KTH Shine

SARAH ABELIN
EDVARD COLLIN
PEDRO ESCRIBANO
FELIX FRIE
MIKKO KIESI
YUCHAO LI
FRIDA LINDBERG
JOAKIM OSCARSSON
SEBASTIAN QUIROGA
VICTOR SUNDIN

MF2058 Mechatronics Higher Course Project
Supervisor: Jad El Khoury
Examiner: Martin Grimheden

Abstract

This project is part of the Mechatronics, Advanced Course, at KTH (The Royal Institute of Technology). The 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 and an additional device for adjusting their cameras. The light rig should be a modular, robust, power saving and user-friendly system.

In order to perform the 3D scans, Pixel Grinder uses a technique called photogrammetry. The leading contributor to the development the equipment within this field is the Institute of Creative Technologies. Their solution among others has been studied in order to know what top products there are within the field today. To achieve ideal scanning results it was concluded that the cameras have to take five consecutive pictures with different lighting conditions: one picture with polarized ambient light, one with unpolarized ambient light and three picture with gradient light from left to right, top to bottom and front to back. For the mechanical structure a decision was made to use an icosphere with LEDs distributing light from every vertex and every edge of the rig. Simulations in Blender indicated that 156 LEDs are needed with an intensity of 4000 lumen per LED node. The LEDs are to be synchronised with the camera shutters to minimize the time they are lit. The cameras will be placed outside of the light rig and can be directed with gimbals.

The LED nodes need to have the possibility of adjusting the settings of each individual LED unit that is under its control. The camera nodes are responsible for configuring the cameras, saving images, triggering capture and communicating actions to the gimbal node. A root node is needed to maintain communication with the user and the two previously mentioned nodes. The LED node will need CAN interface, the Camera node will need Ethernet and the root node will need both interfaces. A critical function prototype was designed including five LEDs synchronised with two cameras. The project will continue until December 2015 resulting in an actual full size light rig with controllable LEDs and cameras.

Contents

Acronyms

AWG American Wire Gauge.

CAN Controller Area Network.

CRI Colour Rendering Index.

DSLR Digital Single-Lens Reflex.

GUI Graphical User Interface.

ICT Institute of Creative Technologies.

IES Illuminating Engineering Society.

KTH The Royal Institute of Technology.

LED Light Emitting Diode.

MCU Micro Controller Unit.

OLED Organic Light Emitting Diode.

POE Power over Ethernet.

PWM Pulse Width Modulation.

QLED Quantum dot Light Emitting Diode.

SDK Software Developer Kit.

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 planned project.

1.1 Background

The project background is here 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 mainly 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, one 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 this project. Pixel Grinder focuses on capturing images of actors and actresses, and by the use of photogrammetry, generates 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 both face scans, and full body scans, two processes that require different camera and light setups. For this project, two products are to be developed. The primary priority is a light rig for the facial scan. 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 which sizes and weight cannot exceed the maximum airline baggage regulations, since Pixel Grinder travels to clients abroad using commercial airlines. The solution should also be scalable, for future implementation for the full body scan.

The secondary product is an aim and focus controlling system for the cameras used in the full body scan. It is believed that with such system, the process from putting up the rig to taking the pictures, will speed up significantly. 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 study that has been made in the MF2058 Mechatronics Advanced Course Spring. The requirements of the project, see appendix A, have been developed in collaboration with the host company, Pixel Grinder, and they form the basis for this project and report. The documented studies made in order to plan the design are to be presented in the report.

1.3 Scope

Pixel Grinder has requested a controllable light rig and a device for adjusting their cameras. Within these broad areas, and their subareas, background and state of the art studies have been made in order to collect information necessary for the development. The project group vision is to design the devices according to the requirements and in as many areas as possible improve the design of relevant devices already on the market. Another aim of the project group is to design a highly flexible and modular system. The state of the art study and the draft plan of the implementation - what has been achieved so far and what the project group is aiming for - is the scope of this report.

1.4 Method

The group working with the project consists of ten technology students, all assigned different responsibilities. Details of these project roles, along with an overall project plan, can be seen in appendix B. The group has agreed on and work according to the developed code of conduct, see appendix C. Weekly meetings have been held with both the company representatives and supervisors at KTH. For the state of the art study, different published articles and company pages have been read in order to collect necessary information. To develop and design different ideas and implementation of a model, concurrent engineering has been utilised.

Chapter 2

Theory

This chapter presents the theory behind the planned project - a study of photogrammetry, how different light settings have effect on it and how to best communicate information on distributed systems. These parts has been studied in order to know what is needed, any why, to plan and produce a first-class product.

2.1 Reverse Engineering

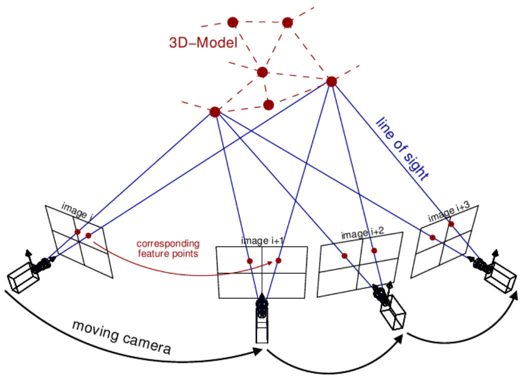
With increasing interest over the recent years, the art of retrieving a model of a 3D object - reverse engineering - continues to grow. For this project, that deals with the scanning of mostly humans, non-contact scanning methods have been looked upon. In order to collect 3D data with high demands on accuracy, there are a few important methods worth mentioning [1] and [2] :

- Stereo disparity: this method receives information about image depth in the same way that human eyes do, and can be done both manually or automatically. This methods are also known as photogrammetry.
- Structured light pattern: this method uses an artificial light pattern to illuminate the surface of the object. Images of the illuminated surfaces are then used to calculate the depth of the image, using knowledge about in what ways the pattern has changed.
- Direct ranging or profiling: this method measures depth by using a transmitter and a sensor to send, retrieve and calculate the time of flight of a laser beam from the device to the object.
- "Shape from" techniques: this method recovers the relative depth from surfaces for example shading, motion, contours or stereo.

This reports focus on the same technique as the host company uses: photogrammetry. This method of receiving 3D information is further discussed.

2.2 Photogrammetry

Photogrammetry is the "science of making measurements from photographs", which means recovering the position in 3D space from a surface's position in 2D. Photogrammetry is basically to snap and compare two pictures of the objects taken in different angles. By finding the similarities in the pictures and through means of triangulation (measurements of angles from two known points in space and through them calculating distance) and trilateration (measurement of distance with help of the geometry of triangles, circles and spheres), the coordinates of these spots are found, an scheme of how it can be achieved is found on the figure 2.1. The discovered coordinates create the final 3D representation.

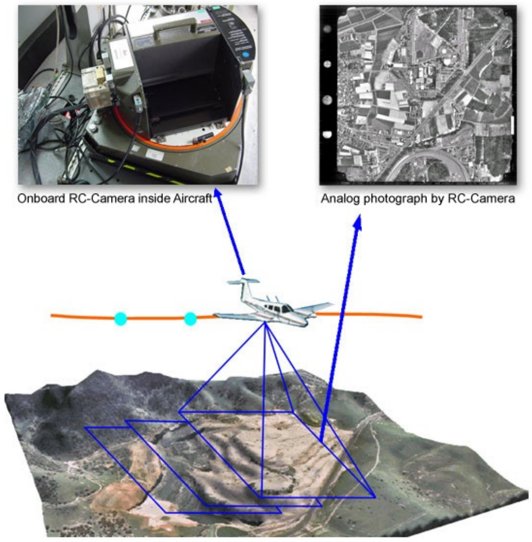


Source: `http://www.tnt.uni-hannover.de/project/motionestimation/`, Accessed: 2015-05-14

Figure 2.1: Triangulation approach to get the 3D model of the object.

Photogrammetry as a measurement science has existed from the 1860th [3] where it was thought of by Albrecht Meydenbauer, an architect for building and terrestrial measurement. Photogrammetry has mainly been used within topography where it was first used in practice from 1930s and forward. Later on other branches were developed.

In figure 2.2 below, terrestial measurement using photogrammetry is illustrated.



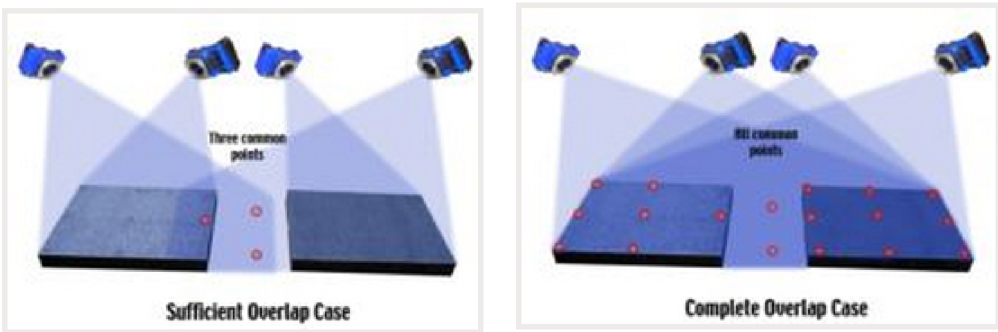
Source: `http://pasco.co.jp/eng/products/aerialphoto/`, Accessed: 2015-05-14

Figure 2.2: General overview of aerial photogrammetry.

With the new millennium, photogrammetry was further developed into a new branch, covering the movie and gaming industry. Photogrammetry as known today is a vast field of measurement science, which serves different purposes within various industries ranging from topographic imaging and GPS to laser scanning and industrial meteorology.

One of the key factors to improve the final model is knowledge about how the cameras are set up. It is important to gather as much information as possible from the model. The targeted object should always be in the middle of the picture and fill most of the photo frame. This means that a higher quantity of pixels from the image taken will be used in the software engine, which will result in a more detailed 3D model.

If there are multiple overlapping photos, every small feature of a surface is captured more than once in different photos as it is shown in figure 2.3 below. For



Source: http://www.geodetic.com/v-stars/what-is-photogrammetry.aspx, Accessed: 2015-05-14

Figure 2.3: Examples of overlapping.

high detailed models there can be multiple captured of the object, ending up with 50-70 photos for small object, 70-100 for medium sized object and 180-200 photos for buildings. If no overlapping exists, however, having a high quantity of photos will not have any effect. It is the overlapping of pictures that results in clear and detailed 3D model.

2.3 Light conditions

Without going too deep into the software process of the taken pictures, let us instead focus on the impact of light settings for the photo shooting. In order to properly compare pictures, blurriness and shadows must be eliminated. Blurriness confuses the software and areas with shadows will have less detail and will therefore not resolve as well in processing. Having low camera aperture reduces the image's depth - its blurriness. Bright, even lighting will allow to have a low aperture. Even lighting also removes shadows and reduce the effect of highlights on shiny objects. Detailed images without depth provide the ideal conditions for photogrammetry. Good lighting conditions also allows for lower camera ISO (sensitivity against light, resulting in less picture "noise") and shorter shutter speed (less motion blur).

To enable ideal use of the photogrammetry software, photographic data from several different light settings are often needed. This could include having the light emitted from different directions, as simulated ambient light, polarized or unpolarized etc., and are done to capture different characteristics of the light bouncing off from the target.

To get an ideal scan of an object, T. Akenine-Möller et al. states [4], it needs to be illuminated for the first four spherical harmonics; full ambient light, gradient light from top, right and front. The full ambient light is used to is used to know how much light that reaches each locations so that every location receives a coefficient for that, lower coefficients for crevices and higher for fully exposed surfaces. With the rest of the four spherical harmonics, lighting from top, right and front, the directions of the surfaces are found by the normals. By lighting the object from above coefficients can be given to each surface depending on how bright they are. The brighter the surface is the more it is pointing upwards and thus gets a higher coefficient. This is repeated for lighting from the front and then from the left. Each surface on the object now have four coefficients for each surface, which represents the surface's orientation and how much light it receives.

The identified variables relevant to creating good light conditions are, apart from the number of sources and the distribution of those, as follows:

- Intensity per light source
- The colour rendering index value of light sources
- Illuminating Engineering Society profile of light sources
- Polarity of the light

2.3.1 Intensity

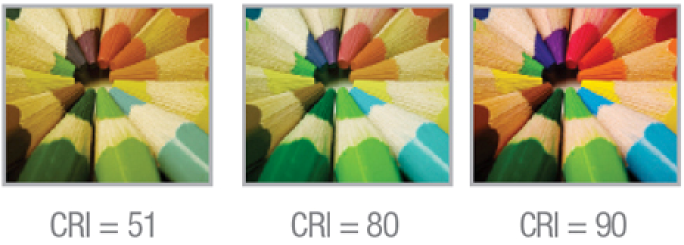
One way to measure the luminous intensity of light is in candela, *cd*. The intensity of a light source is its power emitted in a certain direction. The luminous intensity of a candle light is approximately one candela.

A light source can also be measured in lumen, luminous flux. Lumen is the total amount of visible light emitted from a light source. The relation to candela is that one lumen is one candela multiplied by the number of steradians, *sr*. This means that a sphere, which has a solid angle of $4 \cdot \pi$, that emits one candela in every direction has a luminous flux of $cd \cdot \pi \cdot sr = 12,57$ lumen.

Luminous emittance is measured in lux, which is defined by lumen per square meter. That means that lux is the luminous flux spread over a certain area.[5]

2.3.2 Colour rendering index

The CRI (Colour Rendering Index) also affects the lighting conditions. The CRI measures a light source's ability to reveal the colour of an object and compares it to an ideal light source (in range of 0-100). High CRI values are desirable in photography applications. The figure below shows an example of photos with different CRI-values.

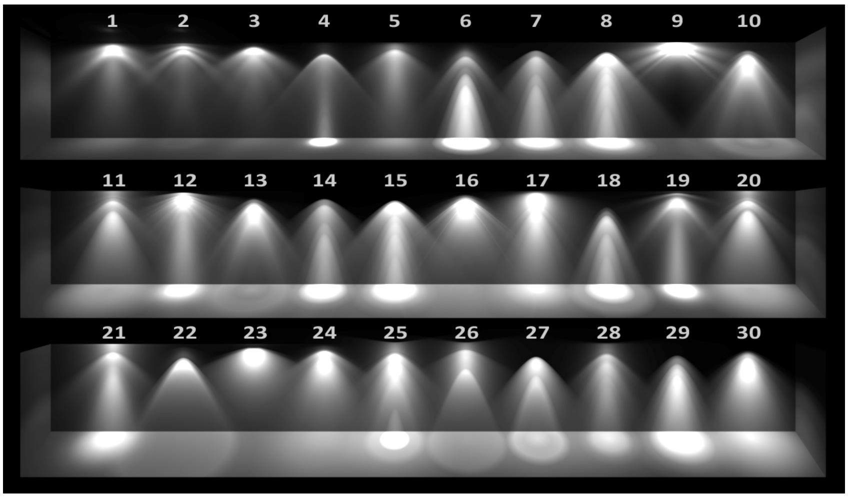


Source: `http://en.dearhe.com/about/?211.html`, Accessed: 2015-05-14

Figure 2.4: Comparison between different CRI Values in three pictures.

2.3.3 Illuminating Engineering Society

Light sources can emit light with different IES (Illuminating Engineering Society). An IES-profile is a digital profile of real world light [6]. It is a profile of how the light is distributed from a light source. The figure below shows numerous IES-profiles.



Source: `https://support.solidangle.com/display/AFMUG/Ai+Photometric+Light`, Accessed: 2015-05-14

Figure 2.5: Examples of different IES profiles.

2.3.4 Polarization

Light is a transverse wave, which means its electric field direction is perpendicular to its transmitting direction. Therefore, an appreciation of its vectorial nature is of great importance. Fig. 2.6 shows the relation of pointing vector, electric field and magnetic field of electromagnetic wave. $\overrightarrow{E}$ represents electric vector field , $\overrightarrow{B}$ for magnetic vector field and $\overrightarrow{V}$ for pointing vector. The pointing vector shows the transmitting direction of the light. The directions of these three vectors follow the right-handed rule [7]. Therefore, the following discussion only takes the electric vector, $\overrightarrow{E}$, into consideration since if both $\overrightarrow{V}$ and $\overrightarrow{E}$ are illustrated, $\overrightarrow{B}$ can be found by following right-hand rule.

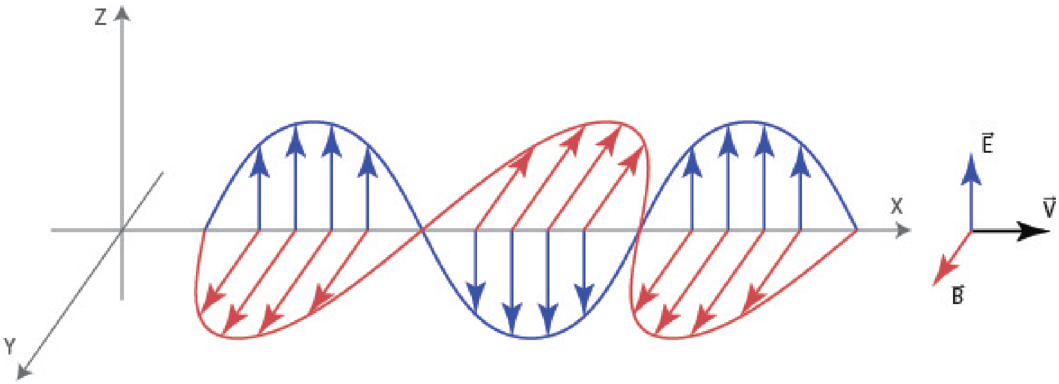


Figure 2.6: Relation of pointing vector, electric field and magnetic field of electromagnetic wave.

In order to clarify the direction of its electric field, a special plane called incident plane, or plane of incidence is introduced. Take the case in fig. 2.7 as an example. When light moves from a medium of a given refractive index n_1 into a second medium with refractive index n_2, both reflection and refraction of the light may occur.

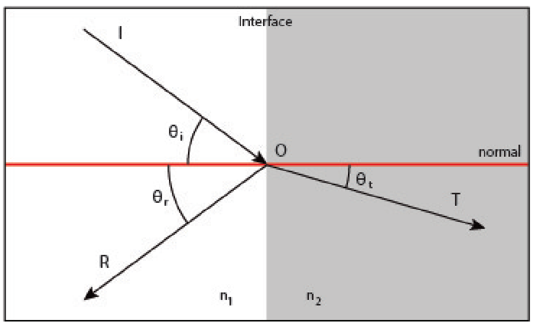
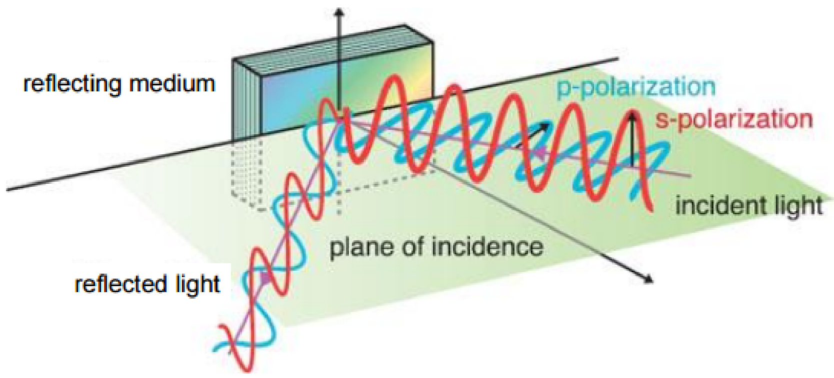


Figure 2.7: Incident plane. A beam of light moves from medium 1 of refractive index n_1 into medium 2 with refractive index n_2. Both reflection and refraction occur. The plane illustrated in the plot is incident plane.

The plane containing the incident, reflected, and refracted rays, i.e. in the plane of fig. 2.7, is called incident plane. Based on the incident plane, the electric field is often decomposed into two components. The one which is perpendicular to the incident plane is called s component, while the other one which is parallel to the incident plane is p component. It is clear that both the s and p components are perpendicular to their pointing vector, or transmitting direction. Fig. 2.8 gives a clear view of the relation of the s component, p component and incident plane. Refraction in this case is ignored [7].



Source: `http://www.ece.rice.edu/~daniel/262/pdf/lecture13.pdf`, Accessed: 2015-05-25

Figure 2.8: Relation of the s component, p component and the incident plane

The symbols of angles in figure 2.7 are used here, where θ_i is the angle of incidence, θ_r is the angle of reflection and θ_t is the angle of refraction. The relations between the angles are given by the following law of reflection

$$\theta_i = \theta_r \tag{2.1}$$

and Snell's law

$$n_1 \cdot \sin(\theta_i) = n_2 \cdot \sin(\theta_t) \tag{2.2}$$

The incident ray is not polarized, which means its electric field contains both the s and p component. The Fresnel equations are introduced below to describe intensity of reflected lights including both the s and p components. Their relative relations are illustrated based on duplicate swing. E_{1p} here stands for the duplicate swing of incident light with the p component and E_{1s} for incident light with the s component. E'_{1p} stands for the duplicated swing of reflected light with the p component and E'_{1s} for reflective light with the s component [8][9].

$$E'_{1p} = \frac{n_2 \cos(\theta_i) - n_1 \cos(\theta_t)}{n_2 \cos(\theta_i) + n_1 \cos(\theta_t)} \cdot E_{1p} = \frac{\tan(\theta_i - \theta_t)}{\tan(\theta_i + \theta_t)} \cdot E_{1p} \tag{2.3}$$

$$E'_{1s} = \frac{n_1 \cos(\theta_i) - n_2 \cos(\theta_t)}{n_1 \cos(\theta_i) + n_2 \cos(\theta_t)} \cdot E_{1s} = \frac{\sin(\theta_t - \theta_i)}{\sin(\theta_t + \theta_i)} \cdot E_{1s} \tag{2.4}$$

These two equations are significantly important because they indicate that the intensity of reflection with s component has nothing to do with the incident light with p component, and vice verse. The reflection of polarized light is still polarized with the same component of incidence.

