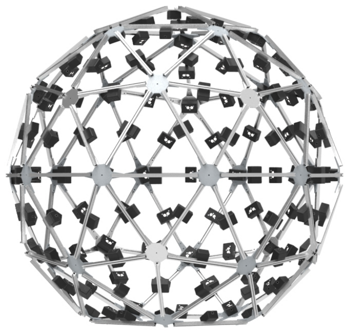


KTH Shine Product Description

SARAH ABELIN
EDVARD COLLIN
PEDRO ESCRIBANO
FELIX FRIE
MIKKO KIESI
PATRIK LEVIN
FRIDA LINDBERG
SEBASTIAN QUIROGA
VICTOR SUNDIN
FRIDA WIKMAN

MF2059 Mechatronics Higher Course Project
Supervisor: Björn Möller
Examiner: Martin Grimheden

Abstract

This project is part of the Mechatronics, Advanced Course, at KTH (The Royal Institute of Technology). The project assignment was given by Pixel Grinder, a company working with face and body scans to create 3D models for the film and game industry. Pixel Grinder has requested a controllable light rig to use during scanning. The purpose of this report is to describe the developed prototype and all decisions taken along the iteration line.

In order to perform the 3D scans, Pixel Grinder uses a technique called photogrammetry that utilises pictures taken with cameras from different directions from the object. To achieve ideal scanning results, the cameras should take five consecutive pictures with different lighting conditions: one picture with polarised ambient light, one with unpolarised ambient light and three picture with gradient light from left to right, top to bottom and front to back. The total amount of light intensity and distribution should also be sufficient for the photogrammetry software to work its best. The designed light rig should be robust, modular and easy to assemble and disassemble to enable travelling with it.

To achieve this, a mechanical structure in the shape of an icosphere has been designed. A total of 153 LED units, with an intensity of 2000 lumen each, are distributing light from every vertex and every edge of the icosphere. To be able to produce all the required lighting conditions, each of the LED units are individually controllable, synchronised with the camera shutters and lit for 5 ms during each of the five shots. The cameras are placed on the rig in between the LED units.

On the light rig, there are 41 micro controllers communicating trigger signals and intensity settings to 3-4 LEDs each. The micro controllers in turn are controlled by a power and data distribution unit, which also supplies the system with power. The logic in the distribution unit is sent by a master unit, from which the synchronisation signals to the cameras also are sent and the communication with the operator's PC is established.

The resulting prototype proved to function in a satisfactory manor. The Pixel Grinder team did some initial tests with the whole system and the results looked promising. With some additional adjustments, like for example optimising the synchronisation and light intensity settings, the company believes that the final result will be of great contribution to their core business.

Preface

During this project, the main supervisor from the Royal Institute of Technology (KTH) has been Björn Möller. He has participated in weekly meetings, contributed with valuable inputs as well as taking charge of the majority of the administrative concerns. For this, the project group is very thankful. The inputs and help from the rest of the staff - Staffan Qvarnström, Tomas Östberg, Martin Edin Grimheden, Jad El-Khoury and Mikael Hellgren - have also been very valuable.

As for the hosting company, it has been an overall pleasure to work, discuss and conclude together with Sigtor Kildal, Nils Lerin, Jonas Törnqvist and Mitra Ashkan Far at Pixel Grinder. Weekly Skype meetings have resulted in a continuous, stable contact and exchange of ideas and knowledge. The project group appreciates the confident that the company has chosen to put into the project group and hope that the prototype will be proven useful in the future.

Special thanks are also to be given to Jan Sköld at Frikab and to the staff at LP Innovation for fast and thoughtful work contributing to the project.

Project group "Shine", Stockholm, 2015-12-16

Contents

Acronyms

CAN Controller Area Network.

CCU Camera Controller Unit.

CSS Cascading Style Sheets.

GND Ground.

GPIO General-Purpose Input/Output.

GUI Graphical User Interface.

HTML HyperText Markup Language.

JSON JavaScript Object Notation.

KTH The Royal Institute of Technology.

LCD Liquid-Crystal Display.

LCU LED control unit.

LED Light Emitting Diode.

MCU Micro Controller Unit.

NAS Network-attached-storage.

NC Normally Closed.

OS Operating System.

PCB Printed Circuit Board.

PDDU Power and Data Distribution Unit.

PDSU Power and Data Splitting Unit.

PWM Pulse Width Modulation.

SPI Serial Peripheral Interface.

SQL Structured Query Language.

TCP Transmission Control Protocol.

UI User Interface.

USB Universal Serial Bus.

Chapter 1

Introduction

This chapter presents the background to the project, the purpose of the report and the methods for collecting information about the project.

1.1 Background

The project background is stated from two perspectives.

1.1.1 Course Perspective

This project is part of the Mechatronics, Advanced Course, at KTH (The Royal Institute of Technology). The project is split between two sequential courses called Mechatronics, Advanced Course Spring Semester (MF2058) and Mechatronics, Advanced Course Fall Semester (MF2059). However, these courses run the same projects as Mechatronics, Business and Management (MF2050). Hence, project members come from different courses.

Overall, the Mechatronics Advanced Course acts as the capstone course for the Mechatronics track of the Integrated Product Development academic masters program. Therefore, the aim of the project is to bring together the different disciplines of the major. For integrated industrial collaboration, the project is done on behalf of a company situated in the business world.

1.1.2 Company Perspective

The company Pixel Grinder, located in Uppsala, Sweden, is the industrial client for the project. The core business of Pixel Grinder is to capture images of actors and actresses, and by the use of photogrammetry, generate 3D-models of those. For the models to be detailed enough, several cameras are required to capture images simultaneously from several angles. Pixel Grinder are currently doing face scans as well as full body scans, two processes that require different camera and light setups. During this project, two products are to be developed. The primary priority is a light rig for facial scans. It is believed that with better light conditioning, less cameras will be needed to generate a high quality 3D-model. The solution needs to be modular and packable in cases of which the size and weight do not exceed maximum airline baggage regulations, since Pixel Grinder travels to clients abroad using commercial airlines. The solution should also be scalable, for future implementation of the full body scan. The secondary priority is an aim and focus controlling system for the cameras used for full body scans. It is believed that with such system, the process from putting up the rig to taking the pictures, will be significantly faster. This will decrease the total time spent with the actors and actresses, thus saving expenses for the company.

1.2 Purpose

The purpose of this report is to present the demonstrator and explain the design decision that have been taken during the project of the MF2058 Mechatronics Advanced Course Fall, as well as the basis developed during the spring semester.

1.3 Scope

Pixel Grinder has requested a controllable light rig that is synchronised with their cameras. The requests made by Pixel Grinder and the course aims of MF2058 have been the basis for development of the Requirements (see appendix A) of the project, which define the scope of the project.

The requirements have been divided into two phases - one concerning the light rig and the other one concerning the control of the cameras. The first phase has been considered mandatory while the second phase has been considered additional and is mainly brought up for discussion. The requirements defining the scope can be summarised as follows:

- The rig shall provide sufficient lighting during five synchronously taken photos with 5 different lighting conditions within one second.
- The intensity of every LED shall be controlled individually. Both polarised and unpolarised light should be provided. The lighting conditions should fulfil the requirements needed for producing high quality 3D models.
- The rig shall be modular and possible to disassemble and fit into one or more Peli 1650 cases. As part of the modularity requirement, the equipment should be easy to replace in case of failure.

1.4 Method

The project team consists of ten technology students, of whom some have been assigned certain responsibilities. Details of project roles, along with an overall description of the chosen agile project management method, can be seen in appendix L.

The group has agreed on and worked according to the developed code of conduct, see appendix N. Weekly meetings have been held with company representatives and with supervisors at KTH. To develop and design different ideas and implementation of a model, concurrent engineering has been utilised.

All prototypes have been designed in the laboratories of KTH, with a few exeptions. The comprehensiveness of the project have led to outsourcing the manufacturing of some of the parts to specialised firms. This was done with parts that either where large in numbers or required special tools to be manufactured.

One major part that was outsourced was the mechanical structure, as it required special tools for manufacturing. Pixel Grinder has a developed relationship with a prototyping firm called LP Innovations located close to their office in Uppsala, and they were hired to manufacture the mechanical rig. The main area in which outsourcing was utilised, however, was the manufacturing of printed circuit boards. In total 202 PCBs of five different designs are used in the final prototype. The quality that it is possible to achieve with the tools that are available at KTH is not deemed sufficient to fulfil the requirements. During the selection of PCB manufacturer the customer support, delivery time and flexibility was of greatest concern. After evaluating 9 different companies Frikab, situated in Bromma, was selected for the manufactoring of PCBs.

Chapter 2

Design Decision Basis

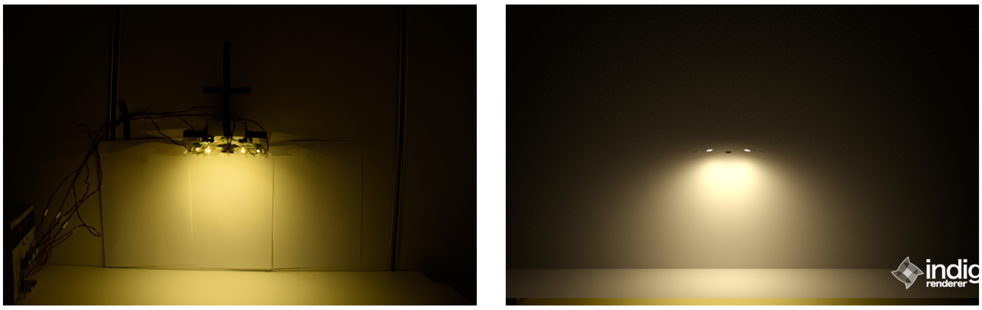
This chapter presents the plan of design for the light conditions, mechanical structure, hardware, casing and software implementations. Why certain choices have been made is discussed and the verifications and validations of these choices are presented.

2.1 Light conditions

One of the first parts that needed to be verified and validated were the light simulations done during the spring term. These simulations are the foundation of the design basis, which implies that any changes in these values affect all the other components in the system, including the power distribution, the geometric structure, the type of micro-controller units and more.

In order to verify our light simulations, a test case was implemented. The test case consisted of two different tests. The first test aimed at verifying the validity of the light simulation results. This was done by comparing a real image (using LEDs of which the specified values were known) with the light simulation results (using the same values). This validated that results provided by the simulation hold in reality. During the second test, a comparison between a light distribution profile in the light dome simulation to one without a light distribution profile was done. The results and data from the test can be seen in the images below.

In the first test, 5 light sources where used on each image. The lights were aligned in a pentagon form from the light dome design. The value for the rated settings was 180 lm for each light. The camera settings were set to a shutter speed of 1/80, an ISO value of 100 and f/stop of 3.5. The real image can be seen on figure 2.1a. The simulation was done with the same settings for the camera and the same light distribution profile for the light sources. The result can be seen in figure 2.1b. From the images it is possible to conclude that the light intensity and distribution was very similar in both images. The only difference was the colour, which was due to the white balance being difference since it was not possible to use the same white balance in Blender as on the real camera.



(a) Light photo for verification.

(b) Light dome simulation with light distribution profile.

Figure 2.1: Light simulations for verification.

In the second test, the light dome simulation used during the spring was used to compare the difference in light intensity and distribution with and without a light distribution profile for the already decided light values of 4000 lm per unit. The camera settings used during the test were set to a shutter speed of 1/200, an ISO value of 100 and f/stop of 14. The image without a light distribution profile can be seen in figure 2.2a and the image with a light distribution profile can be seen in figure 2.2b. As can be seen in the pictures, the light was more intense on the image with the light distribution profile. What this means is that the possibility to use less power-full LED units is available. However, since one of the LEDs has a polarisation filter in front of it which reduces the light intensity by half, the intensity will be correct.

(a) Light dome simulation without light distribution profile.

(b) Light dome simulation with light distribution profile.

Figure 2.2: Light dome simulations.

2.2 Mechanical design

As stated in the spring report, the mechanical rig needs to fulfil certain criteria stated by the company concerning distribution of light, modularity, stability and weight. In order to distribute the light, it was decided that the structure should be shaped as a 2V geodesic dome, see figure 2.3. Left to determine was how the rods in the structure should be attached, how the nodes were to be designed and how the whole dome should be placed on the ground.

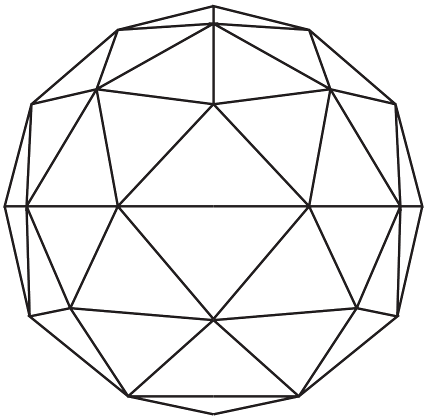


Figure 2.3: The outline structure of a 2V geodesic dome.

Several options for rods and fastenings were considered, and the designs were continuously discussed with the company. For the different iterations, focus was held at:

- Enabling fast and easy assemble and dissemble.
- Having parts that are easily packed and transported.
- Reducing total weight of the mechanical structure.
- Securing stability and robustness.
- Enabling PCB attachment in both nodes and rods.
- Enabling for neat and simple cable attachment along the outline of the structure.
- Keep the price of manufacturing the structure low, e.g. by having simple designs and by using standard profiles.

After several ideas on types of rods and nodes, the final iteration resulted in having bent aluminium sheet metal as nodes and standard Norcan N 0805 beams as rods. This design will keep both the cost and weight to a minimum, and provide a safe,stable and modular structure. The rods are attached to the nodes with adjustable screws that will not have to be completely detached every disassemble.

The complete sketches of the nodes and rods can be seen in appendix F. A virtual prototype of the design can be seen in figure F.2.

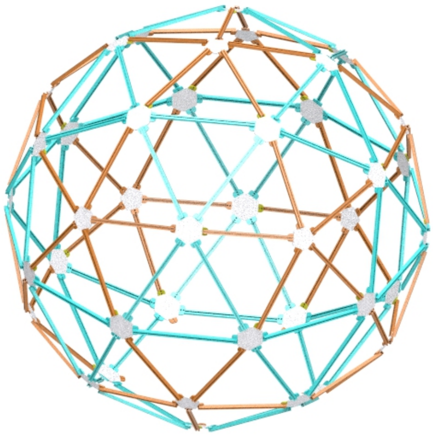


Figure 2.4: The final design of the mechanical dome, consisting of nodes properly in place and with short (red) and long (blue) rods.

In order for the object to enter the dome, one node and its rods have to be temporarily removed. This is going to be solved by having magnetic locks, as shown in figure 2.5.Here, the rods from the dome to the floor are also shown.

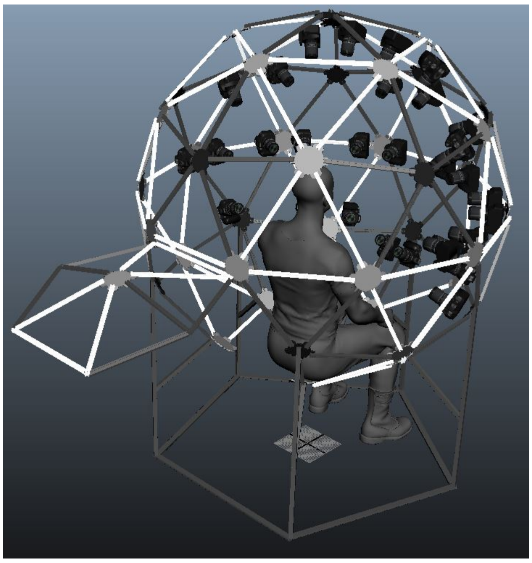


Figure 2.5: A model of the dome and ground fastenings with a person sitting inside and the dome door opened.

2.3 Hardware design

This section covers the hardware present in the project. Requirements for the subsystems and the iterations for the designs are explained.

2.3.1 System overview

Concerning hardware content, this project handles 153 LED units, 41 LCUs, seven PDSU (Power and Data Splitting Unit)s, one master unit and one power supply. The hardware subsystems' interconnections and logical placement can be seen in figure 2.6.

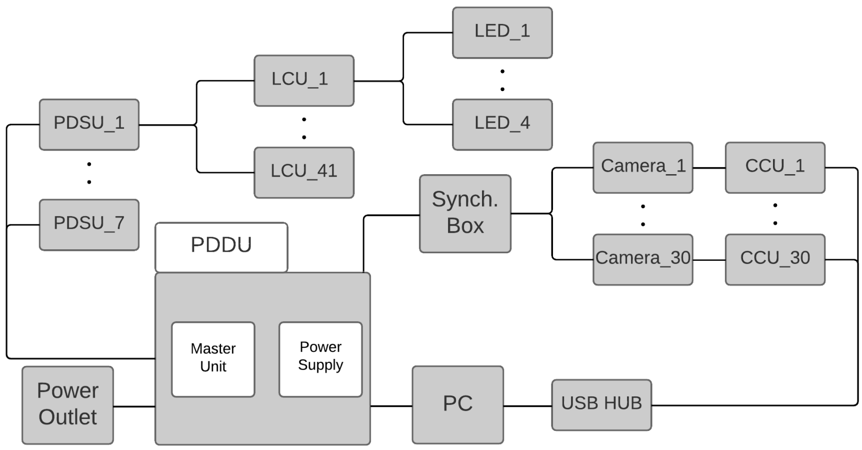


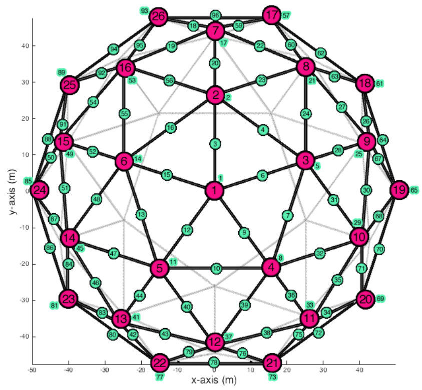
Figure 2.6: An overview of the hardware comprised in the project.

The PDDU (Power and Data Distribution Unit) supplies the whole system with power (48V and 20 amp) and the PDSU divides the current across the dome, making sure that no current limits are broken in wires or devices. The master unit (placed in the PDDU) communicates CAN (Controller Area Network) to all LCUs and synchronisation signal to the synchronisation box, which forwards the signal to the cameras. The LCUs are connected to three or four LED units and supply them with power, a PWM (Pulse Width Modulation) signal and a trigger signal. The PDDU unit also provides its master unit with 5V and the LCUs with 12V.

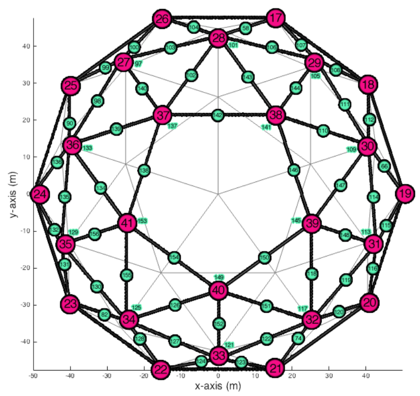
More details on the specific devices (decisions concerning the design, iteration etc.), cabling between them, cases and fastenings are stated in following sections.

2.3.2 LED logistics

To get an even distribution of light in the rig, a set of LEDs were placed in every node of the geodesic dome as well as in the middle of every rod. This resulted in a total of 153 LED units. These units are controlled by 41 LCUs placed on every node of the rig. To get a good overview of all the LCUs and their physical placement they were numbered from 1-41, placing number one on top of the rig and then letting the LCU numbers increase in a circular pattern downwards along the rig structure. Every LCU is capable of controlling four LED units. However, as there are 41 LCUs and 153 LED units, not all LCUs need to control four each. To follow the geometric structure as much as possible, the controlling of the LED units was divided as follows: the first LCUs only controls the LED unit placed on that node, LCU 2-6 control three LED units each and the rest of the LCUs all control four LED units each. The LED units are indexed with a number from 1-153. A table on which LCUs control which LED unit is presented in appendix E. The naming and physical placement of the LCUs and LED units is demonstrated in figure 2.7a for the upper half and in figure 2.7b for the lower half.



(a) Light rig seen from above, showing the placement of the LCUs and LED units on the upper half of the dome.



(b) Light rig seen from above, showing the placement of the LEDs and LCUs on the lower half of the dome.

Figure 2.7: The light rig seen from above.

2.3.3 LED unit hardware

In this section the hardware choices, interfaces and the final performance of the LED units are discussed.

According to the project requirements presented in appendix A each LED unit has to to provide both polarised and unpolarised light. These are to be provided by two different LEDs .The polarised and unpolarised LEDs have to provide intensities of 4000 and 2000 lumen correspondingly and the user has to be possible to adjust the intensities.

During the spring it was decided that the intensities would be adjusted by driving the LEDs using PWM. The functionality of this solution was verified by taking photos of an LED that was driven with different PWM-dutycycles. The results are shown in figure 2.8.

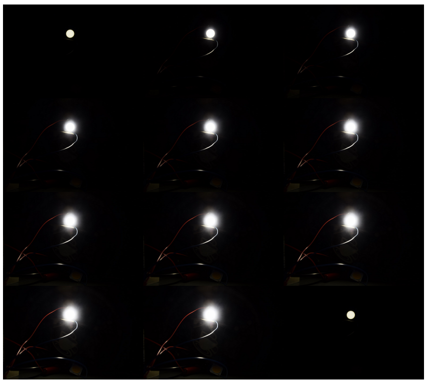


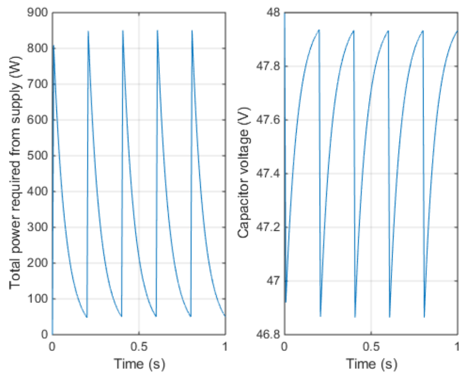
Figure 2.8: Results from PWM verification. Photos of PWM-dutycycle 0, 10, 20,...,100%.

The LEDs chosen for the LED unit were Luminus Devices CHM-14-40-90-36-AC00-F2-3. The selection was based on the LED comparison conducted during the spring. The results of the comparison are shown in appendix C. The chosen LED had the lowest overall cost and sixth best energy efficiency out of the compared LEDs. Additionally it has a wide range intensity it can provide depending on the forward current, ranging from 1000 to 4000 lm. Therefore both the polarised and unpolarised LED can be of the same type.

