



Master Degree in Electrical and Computer Engineering
2021/2022 – Spring Semester

**Distributed Real-Time Control Systems
(Sistemas de Controlo Distribuído em Tempo-Real)**

PROJECT DESCRIPTION AND PART 1 MILESTONES
(First 3 lab sessions)

Real-Time Cooperative Decentralized Control of an Illumination Systems with a network of Raspberry Pi Pico microcontrollers.



Prepared by

Alexandre Bernardino, João Pedro Gomes, Ricardo Ribeiro



Instituto Superior Técnico

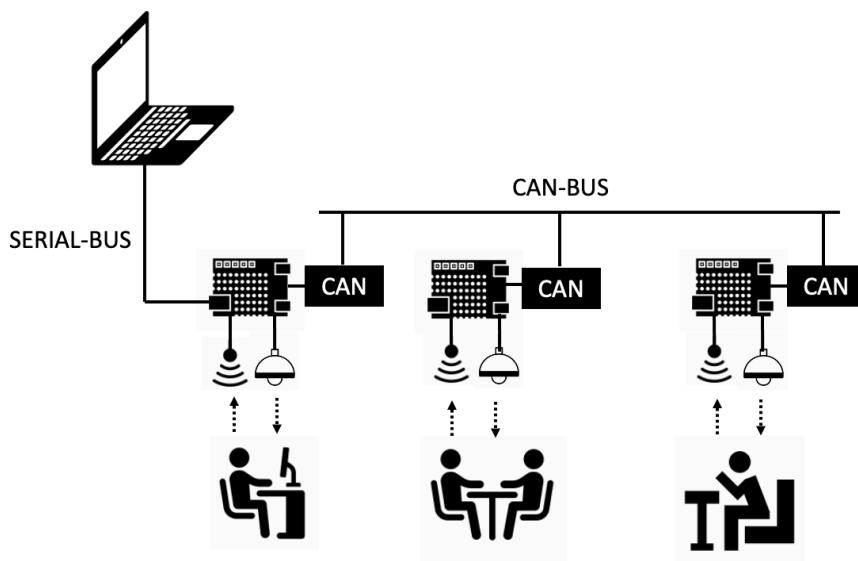
Department of Electrical and Computer Engineering

Scientific Area of Systems, Decision and Control

Version 1.0

Abstract

The objective of this project is to design a real-time control system for a distributed illumination system in a small-scale model of an office space. Conceptually, each desk is served by a smart luminaire installed in the ceiling immediately on top of it. The luminaire has a light emitting source to illuminate the desk, a luminance sensor to measure the light reflected from the desk, a presence sensor to determine if the desk is occupied or free, and computational and communication elements to perform the control of the system. In the project we will simulate the office with a small opaque cardboard box. Each luminaire consists of a breadboard with a microcontroller that drives a Light Emitting Diode (LED) and measures illuminance with a Light Dependent Resistor. All luminaires will be connected in a network via a communication bus to allow message exchange. The objective of the project is to control the dimming level of each LED in a coordinated way to maximize the comfort of the users while minimizing the energy consumption.



Logistics

There are weekly laboratory sessions of 3hr each where students will receive guidance from the teaching staff to execute the project.

In the first lab session, students get individual kits with the equipment for the execution of the project that they can take home and keep until the end of the period. This will allow more autonomy and time of contact with the project's hardware.

In the remaining lab sessions, students must bring the equipment to demonstrate their progress and learn best practices for executing the different phases of the project.

The equipment must be returned by the end of the semester after all assessments have finished.

Important notes:

1. It is recommended that students use their own laptops in the lab for interfacing with the project's hardware. This facilitates switching from work at home to work in the lab.
2. In the first lab session bring a plastic bag to carry the take-home equipment.
3. In the second lab session bring a shoe box (or similar) to assemble the office model.

Project Schedule

The project is divided in two phases. Each phase will have a duration of 3 lab sessions.

In the first phase (from March 7th 2022 to March 25th 2022) each student will work individually in the construction of one of the luminaires of the distributed system and in the development of its local controller. In the end of this phase each student will deliver a written report and the developed code.

In the second phase, students will work collaboratively in groups of at most three students to develop the cooperative distributed controller. Each student of the group may specialize in a particular component of the system and be evaluated for that: distributed control or communications or, concurrent programming. A final demonstration of the project will be made in the exam preparation week. The developed code must also be delivered.