The polarized light is, however, necessary to get rid of what is called as reflection. There are two kinds of reflections which are illustrated in figure 2.9 below.

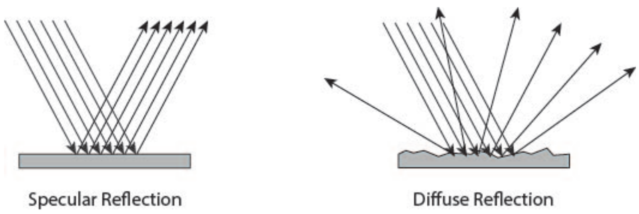


Figure 2.9: Specular Reflection and Diffuse Reflection

The first kind, specular reflection, destroys details of an object. Diffuse reflection, which in nature is the sum of several specular reflection due to the rough surface of object, is the one showing details of an objects [8]. As explained above, reflection of polarized light is still polarized with the same component while the emergent light after diffuse reflection becomes unpolarized (containing an equal mix of s- and p-polarization) because of the change of incident plane.

2.4 Handling communication

Handling and transferring data have a vital role in both academic and industrial fields. In order to make use of different programming code systems simultaneously and separate collaborative systems, a common intelligent language must be implemented. Information from one system, or from a specific written code, has to be able to be interpreted correctly by another system, possibly in another code language. A universal understanding - a communication interface - must be designed. If such a design also enables for a non-computer, a human, to be able to understand the communication, the collaboration between the designers are greatly improved.

Several tools, using different markup languages, are available today from the web. A pair of those is further discussed below.

2.4.1 Markup Languages

Extensible Markup Language (XML) and JavaScript Object Notation (JSON) are markup languages, or formats, famous in industries. They are especially used concerning program languages such as Java, C++, etc. where the codes are able to create objects from reading the transferred code [10]. Both these formats are also human readable and would without any major difficulties be able to be understood simply by reading the produced code.

In XML data representation format, sentences are bounded by system tags. Each chunk of words is initialized by **<tag name="[specific chunk tag name] [head word]">** and finished with **</tag name>**. It is, through this, difficult to misinterpret where a certain section of a text belongs. JSON represents data in text format where the different chunk of words are separated by structural tokens **([, ,], , :, ',', '"', etc.)**, literal tokens **('true', 'false', 'null', etc.)** and whitespace. Each data in a JSON generated code consists of objects, arrays of objects or other permitted typed within these [10].

It has been argued [11] that JSON, in comparison to XML and other more heavy weighted formats, has a higher rate of flexibility in combination with a lightweight structure to avoid unnecessary occupation of data space. It is, even though based on JavaScript, language independent and can be parsed easily and without special requirements. In this way, the author states, it provides an efficient method of transferring and reconfiguring data in distributed embedded systems, creating both flexible and adaptive systems.

Example implementations for the project can be seen in appendix G.

Chapter 3

State-of-the-Art Study

This chapter presents the State-of-the-Art within light rigs used for photogrammetry. The host company's version has been studied and their way of solving the light conditions, camera setup and communication are presented. Other companies currently working with light rigs, of the similar kind as the host company has requested, have also been looked into. Studies of some of the individual parts, LEDs, polarizing filters, devices for enabling camera movement and means for communication, are also presented.

3.1 Host company solution

Pixel Grinder currently employs two different camera and flash configurations, one high-detail scanner used primarily for facial scans and one full body scanner.

The facial scanner consists of 16-24 DSLR (Digital Single-Lens Reflex) cameras distributed in a spherical configuration with the subject in the center. The subject is lit during captures by 5 professional flash units.

Pixel Grinder currently use Nikon D5300 DSLR cameras in all their camera rigs. These specific cameras are used due to the image quality and relative low price. The cameras are configured with a shutter speed in the range 1/100 to 1/200 s and a aperture between f11 and f14. The shutter speed is in this range to ensure sharp images even if the subject is moving. A high aperture value is used to increase the depth of field.

The cameras are triggered with an external signal. This signal is propagated from a manually operated triggering remote switch via custom made synchronization modules. Each module can serve 12 cameras and apart from the synchronization signal they supply power to the connected cameras via an additional cable. All modules are connected in series and therefore the synchronization signal is relayed to all connected cameras with minimal time difference.

Polarization filters are mounted on each camera and they are aligned to be perpendicular to the polarization filters that are mounted on the flashes. These flashes are synchronized by being connected in series and are triggered simultaneously.

The cameras are mounted on a custom made fixture and the aiming of the cameras are done manually by the operator one camera at a time. The focus of the cameras are done by utilising the automatic function available on each camera. The setups are verified by examining shots from each camera on the computer and additional adjustments are done when necessary. When changing scanning subject the cameras' aim and focus need to be altered due to variations in subject height and size.

Each camera is connected to the computer via a USB-hub, this connection enables image transfer and remote configuration of camera settings from the computer. Due to the fact that USB communication is serial both these functions become increasingly time consuming when the number of cameras are high. Transferring all images from one capture currently takes approximately 10 seconds with a 20 camera setup, at fresh restart.

3.2 Photogrammetry 3D scanning solutions of today

Today, there are an increasing number of companies and hobbyists developing light rigs for camera shootings of different kinds. Charged flashes are commonly used, sometimes for example in combination with umbrellas, to create the ideal light for their specific photos. For such a light rig as the host company has requested - a controllable rig with the possibility to illuminate the object with several different light settings within a limited amount of time - there are only a few companies whose work could match the request. There are today no off-the-shelf alternatives for this specific kind of light rig, the ones available are transported and set up on request and are done so for a high price.

The leading contributor to the development of overall photogrammetry based 3D scanning equipment is the ICT (Institute of Creative Technologies). ICT has participated in the creation of films like Avatar and Spider-Man and has created several different solutions for virtual realizations of objects. The most recent creation contributing to accuracy and high quality of their scans is the Light Stage X - a geodesic dome with distributed polarised and unpolarised LED lights. Another company also working within the field of controllable light rigs is the British company Esper Design. The main difference is in modularity - the Esper Design is designed to be easily transported hence focuses less on quality than ICT. Esper Design has also put efforts into making the controllable light rig available on market. [12].

3.2.1 ICT Light Stage 1

The first solution developed by ICT consisted of a single 250 W light bulb, fixed on a rotating arm, with a single high-speed camera aimed at the object [13], as is illustrated infigure 3.1 below.

Source: http://www.pauldebevec.com/Research/LS/lightstage0011.jpg, Accessed: 2015-05-23

Figure 3.1: Light Stage 1

Although a minuscule amount of building material was used, effectively keeping the cost to a minimum, capturing an object took 60 seconds. This meant that i.e. face expressions, where your eyes need to remain open, were quite difficult to obtain.

3.2.2 ICT Light Stage 2

A further development of the first Light Stage lead to Light Stage 2 which is illustrated in figure 3.2 below.

(a) Light Stage 2 Stationary

(b) Light Stage 2 Rotating

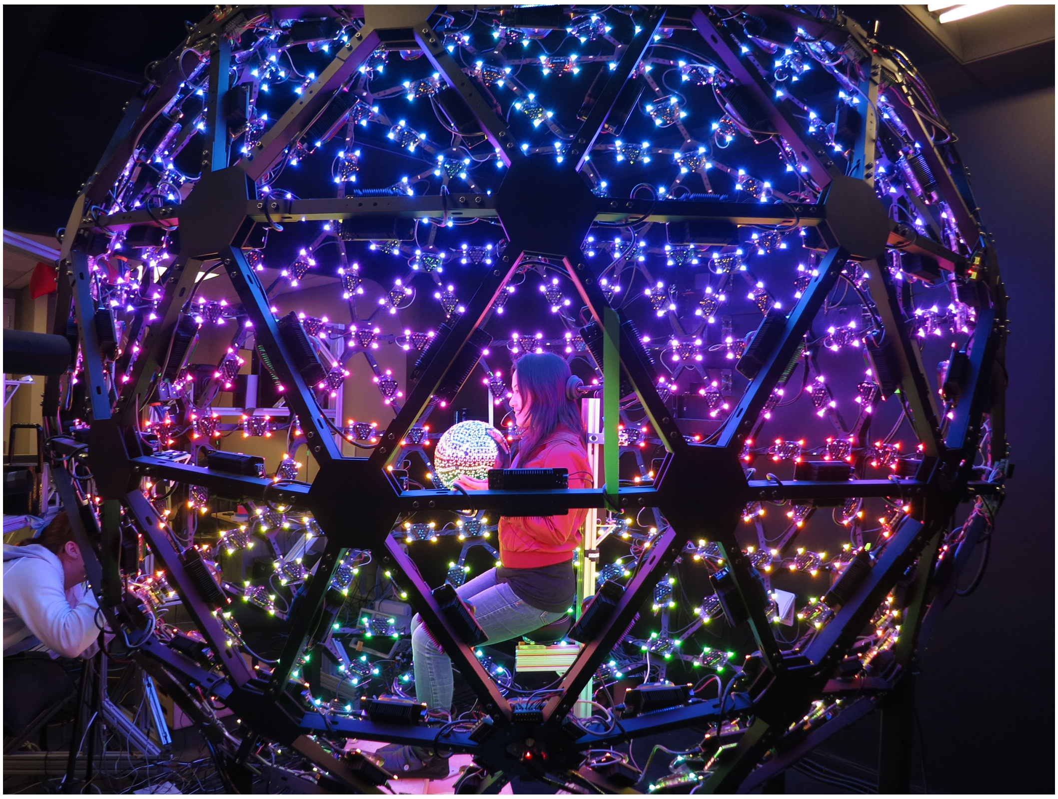
Source: http://gl.ict.usc.edu/Research/ls2/, Accessed: 2015-05-13

Figure 3.2: Light Stage 2

Light Stage 2 consisted of 32 light sources evenly distributed on a circular arm[13] and contributed to a decrease of scanning time to approximately 8 seconds.

3.2.3 ICT Light Stage X

In terms of geometric reconstruction of a face, a real breakthrough in the development of ICT's light stages was their "gradient illumination scanning process" - used in e.g. the movie Avatar. This involved building a dome with an even distribution of light sources, as well as distributing cameras evenly - as to capture a single face expression in one shot. The Light Stage X is illustrated in figure 3.3 below.



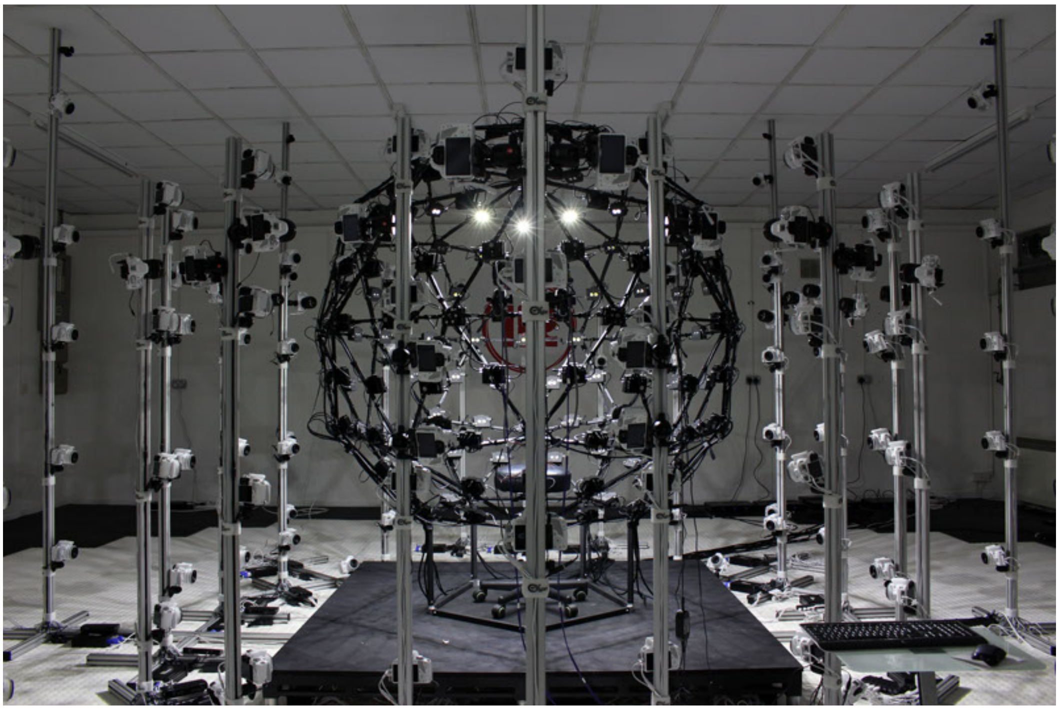
Source: `http://www.arl.army.mil/www/?article=2494`, Accessed: 2015-05-13

Figure 3.3: Light Stage X

Light Stage X uses 156 white LEDs as well as 4x156 RGB LEDs distributed evenly around the dome. The gradient illumination scanning process involves putting removable polarisers on all of the lights in the whole dome, meaning you can cross-polarize the whole sphere of illumination. In addition, if you put polarisers on the cameras, you can take a photograph of the specular mapping of a persons face which - in conjunction with gradient lighting - allows for an extremely detailed digitalization of a persons skin.

3.2.4 Esper Design

A similar - albeit more mobile - solution of the same principle as the Light Stage X is the British company Esper Design, illustrated in figure 3.4 below.



Source: `http://www.esperdesign.co.uk/images/gi-lite.jpg`, Accessed: 2015-05-13

Figure 3.4: Esper Design

It comprises of the same amount of lighting directions - 156 - however using an extra set of polarised LEDs i.e. 312 LEDs in total. It is also packable in 4 Pelicases and therefore easily transported [12]. According to Pixel Grinder, Esper Design can deliver and set up their light rig for approximately 1,000,000 SEK if requested.

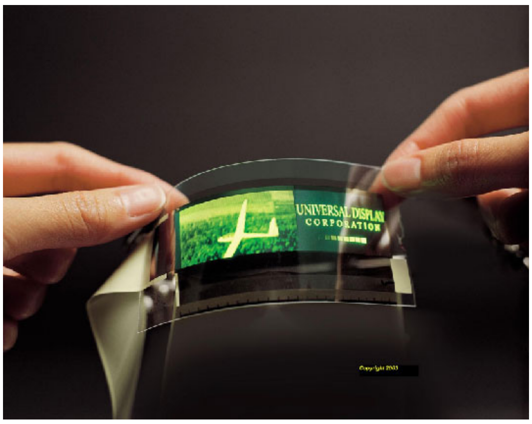
3.3 The State of the Art within specific areas

In addition to studying the overall solutions available today for light and camera rigs, one can also chose to study the individual parts of the rigs to see what the top products within those fields are.

3.3.1 Light Emitting Diodes

A main part in our project is to create a prototype LED (Light Emitting Diode) light rig for Pixel Grinders facial scanning rig. The rig is supposed to be used for creating perfect lightning conditions for the facial scanning sequence build on the first four spherical harmonics. LEDs are a light solution for the 20th century. Unlike other light solutions, like incandescent and luminous lights, LEDs are created to be energy and light efficient, of low-cost. They come in wider variety of sizes and are made to last longer, which makes them more suitable for our project.

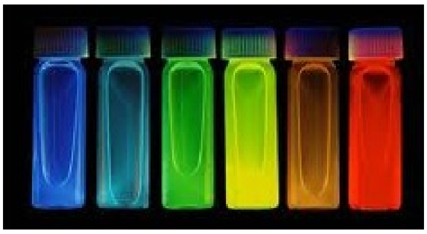
The latest within LED technology can be branched into two different areas: OLED (Organic Light Emitting Diode) and QLED (Quantum dot Light Emitting Diode) technologies. Both these are the future step of LEDs. OLED technology is a form of LED made out of an organic compound as semiconductor. The electroluminescent is thereby produced by the organic film situated between the anode and cathode of the circuit instead of the regular semiconductor. OLEDs are due to its organic composition less energy consuming than regular LEDs, at the same time they produce brighter light and give the possibility to create flexible materials capable of displaying light, as it can be seen in figure 3.5 below.



Source: `https://www.nsf.gov/od/lpa/news/03/images/foled_universal_display_rev.jpg`, Accessed: 2015-05-14

Figure 3.5: Flexible organic light emitting diod

QLEDs are the next step within LED technology. A QLED has a structure similar to an OLED but the main difference is in the composition of the light emitting material. This is made of Cadmium Selenide (CdSe) in a nano-crystal form. Which is smaller and at the same time has a smaller emission width. This makes QLEDs deliver a higher luminance efficiency advantage over OLEDs at a specific colour point. QLEDs have the capacity to become more than twice as power efficient and have a lower manufacturing cost than OLEDs.

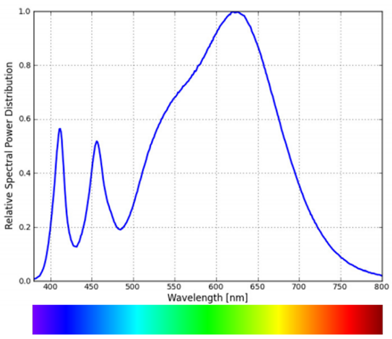


Source: `http://fr.cdn.v5.futura-sciences.com/sources/images/actu/rte/magic/30396_Quantum-Dots_QDVision.jpg`, Accessed: 2015-05-14

Figure 3.6: QLED solution

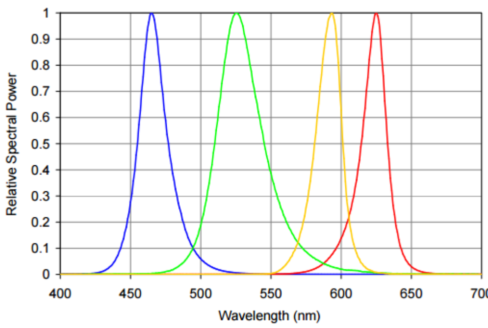
White vs RGB light emitting diodes

The main difference between white and RGB LEDs, besides the light colour control capabilities, is the spectral power distribution of the LEDs. This is related to the CRI value described in section 2.3.2. The figures below show the typical spectral power distribution of a regular white LED, see figure 3.7a, and of a RGB LED, see figure 3.7b.



Source: `http://www.mouser.com/ds/2/602/DS138-542373.pdf`, Accessed: 2015-05-23

(a) MC-DC2 Connector



Source: `http://www.mouser.com/ds/2/228/LZC-03MA07-253130.pdf`, Accessed: 2015-05-23

(b) MC-DC2 Tip

Figure 3.7: MC-DC2 Cable Connectors

Notice how the light of the regular white LED is much more evenly distributed over the visible spectrum, while the RGB LED has normal distributed light a specific wavelengths.

3.3.2 Polarizing filters

Polarizing filters play an important role in the 3D-scanning system as they eliminate the light that decreases the quality of the photos. Searching for new technologies with regard to polarizing filters could be useful for making better choice.

In academic community, many different methods used to create polarizing filters are promising. One way is to creatively manufacture or optimize some special material to get better filters. By using a two-step hybrid optimization technique, researchers from University of South Carolina are able to get polarization performance from grating-based metal-free polarizing filters [14]. Elaborate structures with the help of selective material could also contributes to good filters. Presented in [15], a tunable fiber polarizing filter is produced by selectively filling the holes of base material with high index liquid. The base material here is solid-core polarization maintaining photonic crystal fiber. With its specific characteristic, this method could be used for making flexible and controllable filters.

This technique is, however, far from a stage of mass production. Products off the shelf are based on mature technologies. Kood Circular Polarizer is a popular choice with low price and good performance while one of its drawback is that it reduced the shutter speed of camera. Hoya PRO1 digital filter has better performance and higher price. Other significant names include Cokin P164 Circular polarizer, B+W Digital MRC F-PRO and Tiffen Digital HT.

3.3.3 Enabling camera movement

In order to design a maneuverable system, a device for enabling remote control of the cameras could be useful. Such a system would consist of two parts: a mechanical structure and a motor control unit. Concerning the latter, there are mainly two alternatives on the market: stepper motors and servo motors.

Stepper motors normally have around 100 poles. Due to the number of poles and the fixed step size, a high precision can be achieved. The motor will move one step each pulse generated by a PWM (Pulse Width Modulation). The main problem of the stepper motor is that power must be applied constantly[16]. The coils need to be kept energized in order to maintain the magnetic field and therefore the torque in the motor shaft. This will generate a great amount of heat, which will cause losses. The cameras will be in a fixed position most of the time in use, for which reason stepper motors will be less beneficial to use.

Servo motors usually consist of a DC motor, a gear and a potentiometer which will act as a controller for the position of the motor. This type of motor is not as precise as the stepper motor since it does not have a fix step size and its performance will depend on the feedback sent by the potentiometer. Normally it has three wires: ground, power and control action. This control action is a PWM signal which depending on its duty cycle the motor will rotate to one position or another. In order to hold the position and torque the PWM signal will be sent approximately every 20 milliseconds[17]. In this case less heat will be produced and therefore less losses.

Nowadays remote control of the cameras is mainly used in photography and filming to give more freedom to the user. One example can be seen in figure 3.8. It can be controlled either by a remote or computer.

Source: `http://pacificmotion.net/remote-repeat-heads/`, Accessed: 2015-05-14

Figure 3.8: Remote control camera system

The remote control of cameras has also been used on devices like drones. The enabling camera movement system is, in this case, called gimbal, but the operating principle is basically the same. The main concern for this application is keep the camera stabilised when the drone is moving. In this case stepper motors are better since they provide smoother movements.