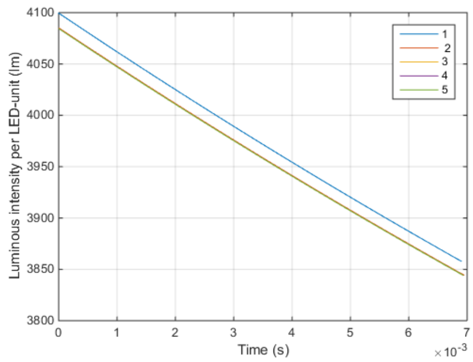
The LED units are driven using three transistors. Two of the transistors are used for controlling whether or not the polarised or unpolarised LEDs are on and they're driven with digital high/low signals. The last transistor is controlled by PWM-signal and it used to vary the intensities of both LEDs. Driving the units in this manner was chosen above having two PWM driven transistor, one for each LED, as a result of most of the potential controllers having abundance of GPIO:s but limited amount of PWMs.

The voltage for driving the unit was chosen to be 48 volts. Main reasons for choosing the exact voltage were the high voltage required by the LEDs, over 40 volts, and the availability of industrial power supplies. The LEDs are powered with large 6.8 mF capacitors. This is due to that lighting up all the LEDs draw up to 8 kW of power for short period of times. Both of the LEDs in the unit are connected in series with resistors, both to set the forward current to get the right intensity and limit the drop in the capacitor voltage during a shot.

The designed circuitry was simulated to evaluate its performance and the total power consumption of the system. The model used for simulation was developed during the spring. The simulation was performed for five consecutive shots of all 156 polarised LEDs. Each shot lasts 7 ms and uses 100 % dutycycle. The results are presented in figure 2.9.



(a) Required supply power and capacitor voltages during the shot sequence



(b) The lighting intensity of the LEDs during each of the five shots

Figure 2.9: Circuit simulation results

According to the simulation results the peak power required from the power supply is 850 watts and the current 17.7 A. The peak current going to a single LED unit would be 114 mA.

The inputs required for the LED unit are:

- GND,
- 48 volts 125 mA,
- PWM,
- GPIO trigger for unpolarised LED, and
- GPIO trigger for polarised LED.

A total of five wires are required for using the LED unit. For external interface a self-locking and widely available connector was desired and therefore standard RJ12 connectors were chosen. The connectors are rated for 1.5 A and 125 V so they can easily handle the requirements for the connection.

The unit has been developed to be able provide the required amount of light for the sequence presented in figure 2.9. During the course of the development three different versions of the LED unit have been designed. These are functional prototype, production prototype and the final PCB manufactured by a subcontractor. These are presented in figure 2.10. The final circuit schematics are presented in appendix D, figure D.1 . When the simulation sequence runs on the real hardware the capacitor voltage behaves in the exact same manner. The main difference is that voltage falls 0.3 volts lower than on the simulation which indicates that the current and therefore the provided light is slightly higher than designed.

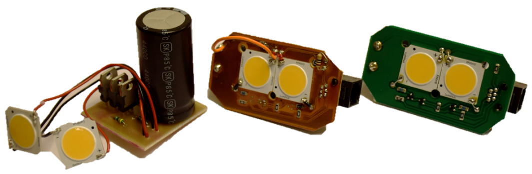


Figure 2.10: Different versions of the LED-unit. The first complete prototype on the left and final production model on the righ

To test the limits of the LED unit a 10-minute long endurance test was run on it. During the test the sequence discussed earlier was repeated every other second summing up to 300 sequences. The unit can provide a stable performance for the tested load which is far more intense than the expected use of the system. During the test significant, but not alarming, heating of the LED and the resistor in series with it was detected. No other components underwent noteworthy heating.

The LED unit has an 10 ohm resistor limiting the charging current of the capacitor. This resistor is rated for 0.66 W, which corresponds to a continuous current of 256 mA. The resistor can, however, handle short pulses of higher current, roughly 1-1.5 A. A current of 4 A leads to power of 160 W which will break the resistor immediately, in less than 1 ms. Therefore the LED unit should never be charged directly from 48 V without external logic to limit the current. This is also the case for connecting the a LED unit with other already charged LED units. If the wiring resistances and power supply are ignored the initial current going through the charging resistor is:

$$i_c = \frac{48}{10\left((\frac{1}{n}+1\right)}A, \tag{2.1}$$

where n is the number of charged LED units in the connection. This means that as few as five charged LED units would provide an initial current of 4 A to the sixth unit that is connected to the network. This would break at least the charging resistor of the connected unit and make it unusable until repaired. Therefore no connections shall be made when the system is powered up!

2.3.4 LCU hardware

This section covers the design for the LCU, which is controlling the behaviour of the LED units. The explainations are divided on the choice of micro controller and the required inputs and outputs.

Micro controller

Requirements for the micro controller:

- 4x PWM signals for controlling the intensity of the LEDs.
- 8x GPIO (General-Purpose Input/Output) pins to trigger either the polarised or non-polarised LEDs.
- SPI (Serial Peripheral Interface) for communication with the CAN-module.
- 1x interrupt pin for the CAN module.

The micro controller that have been chosen is the Arduino Nano because it meets all the requirements, is easy to use and is small compared to other MCU (Micro Controller Unit)s. See appendix B, table B.2 for the evaluation of other options. The reason for not creating a PCB (Printed Circuit Board) from scratch is that it is more convenient to use a already existing, tested PCB with available code libraries that both makes it easy to use and saves time. Another aspect is that if anything were to happen to the PCB it would be easy to replace. The price is another factor to take into account. The price of an Arduino Nano is around 350 kr, which is quite much and in would probably be cheaper if the PCBs were created from scratch, but since the design of the Arduinos is open-source they can be found from around 25 kr in China, making them cost efficient.

The Arduino Nano has been place on the PCB together with a CAN-module and connectors and can bee seen in figure 2.11. The schematics can be found in appendix D, figure D.2.

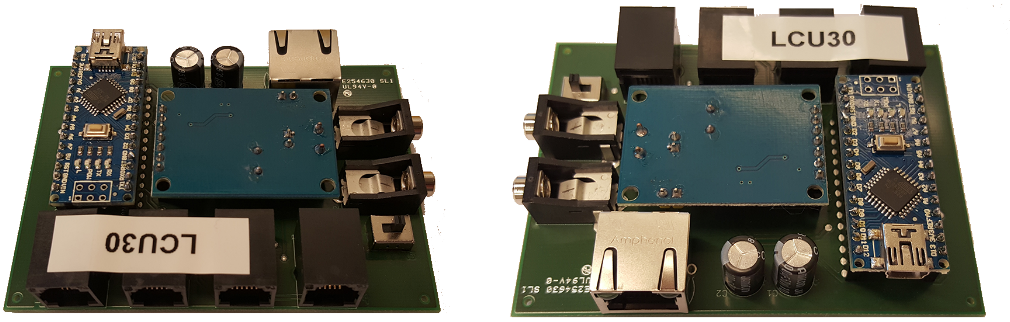


Figure 2.11: Finished LCU PCB.

Connections

The inputs and outputs that the LCU needs to have are:

Inputs:

- GND (Ground)
- 48 volts.
- 12 volts.
- CAN high.
- CAN low.

Outputs (4x):

- GND.
- 48 volts.
- PWM.
- Trigger for unpolarised LED.
- Trigger for polarised LED.

For the input connector a eight wire RJ45 have been chosen. The cable and connector can cope with the load and for safety reasons GND, 48 volts and 12 volts go through two wires each. The PCB is also equipped with two capacitors, one for the 48 volts line and one for the 12 volts line, to minimise ripples and spikes in the voltage supply. This cable and connector have been chosen since it is cheap and can be found everywhere. It is also self-locking so the user can know when it is connected properly. The output connectors that have been chosen are RJ12 connectors and they have been chosen for the same reasons as the input connectors.

When the PCB was designed it was not decided if the CAN signals would go through the RJ45 connector so there are two 3.5 mm 2 pol connectors as well, which the CAN signals could be sent through if needed. There are two since one in for the incoming signal and the other one is so that the signal could pass through and to the next LCU.

2.3.5 PDDU hardware

The PDDU controls the current supply for the system as well as CAN communication to the LCUs. The system consists of a power supply, a master unit (software control and data supply) and a PDSU (which is not physically in the PDDU but is considered a part of the system).

Power supply

A simplified circuit scheme of the 48V circuitry is illustrated in figure 2.12 below.

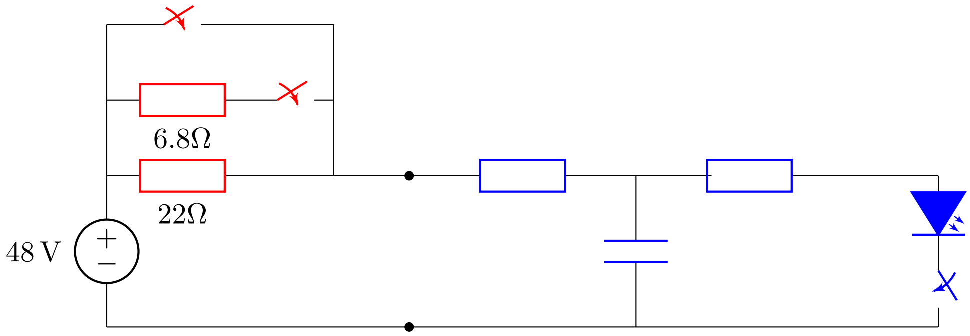


Figure 2.12: An overview of the 48V circuit, supplying the LEDs

The charging circuit red in figure 2.12 is supposed to enable a smooth charging procedure of the capacitors of the LEDs blue in figure figure 2.12, which in turn store energy that is used during the shooting sequence. The LEDs require peaks of about 17 amperes according to section 2.3.3, which is why power supplies capable of supplying 20 amperes were selected. The power resistors of the brand HS100 are able to sustain 100W continously, which is why the first stage was selected to be 22Ω, yielding a peak power of $48 \cdot \frac{48}{22}^2 \approx 104$W. The second stage, which is 6.8Ω starts when the capacitors are charged to 24V and yields a peak power consumption of yields $\frac{24 \cdot (22+6.8)}{22 \cdot 6.8} \approx 106$W. The third stage is when all switches in the charging circuit are closed, enabling the LEDs to draw current directrly from the 48V power supply. The 12V circuitry illustrated is supplying the LCU as well as the master unit with power, as is illustrated in figure 2.13 below.

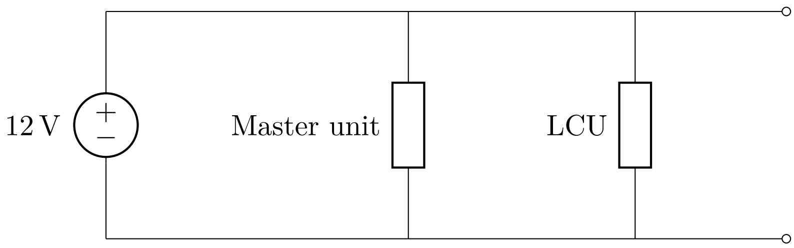


Figure 2.13: An overview of the 12V circuit, supplying the master unit and the LCUs

The master unit and the LCU in turn converts the 12V into 5V. The charging circuit illustrated in figure 2.13 was first modelled in Proteus 8 Professional where both the hardware and the software can be tested, as is illustrated in figure 2.14 below. This can later be used to verify the behaviour of the charging circuit.

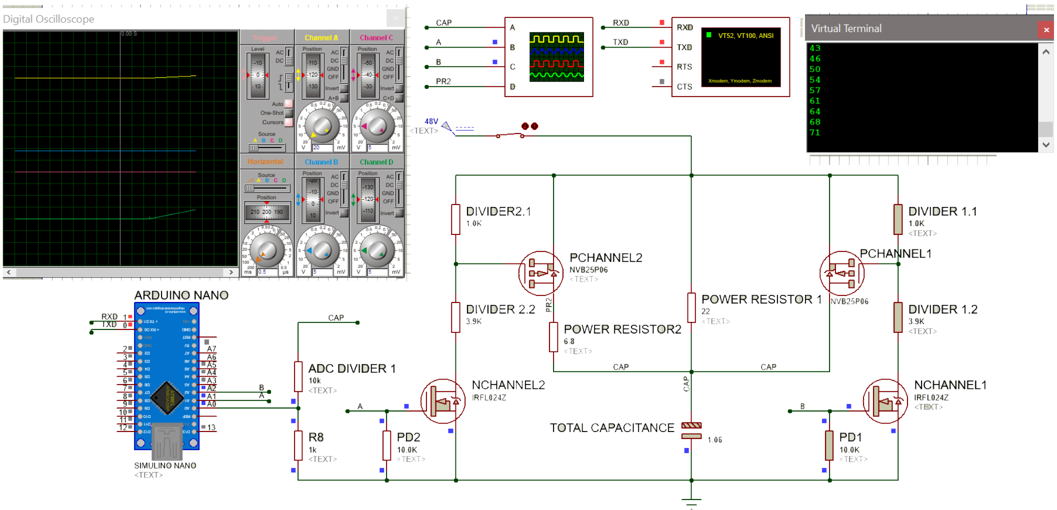


Figure 2.14: The first step of the charging procedure is when all current flows through the 22Ω power resistor.

As the figure illustrates, you are able to display oscilloscope measurements of the voltage level, as well as output the ADC value through a virtual terminal. The circuitry was able to be based on a combination of PMOS and NMOS transistors as well as voltage dividers. The voltage level of the PMOS gate pin is pulled high to 48V and the NMOS transistor pulls it down to $48V \cdot \frac{3.9}{1+3.9} = 38V$ which within the typical range of 4 to 20V below the high side voltage to open. In figure 2.15 below the charging procedure in the real world environment is illustrated. A total of 42 LEDs are charged through the 22Ω power resistor.

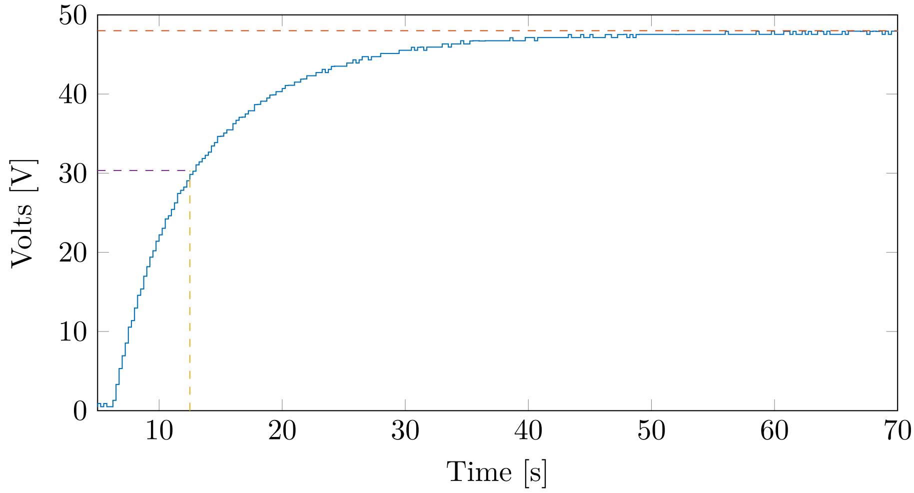


Figure 2.15: Illustration of the charging procedure for 42 LEDs.

Notice how the 42 LEDs reach $48 \cdot (1 - e^{-1})$Volt after about 6 seconds and are fully charged after 50 seconds. As a comparison, when the quick charging procedure is implemented, the time it takes to charge the LEDs is significantly reduced, as seen in figure 2.16 below.

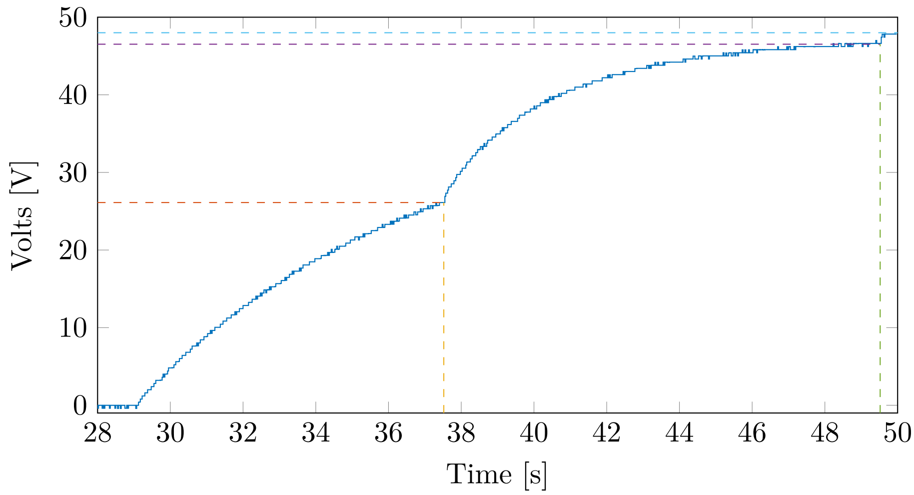


Figure 2.16: Illustration of the charging procedure for 42 LEDs, using quick charge procedure.

Notice how the 42 LEDs are now fully charged in about 20 seconds.

In figure 2.17 below the leakage of charge from the capacitors of the 42 LEDs is illustrated. This is when the power supply is turned off.

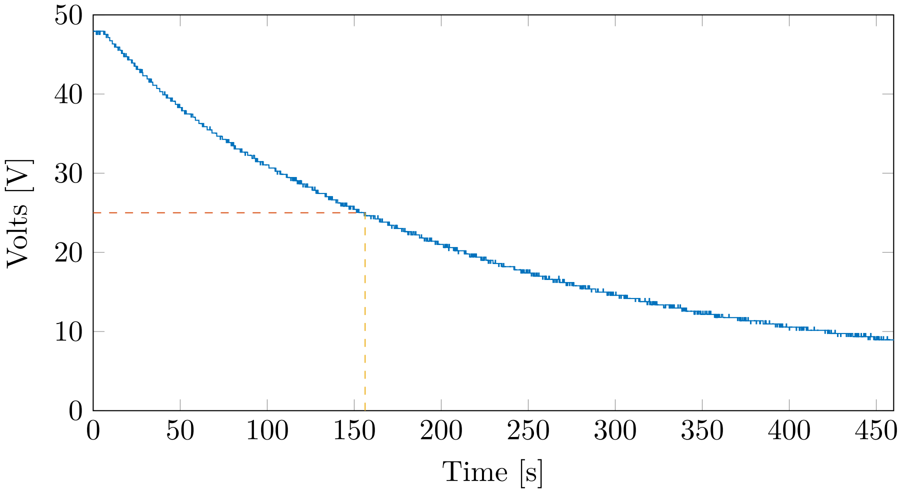


Figure 2.17: Illustration of the discharging procedure for 42 LEDs

As the figure illustrates, the voltage has not reached zero volt after 450 seconds. However, after about 155 seconds the voltage has dropped to 25V, which is what the input resistance at 10Ω of the LED units is able to cope with continuously. In order to discharge the capacitors faster, a discharging terminating resistance and a NC (Normally Closed) relay, i.e. only connected when the system is not powered, is needed in series with ground. In that way, as soon as the system is turned off, discharging will begin automatically as is illustrated in figure 2.18 below.

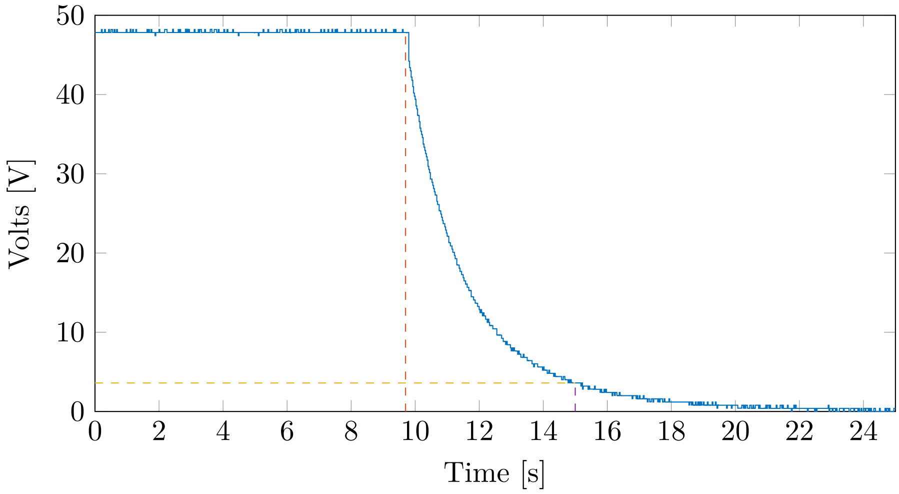


Figure 2.18: Illustration of the discharging procedure for 42 LEDs, when a relay and 4.7Ω power resistor in series.

Master unit hardware

As briefly discussed in section 2.3.1 the master unit's task is to communicate with the LCUs via CAN, as well as handle synchronisation signals to the CAN network and sync box of the cameras. The master unit should also be able to communicate with the CCU (Camera Controller Unit) and user via the controller unit. This should be done via TCP (Transmission Control Protocol), as well as handle the required settings needed by the system. The demands on the master unit can be summarised as follows:

- Provide CAN communication
- Provide TCP communication
- Provide synchronisation signal
- Storage for settings files
- Control for the different units

The stakeholders also required for the whole system to be transportable in travel cases and the team had a goal for the system to be modular. If anything would happen it should be easy to replace components anywhere in the system. All these aspects were taken into account when choosing hardware.

Different micro-controllers were looked into and due to that the unit needs to handle TCP and to have the required storage quantity, a Raspberry Pi became a good possibility. A Raspberry Pi is a single board minimal computer designed to provide computational capabilities as in everyday computers as well as fulfil overall requirements by high end micro-controllers. The mini computer is also driven by Open Source development, which enables for a lot of possibilities developing software. The mini PC provides a set of USB (Universal Serial Bus) ports, memory card space, GPIO pins, Ethernet connectivity and a Linux based OS (Operating System) (if this would be desired to be installed). This mini PC thus fulfils all the requirements set by the system constrains. The Raspberry Pi is also rather cheap and available practically everywhere in the world and thus enables for the modularity and replacement possibilities set by the team.

Although a mini PC is optimal, when developing the LCUs it is desired to have uniform system hardware for the different purposes. This led the team to use an Arduino Nano as a CAN master unit, this unit can be controller by the Raspberry Pi master unit via USB and thus provide a similar behaviour in the CAN network as the other units. The choice of an Arduino Nano for the LCUs and the CAN master unit is explained more in detail in section 2.3.4.

The master unit's hardware is conclusively a Raspberry Pi version 2 model B connected via USB to an Arduino Nano. Three GPIO pins are used to provide a synchronisation signal to the sync box which relays this signal to the cameras as explained in section 2.5.5. The master unit is placed in the power supply unit and connects to the system via the connections provided by the power supply unit as described in section 2.3.5. A capture of the master unit can be seen in figure 2.19 and its placement on the PDDU in figure 2.20.

