universidade de aveiro deti departamento de electrónica, telecomunicações e informática

42088 – Industrial Project

Project Final Report

Project name:	Low-Cost LED Analyser
Customer:	Altice Labs
Customer supervisor:	Eng. Hélder Alves Eng. Nuno Balseiro
Date issued:	14/01/2026
Team members:	Main Contact: Jorge Neto, jorgen@ua.pt e 964283760 Other members of the team: Guilherme Santos, gui.santos@ua.pt e 962013357 Martim Gil, martim.freitas.gil@ua.pt e 930556615 Patrícia Moltchanova, patriciamoltchanova@ua.pt e 969828322 Celina Brito, celinarb@ua.pt e 960198366
Supervisor:	Professor Francisco Fontes

Management summary:

At the end of the development period, the project reached a stable and functional state, where all measurements were done to conclude that the theoretical voltage values in their respective location were achieved, and tests proved that all sensors were functional and as calibrated as possible, leading to the exposition date, where we demonstrated a solid setup in which real-time tests were made. In terms of schedule, the main milestones were achieved within the planned schedule, with only minor delays related to PCB validation and component finding, which did not compromise the final delivery.

Regarding cost, the project remained within the initially defined budget, with all expenses aligned with the estimated component and fabrication costs. Concerning scope, the main objectives of developing a functional LED analyser prototype, including optical sensing, I^2C communication, multiplexing, and modular software architecture were achieved. At project closure, a working hardware and software prototype was available, supported by validated drivers and unit tests, ready for further calibration and integration into a production environment.

Project status

This project reached its final stage with the successful development of a functional low-cost LED analysis system, aligned with the initial specifications provided by the proposing company. The original challenge consisted of designing a solution capable of objectively evaluating the colour, luminous intensity, and wavelength characteristics of LEDs used in network devices such as routers and gateways, while reducing dependence on subjective human inspection and costly commercial equipment.

The company's specifications emphasised the need for:

- Objective and repeatable LED measurements;
- A compact and cost-effective solution;
- Applicability to industrial and testing environments;

To address these requirements, the team proposed a system based on affordable optical sensors (VEML3328), controlled by a Raspberry Pi, and supported by a modular software architecture. The addition of plastic optical fibres to the testing phase allowed the isolation of the LED under test from ambient light, significantly improving measurement consistency. Still in the protocols to mitigate the noise originated by ambient light, 3D supports were produced to fully "black out" the sensor's area. The use of an I^2C multiplexer enabled the analysis of multiple LEDs using a single processing unit, meeting scalability expectations.

Fig. 1 – Two prototypes of the LED Analyser

Project summary

The objective of this project was to develop a low-cost system using a Raspberry Pi capable of objectively analysing the colour, luminous intensity, and wavelength characteristics of LEDs commonly used in network devices at Altice Labs, such as routers and gateways. In industrial environments, LED validation is often performed through visual inspection or expensive optical instruments (such as the Feasa LED analyser, used as a reference in our work), which can lead to inconsistent results or increased operational costs. This project addresses this gap by proposing an accessible solution with repeatable and consistent results, considering the simplicity and robustness of specific components.

The system is based on a modular hardware and software architecture. Colour and light intensity measurements are obtained using dedicated RGB optical sensors, while plastic optical fibres are employed to isolate and conduct the LED light to the sensor, which is under test, therefore minimising the influence of ambient light. An I²C multiplexer enables the sequential measurement of multiple LEDs using a single processing unit, improving scalability and reducing hardware redundancy, this is needed since the Raspberry Pi only has one communication channel. A Raspberry Pi is responsible for sensor control, data acquisition, and processing, chosen for its processing capabilities, wide ecosystem and ease of integration.

On the software side, a layered architecture was adopted, separating low-level hardware communication from data processing and system control. Unit testing was implemented to validate core drivers and ensure system reliability. A REST-based interface allows communication between the system and the user, enabling future integration with external applications or automated testing environments. The developed prototype successfully demonstrated the ability to acquire consistent RGB and intensity measurements across multiple LEDs, validating the feasibility of the proposed approach. The results confirm that the system represents a functional and cost-effective alternative to commercial LED analysis equipment, particularly suited for academic, prototyping, and light industrial testing contexts.

Project definition and outline

Phase 1 – Inception:

- In the initial phase, we did research and studied the necessary concepts required to proceed with the project, such as the correlation between RGB, intensity and wavelength, and searched for components in the market, to choose the.

Phase 2 – Elaboration:

- We analysed and validated the schematic previously done and acquired by Altice Labs. With this, we updated some components.
- Definition of the final system to implement, from hardware to software.
- Developed the architecture of the project, establishing a clear and detailed foundation that enables the start of the construction phase with reduced technical risk.
- Tasks were distributed according to each member's expertise, enabling parallel development and efficient coordination across the project.

