6.

The aforementioned stack is placed on a sliding mechanism by tight springs allowing for a little movement/vibration from the vibration motor which is hung underneath the stack. A closer look at this sliding mechanism, also called the tray, is given in figure 6.7 and 6.8.

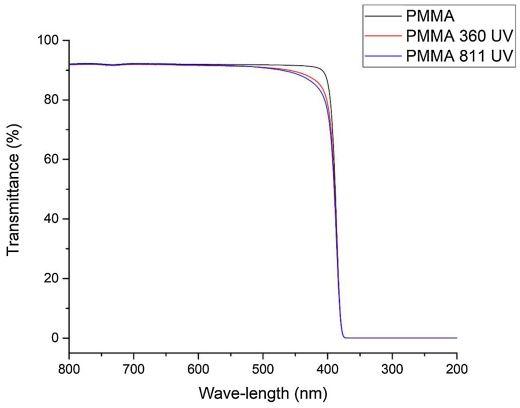


Figure 6.5: transmittance of PMMA (Plexiglas) for different wavelengths of light [46]

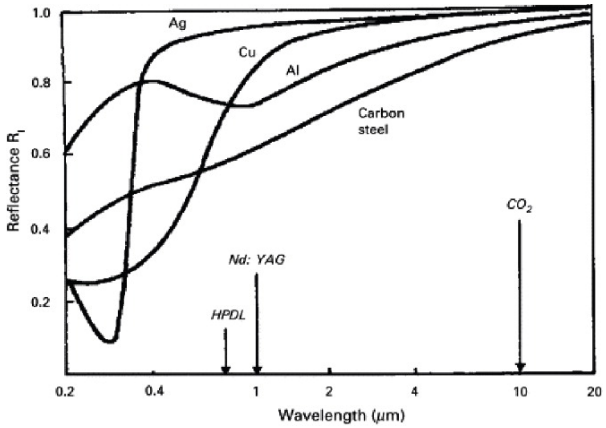


Figure 6.6: Reflectance of aluminum against different wavelengths [47]

Next to this, a screen and buttons/dials are added to the design to interact with the machine. The Control unit subgroup will go into more detail about this [19]. Some other conveniences have also been added such as a sliding door, rubber strips to seal the Plexiglass housing, and screws to screw everything in place.

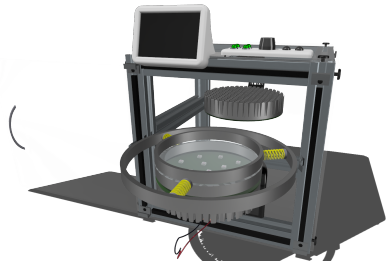


Figure 6.7: V2 tray

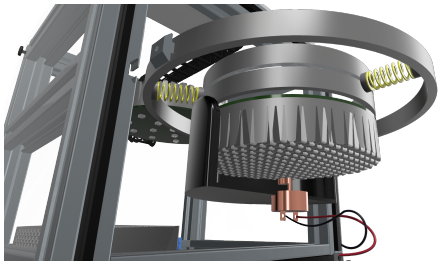


Figure 6.8: Bottom of V2

6.2.4. V3, final design

V3 offers the final design where all the problems of V2 are tackled. The main concerns with V2 were:

- Realistic PCB placement
- Actual PCB design
- Plexiglas design
- Complex spring usage
- Complex sliding tray mechanism
- Sliding door does not seal well enough
- PCBs for the LEDs are bigger; a stack of PCBs was created

- 3D printed parts need to be added

At the time V3 was created, the PCBs were already in production, therefore all designs were finished and added to the third iteration of the device's design. All PCBs have mounting holes making placement easier. These mounting holes were also added to the Plexiglas casing for easy assembly. The files for the plexiglass casing were sent to the 3ME PMMA laser cutter after which two main problems arose. The laser cutter was set up wrong and made sharp and unfinished edges and the holes for mounting were too tight of a fit. This worked in the 3D design, but in the real world, more tolerance was necessary. So oval holes were added to the Plexiglass to be able to adjust the plates. The V2020 frame was filled with rubber sealant to make sure the edges of the plates also block any outgoing UV-C light. The plates could then be screwed onto the V2020 framing with special V2020 nuts and bolts. the front plate was put together with a hinge to form a door that opens downward this part was removed from the render for better visibility. The final design for V3 is shown in figure 6.9

One other major redesign was the mounting mechanism for the stack. Using springs was too complex for the scope and time limit of this project, therefore elastic cord was used to create a weave across the stack. This makes mounting easier. The elastic cord can also be loosened or tightened according to the final weight. The new stack mounting mechanism is shown in figures 6.10 and 6.12.

The upper part of the stack which houses the top PCBs is kept in place with a Plexiglas plate so when the tray is opened the top part stays in place and does not have to be unscrewed to access the seed.

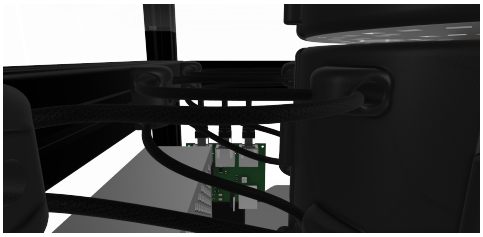


Figure 6.10: V3 stack mounting mechanism - the weave

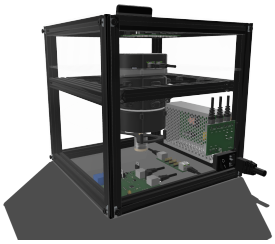


Figure 6.11: rear view of V3

The final V3 design of the device includes some materials that needed to be 3D printed such as:

- Cable Clips
- PCB Stack housing
- LED screen housing
- PCB mounts
- Motor mounting

The PCB stack housing, cable clips, and Motor mount create the final stack of V3. The cable clips are designed such that these fit onto the V2020 frame. The final fitment of these parts is displayed in figures 6.12 and 6.13. This tray is kept in place with bolts that are fastened to the main frame, so the tray can slide in easily.

Figure 6.9: final iteration of the UV-C disinfecting device - V3

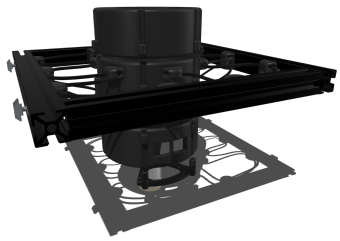


Figure 6.12: Stack & mounting - V3

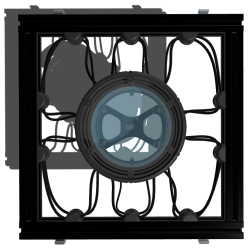


Figure 6.13: Stack & mounting from above - V3

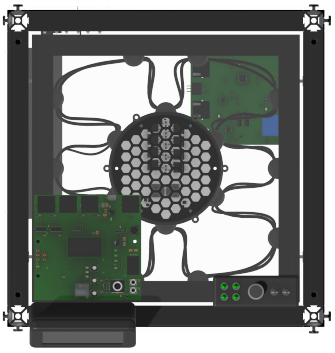


Figure 6.14: Final device from above - V3

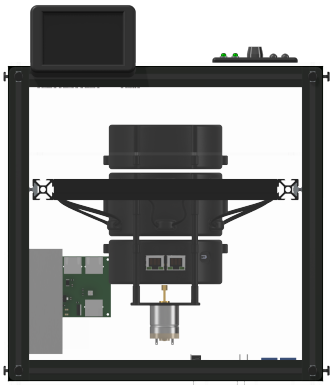


Figure 6.15: Final device from the front - V3

The schematic given in figure 6.16 gives a representation of how each module works and how the modules will work together and communicate.

Notice an over-current signal is missing from the motor controller and this figure does not display power connections.

The motor control unit's expendability could be useful in the final testing phase if overlooked items are needed or the setup should be modified for future experiments.

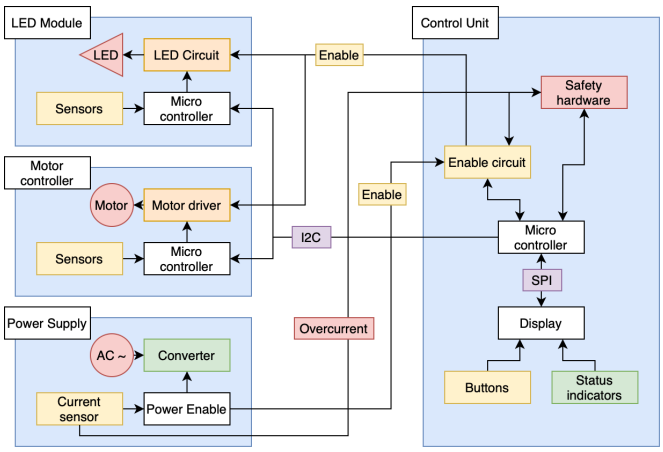


Figure 6.16: Schematic representation of the final workings of the device [18]

7

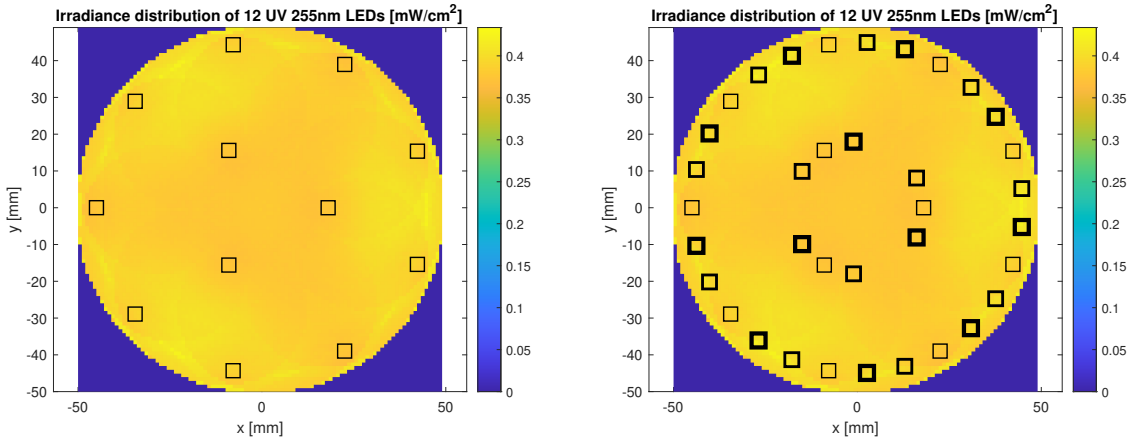
Conclusions, recommendations, and future work

7.1. Conclusions

At the time of writing, the seed disinfection device has been made and all components are fabricated and soldered together. After some time in debugging minor inconveniences, all components work together as one machine that disinfects seeds with the desired setting. A testing plan was made which was followed to determine scientifically which wavelengths, intensities and durations are the best for seed disinfection with the machine. This testing plan and its results are described in a test results report which is included in appendix G. In order to test whether the seed disinfection was successful the seeds were sent to Rijk Zwaan for PCR testing and planting. Unfortunately, the results from the PCR tests were inconclusive for these test runs and the results from planting are outside of the scope of this thesis, since these results take much more time.

7.1.1. Simulated LED Irradiance

After running the optimization script that calculates the best uniform irradiance on the UV-C exposed plate, radii of 17mm for the inner ring and 43mm for the outer ring seemed to give the best results. Applying these parameter values to the script for plotting the irradiance, the following results were acquired. In figure 7.1a the irradiance is shown for one LED type. Figure 7.1b is the same, but now with the 24 positions of the other 2 LED types added to the plot to prove that it all fits.



(a) Top view

(b) Side view

Figure 7.1: Final LED Irradiance simulation with radii 17mm and 43mm

7.1.2. Power Supply Unit

After manufacturing the PCB with the discussed schematics, some problems arose that were solved during the assembly stage:

- Footprint PS2 (24V-5V DC-DC converter) was wrong
- Mounting holes were different from the mounting holes of other PCBs
- Missing pull-down resistor at MOSFET, to be put in the next version of the design
- Drill hole file of a previous design was sent to the manufacturer
  - As a result, the 5V plate area was connected with the ground plate, causing dysfunction of components which use 5V for operation
  - By drilling out manually the hole, the 5V area was disconnected from the ground, resulting in proper functioning components again
- Reversed polarity design at the non-inverting amplifier circuit
- Wrong side of implementing the gain resistors of the non-inverting amplifier circuit
- One gain resistor used in the same circuit was not needed after all
- The resistors behind the power and status LEDs are too high resulting in a bit of dimmed light. In a future work the resistor values should be around 640Ω for full possible power on the LEDs [48]

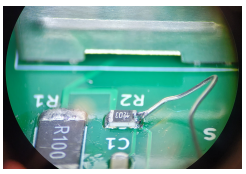


Figure 7.2: New connection between the RC filter and the op-amp

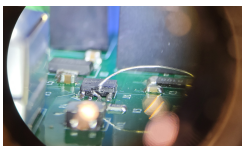


Figure 7.3: What the op-amp looks like now

After applying the needed changes to the PCB, the PCB was tested by connecting it to the AC/DC power supply and an Ethernet tester. The relay was tested by putting 5V manually on the PWR_ENABLE flag. The status LED and relay worked as designed, since a voltage of 24.3V could be measured at the Ethernet ports. Then, the over-current protection was tested using 50W power resistors to mimic the load of the system during normal operation and during overload. After some tweaking, the voltage measured at the OC_PC_FLAG matched the expectation.
Hereafter, the PCB was put into the prototype together with a power switch for user safety. The entire power supply result of this is shown in figure 7.4 hereunder.

Figure 7.4: Final power supply setup

7.1.3. Motor Control Unit

In the first version of the PCB, there were a couple of minor problems, which were resolved easily. The following problems arose:

- Boot-loader was reversed
- Footprint of 12V converter did not match
- Footprint LM358 pins
- Over-current resistors were too high

- Relay pull-down resistor
- Reverse polarity protection MOSFET's footprint did not match

The bootloader's footprint was in reverse, therefore the connector did not match. Thankfully this was easily resolved by using separate wires for the connector. The boot-loader facilitates software that the chip uses for serial communication. This was not too much of a hassle since this only needs to be done once because, from that point on, the USB interface can be used to program the micro-controller.

Also, the footprints in kiCad for the LM358 and 12V converter did not match the actual components that were ordered. Because of this a couple of pins had to be rewired outside of the PCB.

As mentioned before, the red square in figure C.5 notes a fault because this package in kiCad, the PCB design software, did not match. Since this part was only used by the reverse polarity protection, it was decided not necessary and was resolved by shorting pins 2 and 3.

Next up the over-current protection did not function properly at first, which was resolved by taking lower resistances as in figure 5.4, which were previously 11X higher. After this, the over-current protection worked as expected, giving a high signal to the control unit when the motor was stalled .

The final PCB is displayed in figure 7.5. It underwent thorough testing and successfully demonstrated its intended functions, validating its design.

The motor with decentralized weight was also vibrating powerful enough for its function.

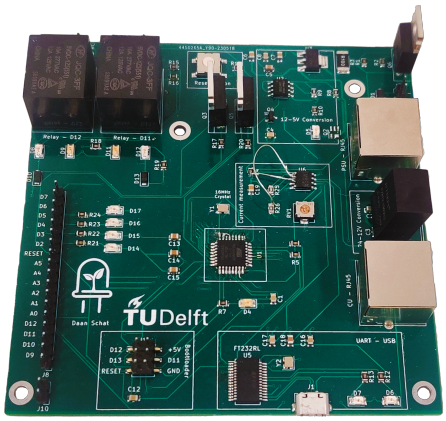


Figure 7.5: Final Motor Controller PCB

7.1.4. Design

At the time of writing, not all of the components have been placed since some of the PCBs were just done with testing outside of the main casing. However, the weave, framing, casing, and power supply were implemented as seen in figure 7.6. Creating the machine as designed in the 3D modelling software.

All items were correctly designed and accurately fabricated, making for an almost seamless fit. In the recommendations, section 7.2.3, more information can be found about the assembly.

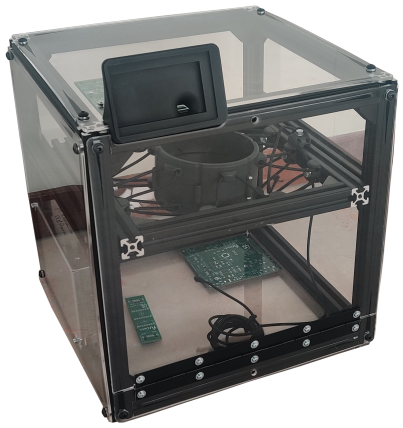


Figure 7.6: final iteration of the UV-C disinfecting device - V3

7.2. Recommendations

7.2.1. Simulations

Though the code is already improved to be more applicable for this project, there are still complications with the processing speed. Especially for the optimization script, it is very time-consuming. For a run where only the radii of rings is variable, it performs around 700 iterations. Since Matlab uses merely one processing core, this process is enduring. An improvement could be made by converting the code into Python, so that more or even all cores on the CPU would be used at the same time, resulting in a faster optimization result.
Furthermore, the optimization simulations were run with setting as few as possible parameters as valuable and as small as possible value range to minimise the number of iterations. These decisions were made based on manually made simulations. If the iteration process of the script would be faster, the script could have been run properly by setting a large value range for variables and set more parameters as variable.

