Solar spectrum simulator design for assessing the efficiency of photovoltaic cells in the aerospace field

Auteur : Culot, Maxence
Promoteur(s) : Redouté, Jean-Michel
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil électricien, à finalité spécialisée en "electronic systems and devices"
Année académique : 2020-2021
URI/URL : http://hdl.handle.net/2268.2/11453

University of Liège - Faculty of Applied Sciences

GRADUATION STUDIES CONDUCTED FOR OBTAINING THE MASTER'S DEGREE IN ELECTRICAL ENGINEER BY CULOT MAXENCE

Solar spectrum simulator design for assessing the efficiency of photovoltaic cells in the aerospace field

Academic promoter:
J-M. REDOUTE

Author:
CULOT Maxence

PhD student:
H. PIERRE

Year 2021

Abstract

The purpose of this Master's thesis is to engineer a LED-based solar simulator. As the photovoltaic cells of CubeSats need to be assessed before launching it into space, a simulation tool is required to replicated the extraterrestrial solar spectrum.

The research started with a thorough analysis of the extraterrestrial solar spectrum which needs to be replicated. Once this solar spectrum have been characterized, a practical solution have been implemented. The next step was to research about the components that could be used to build up the simulator. It is mainly power LEDs with different wavelengths, LED drivers and power converters. Afterward, these components has been interfaced on a custom printed circuit board. Finally, the board have been programmed through a graphical user interface to have a complete monitoring of the output spectrum. The thesis report is an overview of these successive phases of the conception.

At the end, this simulation tool will be used in the aerospace field to assess that the photovoltaic cell harvesting is sufficient enough to power all the electronics of the satellite before making it ready for spatial missions.

Acknowledgements

The completion of this Master's thesis could not have been possible without the expertise of Hervé Pierre, my thesis advisor. He provides me with tools and guidelines from the beginning up to the end of this journey and I would like to thank him for his support. I would also thank Sir Jean-Michel Redouté for his acceptance of being my academic promoter, his interest in the researches I did and his valuable advice.

Last but not least, I would like to thank my family. Without them, none of this work would be possible.

Contents

List of Figures

List of Tables

1 Introduction

Before planning to launch a new satellite into space, one must ensure the effective operation of every critical part. These parts are tested individually and then are assembled together to constitute the satellite. The University of Liège is developing CubeSats for several years up to now and require simulation tools to test and validate their researches. FIGURE 1 shows a picture of *OUFTI-1*, the first Belgian CubeSat sent into space. More specifically, a CubeSat is featured with photovoltaic cells around its case to feed its electric power system (i.e. *EPS*). In this context, these photovoltaic cells need to be tested to guarantee they can harvest enough power from the Sun to supply all its electronic components.
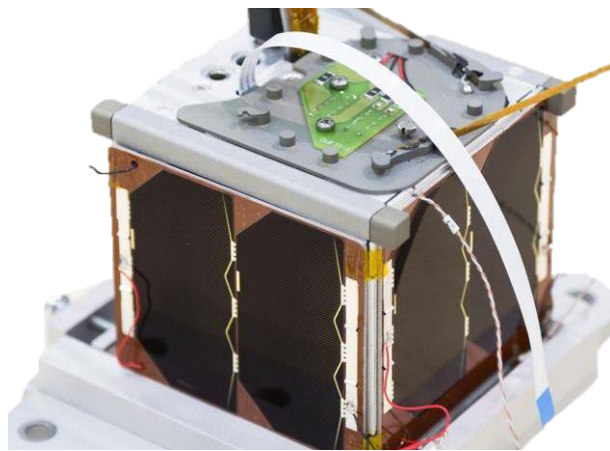
Figure 1 – *OUFTI-1* CubeSat [51].

The goal of this Master's thesis is thus to build a so-called "solar simulator" to replicate the solar power that the CubeSat will receive in space. For this purpose, the extraterrestrial solar spectrum will be replicated using various LEDs. These power LEDs have different wavelengths in order to cover as best as possible this solar spectrum. In addition, a graphical interface will be designed to control the intensity of the LEDs which allows us to modify the simulated spectrum. Finally, a test script will be written to match the simulation requirements. This script consists of cycles with sixty minutes ON time and thirty minutes OFF time.

This new simulator will determine whether or not the photovoltaic cells are able to harvest enough power from the Sun to supply the required operations and thus be able to be energy sufficient while doing spatial missions.

2 Theoretical solution

In this section, the theoretical concepts will be explained as what is the solar spectrum, which spectrum the simulator should replicate and how the solution is going to be implemented.

2.1 Solar spectrum on Earth's surface

The first thing that needs to be clarified is which solar spectrum will the solar simulator replicate. Indeed, the solar spectrum that the simulator will output should be as close as possible from the one the CubeSat will receive during its spatial missions. As the CubeSat obviously lies into space, the replicated spectrum should be the one outside terrestrial atmosphere. The CubeSat can not be simply exposed to the Sun received on Earth's surface to simulate whether or not the power harvesting is sufficient enough to feed the electric power systems. Indeed, the solar spectrum received on Earth is not the same as the one received into space. It is due to the scattering and absorption of Earth's atmosphere [38].

2.1.1 Scattering

Scattering is a physical phenomenon that describes the deviation of electromagnetic radiations due to the interactions with particles or gases in the atmosphere [FIGURE 2].
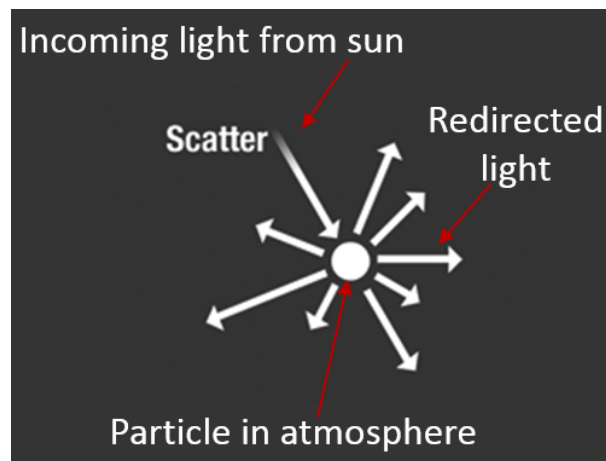
Figure 2 – Scattering phenomenon [53].

The light is redirected, meanings that a part of it is spread while another portion is reflected. How much scattering depends on several factors as the size of the particles, the wavelength of the incoming radiations, the atmosphere depth the radiations pass through, etc.

It exists three types of scattering according to the particle sizes:

- **Rayleigh scattering:** occurs when the particle diameter is smaller than the wavelength of the incoming radiations. These particles are usually NO_2 and O_2 and explain why shorter

wavelengths are more scattered than longer ones. The shorter wavelengths being the blue ones in the visible spectrum, the Rayleigh scattering is the physical phenomenon under the blue color appearance of the sky.

- **Mie scattering:** occurs when the particle diameter is similar to the radiation wavelength. It happens in lower portions of atmosphere where the particles are more abundant and bigger (dust, smoke, water vapor, ...). The phenomenon affects larger wavelengths.

- **Non-selective scattering:** occurs when the particle diameter is larger than the radiation wavelength. These particles are mostly water droplets and they scatter the radiations even outside the visible range. The phenomenon is responsible for the fog and cloud formation.

2.1.2   Absorption

The absorption of electromagnetic radiations through the atmosphere is mainly due to three elements/particles:

- **Water vapor:** strongly absorbs light between 5.5-7 μm and > 27 μm. This absorption by water droplets is highly dependant on the Earth area where the receiver is placed (e.g. desert vs. Amazonian forest).

- **Carbon dioxide** CO_2**:** It is well known as a greenhouse gas and absorbs a part of the light infrared spectrum.

- **Ozone** 0_3**:** The ozone absorbs a portion of ultraviolet radiations and protects our skin against cancers.

The combination of these two phenomena, scattering and absorption, leads to a power decrease between the solar spectrum outside atmosphere and the one on Earth's surface. FIGURE 3 illustrates this balance done at Earth's atmosphere. It explains why a solar simulator needs to be build to test real world conditions while assessing the photovoltaic cells of CubeSats.
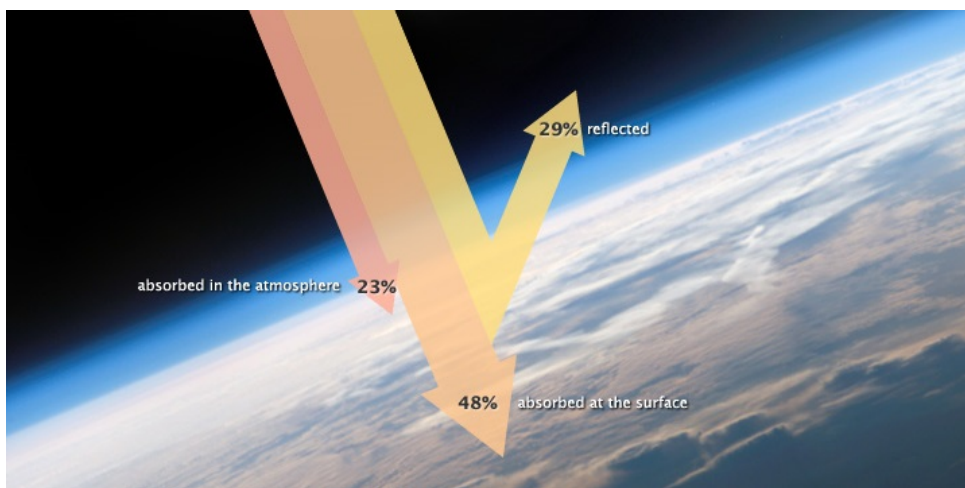
Figure 3 – Solar balance at Earth's Surface [43].

2.2 Solar spectrum on Sun's surface

The solar simulator has thus to replicate the exact solar spectrum outside atmosphere. On the first place, one need to define what is this spectrum at the Sun's surface. The Sun is an energy source that behaves roughly like a black body, i.e. it absorbs all electromagnetic radiations falling on it [8]. EQUATION 1 computes the black body spectral irradiance using the Plank's equation [9]:

$$F(\lambda) = \frac{2\pi hc^2}{\lambda^5 \left(\exp\left(\frac{hc}{k\lambda T} \right) - 1 \right)},$$ (1)

with

- $h = 6.62607015 \times 10^{-34} \left[\frac{J}{Hz} \right]$, the Plank constant

- $c = 3 \times 10^6 \left[\frac{km}{s} \right]$, the speed of light

- $k = 1.380649 \times 10^{-23} \left[\frac{J}{K} \right]$, the Boltzmann constant

- $T = 5778 \, [K]$, the black body temperature (i.e. the surface of the Sun)

- $\lambda \, [\mu m]$, the wavelength

- $F(\lambda) \left[\frac{W}{m^2 \mu m^{-1}} \right]$, the spectral irradiance

Solving this Plank's radiation law with the following parameters for the Sun power source gives the distribution seen on FIGURE 4.
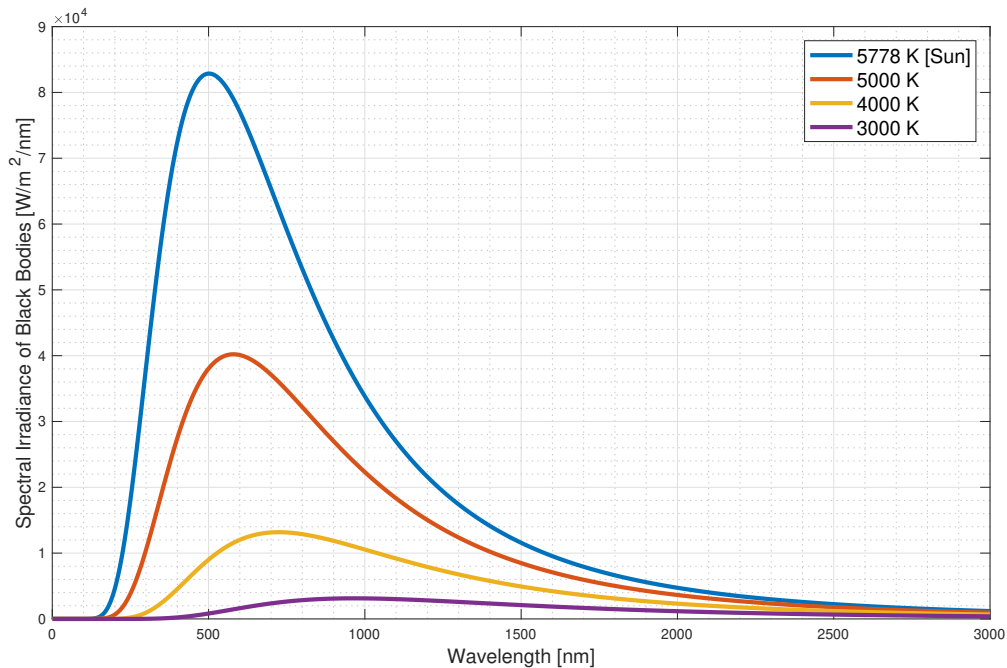
Figure 4 – The black body radiation as function of temperature.

These curves represent the irradiance of black bodies at given temperatures. As it can be seen from this figure, the higher the black body temperature, the lower the wavelength at the irradiance peak. This shift in the peak towards lower wavelengths when the temperature increases is characterized by the Wien's displacement law. EQUATION 2 characterizes this relationship [54]:

$$\lambda_{\text{peak}} = \frac{b}{T}, \tag{2}$$

with

- $b = 2.897771955 \times 10^{-3}$ $[\text{m K}]$, the Wien's displacement constant

- T $[\text{K}]$, the black body temperature

- λ_{peak} $[\text{m}]$, the peak wavelength

This equation is obtained by deriving the Plank's equation according to the wavelength [cf. EQUATION 1]. For the Sun, being approximated as a black body with a surface temperature of 5778 K, EQUATION 3 computes the wavelength where the irradiance is maximal:

$$\lambda_{\text{peak @5778K}} = 501.52 \, \text{nm}. \tag{3}$$

EQUATION 4 computes the overall power density obtained by integrating the Plank's equation [44]:

$$H = \sigma T^4 = 6.32 \times 10^7 \ \text{W/m}^2, \tag{4}$$

with $\sigma = 5.67 \times 10^{-8}$ $\text{W/m}^2\text{K}^4$ being the Stefan-Boltzmann constant. Multiplying the solar power density by the surface of the Sun gives the total power it emits [EQUATION 5]:

$$H_{\text{tot}} = H S_{\text{sun}} = 4\pi H R_{\text{sun}}^2 = 3.86 \times 10^{26} \ \text{W}, \tag{5}$$

with $R_{\text{sun}} = 696\,340$ km being the sun radius.

2.3   Air mass concept

Once the solar power density has been well defined, the electromagnetic radiations have still to reach the surrounding of the Earth. The overall power is thus attenuated due to absorption and scattering by space debris and spread out over a distance as it can be seen on FIGURE 5.
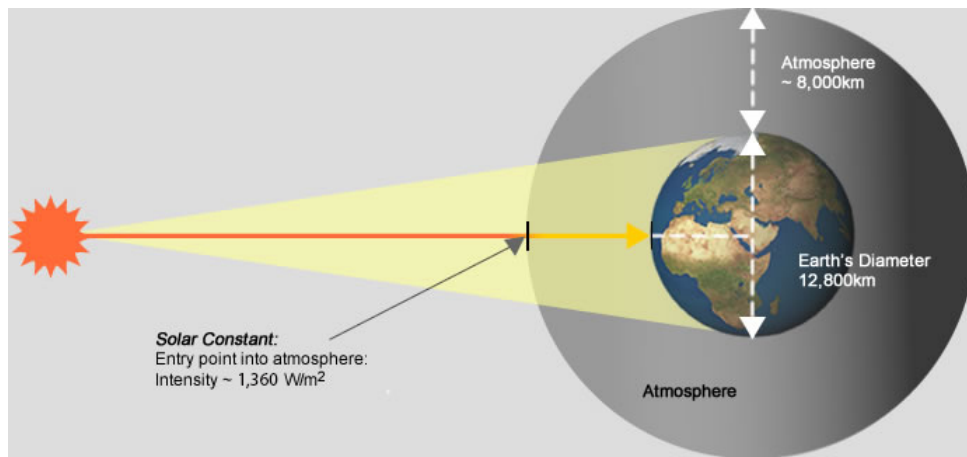
Figure 5 – Projection of the solar irradiance on the Earth's atmosphere and surface [18].

The distance between the Sun and the Earth is not constant and depends on the time of the year. Indeed, as the Earth follows an elliptical orbit, this distance varies from 147.1 millions to 152.1 millions kilometers. FIGURE 6 illustrates the Earth's orbital path.
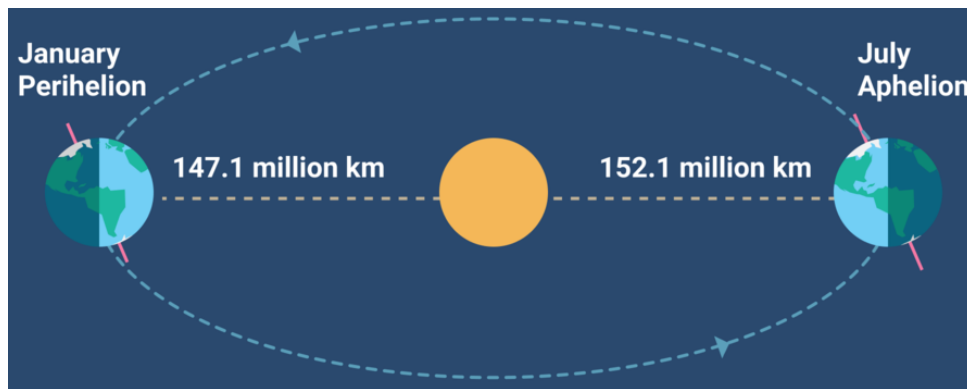
Figure 6 – Earth's orbit around the Sun [24].

To compute the spectrum irradiance outside Earth's atmosphere, the Astronomical unit will be used [7]. This unit is equal to the average distance between Earth and the Sun, defined as 149.6 millions kilometers. EQUATION 6 shows the computation of the spectrum projection taking into account this distance.

$$H_{\text{AM0}} = H \frac{R_{\text{sun}}^2}{D_{\text{sun-earth}}^2} = 1369.3 \text{ W/m}^2. \tag{6}$$

This irradiance is called *AM0* which stands for no air mass between the Sun and the receiver (i.e. outside atmosphere) [1]. This *AM0* spectrum is also known as the extraterrestrial solar spectrum and will be used as reference to test the photovoltaic cells into space. In comparison, the *AM1.5* spectrum represents the spectrum at the Earth's surface. The AM_x spectra are normalized from standard *ASTM G173-03* and characterize the standard spectrum distributions [FIGURE 7].
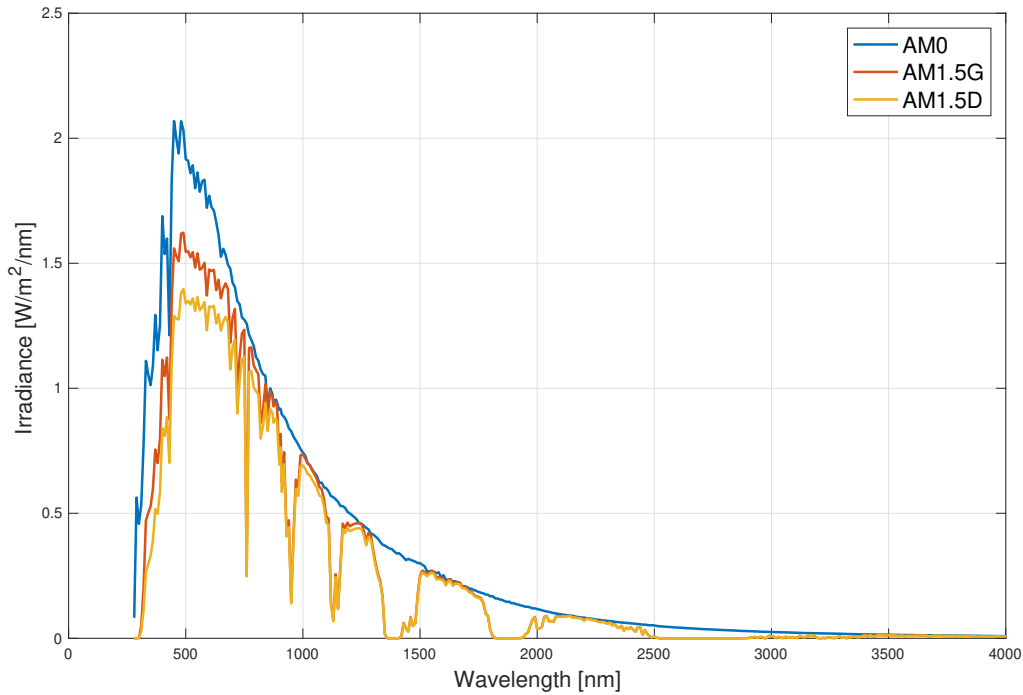
Figure 7 – Standard $AM0$ and $AM1.5$ spectra [5].

The *AM1.5G* represents the global standard spectrum at Earth's surface with direct and diffuse radiations while the *AM1.5D* takes into account only direct radiations. The *AM1.5D* is reduced for 28% compared to the extraterrestrial spectrum. The account of diffuse radiation increases the spectrum by the 10% and constitutes the *AM1.5G* spectrum. By comparing the terrestrial spectrum with the extraterrestrial one, the drops in the spectrum are due to the absorption by atmosphere molecules like O_3, H_2O, CO_2, etc. Integrating the *AM0* spectrum gives the power density of the extraterrestrial spectrum [EQUATION 7]:

$$H_{\text{AM0}} = \int_0^{4000} H(\lambda)d\lambda = 1350.2 \text{ W/m}^2. \tag{7}$$

This value is consistent with what have been found on EQUATION 6. For the solar simulator design, it is this *AM0* spectrum which will be used as reference. The goal is thus to match as best as possible the *AM0* irradiance according to the wavelength of the radiations.

2.4 LED model

In order to replicate this *AM0* spectral density, power LEDs will be used. Indeed, LEDs can be considered as emitters in a narrow band around their peak wavelengths. FIGURE 8 shows the power spectral density of some *Lumileds®* LEDs.

Figure 8 – Typical normalized power vs. wavelength for *Lumileds*® LUXEON Rebel Color line series at 350mA [26].

A simple model but yet accurate to characterize the LED power spectral density (i.e. *PSD*) is a Gaussian model [2, 50]. The distribution follows the normal distribution [EQUATION 8]:

$$f(\lambda) = ae^{\frac{-(\lambda-\lambda_0)^2}{2\sigma^2}}, \tag{8}$$

with:

- a, the radiometric power

- λ_0, the peak wavelength

- σ, the standard deviation

To build the *AM0* spectrum, the power spectral densities of the LEDs will be summed according to their radiometric power. For this purpose, various LEDs with diverse wavelengths will be used to have the highest spectral matching possible. Each of these parameters lies in the LED documentation.