Important note:

Although the second phase of the project and final demonstration will be made in group, each student must take responsibility for a fair part of the work.

Project Evaluation

The grade of the project is computed as the average of the mid-term report and the final demonstration. It has a minimum grade of 9.5 for approval.

The mid-term report has a maximum of 10 pages and will be delivered at week 5 (April 4th 2022).

The final demonstration will be made at the end of the exam preparation week (April 26th – 29th). The contribution of each student to the work done should be clearly stated. Grades may be individualized if the quality is inhomogeneous throughout the group.

Software, hardware schematics and other material developed to execute the project should be submitted jointly with the mid-term report and final demonstration, within the prescribed time limits.

The assiduity and punctuality of the student in the lab sessions is also considered for assessment purposes. Unjustified absences may be penalized.

Important Note:

The reports and the software developed must be original. All forms of plagiarism will be pursued to the full extent of IST / UL regulations and the Portuguese law.

Take home Kits

- 3 RPI pico microcontrollers
- 3 GPIO Headers for RPI pico
- 3 USB 2.0 A – micro USB cables for RPI pico
- 3 Small Breadboards
- 3 Light Emitting Diodes 10mm 85000-100000 mcd
- 3 Light Dependent Resistors 150 ... 300kOhm
- 3 Resistors 47 Ohm for LED driving circuit
- 3 Resistors 10 kOhm for LDR reading circuit
- 3 Capacitors 1 microF for LDR reading circuit
- 2 Female Jumpers for IO shunts
- 2 Male 2-pin Headers for IO shunts
- 20 Male-Male Jumper Wires
- 1 USB 2.0 Hub, 4 ways
- 1 Soldering Kit (Soldering Iron, Support Base w/ sponge, Desoldering Pump)
- 1 Multimeter
- 1 Tool set

Important notes:

1. In case of gross misuse of the equipment or violation of its operational limits, you will be requested to replace the damaged equipment.
2. You must always bring to every lab session the equipment kits (except the soldering kits) since it is required for the lab work.

Project Description

1. Introduction

With the surge in electricity prices, a large research effort has been devoted during the last decade to the development of efficient illumination systems. Advances in semiconductors have brought us high power LEDs that allow up to 85% saving in energy consumption and high versatility of use due to their small size and dimming ability. Additionally, improvements in technology and reductions in price are making LED the favourite illumination devices for homes, cars, offices, cities, and smart building spaces (www.ledmarketresearch.com).

The flexibility of LED lighting is powering another recent trend in energy optimization and comfort control. Recent research is being devoted to adaptive and distributed lighting control systems that account for the occupation status of spaces and the intensity of external illumination [1][2][3]. The power used to achieve the required comfort luminance levels for occupied spaces can be controlled with cheap luminance and presence sensors, while reducing it in unoccupied spaces. Lights in streets, buildings and public spaces are already being turned on and off using motion detection sensors, but more versatile systems can be developed to consider external illumination and jointly coordinate the activation of multiple interacting luminaires. In this project we will consider an office-like scenario where desks have presence and luminance sensors, and luminaires have light dimming and communication abilities to synchronize with their neighbours.

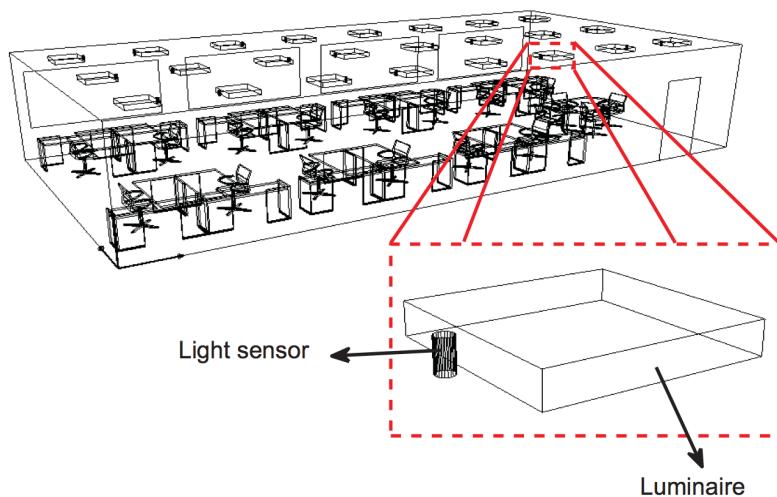


Figure 1. Scenario envisaged in the project. Each desk is served by a luminaire containing a light sensor, presence sensor and communication links to its neighbours. Note that the sensor measures the reflected light in the table and not the direct light from the lamp. The control of the lighting attains fixed levels of illumination at the desk plane (high for occupied desks and low for unoccupied desks) while minimizing the global energy consumption and considering the daylight illumination and disturbances from neighbouring luminaires.