Last thing to mention is the optimized LED layout. The layout that was designed for the prototype, radii of 15mm and 40mm were used. However, later, when the design had already been made and the optimization script was written, radii of 17mm and 43mm seemed to have the best uniform irradiation. The difference is small, but in a future work, it should be kept in mind that the optimized radius values should be used.

7.2.2. Power Supply

First notable thing that can be improved about the whole PSU PCB design is the used space on it. During the design process, the mounting holes were designed after all the components already had been placed and copper routes had been drawn. This resulted in creating the PCB larger than needed, since, as can be seen in figure D.1, there's a lot of unused space. Reducing the size of the PCB could be beneficial for a future new design of the prototype.

Furthermore, the Ethernet ports could be better placed on the PCB. During the PCB design, the prototype design was not finished yet, so the ports were placed what was best for the circuit to reduce vias in the PCB [49]. Now that the placing of other modules is known, all the Ethernet ports could have probably better been placed next to each other.

To stay on the topic of Ethernet connections, another improvement in a future design would be to use the full potential of these cables. The final power consumption by the system was known only when the system was built, since then the current measurements could be done. This results in that only then it was known what power rating the cables also should have. For the Control Unit and Motor Controller, which have a low power consumption as described in table 4.2.4, Ethernet cables are more

than fine to use. However, for the LDU modules, which use over 50W each, the story becomes different. It is difficult to trace back what the exact power rating is for these cables, however, for type 1 Power-over-Ethernet cables, which have a similar function, this is 15.40W. Fortunately, the intensive power consumption over these cables is for s hort period of time, namely for when the LEDs are illuminating seeds. A solution to prevent damages to the cable nonetheless, would be to use the other six of the eight wires within an Ethernet cable too. In practice this would look like four wires being connected to the V_{cc} and the other four wires to the over-current protection which leads to the ground. Together, the cable would in this way have a capability of $4*15.40W = 61.60W$, which is enough for the LDU modules.

Last thing, for safety reasons it could be considered to place a fuse in the switch. In the case that the overcurrent protection suddenly doesn't work anymore, or the Control Unit can't turn off the relay, the fuse would be a good back-up shutdown system.

7.2.3. Motor Controller

Though the motor control module functioned properly, there is still room for improvement. First of which, it is recommended to use pull-down resistors for the onboard relays because the pins on the ATMEGA are floating (meaning no consistent voltage) when it is not yet programmed. Because of this, the relays turn on and off inconsistently if not programmed yet. A pull-down resistor also makes the relay normally open, meaning not connected, which is desirable in case of an error. This is not much of a problem, however, it would be better to include this in the next iteration. Another resistor-related note is the LED current limiting resistors. These were picked too high causing the LEDs to appear a bit dim. This can easily be resolved by picking a lesser-value resistor.

Because of the order size, IRLZ44N MOSFETs were selected to operate the relay and motor control part. These, however, have higher ratings for their current handling capabilities than needed, making them too big for the design. A better fit would be the bss138 N-Channel MOSFET, which can handle less current, which is not important for this use-case, but is way smaller. For the motor control circuit, a bit more current is needed, namely 1A. After this threshold, the over-current protection is activated.
Next to these technical improvements, there are additional improvements for the PCB design. One such design improvement could be reducing the overall size of the PCB to optimize space utilization within the device. By shrinking the dimensions, the PCB could fit more efficiently into compact devices or constrained environments. Though, the intention of the module is modularity for the proof of concept. In an actual commercial design, the module does not need expendability and access to components etc., it would be tailored to fit and therefore also need fewer components previously needed for modularity.

Another desirable addition could be the inclusion of a screw terminal specifically designed for motor connections. This screw terminal would provide a secure and convenient way to connect the motor to the PCB, ensuring reliable power transmission and ease of maintenance.

7.2.4. Design

The final design worked as expected and designed. Though, improvements are still possible. First of which is a better way to open the door, for instance by making use of a door catch.

Implementing the UV-C LED seed disinfection technology on an assembly line can revolutionize seed processing operations. By integrating machines into automated setups, handling is simplified and efficient. A conveyor belt system ensures smooth and scalable seed transfer, while sensors and automatic controls enable precise timing and dosage adjustments. This integration maximizes efficiency, improves consistency and reduces manual labour, benefiting large seed-handling companies. This would still need a full redesign of the technology and therefore need a full analysis. Though, many design choices can be reused.

References

[1] Texas Instruments. *KeyStone Architecture Universal Asynchronous Receiver/Transmitter (UART)*. SPRUGP1. Nov. 2010. URL: https://www.ti.com/lit/ug/sprugp1/sprugp1.pdf?ts=1686421391467&ref_url=https%253A%252F%252Fwww.google.com%252F.

[2] NXP. *I2C-bus specification and user manual*. UM10204. Rev. 7.0. Oct. 2021. URL: https://www.nxp.com/docs/en/user-guide/UM10204.pdf.

[3] G. Lenssen. *Welkom bij Rijk Zwaan*. Company Visit. 2023.

[4] HEPACART. *Far-UV VS. Near-UV*. URL: https://www.hepacart.com/blog/far-uv-vs.-near-uv.

[5] WHO. *Radiation: Ultraviolet (UV) radiation*. URL: https://www.who.int/news-room/questions-and-answers/item/radiation-ultraviolet-(uv).

[6] W. Kowalski. *Ultraviolet Germicidal Irradiation Handbook*. 1st ed. Berlin: Springer Berlin, Heidelberg, 2014. DOI: https://doi.org/10.1007/978-3-642-01999-9.

[7] Philips. *UV-C desinfectie*. URL: https://www.lighting.philips.nl/producten/uv-c.

[8] A. Haji Malayeri et al. "Fluence (UV Dose) Required to Achieve Incremental Log Inactivation of Bacteria, Protozoa, Viruses and Algae". In: ().

[9] Nadia Storm et al. "Rapid and complete inactivation of SARS-CoV-2 by ultraviolet-C irradiation". In: *Scientific Reports* 10.1 (Dec. 2020). ISSN: 2045-2322. DOI: 10.1038/s41598-020-79600-8. URL: https://doi.org/10.1038/s41598-020-79600-8.

[10] BioClimatic. *Werking UVC-desinfectie*. URL: https://www.bioclimatic.nl/werking-uvc-desinfectie.html.

[11] Vijay Kumar Sharma and Hilmi Volkan Demir. "Bright Future of Deep-Ultraviolet Photonics: Emerging UVC Chip-Scale Light-Source Technology Platforms, Benchmarking, Challenges, and Outlook for UV Disinfection". In: *ACS Photonics* 9.5 (May 2022), pp. 1513–1521. DOI: 10.1021/acsphotonics.2c00041. URL: https://doi.org/10.1021/acsphotonics.2c00041.

[12] Soo-Ji Kim, Do-Kyun Kim, and Dong-Hyun Kang. "Using UVC Light-Emitting Diodes at Wavelengths of 266 to 279 Nanometers To Inactivate Foodborne Pathogens and Pasteurize Sliced Cheese". en. In: *Appl Environ Microbiol* 82.1 (Sept. 2015), pp. 11–17.

[13] Nan Jiang et al. "Effects of ultraviolet-c treatment on growth and mycotoxin production by Alternaria strains isolated from tomato fruits". In: *International Journal of Food Microbiology* 311 (2019), p. 108333. ISSN: 0168-1605. DOI: https://doi.org/10.1016/j.ijfoodmicro.2019.108333. URL: https://www.sciencedirect.com/science/article/pii/S0168160519302636.

[14] Dongqi Guo, Lixia Zhu, and Xujie Hou. "Combination of UV-C Treatment and Metschnikowia pulcherrimas for Controlling Alternaria Rot in Postharvest Winter Jujube Fruit". In: *Journal of Food Science* 80.1 (2015), pp. M137–M141. DOI: https://doi.org/10.1111/1750-3841.12724. eprint: https://ift.onlinelibrary.wiley.com/doi/pdf/10.1111/1750-3841.12724. URL: https://ift.onlinelibrary.wiley.com/doi/abs/10.1111/1750-3841.12724.

[15] "Effects of Ultraviolet Light on the Eye: Role of Protective Glasses". In: *Environmental Health Perspectives* 96 (Dec. 1991), pp. 177–184. DOI: https://doi.org/10.1289/ehp.9196177. URL: https://ehp.niehs.nih.gov/doi/epdf/10.1289/ehp.9196177.

[16] Hans-Joachim Burkardt. In: 38.2 (2000), pp. 87–91. DOI: doi:10.1515/CCLM.2000.014. URL: https://doi.org/10.1515/CCLM.2000.014.

[17] I.E. Lager et al. *Bachelor Graduation Project*. Bachelor Graduation Project Electrical Engineering. 2023.

[18] R. Imbens and L. Klootwijk. *UVC LED driving and sensing unit. UVC Sterilisation*. Thesis. 2023.

[19] Erman Ergül and Erik van Weelderen. *Software and Control. BAP Team UVO: UV-C Seed Disinfection*. Thesis. 2023, p. 50.

[20] Global Light Association. “UV-C SAFETY GUIDELINES”. In: *Global Lighting Association* (2020). URL: https://www.globallightingassociation.org/images/files/publications/GLA_UV-C_Safety_Position_Statement.pdf.

[21] Neelofer Hamid and Faiza Jawaid. “Influence of Seed pre-treatment by UV-A and UV-C radiation on germination and growth of Mung beans”. In: *Pakistan Journal of Chemistry* 1 (2011), pp. 164–167.

[22] Roy Bakker and Marcel Brouwers. *Smart Personal Protective Equipment. UVGI Group*. Thesis. 2020, p. 93.

[23] *Datasheet Deep UV LED - 255nm*. UV CA3535 series (CUD5GF1B). Seoul Viosys. 2020. URL: https://media.digikey.com/pdf/Data%5C%20Sheets/Seoul%5C%20Semiconductor/CUD5GF1B_R0.1_Prelliminary_2-3-20.pdf.

[24] *Datasheet Deep UV LED - 275nm*. UV CA3535 series (CUD7GF1B). Seoul Viosys. 2017. URL: http://www.s-et.com/upload2/CUD7GF1B_R2.0.pdf.

[25] Parr A. C. “The Candela and Photometric and Radiometric Measurements. Journal of research of the National Institute of Standards and Technology”. In: *Journal of research of the National Institute of Standards and Technology* 106(1) (2001), pp. 151–186. DOI: 10.6028/jres.106.007.

[26] A. Estrada-Hernández et al. “Luminous flux and correlated color temperature determination for LED sources”. In: *Sixth Symposium Optics in Industry*. Ed. by Julio C. Gutiérrez-Vega, Josué Dávila-Rodríguez, and Carlos López-Mariscal. Vol. 6422. International Society for Optics and Photonics. SPIE, 2007, 64220O. DOI: 10.1117/12.742589. URL: https://doi.org/10.1117/12.742589.

[27] Ivan Moreno et al. “Simple function for intensity distribution from LEDs”. In: *Nonimaging Optics and Efficient Illumination Systems IV*. Ed. by Roland Winston and R. John Koshel. Vol. 6670. International Society for Optics and Photonics. SPIE, 2007, 66700H. DOI: 10.1117/12.735820. URL: https://doi.org/10.1117/12.735820.

[28] M. Quazi et al. “Laser-based Surface Modifications of Aluminum and its Alloys”. In: *Critical Reviews in Solid State and Materials Sciences* 41 (Oct. 2015), pp. 1–26. DOI: 10.1080/10408436.2015.1076716.

[29] T Robert Harris et al. “Aging and performance comparison of common UVC reflective media in UV Air disinfection”. In: (). URL: https://www.porex.com/wp-content/uploads/2023/02/Aging-performance-comparison-of-common-UVC-reflective-IUVA-UV-Materials-Comparison-Porex-GTRI-Final.pdf.

[30] *Reflectance in Thin Films. UVGI Group*. Technical paper, p. 19. URL: https://materion.com/-/media/files/advanced-materials-group/me/technicalpapers/reflectance-in-thin-films_all.pdf.

[31] Feng Zhou and Wei Xiong. “Using PWM output as a digital-to-analog converter on dsp”. In: *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*. Vol. 2. IEEE. 2010, pp. 278–281.

[32] Robert Keim. “Low-Pass Filter a PWM Signal into an Analog Voltage”. In: (2016). URL: https://www.allaboutcircuits.com/technical-articles/low-pass-filter-a-pwm-signal-into-an-analog-voltage/.

[33] Jan FJ Wouters et al. “A novel active feedback flyback with only 100-mV inductive overshoot for a standard low-voltage CMOS inductive load driver, in a single-chip controller for 73 relays in a POTS/ADSL splitter application”. In: *IEEE journal of solid-state circuits* 40.7 (2005), pp. 1541–1548.

[34] CN Dunn et al. “A Logic Controlled HVIC Combining DMOS and GTO Switches, an NMOS Relay Driver and a CMOS Comparator for Telecommunication Applications”. In: *ECS Proceedings Volumes* 1989.1 (1989), p. 292.

[35] *Datasheet DC/DC Converter K78xx-1000R3*. K7805-1000R3. MORNSUN Guangzhou Science and Technology Co. 2022. URL: https://4donline.ihs.com/images/VipMasterIC/IC/MNSN/MNSN-S-A0015864214/MNSN-S-A0015864214-1.pdf?hkey=EF798316E3902B6ED9A73243A3159BB0.

[36] *Datasheet Linear Regulator AMS1117-3.3*. AMS1117-5.0. UMW(Youtai Semiconductor Co., Ltd.) URL: https://datasheet.lcsc.com/lcsc/2110111730_UMW-Youtai-Semiconductor-Co---Ltd--AMS1117-5-0_C347223.pdf.

[37] C.K. Alexander and M.N.O. Sadiku. *Fundamentals of Electric Circuits*. COLLEGE IE OVERRUNS. McGraw-hill Education, 2017. ISBN: 9781259251320. URL: https://books.google.nl/books?id=e4-gjwEACAAJ.

[38] *Datasheet Voltage Regulator K78XX*$_1 000R3$. K7812$_1 000R3$. jetekps. URL: https://datasheet.lcsc.com/lcsc/2303300930_JETEKPS-K7812-1000R3_C5378042.pdf.

[39] *Datasheet Atmel 8-bit Microcontroller with 4/8/16/32KBytes In- System Programmable Flash*. ATmega328P. Atmel. URL: https://datasheet.lcsc.com/lcsc/1809101618_Microchip-Tech-ATMEGA328P-AU_C14877.pdf.

[40] *Datasheet FT232R USB UART I.C.* FT232RL-REEL. FTDI Chip - Future Technology Devices International Ltd. URL: https://datasheet.lcsc.com/lcsc/1810311811_FTDI-FT232RL-REEL%5C_C8690.pdf.

[41] *Datasheet Motor RS-360 365SA*. RS-360. HSMotors.

[42] ElectronicsTutorials. *Operational Amplifier Basics*. URL: https://www.electronics-tutorials.ws/opamp/opamp_1.html.

[43] Arduino. *analogRead()*. URL: https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/.

[44] Led professional. *Transparent Material Considerations for UV Optics in Horticultural Lighting Applications*. URL: https://www.led-professional.com/resources-1/articles/transparent-material-considerations-for-uv-optics-in-horticultural-lighting-applications.

[45] Nantong Foric Optical Glass. *Round quartz*. URL: https://nl.aliexpress.com/item/1005003431984204.html?spm=a2g0o.order_list.order_list_main.29.20c979d2kaePC1&gatewayAdapt=glo2nld#nav-specification.

[46] Karollyne Gomes de Castro Monsores et al. “Influence of ultraviolet radiation on polymethylmethacrylate (PMMA)”. In: *Journal of Materials Research and Technology* 8.5 (2019), pp. 3713–3718. ISSN: 2238-7854. DOI: https://doi.org/10.1016/j.jmrt.2019.06.023. URL: https://www.sciencedirect.com/science/article/pii/S2238785419305861.

[47] M. Quazi et al. “Laser-based Surface Modifications of Aluminum and its Alloys”. In: *Critical Reviews in Solid State and Materials Sciences* 41 (Oct. 2015), pp. 1–26. DOI: 10.1080/10408436.2015.1076716.

[48] *Datasheet 1206 Package Chip LED with Inner Lens*. DSE-0001603. Everlight Electronics Co., LTD. 2009. URL: https://datasheet.lcsc.com/lcsc/1809191118_Everlight-Elec-11-21-R6C-AR2S2B-2T_C264317.pdf.

[49] *Complete Guide to Types of PCB Vias: Designs and Routing*. https://resources.altium.com/p/pcb-via. Accessed: 2023-06-15.

