Measurement Light Bulb

EE3L11

Electrical Engineering Bachelor Thesis

Subgroup:
Control

Authors:
R.A. Damsteegt
M.J.J. van Dunné

Supervisors:
Prof. dr. E. Eisemann
Dr. M. Billeter
N. Salamon

July 22, 2019

DELFT UNIVERSITY OF TECHNOLOGY

FACULTY OF ELECTRICAL ENGINEERING, MATHEMATICS
AND COMPUTER SCIENCE

ELECTRICAL ENGINEERING PROGRAMME

Abstract

This document describes the design process the prototype of a measurement light bulb which is able to project spherical harmonics of orders 0-1-2. Photographs of these projections can be combined to represent many light distributions. The measurement light bulb can be used in the field of research focused on the computation of the illumination impact of lighting, more specifically, simulating different light sources. Our prototype consists of a laser beam rotating over two axes, which allows the device to project onto a sphere around itself. The device can be controlled wirelessly using Bluetooth.

The lamp can project spherical harmonics in a resolution of $4°$ and 256 monochrome light levels. The time one projection takes is about one second, which allows for quick measurements. The dimensions and the weight of the lamp are such that it is portable. At this stage, the prototype can only operate in a dark environment, due to the use of a low powered laser.

In this document, the focus will be on the design of control, communication and the PCB.

Preface

We would like to wholeheartedly thank everyone involved in this project.

Our supervisors Elmar Eisemann, Markus Billeter and Nestor Salamon for giving excellent guidance during this project. They were always compassionate to us and brought excellent knowledge to the table. They also delivered very helpful advice regarding the writing of our theses.

Ioan Lager also receives our gratitude for organising the bachelor graduation project and keeping things on track.

Another person we would like to thank is Jianning Dong, who gave us advice regarding motors and their corresponding controllers.

We would also like to thank the Intelligent Systems Department for letting us use their cluster computer, which allowed us to run a heavy optimisation algorithm for pattern calculations, and letting us use their 3D printer, which allowed us to easily produce custom parts.

Lastly we would like to thank Martin Schumacher for being supportive of our project and his dedication to make sure all our components were delivered in time.

Contents

1 Introduction 1
- 1.1 Prerequisite knowledge about spherical harmonics 1
- 1.2 Related work 2
- 1.3 Problem definition and challenges beyond previous work 3
- 1.4 Criteria for the prototype 3
- 1.5 Project dynamics 3
- 1.6 Thesis outline 3

2 Design overview 5
- 2.1 Design 5
- 2.2 System breakdown 5

3 Requirements for the system 7
- 3.1 Assumptions on the system and its environment 7
- 3.2 System requirements 7

4 Pattern calculation 11
- 4.1 Theory 11
  - 4.1.1 Sources of projection distortion 11
  - 4.1.2 Solution to projection distortion 13
- 4.2 Requirements for the pattern calculation 13
- 4.3 Implementation 14
  - 4.3.1 Issues 15
- 4.4 Results and discussion 15

5 Wireless communication 17
- 5.1 Requirements for the wireless communication 17
- 5.2 Implementation 17
  - 5.2.1 Wireless protocol 17
  - 5.2.2 Hardware 18
  - 5.2.3 Software 20
- 5.3 Results and discussion 20

6 System integration 21
- 6.1 Theory 21
  - 6.1.1 Overview of the subsystems 21
  - 6.1.2 Solution 21
- 6.2 Requirements for the system integration 22
- 6.3 Implementation 22
  - 6.3.1 PCB design 22
  - 6.3.2 Microcontroller 23
  - 6.3.3 Software 24
- 6.4 Results and discussion 24

7 Prototype implementation and validation results **25**
7.1 Fabrication of the prototype 25
7.2 Results of complete system tests 26

8 Discussion of results **27**

9 Conclusion and future work **29**
9.1 Conclusion 29
9.2 Future work 29

Appendices **35**

A Design choice **35**
A.1 Design options 35
A.2 Design choice 36

B Wireless circuit simulation **37**

C Pattern calculation CUDA algorithm **39**

Chapter 1

Introduction

Almost nothing is as much part of our everyday life as light. It is therefore no surprise that light can have a large impact on us: correct lighting can set the right mood and is thus also an important element in interior design. Designing a room requires knowledge of how the room is to be lit, what it is going to look like and how this influences the people inside. Simulating lighting plays a big part in this. Computer programs have been developed specifically for this purpose. However, simulating light, either from a lamp or from the sun, turns out to be hard, since the way it spreads across a room depends on a large number of factors, a lot of which are unknown. Due to this, computer simulation is very costly and difficult to control. This is where our project comes in.

Instead of simulating lamps via computer programs, we opt for a simulation with a physical measurement light bulb. This measurement light bulb has been built to project a finite set of special light patterns: spherical harmonics (SHs), which can be combined to represent many light distributions.

The principle is to capture the device's illumination impact from a given point in the room with the aid of a camera. The photos, resulting from capturing several spherical harmonics projections from the measurement light bulb, can be combined in a post-process to yield the result that would have been obtained by a different light source placed at the same location. For any light source, a specific combination of SHs coefficients exist so that their weighted linear combination yields an approximation of the light's emission pattern. In consequence, by combining the photos using the same coefficients, we can predict the illumination of the given light source. The resulting technique is a lot more efficient than simulating all reflections of the room.

When the light bulb is able to project high resolution SHs and the reflectance properties can be captured, many sorts of light sources could be simulated. Hereby, our work can become a useful tool for interior designers to simulate lighting in a room, or even an entire building. It is possible to have a preview of the results of lamp arrangements without physically installing these. Hereby, the user saves a lot of time and as the capture is only performed once, it enables the exploration of many design options in a post-process.

In this chapter, some prerequisite knowledge about spherical harmonics will first be discusses. This is followed by a brief overview of some related work. After this, the problem definition is laid out. Next come the criteria for the prototype, an overview of the project dynamics and lastly an outline of the entire thesis.

1.1 Prerequisite knowledge about spherical harmonics

As indicated in the introduction, the idea of our solution is to approximate the light emission pattern of a light source using spherical harmonics and rely on our measurement light bulb to actually emit the spherical harmonics pattern into the scene. In order to provide a better understanding, we will first briefly revisit spherical harmonics.

Spherical harmonics are a set of orthogonal basis functions over a sphere. In principle, they can be thought of as the equivalent of a Fourier basis on a spherical surface. In consequence, they can be used to represent various spherical functions by projecting the function into their spherical harmonics basis.

The higher the order of the basis, the more accurate the representation of the function becomes (similar to adding frequencies for the Fourier case).

The general equation for complex spherical harmonics is given in Equation 1.1. However, since complex light cannot be projected, they are transformed to form a real basis.

$$Y_l^m(\theta, \phi) = Ne^{im\phi}P_l^m(cos(\theta)) \tag{1.1}$$

In Equation 1.1, l is the order and m is the index of the spherical harmonic function Y, N is the normalisation constant and P is the Legendre polynomial. ϕ is the longitude and θ the colatitude of the sphere as can be seen in Figure 2.2.

In our work, we will focus on a lower dimensional SH space up to order two (as illustrated in Figure 1.1). In the future, higher order representations could be added, but it has been shown that the first three bands can be used to make very accurate representations [Ramamoorthi and Hanrahan, 2001].

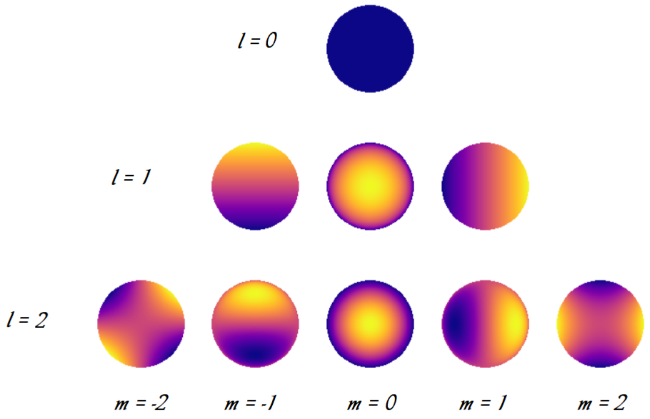


Figure 1.1: The first three order of spherical harmonics. l indicates the order, m the index. Source: [Luger, 2018]

1.2 Related work

The goal of illuminating an environment in a post-process instead of actively setting up lights has received much interest in the past from the vision and graphics community but none focused on building a portable device to support the simulation of arbitrary emission patterns.

For omnidirectional light sources, an algorithm exists, which served as the inspiration for our work. In this case, a lamp is used to sweep the environment, while recording a video. The captured imagery can then be used to relight the scene [Salamon et al., 2017].

Multiple projects study the interaction of light and objects by projecting light from a fixed measurement setup around the object [Debevec et al., 2000; Hawkins et al., 2001; Levoy et al., 2000; Tunwattanapong et al., 2013]. A big disadvantage is that most of these projects require large constructions that are hard to bring to different locations and cannot be used to project outwards into an entire room.

Until now only a couple of projects exist that direct light patterns outward into an entire room. Most of these projects are about screens to be looked at. These screens can take the form of a sphere, or a spinning ring of LEDs that creates an image through persistence of vision (POV) [GreatScott!, 2017; Groenendijk, 2018; Yamada et al., 2017]. While it might seem possible to use such screens as a light source and emit patterns into the room, their light is diffuse and thus does not allow us to derive a clear projection of the displayed pattern to the surroundings.

1.3 Problem definition and challenges beyond previous work

For this project, the aim is to make a portable setup that projects spherical harmonics outward, making it possible to cover an entire room as opposed to just objects. This project differs from the ones in Section 1.2. Firstly, we need an outward projection instead of an inward projection. Secondly, a certain resolution is required in order to be able to project spherical harmonics, even at varying distances, since not every wall has the same distance to the measurement light bulb.

1.4 Criteria for the prototype

The main criterion for the prototype is that it needs to be able to project spherical harmonics at such an intensity and resolution that the impact of the illumination can be captured by a camera. The photos of these projections can then be used to accurately simulate any light source. This process should be easy and relatively fast, allowing the user to intuitively use this device to simulate lighting options for the room. Another criterion is that the device should be safe to use, portable and cheap, allowing research groups to build one and easily test it out themselves.

1.5 Project dynamics

The project proposer is the Computer Graphics and Visualization (CGV) group, thus the prototype is to be delivered to them. The bachelor graduation project is executed in a group of six people, that is divided into three subgroups of two people. Each subgroup is responsible for a specific part of the project and hands in a corresponding thesis. The distribution of the subgroups is as follows:

- Rob Damsteegt & Jippe van Dunné → control, communication and PCB design
- Sebastian Jordan & Ids van der Werf → power distribution, motors and construction
- Bob van Nifterik & Jurgen Wervers → light sources, the corresponding drivers and camera synchronisation

All of these three theses will cover the project globally and their own subsystems in more depth. The thesis outline given in the next section is therefore specific to this thesis.

1.6 Thesis outline

The thesis will start off with a general design overview, followed by a programme of requirements for the complete system. After this, the design parts specific to this thesis are laid out. These are three chapters about the pattern calculation, the wireless communication and the system integration. After these, the complete prototype is treated and its results are presented. Following this come the last two chapters of the thesis: 'Discussion of results' and 'Conclusion and future work'.