2. Summary of the Project

Stages. In the first stage of the project each student will create a model of a small office with 1 luminaire. In the second stage, each group will extend the system to 3 luminaires. For practical reasons, the presence sensors will not be physically implemented, but instead simulated by setting variables in the microcontrollers through a computer interface. A small window should be simulated by creating an opening in the box, to exploit energy harvesting and simulate external disturbances.

Objective. The objective of the project is to minimize the energy consumption and maximize user comfort. Energy minimization will be achieved by controlling the dimming level of each LED such that occupied desks have luminance levels above a certain value (HIGH) and unoccupied ones have a luminance level above a lower value (LOW). User comfort should be maximized by keeping the illumination always above or equal to the minimum levels (visibility), while minimizing the up-and-down variations of the illuminance (flicker) during desk occupation. These variations may be due to noise, external disturbances, or interference caused by the other luminaires in the shared space. Noise and external disturbances can be compensated by a local feedback control loop at each luminaire, but internal disturbances due to interference from other desks can be predicted and compensated through proper communication and synchronization between luminaires (global control).

3. The Plant

Luminaire nodes. Each luminaire will be simulated with the provided equipment. The following diagrams illustrate a possible LED driver circuit and an illuminance reading circuit.

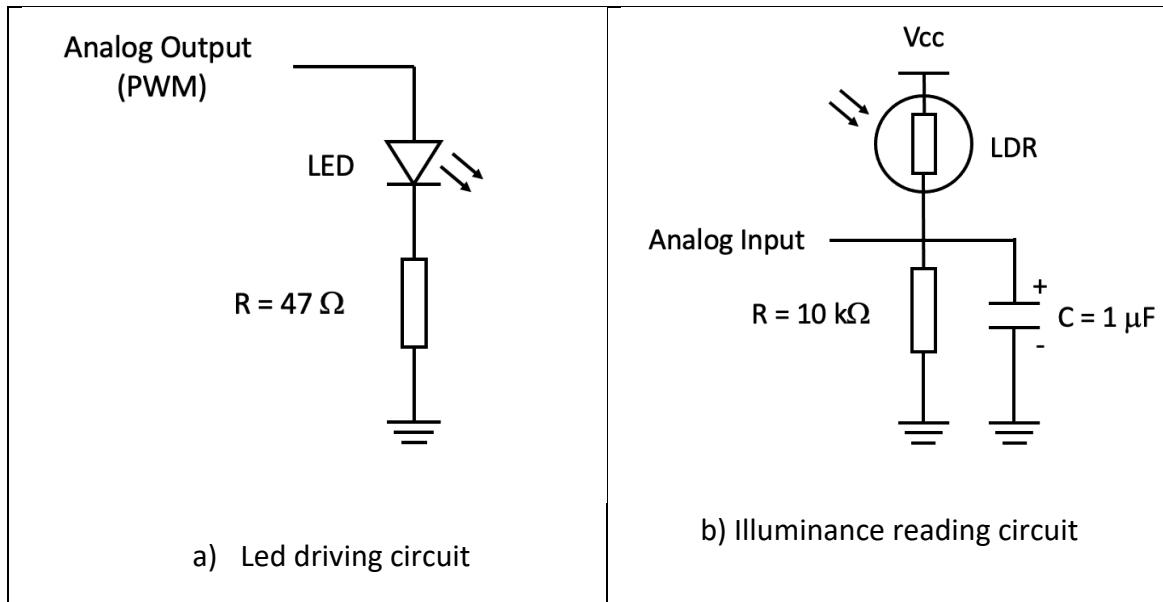


Figure 2 – Schematics for the luxmeter (a) and LED driver (b) circuits.