A

Programming code

A.1. LED Layout

A.1.1. Square plate irradiance

```
% ------------------------------------------------------------------------
% NAME: get_irradiance_grid_sum_square
% INPUT:
% > n_rings: number of rings encircling the center LED
% > leds_per_ring: vector of size n_rings indicating the
% number of UV LEDs in each ring (in to out)
% > center_distance: vector of size n_rings indicating the
% radius of each ring (in to out) [mm]
% > tilt_angle: vector of size n_rings indicating the
% zenith angle of each UV LED in each ring (in to out) [°]
% > led_distance_to_grid: distance from LEDs to the grid [mm]
% > led_size: size of the UV LED (width and height) [mm]
% > radiation_pattern: normalized intensity distribution table
% corresponding to degrees [0°-90°]
% > led_radiant_flux: radiant flux of the UV LED [mW]
% > perimeter_radius: radius of the perimeter lining out the
% surface on which the UV LEDs provide radiation [mm]
% > grid_dim: size of the grid
% > grid_length: width and height of the grid [mm]
% > is_reflected: logical value (true/false) indicating if
% reflections from the walls surrounding the irradiated surface
% should be accounted for
% > reflection_coefficient:
% coefficient indicating the fraction of
% reflected UV light intensity
% > reflection_resolution:
% resolution with which the position of
% reflection is computed [°]
% > create_plot: logical value (true/false) indicating if a
% plot is to be created of the irradiance on the grid [mW/cm^2]
% > plotallwavelengths: logical value (true/false) indicating if a
% plot of the other 2 wavelengths is also  to be created
% OUTPUT:
% > irradiance_grid: 2-dimensional vector indicating the
% irradiance of each grid element [mW/cm^2]
% > led_position_array: array of size
% n_rings x max(leds_per_ring x 3 containing the positions of the
% UV LEDs in each ring (x, y, z) [mm]
% FUNCTION:
% This function computes the irradiance of all n_rings rings of UV
% LEDs consisting of n_leds(i). The LEDs in each ring are spaced with
% equal angle, and the LEDs of two adjacent rings are spaced as far
% apart. Each UV LED has the same defined radiation_pattern,
% radiant_flux, and distance from the grid. The irradiance is
% determined on a two dimensional grid for each grid element.
% ORIGINAL AUTHORS: Roy Bakker & Marcel Brouwers
```

```
% EDITOR: Maxim Mazurovs
% DATE: 8-05-2023
% -------------------------------------------------------------------------
function [irradiance_grid, led_position_array] = ...
get_irradiance_grid_sum_square( n_rows, ...
leds_per_row, ...
led_distance_to_grid, ...
led_size, ...
radiation_pattern, ...
led_radiant_flux, ...
perimeter_dim, ...
grid_dim, ...
grid_length, ...
is_reflected, ...
reflection_coefficient, ...
reflection_resolution, ...
create_plot, plotallwavelengths)

% declare and define variables
irradiance_grid = zeros(grid_dim, grid_dim);
led_position_array = zeros(n_rows, max(leds_per_row), 3);

row_height = zeros(n_rows,1);
%change 0 to 30 to stretch LEDs to the sides
div = (perimeter_dim+0)/(n_rows+1);
for n = 1:n_rows
    row_height(n) = (perimeter_dim+0)-((perimeter_dim+0)/2)-n*div;
end

% go through all rings of UV LEDs [1, n_rows]
for n = 1:n_rows
% compute the irradiance on the grid for the current row of UV LEDs
% [mW/cm^2]
[irradiance_grid_tmp, led_position] = ...
get_irradiance_grid_square( n, row_height(n), ...
leds_per_row(n), ...
led_distance_to_grid, ...
radiation_pattern, ...
led_radiant_flux, ...
perimeter_dim, ...
grid_dim, ...
grid_length, ...
is_reflected, ...
reflection_coefficient, ...
reflection_resolution);
% add the irradiance on the grid of the current ring to that of the
% previous rings [mm]
irradiance_grid = irradiance_grid + irradiance_grid_tmp;
% store the position of all UV LEDs in the current ring in the LED
% position array
% led_position_array(n+1, 1:leds_per_ring(n), :) = led_position;
led_position_array(n, 1:leds_per_row(n), :) = led_position;
end

if plotallwavelengths == true
    maxi = leds_per_row(end);
    tot = zeros(2,maxi);

    % Rotate LED structure for 2 different angles to have 3 diff. wavelengths
    % Reorder matrices for rotation
    for n = 1:n_rings
        old_x = led_position_array(n,:,1);
        old_y = led_position_array(n,:,2);
        tot(1,1+(n-1)*maxi:maxi+(n-1)*maxi) = old_x;
        tot(2,1+(n-1)*maxi:maxi+(n-1)*maxi) = old_y;
    end
end

% if create_plot is true, plot the irradiance on the grid of the whole UV
% LED system defined by the parameters of this function
if create_plot == true
```

```
figure();
% map the grid to the right dimensions, namely a square grid with
% x = [-grid_length/2, grid_length/2] and
% y = [-grid_length/2, grid_length/2] [mm]
mapped_coordinates = -grid_length/2: ...
grid_length/grid_dim: grid_length/2-grid_length/grid_dim;
% create a pseudocolor plot of the irradiance
s = pcolor(mapped_coordinates, mapped_coordinates, ...
irradiance_grid);
% turn off the grid outline
s.EdgeColor = 'none';
% turn on the colorbar legend
colorbar;
% create an appropriate title for the current UV LED system
title(sprintf(...
"Irradiance distribution of %d UV 255nm LEDs [mW/cm^2]", ...
sum(leds_per_row)));

% label the axes
xlabel("x [mm]");
ylabel("y [mm]");
axis equal;
hold on;

% using the LED position array, indicate the position of the UV LEDs by
% drawing a square of size led_size at every LED position [mm]
for ring = 1:n_rows
for led = 1:leds_per_row(ring)
rectangle('Position', ...
[(led_position_array(ring, led, 1)-led_size/2), ...
(led_position_array(ring, led, 2)-led_size/2), ...
led_size, led_size], ...
'LineWidth', 1);
end
end

end
end

% get_irradiance_grid_sum_square(4,[3,3,3,3],30,3.9,radiation_pattern_lg,3.5,100,100,100,
    false,0.5,1,true,false)
```

```
% ------------------------------------------------------------------------
% NAME: get_irradiance_grid_ring
% INPUT:
% > ring_radius: radius of the ring formed by UV LEDs [mm]
% > ring_angle: angle added to all UV LEDs in ring [rad]
% > ring_tilt_angle: zenith angle of each UV LED in the ring [°]
% > n_leds: number of UV LEDs in the ring
% > led_distance_to_grid: distance from LEDs to the grid [mm]
% > radiation_pattern: normalized intensity distribution table
% corresponding to degrees [0°-90°]
% > led_radiant_flux: radiant flux of the UV LED [mW]
% > perimeter_radius: radius of the perimeter lining out the
% surface on which the UV LEDs provide radiation [mm]
% > grid_dim: size of the grid
% > grid_length: width and height of the grid [mm]
% > is_reflected: logical value (true/false) indicating if
% reflections from the walls surrounding the irradiated surface
% should be accounted for
% > reflection_coefficient:
% coefficient indicating the fraction of
% reflected UV light intensity
% > reflection_resolution:
% resolution with which the position of
% reflection is computed [°]
% OUTPUT:
% > irradiance_grid: 2-dimensional vector indicating the
% irradiance of each grid element [mW/cm^2]
% > led_position_array: array of size n_leds x 3 containing the
```

```
% positions of the UV LEDs (x, y, z) [mm]
% FUNCTION:
% This function computes the irradiance of a ring of UV LEDs
% consisting of n_leds spaced with equal angle, each having the same
% defined radiation_pattern, radiant_flux, and distance from the
% grid. The irradiance is determined on a two dimensional grid
% for each grid element.
% AUTHOR: Roy Bakker & Marcel Brouwers
% DATE: 7-5-2020
% ------------------------------------------------------------------------
function [irradiance_grid, led_position_array] = ...
    get_irradiance_grid_square( rownumber, row_height, ...
    n_leds, ...
    led_distance_to_grid, ...
    radiation_pattern, ...
    led_radiant_flux, ...
    perimeter_dim, ...
    grid_dim, ...
    grid_length, ...
    is_reflected, ...
    reflection_coefficient, ...
    reflection_resolution)
% declare and define variables:
led_position_array = zeros(n_leds, 3);
led_position = [0, 0, led_distance_to_grid];
irradiance_grid = zeros(grid_dim, grid_dim);
% compute the solid angle of the UV LED weighted with its normalized
% intensity distribution (figure 1) [sr]
weighted_solid_angle = 0;
for theta = 0:90
    weighted_solid_angle = weighted_solid_angle + ...
        2*pi * radiation_pattern(theta+1, 2) * ...
        sin(theta/180*pi) * 1/180*pi;
end
% compute the normalized intensity distribution table I(phi, theta); the
% rows correspond to the azimuth angle (phi) with range [0°, 359°] and
% steps of 1°; the columns correspond to the zenith angle (theta) with
% range [0°, 180°] and steps of 1°; the range of the normalized intensity
% values is [0, 1]
tilted_radiation_pattern = ...
    get_tilted_radiation_pattern( radiation_pattern, 0);
% compute the absolute radiant intensity [mW/sr]
radiant_intensity = led_radiant_flux .* ...
    tilted_radiation_pattern ./ weighted_solid_angle;

row_distr = zeros(n_leds,1);
div = perimeter_dim/(n_leds*3+1);

%code for offset after each row
for n = 1:n_leds
    if rownumber > 3
        rownumber = rownumber - 3;
    end
    if rownumber == 1
        shift = 0;
    end
    if rownumber == 2
        shift = div;
    end
    if rownumber == 3
        shift = 2*div;
    end

    %uncomment the following line to plot LEDs without
    %row offset (straight line placed LEDs)
    shift=div;

%     row_distr(n) = perimeter_dim/2-perimeter_dim+div+3*(n-1)*div+shift;

    % uncomment the following line to stretch LEDs to the sides
    row_distr(n) = perimeter_dim/2-perimeter_dim+div+3*(n-1)*div+shift+(n-2.5)*shift;
```

```
100 end
101
102 % compute the irradiance on the grid for all UV LEDs combined [mW/cm^2]
103 for n = 1:n_leds
104     % compute the x and y coordinates of the current UV LED [mm]
105     led_position(1) = row_distr(n);
106     led_position(2) = row_height;
107     % compute the azimuth angle of the light direction of the UV LED [rad]
108     led_direction = 0;
109     % store the position of the current UV LED [mm]
110     led_position_array(n, :) = led_position;
111     % compute the irradiance on the grid for the current UV LED and add
112     % this to the irradiance on the grid for the previous UV LEDs
113     % [mW/cm^2]
114     irradiance_grid = irradiance_grid + ...
115         get_irradiance_grid_single_square( radiant_intensity, ...
116         led_position, ...
117         led_direction, ...
118         perimeter_dim, ...
119         grid_dim, ...
120         grid_length, ...
121         false, ... %is_reflected
122         reflection_coefficient, ...
123         reflection_resolution);
124 end
125 end
```

A.1.2. Circular plate irradiance

```
% ------------------------------------------------------------------------
% NAME: get_irradiance_grid_sum_ring_extended
% INPUT:
% > n_rings: number of rings encircling the center LED
% > leds_per_ring: vector of size n_rings indicating the
% number of UV LEDs in each ring (in to out)
% > center_distance: vector of size n_rings indicating the
% radius of each ring (in to out) [mm]
% > led_distance_to_grid: distance from LEDs to the grid [mm]
% > led_size: size of the UV LED (width and height) [mm]
% > radiation_pattern: normalized intensity distribution table
% corresponding to degrees [0°-90°]
% > led_radiant_flux: radiant flux of the UV LED [mW]
% > perimeter_radius: radius of the perimeter lining out the
% surface on which the UV LEDs provide radiation [mm]
% > grid_dim: size of the grid
% > grid_length: width and height of the grid [mm]
% > is_reflected: logical value (true/false) indicating if
% reflections from the walls surrounding the irradiated surface
% should be accounted for
% > reflection_coefficient:
% coefficient indicating the fraction of
% reflected UV light intensity
% > reflection_resolution:
% resolution with which the position of
% reflection is computed [°]
% > create_plot: logical value (true/false) indicating if a
% plot is to be created of the irradiance on the grid [mW/cm^2]
% > plotallwavelengths: logical value (true/false) indicating if a
% plot of the other 2 wavelengths is also  to be created
% OUTPUT:
% > irradiance_grid: 2-dimensional vector indicating the
% irradiance of each grid element [mW/cm^2]
% > led_position_array: array of size
% n_rings x max(leds_per_ring x 3 containing the positions of the
% UV LEDs in each ring (x, y, z) [mm]
% FUNCTION:
% This function computes the irradiance of all n_rings rings of UV
% LEDs consisting of n_leds(i). The LEDs in each ring are spaced with
% equal angle, and the LEDs of two adjacent rings are spaced as far
% apart. Each UV LED has the same defined radiation_pattern,
% radiant_flux, and distance from the grid. The irradiance is
% determined on a two dimensional grid for each grid element.
% ORIGINAL AUTHORS: Roy Bakker & Marcel Brouwers
% EDITOR: Maxim Mazurovs
% DATE: 8-05-2023
% ------------------------------------------------------------------------
function [irradiance_grid, led_position_array] = ...
get_irradiance_grid_sum_ring_extended( n_rings, ...
leds_per_ring, ...
center_distance, ...
led_distance_to_grid, ...
led_size, ...
radiation_pattern, ...
led_radiant_flux, ...
perimeter_radius, ...
grid_dim, ...
grid_length, ...
is_reflected, ...
reflection_coefficient, ...
reflection_resolution, ...
create_plot, plotallwavelengths)

% declare and define variables
ring_angle_shift = 0;
irradiance_grid = zeros(grid_dim, grid_dim);
led_position_array = zeros(n_rings, max(leds_per_ring), 3);
tilt_angle = zeros(1, n_rings);

```

```
% go through all rings of UV LEDs [1, n_rings]
for n = 1:n_rings
% if the current ring is not the inner most ring...
if n > 1
% compute the optimal absolute angle of the current ring using the
% number of the number of LEDs of the current and previous ring
% [rad]
ring_angle_shift = ring_angle_shift + ...
optimal_angle_between_rings( leds_per_ring(n-1), ...
leds_per_ring(n), ...
pi/1000, ...
false);
end
% compute the irradiance on the grid for the current ring of UV LEDs
% [mW/cm^2]
[irradiance_grid_tmp, led_position] = ...
get_irradiance_grid_ring( center_distance(n), ...
ring_angle_shift, ...
tilt_angle(n), ...
leds_per_ring(n), ...
led_distance_to_grid, ...
radiation_pattern, ...
led_radiant_flux, ...
perimeter_radius, ...
grid_dim, ...
grid_length, ...
is_reflected, ...
reflection_coefficient, ...
reflection_resolution);
% add the irradiance on the grid of the current ring to that of the
% previous rings [mm]
% irradiance_grid = irradiance_grid + irradiance_grid_tmp;
irradiance_grid = irradiance_grid + irradiance_grid_tmp;
% store the position of all UV LEDs in the current ring in the LED
% position array
% led_position_array(n+1, 1:leds_per_ring(n), :) = led_position;
led_position_array(n, 1:leds_per_ring(n), :) = led_position;
end

maxi = leds_per_ring(end);
tot = zeros(2,maxi);
disp(led_position_array);
% Rotate LED structure for 2 different angles to have 3 diff. wavelengths
% Reorder matrices for rotation
for n = 1:n_rings
    old_x = led_position_array(n,:,1);
    old_y = led_position_array(n,:,2);
    tot(1,1+(n-1)*maxi:maxi+(n-1)*maxi) = old_x;
    tot(2,1+(n-1)*maxi:maxi+(n-1)*maxi) = old_y;
end

% Rotate using rotation matrix for the 2 other wavelengths
theta1=2*(40/3); %Clockwise by x degrees
theta2 = 7*(40/3);
R1=[cosd(theta1) -sind(theta1); sind(theta1) cosd(theta1)];
R2=[cosd(theta2) -sind(theta2); sind(theta2) cosd(theta2)];
newarray = R1*tot;
newarray2 = R2*tot;

x_rotated = newarray(1,:);
y_rotated = newarray(2,:);
x_rotated2 = newarray2(1,:);
y_rotated2 = newarray2(2,:);

%Transform the rotated matrices to their original form of 3 dimensions
rotated_matrix = zeros(n_rings,maxi,3);
rotated_matrix3 = zeros(n_rings,maxi,3);
for n = 1:n_rings
    rotated_matrix(n,:,1) = x_rotated(1+(n-1)*maxi:n*maxi);
    rotated_matrix(n,:,2) = y_rotated(1+(n-1)*maxi:n*maxi);
    rotated_matrix3(n,:,1) = x_rotated2(1+(n-1)*maxi:n*maxi);
```

```
    rotated_matrix3(n,:,2) = y_rotated2(1+(n-1)*maxi:n*maxi);
end
rotated_matrix(:,:,3) = led_position_array(:,:,3);
rotated_matrix3(:,:,3) = led_position_array(:,:,3);

% if create_plot is true, plot the irradiance on the grid of the whole UV
% LED system defined by the parameters of this function
if create_plot == true
figure();
% map the grid to the right dimensions, namely a square grid with
% x = [-grid_length/2, grid_length/2] and
% y = [-grid_length/2, grid_length/2] [mm]
mapped_coordinates = -grid_length/2: ...
grid_length/grid_dim: grid_length/2-grid_length/grid_dim;
% create a pseudocolor plot of the irradiance
s = pcolor(mapped_coordinates, mapped_coordinates, ...
irradiance_grid);
% turn off the grid outline
s.EdgeColor = 'none';
% turn on the colorbar legend
colorbar;
% create an appropriate title for the current UV LED system
title(sprintf(...
"Irradiance distribution of %d UV 255nm LEDs [mW/cm^2]", ...
sum(leds_per_ring)));

% label the axes
xlabel("x [mm]");
ylabel("y [mm]");
axis equal;
hold on;

% using the LED position array, indicate the position of the UV LEDs by
% drawing a square of size led_size at every LED position [mm]
for ring = 1:n_rings
for led = 1:leds_per_ring(ring)
rectangle('Position', ...
[(led_position_array(ring, led, 1)-led_size/2), ...
(led_position_array(ring, led, 2)-led_size/2), ...
led_size, led_size], ...
'LineWidth', 1);
if plotallwavelengths == true
    rectangle('Position', ...
    [(rotated_matrix(ring, led, 1)-led_size/2), ...
    (rotated_matrix(ring, led, 2)-led_size/2), ...
    led_size, led_size], ...
    'LineWidth', 2);
    rectangle('Position', ...
    [(rotated_matrix3(ring, led, 1)-led_size/2), ...
    (rotated_matrix3(ring, led, 2)-led_size/2), ...
    led_size, led_size], ...
    'LineWidth', 3);
end
end
end

end
end
```

