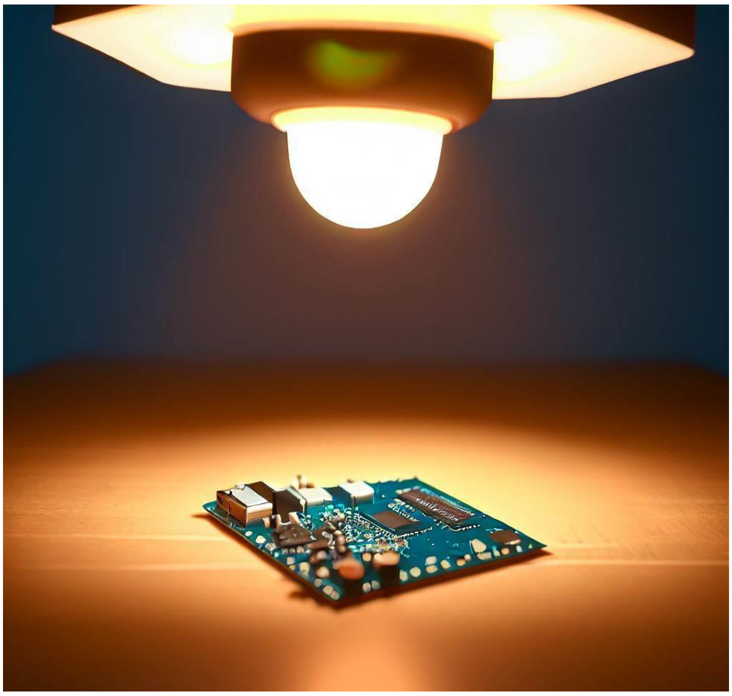
Internship report

3 APRIL 2023 – 28 JULY 2023



Detection and processing of visible-light signals for indoor navigation

Internship supervisor:
Mr. Ernesto **CIARAMELLA**

Polytech supervisor:
Mrs. Nancy **EL RACHKIDY**

Host organisation:
TeCIP Institute, Scuola Superiore Sant'Anna
Via Giuseppe Moruzzi, 1, 56124 Pisa PI, Italia

Institution:
Polytech Clermont
2 Av. Blaise Pascal, 63100 Aubière, France

Louis FOUCHER
ELECTRICAL ENGINEERING – 4A – OPTION B | 2022-2023

Thanks

Before going any further into this professional experience, it seems appropriate to begin this report by thanking those who taught me so much during the course to make it a very worthwhile experience.

First of all, I would like to thank my tutor, Professor Ernesto CIARAMELLA, of the Scuola Superiore Sant'Anna, at the TeCIP institute, for allowing me to do this placement, for the time he gave me and for all the knowledge I was able to acquire during these four months.

I would also like to thank Doctor Giulio COSSU who also supervised me during my internship, as well as all the Institute members for their welcome and the time they gave me.

And, I would also like to thank Mrs. Nancy EL RACHKIDY, my Polytech tutor, for following me during my four months of placement.

Résumé

Ce rapport présente le travail réalisé dans le cadre du sujet « **Detection and processing of visible-light signals for indoor navigation** ». L'objectif a été de créer un système permettant le positionnement intérieur, dans un bâtiment, d'une personne ou d'un objet, en utilisant des lampes LED modulées en fréquence, une carte Arduino et une photodiode. Les lampes servent d'émetteurs de signaux optiques, la photodiode sert de récepteur de signaux optiques et la carte Arduino sert de contrôleur pour analyser et traiter le signal, afin de fournir une position. Ce projet s'est déroulé en plusieurs étapes. Tout d'abord, le choix de la carte Arduino a été effectué. Ensuite, pour traiter la partie du positionnement, il a fallu commencer par être capable de détecter la présence sous une lampe. L'ajout d'une deuxième lampe a permis de déterminer sous quelle lampe se trouve le capteur. Enfin, en ajoutant une troisième lampe, la position du capteur a pu être triangulée, pour fournir ses coordonnées X et Y par rapport à un point d'origine choisi. Le système est alors capable de fournir les coordonnées du capteur et donc sa position.

Mots clés :

- Positionnement
- Traitement du signal
- Communication par lumière visible
- Arduino
- Analyse

Abstract

This report presents the work carried out as part of the "**Detection and processing of visible-light signals for indoor navigation**" project. The aim was to create a system for positioning a person or object inside a building, using frequency-modulated LED lamps, an Arduino board and a photodiode. The lamps act as optical signal transmitters, the photodiode as optical signal receiver and the Arduino board as controller to analyse and process the signal to provide a position. This project took place in several stages. First, the Arduino board was chosen. Then, to process the positioning part, we had to start by being able to detect the presence of a lamp. The addition of a second lamp made it possible to determine under which lamp the sensor was located. Finally, by adding a third lamp, the position of the sensor could be triangulated to provide its X and Y coordinates relative to a chosen point of origin. The system is then able to provide the coordinates of the sensor and therefore its position.

Key words :

- Positioning
- Signal processing
- Visible light communication
- Arduino
- Analysis

Table of contents

Table of figures

Table of tables

Glossary

ADC : An electronic device that converts an analog signal into a digital signal. It transforms physical quantities such as voltages or currents into digital values that can be processed by a digital system.

Analog signal : Continuous representation of a physical quantity. It is characterised by an infinite number of possible values, which means that it can vary fluidly and precisely.

Bandwidth : Capacity or maximum speed of a communication channel for transferring data.

Bluetooth : Short-range wireless communication technology that enables data to be exchanged between electronic devices. It uses radio waves to establish reliable and secure connections.

Camera vision : Use of cameras to capture modulated light signals in order to receive the transmitted data. Cameras are used as optical receiving devices to detect and interpret light variations and convert optical signals into digital data.

FFT : Mathematical technique used to analyse a signal by breaking it down into its constituent frequency components. It allows us to move from a time domain to a frequency domain by calculating the amplitudes and phases of the different frequencies present in the signal.

Infrared : Electromagnetic radiation with a wavelength longer than that of visible light. This radiation is invisible to the naked eye but can be detected using special devices.

Li-Fi : Wireless communication technology that uses visible light to transmit data. It works by modulating light intensity at extremely high speeds, invisible to the naked eye.

Optical fibre : Cable made up of thin glass or plastic wires used to transmit light signals. These light signals, generally laser or LED pulses, carry data in the form of optical signals.

OWC : A communication method that uses optical signals to transfer information between devices. It uses technologies such as light intensity modulation or phase modulation to transmit data. OWC can use infrared or visible light sources, such as LEDs or lasers, to transmit optical signals.

RFID : Technology using RFID tags to identify and track objects remotely using radio signals. RFID tags are small electronic chips that contain information specific to the object to which they are attached.

Ultrasound : High-frequency sound waves, higher than the human ear can perceive. They are generated by special transducers that convert electrical signals into sound vibrations.

VLC : Technology that uses visible light to transmit data. It works by modulating light intensity at high speeds, enabling information to be transferred via rapid variations in light.

WDM : This involves mixing several optical signals on the same optical fibre in order to multiply the fibre's bandwidth.

Wi-Fi : A wireless communication standard that uses radio waves to establish connections between devices. It allows devices to connect to a local network or the internet without using physical cables.

List of signs and abbreviations

ADC	Analog-to-Digital Converter
FFT	Fast Fourier Transform
LED	Light-Emitting Diode
Li-Fi	Light Fidelity
OWC	Optical Wireless Communication
RFID	Radio Frequency IDentification
VLC	Visible Light Communication
WDM-LED	Wavelength Division Multiplexing-Light Emitting Diode
Wi-Fi	Wireless Fidelity

Introduction

These days, technology is all around us, and we use it every day, whether to telephone, send messages, watch a series or locate ourselves. Nowadays, it's essential to be able to locate yourself using the GPS on your phone, but it's not precise enough if you want to locate someone or something, or yourself, in a building such as a museum or hospital. This internship is in the field of indoor positioning, which is the subject of my internship at the TeCIP institute of the Scuola Superiore Sant'Anna: "**Detection and processing of visible-light signals for indoor navigation**". The aim of this project was to develop a system for locating an object using commercially available LED lamps, a photodiode and an Arduino microcontroller.

This report aims to show what I have realized and developed during my internship.

After a presentation of the school and institute where the work placement took place, will be developed the various stages of research and development involved in achieving indoor positioning using LED lamps. Finally, a feedback on this internship will be provided with my personal point of view and the lessons that it taught me.

1. The university

1.1. Scuola Superiore Sant'Anna

Figure 1 : Scuola Superiore Sant'Anna

The Scuola Superiore di Studi Universitari e di Perfezionamento Sant'Anna di Pisa was founded in 1987 as a university institution with a special system, and looks back on a long tradition linked to the city of Pisa and to the history of its university and colleges.

In 1987, the Scuola Superiore Sant'Anna thus becomes the second Higher School of Pisa, on the model of the Scuola Normale Superiore, and is dedicated to the study of applied sciences: Economic and managerial sciences, Legal Sciences, Political Sciences, Agricultural Sciences and Plant Biotechnology, Medical Sciences and Industrial and Information Engineering.

At the same time, the School becomes part of the Italian university system as a university with a special order. Its headquarters is the Sant'Anna Conservatory, established in Pisa in 1785 by Grand Duke Pietro Leopoldo of Lorraine.

Over the years the Sant'Anna School of Advanced Studies has achieved not only national but also international relevance in the academic and scientific panorama, in line with challenges presented by the cultural, social, technological and scientific context.

The Sant'Anna school is a free public university dedicated to applied sciences and based on the valorisation of merit and talent. The fields of study and research are divided into two academic classes: the Social Sciences class, with the fields of Economics, Law and Political Sciences, and the Experimental Sciences class, with the fields of Agricultural Sciences and Plant Biotechnology, Engineering, Medicine and Surgery.

This school offers undergraduate and postgraduate courses that are accessed by competitive examination, in order to ensure the level of excellence that the School pursues at the institutional level:

- **Honor Courses:** first and second level undergraduate courses, which integrate and complete the courses of the University of Pisa with a training of excellence and an early start to research.
- **Master's degrees:** courses in agreement with Italian and foreign universities, related to highly specialized training projects, with strong connotations in terms of interdisciplinary, internationalization and innovation.
- **Master's Diplomas and Higher Education:** ostgraduate training courses aimed at deepening knowledge in innovative sectors and of particular strategic importance to the industry.
- **PhD Programmes:** aimed at university graduates, they are focused on research areas of particular scientific and social relevance.
- **Seasonal Schools:** training courses of excellence with a strong interdisciplinary character, focused on the frontier research themes of the School.

Research is conducted in institutes, departments of excellence and interdisciplinary research centres:

- Institute of Biorobotics
- Institute of Law, Policy and Development (DIRPOLIS)
- Institute of Economics
- Institute of Management
- **Institute of Telecommunications, Computer Engineering, and Photonics (TeCIP)**
- Institute of Mechanical Intelligence
- Research Centre for Plant Production
- Plant Science Research Centre
- Department of Excellence in Economics and Management in the Age of Data Science (EMBEDS)
- Department of Excellence Robotics and AI
- Interdisciplinary Research Centre for Health Sciences

1.2. TeCIP institute

The Telecommunications, Computer Engineering, and Photonics Institute (TeCIP) of the Scuola Superiore Sant'Anna in Pisa was created in 2001 as a Centre of Excellence funded by the Italian University, Scientific and Technological Research Ministry.