Phase 3 – Construction I:

- Completion of the full Bill of Materials (BOM) and acquisition of all required components.

- Development and iteration of several PCB versions, culminating in a finalised design sent for fabrication.
- Preparation and validation of the mechanical holder design for optical alignment.
- Start of software development, with the modular architecture defined and core modules under implementation.
- Ordering of the final PCB and components in preparation for system assembly and integration.

Phase 4 – Construction II:

- Once the ordered components were obtained, we proceeded to assemble the PCB in Altice Lab's facilities, which enriched our knowledge about assembly techniques and procedures.
- Validation of power and circuit behaviour through voltage measurements at key nodes.
- Regarding the software, we were able to undergo a first phase of physical tests, even though not fully correct and missing the integration with the API, we achieved results that made the team conclude that this section of the project was nearly completed, with only a need for calibration to finish the section.
- First printing of 3D case and POF supports and development of a second case for a second PCB, which was assembled with the remaining components to ensure that with two prototypes, the software team would have two comparative metrics during the calibration phase.

Phase 5 – Transition:

- Two full functional prototypes, both with a suitable 3D case showcasing a stable and practical design.
- Software progressed to improve test values, turning them into more accurate data, as a consequence of the calibrations made during Phases 4 and 5.

Project architecture

The hardware architecture of the proposed LED Analyser was designed with the primary objectives of low cost, modularity, scalability, and measurement reliability. The system is centred around a custom-designed printed circuit board (PCB) that interfaces multiple optical sensors with a Raspberry Pi, enabling controlled and repeatable acquisition of LED colorimetric data.

The custom PCB integrates all sensing, multiplexing, and interface components into a compact and structured layout. Proper decoupling capacitors are placed near each sensor and the multiplexer to minimise noise and voltage fluctuations. Finally, pull-up resistors were placed on the I²C lines to ensure reliable communication across all channels.

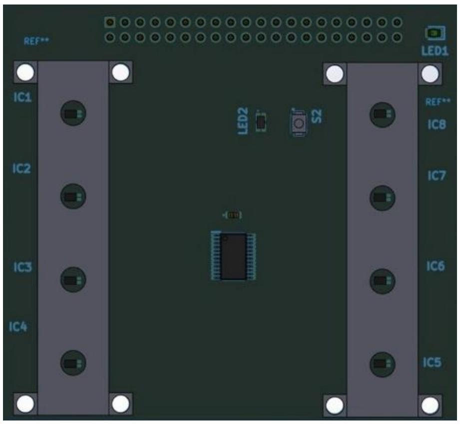

Fig. 2 – 3D view of the PCB layout

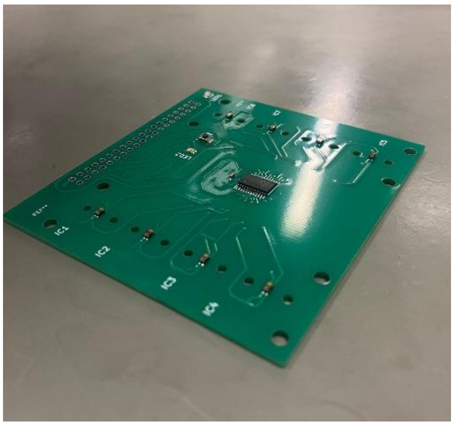

Fig. 3 - Fabricated PCB prototype

At the sensing layer, the system employs multiple VEML3328 RGB colour sensors, selected for their ability to independently measure red, green, blue, and clear light channels with high resolution and low power consumption. The inclusion of the clear channel enables improved luminance estimation and normalisation of RGB values under varying lighting conditions. Since all VEML3328 sensors share the same fixed I²C address, a TCA9548A 1-to-8 I²C multiplexer is used to enable individual sensor selection. The multiplexer allows the Raspberry Pi to communicate with each sensor independently by switching I²C channels.