A.1.3. LED layout optimization script

```
% ------------------------------------------------------------------------
% NAME: irradiance_optimization
% INPUT:
% > n_rings_boundaries: boundaries outlining the range of numbers
% to try for the number of UV LED rings (min, max)
% > leds_per_ring_boundaries:
% boundaries outlining the range of numbers
% to try for the number of UV LEDs in each ring (min, max)
% > distance_rings_min: minimum distance between: each UV LED ring;
% the inner UV LED ring and center UV LED; outer UV LED ring and
% perimeter of the surface to be irradiated [mm]
% > tilt_angle_boundaries:
% boundaries outlining the range of angles to
% try for the tilt angle of each UV LED ring (min, max) [°]
% > led_distance_to_grid_boundaries:
% boundaries outlining the range of distances
% to try for the distance between all UV LED rings and the
% surface to be irradiated (min, max) [mm]
% > overdose_factor_max: maximum value allowed for optimization
% variable 2 = I_max / I_min; in words, this factor describes the
% ratio of maximum measured irradiance on the surface over the
% minimum measured irradiance on the surface
% > radiation_pattern: normalized intensity distribution table
% corresponding to degrees [0°-90°]
% > perimeter_radius: radius of the perimeter lining out the
% surface on which the UV LEDs provide radiation [mm]
% > grid_dim: size of the grid
% > grid_length: width and height of the grid [mm]
% > is_reflected: logical value (true/false) indicating if
% reflections from the walls surrounding the irradiated surface
% should be accounted for
% > reflection_coefficient:
% coefficient indicating the fraction of
% reflected UV light intensity
% > reflection_resolution:
% resolution with which the position of
% reflection is computed [°]
% OUTPUT:
% > n_rings_opt: number of UV LED rings for optimal power
% usage
% > leds_per_ring_opt: vector of size n_rings indicating the
% number of UV LEDs in each ring (in to out) for optimal power
% usage
% > center_distance_opt: vector of size n_rings indicating the
% radius of each ring (in to out) for optimal power usage [mm]
% > tilt_angle_opt: vector of size n_rings indicating the
% zenith angle of each UV LED in each ring (in to out) for
% optimal power usage [°]
% > led_distance_to_grid_opt:
% distance from LEDs to the grid for optimal
% power usage [mm]
% FUNCTION:
% This function goes through all possible permutations in layout of
% the UV LEDs within the bounds for each variable indicated by the
% parameters of this function.
% The function optimizes for the so-called optimization_variable1:
% N/I_min. Here, N is the total number of UV LEDs used in the layout
% and I_min is the minimal irradiance found on the irradiated
% surface. Optimizing for this variable means finding the UV LED
% layout which gives the lowest possible value for this variable. The
% corresponding layout provides the least amount of power consumption
% for the highest irradiance on the surface.
% The second value which is used to optimize the UV LED layout is
% called optimization_variable2: I_max / I_min. A more descriptive
% name would be the overdose factor: the ratio of maximum measured
% irradiance on the surface over the minimum measured irradiance on
% the surface. The position on the grid with the lowest measured
% irradiance (I_min) has to be given a dose D, meaning that the
% position with the highest measured irradiance (I_max) is given an
```

```
% overdose of D*(I_max/I_min). From this description it should be
% obvious that this overdose factor should be kept as small as
% possible, therefore the optimization_variable2 is only used to
% check if it is below the overdose_factor_max parameter.
% AUTHOR: Roy Bakker & Marcel Brouwers
% DATE: 19-5-2020
% ------------------------------------------------------------------------
function [optimization_variable2, center_distance_opt] = ...
    irradiance_optimization_shorted( n_rings_boundaries, ...
    leds_per_ring, ...
    distance_rings_min, ...
    overdose_factor_max, ...
    radiation_pattern, ...
    perimeter_radius, ...
    grid_dim, ...
    grid_length, ...
    is_reflected, ...
    reflection_coefficient, ...
    reflection_resolution)

% declare and define variables
break_flag = 0;
center_distance_resolution = 1;
optimization_variable1 = inf;
optimization_variable2 = inf;
irradiance_min_opt = -1;
irradiance_max_opt = -1;
n_rings_opt = -1;
leds_per_ring_opt = -1;
center_distance_opt = -1;

% compute the maximum number of iterations needed to have tried every
% possible permutation of distance between UV LED rings
center_distance_iterations = 0;
max_center_distance_iterations = 0;
for n = n_rings_boundaries(1):n_rings_boundaries(2)
    if perimeter_radius < (n+1) * distance_rings_min
        continue
    else
        % the formula for calculating the number of iterations needed to
        % have tried every permutation of distance between UV LED rings
        % for n number of UV LED rings
        max_center_distance_iterations = ...
            max_center_distance_iterations + ...
            nchoosek((perimeter_radius - ...
            ((n+1)*distance_rings_min) + 1) + n - 1, n);
    end
end
% go through every possible value of number of UV LED rings bounded by the
% n_rings_boundaries parameter
for n_rings = n_rings_boundaries(1):n_rings_boundaries(2)
    % declare and initialize the vector of size n_rings indicating the
    % radius of each ring (in to out) [mm]
    center_distance = zeros(n_rings, 1);
    % go through every possible value of the sum of radii of each UV LED
    % ring
    for center_distance_sum = ...
            0: center_distance_resolution: perimeter_radius^n_rings
        % compute the vector of size n_rings indicating the radius of each
        % ring (in to out) using the current value of the sum of radii of
        % each UV LED ring [mm]
        for n = 1:n_rings
            % compute the radius of the n'th UV LED ring (in to out) [mm]
            center_distance(n) = rem(floor(...
                (center_distance_sum-center_distance_resolution) / ...
                perimeter_radius^(n-1)), perimeter_radius) + 1;
            % if the computed radius is larger than the outer boundary of
            % the surface to be irradiated minus some margin, skip the
            % whole irradiance calculation as the current UV LED layout
            % would not be valid
            if center_distance(n) > perimeter_radius - distance_rings_min
```

```
141                 break_flag = 1;
142                 break;
143             end
144             % if the most inner UV LED ring has a radius smaller than the
145             % minimum radius, skip the whole irradiance calculation as
146             % the current UV LED layout would not be valid
147             if n == 1
148                 if center_distance(1) < 10
149 %                 if center_distance(1) < distance_rings_min
150                     break_flag = 1;
151                     break;
152                 end
153                 % if the computed radius of the current observed UV LED ring is
154                 % smaller than the that of the previous ring plus some
155                 % margin, skip the whole irradiance calculation as the
156                 % current UV LED layout would not be valid
157             else
158                 if center_distance(n) < ...
159                         center_distance(n-1) + distance_rings_min
160                     break_flag = 1;
161                     break;
162                 end
163             end
164         end
165         % if the current iteration needs to be skipped, break from the
166         % current iteration and clear the flag indicating such action
167         if break_flag == 1
168             break_flag = 0;
169             continue;
170         end
171         % increment the iteration counter for the center distance loop
172         center_distance_iterations = center_distance_iterations + 1;
173
174         tilt_angle = zeros(n_rings,1);
175
176         % compute the irradiance on the surface with the
177         % current LED layout permutation
178         [irradiance_grid, ~] = ...
179             get_irradiance_grid_sum_ring_extended(...
180             n_rings, ...
181             leds_per_ring, ...
182             center_distance, ...
183             tilt_angle, ...
184             30, ... % led_distance_to_grid
185             3.9, ... % led size
186             radiation_pattern, ...
187             3.5, ... % led radiant flux
188             perimeter_radius, ...
189             grid_dim, ...
190             grid_length, ...
191             is_reflected, ...
192             reflection_coefficient, ...
193             reflection_resolution, ...
194             false, false);
195         % compute the minimum and maximum weighted irradiance
196         % values on the surface [1/cm^2]
197         [irradiance_min, irradiance_max, ~, ~] = ...
198             compute_irradiance_uniformity(...
199             irradiance_grid, ...
200             perimeter_radius, ...
201             grid_dim, ...
202             grid_length);
203         % check if the overdose value, optimization_variable2,
204         % is lower than the maximum allowed overdose value
205         if irradiance_max / irradiance_min <= ...
206                 overdose_factor_max && ...
207                 irradiance_max / irradiance_min <= ...
208                 optimization_variable2
209             % compute the new "best" optimization_variable1
210             % and optimization_variable2 values
211             optimization_variable1 = ...
```

```
212                 (sum(leds_per_ring)+1) / irradiance_min;
213             optimization_variable2 = ...
214                 irradiance_max / irradiance_min;
215             % store the minimum and maximum irradiance values
216             % for the current "best" UV LED layout
217             irradiance_min_opt = irradiance_min;
218             irradiance_max_opt = irradiance_max;
219             % store the UV LED layout parameters which provide
220             % the current "best" UV LED layout
221             n_rings_opt = n_rings;
222             leds_per_ring_opt = leds_per_ring;
223             center_distance_opt = center_distance;
224         end
225
226         % print various UV LED layout characteristics of the
227         % current iteration
228         fprintf("distance iteration = %d/%d, ", ...
229             center_distance_iterations, ...
230             max_center_distance_iterations);
231         fprintf("n_rings = %d, {", n_rings);
232         for n = 1:n_rings
233             fprintf(" n_leds(%d) = %d; ", n, leds_per_ring(n));
234         end
235         fprintf("}, {");
236         for n = 1:n_rings
237             fprintf(" r(%d) = %d; ", n, center_distance(n));
238         end
239         fprintf("}, opt. variable = %.3f, ", ...
240             (sum(leds_per_ring)+1) / irradiance_min);
241         fprintf(" I_min = %.3f, I_max = %.3f, ratio = %.3f", ...
242             irradiance_min, irradiance_max, irradiance_max/irradiance_min);
243         % print various UV LED layout characteristics of the
244         % current "best" implementation
245         fprintf("\n\tn_rings = %d, {", n_rings_opt);
246         for n = 1:n_rings_opt
247             fprintf(" n_leds(%d) = %d; ", ...
248                 n, leds_per_ring_opt(n));
249         end
250         fprintf("}, {");
251         for n = 1:n_rings_opt
252             fprintf(" r(%d) = %d; ", ...
253                 n, center_distance_opt(n));
254         end
255         fprintf("}, optimum (N/I_min) = %.3f, ", ...
256             optimization_variable1);
257         fprintf("optimum (I_max/I_min) = %.3f, ", ...
258             optimization_variable2);
259         fprintf(" I_min = %.3f, I_max = %.3f\n", ...
260             irradiance_min_opt, irradiance_max_opt);
261     end
262 end
263 end
```

A.2. Motor Controller

```
//Code for the microcontroller on the Motor Control Unit
//Authors: Daan Schat & Maxim Mazurovs
#include "communication_protocol.hpp"
#include <Wire.h>
using namespace UVO_CommunicationProtocol;

#define MCU_Adress UVO_CommunicationProtocol::MOTOR_CONTROLLER_ADDRESS
#define Pin 11

void receiveEvent(int howmany);
void requestEvent();
void currentsensorhandler(byte* cucommand);
void driverhandler(byte* cucommand);

int DutyCycle   = 0;
cons  int CurrentThreshold = 800;
const int motorPin    = 10;      // PWM output pin for the motor
const int Relay1      = 11;      // PWM output pin for the motor
const int Relay2      = 12;      // PWM output pin for the motor
const int currentPin  = A0;      // Analog input pin for current measurement
const int fadeTime    = 1500;    // Duration for each fade cycle in milliseconds
const int dutyHigh    = 150;     // Max 255, at least dutyLow
const int dutyLow     = 10;      // Min 0, lower than dutyHigh
const int dutyTime = dutyHigh - dutyLow;
bool succes = false;
bool succescurrent = false;
bool errorstate = false;
byte currentbytevalue[4];        //output value to send measured current to CU via I2C

void setup(){
  pinMode(motorPin, OUTPUT);      // Set motorPin as an output
  pinMode(Relay1, OUTPUT);        // Set motorPin as an output
  pinMode(Relay2, OUTPUT);        // Set motorPin as an output
  pinMode(currentPin, INPUT);     // Set currentPin as an input
  digitalWrite(Relay1, 0);        // Pull down Relays
  digitalWrite(Relay2, 0);
  Serial.begin(9600);             // start serial for output
  Wire.begin(MCU_Adress);         // join i2c bus with address 80
  Wire.onReceive(receiveEvent);   // register event
  Wire.onRequest(requestEvent);   // register event
}

void loop() {}

void receiveEvent(int howMany) {
  int commandlength = Wire.available();
  byte cucommand[commandlength];
  if( Wire.available() > 0)
  {
    Wire.readBytes(cucommand, commandlength);                    //store command from CU to a
        byte array
    switch (cucommand[0]) {                                      //read out first value of said
        array
      case PackageTypeToken::REQUEST_SENSOR_DATA:
        currentsensorhandler(cucommand);
        break;
      case PackageTypeToken::SET_DRIVER_RELATIVE_INTENSITY:
        driverhandler(cucommand);
        break;
      }
  }
}

void floatToBytes(float val,byte* bytes_array){
  // Function to convert a float to 4 bytes, the format to send data to Control Unit
  union {
    float float_variable;
    byte temp_array[4];
  } u;
```

```
68    u.float_variable = val;
69    memcpy(bytes_array, u.temp_array, 4);
70  }
71
72  void currentsensorhandler(byte* cucommand){
73    switch (cucommand[1]){
74      case MotorControlToken::SensorToken::CURRENTSENSOR_MOTOR:
75        //read out current value from pin which is an int between 0 and 1024 (10 bits)
76        int voltagevalue = analogRead(currentPin);
77
78        //calculate the voltage across the pin
79        float voltagefloatvalue = voltagevalue * 5.0 / 1024.0;
80
81        //calculate the current going through the shunt resistor (voltage/(gain*resistance))
82        float currentfloatvalue = voltagefloatvalue / (66 * 0.1);
83
84        floatToBytes(currentfloatvalue,&currentbytevalue[0]);
85        succes = true;
86        succescurrent = true;
87    }
88  }
89
90  void driverhandler(byte* cucommand) {
91      switch(cucommand[1]){
92        case MotorControlToken::DriverToken::PWM_MOTOR:
93          DutyCycle = int(cucommand[2]);                    //convert uint_8 format to int
94          succes = true;
95      }
96      analogWrite(motorPin, DutyCycle);                     //set motorpin to duty cycle value
97      //Serial.println(DutyCycle);
98    }
99
100 void requestEvent() {
101   // Send response back to CU
102   if (succes){
103     if (succescurrent){
104       Wire.write(currentbytevalue, 4);              //send 4 bytes with measured current
              value
105       succescurrent = false;
106     }
107     else {
108       Wire.write(UVO_CommunicationProtocol::ACK);   //send acknowledge to let know that
              action was successfull
109     }
110     succes = false;
111   }
112   else {
113     Wire.write(UVO_CommunicationProtocol::NACK);    //send NACK to let know that action was
          not successfull
114   }
115 }
```