2.4.1   Radiometric power

By looking at some LED datasheets, the power is given either in radiometric units or in lumens. When the power is expressed in lumens, it needs to be converted into watts to compute the LED PSD. Lumens are units measuring a luminous flux emitted by a source with respect to the

visible light that humans see. The luminous flux is thus weighted according to the human eye sensitivity. EQUATION 9 shows how to retrieve the radiometric power from the luminous flux in lumens [10, 36]:

$$\Phi_e[\lambda] = \int_0^{4000} \frac{\Phi_v[\lambda]}{K_m V(\lambda)} d\lambda, \tag{9}$$

where

- $\Phi_e[\lambda]$ is the radiometric power

- $\Phi_V[\lambda]$ is the power expressed in lumens

- $K_m = 683$ lm/W is the maximal spectral luminous efficiency at *555* nm

- $V(\lambda)$ is the photopic luminous efficiency function

The plot of this luminous efficiency function can be found on FIGURE 9. It represents the human eye sensitivity and reaches a peak at *555* nm which corresponds to a green color.
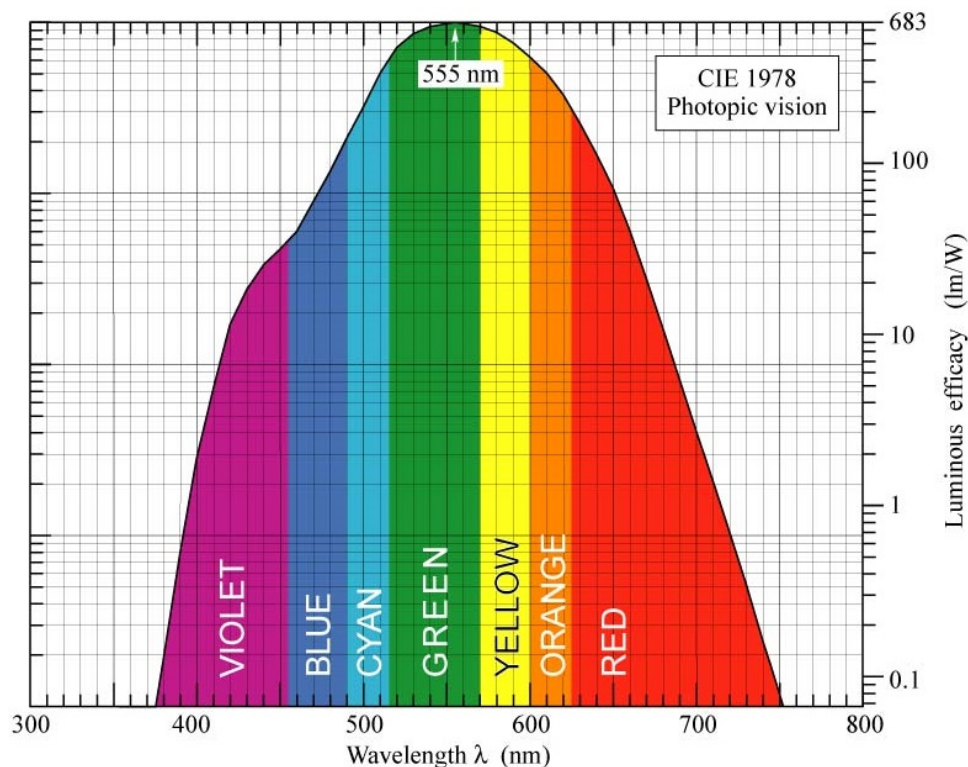
Figure 9 – Photopic sensitivity function [31].

Converting all the powers in watts allows us to sum up the power spectral densities as they all have the same units.

2.4.2   Peak wavelength

The peak wavelength is the wavelength where the maximum of the PSD lies as contrary to the dominant wavelength which represents the main wavelength that a human eye will see.

2.4.3   Standard deviation

The standard deviation of the LED PSD is characterized through the full width at half maximum (i.e. FWHM). It represents the difference between the two values of the spectrum at half the maximum. EQUATION 10 computes the standard deviation from the FWHM [57]:

$$\sigma = \frac{\text{FWHM}}{2\sqrt{2\ln 2}}. \tag{10}$$

2.4.4   Spectral irradiance

To achieve a spectral density (i.e. W/m^2), the radiometric power needs to be radiated on a given surface. The surface is a spherical cap and is linked to the solid angle definition [FIGURE 10]:
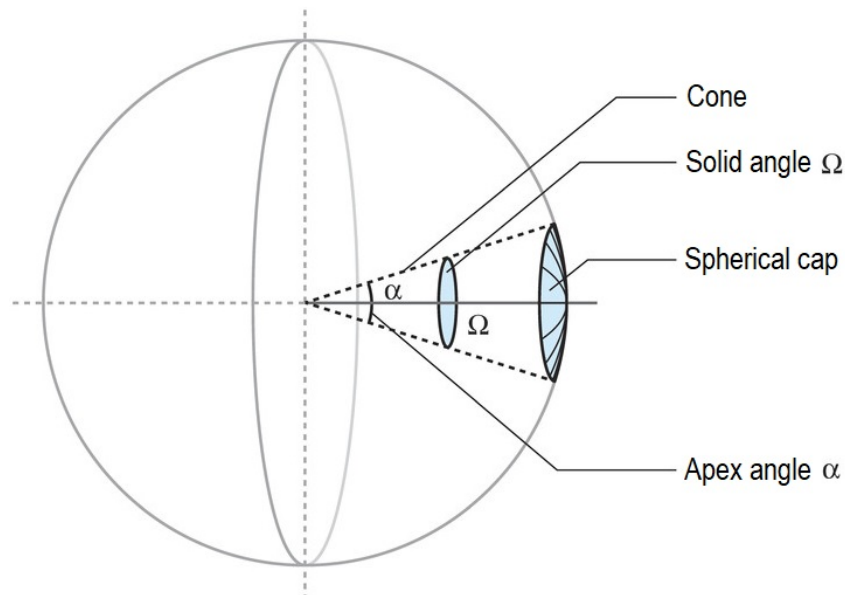
Figure 10 – Spherical cap induced by a solid angle Ω [6].

EQUATION 11 computes the spherical cap surface [40]:

$$S = \Omega R^2 = 2\pi \left(1 - \cos {}^{\alpha}\!/\!{}_{2}\right) R^2, \tag{11}$$

where:

- α is the apex angle of the LED, also known as the viewing angle

- R is the distance between the LED and the surface to irradiate. In this case, this distance will be between the LED and the photovoltaic cells

- S is the projected surface. This surface should be larger than the photovoltaic cells of the CubeSat and was fixed to be at least equal to $30x30$ cm^2

The typical viewing angle of the LEDs used in this solar simulator is about *120°*. For this reason, the luminous flux is spread over a wide surface if the distance from the LEDs to the photovoltaic cells is to high. The higher the distance, the lower the power density. Indeed, if one compare the spherical caps between two distances respectively of *20* cm and *1* m with an apex angle of *120°*, the surfaces are equal to [EQUATION 12]:

$$S_{@20\,cm} \approx 60x60 \text{ cm}^2 \qquad \text{and} \qquad S_{@1\,m} \approx 300x300 \text{ cm}^2. \tag{12}$$

To achieve a suitable power density without using thousands of LEDs, one need to keep this surface reasonable. This was an iterative process but, after simulating, a surface of $60x60$ cm^2 was chosen by placing the simulator at 20 cm long from the photovoltaic cells. Another studied option was to keep the same surface goal but to extend the distance to 1 m. To keep the same power density, some lenses could be used. These lenses, placed on top of the LEDs, channel the light beams on a smaller apex angle. This solution was not kept as there were issues for the simulation and later in the PCB design. Indeed, the lenses have an efficiency below 90% which causes losses in the power density. Furthermore, this efficiency is hard to model as it is wavelength dependant and that datasheets do not give all the details about this dependence. Finally, placing lenses on top of the LEDs required some room around it while designing the PCB and would have lead to an useless wider version of it. For all these reasons, it was chosen to reduce the distance between the simulator and the photovoltaic cells instead of using lenses.

2.4.5   Assumptions and model limitations

The theoretical model build here above contains all elements to start the component research and to start simulations about what the power density should look like. However, there are several assumptions that were took to build it and limitations as well to be aware of.

First, the Gaussian model is a pretty simple model but it does not match at *100%* the LED PSD. Indeed, the PSD is not always symmetric from the peak wavelength. This asymmetry is measured by the skewness factor and is not always indicated on datasheets [46]. As it can be seen on FIGURE 11, the PSD is not symmetric and it exists secondary peaks that are not considered in the Gaussian model.
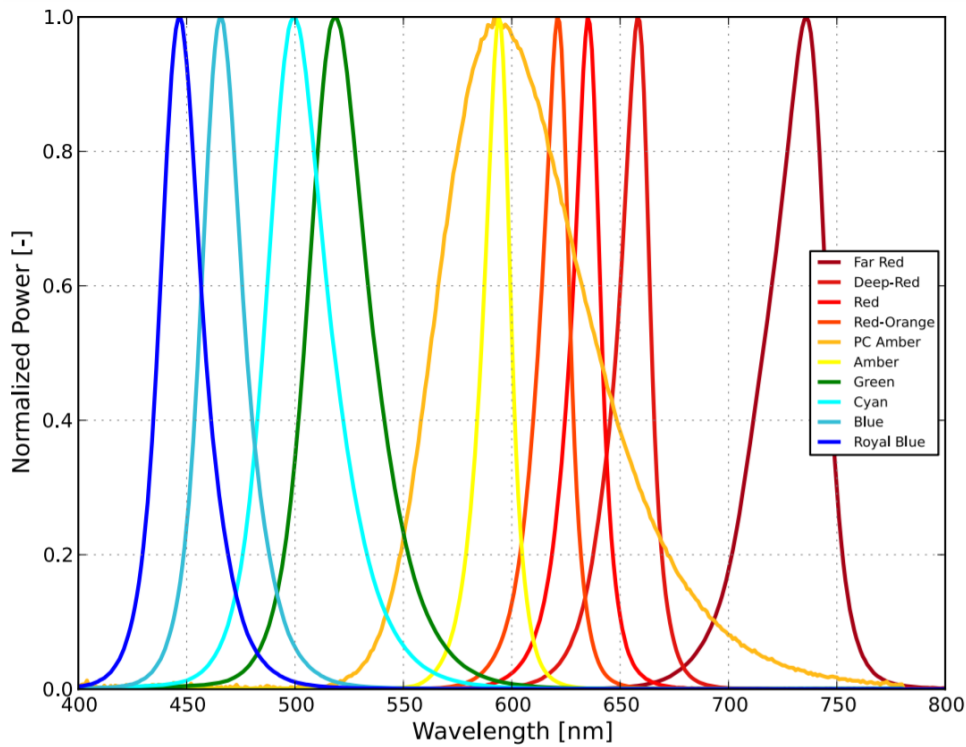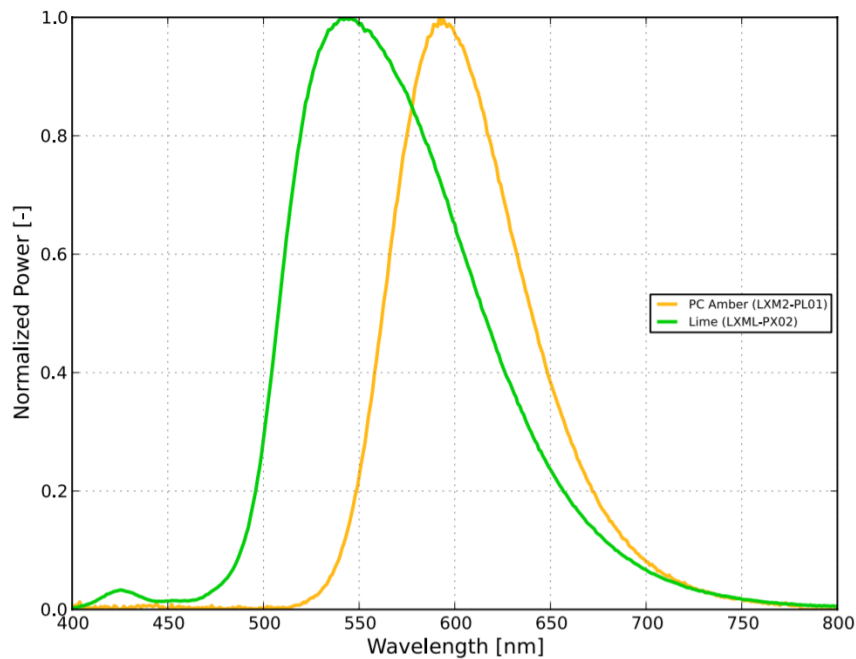
Figure 11 – Typical normalized power vs. wavelength for *Lumileds*® LUXEON Rebel color line series at 350mA.

Second, there is unpredictable variation of the peak wavelength depending on the LED manufacturing. Datasheets provide us with minimal and maximal margins but if the peak is shifted by *20* nm, the overall PSD is strongly affected. This problem can not be solved and will be handled during the testing phase [cf. SECTION 7].

Third, for the LED power conversion from lumens to watts, an unique value has been used according to the peak wavelength of the LED. Indeed, to have an easier and faster way to compute the power spectral densities in watts, the simulation does not take into account a convolution between the LED spectrum and the luminous efficiency function. Instead, the power is computed by dividing the power in lumens by the luminous efficiency value at the peak wavelength of the LED. This approximation is valid till the delta lambda of the LED spectrum is small compared to the delta lambda of the luminous efficiency function. If this assumption is valid, one can consider the luminous efficiency being constant over the whole LED spectrum.

Finally, the surface over which the LEDs irradiate power is not a plane but a spherical cap. As the photovoltaic cells are plane cells, the real irradiated surface is the intersection between a plane and the spherical cap. Considering the spherical cap being much larger than the photovoltaic cell plane, the spherical cap was assumed to be similar to a plane with the same surface. Moreover, to compute this surface, the LEDs were assumed to be stacked in an infinitesimal point in space which constitutes the sphere center. However, the LEDs are much wider than a simple point in space and they take some space on a PCB. To build a more accurate model, the

LEDs placed on the PCB can be surrounded by a circle of diameter 6 cm. Taking into account the same *120°* apex angle, the center point of the sphere is located $\approx$ 5 cm behind the PCB plane [EQUATION 13]:

$$D_{\text{circle - central point}} = \sqrt{\frac{S_{\text{circle}}}{\Omega}} = 5.12 \text{ cm.} \tag{13}$$

In this context, the effective distance between the LEDs and the photovoltaic cells is not 20 cm but 25 cm. EQUATION 14 computes the surface using a distance of 25 cm:

$$S_{@25\,\text{cm}} \approx 76\text{x}76 \text{ cm}^2. \tag{14}$$

This surface is wider than the original 60x60cm^2 and thus the overall power density decreases. However, this computation could not be performed before manufacturing the PCB as there was not possible way to figure out which space the LEDs will take on the PCB.

The validity of these assumptions and shortcuts in the theoretical solution will be discussed in the SECTION 7.

3 Component selection

Once the spectrum goal has been clearly established, the components required to build the simulator have to be gathered. This section is dedicated to the explanations of the component selection.

3.1 Single color power LEDs

To replicate the extraterrestrial solar spectrum, power LEDs will be used and characterized according to the model detailed here above [cf. SECTION 2]. The goal is to use several LEDs with diverse wavelengths to recreate the shape of this spectrum. To assess the relevance of the researched LEDs, a *Matlab®* program has been build which simulates the power spectral density of the LED according to its main parameters.

First of all, there is a main characteristic that needs to be defined before searching LEDs. This characteristic is the LED typical forward current. The irradiated power emitted by the LED is directly related to this current. Indeed, taking small 20 mA LEDs is not suitable as it will required hundreds of these per wavelength to achieve the *AM0* spectrum irradiance. At the opposite, taking 1000 mA LEDs means that there will be only one LED per wavelength as each LED emits a lot of radiative power and thus the spectrum will be less uniform. The ideal solution is thus to use LEDs with typical forward current in the range from 100 mA to 500 mA. In such a way, to achieve the spectral matching, the number of LEDs needed remains manageable and there are several LEDs per wavelength to have a better spatial uniformity.

As an example, lets compute the simulation results for a specific LED. This LED will be the *LXML-PX002-0000* from the *Lumileds® Rebel Color line* series. It is a green LED with a typical forward current of 350 mA. This LED has a peak wavelength of 567 nm and a luminous power of 184 lm at 350 mA. Knowing these characteristics, EQUATIONS 15, 16 and 17 show how to compute the irradiance for such a LED:

$$P_{\mathrm{W}} = \frac{P_{\mathrm{lm}}}{\eta_{\text{luminous fct}}}, \quad \text{with} \quad \eta = V(\lambda_{\text{peak}})K_{\mathrm{m}} = 0.97 K_{\mathrm{m}}. \tag{15}$$

$$P_{\mathrm{W/m^2}} = \frac{P_{\mathrm{W}}}{Surface}, \quad \text{with} \quad Surface = 2\pi R^2(1 - \cos \alpha/2). \tag{16}$$

$$Irradiance = P_{\mathrm{W/m^2}}(\lambda) = P_{\mathrm{W/m^2}} e^{\frac{-(\lambda - \lambda_{\text{peak}})^2}{2\sigma^2}}, \quad \text{with} \quad \sigma = \frac{\text{FWHM}}{2\sqrt{2\ln 2}}. \tag{17}$$

FIGURE 12 shows the irradiance of the LED computed here above.
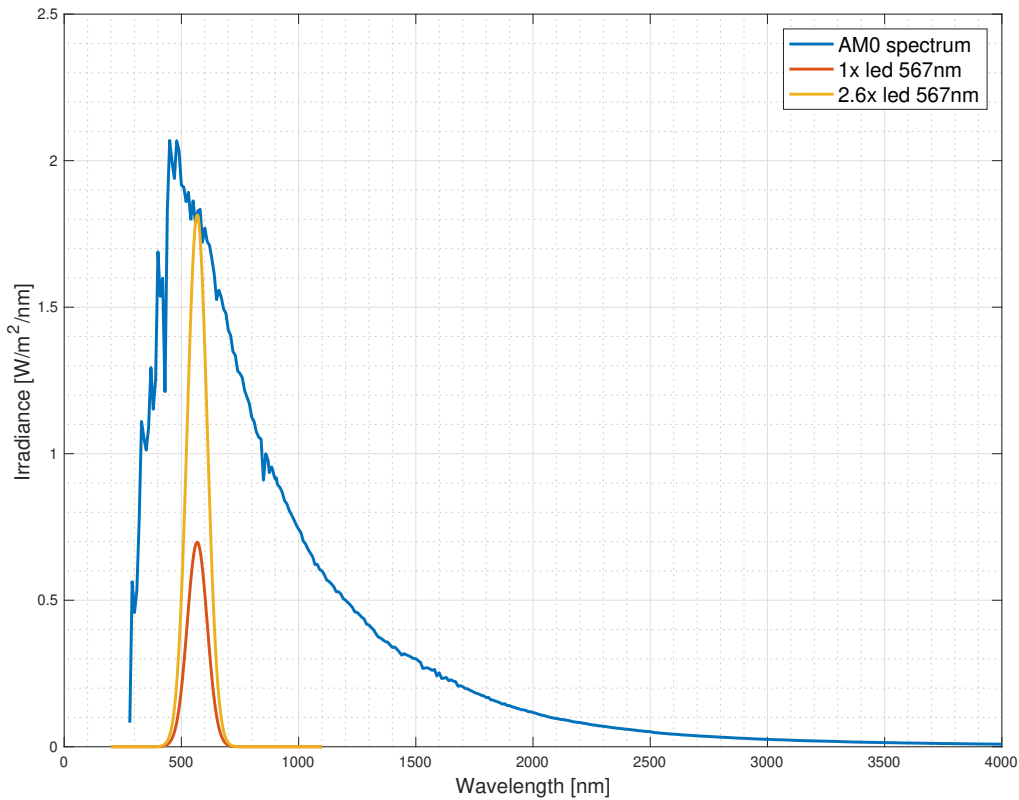
Figure 12 – LED irradiance compared to the $AM0$ spectrum.

As it can be seen, the goal is to bring the LED irradiance as close as possible from the *AM0* spectrum irradiance. This is achieved either by adding several LEDs or by controlling the forward current flowing through these LEDs. For the first case, if the number of LEDs is double, the irradiance is doubled as well as it is directly proportional to the LED radiative power. For the second case, the yellow curve in the figure is the irradiance of *2.6* LEDs. This floating point number is achieved by controlling the current flowing through the LEDs. Indeed, a *2.6* LEDs is equivalent as using three LEDs with a current being *14%* lower than the typical 350 mA current. EQUATION 18 shows the effective current in the three LEDs to achieve this floating point number:

$$I_{\text{each LED}} = \frac{n_{\text{LEDs computation}} I_{\text{typical}}}{n_{\text{LEDs}}} = \frac{2.6 \times 350 \text{ mA}}{3} = 303 \text{ mA}. \tag{18}$$

This linear relationship is only a first order approximation assuming the radiative power is directly proportional to the forward current. The exact ratio between the current and the radiative power is given in the datasheet of each LED [21, 28, 29, 30, 39, 55, 56, 58]. FIGURE 13 shows this relationship for some *Lumileds® Rebel Color line* LEDs.
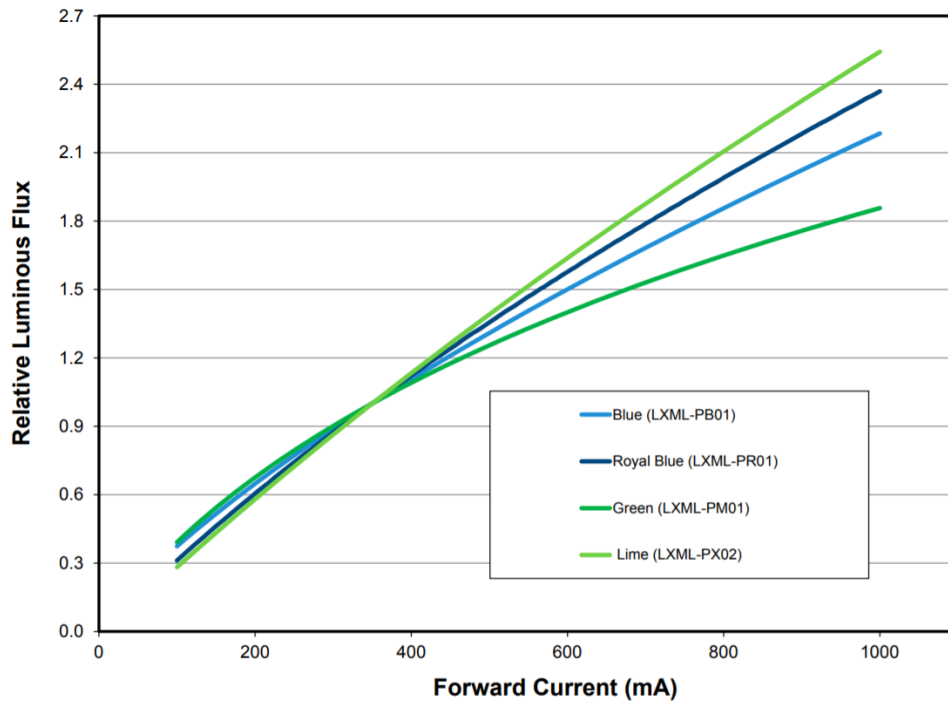
Figure 13 – Relative luminous flux vs. forward current for LXML-PX02, LXML-PM01, LXML-PB01 and LXML-PR01 at test temperature.

As it can be seen, this linear relationship is a good approximation if the current value remains around the typical forward current of the LEDs. The higher the variation between the typical and the driven current, the lower the ration between the relative luminous flux and the real one. As the LEDs are used close to their normalized current, the linear approximation is valid and will be used for all the computations.

Another solution would be to use less LEDs with a higher current (i.e. > 350 mA). However, it is not possible and this limitation is due to the LED drivers used [cf. SECTION 3.2]. These drivers can not output more than 300 mA per channel [15]. This leads to the use of more LEDs driven with lower currents than less LEDs driven with higher currents. There are thus more LEDs required to generate the *AM0* spectrum.

This process is repeated for several LEDs with diverse wavelengths until the simulator irradiance matches the *AM0* spectrum. In this project, up to *24* different wavelengths were used to match as best as possible the extraterrestrial spectrum irradiance. The theoretical spectrum that should be achieved with this solar simulator is shown on FIGURE 14.
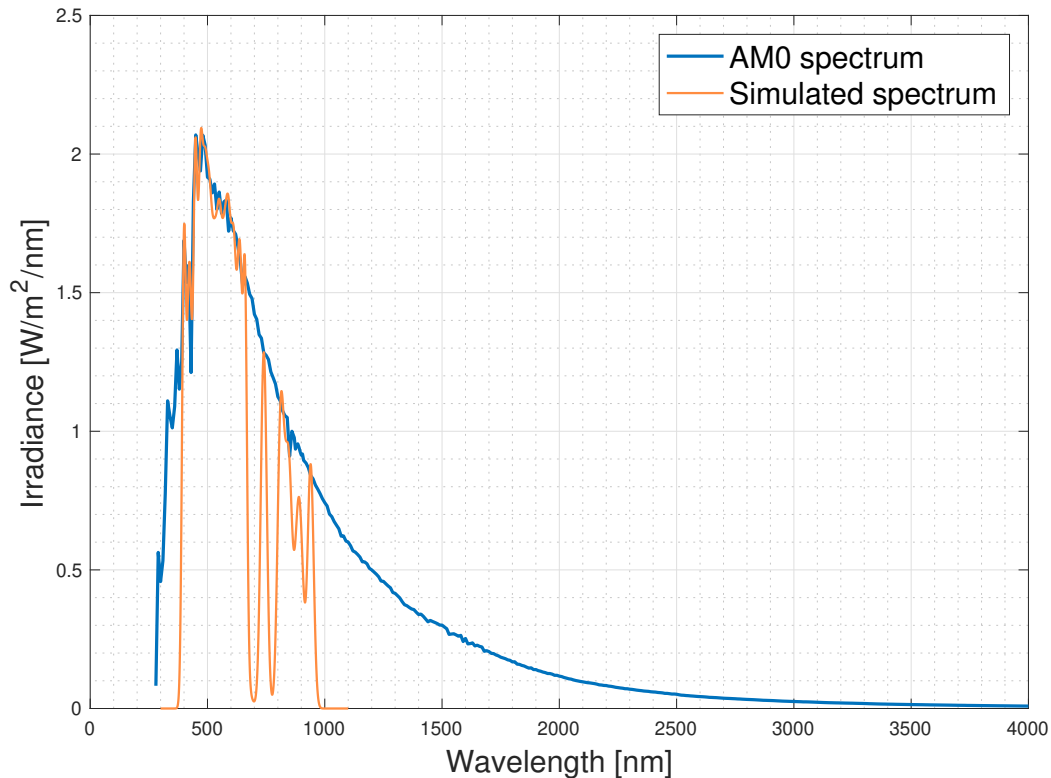
Figure 14 – Solar simulator spectrum compared to the $AM0$ spectrum.