Dimming of the LED is implemented via PWM in one of the digital output ports. The illuminance can be measured via the LDR in a voltage divider circuit. The capacitor in the voltage divider helps to reduce the noise in the sensing circuit. The PWM frequency should be configured to further reduce the noise on the analog input. Both the LED and the LDR should be pointing “vertically” - the sensor should measure the reflected light in the desk and not the direct light from the lamp.

Office Model. The illumination system should be implemented in a reduced scale model. An opaque box with a cover should be used to completely block the external illumination. The box should be large enough to contain the 3 luminaires, but not too large. If the box is too large, the LED intensity may be insufficient to properly illuminate the LDR (note that the LDR receives mostly light reflected on the sides of the box). To improve light reflection, if needed, the interior of the box may be covered with white paper. Small openings in the box should be made to pass cables and wires. Try to insulate these openings as much as possible to prevent uncontrolled light from entering the box. An opening in the box should be available to let external light get inside the box and test the system reaction to external light.

4. First Stage

4.1. Description

During the first stage of the project, students will implement the luminaire and its local control system and test it in the reduced scale office model. The local controller is unaware of the existence of other luminaires and has the objective to keep the illumination level as close as possible to the desired value, despite disturbances due to variations in external illumination, cast shadows or changes in the light reflection paths. The controller should both have a fast response to changes in the desired illumination level and avoid flicker and overshoot in response to disturbances and noise. Because these objectives are often conflicting when using a feedback controller, it is recommended to also use a feedforward component.

Each luminaire can be decomposed in the following modules: (i) the illuminance measurement system (luxmeter), (ii) the LED actuation system (driver), (iii) the individual luminaire controller (local controller), (iv) a calibration procedure to identify the office model system response (controller calibration), and (v) a simple interface with a PC to receive commands and send data for analysis.

The Illuminance Measurement System (Luxmeter)

The purpose of this system, described in Fig. 2b), is to measure the illuminance in LUX units. The LDR is a non-linear element, i.e., its gain (ratio of the variation of the illuminance to variation of the measured voltage or current) varies with the operating point. However, its relationship to the standard illuminance unit — the LUX — is known. The LDR readings should be converted to LUX units with the help of the LDR characteristic response in the datasheet. Note that the LDR datasheet indicates a range of resistance values for each illumination intensity (see Fig. 3). Consider the middle of the range specified in the data sheet.

Illuminance Vs. Photo Resistance

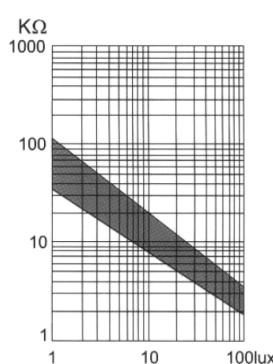


Figure 3 – Illuminance vs Resistance characteristic curve of the light sensor. Note that axes are in logarithmic scales.

The LED Actuation System

The microcontroller does not have a pure analog output. Instead, it emulates it using a switching digital signal with Pulse Width Modulation (PWM), whose ratio of the duration of 1's to the duration of 0's (duty cycle) is proportional to the analog voltage required. The frequency of the PWM signal should be at least 10x higher than the cut-off frequency of the input filter (implemented by the input capacitor), so that the switching does not have a significant influence on the luminance measurement signal. Note that the human eye is a low pass filter that is insensitive to flicker at frequencies above around 100Hz, so we cannot notice the LED pulsing. However, the LDR sensor is sensitive to higher frequencies and may inject noise in the reading circuit.

Office Illumination Model

To have a dynamical model of a physical system is always an important component in a control system. In our problem, this model represents how much light emitted by the LED is reflected in the walls of the office model and reaches the LDR (gain), and how fast variations in illuminance can be read by the measurement system (time constant). Having a good dynamical model allows us to predict the values of the Plant after some actuation and compare this with the current measurements to detect external disturbances. Note that each system may have a different dynamical model not only because the luminaires may be different, but also because the light paths travelled in each office model are different (different size, different reflectivity of internal surfaces, different arrangement of the elements inside the box). Every time the luminaire is moved or the configuration of items in the office changes, the gain of the dynamical model will change. To identify the model, a set of experiments should be planned to feed the Plant with different actuations and characterize its response. These tests should be carried out carefully, to prevent changes in the operating conditions during the process (e.g., external illumination). Also check that the PWM frequency for the LED is high enough to prevent unnecessary noise in LDR measurements. It may also be useful to implement a digital filter in the LDR readings to reduce noise, e.g., acquire many samples during a time interval and compute the mean or the median of the values.

