UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

MASTER'S THESIS

DESIGN AND IMPLEMENTATION OF AN AUTOMATED MEASURING SYSTEM FOR A GONIOPHOTOMETER

ABDÓN ALEJANDRO VIVAS IMPARATO
2017

TITLE: DESIGN AND IMPLEMENTATION OF AN AUTOMATED MEASURING SYSTEM FOR A GONIOPHOTOMETER

AUTHOR: ABDÓN ALEJANDRO VIVAS IMPARATO

TUTOR: MIGUEL ÁNGEL EGIDO AGUILERA

DEPARTMENT: INSTITUTO DE ENERGÍA SOLAR

ABSTRACT

The IES (*Instituto de Energía Solar*, spanish for Institute of Solar Energy) of the *Universidad Politécnica de Madrid* (Technical University of Madrid) decided to build a goniophotometer, which is an instrument to measure the angular dependence of a photometric quantity. Although the purpose of this decision is local quality testing, it is planned to be replicated in countries where the IES has rural electrification projects.

The aim of this project is to design the automated measuring system for that goniophotometer. As it is intended to implement this design in different geographical locations, it must be as replicable as possible, i.e., the lesser the dependence on a specific component, the better. In consequence, an architecture was designed instead of a particular automated measuring system. This way, by using similar components, it can be implemented anywhere in the world. However, as a proof of concepts (and to be used at the IES), this project also involves an implementation of that architecture.

Concerning the IES goniophotometer, it is a fixed-sensor, moving-light source, type C goniophotometer: Two motors rotate the light source in the polar and azimuth axes while a sample of the illuminance fallen on a sensor is taken in all directions. Afterwards those samples are converted into luminous intensities using the inverse square law, and the total luminous flux is calculated from them.

The resultant architecture of the automated measuring system was named after its only non-generic and core element: KNDL. It is a freely distributed Java program developed during this project. Its purpose is to collect, sort, process, and store the measured data. These data are received from a control device (i.e. a microcontroller) which orchestrates the rest of the elements: two reference subsystems, two stepper motors (and their respective motor drivers), and a sensing subsystem.

For the implementation an Arduino Board was used as a control device, an optocoupler and a metallic disk with a notch for each reference subsystem, two H-bridge motor drivers, two bipolar stepper motors, and two types of sensing subsystems: A separated lux meter and an analog-to-digital converter, and an integrated circuit.

Finally, a few light sources were measured in order to generate results. Two set of tests were performed: One to compare plots of the resulting luminous intensity distribution with the shape of the light sources, and another to compare the resulting total luminous flux of the light sources with that of another system (more specifically, an integrating sphere).

RESUMEN

El Instituto de Energía Solar (IES) de la Universidad Politécnica de Madrid decidió construir un goniofotómetro, que es un instrumento utilizado para medir la dependencia angular de una magnitud fotométrica. Aunque el propósito de esta decisión sea para test de calidad local, el IES planea replicarlo en otros países donde tiene proyectos de electrificación rural.

El objetivo de este proyecto es diseñar el sistema de automatización de medida de dicho goniofotómetro. Debido a que se pretende implementar el diseño en distintas zonas geográficas, éste debe de ser lo más replicable posible, es decir, mientras menos dependencia tenga con algún componente en específico, mejor. Por lo tanto, se diseñó una arquitectura en lugar de un sistema de automatización de medida en particular. De esta manera y usando componentes similares, dicha arquitectura puede ser implementada en cualquier parte del mundo. Sin embargo, este proyecto también involucra la implementación de esa arquitectura como prueba de conceptos (y para ser utilizada en el IES).

En cuanto al goniofotómetro del IES, se trata de uno tipo C, con sensor fijo y fuente luminosa móvil. Dos motores hacen girar la fuente luminosa en los ejes polar y azimutal, a la vez que una muestra de la iluminancia que cae en un sensor es tomada en todas las direcciones. Posteriormente, dichas muestras son convertida a intensidad luminosa usando la ley de la inversa del cuadrado y el flujo luminoso total es calculado a partir de ellas.

La arquitectura resultante del sistema de automatización de medida fue nombrada tras su único elemento no genérico, que además es el núcleo de ésta: KNDL. Se trata de un programa hecho en Java durante este proyecto y que es distribuido de forma libre. Su trabajo es el de recolectar, ordenar, procesar y almacenar los datos de las medidas. Estos datos son recibidos desde un dispositivo de control (un microcontrolador) que orquestra al resto de los elementos: Dos subsistemas de referencia, dos motores paso a paso (junto con sus respectivos controladores) y un subsistema de detección.

Para la implementación se utilizó una placa Arduino como dispositivo de control, un optoacoplador y un disco metálico con una muesca para cada subsistema de referencia, dos puentes H como controladores de los motores, dos motores paso a paso bipolares, y dos tipos de subsistemas de detección: Un luxómetro y un conversor analógico digital separados, y un circuito integrado.

Finalmente, se midieron unas fuentes luminosas para generar resultados. Dos conjuntos de test fueron llevados a cabo: Uno para comparar gráficas de la distribución de intensidad luminosa resultante con la forma de las fuentes luminosas, y otro para comparar los resultados del flujo luminoso total calculado con el mismo medido con una esfera integradora.

KEYWORDS

Instrumentation, Automation, Goniophotometer, Photometry, Illuminance, Luminous Intensity, Luminous Flux, FWHM, Arduino, Java, MATLAB.

A mi tío Tony Eblen.
Junto a él aprendí a construir un sismógrafo y un magnetómetro terrestre.
Lo mínimo que puedo hacer en agradecimiento es dedicarle un goniofotómetro.

ACKNOWLEDGEMENTS

A Inés por escucharme todo el tiempo y darme su apoyo.
A mis padres, siempre, por su apoyo incondicional.
A Miguel Ángel por haberme guiado en este camino.
A Zully por ayudarme siempre con las revisiones finales.
A Juan Carlos y a Jorge, por sus ideas y recomendaciones.
A Félix por prestarnos las lámparas del CIEMAT.
A la yayi por su caja de zapatos donde guardé mi modelo casero de goniofotómetro.
A Irene y a Lua por introducirme a LaTeX, a Pablo por prestarme su Arduino y a Mónica por sus inspiradores cantos de sirena.
A mis compañeros capybaros del máster, por su compañía excepcional durante esta etapa de estudios.
Por último, y no menos importante, a la pizza. Sin ella no habría podido sobrevivir a mis épocas más difíciles en este proyecto.

Contents

1 Introduction **1**
- 1.1 Objectives . . . 3

2 Background Theory **4**
- 2.1 Basic concepts of light measurements . . . 4
- 2.2 Goniohotometer . . . 6
- 2.3 Spherical coordinate system convention . . . 7

3 Procedure **9**
- 3.1 Infrastructure . . . 9
- 3.2 Proposed architecture: KNDL architecture . . . 10
  - 3.2.1 KNDL Architecture in detail . . . 11
- 3.3 KNDL (Software) . . . 12
  - 3.3.1 KNDL protocol . . . 13
  - 3.3.2 Calibration process . . . 16
  - 3.3.3 Calculation of results . . . 16
- 3.4 Hardware Implementation . . . 18
  - 3.4.1 Connection diagram . . . 23
  - 3.4.2 BOM . . . 23
- 3.5 Software Implementation (Arduino Code) . . . 25

4 Results **27**
- 4.1 LID tests . . . 27
  - 4.1.1 Post-processing . . . 27
  - 4.1.2 LID results . . . 28
- 4.2 Luminous flux tests . . . 34
  - 4.2.1 Sensing subsystem #1: MESA MS-LUX lux meter + MCP3422 ADC . . . 35
  - 4.2.2 Sensing subsystem #2: TSL2561 Light sensor . . . 36

5 Conclusion **38**
5.1 Future research and possible improvements 40

A KNDL user guide **43**

ABBREVIATIONS

ADC - Analog-to-Digital Converter

BOM - Bill Of Materials

CIEMAT - *Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas* (Center of Energetic, Environmental and Technological Research)

CSV - Comma-Separated Values

FWHM - Full Width at Half Maximum

GUI - Graphical User Interface

IC - Integrated Circuit

IES - *Instituto de Energía Solar* (Institute of Solar Energy)

JVM - Java Virtual Machine

LED - Light-Emitting Diode

LID - Luminous Intensity Distribution

QTM - Quality Test Method

UPM - *Universidad Politécnica de Madrid* (Technical University of Madrid)

USB - Universal Serial Bus

Chapter 1

Introduction

Since the second edition of the IEC/TS 62257-9-5 [1] Technical Specification came out in April 2013, photovoltaic systems benefit from its recommendations for small renewable energy and hybrid systems for rural electrification. In its QTM (Quality Test Method) section, the referred technical specification recognizes the goniophotometer as a test method to obtain the LID (Luminous Intensity Distribution), and in turn the luminous flux as well as the vertical and horizontal FWHM (Full-Width at Half Maximum).

On the other hand, when designing isolated photovoltaic systems, it is important to assure the quality of the equipment to be used before transporting it to the facility in order to avoid an unexpected increase of cost, workmanship and/or execution time of the project. This is more appreciated when the facility is in a hard-to-reach place such as mountains or when the roads are impaired or dirt tracks (common in rural electrification projects).

That being said, the *Instituto de Energía Solar* (IES from now on) of the *Universidad Politécnica de Madrid*, decided to design and fabricate a goniophotometer. The aim of this project (apart from local quality testing) is to implement the same (or alike) design in Bolivia, where the IES has some rural electrification projects. Extending this idea, it was decided to follow an open-source philosophy so the design could be implemented anywhere in the world. Therefore, the design must fulfill the following requirements:

1. If plausible, it must **occupy the smallest space possible**. It would be better if the device can be located inside the laboratory and not in a separate room. This can be achieved easily because the size of the goniophotometer depends on the maximum length of the light source it can measure. Fortunately, we are not going to measure street lights nor fluorescent lamps, but small lamps.

2. It must be an **automated goniophotometer**. Operating it manually

is error prone and low resolution.

3. Its design must be **as replicable as possible**. Each component should have a high geographical availability or at least should be easy to make or substitute.

4. It should be as **easy-to-use** as possible. It is uncertain if the person who is going to use it is an expert in computers.

5. It should be **affordable**.

In accordance with the first requirement, the IES decided to fabricate the goniophotometer shown in figure 1.1, with the light sensor located at the top of the structure.

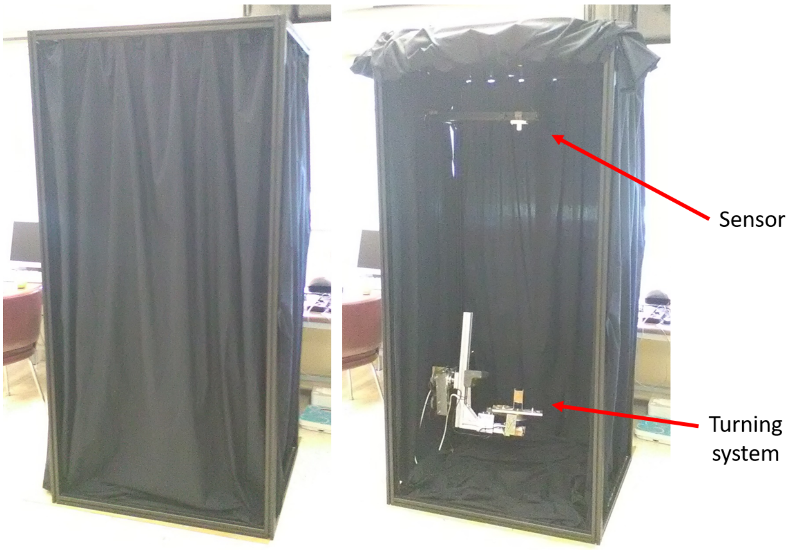

Figure 1.1: IES Goniophotometer.

Given the above, the target of this Master's Thesis is to develop an automated measuring system for the IES goniophotometer (figure 1.1), focusing in the replicability of the design. In consequence, an architecture must be designed instead of a specific implementation. This way, any component could be selected depending on its availability wherever the architecture is being implemented.

1.1 Objectives

Main Objective

To design and implement an automated measuring system for the IES goniophotometer.

Specific Objectives

- To design an architecture to build an automated measuring system for a generic goniophotometer.
- To implement that design as a proof o concepts and as part of the IES goniophotometer.
- To verify the correct operation of the implementation by measuring a few light sources and generating results.
- To compare the results with measures taken with professional equipment and evaluate the difference.

Chapter 2

Background Theory

A few concepts about light measurements and goniophotometers will be summarized in this chapter. The convention used for spherical coordinates will also be exposed. The concepts explained in section 2.1 were mainly extracted from [2] and [3].

2.1 Basic concepts of light measurements

Luminous flux, Φ_v

The luminous flux is defined as the total *visible* radiant power (light energy per second) emitted by a light source, measured in lumen (lm); i.e. the quantity of light (luminous intensity) emitted by a source in all directions.

Luminous intensity, I_v

The luminous intensity is defined as the amount of *visible* radiant power per unit solid angle, measured in candela ($\mathrm{cd} = \frac{\mathrm{lm}}{\mathrm{sr}}$). It describes the quantity of light radiated in a particular direction. When having the luminous intensity of a light source in all directions, the luminous flux can be calculated using equation 2.1.

$$\Phi_v = \int\int I_v d\Omega = \int_0^{2\pi}\int_0^{\frac{\pi}{2}} I_v(\theta,\varphi)\sin\theta d\theta d\varphi \tag{2.1}$$

Ω being the solid angle, θ the polar angle, and φ the azimuth angle.

Illuminance, E_v

The illuminance is defined as the luminous flux per area unit, measured in lux ($\mathrm{lx} = \frac{\mathrm{lm}}{\mathrm{m}^2}$). It describes the luminous flux falling on a surface. By definition, it can be related with the luminous flux as shown in equation 2.2 and it is the measure given by lux meters.

$$\Phi_v = \int\int E_v dS = \int_0^{2\pi}\int_0^{\frac{\pi}{2}} r^2 E_v(\theta,\varphi)\sin\theta d\theta d\varphi \tag{2.2}$$

S being the surface where the light falls on, r the distance between the light source and the surface, and θ and φ the polar and azimuth angles respectively.

Figure 2.1 illustrates these three concepts of luminous flux, luminous intensity, and illuminance.