As it can be seen, there is a pretty exact matching in the range between 390 nm and 700 nm. Up to 940 nm, there are some LEDs in the infrared range but not enough of them to cover the whole spectrum. Indeed, one are at the border between the deep red color and the infrareds used for telemetry applications. In this context, there are wavelengths at which there is not LED available. This explains why there are some gaps in the simulated spectrum. After 1000 nm, there is no more LED available to cover the spectrum and thus this region will not be tested in our simulations. At the opposite, going below 390 nm brings us in the UV-range. The solar simulator will not convert this range as the LEDs emitting at such wavelengths are really expensive (*UVA* and *UVB* LEDs). The budget being limited, the simulator will not be featured with LEDs covering this region of the spectrum.

TABLE 1 shows the LEDs used, their number as well as their wavelengths:

N°	Manufacturer	Part number	Wavelength [nm]	Number of LEDs
1	**Lumileds**	LXML-PR01-0500	447.5	3
2	**Lumileds**	LXML-PB01-0040	470	1
3	**Lumileds**	LXML-PE01-0080	505	3
4	**Lumileds**	LXML-PM01-0100	530	3
5	**Lumileds**	LXML-PX02-0000	567.5	4
6	**Lumileds**	LXM5-PL01	590	1
7	**Lumileds**	LXM2-PL01-0000	599	2
8	**Lumileds**	LXM2-PH01-0060	610	1
9	**Lumileds**	LXM2-PD01-0060	620	1
10	**Lumileds**	LXM3-PD01	660	3
11	**Wurth Elektronik**	150141SS63140	640	4
12	**Lumileds**	L128-RED1003500000	630	1
13	**Lumileds**	L128-FRD1003500000	740	4
14	**Luminus Devices**	SST-10-IR-B130-K850	850	2
15	**Luminus Devices**	SST-10-IR-B90-H940	940	3
16	**Vishay**	VSMF3710-GS18	890	6
17	**Lumileds**	L1CU-BLU1000000000	485	3
18	**Inolux**	IN-3531SCUV-U50	400	3
19	**Inolux**	IN-3531SCUV-U70	420	2
20	**Lumileds**	L1CU-VLT1000000000	430	2
21	**Lumileds**	L1CU-RYL1000000000	460	1
22	**Osram**	LCG H9RM-KZLZ-1-Z	550	1
23	**Luminus Devices**	SST-10-IRD-B130-S810	810	2
24	**Vishay**	VSMG2700-GS08	830	6

Table 1 – Characteristics of the LEDs used in the solar simulator implementation.

These LEDs are chosen as they gather several key characteristics:

- They are available on suppliers where the University of Liège can buy the components. For this project, the two main suppliers used are *Mouser®* and *Farnell®*.

- There typical forward currents lie in a range from 100 mA to 500 mA.

- Their datasheets are complete and contain all required information. This feature might seems silly but it is really substantial to know the LED features and specifications to achieve the best simulation results.

- They are packed into small packages to be compact on the PCB.

- If possible, the LEDs are chosen as much as possible in the same manufacturer and family. It eases the computation, the researches and later the routing of the PCB as the footprints are identical.

For the testing phase, one will refer to the spectrum build with these LEDs to see if the practical simulator follows what have be computed theoretically.

3.2   LED Drivers

In order to control the current in the LEDs, drivers are used. The goal of these drivers is to control the current flowing through the LEDs such that their radiative powers, related to the forward currents, match the extraterrestrial solar spectrum. Moreover, having to ability to adjust the currents lets us some room for modifications and adaptations. Indeed, if the output spectrum does not match the expected one, one can adjust the currents in the LEDs to modify it. This feature is really valuable as updating the spectrum according to various parameters like the distance between the simulator and the photovoltaic cells, theoretical approximations, temperature variations, etc allows an almost perfect matching at any time.

For this simulator, a suitable driver would be a driver which can output a current above 200 mA and which is controllable through a PWM interface. Moreover, to reduce the number of components on the board and thus to decrease its size, several channel drivers are more appropriate. The driver that incorporates all these features is the *AL1783* from *Diodes Incorporated®*. A picture of this driver can be seen on FIGURE 15.

Figure 15 – *AL173* driver from *Diodes Incorporated®*.

This is a three-channel linear driver, each channel being featured with an individual PWM dimming input. It can be supplied with up to 60 V and each channel outputs a maximum current of 300 mA. However, by connecting some channels in parallel, the output current can be increased at the price of reducing the number of channels available. As an example, connecting two chan-

nels in parallel can output a current of 2×300 mA. FIGURE 16 shows a typical application of this driver.
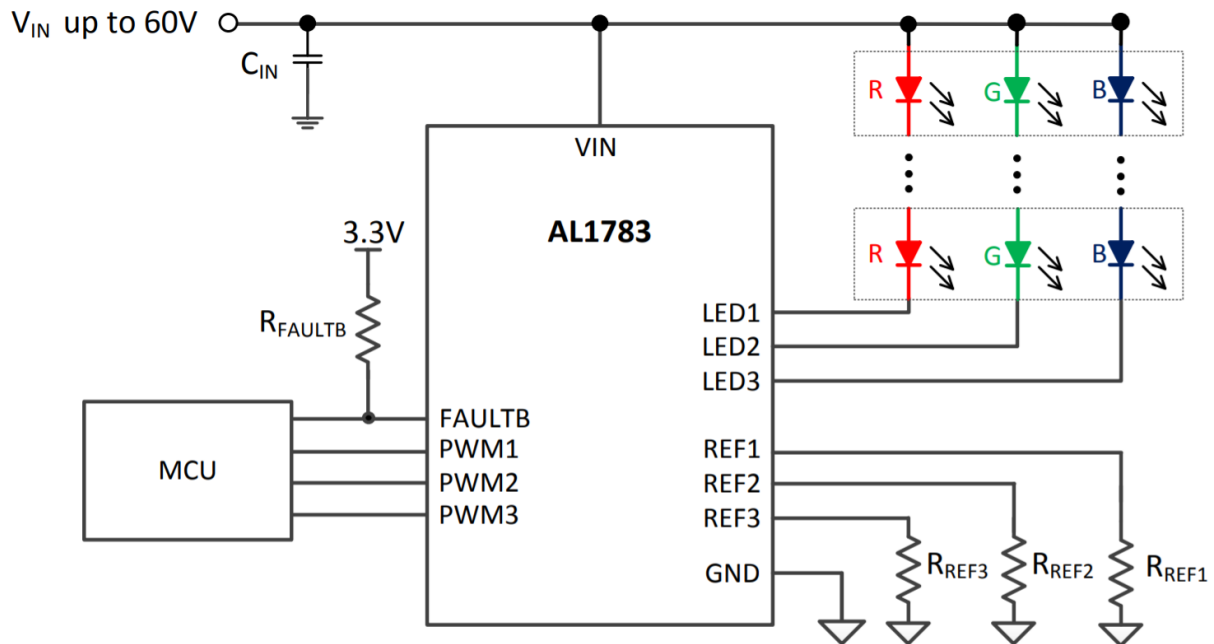
Figure 16 – Typical application scheme of the *AL1783* driver.

This schematics shows the driver controlling three LEDs in current. To adapt this driver for our situation, one need to analyze and understand what the PWM inputs, the reference inputs, the input voltage as well as the LED output pins do.

3.2.1 PWM inputs

The PWM inputs are used to control the current. Indeed, the current can be adjusted from less than 1% up to 100% of its maximal value. This control is done through the duty cycle of the PWM signal at each input and is represented on FIGURE 17.
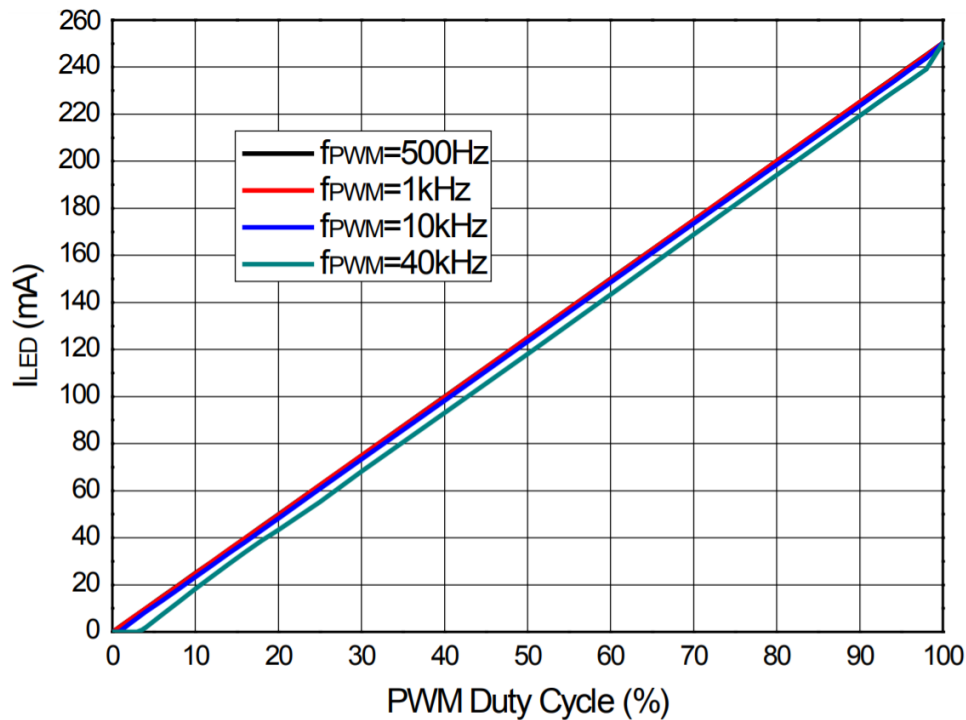
Figure 17 – LED Current vs. PWM Duty Cycle ($R_{\text{REF}} = 24 \text{ k}\Omega$, $I_{\text{LED}} = 250 \text{ mA}$, $0{\sim}100\%$ Duty).

This figure shows the relationship between the current in the LED and the duty cycle of the PWM signal. As it can be seen, when the duty cycle equals 100%, the current in the LED is equal to its maximal value [cf. SECTION 3.2.2]. When the duty cycle drops, the current in the LED also drops with the same ratio (i.e. linear relationship). Regarding the frequency of the PWM signal, the three 500 Hz, 1 kHz and 10 kHz curves can be considered as identical but there is a change in the relationship when a PWM frequency greater than 10 kHz is used. However, this do not impact our computation as the PWM frequency used will be set around 1 kHz [cf. SECTION 3.3].

By controlling this duty cycle, one control the forward current in the LED and are thus able to adjust the radiative power such that it matches our expectations.

3.2.2   Reference inputs

As explained before, the PWM duty cycle can adjust the current between 0 and 100% of its maximal value. The reference inputs are used to set the maximal current that can flow through the LED when the duty cycle is at 100%. These current values are set through external resistors. FIGURE 18 shows the relationship between the resistor value and the maximal current available.
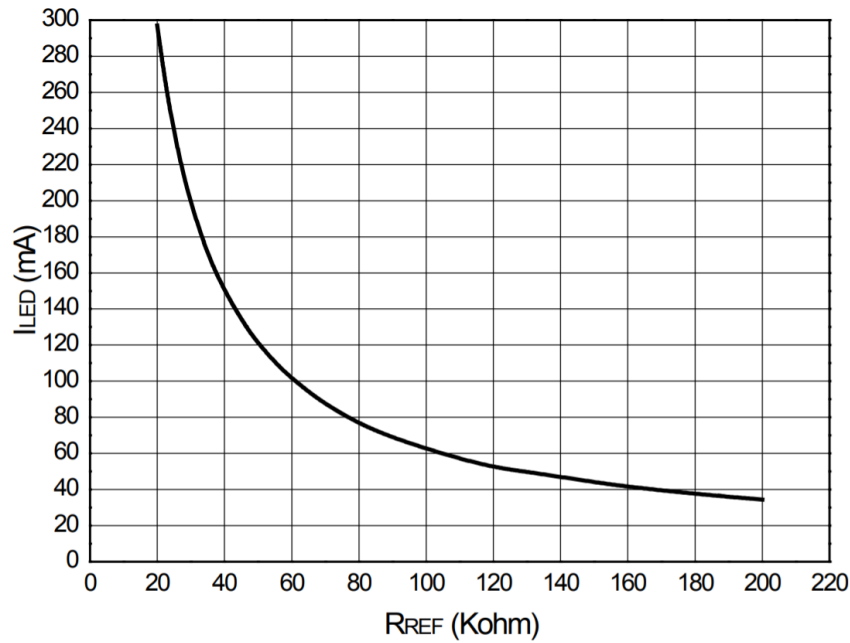
Figure 18 – LED Current vs. R_{REF}.

As it can be seen, choosing the appropriate resistor value sets the maximal current between 30 mA and 300 mA. As the ratings do not recommend to go over 250 mA, one decided to do not exceed this value which corresponds to 24 kΩ resistors. The maximal current value is chosen individually for each LED according to the two conditions of EQUATION 19:

$$I_{\mathrm{LED}} \leqslant 250 \text{ mA} \quad \text{and} \quad I_{\mathrm{LED}} \leqslant I_{\mathrm{max\ LED}}. \tag{19}$$

Indeed, each LED has a recommended maximum current to do not exceed. For security purpose, the resistor values will be chosen always such that the current flowing through the LEDs is below this maximum rating. TABLE 2 shows the resistor values as well as the maximum current in the channel according to these values.

N°	Part number	$I_{typical}$ [mA]	I_{max} [mA]	Resistor [$k\Omega$]	$I_{channel}$ [mA]
1	LXML-PR01-0500	350	1000	24	250
2	LXML-PB01-0040	350	1000	24	250
3	LXML-PE01-0080	350	1000	24	250
4	LXML-PM01-0100	350	1000	24	250
5	LXML-PX02-0000	350	1000	24	250
6	LXM5-PL01	350	700	24	250
7	LXM2-PL01-0000	350	700	24	250
8	LXM2-PH01-0060	350	700	24	250
9	LXM2-PD01-0060	350	700	24	250
10	LXM3-PD01	350	700	24	250
11	150141SS63140	50	100	75	75
12	L128-RED1003500000	200	300	24	250
13	L128-FRD1003500000	200	350	24	250
14	SST-10-IR-B130-K850	350	350	24	250
15	SST-10-IR-B90-H940	350	350	24	250
16	VSMF3710-GS18	100	200	40	150
17	L1CU-BLU1000000000	350	1300	24	250
18	IN-3531SCUV-U50	350	800	24	250
19	IN-3531SCUV-U70	350	800	24	250
20	L1CU-VLT1000000000	350	1500	24	250
21	L1CU-RYL1000000000	350	1300	24	250
22	LCG H9RM-KZLZ-1-Z	350	500	24	250
23	SST-10-IRD-B130-S810	350	1500	24	250
24	VSMG2700-GS08	100	200	40	150

Table 2 – Current ratings and resistor values for each LED.

3.2.3   Resistors

The resistors used to set the current need to remain stable regarding temperature variations. Indeed, as the LEDs will be ON, they will dissipate heat [cf. SECTION 5.1]. This heat will warm up the PCB and will increase the temperature of surrounding components. As the resistors are temperature dependent (either positive or negative), their values will change which has an impact on the LED currents. In summary, if these currents are not stable, the radiative powers of the LEDs will change according to the PCB temperature. It means that the simulator will not have a constant spectrum as the time goes on due to the PCB warming. This case is not practicable for the temporal uniformity of the spectrum.

This problem could not be solved, except doing a complex feedback loop which would require a constant monitoring of the spectrum which is not workable. However, this problem can be much attenuated by choosing specific resistors with a low temperature dependence. This temperature dependence is characterized with a *TCR* which stands for *Temperature coefficient of resistance* [60]. EQUATION 20 defines this TCR coefficient:

$$TCR = \frac{R_2 - R_1}{R_1 \cdot (T_{\text{operation}} - T_{\text{ref}})}.$$ (20)

It is often expressed in ppm/°C which stands for "parts per million per Celsius degree" while the usual reference temperature is 25°C. R_1 is the initial resistor value at $T = T_{\text{ref}}$ and R_2 is the resistor value at the operating temperature. The lower the TCR, the lower the temperature dependence and thus the resistor variation.

As example, the $40 \text{ k}\Omega$ resistors used for setting the LED current at $250mA$ have a temperature dependence of $5ppm/°C$ [52]. This means that a temperature increase of $20°C$ leads to a resistance variation of $\pm 4\Omega$. This dependence is guarantee in a range between 0 and $70°C$. Outside this range there is no available information to characterize it. As the board will be maintained below $70°C$ there is not problem with it.

As it could be inferred from the previous computation, choosing 5ppm/°C or 20ppm/°C resistors does not drastically change the simulator behavior as the nominal values are around several tens of kΩ. However, it is always a good practice to master these details before defining if they worth the extra money spent on it. In the case, it was completely overkill and thus the other resistors are not chosen to have a specifically low TCR.

3.2.4 Input voltage and LED outputs

To power the LEDs and the drivers, the input voltage can be in a range from 6 V to 60 V. The voltage required to supply all of it is depending on the number of LEDs placed in series and on the forward voltage of these LEDs. As the maximum number of LEDs in series is three and that the maximal LED forward voltage is about 3 V, EQUATION 21 computes the minimum supply voltage for this configuration:

$$V_{\text{min}} = N_{\text{LEDs}}^{\text{br}} \times V_{\text{LED}}^{\text{forward}} = 3 \times 3 = 9 \text{ V}.$$ (21)

This voltage is the minimum required to power 3 LEDs in series. In our simulator, there are also tracks with either two LEDs in series or even only one. A simple solution would be to set the voltage at the same maximal value for all the drivers. However, as the drivers are linear regulators, they will dissipate the excess amount of power that do not go through the LEDs. Indeed, it will act as a dividor between the power supplied to the LEDs, through the current control, and the power that needs to be dissipated by the chip. If the input power, directly related to the input

voltage, is oversized, the drivers will have to dissipate the excess of power. For this reason, it is not wise to feed all the drivers with the same voltage source.

On the first place it is completely inefficient as there will be power wasting in dissipation. Moreover, as the power will be dissipated as heat, it will warm up the PCB and lead to a decrease of the LED efficiency, etc [cf. SECTION 5.1].

The solution that was designed is to use various voltage sources as inputs for the different LED rails. A rail is considered to be a series of one, two or three LEDs in this case. EQUATIONS 22, 23 and 24 compute the voltage of the sources that are used:

$$V_{\text{min 3 LEDs}} = N_{\text{LEDs}}^{\text{br}} \times V_{\text{LED}}^{\text{forward}} = 3 \times 3 = 9 \text{ V} \implies V_{\text{supply 3 LEDs}} = 10 \text{ V}. \tag{22}$$

$$V_{\text{min 2 LEDs}} = N_{\text{LEDs}}^{\text{br}} \times V_{\text{LED}}^{\text{forward}} = 2 \times 3 = 6 \text{ V} \implies V_{\text{supply 2 LEDs}} = 7 \text{ V}. \tag{23}$$

$$V_{\text{min 1 LED}} = N_{\text{LEDs}}^{\text{br}} \times V_{\text{LED}}^{\text{forward}} = 1 \times 3 = 3 \text{ V} \implies V_{\text{supply 1 LED}} = \cancel{3.3 \text{ V}} = 7 \text{ V}. \tag{24}$$

The supplied voltages are higher than what have been computed to have margins in case of higher LED forward voltages, etc. As the $V_{\text{min 1 LED}}$ was lower than 6 V, the drivers that features only one LED are supplied with the same voltage source as the two LEDs ones. It will dissipated more heat but there is not choice to keep those drivers working. The drivers will thus be powered with 7 V and 10 V voltage sources.

3.3   PWM generators

To modulate the current, each channel needs to have its own PWM signal. These signals are generated through *PCA9685* which are PWM generator chips [37]. These chips are manufactured by *NXP Semiconductors*® and each chip features up to *16* channels with individual PWM signal. These PWM signals will be connected to the driver inputs to control the currents. As there are *24* different LED wavelengths to control, there are two chips for a total of *32* PWM signals.

The *PCA9685* has an I^2C interface through which the PWM signals can be encoded on 12 bits. It means that the duty cycles can be adjusted from 0 to 100% with a accuracy of [EQUATION 25]:

$$N_{\text{steps}}^{\text{br}} = 2^{12} = 4096 \text{ steps} \longrightarrow \Delta_{\%} = 0.0244\%. \tag{25}$$

The minimal duty cycle step being below 0.1%, it should be accurate enough for our application. Moreover, the frequency of the PWM signals can be adjusted from its internal register. As there is no need of unique feature, the PWM signals will have a standard 1 kHz frequency.

3.4   Raspberry Pi

To program the PWM generators, the I^2C bus of a Raspberry Pi 3 is used. The Raspberry, under Raspbian as operating system, will run some programs to operate the solar simulator and to control the power of each LED. This Raspberry will be connected outside of the main board through its main header. FIGURE 19 shows a schematic of this Raspberry Pi 3.
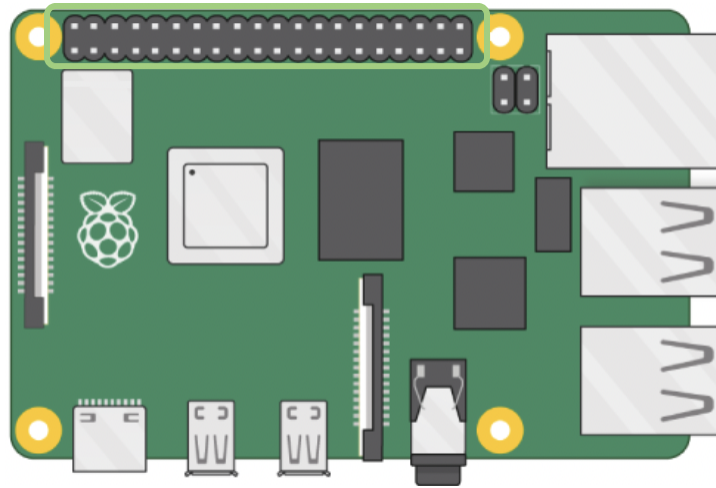
Figure 19 – Raspberry Pi schematic representation [19].

3.5   Power converters

To supply the LEDs, the Raspberry Pi, the drivers and all the components of the board, a power supply is required. This power source will be a 12 V voltage source that is connected to a main AC/DC charger. However, the board requires other voltage sources as well to power the drivers (7 V and 10 V), the PWM generators (3.3 V), etc. For this purpose, the main 12 V voltage source will be transform into 10 V, 7 V, 5 V and 3.3 V. This conversion is done using LDO and DC/DC buck converters. TABLE 3 shows the various voltage sources and for which purpose they are used:

Voltage source	Source	Purpose
12V	Main AC/DC charger	Board supply
10V	*12V-10V* buck converter	Driver supply
7V	*12V-7V* buck converter	Driver supply
5V	*12V-5V* buck converter	Raspberry Pi supply
3.3V	*5V-3.3V* LDO	PWM generator supply

Table 3 – Summary of the voltage source origins and purposes.

3.5.1 LDO

A simple way to convert the 12 V into 3.3 V is to use a low dropout regulator (LDO). The used one is the *TPS7B8133DGNR* from *Texas Instruments®* [49]. It can be supplied with an range of voltages between +3 V and +40 V and output a accurate +3.3 V. This 3.3 V voltage source will be used to supply the PWM generators. It is a fast and cheap way of converting voltage sources; nevertheless the output current is limited. Indeed, the maximal output current can not exceed 150 mA. This current is high enough to power the two *PCA9685* but is clearly undersized to feed the LED drivers. For this application, DC/DC buck converters have to be designed.