The core research domains in the TeCIP Institute are:

- Telecommunication networks, systems and components implemented with partial or full use of photonic technologies and relevant software control techniques;
- Cyber-physical systems, real-time computing, artificial intelligence and cyber-security;
- Photonic Integrated Circuits for datacom, sensors and networks and biophotonics.

The research area works on numerical modelling, design and experimental realizations of innovative system solutions, including propagation effects. It covers optical fiber communications and the rapidly emerging area of optical wireless communications.

The area has a long experience in fiber systems, also gained in industrial context. The research team has been active in optical system and subsystem developments for both core and access networks. In this field, it took part in various EU-sponsored projects, including one (FP7-COCONUT), of which it was the coordinator. In this area, the researchers also realized several innovative demonstrations of all-optical processing functionalities.

In the last ten years, the area has expanded its research interest to optical wireless communications. It organized and hosted the ever-first demonstration of a 1.2 Terabit/s (32x40 Gbit/s) free space optical system, in outdoor environment. In the following it made several break-through demonstrations of visible-light communications (VLC), showing the first Gbit-class system made of a common LED and the first WDM-LED link. In this area it also demonstrated underwater VLC in a real harbour, using its home-made VLC modems. Recently, it is deeply involved in developing intra-spacecraft terminals by using OWC in the infrared region, these terminals are today being tested in a real spacecraft. It is also strongly involved in the merging area of OWC for inter-satellite communications and for deep space. Most of these activities are carried out within the framework of Hydron initiative run by ESA.

2. Mission

2.1. Context

Today we want to know our location and the location of objects around us. For examples, it can be to know if the bus arrives soon, to have real time information on the frequentation of certain places. That's why I was given the project "Detection and processing of visible-light signals for indoor navigation" which consists in putting an electronic map on a robot with a photodiode and to be able to locate inside where the robot is thanks to the lamps present to light the place.

2.2. Indoor positioning

The principle of indoor positioning is to know the position of an object, as GPS does outdoors, but inside an enclosed space, where GPS signals are unavailable or unreliable. There are several solutions for indoor positioning, such as :

- Wi-Fi
- RFID
- Bluetooth
- Ultrasound
- Infrared
- Camera vision
- VLC (Visible Light Communication)

Given the field of research in my laboratory, I'm going to use and explain the use of VLC for indoor positioning. Indoor positioning using optical communication systems such as Li-Fi (Light Fidelity) can be either frequency-modulated or amplitude-modulated. To use Li-Fi, you need a receiver, which can be a telephone camera, a photodiode or a device specially designed to integrate Li-Fi receivers. You also need a transmitter to generate a modulated signal, which can be LED lamps, dedicated Li-Fi bulbs or light panels.

2.3. Experimental setup

This project is the continuation of a previous project using a telephone camera. This project is intended to be used on robots or hospital beds, for example. So, I'll be using LED lamps as transmitters, which are frequency modulated and a photodiode as receivers. A microcontroller will be used to operate the system. The setup looks like this :

Figure 4 : LED lamps

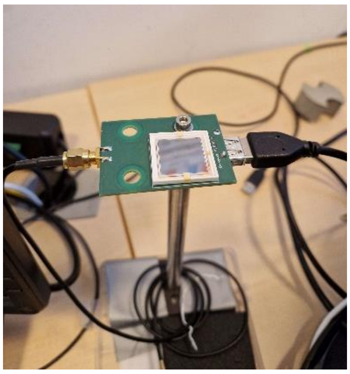


Figure 3 : Photodiode

Figure 2 : Setup as a whole

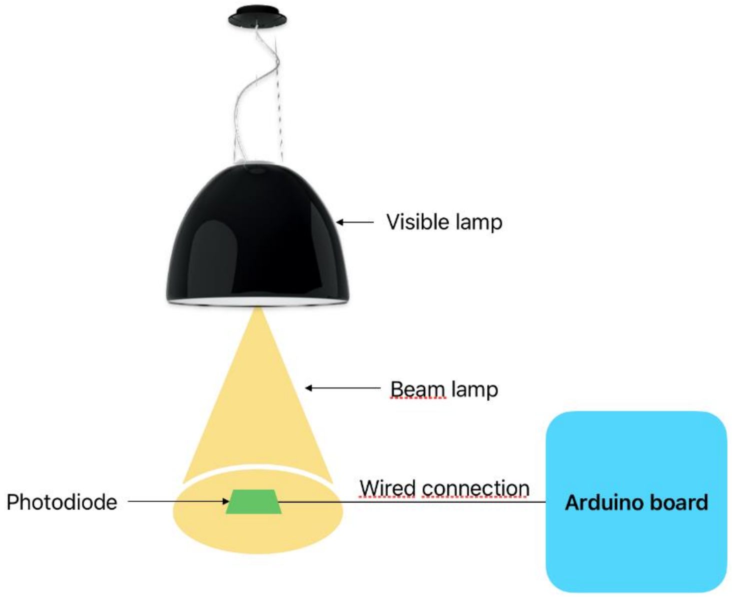


Figure 5 : Experimental setup diagram

The first step was to choose the microcontroller that would run the project.

2.4. Choice of board

I had at my disposal several models of board of the Arduino brand:

- Arduino UNO
- **Arduino Nano RP2040 connect**
- Arduino MKR WiFi 1010
- Arduino Nano 33 BLE

Figure 6 : Arduino UNO

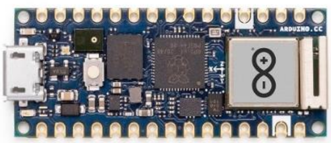


Figure 7 : Aduino Nano RP2040 Connect

Figure 9 : Arduino MKR WiFi 1010

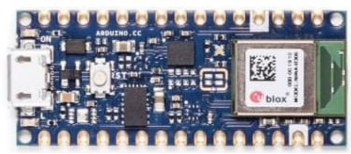


Figure 8 : Aduino Nano 33 BLE

Table 1 : comparison of the different boards

Name	Chip	Memory		Clock
		FLASH	SRAM	
Arduino UNO	ATmega328P	32 KB	2 KB	16 MHz
Arduino Nano RP2040 connect	RP2040, DUAL CORE ARM CORTEX M0+	16 MB	264 KB	133 MHz
Arduino MKR WiFi 1010	SAMD21 Cortex®-M0+ 32bit low power ARM® MCU	256 KB	32 KB	48 MHz
Arduino Nano 33 BLE	nRF52840	1 MB	256 KB	64 MHz

I started to get to grips with the subject using the Arduino Uno, as this is what I was most comfortable with. But I quickly realized that this board was not powerful enough to sample the light signal of the lamp which can provide a modulated signal of 8 kHz, so to be able to sample it correctly is to respect Nyquist's theorem, which says that the sampling frequency must be at least twice as high as the maximum frequency that we want to observe, I had to change board. Another important point is that the card must have a fairly high flash memory because it must be able to store tables of values containing a large number of samples in order to be able to obtain usable results.

So I studied the boards available to me to find the one that would be the best for this project. I opted for the Arduino Nano RP2040 Connect board, this board embeds a Rasberry Pi 2040 microcontroller, clocked at 133 MHz. Moreover, it is equipped with 16 MB of Flash memory and 264 KB of SRAM memory, which is more than enough to store the values needed for the analysis to work properly.

Once the board has been chosen, it's time to start programming the system.

2.5. Lamp differentiation

For this step, I first had to make the ADC of the Arduino board sample at a frequency of 100 kHz which equates to 100k samples per second. To do this I use the ADCInput library from the board package Raspberry PI Pico/RP2040 by Earle F. Philhower, which allows me to configure the sampling frequency of the ADC of my Arduino, this is done with the following command line:

```
adc.setFrequency(samplingFrequency);
```

Here is a diagram of the system:

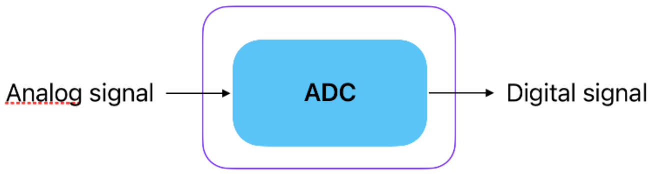


Figure 10 : Système diagram

I sample the frequency-modulated light signal supplied by the lamp by filling an array with 8192 samples of this signal so that I can then calculate the FFT and analyse it. To calculate the FFT, I use Enrique Condes' arduinoFFT library, using the following command lines:

```
(1) FFT.Windowing(vReal, numberOfSamples, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
    //Compute of the FFT
(2) FFT.Compute(vReal, vImg, numberOfSamples, FFT_FORWARD);
    //Compute of the amplitude of each values
(3) FFT.ComplexToMagnitude(vReal, vImg, numberOfSamples);
```

1. Used to window the Fourier transform to reduce the noise present in the sampled signal, making the characteristics of the signal clearer and easier to interpret. In the function parameters, I enter the values I have retrieved from my ADC, the number of samples I'm doing, as well as the type of windowing I want to apply, in this case Hamming, and whether I'm doing a direct or inverse Fourier transform, in this case direct.

2. Is used to calculate the FFT, with parameters such as the ADC values (vReal), the pure imaginary values which are currently only 0 (vImg), the number of samples and here again the FFT direction.
3. The last line is used to calculate the modulus for each value, so I enter the real values (vReal), the imaginary values (vImg), which correspond to the result of the FFT calculation in the complex domain from the previous command, and the number of samples as parameters for the function. The function then returns the result of the module for each value on the vReal array, which avoids having to create a new array of values. I then retrieve these values and put them in dB to produce a graph representing my FFT with the frequency on the x-axis and the amplitude in dB on the y-axis. I can observe the peaks corresponding to the flicker frequencies of the lamps and see that these peaks are located at the right frequencies.