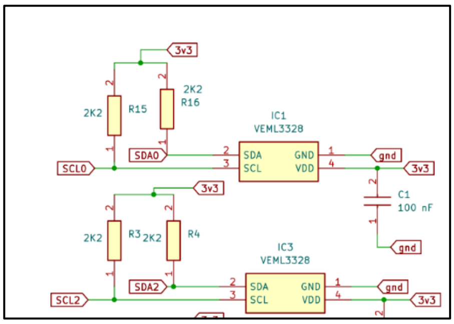

Fig. 4 - Schematic of the sensor circuit

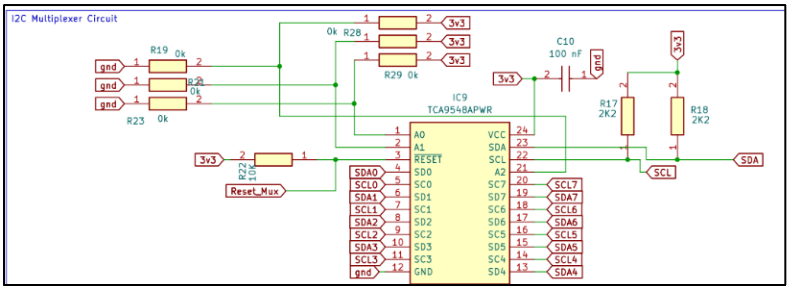

Fig. 5 - Schematic of the Multiplexer circuit

Regarding the processing and control unit, the Raspberry Pi serves as the central processing unit of the system. It is responsible for sensor configuration, channel selection, data acquisition, and higher-level data processing. Communication between the Raspberry Pi and the sensing board is performed exclusively over the I²C bus, ensuring a simple and robust interface.

Our software can be divided into three parts, as represented in Figure 6. The control unit is responsible for reading the data from the sensors, the user interface is to ensuring a good user utilisation, and the rest API is responsible for ensuring the connection between the other two modules.

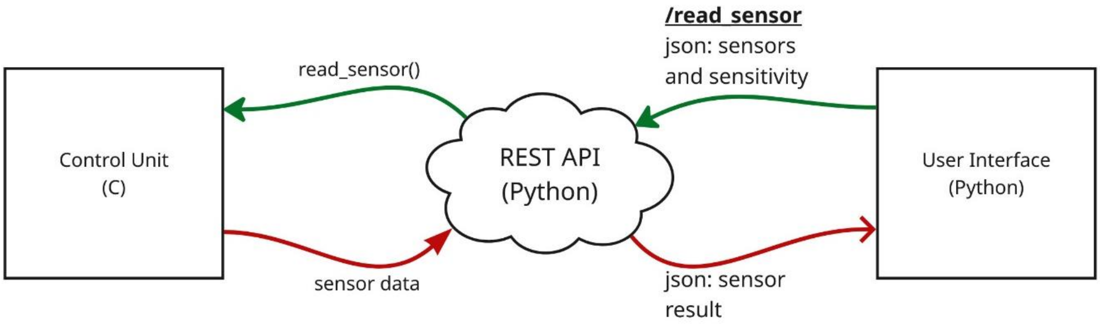

Fig. 6 - High-level software representation

The control Unit is made up of three modules, which are represented in Figure 7:

- VEML3328 Driver: Module that controls the sensors.
- TCA9548A module: Module to control the Multiplexer.
- I2C Driver module: Module for the communication between the Raspberry Pi and the circuit.

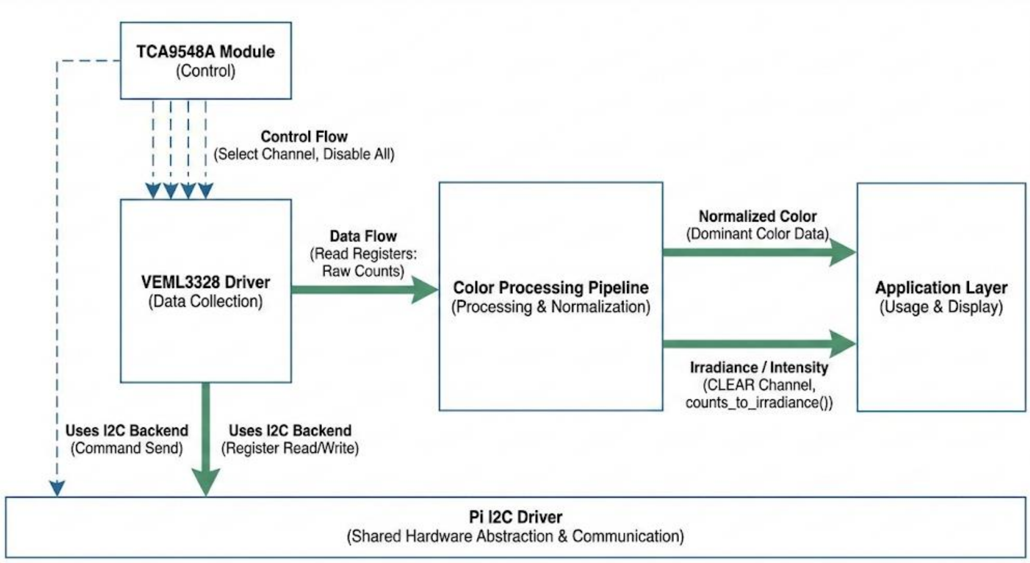

Fig. 7 – Control Unit Modules

For usability we also developed a user interface as shown in figure 8, this interface allows choosing which sensors to use and also allows to change the measuring sensitivity setting, and it presents the results in a table and also in a colour graph. For visibility whenever the user hovers the mouse pointer over a table value this value is enhanced in the graph.