Chapter 4

Design Decision Basis

This chapter presents the plan of design for the mechanical structure, the hardware and software implementations as well as the economical aspect to the project. Why certain choices have been made is discussed, and the prototypes designed this far are presented.

4.1 Mechanical

In order to fulfill the criteria stated in the company's requirements, the light rig must be designed in such a way that the light distributes in an even, bright and soft enough way. It shall also be portable, meaning that it can be assembled and disassembled with ease. The light rig must also be transparent enough to allow cameras to be taking photos through it. The constraints of the light distribution are identified as following:

1. The lightsources need to be evenly distributed around the target, to provide soft light with canceling shadows.

2. The lightsources must achieve a balance between intensity and distance to target.

3. The lightsources will need access to both communication and powerlines.

4. The mechanical structure holding the lightsources in place, must allow for variation in camera positioning, while not obstructing the view of the cameras.

4.1.1 Structure geometry

To achieve even light distribution from all directions in a three dimensional space, a spherical geometry is feasible since all light sources can emit equally intense light when placed on a fixed radius from the target. Such a spherical geometric structure would have to be approximated, and a naive way to do so would be to use an elliptic geometry model, as they are commonly used to approximate globes to map the earth. That is, to create a network of longitudes and latitudes as seen in figure 4.1.

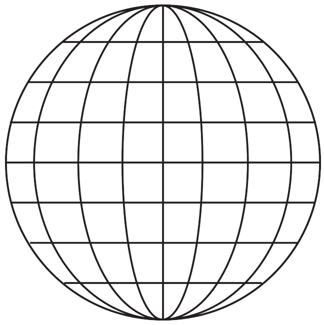


Figure 4.1: Elliptic geometry model.

The advantages of the longitude-latitude system, is its ease of addressing individual points, and potential light sources by mapping each longitude-latitude intersection to a Cartesian coordinate system and that the number of longitudes and latitudes could be increased to support more light sources. However, this approximation poses one problem. The contour lines of the sphere would not be of equal length, which implies that, even though the intersections are trivial to address by a Cartesian coordinate system, their distances along the contour lines would vary with the latitude coordinate. To position the light sources evenly on top of such a grid, becomes a non-trivial task.

To solve this, the sphere can be approximated using a model based on icosahedral symmetry. A regular icosahedron consists of 12 vertecies and 30 edges, as shown in figure 4.2.

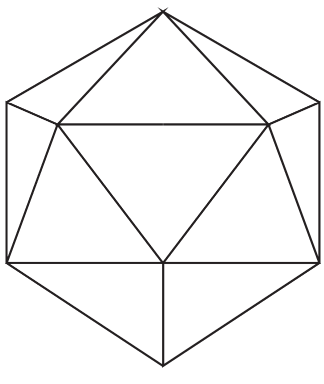


Figure 4.2: Icosahedron.

If a light source would be placed on every vertex and edge, the total number of lightsources would be 42. Preliminary estimations by Pixelgrinder sets the number of light sources to around a 100, which requires the icosahedron to be subdivided once into a Pentakis icosidodecahedron as seen in figure 4.3.

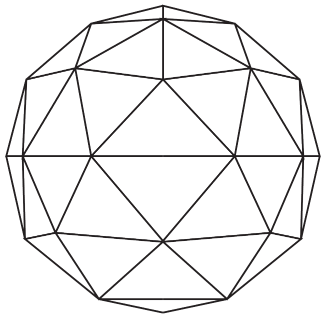


Figure 4.3: Pentakis icosidodecahedron.

A Pentakis icosidodecahedron consists of 42 vertices and 120 edges, both of two different configurations. The edges form 20 equilateral triangles and 60 isosceles triangles. While this might pose a difficulty in assembly, it ensures that all vertices are at the same distance from the target.

4.1.2 Size and Modularity

At the moment when Pixelgrinder are travelling with there equipment, which include the camera rig and flashes, they pack it into Peli cases which they check in as luggage on flights. They want to be able to do the same with the light rig which will put some constrains on the structure. The light rig needs to be disassemble into part which will fit into these cases and distributed so that it wont weigh more then what you are allowed to carry as check-in luggage.

Because they are travelling with their equipment it needs to be easy to disassemble and assemble the light rig. Often they also work under time pressure which even more stresses the need for it to be easy to be assemble. A structure based on a elliptic geometry model could be built out of rings that for the longitudes and latitudes which would make it easy to assemble but it would be hard to disassemble it small enough parts to travel with. of course the rings could be disassemble too but then the complexity would increase.

If a structure was used that was based on a icosahedron the requirements for the ability to travel with it would probably be met because they naturally are built of parts that can be disassembled. A icosahedron consists of different vertices and edges and for example a pentakis icosidodecahedron consists of four different parts, two different vertices and two different length of the edges. So even if it is not a simple structure it would be fairly easy to assemble it with some kind of instruction. Another advantage with a icosahedron is that it is scalable by just changing the length of the edges.

When it comes to the size of the light rig there are many factors that needs to be taken into account. First of all there is that the the object, in this case a human head, must fit inside the rig and have access to it. The light must be at a distance so that the light becomes diffuse but not so far away that the strength of the lights must be huge. Then the cameras must be able to take picture through the rig at the object and also be able to move so that specific parts of the object can be targeted. This has been discussed with the company and some tests have been made. The size that it is leaning towards is around 1.5 meters

A comparison between the the different structures can be seen in table 4.1

Table 4.1: Comparison between geometry structures

	Elliptic geometry model	**Icosahedon**
Modular	Yes	Yes
Complexity	Simple	Complex
Assemble	Easy	Harder
Even light distribution	No	Yes
Scalable	No	Yes

4.1.3 Light Simulations

To get an idea of how strong and how many lights are needed to fulfil the requirements from Pixelgrinder simulations has been made. The program that has been used for the light simulations is Blender. Blender is free 3D-modelling and animation software which can for this purpose be used to simulate light conditions. In blender different light sources can be placed where they are wanted and the characteristics of the light sources can be changed. To get a good light simulation a good rendering software is needed and the one used is called Indigo.

Indigo is an unbiased renderer which can be used as an add-on to blender so all the settings are done there. Unbiased means that the light paths will be created as physically accurate that is statistically possibly using the Monte Carlo algorithm. Indigo renders the image over and over again with the Monte Carlo algorithm and takes the average which means that if given enough time the result of the rendering will approach the most accurate result [18]. Because of this there obviously is a trade-off between accuracy of the result and the time being spent rendering and in this project accuracy is very important. There are biased renderers which work faster and uses other methods to generate the light paths that could give reasonable result but that would demand a lot of time being spent on tuning them which probably would not save time in the end anyway.

Another useful thing with the Indigo renderer is that it has settings for camera that catches the rendering which is very useful for this project. Simulations and renderings in blender and indigo can give an indication to this project on the number of light sources needed their strength and their placement.

The settings of the camera in the simulations have been set accordingly to the settings which the company uses during scanning and at presented in section 4.1.3.

Table 4.2: Camera settings while scanning.

Camera settings	
Focal Lenght	80 mm
Film ISO	100
Exposure	1/200 sec
Aperture	f14

The first simulations have been made light sources places in every vertex of a Pentakis icosidodecahedron with a diameter of 1.5 meters except in the bottom were they have been removed to allow access into the light rig. A pentakis icosidodecahedron has been used just to be able to evenly distribute the light sources. In total there are 41 light sources and the light intensity of of each light source is the same and different intensities has been simulated and the result can be seen in figure 4.4.

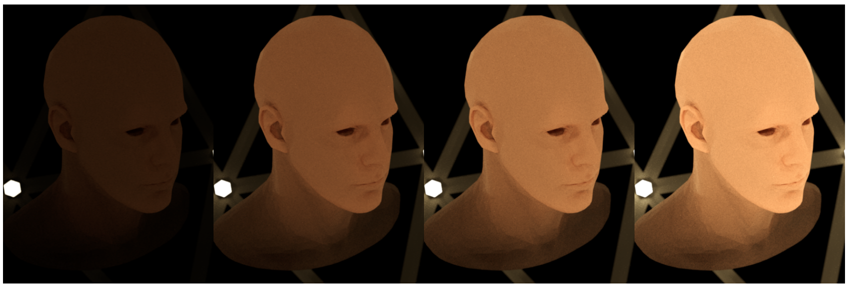


Figure 4.4: 42 light sources with light intensities of 1000 lm, 5000 lm, 10000 lm, 20000lm.

As seen in figure the light intensity that each light source needs to have to be able to generate a good image is about 20000 lumen which is quite high. In order to decrees the intensity of each light source he number of light sources needs to be increased. In the next simulations there are still a light source at each vertex but there has been light sources added on the middle of each edge.

In total there are now 156 light sources and the simulation with different light intensities can be seen in figure 4.5.

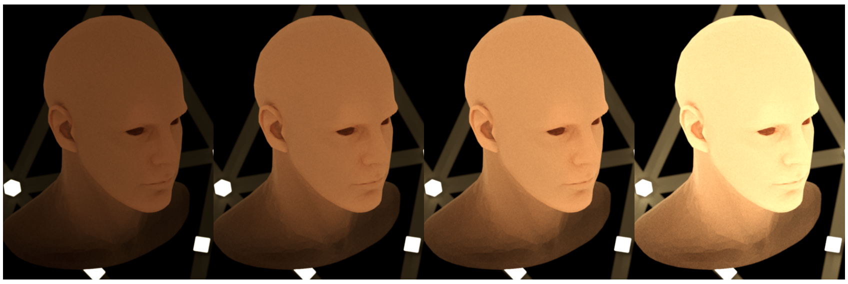


Figure 4.5: 156 light sources with light intensities of 1000 lm, 2000 lm, 4000 lm, 10000lm.

As seen in the figure the intensity of each light source now needs to be much lower, about 4000 lumen per light source instead of 20000 lumen, to generate a good image. Of course the number of light sources has increased but for the purpose of this project it is better to have them more distributed.

These simulations only give an approximation to what light conditions that are needed. There still requires work to verify these simulations.

4.2 Hardware

In order to design a modular, robust, power saving and user-friendly system, a well-designed hardware system must be implemented. It needs to deal with the synchronization between the cameras and the LEDs and communication between a host computer and several controlling nodes. The plan, and partly the implementation, is presented below.

4.2.1 Overview

In below, figure 4.6 illustrates a simplified hardware overview, visualising the idea of modularity and scalability for the hardware part of the project.

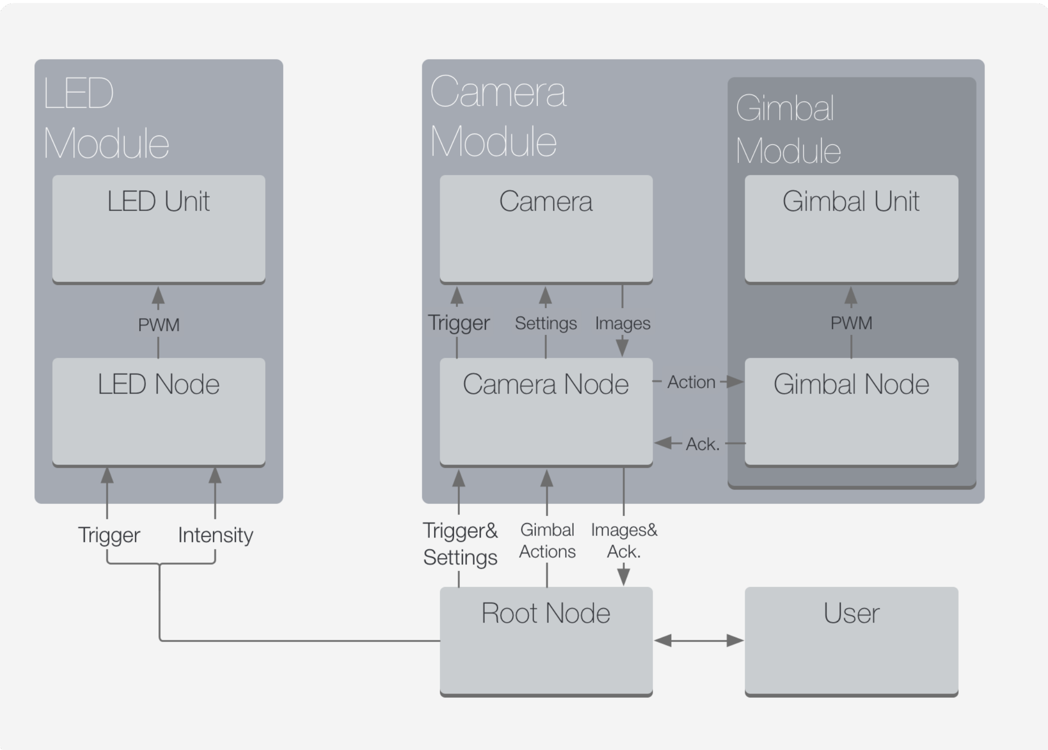


Figure 4.6: Hardware Overview

As can be seen in the figure, the LED module consists of a LED node, which receives Settings and Synchronisation signals from the Root node, and then sends a PWM to the LED unit controlling the intensity. Camera Settings and Synchronization with the LED module is a necessary connection between the Camera module and the Root node, and is achieved through a connection between the Camera node and the Root node. The user interacts only with the Root node.

As explained in the mechanical part, section 4.1, an even distribution of light is required for the complete system. This is achieved with an icosphere structure with LEDs attached to it. An example of such a setup is illustrated in figure 4.7 below.

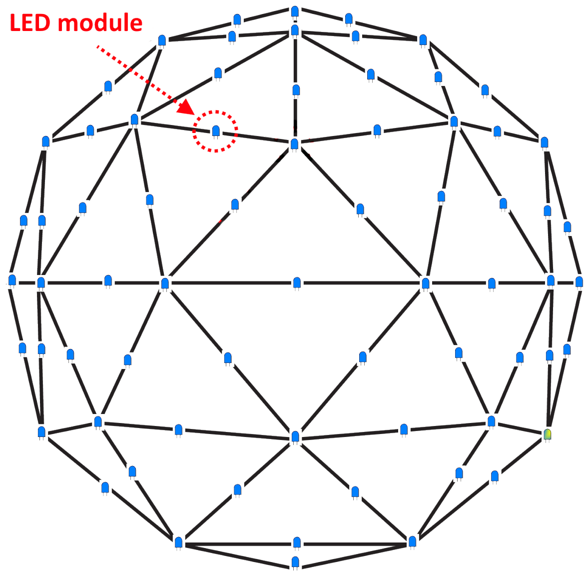


Figure 4.7: Example of LED unit placement on an icosphere, with one unit placed on each rod and mechanical node.

The amount of light directions identified in section 4.1.3 is 156, which means the amount of LED units will be fixed to 156. This corresponds to having light sources on the middle of each rod as well as on each mechanical node of the sphere, as is illustrated in figure 4.7. There is a relationship between the size of the light rig, the number of light sources, the capabilities of a LED node and the number of LED units in a module. Given a fixed number of light sources for the rig, one solution for the hardware rig would be to have a ratio of 1:1 in LED units to LED nodes within each LED module, i.e. each LED unit illustrated in figure 4.7 has their own MCU (Micro Controller Unit). This would put high demands on the communication but require less PWM pins on the actual LED node.

Another solution involves increasing the amount of LED units to 271, where two LED units are placed on each rod, and an MCU on each mechanical node, which is responsible for controlling the LEDs closest to it, as illustrated in figure 4.8.

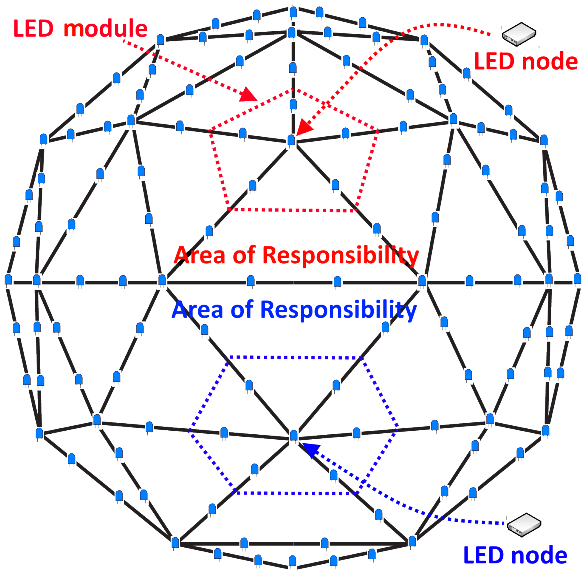


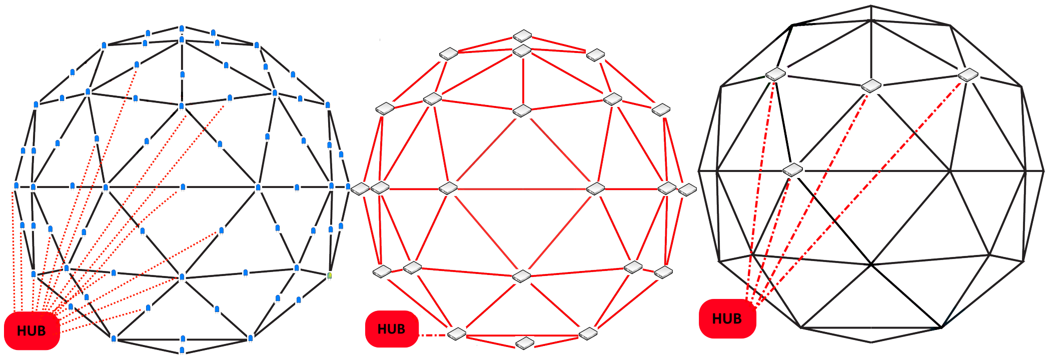
Figure 4.8: Alternate LED unit placement on an icosphere, with two units placed on the middle of each rod and one on each mechanical node.

This setup decreases the required MCUs, i.e LED nodes, by a factor of 6.3 and making the communication between modules less critical, while increasing the number of PWM outputs each LED node requires. Moreover, this setup with 271 LED units has not been simulated, but would require less than the 4000lm per LED unit as mentioned in section 4.1.

The modularity of the system is thereby constrained by the versatility of the modules. This versatility is also what makes the system complex.

4.2.2 Wiring Aspect

In order to reduce the amount of cables required in the system - hence increasing the portability - a comparison between different wiring alternatives must be made. Here, having wires in all rods is compared to having a new wire for each node going from a central hub. It is illustrated in figure 4.9 below.



(a) Wires to all LED units (b) Wires in all rods (c) Wires to all mechanical nodes

Figure 4.9: Comparison between different icosidodecahedron wiring concepts.

If a dedicated wire were to be drawn from the central hub to the placement of all 156 LED units (extreme case), as illustrated in figure 4.9a and an assumption is made of an average cable length of $\frac{1}{2}$ circle circumference, the total cable length would be $156 \cdot \frac{1}{2} \cdot 1.5 \cdot \pi \approx 368$ m. In comparison, according to section 4.1, there will be 120 rods of 2 different lengths, with an average length of 87.4 cm according to [19]. Therefore, in the most extreme case where all rods are connected to i.e. a CAN (Controller Area Network) bus, as is illustrated in red color of figure 4.9b, the total length of cable would be around 105 m. However, if a wire were to be drawn to a node using the setup illustrated in as is illustrated in figure 4.9c, the total length of cable drawn from the hub would be $42 \cdot \frac{1}{2} \cdot 1.5 \cdot \pi \approx 99$ m. The difference compared to the CAN solution is minimal, which is why CAN is the most sensible choice and allows for expandability both hardware and software wise, as well as being mounted directly to the rods.

CAN as interface for LED nodes

The total length of a CAN frame for an 8 byte message is 114 bits. According to [20], the maximum speed of CAN is 500 kbits/s, if the total bus length is below 100m. Therefore the number of frames that are able to be sent by CAN is 4385 Frames/s based on equation 4.1.

$$\frac{500 \text{ kbits/s}}{114 \text{ bits/Frame}} \approx 4385 \text{ Frames/s} \tag{4.1}$$

This is regarded to be enough for sending configuration settings (PWM intensity settings) to the LED modules, granted the PWM settings fit in a single frame as displayed in figure 4.10 below.

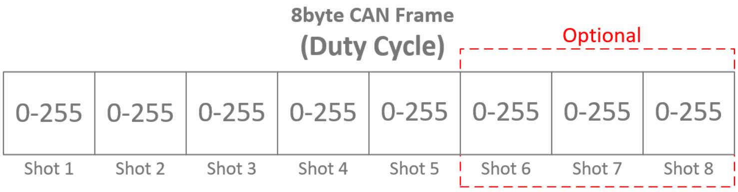
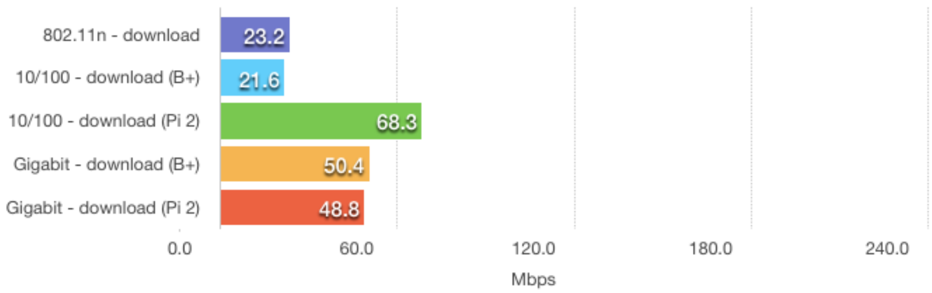


Figure 4.10: Settings Frame

As figure 4.25 illustrates, there are three bytes that could optionally include an additional set of shot settings.