By adding this new step, the system looks like this:

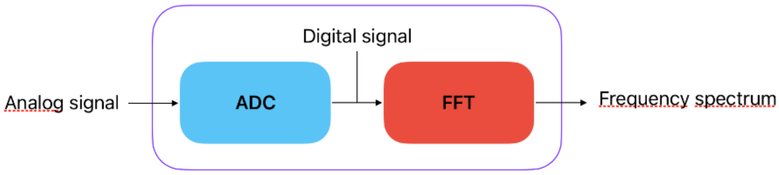


Figure 11 : System diagram step 2

Here the lamp has been configured with the following two frequencies :

- Frequency 1 : 2500 Hz
- Frequency 2 : 6000 Hz

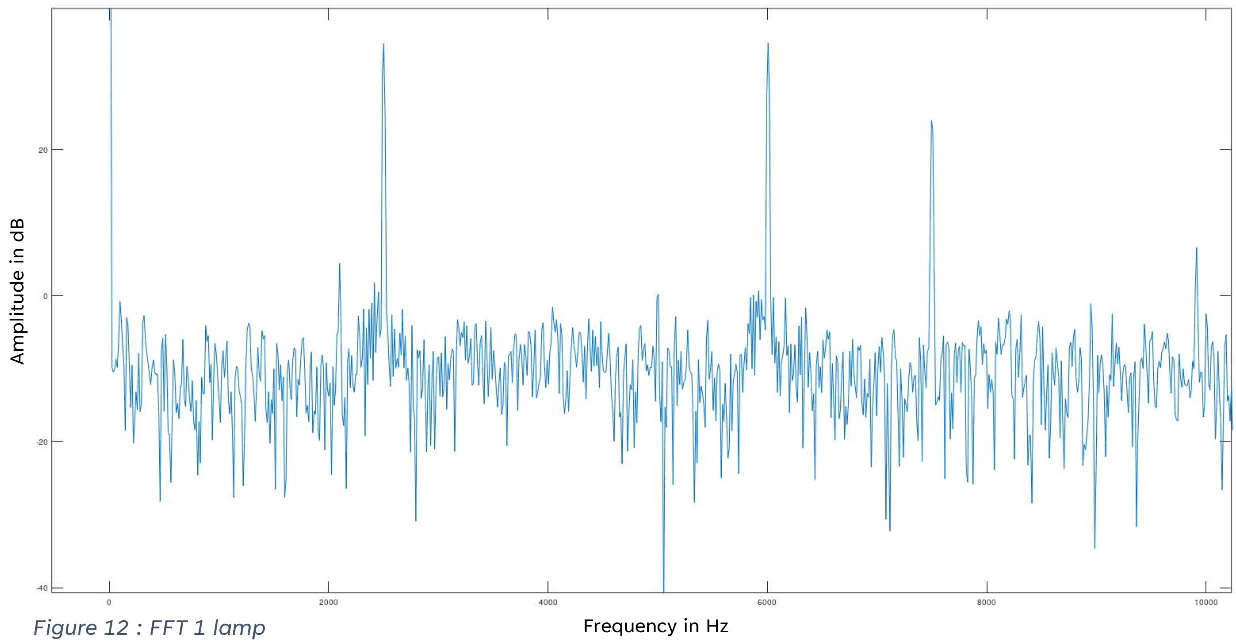


Figure 12 : FFT 1 lamp

On the graph, we can see two frequency peaks, one at 2500 Hz and the other at 6000 Hz. We observe a third peak which corresponds to a harmonic, whose frequency is 3x*Frequency1*, i.e. 7500 Hz. This harmonic is present because the lamp sends a square signal and not a sinusoidal one. Other harmonics are present in the FFT but we don't see them here because the lamps cannot provide a frequency greater than 8000 Hz, there is no point in looking further into the frequencies as we are not interested.

```
Peak 1: Frequency : 2502        Amplitude : 34.20
Peak 2: Frequency : 6006        Amplitude : 34.31

In the new order
Peak 1: Frequency : 6006        Amplitude : 34.31
Peak 2: Frequency : 2502        Amplitude : 34.20

Under the lampe 1
Frequency : 6006        Amplitude : 34.31
Frequency : 2502        Amplitude : 34.20
----------------------------------------------------
```

Figure 13 : Serial Monitor with 1 lamp

I display the peaks found in the serial monitor. You can see that the peaks are at about 2500 Hz and 6000 Hz and that their amplitudes are almost the same and worth about 34 dB. Here the transducer is located just below the lamp, at a distance of about 1m50. For this distance between the lamp and the sensor we have the maximum amplitude achievable.

The program works with one lamp, it is the same principle with two lamps. Instead of having 2 peaks, we will have 4 peaks (without counting the harmonics). We get the following results for a frequency configuration such as :

- Lamp 1 :
  - Frequency 1 : 2500 Hz
  - Frequency 2 : 6000 Hz
- Lamp 2 :
  - Frequency 1 : 4000 Hz
  - Frequency 2 : 5500 Hz

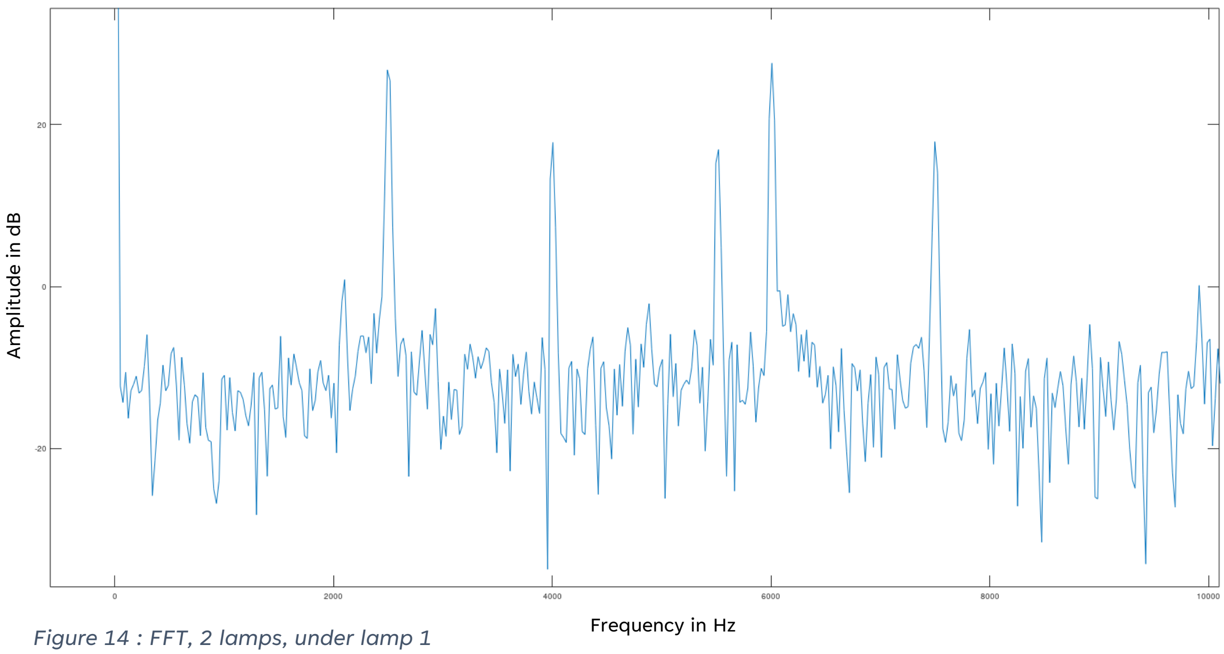


Figure 14 : FFT, 2 lamps, under lamp 1

```
Peak 1: Frequency : 2502.441406 Amplitude : 34.600277
Peak 2: Frequency : 4003.906250 Amplitude : 26.191710
Peak 3: Frequency : 5505.371094 Amplitude : 25.992742
Peak 4: Frequency : 6005.859375 Amplitude : 34.732004

In the new order
Peak 1: Frequency : 6005.859375 Amplitude : 34.732004
Peak 2: Frequency : 2502.441406 Amplitude : 34.600277
Peak 3: Frequency : 4003.906250 Amplitude : 26.191710
Peak 4: Frequency : 5505.371094 Amplitude : 25.992742

Under the lampe 1
Frequency : 6005.859375 Amplitude : 34.732004
Frequency : 2502.441406 Amplitude : 34.600277
```

Figure 15 : Serial monitor with 2 lamps, under lamp 1

We notice that our FFT is composed of two pairs of peaks and that each pair has about the same amplitudes. As the sensor is always located under lamp 1, we can see that the pair of peaks with the greatest amplitudes is the one corresponding to the frequencies of lamp 1, which corresponds to where the sensor is located. If we place the sensor under lamp 2, then we get the following result:

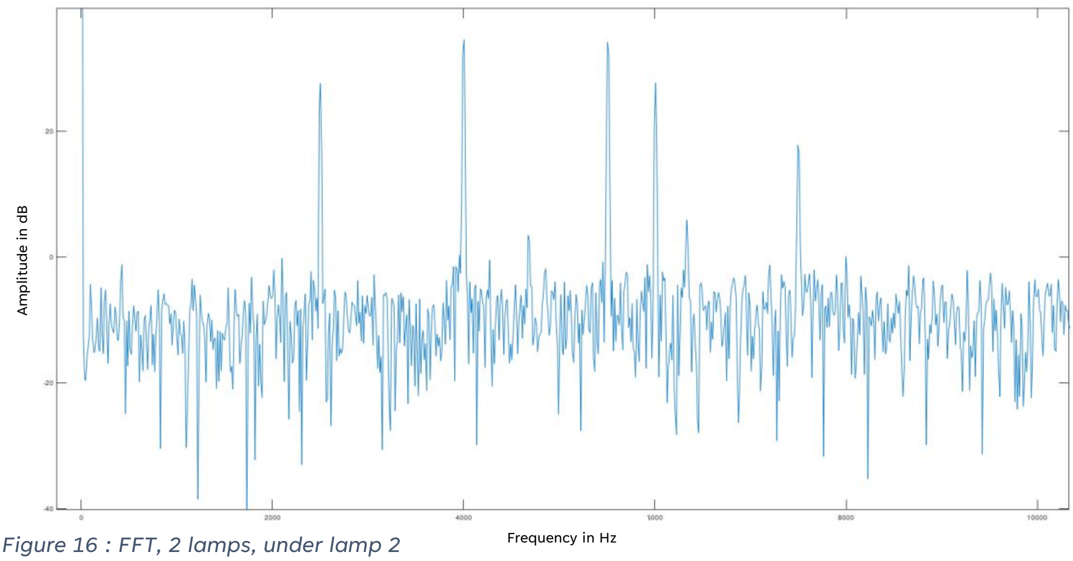


Figure 16 : FFT, 2 lamps, under lamp 2

```
Peak 1: Frequency : 2502        Amplitude : 26.38
Peak 2: Frequency : 4004        Amplitude : 33.98
Peak 3: Frequency : 5505        Amplitude : 33.70
Peak 4: Frequency : 6006        Amplitude : 25.14

In the new order
Peak 1: Frequency : 4004        Amplitude : 33.98
Peak 2: Frequency : 5505        Amplitude : 33.70
Peak 3: Frequency : 2502        Amplitude : 26.38
Peak 4: Frequency : 6006        Amplitude : 25.14

Under the lampe 2
Frequency : 4004        Amplitude : 33.98
Frequency : 5505        Amplitude : 33.70
----------------------------------------------------
```

Figure 17 : Serial monitor with 2 lamps, under lamp 2

Here the sensor is under the lamp 2, we notice that the amplitudes of the peaks are reversed between the couple corresponding to the lamp 1 and that corresponding to the lamp 2. The program therefore works correctly and allows us to tell whether we are under lamp 1 or lamp 2, using the amplitudes of the peaks.