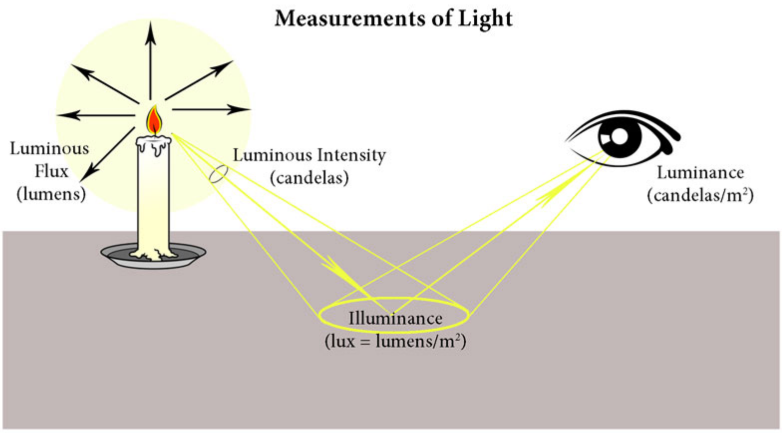

Figure 2.1: Illustration of the luminous flux, luminous intensity, illuminance, and luminance[4].

Photometric center

The photometric center refers to the point in a lamp or luminaire from which the photometric distance law operates most closely in the direction of maximum intensity[5].

The inverse square law

The inverse square law (figure 2.2) states that the illuminance E at a point on a surface varies directly with the luminous intensity I of the source, and inversely as the square of the distance d between the source and the point[2]. *If the surface at the point is normal to the direction of the incident light*, the law may be expressed as in equation 2.3.

$$E_v = \frac{I_v}{r^2} \tag{2.3}$$

It is important to emphasize that this law strictly applies to a point source. However, it can also apply to non-point sources at sufficiently large distances so they can be considered as a point-source [5]. A distance five times larger than the maximum luminous dimension of the source should be enough [2][3][6], though for non-Lambertian light sources it may be a greater distance.

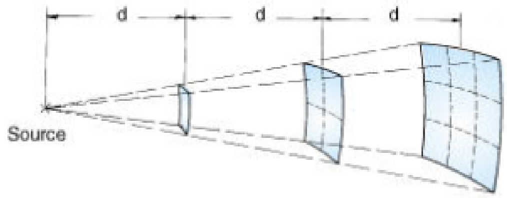

Figure 2.2: Illustration of the inverse square law[2].

2.2 Goniohotometer

A goniophotometer is an instrument to measure the angular dependence of a photometric quantity. There are various types of goniophotometers depending on the moving part: the light source, the photometer head, or a moving mirror [7]. When the moving part is the light source, then there are 3 types of goniophotometers (figure 2.3). According to the facility of the IES goniophotometer, it corresponds to a type 3 goniophotometer (or type C [2] [8]).

A goniophotometer can be used to measure the LID (Luminous Intensity Distribution), luminous flux, and efficiency of lamps and luminaries. Usually the sensor measures illuminance (lux meter) and the luminous intensity is calculated using equation 2.3, so it is important for the distance between the light source and the lux meter to be at least five times the larger than the largest luminous dimension of the light source (see section 2.1).

The luminous flux is then calculated using equation 2.1, or obtained directly from equation 2.2. In fact, even if equation 2.3 does not apply, the luminous flux can still be calculated from the illuminance providing that the detector has good cosine response [9]. This is because irradiance is not a property of the lamp, but is instead the light falling onto a surface.

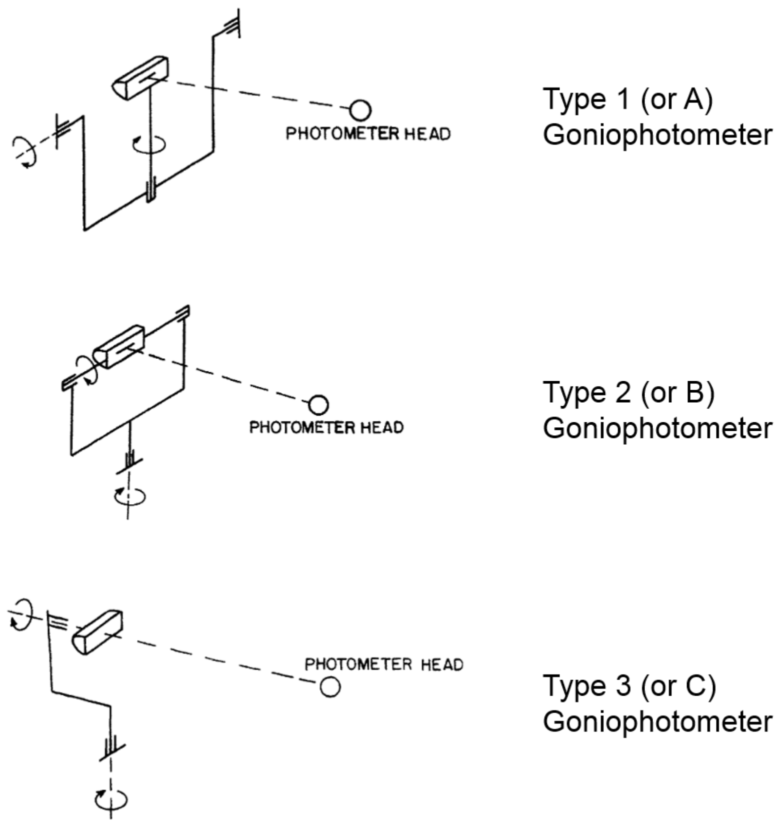

Figure 2.3: Types of goniophotometers. Images taken from [7].

2.3 Spherical coordinate system convention

It is important to highlight that the spherical coordinates convention used throughout this project was that of figure 2.4: r is the radial distance, θ is the polar angle, and φ is the azimuth angle.

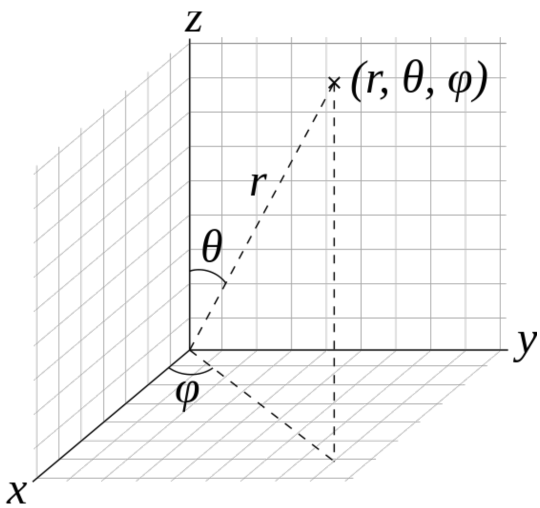

Figure 2.4: Spherical coordinate system convention used during this project.

Chapter 3

Procedure

3.1 Infrastructure

As seen in figure 1.1, the IES goniophotometer is a set of profiles that forms a rectangular prism with a two-degrees-of-freedom turning system at the bottom, all covered by black curtains. Remember that, as said in section 2.2, the IES goniophotometer is a type C goniophotometer. The black curtains are used to avoid as much stray light as possible to be reflected inside the goniophotometer, and to keep it from outside light.

Usually goniophotometers occupy a lot of space. They are isolated in a whole room. But this is a consequence of measuring large light sources such as street lights or fluorescent lamps. However, if the light source is not going to be that spacious, smaller goniophotometers can be used as in [10][11][12], for example. Keeping in mind the five-times rule (see section 2.1) and taking into account that the largest luminous dimension of the IES lamps intended to be measured is 13.2cm, the distance between the photometric center and the lux meter must be at least 66cm. However, the IES goniophotometer has a configurable measure distance that can be up to 150cm (theoretically, a light source with a maximum luminous dimension of 30cm can be measured). All in all, the IES goniophotometer can be relatively small compared to those dark-room goniophotometers [13][14][15].

A detailed view of the turning system of the IES goniophotometer is shown in figure 3.1. The colored lines indicate the two rotation axis: the green line indicates the polar angle rotation axis and the red line the azimuth rotation axis.

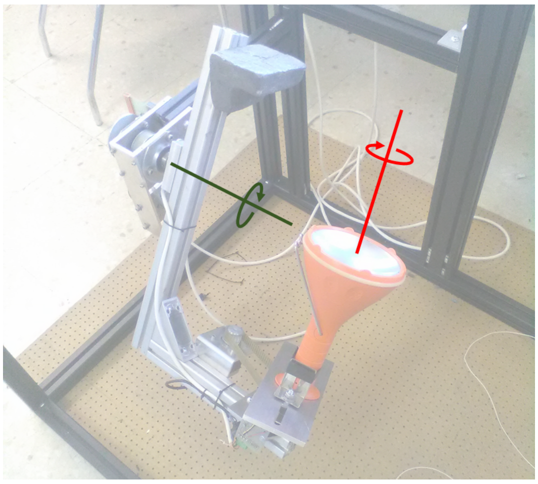

Figure 3.1: Turning system of the IES Goniophotometer.

3.2 Proposed architecture: KNDL architecture

Figure 3.2 shows a diagram of the designed architecture. Its first element is a software named KNDL (pronounced 'candle') running in a JVM (Java Virtual Machine)[1]. The computer executing KNDL is connected to a control device (i.e. a microprocessor) through a USB cable, which in turn is connected to the rest of the components: An ADC (Analog-to-Digital Converter), two motor drivers, one lux meter, two bipolar motors, and two reference subsystems.

In a normal operation, a user initiates the measure via KNDL using the correct configuration for his/her specific goniophotometer implementation (see appendix A). KNDL then just sends a start command and waits for the measures to process them and store the results. The control device then places the motors in the right position to make the first light measure, and when it is done, it moves the motors to the next position according to the user configuration and takes another measure. This process takes as long as the user specified in the resolution configuration (can be hours). When the control device is done, KNDL makes some final processing and stores the results in a CSV file.

As KNDL is the only non-generic element in this architecture, this architecture will be referred from now on as **KNDL architecture**.

[1]KNDL was compiled with JDK (Java Development Kit) 1.7

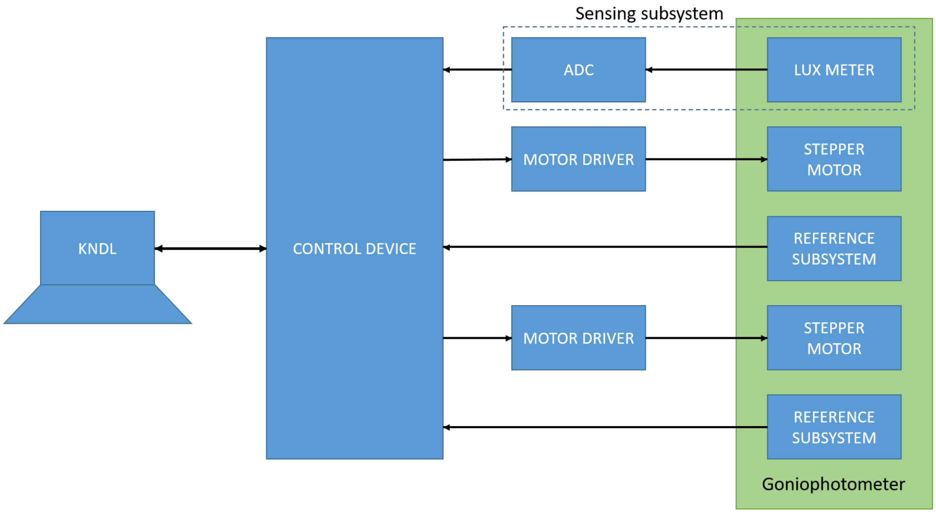

Figure 3.2: KNDL architecture diagram.

3.2.1 KNDL Architecture in detail

This section is intended to explain the function of each element of the KNDL architecture, as well as to mention some minimum requirements regarding these.

Sensing subsystem

The ADC and the lux meter form the sensing subsystem of the architecture. Their function is to take, digitize and send the illuminance measure to the control device. KNDL provides a constant (k in equation 3.1) as part of its configuration in case the volts-to-lux relation of the lux meter is linear, or at least it can be converted to a linear equation like equation 3.1. However, if the volts-to-lux relation of the lux meter is not as specified in equation 3.1, the conversion can still be handled by the control device and sent to KNDL. In such cases, k must be set to 1 in KNDL's configuration.

$$E_v = kV \tag{3.1}$$

Reference subsystems and bipolar motors

They are the elements of the architecture that provide motion. Their function is not only to move the structure to each measure position, but to also provide a way to infer that position.

KNDL architecture is designed to use stepper motors in order to keep track of the angle during the measuring process. This way, by knowing the degrees per step of the stepper motors, and counting the number of steps taken from a reference angle (origin), it is possible to know the relative angle between a given location and that reference angle. Part of KNDL configuration is to let it know the step angle of the goniophotometer (it may not be the same as the motors' step angle if a power transmission system is used).

The reference subsystem is a system that somehow tells the control device when a stepper motor is positioned in the reference angle. There are many ways to achieve this goal. One is mentioned later in section 3.4.

Motor drivers

A microprocessor is not normally able to drive a stepper motor by itself. In turn, it can send control signals to a motor driver which provides the motor with the necessary amount of current for it to move.

Control device

The job of the control device is to move the motors and to send each measure taken to a serial port in a computer where KNDL will do the rest. The selection of the best control device depends on each implementation: The type of motor drivers used, the communication bus used by the ADC/sensing element, and the implementation of the reference subsystem.

Apart from all this, the control device must be programmed to follow KDNL protocol, described in section 3.3.1.

3.3 KNDL (Software)

KNDL was developed in Java so it can run in any operating system with a JVM installed. It is intended to be as compatible as possible and easy to maintain. It has an intuitive and user-friendly GUI (Graphical User Interface) and some advanced functions that make it more comfortable to use. However, its basic functions are sufficient to generate results (see appendix A for more information).

KNDL was not developed to work with the IES goniophotometer exclusively, but also as a software for collecting and storing data for any goniophotometer that implements KNDL architecture (section 3.2).

3.3.1 KNDL protocol

The communication between KNDL and the control device consists of Strings sent through the serial port. When KNDL sends a command, it expects the control device to execute a procedure and send back information. Table 3.1 shows the commands that KNDL can send to the serial port along with the description of each of them. Note that right after sending a start command, KNDL sends 4 configuration parameters. Those parameters are the following:

1. Step angle (degrees per step) corresponding to the polar angle.
2. Step angle (degrees per step) corresponding to the azimuth angle.
3. Resolution of the measure in steps per sample (Polar angle).
4. Resolution of the measure in steps per sample (Azimuth angle).

The enumeration corresponds to the actual order in which the parameters are sent.