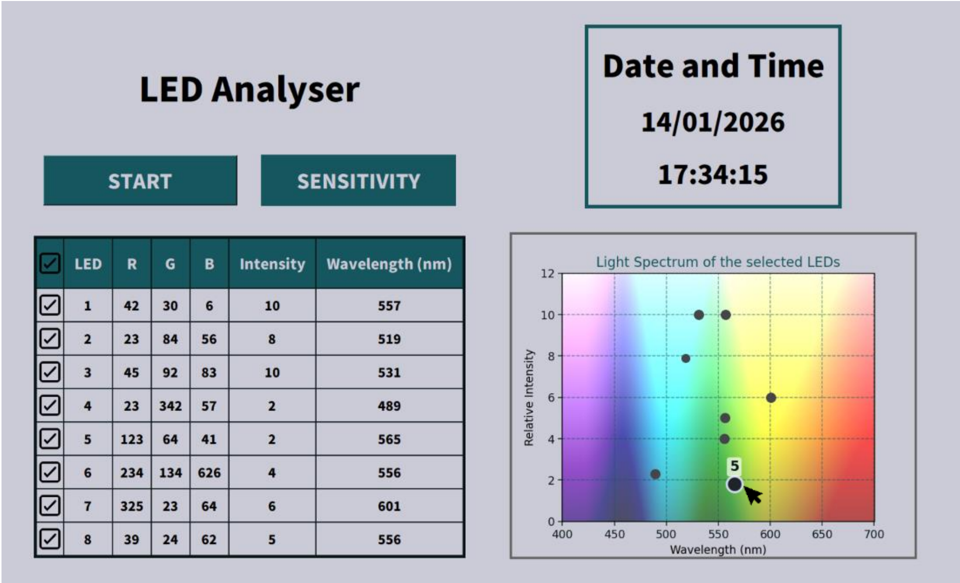

Fig. 8 – User interface

Design and production documents

To face the challenge of developing a project with the magnitude of this particular one, not only due to how desirable and active the market of LED analysers but also to the complexity of the project itself, represented by high rates of software, hardware and also mechanics work that made the team getting out of its comfort zone and learn about new software to develop the schematic/PCB and 3D components and in some cases new programming languages.

For the schematic and PCB design, the team selected KiCad as the main design software. It was chosen during the initial phase of the project, while the work plan was being developed. KiCad proved to be a solid and reliable choice, allowing the team to quickly become familiar with the tool and successfully achieve the expected design outcomes.

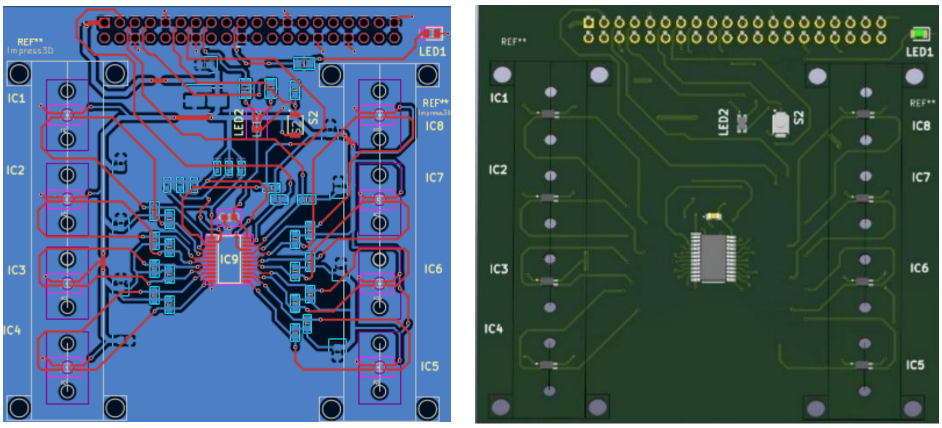

Fig. 9 – Final printed board circuit layout

Concerning the software development of this prototype, the team chose Visual Studio Code as the integrated development environment (IDE) for both the control unit and the REST API, as it was already familiar to the team. Although the project has a stronger focus on hardware development, a thorough study of the required programming languages and frameworks was carried out, which resulted in fully functional software implementations. Additionally, to facilitate the access of the team to the (in-time) current state of our team's work, a shared repository was created on GitHub.