Chapter 2

Design overview

At first, some different design options were drafted. Of these options, one was chosen. An overview of these design options and the reasoning behind the choice can be read in Appendix A.

In this chapter, an overview of the final design will be given. First a general overview is given, followed by a short breakdown of all the subsystems and their placement on the construction.

2.1 Design

A basic overview of the design is given in Figure 2.1. It consists of a laser which is aimed at a fast spinning mirror, driven by a motor (marked 'M'). This construction is driven by a motor (marked 'B'), which rotates slower, such that a small angular step is taken for each complete rotation of Motor M. This allows the laser beam to turn over two axes and to project onto a sphere around itself, as depicted in Figure 2.2. In this figure the colatitude is indicated by θ and the longitude is indicated by ϕ.

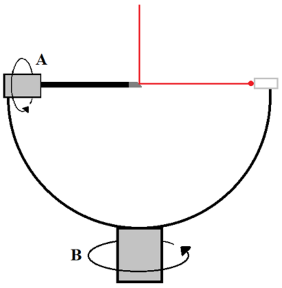


Figure 2.1: Impression of the measurement light bulb design

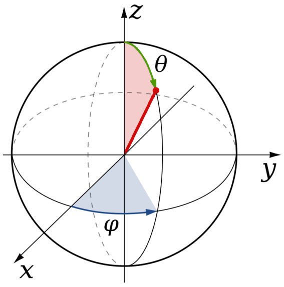


Figure 2.2: Depiction of red laser beam moving over two axes. Source: [Ibrahim, 2009]

In this process, the laser 'draws' the spherical harmonics on the walls of the room it is positioned in. Since spherical harmonics contain positive as well as negative values, but 'negative light' does not exist, the positive and negative parts are projected with separate patterns. This can later be compensated for in post-processing.

These projections are captured by a camera, of which the exposure time is controlled to capture an entire projection, so that every part of the room is lit once.

This system is controlled by a microcontroller, which regulates the rotation speeds for both axes, controls the laser and handles communication with the user. This controller, together with some other subsystems, is placed on a printed circuit board (PCB), which is mounted on the moving construction. Power is fed to the system through the lower axle using a slip ring. Control signals for Motor B and the camera are also sent over this slip ring.

2.2 System breakdown

The system consists of seven subsystems which are discussed in this section. The location of the subsystems in the final design can be seen in Figure 2.3.

Controller The controller is the one that binds all the subsystems together. It monitors speeds and angles, reads pattern data and converts it to a drive signal for the laser and controls the motors.

Motor M(irror) This motor rotates the mirror. Motor M is implemented with a DC motor and a driver to operate it. The motor as well as its driver are placed upon the moving construction. Attached to the axle of this motor is the mirror. The driver has a pulse-width modulation (PWM) signal as input, which dictates the speed. The speed of this motor is monitored by a Hall effect switch, which gives a pulse on each rotation.

Motor B(ase) This motor is used to rotate the upper part of the construction. Motor B is implemented with a stepper motor and a driver to operate it. Motor B and its driver are located in the base of the construction and control signals are transmitted from the controller through the slip ring. This slip ring is placed upon the lower axle and the stepper motor is moved off axis and connected with a belt. The stepper motor driver has two inputs, a direction input which determines in which direction it spins and a step input, which makes the motor rotate a step for each pulse sent to it. The speed and orientation of this motor are monitored by a Hall effect switch.

Laser The laser and its driver are also placed upon the moving part. The laser is focused on the mirror using a lens. The laser driver takes a PWM signal as input and produces an accurate drive signal for the laser.

Bluetooth module The Bluetooth module is for connecting the device to for example a phone, which allows the user to control it from a distance. It also enables transmission of pattern data from the user device to the lamp.

Camera The camera is connected to the controller via the slip ring. It is synchronised with the complete system using the microcontroller to make be able to capture exactly one projection with as little noise as possible.

Power supply The power supply is partly on-board and off-board. The on-board power supply contains some voltage regulators, which produce the voltages required for all the different subsystems. This on-board power supply takes a 12 V DC input supplied from the off-board part.

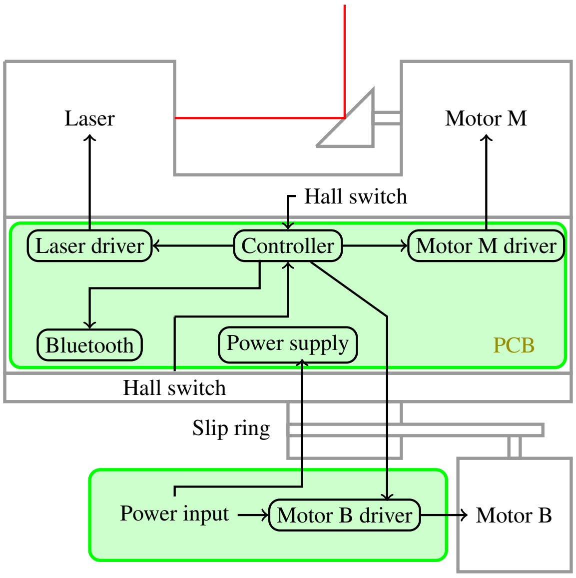


Figure 2.3: Design overview. Circuit boards are depicted by a green frame.

Chapter 3

Requirements for the system

In this chapter, an overview of the requirements of the system is given.

In general, the product should be able to project SHs at a certain light intensity so that the reflections of the pattern can be captured by a camera. The required camera settings are strongly dependent on the type of room, e.g. the amount and direction of lighting in the room. Therefore, the assumptions on the environment in which we aim the product to be working will be specified first. After that, the requirements for the system in order for it to work in the assumed environment will be stated.

3.1 Assumptions on the system and its environment

To be able to simulate lamps, all different light colours need to be simulated. However, we assume that a proof of concept of this device can also be given if the measurement light bulb only projects in one colour. This will not allow for every lamp to be simulated, since only monochromous simulations are possible.

We also assume that the lamp can be tested in a dark environment, in case the lamp is not bright enough to create visible projections in a lit up room. This allows for a proof of concept even if the light source is not powerful enough for projections in daylight.

3.2 System requirements

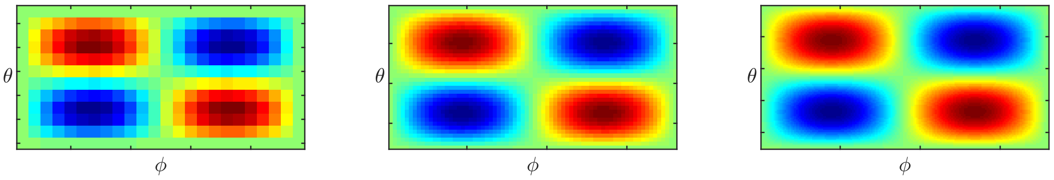
In Section 1.4, some system criteria for the prototype were specified. In this section, these criteria will be further specified and quantified. First some definitions will be elaborated, followed by the requirements.

Some definitions that are important for understanding the requirements are given below.

Pattern: This is the projection of one spherical harmonic (SH), of which one or multiple photos can be taken.
Measurement: This is a complete set of patterns, which together form a complete picture. For example, one pattern could be an SH of any of the orders that are desired to display. If the first three orders of SHs are displayed, this means 9 (=1+3+5) SHs in one measurement. Since all the SHs require a positive and negative projection (except for order 0), this results in 17 photographs.

The requirements for the system are listed below and in Table 3.1, where they are specified in three categories: must have, should have and could have.

Angular resolution The angular resolution of the projection is defined in degrees. This resolution is the same for both the longitudinal as the latitudinal direction. These values were chosen based on simulations. 15° allows for the SHs to be distinguished, but for an accurate representation, an angular resolution of at least 5° is required. A plot for indication can be seen in Figure 3.1.



(a) Angular resolution of 15° (b) Angular resolution of 5° (c) Angular resolution of 2°

Figure 3.1: Three plots of the spherical harmonic function of order 2, index -1 at different angular resolutions

Projection time This is the amount of time one pattern projection takes. It is proportional to the resolution and inversely proportional to the speed of the motors. It is required that it takes a finite time to project the pattern correctly, but ideally, this is in the order of a minute or a second.

Power Since portability is a criterion, the choice of the power source is important. The most simple source of power is a regular power outlet, but for more portability, an implementation with a battery would be more suitable.

Start-up time This is the time required between switching on the device and the start of the first pattern projection. This involves spinning up the motors, buffering the projection from the memory and synchronising the system. Between pattern projections, the time required for setup is less, because the motors are kept spinning and only the memory needs to be updated for the next pattern.

Dimensions For better portability, smaller dimensions are preferable. The prototype should fit through a door, so measurements of any room can be taken.

Phase drift Phase drift is the deviation in average speed of the projection over one of the two axes. Ideally this average speed deviation would be zero, because then the pattern does not shift over the sphere. The allowable amount of phase drift is the same for both axes. The values for this are chosen based on the angular resolution, meaning that they are the same amount of degrees per revolution (divided by two, to account for the positive and negative direction of the phase drift).

Maximum phase error This is the maximum phase error of the projection caused by vibrations or instabilities of the system. This means that there can still be a phase error while the phase drift is zero. The maximum phase error is the same for both axes and has the same bounds as the phase drift.

Colour of the light source To simulate any lamp, the projection should contain all colours. However, monochromous design can also suffice for a prototype.

SH orders The more SH orders are projected, the more accurate the resulting computer simulations can be. It is required that at least the SHs up to order two can be projected for simple simulations.

Brightness This involves the power of the light source on the device. The power should at least be high enough to be able to capture projections in a dark room. However, higher power would allow for capturing in brighter environments, ideally in daylight.

Light levels The amount of different light levels contributes to the resolution of the projection. The more light levels, the less quantisation noise occurs in the projection. However, since the camera has $14\,\mathrm{bit}$ light levels, a higher accuracy than this is not visible.

Camera synchronisation For capturing these projections, synchronisation with the camera is desired. This allows the exposure time to contain exactly one pattern projection, which minimises the amount of captured noise light.

User interface For controlling the device, a user interface (UI) is required. Any working UI suffices for the prototype but a graphical UI (GUI) is preferred for a more intuitive experience.

Table 3.1: Overview of complete system requirements

Type	Must have	Should have	Could have
Angular resolution	15°	5°	<5°
Projection time	”converges”	$1\,\mathrm{min}\,\mathrm{pattern}^{-1}$	$1\,\mathrm{s}\,\mathrm{pattern}^{-1}$
Power	$230\,\mathrm{V}$, $16\,\mathrm{A}$	-	battery powered
Start-up time	$<15\,\mathrm{min}$	$<1\,\mathrm{min}$	$<10\,\mathrm{s}$
Dimensions	$1\,\mathrm{m}$ x $1\,\mathrm{m}$ x $1\,\mathrm{m}$	”fits through door” $<0.8\,\mathrm{m}$ for smallest rib	$0.4\,\mathrm{m}$ x $0.4\,\mathrm{m}$ x $0.4\,\mathrm{m}$
Phase drift	$7.5\,{}^\circ\,\mathrm{rev}^{-1}$	$2.5\,{}^\circ\,\mathrm{rev}^{-1}$	$<2.5\,{}^\circ\,\mathrm{rev}^{-1}$
Maximum phase error	15°	5°	<5°
Colour of the light source	monochrome	-	RGBW
SH order	0-1-2	-	higher orders
Brightness	Capturable in completely dark room	-	Visible in daylight
Light levels	256 is not here; 8 ($3\,\mathrm{bit}$)	256 ($8\,\mathrm{bit}$)	16384 ($14\,\mathrm{bit}$)
Camera synchronisation	Change settings and take pictures manually	Change settings manually, use microcontroller to take pictures	Everything done by microcontroller
User interface	working UI	-	GUI

Price A requirement that does not have a must/should/could have value is price. However, the goal is to make this prototype relatively cheap, since that makes it accessible to research groups to reproduce and use for themselves.