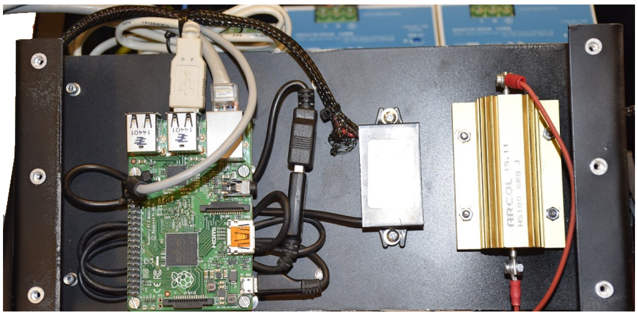


Figure 2.19: The master unit

Figure 2.20: The master units placement in the PDDU

Power and data splitting unit

The PDSU is used to minimise the required length of cables by moving the division into individual cables to the LCU closer to them. It is connected to the PDDU from which it receives 48V, 12V, CAN HIGH, CAN LOW and ground connections. It is designed to grant these connections to multiple connected LCUs.

The number of possible LCUs was chosen with regards to the current that the LCU units draw at peak load. The fact that an even distribution was desired also contributed to the choice. The peak current was calculated to be 105 mA per LED unit right after a flash, this means that each LCU has a peak current of approx. 400 mA.

The cables selected for handling the input to the PDSU unit handles up to 3 A each. This made a division by six appropriate for the unit. This is also beneficial in terms of usability since it grants an even distribution of the power and data splitting units around the sphere.

A capture of the PDSU can be seen in figure 2.21.

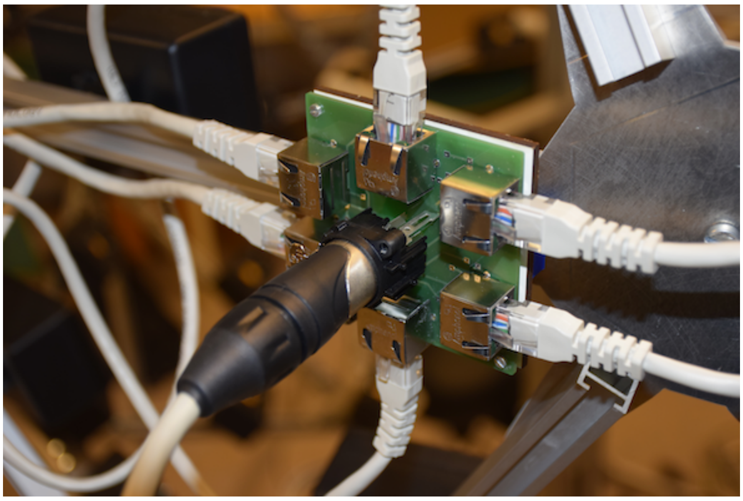


Figure 2.21: A capture of the power and distribution splitting unit with all cables connected.

2.3.6 Cabling

According to standard SS-EN60204 (Machinesafety-elecrical equipment), which amongst other things gives guidance for selecting cables, the maximum voltage drop should be no more than 5% of the nominal voltage.

The electric resistivity, ρ is defined as

$$\rho = R \cdot \frac{A}{l} \tag{2.2}$$

where R, is the electrical resistance, A, the cross-sectional area and l, the length of the material.

The electrical resistivity of copper is $0.017\mu \cdot \Omega \cdot m$, and the voltages used are 240, 48 and 12 V. If the longest wire is estimated to be 5meters long, and the voltage drops are not supposed to be more than $240 \cdot 0.05 = 12V$, $48 \cdot 0.05 = 2.4V$ and $12 \cdot 0.05 = 0.6V$, their resistances cannot be higher than $\frac{12}{16} = 0.75\Omega$, $\frac{2.4}{20} = 0.12\Omega$ and $\frac{0.6}{20} = 0.03\Omega$ if the power supply is delivering its maximum of 20 A, and drawing 16 A from the socket. The minimum cross-sectional area is calculated using the following equation.

$$A = \frac{\rho \cdot l}{R} \tag{2.3}$$

If the resistances are 0.75Ω, 0.12Ω and 0.03Ω respectively, the length 5m and $\rho = 0.017\mu \cdot \Omega \cdot m$, the minimum cross sectional areas are as follows for the cord from the socket to the power supply,

$$\frac{0.017\mu \cdot \Omega \cdot m \cdot 5m}{0.75\Omega} \approx 0.11mm^2$$

and for the 48V power cables,

$$\frac{0.017\mu \cdot \Omega \cdot m \cdot 5m}{0.12\Omega} \approx 0.7mm^2$$

and finally the 12V power cables needs to be

$$\frac{0.017\mu \cdot \Omega \cdot m \cdot 5m}{0.3\Omega} \approx 2.8mm^2$$

.

However, since the 48V and 12V power cables are split into seven different XLR wires, they do not need a greater cross sectional area than $\frac{2.8mm^2}{7} = 0.4mm^2$, which is equivalent to AWG21. Furthermore, the XLR wires split into 6 in the power and data splitting unit, yielding a need for wires with a cross-sectional area of $\frac{0.4mm^2}{6} \approx 0.067mm^2$ i.e. AWG29. Regarding RJ45 and RJ12 cables, used for connection between the PDSU and the LCU, and between the LCU and the LED, they are typically of AWG25 and AWG28 standard respectively i.e. they are within the range of what is required.

2.3.7 Synchronisation of LEDs and Cameras

In order to take high-quality images synchronisation between the cameras and LEDs is required. Synchronisation needs to be achieved for each shot in the five shot sequence. The shutter speed used for the cameras is 1/200 meaning that the shutter is open for 5 ms. In order for the cameras to receive enough light the LEDs should be lit for the whole exposure time. Having the LEDs on for longer time would grant a larger window to take the picture but would in return increase the power consummation. This would also lead to a longer re-charge, increasing the time between shots. In the following subsections the triggering of cameras and LEDs are discussed and the solution described.

Camera triggering

There are several possible ways to triggers the Nikon DLSR cameras used in the project. It is required that all the cameras are triggered simultaneously and that they are triggered at the same time as the LEDs are triggered.

The wireless solution that Nikon provide is based on WiFi. Each camera becomes a WiFi hotspot which one can connect to via either a smart-phone, a tablet or personal computer and it grant access to the camera controls. This solution is of limited use when multiple cameras are used since it makes it troublesome to connect to multiple cameras from one unit. This solution was also considered unstable when tested and considered unsuitable for the uncontrolled environment in which it would operate. Another solution is to use the remote-trigger-wire-interface of the DSLR cameras. This approach has been explained more in detail in the spring report. In short the cameras capture an image when the cable is connected to ground.

Pixel Grinder has been previously utilising the latter solution and it is incorporated in the camera powering system that they deploy. The wire-interface has been chosen to it being proven both robust and accurate when using large number of cameras. The delay between triggering signal and shutter opening was measured to be approximately 80 ms. The trigger signal is sent from the GPIO pins of the master unit through an optocoupler since the sync boxes, supplied by Pixel Grinder, operate at 5V and the master unit at 3.3V.

LED triggering

The LEDs are triggered by the master unit via the CAN network. The CAN-network was chosen as the triggering method since it was implemented for conveying the settings of the shooting sequences to the LCUs. The triggering signal is sent as a CAN message to the LCUs which then trigger the LEDs enabling the corresponding PWM-signals and setting the right GPIO pins to digital high. Since the triggering signal is sent to the LCUs through CAN they receive it simultaneously and all the LEDs can be therefore triggered at the same time. The delay between the triggering signal being sent from the master unit and the LEDs being lit has been measured to approximately 10 ms. Since the camera triggering takes approximately 80ms to execute the LED trigger signal is delayed by 70 ms. This causes all the LEDs and the cameras to be triggered at the same time and the LEDs remain lit until the end of the cameras exposure time.

As figure 2.22 below illustrates, the trigger CAN message is sent once every 0.2 seconds, i.e. five times per second.

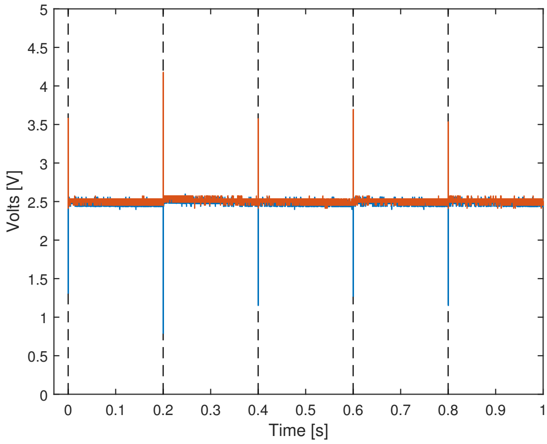


Figure 2.22: Plot of how the of the CAN trigger message and the cameras are synchronised

CAN termination resistance and ringing

The CAN network have been tested with two different setups, figure 2.23 below illustrates the difference with using one or two terminating resistances on the ends of a 4 meter XLR wire.

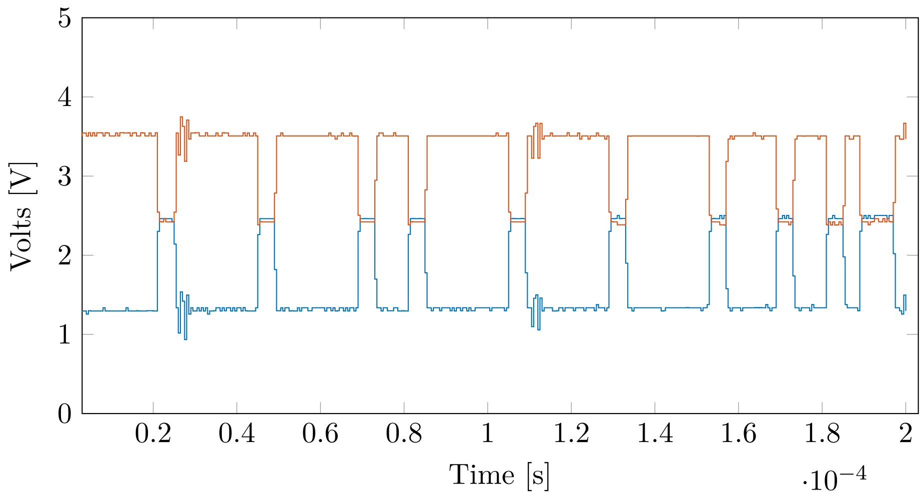


Figure 2.23: An oscilloscope plot of the CAN synchronisation signal illustrating the difference in using 120Ω terminating resistance on both ends of a 4m XLR cable

When looking at the synchronisation signal illustrated in figure 2.24 which is sent to the LCUs, the performance of the CAN message is similar to that of figure 2.23.

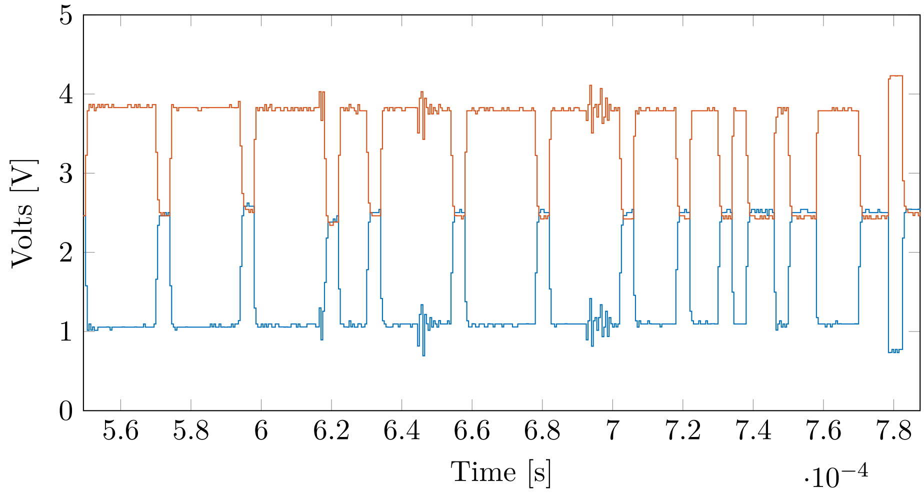


Figure 2.24: Illustration of the synchronisation message which was sent to the 42 LEDs

2.4 Cases and fastenings

In order to ensure robustness and security in the hardware systems, the circuit boards must be enclosed by cases. The cases should also enable for easy packing when travelling and a secure way in which they can be fastened to the mechanical rig. The iterations for each case and the fastenings are discussed in this section.

2.4.1 LED unit cases and fastenings

The casing of the LED unit has three main functions. Firstly the cases need to ensure as little contact between user and PCB as possible. Secondly they need to allow for the light to shine through whilst protecting the LEDs from getting damaged during the transportation and packing processes. Finally the cases must enable secure and easy fastening to the mechanical rig, see section 2.2.

As a first iteration step a custom made casing was designed and 3D-model was generated in Solid Edge, see figure 2.25. This design would enable secure fastening and transportation as well as approved attachment to the rig (by a screw through the back of the case). Also being custom design the PCB of the unit fits the unit perfectly. The design was 3D-printed and proven to work as it should. In order to produce a sufficient amount, however, it was concluded that neither the time required for 3D-printing or the cost of having them manufactured by another company were considered low enough for the design to be feasible to be taken into use for all of the units. A complete assembly description of the case can be seen in appendix G.

Figure 2.25: The first design of the LED case, with PCB and screws attached.

Therefore an alternative solution had to be found. A commercial plastic enclosure was found to be suitable, see figure 2.26. The layout of the LED unit PCB had to be altered slightly in order to make fit the case properly. The connector, LED holes, and screw holes as seen in figure 2.25, were then laser cut from the case.

https://www.elfa.se, visited 2015-12-05

Figure 2.26: Plastic enclosure from Elfa.

The LED units shall, as explained in section 2.1, be fastened on the inside of the dome on all the nodes and rods. They should also be rotatable in order for the user to adjust the polarisation direction. Separate solutions were chosen for mounting the units on nodes and rods. For the LED units on the nodes the fastening was solved by adding an M6 pop-nut in the cases, allowing for the case to be fastened to the hole in the node with an M6x16 screw. For the rods, the fastening was solved by adding a piece of plastic that allows attachment to the rod profile whilst allowing the rotation. Both the fastening systems can be seen in figure 2.27.

Figure 2.27: Capture of the two fastenings, rod fastening left and node fastening right.

2.4.2 LCU cases and fastenings

When designing the case for the LCU to be mounted in, the key requirements were:

Table 2.1: Key requirements for the design of the LCU-case

1. Keep PCB safe from potential damage from operator use and transportation
2. Enable easy connection of all cables
3. Incorporate a way to attaching the case on the geodesic dome

With this key requirements in mind a case was designed in Solid Edge with the intention of using a 3D-printer to produce it. The decision was based on the fact that the case will be highly tailored to the project and buying a generic case and modifying it will prove as expensive and more complicated. The design in Solid Edge is illustrated in the figure below.

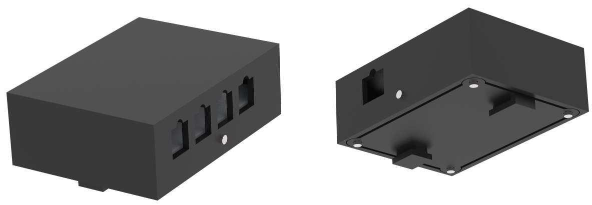


Figure 2.28: A illustration of the plastic case protecting the LCUs

The case will be printed in plastic in a 3D printer. This will provide enough protection for moderate ware and tare. The case has four Ethernet connections that have been fitted with extra room at the top to provide easy disconnection of the cables. On the bottom of the case in figure 2.28 a T-shaped wedge can be observed. The shape will enable the case to be attached to the rods on the geodesic dome. One end of the T-shape is equipped with a rectangular stop-piece. This will hinder the case from sliding down the rod. The entire case, including the screws and nuts needed for assembly, are presented in figure figure 2.29.

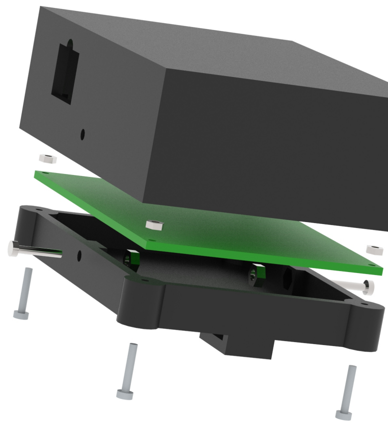
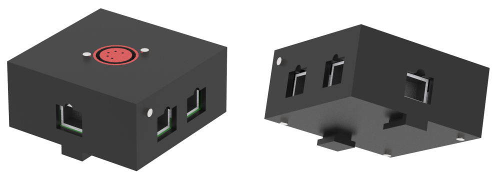


Figure 2.29: The picture illustrates the assembly process and includes all necessary parts

A detailed draft of the LCU-case is located in appendix I.

2.4.3 PDSU cases and fastenings

The design of the PDSU case was done with the requirements stated in table table 2.1 as a foundation. With these key requirements in mind a case was designed in Solid Edge with the intention of using a 3D-printer to manufacture it. The decision was based on the fact that the case will be highly tailored to the project and buying a generic case and modifying it will prove to be expensive and more complicated. The design in Solid Edge is illustrated in the figure below.



Solid Edge, visited 2015-11-10

Figure 2.30: The picture highlights the case design for the PDSU.

The casing will be 3D-printed in plastic, which provides protection against medium ware and tare and transport. The top is thin to allow the XLR-contact, illustrated in red in figure 2.30, to be connected. The T-shaped wedge on the bottom of the case allows for the case to be mounted on the rods of the geodesic dome. One end of the T-shape is equipped with a rectangular stop-piece. This will stop the case from sliding down the rod. The entire case, including the screws and nuts needed for assembly, are presented in figure figure 2.31.

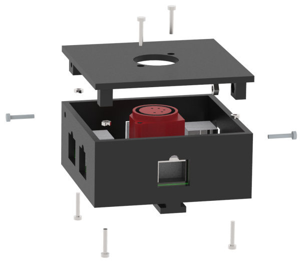


Figure 2.31: The picture illustrates the assembly process and includes all necessary parts.

A detailed draft of the PDSU-case is located in appendix H.

2.4.4 PDDU cases and fastenings

Illustrated in figure 2.32 below is a 3D model of the PDDU which holds the power supplies and the master unit.

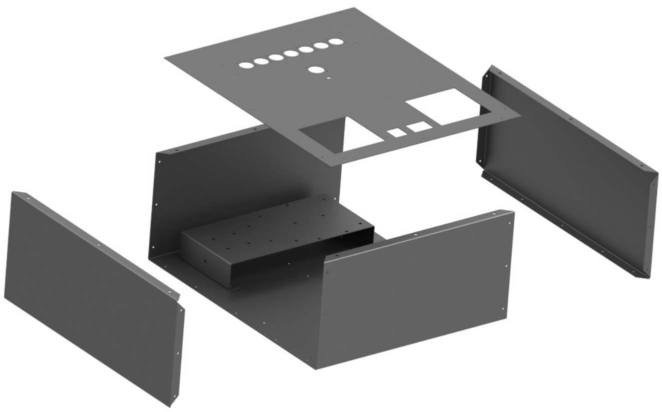


Figure 2.32: A 3D rendered illustration of the enclosure for the power supply, containing the master unit and the charging circuitry

The different parts are fastened using 3mm blind rivets, M3 blind rivet nuts and M3 screws. In order for the case to be modular, the rack holding the PCBs is fastened using M3 screw to M3 rivet nuts. Illustrated in the drawings, which are found in appendix J, is also the holes for attaching the DIN-rail, to which the power supplies are attached.

2.5 Software design

This section covers the logics and software implemented on the hardware units. It also brings up the communication between the user and the system and the implemented JSON file used.

2.5.1 System Overview

The system's architecture is distributed and an overview can be seen in figure 2.33. It shows how the logic of the system is divided into several units working as a single system. As seen in the image, the user interacts with system via the UI (User Interface), sending orders to be performed by the server. The server relays and broadcast the required information and actions to be performed by either the LCUs and/or the CCUs that respectively control the LEDs and the cameras.

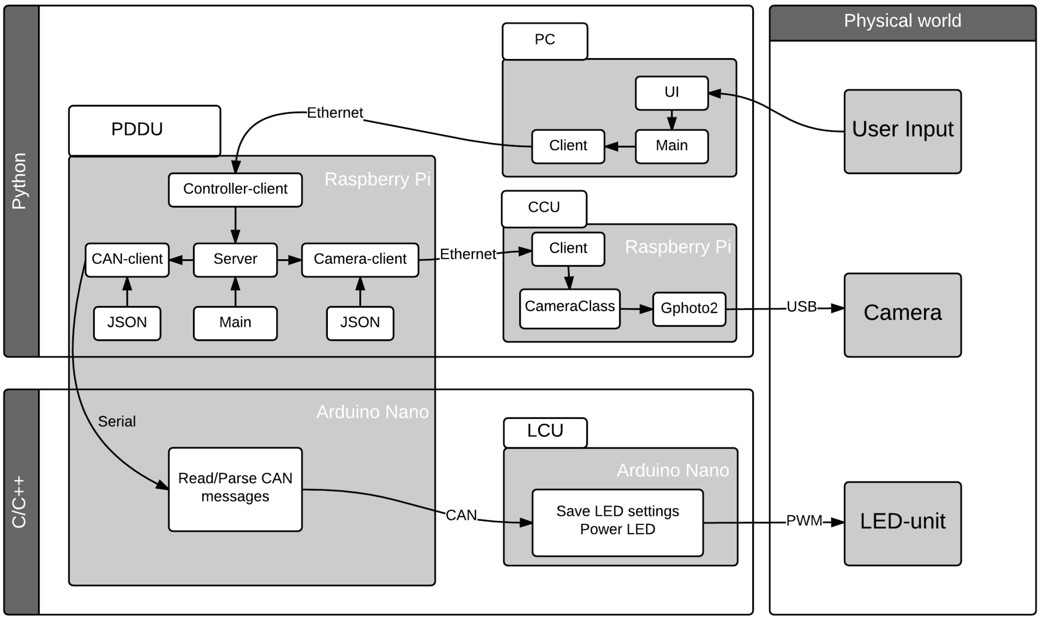


Figure 2.33: The software systems overview

2.5.2 Calculating gradient light

In the last three shots in the sequence of five shots, the light setting is required to be three gradients from three different positions in the rig: one shot with gradient from left to right, one from top to bottom and one from front to back. To get the right intensity for every LED on the rig, the x, y and z coordinates for the physical placement of every LED were calculated with origo placed in the middle of the dome. The coordinates for each LED are shown in appendix E. The position of every LED is seen as a vector from origo. Creating a gradient from top to bottom will let the LED on top of the rig shine with 100 percent intensity and letting the intensity for the rest of the LEDs decrease downwards to the bottom where the intensity is 0 percent. To calculate the intensity of every LED, the vectors to all the LEDs are projected onto the vector of the LED from where the gradient should start. The percentage p (here from 0 to 1) of the length of the projected vector compared to the original is calculated with the following equation

$$p = \frac{\|v_2\| cos\theta}{\|v_1\|} = \frac{\|v_2\|}{\|v_1\|} * \frac{v_1 v_2}{\|v_2\|\|v_1\|} = \frac{v_1 v_2}{\|v_1\|} \tag{2.4}$$

However since that only is the percentage for the projection from origo i.e. its projection on the radius, and the gradient percentage goes over the whole rig, the percentage of the length of the diameter is needed. To achieve this, the percentage of the radius is divided by two to get the percentage of the diameter. Then 50 percent is added to the projection since the intensity in origo will always be 50 percent.