2.6. Code improvement and testing

2.6.1. Making the programme more robust

In order to have a more robust programme, I tried and used several solutions. The problem was that there was a lot of noise in my signal, which can be seen in the figures above. This noise can be problematic because if it gets too high the programme could confuse the noise with the signal. To reduce this noise I used the averaging method. I first tried to average over ten data acquisitions by summing the moduli for each frequency and then dividing by the number of acquisitions, in this case ten. This gave me the following result:

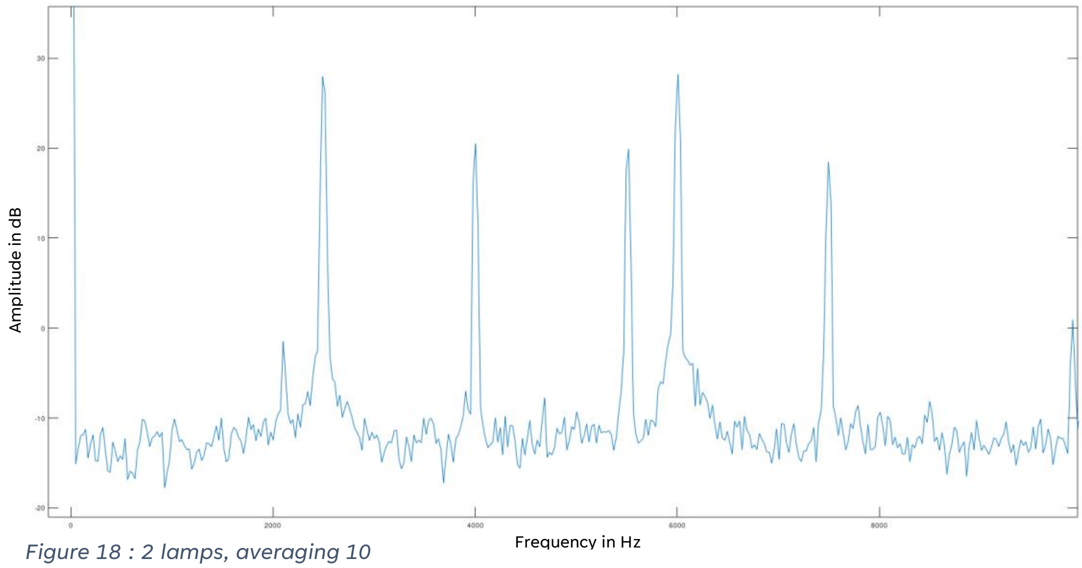


Figure 18 : 2 lamps, averaging 10

We can see that compared with Figure 18, the noise has been significantly reduced. But the downside of this method is that as the data is acquired and the FFT calculated several times, you have to wait almost 10 seconds before getting the result, taking 8192 samples. I also did the test by making 100 acquisitions:

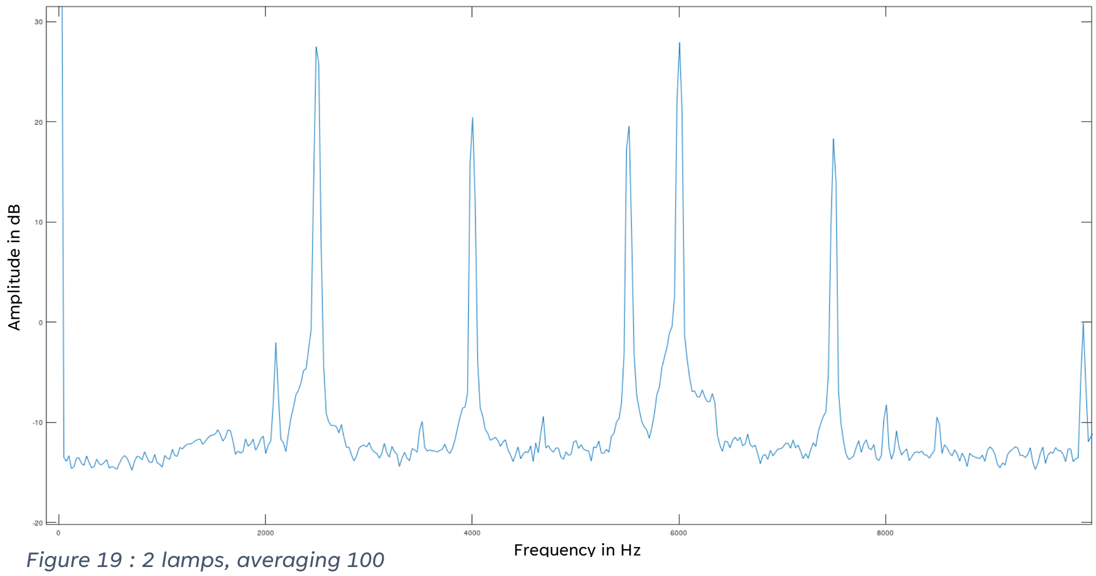


Figure 19 : 2 lamps, averaging 100

As you can see, the accuracy has improved still further, but the time taken to get the result has also increased, since it will take more than a minute to get the result. The precision gained compared to the time lost was not great enough, so I stayed with an averaging of 10 acquisitions.

I also wanted to test averaging not the module but the FFT results in the complex domain. This didn't work because the lamps used and the Arduino don't have the same phase and so I lost not only the noise but also the peaks in my signal when I plotted the FFT.

So I went back to my first attempt, which is working, with 10 acquisitions. To make the programme even more robust, instead of averaging over the module, I average over the module squared, which gives a greater gap between the maximum noise and the maximum peaks and therefore greater reliability.

2.6.2. Minimized surface test

Then I did some tests by changing the acquisition area of the sensor, using an opaque box and reducing the acquisition area between each result and see when we are able to see the peaks and analyse them. I took the results by making 100% of the surface, 75%, 50%, 25%, 0%, from right to left, then the same thing starting from left to right. The sensor is placed under lamp 2, roughly in the middle of the table on the right. The results of this test give me the following graph:

Figure 20 : minimized surface of sensor

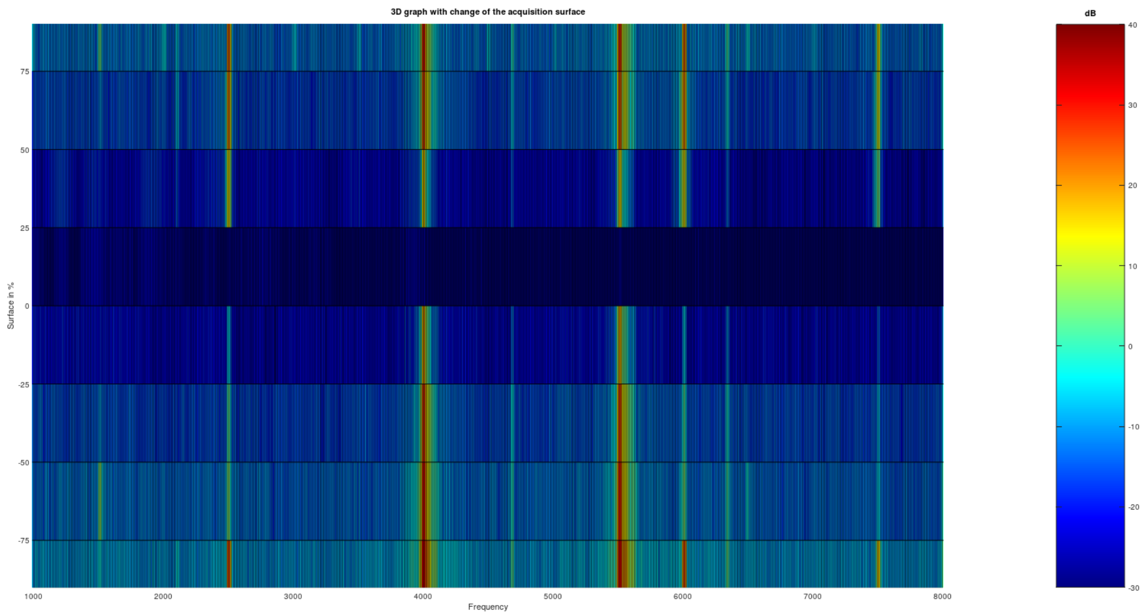


Figure 21 : plot minimized surface of sensor

On this graph we can see the frequency peaks, which correspond to the yellow bands on the graph. You can see that in the first part, from 100% to 25% on the vertical axis, the 4 peaks fade slightly as the surface shrinks, but they're there and you can see them clearly. From 25% to 0% is when the sensor is completely covered by the box and you can see that there are no peaks because the entire sensor is covered. Then we reopen the surface of the sensor a little and we immediately notice the 2 major peaks that correspond to the lamp we're standing under. But we also notice that the peaks corresponding to the lamp 1 are very attenuated, this is due to the fact that the box used for the test creates a shadow which prevents the sensor from seeing the signal correctly.

2.6.3. Angle of sensor test

To check the robustness of the system, I also tested whether we still had a usable result by setting an angle to the sensor. I tested with the sensor under lamp 2, using five different angles: 90°, 45°, 0°, -45°, -90°. The sensor is placed under lamp 2, roughly in the middle of the table on the right. Here's the resulting graph:

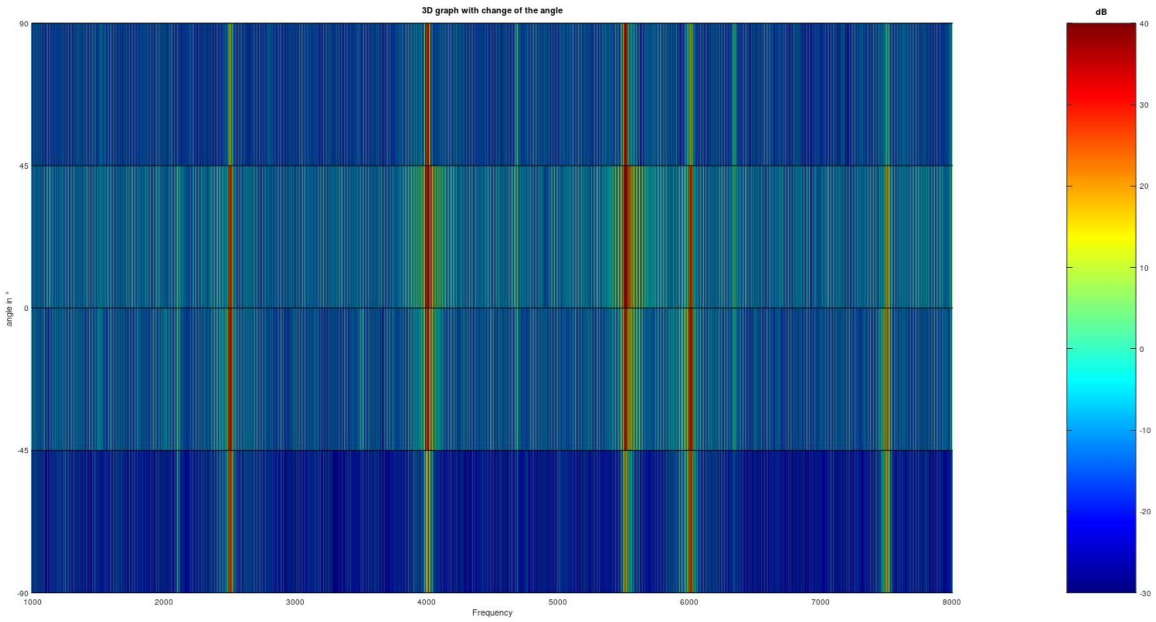
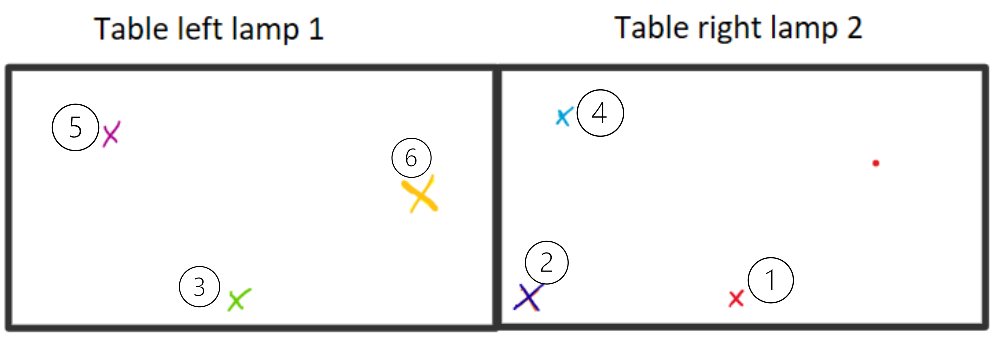


Figure 22 : plot with change of angle of sensor

As you can see from this graph, I can see the four peaks in all positions, with different amplitudes of course. As I'm basing the position of the sensor on the amplitudes of the peaks, we can see that there will be an error if the sensor is turned towards lamp 1, because the light signals received from lamp 1 will be more direct than those from lamp 2, under which the sensor is located. But you can see the 4 peaks in all cases, which is a good thing from the point of view of the robustness of the system.