Ethernet as Interface for Camera Nodes

One of the host company's wishes was for an increased image transfer speed compared to using an USB-hub where all cameras are connected. According to [21], the file size of the Nikon D5300 is varying between 1MB to 23.8MB depending on format. If CAN would be used, the total time of transferring a singe image from one camera will according to equation 4.1 in the worst case be 380 seconds. As a reference, the capabilities of 802.11 n WiFi and LAN (10/100) on the Raspberry Pi CAN be seen in figure 4.11 below.



Source: `http://www.midwesternmac.com/blogs/jeff-geerling/getting-gigabit-networking` , Accessed: 2015-05-14

Figure 4.11: Real bandwidth used by Raspberry Pi 2 and B+ for normal applications

Assuming the maximum speed over Ethernet is 68.3 Mbits/s, the total time for transferring a single image from one camera will in the worst case be 2.78 seconds, compared to 380 seconds when using CAN, and the 10 seconds mentioned in section 3.1. Therefore, it is concluded Ethernet should be used for communication to the camera node.

4.2.3 LED Module

The different LED units are controlled through a PWM via a micro controller. One LED node and the complete module is generally illustrated in figure 4.12 below.

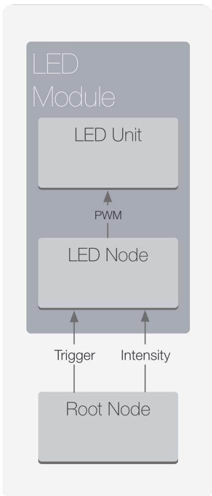


Figure 4.12: LED Module Overview

The even distribution of light is given by LED units, LED circuits with polarized and un-polarized LEDs, as illustrated in figure 4.13.

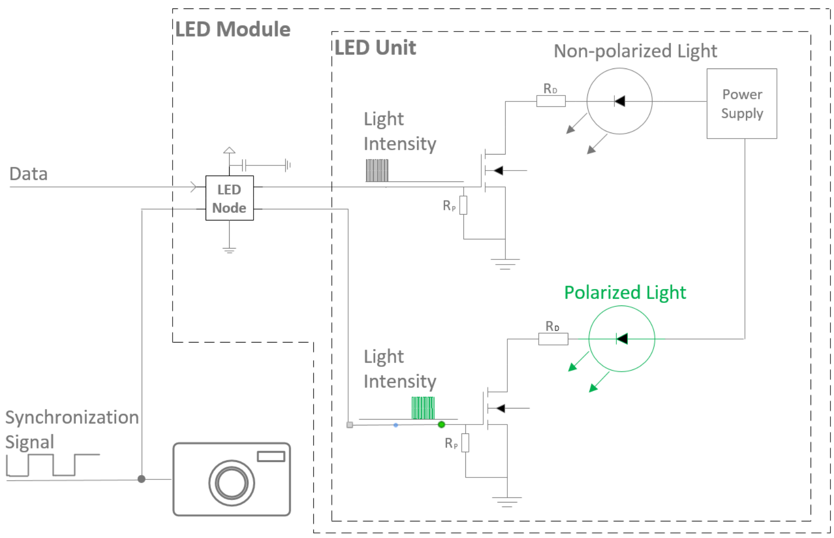


Figure 4.13: LED and Camera Synchronization

In section 4.2.3, it was not concluded wheter or not each LED module should have its own MCU as is illustrated in figure 4.13. The main principle is though that each LED in the LED unit is controlled by a PWM from some MCU. The requirements of the LED node are therefore, in order to be able to fulfil the requirements stated in , shown in table 4.3.

Table 4.3: LED node MCU Requirements

Requirement	**Reason**	**Min**	**Max**
CAN communication	To reduce amount of cables	1	1
PWM signals	For individual LED	2	14
I/O Input	For analog synchronization	1	1

A LED node uses the settings it receives through the CAN bus to control the intensity of each LED unit with PWM signals. An analogue interrupt signal (to which all LED nodes and Cameras are connected to) - or synchronisation frame sent over CAN - is used to trigger a change of the different lightning settings. More about this in section 4.2.5.

LED power consumption

In order to take a picture with the camera setting defined in the system requirements - see appendix A, a significant amount light is required. According to the preliminary simulations made in section 4.1.3 the LED-dome will require roughly 156 LED-units each of which have to deliver 4000 lumen.

The power required for such lighting intensity is significant and therefore an estimation on the power consumption was made. For this estimation the properties of 19 different high-power LEDs were compared. The LEDs chosen for the comparison provided neutral white light (4000 K), had a CRI-value of 90 or higher, of reasons mentioned in section 3.3.1 and nominal luminous intensity between 700 and 4200 lumen. The full comparison is displayed in appendix E. The comparison shows that the nominal lumen/watt ratio varied in a range $74.2-119.7$. This implies that when fully illuminated the power consumption of the dome varies between $33.4 - 53.5$ W per LED-unit and $5.21 - 8.41$ kW for the whole system.

The difference between LEDs is significant, with over 60% increase in power from lowest to highest. However, regardless of LED there is a high demand on power. While shooting, the LEDs are off for most of the time and therefore it is not reasonable to supply the system with a power matching only the few moments when the LEDs are turned on. The required power can be lowered by storing the energy between the shots and discharging when the camera shutter is open. In the next section, supplying the system with lower power using capacitors to store and provide energy for the LEDs, is discussed.

Using capacitors as energy storage

In this section the feasibility of driving the LED dome with capacitors is evaluated. Generic equations are derived and simulation for one type of LED is performed.

The evaluation is performed for a single LED-unit and the circuit diagram used is displayed in figure 4.14. The unit is attached to the main power supply, which provides the supply voltage, U_s. The capacitor is charged through two resistors and a switch S_1. The unit has a capacitor with a total capacitance of C. The LEDs are connected in parallel with the capacitor through the switch S_2 and can be armed with a resistor if the supply voltage does not match the voltage required to generate desired light intensity. In this evaluation only one set of LEDs is used, ignoring the other set needed for polarised light.

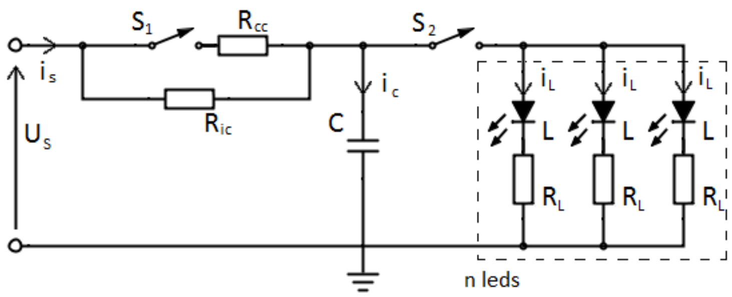


Figure 4.14: The circuit used for simulating the usage of capacitors to drive LEDs.

The circuit has three possible modes: initially charging the capacitor, continuous charging and discharging. These are implemented using the two switches that are displayed in table 4.4. Initial charge is used when the capacitor voltage is far below the supply voltage, continuous charging with voltages near the supply voltages and lights on when taking a single shot.

Table 4.4: Capacitor circuit modes of operation

State	S_1	S_2
Initial charge	open	open
Continuous charge	closed	open
Lights on	closed	closed

For the purpose of this evaluation, linearised LED model is sufficient. The linearisation is expressed in equation (4.2). The parameters are determined according to the forward voltage - forward current relationship diagram found in the data sheet of the specific LED.

$$u_L = a_r \cdot i_L + b_r \tag{4.2}$$

With the linearised LED model differential equations for the modes can be derived. The equations for initial and continuous charging & lights on modes are shown respectively in equations (4.3) – (4.5),

$$\dot{u_c} = \frac{U_s - u_c}{R_{ic} \cdot C} \tag{4.3}$$

$$\dot{u_c} = \frac{U_s - u_c}{R_{tc} \cdot C} \tag{4.4}$$

$$\dot{u_c} = \frac{1}{C}\left(-\frac{n_l \cdot R_{tc} + a_r + R_L}{R_{tc}\,(a_r + R_L)} \cdot u_c + \frac{U_s}{R_{tc}} + \frac{n_l \cdot b_r}{a_r + R_L}\right) \tag{4.5}$$

where n_l is the number of LEDS and R_{tc} the equivalent resistance of the parallel connected R_{ic} and R_{cc}.

The defining constraints for the circuit are the supply voltage and power, the time LEDs are on for a single shot and the required light intensity that affects the choice and number of LEDs. A longer illumination time increases the need for stored energy and therefore the capacitance.

Having established the required equations a simple Simulink simulation was designed. The following conditions were assumed:

- There are 156 LED-units generating 4000 lumen each.
- Supply voltage U_s = 40 V and maximum supply power 1 kW.
- LEDs: three Osram GW KAGHB1.CM-RSRU-40H3 per node.
- LEDs on for 10 ms per shot.
- Sequence: All polarised or nonpolarised LEDs on for every shot.

The resistors R_{ic} and R_{cc} were chosen so that maximum supply power never was exceeded. R_L was chosen so that the three LEDs generate 4000 lumen driven with supply voltage. Capacitance was set to 5 mF.

The total power, capacitor voltage and generated luminous intensity were extracted from the simulation results and are shown in figures 4.15 and 4.16 below.

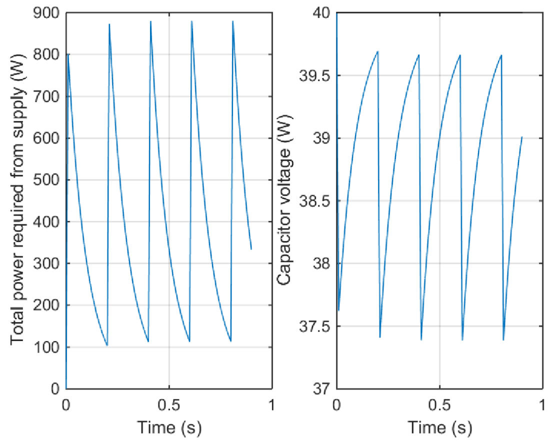


Figure 4.15: The total power and capacitor voltage during a single shot

The supply power stays under the 1 kW throughout the sequence, illustrated in figure 4.15 as desired. However, the time between shots is not long enough to fully charge the capacitor between the shots leading to lower lighting intensity on the latter shots. The luminous intensity illustrated in figure 4.16 falls significantly during a single shot, ending up at 82 % of the desired intensity at the end of the fifth shot.

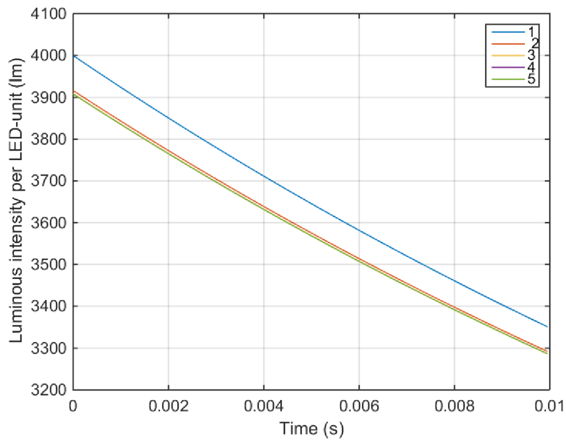


Figure 4.16: The luminous intensity for the five individual shots.

To prevent the intensity from falling too low during a single shot the time LEDs are on can be reduced by well set up synchronisation of the lights and cameras. For a single shot it is also important that the average lighting level is high enough so the initial intensity value can be increased by the LED resistors. The final option is to enlarge the capacitance. Increasing capacitor size, however, increases the drop of intensity between the shots since the supply will not be able to charge it to as high a voltage. Issues related to insufficient power supply could be increased by increasing supply power or using more efficient LEDs.

In conclusion supplying the dome using capacitors seems feasible but requires plenty of optimisation between the different components. Good synchronisation of camera and lights is paramount for proper and efficient power supply of the dome. The model developed can be used for the evaluation of wide range of options. Before basing major decisions on the equations derived here the model has to be verified with a prototype.

LED cost estimation

To evaluate the commercial feasibility a cost estimation for the previously compared LEDs was made and can be seen in table 4.5. The number of LEDs for each node has been approximated and is displayed in table E.1 and multiplied by two as both polarised and non-polarised ones are needed. The total cost for all the LEDs was based on prices at Mouser.

The price range varies from 31,400 SEK all the way up to 104,000 SEK. Two types of LEDs cost below 35,000 SEK which is more than 10,000 SEK less than the proceeding one. These LEDs also ranged in the higher range on energy efficiency. The LEDs in question were Luminus Devices CHM-14-40-90-36-AC00-F2-3 and Lumileds L2C3-4090109E06000.

Table 4.5: Comparison on the prices the 19 different high-power LEDs

Manufacturer article number	**Total Power (kW)**	**LEDs/ node**	**LEDs total**	**Cost (SEK)**
LHC1-4090-1202	5.21	10	1560	83 848
LHC1-4090-1204	5.54	2	312	51 804
LHC1-4090-1208	5.54	4	624	55 642
LHC1-4090-1203	5.54	6	936	58 701
LHC1-4090-1205	5.54	4	624	65 224
L2C3-4090109E06000	5.99	8	1248	34 401
CHM-14-40-90-36-AC00-F2-3	6.18	4	624	31 400
L2C1-4090120206A00	6.33	12	1872	77 403
GW KAHNB1.CM-TUUQ-40S3-T02	6.88	2	312	64 502
GW KAGHB1.CM-RSRU-40H3	7.13	6	936	48 312
GW6DGA40NFC	7.27	6	936	67 619
GW6DGC40NFC	7.62	4	624	71 746
GW6DGD40NFC	7.72	2	312	46 349
GW MAFJB1.CM RUSS-40S3	7.72	4	624	73 478
GW6DGE40NFC	7.86	2	312	60 981
GW MAEGB1.CM QPQS-40S3-0-T02	8.00	12	1872	104 242
GW6BGR40HED	8.08	8	1248	58 066
GW6BGS40HED	8.32	8	1248	68 456
GW MAGMB1.CM-TQTT-40S3-1050-T02	8.41	8	312	52 756

4.2.4 Camera Module

The Camera module is illustrated in figure 4.17 below.

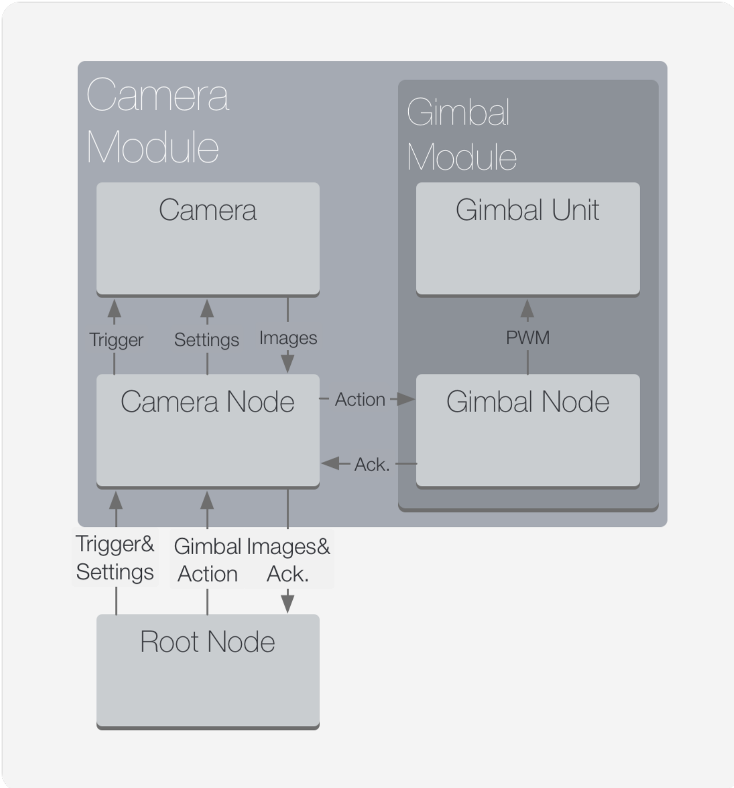


Figure 4.17: Camera Module Overview

The Camera Node is connected to the Root Node and handles all information sent to and from the Camera module. For the user to be able to select what cameras that are aim/zoomable an additional Gimbal Module is connected to the Camera Node.

The table table 4.6 below describes the requirements of the Camera Node.

Table 4.6: Camera Node MCU Requirements

Requirement	**Reason**	**Min**	**Max**
Ethernet Connection to Root Node	Settings, Images and LiveView	1	1
Data Connection to Gimball Node	Pitch, Yaw and Zoom	1	1
USB Connection to Camera	Settings and Camera Triggering	1	1
PWM signals	For Alternate Triggering of the Camera	1	1

A commercially available and renowned micro controller is the Raspberry Pi 2, which was used during the project for testing the requirements mentioned in table 4.6. Due to the lack of hardware PWM pins on the Raspberry Pi 2, a Gimbal unit would have to be controlled using regular I/O pins (software PWMs). They are more software intensive and will need more current. Therefore, another module needs to be added for the ability of controlling a three degrees of freedom Gimbal unit. Its requirements are shown in table 4.7 below.

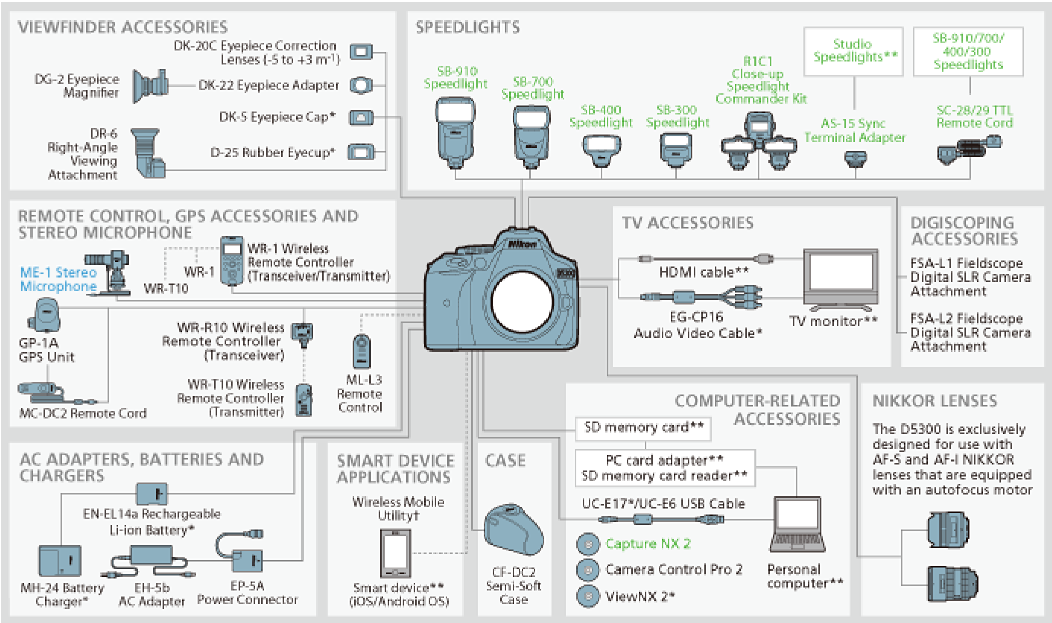
Table 4.7: Gimball Node Micro Controller Requirements

Requirement	**Reason**	**Min**	**Max**
Data Connection to Camera node	Pitch, Yaw and Zoom	1	1
PWM signals	For three motors	3	3

Since the Raspberry Pi 2 has four USB ports, data connection to the Camera Node mentioned in table 4.7 can be implemented using USB.

Camera Interface

The camera used for system implementation is a Nikon D5300. The system chart of a Nikon D5300 is illustrated in figure 4.18 below.



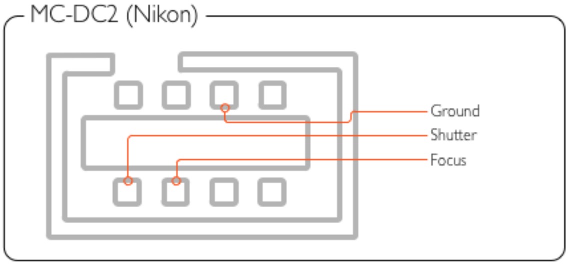
Source: `http://imaging.nikon.com/lineup/dslr/d5300/img/compatibility01/spec01.png`, Accessed: 2015-05-14

Figure 4.18: Nikon System Chart

According to the system chart, a computer-related accessory that could be used for serial communication with the camera is the UC-E17*/UC-E6 USB Cable. The cable enables PC to Camera control through a software interface e.g. the Nikon SDK. An analogue way to capture pictures and trigger the cameras auto-zoom functionality is by using the MC-DC2 Remote Cord.

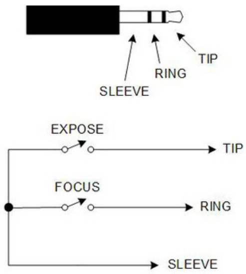
MC-DC2 Remote Cord

One alternative to using serial communication with the cameras is analogously triggering them to capture images. The figure 4.19 illustrates the connectors of the MC-DC2 cable.



Source: `http://www.triggertrap.com/news/making-your-own-triggertrap-connection-leads/`, Accessed: 2015-05-14

(a) MC-DC2 Connector



Source: `http://cdn.instructables.com/F8Z/YJAB/G2HE089F/F8ZYJABG2HE089F.LARGE.jpg`, Accessed: 2015-05-14

(b) MC-DC2 Tip

Figure 4.19: MC-DC2 Cable Connectors

The maximum capacity of the Nikon D5300 is 5 frames per second. If all channels of the MC-DC2 cable are shorted to ground, the camera connected to the cable will capture a frame, as shown in figure 4.20 below.

