LIP-STUDENTS-25-11

Building a Station for Monitoring Ambient Radiation

Lucas Silva[1,a], João Antunes[1,b]

[1] *Faculdade de Ciências da Universidade de Lisboa, Lisboa, Portugal*

Project supervisors: Luís Peralta, Tomás Campante *September 29, 2025*

Abstract. This work aims to portray the development of a sensor station with affordable specialized sensors, which monitored weather variables throughout various days. It will go over the issues and solutions encountered in the process of assembling the station, both software- and hardware- wise, the results and takeaways, while also comparing some plots obtained with references from weather institutes. A few notes about the ease of recreating it, as well as a GitHub repository with the same code used is also presented. Overall, the work developed is able to qualitatively capture the behavior of temperature, air pressure, and light (visible, infrared and ultraviolet radiation) through several days.

KEYWORDS: Sensors, Arduino, I^{2}C Protocol, Ambient Radiation, Processing

1 Introduction

While in the current era the monitoring of weather variables such as temperature or the UV (ultraviolet) index for the use of the average consumer is largely performed by specialized institutions, it has also become very easy for anyone to set up their own weather station at home for an accessible price, whether through pre-built devices or DIY (do it yourself) projects with sensors and micro-controllers. During this internship, a sensor station was devised and the results obtained from it were analyzed.

This work aimed to develop a small, easily reproducible sensor station with inexpensive sensors interfaced with Arduino, which will acquire data on weather variables through the day and plot them over time, using a Raspberry Pi and the Processing platform. As such, it becomes possible to obtain plots for how light intensity, UV index, temperature, and air pressure progress throughout various days, and perform a qualitative analysis of them.

2 Materials and Components

This project requires the use of specialized sensors for measuring the environmental variables and micro-controllers and processing units for acquiring and managing the information. For this, the following specialized sensors were used.

Sensors

To monitor the weather, various sensors were chosen, as these specialized sensors focus on specific physical quantities.

Three radiation sensors were implemented. The ML 8511, that reads the UV index [1], the SI 1155, which reads visible and infrared radiation, as well as the UV index [2]. For ambient light intensity, the TSL 2561 was used, as it detects ambient light illuminance mimicking the response of the human eye [3].

Finally, the MPL 115A2 was selected to measure both temperature and air pressure [4], providing some insight into atmospheric properties.

Micro-Controllers and computers

All sensors were required to be connected to micro-controllers, that were then connected to computers for collecting and processing the data.

The Arduino Uno was implemented as a simple and straightforward interface between the sensors and the computer, while a Raspberry Pi 5 served as the unit to store, manage, and showcase the data. Some Raspberry Pi peripherals also helped improve its capabilities, such as an SD card, a cooling fan, and a micro-HDMI to HDMI cable.

3 Software Development

Arduino Code

The Arduino IDE (integrated development environment) was used for the development of the codes that performed the measurements. Note that most sensors have their own libraries that are required for their operations. These libraries also include example codes that perform the measurements for each sensor, so these codes were adapted into a single file, with some changes to ensure compatibility between them, allowing the sensors to take various measurements over time and average the results.

However, it was soon understood that some of these sensors could not work together due to their communication protocol properties.

The I^{2}C Protocol

The I^{2}C (inner-integrated circuit) protocol is a widely used communication protocol due to its simplicity and cost-effectiveness. It was developed by NXP Semiconductor in 1982 and, since 2006, its implementation no longer requires a license, which further increased its use. [5]

[a]e-mail: lucas.ligeiro.silva@gmail.com
[b]e-mail: antunesjoaomaria07@gmail.com

It can connect multiple devices through only 2 lines, the serial data line (SDA) and serial clock line (SCL).

Each I^{2}C target device, such as the sensors used, has an associated 7-bit I^{2}C address (in some cases 10 bit addresses are also used, however, none of the sensors used have 10 bit addresses). When initiating communication with a particular target device, the controller uses the target device address to send or receive data in the following I^{2}C frames.