2.6.4. Effect of reflections



Lamp 2 :

		2		
1	Without = 34 With = 32	Without = 32,5 With = 30,5	Without = 25 With = 25	3
4	Without = 32,4 With = 30,7	Without = 23 With = 21	Without = 28,6 With = 27,4	6
		5		

Lamp 1 :

		2		
1	Without = 26,7 With = 26,2	Without = 31,5 With = 30	Without = 34 With = 32	3
4	Without = 29 With = 28	Without = 33 With = 31,5	Without = 33,5 With = 32	6
		5		

Figure 23 : Reflexion effect

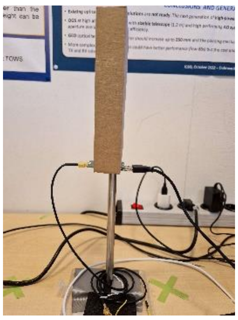


Figure 24 : Sensor with box

Figure 25 : Sensor without box

These results show the difference in amplitude of the frequency peaks when I put the box on the sensor, which produces the effect of a spatial filter by taking only the direct light rays for each of the lamps. To get the best results, I tilted the box either towards lamp 1 or towards lamp 2 to always get the greatest amplitude possible, to get significant results. You can see that the effect of reflection on the amplitude is about 2 dB and that this difference decreases the further the sensor is from the light source.

2.6.5. Improve speed

Looking at the results presented above, we can see that the time taken to obtain a result is a little long, due to the large number of data acquisitions. As a reminder, I acquired 8192 samples 10 times. The programme isn't fast enough because of the large number of steps involved in calculating the FFT, which is the step that takes the longest in the programme, around

800 ms for each calculation. Since this calculation is done 10 times, the programme takes around 8 seconds before giving a result.

Below is a graph showing the time between the start of the programme and the moment when the results are displayed:

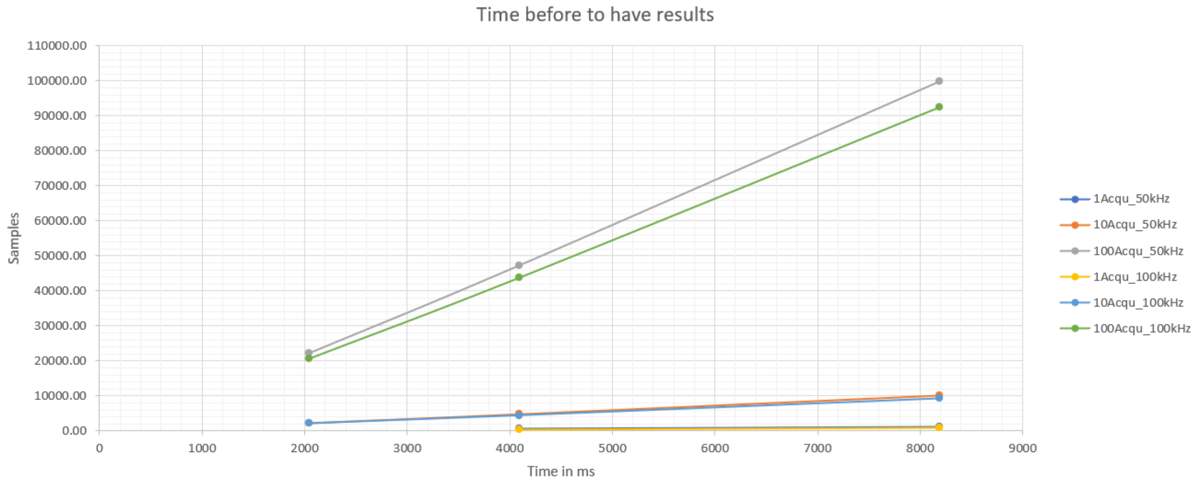


Figure 26 : Run time graph

As can be seen from the graph, the time taken to obtain the results fluctuates greatly depending on the number of samples. The number of samples doesn't go below 2048, because if we use fewer samples the precision is too low for the results to be correct.

Initially, I use a sampling frequency of 100,000 Hz, but this figure can be reduced to just twice the maximum frequency according to Nyquist's theorem, $f_s \geq 2f_{max}$, which states that the maximum frequency is equal to half the sampling frequency. Theoretically, with a sinusoidal signal, as the maximum observable frequency is 8 kHz, a sampling frequency of 16 kHz is sufficient. Here, we are observing a square-wave signal with harmonics worth 3xPeakFrequency, which means that for a peak at 2 kHz, we will have a harmonic at 6 kHz. The highest observable frequency is 8 kHz, so the highest harmonic will be at 24 kHz. You could then say that a sampling frequency of 25 kHz is sufficient, but remember Nyquist's theorem: the highest displayable frequency is equal to half the sampling frequency, so the highest frequency would be 12.5 kHz. This is when the phenomenon of aliasing comes into play. It is characterised by the folding of the spectrum of the DC signal onto itself. This will disturb the frequency range that I observe, as shown below:

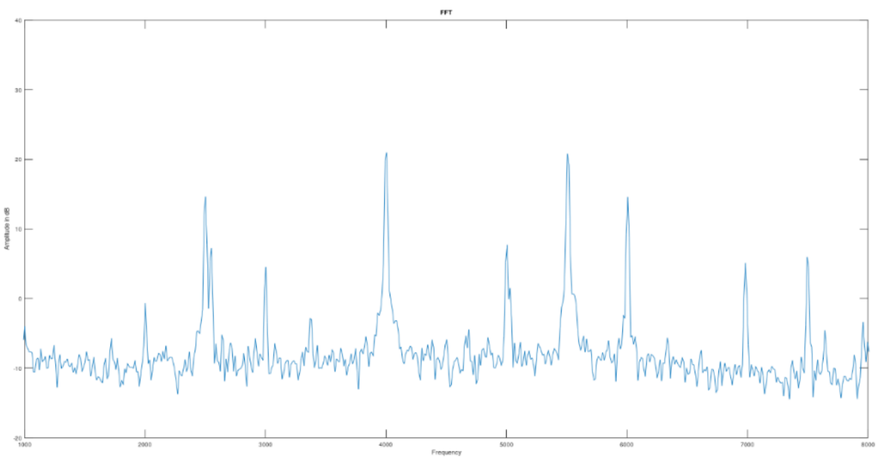


Figure 27 : Aliasing

For this phenomenon not to occur, I need at least a 48 kHz sampling frequency so that the largest possible harmonic is not subject to aliasing. I've therefore set my sampling frequency to 50 kHz to increase speed, but I can't go below this frequency for technical reasons.

In order to be faster, I also reduced the number of samples to 4096, which halved the calculation time since the number of samples was divided by 2. Moreover, I chose 4096 samples because this number of samples had the best precision-computation time ratio.

2.7. XY position

2.7.1. Lamp beam

The aim of this stage is to find out the light field of the lamps I'm using. I took measurements of the amplitude of the signal received at several known distances, for example every 5 or 10 cm starting from 0, i.e. just under the lamp where the light signal received is strongest. We then plot the curve representing the light field from the lamp and deduce an equation, which is not exactly the same for all lamps. In this installation, two out of three lamps have the same light field. Using the amplitudes of the FFT peaks, we can find out how far away the sensor is, which brings us to the calculation part to find the location of the sensor.

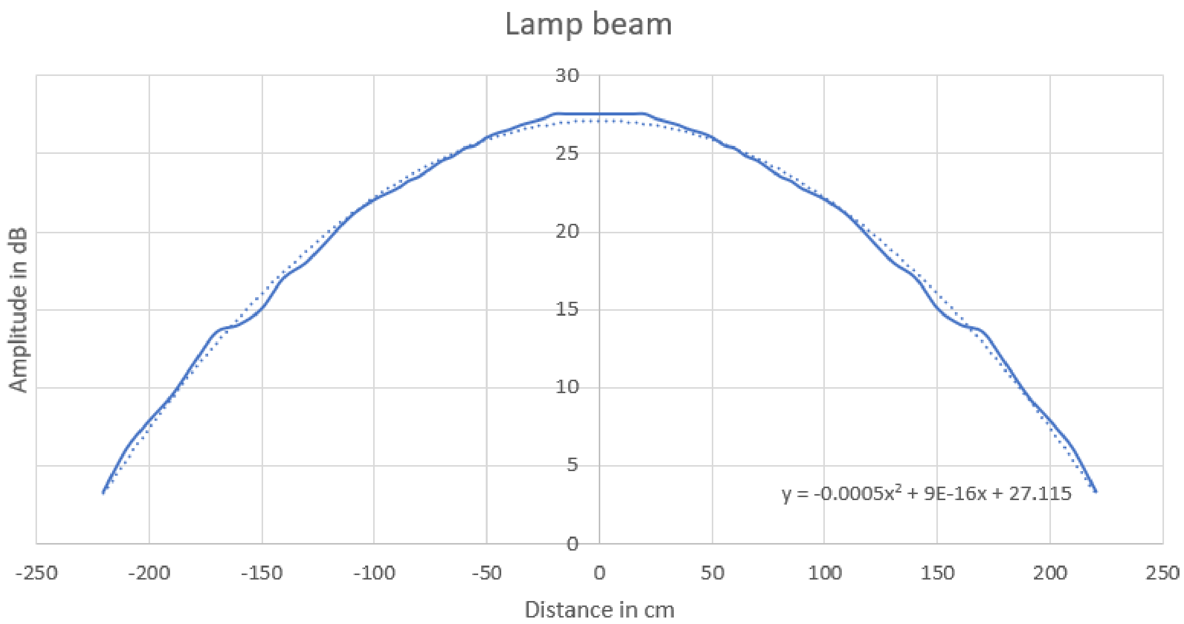


Figure 28 : Lamp beam plot

2.7.2. Trilateration

Trilateration is a method of locating an object using distance. This method is similar to triangulation, except that instead of measuring angles, it involves measuring distances.