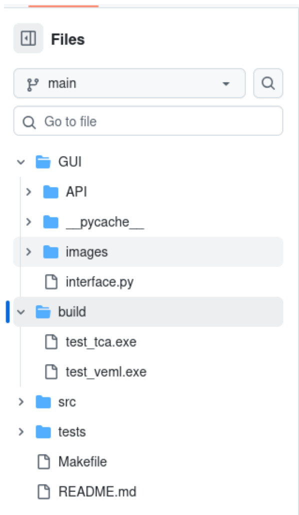

Fig. 10 – Github

Tests

To overcome the different stages of these projects, the team continuously set goals for each stage, determined by tests that would conclude whether we proceeded to the upcoming stages or not.

Due to the development of all the hardware using the previous schematics produced and shared by Altice Labs, the first row of tests started once we completed PCB assembly. We had to make sure everything was running smoothly as expected, so we applied a 3.3V input voltage and tested, using a multimeter, the current and voltage in different lines and components of our board, ending up matching the values expected. This test allowed us to start mounting the PCB with the Raspberry Pi, getting us close to a simplistic prototype and progressing to more complex phases of the project.

An important test conducted by our team was the first one made with the software structurally built, and this test took place in a room on Altice Labs' facilities:

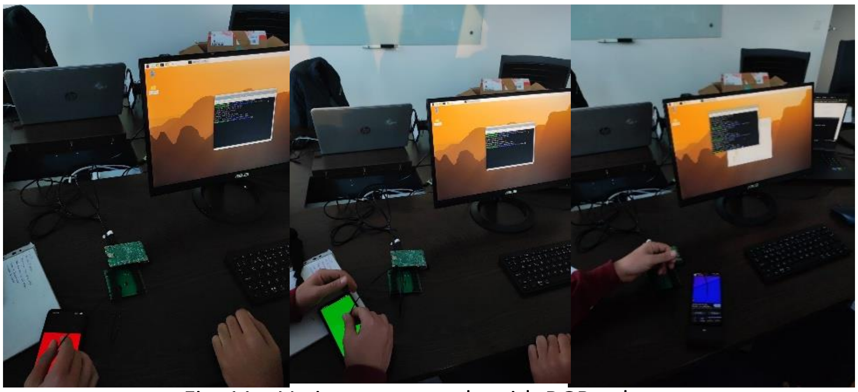

Fig. 11 – Various tests made with RGB colours

These tests were conducted using a POF (plastic optical fibre) to guide light from a smartphone screen with images representing colours defined by RGB components, assessing the accuracy of our results. However, as expected, the first values obtained showed some ambient light and screen filter noise, which made our software team confident that, in other circumstances and with multiple calibrations, the reliability of our results would improve considerably. In fact, this is exactly what happened.

Getting close to the DETIXchange, Altice Labs provided us with a configurable router that had RGB LEDS and more trustworthy RGB components due to the lack of filter seen in smartphone p.e. and with the possibility of defining each component's red, green and blue, giving us something to compare to our results. Allied to the high number of calibrations, we were able to obtain better results, leading to a successful exposition day where the team repeatedly tested the setup (LED analyser + Router), with the results turning out to be consistent, although the high rate of simulation made. It’s worth mentioning that during the presentation day, the software regarding both the control unit and REST API worked simultaneously, demonstrating all the work developed by the team’s software section during the last few months.