This is displayed in figure 2.34.

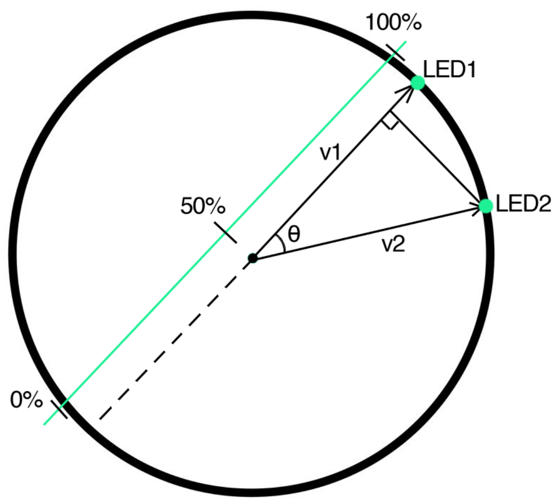


Figure 2.34: Projection of vector of LED2 onto vector of LED1 to caluclate light intensity for the gradient

The final equation for calculating all the LED intensities (here from 0 to 1) therefore is

$$p = 0.5 + \frac{v_1 v_2}{\|v_1\|} * 0.5 \tag{2.5}$$

2.5.3 LCU software

In this section the software solution for the LCU unit is explained. The primary function of the code is to synchronise the LEDs with the cameras, and the code will be implemented on an Arduino Nano connected to each LCU. This section does not cover all details of each line of code, but gives an overview.

The LCU handles the CAN communication between the master node and LEDs with the help a of CAN-module. CAN communicates the synchronisation of the LEDs with the cameras (turning them on at the same time as the the cameras are capturing pictures) and changing the individual intensity for each shot. Each LCU has been assigned to 3 or 4 CAN IDs. Each ID is unique and the total quantity corresponds to the number of LED units in the system. The LCU only uses CAN ID and data frame, therefore no other frames need to be changed. The ID will be used either for triggering (ID=0x0000) or for changing the parameters of a single LED addressed to the ID used in the CAN message. The data frame specifies the setting for that specific LED.

In table 2.2 the CAN messages turning ON and OFF all the LED distributed around the rig is shown. As this message is not filtered, it is listened to by all LCUs and they will trigger the LEDs when doing so. The LEDs will be on for 5 ms, which is implemented in each LCU by a delay.

Table 2.2: CAN messages for turning ON and OFF the LEDs.

Synchronisation of the LEDs		
ID received	Data Frame	Function
0x0000	[0,0,0,0,0,0,0,0]	Sync signal turn ON the LED

The triggering of the LEDs for each shot could have been done directly in the Arduino. However, the synchronisation would have depended on all internal clocks in the LCUs. Therefore, it was considered a better solution to only depend on one clock, the master node clock, since the synchronisation then would have smaller risk of failure. Therefore each time the LEDs need to be ON for a shot, the master node sends a signal to the LCUs.

table 2.3 shows how to change the shot intensity for each LED which is the default case. It also shows the extra cases already programed on the LCU like test cases in order to show if all LEDs are working properly. Extra cases can be added easily. Each LCU has a mask and filters set properly according to the physical position over the rig. Therefore the LCU will only listen to at most 5 different IDs, the synchronisation ID, 0x0000, and the 3 or 4 IDs for the LEDs it is controlling.

The IDs for the LEDs in table 2.3 have been represented as 0x—A where "-" means the numbers that compound the ID for a LCU and A is a number between 1 and 4 that defines to which LED on the LCU are we talking to. In the data frame the first byte means the type of message (change settings, test, etc.) and the rest of the bytes are the information regarding this type of message

Table 2.3: CAN messages for changing parameter and extra functions.

Predefined Functions		
ID received	Data Frame	Function
0x—A	[0,S1,S2,S3,S4,S5,0,0]	It will change the Intensity (from 0 to 100) for each shoot for the LED A.
0x—A	[1,0,0,0,0,0,0,0]	Check the non-polarised LED flashing the LED
0x—A	[1,1,0,0,0,0,0,0]	Check the polarised LED flashing the LED
0x—A	[2,0,0,0,0,0,0,0]	It will turn ON or OFF (depending on the actual state) both polarise and nonpolarised LEDs

2.5.4 PDDU software

The PDDU's software (controlled by the master unit) holds the logic and handles the communication between all devices. The master unit's software is divided into two parts: the controller interface and the server. The server is similarly divided into two sub-parts: the Raspberry Pi's software and the Arduino Nano's software, where the latter is a slave unit to the first.

As presented in section 2.3.5, the requirement of the master unit was to handle all communication from the different parts of the distributed system. This includes the high level communication between the controller, the master unit server and the CCUs via Ethernet, and low level communication of byte arrays via CAN. In order to allow for seamless communication between the CCUs, LCUs and the user, several system architectures were evaluated with mainly two different approaches for the server and controller interface.

The first possibility was built on the idea of making the server a database handling unit, having a web-based controller interface for the user and handle the settings from all the devices in a dynamical way. The system would in this case have relayed all the data, the different settings from the different devices, and stored them accordingly, making it possible for the user to access them and change them via a web-based interface. The reason for considering this was the possibility to easily develop a GUI (Graphical User Interface) using HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) programming. However, this required the data to be stored in a SQL (Structured Query Language) database, which required knowledge not previously known to any of the team members. Another problem with this solution was that it did not provide an easy way of enabling the CAN communication due to the difference in communication protocols between TCP and CAN.

The second possibility was to implement a server based program using sockets for communication between the different devices, utilising command based clients. This approach provided the possibility to use python programming for the socket communication as well as the command based clients. In this alternative, the possibility for the settings to be stored in JSON (JavaScript Object Notation) files was available, which can both be read, changed and interpreted by humans and computers.

With a desire to achieve a robust, modular, user-friendly and reliable system being easy to improve, the choice was made to go with the second alternative. This would provide more flexibility, both in what could be done with the system as well as what the end user could improve if desired after the system was delivered.

With the overall architecture for the master unit's software being decided, the development of the communication and the command based clients was initiated. Firstly, the communication over CAN was developed. In order to do this, an early implementation of the CAN network communication protocol presented in table 2.2 and table 2.3 was developed. Based on this protocol, serial communication between the Arduino Nano and the Raspberry Pi was implemented using the required data fields msgType, ID and CAN message, for the CAN communication. Later, the controller of this communication was developed on the Raspberry Pi, enabling different messages to be sent as well as JSON files to be read and configurations to be sent over the CAN network.

The next step was to develop the TCP capable communication between the different units, clients. This was done by implementing a server handling the communication relay between the different units. Firstly, communication was established between two units, with one acting as server. In order to enable a robust and reliable system, the communication needs to be very reliable and thus the different communications should work in parallel, in this case as different thread instances. A new version of the server was developed which spawns a thread on connection, allowing for the communication to a specific unit not to interfere with the rest of the communication if there is an error or disturbance that could affect the code running. In this case, any crashes of a specific client communication will terminate at the specific thread and the system will keep on running.

As this was being implemented, the three different types of clients were developed in parallel. The two first types, for the controller and camera. The first type allows all the commands controlling the system to be sent to the different units. The second one handles the specific commands that interact with the camera software, more of which can be found in section 2.5.5. A CAN client which handles the communication to the CAN network via the Arduino Nano was derived from the previous CAN communication software. A new version of the JSON file reading software was developed and implemented in the CAN communication software in accordance to how this was created in section 2.5.6.

The last thing to be implemented in the master unit's software was the controller client's UI. This is the interface that runs on a computer and allows the user to control the distributed system. This was done as a text based command control which handles the communication protocol enabled by the server.

The overall functionality of the master unit is described as part of the system overview in figure 2.33. The main function of the master unit is to relay information from one unit to another. A more detailed logic description can be seen in figure 2.35.

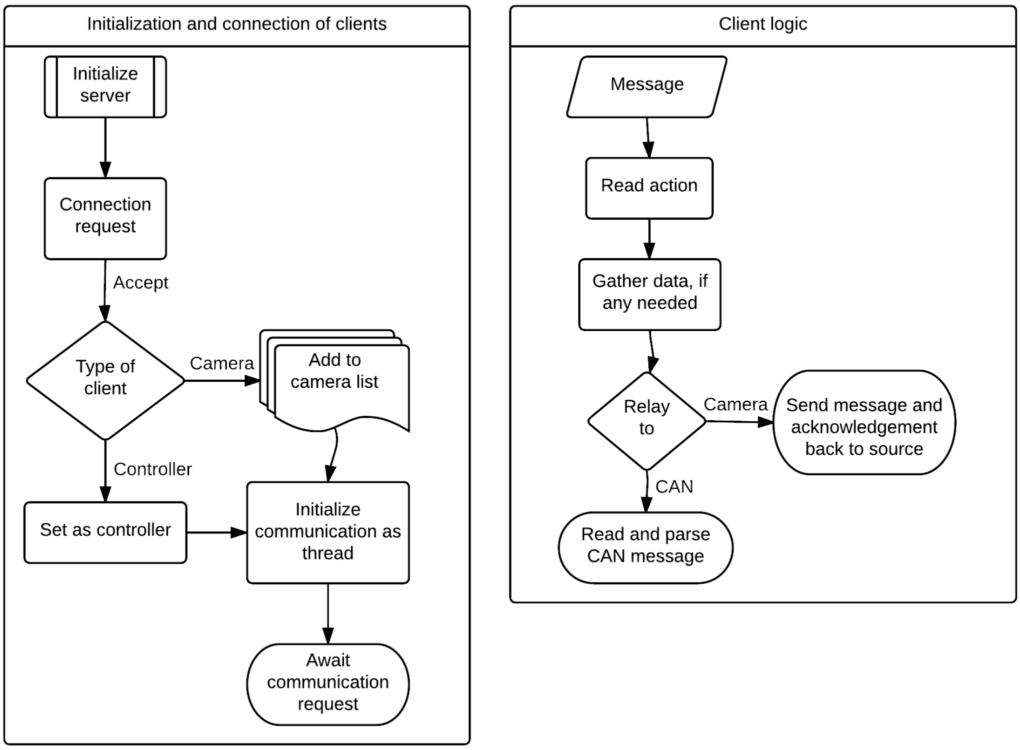


Figure 2.35: The master unit's logic

Since the main task of the master unit is to handle the communication between the different units, a detail explanation of the messages handled by the unit is presented in table 2.4. All messages to the master unit from the controller and CCUs follow the structure {{server action},{unit ID},{data field}}. As presented in the table, the different units that are able to send information to the master unit are displayed. The different server actions enabled by these units are also available and in the case of a send message, the unit ID is also available. The data field presents the way the data field needs to be sent, how it will be interpreted in the end unit and what can be sent.

Table 2.4: Text based messages to the master unit

From	Server action	unit ID	Data field
Controller			
	send	{{unit type}_{ID number}	
		Camera_X	{{command} {value}} {{iso} {value}} {{aperature} {value}} {{shutterspeed} {value}} {{focusmode} {value}}
		CAN	{{command} {config}} {{sendSync}{}} {{sendConfigurations}{}} {{quit}{}} {{sendSettings} {{LED ID} _msgType-S1-...-S7}} blink {{LED ID}_{Polarized or unpolarized}}
	quit		
Camera			
	send	{{ID type}_{ID number}	
		Controller	{Info} {Error while changing...} {Settings changed to...}

2.5.5 Camera control software

To enable remote control of the cameras during a photo shoot some software needed to be created. The purpose of this software is to allow the user to change settings for individual cameras from their laptop. To enable this functionality several different communication channels needed to be established. To view the path of the information travel, from user-input to the camera, please see figure 2.33.To enable the user to manipulate the camera remotely, the camera needs to be connected to a Raspberry Pi, a single board computer, with a USB cable. The software, which is controlling the camera, is located on a Raspberry Pi and will be mounted in close proximity to the camera. The following discussion will only focus on the communication from the camera perspective and the software that enables the needed functionality.

The software, written in Python, is receiving information over a socket from the Master Unit which is acting as a server. The information transferred over the socket contains the commands the user wants the software to execute. Available commands are listed in table 2.5 below:

Table 2.5: Available commands for controlling the camera.

Commands	Description
mkdir	Make a new directory on the Raspberry pi
iso	Set an iso value on the camera
aperture	Set an aperture value on the camera
shutterspeed	Set the shutterspeed on the camera
focusmode	Set the focusmode on the camera
getsettings	Retrieve the current settings on the camera.
capture	Capture an image and saves it to the Raspberry pi

The list of commands is based on the stakeholders requirements and contains all the essential functionally that will be useful during a photo shoot. The software logic and information flow is visualised in figure 2.36 below:

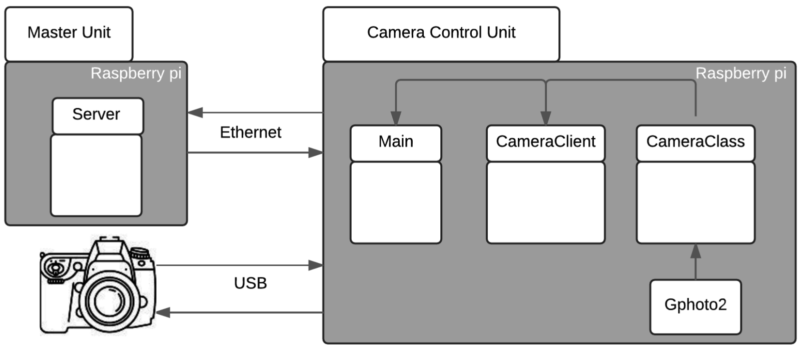


Figure 2.36: Camera software overview including the communication path

To elaborate on 2.36 the program Main will include the class CameraClient, as indicated by the arrow, and spawn an instance of the class. The CameraClient will then proceed with establishing a connection to the Master Unit over Ethernet. The communication is done by establishing a socket on both sides allowing bi-directional communication. The messages received by the Camera Control Unit will be string messages containing either the commands listed in table **??** or a request for the

CameraClient to close down the socket. If the message is a command CameraClient will call the appropriate function in CameraClass, which is able to manipulate the camera by using the software Gphoto2. Communication between Gphoto2 and the camera is done over USB. After executing a command, a response is sent back to the user via the socket. The response contains information on if the command was successfully executed as well as the new state of the camera.

Prior iteration of the software also included the ability to capture images and auto-focus by using the remote shutter release described in section 2.3.7. This was achieved by using the GPIO pin available on the Raspberry pi. This functionality was moved to the Master Unit to ensure that all the LEDs and all cameras are kept synchronised when taking a picture.

2.5.6 JSON implementation

In order to save the light settings in a modular and comprehensive way, they were stored in JSON files. An advantage with JSON files is that they can be read, changed and interpreted by humans as well as computers.

In order to make the solution scalable, a program for creating JSON files was developed. The program uses the gradients calculated as described in section 2.5.2, which are saved in lists where every element corresponds to an LED and contains a number between 0 and 255 representing the light intensity.

The basis of the JSON file is a list of LEDs, with their first field being their ID (a number between 1 and 153, the number of LEDs). The second field contains the settings, divided into default and special. These two field have the same structure, containing a field for message type and a field called shots, which is a list of 5 numbers between 0 and 255. These numbers represent the different light intensities for the 5 sequential shots.

At the moment, the default field is in use, where the 5 shots are defined to be ambient unpolarised, ambient polarised, gradient from left to right, gradient from top to bottom and gradient from front to back. The special field is made for future work, to be used if Pixel Grinder wish to create other light settings in addition to the default one.

The program creates a JSON file, which is interpreted in the master unit and used for constructing a message for the CAN communication, containing the data fields msgType, ID and CAN message.

2.5.7 User interface

To make the rig easy to use and control, a text based user interface was set up where the user can control the rig from a PC. The user interface has three main functions: trigger a sequence of shots, change settings and perform a test on the rig.

The trigger function sends a synchronisation signal through the network and a sequence with predefined number of shots as well as light and camera settings, is started.

Through the user interface it is also possible to change the settings of the light rig. The settings function lets you change sequence settings, LED settings and camera settings. In the sequence settings it is possible to choose from a number of predefined sequences saved in a JSON files. The camera settings function makes it possible to change the settings for all cameras at once or to change the settings for one specific camera by choosing the camera ID. The attributes to be changed on the cameras are the shutter speed, the ISO value, the aperture and the focus mode. The different values for these attributes are all saved in text files. In the LED settings the intensities of the different LEDs are changed. To change a specific LED the LED ID is chosen and a value between 0-255 for the desired intensity is entered.

To be able to test the function of the LEDs in the rig a test function is also implemented in the user interface. The test function initially lets the user choose between testing several LEDs at once or to test one specific LED blink. When testing several LEDs one can choose from testing only the polarised, only the unpolarised or all LEDs at once. When testing the polarised or the unpolarised LEDs the LEDs of the LED type chosen will blink one at a time in a spiral from top to bottom. When testing all LEDs, all LEDs will shine at the same time with low intensity. A blink function is implemented to test specific LEDs by letting it blink once. The desired LED ID is entered as well as the LED type to choose the specific LED on that LED unit.

Chapter 3

Demonstrator

This chapter presents the final critical function prototype designed for this project and it also presents the expenses of the project.

3.1 Prototype Description

In this project, a light rig with the shape of a geodesic dome with 1.6 meters in diameter was designed and constructed. 41 LCUs and 153 LED units were mounted on the rig together with 30 cameras. One LCU controlling four LEDs was discovered defective so a total of 40 functioning LCUs were used on the prototype. In addition to the four LED units, connected to the faulty LCU, one other LED unit was discovered broken resulting in a total of 148 functioning LED units.

The light conditions worked well and the light was evenly distributed around the rig in an approved manor. Pixel Grinder concluded that the resulting 3D-models were clearly better compared the previous solution used with regular flashes mounted around the rig. An initial scanning of 10 people during the demonstration confirmed this, and the resulting pictures proved the spread, intensity and gradient of light to be correct. The five pictures taken with camera 12 can be seen in figure 3.1. Even though the gradients might be hard to distinguish the gradients at first sight (especially the two last ones), they have great impact to how well the software preforms.

The light intensity that the light rig produces is easy to control, both on individual LED units as well as sending predefined light settings for all LED units at the same time. The polarised and unpolarised ambient light and the three gradients were all working as expected. All LED requirements regarding maximum intensity, CRI value, frequency and power are fulfilled. LED settings are saved in JSON files in the computer though current settings of the rig cannot be retrieved from the rig.

The 30 cameras mounted on the rig and can easily be placed and replaced on desired rods. They are attached with standard camera fastenings and are given uncompromising view of the object to be scanned. The fastenings of the cameras do not in any way interfer with the placements of hardware or cables.

Figure 3.1: Capture of Sarah, from left to right: 1. Full unpolarised 2. Full polarised 3. Gradient left to right 4. Gradient top to bottom 5. Gradient front to back

The synchronisation worked as planned. All cameras synchronised with the flashing of the LED units meaning that all shots were able to capture images during the time the LEDs were turned on. The settings used on the cameras were a shutter speed of 1/200 and aperture of f-13 as stated in the requirements. Five pictures with the proper light settings were able to be taken within a second.

The aim, positioning and zoom of the lenses are controlled manually on the rig. The settings for the cameras can be controlled remotely via the user interface on the personal computer. Focus mode can be set remotely but focusing the lenses is done manually if auto-focus mode is turned off. The view from the cameras must also be done manually. At the moment the rig is not compatible with different camera models.

The rig can be disassembled and fitted into several Peli 1650 cases. The total weight of the rig is not established. It is, however, possible to divide the parts between the cases in a way that every case does not weight more than 31 kg, as stated in the requirements. The disassembled parts can be divided into mechanical and electrical parts. All parts are listed in the table 3.1.

The light rig can approximately be assembled within one working day for two to three persons. The physical placement of the LCUs and the LED units are mapped as well as the connection between them. There are no fixed placements for the PDSUs since they can be connected to optional LCUs. Each PDSU is connected to the six LCUs placed closest to it.

The cabling parts of the equipment are standardised and easy to replace. However, the LCUs and LED units are more complex parts developed specially for this rig and are harder to replace if broken.

A caption of the final prototype is shown in figure 3.2.

Table 3.1: List of components the rig is disassembled into.

Mechanical parts	**Electronics**
Aluminium 6-nodes	153 LED units
Aluminium 5-nodes	41 LCU
Aluminium rods - short	7 PDSU
Aluminium rods - long	PDDU box
Aluminum rods for bottom structure	3 Synchronisation boxes
Fastening bolts and screws	30 Cameras
	30 Camera fastenings
	200+ Cables

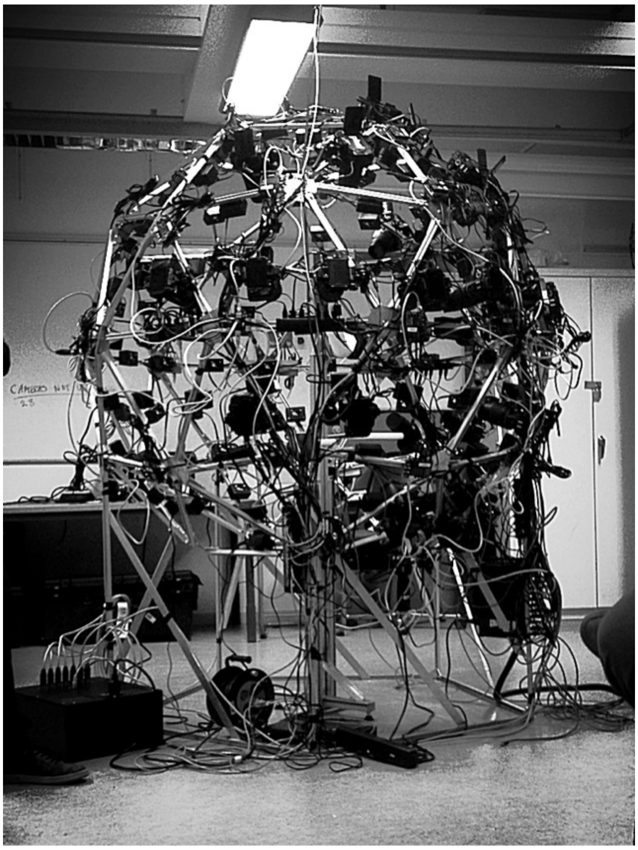


Figure 3.2: Final Prototype

3.2 Economics

The total expenses for the light rig including all components, various connectors, the manufactured PCBs, cables etc. added up to a cost of 92.158 SEK. Here, the cost for material and manufacturing of the mechanical rig (79.503 SEK) is not considered as it was administrated by the company. Furthermore, the project group had prototyping expenses (for not fully implemented ideas, exploration, testing etc.) of 15.759 SEK. The total amount of purchases made added up to 107.918 SEK (here, one could argue that such a cost is minor compared to the vast amount present computers). The expenses and purchases have continuously been discussed and approved by the Pixel Grinder team. A complete table of all purchases made can be seen in appendix K.