B

LED Layout Optimization

B.1. Matlab optimization script output

```
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378
distance iteration = 489/666, n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r
(1) = 33;  r(2) = 45; }, opt. variable = 55.146,  I_min = 0.236, I_max = 0.421,
ratio = 1.788
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378
distance iteration = 490/666, n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r
(1) = 34;  r(2) = 45; }, opt. variable = 56.430,  I_min = 0.230, I_max = 0.423,
ratio = 1.837
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378
distance iteration = 491/666, n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r
(1) = 35;  r(2) = 45; }, opt. variable = 57.555,  I_min = 0.226, I_max = 0.425,
ratio = 1.881
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378
distance iteration = 492/666, n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r
(1) = 36;  r(2) = 45; }, opt. variable = 58.872,  I_min = 0.221, I_max = 0.427,
ratio = 1.933
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378
distance iteration = 493/666, n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r
(1) = 37;  r(2) = 45; }, opt. variable = 60.248,  I_min = 0.216, I_max = 0.429,
ratio = 1.986
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378
distance iteration = 494/666, n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r
(1) = 38;  r(2) = 45; }, opt. variable = 61.004,  I_min = 0.213, I_max = 0.429,
ratio = 2.015
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378
distance iteration = 495/666, n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r
(1) = 39;  r(2) = 45; }, opt. variable = 62.123,  I_min = 0.209, I_max = 0.431,
ratio = 2.057
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378
distance iteration = 496/666, n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r
(1) = 40;  r(2) = 45; }, opt. variable = 63.323,  I_min = 0.205, I_max = 0.431,
ratio = 2.100
    n_rings = 2, { n_leds(1) = 3;  n_leds(2) = 9; }, { r(1) = 18;  r(2) = 45; },
optimum (N/I_min) = 52.532, optimum (I_max/I_min) = 1.526,  I_min = 0.247, I_max =
0.378

ans =

    1.5260

>>
```