The test environment for XY positioning is shown in the diagram below:

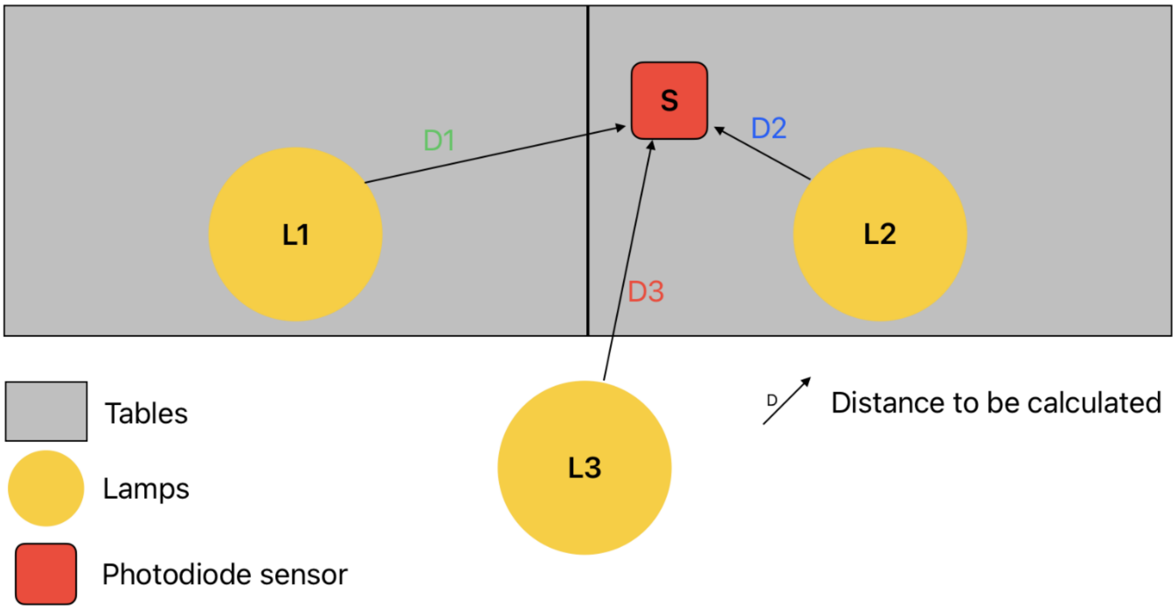


Figure 29 : test environment XY positionning diagram

This method consists of measuring the distances between three points with known coordinates and the point with unknown coordinates. Here the three known points are the lamps and the unknown point is the photodiode. To find out the coordinates we will use the amplitude of the peaks present in the FFT corresponding to each of the lamps. Using the equations from the previous step, we can find the distance between each lamp and the photodiode, D1, D2, D3. This gives us the coordinates (x; y) of the sensor, as shown below:

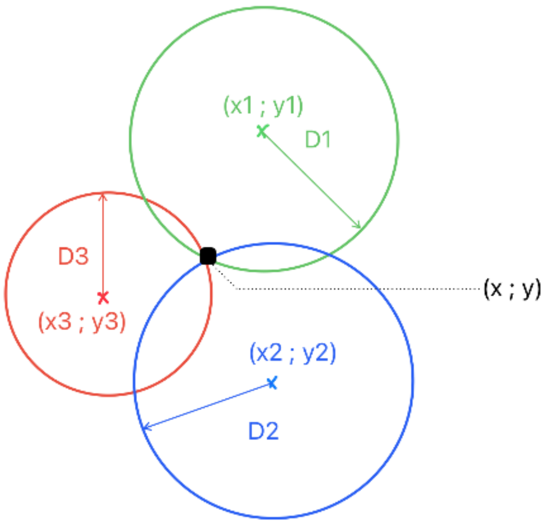


Figure 30 : trilateration operating diagram

On the output we read the amplitudes of the 6 peaks of the FFT, which will give us 3 distances by averaging the two amplitudes corresponding to a lamp, and then I display the result of the trilateration, i.e. the coordinates of the unknown point, the sensor.

```
Peak 1: Frequency : 1501        Amplitude db : 21.69
Peak 2: Frequency : 2502        Amplitude db : 21.11
Peak 3: Frequency : 4004        Amplitude db : 27.48
Peak 4: Frequency : 4614        Amplitude db : 20.02
Peak 5: Frequency : 5505        Amplitude db : 27.05
Peak 6: Frequency : 6006        Amplitude db : 21.39

For y = 21.25 dB, x is approximately equal to : 119.03 cm
For y = 27.26 dB, x is approximately equal to : 32.59 cm
For y = 20.85 dB, x is approximately equal to : 165.13 cm
Estimated coordinates of the point : (199.48, 28.94)
----------------------------------------------------------------------------
```

Figure 31 : Coordinates XY

Here are the different steps the system follows to give the coordinates of the sensor using visible light:

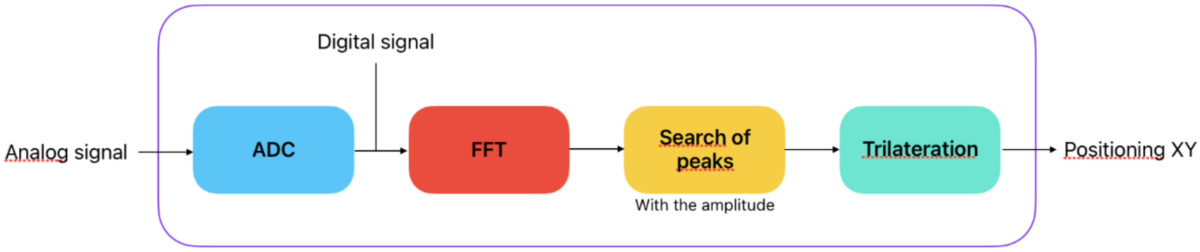


Figure 32 : Complete system diagram

2.8. System errors and limitations

After building the part of the system that allows XY positioning, I carried out a few tests to determine the position error between the actual position and the position measured by the system at different points. For each point, the sensor was placed in a precise position and measured. To obtain data for as much of the test table as possible, which is 280 cm long (X axis) and 80 cm wide (Y axis), points were taken every 28 cm, from 0 cm to 224 cm, on the X axis and every 16 cm, from 16 cm to 80 cm, for the Y axis. This method provides feedback on the operation of the system over almost the entire test table. The graph below shows the distance error in centimetres for each of these measured points:

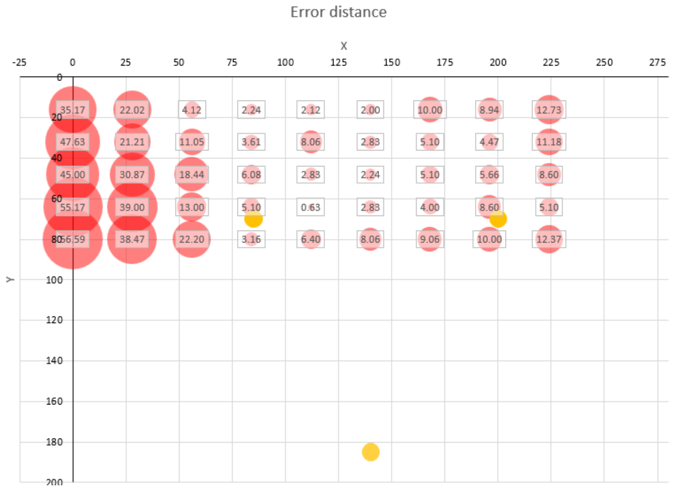


Figure 33 : Graph distance error

On this graph we can see the yellow dots representing the lamps, as well as each red dot with the distance error in centimetres between the actual position and the measured position. To find this error each point has coordinates, for example the first is in (0; 16), and the system gives us the coordinates of the sensor which it will have calculated (x_M; y_M) and with the following formula, $d_{err} = \sqrt{(x_M - x_E)^2 + (y_M - y_E)^2}$ with x_E and y_E the known real coordinates and x_M and y_M the measured coordinates, we know the distance error.

You can also see on the graph that the accuracy is quite good except on the first two rows of points where the error is always greater than 20 cm. This phenomenon is only on one side and I found 3 hypotheses that could explain it:

- **Hypothesis 1:** Due to the distance of the sensor from the lamps which is too large,
- **Hypothesis 2:** Due to the due to the lamp's radiation not being the same all around the lamp, if for the same distance Sensor/Lamp the amplitude is not the same depending on whether you are on the right or the left,
- **Hypothesis 3:** it may be due to the effect of using equations that are too approximate.

To remedy this big difference in error, I first changed the lamp equations from an approximation of the curve using a polynomial trend curve of degree 2, to a polynomial trend curve of degree 6. This makes it possible to stick much more closely to the measured lamp beam curve. These new equations give the following graph:

Figure 34 : graph distance error new equation

On this graph, we can see that some errors are greater than on the previous graph, but the errors are more homogeneous and this is confirmed by the average error, which is 10.35 cm here, compared with 14.2 cm on the previous graph. It has therefore been decided to keep these new equations.

We can see that although the errors are more uniform, they are still greater at the extremities, which is due to the distance of the sensor from the lamp, which is quite large. The system works very well in a precise zone, but the accuracy deteriorates as you move away from this zone. Despite this, an accuracy of 20 cm is still good. A second problem is certainly that the radiation from the lamp is not exactly the same all around the lamp.

To support this hypothesis, I took several points always at the same distance from the lamp to see if the amplitude was always more or less the same, which gave me the following results:

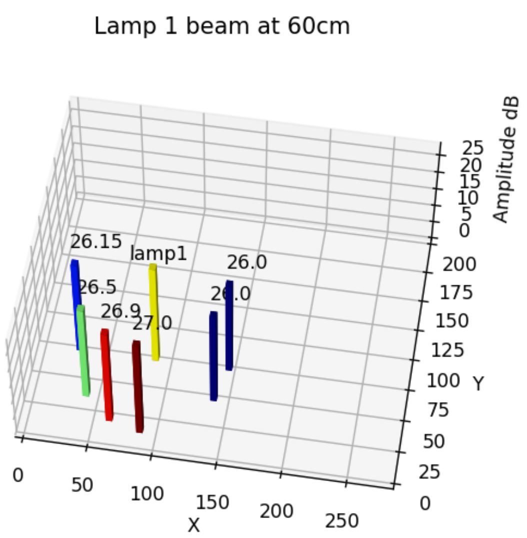


Figure 36 : Lamp 1 beam at 60cm

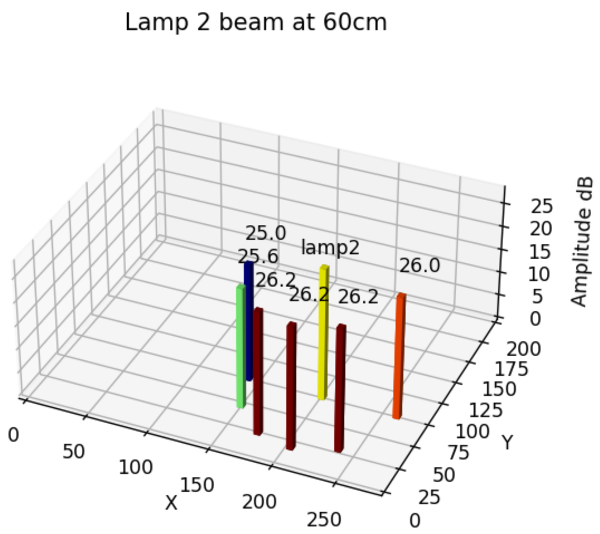


Figure 35 : Lamp 2 beam at 60cm

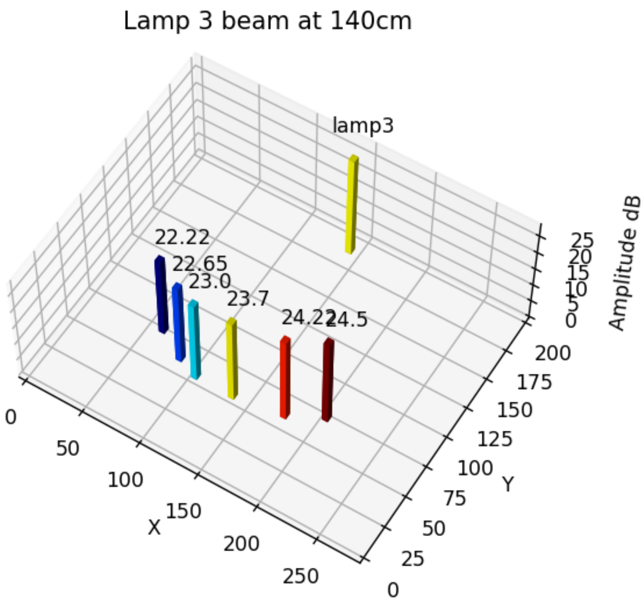


Figure 37 : Lamp 3 beam at 140cm

These graphs show that for lamps 1 and 2 the difference in amplitude depending on where the sensor is in relation to the lamp does not exceed 1 dB, whereas for the 3rd lamp we already have a difference of more than 2 dB. The tests were carried out on the test table, so the distance between the sensor and the lamp is greater for lamp 3, which perhaps explains why the difference in amplitude is greater. However, there is a difference in radiation from the lamps depending on where the sensor is, especially from lamp 3, which creates a distance error when calculating positioning. To reduce this error, the lamp should be changed so that the radiation is uniform all around the lamp.