3.5.2 DC/DC buck converters

To power the drivers and the LEDs with much higher currents, the voltage source conversions are done using DC/DC buck converters. Those converters are designed using the *Texas Instruments®* power designer. The 12 V to 5 V conversion will be used as example but the others voltage sources are obtained in the same way.

The key parameters to search for a buck converter is to feed the power designer with the input voltage range (here 12 V), the desired output voltage (here 5 V) as well as the output current. This maximal output current was set at 3 A. To achieve this conversion, the $TPS56339$ buck converter is used [48]. The power designer provides us with the component values, their references as well as simulations. FIGURE 20 shows the efficiency as a function of the output current for the 12 V to 5 V conversion. With 3 A current drawing, the efficiency for the 5 V, 7 V and 10 V are respectively about 92.1%, 94.3% and 95.2%
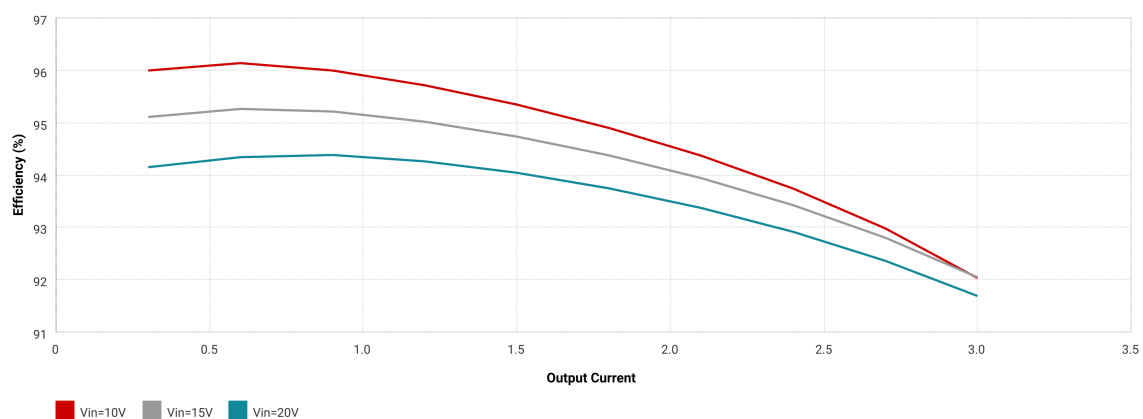
Figure 20 – Efficiency vs. output current of the DC/DC buck converter [23].

This maximal output current was set after computing what the LEDs would draw. It is done by summing all the current that LEDs would consume from each voltage source:

- 5 **V:** This voltage source will be used to power the Raspberry Pi which will not consume more than 2 A.

- 10 **V:** This voltage source will power 11 rails of LEDs. Taking into account the maximal consumption of each rail, the total equals 2.35 A. This is below 3 A and the LEDs will never be power ON at 100% simultaneously.

- $7V$: This voltage source will power 18 rails of LEDs. Taking into account the maximal consumption of each rail, the total equals 4.25 A. As it is higher than 3 A, there are two identical buck converters to split the load. The first one is connected to a set of rails consuming up to 2.3 A while the second one is connected to another set with a maximal drawn current of 1.9 A. In such a way, the same converter design is kept and there is room for upgrades [cf. SECTION 8].

4 Board design

This section will be dedicated to the design of the solar simulator on *Altium designer®*.

4.1 Schematics

Once the components have been chosen, they need to be interfaced to each other before routing the PCB. One are going to analyze some of the key elements in this schematic. The full version can be found on APPENDIX A.

4.1.1 Power input

The power input of the PCB is a DC jack that takes as input a 12 V voltage source [47]. FIGURE 21 shows the schematic of this power input.
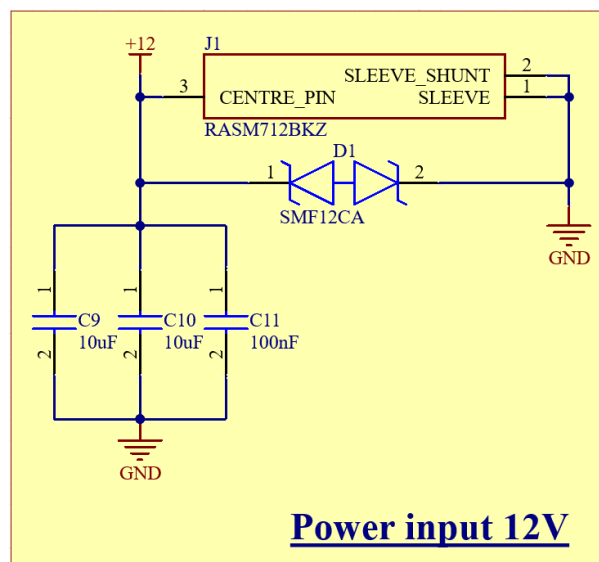
Figure 21 – Schematic of the power supply input.

As it can be seen, the center pin of the DC jack is label as the $+12$ V input while the sleeve is connected to the ground. In such a way, the circuit ground will be connected to the mains ground. There is a TVS diode in parallel which protects the circuit from disruptive transient voltages resulting from lightning, etc [25]. It is a bi-directional diode with a 12 V working voltage and a maximum clamping voltage of 20 V. The three decoupling capacitors in parallel are used as energy tank to stabilise the 12 V input. Note that there is no diode for polarity protection as there is no possible way to invert the + and - terminal due to the circular jack.

4.1.2   DC/DC buck converters

As explained above, the conversion between the 12 V input and the other voltage sources are done through DC/DC buck converters. FIGURE 22 shows the schematic of one of these.

12V to 10V DC/DC buck converter
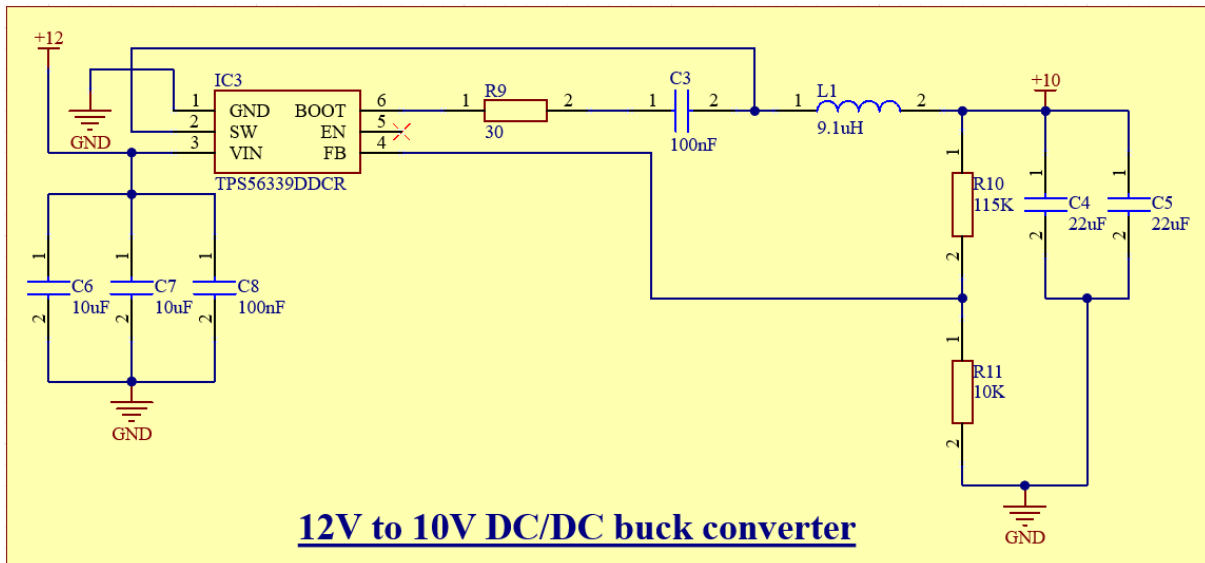
Figure 22 – Schematic of the $12\,V$ to $10\,V$ DC/DC buck converter.

As it can be seen, the *TPS56339* is connected to external components to convert the 12 V from the main power supply to a 10 V voltage source. There are decoupling capacitors at the input and the output of the converter which act as an energy tank and stabilise the voltages to their nominal values. The values of the external components follow the recommendations from the *Texas Instrument*® power designer and from the buck converter datasheet. The output voltage is set through the ratio between the feedback resistors R_{10} and R_{11} while the enable pin needs to be left floating in order to activate the buck converter. EQUATION 26 shows the feedback resistor values to achieve a 10 V output voltage:

$$R_{10} = \frac{V_{\text{OUT}} - V_{\text{REF}}}{V_{\text{REF}}} \cdot R_{11} = \frac{10 - 0.802}{0.802} \times 10 \text{ k}\Omega = 114.69 \text{ k}\Omega. \tag{26}$$

The V_{REF} voltage is provided by the datasheet while the value of R_{11} is recommended to be larger or equal to 10 kΩ. Using large values for resistors improves the efficiency of the buck converter. The datasheet provides also equations for the computation of the inductor value as well as the input and output capacitors. These components are chosen to have a specific maximal ripple current, a given filtering, etc. In this case, the standard values already computed by the power designer will be used.

This topology is exactly the same for the two 7 V and the 5 V converters. The only changes are about the inductor values and the feedback resistors.

4.1.3   LDO

The 3.3 V voltage source being used only to power the PWM generators, there is no need of a complex and cumbersome buck converter. FIGURE 23 shows the simple schematic of the LDO used.
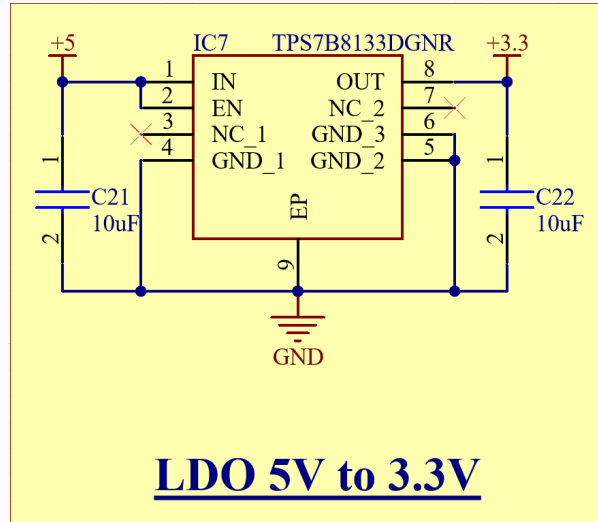
Figure 23 – Schematic of the LDO regulator.

The input is set at 5 V while the output is labelled as the 3.3 V voltage source. This 5 V input comes from the 12-5 V buck converter that powers the Raspberry Pi. Finally, there are decoupling capacitors on both sides to act as energy tank and to smooth the voltages.

4.1.4 Interface

The interface is the connection between the board and the Raspberry Pi 3. It consists of a SMD header that is going to be connected to the Raspi main header. The schematic of this interface can be seen on FIGURE 24.
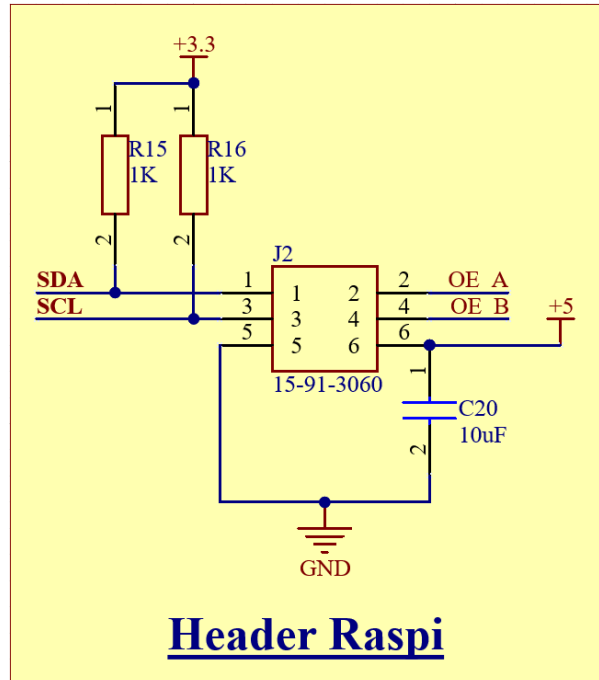
Figure 24 – Schematic of the interface header.

The OE_A and OE_B come out from the PWM generators and are used to control simultaneously all the LEDs. These pins should be connected to the ground to enable the PWM signals. The SDA and SCL are the two lines of the I^2C bus and are connected on the same ports on the Raspberry Pi 3. There are two pull-up resistors for this I^2C bus as the bus needs to be pulled up to 3.3 V. However, there are already pull-ups included in the Raspberry which is the reason why these resistors are not populated and will be solder only in case of backup.

Finally, the board ground is connected to the Raspberry ground which is power with the 5 V voltage source. This tiny computer should not exceed a consumption of 2 A and as the 5 V comes out of the DC/DC buck converter which can output up to 3 A, there should be much enough power to supply it.

4.1.5   PWM generators

FIGURE 25 shows the schematic of the two *PCA9685* PWM generators.
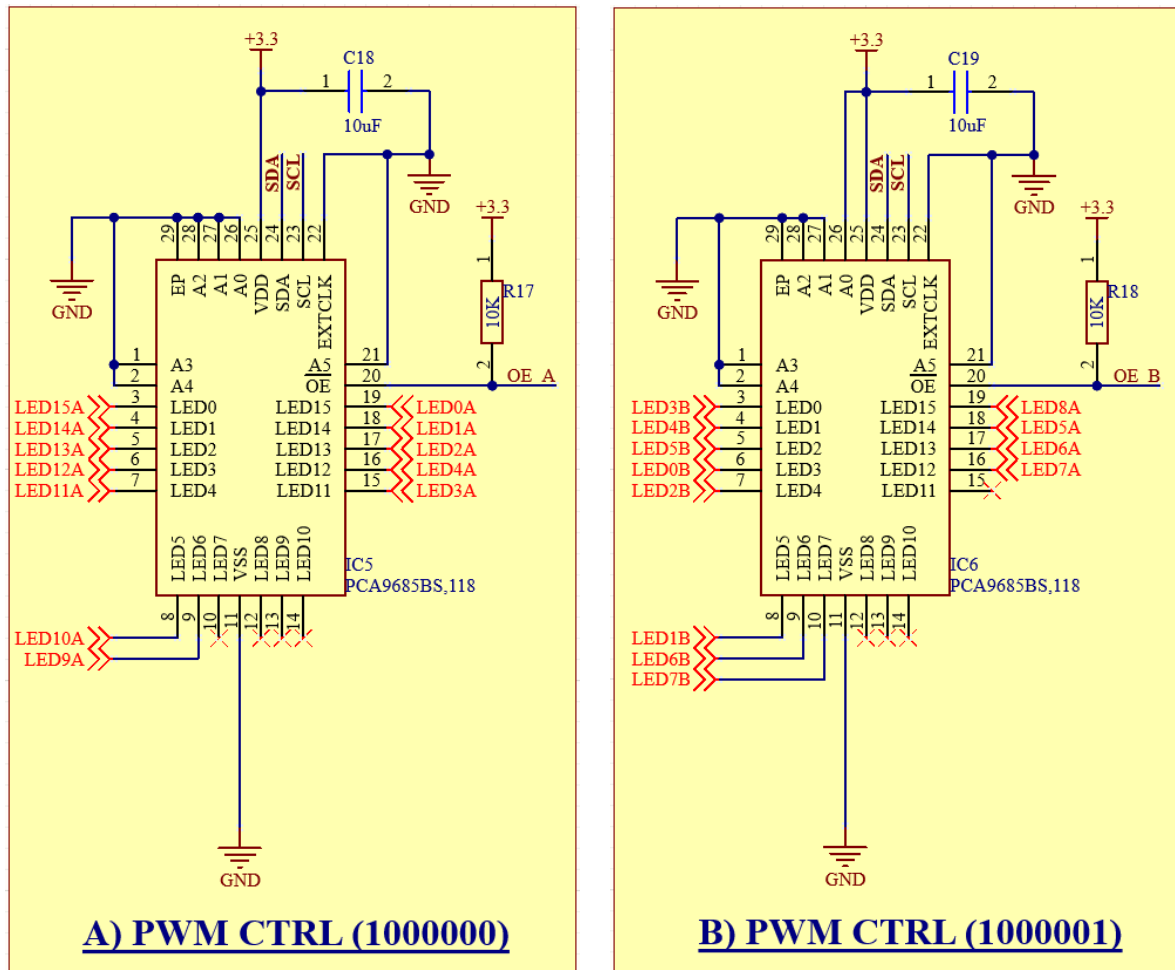
Figure 25 – Schematic of the PWM generators.

For both PWM generators, they are powered with the 3.3 V voltage source coming out of the LDO. There is a decoupling capacitor to smooth the voltage and acts as energy tank. They are both connected to the I^2C bus of the Raspberry Pi through the SDA and SCL ports. The addresses of the two devices are set according to FIGURE 26.
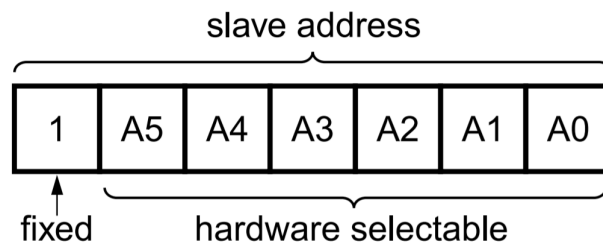
Figure 26 – Selection of the addresses through hardware pins.

The low or high value on the address pins A_x is achieved either by connecting the pin to the ground GND or to the supply VDD. In this case, the address of the first PWM generator is set

---

to *1000000* while the second is set to *1000001* by connecting A_0 to VDD.

For both ICs, the EP pin is connected to the ground. This is not a pin but it corresponds to the exposed pad under the chip. It needs to be solder to the ground to enhance thermal performances and to allow generated heat to be extracted. There are also the $\overline{OE}$ pins, *Output Enable*, which are connected to the Raspberry Pi with pull-up resistors.

Finally, the output PWM pins will be connected to the corresponding driver channels. As they are not on the same schematic sheet, some *Off sheet connectors* are used and act as labels. If some of them are not required, the corresponding pins are left floating.

4.1.6   LED drivers

FIGURE 27 shows one of the schematics with the *AL1783* LED driver.
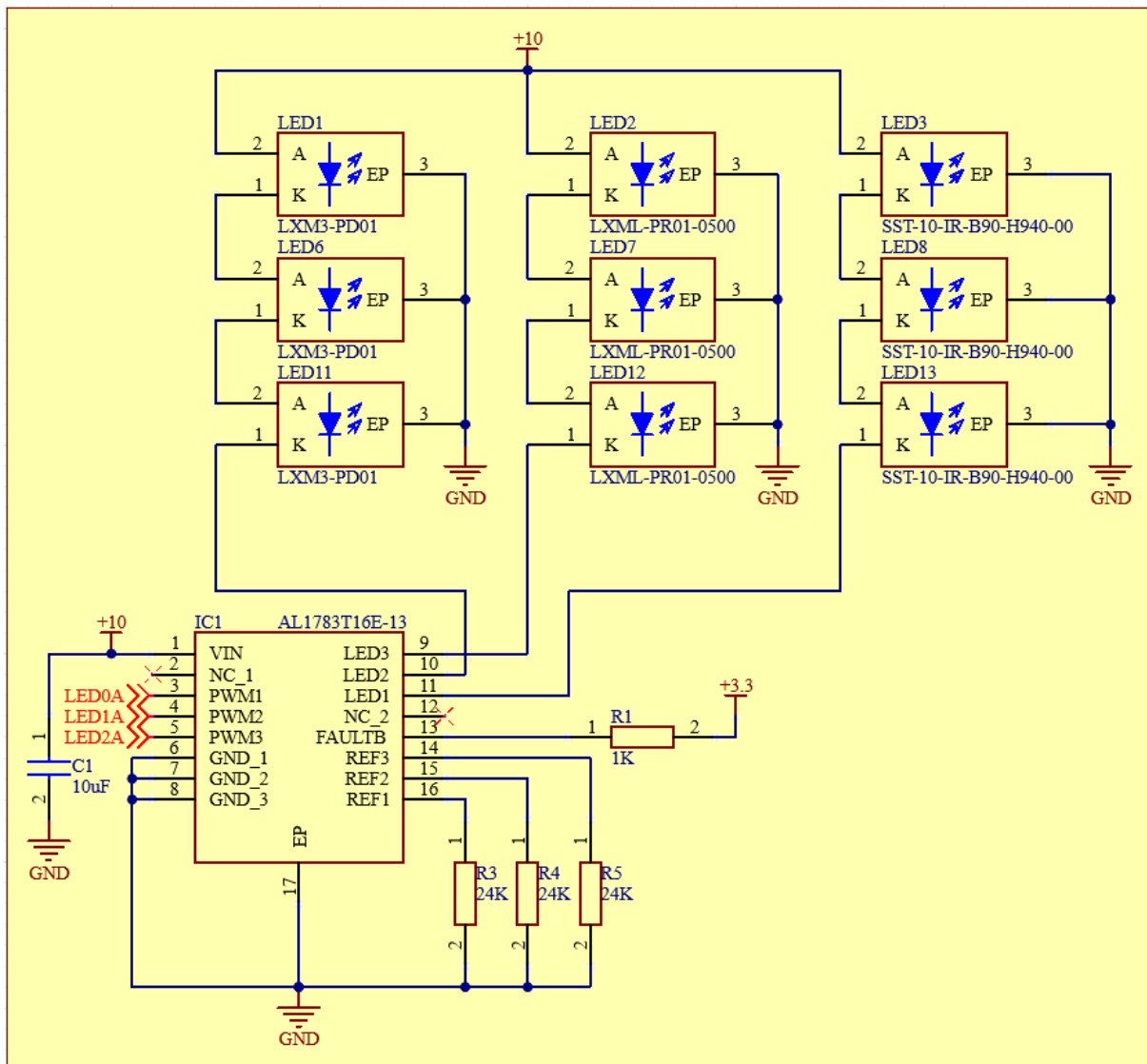
Figure 27 – Schematic of the LED driver.

As it can be seen, the PWM signals coming out of the PWM generators are connected to the PWM input pins of the chip. It is these PWM that will be used to dim the LEDs. The REF_x pins are connected to a given resistor value and fix the maximal current in the corresponding channel [cf. Section 3.2.2]. The *FAULTB* is a fault report pin that needs to be pulled up. As this feature of the ICs is not used in this project, there is no connection to the Raspberry Pi.

The IC is fed with a given voltage source according to the number of LEDs in series on its channels. The design was made such that each driver has the same number of LEDs in each of its channels. The drivers which are connected to three LEDs in series are fed with 10 V and the ones with one or two LEDs in series with 7 V. These voltages sources came out of the DC/DC buck converters as they required substantial currents. As always, there is a decoupling capacitor at the input to smooth the voltage and to acts as energy tank.

The LEDs which need to be driven are connected to the channels LED_x. Each channel drives a set of LEDs with the same wavelength. In such a way, the radiative power at each wavelength can be individually controlled and monitored. The LEDs are connected through their cathodes at the channel input as the current goes from the power source to the driver. When the LEDs feature an exposed pad, this one is also connected to the ground.

Finally, as it is a linear driver which will dissipate a significant amount of power to control the LEDs, the exposed pad needs to be solder to the ground as well to enhance thermal performances.

4.2   PCB design

Once the schematic is done, the components need to be placed and routed on the board. This board will constitute the PCB of the solar simulator. In this section, some followed design rules will be explained as well as key features of the PCB.

4.2.1   Layers

One of the first thing that needs to be defined is how much layers to PCB will be made of. In this case, two layers are not enough as the LEDs need to be packed as close as possible with a lot of wiring which is why a four-layers PCB has been chosen. FIGURE 28 shows the layer stack which sets up the PCB.

#	Name	Material		Type	Weight	Thickness	Dk	Df
	Top Overlay			Overlay				
	Top Solder	Solder Resist	⋯	Solder Mask		0.01016mm	3.5	
1	Top Layer		⋯	Signal	2oz	0.07mm		
	Dielectric 2	PP-006	⋯	Prepreg		0.07112mm	4.1	0.02
2	Layer 1	CF-004	⋯	Signal	1oz	0.035mm		
	Dielectric 1	FR-4	⋯	Dielectric		0.32004mm	4.8	
3	Layer 2	CF-004	⋯	Signal	1oz	0.035mm		
	Dielectric 3	PP-006	⋯	Prepreg		0.07112mm	4.1	0.02
4	Bottom Layer		⋯	Signal	2oz	0.07mm		
	Bottom Solder	Solder Resist	⋯	Solder Mask		0.01016mm	3.5	
	Bottom Overlay			Overlay				

Figure 28 – Overview of the layer stack.