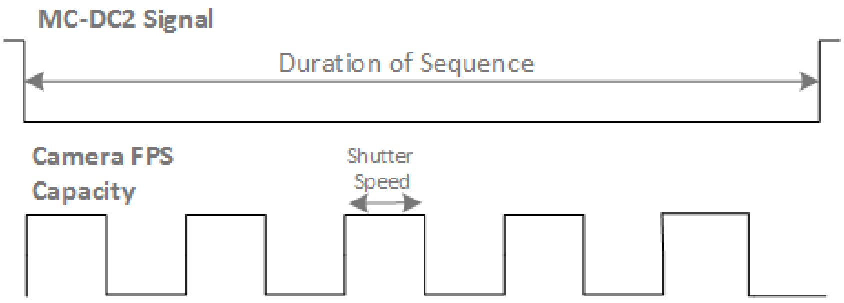


Figure 4.20: Illustration of the camera shot interval when MC-DC2 signal is shorted

4.2.5 Synchronisation of Camera and LEDs

Depending on the placement of the LED node, and in which sequence and type of light that should be emitted, different settings for activating the non-polarised and polarised LEDs will be required. In figure figure 4.21 below, a falling edge indicates a start of a image capture sequence, whilst a rising edge indicates the LED node should use a new setting by the time the next falling edge arrives.

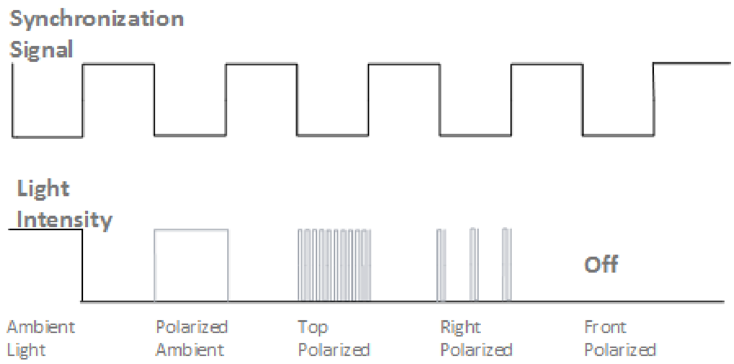


Figure 4.21: Synchronised signals

Instead of shorting the triggering signal of the camera for the duration of the whole sequence, as was illustrated in figure 4.20, the same signal as illustrated in figure 4.21 could be used for the synchronization of the camera and the LEDs, as was illustrated in figure 4.13.

The time for the transistor and LED to be turned on completely is typically within microseconds, whilst the camera shutter is slower due to the fact it is mechanical. The LEDs can be controlled to be lit the exact time the camera shutter is open, which is given by $t_{shutter} = {}^{1}\!/_{200} = 5$ ms. For ideal synchronisation of the camera shutter delay, the time it takes for the camera shutter to open can be determined by trying to capture an image of a brief LED flash, which is delayed in relation to the start of a falling edge - also used for camera triggering. The delay of the Nikon D5300 was found to be 80ms.

4.2.6 Power Distribution

Wire Gauge

The amount of current a wire can carry depends on several different factors e.g. wire length and ambient temperature. Generally a thicker wire can conduct more current. In the AWG (American Wire Gauge), diameters can be calculated by applying the formula,

$$D(AWG) = 0.005 \cdot \frac{92}{25.4}^{\frac{36-AWG}{39}} \text{ mm} \tag{4.6}$$

For every 6 gauge decrease, the wire diameter is doubled. For every 3 gauge decrease the cross sectional area is doubled [22]. The table table 4.8 below illustrates some AWG gauges.

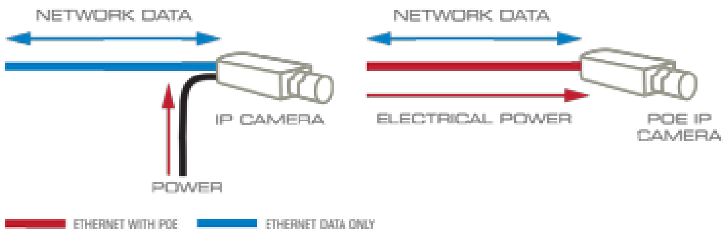
Table 4.8: American Wire Gauge

AWG Gauge	**Conductor Diameter (mm)**	**Ohms per km**	**Maximum Amps**
14	1.62814	8.282	5.9
15	1.45034	10.44352	4.7
16	1.29032	13.17248	3.7
17	1.15062	16.60992	2.9
18	1.02362	20.9428	2.3
19	0.91186	26.40728	1.8
20	0.8128	33.292	1.5
21	0.7239	41.984	1.2
22	0.64516	52.9392	0.92
23	0.57404	66.7808	0.729
24	0.51054	84.1976	0.577
25	0.45466	106.1736	0.457

In the metric gauge scale, the gauge is ten times the diameter in millimeter. However, as the AWG gauge goes down, the diameter increases. Due to this confusion, a wire is measured in the metric scale is typically measured in millimetres rather than metric gauges.

Power over Ethernet

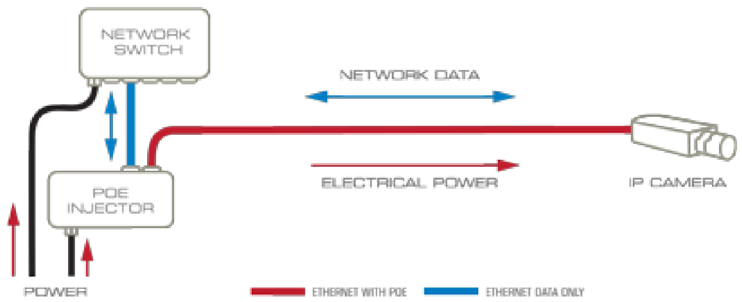
In order to decrease the amount of wires going to and from components, e.g. data and power cables as illustrated in figure 4.22 below, there is a commercially available standard called POE (Power over Ethernet) that can be used to solve this issue.



Source: `http://www.veracityglobal.com/media/78155/poe-explained-one-diagram-one.png`, Accessed: 2015-05-17

Figure 4.22: External power source compare to Power over Ethernet.

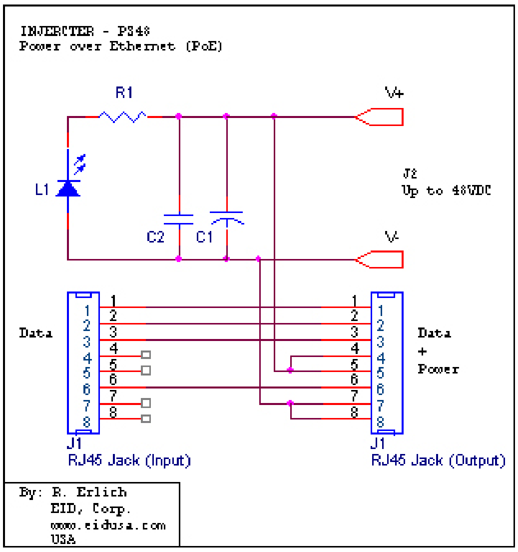
The benefits of POE are, apart from other things, flexibility and time saves time during installation. In the most recent standard, IEEE 802.3af, a single port on a POE enabled switch can supply as much as 25.388W, using 44V supply voltage and a 24AWG wirea table 4.8. The specific CAT6 cable at [23] is specified to 23AWG, which - according to table 4.8 - is able to conduct 0.729 A, yielding a power capacity in the cable of 32 W. If a separate POE injector is used, as illustrated in figure 4.23 below, and a higher voltage than the 44 V in IEEE 802.3af can be applied and an even higher power throughput can be achieved.



Source:
`http://www.veracityglobal.com/media/78155/poe-explained-one-diagram-three.png`, Accessed: 2015-05-17

Figure 4.23: Overview of Power over Ethernet with injector.

According to section 4.2.3, a single LED node requires between 33.4-53.5 W, implying that at least 45.8-73.4 V would have to be applied in order for a single pair to be able to provide the power needed using the 23AWG cable found at [23]. In figure 4.24 below, a POE injector schematic is illustrated.



Source:
http://www.veracityglobal.com/media/78155/poe-explained-one-diagram-three.png,
Accessed: 2015-05-17

Figure 4.24: Power Over Ethernet Injector Schematic

As figure 4.24 illustrates, the pairs 4& 5, 7& 8 can be used for power throughput - yielding a total power capacity of each cable at 2x35W i.e. 70W when using 48V.

4.2.7 Hardware summary

An overview of an imagined hardware solution is displayed in figure 4.25 below. A more detailed interface overview can be found in appendix D.

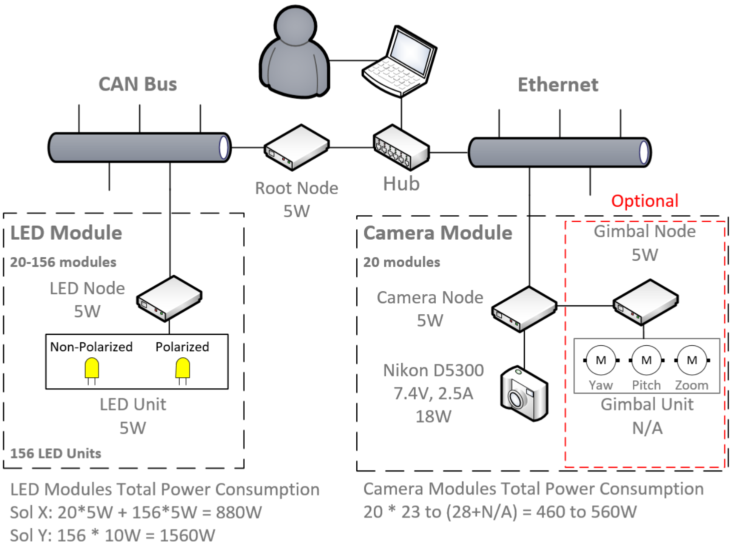


Figure 4.25: Connections Overview

As was concluded in section 4.2.3 and illustrated in figure 4.25, the LED nodes will need a CAN interface, the Camera nodes will need Ethernet and the Root node will need both interfaces.

Included in figure 4.25 is also a rough power consumption estimation. The power consumption per MCU is estimated to 5W, based on the different MCUs in appendix F. The Nikon D5300 consumes 7.4V and 2.5A i.e. 18W. The Gimbal unit has not been investigated into detail so therefore no assumption regarding its power consumption is made.

One MCU included in appendix F, which is interesting from several aspects, economy amongst others, is the mBed Nucleo-F303RE. It has 17 PWMs and CAN, which fulfils the most extreme requirements of table 4.3. It is also compatible with Arduino shields, making it able to interact with i.e. an Ethernet shield and shoulder the role as a Root node. However, if all 156 LED units would be assigned their own MCU in accordance with figure 4.9a, the total cost of the MCUs would be $156 \cdot (88 + 11) = 15444sek$. On the other hand, the setup displayed in figure 4.8 implies the need of 42 Nucle-F303REs yielding a total LED node cost of $42 \cdot (88 + 11) = 4158sek$. Moreover, a need for Raspberry Pi 2s has been identified in section 4.2.4 at a total cost of $20 \cdot 399 = 7980sek$. Attaching an Arduino Micro to the Raspberry Pi 2, priced at 149sek according to appendix F, grants it access to hardware PWMs for controlling a Gimbal unit.

4.3 Software

In order to design a modular, user-friendly and stable system, a well organized software system must be implemented. The plan, and partly the implementation, of this is presented below.

4.3.1 System logics

LED node

To be able to fulfil the requirements of capturing five photos in one second with the specified light settings the LED node needs to have the possibility of adjusting the settings of each individual LED unit that is under its control. The intensity of the LED units will be controlled with PWM signals, therefore the LED node needs to be able to configure timers coupled with the PWMs for all connected LED units. The resolution of the intensity depends on the selected hardware for the LED nodes. A maximum resolution of 16 bit is to be expected as that is the maximum value of most micro controllers. This means that the LED node should be able to read incoming intensity settings and translate these to corresponding duty cycle of all the PWM signals.

The LED node needs to be able to receive a configuration file including information of position and identity of all LED units that are connected to it. This could either be stored in memory or received from another node upon connection to the system.

Since the LED node is responsible of setting the duty cycle of each PWM signal it needs to translate the received intensity signal to a corresponding TOP value for the timers. There are several solutions to this but two major paths: either the LED node is controlled from the centralised root node and receives direct orders with specified intensity for each LED unit or these calculations are distributed and receive environmental variables from which it calculates corresponding intensity signals for each LED unit. The first case implicates that there is a one-to-one relationship between incoming settings and LED units, which puts larger demand on communications. It also requires that the root node knows the ID and position of all LED units. This could be solved by having the LED node relaying this information since it, as mentioned above, needs to be available on the LED node.

Compared to the first path the second one is less demanding on the communication since the same message could be sent to all LED nodes from the root node. It would then be the responsibility of each individual LED node to map this environmental setting to specific settings for all the LED units that are connected to it. This would be far more demanding concerning processing power on the LED nodes since the above mentioned mapping has to take place on each LED node. The amount of information regarding setup and configuration that would need to be transferred up in the hierarchy would be minimal with this kind of setup.

One of the main goals of the project is to create a flexible system that requires a minimum of maintenance. The solution that fulfils these goals to the greatest extent is the one which is distributed, to minimise the information that is needed by each node is of great importance in this aspect. Even if this solution requires a large amount of configuration of LED nodes compared to the centralised solution, as seen in table 4.9, this is something that could be automatised and since it is a non-recurring process it will not impact the requirements of maintenance negatively.

Table 4.9: System design requirement implications on LED nodes

Design	CPU	Communication	Configuration	Flexibility
Centralised	Low	High	Low	Low
Distributed	High	Low	High	High

Root node

The root node needs to maintain communication with the user and the two previously mentioned nodes. Upon start the system should be able to read predefined settings from a configuration file, of which there is an example in appendix G and propagates these to all the concerned nodes. The user will be presented with a GUI (Graphical User Interface) from which control of all camera and LED nodes should be granted. The user should be able to configure number of images taken and the lighting environment for each image in a sequence in the GUI. The flow chart in figure 4.26 illustrates the capture procedure from system start-up to finished sequence.

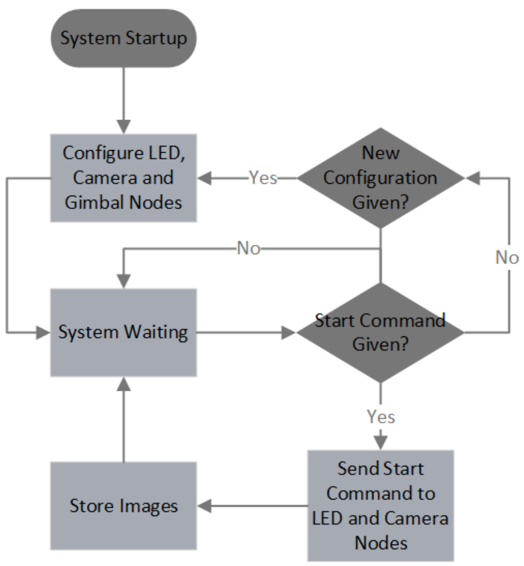


Figure 4.26: Simplified Flow Chart

If the system is designed to be centralised the processing power required of the root node will be high, table 4.10. This is due to the fact that the root node then needs to map the lighting environment to the specific intensity of each LED unit. These intensity settings then need to be transferred to via correct LED node and with correctly configured IDs. The requirements on the root node with a distributed design is considerably less complex since it only needs to propagate the lighting environment to all LED nodes on the selected communication protocol. If these differences in requirements, that these two solutions put on the root node, are taken into account the balance tilts even further in favour of the distributed design.

Table 4.10: System design requirement implications on root node

Design	CPU	Communication	Configuration	Flexibility
Centralised	High	High	High	Low
Distributed	Low	Low	Low	High

Camera node

The camera node is responsible for configuring the camera, saving images, triggering capture and communicating actions to the gimbal node. To control and gather images from the camera there are two major possibilities under consideration, the Nikon's proprietary SDK (Software Developer Kit) and gPhoto2.

Nikon SDK

The Nikon SDK is written in C++ code and is Windows and Mac compatible. The SDK provides among other things the ability to remotely capture pictures, change apparatus and stream the LiveView feed from the camera to the computer as can be seen in figure 4.27 below. However, in order to be able to run the SDK on an

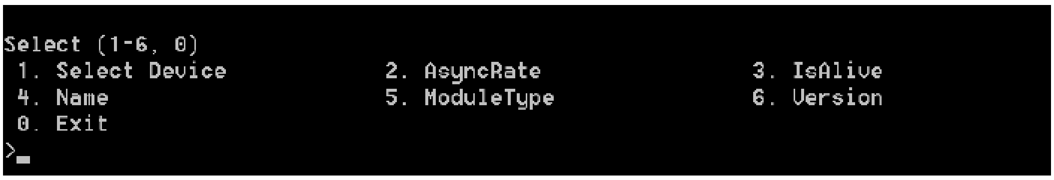


Figure 4.27: Nikon SDK Interface

embedded device it has to be modified.

GPhoto2 is a free application that enables control of cameras that is made to run on Unix systems. It supports a wide range of cameras and it is continuously developed. The software grants the user the ability to remotely capture pictures, access live-view and to change settings. No changes to the code would be needed if the camera is changed, as long as the new camera is one of the 1800 supported cameras.

Since the SDK is proprietary and only works for Nikon cameras it hardly fulfils the goal of having a system in which the modules are interchangeable. It is even model specific meaning that it has to be changed or updated if the cameras were to be changed. This is not the case with gPhoto2 since it supports more than 1800 cameras, and new cameras are continuously being added.

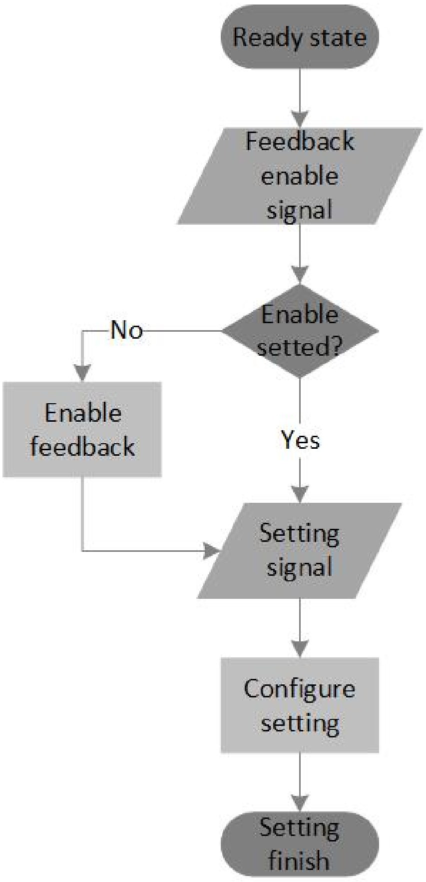
4.3.2 Software Interfaces

The various modules in the system should be interchangeable among its group with minimal overhead. This would facilitate the assembly process but makes system design increasingly difficult. To meet the goal of creating such a highly modular system well defined interfaces between the various nodes is important. The main interfaces has previously been illustrated in figure 4.6 and will be discussed further into detail here from a software perspective.

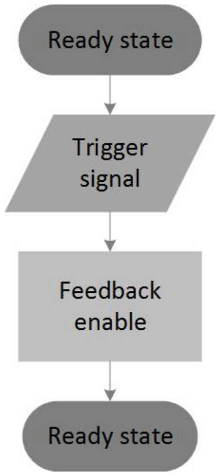
LED node

The LED node has two interfaces, towards the LED unit(s) and towards the root node. The LED node needs to be able to address all connected LED units with unique IDs. These could be mapped against the mapping already used by the root node but for simplicity they should use same the IDs. The individual LED node should also have an unique ID that is addressable from the root node. Since CAN network is to be used for the communication between root node and LED node, function algorithms are designed. In order to reduce the communication intensity, broadcast instead of peer-to-peer transmission is used. For broadcast, however, the master node cannot make sure all the nodes are getting the information since if one node sends out a confirmation after getting the message, the root node would stop sending. Therefore, a feedback checking function is also introduced.

The figure 4.28 below illustrates the procedure of the two functions in the LED nodes. Notice that these two functions are attached to interrupts.



(a) Setting function procedure



(b) Enabling feedback procedure

Figure 4.28: Functions of LED nodes regarding to CAN communication

The lighting environment could be represented by three coordinates and an intensity. This would then require the message to consist of four variables. The size of these variables will decide with what resolution the user can specify an origin for the light. The intensity variable is used to configure the gradient of the light. Using this representation of light environment means that no difference needs to be done in regard to what information is sent to which LED node. This design is better suited for a communication protocol that utilises multi-casting of messages.

Peer-to-peer communication enables messages to be sent to individual nodes, this functionality comes with an increased latency, and a comparison to multi-casting is illustrated in table 4.11 below.

Table 4.11: Communication technique implications

Distribution	**Addressability**	**Speed**	**Configuration**	**Flexibility**
peer-to-peer	High	Low	High	Low
multi-cast	Low	High	Low	High

In order to maintain modularity, the LED nodes will run the same code, although a markup language, such as the ones discussed in section 2.4.1 and exemplified in appendix G, will be used for creating a configuration file. This configuration file will be used for storing node specific configurations of different nodes in the system, e.g. information about the position and number of connected LED units.

Camera node

The camera node will communicate with the camera, the Gimbal node and the Root node. According to figure 4.18, the communication with the camera has the possibility to be done via either its WiFi or USB (Universal Serial Bus) interface. The WiFi interface requires that the Camera node is connected to a WiFi hosted by each camera i.e. a hot-spot must be created on the camera. This indicates that it is dependent on one Camera node per camera, a dependency that is not shared by the USB interface, which is one reason to why WiFi is discarded.

Over the interface to the Gimbal node actions are communicated. These commands are propagated from the user and will consist of a pitch, yaw and zoom. The choice of communication interface with the Gimbal node depends on the choice of MCU. When the Gimbal node has successfully done the desired actions an acknowledgment is sent to the Camera node, as seen in table 4.12.

