



2nd Year Internship report

SYSMER

2024-2025



Low-cost Underwater Communication Device for Fish-type Robot

Student :

Matys Adéas BIECHE

School Supervisor :

M. Cedric ANTHIERENS

Internship Supervisor :

Pr. Takuya HASHIMOTO

Tokyo University of Science

Hashimoto Lab

6-chome-3-1 Niijuku, Katsushika-Ku, Tokyo 125-0051

Engagement de non plagiat

Je soussigné, BIECHE A. Matys

N° carte d'étudiant : 22304513

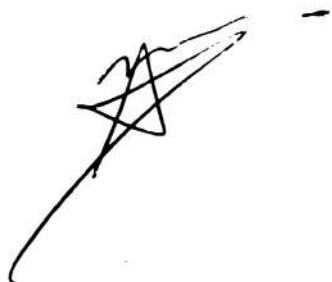
Déclare avoir pris connaissance de la charte des examens et notamment du paragraphe spécifique au plagiat.

Je suis pleinement conscient(e) que la copie intégrale sans citation ni référence de documents ou d'une partie de document publiés sous quelques formes que ce soit (ouvrages, publications, rapports d'étudiant, internet etc...) est un plagiat et constitue une violation des droits d'auteur ainsi qu'une fraude caractérisée.

En conséquence, je m'engage à citer toutes les sources que j'ai utilisées pour produire et écrire ce document.

Fait le 2025/05/06

Signature :

A handwritten signature in black ink, appearing to be 'A. Matys', written over a horizontal line.

Acknowledgment

First and foremost, I would like to express my deepest gratitude to Professor Takuya HASHIMOTO, my supervisor during my internship at Tokyo University of Science (TUS), for allowing me to work in his laboratory. This experience was truly remarkable. Not only did I enjoy working on my internship project, but I also had the chance to meet incredible people and immerse myself in Japanese culture.

I am also sincerely thankful to all the bachelor's and master's students in the lab for their warm welcome and constant willingness to help. I am especially grateful to Shingo NISHIMARU, Gunho LEE, and Taiki MATSUTANI, who were part of my project group, as well as to Taiyo ICHIKAWA, Hayato IMANO, Takuo SHIBAHARA, and Hanako KOGUCHI. I greatly appreciated our discussions and the opportunity to exchange ideas on our various projects.

I would also like to thank Salma LEMBIRIK, a fellow intern from ENSTA Bretagne, with whom I shared many valuable conversations about our experiences in Japan and our respective work. I am also grateful to Antoine MORVAN, who previously worked on this subject and kindly answered my questions. Finally, I am very grateful to Stéphane WEBER, a SeaTech alumnus living and working in Japan, for his generous support and guidance throughout my stay.

This experience has been enriching and educational, and I am grateful to everyone who made it possible.

Résumé

Durant ma deuxième année en école d'ingénieurs à SeaTech, j'ai eu l'opportunité d'effectuer un stage de 18 semaines en tant qu'assistant ingénieur à l'Université des Sciences de Tokyo au campus de Katsushika-ku dans le laboratoire de robotique HashimotoLab

Lors de ce stage j'ai pu concevoir deux systèmes de communication sous-marin à bas-coûts pour robots poissons. En effet, la communication sous-marine est un sujet de plus en plus traité, notamment du au fait de la multiplication des systèmes sous-marins amateurs. Malheureusement, les solutions existantes sont trop coûteuses pour ce type d'application et/ou inexploitable.

Ce rapport sera donc l'occasion de mettre en lumière les compétences acquises, mais aussi les découvertes et recherches réalisées au cours de mon stage, visant à répondre à cette problématique.

Mots clés : Communication sous-marine, Mécatronique, Bas-couts, Robot Poisson

Abstract

During my second year at SeaTech engineering school, I had the opportunity to complete an 18-week internship as an assistant engineer at the Tokyo University of Science, on the Katsushika-ku campus, in the HashimotoLab robotics laboratory. Throughout this internship, I was able to design two low-cost underwater communication systems for robotic fish.

Underwater communication is an increasingly important topic, especially due to the growing number of amateur underwater systems. Unfortunately, existing solutions are often too expensive for these applications and/or are not suitable.

This report will therefore highlight the skills I acquired, as well as the discoveries and research carried out during my internship, all focused on addressing this challenge.

Keywords : Underwater communication, Mechatronics, Low-cost, Fish-type Robot

Sommaire

1	Tokyo University of Science	6
1.1	Hashimoto Laboratory	6
1.2	Work Environment	8
2	Context & Internship Topic	9
2.1	Typical Architecture	9
2.2	Analysis of Previous Work	10
2.2.1	Work by Morvan	10
2.2.2	Insights from the Pilot Study	10
2.2.3	Key Technical Considerations	11
3	Low-cost Underwater Communication Device for Fish-type Robot	12
3.1	SakanaCOM : Intuitive User Interface	12
3.2	Acoustic Device	13
3.2.1	Components and pricing	14
3.2.2	Emitter Hardware	14
3.2.3	Emitter Software	17
3.2.4	Receiver Hardware	18
3.2.5	Receiver Software	21
3.3	AI-Enhanced Visual Device	22
3.3.1	Components and pricing	23
3.3.2	Emitter	24
3.3.3	Receiver	25
3.3.4	AI-Model	26
3.4	Experiments & Results	27
3.4.1	Acoustic Device	27
3.4.2	AI-Enhanced Visual Device	27
4	Further Work	29
4.1	Acoustic Device Improvements	30
4.2	Visual Device Enhancements	30
4.3	Extended Testing	31
5	Conclusion	32

Introduction

Underwater communication devices are usually expensive and complex to develop due to the challenging environment and its limitations. As the number of remotely operated underwater robots increases, there is a growing need for specialized and reliable communication systems that can support both control and positioning functions.

The Fish-type Robot at Hashimoto Lab requires such a system to enable remote control and accurate localization within a pool.

This report details the development of two low-cost underwater communication systems intended to serve as practical solutions for researchers and engineers working with autonomous underwater vehicles (AUVs), remotely operated vehicles (ROVs), and other marine technologies.

By prioritizing affordability and straightforward implementation, the system seeks to facilitate wider adoption of underwater communication technologies across diverse scientific and industrial applications.

1 Tokyo University of Science

Tokyo University of Science (東京理科大学) was founded in 1881 as the first institution in Japan dedicated to teaching the sciences. Along with the advance of science and technology, departments covering a wide range of fields were set up, and today, TUS is the only general university for science and technology in Japan. Recognized as a prestigious university with over 19,000 students, the university is divided across five campuses, offering both undergraduate and graduate programs.

The Katsushika campus houses a research facility with 11 departments, conducting research in various fields, including mechanical engineering, where I completed my internship.



Figure 1.1 *Tokyo University of science*

1.1 Hashimoto Laboratory

I am currently doing my internship at the Hashimoto Laboratory (橋本研究室), a mechanical robotics laboratory founded by Professor Takuya HASHIMOTO, who is also my supervisor during this internship. The laboratory's research mainly focuses on the application of robotics in the fields of medicine, human-robot interaction, social interaction and sports, with a strong emphasis on mechanics. However, the lab also undertakes projects outside these main themes, such as a welding robot and Fish-type robots, which fall into this latter category.



Figure 1.2 *Robots in the Laboratory*

The laboratory brings together about twenty students, mainly undergraduate and master's students, each working on their own projects, which are often complementary. The lab is well equipped, and everyone is free to use the 3D printers, soldering irons, and other equipment as they wish, allowing great flexibility in project development. The lab also has a pool for testing the fish robot.



Figure 1.3 *Laboratory Overview*

1.2 Work Environment

The work organization in the lab gives students considerable freedom and autonomy in how they approach their tasks, select the necessary materials for their projects, and define their working hours. There are no strict rules regarding students' presence in the lab, except that a certain number of hours must be completed by the end of the week. Each student has their own workstation with a desk and computer, which allows students working on complementary projects to be grouped together and encourages collaboration by placing them in clusters.

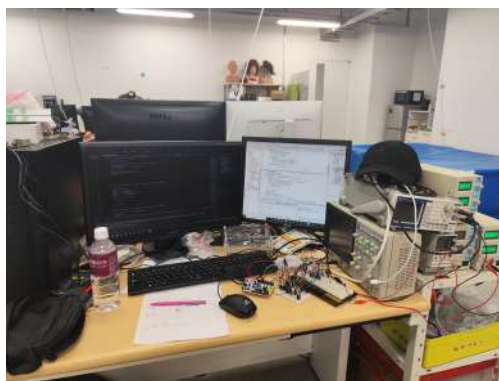


Figure 1.4 *My Desk*

Regarding supervision, students are required to produce a weekly report for the lab director on the work accomplished during the week, during a meeting organized among members of the same working group. These weekly meetings are used to present project progress and discuss encountered problems as well as possible solutions.