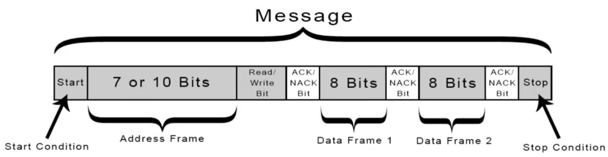

Figure 1. Example of an I^{2}C message. Image retrieved from reference [6].

This way, the controller, in this case, the Arduino, would select the addresses of the sensors and be able to obtain data from each of them individually through its SCL and SDA pins simultaneously.

All sensors, with the exception of the ML 8511 which simply relies on the comparison of two analog voltage values, use the I^{2}C protocol to communicate. However, both the SI 1145 and the MPL 115A2 have the same address, 0x60.

This conflict prevents these two sensors from being connected to the same I^{2}C lines. Despite this, the Arduino Uno only has one I^{2}C bus (i.e., communication system).

To solve this, various approaches were attempted, such as trying to change the address of one of them through code, trying to use an Arduino Mega since it apparently has two I^{2}C pair of ports, employing libraries that enable the configuration of an I^{2}C bus on arbitrary GPIO (General Purpose Input/Output pins), or even using the Raspberry Pi's own I^{2}C pins.

However, many of these solutions proved either incompatible with Arduino or too impractical given the limited time available. A more straightforward method was chosen, which employed two Arduino Uno boards for the station setup.

3.1 Processing Code

Multiple codes were developed to save the measurements and plot them over time using Processing, a visually oriented IDE based on the Java programming language. A final version of the codes was reached, that reads the output from the two Arduinos' Serial.print() and stores the listed values into their own arrays as well as the time of the day they were taken.

Additionally, it saves all arrays and timestamps into `.csv` files, as well as plotting the values over time at each step, which means that even if the program is unexpectedly interrupted, for example, in the occurrence of a power outage, the values are still saved until the last possible moment.

4 Hardware Development

4.1 The Sensors' Mount

The mount that would ultimately hold the sensors was assembled, and male headers were soldered onto copies of each sensor for breadboard testing.

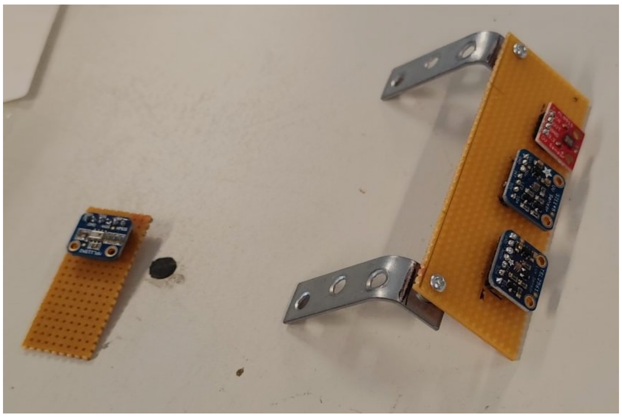

Figure 2. The sensors' mount: to the right, the main base with three light sensors, to the left, the temperature and air pressure sensor.

For this work, the experiments were conducted in a laboratory in the C8 building of the Faculty of Sciences of the University of Lisbon, which featured a south facing window. As such, the sun rises to the left and sets to the right relative to the window. The design intent shown in figure 2 was to position the three light sensors sitting on the windowsill, while the temperature and barometric sensors were suspended off of the larger mount. This arrangement was chosen to minimize the influence of the windows on the environmental measurements.

4.2 Wire-out

Given that two Arduino boards were used due to the compatibility issue discussed in section 3, the wiring was designed to isolate the temperature and air pressure sensor from the rest. A diagram of the wiring is presented in figure 3.

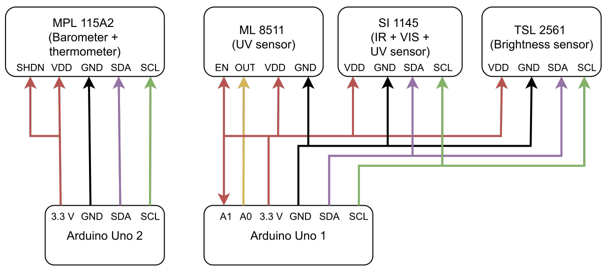

Figure 3. Diagram of the wiring of the mount.