The camera settings and synchronisation signals are received from the Root node as seen in table 4.12. Neither of these signals are data intensive since they consist of a list of variables with integer values.

Table 4.12: Camera Node Inputs

Camera Node Inputs	**Type**
Root Node Connection	Settings and Sync.
Gimbal Node Connection	Acknowledgments

The output required towards the Root node is images and acknowledgements. The acknowledgements are similarly to the inputs lightweight in regard of message size, this is not the case with the images. Without alteration or compression they are between 6 and 11 MB depending on light conditions. Since there are five photos taken in each sequence the amount of data that needs to be sent after each sequence will add up to 30-55 MB per Camera node. This increases the demand on the interface Root node towards Camera node in regards to throughput.

Table 4.13: Camera node outputs

Camera node outputs	**Type**
Root node connection	Images and Acknowledgments
Gimbal node connection	Actions

Root node

The Root node needs to maintain communication with the user and the two different nodes: Camera and LED. The communication with the user will consist of status messages, configuration parameters, image-data, acknowledgements and actions. Those have been described in earlier sections. They need to be presented in a GUI that is accessible for the user. This could either be an application that is running on the user's hardware or be a web-server hosted by the Root node. The Root node needs to be able to handle the inputs specified in table 4.14.

Table 4.14: Root node inputs

Root node inputs	**Type**
LED node connections	Heartbeat
Camera node connections	Images and Acknowledgments
User Connection	Settings, Sync. and Actions(Optional)

The heartbeats from LED nodes are for the Root node to get information of number of connected nodes and units. The Camera node will transfer images that will become accessible for the Root node. Apart from images the Camera node will relay acknowledgements of sent Gimbal actions, these are used by the Root node to maintain correct view of current positions of all cameras.

The user will have the possibility to enter a variety of inputs to the Root node via the GUI. Primarily the lighting environment will be configured by users and handled by the Root node. The user also has the possibility to start capture sequences and send actions to the Gimbal nodes.

Based on all these inputs the Root node is responsible of sending corresponding outputs to correct nodes. These outputs are compiled in table 4.15.

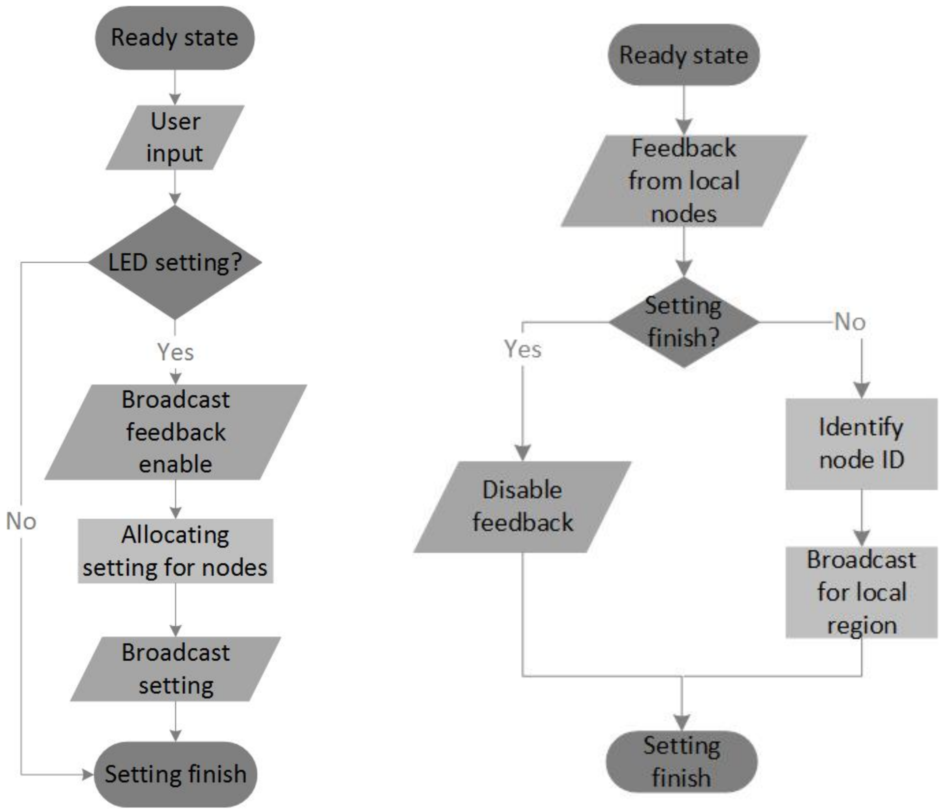
Table 4.15: Root node outputs

Root node outputs	**Type**
LED node connections	Environments and Sync.
Camera node connections	Settings, Images, Sync. and Actions(Optional)
User Connection	Images, Status and Acknowledgments

The data structure of these messages will vary depending on the protocol that is used for specific communication channels. Towards the camera a more extensive data structure is needed for the messages since there are many camera settings that should be controlled.

Based on CAN communication, functions for sending settings for the LED nodes and feedback checking are designed in this stage. Feedback functions on the LED nodes are designed as periodic functions. By disabling this function on the LED node, the intensity of the CAN bus could be lower.

The figure 4.29 illustrates the procedure of the two functions mentioned above on the Root node.



(a) Broadcast function procedure (b) Feedback checking procedure

Figure 4.29: Functions of Root node regarding to CAN communication

4.3.3 Software summary

In this section, both centralized and distributed calculation architectures have been carefully looked into, with a clear preference of the latter. Based on the distributed architecture, key functions attached to different nodes are discussed and listed.

LED node

As the most important part of even distribution light sphere, LED node demands precise information transmission while the task is time critical since the setting is finished before taking the picture. Based on this feature, a set of functions has been designed and listed in table 4.16 below.

Table 4.16: LED node software function summary

Function	**Description**	**Type**
Receiver	Receiving message from Root node and configuring it as local PWM command	interrupt attached
Feedback	Sending feedback of setting status to Root node	timer attached
Setting	Changing PWM duty circle between two shots	timer attached
Output	Sending PWM signal	loop function

For instance, when the LED node receives a message from the Root node it should configure the local PWMs accordingly.

Root node

The Root node, as a bridge between users and the system, needs to be able to handle most of the data within the system as well as the distributed architecture which has been chosen. A set of functions has been designed and listed in table 4.17 below. Since the Root node needs to communicate with user, Camera nodes and LED nodes, scheduling the tasks properly is something which still needs to be planned.

Table 4.17: Root node software function summary

Function	**Description**	**Type**
Receiver	Receiving user message and configuring it	interrupt attached
LED-Setting	Broadcasting setting signals LED nodes	sub-function of Receiver
Feedback-regarding	Dealing with feedback from LED nodes	interrupt attached
Camera-Setting	Sending Camera setting command to Camera node	sub-function of Receiver
Relay	Sending liveview of camera to user	loop function

Camera node

The Camera node is designed for adjusting the camera settings as well as communicating with an optionally attached Gimbal node. A set of functions are listed in table 4.18. How to actually implement the changes of camera settings has not been looked into detail, however the key functions have been defined without any detail-level implementation.

Table 4.18: Camera node software function summary

Function	**Description**	**Type**
Receiver	Receiving message from Root node and configuring it	interrupt attached
Setting	Sending setting message to camera	sub-function of receiver
Movement	Sending movement command to Gimbal	sub-function of receiver
Liveview	Sending liveview to Root node	loop function

Gimbal node

The Gimbal node is dedicated for the ability of adjusting the yaw, pitch and zoom of the camera. It could be optionally attached to cameras which for instance would be hard to reach. An example code implementation has been accomplished and the functions are listed in table 4.19.

Table 4.19: Gimbal software function summary

Function	**Description**	**Type**
Receiver	Receiving message from Camera node and configuring it as local movement command	interrupt attached
Yaw	Sending PWM to one servo for left/right setting of camera	sub-function of receiver
Pitch	Sending PWM to one servo for up/down setting of camera	sub-function of receiver
Zoom	Sending PWM to one servo for zooming in/out of camera	sub-function of receiver

All the functions attached to the different nodes mentioned in the tables above have been designed from top level, with interfaces between each other taken into consideration. In order to make the code reusable and the system modular, the logic designs involved in this stage focus on general behaviour, without taking specific situation into account.

Chapter 5

Demonstrator

This chapter presents the critical function prototype designed for this part of the project.

5.1 Critical Function Prototype

To try out some of the design ideas and test some basic functionalities a critical function prototype was designed. A model of the icosphere structure was built by 3D printing the junctions at the vertices and building the linkages with straws as can be seen in figure 5.1 below.

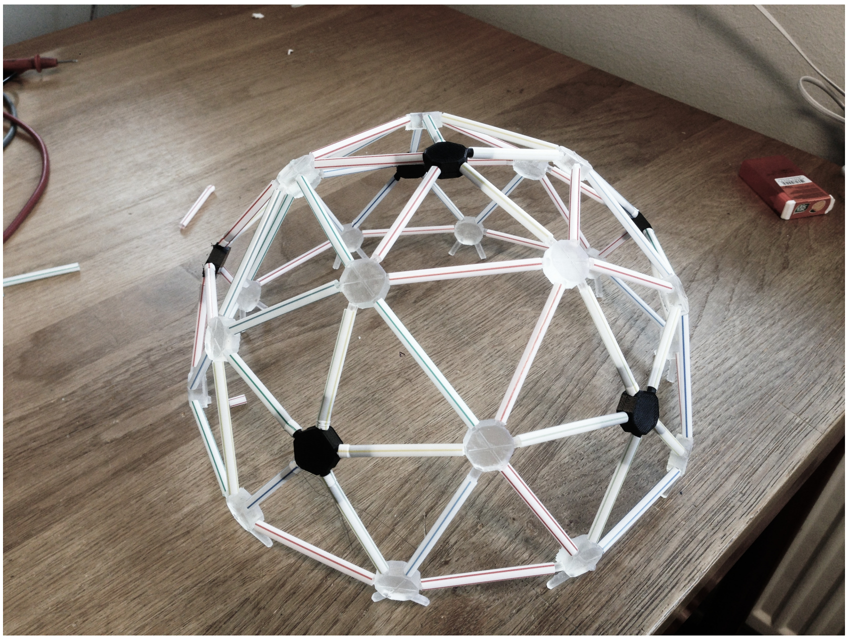


Figure 5.1: Prototype of the icosphere structure for the light rig

Five LEDs where fastened at five nodes inside the light rig prototype. The LEDs are driven with LEDdrivers, which are circuit boards made to drive the high power LEDs with the required current. The low current signals of the MCUs is amplified with a transistor in the circuit. A schematic of a LED circuit is presented below in figure 5.2 below.

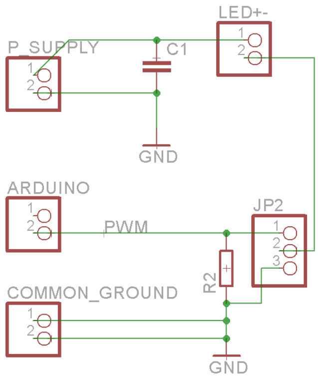


Figure 5.2: Design of LED driver circuit board where JP1 is a transistors

The LEDs and the cameras are controlled by three Arduino Uno MCUs where one is the master node controlling the cameras, the bottom LED and sending signals to the other two slave nodes. The two slave nodes are controlling the front and left LEDs as well as the top and right LEDs.

Table 5.1: Duty cycle settings for the different lighting conditions

LED Position	**Polarized Ambient**	**Non-polarized Ambient**	**Right-to-left**	**Top-to-bottom**	**Front-to-back**
FRONT LED	100%	100%	20%	20%	100%
RIGHT LED	100%	100%	100%	20%	0%
LEFT LED	100%	100%	0%	20%	0%
TOP LED	100%	100%	20%	100%	0%
BOTTOM LED	100%	100%	20%	0%	0%

Pushing an external button will cause an interrupt in the master node starting a sequence of five consecutive shots. The shutter speed on the camera has a speed of $1/200 = 5$ ms and the LEDs are synced to be on for the same time as well, at the precise time the shutter is open. Between every shot the settings for the LEDs, i.e. the duty cycle of the PWM signals, is changed to get five different lighting conditions in the five pictures. The first two shots simulate ambient light condition with and without polarized light, although no polarization filters have been used in this prototype. The last three shots simulate light conditions with light from left to right, top to bottom and front to back. The duty cycle settings for the different light conditions can be found in table 5.1. The critical function prototype with LED drivers and MCUs can be seen in figure 5.3 below.

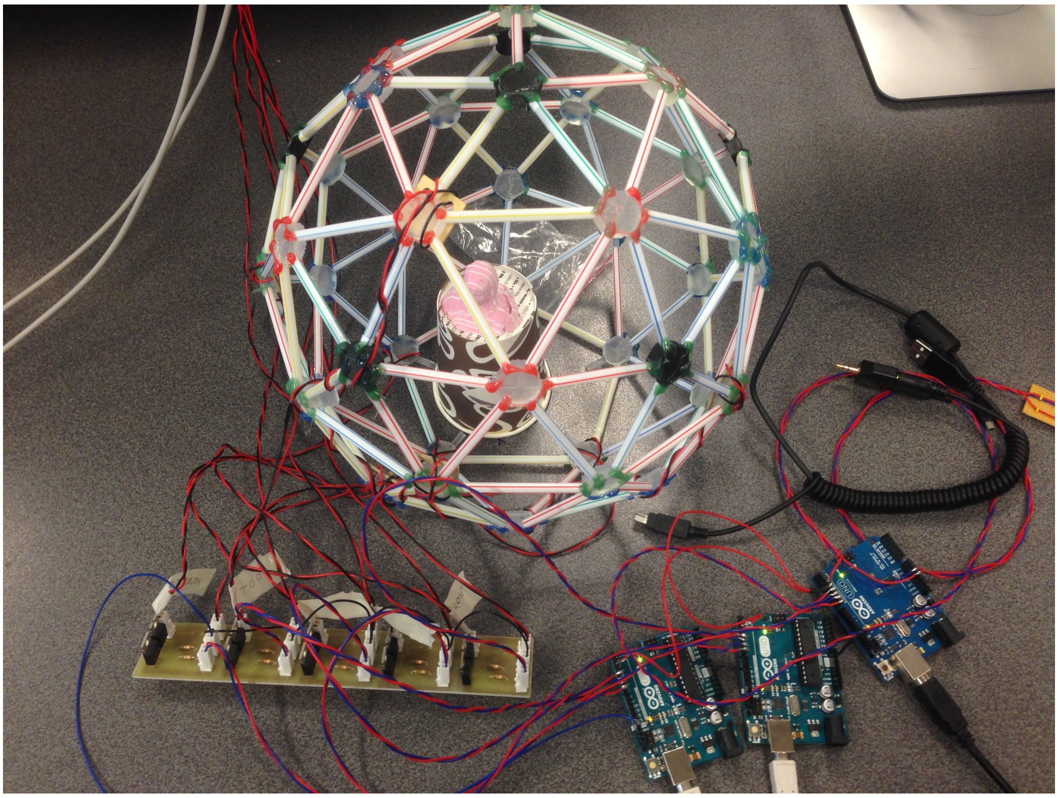


Figure 5.3: Critical function prototype including, ligt rig, LED driver boards and micro controllers

The demostrator sequence and results can be seen in figure 5.4 below.

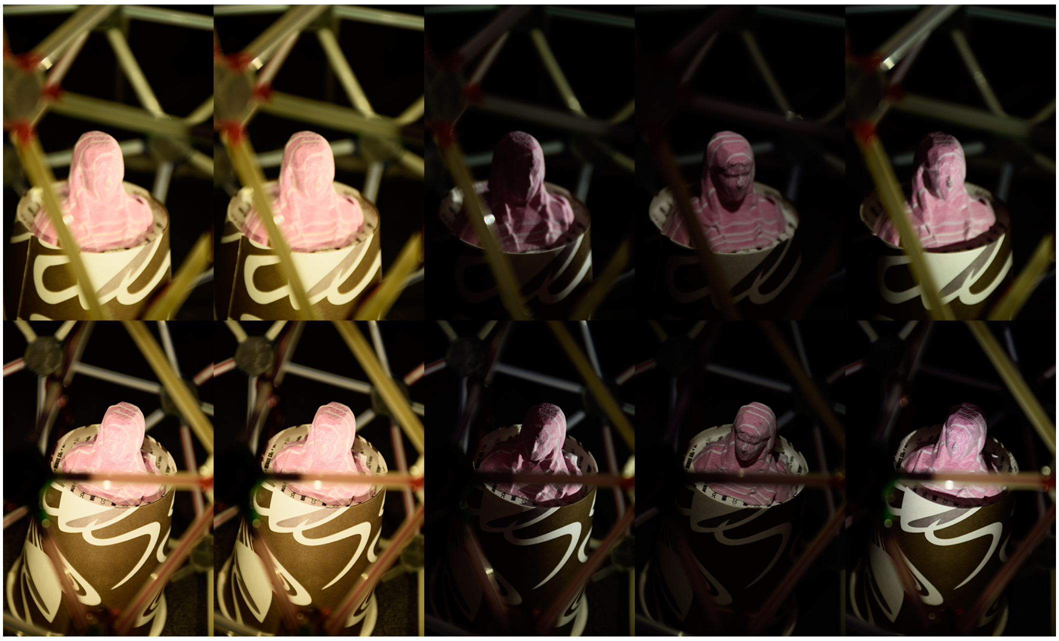


Figure 5.4: 5 synchronized pictures from 2 cameras with the 5 light settings

The results are not sufficient for verifying the simulations made in section 4.1.3, although they give an idea of what could be achieved. In order to get results which could be used for verification purposes, more LEDs would have to be attached and distributed.

5.2 Movement of the camera

In this section it is explained how the movement of the camera has been controlled. For this a Raspberry Pi 2 has been used as a master node and the camera node consists of a Raspberry Pi 2 and an Arduino, which will generate the control signal for the servo motors in this demonstration. The Raspberries are connected to each other via an Ethernet cable and the Raspberry Pi on the camera node is connected to the Arduino via USB.

The code that has been run in the master node basically establishes the communication between the two Raspberry Pi and the necessary iterations between them. The function send_to_master seen in the figure 5.5 aims to receive a user input which can be: w,s,a and d.

```
def send_to_master(socket):
        while True:
            userInput = raw_input('input (wasd)')
            if len(userInput) == 1:
                break
            print 'Please enter only one character'
        command = userInput.lower()
        socket.send(command)
```

Figure 5.5: Function send_to_master

The camera moves according to the command introduce. The functionality implemented so far is just for w and s and only for the x direction since this test is mainly to see how the communication works between the different nodes. The function receive_from_master in section 5.2 will check if the sending of the commands was successful or not.

```
def receive_from_master(socket):
        data = socket.recv(BUFFER_SIZE)
        message = "received data:" + data
        return message
```

Figure 5.6: Function receive_from_master

The aim of the code in the Raspberry Pi located on the camera node is to translate the commands sent from the computer into the proper input for the servo motor controller.

The main function in this code is the transformAction in section 5.2 below.

```
def transformAction(action):
        global x
        global y
        global f
        if action == "w":
                if x <170:
                        x=x+10
                        setAngle("x",x)
                        return "Success"
        if action == "s":
                if x >10:
                        x=x-10
                        setAngle("x",x)
                        return "Success"
        else:
                return "invalid action"
```

Figure 5.7: Function transformAction

It will check which command was sent. An increment or decrement of the actual position of the servo will be produced according the specified commands. This function calls SetAngle in section 5.2, which will send the commands to the Arduino through serial communication.

```
def setAngle(dir,angle):
        angle = str(angle)
        while len(angle)<3:
                angle = "0"+angle
        command = dir+angle
        print "command: ", command
        print len(command)
        ser.write(command)
        if ser.read() == "S":
                return True
        else:
                print "Error setting: ", angle
                return False
```

Figure 5.8: Function SetAngle

The final step of this process is managed by the controller of the servos which in this case is an Arduino. This code is simple since Arduino has a specific library to control the servos. This program will receive the parameter introduced by the user and will move the correct motor to the correct position. It will return if the process was executed successfully or not.

Chapter 6

Discussion

This chapter presents a discussion on the studies that have been carried out and conclusions that are drawn from these.

6.1 Discussion, conclusions and future work

In the State of the Art study, the leading projects within the area were examined and evaluated. Following the stated requirements, it can be concluded that the final model of this project is going to have similar qualities as the ICT Light stage X and the Esper Design. Improvements are however going to be made, in as many sub-parts as possible (e.g. glsled control, camera movement etc.) The mechanical structure of the light rig is suggested as to be a pentakis icosidodecahedron of about 1.5 meters in diameter. This is due to the fact it provides an even distribution of light sources, and since it is built using two different lengths of rods, and two different mechanical nodes, it is intended to be easily assembled and disassembled. At least 156 light sources will be distributed at the vertices and edges of the structure to be able to produce an even gradient of light. The simulations indicate that each light source must produce about 4000 lumen, but this still needs to be verified. Hardware wise the system is intended to be highly modular and consist of LED and Camera modules. Each module consists of a controlling node and a actuating unit, and whole system will be controlled by the user through a Root node. The LED nodes will be connected to the Root node through a CAN bus, and the Camera nodes will be connected to the Root node through Ethernet, mainly in order to provide high bandwidth for image and LiveView transferring. Generating the necessary lighting intensity using 156 LEDs of the types investigated, will require at least 5 kilowatts of power when the glsleds are turned on. Based on the rough estimates made in figure 4.25, the total power consumption of 20 cameras and their peripherals is around two kilowatts. Therefore, the system is estimated to require surges of about 10 kilowatts. The exact figures are highly dependant on the load of the glsmcus used, as well as the final design of LED units. However, it is argued capacitors could be used as reservoirs at the individual LED modules, which would easy the burden on a power supply when the LEDs are activated. Simulations have been made of the surges, but the required luminosity simulated in section 4.1.3 have to be verified before making further decisions. This could be done in steps, using the critical functions prototype and comparing the real world results with a simulation model of the prototype in Blender. The next step could be building either a smaller section of the larger dome, or the full-scale dome, and the results compared to the current full scale model in Blender. The theory behind polarization phenomenon has been looked into in order to get a better understanding of filters and the characteristics of best filters available on the market. A loss of luminosity at 50% has been identified when light travels from an LED to a camera and both has polarization filters mounted. Based on the glscan bus, the logic of communicating function has been carefully designed. The Camera modules will also be expandable with a Gimbal module, controlling the yaw, pitch and zoom of the camera. The Gimbal module will be developed separately and is intended to be implementable as a plug-in device on any or all of the cameras which would require pitch, yaw and zoom.