Figure 1.5 *Oral Presentation*

Every two weeks, an individual oral presentation must be given in front of all lab members about our project. This presentation also covers our current progress, the results of our tests, and/or our solutions in more detail than in the weekly meetings. After the presentation, there is a Q&A session followed by a discussion with the entire lab to redirect the project if necessary, provide additional feedback, or clarify certain points.

2 Context & Internship Topic

I was asked to develop a low-cost underwater communication device that could be integrated into a small, fish-type robotic platform. This bio-inspired robot is designed for autonomous navigation in aquatic environments, using fin-based propulsion and minimal onboard electronics. To enable coordinated behaviors such as swarm navigation, remote control, or data reporting, it requires a lightweight communication system capable of transmitting position and control information underwater.



Figure 2.1 *The Fish-type Robot*

I was given full freedom in the way I should tackled this problematic, with only two key constraints : the final system should cost less than \$200 and be compatible with the robot's physical and electrical constraints.

Before starting my own design, I reviewed two important documents : the work of a previous intern, Morvan [1], based on a pilot study by Slzachetko et al. [2]. These two references, although different in scope, provided essential technical insights and helped shape the direction of my work.

2.1 Typical Architecture

Before diving into the main topic, it's important to first outline the typical architecture of an underwater communication system. These systems generally consist of two primary nodes : an emitter and a receiver. The transmitter is responsible for encoding and sending data as signals through the water, while the receiver captures these signals and decodes them back into usable information. Due to the challenging underwater environment, such systems often include additional components like digital signal processing units, synchronization mechanisms, and error correction algorithms. This basic two-node setup forms the foundation for point-to-point underwater communication.

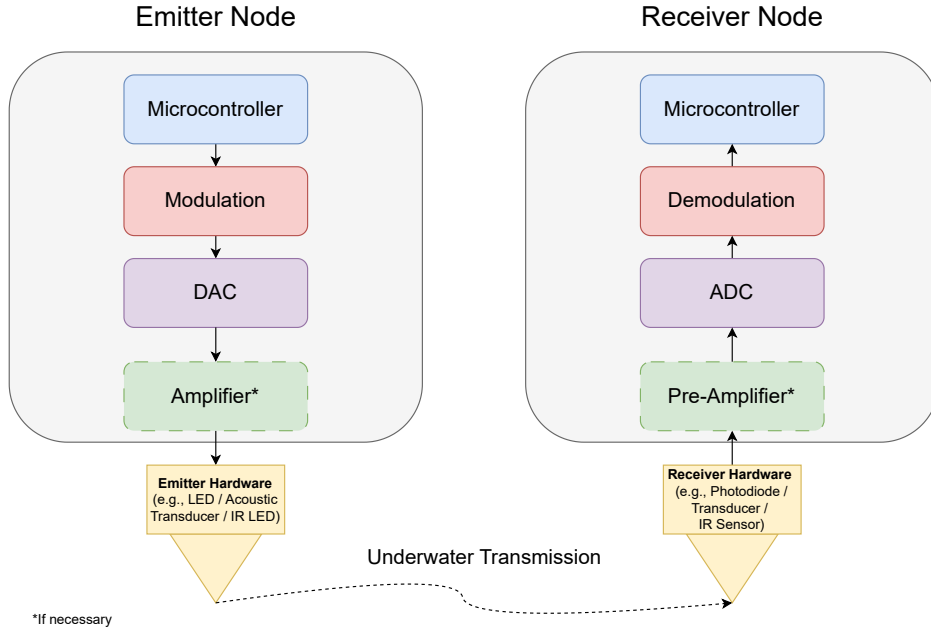


Figure 2.2 *Typical Architecture of underwater communication systems*

2.2 Analysis of Previous Work

2.2.1 Work by Morvan

Morvan’s internship work aimed to design a low-cost acoustic modem based on the findings of an earlier feasibility study. Although the system did not reach full operational reliability, his work laid essential groundwork by identifying key challenges such as impedance mismatches, power limitations, and unpredictable system behavior [1].

Given the inherent complexity of underwater acoustic communication, his well-documented exploration has been extremely valuable in guiding my own development process. By thoroughly exposing many of the practical pitfalls, his work significantly simplified my task and allowed me to focus directly on improving and refining the most critical aspects of the system.

2.2.2 Insights from the Pilot Study

The pilot study by Slzachetko et al. [2] served as an initial proof of concept for low-cost underwater acoustic communication using off-the-shelf components. The system was said to employ QPSK modulation, but most of the presented results appear to correspond instead to BPSK, creating some ambiguity about the actual implementation. Despite these inconsistencies, the study was instrumental in showing that such a system could potentially operate

for under 100\$, at least in controlled conditions.

While the study did not directly shape the early design of my system, it reinforced the feasibility of a low-cost approach. Toward the later stages of my work, I was able to get in touch with the lead researcher, which allowed me to clarify certain technical aspects of the study and validate some of my own choices.

2.2.3 Key Technical Considerations

Both Morvan's work and the pilot study point to several critical factors that must be addressed when designing this type of underwater communication device :

- **Power Supply** : A well-distributed and appropriately managed power supply is crucial to ensure reliable operation and to avoid excessive current draw or potential damage to components.
- **Impedance Matching** : Proper impedance matching between the transducer, amplifier, and driving circuits is crucial to minimize signal loss, maximize efficiency and makes everything work together.
- **Modulation Scheme** : Modulation directly affects the reliability, power efficiency, and software complexity of the communication system, making its selection crucial for overall system performance.
- **Component Selection** : Low-cost components can be effective, but their limitations must be understood and compensated for through careful circuit and algorithmic design.

During this internship, I assumed calm water conditions, as experiments were conducted in a laboratory pool. Therefore, this issue was not addressed.

Throughout the design process, I did my best to take these key aspects into account.

3 Low-cost Underwater Communication Device for Fish-type Robot

In this section, I will describe the two main communication systems I designed and implemented for the fish-type robot, each targeting specific challenges in underwater robotics. The first is an acoustic system, building upon the work presented in the previous sections. It represents a conventional and expected approach to underwater communication. The second is a visual system, which uses light signals detected by a camera and enhanced through artificial intelligence. This approach is more unconventional and was developed to explore alternative communication strategies.

The goal was to investigate low-cost, efficient communication solutions compatible with the constraints of small underwater robots, such as limited power, size, and processing capacity. These two approaches, based on complementary modalities, that are acoustic propagation and visual recognition, were developed to suit different operational contexts.

Rather than detailing every prototype created during the internship, I will focus on the most relevant and representative system versions.

All the source code for these systems is publicly available on GitHub.[3]

3.1 SakanaCOM : Intuitive User Interface

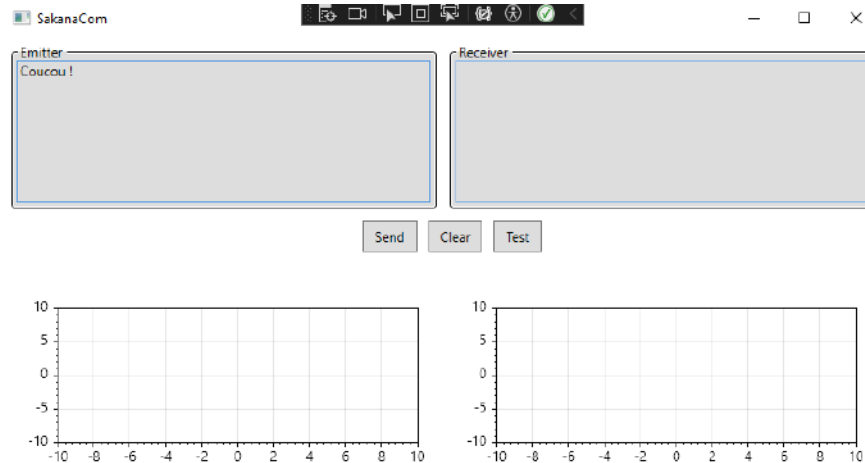


Figure 3.1 *SakanaCOM : User Interface*

Before proceeding with development, I designed a C# (WPF) graphical user interface (GUI) to facilitate debugging and monitoring of my system. This interface enables two-way UART communication between a PC and a microcontroller, using the following communication protocol :

Start Of Frame (SOF)	Command	Payload Length	Payload	Checksum
0xFE	2 octets	2 octets	n octets	1 octet

It includes two real-time graphs capable of displaying various data, such as modulation constellations or buffer values, which greatly aid in debugging. Additionally, the interface features a configurable test button, which I primarily used to send periodic messages for testing purposes.

The GUI is structured to be easily modifiable, allowing for quick adaptation to future development needs, protocol changes or adding new commands.