The Individual Luminaire Controller

The individual luminaire controller (local controller) should be implemented as a PID controller with feedforward. The feedforward term has the role of producing fast changes of the LED in response to changes in the desired reference values at the desks. The feedback term has the role of responding to external disturbances. Start with the development of the feedforward term, whose objective is to drive the LED in open-loop to achieve some illuminance reference. Of course, the obtained value will not be exactly the desired one due to external disturbances and model errors, but it is

important to speed up the response of the system. On top of that, implement the PID feedback controller to cope with disturbances and modelling errors. Do not forget to implement adequate integrator anti-windup functions, to cope with actuator limits. **Use a sampling rate of 100Hz.**

Interfacing with PC

To read and write data from/to the microcontroller, a simple PC interface can be implemented using the program “serial monitor” in the Arduino IDE. The students should implement a simple character-based protocol to read LDR values in LUX, set LED PWM values, set upper (occupied) and lower (unoccupied) reference values for the local controller, set desk occupancy, turn on/off the feedforward term, turn on/off the anti-windup term, continuously stream a variable, get a batch of buffered data (e.g last minute), or to perform any other operations that you may find useful for testing/debugging or reporting.

To consider the extension to the control of multiple luminaires, include the index of the luminaire on the command arguments. A possible list of commands is presented in the next table. You may extend the list with other commands you find useful.

Command	Client Request	Server Response	Observation
Set duty cycle at luminaire i	“d <i> <val>”	“ack” or “err”	<val> is a number expressing duty cycle.
Get current duty cycle at luminaire i	“g d <i>”	“d <i> <val>”	<val> is a number expressing duty cycle
Set illuminance reference at luminaire i	“r <i> <val>”	“ack” or “err”	<val> is a number expressing the illuminance reference in LUX
Get current illuminance reference at luminaire i	“g r <i>”	“r <i> <val>”	<val> is a number expressing the illuminance reference in LUX.
Get measured illuminance at luminaire i	“g l <i>”	“l <i> <val>”	<val> is a number expressing the measured illuminance in LUX.
Set current occupancy state at desk <i>	“o <i> <val>”	“ack” or “err”	<val> is a Boolean flag: 0 – unoccupied, 1 – occupied.
Get current occupancy state at desk <i>	“g o <i>”	“o <i> <val>”	<val> is a Boolean flag: 0 – unoccupied, 1 – occupied.
Set anti-windup state at desk <i>	a <i> <val>	“ack” or “err”	<val> is a Boolean flag: 0 – off, 1 – on.
Get anti-windup state at desk <i>	g a <i>	a <i> <val>	<val> is a Boolean flag: 0 – off, 1 – on.

Set feedforward control state at desk <i>	w <i> <val>	"ack" or "err"	<val> is a Boolean flag: 0 – off, 1 – on.
Get feedforward control state at desk <i>	g w <i>	w <i> <val>	<val> is a Boolean flag: 0 – off, 1 – on.
Set feedback control state at desk <i>	b <i> <val>	"ack" or "err"	<val> is a Boolean flag: 0 – off, 1 – on.
Get feedback control state at desk <i>	g b <i>	b <i> <val>	<val> is a Boolean flag: 0 – off, 1 – on.
Get current external illuminance at desk <i>	"g x <i>"	"x <i> <val>"	<val> is a number expressing external illuminance in lux.
Get instantaneous power consumption at desk <i>	"g p <i>"	"p <i> <val>"	<val> is a number expressing instantaneous power at desk <i> in Watt. Assume each led nominal power = 1W.
Get elapsed time since last restart	"g t <i>"	"t <i> <val>"	<val> is a number expressing elapsed time in seconds.
Start stream of real-time variable <x> of desk <i>. <x> can be "l" or "d".	"s <x> <i>"	"s <x> <i> <val> <time>"	Initiates a real-time stream of values. Every time a new sample of a certain variable is available, it is sent to the client in a string with the format indicated. <time> is an increasing timestamp in milliseconds.
Stop stream of real-time variable <x> of desk <i>. <x> can be "l" or "d".	"s <x> <i>"	"ack" or "err"	Stops the real-time stream of values.
Get last minute buffer of variable <x> of desk <i>. <x> can be "l" or "d".	"b <x> <i>"	"b <x> <i> <val1>, <val2>, ...<val_n>"	Values are returned in a string of comma separated numbers. The string is terminated with the newline character.