Safety Safety constraints should also be taken into consideration. Some parts of the light bulb may be moving, which the construction should support. Also, the light bulb needs power. The system that takes power from the mains should be safe and intuitive to use in order to avoid electric shocks or a short circuit. A safety system should be included in the design, so that in case of any failure or danger, the system shuts down by itself.

Chapter 4

Pattern calculation

When using the measurement light bulb, the user will have a certain pattern in mind which is to be projected: the intended projection. To map this projection to a pattern which can be used by the device, the transfer from pattern to projection must be taken into account. Due to the physical properties of the device, the input pattern will be distorted when projecting. This part of the design will focus on how to predict this distortion in such a way that the original pattern can be altered to counteract these distortions. This will allow the user to change their ideal pattern in such a way that, when fed as an input to the device, the projection will be as close as possible to the intended projection.

The first section will go into the theory behind the different origins of pattern distortion and in what way they influence the projection. A possible solution to the problem will also be proposed, involving the necessary theory. The second section will go into the requirements of the solution, such as the allowed error and required resolution. The third section explains the implementation of the solution and the design choices that are made. The last section will take a look at the results, quantify the performance of the implementation, and relate these quantities back to the requirements.

4.1 Theory

This section will elaborate on several sources of projection distortion and propose a solution to minimise the errors introduced.

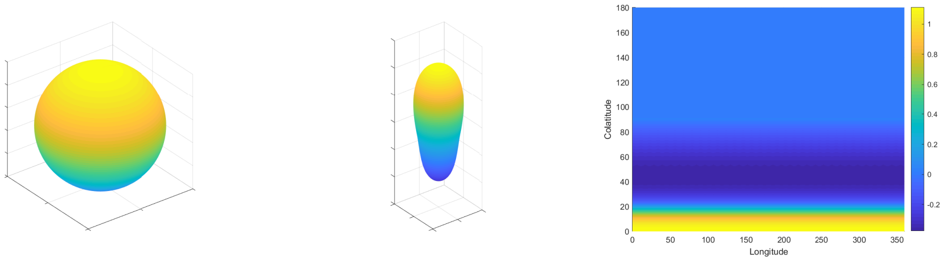
4.1.1 Sources of projection distortion

To perfectly recreate any pattern, an infinitely small light beam in combination with an infinitely small projection resolution would be required. This is physically impossible which restricts the patterns that can be projected accurately. The distortion introduced by the laser will be examined first, the distortions due to the mechanical limitations will be treated later.

Wide laser beam To project the patterns, a laser is used. Since it does not have an infinitely small dot and the entire sphere must be covered within a reasonable time, the laser light is diverged into a beam of several degrees. First of all, this puts restrictions on the frequency contents of the pattern since its brightness slope cannot be larger than that of the laser dot.

Secondly, longitudinal circles combined with a wide beam will cause overlap between different parts of the path of the beam, especially near the poles. An example of this effect can be seen in Figure 4.1. 4.1a shows the intended pattern and 4.1b what the projection would look like when it is projected by the lamp when the projection is mapped to a pattern directly. 4.1c shows the difference between the intended pattern and projection. It can be seen that the pole is too bright.

Circular symmetry The previous issue of overlap can become even harder when the intensity distribution of the laser is not circularly symmetric. Since the laser is pointed at a rotating mirror, the beam reflected off



(a) Intended pattern (b) Projection of intended pattern (c) Error with intended pattern

Figure 4.1: Effect of overlap when intended pattern is used as input pattern

the mirror will rotate as well. When the laser is circularly symmetric, this effect cannot be seen and does not pose a problem. The ideal model of a laser has a Gaussian distribution, but due to all kind of external effects this distribution can change [Bretenaker, 2015].

Round off Another distortion introduced by the laser is round off due to a limited amount of light levels of the laser. These introduced round off errors are at most half of the least significant bit of the laser intensity.

Laser driver A distortion introduced by the laser driver originates from the nonlinearity of the laser driver circuitry. This means that the laser intensity is not completely linear with respect to the drive signal. This can have several causes, but it can be counteracted if a function of this relation is available.

Stepper resolution The mechanical construction of the device also distorts the projection. This is due to the fact that the beam follows longitudinal circles, as can be seen from Figure 4.2, which are a certain distance apart, determined by the step size of the stepper motor. The pattern only gets sampled on these circles and the space between these circles is illuminated due to the wide beam which leads to a type of interpolation. The errors that these interpolations introduce should be minimised.

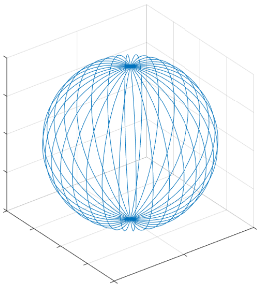


Figure 4.2: Path followed by laser beam

Microcontroller speed Another cause of distortion of the pattern comes from the microcontroller. Similar to the problem described above, where the sphere is discretised into several longitudinal circles, the limited speed of the microcontroller will cause discretisation along the circles. The distortions introduced by this effect are very small, since the microcontroller can update the laser intensity very fast compared to the width of the beam and the speed at which the laser dot moves.

Negative light A final cause of distortion comes from the way the patterns are projected. From the general equation of SHs (Equation 1.1) it can be seen that all harmonics except order 0 contain both positive and negative parts. Since negative light cannot be projected, the spherical harmonic functions are split into two patterns. One pattern contains the positive part of the SH, the other contains the absolute value of the negative part. This implies that all these patterns will consist of two separate projections.

An alternative is to shift and scale the SHs [Tunwattanapong et al., 2013, Section 4.1]. Although simply shifting and scaling the SHs will result in half the measurements, the results cannot be used. Dark spots in the image will then be seen as strongly negative light, even if the part of the room is hidden behind an object and light could never reach it.

An alternative used by Tunwattanapong et al. is to take a complementary pattern and make a second measurement. The results can then be subtracted to obtain the total measurement. This means that every pattern will again consist of two separate measurements. The entire brightness range will be comparable with the range when separating positive and negative parts and projecting them separately.

By splitting the pattern in a positive and negative part, the region where the intensity goes from positive to negative will contain very high frequency components which can not be reproduced and therefore lead to distortions of the pattern.

4.1.2 Solution to projection distortion

The most important sources of projection distortions have been examined in the previous section. The introduced distortions can be predicted and calculated. The steps of an algorithm to turn a pattern into the projection are:

1. Determine the input pattern of the system.
2. Draw a path over a sphere consisting of longitudinal circles to model the mechanical limitations of the device.
3. Evaluate the input pattern over this line. Laser driver nonlinearity can also be introduced at this point.
4. Perform a spherical convolution of the previously calculated path with the light intensity distribution of the laser beam to model the wide laser beam.

The overlap, wide beam, potential asymmetrical intensity distribution and drive limitations of the laser are all modelled this way. The limitations of the mechanical construction are also modelled by taking longitudinal circles with a certain distance between them.

The described algorithm can calculate the observed projection for some input pattern. However, we are looking for the inverse solution: we want the input pattern to be modified such that the projected pattern looks as the intended one. One solution is to directly reverse the algorithm, but this is hard to do since sampling introduces loss of data and spherical deconvolutions are hard to do.

Our proposed solution takes a gradient descent optimisation approach where the intended projection can be provided and through an iterative approach, the input pattern will be optimised, as will be explained in Section 4.3 paragraph ”Learning method”.

4.2 Requirements for the pattern calculation

The program should optimise the intended projection entered by the user in such a way that the projection of the final pattern resembles the projection given. To quantify the performance of the program, several new requirements are added based on the requirements of the whole system.

Light intensity error The most important requirement is the intensity final error bound that can be achieved. Ideally, it should be equal to or better than half of the least significant bit (LSB) of the amount of light levels. This would mean that the actual value is rounded off which is the best possible with a limited amount of light levels. However, due to all the different distortions it is unlikely that an error this small can be obtained since all of their errors add up.

Angular resolution Another important requirement is the angular resolution which can be used. Since the angular resolution of the device should be at most 5° (Table 3.1), the laser dot will have about this resolution. The simulation will have to be done at a lot higher resolution to obtain realistic results and include the intensity distribution of the laser dot.

Run time Another requirement involves the run time of the algorithm. This is not an important requirement since the algorithm can run on a regular computer and every pattern needs to be calculated only once. The run time should be within reasonable length however and not take up multiple days for a single pattern.

To give some quantifiable meaning to the requirements mentioned above, Table 4.1 is made with several values which are assumed to be reasonable. The light intensity error is defined with respect to 256 light levels (8 bit), which was chosen because an 8 bit value is the fundamental data size for a controller and still allows for a high enough laser PWM frequency.

Table 4.1: Requirements related to the pattern calculation algorithm

Type	Must have	Should have	Could have
Light intensity error	8 LSB	2 LSB	< 0.5 LSB
Resolution	2°	0.5°	$< 0.5°$
Run time	1 day	3 hours	< 1 hour

4.3 Implementation

The solution to this problem is a gradient descent algorithm implemented in MATLAB which can determine the optimal input pattern to achieve a certain projection. A problem of this simulation is that the sphere is discretised while the movement of the device is continuous. This can be approximated by taking a high resolution, although, it turns out that this solution falls short, as will be explained in Section 4.3.1.

The script consists of a learning loop where the input pattern is used to calculate the projection. The projection is compared with the intended projection and the error is determined. This error can then be referred back to the input pattern and its values are slightly altered to decrease the error. Some parts of this loop will be examined in further detail.

Learning method To refer the error back to the input pattern, a second spherical convolution is done with the laser intensity distribution. This indicates how much of the error around a certain point is due to an error at that point. This also makes sure that high frequency errors get averaged out and do not influence the pattern. Only if the average projection intensity is too large or too small over a significant part of the projection area, then the value of the input pattern at that point is altered. If the error is positive, meaning the projection is brighter than it should be, the value of the input pattern is decreased. If the error is negative, the value of the input pattern is increased.