Command	**String sent**	**Description**
START	STR	Starts a measure. Right after sending the START command, KNDL sends 4 configuration parameters so the device can use them to measure.
STOP	STP	Stops a measure in process.
CALIBRATE	CAL	Starts a calibration process.

Table 3.1: Commands sent from KNDL to the control device.

When the control device sends data back to KNDL, it must use a semicolon as a separator so that KNDL can understand it. There are three different responses that the device can send (Table 3.2). However, any String sent will be shown in KNDL's monitor, so the device can send messages that may concern the goniophotometer's operator.

Having described the commands and responses of KNDL's protocol, figures 3.3 and 3.4 show the protocol sequences for the measure process and the calibration process respectively. Remember that 'THETA' refers to the polar angle and 'PHI' refers to the azimuth angle (See section 2.3).

Response	Format	Description
ACKNOW-LEDGE-MENT	ACK;	Acknowledges the command from KNDL (Table 3.1). When receiving the START command, the acknowledgement must be sent after receiving the configuration of the measure.
DATA	**When calibrating:** DAT;[CALIBRATION_RESULT]; **When measuring:** DAT;[THETA];[PHI];[VALUE];	**When calibrating:** To send the calibration result (an offset to be subtracted from each sample, in volts or lux, depending on the unit the control device sends to KNDL). **When measuring:** To send the coordinates and value of each illuminance sample taken (in volts, if it has not been converted yet, or in lux).
END	END;	To inform KNDL that a measure has ended and the device will enter into IDLE state again.

Table 3.2: Responses sent from the control device to KNDL.

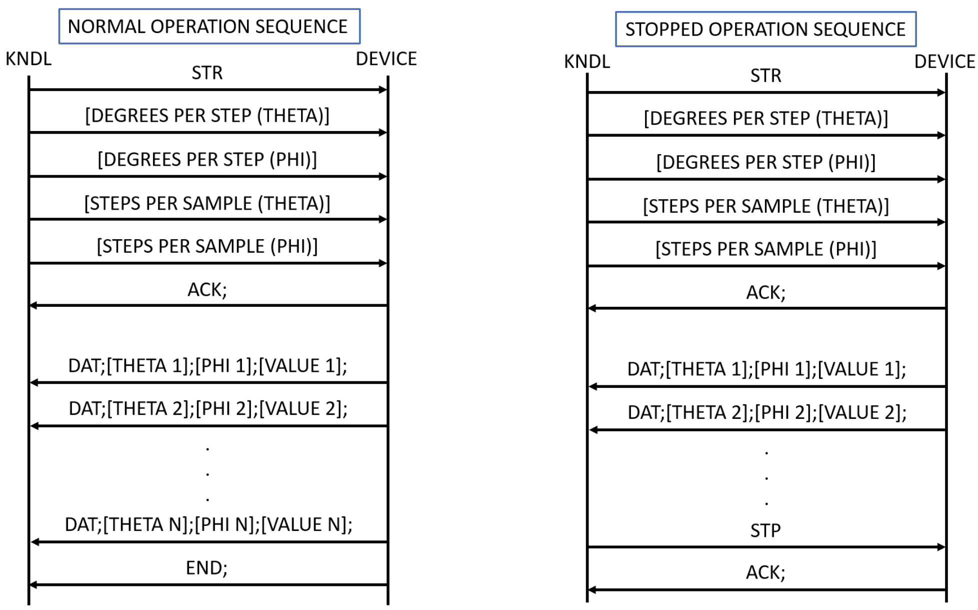

Figure 3.3: Protocol sequence of a successful and a stopped measure process.

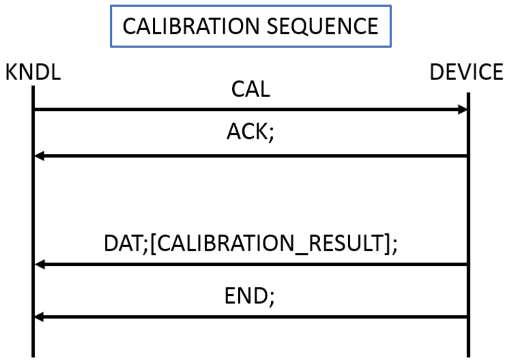

Figure 3.4: Protocol sequence of a calibration process.

The commands are sent each time an action button is pressed on the GUI. Figure 3.5 shows KNDL main window. When pressing the buttons 'Start measure', 'Stop measure' and 'Calibrate'; KNDL sends START, STOP and CALIBRATE commands through the serial port respectively. For information about how to use KNDL, see appendix A.

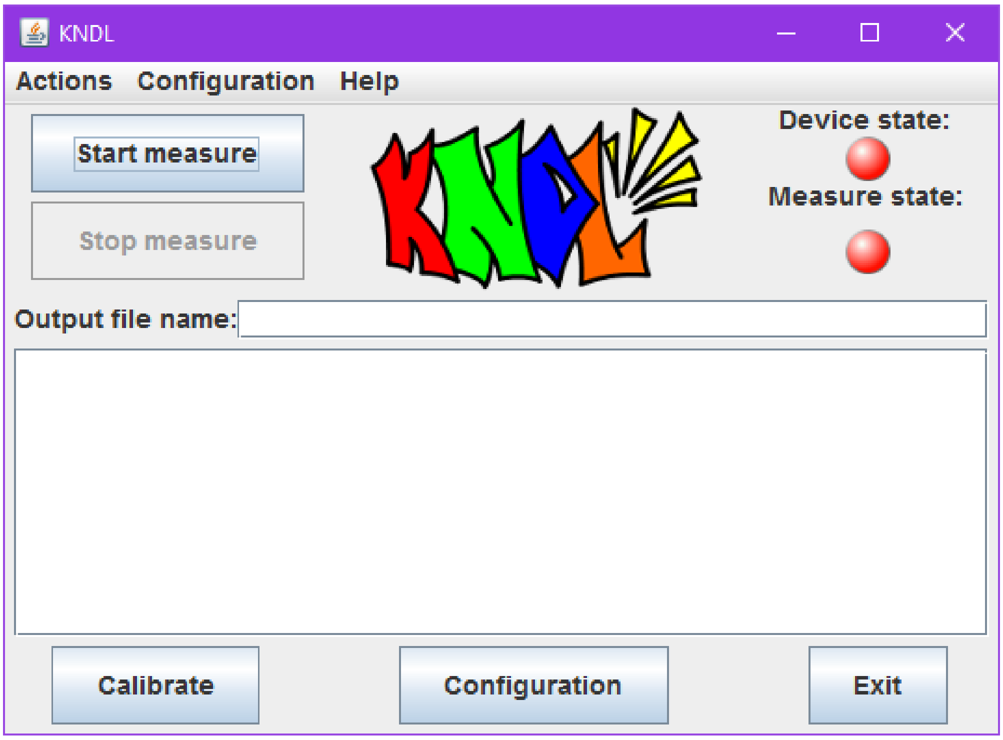

Figure 3.5: KNDL main window.

3.3.2 Calibration process

KNDL gives you the possibility to define an offset to be subtracted from each measure if you cannot manage to get 0 lx from your goniophotometer when the light source is off. This is done by means of a calibration process. The idea is for you to implement it in the control device the way you think is better. For example, you can take a single measure, or the minimum out of five measures if your sensor is a bit noisy. When your control device sends the result to KNDL, it will subtract that value to each sample taken during a measure. The offset can be reseted from the advanced configuration menu (see appendix A).

It is recommended to avoid the use of this functionality by building your goniophotometer as good as possible.

3.3.3 Calculation of results

KNDL packs three results from each measure in a CSV file: The luminous flux, the horizontal and vertical FWHM, and the LID. These calculations are possible thanks to a Java object that associates the spherical coordinates of each sample taken with their luminous intensity value.

The luminous intensity is calculated from each illuminance value received from the control device and the distance between the light source and the sensor applying equation 2.3. The source-to-sensor distance, in meters, is part of KNDL's configuration.

The horizontal and vertical FWHM are easy to calculate. The procedure is to search for the polar angles for which the value of the luminous intensity is the closest to half the maximum value of the luminous intensity for azimuth angles 0°, 90°, 180° and 270°. By summing up the results for $\varphi = 0°$ and $\varphi = 180°$, you obtain the horizontal FWHM, and by summing up the results for $\varphi = 90°$ and $\varphi = 270°$, you obtain the vertical FWHM (depending on the reference you took when measuring). Note that it is important to place the light source in the correct position before taking a measure.

As for the luminous flux calculation, it seems more difficult to come up with an algorithm parting from equations 2.1 or 2.2. By putting them into practice, equation 3.2 is obtained[16]. However, it is still difficult to implement an algorithm parting from that equation. In turn, it should be easy to geometrically see what does equation 2.2 says, and come up with an efficient algorithm with that picture in mind.

$$\Phi_v = 2\pi r^2 \sum_{\theta=0}^{\pi} \left[\sum_{\varphi=0}^{2\pi} E_{m,n}(\theta, \varphi)[\cos\theta_l - \cos\theta_h] \right] \tag{3.2}$$

To visualize it easily, instead of taking into account a full sphere, consider the semi-sphere from figure 3.6, 1st image. On the left is a top view of the semi-sphere and on the right is an oblique perspective of it. The light source would be located in the bottom of the semi-sphere, at the center of the base. The lines represents all the points the sensor passes through with a resolution of 45° on each angle (polar and azimuth). Note that this is relative, as the sensor is fixed and the light source is the one rotating in the bottom-center of the semi-sphere.

The points in which a sample of the illuminance is taken are the intersections of those lines (the colored points of figure 3.6, 2nd image). What equation 2.2 is telling us, is to integrate the values of the illuminance measured along a whole sphere (or semi-sphere in this case) in an infinitesimal matter, but as in practice we have a resolution, we divide the semi-sphere in shares (figure 3.6, 3rd image) and multiply each sample by the area of its corresponding share of the semi-sphere. This is a vast approximation as this example uses a resolution of 45°, but in practice you will have to use better resolutions.

Furthermore, there is no need calculate the area of the share for each sample as that area is the same for the shares contained in the same ring[2]. In figure 3.6, 4th image, you can see that all the shares of the blue ring have the same area. The same happens with the red and yellow shares.

In conclusion, it is enough to just calculate the area of the shares contained in one slice[3] and use the distributive property of multiplication over addition (sum of the samples contained within each ring. See figure 3.6, 5th image) to finally get the total luminous flux of the semi-sphere.

[2]Shares of the samples that have the same polar angle. This term is taken from the top view of the semi-sphere. In figure 3.6, 4th image, the rings are represennted as shares of the same color.

[3]Shares of the samples that have the same azimuth angle. This term is taken from the top view of the semi-sphere as well, if you look at it as a pizza divided in slices. In figure 3.6, 4th image, one of the slices is emphasized.

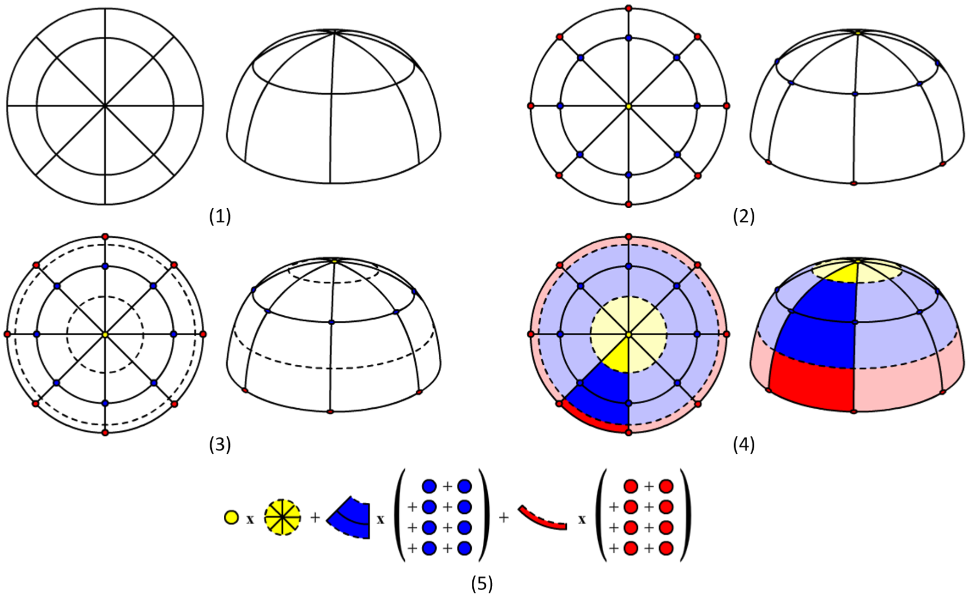

Figure 3.6: Graphical representation of the luminous flux calculation algorithm. (1) All the points in which the sensor passes through. (2) The points where a sample is taken are marked. (3) Shares of the semi-sphere marked. (4) Rings colored and one slice emphasized. (5) Graphical representation of the calculation to be made in order to get the luminous flux from the illuminances taken and the area of each share.

3.4 Hardware Implementation

The hardware implementation is composed of an Arduino board, a MCP3422 ADC, a MESA MS-LUX lux meter, two L298 dual full bridge driver modules, two RS 535-0502 bipolar motors, and two HOA2001 optocouplers. A diagram positioning each of these components in KNDL architecture is shown in figure 3.7.

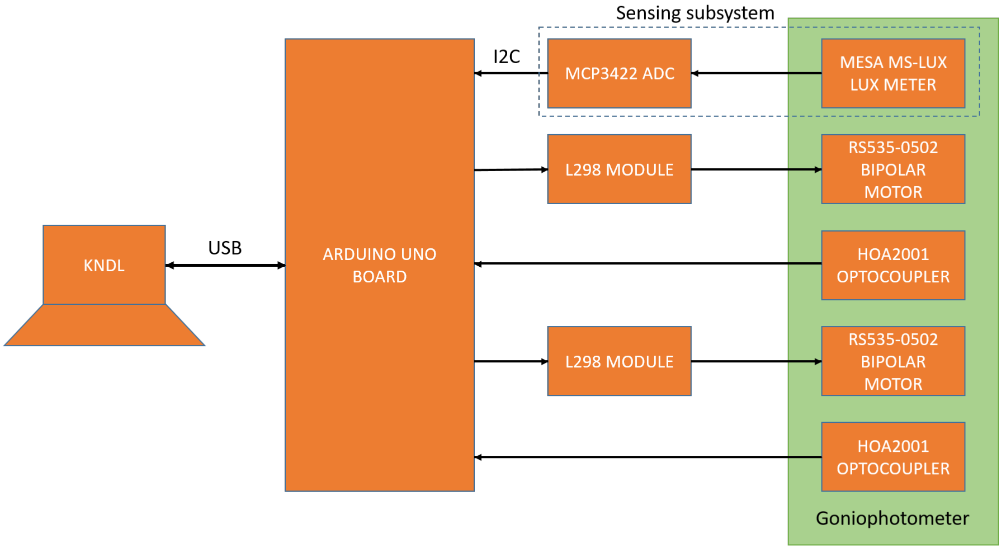

Figure 3.7: Implementation of KNDL architecture used for the IES goniophotometer.