Indeed, as the bottom layer is only a ground plane [cf. SECTION 5.1], there is not enough room on the top layer to route all the tracks. In this case, the two inner layers are dedicated to the routing of tracks that can not fit on the top layer. The parameters as the layer thickness, the dielectrics used, etc are the Altium default ones and are standard for manufacturing the PCB without additional costs.

4.2.2   Vias

To connect tracks from one layer to another, vias are used. As the bottom layer needs to be a full plane of ground for sticking an heat sink on it, the ideal solution would be to use blind and buried vias to go through layers instead of through hole vias. The cross section of such vias can be see on FIGURE 29.

#	Name	Type	Thickness		#	Thru 1:4	Blind 1:2	Blind 1:3	Buried 2:3
	Top Overlay	Overlay							
	Top Solder	Solder Mask	0.01016mm						
1	Top Layer	Signal	0.07mm		1				
	Dielectric 2	Prepreg	0.07112mm						
2	Layer 1	Signal	0.035mm		2				
	Dielectric 1	Dielectric	0.32004mm						
3	Layer 2	Signal	0.035mm		3				
	Dielectric 3	Prepreg	0.07112mm						
4	Bottom Layer	Signal	0.07mm		4				
	Bottom Solder	Solder Mask	0.01016mm						
	Bottom Overlay	Overlay							

Figure 29 – Overview of the different categories of vias.

However, these specific vias are expansive to manufacture. For this purpose, only through hole vias are used in this PCB. In order to avoid short circuits between the vias and others signals, they are tented [14]. It means that they are covered with a layer of solder mask. In such a way, the vias are protected against corrosion and it provides the vias with a high electrical isolation ($\sim 10^{12}\Omega$).

4.2.3 LED implementation

The key element of the solar simulator being the spectrum emitted by the LEDs, they need to be as close as possible for spatial uniformity. They are thus the first elements placed on the center of the PCB. After testing various designs and implementations, a circular shape appears to be the most compact way to place the LEDs. They are put as close as possible from each other while leaving some room for thermal vias and future tracks. FIGURE 30 shows the final implementation.
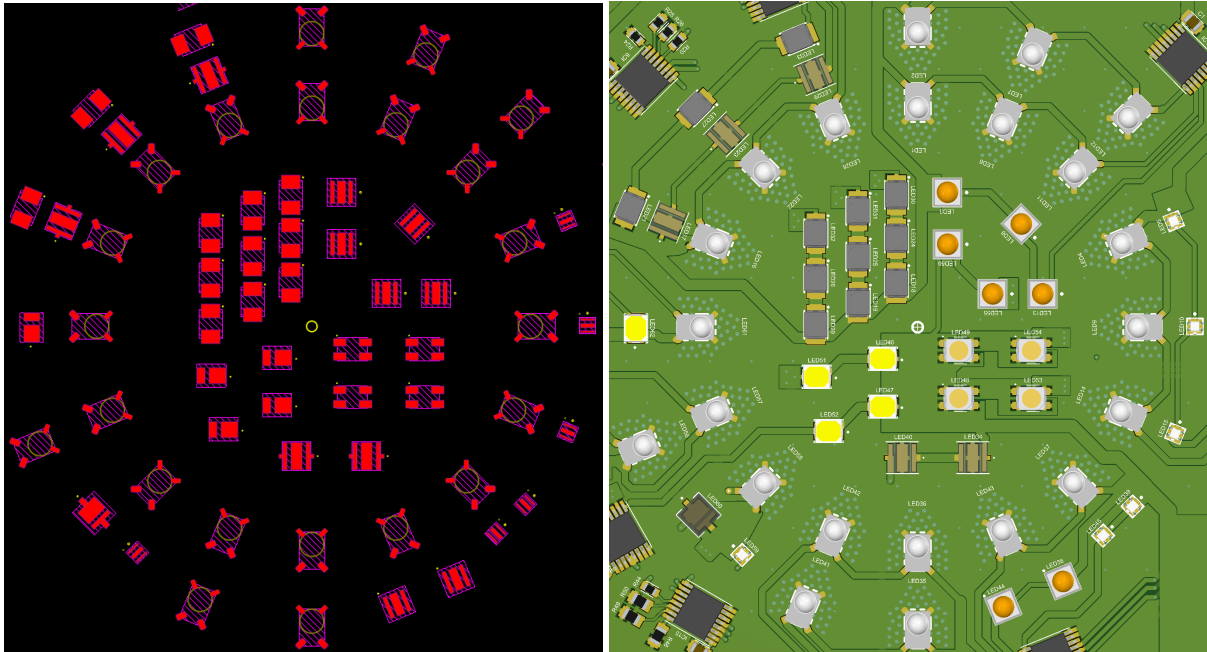
Figure 30 – Overview of the LED implementation on the PCB.

On the left figure, 3D bodies of the LEDs are shown while the right figure is the 3D view with the footprints, the tracks, the vias, etc. As it can be seen, there is just enough room to route tracks around the footprints.

Moreover, the LEDs are placed according to which LED driver they belong. Grouping the LEDs in such a way leads to a better routing. Indeed, if the LEDs were split randomly across the board, the routing would not have been optimal as the tracks would have to cross the board in various directions to reach the given LEDs. In this case, the tracks are clear, clean and as short as possible. However, the disadvantage of this method is that LEDs with the same wavelength (i.e. from the driver) are packed on the same area. It might decrease the spatial uniformity of the spectrum but there are no other better choices to make the PCB routable. This spatial uniformity will be validated while doing the simulations [cf. SECTION 7].

4.2.4 LED drivers

The white circles on FIGURE 31 shows the position of the *AL1783* LED drivers on the PCB.
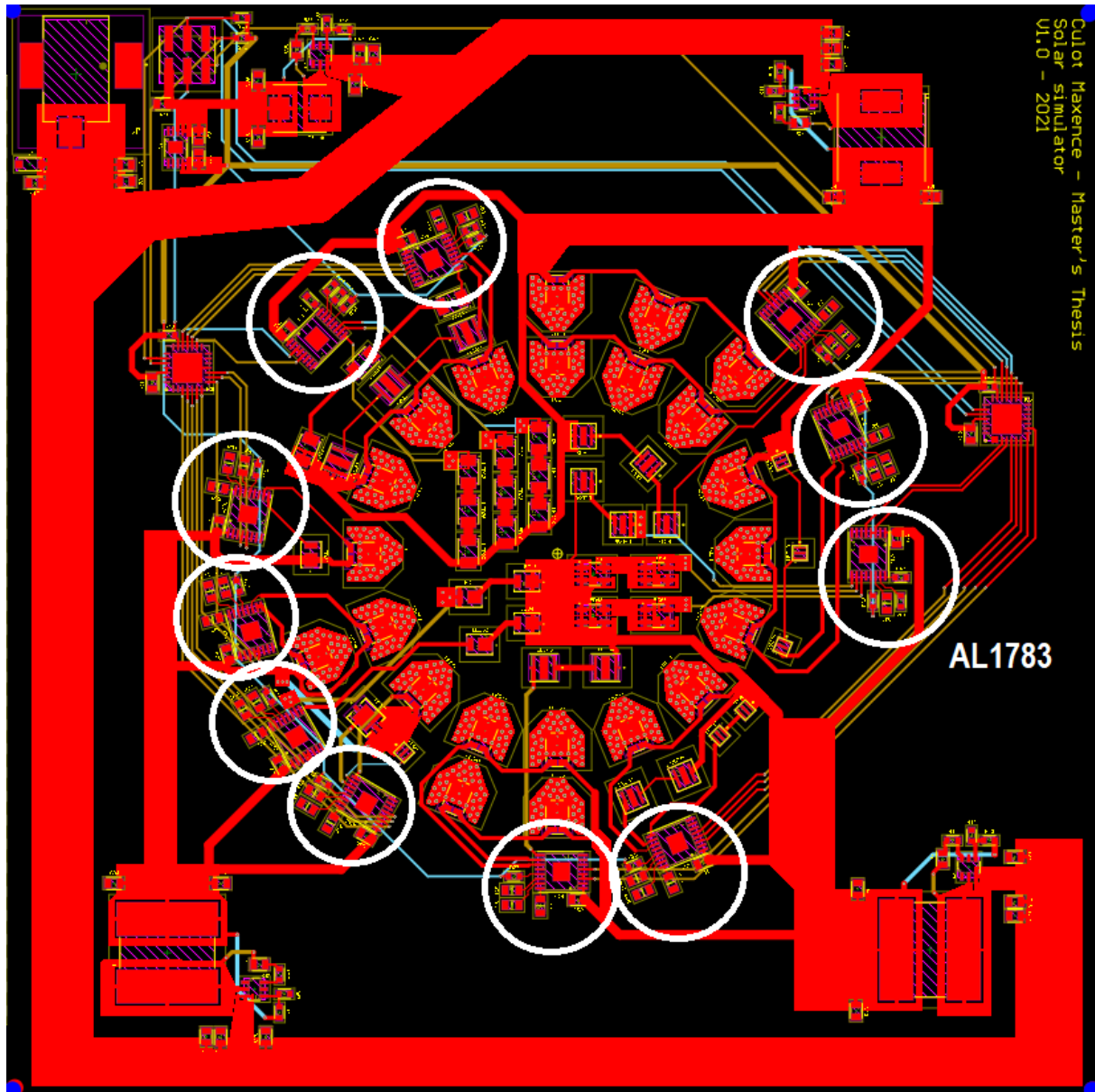
Figure 31 – Overview of the driver implementation on the PCB.

As it can be seen, they are placed around the periphery of the imaginary circle made by the LEDs. The LED drivers are as close as possible from the rails which they control.

4.2.5   PWM generators

The white circles shows the location of the PWM generators on FIGURE 32.
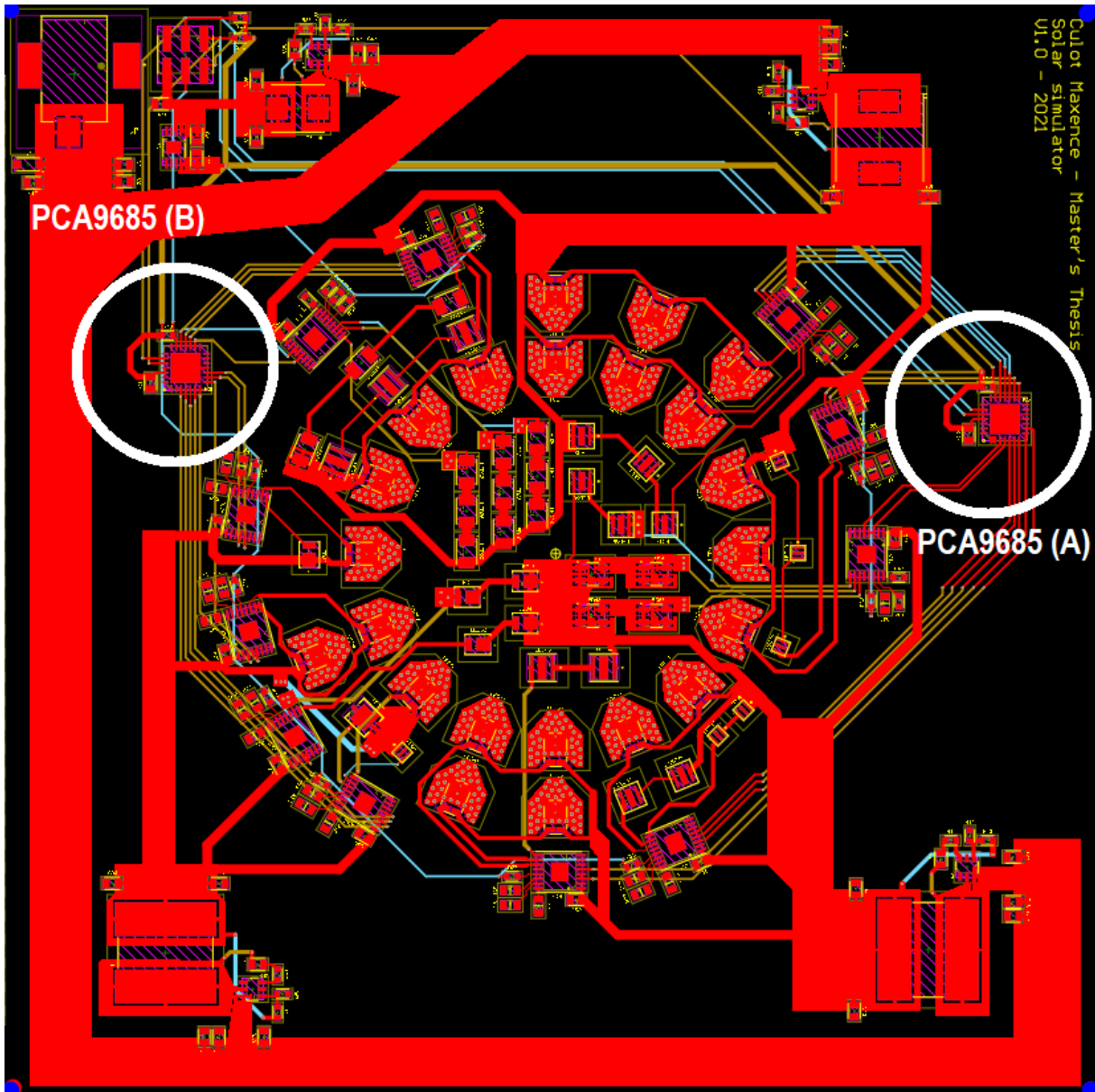
Figure 32 – Overview of the PWM generator implementation on the PCB.

They are placed around the drivers where the PWM signals should be monitored. The drivers belong either to the PWM generator *A* or *B*. The choice is done as a function of the path between the two. Each driver is connected to the closest PWM generator if there are enough free PWM pins available on the chip. As there is a pretty symmetric repartition, each PWM driver is loaded with approximately the same number of drivers (5 drivers for A and 6 drivers for B).

4.2.6   DC/DC buck converters

The DC/DC buck converters are placed on the board periphery, the closest possible from where the voltage source they deliver is required. The position of these converters is represented by the white circles on FIGURE 33.
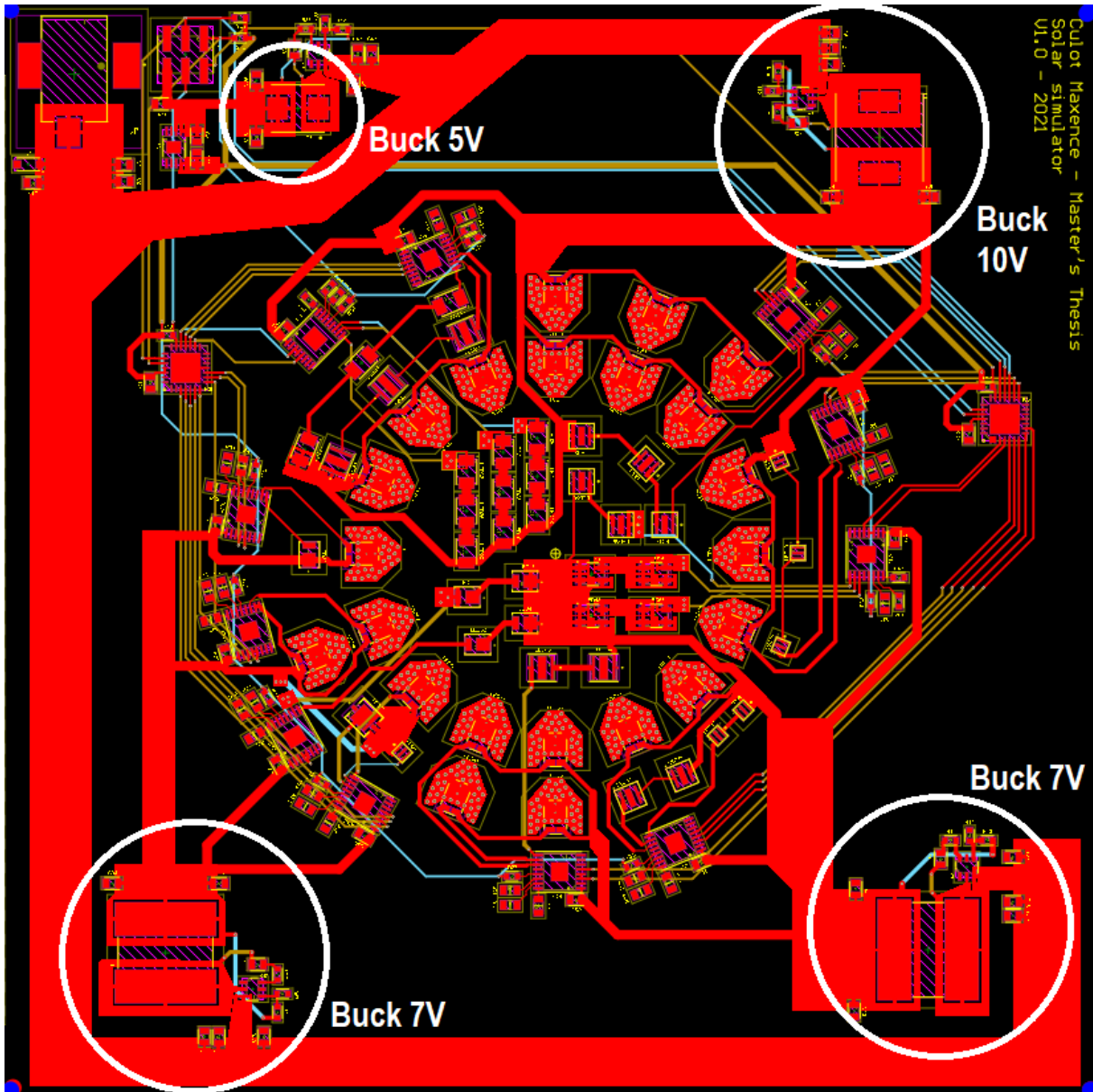
Figure 33 – Overview of the DC/DC buck converter implementation on the PCB.

The two 7 V output converters are placed on the bottom of the board and supply the LEDs and drivers around while the 10 V converter is on the top corner. As it can be seen, their 12 V input is connected to the converters not via a track but through a power plane. Indeed, as there are a lot of currents carried (up to 3 A per converter), width tracks are not enough. Large power planes decrease the path resistance and avoid bottleneck issues. The 12 V input is thus carried through a large power plane that feed all the converters around the board.

The detailed topology of a buck converter is shown on FIGURE 34.
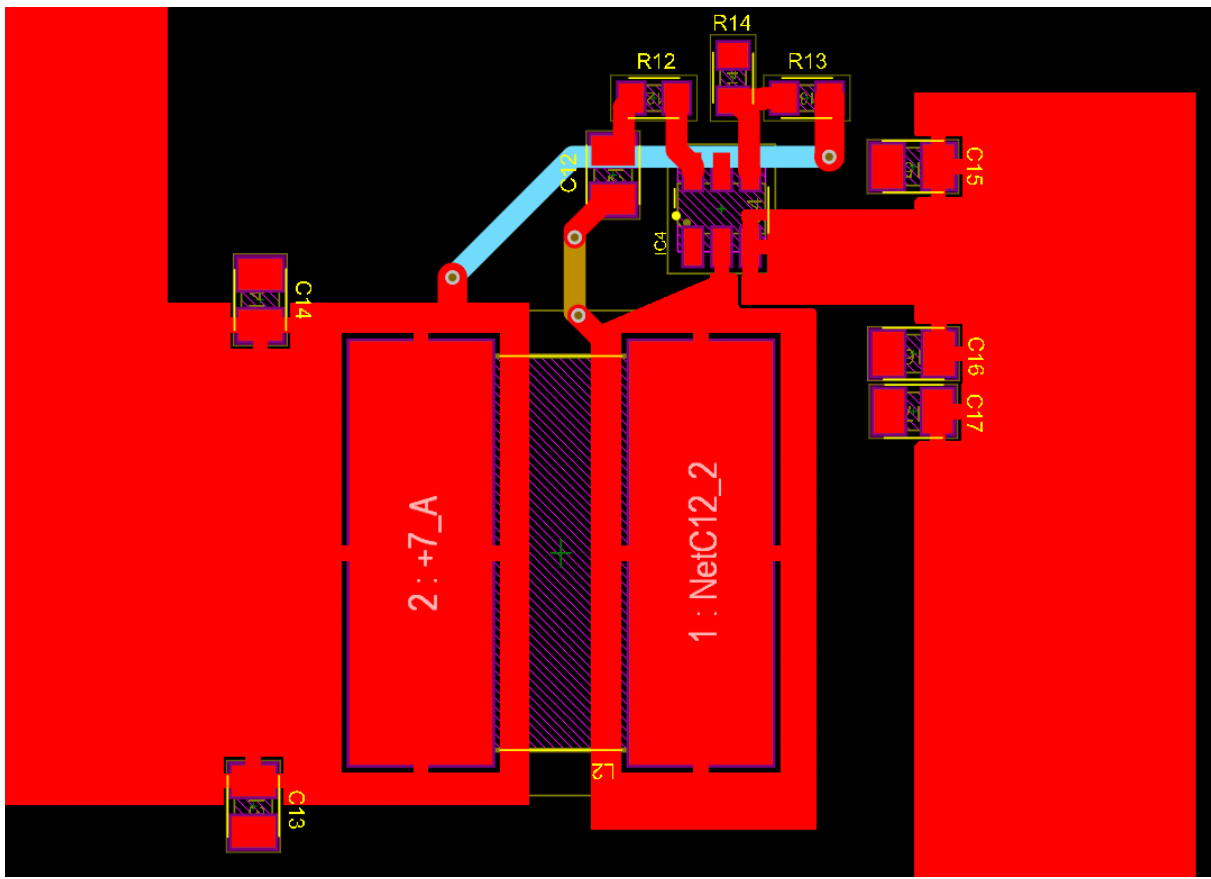
Figure 34 – Detailed routing of the $12\,V$ to $7\,V$ DC/DC buck converter.

This topology follows as closely as possible the recommendation of *Texas Instrument®*. The input V_{in}, the output $+7$ V and the SW pins are wired using power planes as they carry currents and thus power. Moreover, special care have been taken to do not wire tracks under the inductor. Indeed, it is an area where there could be interferences due to the electromagnetic field. All the four buck converters of the board have to same topology.

4.2.7   Main power supply and interface

On the top left corner lies the main power input, the 5 V buck converter and the interface header. The component placement can be seen on FIGURE 35.
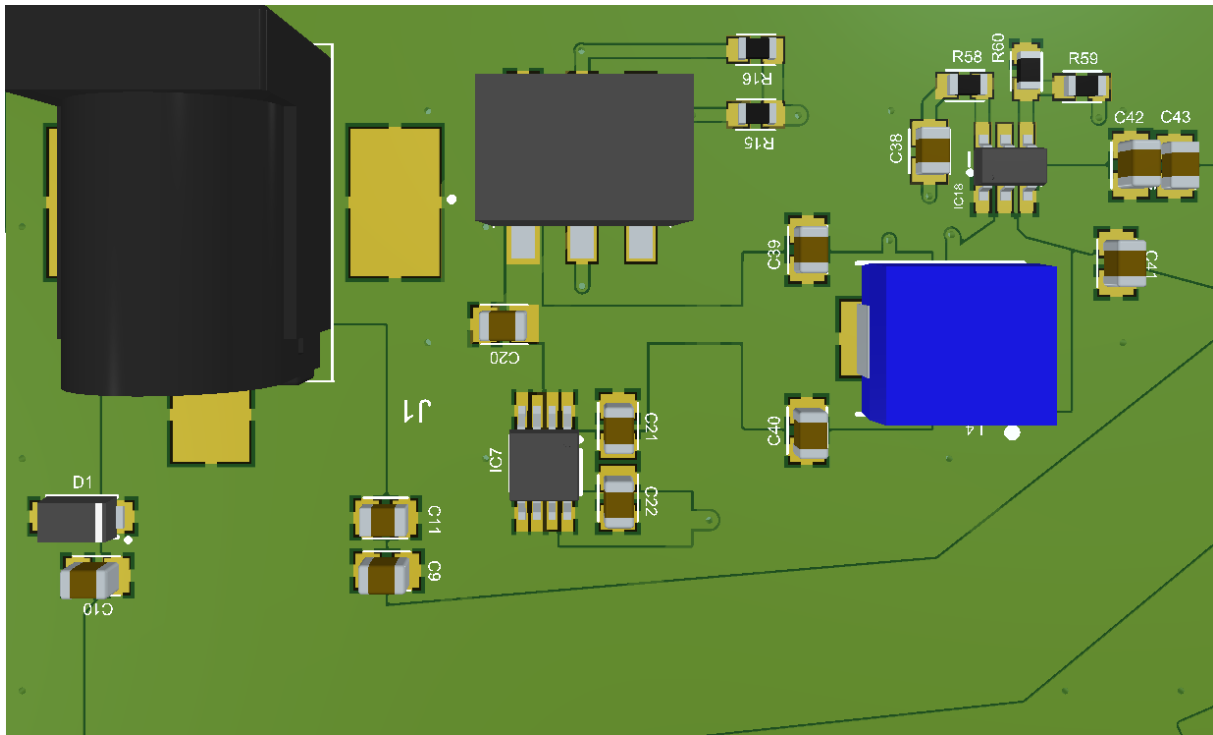
Figure 35 – Overview of the main power supply implementation on the PCB.