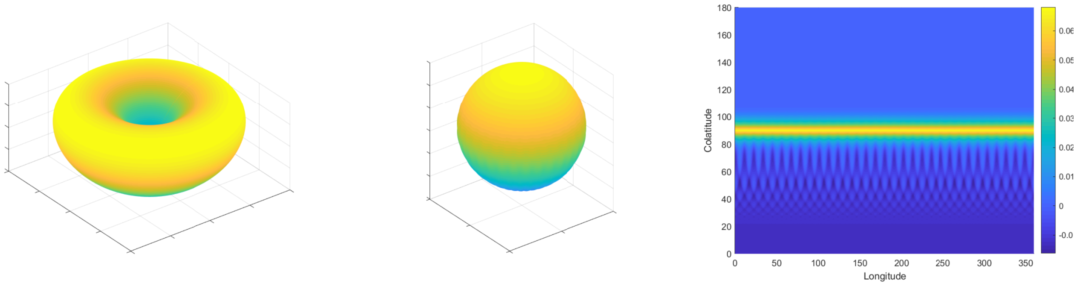
Convolution optimisation Calculating the projection and referring the error back to the input pattern both involve spherical convolutions. These are the most computationally expensive part of the entire program. MATLAB does not have built in functions which can do these convolutions, so a custom implementation is made. To get the convolutions to execute fast, custom CUDA code (Appendix C) is used which brings single iteration loop times back from several minutes to seconds. This allows us to execute a lot more iterations within the same amount of time.

Using GPUs for the calculation speeds up the process, if a compatible GPU is available. To speed up the calculations even more, the programs are run on the HPC cluster. This allows high end GPUs to be used and the different patterns can be split up among different jobs which can run in parallel and overnight.

4.3.1 Issues

Implementing the program revealed some issues about the algorithm. Testing shows that the run time complexity of the convolution algorithm lays somewhere between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ where n is proportional to the resolution. This puts a severe upper bound on the highest resolution that can be used within reasonable time.

To speed up the convergence time of the algorithm, an initialisation pattern is provided which the algorithm then optimises. We use the intended pattern which is multiplied with the sine of the polar angle at each point. This gives a good approximation of how the overlap distortion can be countered. This takes work away from the optimisation algorithm which can then only focus on countering the less predictable distortion factors, which will decrease the amount of iterations needed to run the algorithm.



(a) Initialisation pattern (b) Projection of initialisation pattern (c) Error with initialisation pattern

Figure 4.3: Effect of overlap after the sine of the colatitude has been multiplied with the intended pattern

The effect of this initialisation pattern can be seen in Figure 4.3 which uses the same intended pattern as the example from Figure 4.1. 4.3a shows the result of applying the multiplication with the sine of the colatitude on the input pattern. 4.3b shows what the projection would look like when the initialisation pattern is projected by the device. 4.3c shows the difference between the intended and projected pattern. The largest absolute error is already $16\times$ smaller than the largest absolute error seen in Figure 4.1c. The largest error can be observed at the equator where the value of the SH would go from positive to negative, as described in ”Negative light” in Section 4.1.1.

4.4 Results and discussion

The optimisation algorithm was developed in such a way that it finds a satisfying input pattern relatively quickly. This took about a thousand iterations, which on the HPC cluster takes about 25 minutes per projection. Since this cluster allowed these jobs to be run in parallel, this took a little less than an hour, which meets the requirements.

In Table 4.2, the calculated absolute maximum error and standard deviation for each projection are depicted. It can be seen that the overall absolute maximum error is less than 2% and the overall maximum standard deviation is less than 0.8%. It can be seen that positive projections typically have the same size errors as their negative counterparts. The same holds for negative and positive indices of the same order.

The resulting errors are in line with the projection distortion requirements and will yield SHs projections that satisfy the criteria of the project. However, the circular asymmetry of the laser and non-linearities in its driver have not been modelled, as these measurements were not finished in time. In a future version, these errors could be accounted for if the distribution of the laser profile and the relation between input PWM and output laser light intensity are known. This could remove the unknown error currently caused by these sources.

Table 4.2: Overview of absolute maximum error and standard deviation for all the projections

SH order	index	+/- part of projection	Abs max error	Std dev
0	0	n.a.	0.019284	0.007690
1	-1	+	0.018856	0.003555
	-1	-	0.018856	0.003555
	0	+	0.005835	0.001640
	0	-	0.005836	0.001640
	1	+	0.018855	0.003555
	1	-	0.018855	0.003555
2	-2	+	0.018862	0.003452
	-2	-	0.018862	0.003452
	-1	+	0.007729	0.001807
	-1	-	0.007730	0.001807
	0	+	0.004239	0.001019
	0	-	0.009413	0.003180
	1	+	0.007729	0.001807
	1	-	0.007729	0.001806
	2	+	0.018861	0.003452
	2	-	0.018862	0.003452

Chapter 5

Wireless communication

The design of the device allows it to project onto an entire sphere. Since the device consists of several spinning objects, it is not very safe to operate it using simple buttons mounted on the construction. Furthermore, since pattern data is not stored locally, but on the controlling user device, this data needs to be transmitted to the lamp.

These issues are combated with a wireless communication channel. This allows the user to control and send pattern data to the device from a safe distance and to read out diagnostic information. The functionality of the interface could be easily extended using software without altering any of the hardware.

This chapter will elaborate on the design choices and implementation of the wireless connection. First, the requirements of the connection, the problems it is supposed to solve and several design choices will be considered. A more in-depth look into the hardware choices will be discussed next, followed by the software used. The chapter will end with an overview of shortcomings and possible solutions to improve the system.

5.1 Requirements for the wireless communication

As mentioned in the introduction to this section, operating the device with physical buttons on the device is a dangerous choice due to the spinning hardware and the laser with which eye contact is to be avoided. Especially when something goes wrong physically, when for example a piece of the construction breaks and the device does not shut down by itself or the user has entered wrong settings and would like to start again, coming close to the device is dangerous. This main problem we intend to solve via a wireless connection.

Furthermore, pattern data cannot be stored on the lamp locally, which means this data needs to be transmitted to it for every projection. This is also needs to be done via wireless communication.

Besides safety and pattern transmission being the most important requirements, there are more requirements to this part of the system. These are listed in Table 5.1.

5.2 Implementation

This section will first discuss the choice of the wireless protocol, followed by its hardware implementation. The section ends with an overview of the software interface used for implementing the wireless communication.

5.2.1 Wireless protocol

For the choice of the wireless protocol, first some options are listed with some considerations, after which one is chosen.

Table 5.1: Wireless connection requirements

Requirement	Explanation
Safety	Operating the moving measurement light bulb using physical buttons is unsafe, which the wireless connection solves. Loss of connection should also be handled swiftly and safely.
Pattern transmission	Pattern data cannot be stored on the lamp itself, so it should be transmitted to it wirelessly.
Easy use	Setting up the connection and using it should be easy and should not take more than a couple of seconds.
Short range	While operating the lamp, the user should always keep an eye on the it in case of any unexpected behaviour. Therefore, the range does not have to exceed the size of an average room which is assumed to be 5 m to 10 m.
Low-cost	The price of the components of the lamp should not be too high since it should be cheap to reproduce.

Options

The list of requirements from Table 5.1 rules out any custom made wireless communication hardware and protocols, since that would violate the ”Easy use” and ”Low-cost” requirements. A commercially available, widely used and well documented wireless solution is therefore required, three of which are listed in the next section with some considerations

Wi-Fi Wi-Fi is used in nearly all smartphones and computers. To address the safety issue regarding loss of connection, the lamp should act as an access point, which means that the connected phone or computer cannot be connected to anything else simultaneously. An alternative would be to connect the lamp to a network together with the controlling device, but then there would need to be a way for the lamp to find the right Wi-Fi network, which increases complexity.

Bluetooth The range and data speeds of Bluetooth are lower than those of Wi-Fi, but still within the requirements. Most smartphones and increasingly computers have a Bluetooth adapter which allows the user to use such a device without having to disconnect from the internet. The safety issue of detecting connection loss is less of a problem since Bluetooth devices are always paired and loss of a paired connection is easy to detect.

Zigbee Zigbee looks a lot like Bluetooth regarding the way paired connections are set up, but with a more robust protocol and higher range. However, these devices are slightly more expensive and would require an external remote unlike Wi-Fi and Bluetooth, where a smartphone or computer can be used as a remote.

Choice

Regarding the considerations of the different wireless technologies listed above, Bluetooth is chosen to fit the requirements the best. It can be used very safe including detection of loss of connection. It is very easy to apply since a smartphone or laptop can be used as a remote if an application is written for it. The usual operating range of Bluetooth is around 10 m which is enough for our application and there are several popular Bluetooth modules available at a low price which are easy to use.

5.2.2 Hardware

This section will explain the choices regarding the hardware required to establish the Bluetooth connection and use it. First, the used module will be motivated followed by the external circuitry required to employ it safely. The software interface which uses the connection will be explained in the next section.

Bluetooth module: HC-06

Since one of the requirements states that the wireless link should be low-cost and easy to acquire, a popular Bluetooth module is chosen. Two of the most popular Bluetooth modules are the HC-05 and HC-06. These two modules look similar from the outside because they use the same hardware but the HC-05 can be configured to work both as a master and slave, meaning that it can both accept and initiate connections. This is not needed and makes using it slightly more involved. The HC-06 can only be used as a slave, meaning the user can connect to it but the module can not initiate any connections. Therefore, the HC-06 is the Bluetooth module of choice.

The module contains an LED output signal to which an LED can be connected which blinks if there is no Bluetooth device paired and stays on when the device is paired. To determine whether the device is paired, the LED output signal goes to some external circuitry which detects whether it is on (paired) or pulsing (awaiting connection).

External circuitry

The main part of external circuitry used is to determine whether the Bluetooth module is paired with an external device or not. It should be able to differentiate between a pulsing and continuously high input and output a binary voltage level to the microcontroller. This saves computational power which would otherwise be needed to monitor the signal to determine whether it is on or pulsing. Monitoring an almost always constant value is a lot easier using an interrupt which does not interfere with any other activities of the microcontroller.

This external circuit can be seen in Figure 5.1. When the LED signal is high, the capacitor gets charged. This will take some time, determined by the values of the capacitor and resistor. When the LED signal is low, the capacitor gets discharged without the resistor limiting the current flow.

To determine the values of the resistors and capacitor, the frequency of the LED is important, which is about 5 Hz. The RC time should well exceed the period of the LED to make sure that false positives do not occur. With a resistance of $10\,\mathrm{k\Omega}$ and capacitance of $47\,\mu\mathrm{F}$, the RC time is calculated to be $0.47\,\mathrm{s}$ which should be enough. The capacitor voltage is then compared to $50\,\%$ of the source voltage. The output is low when disconnected and becomes high when it connects. If the connection is lost, the capacitor will be completely discharged directly after the first falling edge. The comparator output signal becomes low again indicating loss of connection very fast.

Simulations of this circuit can be found in Appendix B.

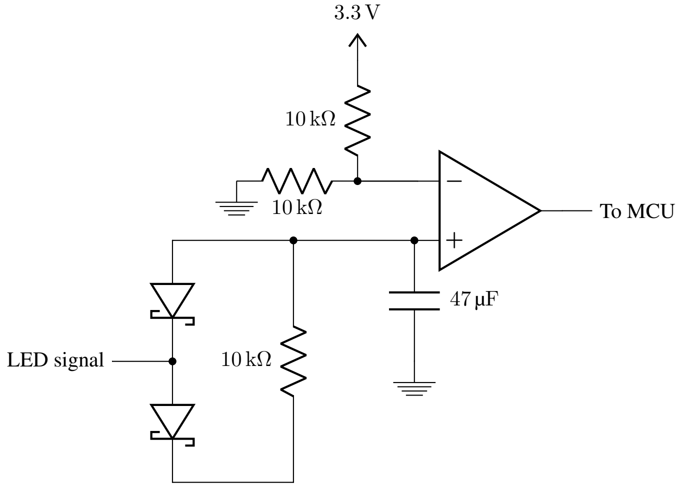


Figure 5.1: External circuitry to translate LED signal to a binary signal.