As part of an implementation of KNDL architecture, these components may be changed in other implementations, as long as they fulfill the minimum requirements described in section 3.2.1. In consequence, there will be no further explanation of why they were chosen. This section's purpose is to describe the implementation chosen for the IES goniophotometer, and not to describe the whole selection process. It is self-evident that they are valid elements. However, some recommendations are made in order to help future selection processes.

Sensing subsystem

The lux meter was reused from other IES projects [17]. It is the MS-LUX lux meter from MESA Systemtechnik GmbH. It has a linear volts-to-lux relation (1-to-1000).

As for the ADC, it was selected based on a test (figure 3.8) using the lux meter mentioned above, an oscilloscope, a dark box and a test lamp. A measure of the output voltage was taken while the lamp was off, and another while the lamp was on. The results showed that the potential difference was less than a volt. In consequence, the MPC3422 ADC from Microchip Technology should be enough ($\pm 2.048V$ differential input range and up to 18-bits resolution). The recommendation is to select the ADC based on an experiment similar to the one described before. This ADC was also selected because of the additional advantage of having public libraries for use with Arduino boards. It communicates using I^2C bus.

Figure 3.8: Test similar to the one used to chose the ADC for the IES goniophotometer.

For verification purposes, the sensing subsystem was also implemented using TSL2561 IC (Integrated Circuit) from TAOS. It has the possibility to calculate illuminance from its output [18] without corresponding with equation 3.1. Thus, the volts-to-lux relation was set to 1 in KNDL's configuration. It also communicates using I^2C.

Optocouplers and motors

In order to know if the current position of one of the motors is aligned with the reference angle, a metallic disk with a notch was attached to the rotor. If the notch passes through a fixed optocoupler, then it means that that position is aligned with the reference angle. This idea is visualized in figure 3.9.

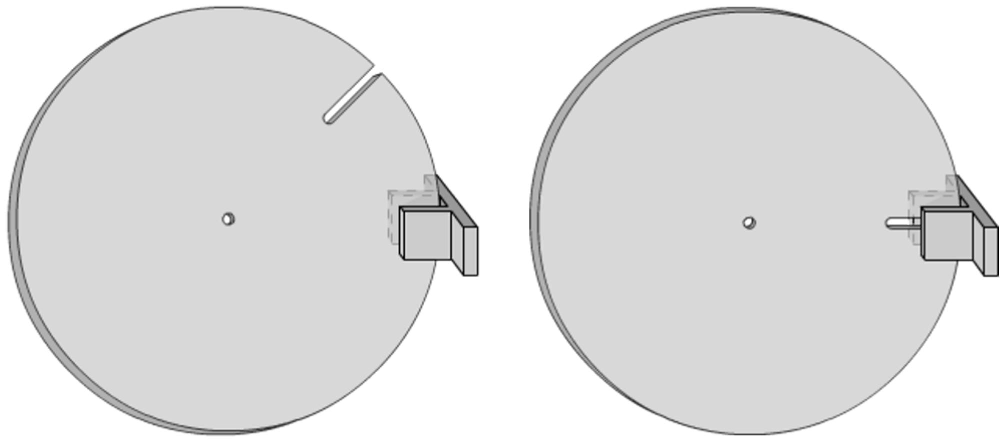

Figure 3.9: Reference subsystem. Left: Position not aligned with the reference. Right: Position aligned with the reference.

The optocouplers used were two Honeywell's HOA2001.

As for the selection of the specific motors, it depends on the torque needed to lift the structure they are moving. In the case of the IES goniophotometer, the selected motors were the RS 535-0502 motors (1.26 N m torque, 4-wire bipolar stepper motors). They were chosen based on an old infrastructure with a lighter load not shown in this document to avoid information overload. However, as seen in figure 3.1, the load of the current infrastructure is balanced on account of a lead on the opposite side of the structure holding the light source, so as long as the speed of the polar angle motor is not too fast, the mentioned bipolar motors should be enough.

Figure 3.10 shows a detailed view of one of the motors. As the step angle of the motors mentioned above is 0.9°, two belt drives (one per motor) were used in order to reduce the step angle of the motors to an overall step angle of 0.5° (the two gears of each belt drive have a 9:5 relation).

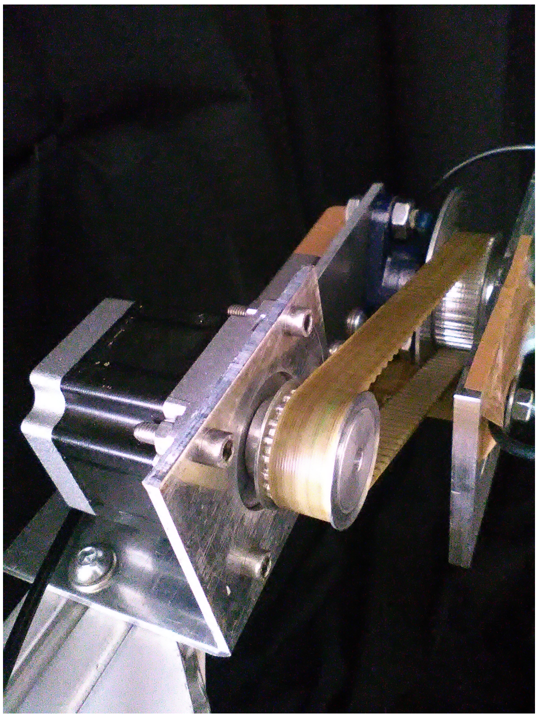

Figure 3.10: Detailed view of the motors and the belt drives.

Motor drivers

In order to move the 4-wire bipolar stepper motors mentioned above, two H-bridges are needed per motor (one per coil). Consequently, two L298 IC were selected. It is a dual full-bridge driver (basically two H-bridges within each IC).

Furthermore, the L298 IC needs a breakout board to be connected to the system, so two L298 motor driver modules (L298 modules from now on) were acquired from Diotronic to save time and effort. Nevertheless, their heat sink was replaced with a greater one (figure 3.11) as their former heat sink was not enough to support 2 A.

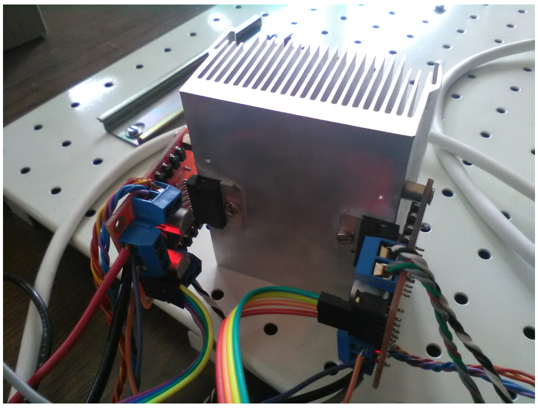

Figure 3.11: L298 modules.

Arduino board

An Arduino board is a good option for making an easy and free-to-use implementation of KNDL architecture's control device. Arduino has a set of libraries to easily control stepper motors (Stepper library), to use the I^2C serial bus (Wire library), and to send data to a computer via USB (Serial library).

The code uploaded to the Arduino board is explained in section 3.5. The specific Arduino board used was an Arduino UNO board, but any Arduino board with at least 14,754 bytes of program storage and 1,024 bytes of memory in its microcontroller, 11 Free GPIOs, I^2C communication, and serialEvent() function support may be used in conjunction with the code written for this implementation. For example, an Arduino Nano , or an Arduino Mega may be compatible as well.

3.4.1 Connection diagram

Figure 3.12 shows the connection diagram of this hardware implementation. Note that this diagram only shows the sensing subsystem with the MESA MS-LUX lux meter and the MCP3422 ADC. The other sensing subsystem would be substituting these two with the TSL2561.

3.4.2 BOM

Below is a Bill Of Materials (BOM) of the main components of the IES goniophotometer's implementation of KNDL architecture. It serves as an idea of how much an implementation of the automated measuring system can cost. This BOM does not take into account discrete components, such as resistors and capacitors, nor cables or power supplies.

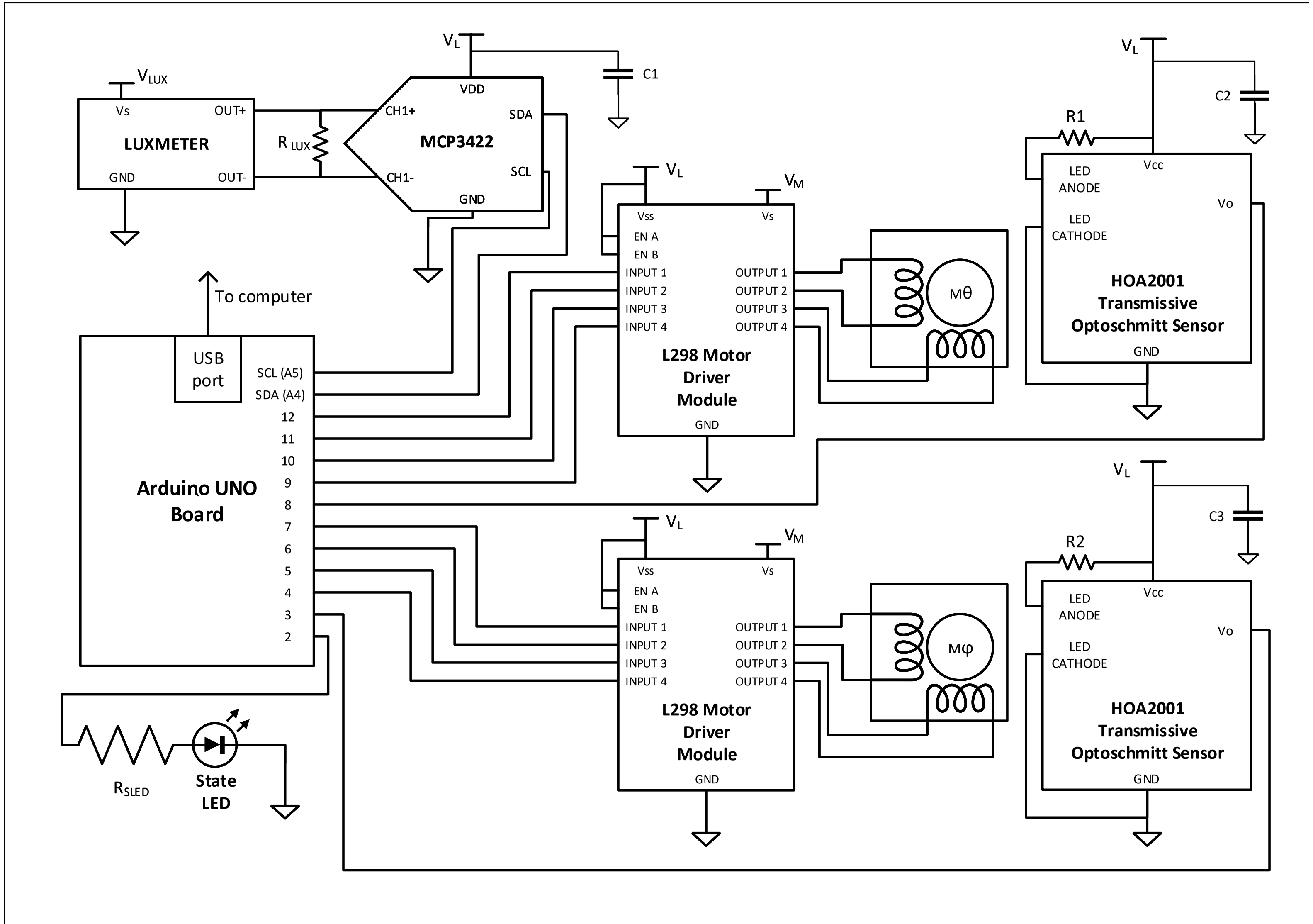

Figure 3.12: Connection diagram. V_L is the logic voltage, V_M is the motor voltage supply, and V_{LUX} is the MS-LUX lux meter voltage supply. $C1 = C2 = C3 = 0.1\mu F$, $R_{SLED} = R1 = R2 = 200\Omega$, $V_L = 5V$, $V_M = 4V$ and $V_{LUX} = 18V$.

Item number	Description	Provider	Stock number	Unit cost (€)	Quantity	Total cost (€)
0	AVR ARDUINO UNO REV 3	DIOTRONIC S.A.	A000066	20.57	1	20.57
1	DRIVER MOTOR L298 DUAL H-BRIDGE	DIOTRONIC S.A.	EF03082	18.03	2	36.06
2	RS Pro Hybrid, Permanent Magnet Stepper Motor	RS Components	535-0502	75.89	2	151.78
3	Honeywell HOA-2001 Through Hole Slotted Optical Switch	RS Components	127-3085	4.77	2	9.54
4	MESA SYSTEMTECHNIK MS-LUX LUXMETER (*)	Adler Instrumentos	---	840.16	1	840.16
5	Microchip MCP3422A0-E/SN (*)	RS Components	669-6089	2.78	1	2.78
6	TSL2561 DIGITAL LIGHT SENSOR - EVALUATION BOARD (**)	DIGI-KEY ELECTRONICS	1568-1002-ND	5.46	1	5.46
			Total Amount using sensing subsystem #1 (€)			1060.89
			Total Amount using sensing subsystem #2(€)			223.41

(*) Onnly sensing subsystem #1
(**) Only sensing subsysttem #2

Table 3.3: BOM of the main components of the IES goniophotometer's automated measuring system.

3.5 Software Implementation (Arduino Code)

As said in section 3.4, an Arduino board was used as a control device for the implementation of the proposed architecture. The purpose of this section is to explain how the Arduino board was programmed in order to follow KNDL protocol (see section 3.3.1).

Basically, the Arduino is in one of the states summarized in table 3.4. At the end of each iteration of the main loop, if it has received a String command through USB, Arduino evaluates it and decides what the next state is going to be.

State	**Command**	**State's description**
MEASURING	START	The device is taking measures and sending the results to KNDL.
IDLE	STOP	The device is waiting for START or CALIBRATE commands. This is also the state in which device is after reboot.
CALIBRATING	CALIBRATE	The device sends to KNDL the average of 5 measures. The light source must be powered off during calibration process.