The power input and the header are packed together on the same board side as it will be easier for connection with the Raspberry Pi and the wall plug. In such a way, all cables exit the board on the same side. Moreover, the 5 V buck converter is placed on this side as its main function is to supply the Raspberry Pi through the header. It is placed thus close from the header and supplied by the DC jack. Finally, the LDO ($IC7$) is placed next to the 5 V voltage source and is routed up to the PWM generators.

4.2.8 Wiring

Once every component have been placed, it is time to route the tracks and power planes. However, this step was not straight forward and there are updates in the component placement as the routing was evolving. It is an iterative process to achieve a clear and clean routing while keeping the components ideally placed. The final routing of the components can be seen on FIGURE 36.
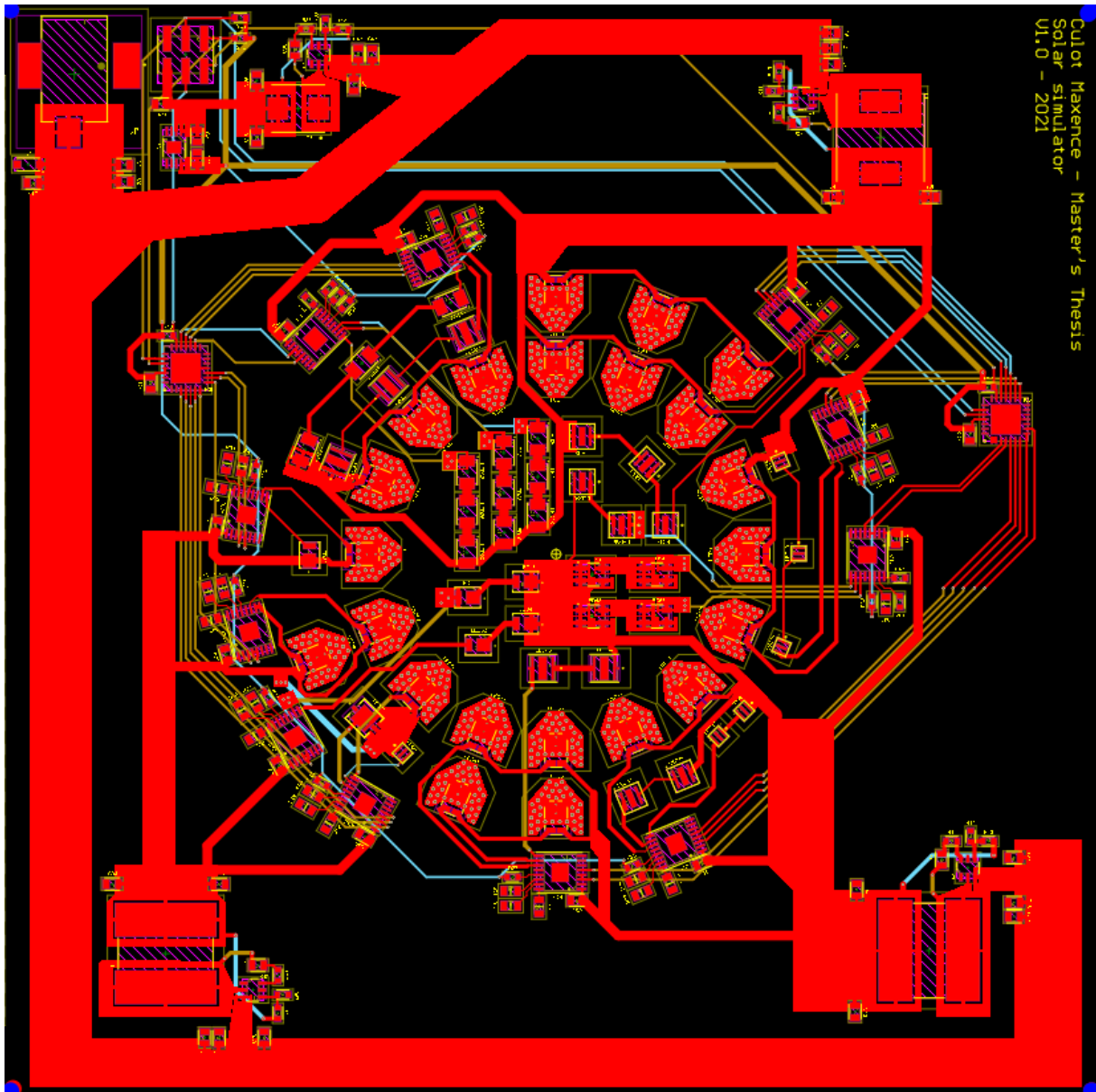
Figure 36 – Overview of the various tracks and power planes on the PCB.

On this figure, the red color is for the tracks lying on the top layer, the brown for the second layer while the light blue is for the third layer. As it can be seen, the tracks and power planes are routed on the first layer as a priority. Moreover, as much as possible, the power planes are placed on the top layer to avoid vias going for one layer to another and leading to bottlenecks. When there is not enough room to route on the top layer, the two inner layers are used, mainly to route the PWM signals for the PWM generators to the drivers.

This figure also shown the various widths used. Planes are used each time there is enough place, otherwise large tracks are used to conduct currents. For PWM signals, as they do not carry any power, there is no chance to have bottleneck issues with thin tracks and vias across the layers. They are thus routed with simple 20 mil tracks which is already to wide.

4.2.9   Ground planes

Finally, two ground planes have been added. One of on the top side and another on the bottom side. They connect all the grounds together without requiring tracks. In such a way, the resistivity do not depend on the track length and is thus much lower as it is equal to $\sim 0.5\text{m}\Omega/\square$ [45].

5 Thermal design and mechanical considerations

As the LEDs will be driven with high currents, a significant amount of power will be dissipated. To avoid damaging the LEDs and the other electronic components, these losses need to be taken into account in the design. In addition, there is a mechanical support that needs to be designed to hold the simulator in a standing position during our tests.

5.1 Thermal design

Driving LEDs generates losses as waste heat. Indeed, they have an efficiency around 30 to 50% which means that 50 to 70% of the power is dissipated as heat and not as luminous power. If one consider a LED model with a narrow band, the efficiency can be computed as the luminous power at the peak wavelength divided by the electrical power supplied. An example of such a computation can be found on EQUATION 27 for the *LXM3-PD01* LED:

$$\eta = \frac{LuminousPower}{ElectricalPower} = \frac{360 \text{ mW}}{350 \text{ mA} \times 2.10 \text{ V}} \approx 49\%. \tag{27}$$

It means that there are ~ 368 mW of heat to evacuate to keep the same temperature. If this heat is not evacuated, the junction temperature of the LED will rise and it leads to several issues. The two mains ones are the lifetime and the luminous flux decreases [59]. Indeed, as the LED junction goes up to its standard temperature (usually $25°$C), the relative luminous flux drops from its original value. FIGURE 37 shows this relationship for some of *Lumileds*® LEDs.
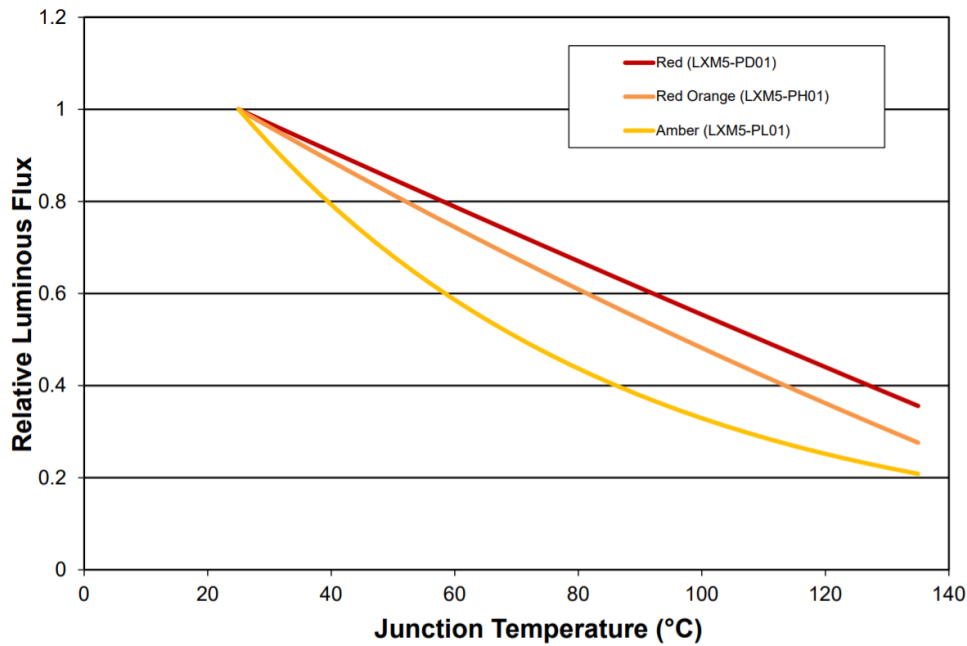
Figure 37 – Typical normalized light output vs. junction temperature for *LXM5-PD01*, *LXM5-PH01* and *LXM5-PL01* at *350mA*.

For this reason, it is required to keep the junction temperature to its nominal value as much as possible. This can be done only by extracting the heat.

5.1.1   Thermal resistance model

Before applying thermal design, it is a good practice to evaluate what would be the junction temperature of the LED. For this purpose, one build a thermal conduction model [20, 41]. This thermal model is equivalent as an electrical circuit where the current source is replaced by the heat generated, the resistors are replaced with thermal resistors and the voltage at each node is the temperature of the node [3]. For this model, one need to take into account the various elements between the power generation source and the dissipation. This first approximation model can be found on FIGURE 38.
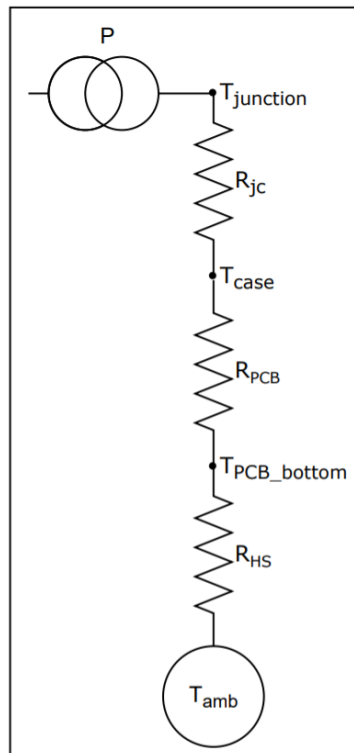
Figure 38 – Thermal resistance model.

where:

- P = input power to be dissipated [W]

- R_{jc} = junction to case thermal resistivity [°C/W]

- R_{PCB} = PCB thermal resistivity [°C/W]

- R_{HS} = heat sink thermal resistivity [°C/W]

- $T_{junction}$ = junction temperature [°C]

- T_{PCB} = PCB temperature [°C]

- T_{HS} = heat sink temperature [°C]

- T_{amb} = ambient temperature [°C]

The goal of such a model is to determine, as a function of the input power and the overall thermal resistivity, the junction temperature of the LED. The junction to case thermal resistance can be found on the LED datasheet and varies usually from 5 to 12°C/W. For our next computations, an average upper value of 10°C/W will be used.

To determine the PCB thermal resistance, the structure of it needs to be analyzed. Indeed, it will depend on the number of layers, the dielectric used, the vias, the solder pad areas, etc [11]. As

there are not only one single LED on the board nor unique footprint for all the LEDs, it is complicated to have an accurate model. The following numbers are thus only rough approximations.

The PCB resistance will be modelled like the heat can escape only through a conduction via, like a tube, under the solder pad [4]. FIGURE 39 shows what this model looks like while EQUATION 28 computes the via thermal resistivity:

$$R_{\text{PCB}} = \frac{L}{A \cdot k} = 3.22°\text{C/W}, \tag{28}$$

where:

- $L = 0.67228$ mm, the via length

- $A = \pi \left(\varnothing_{\text{vias}}^2 - \varnothing_{\text{hole}}^2 \right) = 0.54$ mm^2, the via surface

- $k = 385$ W/m°C, the copper thermal conductivity [35]
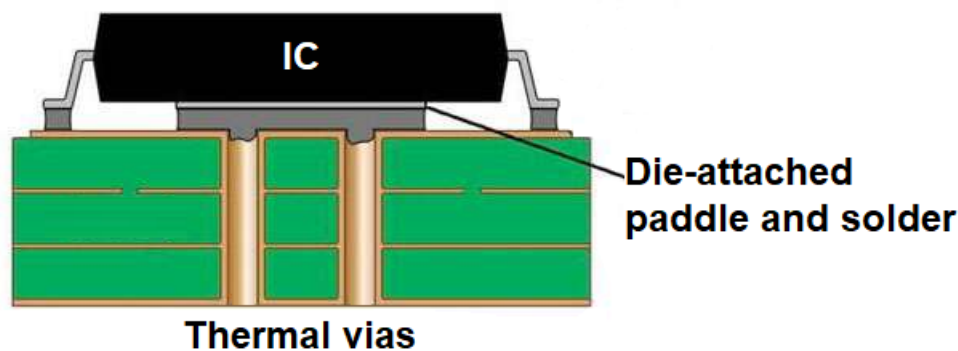
Figure 39 – Thermal via design for heat transfer into the circuit board substrate [12].

This value is only a first order approximation and is only valid for one via. However, as this matches what has been calculated by *Texas Instrument* tests (3.76°C/W [33]) and Osram Opto Semiconductors (8°C/W [42]) with similar configuration, the reality should not be so far away. Nevertheless, this model is not exact for our circuit. In our case, the vias are not placed on the expansion pad but on the surrounding of the component and their sizes are not constant as it depends of the footprint of the components. For this reason, this value will be mutiplied by 10 in the next computation. This margin will provide us with security as it is better to oversize the thermal solution than to undersize it. The approximated thermal resistance of our PCB is thus about 30°C/W.

Finally, the heat sink is the key element to bring the junction temperature to lower values. Without any heat sink, the thermal resistance on the bottom of the PCB will be very high as the only way to extract heat is trough convection and radiation. Ignoring the convection on a regular

room, it remains only radiation. The radiation resistivity between the cooper on the bottom of the PCB (without solder mask) and the air can be approximated as [EQUATION 29]:

$$R_{\text{rad/cm}^2} = 1000°\text{C/Wcm}^2. \tag{29}$$

As it can be seen, it is unsuitable as a power of 1 W will increase the junction temperature by $1000°$ C if we consider a surface of $1\,\text{cm}^2$ per LED. Mounting an heat sink on the board bottom side will replace this resistivity by a much lower one [32]. FIGURE 40 represents the PCB with the heat sink mounted on it.
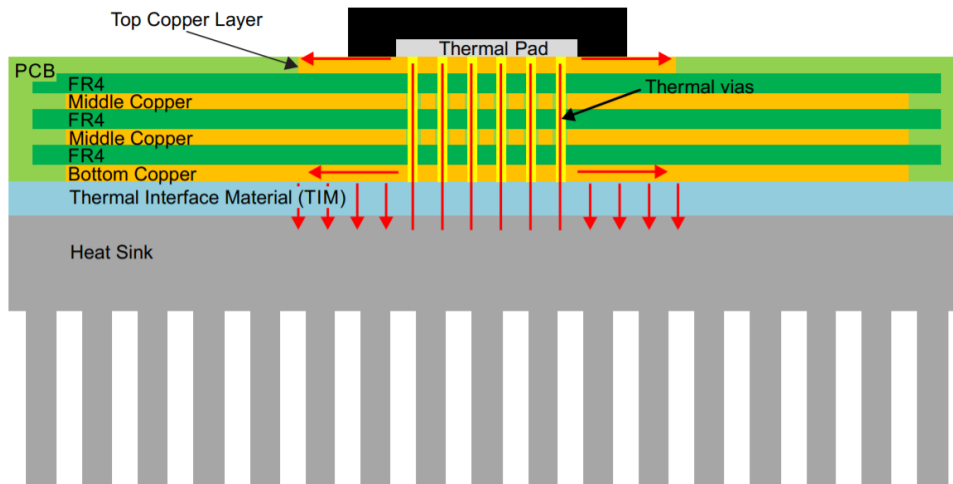
Figure 40 – Typical thermal layout of PCB with mounted heat sink [22].

As the heat sink thermal resistivity varies with air velocity, cooling fin size, etc, one will take an order of magnitude of $6°$ C/W [17]. This leads to an overall thermal resistivity about [EQUATION 30]:

$$R_{\text{overall}} = R_{\text{jc}} + R_{\text{PCB}} + R_{\text{HS}} \approx 10 + 30 + 6 = 46°\text{C/W}. \tag{30}$$

Taking into account that the LED dissipate ~ 368 mW and that the room temperature is at $25°$ C, EQUATION 31 computes the junction temperature:

$$T_{\text{junction}} = T_{\text{amb}} + P \cdot R_{\text{overall}} = 25 + 0.368 \times 46 \approx 42°\text{C}. \tag{31}$$

This final value of the junction temperature remains completely manageable and preserves a luminous flux about 90% from its original value. However, all these calculations are done like there is only one single LED on a smaller board than the original PCB. The reality is that the board is much larger and featured with a more than *60* LEDs on it. The only change in the model is that instead of using *60* heat sinks, there is only one large on the bottom of the board with the same thermal resistivity.

The final approximation done is regarding the interface material between the bottom of the PCB and the heat sink. Indeed, sticking the PCB without interface material increases the thermal

resistivity of the overall model. FIGURE 41 illustrates the micro gaps between the two surfaces which lead to contact losses and thus to a higher resistivity.
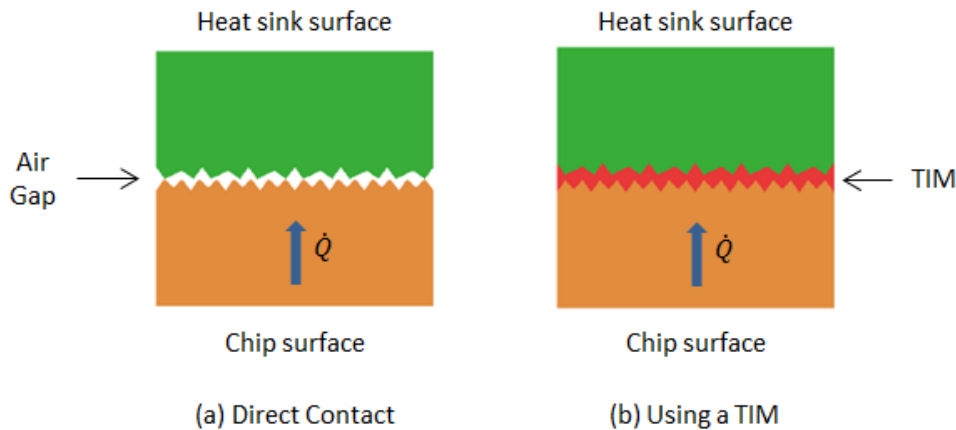
Figure 41 – Thermal interface material application [34].

Using a TIM, *Thermal Interface Material*, fills the gaps and decreases the thermal resistivity. The approximation done is that the TIM have a perfect thermal conductivity (i.e. $R_{\text{TIM}} = 0°$ C/W). If one use thermal grease (0.2° C/W), mica (0.4° C/W) or silicon pads (1.1° C/W) this approximation is pretty valid as it is much lower than the overall 46° C/W resistivity.

5.1.2 PCB design practices

There are several features implemented on the PCB to enhance thermal performances. First, thermal vias are placed on the component footprints. These vias are high conductivity paths for heat to be conducted up to the heat sink. FIGURE 42 shows the footprint of the *Lumileds®* Rebel Color series with thermal vias around the expansion pad.
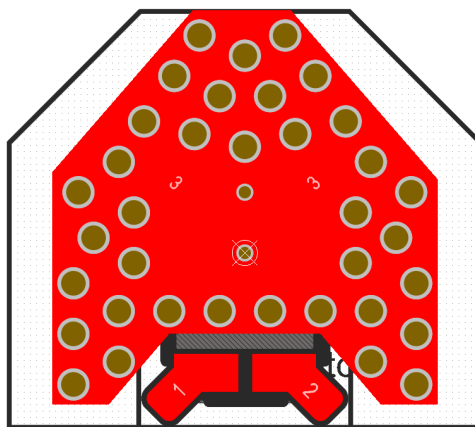
Figure 42 – *Lumileds®* LUXEON Rebel color line series footprint [27].

As it can be seen, there is a lot of vias around the exposed pad (3). Although it is not the case for every LED footprint, those LEDs are the most critical to be featured with thermal vias as these are the ones driven with the highest currents (up to 250 mA). In addition, the exposed pad

of all LEDs are connected to the ground plane. This GND will thus carry all the heat and, by connecting it to the heat sink, will dissipate through it.

To enhance even more the overall thermal performance, via stitching is applied on the ground planes. In such a way, the bottom and top ground planes are well interconnected through these vias and it add lots of thermal paths for heat to escape. These vias are placed on a grid at $5mm$ from each other. This provide a nice via covering on the full board.

Finally, at the bottom of the board, the solder mask is removed to expose the copper. APPENDIX B shows the removed back side of the PCB. As this solder mask is insulating, removing it allows the heat sink to be directly connected to the bottom layer of copper and enhances thermal conductivity.

5.2   Mechanical design

Once the simulator is build, it needs to be fixed to a mechanical support to maintain it in a standing position. As the heat sink will cover the back of the PCB, mechanical prolongations are designed such that the board can be mounted on a support without disturbing the heat sink assembly. FIGURE 43 shows these extensions connected to the board.
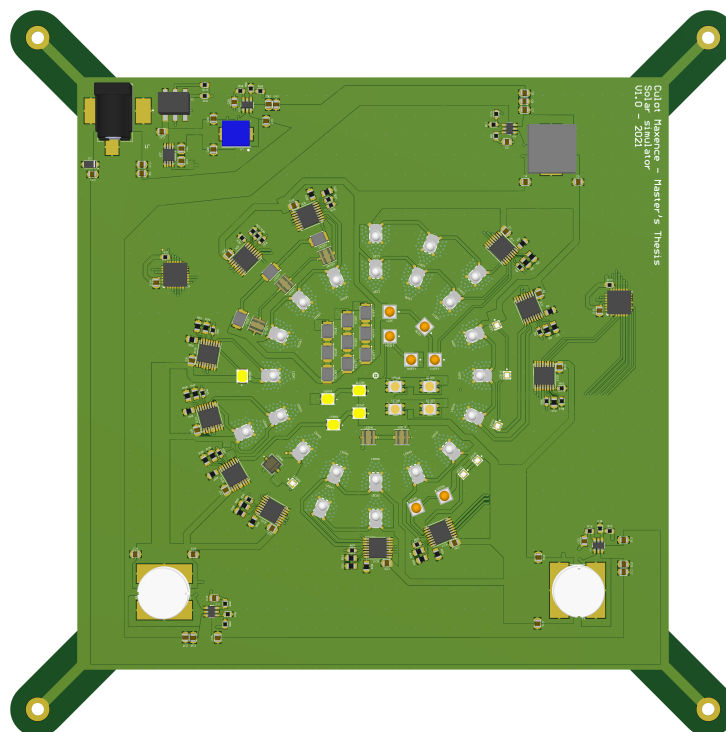
Figure 43 – 3D view of the PCB front face.

