FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Communications Module for a Light Autonomous Underwater Vehicle

Bernardo António Correia Gabriel

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: João Tasso de Figueiredo Borges de Sousa

October 28, 2024

Abstract

This dissertation presents the research, development, and implementation of a communications module for a Light Autonomous Underwater Vehicle, a project undertaken at the Underwater Systems and Technology Laboratory, LSTS, at the Faculty of Engineering of the University of Porto. The objective is to enhance the flexibility, power efficiency, and overall endurance of LAUVs, enabling more effective autonomous marine research.

The introduction provides the motivation for the project, emphasizing the critical role of oceans in climate regulation, biodiversity, and human livelihoods, as well as the need for advanced marine research technologies. The background chapter offers a comprehensive overview of LAUVs and their computational architecture, underlining their significance in marine research and the need for modular, adaptable hardware.

Then, the key challenges in payload integration, system flexibility, and power efficiency are identified, highlighting the necessity for a modular hardware approach to address rapid technological advancements and reduce costs. Subsequently, the state of the art reviews existing solutions and technologies in AUV hardware architectures, providing a critical analysis of current approaches and their limitations.

The methodology chapter details the design of the communications module, including the selection of the microcontroller, serial transceivers, power supply components, and interface options. It describes the schematic and PCB design processes and prototype assembly.

Afterward, the outcomes of the verification and validation process are presented, showcasing the performance and potential applications of the developed communications module. It discusses the tests conducted on the general board and specific serial ports.

The document ends with a summary of the achievements of the project, highlighting its contributions to the field of marine robotics and providing insights into future work and potential improvements to further enhance the LAUV's capabilities.

This work contributes to the broader field of marine technology by researching a robust, flexible, and efficient approach for a communications module for the LAUV, advancing the capabilities of autonomous underwater research systems.

Resumo

Esta dissertação apresenta a pesquisa, desenvolvimento e implementação de um módulo de comunicações para um Light Autonomous Underwater Vehicle, um projeto realizado no Laboratório de Sistemas e Tecnologia Subaquática da Faculdade de Engenharia da Universidade do Porto. O objetivo é melhorar a flexibilidade, a eficiência energética e o desempenho global dos LAUVs, permitindo uma investigação marinha autónoma mais eficaz.

A introdução fornece a motivação para o projeto, enfatizando o papel crítico dos oceanos na regulação do clima, na biodiversidade e nos meios de subsistência humanos, bem como a necessidade de tecnologias avançadas de investigação marinha. De seguida, é dada uma visão geral abrangente dos LAUVs e da sua arquitetura computacional, sublinhando a sua importância na investigação marinha e a necessidade de hardware modular e adaptável.

No capítulo seguinte são identificados os principais desafios na integração de novos dispositivos, flexibilidade do sistema e eficiência energética, destacando a necessidade de uma abordagem modular para enfrentar os rápidos avanços tecnológicos e a reduzir custos. Subsequentemente, o estado da arte analisa as soluções e tecnologias existentes nas arquiteturas de hardware dos AUVs, fornecendo uma análise crítica das abordagens atuais e suas limitações.

O capítulo de metodologia detalha o desenvolvimento do módulo de comunicações, incluindo a seleção do microcontrolador, transceptores para comunicações série, componentes de alimentação e opções de interface. Descreve também os processos de desenho do esquemático e da PCB, e a montagem de um protótipo.

Posteriormente, são apresentados os resultados do processo de verificação e validação, mostrando o bom desempenho e as potenciais aplicações do módulo de comunicações desenvolvido, expondo os testes realizados.

O documento termina com um resumo dos desenvolvimentos do projeto, destacando as suas contribuições para o campo da robótica marinha e fornecendo uma introspeção sobre trabalhos futuros e potenciais melhorias para aumentar ainda mais as capacidades do LAUV.

Esta investigação contribui para um lado mais amplo da tecnologia marinha ao desenvolver um módulo de comunicações robusto, flexível e eficiente para LAUVs, melhorando as capacidades dos sistemas autónomos de investigação subaquática.

United Nations Sustainable Development Goals

Table 1 specifies this project's contribution towards the United Nations Sustainable Development Goals, with possible performance indicators and metrics.

Table 1: United Nations Sustainable Development Goals

Goal	Target	Contribution	Performance Indicators and Metrics
4	4.3	Projects like this can be used in educational tools and kits to teach students about electronics, programming, and embedded systems, providing hands-on experience and practical skills in STEM fields.	Number of technical and vocational training programs that incorporate the PCB designed in this project.
	4.4	By developing accessible and affordable electronic development boards, you can help enhance technical skills that are crucial for employment in the tech industry, fostering a skilled workforce.	Number of certification programs that include training on using the PCB.
8	8.2	Innovations in microcontroller-based projects can lead to new products and solutions, boosting economic productivity and technological advancement in various sectors, including manufacturing, IoT, and automation.	Number of new products developed using the PCB designed in this project.
9	9.5	This project can support research and development in electronics and embedded systems, facilitating the creation of innovative solutions and the upgrading of industrial technologies.	Number of research projects utilizing the PCB designed in this project.
13	13.3	This project can support climate monitoring and environmental data collection systems, which are critical for climate action initiatives and raising awareness about environmental impacts.	Number of educational and awareness programs on climate action using data gathered using the PCB designed in this project.