6.2 Economics discussion

In order to validate the possible choices, function as well as the economical aspect must be taken in account. The project group is striving to design a well-functioning, easy-to-replace model at a moderately low price. This vision is going to affect the choices made during the design process in the autumn. All parts, including MCUs, MCUs, cables as well as mechanical parts, must be evaluated in terms of quality, price and how easy they are to acquire if needed to be replaced. As for MCUs alone, there must also be a comparison between what tasks a high standard, expensive one can perform compared to if the tasks were to be distributed to a larger amount of cheap ones. Some of these comparisons have been made in previous chapters while others will be done during the autumn.

All economical choices made during this project are discussed with the host company prior to ordering. The purchases made during the research and development process this spring, and partly for the critical function model, can be seen in appendix H. Based on the estimations in section 4.2.3 and section 4.2.7, it is believed the least expensive solution for LEDs and MCUs would require a budget of $31400 + 7980 + 4158 = 43538$ SEK. It has been argued that the total price of a pentakis icosidodecahedron made in steel, would account for a negligible part of the total cost, and has therefore not been investigated into detail. Moreover, as is illustrated in the calculation above, LEDs would account for the major part of the total cost.

Bibliography

[1] Z. Majid, A. Chong, A. Ahmad, H. Setan, and A. Samsudin, "Photogrammetry and 3d laser scanning as spatial data capture techniques for a national craniofacial database," *Photogrammetric Record*, vol. 20, 3 2005.

[2] V. Raja and K. J. Fernandes, *Reverse Engineering - an Industrial Perspective.* Springer-Verlag London, 2008.

[3] J. Albertz, "Pioneer of photogrammetric documentation of the cultural heritage." `http://www.theulegium.de/fileadmin/user_upload/Texte/Meydenb.pdf/`. Accessed April 20, 2015.

[4] T. Akenine-Möller, E. Iaines, and N. Hoffman, *Real Time Rendering, Third edition*, vol. 3 of *1.* A K Peters, Ltd, 2008.

[5] D. D. AS, "Illuminance." `hhttp://www.dnp-screens.com/DNP08/Technology/Basic-Visual/What-is-light/Illuminance.aspx`. Accessed April 20, 2015.

[6] CGArena, "Understanding ies lights." `http://www.cgarena.com/freestuff/tutorials/max/ieslights`. Accessed April 21, 2015.

[7] R. Feynman, *The Feynman Lectures on Physics*, vol. 1 of *3.* Boston: Addison Wesley Longman, 1 ed., 2 1977.

[8] K. Zhao and X. Zhong, *Optics*, vol. 1 of *2.* Beijing: Beijing University, 1 ed., 6 2008.

[9] A. I. Lvovsky, "Fresnel equations," *Encyclopedia of Optical Engineering*, vol. 1, pp. 1–6, 2 2013. Published online.

[10] S. Chatterji, S. Sengupta, B. Rao, and D. Banerjee, *A tool for converting different data representation formats.* IEEE, 2014.

[11] P. Wehner, C. Piberger, and D. Gohringer, *Using JSON to manage communication between services in the Internet of Things.* IEEE, 2014.

[12] E. Design, "Esper design." `http://www.esperdesign.co.uk/`. Accessed May 5, 2015.

[13] U. of Southern California Institute of Creative Technologies, "Light stages." http://gl.ict.usc.edu/LightStages/. Accessed May 5, 2015.

[14] C. Lee, "Total internal reflection-based high efficiency grating design for a metal-free polarizing filter applications using hybrid optimization procedure," *Advances in Optical Technologies*, vol. 2014, 2 2014.

[15] J. Guo, Y. Liu, Z. Wang, T. Han, W. Huang, and M. Luo, "Tunable fiber polarizing filter based on a single-hole-infiltrated polarization maintaining photonic crystal fiber," *Optics express*, vol. 22, 3 2014.

[16] P. Acarnley, *Stepping Motors: A Guide to Theory and Practice*. IEE, 4 ed., 2002.

[17] B. E. Company, *Motors and Drivers*. Blador, 1994.

[18] I. Renderer, "Indigo renderer, unbiased ray tracer." http://www.indigorenderer.com/what-unbiased-ray-tracer. Accesed Apil 28, 2015.

[19] Domerama, "2v geodesic dome calculator." http://www.domerama.com/calculators/2v-geodesic-dome-calculator/. Accessed May 19, 2015.

[20] NationalInstruments, "What is the maximum cable length for a can bus?." http://digital.ni.com/public.nsf/allkb/D5DD09186EBBFA128625795A000FC025. Accessed May 19, 2015.

[21] Nikon, "Nikon digital slr camera d5300 specifications." http://imaging.nikon.com/lineup/dslr/d5300/spec.htm. Accessed May 20, 2015.

[22] H. E. Physics, "American wire gauge (awg) & metric gauge wire sizes." http://web.hep.uiuc.edu/home/serrede/P436/Lecture_Notes/American_Wire_Gauge.pdf. Accessed May 17, 2015.

[23] Belden, "Belden 7812e 0061640 cable, 7812e, cat6, 4pair, 500m." http://se.farnell.com/belden/7812e-0061640/cable-7812e-cat6-4pair-500m/dp/1218048. Accessed May 17, 2015.

Appendix A

Requirements

The requirements from Pixel Grinder can be summarized in to 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 be 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.
- The rig shall be compatible with different cameras.
- The ring shall be used for facial scanning.
- The number of the cameras should be under 20.

- The equipment used should be easy to replace in case of failure.
- The rig should be assembeled under ? time.

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 able 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 opening of ? cm2.
- The nodes shall be evenly distributed on a sphere around the object.
- The ring shall have an opening for the object.
- 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 able to control the light intensity.
- It shall provide live view from the cameras to the user.

- It shall be able to synchronize the cameras with the LEDs (Taking pictures).
- It shall be able to allow the user to move the cameras pitch, yaw and zoom.
- It shall provide serial 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 pull out 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 x .
- Each node needs at least X w .
- Each camera shall have X volts and X amps
- 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

Project Plan

Shine - Project Plan

Joakim Oscarsson & Victor Sundin

May 12, 2015

Chapter 1

Project Background

Course Perspective

This project is part of the Mechatronics, Advanced Course, at the Swedish Royal Institute of Technology. The project is mainly 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, one aim of the project is to bring together the different diciplines of the major. For integrated industrial collaboration, the project is done on behalf of a company situated in the business world.

Company Perspective

The company Pixel grinder, located in Uppsala, Sweden, is the industrial client for this project. Pixel grinder focuses on capturing images of actors and acresses, and by the use of photogrammetry, generates 3d-models of those. For the models to be detailed enough, several cameras are required to captrue images simultaneously from several angles. Pixel grinder are currently doing both face scans, and full body scans, two processes that requires different camera and light setups. For this project, two products are to be developed. The primary priority is a light rigg for the facal scan. It is beleaved, that with better light conditioning, less cameras will be needed to generate a posibly even better quality 3d-model. The solution must be modular since Pixel grinder intends to travel to abroad clients with the equipment. The solution should also be scalabe, for future implementation for the full body scan. The secondary product is an aim and focus controling system for the cameras for the full body scan. It is beleaved that with such system, the process from putting up the rigg, to taking the pictures, will speed up significantly.

Chapter 2

Project Goals

The final goal of the project, is to, by december 2015,

- Have a full working light rig that Pixel grinder can start using.
- Have a finnished concept and a working prototype of the camera aim and focus system.
- Have a technical report, describing the finnished products and their development processes.

A subgoal, of the project, is to, by the end of may 2015,

- Have a critical functions prototype of the light rigg.
- Have the chapters of the report ready, covering the background, state of the art study and descision basis for the product design.

Chapter 3

Project Organisation

Role Classification

The project members, and their dedicated roles are:

- **Abelin, Sarah** : HR Manager
- **Collin, Edvard** : Mechanical Manager
- **Escribano, Pedro** : Requirements Manager
- **Frie, Felix** : Software Manager & Project Manager (Fall semester)
- **Kiesi, Mikko** : Document Manager
- **Li, Yuchao** : Trello Master
- **Lindberg, Frida** : Company Contact
- **Oscarsson, Joakim** : Editor-In-Chief & Project Manager (Spring semester)
- **Quiroga, Sebastian** : Quater Master
- **Sundin, Victor** : Hardware Manager & Project Manager (Spring semester)

The roles are defined as:

Company Contact: Is responisble for the concact with Pixel Grinder and relaying communication to the rest of the group. Is also responsible for the budget of the project.

Document Manager: Is responisble for maintaining a tidy subversion repository - as well as keeping the repository up to date.

Editor-In-Chief: Is responsible for the report and theoretical study.

Hardware Manager: Is respinsible for the implemented hardware solutions.

HR Manager: Is responisble for human-resource realated questions. Is the primary person to speak to anonymously if personal problems occur between group members. Should also see to that everyone gets to learn the tools used in the project.

Mechanical Manager: Is responsible for file formats and CAD programs in use, as well as mechanical virtual prototypes.

Project Manager: Is responsible for project meetings, and makes sure that the team follows the plan.

Quater Master: Is responsible for the camera equipment as well as keeping the room (A311) tidy.

Requirements Manager: Is responisble for the requirements as well as verification and validation. Continous contact is maintained with the company contact.

Software Manager: Is responsible for maintaining the interface between different distributed software implementations, so that each development team can focus on their area.

Trello Master: Is responisble for updating Trello with the general tasks decided each week in accordance to the weekly meetings. Assigns the different persons in Trello to the tasks assigned to them at the meeting

Stakeholder Identification

Primary stakeholders are defined as stakeholders with which the project members will have direct contact, and are identified as:

- The teaching team
- The company Pixel Grinder
- The project members
- other occupants of the project room (A311)

Secondary stakeholders are defined as skateholders with which the project members will not have direct contact, and are identified as:

- The film industry
- The gaming industry

Contact Information

Frida Lindberg

Role: Company Contact
Phone-number: XXX-XXX XX XX
E-mail: fril@kth.se
Skype: kth_frida

Joakim Oscarsson

Role: Project Manager (Spring), Editor-In-Chief
Phone-number: XXX-XXX XX XX
E-mail: oscarsso@kth.se
Skype: oscarsson.joakim

Victor Sundin

Role: Project Manager (Spring), Hardware Manager
Phone-number: XXX-XXX XX XX
E-mail: vsundin@kth.se
Skype: victorsundin

Felix Fri

Role: Project Manager (Fall), Software Manager
Phone-number: XXX-XXX XX XX
E-mail: felixfri@kth.se
Skype: felix.fri

Sarah Abelin

Role: HR Manager
Phone-number: XXX-XXX XX XX
E-mail: sabelin@kth.se
Skype:

Edvard Collin

Role: Mechanical Manager
Phone-number: XXX-XXX XX XX
E-mail: ecollin@kth.se
Skype: edvard00

Pedro Escribano

Role: Requirements Manager
Phone-number: XXX-XXX XX XX
E-mail: pedroec@kth.se
Skype: escacri1

Mikko Kiesi

Role: Document Manager
Phone-number: XXX-XXX XX XX
E-mail: kiesi@kth.se
Skype: mikko.kiesi

Yuchao Li

Role: Trello Master
Phone-number: XXX-XXX XX XX
E-mail: yuchao@kth.se
Skype:

Sebastian Quiroga

Role: Quater Master
Phone-number: XXX-XXX XX XX
E-mail: squiroga@kth.se
Skype: live:sebbzquiroga

Chapter 4

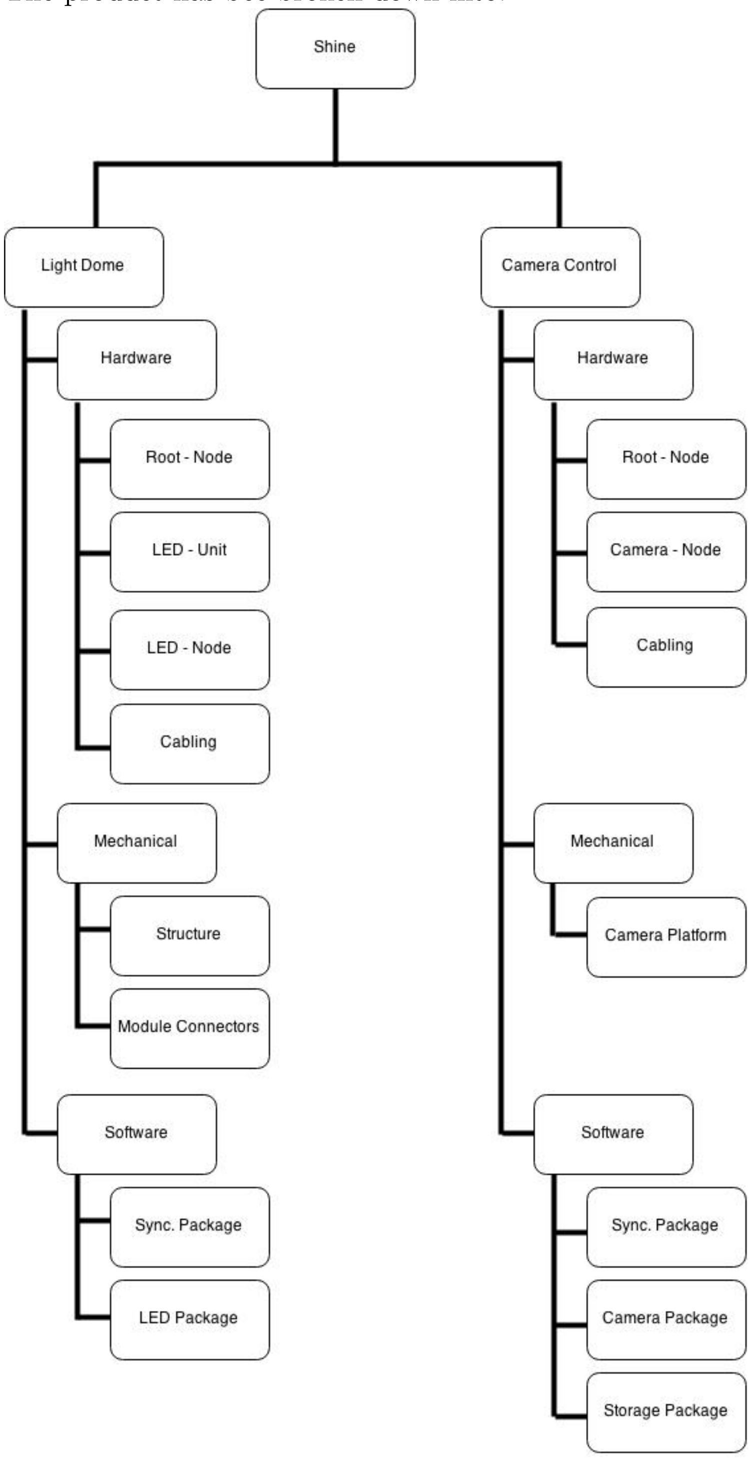
Project Model

Approach Selection

This project takes an agile approach to the every day tasks, but with an overarching plan to keep track of milestones and deadlines.

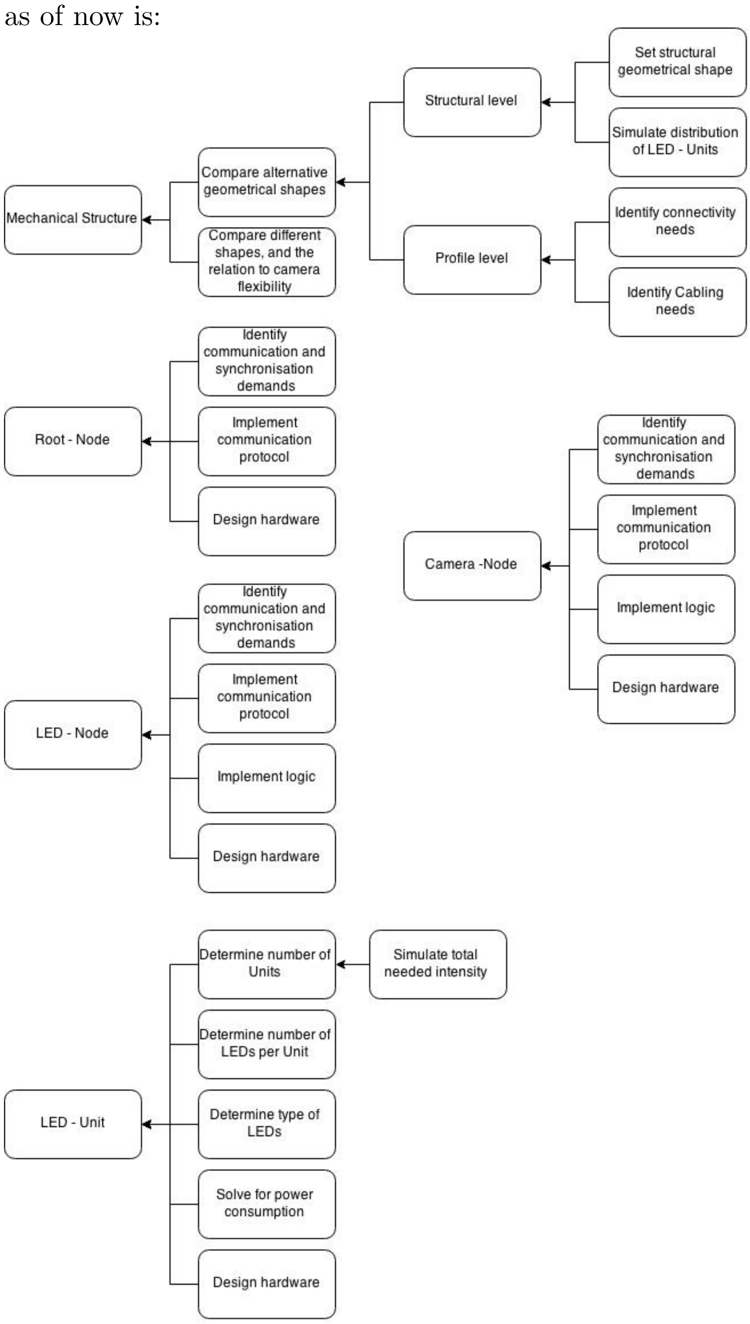
PBS

The product has bee broken down into:



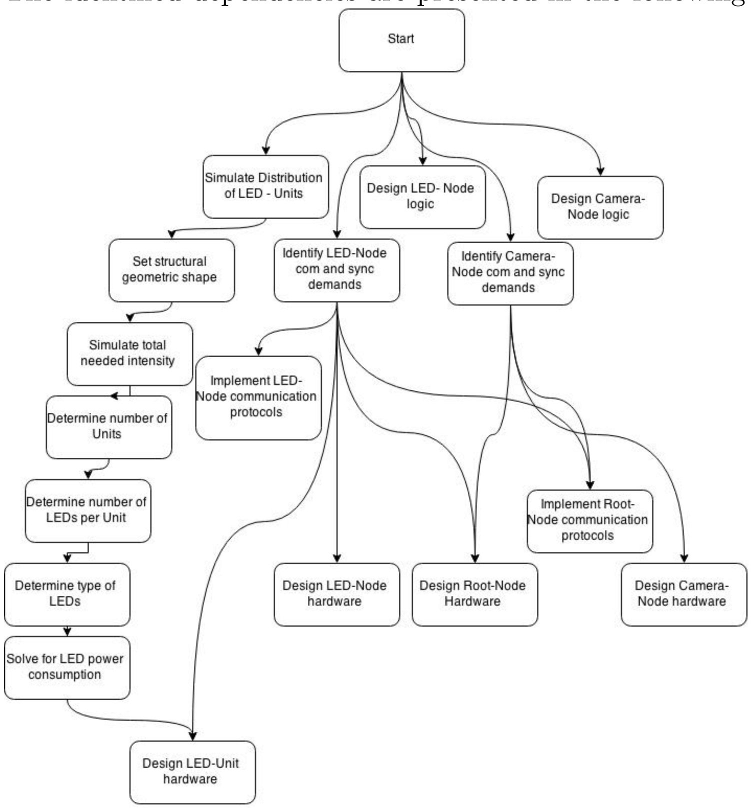
WBS

The work breakdown structure is a living document, since new subtasks arise along with the progress of the project. The work breakdown structure as of now is:



Dependencies Identification

The identified dependencies are presented in the following PERT-shcedule.



Chapter 5

Document Rules

Communication Methods

Communication within the group is primarly done via slack when direct contact is not feasable or possible. Communication with Pixel grinder is primarly done via Skype.

Versioning System

Versioning will be handled by subversion, at a repository under KTH domain.