Cultural differences

When I came to Italy to do my internship, I thought I wouldn't feel much of a change of scenery, given that Italy borders France. In the end I really felt out of place. Right from the start on the road to get to Pisa, I had to take the motorway and I had to pay 2 or 3 euros in tolls every 30 minutes and I really wondered why they didn't just give you a ticket when you get on the motorway and you pay once when you get off. Once I'd arrived in Pisa, I wanted to take a walk in the city parks, and I realised that they weren't mown or well-maintained like in France. I also saw the staff who maintain the verges, roundabouts, etc. mowing the lawn without a mower, using only a brush cutter.

I'd been warned before I went to Italy that Italians were very bad drivers, and well, I wasn't disappointed! They don't pay any attention to what's around them, and every time I try to cross the road there's always a fear of being run over. They're really very dangerous at the wheel and unpredictable. And you can see it on their cars - to find one that's not damaged, you really have to look.

Walking around the city, I was also very surprised by the fact that everything is open on Sundays, from clothes shops to restaurants, small boutiques where you can buy anything and everything, and supermarkets. It makes the city just as lively on Sundays as on other days of the week.

As I had friends in Florence, they were not too far away and we could see each other from time to time. I've found that the bus ticket system is very clever. There's just one bus application for the whole of Tuscany, which is great, but what's more, Tuscany is divided into regions and the bus tickets you buy in one region are valid for the whole region. In my case, Pisa and Florence are in the same region, so the tickets I bought to take the bus in Pisa can also be used to take the bus in Florence. I thought this was great and wondered why we didn't have the same thing in France.

For food shopping, I was very surprised not to find a big pack of “Pom’Potes” in the supermarket, they're sold in packs of four maximum. In Italy, of course, the pasta section is very well stocked and they're much better than what we find in France but there isn't the "coquillette" variety of pasta that so many people in France have grown up with. Frozen pizzas are also much better, which makes sense when you're in the land of pizza.

I will keep a really good memory of this experience abroad. I had the opportunity to see the ways of living in Italy, different than in France, and to meet nice people. I visited too Tuscany and Rome, very beautiful places and with a rich historical past.

Conclusion

This report presents the work carried out during my four-months internship at the TeCIP institute of the Scuola Superiore Sant'Anna in Pisa, on the project "**Detection and processing of visible-light signals for indoor navigation**". The aim of this placement was to set up an indoor positioning system using an Arduino board, LED lamps and a photodiode.

I was able to develop this positioning system by going through several stages, such as choosing the Arduino board, differentiating the lamps, and finally succeeding in having a positioning system that can tell you where the sensor is on the test table by giving its X and Y coordinates. What's more, I was able to carry out a number of tests to find out the system's limitations and errors, so as to make the product as robust as possible and be able to give correct results and use the system in as many cases as possible.

Thanks to this placement, I was able to familiarise myself with optical wireless communication technologies, and more specifically visible light communications. This is a fast-growing field that is still evolving and improving, and I'm glad to have been able to work in this very interesting area.

Personal review

This project was an opportunity for me to apply the methods and knowledge I'd acquired throughout my studies, especially the knowledge I'd acquired at engineering school over the last two years in Electrical Engineering. Working on a concrete subject in the field of research rather than in industry was very rewarding.

I was able to deepen my knowledge of signal processing by observing its concrete application. This greatly helped me to better understand and master this subject. What's more, this brought me to improve my working methods and get to know better my preferences.

All in all, this internship has been a very formative and fulfilling experience, which has enabled me to consolidate my academic knowledge while familiarising myself with the world of research.

Table of annexes

Annex 1 : Check list – Rapports stage GE4

Couverture	✓	Logos, titre, prénoms + noms des étudiants, prénoms+noms des tuteurs, mention « rapport de stage » ou « rapport de projet », année, département ; nom de l'école en entier (Polytech Clermont)
Résumé en français et en anglais, mots clés	✓	Résumé (200 mots maximum) + mots clés Abstact + key words Entreprise, problématique, méthode, résultats, analyse des résultats Les 2 résumés sont sur une seule page
Remerciements	✓	Ordre (en gras ce qui est obligatoire) : **tuteur entreprise**, personnels entreprise, **tuteur école**, enseignants école, autres « Mr. » = Mister ; Monsieur = « M. » Attention à l'accord des participes passés
Sommaire	✓	2 niveaux de titres
Table des figures et des tableaux	✓	Toutes les figures et tous les tableaux sont répertoriés. Classement dans l'ordre d'apparition des figures et tableaux dans le texte « Figure 1 : Titre, numéro de page »
Glossaire	✓	Les mots du domaine présentés en ordre **alphabétique** + définition En fin de lexique : une mention indiquant que « tous les mots suivis d'un astérisque sont définis dans le lexique » Dans le texte : un astérisque à chaque mot défini dans le lexique.
Table des abréviations	✓	Obligatoire s'il y a utilisation d'abréviations Présentées par ordre alphabétique
Introduction	✓	Accroche, sujet, problématique, entreprise, enjeux, méthode de travail, annonce du plan
Contexte	✓	Pour évaluer votre travail, il faut clairement définir d'où vous partez et ce que avez à accomplir (de manière chiffrée, par un cahier des charges par exemple)
Résultats	✓	Il faut également définir quels sont vos résultats (quel pourcentage du cahier des charges a été réalisé?).
Figures	✓	Elles sont TOUTES : numérotées, titrées, référencées dans le texte, sourcées
Différences culturelles	✓	Présence d'une partie sur les différences culturelles (en fin de rapport, avant la conclusion)
Conclusion	✓	Rappel de la problématique, rappel synthétique des résultats, distance critique par rapport à ces résultats, ouverture vers un autre sujet ou une autre problématique
Bilan	✓	Prise de recul, analyse des compétences acquises
Bibliographie Sitographie		Application des consignes de présentation des items ne pas confondre bibliographie et sitographie (sites webs d'entreprises par exemple) ATTENTION à bien citer dans le texte **toutes** les références
Table des matières	✓	Tous les niveaux de titres + tous les « objets » du rapport (les tables, les remerciements etc)
Annexes	✓	Table des annexes (« Annexe 1 : TITRE, numéro de page ») les annexes sont numérotées et classées en ordre d'apparition dans le texte, elles sont TOUTES appelées dans le texte
Fiche du tuteur	✓	Présence de la fiche de validation du stage par le tuteur. (c'est une annexe)

Annex 2 : Form validating the work placement by the tutor

UNIVERSITE CLERMONT-AUVERGNE
POLYTECH CLERMONT

Année universitaire / Academic year: 2022/ 2023

Attestation de lecture de rapport de stage
Internship report reading certificate

Je soussigné(e), M./Mme
I, the undersigned,
Prof. Ernesto Ciaramella

(tuteur d'entreprise), atteste avoir lu et autorise l'envoi au jury le rapport de stage intitulé
(intern's supervisor in the Company) attest to having read and authorizes the sending to the jury of the internship report entitled

Detection and processing of visible-light signals for indoor navigation ………………………………………