The next section will focus on the software required on both the microcontroller and external device which is used as a remote. It will also talk about the high level protocol, what messages can be sent to the device and what messages will be returned by the device.

5.2.3 Software

The second part to the wireless connection is the software on the measurement light bulb and the controlling device. The software on the controller of the lamp is written in C and further elaborated upon in Section 6.3.3. The software on the controlling device is written in Python. This program has a command line interface which allows the user to control the lamp. After connecting the controlling device to the lamp, one of four commands can be used to control it. These commands, together with the character that is sent and the meaning of the command, can be seen in Table 5.2.

Table 5.2: Overview of commands from controlling device to measurement light bulb

Command	Character	Meaning
Transmit pattern	`p`	be ready to receive pattern data
Go	`g`	start projecting currently loaded pattern
Stop	`s`	stop projecting and shut down motors
Diagnostics	`d`	send diagnostics to controlling device

After the 'Transmit pattern' command is sent, the first 8 characters of the pattern name are transmitted, followed by $80.4\,\mathrm{kB}$ worth of pattern data. The pattern name allows the user to see which pattern is loaded. The diagnostics request will result in a response from the measurement light bulb, which contains the motor speed and name of the pattern which is currently loaded.

5.3 Results and discussion

The described system has been implemented and tested and provides a basic but functional user interface. The connection detection circuit works as intended, just like the Bluetooth connection with the controlling device. Only the command concerning the transmission of the pattern does not entirely work as intended. This is due to a limitation in the microcontroller, which leads to the pattern having to be divided across multiple transmissions. This is not a problem in itself, but requires the implementation to change and is currently still work in progress.

The software does show some limitations, especially on the controlling device. Loss of connection is not detected and the application is single threaded, resulting in blocking behaviour. Future work could involve improving the software on the controlling device, since this has not gotten a lot of attention since it is related more to high level software engineering.

Chapter 6

System integration

The system consists of a couple of subsystems, which all need to be bound together and controlled in order for the whole system to function properly. This part of the design will focus on how to bind all these parts together on the moving construction, keeping weight and size in mind as not to put too much strain on the motors.

The first section will go into the theory of the subsystems, their control and also propose a possible implementation with some background information. The second section states the requirements of this implementation. The third section explains the implementation of this system integration and the choices that were made in this design. Finally, the fourth section goes over the results of the implementation, which links back to the requirements in the second section.

6.1 Theory

Like stated in the introduction, there are some subsystems which need to be controlled by an overlaying controller. These subsystems are the following: two motor drivers, a laser driver and a Bluetooth module. All these circuits and a circuit for managing power are placed upon the moving part of the construction, except for one of the motor drivers.

An overview of the different subsystems, what they do and where they are located on the lamp is given first. After that, a method which integrates these subsystems will be explained.

6.1.1 Overview of the subsystems

The motor driver for Motor B is not placed upon the moving part, but near the motor itself and has two inputs: a direction input which determines in which the direction it spins and a step input, which makes the motor rotate a step for each pulse sent to it.

The other motor driver is for driving Motor M and is placed upon the moving construction. This driver has a PWM input which dictates the speed.

The laser driver is also placed upon the moving part and takes a PWM signal as input.

The Bluetooth module is for connecting the measurement light bulb to the user's controlling device, which allows the user to control the lamp. It also enables transmission of pattern data from the controlling device to the lamp.

The on-board power supply consists of some voltage regulators, which produce the voltages required for all the different subsystems. It has a 12 V DC input. The laser driver circuit uses 12 V straight from the DC input, the driver for Motor M 7.2 V and the microcontroller and Bluetooth module 3.3 V.

6.1.2 Solution

A possible small and light implementation of this complete circuit is a printed circuit board or PCB. This PCB contains all the circuitry for the subsystems and as a controller, it contains a microcontroller.

However, when designing a PCB, some factors should be accounted for, to make sure that all the circuits function accordingly. For example, paths on the PCB should not be too long for some components, or not too thin for certain current levels.

Several guidelines were found on the internet, which provided some insight in basic PCB design techniques [Electronics Notes; Eurocircuits; STMicroelectronics, 2017, 2018; Texas Instrument, 2013].

After the PCB is built, the controller can be programmed. For this, a piece of software needs to be written. This implements all the required features for the device to function correctly.

Lastly, user control is implemented using a computer application. As has been explained in Chapter 5, this application connects to the system via Bluetooth and can send commands and pattern data to it.

6.2 Requirements for the system integration

This section will first list the requirements for the PCB. After this, some requirements for the microcontroller are stated. The next section about the implementation will use these requirements to explain several design choices.

The PCB should have the right size for the construction, which means it needs to stay within the boundaries of the lower layer of the construction. The dimensions for this are $5.3\,\mathrm{cm}$ x $20\,\mathrm{cm}$. One of the cheapest vendor for custom made PCBs is JLCPCB and this will be used for producing the design. At JLCPCB, the price is lower if it stays between $10\,\mathrm{cm}$ x $10\,\mathrm{cm}$, which is not strictly required, but would lower the price of the total device, which is a goal.

The chosen microcontroller should be able to have enough computational power to keep up with all the subsystems. This means the following:

- $8\,\mathrm{bit}$ resolution PWM signal at a high frequency to the laser driver, which allows for a smooth projection. Its duty cycle should be updated several times per revolution to realise the required angular resolution, as has been explained in Section 4.1.1 ”Microcontroller speed”.
- $16\,\mathrm{bit}$ resolution PWM signal at a frequency which needs to be at least twice as high as the motor frequency to the driver for Motor M.
- At least $80\,\mathrm{kB}$ of RAM. Patterns are projected at a resolution of $0.9°$, which results in a matrix of 200x400 with each entry being one byte for $8\,\mathrm{bit}$ light levels, which has a size of $80\,\mathrm{kB}$.

The software should allow the user to project any of the spherical harmonics functions of the first three bands intuitively. It should also implement some safety features, to make sure the device cannot hurt itself or its user.

6.3 Implementation

In this section, the implementation of this integrated system is discussed. First, the PCB design is explained, followed by the choice of the microcontroller. Lastly, the software for the controller is discussed.

6.3.1 PCB design

The final PCB design can be seen in Figure 6.1. A 3D rendered picture of this can be seen in Figure 6.2 with the subsystems indicated. This design was made using KiCad [KiCad Developers Team, 2019], which is an open-source software suite for Electronic Design Automation.

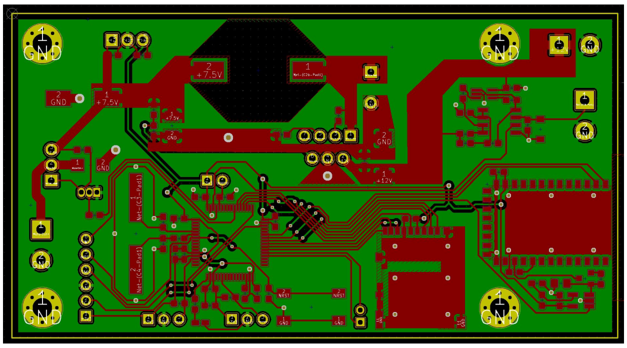


Figure 6.1: Picture of the PCB design

The final design stays within the boundaries for the cheaper option at JLCPCB of 10 cm x 10 cm and contains two copper layers. It features all the mentioned subsystems: the microcontroller, one motor driver, a laser driver, a Bluetooth module and a power supply circuit. It also contains an SD card reader, which was originally planned to store the pattern data. However, since we did not get this interface to function, it was decided that pattern data would be transmitted to the system via Bluetooth from the user device.

Paths that need to transport a lot of current are made wide enough for this capacity, as can be seen in Figure 6.1. The green layer in Figure 6.1 is the ground plane. This complete bottom layer, with some small exceptions, is a copper ground layer. This reduces interference through ground loops and avoids crosstalk between adjacent traces.

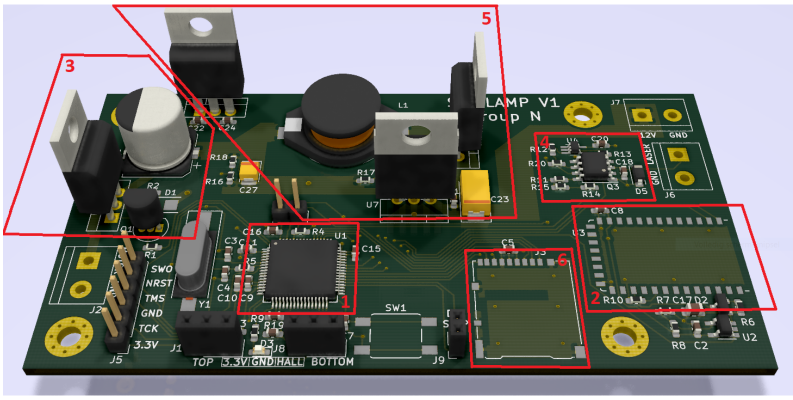


Figure 6.2: Raytraced picture of the PCB design. Subsystems are indicated with numbers. 1: microcontroller, 2: Bluetooth module, 3: Motor M driver, 4: laser driver, 5: power supply, 6: SD card reader

6.3.2 Microcontroller

This section will discuss the choice of the microcontroller. First some options are listed with some important properties, after which one is chosen for the final design.

Options

For the microcontroller, several options were considered. These are listed below.

Arduino Arduinos are widely used as simple and cheap microcontrollers for all kinds of applications. This is an advantage, since there is a lot of documentation available and wide support for all kind of applications. However, the processing power is limited and the available amount of memory (2 kB of RAM and 32 kB of flash in case of the ATmega328P) is quite small.

Raspberry Pi Raspberry Pis are also widely used and cheap, but compared the Arduino and STM32, the main difference is that they run an operating system. This makes them less ideal for real-time applications, which is the case with this project.

STM32 STM32 microcontroller are also popular and a lot of different versions are available. They are cheap, fast and very well documented. They do not run an operating system and have more memory than Arduinos.

Choice

The choice for the microcontroller is a STM32F401. This has been chosen because of its high clock speed (84 MHz), relatively large memory (96 kB of RAM and 512 kB of flash) and low price (around €7). It also has more than enough timers to produce all the required PWM signals.

6.3.3 Software

The software for the microcontroller is written using STM32CubeMX, the IDE supplied for STM32 microcontrollers [STMicroelectornics]. The software for the control application is written in Python [Python Software Foundation].

A simplified state diagram can be seen in Figure 6.3. This diagram does not include all the possible transitions and actions, but is for illustrative purposes.

First, the user needs to connect to the device via Bluetooth. After this, a Python program can be used to select which SH is to be projected and send the data of this projection to the measurement light bulb. The lamp stores this data in its memory and waits for a command to start up its motors. If the motors run at the correct speed, it starts its projection. When the projection is finished, the system waits for a command to either stop and turn the motors off, select a new pattern, or to start projecting the current pattern.