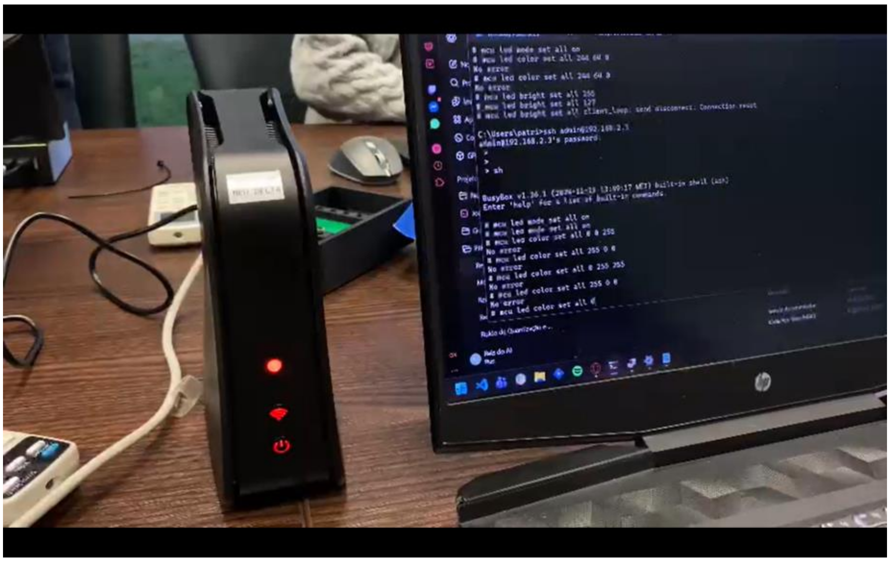

Fig.12 – Testing router

Key milestones

ID	Description	Original planned completion date	Current planned completion date	Actual completion date
1.1	Project definition and organisation	22/10/2025	22/10/2025	22/10/2025
2.1	Study about the concepts	24/10/2025	24/10/2025	24/10/2025
2.2	Analysis and validation of the given schematic and components	29/10/2025	29/10/2025	31/10/2025
2.3	Development of the schematics	05/11/2025	05/11/2025	05/11/2025
2.4	Better analysis of risks and better planning	05/11/2025	05/11/2025	05/11/2025
3.1	Test of final schematics on breadboard	12/11/2025	5/12/2025	10/12/2025
3.2	Programming control software (I2C, MUX, Sensors)	19/11/2025	28/11/2025	10/12/2025
3.3	Programming the REST API	26/11/2025	7/12/2025	7/12/2025

ID	Description	Original planned completion date	Current planned completion date	Actual completion date
3.4	Development of the POF holder/device case	26/11/2025	7/12/2025	19/12/2025
4.1	PCB assembly and initial prototype testing	10/12/2025	10/12/2025	10/12/2025
4.2	Final product testing	17/12/2025	17/12/2025	19/12/2025
4.3	Final calibrations	17/12/2025	05/01/2026	05/01/2026

Progress and deviation from plan

During these kinds of projects, where the team is consistently fighting against time to achieve certain goals and milestones, errors and mistakes are common. Still, compared to the same mistakes and errors expected at the beginning of the prototype development, the team can conclude that, besides those normal upsets, the team managed to take measures to avoid big delays in deadlines for certain assignments. With no problems, the team can say that it affected the end delivery of the finalised prototype.

One of those upsets happened before ordering the PCB, where the team noticed that the 40-pin GPIO would be in the same layer as the sensors, which would compromise its functionality, since when plugging the Raspberry Pi to the PCB using the GPIO pins, the Raspberry Pi would cover the sensors, leaving us with no access to the sensors. This problem came up when changing the layer in KiCad, the pins from the GPIO were inverted, which led to our prototype ending up being different to the expected one.

Therefore, we devised two solutions to this problem: one involved deviating the sensors from the Raspberry Pi, as shown in Fig. 13. The other solution was to skip a row of pins, resulting in the last prototype, as seen in the right-side image of Fig. 1.

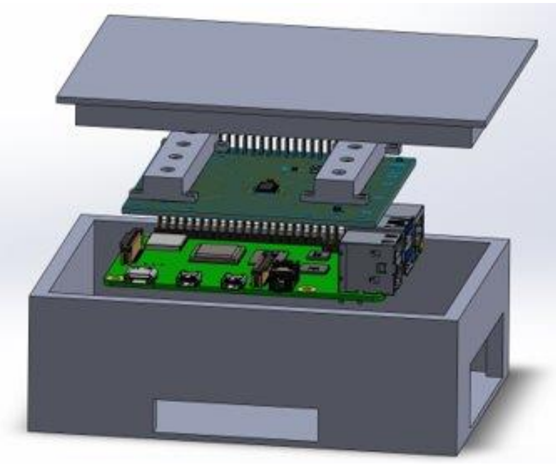

Fig 12 – First Design developed

Fig 13 - First functional prototype

By looking at Fig. 12 and 13, it's clear that this "build" mistake also affected the initial 3D structure that had to be rectified to be able to fit the PCB and Raspberry Pi bundle. Additionally, once the first board was functional, the team decided to assemble a second board, this time doing it similarly to the original idea, with the final deliverable product being the one in Fig. 1.

Risks

At this final phase, the risk of us failing the deadlines was only one. Since both the software and hardware have already undergone testing, such as individually and combined, we were convinced that our work was fine and with no errors. The only risk we had was the possibility of any of our members being unavailable. Nevertheless, we were confident that in case it happened, any of the other members would be able to fill in for them. The reviewed risk matrix is in the appendix.