3.2 Acoustic Device

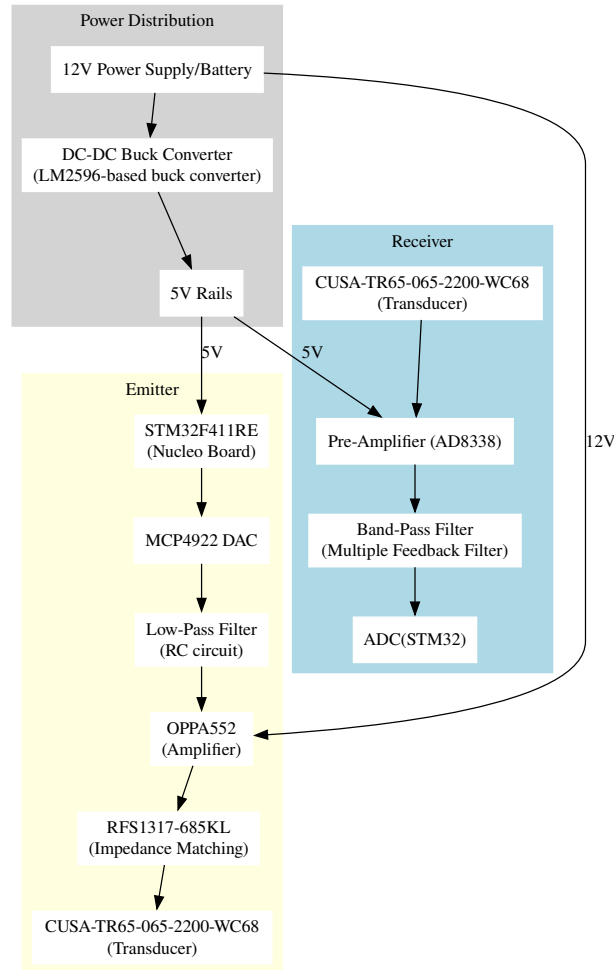


Figure 3.2 *Acoustic Communication Device Architecture*

The first device I designed corresponds to a classic underwater acoustic communication system. While acoustic communication is a well-established method in underwater robotics, the particular design I developed stands out due to its simplicity, affordability, and adaptability for small-scale robots. In contrast to the complex and expensive commercial solutions typically used in the field, this device offers a cost-effective alternative.

Several key decisions shaped the development of this system. First, I chose to use the same transducer as Morvan, which operates at a resonant frequency of 40kHz. This choice had a direct impact on the rest of the design, both in terms of hardware and software. All electronic stages had to be tuned to this frequency, requiring careful impedance matching between components to ensure efficient power transfer and minimal signal loss.

Moreover, components such as amplifiers needed to offer high precision and fast slew rates with minimum power to accurately reproduce the 40kHz signal. Achieving this level of performance on a limited budget proved to be a significant challenge, as higher-specification components typically come at a higher cost.

3.2.1 Components and pricing

The system was designed using off-the-shelf components, carefully selected to balance performance, power efficiency, and cost. The main components are :

Component	Estimated Price
Nucleo STM32F411RE	$\approx 15 \text{ €}$
MCP4922 DAC	$\approx 5 \text{ €}$
RC Low-Pass Filter (passive components)	$< 2 \text{ €}$
OPA552 Amplifier	$\approx 8 \text{ €}$
RFS1317-685KL (Impedance Matching Transformer)	$\approx 5 \text{ €}$
CUSA-TR65-065-2200-WC68 Transducer ($\times 2$)	$\approx 20 \text{ €}$
AD8338 Pre-Amplifier	$\approx 10 \text{ €}$
Band-Pass Filter (passive components)	$< 2 \text{ €}$
LM2596 DC-DC Buck Converter	$\approx 3 \text{ €}$
Misc (cables, PCB, connectors)	$< 10 \text{ €}$
Total	between 75€ and 85€

TABLE 1 – Estimated prices of the acoustic communication components

Thus, fully respecting the low-cost constraint.

3.2.2 Emitter Hardware

The development of the emitter node proved to be particularly challenging. Amplifying a 1.5V signal to at least 10V, while ensuring proper impedance matching with the transducer

and managing current draw, was far from straightforward, especially when considering it was my first time designing a system involving analog electronics.

The node can be described in 4 distinct parts. The DAC conversion, the amplification part and the impedance matching part before the transducer part.

Emitter Circuit

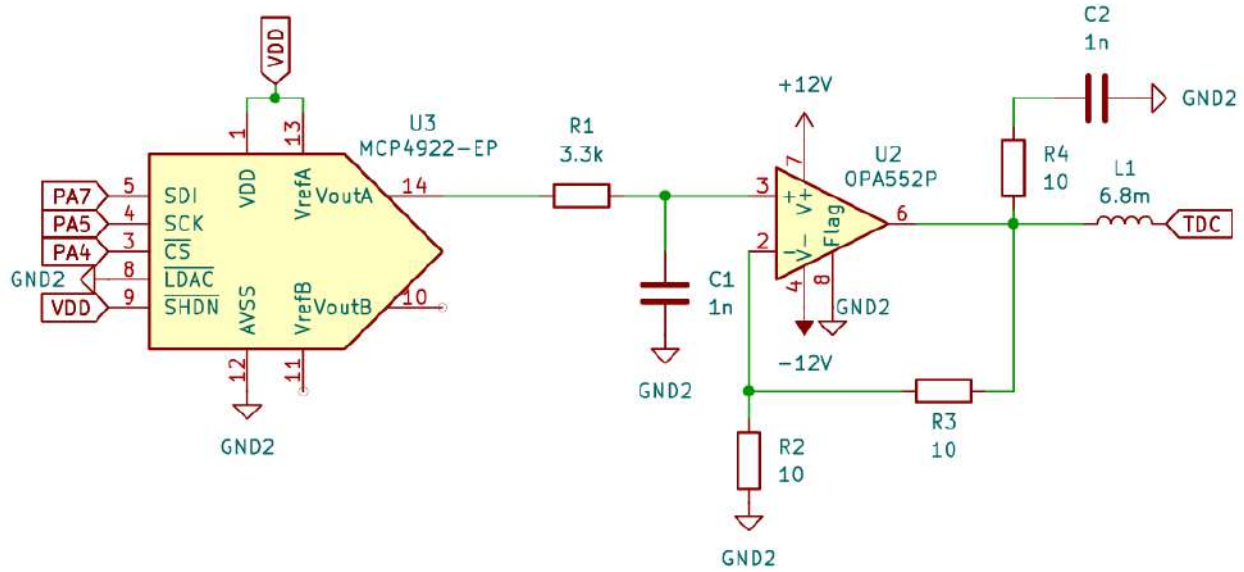


Figure 3.3 *Emitter Circuit*

The DAC is connected to the appropriate STM32 pins, enabling it to output the generated waveform to the amplifier stage. This stage both filters out high-frequency noise and increases the signal amplitude, with the gain determined by resistors R2 and R3. Notably, R3 could be replaced with a variable resistor to allow dynamic gain adjustment depending on the required output.

Following amplification, the signal passes through an impedance matching network before reaching the transducer. This step is essential to maximize power transfer and minimize reflections. In this design, impedance matching was achieved using the RFS1317-685KL transformer, which introduces an inductance of approximately 6.8mH. This value was chosen based on the following equation :

$$L_M = \frac{\sqrt{R_S \times R_L}}{2\pi f}$$

where :

- $R_S = 10 \Omega$ (amplifier output impedance)
- $R_L = 2000 \Omega$ (transducer impedance)
- $f = 40 \text{ kHz}$ (resonant frequency of the transducer)

Plugging in the values :

$$L_M = \frac{\sqrt{10 \times 2000}}{2\pi \times 40000} \approx 6.7 \text{ mH}$$

At this frequency, the inductor exhibits a reactance of :

$$Z_L = 2\pi f L = 2\pi \times 40000 \times 6.8 \times 10^{-3} \approx 1700 \Omega$$

which closely matches the transducer's input impedance, thus improving energy transfer.

To mitigate voltage spikes and oscillations caused by energy stored in the magnetic field due to fast switching conditions such as our sent signal, a snubber network was also integrated. Typical values for this snubber circuit are :

$$R_{\text{snub}} = 10\text{--}100 \Omega \quad \text{and} \quad C_{\text{snub}} = 1\text{--}10 \text{ nF}$$

This protects both the amplifier and the transducer from transient effects and ensures more stable operation.

Thermal Considerations : Assuming a 12 V supply and a 200 mA output current, the power dissipated by the amplifier is calculated as :

$$P = I^2 R_{\text{LOAD}} + V_{\text{DROP}} \times I$$

With $R_{\text{LOAD}} \approx 10 \Omega$ (inductor DCR + snubber), $I = 0.2 \text{ A}$, and $V_{\text{DROP}} \approx 2 \text{ V}$:

$$P = (0.2)^2 \times 10 + (12 - 10) \times 0.2 = 0.4 + 0.4 = 0.8 \text{ W}$$

Assuming the amplifier is in a TO-263 package with thermal resistance $\theta_{JA} = 65^\circ\text{C/W}$ and ambient temperature $T_A = 25^\circ\text{C}$:

$$T_J = T_A + P \cdot \theta_{JA} = 25^\circ\text{C} + 0.8 \times 65 = 77^\circ\text{C} < 160^\circ\text{C}$$

This confirms that the amplifier can operate safely under the given conditions, even without additional heatsinking.