C

PCB Schematics

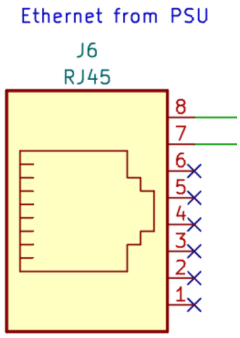


Figure C.3: Ethernet - PSU - pin 8 = GND, pin 7 = +24 V

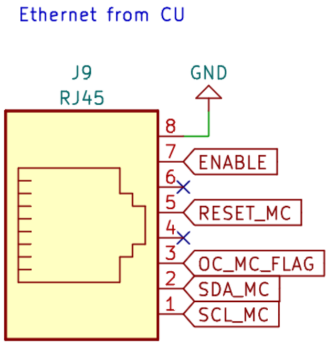


Figure C.4: Ethernet - Control Unit

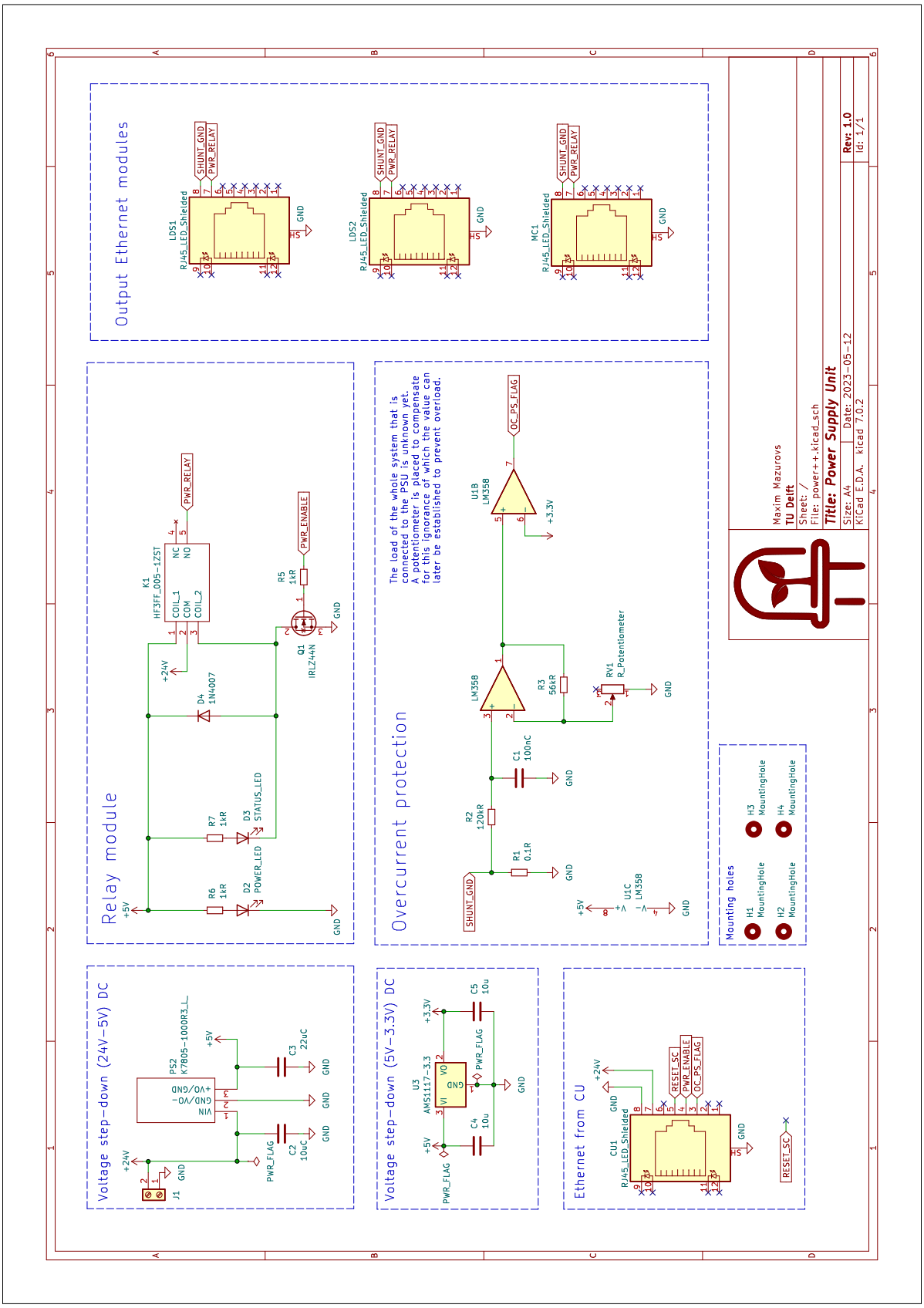


Figure C.1: Power Supply Unit schematic

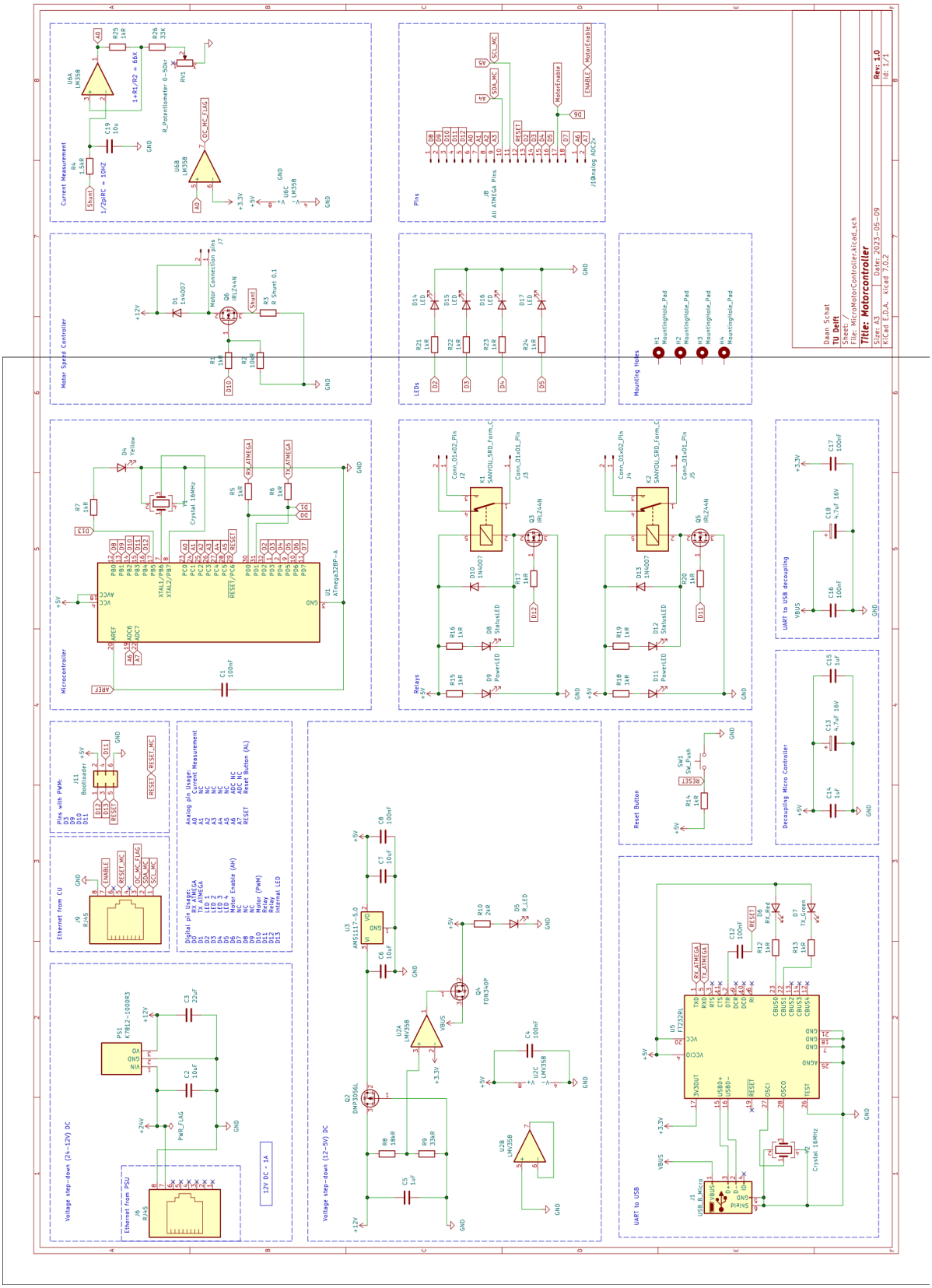


Figure C.2: Motor & Micro-controller schematic

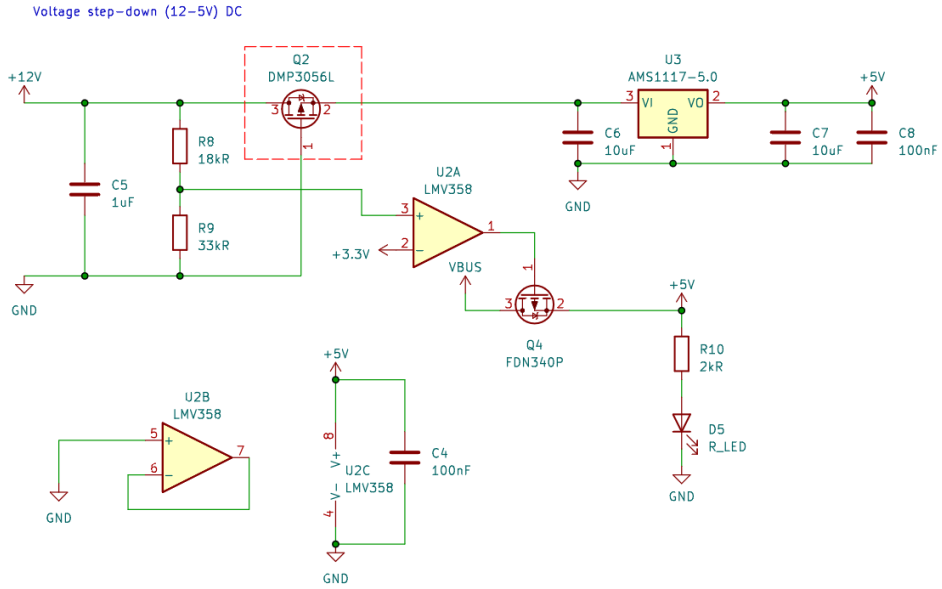


Figure C.5: 12-5V conversion

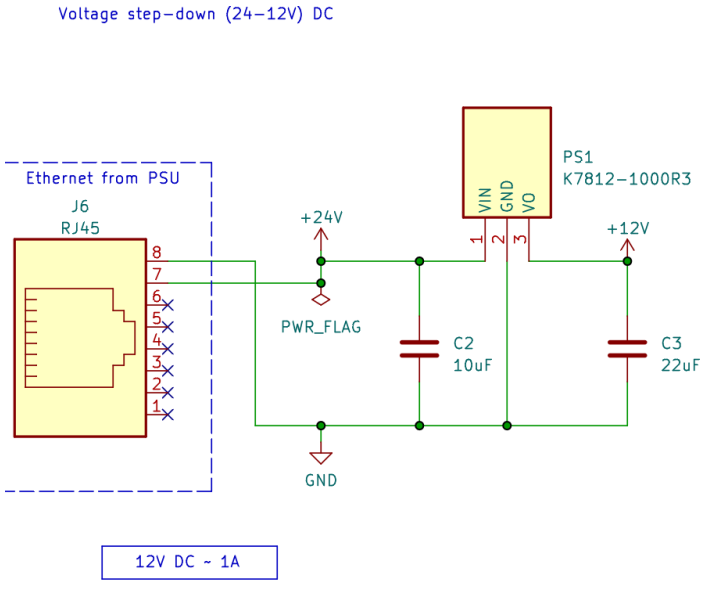


Figure C.6: 24-12V conversion

D

PCB Designs

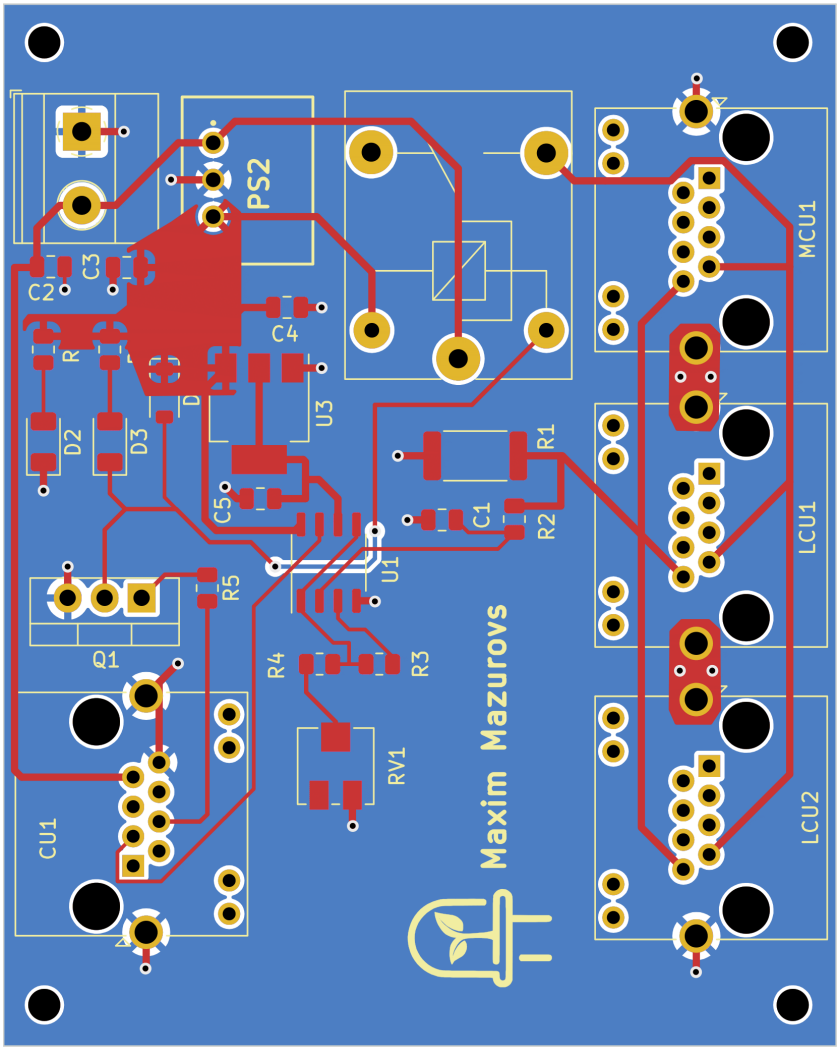


Figure D.1: Power Supply Unit PCB Design

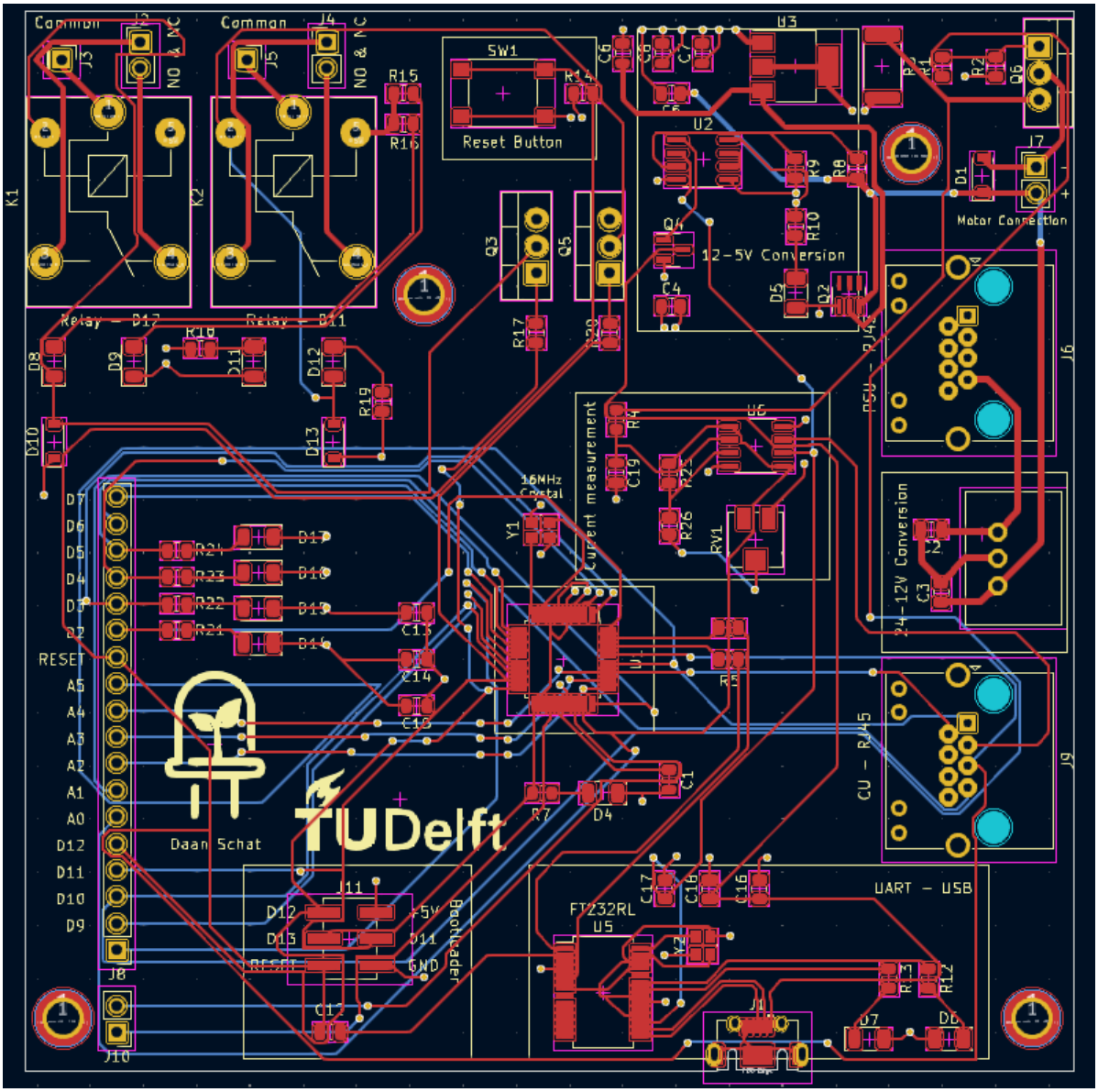


Figure D.2: Motor Controller & Microcontroller PCB Design

E

PCB 3D models

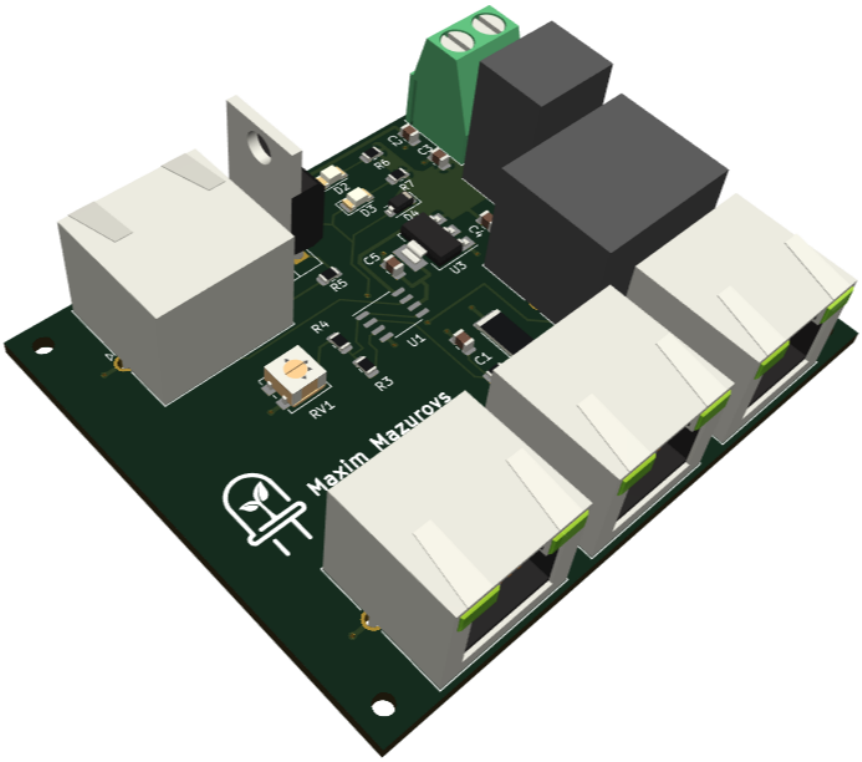


Figure E.1: Power Supply Unit PCB 3D Model

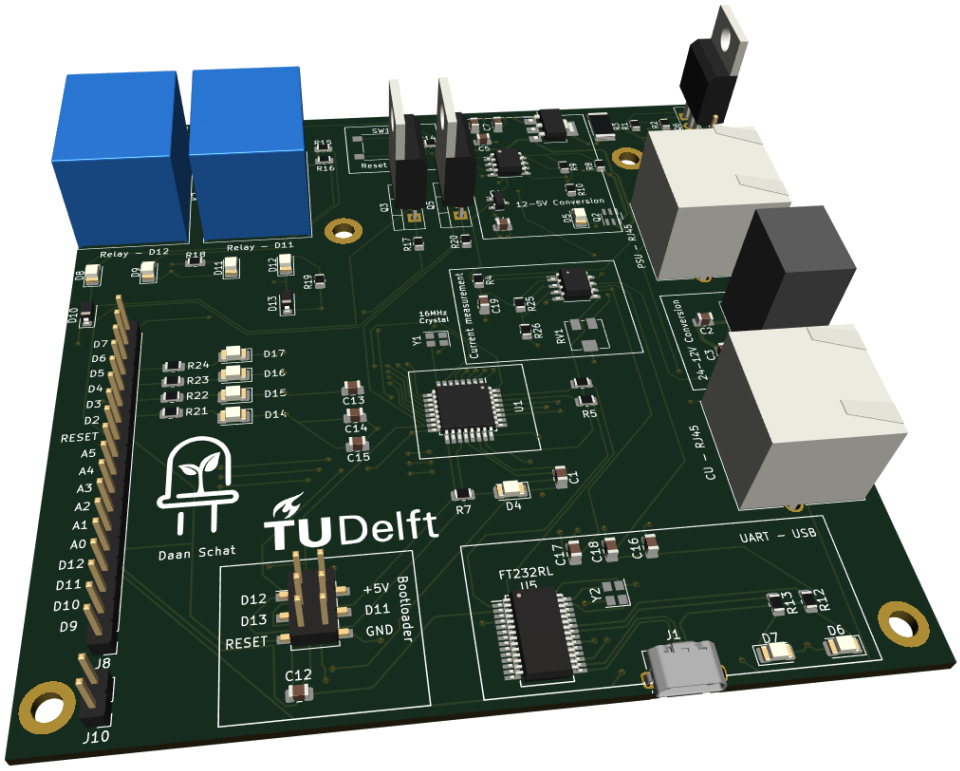


Figure E.2: Motor & Micro-controller PCB 3D Model

F

Machine 3D Model

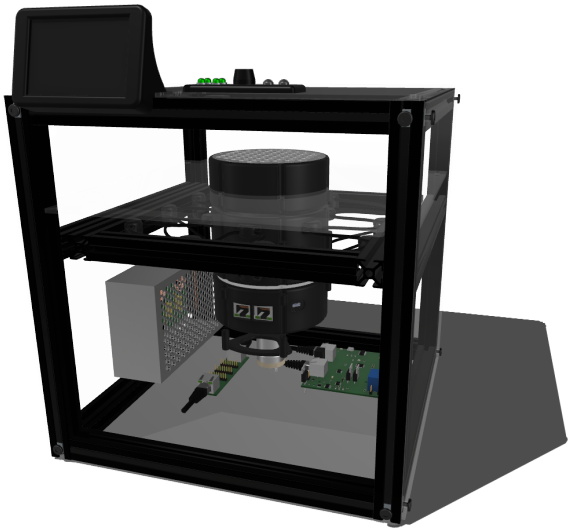


Figure F.1: Full device

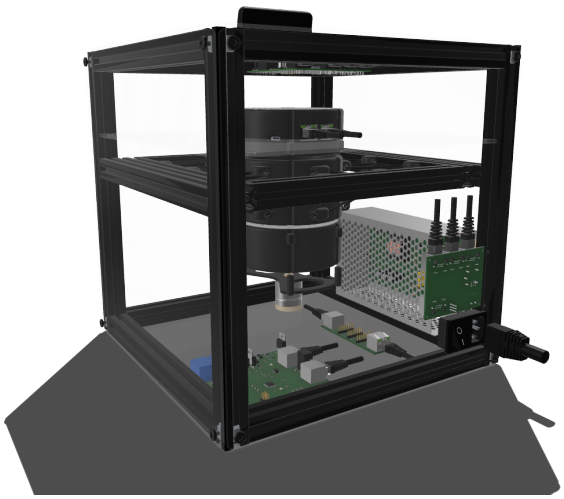


Figure F.2: Rear view of the device

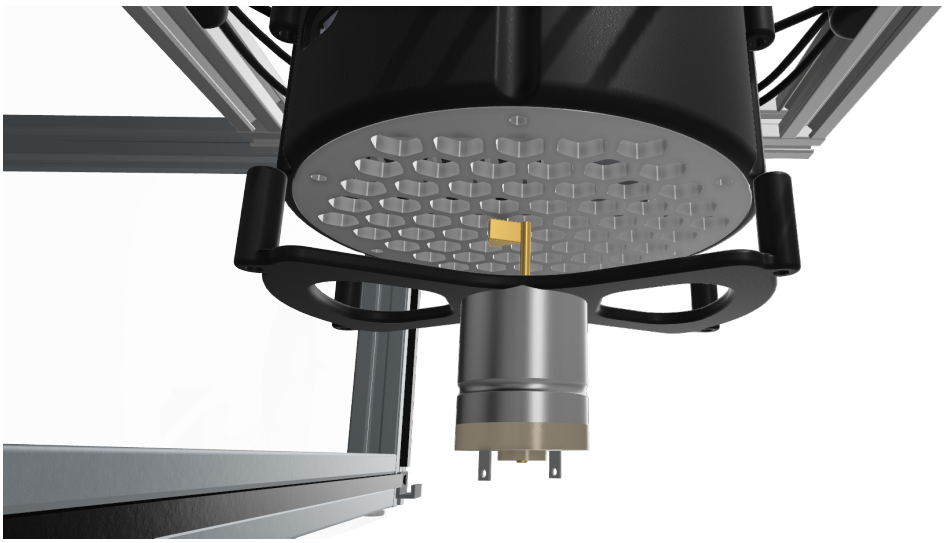


Figure F.3: Motor Mount

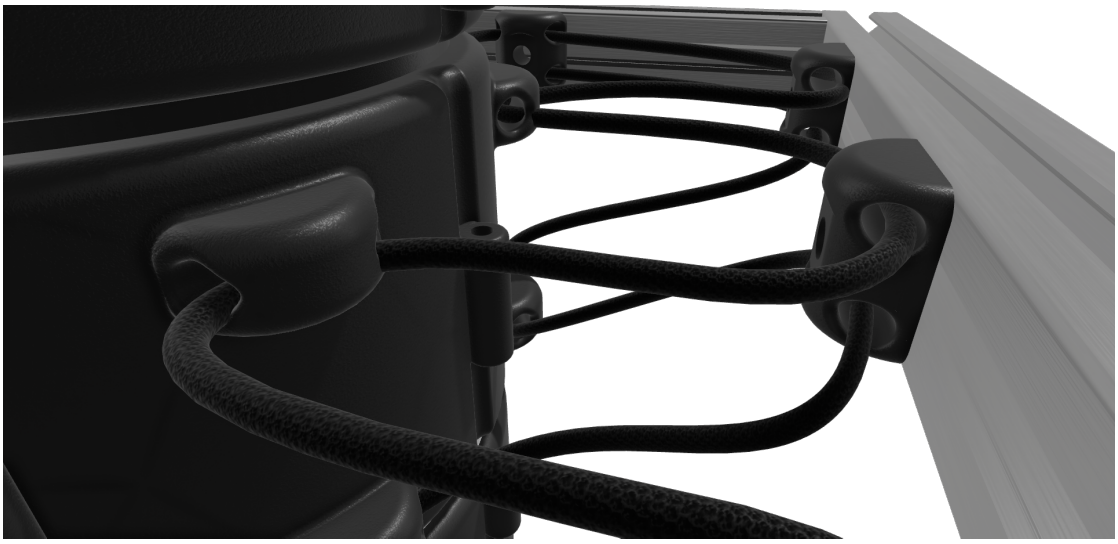


Figure F.4: ”Stack” mounting mechanism

G

Test results and discussion

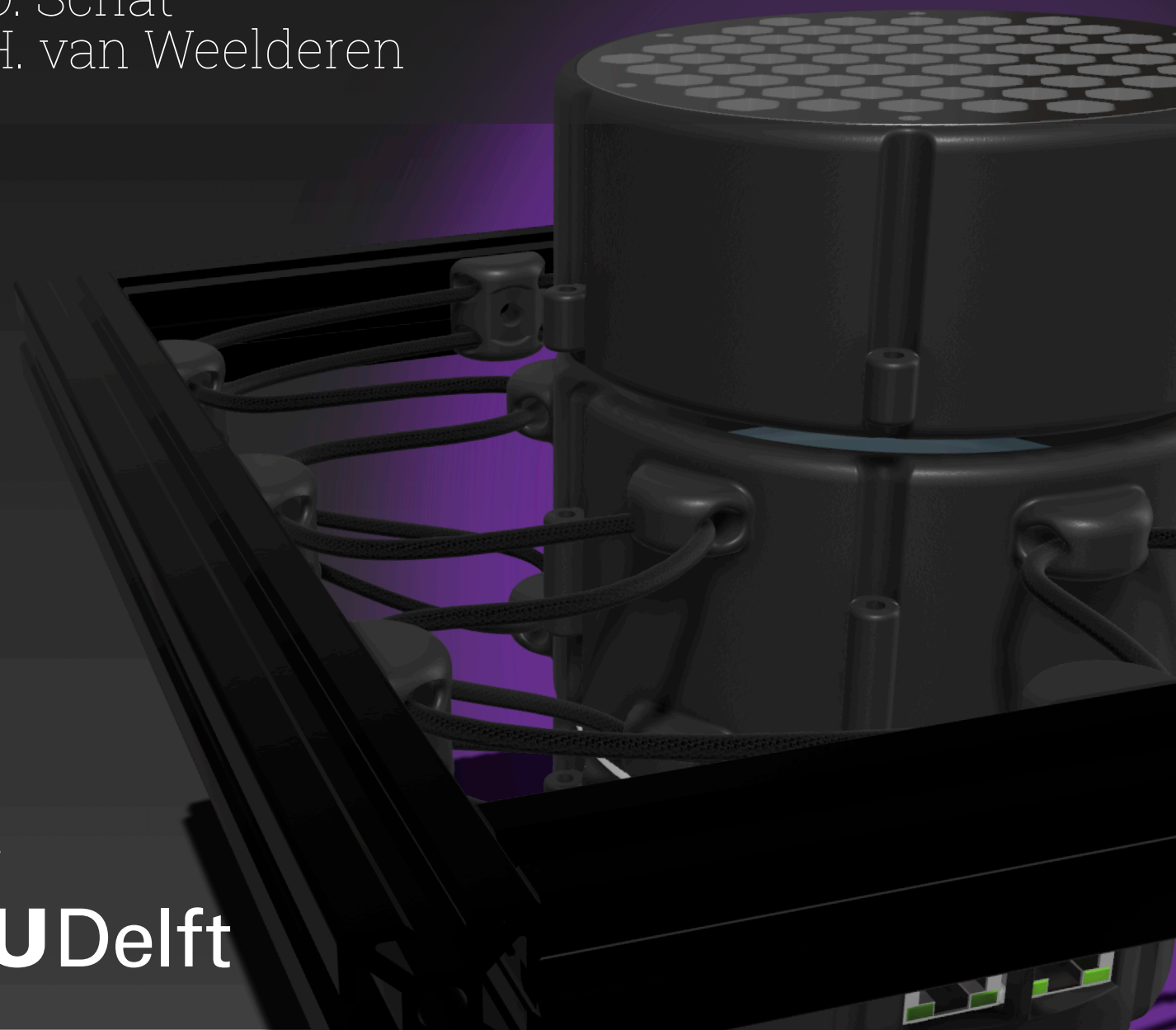
UV-C LED Seed Disinfection

Test results

EE3L11: Bachelor Graduation Project

E. Ergül
R.W.L. Imbens
L.C. Klootwijk
M. Mazurovs
D.O. Schat
E.H. van Weelderen

Delft University of Technology



TU Delft

Preface & acknowledgement

This report presents the experiments of the research project aimed at the validation of the custom-built UV-C LED disinfection device, targeted to disinfect fungi from seed. The project represents a collaborative effort by the team of students that set out to develop this device. Drawing upon the team's collective expertise in Electrical Engineering, there is designed and constructed a specialized UV-C LED device capable of irradiating seed with far-ultraviolet light, known to be capable of disinfection. The objective of this study was to evaluate the device's performance in seed disinfection and to test different setups in terms of wavelength to obtain meaningful results in hopefully eradicating pathogens, thereby enhancing seed quality.