Financial status

Development cost per unit

Considering the development cost of building one functional device, the following table shows an estimate of the cost to produce one prototype like ours:

#	Category	Description	Quantity	Status	Value
1	Component	VEML3328	8	Purchased	10,96 €
2	Component	Raspberry Pi 4 Model B 4GB	1	Purchased	62,91 €
3	Component	CRGCQ0603F2K2 (2.2K ohms resistor) (SMD)	16	Purchased	00,28 €
4	Component	C0603C104K5RACTU (100 nF capacitor) (Ceramic)	8	Purchased	00,07 €
5	Component	TCA9548APWR (3:8 Mux)	1	Purchased	01,14 €
6	Component	RC0603FR-0710KL (10k ohms resistor) (SMD)	3	Purchased	00,03 €
7	Component	Electrolytic capacitor ALUM 10UF 20% 16V SMD (865080340001)	1	Purchased	00,16 €
8	Component	CRS0805-FX-1000ELF (100 ohms resistor) (SMD)	4	Purchased	01,04 €
9	Component	LTST-C170GKT (Green LED)	1	Purchased	00,08 €
10	Component	B3U-1000P (Push Button)	1	Purchased	01,28 €
11	Component	PRT-16763 (40-pin Connectors)	1	Purchased	04,16 €
12	Component	RC0402JR-070RL (0-ohm resistor)	9	Purchased	00,81 €
13	Component	APHBM2012ETSGTC (Green and red bicolour LED)	1	Purchased	00,62 €
14	Component	SMD_22 (POF Surface Mount)	8	Purchased	13,36 €
15	Component	Kemo S109 – Optical fibre cable - 2m	1	Purchased	03,27 €
16	Fabrication	PCB	1	Purchased	10,52 €
17	Fabrication	PCB Stencil	1	Purchased	07,00 €
18	Fabrication	Mechanical Components (Box, POF supports)	1	Purchased	02,16 €
				Total	**119,85 €**

Besides the cost of a single device, we also present the cost of the entire product development:

#	Category	Description	Quantity	Status	Value
1	Component	VEML3328	18	Purchased	24,66 €
2	Component	Raspberry Pi 4 Model B 4GB	1	Purchased	62,91 €
3	Component	CRGCQ0603F2K2 (2.2K ohms resistor) (SMD)	40	Purchased	00,70 €
4	Component	C0603C104K5RACTU (100 nF capacitor) (Ceramic)	22	Purchased	00,20 €
5	Component	TCA9548APWR (3:8 Mux)	3	Purchased	03,42 €
6	Component	RC0603FR-0710KL (10k ohms resistor) (SMD)	6	Purchased	00,05 €
7	Test component	VEML3328-SB (sensor testing board)	1	Purchased	11,49 €
8	Component	Electrolytic capacitor ALUM 10UF 20% 16V SMD (865080340001)	4	Purchased	00,64 €
9	Component	CRS0805-FX-1000ELF (100 ohms resistor) (SMD)	8	Purchased	02,08 €
10	Component	LTST-C170GKT (Green LED)	3	Purchased	00,23 €
11	Component	B3U-1000P (Push Button)	2	Purchased	02,56 €
12	Component	PRT-16763 (40-pin Connectors)	2	Purchased	08,32 €
13	Component	RC0402JR-070RL (0-ohm resistor)	9	Purchased	00,81 €
14	Component	APHBM2012ETSGTC (Green and red bicolour LED)	2	Purchased	01,44 €
15	Component	SMD_22 (POF Surface Mount)	16	Purchased	26,72 €
16	Component	Kemo S109 – Optical fibre cable - 2m	1	Purchased	03,27 €
17	Fabrication	PCB Stencil	1	Purchased	07,00 €
18	Fabrication	PCB	5	Purchased	52,60 €
19	Fabrication	Mechanical Components (Box, POF supports)	2	Purchased	04,32 €
				Total	**213,42 €**

Group contribution

Group member	Major contributions	Work share (%)
Jorge Neto - Project Coordinator	Team coordination, communication with Altice Labs, component selection, and hardware assembly and testing	20
Martim Gil - PCB & Power Systems	PCB design, circuit validation, DC–DC converter integration, and connector layout.	20
Patricia Moltchanova - Mechanical & Optical Design	Design of enclosure and optical fibre alignment system, 3D printing, and assembly.	20
Guilherme Santos - Firmware & Data Acquisition	Programming the Raspberry Pi for sensor communication (I2C), calibration routines, and data logging.	20
Celina Brito - Interface & API Development	Development of the graphical interface, data visualisation, and REST API for integration.	20

Comments and remarks

The development of this project reinforced the importance of structured planning, task distribution, and iterative validation in multidisciplinary engineering projects. The close collaboration with Altice Labs allowed the team to experience industrial workflows and constraints, improving both technical and organisational skills. Overall, the project was a strong learning opportunity that bridged academic knowledge with real-world engineering practice.