Chapter 4

Discussion

This chapter presents a discussion on the decisions that have been concluded and the conclusions that are drawn from these. It also reports from the performance of the project as well as future work.

4.1 Discussion and conclusions

In summary, a light rig easy to assemble, disassemble, travel with, operate and with the ability to create lighting conditions about five times as good as with the equipment used before, was designed and implemented. In relation to the project's scope, the majority of the goals were reached. The final prototype worked in a satisfying way, both in the project group's and the company's perspective. The company believes that the rig will improve their future scanning results and simplify their coming scanning projects.

Most of the requirements set up in the beginning of the project were fulfilled. Time limitations and other obstacles that arose along the iteration line, some limitations had to be made. Close discussions with the company led to decisions on what should parts that should be focused on. Controlling the camera movement was one of the functions not implemented due to lack of time and by the concentration of efforts put into refining the light rig. It was not, however, prioritised by the company since their primarily wishes for the rig were the lighting conditions and the synchronisation of LEDs. Designing a graphical user interface was also postponed since it was not critical for the operation of the rig. Further testing of would have been beneficial in order to verify the functionality of the rig and to refine the control. Analysis of the few malfunctioning units and the reasons for these errors should also be made. An investigation in regard to robustness and what components that are most likely to become defective is also of interest, as well as looking further into and simplifying the process of replacing faulty components. A more detailed discussion on the different parts of the rig, their result and performance as well as possibly improvements is described in following sub chapters.

4.1.1 Mechanical structure

During the design of the mechanical rig, focus was held at enabling secure fastenings, shortening assemble time and keeping the weight, cost and manufacturing time limited. This was achieved and approved by the company. If it would prove, however, that the time or complexion of assembly would become unreasonable large, some adjustments could be done. With an increased manufacturing costs, the design of the nodes could be done in such a way that would that would enable fast, snapping fastenings of rods and PCBs. This would shorten och ease the assembly process greatly.

In the beginning of the design process, it was not stated how the cameras would be mounted. The original idea from the company was to have a second rig only for cameras, which would be situated around the light rig. As our design for the mechanical rig also enables for camera attachment, a second rig is not necessary. This lowers both the assembly time and the total weight.

The idea of having a removable section of the dome, see section 2.2, was not fully implemented. In order to achieve this, proper magnetic locks and hinges need to be added. As the rig is constructed now, the person getting in to the rig needs to crawl underneath the bottom rods. Something could also be done to improve the stability of the rig. As it is now, especially when one node is removed, the rig becomes rather easily swayed when pressure is applied. This could be solved by adding extra rods attached to the dome from ground.

4.1.2 LED unit and reflections on power consumption

As described in section section 2.3.3 the performance of the LED unit is in line with the requirements. However, as the circuit is extremely simplified an external charging requires a procedure control as described in section 2.3.5. This means that the system power supply has to be designed for the specific amount of LEDs and charging the dome takes a significant amount of time. It is possible to break the charging resistors in the unit by inserting wrong current.

The previously mentioned defects in the charging procedure design reduce the reliability of the system significantly. Therefore it would be beneficial to implement an internal charging control procedure. One possible solution has been sketched and is presented in the figure 4.1.

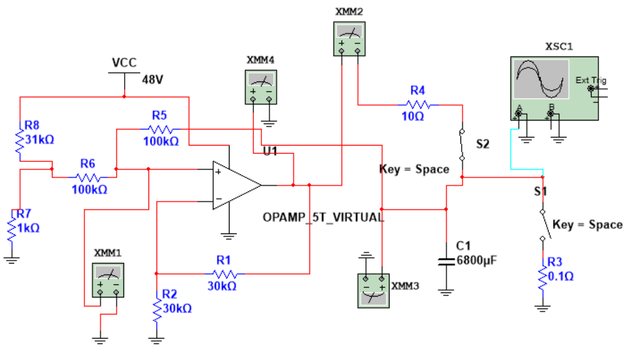


Figure 4.1: Potential solution for improved LEDunit charging.

Here an operational amplifier is used to control the input voltage. It functions as a summing amplifier outputting the capacitor voltage and adding a fixed value corresponding to the desired charging current. This gives a constant, smooth and fast charging of the capacitor without external control. The circuit was tested with a simple Multisim-simulation and the results can be seen in figure 4.1.

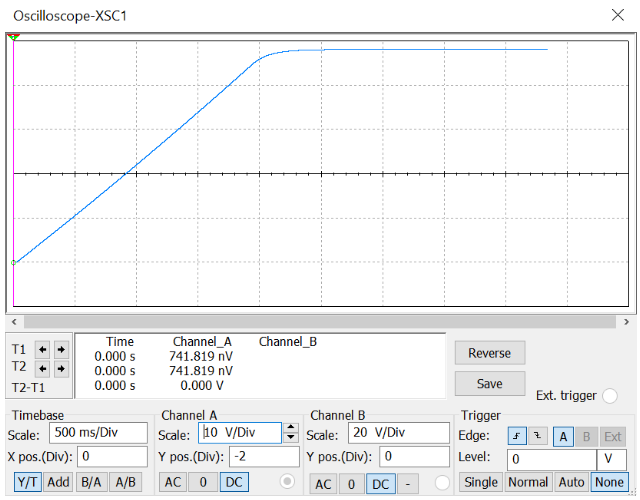


Figure 4.2: Simulated chargin of the capacitor using a charging current of 150 mA

Another point of improvement for the LED unit would be to change the resistors connected in series with the LEDs. Currently those are under significant load and experiencing major heating during shooting. This is good in the sense that they act as a fuse protecting the far more expensive LEDs from overheating, but might lower the operating-life of the LED-units slightly.

Finally assembling the LED units is currently extremely time consuming. For the next version of the unit the assembly procedure should be rethought to make it faster and easier.

4.1.3 LCU hardware and software

The LCU in its current state works well even though there is room for some improvement.

Hardware

One of the things that was thought of when developing the LCU was to include a small LCD (Liquid-Crystal Display)-screen to be able to print information. Unfortunately, the Ardunio Nano did not have enough pins to support the LCD-screen that would have been suitable for this. A LCD-screen is not essential for the LCU to work, but it could be used to display status, ID and settings information of the LCU to make it easier to see that everything works and is connected correctly.

Another way to improve the LCU would be to design it totally from scratch, instead of attaching an Ardunio Nano and a CAN-module separately those components would be integrated into the PCB. Although it is convenient to have the Arduino and CAN module separately it would be more professional and save more space to include the components directly on the PCB. If this would be done it has to be decides if it would be made in a way so that the Arduino libraries still could be used; either by designing an Arduino on the PCB with their open source schematics or only to use components that precisely fit the purpose (without using the Arduino library).

Regarding the connectors on the LCU, firstly the two extra 3.5 mm connectors for the CAN communication would be removed since the CAN communication worked through the PDSU. Even though the RJ12 and RJ45 connectors are good and fairly cheap it could be considered to change these to something more robust since they are going to be connected and disconnected plenty of times. Connectors that could be considered would be mini-XLR or LEMO connectors. These connectors are robust and are made to be connected and disconnected and they also have a good locking mechanism.

Software

The actual software in the LCU covers the minimal requirement; five shots with the different light conditions. Some test cases have been implemented in order to check the LED units in order to see that they are working perfectly. Nevertheless, the functionality of this software could be expanded much further.

One example is to expand the code for having more than five shots in a row. More test cases could be implemented, etc. All of it is not too difficult since all the code has been well documented and someone with knowledge in C and Arduino could expand the capabilities of the LCU.

One of the main drawbacks with the LCU is that every individual unit has a fixed ID. This means that they must be placed in a specific node every time the dome is built. A better implementation would be to have a dynamic ID which would be set in accordance with where they have been placed when the dome is built.

4.1.4 PDDU software

The PDDU (master unit) software is implemented right now to handle only the tasks explained in previous chapters. The software has been designed and tested with regard to communication robustness in accordance to what the system is supposed to cope with. Unfortunately, it is not safe to say that all possible error sources have been covered. Since it has been desired that the system is modular it is difficult to verify if this has been completely achieved. The authors are not the end users of the product and there might therefore be cases for usage of the product that have not been covered when designing the overall product.

Meanwhile, there are some features that with more time could have been implemented and tested. An example of this is a self diagnostic solution for the system controlled by the master. This could be a part of a start-up procedure enabling the system to run a test sequence, verify proper communication between all the units and that the system will work as intended. This could help the user by providing feedback as if for example one cable has not been connected and thus one LCU or a group of LCUs will not work properly. A solution such as this would check the status of the network by checking the status of each unit.

It might also be interesting to put the actual computational power of the master unit to good use, by for example having it handling more settings, data, images, as well as making it possible to control the system with a smartphone.

4.1.5 PCB cases

The cases for the LCU and PDSU described in section 2.4.2 and section 2.4.3 meet the requirements stated in table 2.1. The cases were 3D-printed to ensure the PCB could fit inside the case and that the case was able to be mounted on the dome, using the T-wedge design. The case was only printed once as a proof of concept. Some slight modification were made to the PDSU case afterwards but should not affect the validity of the model. The 3D-models of the cases will therefore act as a template for the company if they wish to produce more cases.

The production of more LCU- and PDSU-cases is based on the availability of 3D-printers. If there is no 3D-printers available or if the plastic used in the process is deemed to weak, purchasing generic cases and modifying them is an option. This approach was used when manufacturing the LED-case, described in section 2.4.1, with great success. The positive effects of this approach is that it offers a lower price for the plastic case and higher structural strength than the printer. The downside is that the cases need to be to be modified afterwards to enable the cables to be connected. Finding a appropriate sized case for the PCB might be problematic which can lead to a PCB re-design.

The LED-case described in section 2.4.1 was produced in great numbers and at a low production cost. The needed holes for the LEDs and the RJ12 connection and was achieved by laser cutting the plastic case. The only downside is that the PCB needs to be slightly modified before assembly to fit the case. To avoid this a redesign of the placement of the components could be done.

This project does not include a unique design for a case for the Raspberry Pis. The reason is that there already exists a vast numbers of designs available for purchase for a low price.

4.1.6 Synchronisation

The synchronisation in the demonstrator is based on empirical tests of the delay in the cameras that are used. For the intended use the synchronisation proved to meet all the requirements. Since this delay has been implemented statically in the software of the LCUs a modification of the program is required if cameras with different delay are to be used. It could be generalised by altering the interface between the LCUs and the master unit to include a configurable delay.

The current implementation of the synchronisation enables the LEDs for precisely the same time as the exposure time of the cameras. The advantage of this is that it minimises the power consumption of the LED units leaving the capacitors with more charge after each shot. The drawback is that this requires the cameras and LEDs to have no offset whatsoever. The system has been designed to cope with the LEDs being active for 7 ms per shot, with this configuration the capacitors should still be able to recharge until it is time for the next shot. Since the synchronisation has been proven to be acceptable while the LEDs are configured to be active for 5 ms there should not be any reason to increase this to 7 ms unless the synchronisations are noted to be off later on.

4.1.7 User interface

To develop the user interface further, the sequence settings function should be expanded by adding possibilities to change the current settings by dynamically retrieving the current settings from the light rig. It should also be possible to create new sequence settings and save them. When creating new sequences it would be of interest to be able to choose from predefined light settings for all LEDs instead of changing the intensity for one LED at a time. A code for calculating gradients from a desired point in the rig to retrieve the light intensity for every single LED is already implemented. Implementing a GUI to get a better overview and make the light rig easier to control is another improvement. This was not a requirement from the company but could be of value to implement in the future.

4.1.8 Camera control and image handling

The camera control and image handling requirements established by the stakeholders at the beginning of the project can be viewed at in appendix appendix A. The appendix states that all requirements were grouped into two phases according to priority, where camera control was considered less important. The reason for this is due to the fact that the company already own a solution which can handle to most essential requirements. The additional requirements were established since they would be convenient for the operator during a photo shoot.

A summary of the software that has been developed, and what of kind functionality it provides, can be seen in section 2.5.5. The following list contains the the areas where future work should be focused.

1. The cameras aim, position, zoom and focus shall be remotely controlled.
2. It shall be possible to allow the user to move the cameras pitch, yaw and zoom.
3. It must be possible to save the camera setup.
4. Camera setups shall be able to be loaded from saved setups.
5. It shall provide live view from the cameras to the user.
6. The software shall be able to access the images from the cameras.

The first two requirements involve control of the camera by using motors. For a structure of the size constructed in this project this is purely a luxury feature since all the cameras are relatively easy to adjust by hand. If this feature becomes highly prioritised in the future, for example in a larger dome, the camera position can be controlled with set of servomotors controlling the yaw, pitch and zoom. A unit that is able to control the movement of a camera is called a Gimbal.

To implement a Gimbal unit the current camera module needs to be slightly modified from the one described in section 2.5.5. The Gimbal would be developed as a new module within the Camera control unit. A more detailed discussion of the elements needed to accomplish this feature is described in the spring report.

Regarding requirement three and four stated above, the ability to save the current camera settings on the Raspberry Pi and to load and apply settings from a saved file is not implemented. The current implementation allows the user to fetch the applied setting on the cameras and change them to something else. Essentially the same functionality described in the requirements is achieved, but in a different way. Different settings can be stored on the laptop of the user. If these settings need to be applied to a camera multiple commands needs to be sent over the network resulting in the same outcome.

The live-view requirement has not been pursued since it is of little use when the the ability to control the camera position is not implemented. This requirement should be developed when a need for controlling the cameras becomes a priority or when reaching the camera becomes troublesome. There is, however, some support for live-view in Gphoto2, which makes the implementation possible with the current software layout.

Regarding requirement six, the software can access pictures taken on the camera but only picture taken via Gphoto2's internal capture function. This is done by predefining the location of the picture on the Raspberry Pi. This way of capturing images was later deemed too slow to be used when several cameras needed to trigger simultaneously. To solve this problem it is recommended to first set up a common storage for all the cameras where the picture can be stored. The most promising solution is to set up a NAS (Network-attached-storage) unit and connect it to the Ethernet Hub. This will enable all the Raspberry Pi:s to save their images to a common storage. The network setup needed would look like this.

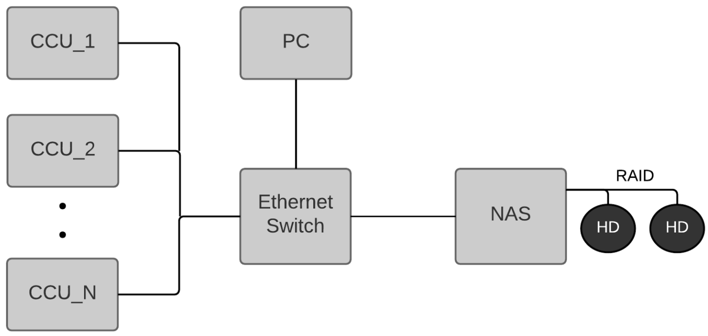


Figure 4.3: Camera software overview including the communication path

To use the network setup described in figure figure 4.3 effectively a program that fetches the images from the cameras and save them on the NAS would need to be developed. There are support for this in the Gphoto2 library. The program should be implemented as an member function in CameraClass described in section 2.5.5. The different cameras will be able to, simultaneously, transfer their images to the NAS over Ethernet. The proposed solution will offer backup if it is configured with a RAID setup, utilising multiple hard drives. This is not implemented at the company at the moment, which leaves them open for the risk of having to redo a shoot. Additionally establishing this functionality will prove useful if live-view is desired since a live-view can be achieved by continuously writing to a storage, which is accessible to the user.

4.1.9 Economics discussion

Since this product was designed as a part of a research study, the labour cost of the work done by the group members has not been calculated. If this was not the case it would constitute the lion's share of the product cost. Even though our labour is not charged it has been valuated when reviewing outsourcing cost to ensure that we maximise the utility of our knowledge and capabilities.

The production of PCBs has been the area in which outsourcing has been used to the greatest extent in the project. This is due to the limited capabilities provided at KTHs premises. This decision has proven to have a very positive impact on the product as the quality of the PCBs that Frikab manufactured by far exceeds the quality of products manufactured by the group. The specific choice of Frikab as supplier of PCBs was based on the support and delivery time they offered. In hindsight, prioritising these factors was crucial for our success. During the process of the manufacturing, our collaboration with Frikab intensified and we got the opportunity to utilise the machines that they have at their disposal to speed up the finalisation of the PCBs. Without their willingness to cooperate we would have had to focus more on hands-on work with the PCBs rather than developing the design of the product. Their exceptional service naturally showed in the pricing of their services, in a spot-check their service was priced approximately 15% higher than the cheapest competitor. This additional cost was considered acceptable due to the benefits that their service granted us.

The cost of the work that we outsourced to Frikab was greatly increased by the fast delivery time that was required due to the time limitations of the project. It would be possible to reduce this cost by approximately 30%. The cost of the PCBs was composed of fixed costs for configuration of machines and tooling and unit cost. Many of the designs were only ordered in a small number of units, causing the fixed costs to be a significant part of the total cost. If additional units of these designs are to be manufactured in collaboration with Frikab, only the unit cost will be needed to take into account since the programming and tooling related to the fixed cost already have been accomplished.

The fact that the labour cost of our group members is not being calculated makes the components mounted on the PCBs the greater part of the budget. Most of these components are in the mid-tier from a financial perspective. This was a level we chose in agreement with the company and it was largely decided on a case to case basis. To select components that were required in large numbers, and that we had high quality requirements on, standard components were investigated and their applicability to our product was tried. An example of this is the use of RJ12 connectors for the LED unit. This is a connector that is used in telecommunication products thus it is widely mass produced and the cost is relatively low given the quality of it.

4.2 Future work

The product that has been developed is considered to have the fundamental functionality to take the Pixel Grinder service to the next level. The mechanical and electronic properties of the product enables operations that fulfil the requirements presented and beyond. The company has expressed a wish of designing a similar, though stationary, light rig for their full-body scans. The mechanical structure is easily scalable, as it only requires proportionally longer rods. For the light distribution restraints, however, it is going to have to be confirmed that enough light sources are implemented (possibly by adding extra LED units to the rods).

A wide-ranging area where additional studies are suitable is in the field of embedded systems. It is expected that the product will serve as a platform for future studies focusing on more advanced control of distributed systems. The computational power available in the product far exceeds what is currently utilised, this enables the development and deployment logic with increased complexity. It is possible that the light could be used beneficially in other situations, for example in combination with high-speed cameras.

To further develop the logic of the distributed system the availability of the communication protocol is essential. The current use of the CAN is confined to sending of configurations and synchronisation signal. This suboptimal utilisation of the communication network further confirms that a development of the logic is possible and advisable. The design of a self-checking verification during start-up could be an example of such an improvement.

The software that has been developed for the product is case specific and is therefore highly dependent on the context in which it is supposed to operate. This is, however, not the case with the developed hardware. The units are all modular and the number of units can potentially be scaled as long as the total power consumption is within the limits of the power supply. It is therefore possible to extend and reschedule the hardware system with minor changes to the software.

Appendix A

Requirements

The requirements from Pixel Grinder can be summarised into two phases. Phase one concerns the light rig and phase two concerns the control of the cameras.

Stakeholder requirements

The Light Rig - To be able to create ideal lighting conditions

- The cameras shall not have a fixed position on the rig.
- The light rig shall allow photos to be taken through it.
- The rig shall be able to provide lighting for 5 synchronous taken photos with 5 different lighting conditions within 1 second.
- The light intensity shall be controllable.
- The rig shall be able to create lighting conditions equal to the first four spherical harmonics.
- The rig shall be able to produce both polarised and unpolarised light separately.
- The light must be strong enough to shoot with a narrow aperture of f13 and fast shutter of 1/200 second.
- The rig shall be disassemblable so that it can fit into a space of one or more Peli 1650 cases.
- The Peli case with all the equipment shall not weight more than 31kg each.
- The rig shall be compatible with different cameras.
- The rig shall be used for facial scanning.
- The equipment used should be easy to replace in case of failure.

- The rig should be possible to assemble in one day.

Camera control - To be able to control the cameras position and setting remotely.

- The cameras aim, position, zoom and focus shall be remotely controlled.
- The camera settings shall be remotely controlled.
- It must be possible to save the camera setup.
- Camera setups shall be able to be loaded from saved setups.
- The images taken by the cameras shall be accessible.

System requirements

The stakeholder requirements has here been translated into mechanical, software and hardware system requirements.
Mechanical

- The rig shall have openings where the cameras fit.
- The nodes shall be evenly distributed on a sphere around the object.
- The rig shall have an opening for the object to enter through.
- The diameter of the sphere shall be 1.5m.
- The number of light sources shall be at least 150.
- Any part of the rig shall not be bigger than that that it can fit in a Peli 1650 case.
- A single part shall not weight more than 31 kg.
- The cameras shall be able to move 120 degrees pitch and yaw.
- The camera position must be stable.
- The camera must be attached to the rig by a standard interface.
- There shall be a mapping for the assembling of the rig.
- The edges of the rig shall be made out of a standard profile.

Software

- It shall be possible to control the light intensity.
- It shall provide live view from the cameras to the user.

- It shall be possible to synchronise the cameras with the LEDs when taking pictures.
- It shall be possible to allow the user to move the cameras pitch, yaw and zoom.
- It shall provide communication between the nodes.
- It shall provide the ability of changing the setting of the camera.
- Each node should have a unique ID.
- The software shall not be camera model dependent.
- The software shall allow for settings to be saved.
- The software shall allow for settings to be loaded.
- The software shall be able to access the images from the cameras.

Hardware

- The LEDs intensity shall be adjustable.
- Each LED must be able to produce 4000 lm.
- The LED shall have a CRI at least of 90 .
- The connections and cables between the nodes must be standard.
- The connections shall be mapped.
- The frequency which set the intensity of the LEDs shall be 8000Hz.
- Each node needs at least 60W .
- Each camera shall have 12V.
- Both polarized and unpolarized LEDs shall be used.
- Hardware interfaces shall not be camera model dependent.
- The camera nodes shall provide control signal for controlling the position of the camera.
- There shall be a memory for saving settings.