Throughout the project, the testing protocol was implemented to assess the device's ability to disinfect cabbage seed. The team prepared and conducted controlled experiments, in which multiple variables such as exposure time, the wavelength of the UV-C light, and the intensity of UV-C radiation could be controlled.

We express our deepest gratitude to all those who contributed to this endeavour, including our team members, advisers, and the support of Rijk Zwaan who provided resources and guidance. It is our hope that this report will not only advance scientific knowledge but also inspire further exploration and adoption of UV-C using LED.

E. Ergül
R.W.L. Imbens
L.C. Klootwijk
M. Mazurovs
D.O. Schat
E.H. van Weelderen
Delft, June 2023

Abstract

This report gives the results of using UV-C LED light for disinfecting plant seeds. This study focuses on destroying Alternaria fungi on rape seeds. The machine used for the tests is a custom developed UV-C radiating device which is enclosed in a safe casing for user-safety and features an integrated control unit for easy selection of variables and parameters. Parameters that can be set are wavelength, intensity and dose. Also, there is a vibration motor built into the seed bed so that seeds can be turned around during testing to ensure an even radiation. The method used for testing is the PCR-method. The results of the tests on the seeds does not give a significant difference, so conclusions can not be determined yet. Other methods are needed to determine the effectiveness of the tests. Based on integration testing of the machine and the literature, the machine has the potential to successfully disinfect the seeds, so doing the recommended future work can be very interesting.

Nomenclature

Abbreviations

Abbreviation	Definition
LED	Light Emitting Diode
UVC & UV-C	Lowest range UV light (100 - 280 nm)
PSU	Power Supply Unit
CU	Control Unit
MCU & MC	Motor Control Unit
LDS	LED Driving & Sensing module
PCB	Printed Circuit board
IC	Integrated-Circuit, usually referring to a chip or module
PoR	programme of requirements
MR	Mandatory requirements
ToR	trade-off requirements

Symbols

Symbol	Definition	Unit
P	Power	[W]
D	Dose	[mJ/cm^2]
t	Time	[s]
λ	Wavelength	[nm]

Contents

1 Introduction

As the world population increases and food production should be more and more efficient and sustainable, improvements and offering more sustainable technologies in the agricultural sector become more and more important. Seed disinfection plays a critical role in ensuring the quality and health of crops. Contamination by pathogenic microorganisms, such as Alternaria, can significantly impact seed quality and crop yield. In the pursuit of effective seed disinfection methods, our research team has conducted a study to evaluate the efficacy of an in-house built UV-C device hopefully capable of eliminating microbial contaminants from seeds.

This technical report presents the findings of the disinfection from this UV-C device and an analysis of some key parameters. With the use of the far-ultraviolet irradiation principles, the device offers a promising alternative to conventional warm seed bathing or even chemical treatments by effectively inactivating fungi.

The primary objective of our research was to assess the performance of the UV-C device in disinfecting cabbage seeds. We aimed to determine the device's impact on seed quality in terms of reduction in Alternaria. By varying exposure time, wavelength, and intensity of UV-C radiation, we sought to identify optimal disinfection parameters that balance effective pathogen elimination.

To achieve accurate and reliable results, the testing procedures adhered to standardised protocols for all tests. We performed multiple replicates of each experiment to ensure robustness and consistency in the data. Moreover, we employed appropriate control groups and reference samples to establish a baseline for comparison and validate the efficacy of the UV-C device.

In the subsequent sections of this report, we will present a detailed description of our methodology, including the setup of the UV-C device and experimental procedures. We will then provide an analysis of the results obtained. Additionally, we will discuss the limitations encountered during the testing process and offer recommendations for further research and potential applications of the UV-C device in seed disinfection practices.

1.1. Problem definition

The disinfection of seeds is a critical process in the seed breeding industry to ensure optimal quality and therefore crop yield. However, traditional seed disinfection methods often rely on inefficient treatments. UV-C might be the next step in industry-leading disinfection methods.

In light of this objective, our research focuses on evaluating the efficacy of our custom-built UV-C device for seed disinfection. The primary objective addressed in this technical report is to determine the performance of this UV-C device in eliminating the fungus Alternaria on cabbage seed. By assessing the effectiveness of the UV-C device, we aim to address the following key questions:

- Can the UV-C device effectively inactivate (Alternaria) fungi on the seed?

- What are the optimal parameters, such as exposure time, the wavelength of UV-C, and the intensity of UV-C radiation for effective seed disinfection without compromising seed viability?
- What are the potential practical applications of the UV-C device in seed disinfection practices, and how does it contribute to sustainable and environmentally friendly agricultural practices?
- How does the UV-C device compare to the conventional warm water bathing method in terms of microbial reduction and overall seed quality?

The testing of Alternaria residue will be performed by Rijk Zwaan, located in De Lier, one of the world leaders in seed breeding. Rijk Zwaan will make use of PCR testing to determine the seed disinfection.

By providing answers to these questions, this technical report aims to contribute to the development and adoption of advanced seed disinfection technologies that offer efficient and more sustainable pathogen elimination while maintaining seed quality.

1.2. Hypothesis

Based on the literature, the hypothesis is that UVC-LED light can be used to disinfect the seeds. The Alternaria fungi can be eliminated by destroying the DNA and RNA structures by using a wavelength of 255 nm and the protein structures can be destroyed by using a combination of 275 nm and 285 nm.

2

Testing setup

Firstly the test setup will be explained. The test setup consists of a seed bed surrounded by two LED panels, which radiate UV-C light onto the seeds. A vibration motor will move and turn all seeds so that the seed surface is radiated as evenly as possible. Figure 2.1 provides an overview of the test setup.

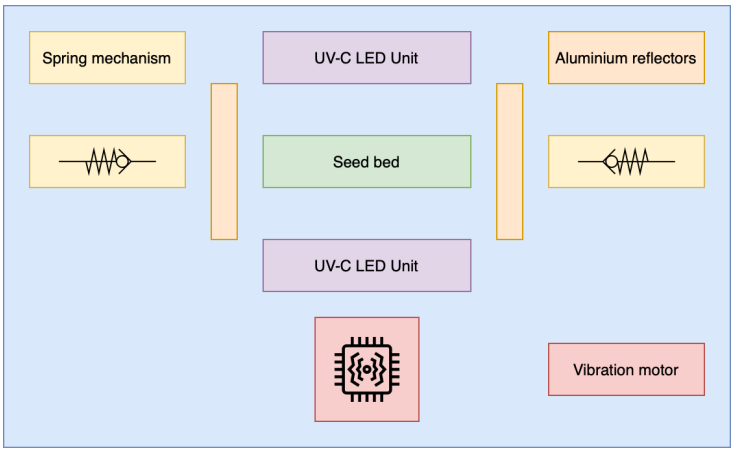


Figure 2.1: Seed distribution on a circular plate with 10cm diameter assuming 2.75 mm seed thickness.

The seedbed is a circular quartz plate with a diameter of 10 centimetres. The surface of the radiated area can then be calculated according to Equation 2.1.

$$A_{SeedBed} = \pi \cdot r^2 = \pi \cdot \left(\frac{d}{2}\right)^2 = \pi \cdot \left(\frac{10}{2}\right)^2 = 78.5 \text{ cm}^2 \tag{2.1}$$

Assuming a rapeseed diameter of 2 mm on average with an additional 0.75 mm margin per seed for stacking and placing seeds, the number of seeds that can be tested in one batch on the plate can be simulated. This can be observed in Figure 2.2.

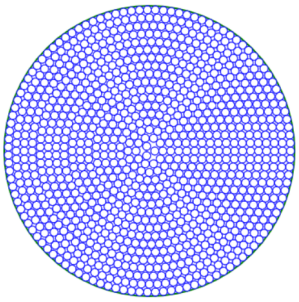


Figure 2.2: Seed distribution on a circular plate with 10cm diameter assuming 2.75 mm seed thickness.

The LEDs used in the setup and the specs are shown in Table 2.1. The optical output power is an essential parameter since the radiation time is dependent on this if a certain dose is desired to be radiated.

Wavelength	Typical Forward Voltage	Typical Forward Current	Dissipated Power	Optical Output Power
255	7.5V	100mA	0.75W	3.5mW
275	6.0V	85mA	0.51W	11.5mW
285	6.5V	90mA	0.585W	10mW
395	3.2V	75mA	0.24W	11.5mW

Table 2.1: Specifications of the LEDs used in the machine.

Radiation dose can be calculated using Equation 2.2. Each LED panel consists of 12 LEDs. Radiating is done from both sides, so the amount of LEDs and radiations surface is doubled.

$$Dose = \frac{P_{total} \cdot t}{2 \cdot A_{SeedBed}} = \frac{2 \cdot 12 \cdot P_{optical} \cdot t}{A_{SeedBed}} \tag{2.2}$$

Rewriting this equation to get the radiation time gives:

$$t = \frac{A_{SeedBed} \cdot Dose}{2 \cdot 12 \cdot P_{optical}} \tag{2.3}$$

3

Methodology

In order to achieve meaningful results, Rijk Zwaan has provided PCR testing to determine the residue of Alternaria and cabbage seed to disinfect. Unfortunately, time and test size constraints will limit the extensiveness of this research. Because of this, ten individual tests with different parameters can be performed, each of which will have a similar test as the control.

Before testing the actual cabbage seed, a pre-test is used to evaluate the radiation pattern of the UV-C LED disinfecting device. A silicon wafer is placed as a substitution for the quartz plate, which will act as a target. The silicon wafer, known for its light absorption characteristics, will be examined after exposure to UV-C. This pre-test aimed to visualize and analyze the distribution of UV-C radiation emitted by the device.'

Extending tests may be possible by the use of yeast. Irradiating yeast in its powder form commonly found in grocery stores can be an indicator of disinfection when added to sugar water. When the yeast is irradiated and killed, it should not create any bubbles in the sugar water.

3.1. Testing parameters for PCR tests

PCR tests can indicate the residue of genetic material, in this case Alternaria, on the seeds and can therefore be used to determine if the device works. Because of the limitations in test size of 10, some meaningful test setups with different parameters should be explored. Table 3.1 gives the chosen parameters for each individual test.
The following parameters can be set and researched; wavelength, motor speed, time and dose. The dose is related to the time and intensity of certain LEDs. It should be noted wavelength 3, $\lambda_3 95$, is not in the UV-C spectrum, but is UV-A. Literature suggested that pre-treatment with longer wavelengths of ultraviolet light can activate the pathogens and therefore make them weaker than in their "sleep".

3.1.1. Selecting proper dose

The dose is proportional to the exposure time and intensity of the LEDs. Choosing a useful exposure time is crucial for optimal test results. Based on literature, Alternaria can be eliminated by radiating with a dose of around $5kJm^{-2}$ [1, 2]. To investigate the relation of doses versus pathogen presence, seeds will be radiated by different radiation doses. The expected relation is a logarithmic curve. The doses which are chosen for different radiation levels are the following:

- $0.25\,\mathrm{kJ/m^2}$
- $1\,\mathrm{kJ/m^2}$
- $5\,\mathrm{kJ/m^2}$

According to Equation 2.3, the efficiencies from Table 2.1, and the area from Equation 2.1, the radiation time can now be calculated for each LED.

$$t_{\lambda_{255nm}(0.25kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 0.25 \cdot 10^{3}}{2 \cdot 12 \cdot 3.5 \cdot 10^{-3}} = 23.2s$$

$$t_{\lambda_{255nm}(1kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 1 \cdot 10^{3}}{2 \cdot 12 \cdot 3.5 \cdot 10^{-3}} = 92.9s$$

$$t_{\lambda_{255nm}(5kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 5 \cdot 10^{3}}{2 \cdot 12 \cdot 3.5 \cdot 10^{-3}} = 464.3s$$

$$t_{\lambda_{275nm}(0.25kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 0.25 \cdot 10^{3}}{2 \cdot 12 \cdot 11.5 \cdot 10^{-3}} = 7.07s$$

$$t_{\lambda_{275nm}(1kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 1 \cdot 10^{3}}{2 \cdot 12 \cdot 11.5 \cdot 10^{-3}} = 28.3s$$

$$t_{\lambda_{275nm}(5kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 5 \cdot 10^{3}}{2 \cdot 12 \cdot 11.5 \cdot 10^{-3}} = 142.2s$$

$$t_{\lambda_{395nm}(0.25kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 0.25 \cdot 10^{3}}{2 \cdot 12 \cdot 11.5 \cdot 10^{-3}} = 7.07s$$

$$t_{\lambda_{395nm}(1kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 1 \cdot 10^{3}}{2 \cdot 12 \cdot 11.5 \cdot 10^{-3}} = 28.3s$$

$$t_{\lambda_{395nm}(5kJm^{-2})} = \frac{78.5 \cdot 10^{-4} \cdot 5 \cdot 10^{3}}{2 \cdot 12 \cdot 11.5 \cdot 10^{-3}} = 142.2s$$

3.1.2. Parameter selection

Test plan	Motor	Hot air [°C]	λ_{255nm}	λ_{275nm}	λ_{395nm}	time [s]	Dose [kJ/m^2]
Test 1, control	off	-	-	-	-	-	-
Test 2	off	-	100%	-	-	92.9	1
Test 3	on	40	100%	-	-	92.9	1
Test 4	on	-	100%	-	-	23.2	0.25
Test 5	on	-	100%	-	-	464.3	5
Test 6	on	-	100%	-	-	92.9	1
Test 7	on	-	-	100%	-	7.07	0.25
Test 8	on	-	-	100%	-	28.3	1
Test 9	on	-	-	100%	-	142.2	5
Test 10	on	-	100%	-	100%	92.9 λ_{255nm}, 28.3 λ_{395nm}	1+1

Table 3.1: Tested parameters

The tests setups in table 3.1 explained:

- Test 1 will be a control without UV treatment, motor vibration and hot air pre-treatment.
- Test 2 tests the effectiveness of the motor.
- Test 3 tests the effectiveness of a hot air pre-treatment. By comparing tests 1, 2, 3 and 6 the effectiveness of the motor and the hot air-pre-treatment can be determined.
- Test 4 tests the effectiveness of λ_{255} with a dose of 0.25 [kJ/m^2].
- Test 5 tests the effectiveness of λ_{255} with a dose of 5 [kJ/m^2].
- Test 6 tests the effectiveness of λ_{255} with a dose of 1 [kJ/m^2].
- Test 7 tests the effectiveness of λ_{275} with a dose of 0.25 [kJ/m^2].
- Test 8 tests the effectiveness of λ_{275} with a dose of 1 [kJ/m^2].
- Test 9 tests the effectiveness of λ_{275} with a dose of 5 [kJ/m^2].
- Test 10 tests the effectiveness of pre-treatment with a longer wavelength, in UV-A, λ_{395} in combination with λ_{255}. Each wavelength will deliver a dose of 1 [kJ/m^2], making the total 2 [kJ/m^2]. To be compared with test 6.

Please note that test 10 actually delivers a total dose of 2 [kJ/m^2]. This is not expected to spoil the result because UV-A is not known for strong germicidal and disinfecting capabilities.

3.2. Testing parameters for silicon tests

3.2.1. Selecting proper dose

Silicon wafers are photoreactive for wavelengths between $300\,\text{nm}$ and $500\,\text{nm}$. The silicon wafers can give an indication about the irradiation pattern and uniformity of radiation on the plate.

The proper dosage of UV-C radiation in which results would be visible is about $40\,\text{mJ/cm}^2$. As stated earlier, the LEDs have an optical power output of:

- 3.5 mW for a 255 nm LED
- 11.5 mW for a 275 nm LED
- 10 mW for a 285 nm LED
- 11.5 mW for a 395 nm LED

As stated earlier, the 285 nm LEDs don't work properly and therefore cannot be tested. the area of the plate is 78.5 cm^2. There are 12 LEDs for each LED type. As the wafer is only photo reactive from the top, only the top PCB will be turned and used for irradiation. Note silicon is mostly reactive in the 300-500 nm region, so the 395 nm LEDs are expected to yield the most promising results.

$$P_{area,255nm} = \frac{3.5 \cdot 12}{78.5} = 0.53\,\text{mW/cm}^2$$

$$P_{area,275nm} = \frac{11.5 \cdot 12}{78.5} = 1.75\,\text{mW/cm}^2$$

$$P_{area,395nm} = \frac{11.5 \cdot 12}{78.5} = 1.75\,\text{mW/cm}^2$$

Meaning, for $40\,\text{mJ/cm}^2$ the device should be turned on for a total of 74.8 seconds for the 255 nm LEDs and 22.8 seconds for the 395 and 275 nm LEDs. Some meaningful parameter testing is given in Table 3.2.