This short overview illustrates the level of detail required in the design of each individual component to ensure proper integration and functionality. For the sake of conciseness, I will not describe every remaining component or circuit decision in the same depth.

As for the software, although behind easier with the help of the GUI I developed, the main challenges here were to activate the internal clocks and peripheral gpios accordingly to what was required to make the DAC working.

3.2.3 Emitter Software

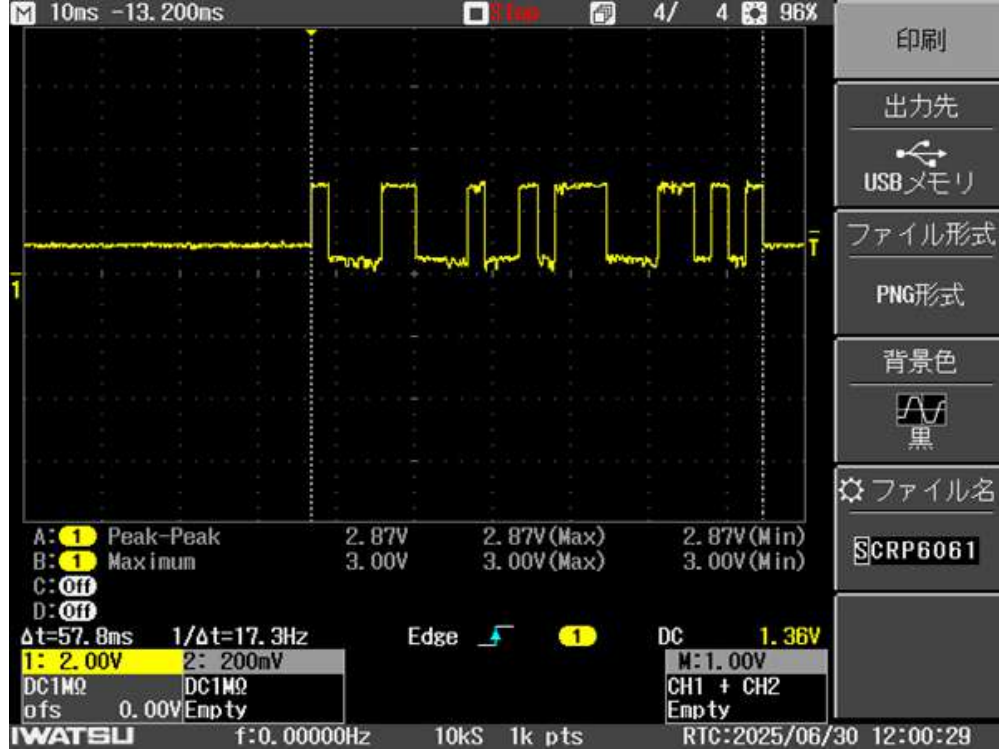


Figure 3.4 *Sent Signal*

The emitter's software architecture was implemented on an STM32F411 microcontroller (Nucleo board).

Generating a continuous modulated signal required precise configuration of the DAC, SPI, DMA, UART, and timers was a quite challenging task considering the system's memory, power, and speed limitations. The PLL was set to run at 84 MHz, and the GPIO pins were initialized with the appropriate alternate functions for SPI (MOSI/SCK) and DAC control lines.

A dedicated transmitter module handles the streaming of samples to the MCP4922 DAC via SPI. A timer triggers interrupts at the desired sampling rate, and during each interrupt, a callback retrieves the next symbol sample from a buffer and transmits it to the DAC.

The modulation block precomputes a step signal representing the bitstream received from the PC. This signal, including a preamble of two bits (1,0) to facilitate synchronization with

the receiver node, is stored in a ring buffer and then fed to the transmitter module.

The SPI peripheral is configured with DMA support, enabling efficient and continuous data transfer to the DAC while minimizing CPU load.

However, the STM32F411's limited memory posed significant challenges when generating high-quality signals. As a result, I opted for a simplified OOK (On-Off Keying) modulation scheme, although the original plan was to implement BPSK or QPSK. For future improvements, I would recommend using a more powerful microcontroller, either from the same manufacturer or another widely available option.

3.2.4 Receiver Hardware

Receiver Circuit

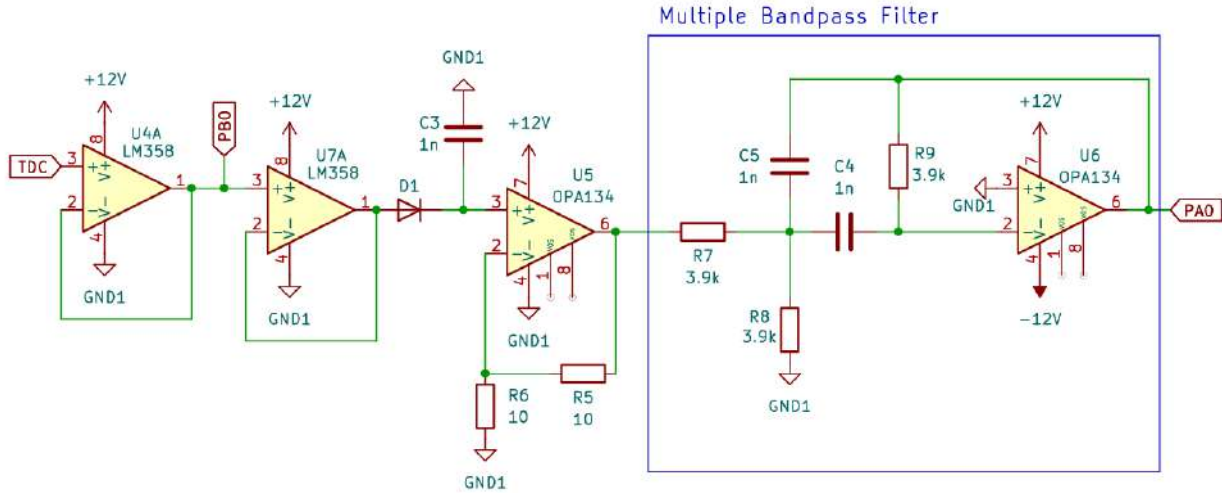


Figure 3.5 *Receivers Circuit*

The receiver circuit is responsible for capturing weak acoustic signals underwater and converting them into clean digital data for processing by the microcontroller. Each stage of the hardware pipeline is designed to ensure signal integrity, robustness to noise, and compatibility with the STM32F4 microcontroller. Below is a detailed breakdown of the components and their roles.

The system uses the same 40 kHz piezoelectric transducer used on the transmitter side to detect incoming acoustic signals. These transducers generate small analog voltage fluctuations in response to sound pressure variations in water. However, these signals are extremely weak (often in the millivolt range), and thus require high-gain, low-noise amplification to be

usable.

The analog signal from the transducer was first designed to be fed into an AD8338 auto-gain control amplifier. This component was specifically chosen for its high sensitivity and very low noise characteristics. It allows a digitally controlled gain (up to 50dB), making it ideal for underwater environments where signal strength may vary significantly with distance or angle. The gain control can be tuned externally via a voltage input, enabling runtime adaptability. I was not able to make it work on my system and did not find the reason. In order to still work, I decided to amplify it by hand, with no auto-gain.

So, instead, the analog signal first goes into two consecutive LM358 op-amp stages (U4A and U7A). Each stages helps with impedance matching with the ADC and EXTI Pins that drain current while activated, but also filters out some of the noises.

After the LM358 stages, the signal passes through a diode (D1) and capacitor (C3), which form a crude envelope follower. This stage helps reshape the modulated signal by clipping and filtering sharp transitions, essentially "widening" peaks that are too narrow for the ADC to detect properly. This design decision addresses the issue of spike length being too short in early tests. (We will get back to that in the next section).