After soldering the wires onto the mount shown in figure 2, labeling them, installing male pin headers to improve connection stability with the Arduino, and twisting the wires together to reduce electrical noise, the station was ready to start measurements.

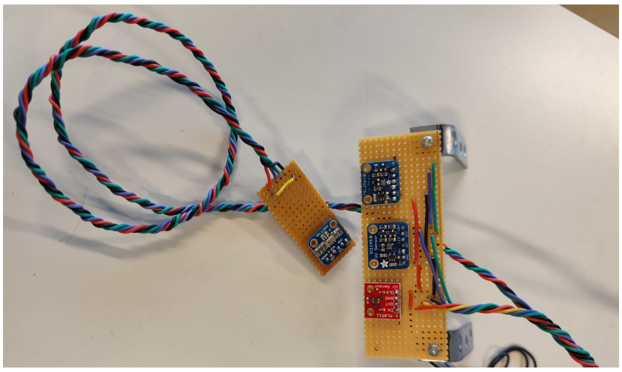

Figure 4. Finalized sensor station, joining the temperature and air pressure sensor though a cable to the three light sensors on the base.

5 Results

After leaving the station acquiring data for a few days, a clear day-to-night cycle was observable through the Processing interface.

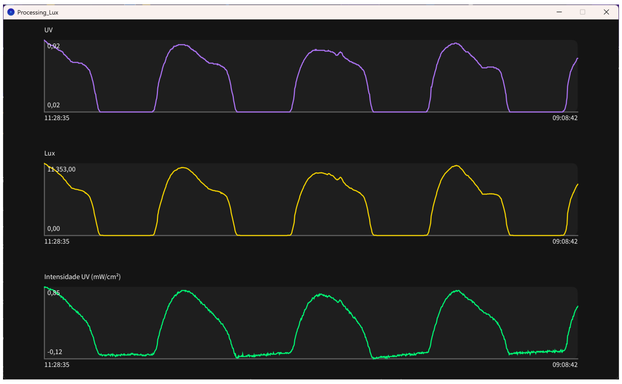

Figure 5. The plots for the light sensors after a few days of measuring.

By plotting the data for a single day using the `.csv` file produced, further conclusions can be drawn. All measurements shown from this point on were acquired during the 27th of July, 2025.

Temperature

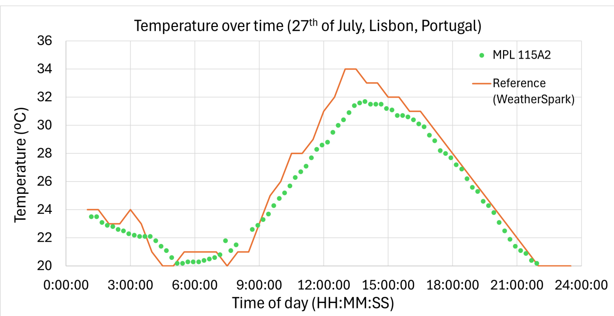

Figure 6. The variation of temperature measured during the 27th of July, as well as the temperature for that day taken from the website WeatherSpark at the Humberto Delgado Airport [7].

The temperature begins to rise after 6:00, as does the sunlight detection for the light sensors, and reaches a peak of 31.7 ºC at approximately 14:00. After that, it decreases as the sunlight intensity becomes weaker and the sun eventually sets.

The sensor readings largely align with the reference data used. There are some discrepancies in the peak readings, but that is probably due to the different locations of the measurements.

Air Pressure

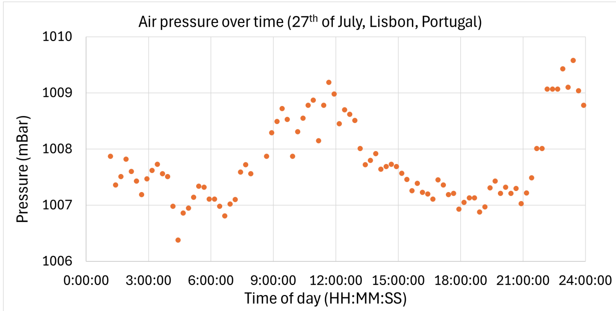

Figure 7. The variation of air pressure measured during the 27th of July.