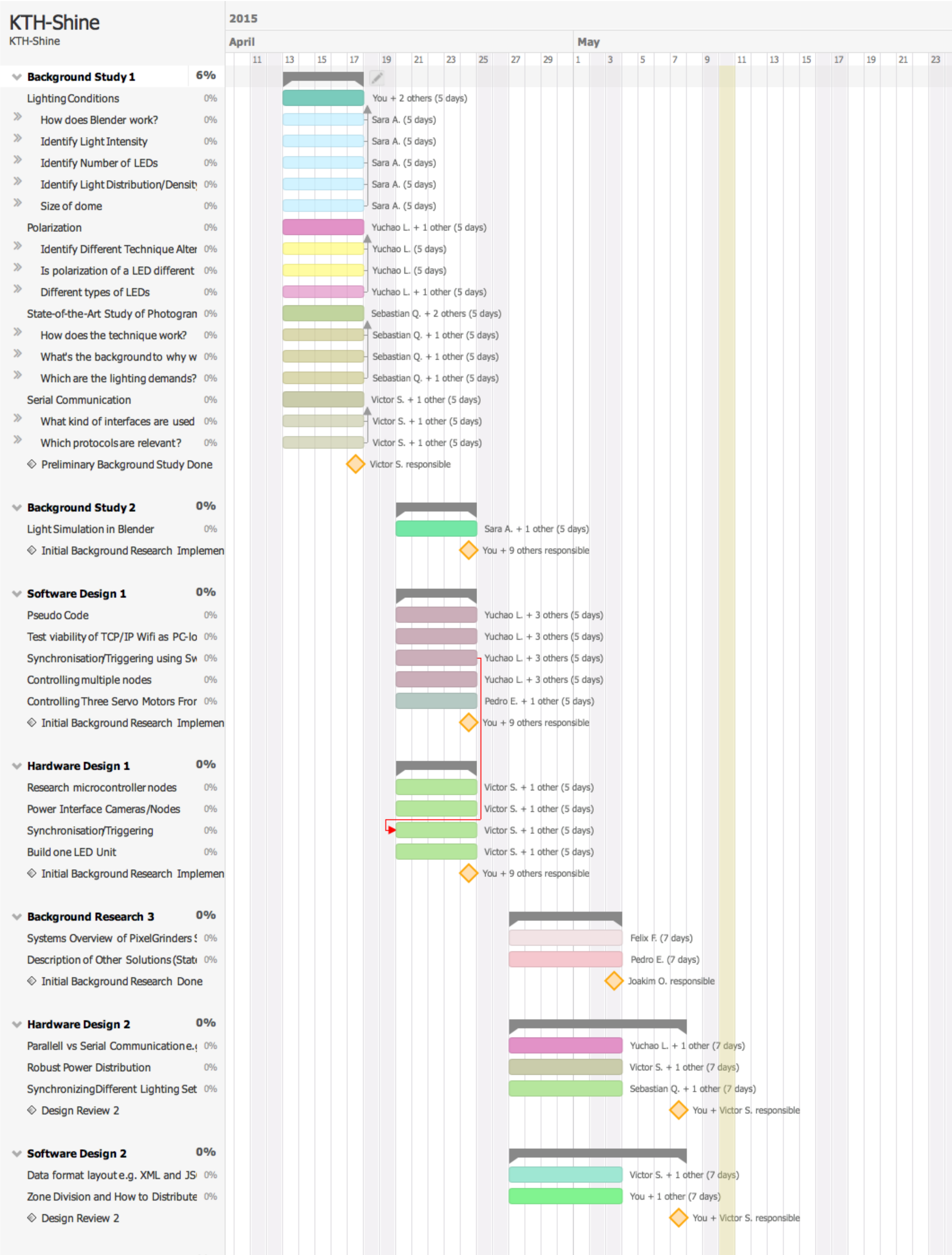
Document Naming System and Rules Description

Futher information is available in appendix Document Rules.

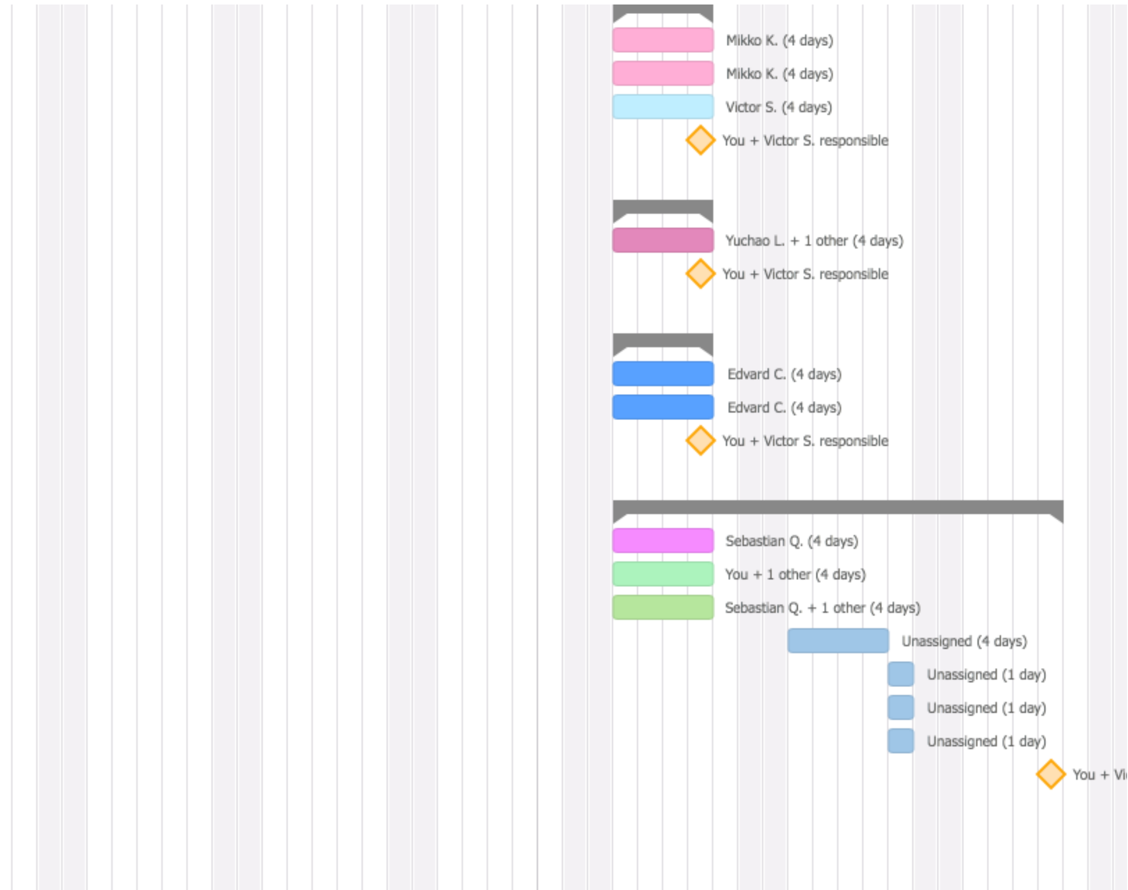
Chapter 6

Appendix

1. Time Plan
2. Risk Analysis
3. Document Rules


KTH-Shine
KTH-Shine
2015
April
May
11 13 15 17 19 21 23 25 27 29 1 3 5 7 9 11 13 15 17 19 21 23
Background Study 1 6%
Lighting Conditions 0%
How does Blender work? 0%
Identify Light Intensity 0%
Identify Number of LEDs 0%
Identify Light Distribution/Densit 0%
Size of dome 0%
Polarization 0%
Identify Different Technique Alter 0%
Is polarization of a LED different 0%
Different types of LEDs 0%
State-of-the-Art Study of Photogran 0%
How does the technique work? 0%
What's the background to why w 0%
Which are the lighting demands? 0%
Serial Communication 0%
What kind of interfaces are used 0%
Which protocols are relevant? 0%
Preliminary Background Study Done
Background Study 2 0%
Light Simulation in Blender 0%
Initial Background Research Implemen
Software Design 1 0%
Pseudo Code 0%
Test viability of TCP/IP Wifi as PC-lo 0%
Synchronisation/Triggering using Sw 0%
Controlling multiple nodes 0%
Controlling Three Servo Motors Fror 0%
Initial Background Research Implemen
Hardware Design 1 0%
Research microcontroller nodes 0%
Power Interface Cameras/Nodes 0%
Synchronisation/Triggering 0%
Build one LED Unit 0%
Initial Background Research Implemen
Background Research 3 0%
Systems Overview of PixelGrinders 0%
Description of Other Solutions (Stat 0%
Initial Background Research Done
Hardware Design 2 0%
Parallell vs Serial Communication e. 0%
Robust Power Distribution 0%
Synchronizing Different Lighting Set 0%
Design Review 2
Software Design 2 0%
Data format layout e.g. XML and JS 0%
Zone Division and How to Distribute 0%
Design Review 2
You + 2 others (5 days)
Sara A. (5 days)
Sara A. (5 days)
Sara A. (5 days)
Sara A. (5 days)
Sara A. (5 days)
Yuchao L. + 1 other (5 days)
Yuchao L. (5 days)
Yuchao L. (5 days)
Yuchao L. + 1 other (5 days)
Sebastian Q. + 2 others (5 days)
Sebastian Q. + 1 other (5 days)
Sebastian Q. + 1 other (5 days)
Sebastian Q. + 1 other (5 days)
Victor S. + 1 other (5 days)
Victor S. + 1 other (5 days)
Victor S. + 1 other (5 days)
Victor S. responsible
Sara A. + 1 other (5 days)
You + 9 others responsible
Yuchao L. + 3 others (5 days)
Yuchao L. + 3 others (5 days)
Yuchao L. + 3 others (5 days)
Yuchao L. + 3 others (5 days)
Pedro E. + 1 other (5 days)
You + 9 others responsible
Victor S. + 1 other (5 days)
Victor S. + 1 other (5 days)
Victor S. + 1 other (5 days)
Victor S. + 1 other (5 days)
You + 9 others responsible
Felix F. (7 days)
Pedro E. (7 days)
Joakim O. responsible
Yuchao L. + 1 other (7 days)
Victor S. + 1 other (7 days)
Sebastian Q. + 1 other (7 days)
You + Victor S. responsible
Victor S. + 1 other (7 days)
You + 1 other (7 days)
You + Victor S. responsible

Hardware Design 3	**0%**
LED Node Power Consumption	0%
Capacitor Storage at LED Node	0%
Cable Current vs Diameter Standard	0%
◇ Design Review 2	
Software Design 3	**0%**
Test Ethernet to CAN to CAN modul	0%
◇ Design Review 2	
Background Research 4	**0%**
Size of Dome	0%
Amount of Light in Dome (Lumen)	0%
◇ Design Review 2	
Critical Functions Prototype	**0%**
Prepare Presentation for Design Rev	0%
Tidying the report	0%
Set up the LED Synchronization Den	0%
Common Control / Integration of Ca	0%
Controlling and Synchronizing One (	0%
Control Three Servo Motors from PC	0%
Integration into one program	0%
◇ Final Presentation	
New Task List (#4)	**0%**



Project Risk Report: KTH-Shine

	Risk source		Probability			Impact			Result	Impact Areas			Mitigation / Response plan	Status
			Low	Medium	High	Low	Medium	High		Cost	Schedule	Performance		
1	Different Ambitions	Victor S.			9			9	81		X		Not defined	New
2	Not enough room to speak ones mind	Joakim O.			9			7	63	X	X	X	Feedback exercise, make a "10 sec rule" among the ones that speak the most.	New
3	Tight couplings	Joakim O.			8			7	56	X	X	X	Adress the issue on the next meeting.	New
4	Sickness	Victor S.		5				7	35		X		Not defined	New
5	Affraid to speak your mind	Victor S.		5			5		25			X	Not defined	New
6	Information Gathering Hindered	Victor S.		5			5		25			X	Maintain good contact with Björn, the company and discuss within the group	New
7	Camera rig can't be set up at KTH	Victor S.	2				5		10		X	X	Not defined	New
8	No actual spot to place the rig is appointed	Victor S.	1				6		6		X	X	Not defined	New

Risk Item Details

	Risk	Date Created	Created By	Date Updated	Updated By
1	Different Ambitions	12/04/2015 2:18PM	Victor Sundin	12/04/2015 2:18PM	Victor Sundin
2	Not enough room to speak ones mind	28/04/2015 12:07PM	Joakim Oscarsson	28/04/2015 12:10PM	Joakim Oscarsson
3	Tight couplings	28/04/2015 12:06PM	Joakim Oscarsson	28/04/2015 12:06PM	Joakim Oscarsson
4	Sickness	12/04/2015 2:24PM	Victor Sundin	12/04/2015 2:24PM	Victor Sundin
5	Affraid to speak your mind	12/04/2015 2:22PM	Victor Sundin	12/04/2015 2:22PM	Victor Sundin
6	Information Gathering Hindered	12/04/2015 2:25PM	Victor Sundin	12/04/2015 2:25PM	Victor Sundin
7	Camera rig can't be set up at KTH	12/04/2015 2:29PM	Victor Sundin	12/04/2015 2:29PM	Victor Sundin
8	No actual spot to place the rig is appointed	12/04/2015 2:27PM	Victor Sundin	12/04/2015 2:27PM	Victor Sundin

Document rules

This document describes the general rules for document naming etc. and is currently work in progress. Suggestions and criticism are more than welcome so we can forge a working document policy that satisfies everyone! -Mikko

Document naming

Document names consist of four parts as in the template below

DC-YYMMDD-XXXX-documentDescription

1. Document class symbol (**DC**) based on the Table 1 presented further in this document.
2. Date on which the document is published expressed with six numbers (**YYMMDD).**
3. Unique document number **XXXX**. Document number is selected based on AD-150405-0000-documentNames
4. A short description/name of the document. In principle as short and describing as possible.

For example this document is named

AD-150405-0001-documentRules

Document is an *Administrative document* written on *5th of April 2015*, its unique document number is *0001* as the previous number 0000 is the list of document numbers. Also this document presents the document rules so description is obviously *documentRules*.

General guidelines

- When adding new documents to the repository pick the document number from AD-150405-0000-documentNames and commit the updated version of it ASAP to avoid confusions.
- When naming files avoid using special characters including Å, Ä and Ö.
- If a document description consists of multiple words use capital letters to separate words like**T**his**F**or**E**xample.

Document classes

Table 1 shows the current document classes used. If you feel like there is not a suitable class for a document/design you are making consider adding one to the list. Keep in mind that the classes are more or less loose.

Table 1, Document classes

Document Class	Class symbol
Administrative Document	AD
Mechanical Design	DM
Electric Design	DE
System Design	DS
Meeting Agenda	MA
Meeting Notes	MN
Report	RP
Research	RS

Appendix C

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.

Appendix D

Interface Overview

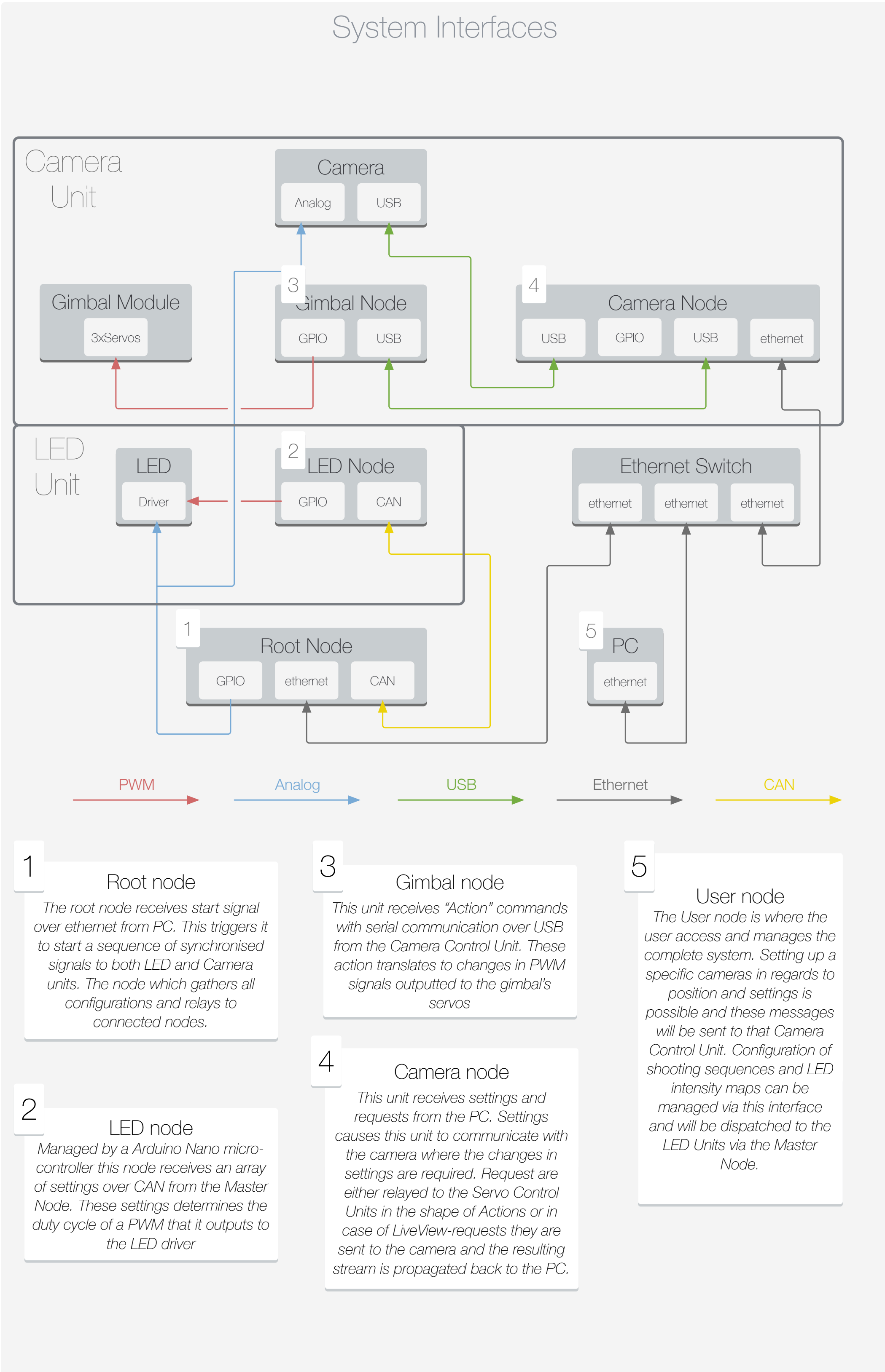


Figure D.1: System overview, illustrating the interface layout of the nodes

Appendix E

LED Comparison

Table E.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 F

Microcontroller Comparison

Root Node

Table F.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 F.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 F.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 F.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 G

Examples of markup language implementation

We developed two examples of files to store the configuration files and how the data would be translated. As a basis for these examples the use case of the root node has been used since it is the broadest case. The first test was done in XML of earlier stated fashion in section 2.4.1 and it can be seen in figure G.1 below.

XML File

```
<cameraNodes>
   <cameraNode id='IPAddress or CAN ID'>
      <liveView>false<\liveView>
      <gimball>
         <setting>
               <pitch>180.0<\pitch> //degrees
               <yaw>180.0<\pitch>   //degrees
               <zoom>100.0<\zoom>    //percent
         <\setting>
      <\gimball>
   <\cameraNode>
<\cameraNodes>
<ledzones>
   <top>
      <ledNode id='IPAddress or CAN ID'>
         <settings>
            <setting>
               <polarized>false<\polarized>
               <intensity>50.0<\intensity>  //dutycycle
               <duration>10.0<\duration>    //milliseconds
            <\setting>
         <\settings>
      <\ledNode>
   <\top>
<\ledzones>
```

Figure G.1: Example XML Formatting

A second example with the same information as contained in figure G.1, however using JSON formatting is illustrated in figure G.2 below.

JSON File

```
{"cameraNodes": [
   {"cameraID": "IPAddress or CAN ID",
   "liveView": false,
   "gimball": {
      "setting": [
            {"pitch" : 180.0}, //degrees
            {"yaw" : 180.0},   //degrees
            {"zoom" : 100.0}   //percent
      ]
   }}
],
"ledzones": [
   {"top" :[
      {"ledNodeID": "IPAddress or CAN ID",
         "settings": [
            {"setting": [
               {"polarized" : false},
               {"intensity" : 50.0},   //dutycycle
               {"duration" : 10.0 }    //milliseconds
            ]}
         ]
      },
   ]}
]}
```

Figure G.2: Example JSON Formatting

As figure G.2, the JSON format is containing the same information is a more compact package. A simplified flowchart of the parsing procedure is illustrated in below.

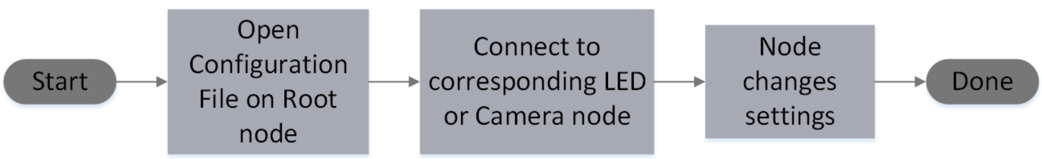


Figure G.3: Parsing Procedure

Appendix H

Purchases

Table H.1: Research and development purchases for the spring of 2015.

Article	**No.**	**Date**	**Reseller**	**Total Price**
Arduino Micro	4	42123	Elfa	664 SEK
Raspberry Pi 2	5	42123	Elfa	1695 SEK
Current supply Rpi	1	42122	Kjell Co	149 SEK
Memory Card	5	42122	Kjell Co	845 SEK
Ethernet shield	3	42123	Elfa	801 SEK
LEDs Natural light	5	42122	Elfa	240 SEK
MCP2551 for CAN	2	42161	Farnell	22 SEK
Nucleo board	2	42161	Farnell	174 SEK
TOTAL				4590 SEK 560 US Dollar