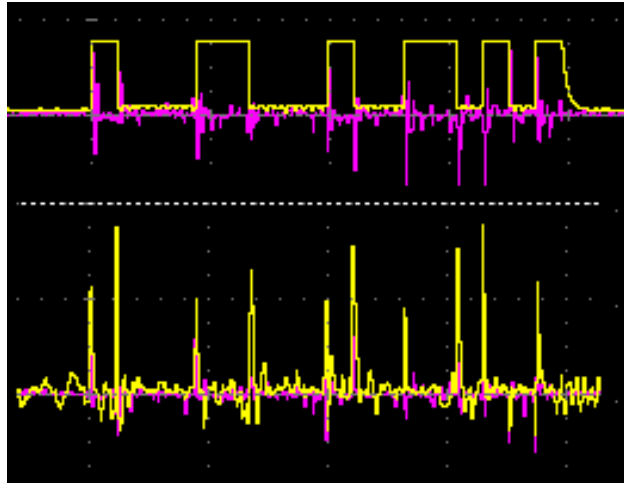


Figure 3.6 *Up : (Purple = Received transducer signal; Yellow = Transmitted signal)*
Down : (Purple = Received transducer signal; Yellow = Received signal amplified)

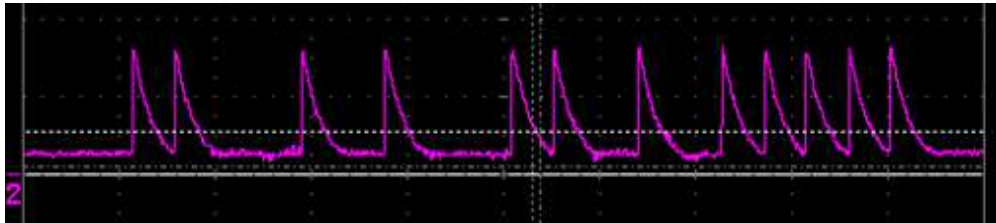


Figure 3.7 *Received signal after filtering and peaks widening*

The reshaped signal is then amplified by a an op-amp (U5 : OPA134), which provides clean gain and isolates the filtering stages from the prior stages. OPA134 was selected for its low-noise and performance, making it ideal for sensitive analog circuits.

Following amplification, the signal passes through a multiple-feedback band-pass filter centered around 40 kHz using 1 nF capacitors (C4, C5) and 3.9 k Ω resistors (R7-R9). The goal here is to isolate the useful signal and attenuate both low-frequency environmental noise (e.g., vibrations, water flow) and high-frequency electrical interference. This filter can be removed for tests in controlled conditions.

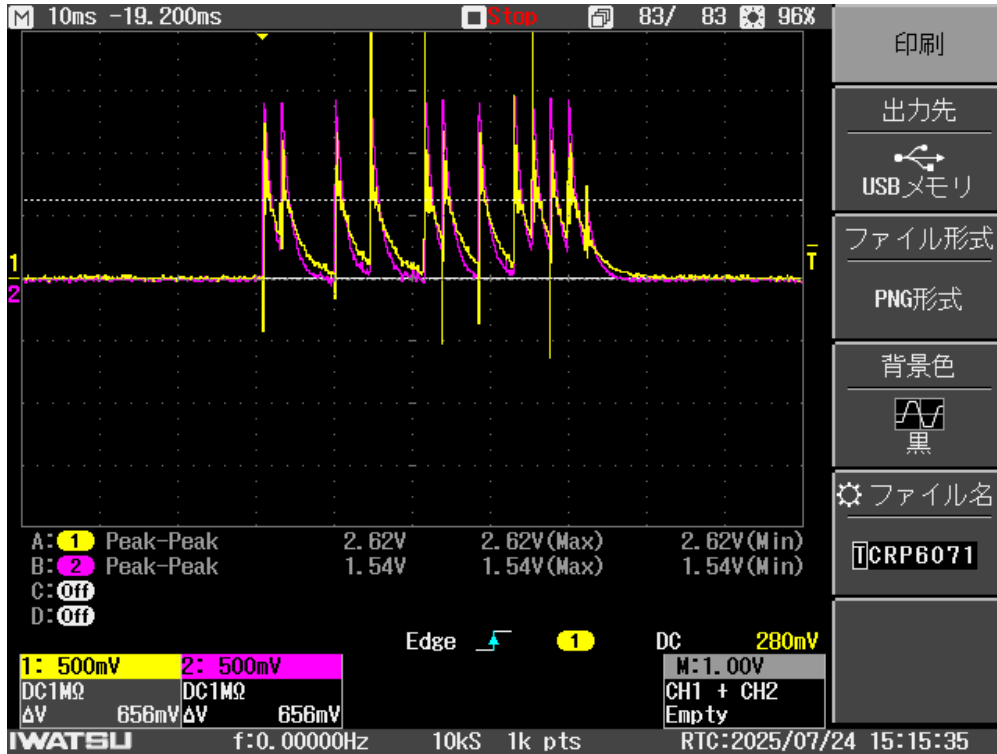


Figure 3.8 *Received signal in real conditions (Yellow : After the transducer; Purple : after filtering)*

The filtered analog signal is directly digitized using the STM32F411RE’s 12-bit ADC.

The entire receiver system is powered by a stepped-down voltage supply from a 12 V source. Careful attention is given to power integrity, including the use of decoupling capacitors near all sensitive analog circuitry to minimize voltage ripple and electrical noise that could otherwise corrupt weak signal measurements.

This hardware pipeline enables reliable underwater reception and demodulation of acoustic data using low-cost and compact components.

3.2.5 Receiver Software

The receiver software plays a pivotal role in reconstructing the transmitted data from the analog signals captured through the ADC. It is implemented on the STM32F411RE microcontroller and handles signal acquisition, demodulation, and protocol-level reconstruction. The design is tailored to support On-Off Keying (OOK) modulation and features robust synchronization and error mitigation mechanisms.

Different way of synchronization, demodulation, adc conversion where tested, but for the speed and precision needed, I was able to achieve the receiving part only with one method that I will describe here.

The signal reception is initiated by an external trigger connected to pin PB0, using an EXTI (external interrupt). To avoid multiple detections from bouncing triggers, the software temporarily disables EXTI after each trigger. It uses a simple timestamp-based reactivation mechanism which ensures the EXTI line is re-enabled only after a cooldown period (100 ms), and only if the pin has returned to a low state. The rising edge triggers and interrupt then starts different actions.

First, it starts the ADC which begins to acquire samples through DMA from analog pin PA0. A timer (TIM3) is simultaneously activated to pace the ADC conversions at a precise rate (480 kHz). At first, the peaks were too small so some of the peaks were not acquired/missed. This was the reason why I added the diode and capacitor in the hardware design to widen the peaks. The received analog data is then stored in a circular buffer. Once the buffer is full, a callback is executed, which halts the ADC and stops the timer, signaling that data processing can begin.

To decode the received analog signal, a custom OOK (On-Off Keying) demodulation method is implemented. The goal is to transform the raw ADC samples into a clean bitstream representing the transmitted data. The steps are as follows :

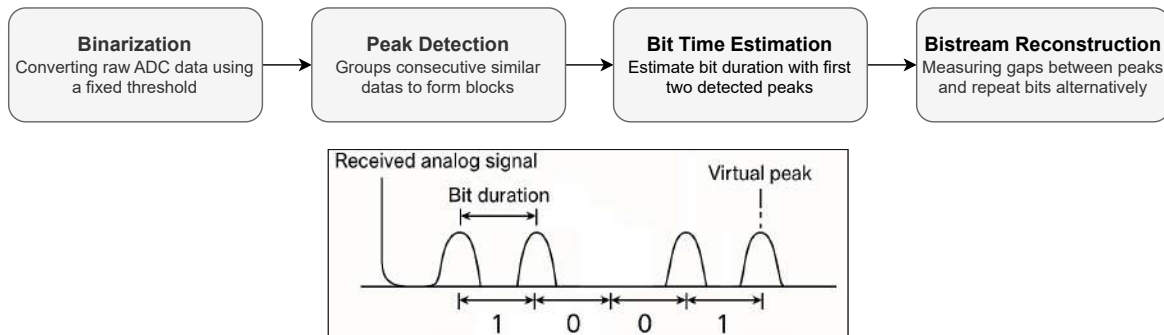


Figure 3.9 *Demodulation process schematic*

- **Binarization** : The raw ADC data is first binarized using a fixed threshold. Values above this threshold are considered as high (signal present) and marked as 1, while values below are considered as low as 0. This step converts the continuous signal into a sequence of digital high/low states, where each peak corresponds to a potential data transition.
- **Peak Detection** : Consecutive "1"s in the binarized signal are grouped to form a single "high block." For each block, the temporal center position (mean index) is recorded. These centers form a list of peak positions, representing the timing of rising and falling edges.
- **Bit Time Estimation** : The time difference between the first two detected peaks gives an estimate of the duration of a single bit (the symbol period). This works because the preamble is always known and consists of a "1" followed by a "0". This reference bit time is then used to decode subsequent symbols.
- **Bitstream Reconstruction** : We then walk throughout the list of peaks, measuring the gap between each pair of consecutive peaks. For each interval, the number of bit durations that fit into this gap is calculated, and the current bit value is repeated accordingly.
 - The first detected interval is always interpreted as a "1".
 - The bits then alternate (1, 0, 1, 0,...) according to the spacing between peaks.
 - If the number of detected peaks is odd, a virtual peak is added at the expected position to maintain a consistent alternation pattern.

Finally, the reconstructed bitstream (excluding the preamble) is grouped into bytes (8 bits each), which are then passed to the higher-level UART protocol handler for further processing.

This algorithm is designed to be robust against noise, small timing variations, and signal degradation. By relying on relative peak spacing rather than absolute sample values, it can handle imperfect conditions such as small amplitude fluctuations or slight sampling jitter.