Appendix B

Microcontroller Comparison

Root Node

Table B.1: Root Node summary

MCU	Interfaces	I/O Pins	PWMs	Power	Ease of Use	Price (SEK)
mBed LPC4088	CAN[1], Ethernet, USB, UART, SPI, I2C	2 × 22	6	N/A	Moderate	718
BeagleBone Black	Ethernet USB UART SPI I2C	2 × 41	7	N/A	A	519
mBed Nucleo F303RE	CAN[1], USB, Ethernet (Shield)[3], USB, UART, SPI, I2C	51	17	N/A	Hard	88

LED Node

Table B.2: LED Node summary

MCU	Interfaces	I/O Pins	PWMs	Power	Ease of Use	Price (SEK)
AT90CAN32	CAN[1], SPI, UART	53	7	N/A	Must be soldered	64
ATMega16M1	CAN[1], SPI, UART	27	10	N/A	Must be soldered	38
mBed Nucleo F303RE	CAN[1], USB Ethernet (Shield)[3], USB, UART, SPI, I2C	51	17	N/A	Hard	88
Arduino Uno	CAN (Shield)[2], USB, UART, Ethernet (Shield)[3], SPI, I2C	14	6	N/A	Easy	279

[1]MCP2551 CAN Transiever needed - Cost 11sek
[2]Arduino UNO plus MCP2515 SPI-CAN Module - Cost: 279+50 = 329sek
[3]mBed Nucleo-F303RE plus Arduino Ethernet Shield - Cost: 88+399=487sek

Camera Node

Table B.3: Camera Node summary

MCU	Interfaces	I/O Pins	PWMs	Power	Ease of Use	Price (SEK)
BeagleBone Black	Ethernet, USB, UART, SPI, I2C	2 × 41	7	N/A	Moderate	519
Raspberry Pi 2	Ethernet, USB, UART, SPI, I2C	40	0	N/A	Moderate	399

Gimbal Node

Table B.4: Gimbal Node summary

MCU	Interfaces	I/O Pins	PWMs	Power	Ease of Use	Price (SEK)
mBed Nucleo F303RE	CAN [1], USB, UART, Ethernet (Shield)[3], SPI, I2C	51	17	N/A	Moderate	88
Arduino Micro	USB, UART, SPI, I2C	20	7	N/A	Easy	150

Appendix C

LED Comparison

Table C.1: Comparison on the properties of 19 different high-power LEDs

Manufacturer	**Manufacturer article number**	**Lumens**	**Voltage (V)**	**Current (mA)**	**Power (W)**	**Lm/W ratio**	**LEDs required**	**Total Power (kW)**
Lumileds	LHC1-4090-1202	850	35.5	200	7.1	119.7	4.71	5.21
Lumileds	LHC1-4090-1204	1800	35.5	450	15.975	112.7	2.22	5.54
Lumileds	LHC1-4090-1208	3600	35.5	900	31.95	112.7	1.11	5.54
Lumileds	LHC1-4090-1203	1200	35.5	300	10.65	112.7	3.33	5.54
Lumileds	LHC1-4090-1205	2400	35.5	600	21.3	112.7	1.67	5.54
Lumileds	L2C3-4090109E06000	925	35.5	250	8.875	104.2	4.32	5.99
Luminus Devices	CHM-14-40-90-36-AC00-F2-3	2550	35	720	25.2	101.2	1.57	6.17
Lumileds	L2C1-4090120206A00	700	35.5	200	7.1	98.6	5.71	6.33
OSRAM	GW KAHNB1.CM-TUUQ-40S3-T02	4190	44	1050	46.2	90.7	0.95	6.88
OSRAM	GW KAGHB1.CM-RSRU-40H3	1400	32	500	16	87.5	2.86	7.13
Sharp	GW6DGA40NFC	1270	37	400	14.8	85.8	3.15	7.27
Sharp	GW6DGC40NFC	2120	37	700	25.9	81.9	1.89	7.62
Sharp	GW6DGD40NFC	2840	37	950	35.15	80.8	1.41	7.72
OSRAM	GW MAFJB1.CM-RUSS-40S3	1640	29	700	20.3	80.8	2.44	7.72
Sharp	GW6DGE40NFC	3770	50	950	47.5	79.4	1.06	7.86
OSRAM	GW MAEGB1.CM-QPQS-40S3-0-T02	710	26	350	9.1	78.0	5.63	8.00
Sharp	GW6BGR40HED	890	36	320	11.52	77.3	4.49	8.08
Sharp	GW6BGS40HED	1080	36	400	14.4	75.0	3.70	8.32
OSRAM	GW MAGMB1.CM-TQTT-40S3-1050-T02	3040	39	1050	40.95	74.2	1.32	8.41

Appendix D

PCB Schematics

D.1 LED unit schematics

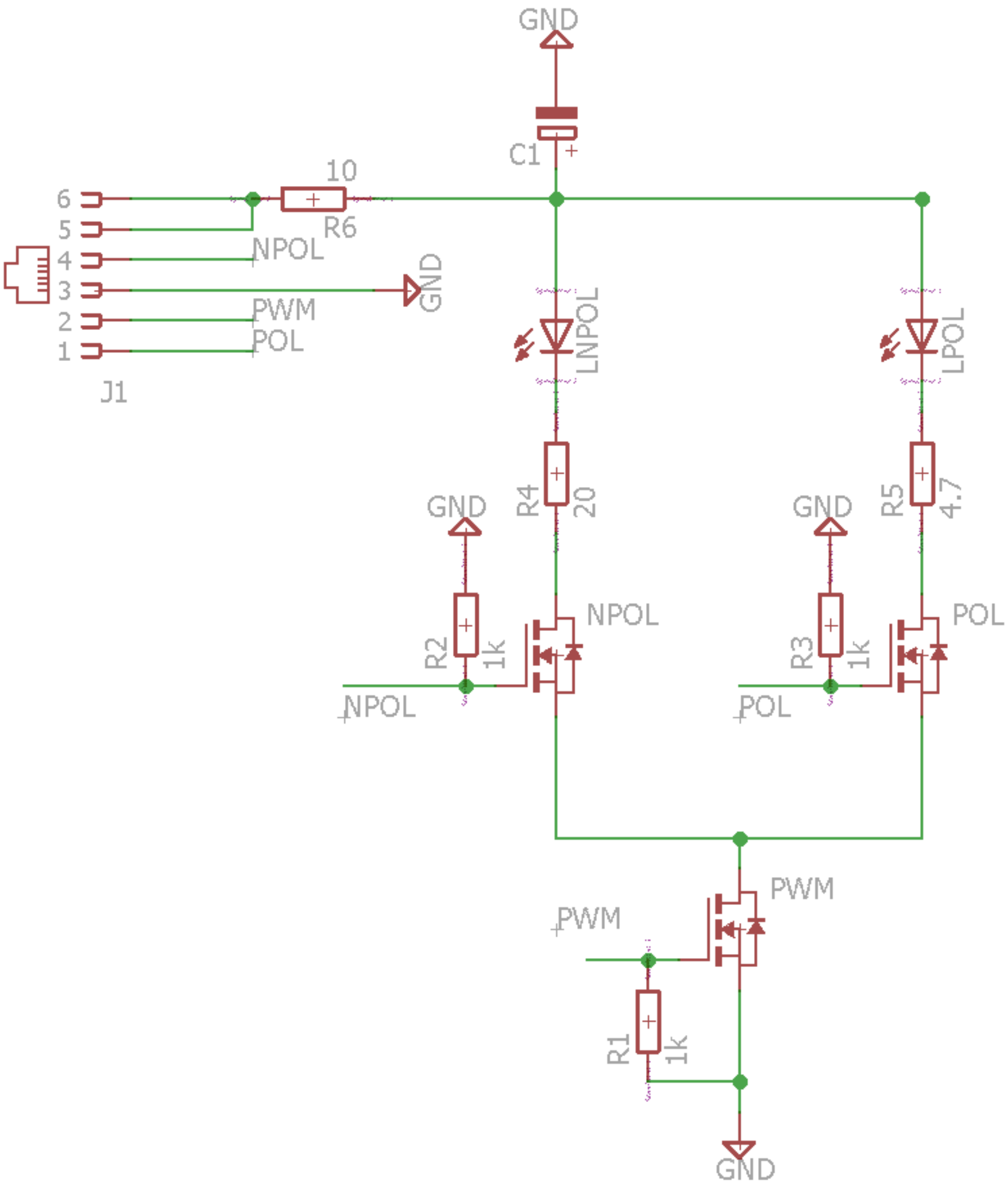


Figure D.1: LED unit Schematics.

D.2 LCU schematics

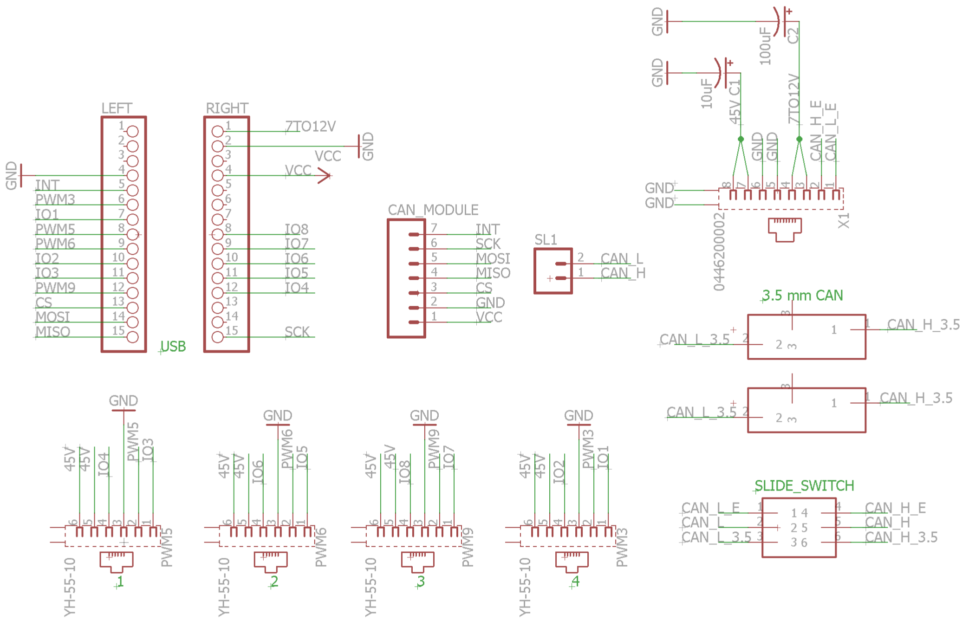


Figure D.2: LCU Schematics.

D.3 Charging circuit schematics

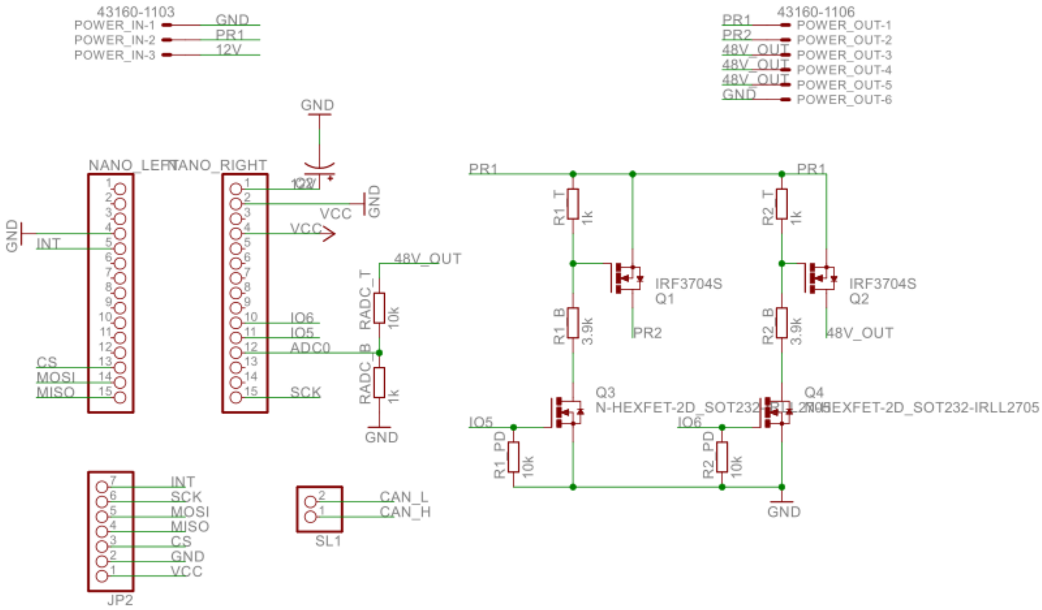


Figure D.3: Charging circuit schematics.

D.4 Connection plate schematics

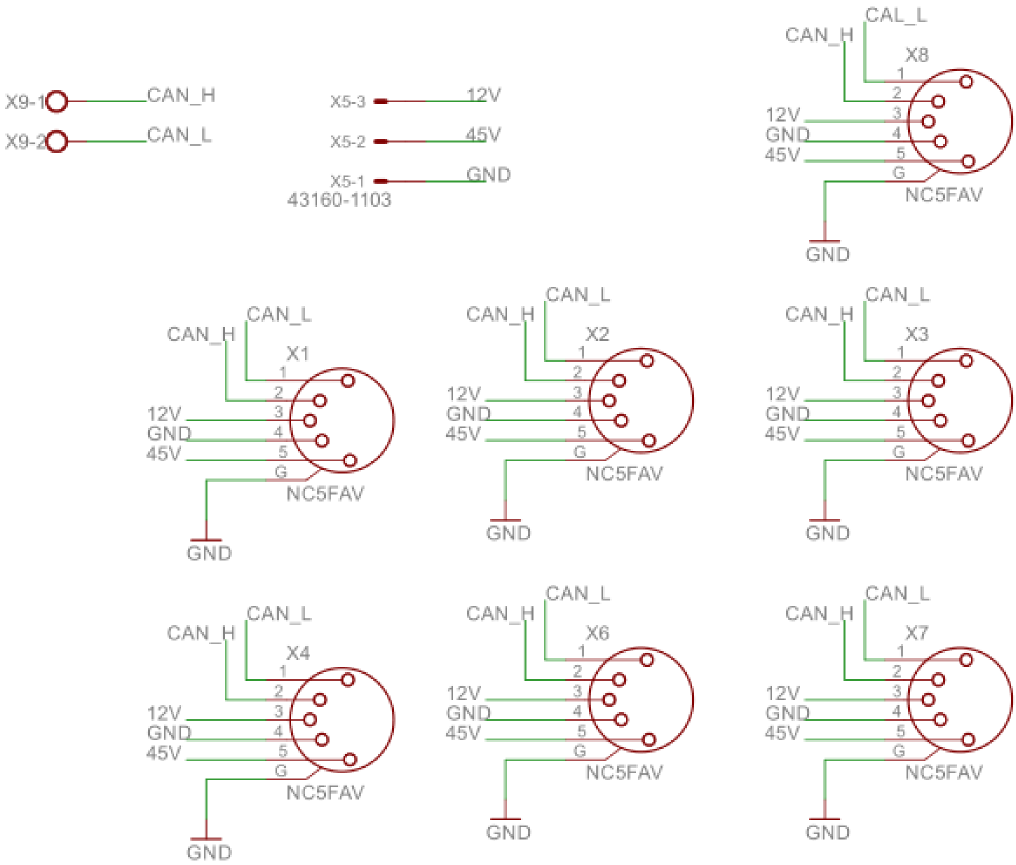


Figure D.4: Connection plate schematics.

Appendix E

LCU-LED Logistics and Coordinates

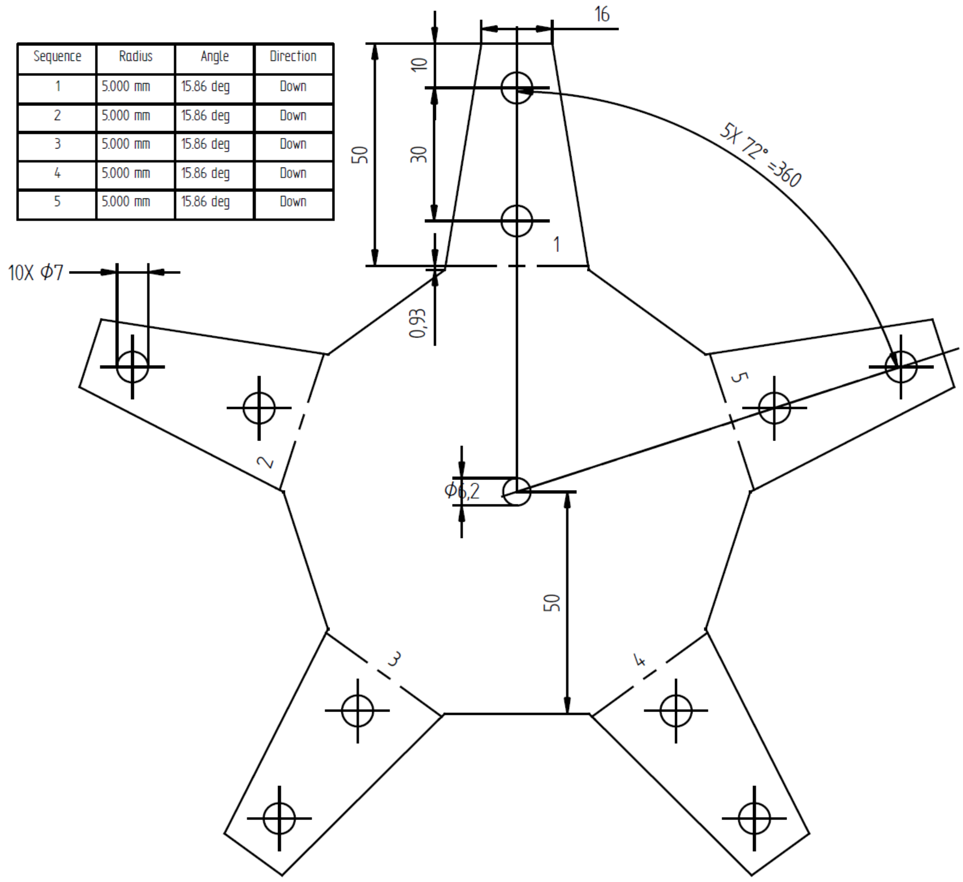
LCU	LED	Coordinates
1	1	[0, 0, 50],
2	2	[0, 26.2866, 42.5325],
	3	[0, 13.1433, 46.2663],
	4	[12.5, 17.2048, 42.5325],
3	5	[25, 8.123, 42.5325],
	6	[12.5, 4.0615, 46.2663],
	7	[20.2254, -6.5716, 42.5325],
4	8	[15.4508, -21.2663, 42.5325],
	9	[7.7254, -10.6331, 46.2663],
	10	[0, -21.2663, 42.5325],
5	11	[-15.4508, -21.2663, 42.5325],
	12	[-7.7254, -10.6331, 46.2663],
	13	[-20.2254, -6.5716, 42.5325],
6	14	[-25, 8.123, 42.5325],
	15	[-12.5, 4.0615, 46.2663],
	16	[-12.5, 17.2048, 42.5325],
7	17	[0, 44.7214, 22.3607],
	18	[-7.7254, 46.1371, 11.1803],
	19	[-12.5, 39.5655, 24.3236],
	20	[0, 35.504, 32.4466],
8	21	[25, 34.4095, 26.2866],
	22	[12.5, 39.5655, 24.3236],
	23	[12.5, 30.3481, 34.4095],
	24	[25, 21.2663, 34.4095],
9	25	[42.5325, 13.8197, 22.3607],
	26	[41.4917, 21.6045, 11.1803],
	27	[33.7663, 24.1146, 24.3236],
	28	[33.7663, 10.9713, 32.4466],
10	29	[40.4508, -13.1433, 26.2866],
	30	[41.4917, 0.33819, 24.3236],
	31	[32.7254, -2.5101, 34.4095],
	32	[27.9508, -17.2048, 34.4095],
11	33	[26.2866, -36.1803, 22.3607],
	34	[33.3687, -32.7848, 11.1803],
	35	[33.3687, -24.6618, 24.3236],
	36	[20.8687, -28.7233, 32.4466],
12	37	[0, -42.5325, 26.2866],
	38	[13.1433, -39.3564, 24.3236],
	39	[7.7254, -31.8994, 34.4095],
	40	[-7.7254, -31.8994, 34.4095],
13	41	[-26.2866, -36.1803, 22.3607],
	42	[-20.8687, -41.8666, 11.1803],
	43	[-13.1433, -39.3564, 24.3236],
	44	[-20.8687, -28.7233, 32.4466],
14	45	[-40.4508, -13.1433, 26.2866],
	46	[-33.3687, -24.6618, 24.3236],
	47	[-27.9508, -17.2048, 34.4095],
	48	[-32.7254, -2.5101, 34.4095],
15	49	[-42.5325, 13.8197, 22.3607],
	50	[-46.2663, 6.9098, 11.1803],
	51	[-41.4917, 0.33819, 24.3236],
	52	[-33.7663, 10.9713, 32.4466],
16	53	[-25, 34.4095, 26.2866],
	54	[-33.7663, 24.1146, 24.3236],
	55	[-25, 21.2663, 34.4095],
	56	[-12.5, 30.3481, 34.4095],
17	57	[15.4509, 47.5528, 0],
	58	[0, 47.5528, 0],
	59	[7.7254, 46.1371, 11.1803],
	60	[20.2254, 40.9812, 13.1433],
18	61	[40.4509, 29.3893, 0],
	62	[32.7254, 31.8994, -13.1433],
	63	[27.9508, 38.471, 0],
	64	[32.7254, 31.8994, 13.1433],
19	65	[50, 0, 0],
	66	[45.2254, 14.6946, 0],
	67	[46.2663, 6.9098, 11.1803],
	68	[45.2254, -6.5716, 13.1433],
20	69	[40.4509, -29.3893, 0],
	70	[40.4509, -21.2663, -13.1433],
	71	[45.2254, -14.6946, 0],
	72	[40.4509, -21.2663, 13.1433],
21	73	[15.4509, -47.5528, 0],
	74	[27.9508, -38.471, 0],
	75	[20.8687, -41.8666, 11.1803],
	76	[7.7254, -45.0427, 13.1433],
22	77	[-15.4509, -47.5528, 0],
	78	[-7.7254, -45.0427, -13.1433],
	79	[0, -47.5528, 0],
	80	[-7.7254, -45.0427, 13.1433],