The plots for air pressure were the most inconclusive among the data retrieved, as no discernible pattern was observed in its plot, and just floated around the same value for most of the time. However, this is to be expected, as air pressure usually does not vary a lot.

Visible and Infrared Radiation

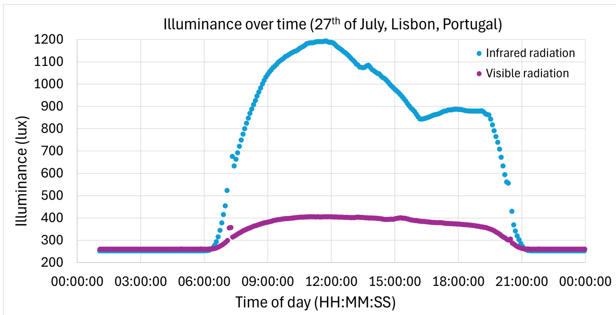

Figure 8. The variation of visible and infrared radiation illuminance measured during the 27th of July.

For visible and infrared radiation, the graphs follow a similar trend, showing that sunlight intensity reaches a peak around noon. However, there is a much higher amount of infrared radiation than visible radiation hitting the Earth.

During night time, these values don't drop below a certain threshold. This is due to the sensitivity of the sensor, which shuts down at night.

Note that this graph, as well as all light sensor graphs afterward, reach a plateau at around 16:00. This is due to the fact that the building the station was in cast a shadow over it at this time.

UV Index

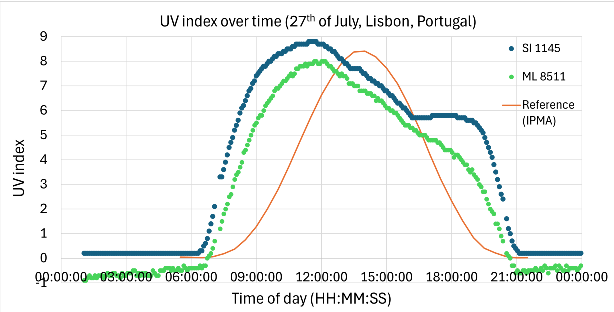

Figure 9. The variation of the UV index measured during the 27th of July, as well as the UV index for that day taken from IPMA (Instituto Português do Mar e da Atmosfera) at the Humberto Delgado Airport [8].

Two sensors were used to measure the UV index, which quantifies how dangerous UV radiation is. During peak sunlight hours, the UV index reached levels in the dangerous zone, hence why it is advised to not be out in the sun for too long at noon.

It is also noted that one of the UV index sensors is much more imprecise than the other, that is, because the ML 8511 relies on the measurement of a photocurrent from an UV photo-diode and its comparison to its supply voltage, whereas the other sensor simply returns the UV index through I^2C. Because of this, the graph of the former sensor shows significantly more fluctuation, even dipping into the negatives at night time.

For comparison, a graph for the UV index on that day was included from IPMA. The graphs are very different, however, this is attributed to how the sensors detect the UV index, how imprecise they are, and the different exposure of the settings the sensors are in.

Ambient Light Illuminance

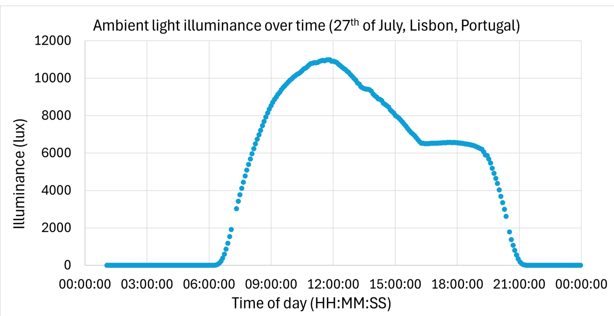

Figure 10. The variation of ambient light illuminance measured during the 27th of July.

The ambient light sensor detects ambient light illuminance with a response that mimics the human eye, providing a measure of perceived luminosity. It also follows the same trend as the other light sensors, which is consistent with the expected.

6 Reproducibility

Finally, the feasibility of independently replicating the station was also evaluated.