3.3 AI-Enhanced Visual Device

During my internship, I came up with the idea of developing a novel device inspired by visual communication systems and the perception techniques we use in the RoboCup MSL League.

Recognizing the robustness of perception methods enhanced by AI, such as object detection with YOLO, I initially designed a system that would transmit digital data by blinking

LEDs in a controlled pattern. A camera would capture these blinking signals, and an AI model would interpret the ON/OFF states to recover the transmitted data.

While this concept worked well in theory, I encountered practical limitations related to camera frame rates, which made reliable detection of fast LED blinking challenging. As a result, I revised the approach. I retained the use of AI to enhance the system, but redesigned the architecture to separate the LED state detection (using a traditional algorithm) from the AI-based data demodulation, creating a more robust and reliable solution.

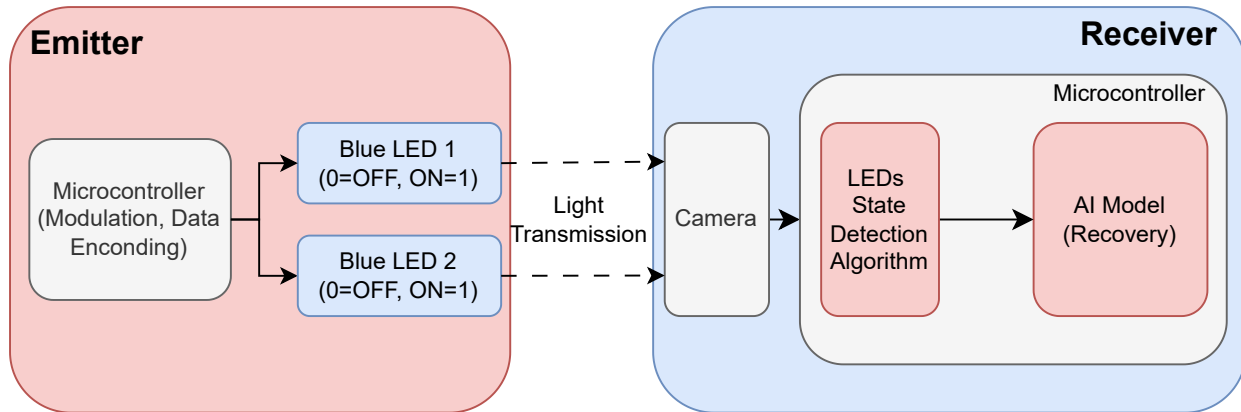


Figure 3.10 *AI-Enhanced Visual Communication Device overview*

This device can be easily integrated into existing projects that already use a camera and a microcontroller or PC.

Note that, to be technically rigorous, the device should more accurately be referred to as an **Optical Communication System with AI-Based Symbol Recovery**. However, for the sake of brevity and readability, we adopt the shorter term AI-Enhanced Visual Communication Device throughout this work.

This system is also the subject of a research article, authored during my internship, which is currently under preparation and will provide a more detailed description of its design, implementation, and performance.

3.3.1 Components and pricing

The components used in testing were chosen solely for demonstration purposes. In real-world applications, such components (microcontrollers, PCs, and cameras) are typically already integrated into the USV. As a result, they usually do not add any significant cost to the overall project, since part of the processing can be offloaded to existing onboard systems.

If your project already includes a microcontroller or PC, adding LEDs and a simple blinking control system would cost less than 20€. The low-cost constraint is fully respected either way.

Component	Estimated Price
Decorative Blue LEDs Tape	$\approx 3\text{€}$
Raspberry Pi 4 Model B	$\approx 60\text{ €}$
Raspberry Pi Cam V3	$\approx 30\text{ €}$
Nucleo STM32F411RE	$\approx 15\text{ €}$
Others (Cable, Case..)	$< 10\text{ €}$
Total	between 90€ and 100€

TABLE 2 – Estimated prices of the components

3.3.2 Emitter

Since blinking two LEDs via a microcontroller is a trivial and widely understood operation, this report omits its implementation details in favor of focusing on more substantial technical contributions.

The modulation technique employed involves blinking the LEDs according to a predefined pattern that encodes a binary message. Each pair of bits is represented by a unique LED state, as follows :

- (1,0) = LEFT LED ON, RIGHT LED OFF
- (0,1) = LEFT LED OFF, RIGHT LED ON
- (1,1) = BOTH ON
- (0,0) = NONE ON

To facilitate synchronization and detection by the receiver, a preamble is included at the beginning of each message. This preamble consists of alternating blinks of the two LEDs, creating a known, repetitive pattern that enables the system to identify the start of valid transmissions.

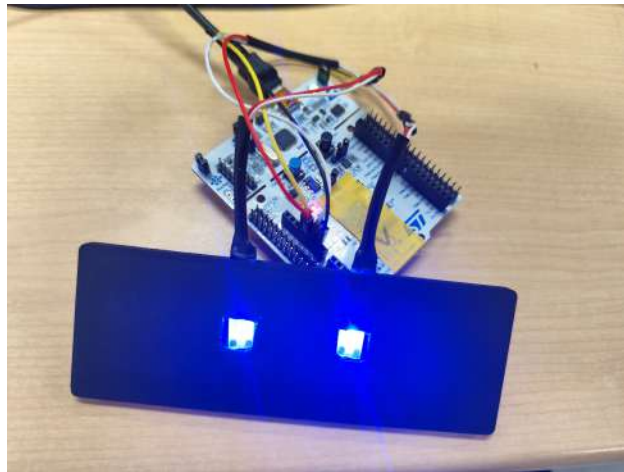


Figure 3.11 *Emitter Node*

The LEDs are housed within a 3D-printed black enclosure, a decision whose purpose is discussed in detail in the receiver section.

3.3.3 Receiver

The receiver node is responsible for detecting and decoding optical signals transmitted via blinking LEDs. The implementation, developed in Python, relies on the OpenCV library for real-time computer vision processing. Video frames are continuously acquired using the Raspberry Pi Camera Module v3. To constrain processing to a relevant area and improve robustness against background noise, the system first isolates a predefined black rectangular region within each frame. This region of interest (ROI) is detected by thresholding the grayscale image and applying morphological filtering to extract the largest contiguous dark contour, containing the LED emitters.

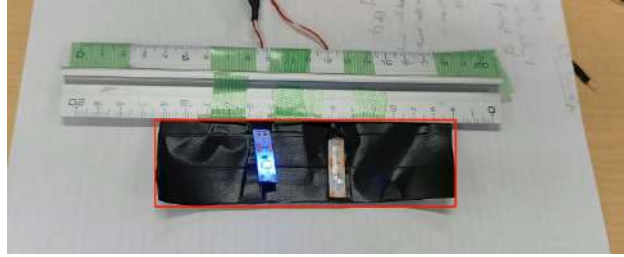


Figure 3.12 *Black Area detection on the first prototype*

Once the ROI is established, the system applies color-based detection in the HSV color space. I added a calibration mechanism to enable dynamic adaptation by allowing the user to manually select a pixel corresponding to an active LED, from which HSV bounds are inferred using a predefined tolerance. These bounds are then used to segment LED candidates within the ROI via binary thresholding and morphological operations.





Figure 3.13 *Led Recognition*

The detected blobs are subsequently filtered by size and spatial consistency, and then tracked across frames to identify the left and right LEDs. A lightweight tracking algorithm compares the centroid of each blob with previously known LED positions, using Euclidean distance as a matching criterion. This labeling persists across frames, even when only one LED is temporarily visible, allowing robust binary state extraction over time.

Each frame is thus reduced to a binary state tuple representing the on/off status of both LEDs. These tuples are appended to a rolling buffer from which the system attempts to decode messages by sending it to the AI-Model.

3.3.4 AI-Model

To improve the robustness of the receiver and to accurately decode transmitted messages even in the presence of noise, a Gated Recurrent Unit (GRU)-based neural network was employed. GRUs are a type of recurrent neural network (RNN) architecture particularly well-suited for sequence modeling tasks, thanks to their ability to retain and forget information across time steps through gated mechanisms. This makes them effective for processing temporally structured data such as blinking LED patterns. The network was implemented using PyTorch and trained on a synthetic dataset of 40,000 message sequences, each consisting of 16 bits encoded as 2-bit LED states over time.

The dataset was artificially generated to simulate realistic communication noise : 15% of the frames in each sequence were either randomly flipped (bit inversion) or dropped entirely, mimicking errors such as LED detection noise, occlusion, or frame loss. These corrupted sequences were then padded or truncated to a fixed input length compatible with the GRU model.

The GRU model itself consists of two recurrent layers with 256 hidden units, followed by a fully connected layer that outputs a 16-dimensional vector. Binary cross-entropy with logits was used as the loss function, and the model was trained for 400 epochs using the

Adam optimizer. The final model achieved a bit-level accuracy of approximately 95.61%, demonstrating high reliability even under noisy input conditions.