These peripheral holes will be used to screw the board the support with $M3$ screws. The design of this support is done with *Fusion 360®*. There are not mechanical calculations about it, it was

just design to be practical and nice looking. FIGURE 44 shows the complete support structure.
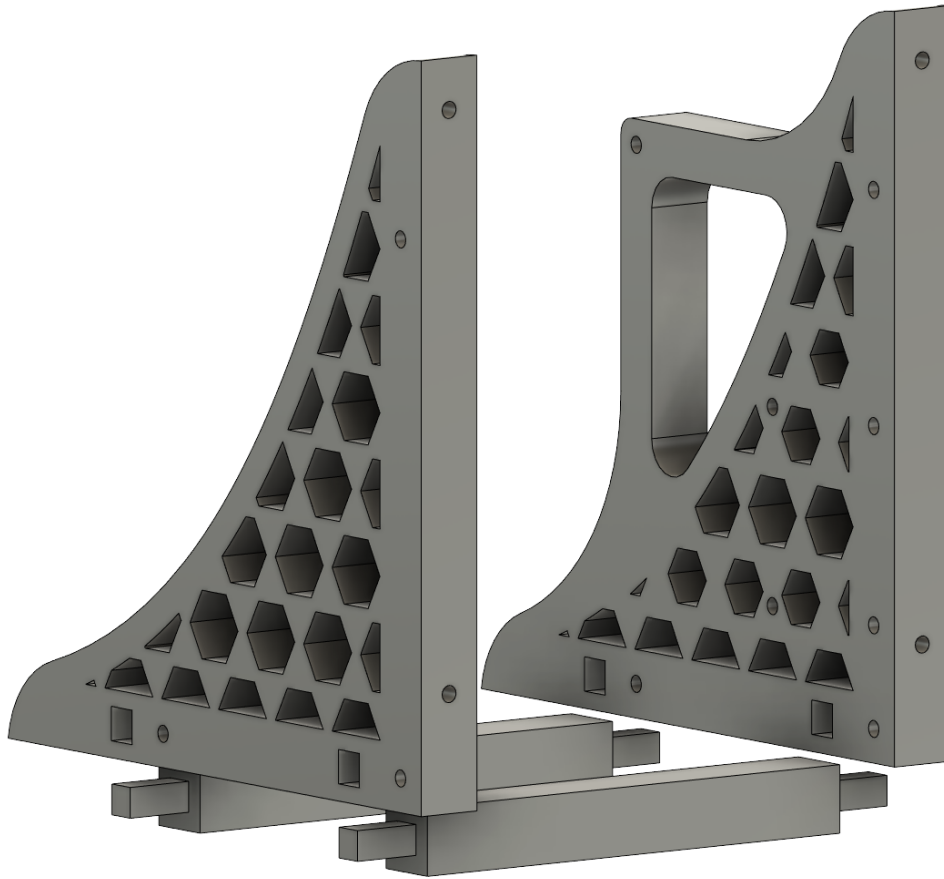
Figure 44 – 3D view of the solar simulator supports.

The holes on the two support front faces correspond to the mounting hole on the PCB. There are reinforcement bars on the bottom to provide the structure with more rigidity. The holes on the side of the supports are designed to screw the Raspberry Pi 3 on it and also to have the opportunity to attach a Raspberry Pi compatible touch screen. This support structure is done in four pieces to be easily 3D printed on a smaller 3D printer. Finally, the alveolar design is done to reinforce the structure stand in a more design looking version. This structure should be able to maintain the simulator and the heat sink in a stable standing position.

After many issues with 3D printers, the two standing supports have been laser cut into plastic sheets of 8 mm thick. As the previous design was done for 20 mm thick supports, the holes on the front faces have been more tricky to realise and have been done by hand with a drilling machine. The final result is less thorough but it will still do the job for which it was designed for.

6 Programming

For driving the LEDs, the PWM signals for each wavelength need to be monitored and controlled to match the expected extraterrestrial solar spectrum. To achieve this control, the *PCA9685* PWM generators are programmed via an I^2C interface with the Raspberry Pi 3 as the main computer. This programming is done with two dedicated python scripts. The first script allows an individual control of the PWM duty cycles and is used to calibrate the simulator. The second one is a test script which turns ON and OFF the simulator according to a design brief to simulate spatial conditions. These two scripts are turned into executable files and are run on the Raspberry Pi.

This Raspberry Pi is working under Raspbian OS and the scripts are written with a Python IDE called *Spyder®*.

6.1 Calibration script

First, to have the ability to control the PWM signals, the Raspberry Pi should be able to communicate with the PWM generators. This is done through the serial I^2C interface. This interface is initialized with the *board* and *busio* libraries using the following command:

```
i2c = busio.I2C(board.SCL, board.SDA)
```

It creates an I^2C instance linked to the SDA and SCL hardware pins [13]. This interface will be used to have a high level communication between the devices.

Second, two *PCA9685* instances are build using the *PCA9685* adafruit library [16]. It is done by specifying the address of the chip and the dedicated communication bus. The first PWM generator address is `0x40` while the second one is `0x41`. The code line below shows an example of such an instantiation for the first PWM generator:

```
PCA_A = adafruit_pca9685.PCA9685(i2c, address=0x40)
```

Third, the channels of each module are controlled individually with the included adafruit library. The frequency is set to $1kHz$ and the duty cycle is controlled from 0% to 100%. The duty cycle value range from 0 to $2^{12} - 1 = 4095$ as this value is coded on 12 bits. However, while running some tests, it appears that the maximal value did not lead to a 100% duty cycle ratio. With an iterative method, it appears that the PWM duty cycle ratio is coded on 16 bits unlike the datasheet specifications. Investigating resources on the subject did not lead to clear answers and one decided that it was a manufacturer mistake. The duty cycle ratio is thus controlled from 0 to $2^{16} - 1 = 65535$ which give us even more room for an accurate monitoring. The following piece of code shows the two mains commands used to access a given channel and to modify the PWM duty cycle on this channel.

```
led_channel_400nm = PCA_A.channels[0]
led_channel_400nm.duty_cycle = dc_ratio_400nm
```

The channel 0 of the first *PCA* is attributed to the control of the 400 nm LEDs through a new variable called `led_channel_400nm`. The same process is repeated for each wavelength according to the hardware wiring on the PCB. Having the controlled wavelength in the variable name allows a clearer access in the code to the wavelengths that need to be adjusted.

On top of this back-end programming, an user-friendly graphical interface has been designed to have a quick and easy access to the parameters. This graphical interface is done using the *Tkinter*® widgets. The easiest way to modify the duty cycle ratios was to use sliders. While sliding the cursor, it update the linked variable and the PWM duty cycle is thus modified for this specific channel. FIGURE 45 shows this interface.

As it can be seen, there is one slider per controllable wavelength. The *Apply changes* is a button that apply the changes done to the sliders only when it is pressed. It avoids having a fluctuating luminous flux while sliding up to the desired value. When the program starts, all slider values are set to 0. It allows to have an OFF phase when starting the simulator to adjust the parameters, etc. When the simulator is operating, as the luminous flux is very bright, the *Emergency OFF* button shutdowns every LED at the same time. In such a way, there is no need to slide all the values down to 0 to shut it down.

Figure 45 – Graphical interface of the calibration script.

The $AM0$ button sets all the duty cycles in a way that the output spectrum emitted by the LEDs matches the $AM0$ spectrum. On the first place, this button initializes the sliders at the theoretically computed values. Once the simulations will be done and that these values are fine-tuned, this button will updated such that it outputs the correct duty cycles. TABLE 4 shows the theoretical duty cycle percentages of each wavelength.

Wavelength	Duty cycle	Wavelength	Duty cycle	Wavelength	Duty cycle
400 nm	77.00 %	**530 nm**	60.67 %	**640 nm**	5.00 %
420 nm	70.00 %	**550 nm**	72.80 %	**660 nm**	67.67 %
430 nm	39.90 %	**567.5 nm**	70.00 %	**740 nm**	66.80 %
447.5 nm	60.67 %	**590 nm**	42.00 %	**810 nm**	47.60 %
460 nm	67.20 %	**599 nm**	56.00 %	**830 nm**	22.22 %
470 nm	49.00 %	**610 nm**	70.00 %	**850 nm**	56.00 %
485 nm	73.73 %	**620 nm**	70.00 %	**890 nm**	79.33 %
505 nm	71.40 %	**630 nm**	64.00 %	**940 nm**	47.60 %

Table 4 – Summary of the theoretical LED duty cycles to achieve the $AM0$ spectrum.

These values are obtained by using the following methodology for each wavelength:

- An equivalent current is calculated by multiplying the theoretical number of LEDs used in the simulation by the typical forward current of the LED.

- This equivalent current is divided by the number of LEDs used in the implementation of the PCB to obtain a current value per LED.

- This current value is divided by the maximal current available in the channel to get a duty cycle value in percentage.

This methodology gives rough estimations of the duty cycles but it is not accurate. Indeed, it would be accurate if the current vs. the radiative power is a linear relationship. As the LEDs are sometimes driven "far away" from their typical current, the radiative power will not be linear with the driven currents. However, as it can be seen from the duty cycle values, there is a lot of margin to adapt the radiative powers such that it matches our expectations. It was designed in a way that the theoretical background gives estimations and that the electronic design lets adjustment margins to counter balance the theoretical shortcuts done.

On the basis of these values, the script is used to calibrate the output spectrum to match the expected *AM0* one. By monitoring the spectrum generated by the solar simulator, one can match as best as possible the required spectrum by adjusting the luminous power at some wavelengths.

Indeed, as the duty cycle goes up, the mean current in the LEDs is higher and the luminous flux increases. Once these values are calibrated, they could be used for the real application script.

6.2   Countdown script

For assessing the efficiency of the photovoltaic cells, the solar simulator should run a script that mimics day/night periods. It means that the simulator should be ON, like the received solar spectrum, during the day time and OFF during the night time. The duration is a cycle of 60 minutes ON and 30 minutes OFF.

This cycle is done by switching the PWM duty cycles between their nominal values and 0 to switch it OFF. The script loads the values established with the calibration phase and use these as nominal values. The update between day and night phases is done with the pre-implemented `after()` function in python. This function is able to call another one after a given amount of time. For this purpose, there are two functions for day and night cycles, `ON()` and `OFF()`. The timer is updated and when it reaches the limited time, it calls the opposite function and loops this process.

On top of it, a graphical interface is build with the same *Tkinter®* widgets as the previous script. FIGURE 46 shows a picture of this graphical interface.
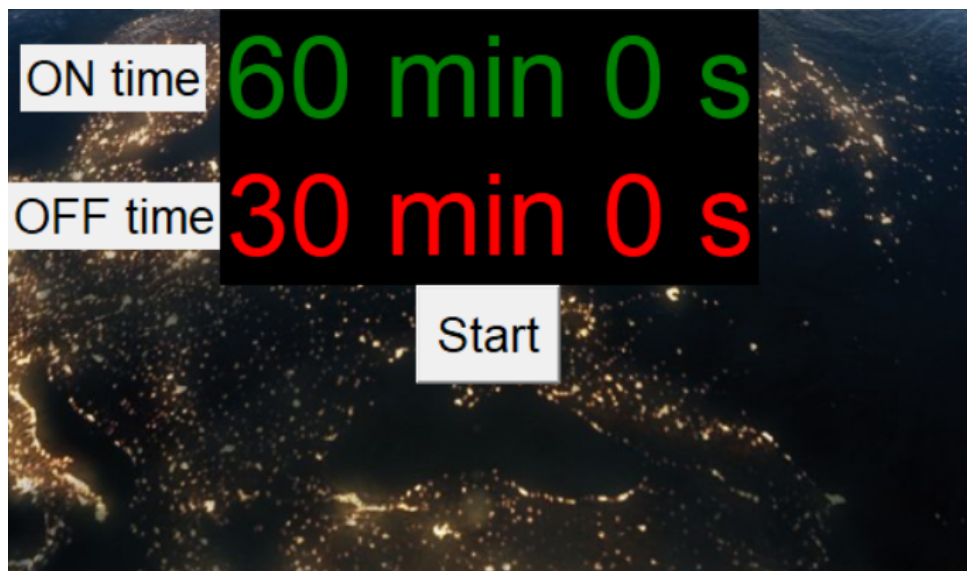

Figure 46 – Graphical interface of the countdown script.

As it can be seen, there is a manually implemented countdown to inform the user about the remaining *ON* or *OFF* time. The start button initiates the cycle. This script loops for ever and will be use for the testing of the photovoltaic cells.

7   Simulations

Once the design phase is done, it is time to build the solar simulator up and to test it. The first thing to do is to put the pieces together and then to assess some of its features.

7.1   Final assembly

After manufacturing the PCB, it has been assembled in the University of Liège by a technician as it required soldering under the microscope, welding flux, etc. The final printed circuit board with the components assembled can be seen on FIGURE 47.
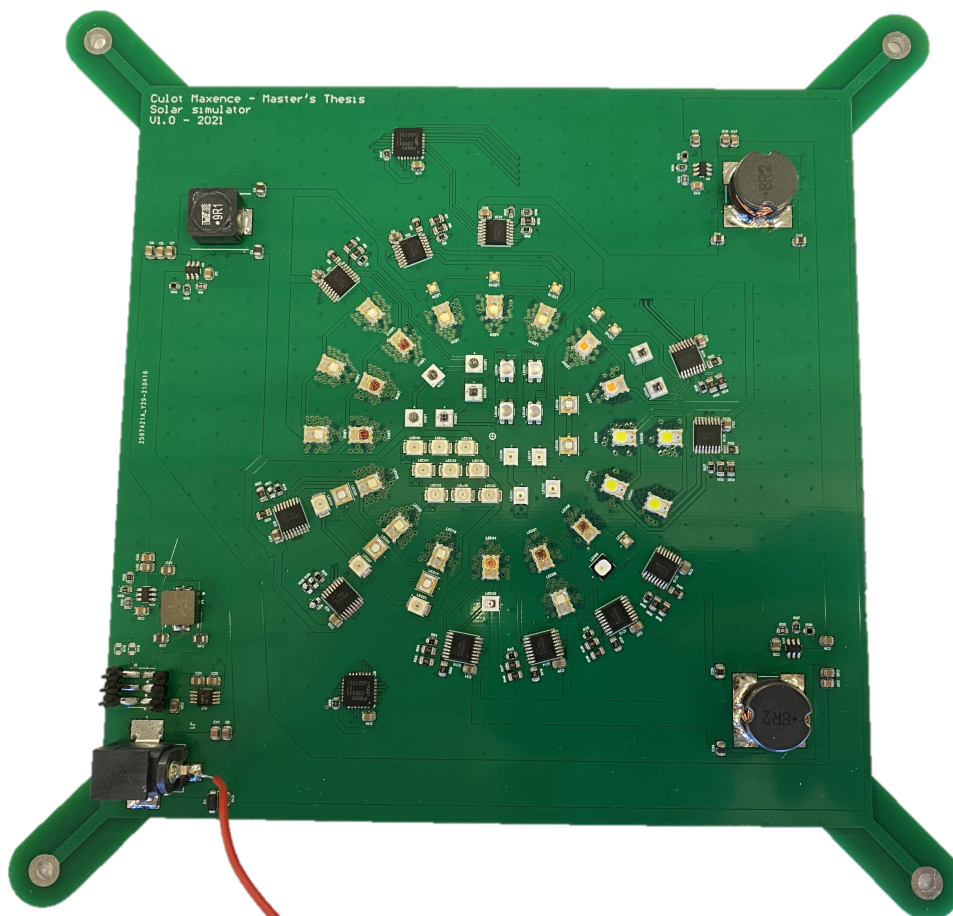
Figure 47 – Picture of the final assembled PCB.

Once the assembly was done, the PCB needed to be fixed to the 3D supports to maintain it in a standing position. The PCB is thus fixed to the supports with M3 screws. The touchscreen as well as the Raspberry Pi are fixed to the support side faces. Finally, the heat sink is attached to the back of the PCB with sticky silicon pads. FIGURE 48 shows the final implementation of the solar simulator prototype.
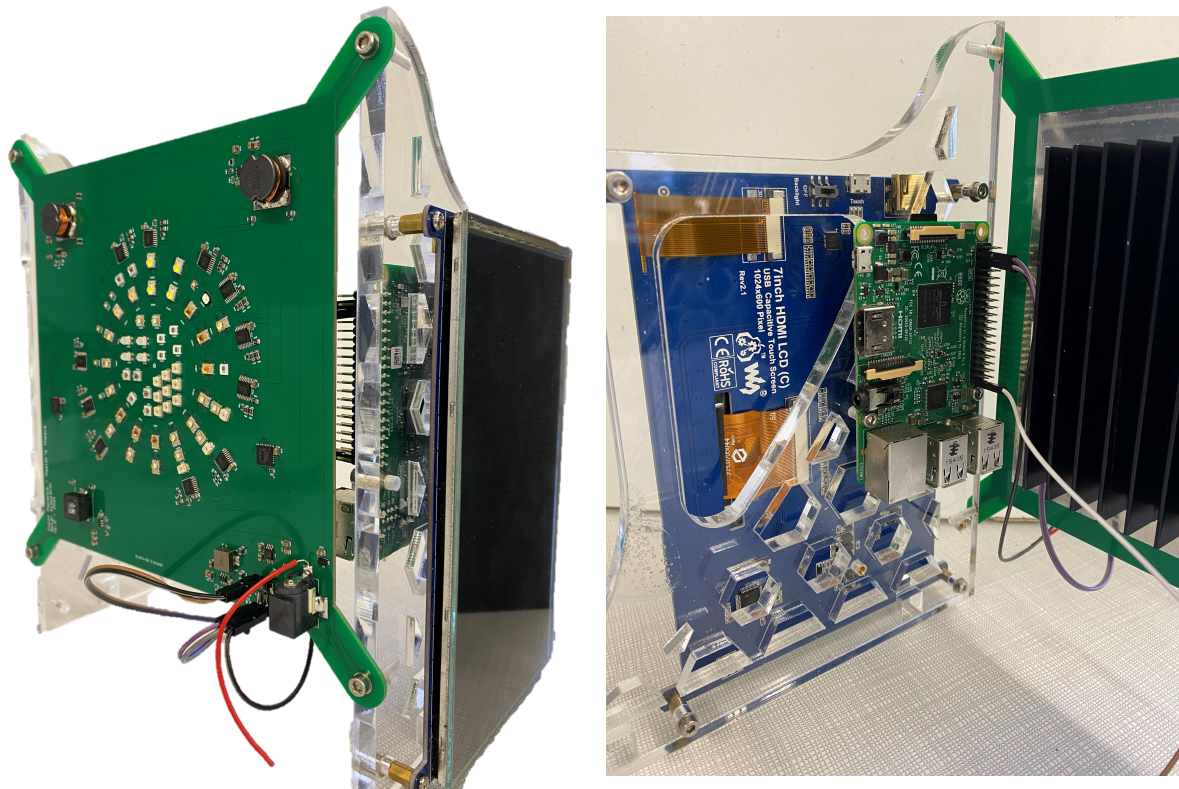
Figure 48 – Front and back views of the solar simulator on its support.

7.2   Tests and measurements

Before connecting the board to the Raspberry Pi, the voltage outputs of the DC/DC buck converters need to be assessed, especially the 5 V output that will power the Raspberry Pi. As the AC/DC charger was not available at the time the simulations were run, a red wire is soldered on the DC jack to power the board through a laboratory power supply. This power supply is calibrated to output a 12 V voltage source. With a voltmeter, the voltages at the outputs of the DC/DC converters are measured. TABLE 5 captures the measurements done.

	Expected voltage	Measured voltage
Buck converter *12V-10V*	*10 V*	*9.99 V*
Buck converter *12V-7V* (A)	*7 V*	*6.92 V*
Buck converter *12V-7V* (B)	*7 V*	*6.91 V*
Buck converter *12V-5V*	*5 V*	*5.00 V*
LDO *5V-3.3V*	*3.3 V*	*3.305 V*

Table 5 – Expected voltage outputs of the board vs. measured voltages.

As it can be seen, the DC/DC buck converter designs succeeded perfectly as they output the expected voltages. The light variations are due to the resistors that are not exactly equal to their

nominal values. However, these variations do not impact the proper operations of the simulator.

Once the voltage sources have been measured, the Raspberry Pi can be connected to the board and the real trials can be performed. The first thing to do is to check if the I^2C bus detects the two PWM generators. This operation is performed through the terminal using the following command:

```
sudo i2cdetect -y 1
```

The two PWM generators are detected on the first trial which enables us to start controlling the LEDs. Individually, each LED is powered ON at 25% of its maximal value, then powered at 100% of its maximal value. This process is repeated for each wavelength and allows a monitoring of the controlled LED. On the first place, it confirms that the LEDs are well powered when they emit light, and on the second place it confirms that they could be controlled through the duty cycles of the PWM signals. Indeed, the higher the PWM duty cycles, the higher the luminous flux of the LEDs.

FIGURES 49 and 50 show the LEDs being powered at their maximal luminous fluxes. As the luminous flux is filtered by the phone camera for the picture, it appears to be not that bright. However, in real conditions, it is blinding if a human directly observes it.
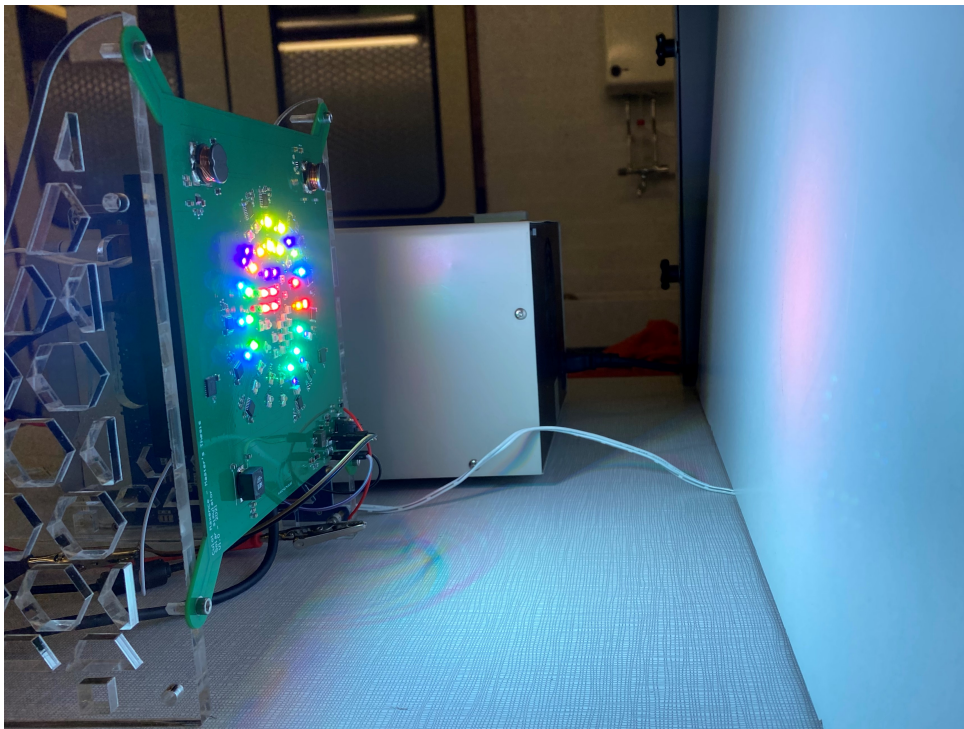

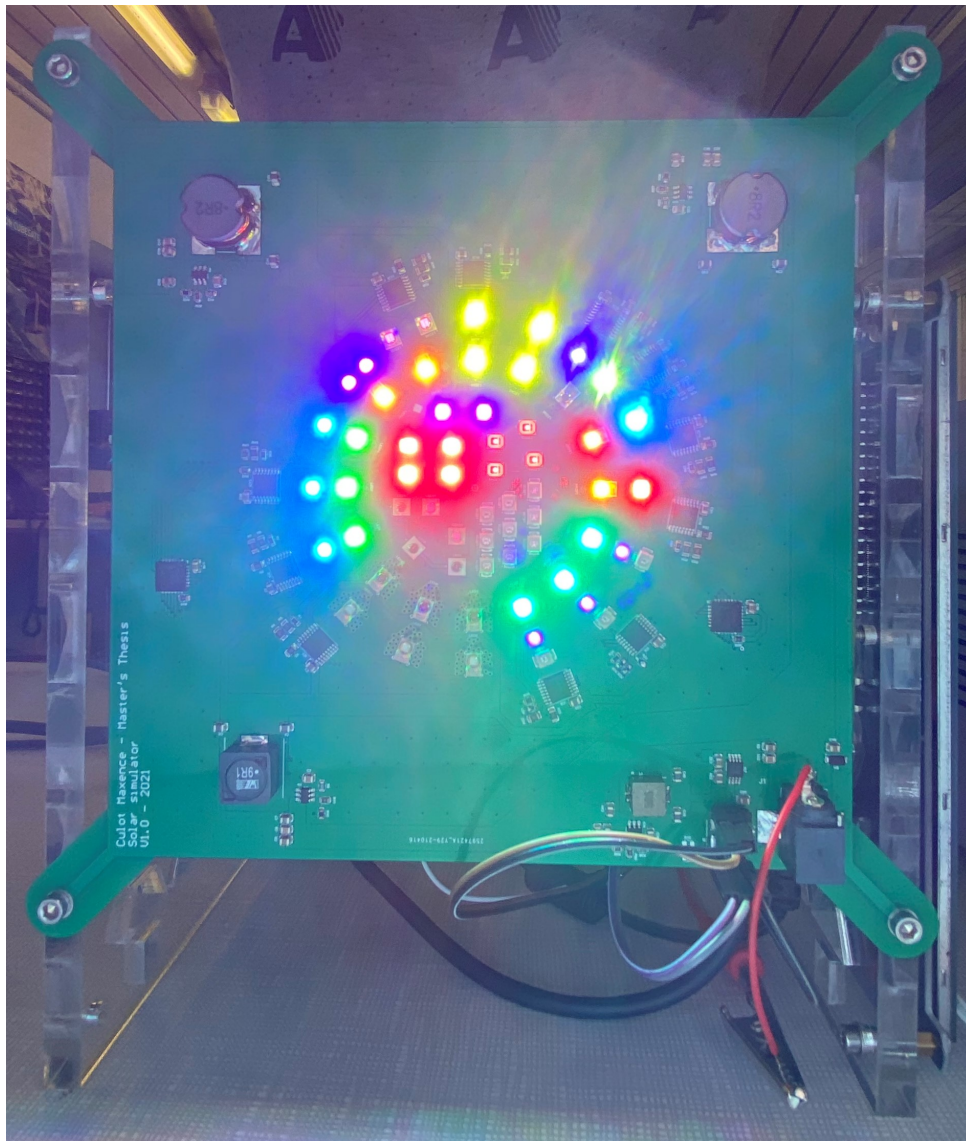Figure 49 – Application of the solar simulator on a white surface.