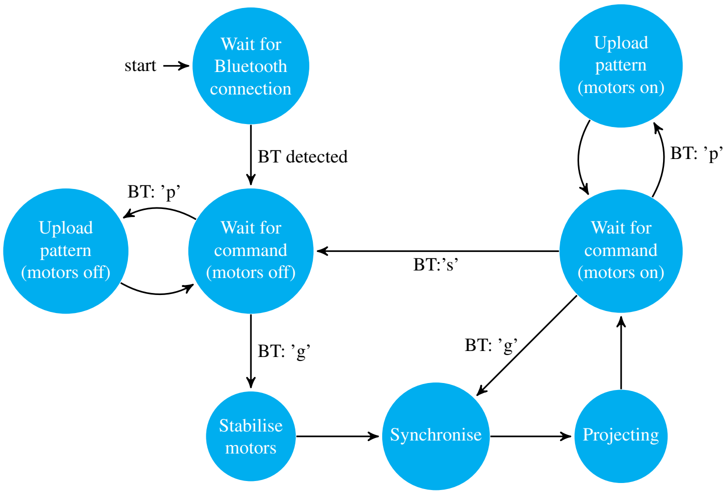


Figure 6.3: Simplified state diagram implemented in software

For safety reasons, if the time it takes for Motor M to complete one rotation changes by more than $0.5\,\%$ ($25\,\mu$s) between consecutive rotations, the motors are turned off. This means that if anything gets caught in the construction, someone touches it or a part falls off, damage is minimised. This percentage was tuned to a safe value by trial-and-error.

6.4 Results and discussion

All the functions of the constructed PCB and the controller software were found to be working. The PCB stayed within the dimensions for the cheaper option and the chosen microcontroller contains all the required features and is fast enough to keep up with the speed of the system. The PWM signal for the laser driver is a $16\,\mathrm{bit}$ signal at approximately $330\,\mathrm{Hz}$ and for Motor M this is an $8\,\mathrm{bit}$ signal at approximately $1.3\,\mathrm{kHz}$. Lastly, the software allows the user to choose any of the spherical harmonics functions up to order two and have them projected. It also contains some safety features to protect it from damage.

Initially, an SD card reader was included for saving the pattern data locally instead, but this did not function due to driver issues. This problem was solved by transmitting them via Bluetooth and saving them temporarily in the RAM of the microcontroller before projecting. However, the card reader is fully connected to the microcontroller, so a programmer more experienced with SD card communication could reprogram the microcontroller to work with an SD card.

Chapter 7

Prototype implementation and validation results

This section will discuss the last steps in the building of the measurement light bulb. First, the fabrication of the prototype will be discussed, followed by the test results of the complete system.

7.1 Fabrication of the prototype

The final prototype consists of all subsystems as shown in Figure 2.3. The subsystems, separately designed by the subgroups, are assembled together to form the prototype.

The laser, implemented in the final prototype, has an opening angle of $4°$ and is focused such that the focal point coincides with the mirror. By using this angular resolution, the spherical harmonic can be projected accurately while still limiting the time needed for each measurement. The construction has two axes over which it can rotate, which allows the laser to project onto a sphere around itself. The measurement light bulb acts as a point source, due to the fact the focal point coincides with the mirror, from which the spherical harmonic is projected into the room.

Motor M spins at $200\,\mathrm{Hz}$, while Motor B steps at $200\,\mathrm{Hz}$ with steps of $0.9°$, thus 400 steps for one rotation. This means that during the projection of one pattern, Motor B rotates $180°$, which corresponds with 200 steps. Therefore the projection of one pattern takes $1\,\mathrm{s}$. The microcontroller controls the speed of the two motors and synchronises this with the intensity of the laser to create a pattern. The synchronisation with the camera is not yet implemented, so the camera is controlled and set up manually.

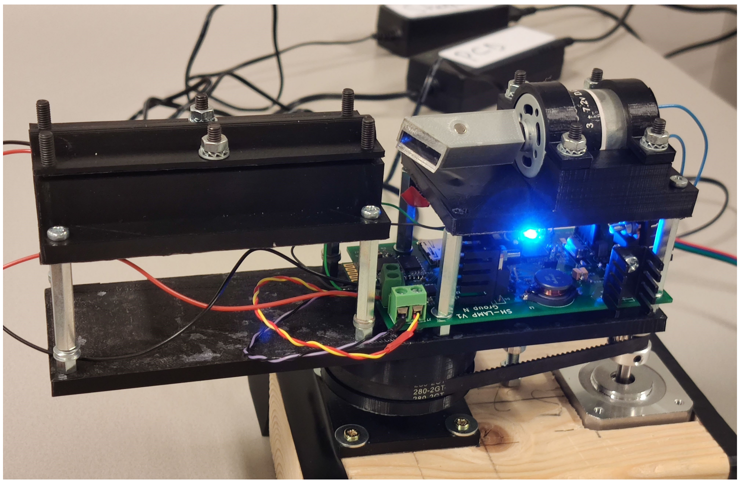


Figure 7.1: A photograph of the prototype

7.2 Results of complete system tests

The goal of this project was to build a measurement light bulb which can project spherical harmonics and to synchronise it with a camera so that the impact of the illumination can be captured. Figure 7.3 shows the result of a projection of the measurement light bulb. The photograph is taken in a dark environment to eliminate as much ambient light as possible, which is required since the laser is relatively low power. The exposure time for this photograph was 8 s. The pattern that is projected is the positive part of the spherical harmonic function of order 2 and index -2, of which a plot is depicted in Figure 7.2.

However, it still contains some errors. The positioning of the Hall effect switch for Motor M causes some offset in latitudinal direction, which should be corrected for. Also, a bug in the projection algorithm causes the spherical harmonic to be projected in two different orientations: once normally, and once rotated 180° over the longitudinal axis. These two errors combined make it that the projection is actually two projections of the same spherical harmonic oriented in slightly different ways.

Also, since the Hall effect switch for Motor B is not yet placed, the measurement light bulb is not yet able to orient itself longitudinally.

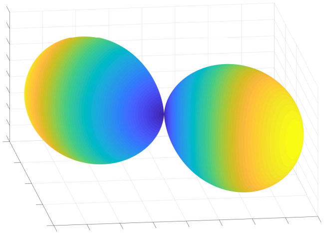


Figure 7.2: Plot of the positive part of the spherical harmonic function of order 2, index -2

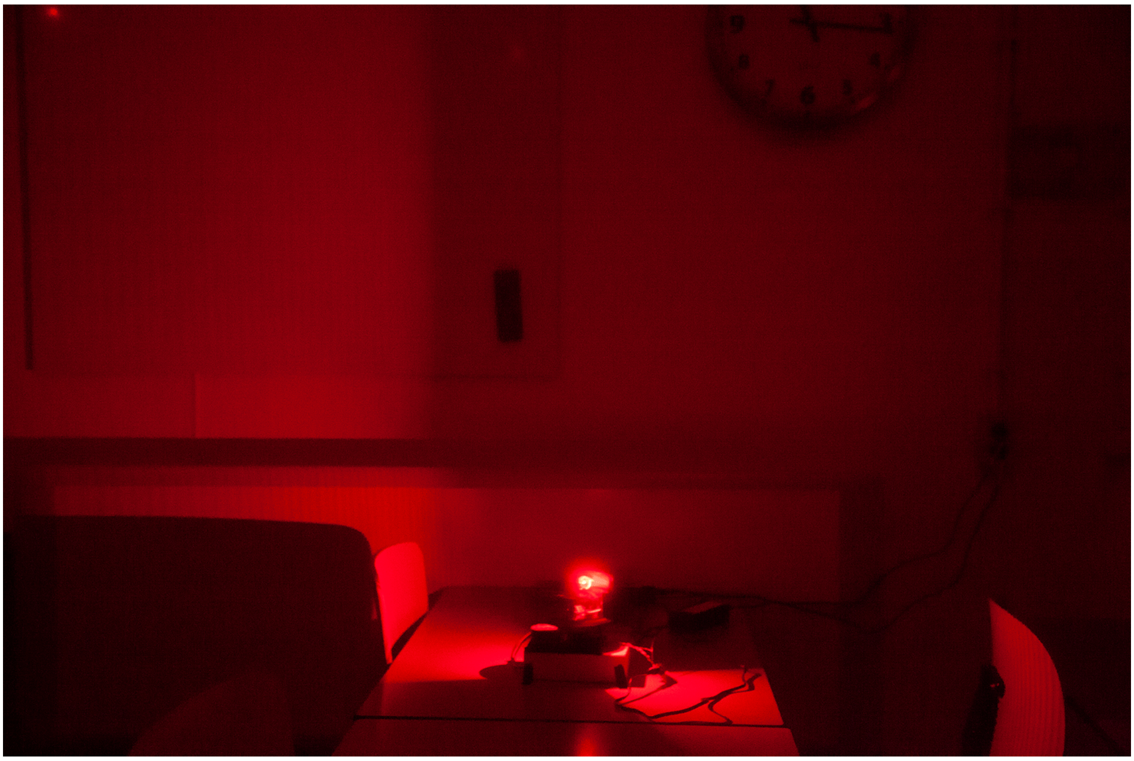


Figure 7.3: A photograph of the first spherical harmonics projection

Chapter 8

Discussion of results

In this section the results of the measurement light bulb are compared with the system requirements, stated in Chapter 3. Table 8.1 shows an overview of the system requirements and the results that are achieved.

Table 8.1: Overview of complete system requirements

Type	Must have	Should have	Could have	Result
Angular resolution	$15°$	$5°$	$<5°$	$4°$
Projection time	”converges”	$1\,\mathrm{min\,pattern^{-1}}$	$1\,\mathrm{s\,pattern^{-1}}$	$1\,\mathrm{s\,pattern^{-1}}$
Power	$230\,\mathrm{V}$, $16\,\mathrm{A}$	-	battery powered	$230\,\mathrm{V}$, $16\,\mathrm{A}$
Start-up time	$<15\,\mathrm{min}$	$<1\,\mathrm{min}$	$<10\,\mathrm{s}$	$1\,\mathrm{min}$
Dimensions	$1\,\mathrm{m}\times1\,\mathrm{m}\times1\,\mathrm{m}$	”fits through door” $<0.8\,\mathrm{m}$ for smallest rib	$0.4\,\mathrm{m}\times0.4\,\mathrm{m}\times0.4\,\mathrm{m}$	$0.24\,\mathrm{m}\times0.24\,\mathrm{m}\times0.20\,\mathrm{m}$
Phase drift	$7.5\,°\,\mathrm{rev}^{-1}$	$2.5\,°\,\mathrm{rev}^{-1}$	$<2.5\,°\,\mathrm{rev}^{-1}$	$0\,°\,\mathrm{rev}^{-1}$
Maximum phase error	$5°$	$5°$	$<5°$	$1.044°$
Colour of the light source	monochrome	-	RGBW	monochrome
SH order	0-1-2	-	higher orders	None
Brightness	Capturable in completely dark room	-	Visible in daylight	Capturable in completely dark room
Light levels	8 (3 bit)	256 (8 bit)	16384 (14 bit)	256 (8 bit)
Camera synchronisation	Change settings and take pictures manually	Change settings manually, use microcontroller to take pictures	Everything done by microcontroller	Change settings and take pictures manually
User interface	working UI	-	GUI	working UI

Angular resolution The final angular resolution of the system is equal to $4°$ and therefore the ’Could have’ requirement is reached.