Parameter selection					
Test plan	Reflector	Scanner	λ_{395nm}	time [s]	Dose $[mJ/cm^2]$
Test 1	no	no	100%	22.8	40
Test 2	no	no	100%	45.6	80
Test 3	yes	no	100%	22.8	40
Test 4	yes	no	100%	45.6	80
Test 5	no	yes	100%	22.8	40
Test 6	yes	yes	100%	45.6	80
Test 7	yes	yes	100%	22.8	40
Test 8	no	yes	100%	45.6	80

Table 3.2: Tested parameters

The following tests are conducted:

- Test 1 tests the radiation pattern of the 395 nm LEDs at the suggested dose.
- Test 2 tests the radiation pattern of the 395 nm LEDs at double the suggested dose.
- Test 3 and 4 tests the same parameters with the reflector put onto the sides of the seed bed.
- Test 5 till 8 tests the same parameters again so that the test set is doubled and on of the sets can be used to directly see results after scanning it.

3.3. Testing yeast

To get an indication of the radiation needed to destroy fungi and to verify the calculations and effectiveness of the tests above the machine can be tested by loading yeast into the seed bed. To test the viability of yeast a sugar solution in water can be used. Testing this requires a similar concentration of

sugar to test normal yeast and yeast radiated with UV-C light. Testing was done by filling up a translucent glass of water with 200 mL of water and mixing 30 grams of sugar. One of the glasses can be used as a control test by monitoring the carbon dioxide (CO_2) production. This can be compared with the production of carbon dioxide from the yeast radiated with the light in the machine. Radiation is done by a dose of $0.5\,\mathrm{kJ/m^2}$.

3.4. Testing protocol

To ensure accuracy and reliability in the evaluation of the UV-C device's performance for seed disinfection, a standardized testing procedure was implemented. The procedure aimed to minimize potential errors and inconsistencies, thereby providing consistent, reliable, and comparable results throughout the testing process.

First, a controlled testing environment was established. This environment maintained similar conditions as the tested seeds in terms of temperature, humidity, and lighting conditions to eliminate any external factors that could influence the test outcomes.

The seed-disinfection device created by our team was employed for each trial to ensure reproducibility. The seeds used in the tests were sourced from the same batch and were of uniform size and quality. To guarantee consistency, a specific number of seeds were selected for each trial, namely 500, and they have been evenly distributed across the test area.

The distance between the UV-C device and the seeds was kept constant throughout the testing process. This distance and LED placement were determined based on prior research and optimization.

To further minimize errors, the exposure time was standardized for each test. Two predetermined durations were chosen based on literature studies for effective seed disinfection of fungi (Alternaria). This ensured that the seeds received the same UV-C dosage during each trial, allowing for an accurate comparison of the disinfection outcomes.

In between tests, the quartz plate, which holds the seed, is thoroughly disinfected by radiation without seeds in order to assure no pathogens are carried over to the next test batches.

Finally, when placing and removing the seeds from the quartz testing plate, no physical contact with hands or non-sterile equipment is permitted. The seeds are placed in sterilized bags outside the device in order to assure safe and sterile transport to Rijk Zwaan.

4

Results

4.1. PCR test results

The test results following from the PCR tests performed by Rijk Zwaan are given in Table 4.1. The subsequent graphs are given in Figures 4.1, 4.2, and 4.3.

	Sample	FAM (A.brassicicola)	VIC (A.brassicicola)	Cy5 (A. brassicae)	TxR (IAC Acat)	Uitslag
S	1.1	20,98	24,80	N/A	19,48	A.brassicicola
	2.1	21,00	24,63	37,74	19,36	A.brassicicola
	3.1	20,21	24,04	N/A	19,49	A.brassicicola
	4.1	21,31	25,14	N/A	19,40	A.brassicicola
	5.1	21,78	25,40	N/A	19,46	A.brassicicola
	6.1	21,14	25,04	N/A	19,49	A.brassicicola
	7.1	20,88	24,48	N/A	19,67	A.brassicicola
	8.1	21,17	25,16	N/A	20,41	A.brassicicola
	9.1	20,70	24,54	N/A	19,62	A.brassicicola
	10.1	21,58	25,42	N/A	19,47	A.brassicicola
	1.2	21,10	25,06	N/A	19,55	A.brassicicola
	2.2	20,75	24,43	38,81	19,68	A.brassicicola
	3.2	21,37	25,27	36,79	19,81	A.brassicicola
	4.2	21,11	24,79	N/A	19,69	A.brassicicola
	5.2	21,81	25,55	N/A	19,76	A.brassicicola
	6.2	22,67	26,38	N/A	20,62	A.brassicicola
	7.2	21,05	24,82	N/A	19,75	A.brassicicola
	8.2	21,05	24,86	N/A	19,45	A.brassicicola
	9.2	21,20	25,04	N/A	19,49	A.brassicicola
	10.2	21,12	25,11	N/A	19,72	A.brassicicola
C	QC_NTC	N/A	N/A	N/A	N/A	Oke
	QC_NPC	N/A	N/A	N/A	21,57	Oke
	QC_NAC	N/A	N/A	N/A	23,20	Oke
	QC_PC	19,69	18,03	19,39	N/A	Oke

Table 4.1: PCR test results from Rijk Zwaan. The S indicates the samples, the C indicates the PCR-control tests.

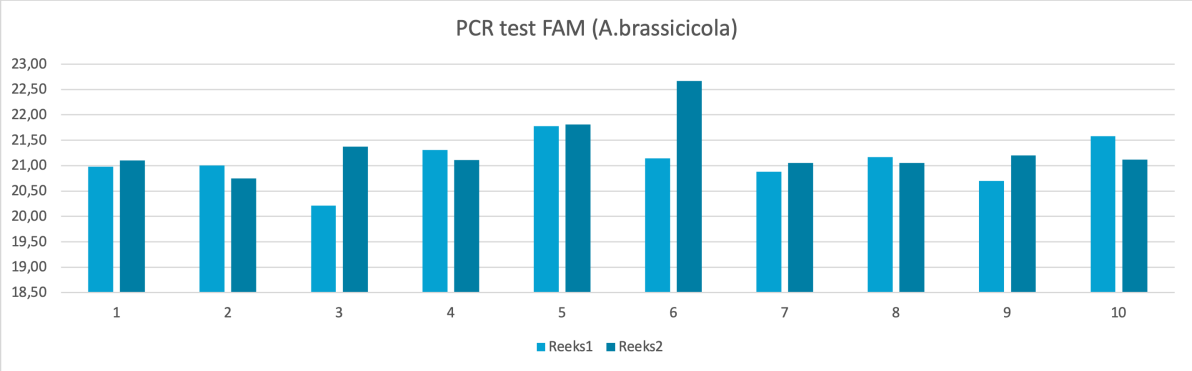


Figure 4.1: The PCR Test tested with FAM.

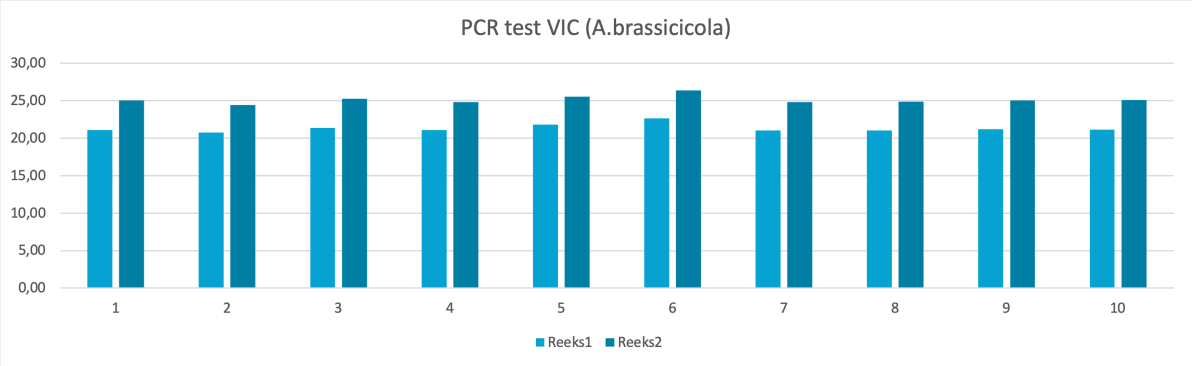


Figure 4.2: The PCR Test tested with VIC.

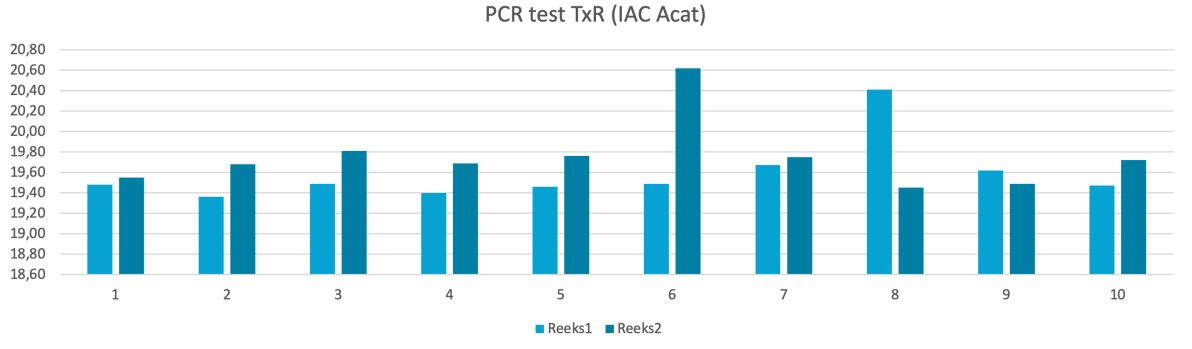
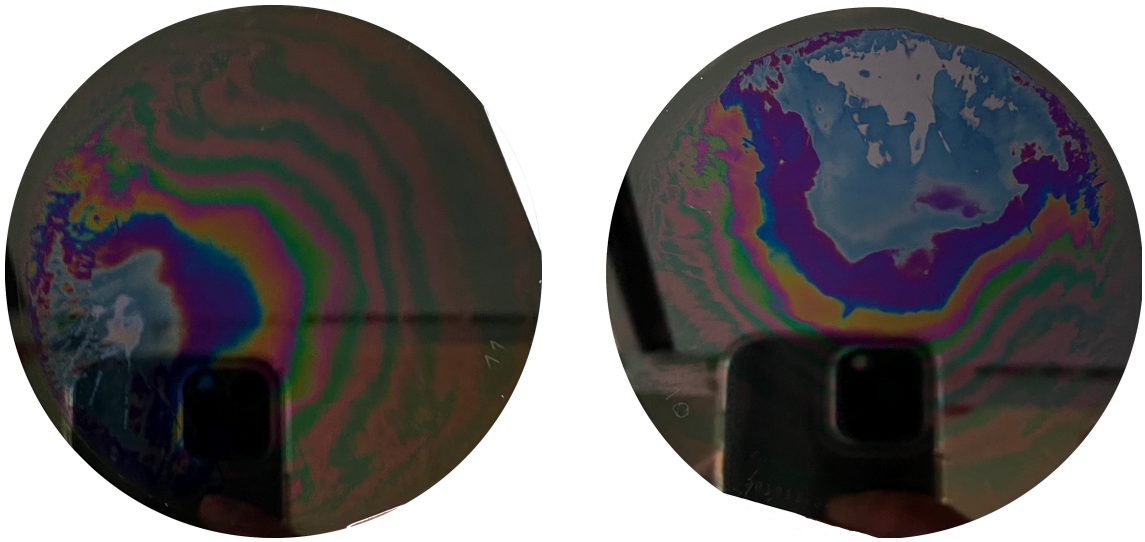


Figure 4.3: The PCR Test tested with TxR.

4.2. Silicon wafer test results

The results for the silicon wafer test will mostly be visual comparison and therefore the data is qualitative. The results are given in Figure 4.4.



(a) The silicon wafer radiated with a dose of $0.4\,\mathrm{kJ/m^2}$.

(b) The silicon wafer radiated with a dose of $0.8\,\mathrm{kJ/m^2}$.

Figure 4.4: Silicon wafer radiated with UV-C light.

The silicon wafers are developed in a basic solution so that the exposed parts are dissolved and the radiation pattern becomes visible.

4.3. Yeast test results

Yeast was tested in two glasses of water. Results are that the yeast from the control test was actively producing CO_2 and the radiated yeast was notably less active. The foam head as a result of the production of carbon dioxide was way thinner than the control test.

5 Discussion

It is important to note that even with strict adherence to standardised procedures, it is possible for some degree of variability to exist due to biological variations among seeds but also slight human errors. However, by implementing controls and consistent testing practices, the impact of such variations can be minimised, ensuring reliable and valuable data for the evaluation of the UV-C device's performance in seed disinfection.

Also, the test results of the PCR tests measure how much DNA or RNA there is still left for a certain test string after the tests have been executed. In the case of these test results, test strings of the pathogen A. brassicicola were used as a reference. From this, it could be investigated how much of these pathogen material is still left in the samples. However, the PCR results do not point out whether these structures are viable or not. Seeds can be disinfected while still detecting their structures. If those structures are not viable anymore, they are not harmful to the plants, which is the desired result. To summarize, the PCR method is not a good method to conclude whether seeds are disinfected, but only to detect if DNA or RNA structures are still present.

Actually, this was not yet known at the beginning of the project, but this is a possible declaration for the test results, which do not show any significant results. For future work, the ELIZA method is probably a better method to test the effectiveness of the machine.

This method does not test the DNA or RNA structures, but it tests the protein structures. Detecting proteins with enzymes is a much more specific process, so the viability of the pathogens can be determined more reliably.

Another method which would probably work better than the PCR method is the blotter or malt agar method used by the International Seed Health Association [4]. In short how it works: treated seeds get placed on a petridish on a medium, filter/blotter paper or malt agar. After, the seeds get incubated and then freezed. In the end, the seeds germinate and Alternaria grows too if it is present on the seeds. This takes a bit more time than the PCR method, however, less time than planting the seeds and examining the leaves.

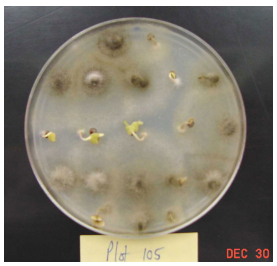


Figure 5.1: Cabbage seed plated onto a semi-selective agar medium to detect Alternaria brassicicola [3]

The results of the silicon wafer test can be used to verify the dose of the LEDs. The radiation pattern cannot be determined reliably because the wafers are developed by hand which is much less accurate than doing it automatically by a machine.

6

Conclusion

From the results in chapter 4, it can be seen that there are no significant differences in the results. Although it cannot be concluded that the machine can successfully disinfect seeds, the PCR method is not the best method to verify this. As explained in the Discussion section, future work is needed to show if the machine can disinfect seeds. This conclusion is colourable since the yeast test results show that UV-C light can eliminate fungi. The integration of the machine is successful. All submodules are tested individually to ensure reliability and safety for both the electronic and user side. The integration is also tested and the machine can select the user input and set the settings into the machine for reliable testing. The verification of the dose of the LEDs is done by studying the colour pattern. It turns out the dose is correctly radiated onto the seed bed. The machine has the potential to show the effectiveness of the UV-C light and future work can be interesting to execute.

References

[1] Nan Jiang et al. “Effects of ultraviolet-c treatment on growth and mycotoxin production by Alternaria strains isolated from tomato fruits”. In: *International Journal of Food Microbiology* 311 (2019), p. 108333. ISSN: 0168-1605. DOI: https://doi.org/10.1016/j.ijfoodmicro.2019.108333. URL: https://www.sciencedirect.com/science/article/pii/S0168160519302636.

[2] Dongqi Guo, Lixia Zhu, and Xujie Hou. “Combination of UV-C Treatment and Metschnikowia pulcherrimas for Controlling Alternaria Rot in Postharvest Winter Jujube Fruit”. In: *Journal of Food Science* 80.1 (2015), pp. M137–M141. DOI: https://doi.org/10.1111/1750-3841.12724. eprint: https://ift.onlinelibrary.wiley.com/doi/pdf/10.1111/1750-3841.12724. URL: https://ift.onlinelibrary.wiley.com/doi/abs/10.1111/1750-3841.12724.

[3] Lindsey du Toit. *Cabbage seed plated onto a semi-selective agar medium to detect Alternaria brassicicola.* https://mtvernon.wsu.edu/path_team/cabbage.htm#alternariablackspot. 2023.

[4] International Seed Health Association. “International Rules for Seed Testing 2022: Validated Seed Health Testing Methods”. In: 2023.Number 1 (2023), i-7–6(6).