de l'élève M./Mme
of the student Mr/ Mrs

Louis Foucher ……………………………………………………………………………………………………

date
27/7/2023

Signature du tuteur
Company supervisor's signature

Annex 3 : Final XY positioning code

```
#include <ADCInput.h>
#include <arduinoFFT.h>
#include <math.h>

#define NUM_POINTS 3
#define EPSILON 1e-9
#define MAX_ITERATIONS 1000

ADCInput adc(A0);
arduinoFFT FFT = arduinoFFT();

// Structure for representing the coordinates of a point
typedef struct {
  double x;
  double y;
} Point;

// Function for solving the system of equations
Point trilateration(Point* reference_points, double* distances) {
  int i;
  double A[NUM_POINTS - 1][2];
  double b[NUM_POINTS - 1];

  // Construction of the coordinate difference matrix
  for (i = 0; i < NUM_POINTS - 1; i++) {
    A[i][0] = 2 * (reference_points[i + 1].x - reference_points[0].x);
    A[i][1] = 2 * (reference_points[i + 1].y - reference_points[0].y);
  }

  // Construction of the distance squares vector
  for (i = 0; i < NUM_POINTS - 1; i++) {
    b[i] = pow(distances[0], 2) - pow(distances[i + 1], 2)
      - pow(reference_points[0].x, 2) + pow(reference_points[i + 1].x, 2)
      - pow(reference_points[0].y, 2) + pow(reference_points[i + 1].y, 2);
  }

  // Solving a system of linear equations
  double det = A[0][0] * A[1][1] - A[1][0] * A[0][1];
  if (det == 0) {
    // The system has no solution
    Point result = { NAN, NAN };
    return result;
  }

  double inv_det = 1 / det;
  double x = (A[1][1] * b[0] - A[0][1] * b[1]) * inv_det;
  double y = (A[0][0] * b[1] - A[1][0] * b[0]) * inv_det;
```

```
    Point result = { x, y };
    return result;
}

double equationL1(double x) {
    return -2 * pow(10, -14) * pow(x, 6) + 5 * pow(10, -21) * pow(x, 5) + 3 * pow(10, -9) * pow(x, 4) + 7 * pow(10, -16) * pow(x, 3) - 0.0005 * pow(x, 2) + 4 * pow(10, -11) * x + 27.791;
    //3 * pow(10, -9) * pow(x, 4) - 5 * pow(10, -18) * pow(x, 3) - 0.0005 * pow(x, 2) - 6 * pow(10, -13) * x + 27.802
    //return - 0.0005 * pow(x, 2) + 9 * pow(10, -16) * x + 27.115; old equation
}

double findXL1(double y) {
    double a = -220.0; // Minimum search value for x
    double b = 220.0; // Maximum search value for xM

    double mid, f_mid;

    for (int i = 0; i < MAX_ITERATIONS; i++) {
        mid = (a + b) / 2.0;
        f_mid = equationL1(mid);
        //Serial.printf("%lf\t%lf\t%lf\n", f_mid, y, mid);

        if (fabs(f_mid - y) < EPSILON) {
            return mid;
        }

        if (f_mid < y) {
            b = mid;
        } else {
            a = mid;
        }
    }

    // If convergence is not reached after the maximum number of iterations
    // or if no solution is found, return an invalid value
    return NAN;
}

double equationL2(double x) {
    return -2 * pow(10, -14) * pow(x, 6) - 1 * pow(10, -20) * pow(x, 5) + 7 * pow(10, -9) * pow(x, 4) + 9 * pow(10, -10) * pow(x, 3) - 0.00065 * pow(x, 2) + 8 * pow(10, -11) * x + 27.611;
    //6 * pow(10, -9) * pow(x, 4) + 15 * pow(10, -8) * pow(x, 3) - 0.00065 * pow(x, 2) - 3 * pow(10, -5) * x + 27.498  old equation
    //return - 0.0005 * pow(x, 2) + 9 * pow(10, -16) * x + 27.115; old equation
}

double findXL2(double y) {
```

```
    double a = -220.0; // Minimum search value for x
    double b = 220.0; // Maximum search value for xM

    double mid, f_mid;

    for (int i = 0; i < MAX_ITERATIONS; i++) {
        mid = (a + b) / 2.0;
        f_mid = equationL2(mid);
        //Serial.printf("%lf\t%lf\t%lf\n", f_mid, y, mid);

        if (fabs(f_mid - y) < EPSILON) {
            return mid;
        }

        if (f_mid < y) {
            b = mid;
        } else {
            a = mid;
        }
    }

    // If convergence is not reached after the maximum number of iterations
    // or if no solution is found, return an invalid value
    return NAN;
}

double equationL3(double x) {
    return -2 * pow(10, -14) * pow(x, 6) - 1 * pow(10, -20) * pow(x, 5) + 5 * pow(10, -9) * pow(x,
4) + 8 * pow(10, -16) * pow(x, 3) - 0.0005 * pow(x, 2) + 8 * pow(10, -11) * x + 31.174;
    //return - 0.0003 * pow(x, 2) + 1 * pow(10, -16) * x + 30.641; old equation
}

double findXL3(double y) {
    double a = -240.0; // Minimum search value for x
    double b = 240.0; // Maximum search value for x

    double mid, f_mid;

    for (int i = 0; i < MAX_ITERATIONS; i++) {
        mid = (a + b) / 2.0;
        f_mid = equationL3(mid);
        //Serial.printf("%lf\t%lf\t%lf\n", f_mid, y, mid);

        if (fabs(f_mid - y) < EPSILON) {
            return mid;
        }

        if (f_mid < y) {
            b = mid;
```

```
        } else {
            a = mid;
        }
    }

    // If convergence is not reached after the maximum number of iterations
    // or if no solution is found, return an invalid value
    return NAN;
}

void setup() {
  Serial.begin(115200);
  while(!Serial);

  const double samplingFrequency = 50000.0;
  const int numberOfSamples = 4096;
  const short nbAcqu = 10;
  const int frequency_1_lampe_1 = 2500, frequency_2_lampe_1 = 6000;
  const int frequency_1_lampe_2 = 4000, frequency_2_lampe_2 = 5500;
  const int frequency_1_lampe_3 = 1500, frequency_2_lampe_3 = 4600;
  int lamp1 = 0, lamp2 = 0;

  adc.begin();
  adc.setFrequency(samplingFrequency);

  // Definition of the variables for the FFT calculation
  double vReal[numberOfSamples];
  double vImg[numberOfSamples];
  float marge_erreur_freq = 50.0;
  float marge_erreur_amp = 2.0;

  float freq_max[10]; // Initialize an array to store the frequencies with max amplitude for each
peak
  float amp_max[10]; // Initialize an array to store the max amplitudes for each peak

  // Initialize a 2D array to store the amplitudes and frequencies for each acquisition
  float amps[numberOfSamples/2];
  float freqs[numberOfSamples/2];

  float marge_erreur = 70.0; // Set the margin of error to 70 Hz
  int index = -1;
  float max_amp = 0.0;
  int peak_count = 0; // Initialize a counter for the number of peaks found
  int t=0, cnt=0, t_mid=0;

  short minimized_surface = 0;
  short pourc_surf = 90;

  // Coordinates of known reference points
```

```
Point reference_points[NUM_POINTS] = {
   { 85, 70 },  //lamp 1
   { 200, 70 }, //lamp 2
   { 140, 185 } //lamp 3
};

delay(1);
int t_start;
//int t_start=millis();

while(1){
  //Reset variables
  memset(vReal, 0, sizeof(vReal)); // Reset
  memset(vImg, 0, sizeof(vImg)); // Reset
  memset(freqs, 0, sizeof(freqs)); // Reset
  memset(amps, 0, sizeof(amps)); // Reset
  memset(freq_max, 0, sizeof(freq_max)); // Reset
  memset(amp_max, 0, sizeof(amp_max)); // Reset
  peak_count = 0;
  index = -1;
  max_amp = 0.0;
  lamp1 = 0;
  lamp2 = 0;

  //Serial.printf("\t\tDébut du calcul\n");
  for (int j=0 ; j < nbAcqu ; j++) {
    //if (j==0)
      //Serial.printf("\tCalcul en cours ...\n", j+1);
    for (int i=0 ; i < numberOfSamples ; i++) {
      //Serial.printf("%d\n", adc.read());
      vReal[i]=(adc.read()*3.3)/4095;
      vImg[i]=0;
      //Serial.printf("%lf\n", vReal[i]);
    }

    //t_start = millis();
    /*The most time-consuming part of the programme*/
    FFT.Windowing(vReal, numberOfSamples, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
    //Compute of the FFT
    FFT.Compute(vReal, vImg, numberOfSamples, FFT_FORWARD);
    //t_mid = millis();
    //Compute of the module of each values
    FFT.ComplexToMagnitude(vReal, vImg, numberOfSamples);
    /*____________________________________________________*/

    //t = t_mid - t_start;
    //Serial.println(t);

    double delta_f = samplingFrequency / numberOfSamples;
```

```
    for (int i = 0; i < numberOfSamples/2; i++) {
      double temp = pow(vReal[i], 2);
      amps[i] = temp+amps[i];
      freqs[i] = i*delta_f;
      //Serial.printf("%.2f\n", amps[i]);
    }
  }

  //Serial.printf("\t\tFin du calcul\n");

  // Compute the average amplitudes for each frequency
  float avg_amps_db[numberOfSamples/2];
  memset(avg_amps_db, 0, sizeof(avg_amps_db));
  for (int i = 0; i < numberOfSamples/2; i++) {
    avg_amps_db[i] = 10*log10(amps[i]/nbAcqu);
    //Serial.printf("%.2f\n", amps[i]);

    //Serial.printf("%d,%.0f,%.2f\n", minimized_surface, freqs[i], avg_amps_db[i]);

    //Search peaks
    if (avg_amps_db[i] > 10) {
      if((freqs[i] >= frequency_1_lampe_1-marge_erreur_freq && freqs[i] <=
frequency_1_lampe_1+marge_erreur_freq) || (freqs[i] >= frequency_2_lampe_1-
marge_erreur_freq && freqs[i] <= frequency_2_lampe_1+marge_erreur_freq)
        || (freqs[i] >= frequency_1_lampe_2-marge_erreur_freq && freqs[i] <=
frequency_1_lampe_2+marge_erreur_freq) || (freqs[i] >= frequency_2_lampe_2-
marge_erreur_freq && freqs[i] <= frequency_2_lampe_2+marge_erreur_freq)
        || (freqs[i] >= frequency_1_lampe_3-marge_erreur_freq && freqs[i] <=
frequency_1_lampe_3+marge_erreur_freq) || (freqs[i] >= frequency_2_lampe_3-
marge_erreur_freq && freqs[i] <= frequency_2_lampe_3+marge_erreur_freq)){
//        if (freqs[i] >= 1000 && freqs[i] <= 8000){
          if (index == -1 || freqs[i] - freqs[index] > marge_erreur) {
            // New peak found
            index = i;
            max_amp = avg_amps_db[i];
            freq_max[peak_count] = freqs[i]; // Store the frequency with max amplitude for this
peak
            amp_max[peak_count] = avg_amps_db[i]; // Store the max amplitude for this peak
            peak_count++; // Increment the peak counter
          } else if (avg_amps_db[i] > max_amp) {
            // New point with higher amplitude found for the same peak
            index = i;
            max_amp = avg_amps_db[i];
            freq_max[peak_count-1] = freqs[i]; // Update the frequency with max amplitude for
this peak
            amp_max[peak_count-1] = avg_amps_db[i]; // Update the max amplitude for this peak
          }
      }
    }
```

```
  }

  //take the no-dB amplitude, to have more fluctuation
  float max_amps[10] = {0};
  for (int i = 0; i < peak_count; i++){
    max_amps[i] = pow(10, amp_max[i]/10);
  }

  // Print the frequencies and amplitudes with max amplitude for each peak
  for (int i = 0; i < peak_count; i++) {
    Serial.printf("Peak %d: Frequency : %0.0lf\tAmplitude db : %.2f\n", i+1, freq_max[i],
amp_max[i]);
  }

  Serial.println();

  //Positionning
  if (peak_count < 2){
    Serial.printf("\n!!!!!!!!!!!!! Something is preventing the sensor from working properly
!!!!!!!!!!!!!!!!!!!!!\n");
  }

  double Measured_Amp1 = (amp_max[1]+amp_max[5])/2;
  double Measured_Amp2 = (amp_max[2]+amp_max[4])/2;
  double Measured_Amp3 = (amp_max[0]+amp_max[3])/2;

  //Serial.printf("%.2lf;%.2lf;%.2lf\n", Measured_Amp1, Measured_Amp2, Measured_Amp3);

  double x1 = findXL1(Measured_Amp1);
  double x2 = findXL1(Measured_Amp2);
  double x3 = findXL3(Measured_Amp3);


  if (!isnan(x1)) {
      Serial.print("For y = ");
      Serial.print(Measured_Amp1);
      Serial.print(" dB, x is approximately equal to : ");
      Serial.print(x1);
      Serial.println(" cm");
  } else {
      Serial.print(Measured_Amp1);
      Serial.println("No solution found for x.");
      x1 = 0;
  }

  if (!isnan(x2)) {
      Serial.print("For y = ");
      Serial.print(Measured_Amp2);
      Serial.print(" dB, x is approximately equal to : ");
```

```
      Serial.print(x2);
      Serial.println(" cm");
    } else {
      Serial.print(Measured_Amp2);
      Serial.println("No solution found for x.");
      x2 = 0;
    }

    if (!isnan(x3)) {
      Serial.print("For y = ");
      Serial.print(Measured_Amp3);
      Serial.print(" dB, x is approximately equal to : ");
      Serial.print(x3);
      Serial.println(" cm");
    } else {
      Serial.println(Measured_Amp3);
      Serial.println("No solution found for x.");
    }

    // Distances measured from reference points
    double distances[NUM_POINTS] = { x1, x2, x3 };

    // Solving the system of equations
    Point coordinates = trilateration(reference_points, distances);

    // Displaying the coordinates of the estimated point
    if (isnan(coordinates.x) || isnan(coordinates.y)) {
      Serial.println("Impossible to determine the coordinates of the point.");
    }
    else {
      Serial.print("Estimated coordinates of the point : (");
      Serial.print(coordinates.x);
      Serial.print(", ");
      Serial.print(coordinates.y);
      //Serial.println();
      Serial.println(")");
    }

    Serial.printf("----------------------------------------------------------------------
---\n");
  }
}

void loop(){

}
```