LCU	LED	Coordinates
23	81	[-40.4509, -29.3893, 0],
	82	[-27.9508, -38.471, 0],
	83	[-33.3687, -32.7848, 11.1803],
	84	[-40.4509, -21.2663, 13.1433],
24	85	[-50, 0, 0],
	86	[-45.2254, -6.5716, -13.1433],
	87	[-45.2254, -14.6946, 0],
	88	[-45.2254, -6.5716, 13.1433],
25	89	[-40.4509, 29.3893, 0],
	90	[-45.2254, 14.6946, 0],
	91	[-41.4917, 21.6045, 11.1803],
	92	[-32.7254, 31.8994, 13.1433],
26	93	[-15.4509, 47.5528, 0],
	94	[-20.2254, 40.9812, -13.1433],
	95	[-27.9508, 38.471, 0],
	96	[-20.2254, 40.9812, 13.1433],
27	97	[0, 44.7214, -22.3607],
	98	[-12.5, 39.5655, -24.3236],
	99	[-7.7254, 46.1371, -11.1803],
	100	[7.7254, 46.1371, -11.1803],
28	101	[25, 34.4095, -26.2866],
	102	[12.5, 30.3481, -34.4095],
	103	[12.5, 39.5655, -24.3236],
	104	[20.2254, 40.9812, -13.1433],
29	105	[42.5325, 13.8197, -22.3607],
	106	[33.7663, 24.1146, -24.3236],
	107	[41.4917, 21.6045, -11.1803],
	108	[46.2663, 6.9098, -11.1803],
30	109	[40.4509, -13.1433, -26.2866],
	110	[0, 21.2663, -34.4095],
	111	[41.4917, 0.33819, -24.3236],
	112	[45.2254, -6.5716, -13.1433],
31	113	[26.2866, -36.1803, -22.3607],
	114	[33.3687, -24.6618, -24.3236],
	115	[33.3687, -32.7848, -11.1803],
	116	[20.8687, -41.8666, -11.1803],
32	117	[0, -42.5325, -26.2866],
	118	[-4.7746, 6.5716, -34.4095],
	119	[13.1433, -39.3564, -24.3236],
	120	[7.7254, -45.0427, -13.1433],

LCU	LED	Coordinates
33	121	[-26.2866, -36.1803, -22.3607],
	122	[-13.1433, -39.3564, -24.3236],
	123	[-20.8687, -41.8666, -11.1803],
	124	[-33.3687, -32.7848, -11.1803],
34	125	[-40.4509, -13.1433, -26.2866],
	126	[-20.2254, 6.5716, -34.4095],
	127	[-33.3687, -24.6618, -24.3236],
	128	[-40.4509, -21.2663, -13.1433],
35	129	[-42.5325, 13.8197, -22.3607],
	130	[-41.4917, 0.33819, -24.3236],
	131	[-46.2663, 6.9098, -11.1803],
	132	[-41.4917, 21.6045, -11.1803],
36	133	[-25, 34.4095, -26.2866],
	134	[-25, 21.2663, -34.4095],
	135	[-33.7663, 24.1146, -24.3236],
	136	[-32.7254, 31.8994, -13.1433],
37	137	[0, 26.2866, -42.5325],
	138	[-12.5, 17.2048, -42.5325],
	139	[-12.5, 30.3481, -34.4095],
	140	[0, 35.504, -32.4466],
38	141	[25, 8.123, -42.5325],
	142	[12.5, 17.2048, -42.5325],
	143	[-7.7254, -2.5101, -34.4095],
	144	[-8.7663, 10.9713, -32.4466],
39	145	[15.4509, -21.2663, -42.5325],
	146	[20.2254, -6.5716, -42.5325],
	147	[-12.5, -17.2048, -34.4095],
	148	[-13.5408, -3.7233, -32.4466],
40	149	[-15.4509, -21.2663, -42.5325],
	150	[0, -21.2663, -42.5325],
	151	[-27.9508, -17.2048, -34.4095],
	152	[-28.9917, -3.7233, -32.4466],
41	153	[-25, 8.123, -42.5325],
	154	[-20.2254, -6.5716, -42.5325],
	155	[-32.7254, -2.5101, -34.4095],
	156	[-33.7663, 10.9713, -32.4466],

Appendix F

Mechanical structure: Node drafts



Sequence	Radius	Angle	Direction
1	5.000 mm	15.86 deg	Down
2	5.000 mm	15.86 deg	Down
3	5.000 mm	15.86 deg	Down
4	5.000 mm	15.86 deg	Down
5	5.000 mm	15.86 deg	Down

Figure F.1: A draft of the node for five rods attachment.

Sequence	Radius	Angle	Direction
1	5.000 mm	15.86 deg	Down
2	5.000 mm	18.00 deg	Down
3	5.000 mm	15.86 deg	Down
4	5.000 mm	18.00 deg	Down
5	5.000 mm	18.00 deg	Down
6	5.000 mm	18.00 deg	Down

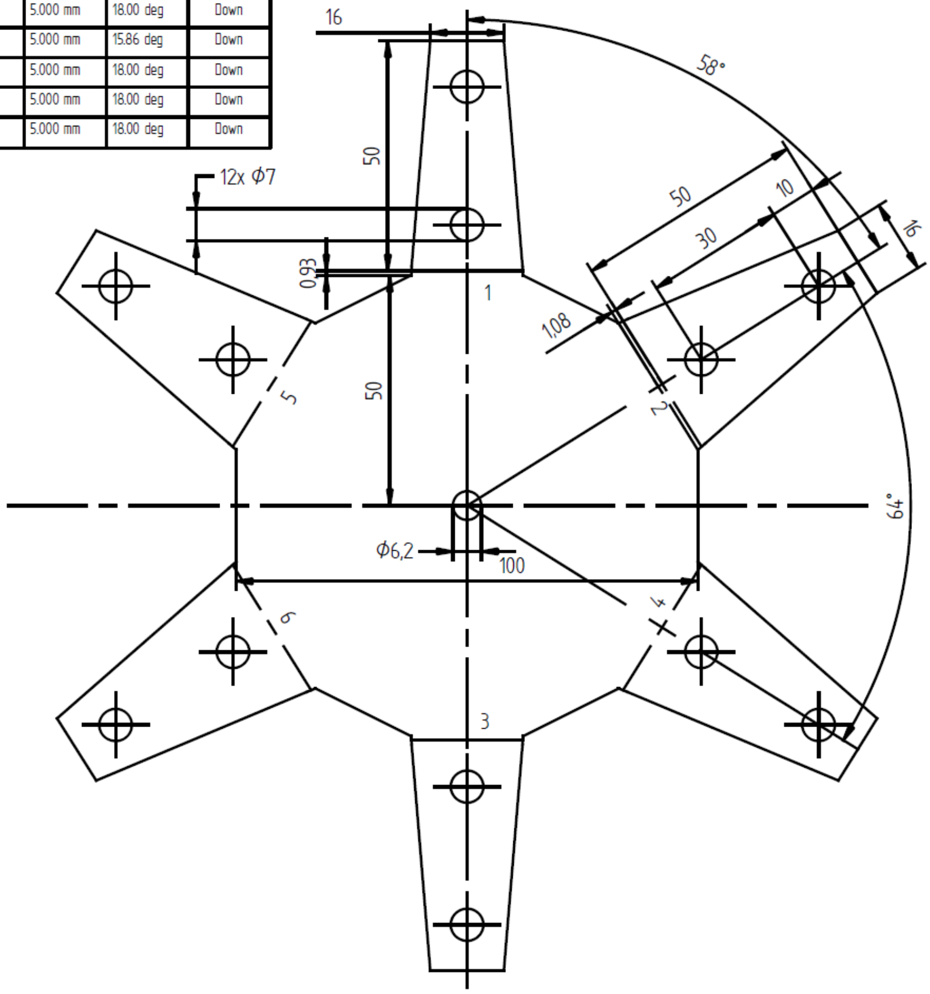


Figure F.2: A draft of the node for six rods attachment.

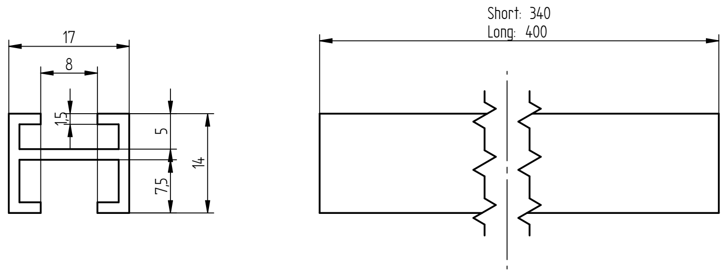
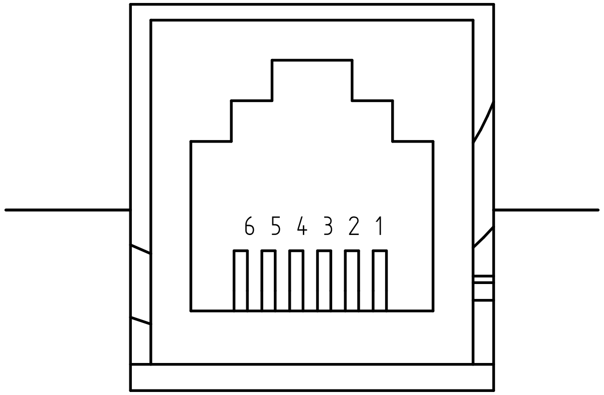


Figure F.3: Profile and length of the rods.

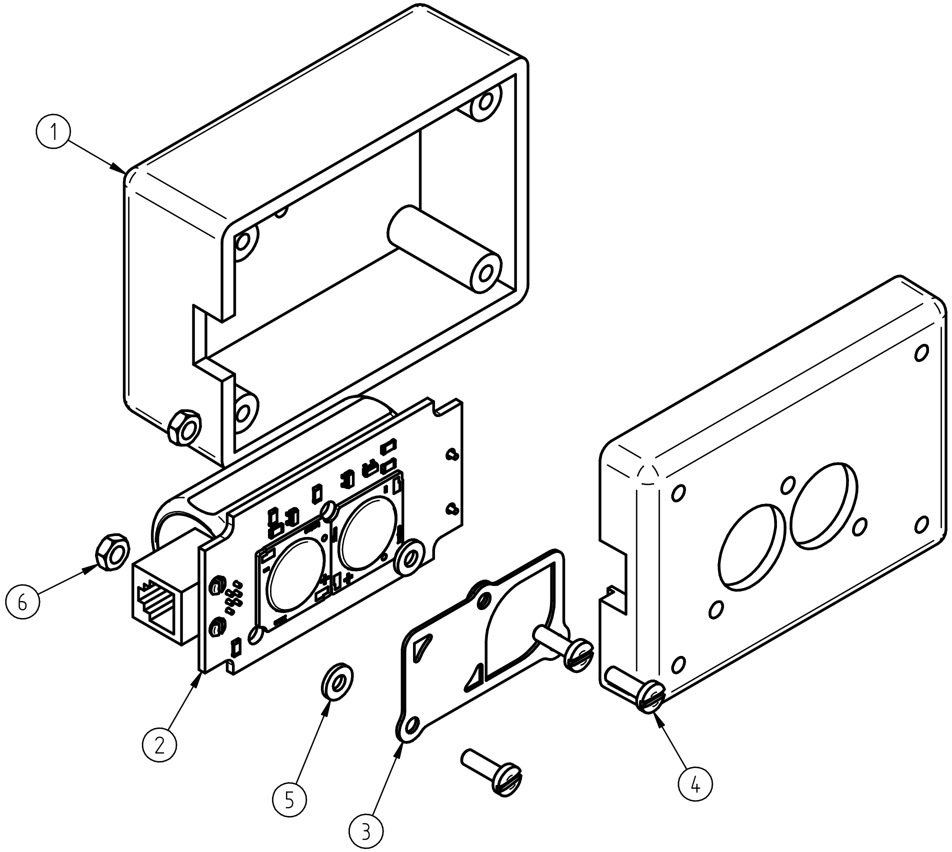
Appendix G

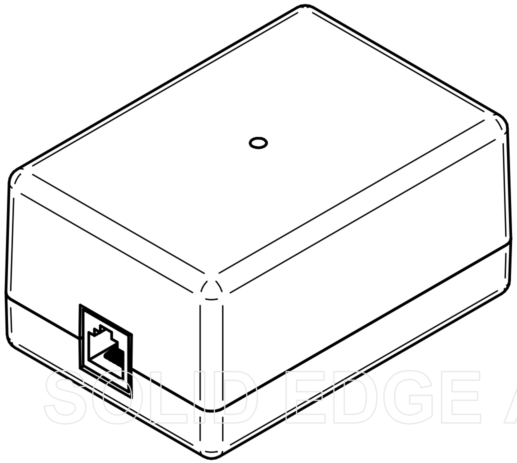
LED Case Assembly

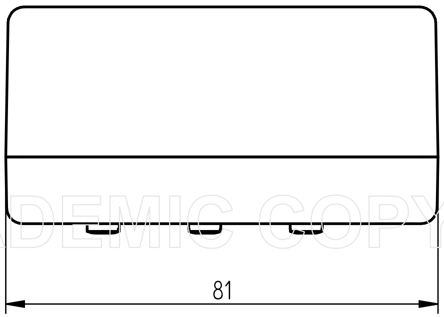
Connection interface	
Pin No.	Signal
1	Polarised LED digital
2	PWM
3	GND
4	non-polarised LED digital
5	48V
6	48V

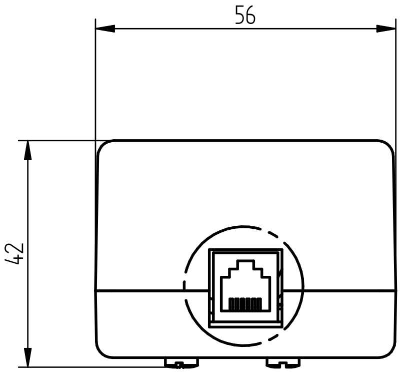


DETAIL A
Connection interface







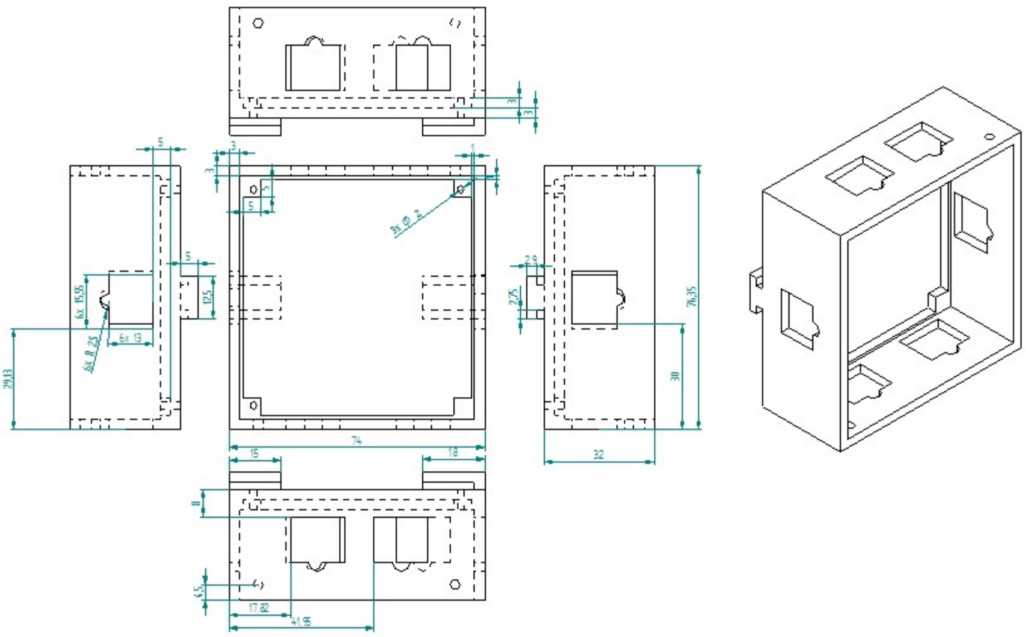


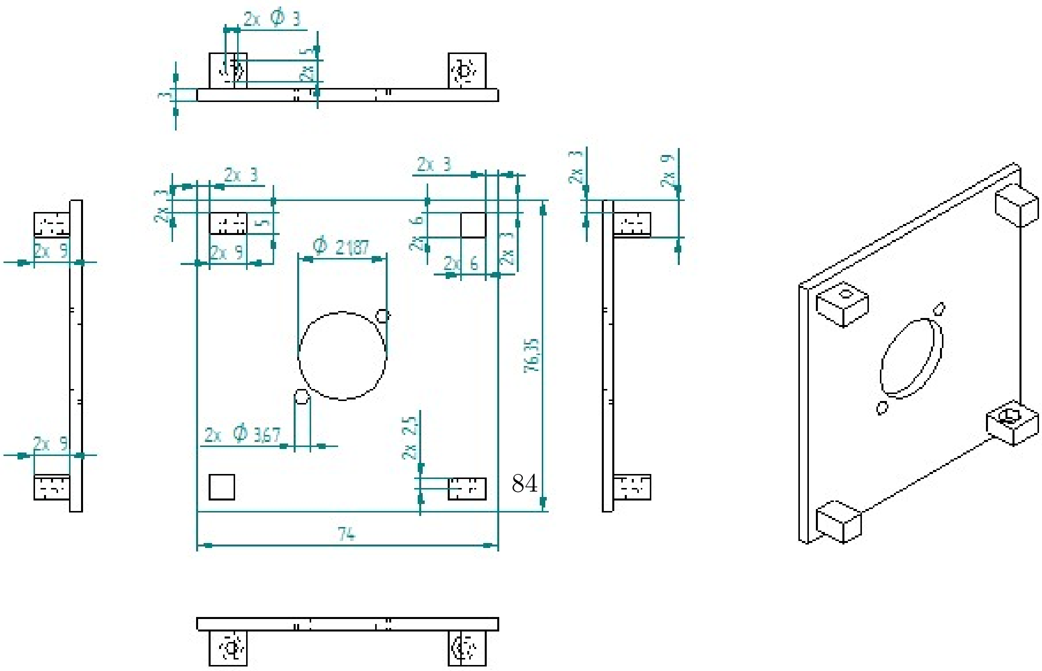
A

Item Number	File Name (no extension)	Quantity
1	Case_Hammond1594B	1
2	LedUnit	1
3	polarisingFilter	1
4	screw_ISO1580_M3x10	3
5	m3washer_thick	3
6	Nut_M3	3