Projection time Since Motor B has a frequency of $0.5\,\mathrm{Hz}$ and one projection only needs a $180°$ rotation, the projection of one pattern takes $1\,\mathrm{s}$. Thus the and therefore the ’Could have’ requirement is reached.

Power The system is powered using the mains. Therefore the ’Must have’ requirement is reached. In order for the system to be powered by one or multiple batteries, the power supply should be designed differently to be compatible with the battery instead of with the mains.

Start-up time The device can start a measurement within one minute of switching the device on. Therefore the 'Should have' requirement is fulfilled. A faster start-up time would require a faster data transmission when a pattern is uploaded to the device, since this is the bottle neck for the start-up time.

Dimensions The measurement light bulb has the dimensions $0.24\,\mathrm{mx}0.24\,\mathrm{mx}0.20\,\mathrm{m}$. Therefore the 'Could have' requirement is reached.

Phase drift The phase drift of the pattern is equal to zero, since the pattern is reset each time Motor M finishes one revolution. Thus, the 'Could have' requirement is reached.

Maximum phase error Motor M is the main cause of the maximum phase error, in colatitudinal direction. The maximum speed deviation of Motor M is $0.29\,\%$, with respect to the steady-state speed. This corresponds to a maximum phase error of 1.044° in one revolution. Therefore the 'Could have' requirement on phase error is reached (phase error $< 5^\circ$).

Colour The colour of the projection is red monochrome light with a wavelength of $650\,\mathrm{nm}$. Therefore the 'Must have' requirement is reached. For this project, a monochrome light source is enough to create satisfying results and give a proof of concept.

SH order The software has currently been developed for spherical harmonics of orders 0, 1 and 2. However, due to some bugs in the software, only one of these can be projected. At the moment this is the function of order 2 and index -2.

Also, some unfinished work relating to the Hall switches give rise to some synchronisation issues, which cause double projections at different orientations.

When these are fixed, the device will be able to project all the specified spherical harmonics correctly. When this has been done, the 'must have' requirement can be fulfilled.

Brightness The prototype is able to capture patterns in a completely dark room. Therefore the 'Must have' requirement is reached. To reach the 'Could have' category, the light source should have a higher light intensity than the ambient light. Then, a camera can capture the pattern in daylight.

Light levels The driver and the microcontroller together allow for 256 light levels ($8\,\mathrm{bit}$), thus the 'Should have' requirement is fulfilled.

Camera synchronisation The microcontroller can synchronise the pattern and the camera to take a photograph. The camera settings are set manually by the user. Thus the 'Should have' requirement is reached. In order for the microcontroller to also control the camera settings, an extra (wireless) connection between the microcontroller and the camera is needed.

User interface A user interface has been implemented via a command line interface using a Python program. It allows the user to control the device intuitively using only four different commands. This means that the 'Must have' requirement is reached.

Price The material costs for the measurement light bulb are roughly €150.

Safety The output power of the used laser is within the TU Delft regulations and is therefore considered safe. The prototype can be controlled wirelessly which eliminates the need of being close to the moving parts of the device when performing measurements. Also, a safety measure has been implemented to make sure that the motors and the laser shut down if something gets stuck in them or a part falls of.

Chapter 9

Conclusion and future work

9.1 Conclusion

The goal of this project was to deliver a proof of concept for the measurement light bulb. The final prototype is able to project a spherical harmonic onto its surroundings. However, the prototype only works in a low light environment. In environments with a higher light intensity, the camera is not able to distinguish the ambient light from the the projected light. Patterns can be uploaded to the prototype using a Bluetooth connection. Also, the control signals, like a starting and stopping signal, are managed by Bluetooth. Although not implemented yet, the prototype could support camera synchronisation, to capture the illumination impact of the measurement light bulb.

9.2 Future work

All requirements for the prototype were reached to some extent (Must have/Should have/Could have), yet there are still things to improve on for future versions of the measurement light bulb. This prototype is only used for a proof of concept. A future design for a measurement light bulb could improve on individual subsystems as well as the system as a whole.

Beam profile The current beam profile has a quasi-Gaussian distribution. A future version of the measurement light bulb could contain a laser with a more accurate Gaussian distributed profile or a circularly symmetric beam profile. This will reduce errors in the projections or omit the need for complicated compensation algorithms for this problem.

Light output intensity The current prototype is only able to capture spherical harmonic projections in a completely dark room. A future model could contain a light source with a higher power output. This would enable the measurement light bulb to perform measurements in lit rooms as well as dark rooms. However, this would require the involvement of a laser safety officer.

Spectrum of the light source The current prototype of the measurement light bulb can only emit red light with a wavelength of $650\,\mathrm{nm}$. A future design of the measurement light bulb could improve on this by using a light source that covers the complete visible light spectrum or even a light source that can cover certain parts of the spectrum upon request.

Linearity of light source As seen in tests, the light intensity as captured with the camera is not completely linear with the PWM signal that was fed to the laser driver. A future model of the measurement light bulb could contain a linear laser driver or could compensate for this problem using software.

Camera The Canon EOS 5D Mark II has two main forms of unpredictable delay. Firstly the shutter delay, which is the time between the signal that activates the camera to take a picture and the opening of the shutter. Secondly the buffer delay, which is the time between the end of one exposure and the start of the next, needed for saving the image file. To implement an accurate and efficient camera synchronisation algorithm, a camera with predictable delays is required. In addition, newer camera models often have better image sensors that have better low-light performance, which would give more accurate results.

Light levels Together with the microcontroller, the laser driver is able to output 256 different light levels (8 bit). The sensor of the camera is able to capture 16384 (14 bit) different light levels. Therefore, in a future model of the measurement light bulb, the amount of light levels could be increased up to 16384 (14 bit), equal to the sensitivity of the sensor in the camera.

Construction Although the construction is functional, a few aspects could be improved. Motor B, its corresponding and the power converters can be put into a 3D printed box, which can be mounted to the bottom of the current construction. This way there will be no loose wires or components. This box could also have a mounting point for a camera tripod. This way the system could easily be placed in many position. Another possible point of improvement is the part of the projection that is obstructed by the construction. This obstruction could be reduced if the construction is designed differently.

Start-up time The start-up time of the measurement light bulb is currently around a minute, but this could be reduced significantly if the symbol rate of the Bluetooth communication was increased. Currently, this is set at 9600 Bd, but this could be increased to 115 200 Bd or even to 1 382 400 Bd, for a approximate speed-up of 10 or 100 respectively.

Motor speed To reduce the projection time, the time it takes to capture one pattern, the motors could be driven at higher speeds. In this way more measurements could be performed within a shorter amount of time.

Higher order spherical harmonics Future work could include higher order spherical harmonics to be able to be projected. The hardware of the device is already capable of this, but the patterns for higher order have not been calculated and stored. If this were to be done however, they could be projected.

Power supply The current implementation uses two AC/DC converters. One for the stepper motor and one for the rest of the system. In a future version this could be reduced to one AC/DC, by changing the topology of the power supply. Another improvement could be the use of a battery to power the system. It is expected that a large battery pack would be needed in order to power the whole system.

User interface The user interface could be improved by transforming it into a graphical user interface. This allows for more intuitive usage.

Bibliography

F. Bretenaker. *Laser Physics*. University of Paris-Sud & École Polytechnique, Paris, 2015.

P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 145–156. ACM, 2000. doi: 10.1145/344779.344855.

Electronics Notes. PCB Design Layout Guidelines. `https://www.electronics-notes.com/articles/analogue_circuits/pcb-design/pcb-design-layout-guidelines.php`.

Eurocircuits. PCB Design Guidelines. `https://www.eurocircuits.com/pcb-design-guidelines/`.

GreatScott! Make your own POV LED globe. `https://www.youtube.com/watch?v=69G522AeRq8`, 2017. Accessed: 25-04-2019.

S. Groenendijk. POV globe project. `https://www.youtube.com/watch?v=kut4uUSUP94`, 2018. Accessed: 26-04-2019.

T. Hawkins, J. Cohen, and P. Debevec. A photometric approach to digitizing cultural artifacts. In *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage*, VAST '01, pages 333–342. ACM, 2001. doi: 10.1145/584993.585053.

M. Ibrahim. Spherical coordinates colatitude longitude clip art. `http://www.clker.com/clipart-29223.html`, 2009.

KiCad Developers Team. KiCad. `http://www.kicad-pcb.org/`, 2019.

M. Levoy, S. Rusinkiewicz, M. Ginzton, J. Ginsberg, K. Pulli, D. Koller, S. Anderson, J. Shade, B. Curless, L. Pereira, J. Davis, and D. Fulk. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of SIGGRAPH 2000*, pages 131–144. ACM, 2000. doi: 10.1145/344779.344849.

R. Luger. Basics: Spherical Harmonics Maps. `https://rodluger.github.io/starry/v0.3.0/tutorials/basics1.html`, 2018.

Python Software Foundation. Python 3.6.7. `https://www.python.org/`.

R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001. doi: 10.1145/383259.383317.

N. Z. Salamon, M. Lancelle, and E. Eisemann. Computational light painting using a virtual exposure. In *Computer Graphics Forum (Proceedings of Eurographics)*, volume 36. Eurographics, John Wiley & Sons, 2017. doi: 10.1111/cgf.13101.

STMicroelectornics. STM32CubeMX. `https://www.st.com/en/development-tools/stm32cubemx.html`.

STMicroelectronics. AN2867 Application note. `https://www.st.com/content/ccc/resource/technical/document/application_note/c6/eb/5e/11/e3/69/43/eb/CD00221665.pdf/files/CD00221665.pdf/jcr:content/translations/en.CD00221665.pdf`, 2017.

STMicroelectronics. AN4488 Application note. `https://www.st.com/content/ccc/resource/technical/document/application_note/76/f9/c8/10/8a/33/4b/f0/DM00115714.pdf/files/DM00115714.pdf/jcr:content/translations/en.DM00115714.pdf`, 2018.

Texas Instrument. AN-1229 Simple Switcher PCB Layout Guidelines. `http://www.ti.com/lit/an/snva054c/snva054c.pdf`, 2013.

B. Tunwattanapong, G. Fyffe, P. Graham, J. Busch, X. Yu, A. Ghosh, and P. Devebec. Acquiring reflectance and shape from continuous spherical harmonic illumination. In *ACM Transactions on Graphics*, volume 32. ACM, 2013. doi: 10.1145/2461912.2461944.

W. Yamada, K. Yamada, H. Manabe, and D. Ikeda. iSphere: Self-luminous spherical drone display. pages 635–643, 10 2017. doi: 10.1145/3126594.3126631.

Appendices

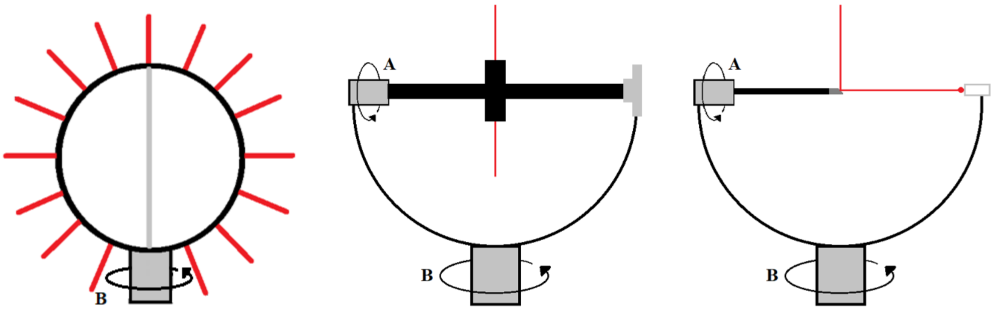
Appendix A

Design choice

In this chapter, several design options will be discussed, the option that was chosen in the end and finally what is to be delivered to achieve this design.