Table 3.4: Arduino possible states.

A simplified diagram of the Arduino's behavior is shown in figure 3.13. When it is in IDLE state, it toggles the state LED so it blinks and let the operator know it is in IDLE state. If it receives the CALIBRATE command, it enters into CALIBRATING state, takes 5 measures, computes and sends the average to KNDL, and returns to IDLE state. If it receives the START command, it enters into MEASURING state. Once there, it receives the configuration from KNDL, initializes the motors and the ADC, and moves to the first measure's position. After that, it enters into a loop in which it takes a measure, sends the result along with the coordinates from where it was taken, and moves to the next measure's position. This loop ends either when the complete set of measures is taken or when the STOP command is received from KNDL.

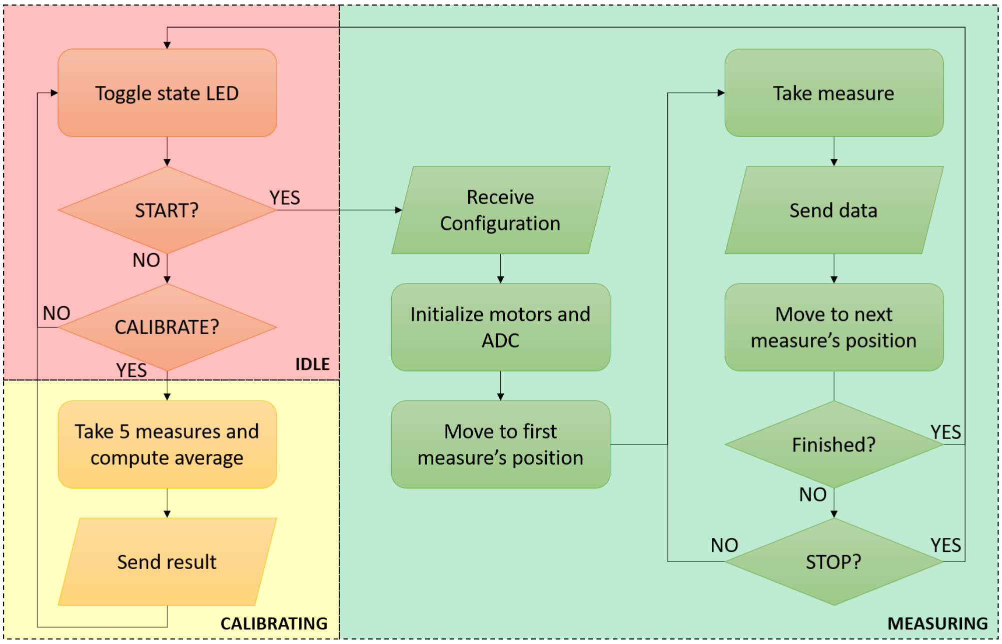

Figure 3.13: Flow diagram of the control device.

Chapter 4

Results

This chapter is divided in two sections. In the first section, a series of LID measures were made to check if the system was working properly overall. The interpretation of these results is based on the shape of the measured light sources. For example, a positive result for a lamp that seems to concentrate light because of its shape would be a LID which shows a somewhat thin beam depending on the dimensions of the lamp. The second section focuses on the validity of the quantitative results. This is achieved by comparing the resulting luminous flux of a few LED light sources with their respective luminous flux measured with an integrating sphere.

4.1 LID tests

These tests merely intended to check that the system was working properly by measuring a few light sources, each with a different form. These are *qualitative* tests.

4.1.1 Post-processing

Since KNDL's output is a CSV file, the LID is given in the form of a table and as such, it cannot be interpreted easily. In consequence, two MATLAB functions were written to produce graphics parting from KNDL's raw files (see appendix A) to evaluate the results. The function of each script is described below:

- `fixedAnglePolarPlot`: Plots the luminous intensity distribution for a fixed polar angle (e.g. figure 4.3, left) and a fixed azimuth angle (e.g. figure 4.3, right).

- `threeDpolarplot`: Generates a 3D representation of the LID (e.g. figure 4.2). This three dimensional plot is a qualitative representation as its axes do not represent anything. The reason of this is that the results are given in spherical coordinates and they must be transformed into cartesian coordinates to be 3D plotted in MATLAB. To know exact values of luminous intesity, `fixedAnglePolarPlot` must be used.

Two more functions were written to assure that the calculations regarding the luminous flux and the horizontal and vertical FWHM were consistent with KNDL's: `calculateLuminousFlux` and `calculateHVFWHM`. As the results given by these two functions were positive and do not provide any additional information, they are not shown here. However, the four MATLAB functions mentioned during this section are distributed alongside with KNDL to support future research.

4.1.2 LID results

The lamps measured during these tests were Phoco's Pico lamp, d.light's S300 lamp, and a LED light bulb from FD (one of the LED light bulbs measured in section 4.2). A picture of the Phocos Pico lamp, and the d.light S300 lamp is shown in figure 4.1.

Figure 4.1: Phocos Pico lamp (left) and d.light S300 lamp (right).

The fixed polar angle chosen for all the polar graphs of this section was that of the horizontal half maximum. As for the fixed azimuth angle, it is 0° for one half of the plot and 180° for the other half (precisely the horizontal azimuth angles).

LID Test 1: Phocos Pico lamp

The first lamp measured was the Phocos Pico lamp. Figure 4.2 shows the resulting 3D plot of its luminous intensity, while figure 4.3 shows the polar plots for a fixed polar and azimuth angle. Note that as the shape of the lamp suggest in figure 4.1, the LID is long and thin (concentrated).

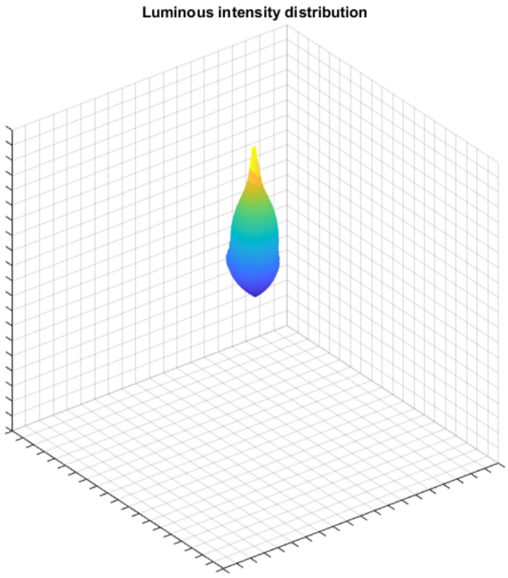

Figure 4.2: 3D representation of the LID of the Phocos Pico lamp.

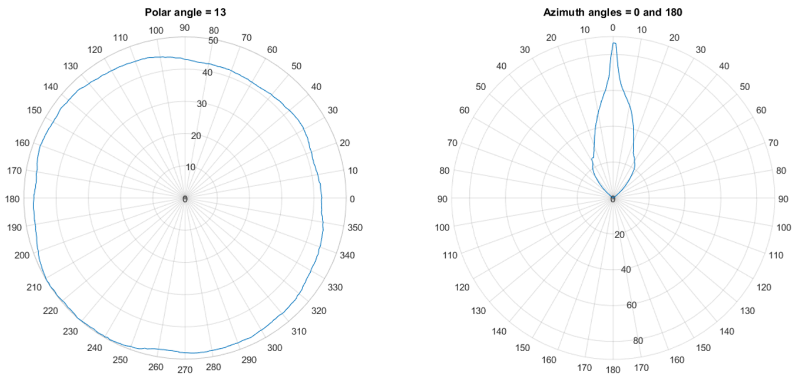

Figure 4.3: Polar plot for fixed angles of the LID of the Phocos Pico lamp.

LID Test 2: Phocos Pico lamp partially covered

For this test, the Phocos Pico lamp was partially covered with dark insulating tape as seen in figure 4.4. The results (figures 4.5 and 4.6) are quite interesting. First of all, compared to the results of the first test, the values of the luminous intensity are lower. This is trivial because the lamp is being covered. What is more interesting is the shape of the 3D plot of this test. At the zenith of the lamp, due to the rays coming from the sides, the lamp still has its maximum value of luminous intensity. In figure 4.6 (left) could be perfectly appreciated that for those angles corresponding to $\varphi = 90°$ and $\varphi = 270°$, the luminous intensity is the lowest, as those points are the ones that are being covered by the insulating tape. However, in figure 4.6 (right) can be seen local maximums at approximately $\theta = 35°$ for fixed angles $\varphi = 0°$ and $\varphi = 180°$. Those points are the ones at which the light finds a path between the photometric center and the sensor without encountering the insulating tape.

Figure 4.4: Phocos Pico lamp partially covered with insulating tape.

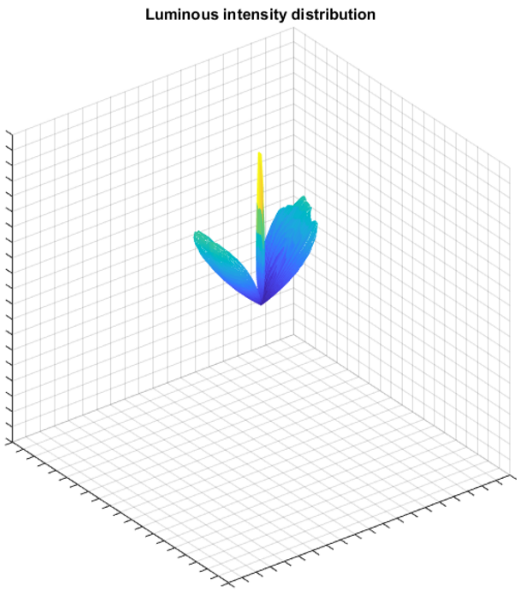

Figure 4.5: 3D representation of the LID of the Phocos Pico lamp partially covered with insulating tape.

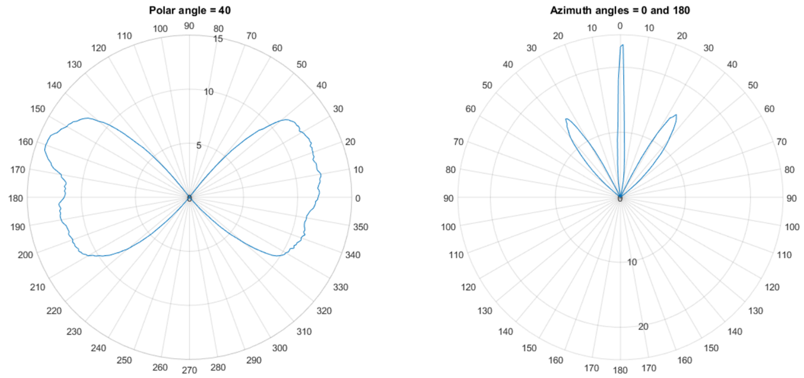

Figure 4.6: Polar plot for fixed angles of the LID of the Phocos Pico lamp partially covered with insulating tape.

The results of this test are conclusive enough to say that the system works as expected in terms of measuring and recording luminous intensity distribution. However, two more test were made in order to assure this assumption.

LID Test 3: d.light S300

Comparing the results of this test (figures 4.7 and 4.8) with the ones from the first test, it is possible to see two main differences. First, the 3D plot is shorter and wider, as the shape of the d.light S300 lamp suggested (figure 4.1). Second, the polar plot for a fixed polar angle seems to be bent (figure 4.8, left). This can be because of imperfections of the goniophotometer (a serious misalignment), or because that is the actual LID of the lamp. To make sure that the goniophotometer is correctly aligned, another test was made with the lamp turned 180°. The results (figure 4.9) show the exact same shape as before, but rotated 180°. Therefore, that is the actual LID of the lamp and the goniophotometer works perfectly.

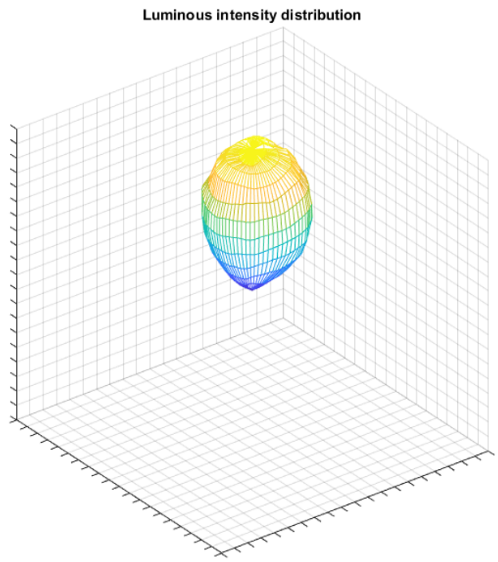

Figure 4.7: 3D representation of the LID of the d.light S300 lamp.

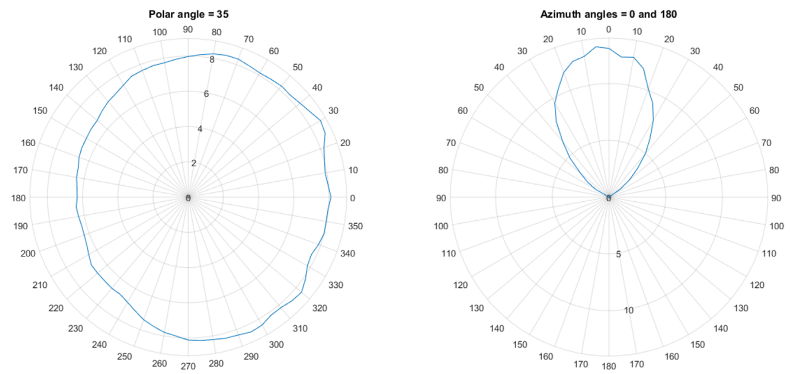

Figure 4.8: Polar plot for fixed angles of the LID of the d.light S300 lamp.

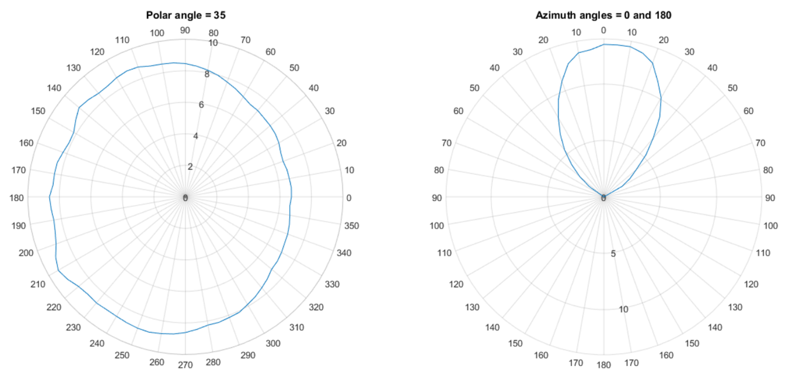

Figure 4.9: Polar plot for fixed angles of the LID of the d.light S300 lamp turned 180°.