Since the sensors are inexpensive and the assembly process and putting everything together is not that hard, this station could be an interesting DIY project. For microcontrollers, Arduino Unos were used, however, cheaper options may also work, as long as they have I^2C communication and ADCs (analog to digital converters).

For running the station, any computer or a Raspberry Pi can be used, but the use of the weaker Raspberry Pi models was not tested.

To simplify the process, a list of all sensors and components used was compiled, along with their approximate prices.

Table 1 shows the sensors used in this project. They are relatively inexpensive, with a total cost of approximately 39 €, depending on the supplier. These include the ML 8511 (UV sensor), SI 1145 (UV index, visible and infrared sensor), TSL 2561 (ambient light sensor) and MPL 115A2 (temperature and pressure sensor).

Sensors	Price (€)
ML 8511	~9
SI 1145	~10
TSL 2561	~10
MPL 115A2	~10
Total	**~39**

Table 1. Sensors used in the station.

Additionally, table 2 provides a breakdown of the necessary components for operating the station, including a Raspberry Pi, Arduino Unos, and miscellaneous items such as wires, power cable, and breadboards. The total cost for these components comes to approximately 138 €, depending on the supplier.

Components	Price (€)
Raspberry Pi 5 (2GB)	~55
Micro SD Card (32GB)	~5
Arduino Uno	~30 × 2
Cables, others	~18
Total	**~138**

Table 2. Additional components to setup the station.

Finally, the source code and wiring diagrams for the station will be available on GitHub. The repository can be accessed through the following link: https://github.com/ved-tig/diy-radiation-monitoring-station.git [9].

7 Conclusion and possible improvements

During the internship, the qualitative behavior of key weather variables such as air pressure, temperature, and radiation were captured. It proved to be both very interesting and rewarding. Throughout its duration, valuable skills were acquired that will be useful in the future. The data collected met some of the expectations, although there are still aspects to improve, such as creating an autoexec file so that, in case of a power outage, the computer would automatically re-launch the app when rebooted, load the already processed data, and resume the measurements, making the process more continuous. Additionally, setting up an SSH (secure shell) connection would have enabled remote access to the data, allowing monitoring even while away.

Acknowledgements

The authors would like to thank Professor Luís Peralta and Tomás Campante for their supervision and guidance throughout the internship. Appreciation is also extended to LIP for providing the opportunity to carry out this internship.

References

[1] L.S. Co., *ML8511 UV Sensor with Voltage Output*, https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/ML8511_3-8-13.pdf (2013), retrieved at September 12th, 2025

[2] S.L. Inc., *Proximity/UV/Ambient Light Sensor IC with I^2C Interface*, https://www.silabs.com/documents/public/data-sheets/Si1145-46-47.pdf (2022), retrieved at September 12th, 2025

[3] T.A.O.S. Inc., *TSL2560, TSL2561 Light-to-Digital Converter*, https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf (2009), retrieved at September 12th, 2025

[4] F.S. Inc., *Miniature I^2C Digital Barometer*, https://cdn-shop.adafruit.com/datasheets/MPL115A2.pdf (2012), retrieved at September 12th, 2025

[5] T.I. Inc., *A Basic Guide to I^2C*, https://www.ti.com/lit/an/sbaa565/sbaa565.pdf?ts=1757524239200, retrieved at September 12th, 2025

[6] S. Campbell, *Basics of the I^2C Communication Protocol*, https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/, retrieved at September 12th, 2025

[7] C.L.V. Inc., *Histórico de condições meteorológicas em 27 de julho de 2025 em Lisboa*, https://pt.weatherspark.com/h/d/32022/2025/7/27/Condi%C3%A7%C3%B5es-meteorol%C3%B3gicas-hist%C3%B3ricas-em-domingo-27-de-julho-de-2025-em-Lisboa-Portugal#metar-01-00, retrieved at September 2nd, 2025

[8] I.P. do Mar e da Atmosfera, *Índice UV observado- Lisboa/ Aeroporto*, https://www.ipma.pt/pt/index.html, retrieved at July 29th, 2025

[9] *DIY Radiation Monitoring Station*, https://github.com/ved-tig/diy-radiation-monitoring-station.git