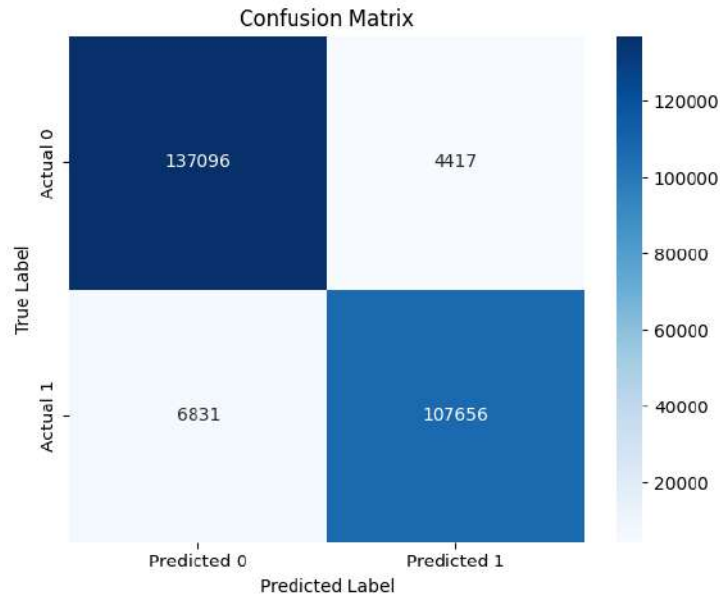


Figure 3.14 *Confusion Matrix*

These results highlight the effectiveness of the GRU.

3.4 Experiments & Results

Some experiments have been done in order to see the limitations of the two systems. Here will be noticed the conditions of the experiments and notable results.

3.4.1 Acoustic Device

Successful underwater communication at a speed of 1 kbps was achieved. However, the range remains limited due to insufficient pre-amplification, as discussed in the corresponding section. Implementing an automatic gain control (AGC) pre-amplifier would significantly improve performance. In the current setup, the maximum reliable reception distance is around 10 cm. With improved pre-amplification methods, a range of 1–2 m should be attainable given the current emitting power.

3.4.2 AI-Enhanced Visual Device

The performance of the AI-Enhanced Visual Device depends on geometric and optical factors such as camera resolution, inter-LED spacing, LED size, and luminance. To predict

the operational range, a simple geometric model was developed.

The camera–LED geometry was modeled to relate physical parameters to image-space size. For camera-to-object distance z , horizontal FOV θ_{FOV} , image width N_{px} , LED diameter d_{LED} , and spacing s_{LED} , the scene width is

$$W(z) = 2z \cdot \tan\left(\frac{\theta_{\text{FOV}}}{2}\right), \quad \text{scale}(z) = \frac{W(z)}{N_{\text{px}}}.$$

Apparent LED diameter and separation (in pixels) are

$$D_{\text{px}}(z) = \frac{d_{\text{LED}}}{\text{scale}(z)}, \quad S_{\text{px}}(z) = \frac{s_{\text{LED}}}{\text{scale}(z)}.$$

Detection requires $D_{\text{px}} \geq 6$ and $S_{\text{px}} \geq 5$.

Using the parameters :

- Camera resolution : 160×120 pixels
- LED diameter : 1 cm
- Horizontal field of view : 66°
- Inter-LED spacing : 10 cm

With $s_{\text{LED}} = 10$ cm, the model predicts reliable detection up to $z \approx 240$ cm ; within this range, both constraints are satisfied.

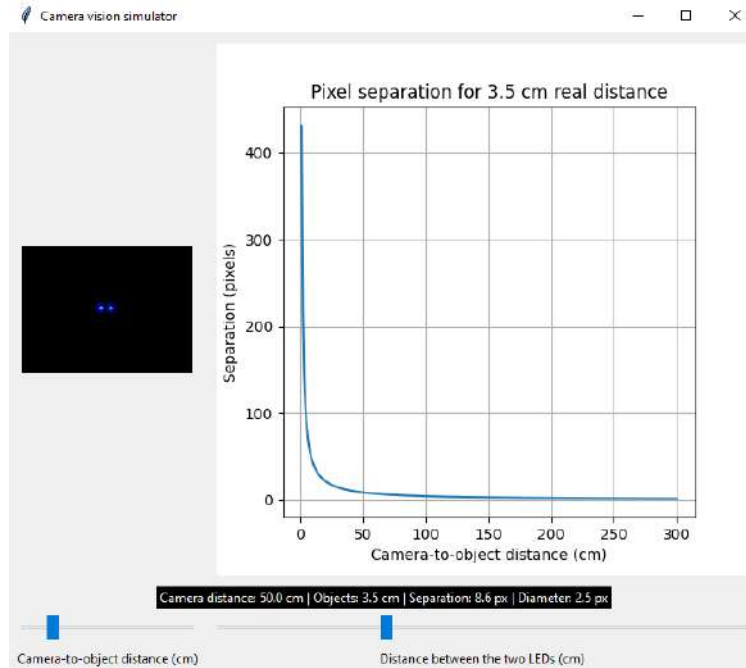


Figure 3.15 *Simulation tool for predicting LED visibility as a function of distance and spacing.*

Pool experiments were then carried out by varying the emitter’s position. Results showed a strong agreement with the model : detection was stable and decoding accuracy matched simulation predictions within the estimated range, while performance degraded rapidly beyond it.

Combined with these physical measurements, GRU-based decoding demonstrated clear improvements over a direct rule-based mapping. The numeric GRU, optimized for fixed-length numeric codes, achieved up to 95.76% bit accuracy under clean conditions and maintained over 80% accuracy even with high simulated noise. This confirms that structured message formats (numeric IDs, short codes) are more resilient for underwater optical communication, while the GRU’s ability to partially recover corrupted sequences provides useful outputs even when exact recovery fails.

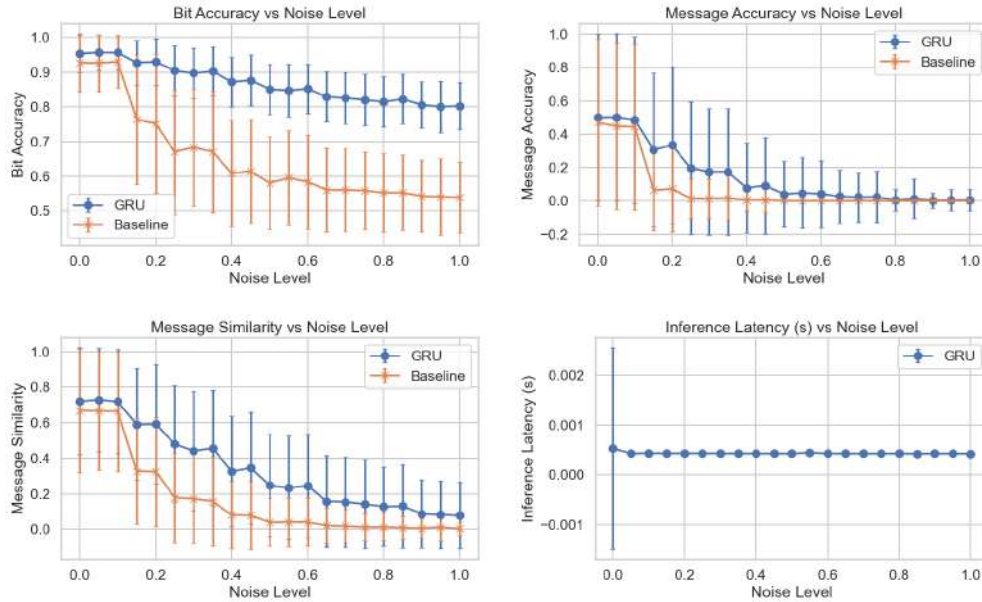


Figure 3.16 *GRU performance compared to a the baseline. (The baseline represents the system performances without AI-recovery)*

At distances beyond the limit, detection degraded as predicted. This validates that in practical conditions, once LED states are extracted, decoding operates deterministically and performance is governed by the known robustness of the GRU to noise and symbol corruption.

4 Further Work

While both the acoustic and AI-enhanced visual communication systems reached a functional prototype stage, several areas could be explored to further develop their capabilities and robustness.

4.1 Acoustic Device Improvements

A primary limitation of the current acoustic system lies in the reception range, which is currently limited by the absence of proper pre-amplification. One essential improvement would be the integration of an automatic gain control (AGC) stage. For instance, successfully implementing the AD8338 or a more integration-friendly alternative would allow dynamic gain adjustment, improving performance across varying distances and orientations.

A second enhancement would involve upgrading the modulation scheme. Although On-Off Keying (OOK) has proven functional, implementing more advanced techniques such as BPSK would significantly increase the data rate and improve resilience to noise. However, these schemes require more precise signal generation and processing, implying the need for more powerful hardware.