A.1 Design options

At the start of the project, the CGV group proposed a basic idea for an implementation of the measurement light bulb. In addition to this, two alternative implementations were thought of. These three basic implementations could be all solutions to the problem. They all work with one or multiple light sources that spin around to project onto a sphere. The reflections of this light in the room are captured with a camera with a sustained shutter time to capture one or more entire rotations. A brief explanation of the three design ideas is given below, of which the corresponding design impressions can be seen in Figure A.1.



(a) Design 1 (b) Design 2 (c) Design 3

Figure A.1: Sketches of basic design options. Red lines indicate lasers, grey rectangles indicate motors, the letters M and B describe the axes of rotation and the black arrows indicate the direction of this rotation.

Design 1 This design consist of a ring with light sources, which can all be controlled individually at high speed. The ring is rotated around axis B. Due to the rotating movement, the ring projects onto a sphere. By varying the light intensity of the individual light sources during the rotation, the system is able to show light patterns on the sphere.

Design 2 This design consists of one or two light sources that rotate(s) around two perpendicular axes. With the rotation around axis M, the light source(s) will project onto a circle. When this part is rotated around axis B, which is perpendicular to axis M, the device will be able to project onto an entire sphere. By varying the intensity of the light source while the device spins around both axes, it is possible to project different light patterns into a room.

Design 3 This design consists of a rotating mirror instead of a rotating light source. The design uses only one light source that is focused on a mirror that rotates around axis M. In this way the rotating mirror spans up a circle. When this part is rotated around axis B, which is perpendicular to axis M, the device will be able to project onto a sphere. By varying the intensity of the light source while the device spins around both axes, it will be possible to project different light patterns into a room.

A.2 Design choice

After simulating and testing parts of the designs that are mentioned above the third design was chosen as the best.

Design 1 is not a good option because of a trade-off between weight and resolution. Adding more light sources increases the resolution, but increases the weight, which makes the device heavier to rotate. For a resolution that meets the requirements, this would require a lot of lights, which would be infeasible.

For design 2, axis M was built. Unfortunately it turned out to be very difficult to align the axle of the motor, as it needs to be fixed to the construction on either side. Due to this alignment problem, the laser, which is mounted on axis M, was not able to move smoothly and could not reach the frequency stability that is desired for projecting a stable pattern.

For design 3, a test version of axis M was built, of which the test results were promising. It has the advantage that the only part which is spinning at a high frequency is the mirror, which is relatively light, whereas the laser is only moving slowly. This makes it easier to stabilise the system.

These results and the difficulties with the other two designs, lead to the decision to implement design three.

Appendix B

Wireless circuit simulation

When the LED signal is pulsing, the capacitor voltage will not reach the threshold of half the source voltage. However, when the LED signal stays high, it will pass this threshold. This process can be seen in Figure B.1 If the connection is lost and the LED signal starts pulsing again, the capacitor will be completely discharged directly after the first falling edge. The comparator output signal becomes low again indicating loss of connection very fast.

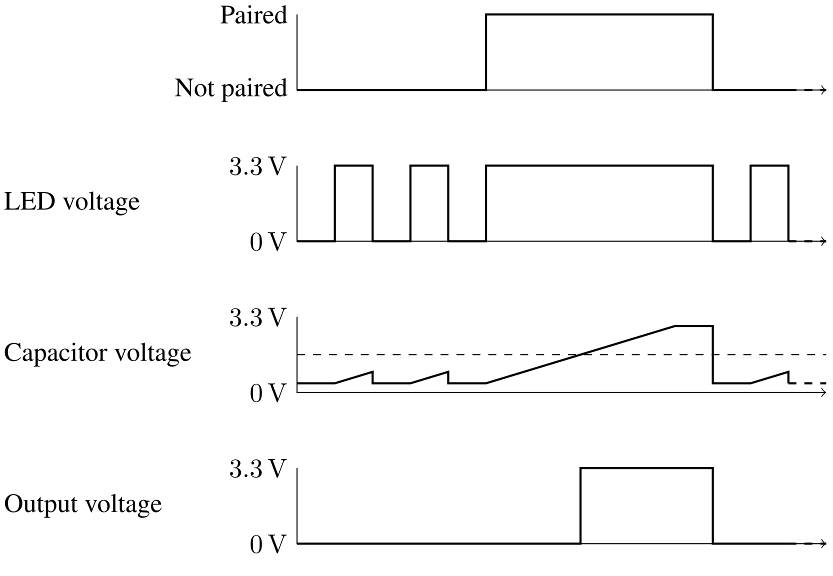


Figure B.1: Voltages in circuit from Figure 5.1

The circuit from Figure 5.1 is also simulated using LTspice. The results of this simulation can be seen in Figure B.2. These results look a lot like the voltages estimated in Figure B.1. The simulations have been done with an LED frequency of 5 Hz and there are no false positives. It can also be seen that the circuit introduces a delay of approximately 300 ms between the devices being paired and the output telling the microcontroller that a device has paired. This is not a problem since it will mostly happen once during an entire measurement. The detection of loss of connection however is instant which contributes to the safety requirement mentioned before.

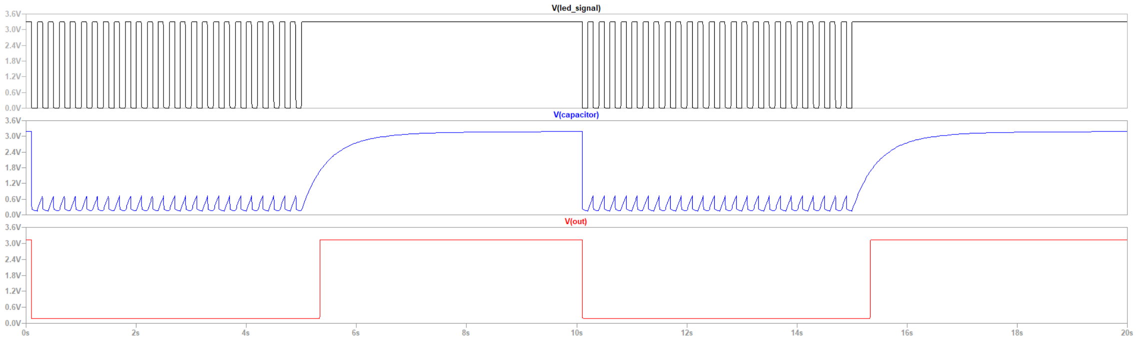


Figure B.2: Simulation results of the circuit from Figure 5.1 using LTspice. From top to bottom: LED signal voltage, capacitor voltage, output voltage to microcontroller.

Appendix C

Pattern calculation CUDA algorithm

```
// Define pi
#define  pi 3.14159265358979323846

/** Work out which piece of the global array this thread should
 ↪  operate on */
__device__ size_t calculateGlobalIndex() {
    // Which block are we?
    size_t const globalBlockIndex = blockIdx.x + blockIdx.y *
     ↪  gridDim.x;
    // Which thread are we within the block?
    size_t const localThreadIdx = threadIdx.x + blockDim.x *
     ↪  threadIdx.y;
    // How big is each block?
    size_t const threadsPerBlock = blockDim.x*blockDim.y;
    // Which thread are we overall?
    return localThreadIdx + globalBlockIndex*threadsPerBlock;

}

/** Main function to calculate the convolution at every point on the
 ↪  sphere **/
__device__ double calcSum(
                            double const aziRot,
                            double const polarRot,
                            const double * azi,
                            const double * polar,
                            double const * pattern,
                            double const openingAngle,
                            const double gridRes,
                            const unsigned int aziLen,
                            const unsigned int numel) {
    // Initialise output variable
    double sum = 0;

    // Calculate start and stop indices to loop over
    int iStart = ((polarRot - openingAngle / 2) / gridRes) * aziLen;
    int iStop = ((polarRot + openingAngle / 2) / gridRes + 1) *
     ↪  aziLen;
```

```
// Limit start and end index to array bounds
if(iStart < 0) {
    iStart = 0;
}

if (iStop > numel) {
    iStop = numel;
}

// Calculate X,Y,Z coordinates of the midpoint on the sphere
double xRot = cos(aziRot * pi / 180) * sin(polarRot * pi / 180);
double yRot = sin(aziRot * pi / 180) * sin(polarRot * pi / 180);
double zRot = cos(polarRot * pi / 180);

// Calculate the max distance at which the beam pointed at the
// coordinates calculated above will shine on the sphere
double maxDistSq = pow(2*sin(openingAngle * pi / 720), 2);

// For every potential point that projects a beam which contains
 ↪  the current point
for(unsigned int i = iStart; i < iStop; i++) {
    // Check if the laser is on
    if (pattern[i] != 0) {
        // Determine at which point the laser is pointed
        double aziAngle = azi[i];
        double polarAngle = polar[i];

        // Convert the spherical coordinates to carthesian
        double x = cos(aziAngle * pi / 180) * sin(polarAngle * pi
         ↪  / 180);
        double y = sin(aziAngle * pi / 180) * sin(polarAngle * pi
         ↪  / 180);
        double z = cos(polarAngle * pi / 180);

        // Calculate the distance between the current point and
         ↪  the
        // point where the laser is pointing
        double dist = pow(x - xRot, 2) + pow(y - yRot, 2) + pow(z
         ↪  - zRot, 2);

        // If the point lies within the beam
        if (dist < maxDistSq) {
            // If the distance is non zero
            if (dist > 0) {
                // Normalise the distance
                dist /= maxDistSq;
                // Approximate Gaussian profile with a sinc
                 ↪  function
                sum += pattern[i] * sin(dist * pi) / (dist * pi);
            } else {
                // If the distance is zero, the sinc computation
                 ↪  will
                // contain a division by zero which will crash the
                 ↪  gpu
```

```
                    sum += pattern[i];
                }
            }
        }
    }
    return sum;
}

/** Main entry point.
 * Works out where the current thread should read/write to global
 ↪  memory
 * and calls calcSum to do the actual work.
 */
__global__ void beamConvolution(
                      double * out,
                      const double * azi,
                      const double * polar,
                      const double * pattern,
                      const double openingAngle,
                      const double gridRes,
                      const unsigned int aziLen,
                      const unsigned int numel ) {
    // Work out which thread we are
    size_t const globalThreadIdx = calculateGlobalIndex();

    // If we're off the end, return now
    if (globalThreadIdx > numel) {
        return;
    }

    // Get our X and Y coords
    double const aziRot = azi[globalThreadIdx];
    double const polarRot = polar[globalThreadIdx];

    // Calculate the sum on this location
    double const sum = calcSum( aziRot, polarRot, azi, polar, pattern,
     ↪  openingAngle, gridRes, aziLen, numel );
    out[globalThreadIdx] = sum;
}
```