LID Test 4: A light bulb

As a final test, a LED light bulb was measured. Unlike the previous tests, a light bulb should have a more uniform and wide distribution. In fact, it was predicted that the results were to resemble a sphere somehow (not exactly a sphere because as the lamp is turned to its back, the illuminance should decrease until it becomes zero). Figures 4.10 and 4.11 show that the results were satisfactory.

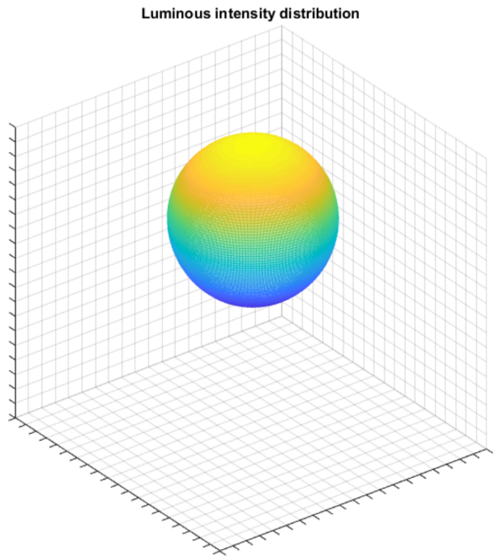

Figure 4.10: 3D representation of the LID of a LED light bulb from FD.

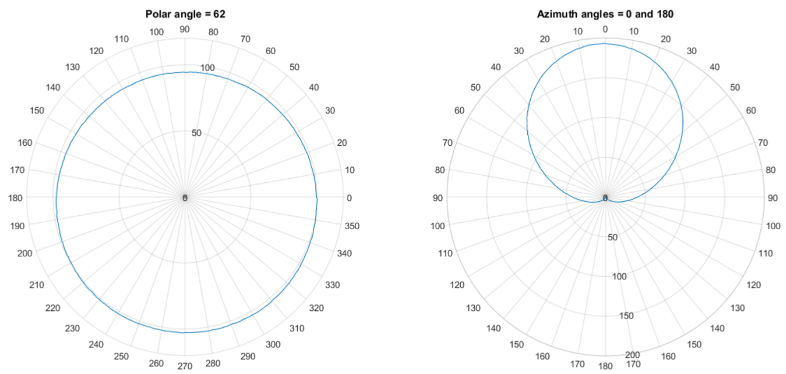

Figure 4.11: Polar plot for fixed angles of a LED light bulb from FD.

4.2 Luminous flux tests

As the graphical representations of the LID were satisfactory, the only thing left was to test if the actual *values* of the luminous intensity were correct. The aim of these tests was to check that the IES goniophotometer works properly by comparing the resulting luminous flux with those of another system.

The light sources measured during these tests were already measured by the CIEMAT (*Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas*) in 2014 with an integrating sphere[19]. The sample n°3 of four of the LED bulb models considered in the CIEMAT reasearch was measured. Moreover, the same labels were used in order to keep consistency: FD (Alcampo), IKEA, Carrefour and Getic-lighting (Hipercor). A picture of these light sources can be seen in figure 4.12.

Figure 4.12: LED bulb models measured by the CIEMAT and selected for this test.

Two sensing subsystems were used to assure the consistency of the results. The first sensing subsystem used was the MESA MS-LUX lux meter along with the MCP3422 ADC. Afterwards, it was substituted with a TSL2561 light sensor. All the measures where taken with a resolution of 1 degree per sample for both polar and azimuth angles. This is not the maximum resolution that can be used with the IES goniophotometer (it is 0.5°. See section 3.4), but it was chosen to save time when measuring[1].

[1]Each time the motors move, the whole system waits until it stabilize and finally the sample of illuminance is taken. The greater the resolution, the longer it will take to complete a measure of luminous flux.

Light reflection consideration

Before showing the results, it is important to mention that some of them were affected by the reflection of light on the floor of the goniophotometer. In such cases, that reflection was subtracted from all the samples of luminous intensity starting from the polar angle correspondent to the half maximum. Needless to say, this reflection does not necessarily affects all those samples, but that was the rule followed due to the uncertainty of the point from which those reflections started to affect the measure.

4.2.1 Sensing subsystem #1: MESA MS-LUX lux meter + MCP3422 ADC

A 16-bit resolution was used for the MCP3422, at which it has a data rate of 15 samples per second[20]. As a result, the use of this sensing subsystem has the advantage of measures not taking too long (4 hours of continuous measuring). In the next section you will see that subsystem #2 is worst in this aspect.

Table 4.1 shows the results for each LED bulb mentioned above. Four measures of the luminous flux were taken for the FD LED bulb and three for the IKEA's, Carrefour's, and Getic-lighting's. Note that the difference between CIEMAT's measures and the IES meassures is between 6% and 16% relative to CIEMAT's measures. However, as shown in table 4.2, the coefficient of variation (CV) of the measures is very low (less than 4% in all cases). This means that the results are very precise even though they are not accurate enough (the relative difference between them and the CIEMAT results is still above (or close to) 10% in most of the cases comparing the mean values).

Light source	CIEMAT luminous flux (lm)	IES luminous flux (lm)	Difference with CIEMAT's
FD	845.6	769.1	-9.0 %
FD	845.6	780.1	-7.7 %
FD	845.6	783.8	-7.3 %
FD	845.6	732.9	-13.3 %
IKEA	634.4	596.2	-6.0 %
IKEA	634.4	597.4	-5.8 %
IKEA	634.4	589.6	-7.1 %
CARREFOUR	1113.5	999.7	-10.2 %
CARREFOUR	1113.5	1004.1	-9.8 %
CARREFOUR	1113.5	990.0	-11.1 %
GETIC-LIGHTING	958.1	805.6	-15.9 %
GETIC-LIGHTING	958.1	828.9	-13.5 %
GETIC-LIGHTING	958.1	811.0	-15.4 %

Table 4.1: Results obtained using the sensing subsystem MESA lux meter + MCP3422.

Light source	Mean (lm)	SD (lm)	CV	Difference with CIEMAT's
FD	766.5	20.1	2.6 %	-9.4 %
IKEA	594.4	3.4	0.6 %	-6.3 %
CARREFOUR	997.9	5.9	0.6 %	-10.4 %
GETIC-LIGHTING	815.1	9.9	1.2 %	-14.9 %

Table 4.2: Statistical study of the measures taken with the sensing subsystem #1. SD = Standard deviation, CV = Coefficient of variation. The difference with CIEMAT's results were calculated using the mean values.

4.2.2 Sensing subsystem #2: TSL2561 Light sensor

As the lack of accuracy exposed above could be caused by the sensing subsystem used, it was replaced with one TSL2561 light sensor to see if the results were consistent. It is a very inexpensive option (see section 3.4.2), but if the results are similar to those of the first sensing subsystem, this subsystem as the cause of the lack of accuracy can be discarded. An integration time of 402 ms was used among the three possibilities[18] in order to achieve more precision, but the downside of this sensing subsystem is exactly that. Such a high integration time means longer measures (18 hours of continuous measuring). Moreover, The library available to use the TSL2561 along with Arduino only allows the calculation of luminous flux with the poorer resolution of 1 lx.

Table 4.3 shows the results of the measures using this second sensing subsystem. Note that the difference between the CIEMAT's measures and the IES measures is still in the range of 6% and 16%.

The statistics for this second sensing subsystem are shown in table 4.4. It seems like it has a little better accuracy comparing CIEMAT's measures with the mean values, but is not enough. However, this statistics show the same precision as before.

In conclusion, the lack of accuracy is not caused by the sensing subsystem. In order to find the cause of this lack of accuracy, a more intensive study of the IES goniophotometer is needed. There are a lot of physical factors that can contribute to it: The size and shape of the goniophotometer itself, the reflection of light on the floor of the goniophotometer, the measured distance between the photometric center and the sensor, etc. For the time being, it can be attributed to the corrections made due to the reflections on the floor of the goniophotometer as all the results are below the CIEMAT measures.

To conclude, table 4.5 shows the statistics taking into account both sensing subsystems. Even calculating the mean values and standard deviation of the two sensing subsystems together, the IES goniophotometer is still precise, but not accurate enough.

Light source	CIEMAT luminous flux (lm)	IES luminous flux (lm)	Difference with CIEMAT's
FD	845.6	780	-8 %
FD	845.6	782	-8 %
FD	845.6	778	-8 %
FD	845.6	770	-9 %
IKEA	634.4	584	-8 %
IKEA	634.4	608	-4 %
IKEA	634.4	615	-3 %
CARREFOUR	1113.5	1053	-5 %
CARREFOUR	1113.5	1032	-7 %
CARREFOUR	1113.5	1057	-5 %
GETIC-LIGHTING	958.1	818	-15 %
GETIC-LIGHTING	958.1	838	-13 %
GETIC-LIGHTING	958.1	888	-7 %

Table 4.3: Results obtained using IC TSL2561.

Light source	Mean (lm)	SD (lm)	CV	Difference with CIEMAT's
FD	778	5	1 %	-8 %
IKEA	602	13	2 %	-5 %
CARREFOUR	1047	11	1 %	-6 %
GETIC-LIGHTING	848	30	4 %	-12 %

Table 4.4: Statistical study of the measures taken with the sensing subsystem #2. SD = Standard deviation, CV = Coefficient of variation. The difference with CIEMAT's results were calculated using the mean values.

Light source	Mean (lm)	SD (lm)	CV	Difference with CIEMAT's
FD	772	16	2 %	-9 %
IKEA	595	8	1 %	-6 %
CARREFOUR	1023	26	3 %	-8 %
GETIC-LIGHTING	823	12	2 %	-14 %

Table 4.5: Statistical study of the results taking into account both sensing subsystems. SD = Standard deviation, CV = Coefficient of variation. The difference with CIEMAT's results were calculated using the mean values.

Chapter 5

Conclusion

It was a success to create a software that can collect, sort, process, and store illuminance data coming from a control device. KNDL not only transforms this data into luminous intensity values, but also can calculate the total luminous flux and the horizontal and vertical FWHM. Furthermore, as the only fixed element of the proposed architecture (KNDL architecture), KNDL does not depend on the operating system as long as it has a JVM installed. In consequence, the replicability requirement was successfully met as the rest of the elements can be adapted to their availability. Moreover, KNDL is freely distributed (see appendix A for explanations on how to download it), so its acquisition is not an issue.

On the other hand, the proposed architecture makes it somewhat difficult to implement the control device. Regarding this issue, the Arduino code used for the IES goniophotometer's implementation is distributed alongside with KNDL. It is a fact that the ease of use of Arduino's libraries has made it a typical choice for open projects all over the world, which is why it has been broadly distributed, making it simple to acquire or to make (as it is open hardware). Thus, it should not be difficult to adapt the provided code to other implementations. Furthermore, if the MCP3422 or the TSL2561 ICs and H-bridge motor drivers are used, the provided Arduino code can be used without any adaptation. As for the rest of the elements of the architecture, they can be selected depending on their availability in the geographical location where the system is being implemented.

In relation to the implementation, most of the elements selected were as inexpensive as possible (see section 3.4.2) except for the only element imposed by the IES to be reused (the MESA MS-LUX lux meter). Nevertheless, as an ADC is needed, any other analog-output lux meter can be used. Besides, a second option for a sensing subsystem was also implemented using the TSL2561 IC. In fact, it is interesting to see that despite the difference in cost and resolution between the two sensing subsystems, they both deliver similar results when it comes to measuring luminous flux. This must be a consequence of the integration of a large quantity[1] of samples to get the luminous flux. Additionally, the fact that the conditions of a goniophotometer are strictly controlled (only one light source in a dark-indoor environment) may also account for these similar results.

Concerning the LID results, they were visualized using MATLAB scripts because they provided an easier way to check whether the measured luminous intensities made sense. This was achieved by comparing the shape of the light sources with the 3D plot and polar plots for fixed angles of their LIDs. Moreover, even though these were qualitative tests, the polar plots obtained with those MATLAB scripts are valid to characterize the light sources as they specify the luminous intensity vs the angle from where light source is viewed. These plots can also be obtained using spreadsheets and KNDL's resulting CSV file. However, the IES goniophotometer must be improved for these results to be good enough (because of the accuracy issue).

As for the results of the luminous flux calculation, they show precise, but not accurate enough measures. However, with a few modifications of the IES goniophotometer, it could still be possible to make an affordable, space-saving goniophotometer. One idea to achieve better accuracy by modifying the infrastructure is to raise the turning system more from the floor. This way the path of light reflected on the floor of the goniophotometer will be longer and may be attenuated before reaching the sensor. The downside of this option is that it diminishes the maximum size of lamps that can be measured with the IES goniophotometer (see section 3.1). Another option could be to place a dark platform with a hole in the middle somewhere between the light source and the sensor. This way only the direct light will reach the sensor. The downside of this second option is the size of the hole: it must be big enough to let all the direct light pass through it, but small enough not to let the reflections pass, so its size varies from one light source to another.

To sum up, the objectives of the project regarding the design and implementation of a replicable, automated measuring system were successfully met. However, the results show that there are still improvements to be made for the IES goniophotometer's infrastructure to achieve better accuracy of the overall system. Thus, further research must be done.