Appendix H

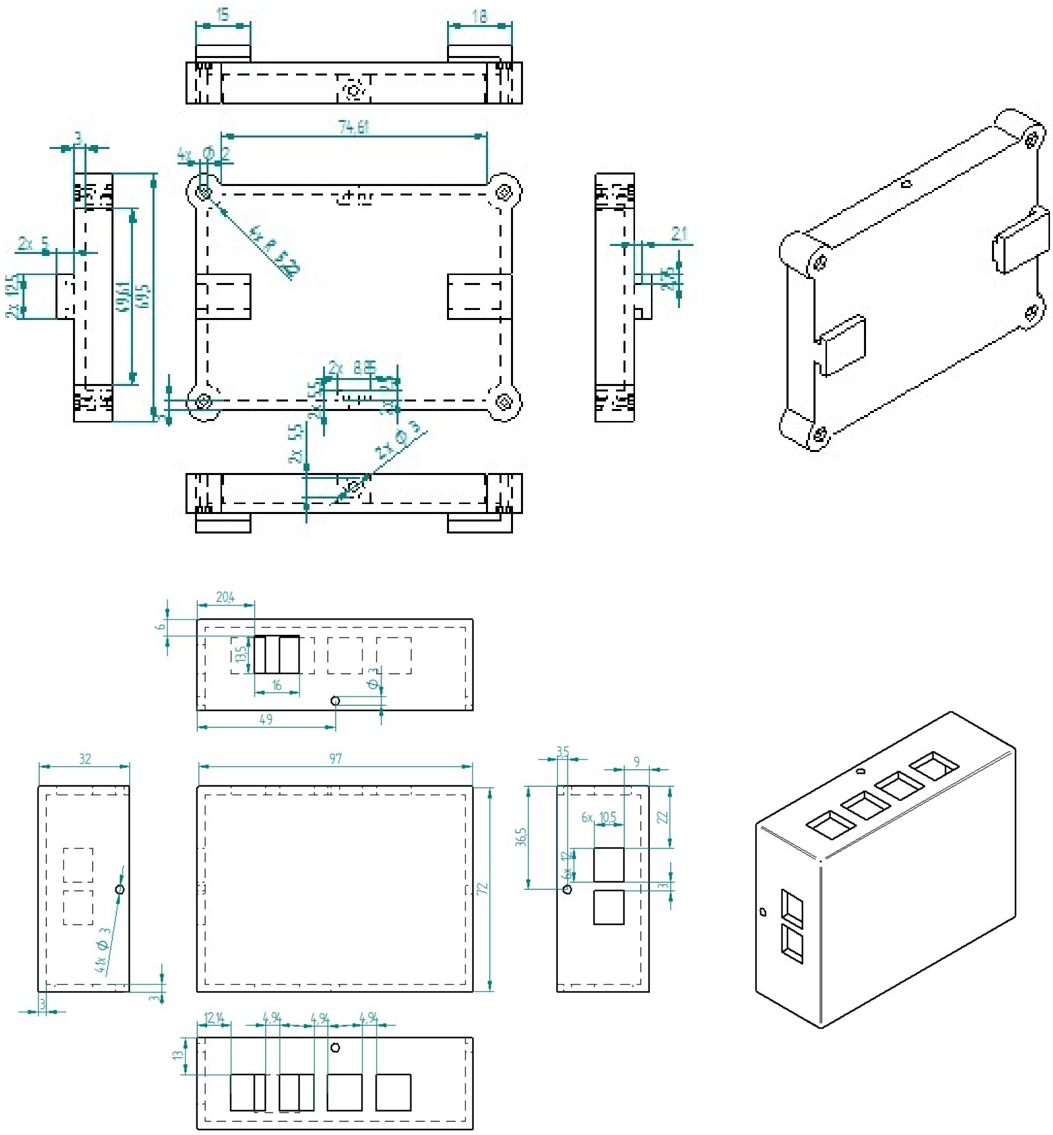
PDSU Case Draft





Appendix I

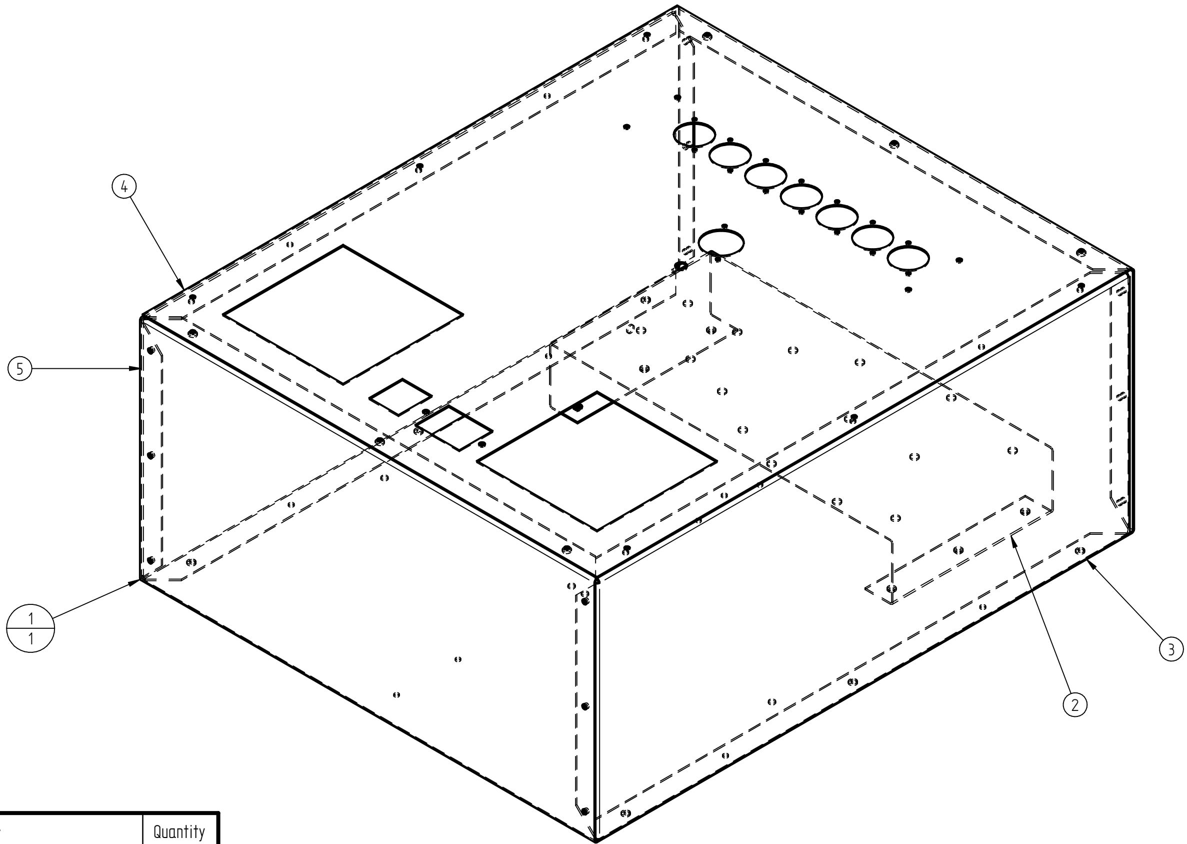
LCU Case Draft



Appendix J

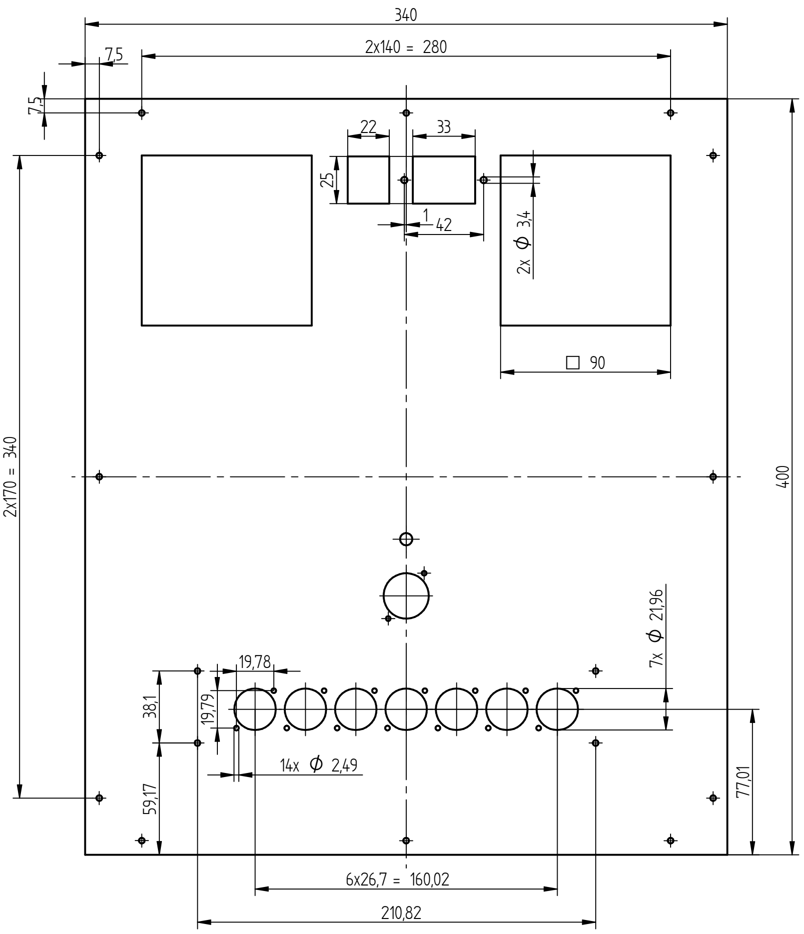
LED Case Assembly

J.1 Overview



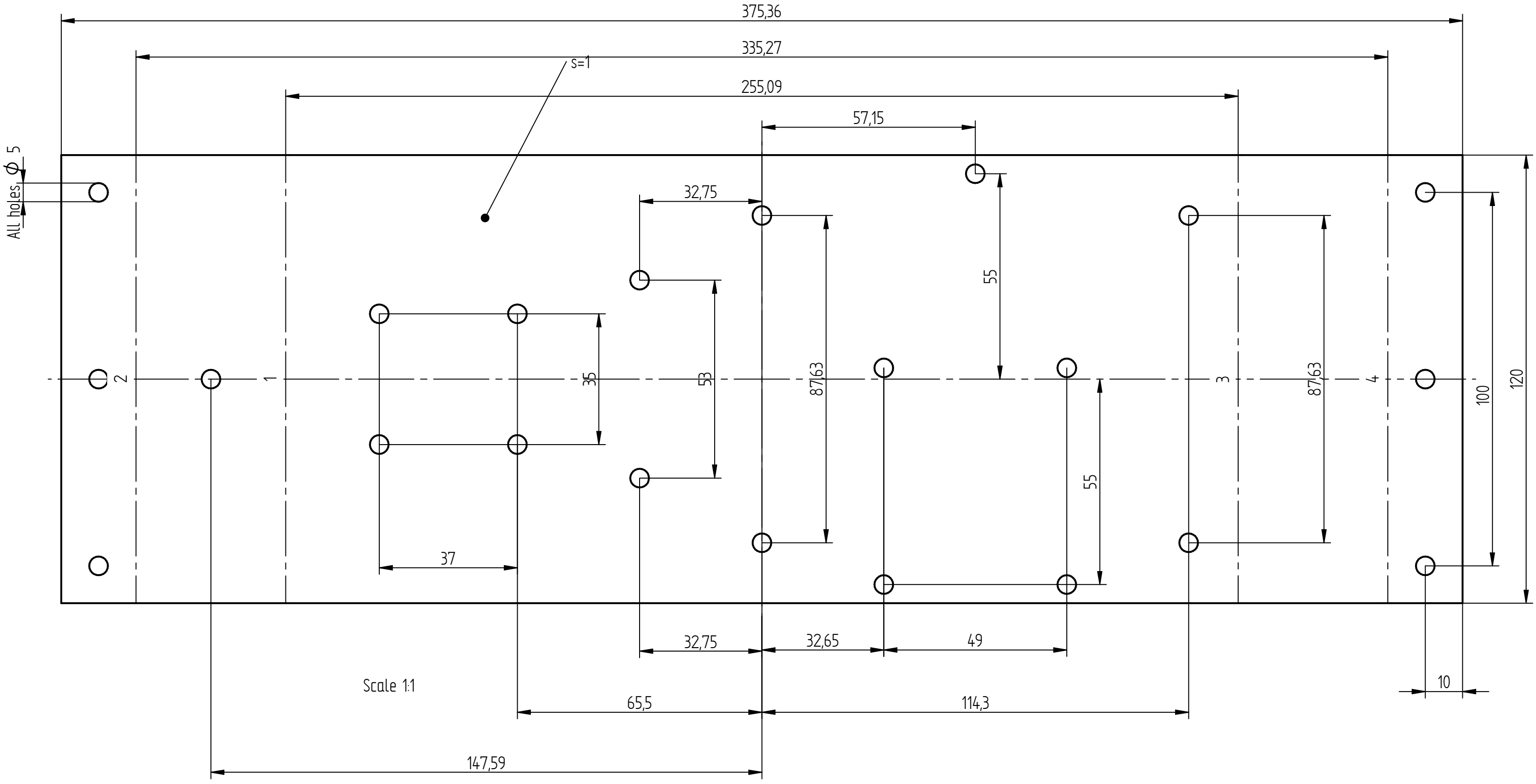
Item Number	File Name (no extension)	Author	Quantity
1	backPlate	Victor	1
2	rack	Victor	1
3	sides	Victor	1
4	lid	Victor	1
5	sides_mir1	Victor	1

J.2 Lid


340
2x140 = 280
7,5
7,5
22
33
25
1
42
2x Ø 3,4
□ 90
2x170 = 340
400
7x Ø 21,96
19,78
38,1
19,79
14x Ø 2,49
59,17
77,01
6x26,7 = 160,02
210,82

Scale: 1:2

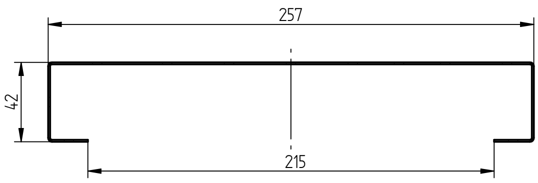
J.3 Rack



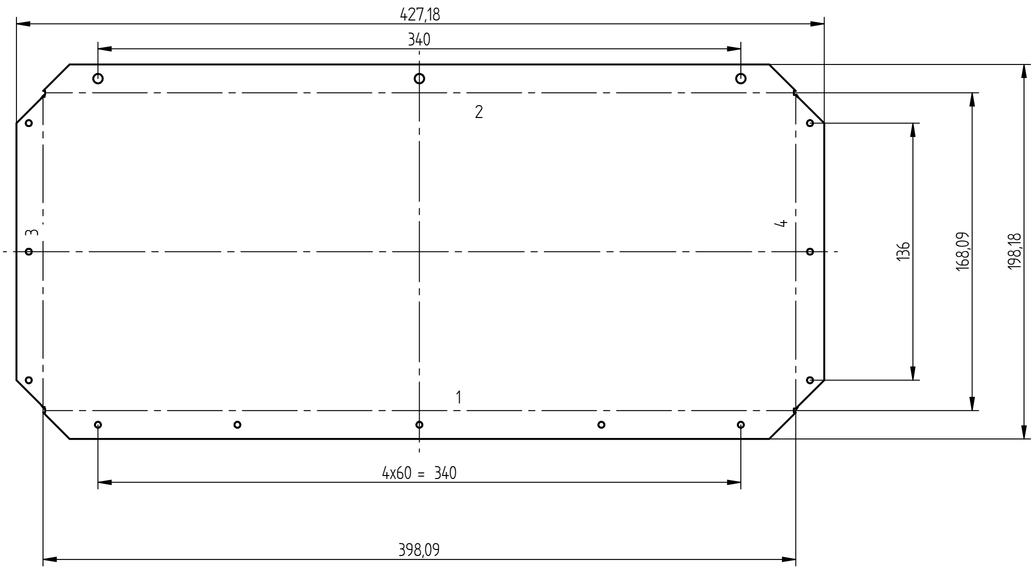
Scale 1:1

Sequence	Feature	Radius	Angle	Direction	Included Angle
1	Flange 1	1.000 mm	90.00 deg	Down	90.00 deg
2	Flange 2	1.000 mm	90.00 deg	Down	90.00 deg
3	Mirror 2	1.000 mm	90.00 deg	Down	90.00 deg
4	Mirror 2	1.000 mm	90.00 deg	Down	90.00 deg

Scale 1:2



J.4 Sides



There are two different sideplates. The layout is the same for them both but the bends different.
In the bend table first direction is the first side, second is the second.

Sequence	Feature	Radius	Angle	Direction	Included Angle
1	Flange 11	1.000 mm	90.00 deg	Down, Up	90.00 deg
2	Flange 13	1.000 mm	90.00 deg	Down, Up	90.00 deg
3	Flange 14	1.000 mm	90.00 deg	Down, Up	90.00 deg
4	Flange 15	1.000 mm	90.00 deg	Down, Up	90.00 deg

Appendix K

Total expenses

Cost of Product

LED Unit

Cost Object	Unit Price	Quantity	Total
PCB start cost	700,0	1	700,0
PCB programming labor	2300,0	1	2 300,0
PCB manufacturing	75,0	160	12 000,0
Capacitors	47,2	160	7 552,0
Fastenings	11,0	160	1 760,0
Casing	55,8	160	8 928,0
High Power LEDs	78,0	320	24 960,0
Subtotal			**58 200,0**

LED Control Unit

Cost Object	Unit Price	Quantity	Total
PCB start cost	800,0	1	800,0
PCB manufacturing and cor	98,0	45	4 410,0
Modular jack, RJ12	10,2	180	1836
Modular jack, RJ45	25,4	45	1 143,0
Capacitors	1,4	82	114,8
Slide switch	5,75	45	258,75
Audio jack, 3.5mm	7	90	630
Arduino nano	20	43	868,6
CAN modules	29	43	1 255,6
Subtotal			**11 316,8**

Cables

Cost Object	Unit Price	Quantity	Total
6-wire RJ12 0.5 m	15,5	160	2 480,0
8-wire RJ45 0.5 m	73,8	39	2 878,2
8-wire RJ45 2 m	120,0	4	480,0
8-wire RJ45 5 m	190,0	20	3 800,0
5-wire XLR 3 m	200,0	7	1 400,0
Subtotal			**11 038,2**

Power & Data Distribution

Cost Object	Unit Price	Quantity	Total
Modular jack, RJ45	25,4	49	1244,6
Capacitor	1,4	14	19,6
XLR 5 pin mount female	44,2	7	309,4
3.5 mm 3 P	22,6	1	22,6
Coaxial cable	58,6	1	58,6
Connection wire 0.75	3,9	3	11,6

Connection wire 0.88	10,0	3	29,9
Connection wire 0.88	10,0	3	30,0
Connection wire 0.88	10,0	3	29,9
Connection wire 0.88	10,0	3	29,9
XLR 5 mount female	35,2	10	352,0
Cut strip	0,7	100	70,0
Power connector 3P	12,0	6	72,0
Crimp housing 3P	5,0	6	30,0
Power connector 6P	24,0	3	72,0
Crimp housing 6P	7,0	3	21,0
DIN rail 35x150 mm	26,5	2	53,0
IEC cord 16 A	208,0	1	208,0
C20 mount male	27,0	1	27,0
Wirewound resistor	66,4	1	66,4
Wirewound resistor	66,4	1	66,4
Transistor P-channel	22,0	10	220,0
Transistor N-channel	8,3	10	82,8
Resistor SMD 1kOhm	1,2	10	12,3
Resistor SMD 3.9kOhm	1,2	10	12,3
Resistor SMD 10kOhm	1,2	10	12,3
DIN power supply 12V	310,0	2	620,0
DIN power supply 48V	1160,0	2	2 320,0
PCB manufacuring	5500	1	5500
Subtotal			**11 603,4**
Total			92 158,4

Cost of prototyping

Power Distribution

Component	Unit price	Quantity	Total price
XLR 5 cable plug	49,4	1	49,4
XLR 5 cable socket	84,7	1	84,7
XLR 5 mount	44,2	1	44,2
XLR 5 mount female	44,2	1	44,2
Modular jack, RJ45	25,4	12	304,8
3.5 mm 3 P	22,6	1	22,6
Coaxial cable	58,6	1	58,6
Connection wire 0.75	3,9	3	11,6
Connection wire 0.88	10,0	3	29,9
Connection wire 0.88	10,0	3	30,0
Connection wire 0.88	10,0	3	29,9
Connection wire 0.88	10,0	3	29,9
XLR 5 mount female	35,2	10	352,0
PCB 160x100mm	76,1	2	152,2
PCB 233x100mm	185,0	1	185,0
Cut strip	0,7	100	70,0
Power connector 3P	12,0	6	72,0
Crimp housing 3P	5,0	6	30,0
Power connector 6P	24,0	3	72,0
Crimp housing 6P	7,0	3	21,0
DIN rail 35x150 mm	26,5	2	53,0
IEC cord 16 A	208,0	1	208,0
C20 mount male	27,0	1	27,0
Wirewound resistor	66,4	1	66,4
Wirewound resistor	66,4	1	66,4
Transistor P-channel	22,0	10	220,0
Transistor N-channel	8,3	10	82,8
Resistor SMD 1kOhm	1,2	10	12,3
Resistor SMD 3.9kOhm	1,2	10	12,3
Resistor SMD 10kOhm	1,2	10	12,3
DIN power supply 12V	310,0	2	620,0
DIN power supply 48V	1160,0	2	2 320,0
Subtotal			**5 394,3**

LED Units

Component	Unit price	Quantity	Total Price
High power LEDs	98,0	25	2 450,0
Transistor N-channel	7,7	50	385,0
Capacitor 6800 uF	66,1	15	991,5
Transistor SMD N-channel	3,8	10	38,2
Resistor SMD 4.7Ohm	0,5	3	1,4
Resistor SMD 1kOhm	2,5	10	25,0
Resistor SMD 10Ohm	2,5	3	7,5
Resistor SMD 20Ohm	0,8	3	2,5
Modular jack, RJ12	13,8	6	82,8
Telephone cable, RJ12	27,7	1	27,7
Fastening M3	126,2	1	126,2
Screw M3	26,4	1	26,4
Casing	67,3	3	201,9
Nuts M3	18,8	1	18,8
Subtotal			**4 384,9**

LED Control Units

Component	Unit price	Quantity	Total Price
Hylslist	14,9	50	745,0
3.5 mm mono	7,0	10	70,0
Nucleon dev card	83,6	2	167,2
CAN tranceiver MCP2551	11,0	2	22,0
Arduino Micro	166,0	4	664,0
High power LEDs	47,7	5	238,5
Modular jack, RJ12	12,1	10	121,0
High power LEDs	46,5	5	232,5
Arduino ethernet shield	267,0	3	801,0
Subtotal			**3 061,2**

Camera Control

Component	Unit price	Quantity	Total Price
Raspberry pi	336,0	5	1 680,0
SD memory card	69,0	5	345,0
HDMI Cable	149,0	5	745,0
Current supply	149,0	1	149,0
Subtotal			**2 919,0**

Total	**15 759,4 SEK**

Appendix L

Project Management

Appendix M

Project Management Method

The project has been characterised by limited knowledge on the time needed for fulfilment of each requirement, a demand for reaching the course aims of MF2058 and absence of and unwillingness to create a strong hierarchy within the group. Therefore, the chosen method of conducting the project management during the autumn has been agile, with short sprints and meetings and vaguely defined project roles, with focus on long-term goals instead of time-based deadlines. Emphasis has been on creating a flexible and transparent working environment.

Goals

The goals have built on the requirements, with focus on the first phase of them. This has given us enough structure to work in an efficient way towards the project deadline in December, while still being flexible to make changes along the way. There has been a priority order to ensure having a working demonstrator at the end of the project. For example, robustness has been prioritised over the amount of requirements fulfilled.

Project roles

There have been no strict project roles except for the project manager, who has been more of a scrum master with responsibility for tasks and a holistic view on the project. Some team members have been more responsible for company contact, ordering of goods etc., but it has not been an issue for someone else to take over if needed.

Sprints

Since an important aim of the project is to bring the different disciplines of the major together, the work has been divided into sprints during which the project members have focused on one certain area of the project. The aim has been to switch area for every sprint, in order to make sure every project member acquire skills from all areas of the project. The division of areas has been done during Monday meetings, based on the interest of the members. No one has been forces to switch area, but encouraged to. During the first period of the fall semester the sprints were two weeks long, but during the second period they were shortened to one week. The reason for this was that the progress for every week was increasing and the handover would be too massive after two weeks. As before, members were not forced to switch areas, but it was important to create a mindset of 1 week sprints. By working with sprints the transparency and flexibility has been high, since everyone have had at least a sound understanding of all parts and have been able to help with solving problems. It has also increased the motivation of the team members, since everyone have, so to speak, shared the good and the bad. No one has been stuck with a task that has been unsatisfying and boring for too long.

Smaller teams

The sprints have been divided among smaller teams, and the team members have not been individually responsible for big tasks. In contrast to working independently, working in small teams increase the problem solving ability and generates fewer errors since more perspectives are brought into the problem solving process. Also, it is in many ways time efficient, since the risk of getting stuck decreases and you rarely procrastinate when you are expected to be a collaborative partner. Still, the teams cannot be too big. The team members have to feel needed and be able to contribute. Also, the bigger the team, the longer time for every decision to be taken.

Meetings

The meeting concept has been developed aiming at creating transparency and being efficient. Therefore, to give the whole team information on the current progress, the meetings have started with a tour round the table where everyone has summarised to the rest of the group what they have been doing since the last meeting. Big decisions have been discussed during meetings, while those not affecting everyone have been discussed during smaller meetings. More about the meeting principles can be read in the Code of Conduct (appendix N).

Communication

Communication has mainly been done through Slack, a mobile-app and desktop application with the feature of creating different channels for different topics. Everyone has had access to all channels, which means that all information has been available to everyone.

Appendix N

Code of Conduct

Code of Conduct

Shine – Pixel Grinder

This code of conduct contains social, ethical and moral guidelines that our organization should follow throughout its practice.

Ethical principles and values

All members of the group are treated with respect. All members of the group are expected to be honest, unbiased and unprejudiced. Everyone should always be able to speak his or her opinion.

Accountability

Take responsibility for your actions. Carry out and finish the tasks you have taken on. Do not be afraid to ask for help if needed. Be open the help other members within the group. Tell each other if you think of a better way of solving a task and be open for suggestions from other group members. Be clear and honest with how much workload you have at the moment. Accept group decisions. Keep company secrets. Always try to do your best.

Standard of conduct

Communication is important, communicate with the group what you are working with and what progress you are making. Use the channels decided upon to share information, files and prototypes. Commit to the organisation. Be eager to learn. Have fun!

Standard of practice

- Do not be late. If you will be late, tell the group that you will be late.
- The time, place and agenda of a meeting should be presented in advance.
- The meetings should be structured and follow an agenda.
- A meeting chairman and meeting secretary should be assigned every meeting.
- Go through the decisions made during the meeting at the end of the meeting.
- Everybody should be on board on all the decisions. The group members can agree to disagree but everyone should accept the decision of the group.
- Meeting minutes are uploaded directly after the meeting is held. If you

miss a meeting it is your responsibility to read what was discussed and decided during the meeting.

- Preferably longer meetings should be held within sub groups and are to be summed up during meetings with the whole group.
- Company contact is performed through decided channels – via skype.

Disciplinary actions

If you feel like someone is violating the code of conduct please speak to the HR-manager of the group. Problems with not following the code of conduct will be discussed within the group or if preferred alone with the HR-manager. If the problems remain the teaching team will be contacted.

If you are late for a meeting you have to bring “fika” for the next meeting.