Project documentation and Annexes

Github link, with all software development and user manual (readme file):
https://github.com/guilhermsts/PI_Software

Risk matrix

Initial analysis

#	Risk	Consequence	Likelihood	Severity	Level	Solution

1	External interference at the POFs should be considered, outside light may interfere with the LED during testing	Lack of repeatability and consistency in results, leading to possible mistakes in the testing	Occasional	Marginal	3	Ensure all POFs have opaque coating and implement optical shielding and mechanical enclosure designs to prevent ambient light ingress
2	Incorrect LED measurements due to mechanical enclosure misalignment or wrong dimensions	If the ports designed for the POFs are not accurately dimensioned, the distance or angle between the LED and the optical sensor may be incorrect. This can cause inaccurate colour or intensity readings, poor repeatability, or require reprinting of the mechanical structure, delaying integration and testing	Occasional	Marginal		We must do a 3D-printed structure that ensures precise alignment with the device under test, thereby ensuring measurement repeatability. To achieve optimal accuracy, the probes must be positioned at a 90º angle relative to the LEDs, and the ports should be slightly undersized in the 3D print so they can be manually adjusted to the desired fit.
3	Delay in the shipment of components	Late deliveries push the start of assembly/integration, compress testing, increase rework risk, and may slip the demo/milestones	Likely	Marginal	2	We will search for equivalent components early in the phase and confirm long-term availability from suppliers
4	Team members' unavailability to finish their tasks,	Project progress will slow down, and dependencies between tasks may be broken, delays in development, integration or testing	Remote	Critical	2	We will reallocate tasks through the team, reallocate work to backups, extend lab hours, and escalate to

	conflicting schedules	phases, reduced quality of deliverables due to rushed work, possible need to reschedule activities or reduce project scope				partners for temporary support if needed
5	Delay in the hardware assembly phase due to errors on the PCB	If the PCB design has errors or the fabrication is defective, assembly and integration will be delayed while waiting for a new board. This can reduce the time available for testing and validation, increase costs, and risk missing project milestones or the final demo	Occasional	Critical	2	Carefully review the schematic and PCB layout, order an early prototype batch to validate the design and connections, keep spare budget and time for a potential re-order and use a reliable manufacturer with fast delivery and proven quality. If the board does not work as expected, perform manual rework or debugging to identify and fix the issue
6	Lack of specific technical knowledge	Team may face delays learning new tools, sensors, or software environments	Occasional	Critical	2	Having a plan and deadlines imposed, so that we know we have to start some tasks earlier than first thought, to learn and get used to them and still be on schedule
7	Delay in milestone completion	If the errors pile up, the team might have delays in the milestone completion	Remote	Marginal	3	Project monitoring through frequent meetings
8	Software malfunction or database corruption	Loss of collected measurement data and inability to control the hardware, causing	Occasional	Marginal	3	Frequent backups and simulated test routines

	leading to data loss or system lock-ups	downtime, invalid test runs, and schedule slippage				
9	Unplanned or unforeseen risks (unexpected events not identified during planning	Unexpected issues may disrupt the project schedule or workflow. These could include hardware damage, software incompatibilities, unavailable resources, or sudden design changes. Such events can cause delays, additional costs, or reduced project quality if not handled quickly.	Occasional	Negligible	3	Maintain schedule and budget buffers, conduct regular progress reviews to detect new risks early, ensure flexible task allocation, and keep spare components or backup plans

Residual risk analysis

	Consequence	**R_Likel.**	**R_Sever.**	**R_Level**
1	Minor measurement variation may still occur due to ambient light, but results remain valid and functional.	Remote	Negligible	4
2	Small positioning deviations may still occur, with negligible impact on measurement accuracy	Remote	Negligible	4
3	Minor schedule impact if alternative components or fast shipping are used.	Remote	Negligible	4
4	Slight delays may still occur	Remote	Marginal	3

5	Even if issues occur, early prototypes and rework reduce the impact, minor reassembly delay is possible, but within the buffer time	Remote	Marginal	3
6	After planning and training, delays are minimal	Remote	Negligible	4
7	Frequent monitoring allows early correction; small deviations may still happen, but will not affect the overall schedule	Remote	Marginal	3
8	After implementing backups, version control, and simulated test routines, the chance of losing critical data or control is very low. Minor issues, such as temporary crashes or loss of non-critical logs, may still occur but will not affect the system's main functionality	Remote	Negligible	4
9	Even with buffers and monitoring, some unforeseen events may still cause minor delays or require rapid adjustments, but they will not significantly affect the final project outcome	Occasional	Marginal	3