[1]For an angular resolution of 1 degree per sample for both polar and azimuth angles, the results are comprised of 64,442 luminous intensity samples

5.1 Future research and possible improvements

Below is a list of possible future research and possible improvements:

- The conclusions were drawn taking the CIEMAT measures as a reference. It would be interesting to compare the results with other systems. Moreover, the IES goniophotometer could be tested against a standard lamp (i.e. a lamp with a defined, stable luminous flux).
- Improvement of KNDL by adding the possibility to generate plots similar to those created with the MATLAB scripts from chapter 4. A nice idea to achieve this is to use Python's package NumPy to do the work. Interprocess communication can be used between Java and Python to exchange data, or Python could simply read it from the resulting CSV file (under the command of Java).
- Essentially, KNDL's main functionality is to collect, sort and store the data that comes from a control device. That data includes a spherical coordinate and the value of a physical quantity (luminous intensity). In consequence, it is possible to easily adapt KNDLS's code to create programs that can characterize the directivity of other transducers, such as speakers or antennas. In fact, the change of some labels should be enough to adapt KNDL to such circumstances. Furthermore, the Arduino code and the MATLAB code can also be reused. Also, by inverting the roles and using standard radiators, one can even characterize the angular dependency of sensors.
- By using a Raspberry Pi as a control device and running KNDL into it, one can combine the computer and control device's functionality into one device. The downside of this solution is that it is more difficult to set up the environment of a Raspberry Pi than it is to just install Java in a computer and upload the code to an Arduino Board. After all, KNDL architecture was designed to be implemented anywhere and the person who implement it may not be a programmer or know much about computers.
- Even though it was not mentioned until now, the black curtains of the IES goniophotometer were enough to keep it from outside light, but as long as it was not direct sunlight. This is not relevant for this document because this problem was avoided by simply closing the blinds at the hours in which the goniophotometer was exposed to direct sunlight. However, an extra layer of opaque-curtains must be used to suppress this situation.

Bibliography

[1] IEC, "IEC TS 62257-9-5 TECHNICAL SPECIFICATION," 2013.

[2] M. S. Rea, *The IESNA Lighting Handbook: Reference and Application*. Illuminating Engineering Society of North America, 9th ed., 2000.

[3] A. E. F. Taylor, *Illumination Fundamentals*. Rensselaer Polytechnich Institute, 2000.

[4] RS Agencies, "Lumens for the laymen," 2015.

[5] CIE, "eilv.cie.co.at," 2014.

[6] Lisun Group, "Basic Theories About Goniophotometer," 2011.

[7] CIE, "THE MEASUREMENT OF ABSOLUTE LUMINOUS INTENSITY DISTRIBUTIONS," 1987.

[8] Instrument Systems GmbH, "The goniometer types A / B / C."

[9] G. Leschhorn and R. Young, "Understand how to measure luminous flux and radiant power," *LEDs MAGAZINE*, vol. 13, no. 8, 2016.

[10] Instrument Systems GmbH, "LEDGON 100 Datasheet."

[11] OxyTech, "Goniophotometer T3 Datasheet."

[12] Viso Systems, "www.visosystems.com."

[13] X -Rite GmbH, "GONIOPHOTOMETER SYSTEMS," 2005.

[14] Instrument Systems GmbH, "Goniophotometer Systems."

[15] Radiant Zemax LLC, "Large Light Source Performance Characterization PM-NFMS $^{\text{TM}}$," 2012.

[16] F. Sametoglu, “Construction of two-axis goniophotometer for measurement of spatial distribution of a light source and calculation of luminous flux,” *Acta Physica Polonica A*, vol. 119, no. 6, 2011.

[17] L. Olivieri, *INTEGRAL ENERGY BEHAVIOUR OF PHOTOVOLTAIC SEMI-TRANSPARENT GLAZING ELEMENTS FOR BUILDING INTEGRATION*. PhD thesis, Universidad Politécnica de Madrid, 2015.

[18] TAOS, “TSL2561 LIGHT-TO-DIGITAL CONVERTER,” 2009.

[19] F. García Rosillo, “INFORME BOMBILLAS LED EQUIPADA CON CASQUILLO E-27 EQUIVALENTES A INCANDESCENTE DE 60W COMERCIALIZADAS BAJO MARCAS BLANCAS Y MARCAS GRANDES SUPERFICIES,” 2014.

[20] Microchip Technology Inc., “MCP3422/3/4 Datasheet,” 2008.

Appendix A

KNDL user guide

KNDL, pronounced 'candle', is a software that runs on a computer as part of an automated goniophotometer. The components of the goniophotometer must implement KNDL Architecture. For more information about KNDL Architecture, go to: `https://github.com/abdonvivas/KNDL`

Downloading KNDL

First, you must have the Java Runtime Environment (JRE) installed. Go to `https://java.com/` and download the JRE from there. Then, download KNDL from `https://github.com/abdonvivas/KNDL`.

Executing KNDL

Windows and Mac

Unzip the file you downloaded and then double-click the file KNDL.jar.

Linux

Open a console and navigate to the folder where you unzipped the file you downloaded. Then, type:

```
$ java -jar KNDL.jar&
```

Main window

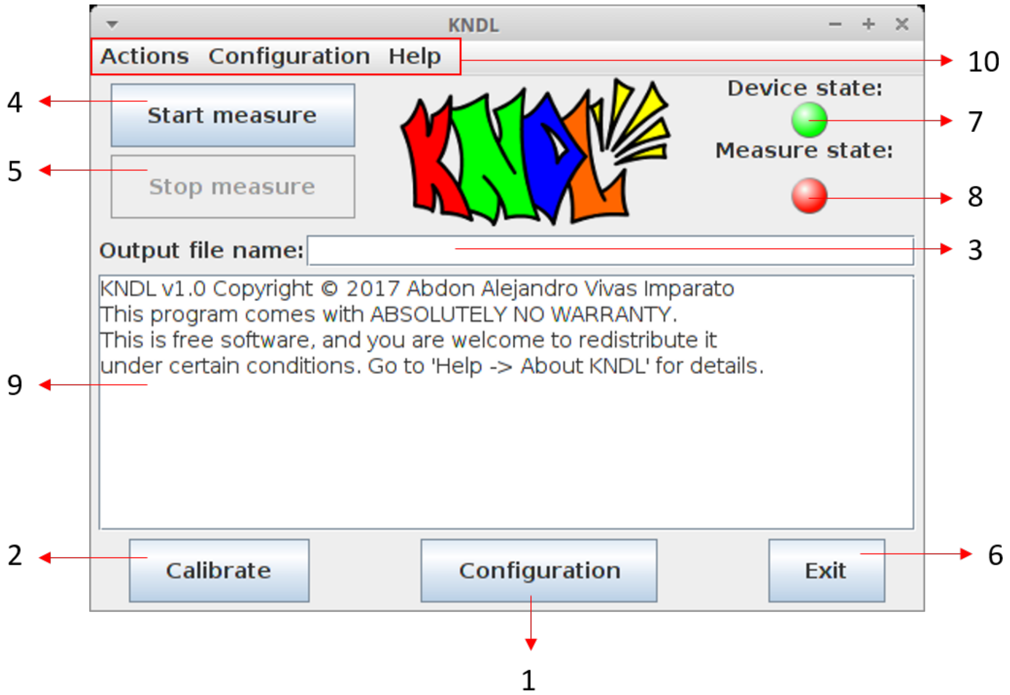

Figure A.1: KNDL's main window.

1. Configuration button: Opens the configuration window.

2. Calibrate button: Starts a calibration process.

3. File name text field: To specify the name of the output file.

4. Start button: Starts a measure process.

5. Stop button: Stops a measure in progress.

6. Exit button: Exits KNDL. This button is disabled during a measure process to avoid closing KNDL by accident. However, KNDL can still be closed via the Actions menu in the menu bar.

7. Device state indicator: Indicates whether a control device is assigned (green) or not (red).

8. Measure state indicator: Indicates whether a measure is in process (green) or not (red).

9. Monitor: Shows the progress of a measure or a calibration process. It also outputs whatever is written from the control device to the serial port during those processes.

10. Menu bar.

Menu bar

Actions Menu

To alternatively start or stop a measure, or exit KNDL.

Configuration menu

To open the configuration window or use advanced functions.

Help menu

Displays information about KNDL.

Configuration window

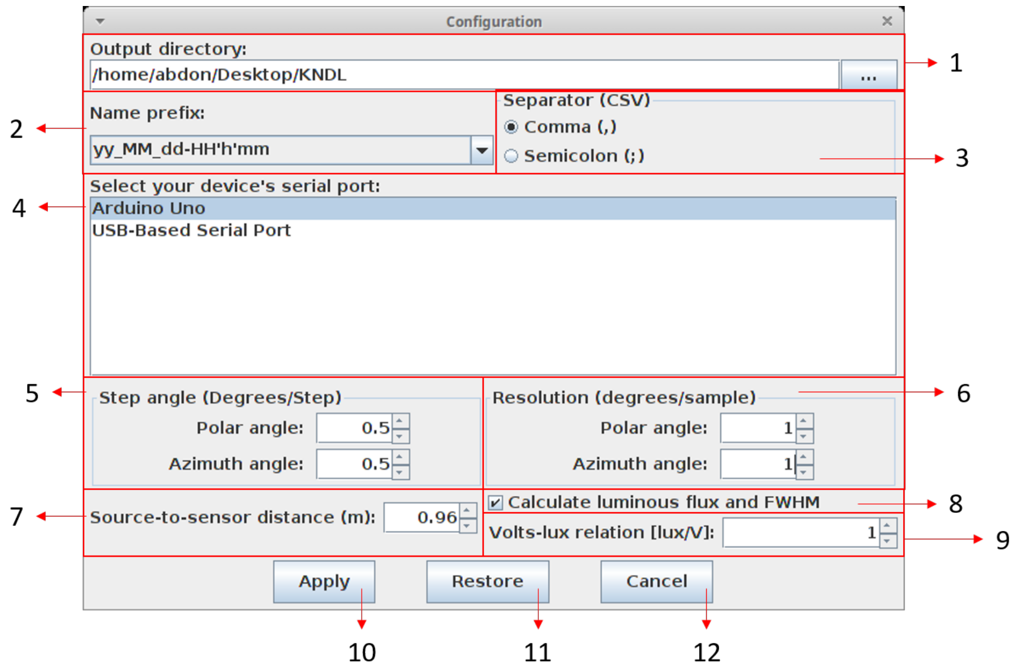

Figure A.2: KNDL's configuration window.

1. Output directory text field: Specifies the directory where you want the output file to be stored. Press the button with the ellipsis to open a file browser dialog.

2. Prefix of the output file name drop-down list:

   - yyyy_mm_dd-HH'h'mm e.g. 2017_01_26-13h30
   - yy_mm_dd-HH'h'mm e.g. 17_01_26-13h30
   - HH'h'mm e.g. 13h30

3. Separator of the CSV file panel. If your decimal mark is a comma, then it would be better if you use a semicolon as a separator.

4. List of the available serial ports. To select the serial port of your control device.

5. Step angle. To specify your goniophotometer's step angle, in degrees per step, for each of the axis. This is not the step angle of your motors, but the total step angle if you are using a power transmission system.

6. Measure resolution. To specify the desired resolution of the measure, in degrees per sample, for each of the axis. The time the system is going to be measuring and the accuracy of the measure depends strongly on these values.

7. Source to sensor distance input. To specify the measuring distance.

8. Calculations checkbox. If you don't want KNDL to make any calculations, unmark this checkbox.

9. Volts-to-lux relation. Specify the relation, in lux per volt, between lux and volts. If your control device sends the illuminance directly instead of a voltage, set this value to 1.

10. Apply button. Once you modify any of the properties, they won't apply until you press this button.

11. Restore all the properties to their default values. They won't apply unless you press the Apply button after restoring them.

12. Cancel button. To exit the configuration window without saving the changes.

Calibration process

To start a calibration process, turn off the light source and press the Calibrate button in the main window. The result of the calibration process is taken as an offset to be subtracted from all the samples taken during a measure.

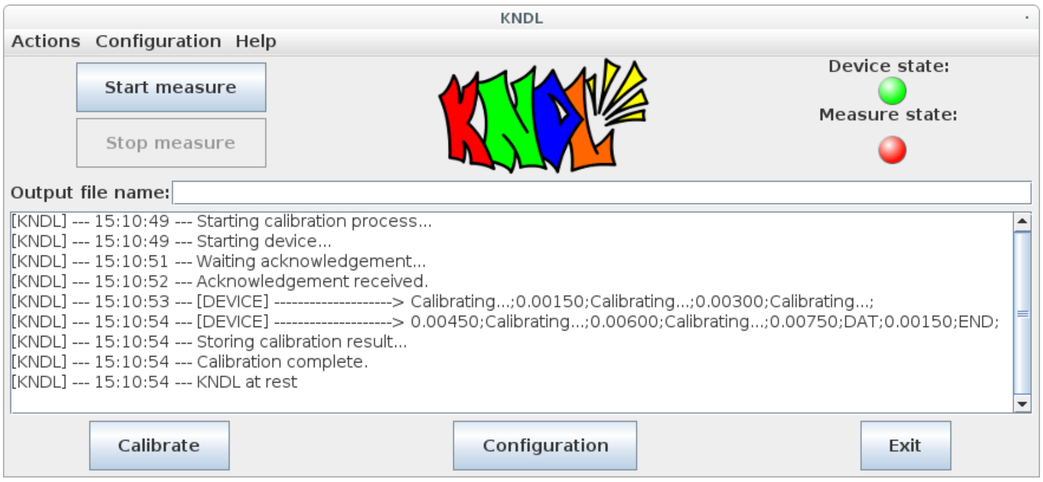

Figure A.3: KNDL's calibration process screenshot.

Measure proccess

After selecting the right configuration for the measure (see Configuration window section) and calibrating (see Calibration process section), you can start a measure. Select a name for the output file (e.g. 'LightSource1'), turn on the light source, and press the start button in the main window. The progress of the measure and all the messages sent from the control device, including the responses (figure A.4 and figure A.5) will appear in the monitor.

Once the measure is over, you can go to the output directory you selected in the configuration window and open the resulting CSV file (figure A.6).