Figure 50 – Overview of the LED colors during the ON phase of the solar simulator.

As it can be seen, each LED has its own wavelength which gives luminous fluxes with different colors. The overall light that came out of the solar simulator appears to be white which is a good thing. Indeed, if the solar simulator outputs the solar spectrum, as all the wavelengths in the visible spectrum should be represented, the overall flux should appears white to the human eyes. It is not a prove of its correct operation but it is an appreciable start.

After testing each LED individually, some of them do not seam to work. As three of these are infrared LEDs, they can not be visually observed without optical cameras. A cheap way of assessing their operations is to use a smartphone lens. After using this hand-made technique, two out of the three infrared LEDs appear to work perfectly. One of these still do not work, as well as some others wavelengths. TABLE 6 outlines the non-operating LEDs.

Wavelength	Part number
447.5 nm	LXML-PR01-0500
505 nm	LXML-PE01-0080
610 nm	LXM2-PH01-0060
660 nm	LXM3-PD01
940 nm	SST-10-IR-B90-H940

Table 6 – Summary of the non-operating LEDs.

The LEDs are analyzed individually to characterize the reason of their non-operation. The three LEDs at 447.5 nm, 660 nm and 940 nm are driven from the same driver. This issue has been investigated with a voltmeter. It was measured that it is fed with the correct voltage. The logical conclusion is that the driver is broken due to a drop on the floor before soldering, a bad batch, etc. To solve this issue, the driver has to be unsoldered and be replaced with another one. As there was no time left for the technician to do it, this replacement could not be done. For the 660 nm LED, the issue is related to a bad soldering as the driver is perfectly working with other wavelengths.

Finally, the 505 nm LED is flashing when it is powered ON. It is due to the forward voltage of the LED that is too high. Indeed, its typical forward voltage is about 3.17 V but it can range up to 3.51 V. If the LEDs have a forward voltage in the typical range, 3x3.17 V< 10 V and the LEDs would work. However, if they have a higher forward voltage according to the LED batch, the sums of the three voltages will be higher than 10 V, the voltage power supply, and the LEDs will thus not work. This issue has been anticipated but not taking into account as it would be dependent on the LED batch. To solve this issue, one ordered more LEDs of the same 505 nm wavelength to pick to ones with the lower forward voltages. Another solution would be to change the feedback resistor ratio of the buck converter to set the voltage to 10.5 V which will be thus higher than the sum of the three forward voltages. Due to timing issues, there is no time to apply one of these solutions. However, the next people dealing will the simulator would have the opportunity to solve it easily.

Except these wavelengths, all the remaining LEDs appear to work as expected and are ready for the calibration procedure.

7.3   AM0 spectrum calibration

Unfortunately, due to delays in the PCB ordering and in the component assembly, the complete board was not ready early enough. It would have required a few more weeks to reserve the *Centre Spatial de Liège* spectrometer to calibrate the simulator. The future people that will work with the simulator would have thus to calibrate it before using it for assessment purposes. However, as the design was done to have a lot of margin while matching the spectrum, there is no doubt that the simulator will output the required spectrum.

8 Upgrades

Although there was not enough time to assess the effective operation of the solar simulator, there are some key features that could be improved or updated to enhance its future operation. As example, the following features can be implemented:

- To improve the spectral covering, LEDs with different wavelengths can be added to the board. The electronic design can be copied-pasted. Indeed, the DC/DC buck converters are oversized such they can supply a few more drivers and thus LEDs can be added without designing new voltage converters. The new drivers simply have to be connected to the free PWM channels of the *PCA9685*. A more specific and detailed research while allowing orders on unconventional suppliers can lead to the identification of LEDs with wavelengths covering the gaps existing in the solar simulator spectrum.

- The overlay layer of the PCB can be upgraded with a larger font size. Indeed, it was too small and it is thus barely readable. In addition, the wavelengths of the LEDs can be written next to the corresponding channels to have visual marks of the controlled wavelengths.

- The design error done could be corrected. Indeed, due to an error in the schematic, the 505 nm and the 890 nm wavelengths are controlled with the same channel on the PWM generator. Their duty cycles could thus not be controlled individually. Swapping the label on the schematic file would solve this issue.

- The graphical interface, although being practical and efficient, can be modified to have a better looking design. Moreover, a smartphone application would be more convenient to use and could be programmed through the bluetooth interface of the Raspberry Pi.

These upgrades are not critical for the effective operation of the simulator but they are appreciable features that do not required a lot of engineering to upgrade the simulator.

9 Conclusion

This Master's thesis aimed to engineer a LED-based solar simulator to assess the effective operation of photovoltaic cells used to power CubeSats. Based on a thorough analysis of the extraterrestrial solar spectrum, a technical solution have been implemented to replicate this spectrum. The final result is a working spectrum simulator prototype which outputs a spectrum that can be controlled to achieve the desired solar irradiance. It allows to replicate the solar spectrum density of 1369 W/m^2 with a high accuracy between 400 nm and 700 nm and up to 940 nm. Moreover, this spectrum is projected on an circular surface equivalent to *60x60* cm^2, largely enough to illuminate the *10x10* cm^2 CubeSat cells. This device will be used to guarantee that the photovoltaic cells operating in CubeSats would harvest enough power from the Sun during spatial missions to feed all the electronic power systems of the satellite.

In summary, the first step of this conception was to characterize the extraterrestrial solar spectrum to replicate it using LEDs. Once the spectrum have been characterized, a detailed research have been conducted about the components that would be suitable to build up this simulator. To achieve it, LEDs with various wavelengths have been used as well as LED drivers. In addition, several power supply converters have been designed to feed all the electronics constituting the simulator. These components are then interfaced together on a custom printed circuit board. Finally, a graphical user interface have been deployed to allow modifications of the spectrum emitted by the LEDs such that it matches the requirements of the photovoltaic cells assessment.

To conclude, this solar simulator is a new assessment device in the field of aerospace in the Liège area. It will be used for future developments and enhancements of the new *OUFTI* CubeSat versions. It will provide engineers with a simulation tool to validate their designs about the electric power supply of the CubeSat before launching it into space. As a final result, this device will contribute to the development of aerospace researches in the University of Liège.

Bibliography

[1] Solar radiation outside the earth's atmosphere. Oct 2020.

[2] T. Messadi A. V. Arecchi and R. J. Koshel. *Field Guide to Illumination.* 2007.

[3] Altium Academy. (1) thermal management - thermal resistance concept. *Youtube*, Apr 2020.

[4] Altium Academy. (3) thermal management - thermal resistance concept. *Youtube*, Apr 2020.

[5] ASTM. G173–03 standard am0 and am1.5 spectra. 2003.

[6] G. Boisclair. Cône d'ouverture, calotte sphérique et angle solide. *Le monde en images*, 2016.

[7] The Editors of Encyclopaedia Britannica. Astronomical unit. *Encyclopedia Britannica*, Aug 2018.

[8] The Editors of Encyclopaedia Britannica. Blackbody. *Encyclopedia Britannica*, Mar 2021.

[9] C.B.Honsberg and S.G.Bowden. Photovoltaics education website. 2019.

[10] Asim Kumar Roy Choudhury. *Principles of colour and appearance measurement: Object appearance, colour perception and instrumental measurement.* Elsevier, 2014.

[11] Cree. Optimizing pcb thermal performance for cree xlamp leds. *Digi-Key Electronics*, Dec 2010.

[12] Altium Designer. Thermal via management for pcbs. 2021.

[13] T. DiCola. Circuitpython basics: I2c and spi. Sep 2017.

[14] P. Dillmann. Via tenting for pcb design. *Macrofab*, Apr 2018.

[15] Diodes Incorporated. *AL1783: 60V three independent PWM channel linear LED driver*, Nov 2017. Rev. 2-2.

[16] B. Earl. Adafruit pca9685 16-channel servo driver. Oct 2012.

[17] Power Electronics. Thermal management and heatsink design. *Youtube*, Jan 2018.

[18] Green Rhino Energy. Extraterrestrial irradiation: The solar constant. 2016.

[19] Raspberry Pi Foundation. Raspberry pi documentation. 2021.

[20] Frank P. Incropera. *Foundations of heat transfer*. J. Wiley, 6th ed., international student edition, 2013.

[21] Inolux. *IN-3531SCUV: High Power UV LED*, Sep 2016. Rev. 1.3.

[22] Texas Instruments. Thermal design of bottom side cooled devices. May 2019.

[23] Texas Instruments. Webench power designer. 2021.

[24] A. Kher. Perihelion, aphelion and the solstices. *Timeanddate.com*, Jun 2010.

[25] Littelfuse. *SMF12CA: TVS Diodes*, May 2020.

[26] Lumileds. *LXM*-****-****: DS68 LUXEON Rebel Color Line*, 2017.

[27] Lumileds. Luxeon rebel altium designer library model. Apr 2019.

[28] Lumileds. *L128: DS236 LUXEON 2835 Color Line*, 2020.

[29] Lumileds. *L1CU: DS198 LUXEON CZ Color Line*, Jan 2021.

[30] Luminus. *SST-10-IR: Surface Mount Series Infrared LED*, 2016. Rev. 2.0.

[31] LEDs magazine. Radiometric and photometric terms. Apr 2005.

[32] MechaTronix. Calculation of the required led cooler thermal resistance rth. 2021.

[33] S. Mohan. Thermal comparison of fr-4 and insulated metal substrate pcb for gan inverter. *Texas Instruments*, Jun 2019.

[34] MyHeatSinks. Thermal interface material basics. 2021.

[35] R. Nave. Thermal conductivity. *Georgia State University*, 2017.

[36] Nichia. Optical unit and calculation. Mar 2016.

[37] NXP Semiconductors. *PCA9685: 16-channel, 12-bit PWM Fm+ I2C-bus LED controller*, Apr 2015. Rev. 4.0.

[38] Government of Canada. Interactions with the atmosphere. Mar 2016.

[39] OSRAM Opto Semiconductors. *LCG H9RM: OSTAR Projection Cube*, May 2012. Rev. 1.0.

[40] J-J. Rousseau. Notion d'angle solide. 2018.

[41] On Semiconductor. A quick pcb thermal calculation for power electronic devices with exposed pad packages. Oct 2017.
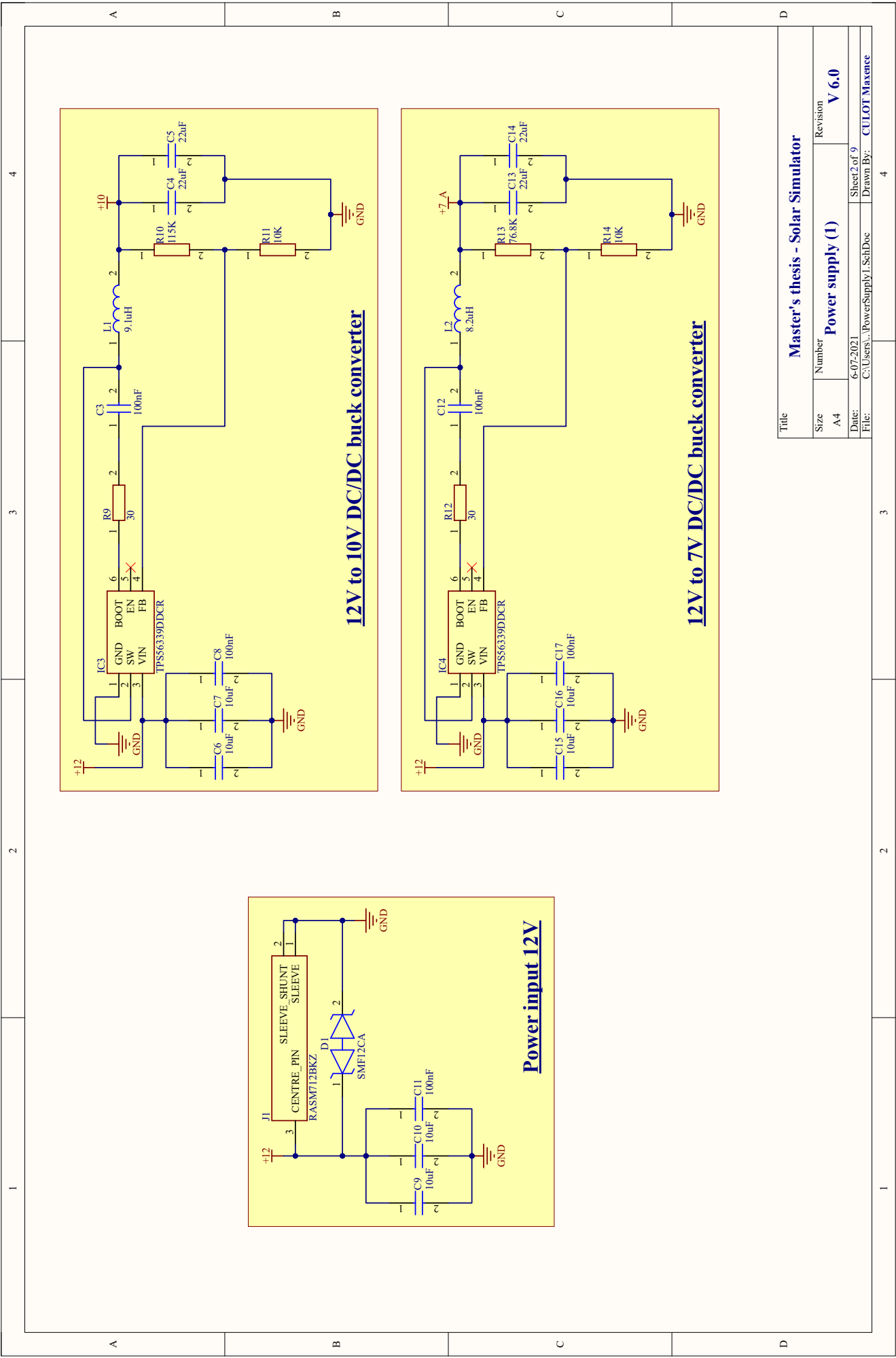
[42] OSRAM Opto Semiconductors. External thermal resistance substrates - led fundamental series. *Youtube*, Jun 2012.

[43] R. Simon. Earth's energy budget. *Nasa Earth observatory*, Jan 2009.

[44] American Chemical Society. Energy from the sun. 2021.

[45] V. Spataro. Counting squares: A method to quickly estimate pcb trace resistance. *EDN*, Apr 2013.

[46] A. Steinwolf. Random vibration testing beyond psd limitations. *University of Auckland*, Sep 2006.

[47] Switchcraft. *RASM712BKZ: DC Power Connectors BKZ Locking Series 2.5mm Surface Mount*, Aug 2018. Rev. A.

[48] Texas Instruments. *TPS56339: 4.5-V to 24-V Input, 3-A Output Synchronous Buck Converter*, May 2019. Rev. A.

[49] Texas Instruments. *TPS7B81: 150-mA, 40-V, Ultra-Low-IQ, Low-Dropout Regulator*, Apr 2020.

[50] A. Thorseth. Characterization, modeling, and optimization of light-emitting diode systems. *Department of Photonics Engineering*, page 135, 2011.

[51] M. Tonneau. Le premier satellite 100% wallon, oufti-1, est prêt pour l'espace. Nov 2013.

[52] TT Electronics. *PCF0805-13-24KBT1: Precision Thin Film Nichrome Chip Resistors*, Jul 2019.

[53] Humboldt State University. Atmospheric scattering. 2019.

[54] Humboldt State University. Blackbody radiation. 2019.

[55] Vishay Semiconductors. *VSMG2700: High Speed Infrared Emitting Diode*, Sep 2013. Rev. 1.4.

[56] Vishay Semiconductors. *VSMF3710: High Speed Infrared Emitting Diode*, Nov 2015. Rev. 2.0.

[57] A. M. Winkler. Gaussian kernels: convert fwhm to sigma. Aug 2011.

[58] Würth Elektronik. *150141SS63140: WL-SMTD SMT Mono-color TOP LED Diffused Dome*, Nov 2019. Rev. 2.0.

[59] X. Yang. An alternative lifetime model for white light emitting diodes under thermal electrical stresses. *EDN*, May 2018.

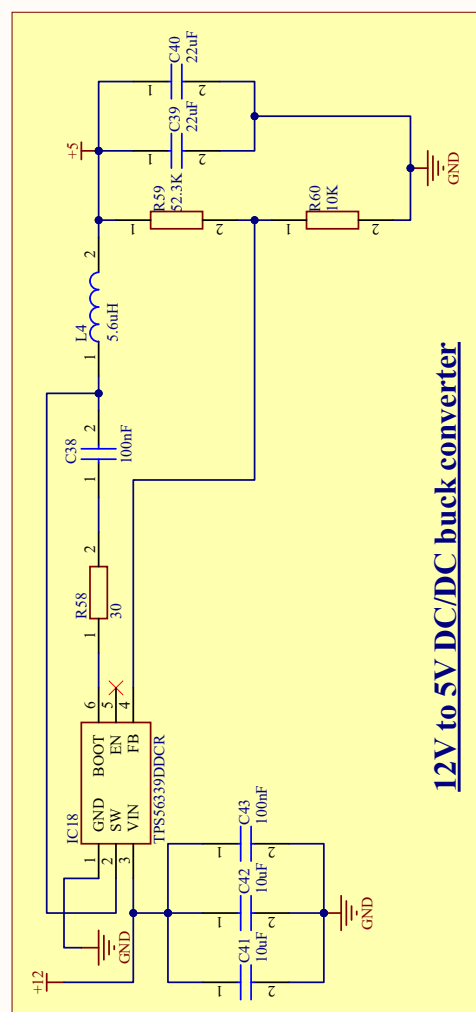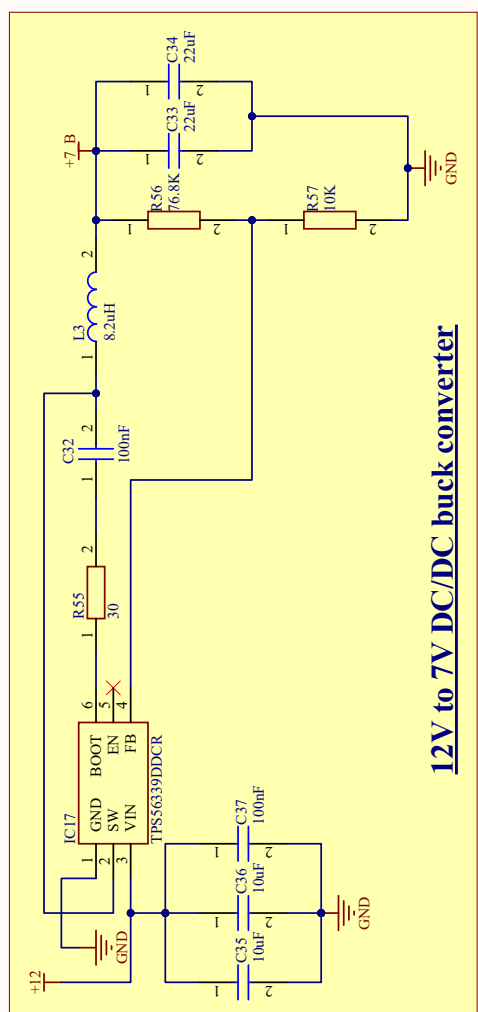[60] B. Yarborough. Temperature coefficient of resistance for current sensing. *Vishay*, May 2020.

A   Appendix: Schematics

12V to 10V DC/DC buck converter

12V to 7V DC/DC buck converter

Power input 12V

B) PWM CTRL (1000001)

R18 10K
+3.3
OE_B

C19 10uF
GND
+3.3
SCL
SDA

IC6
PCA9685BS,118

EXTCLK
A5 OE
LED15
LED14
LED13
LED12
LED11
SCL
SDA
VDD
A0
A1
A2
EP
A3
A4
LED0
LED1
LED2
LED3
LED4
LED10
LED9
LED8
VSS
LED7
LED6
LED5

LED8A LED5A LED6A LED7A
LED3B LED4B LED5B LED0B LED2B
LED1B LED6B LED7B
GND

A) PWM CTRL (1000000)

R17 10K
+3.3
OE_A

C18 10uF
GND
+3.3
SCL
SDA

IC5
PCA9685BS,118

EXTCLK
A5 OE
LED15
LED14
LED13
LED12
LED11
SCL
SDA
VDD
A0
A1
A2
EP
A3
A4
LED0
LED1
LED2
LED3
LED4
LED10
LED9
LED8
VSS
LED7
LED6
LED5

LED0A LED1A LED2A LED4A LED3A
LED15A LED14A LED13A LED12A LED11A
LED10A LED9A
GND

Header Raspi

+3.3
R16 1K
R15 1K
SDA
SCL
J2
15-91-3060
OE_A
OE_B
+5
C20 10uF
GND

LDO 5V to 3.3V

IC7 TPS7B8133DGNR
IN
EN
NC_1
GND_1
OUT
NC_2
GND_3
GND_2
EP
+3.3
C22 10uF
+5
C21 10uF
GND

Master's thesis - Solar Simulator

LED drivers (3)

V 6.0

CULOT Maxence

Master's thesis - Solar Simulator

LED drivers (4)

Revision
V 6.0

Title

Size: A4
Number
Date: 6-07-2021   Sheet 8 of 9
File: C:\Users\MC\Desktop\PCB_V4\Main4.SchDoc   Drawn By: CULOT Maxence

LED47 L128-FRD1003500000
LED52 L128-FRD1003500000
LED46 L128-FRD1003500000
LED51 L128-FRD1003500000

+7_A
+7_B

IC12 AL1783T16E-13
LED3 LED2 LED1 NC_2 FAULTB REF3 REF2 REF1
VIN NC_1 PWM1 PWM2 PWM3 GND_1 GND_2 GND_3
EP

LED0B
C27 10uF
GND

+3.3
R35 1K
R39 24K
R38 24K
R37 24K
GND

LED50 SST-10-IR-B130-K850-00
LED55 SST-10-IR-B130-K850-00
A K EP
GND

LED49 15014SS63140
LED54 15014SS63140
LED48 15014SS63140
LED53 15014SS63140

+7_A

IC13 AL1783T16E-13
LED3 LED2 LED1 NC_2 FAULTB REF3 REF2 REF1
VIN NC_1 PWM1 PWM2 PWM3 GND_1 GND_2 GND_3
EP

+7_A
LED14A
LED15A
C28 10uF
GND

+3.3
R36 1K
R40 24K
R41 75K
R42 75K
GND

Master's thesis - Solar Simulator

Revision
V 6.0

LED drivers (5)

Number

Title

Size
A4

Date: 6-07-2021   Sheet 9 of 9
File: C:\Users\MC\Desktop\PCB V4\Main5.SchDoc Drawn By: CULOT Maxence

B Appendix: 3D view of the PCB