The current microcontroller, STM32F411RE, while cost-effective and compact, limits signal quality due to its processing and memory constraints. Future designs could benefit from migrating to a more powerful microcontroller such as the STM32H7 series or even a small FPGA to manage higher-order modulations and real-time signal processing.

Finally, the current setup is not yet optimized for full underwater integration. Designing a compact, waterproof enclosure with embedded transducer connectors would be necessary for real-world deployment outside of controlled pool conditions.

4.2 Visual Device Enhancements

The AI-enhanced visual device shows promising results. Nonetheless, its performance remains dependent on camera characteristics, ambient lighting, and LED placement.

Improvements could begin with the optical system. The current prototype uses basic decorative LEDs; replacing them with high-power, narrow-angle LEDs and integrating them into a custom optical housing would enhance visibility and increase range. Moreover, using cameras with higher frame rates would allow faster blinking patterns and therefore increase the data transmission rate.

The current GRU-based neural network already shows strong resilience to noise and partial data loss. However, exploring more modern deep learning architectures, such as Transformers or CNN-RNN hybrids, could further increase decoding performance under real-world disturbances.

In addition, the integration of a Kalman filter-based decision layer could be considered to determine whether or not the GRU output should be used to correct the received signal. This approach would mitigate cases in which the AI inadvertently alters an already accurate

message, thereby introducing errors instead of improving the result.

Another aspect concerns adaptation to realistic underwater conditions. The current system performs well in laboratory pools but would need to be tested and calibrated for light attenuation, reflection, and color shift present in open water. Techniques such as dynamic thresholding, color correction, or temporal smoothing could be integrated to address these challenges.

The system currently supports only one-way communication. Adding a visual feedback channel would make bi-directional exchanges possible and enable mechanisms such as message acknowledgment, error correction, or synchronization.

4.3 Extended Testing

Although initial results in laboratory conditions are encouraging, further validation is required. Future experiments should include trials in open water, such as lakes or coastal areas, to assess real-world viability. In particular, testing under different lighting conditions, distances, angles, and movement scenarios would help evaluate the limits and reliability of each system.

Long-term deployment tests should also be conducted to assess waterproofing durability, thermal stability, power consumption, and overall endurance of both systems.

In summary, while the foundation has been successfully laid, these proposed extensions would substantially increase the applicability and robustness of the developed systems for future underwater robotic missions.

5 Conclusion

This internship at the Hashimoto Laboratory far exceeded my expectations. Over the course of four months, I was able to tackle challenging technical problems, develop innovative solutions, and deepen my expertise in robotics, electronics, and artificial intelligence. In particular, designing and validating two functional low-cost underwater communication prototypes allowed me to apply and expand my engineering skills while contributing to a field where accessibility and efficiency are increasingly important.

Beyond the technical dimension, this experience allowed me to immerse myself in a different academic and cultural environment, while working at the intersection of robotics, electronics, and artificial intelligence. It was particularly inspiring to contribute to innovative, low-cost solutions in underwater communication, a field where accessibility and efficiency are becoming increasingly important.

Ultimately, this internship has reinforced my interest in robotics and embedded systems. It has not only validated my academic direction but also broadened my vision of the career paths I can pursue. I leave with new technical skills, valuable insights, and a deep appreciation for interdisciplinary collaboration in research and development.

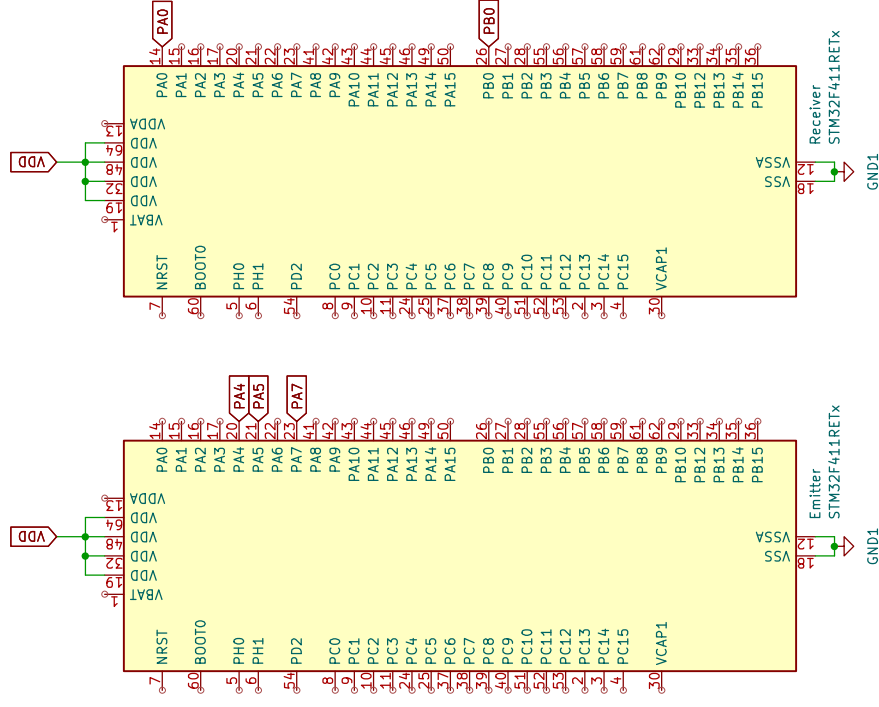
In short, this has been a defining chapter in my academic journey, that I will carry forward into my future studies and professional endeavors.

Bibliography

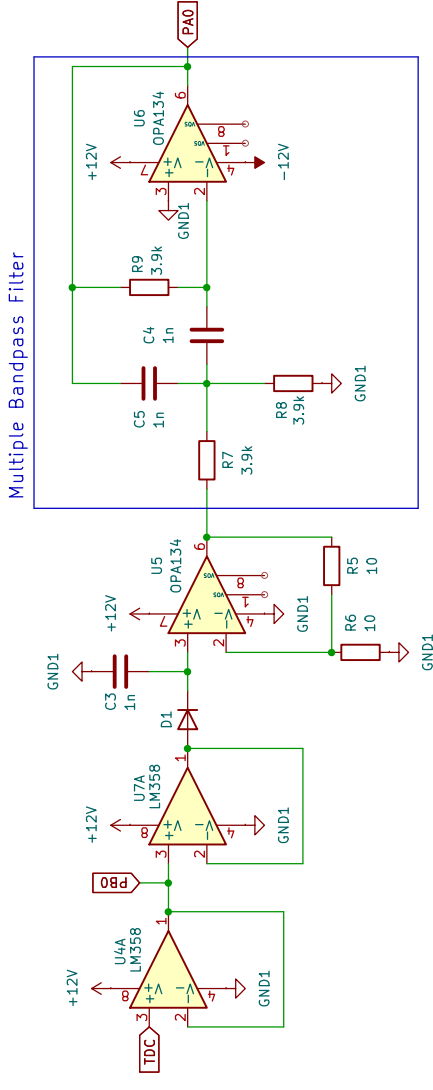
- [1] Morvan Antoine. Rapport de stage 2024. https://www.ensta-bretagne.fr/jaulin/rapport2024_morvan.pdf, 2024.
- [2] Boguslaw Slzachetko. Low-cost underwater communication system : A pilot study. *Applied Sciences*, 2022.
- [3] Bièche Matys Adéas. Low-cost underwater communication system for fish-type robot - source code. <https://github.com/AdeMBCH/Low-Cost-UnderWater-Communication-System-for-Fish-Type-Robot>, 2025.

Annexes

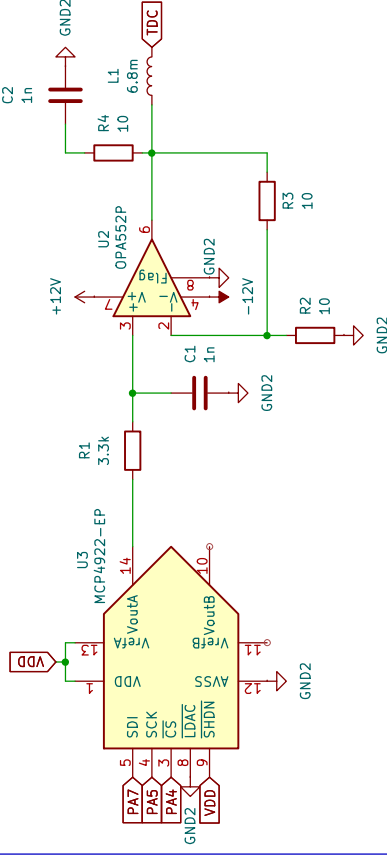
Underwater communication device for fish-type robot



Receiver Circuit



Emitter Circuit



Notes :

- This is my first time working on schematics and electronics, so I'd appreciate any advice you can share
- In controlled conditions, the multiple bandpass filter may be omitted from the receiver circuit.
- Amplifier supply voltages (V+ and V-) can vary based on the chosen power source.
- Resistor and capacitor values can change according to the power source and the target gain.
- Both STM32 boards are connected to the same ground reference