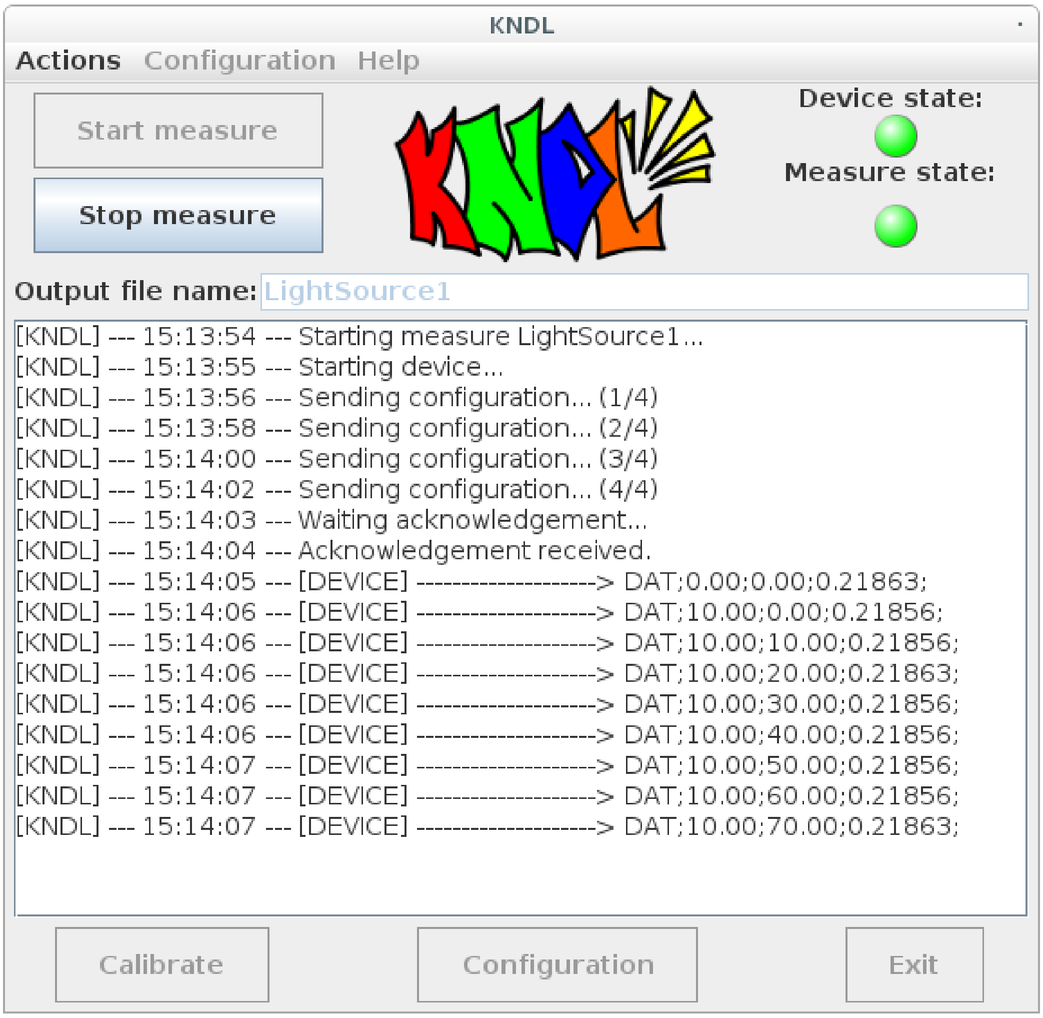

Figure A.4: Screenshot of the starting of a measure.

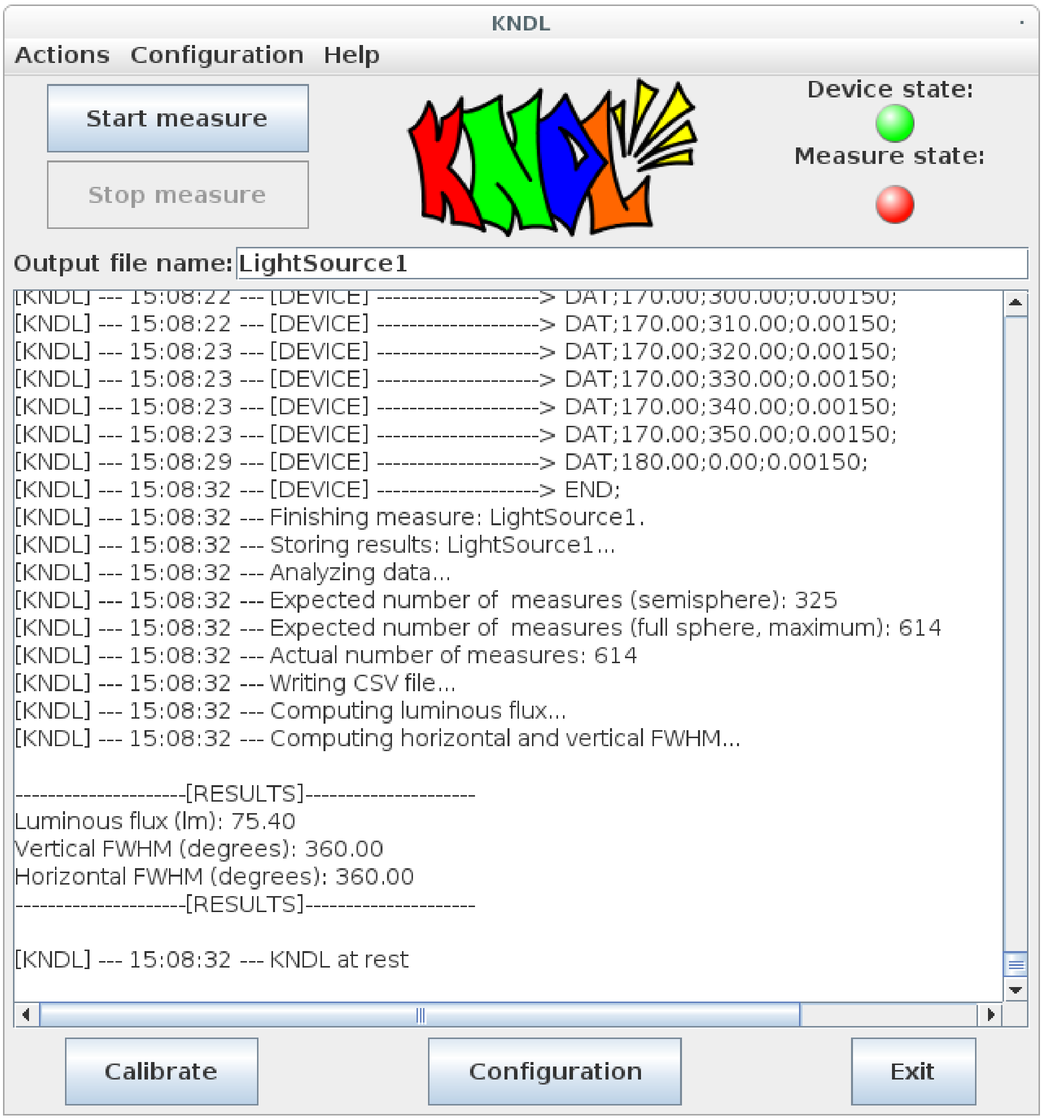

Figure A.5: Screenshot of a finished measure.

	A	B	C
1	Luminous flux (lm):	784.2	
2	Vertical FWHM (degrees):	122	
3	Horizontal FWHM (degrees):	121	
4	Source-to-sensor distance (m)	0.727	
5	Calibration offset (V)	0.00186	
6			
7	Polar angle (°)	Azimuth angle (°)	Luminous intensity (cd)
8	0	0	203.8219191
9	1	0	204.0861836
10	1	1	203.9223396
11	1	2	204.0544718
12	1	3	204.0544718
13	1	4	204.1548923
14	1	5	204.0544718
15	1	6	204.1178953
16	1	7	203.9857631
17	1	8	203.9857631
18	1	9	204.0227601
19	1	10	204.0544718
20	1	11	203.9857631
21	1	12	204.0544718

Figure A.6: Screenshot of a resulting CSV file.

Advanced functions

You can access the advanced functions by opening the configuration menu in the menu bar and selecting the ‘Advanced configuration’ item. The advanced functions are the following:

Reset the offset value

Reset the offset (calibration result) to zero.

Store raw files

To store an additional CSV file in a subdirectory called ‘RAWS’ in the output directory. The additional CSV file contains no explanation of the results, nor the result of the calculations, but only the coordinates and illuminance of each sample taken in a measure (figure A.7). The first column is the polar angle, the second column is the azimuth angle, and the third column is the luminous intensity value measured in that position. This is useful if you want to use the data to generate extra results such as plots using other programs (e.g. MATLAB or python). For example, the plot shown in figure A.8 was generated by processing a raw file using MATLAB.

	A	B	C	D
1	0	0	193.084523536	
2	1	0	192.9977499747	
3	1	1	193.0377993107	
4	1	2	193.0377993107	
5	1	3	192.9577006388	
6	1	4	192.9577006388	
7	1	5	192.9577006388	
8	1	6	193.0377993107	
9	1	7	192.9577006388	
10	1	8	193.0377993107	
11	1	9	193.0377993107	
12	1	10	193.084523536	
13	1	11	193.0377993107	
14	1	12	193.084523536	
15	1	13	192.9577006388	
16	1	14	192.7908284056	
17	1	15	192.9577006388	
18	1	16	192.9577006388	
19	1	17	193.2046715438	
20	1	18	193.1646222079	

Figure A.7: Example of a raw file.

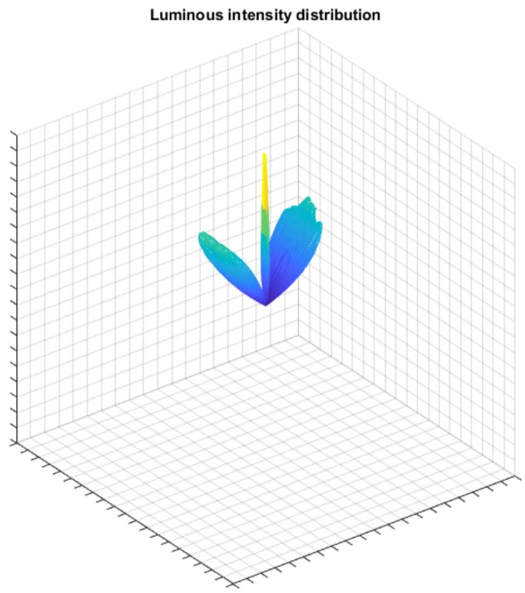

Figure A.8: 3D representation of a lamp partially covered with insulating tape.

Store log files

To store the contents of the monitor in a log directory located in the KNDL root directory at the end of each measure. This is useful as each measure is supposed to take a long time and if something goes wrong, then you can trace back the error by looking at the log file. For example, if for some reason you want to know the exact hour a sample was taken in a specific position, you can see it in the log file. Also, if something goes wrong and the JVM or the operating system crashes before storing the final results, you can still recover the measures from the log file as it is being updated continually.

Change the default device

If you want KNDL to recognize your control device each time KNDL starts, change the regular expression KNDL uses to find a default control device to one that matches your device's description or the serial port to which your control device is connected.

Set current properties as default properties

As KNDL is an open source software, the default configuration is more like a developer configuration. Once you specify some static parameters of your goniophotometer as the system resolution or the volts-to-lux relation, you may want those values to be the default values so every time you press the Restore button in the configuration window they won't restore to the developer's values.

Before selecting this option, make sure the current configuration is the one you want to use by default. See Configuration window section for more information.

License

KNDL, pronounced 'candle', is a software that runs on a computer as part of an automated goniophotometer. The components of the goniophotometer must implement KNDL Architecture. For more information about KNDL Architecture, go to: `https://github.com/abdonvivas/KNDL`

List of Figures

1.1 IES Goniophotometer. 2

2.1 Illustration of the luminous flux, luminous intensity, illuminance, and luminance[4]. 5
2.2 Illustration of the inverse square law[2]. 6
2.3 Types of goniophotometers. Images taken from [7]. 7
2.4 Spherical coordinate system convention used during this project. 8

3.1 Turning system of the IES Goniophotometer. 10
3.2 KNDL architecture diagram. 11
3.3 Protocol sequence of a successful and a stopped measure process. 14
3.4 Protocol sequence of a calibration process. 15
3.5 KNDL main window. 15
3.6 Graphical representation of the luminous flux calculation algorithm. (1) All the points in which the sensor passes through. (2) The points where a sample is taken are marked. (3) Shares of the semi-sphere marked. (4) Rings colored and one slice emphasized. (5) Graphical representation of the calculation to be made in order to get the luminous flux from the illuminances taken and the area of each share. 18
3.7 Implementation of KNDL architecture used for the IES goniophotometer. 19
3.8 Test similar to the one used to chose the ADC for the IES goniophotometer. 20
3.9 Reference subsystem. Left: Position not aligned with the reference. Right: Position aligned with the reference. 21
3.10 Detailed view of the motors and the belt drives. 22
3.11 L298 modules. 23
3.12 Connection diagram. V_L is the logic voltage, V_M is the motor voltage supply, and V_{LUX} is the MS-LUX lux meter voltage supply. $C1 = C2 = C3 = 0.1\mu F$, $R_{SLED} = R1 = R2 = 200\Omega$, $V_L = 5V$, $V_M = 4V$ and $V_{LUX} = 18V$. 24

3.13 Flow diagram of the control device. 26

4.1 Phocos Pico lamp (left) and d.light S300 lamp (right). 28
4.2 3D representation of the LID of the Phocos Pico lamp. 29
4.3 Polar plot for fixed angles of the LID of the Phocos Pico lamp. 29
4.4 Phocos Pico lamp partially covered with insulating tape. . . . 30
4.5 3D representation of the LID of the Phocos Pico lamp partially covered with insulating tape. 30
4.6 Polar plot for fixed angles of the LID of the Phocos Pico lamp partially covered with insulating tape. 31
4.7 3D representation of the LID of the d.light S300 lamp. 32
4.8 Polar plot for fixed angles of the LID of the d.light S300 lamp. 32
4.9 Polar plot for fixed angles of the LID of the d.light S300 lamp turned 180°. 32
4.10 3D representation of the LID of a LED light bulb from FD. . . 33
4.11 Polar plot for fixed angles of a LED light bulb from FD. . . . 33
4.12 LED bulb models measured by the CIEMAT and selected for this test. 34

A.1 KNDL's main window. 44
A.2 KNDL's configuration window. 45
A.3 KNDL's calibration process screenshot. 47
A.4 Screenshot of the starting of a measure. 48
A.5 Screenshot of a finished measure. 49
A.6 Screenshot of a resulting CSV file. 50
A.7 Example of a raw file. 51
A.8 3D representation of a lamp partially covered with insulating tape. 52

List of Tables

3.1 Commands sent from KNDL to the control device. 13
3.2 Responses sent from the control device to KNDL. 14
3.3 BOM of the main components of the IES goniophotometer's automated measuring system. 25
3.4 Arduino possible states. 25

4.1 Results obtained using the sensing subsystem MESA lux meter + MCP3422. 35
4.2 Statistical study of the measures taken with the sensing subsystem #1. SD = Standard deviation, CV = Coefficient of variation. The difference with CIEMAT's results were calculated using the mean values. 36
4.3 Results obtained using IC TSL2561. 37
4.4 Statistical study of the measures taken with the sensing subsystem #2. SD = Standard deviation, CV = Coefficient of variation. The difference with CIEMAT's results were calculated using the mean values. 37
4.5 Statistical study of the results taking into account both sensing subsystems. SD = Standard deviation, CV = Coefficient of variation. The difference with CIEMAT's results were calculated using the mean values. 37