4.2. Performance metrics:

To properly validate an engineering solution, it is fundamental to define appropriate performance metrics, expressing in a quantitative way the requirements addressed in its formulation. For this project, we aim to minimize the energy spent in illumination while providing comfort to the users. The energy is the accumulation (integral) of the instantaneous power along time. Suppose the maximum power of luminaire j is denoted as P_j . Then, a formula to compute the energy consumed at each desk is:

$$E_j = P_j \sum_{i=1}^N d_{i-1}(t_i - t_{i-1})$$

where i is the index of the control samples, t_i is the time in seconds of the i -th sample, and d_i is the led duty cycle value (between 0 and 1) at sample time t_i . The units of this metric are Joule [J].

While energy minimization is simple to formulate, comfort criteria are more subjective. We can consider the following rules:

- a) The system should prevent periods of illumination below the minimum settings defined by an occupation state. A metric to assess this criterion can be defined as the average error between the reference illuminance (L) and the measured illuminance (l) for the periods when the measured illuminance is below the reference. Let us call this quantity the **Visibility Error**:

$$V = \frac{1}{N} \sum_{i=1}^N \max(0, L(t_i) - l(t_i))$$

where N is the total number of samples used to compute the metric and t_i are the sampling times.

The previous expression refers to a single desk. The total average error should be computed as the sum of the average errors at each desk. The units of this metric are [LUX].

- b) The system should prevent frequent ups-and-downs of illuminance (flickering) while the reference is at a constant value. A metric to assess this criterion can be defined as the average magnitude of the signal derivatives when it changes sign, during periods of constant occupation. Let us first define the flicker at time t_i as f_i

$$f_i = \begin{cases} (|l_i - l_{i-1}| + |l_{i-1} - l_{i-2}|) / (2T_s) & \text{if } (l_i - l_{i-1}) \times (l_{i-1} - l_{i-2}) < 0 \\ 0 & \text{otherwise} \end{cases}$$

where T_s is the sampling period, l_i is the measured illuminance at time t_i , and $|A|$ denotes the absolute value of A . Transient periods due to explicit variation of the reference should be excluded from the formula.

We can now define the **Flicker Error** as:

$$F = \frac{1}{N} \sum_{i=1}^N f_i$$

Again, the previous expression refers to a single desk. The total average flicker should be computed as the sum of the flicker error at each desk. The units of this metric are [LUX/s].

In the report, identify factors that can influence this metric.

The computations should be done at all control cycles, i.e. with a frequency of 100Hz.

A few more commands regarding these metrics can be implemented to interface with the PC.

Command	Client Request	Server Response	Observation
Get accumulated energy consumption at desk <i> since the last system restart.	"g e <i>"	"e <i> <val>"	<val> is floating point number expressing accumulated energy consumption at desk <i> in Joule. Assume each led nominal power = 1W.
Get accumulated visibility error at desk <i> since last system restart.	"g v <i>"	"v <i> <val>"	<val> is floating point number expressing the accumulated visibility error in lux.
Get accumulated flicker error at desk <i> since last system restart.	"g f <i>"	"f <i> <val>"	<val> is floating point number expressing the accumulated flicker error in lux/s.

4.3. Implementation Tips

For the report, it is very important to collect data from your system to make plots of the different signals in the control system (references, control values, measurements) and compute metrics that show the correct operation of the system. Implement functions in the PC interface that allow you to collect this data.

You can use one of the other luminaires to inject deterministic disturbances in the system and test the ability of your controller to attenuate them.

Note that serial communications use precious microprocessor time. Choose messages with short size and a high baud rate. Compute communication delays and verify that the communication time can be accommodated within the available control loop period.

You can copy text from the Serial Monitor. Format your messages so that you can use the copied text to graphically visualize the data in Matlab or Excel. Note that the Arduino IDE has a Serial Plotter interface that can display data graphically if the messages are properly formatted.

You can assign different IDs to each luminaire by using the provided headers and jumpers to shunt digital input ports, and read these values at start-up.

4.4. Milestones for each lab session:

Although it is not mandatory to follow a strict agenda in the execution of the project, there are some minimum objectives that should be met each week, to ensure a timely execution of the project. Note that objectives are quite hard to achieve during the 3h of the session, so some items must be prepared at home before the session, and others completed autonomously after the session.

Session 1 (March 7th - 11th) – Intro to the Hardware and Software:

- Reception of the material needed for the project.
- Installation of the Arduino IDE and Arduino-core for RPI pico, in the PC that interfaces the microcontrollers.
- Loading the first Arduino IDE sketch into the assembled luminaire. The first upload requires holding the BOOTSEL button before powering up the microcontroller (connecting to USB). You can use the example ‘FADE’ included in the rp2040 examples to see the RPI pico built-in LED fade in and out, or the example ‘Dimmer’ included in the Communication Examples to set the dimming level of the Built-in led given user input in the SerialMonitor.
- Assembling one luminaire. Soldering the header pins of one microcontroller and assembling the LED drive and LDR measurement circuits in the breadboard (the other luminaires should be assembled at home)
- Development of basic programs to (i) repeatedly read values from the LDR and send those values to serial output; (ii) read values from the serial input and set the duty cycle of the LED to this value). See examples ‘AnalogReadSerial’ and ‘Dimmer’.
- Implementation of a function that converts the voltage read at the analog input port to LUX (you must use the expressions of the voltage divider and the log-log characteristic curve provided in the datasheet).

Session 2 (March 14th – 18th) – System Assembly and Identification:

- Use a shoe box (or similar) to create a scaled model of an office space. It is important that the box can isolate well the external light.

- Write a program to perform step changes in the LED actuation and collect the LDR signal in LUX and Volt at a high sampling rate.
- Use the steady state values of the curve in LUX to compute the static gain of the system (LUX/Input Units). Note that, theoretically, this gain should be constant. Improve the calibration of the slope of the log-log LDR characteristic curve to satisfy this constraint.
- Due to the sensibility of the static gain to changes in the box structure, devise and implement a **simple method** to calibrate the static gain every time the system boots. This way the system is always working with calibrated gains, even if the elements inside the box moved due to transportation.
- Use the transient values of the curve in Volt to estimate the time constant $\tau(x)$. Note that, theoretically, this value should depend on the current illuminance x . For a better precision in the computation of $\tau(x)$ perform experiments at different illuminance levels using both positive and negative steps.
- Write down a rule that expresses the time constant as a function of the initial and final illuminance.
- Create a C++ class for a system simulator, that can predict the values read by the sensor, given the actuation commands. Evaluate the quality of your simulator by comparing simulated with real responses.

Session 3 (March 22nd – 25th) – Local Controller

- Feedforward Control. Using the knowledge of the static gain computed in the previous session, write, and test a feedforward controller for your system. Check the properties of the feedforward controller in terms of accuracy and response time to step changes in the references.
- Feedback control. Implement the feedback controller with anti-windup. Check how it behaves in the presence of external disturbances, in terms of accuracy, overshoot and oscillations (flicker).
- Integrate the feedback controller with the feedforward controller. Generate appropriate references to the feedback controller, using the simulator developed in the previous session, to prevent overshoot in the step responses.
- Implement the metrics for evaluation and quantify the effects of different controller configurations.

4.5. Guidelines to the mid-term report:

- a) Take pictures of the interior and of the exterior of the box enclosing the luminaires. Make sure that you illustrate the position of the LED, the LDR and the emission / reflection path.

- b) Show plots of the steady state characteristic of the system. Show step responses of the system in different illuminance conditions.
- c) Characterize the jitter in your control system. How much does the sampling rate deviate from the desired one?
- d) Characterize the error in your feedforward controller. Implement a simulator of your system and compute the average mean squared error between the simulation and the actual measurements.
- e) Characterize the dynamic characteristics of the feedback controller, in different illuminance conditions (overshoot, damping factor).
- f) Illustrate any improvements that you make to the basic feedback controller (feedforward term, anti-windup, etc) with plots of the time responses.
- g) Comment the code for your controller.
- h) Characterize the processing time taken by the control computation, serial communications, and other computations.
- i) Always use SI units for pertinent quantities. Check the datasheets to verify the conversions from electrical to physical units.
- j) The report should describe the problem, the adopted solutions, the obtained results, and discuss the pros and cons of your design choices.
- k) The report must be complete but succinct, with less than 10 pages including graphics. Avoid redundancies and overly technical content. Try to summarize as much as possible but do not leave out the important issues.
- l) Graphics should be self-contained, i.e., fully labelled and with complete captions.