SDG	Target	Contribution	Performance Indicators and Metrics
14	14.2	AUVs can gather important data on the health of marine ecosystems, including coral reefs, seagrass beds, and fish populations. This information aids in the development and implementation of strategies to protect and sustainably manage these ecosystems.	Number of areas identified for restoration or conservation based on AUV data.
	14.a	The deployment of AUVs can significantly enhance scientific research by providing detailed, real-time data on various aspects of ocean health. This contributes to the global understanding of marine environments and supports informed decision-making.	Number of new technologies or methods developed based on AUV capabilities.
17	17.8	By making the PCB design and applications open-source or collaborating with educational institutions and industry partners, you can promote knowledge sharing and collaborative innovation, helping to achieve sustainable development goals through technology.	Number of platforms or forums for sharing knowledge and expertise related to the PCB designed in this project.

Acknowledgments

I want to express my gratitude to Professor João Borges Sousa for his supervision during the development of this work. His guidance and proactive availability were essential for completing the present dissertation. This acknowledgment is extended to all LSTS staff who have been giving me constant support on my accomplishments, shaping my academic and professional paths.

A special regard to Fulbright, in the person of Otília Macedo Reis and Sofia Wahnon, and all the people involved, for giving me the possibility to travel to California to complete my research with a scholarship funded by FLAD, for which I manifest my abiding gratefulness.

To MBARI and the David and Lucile Packard Foundation, a word of appreciation for hosting me during my project and funding the fabrication of the final prototype. During my time at MBARI, Brett Hobson and his team were very enthusiastic and encouraging, to whom I will always be in debt.

I would like to show appreciation for projects GROOM II, reference 951842, and JUNO — Robotic Exploration of Atlantic Waters, reference 2021/0008, which allowed the unfolding of part of my Master's thesis activities.

My gratitude to the university colleagues I have met along the way, who became great friends and have been there for the good and the bad. My respect also goes to Professor Bruno Ferreira, who introduced me to marine robotics and sparked my interest in this field.

Ultimately, I want to thank my family for shaping my personality and giving me all the support I always needed, as well as my girlfriend Carolina for her companionship and understanding, even across 9000 Km. A final remark to all my friends and all the people with whom I crossed paths.

"If I Can't Scuba, Then What's This All Been About?"

Creed Bratton

Contents

List of Figures

List of Tables

Abbreviations

AI - Artificial Intelligence
AMD - Advanced Micro Devices, inc.
API - Application Programming Interface
AUV - Autonomous Underwater Vehicle
CPU - Central Processing Unit
DUNE - DUNE: Unified Navigation Environment
DVL - Doppler Velocity Logger
EDA - Electronic Design Automation
FLAD - Fundação Luso-Americana - para o Desenvolvimento
GB - Giga Byte
Gbps - Gigabit per second
GPS - Global Positioning System
GPU - Graphics Processing Unit
GROOM II - Gliders for Research, Ocean Observations and Management: Infrastructure and Innovation
I2C - Inter-Integrated Circuit
INS - Inertial Navigation System
IMC - Inter-Module Protocol
I/O - Input/Output
IoT - Internet of Things
LAUV - Light Autonomous Underwater Vehicle
MBARI - Monterey Bay Aquarium Research Institute
MCU - MicroController Unit
mSATA - mini Serial Advanced Technology Attachment
NRE - Non-Recurring Engineering
LSTS - Laboratório de Sistemas e Tecnologia Subaquática
PCB - Printed Circuit Board
PCIe - Peripheral Component Interconnect express
RAM - Random Access Memory
ROV - Remote Operated Vehicle
SBC - Single Board Computer
SDK - Software Development Kit
SMD - Surface-Mount Device
SPI - Serial Peripheral Interface
SoC - System on a Chip
SoM - System on Module
STEM - Science Technology Engineering and Mathematics
UART - Universal Asynchronous Receiver-Transmitter
USB - Universal Serial Bus

Chapter 1

Introduction

This chapter will state the motivation that supports this dissertation and describe the document structure.

1.1 Motivation

With the growing recognition of the oceans' critical role in regulating climate, supporting biodiversity, and sustaining human livelihoods, marine research is becoming a trend. Technological advances have made deep-sea exploration more accessible, revealing some of the oceans' mysteries and highlighting the urgency of addressing issues such as pollution, overfishing, and climate change. Increased public and governmental awareness of the importance of marine ecosystems has promoted investment in ocean sciences, aiming to protect, monitor, and manage these vital resources for future generations.

This popularity has contributed to the development of diverse marine systems, which have evolved into increasingly autonomous systems. Nowadays, most of these systems are designed for a given application and tend to have low pliability for upgrades. In terms of the hardware used inside these vehicles, with the constant development of technologies, the poor flexibility of these systems leads to architectures that quickly become outdated, which can cause an increase in costs and hinder the implementation of newer technologies.

The push for more and more autonomy in these systems goes along with the interest in extending their endurance, which improves the persistence and reliability of these research methods. This however brings even more challenges for the design and construction of an autonomous system, as they face constraints regarding power availability and computational performance while requiring the handling of communications with various payload devices, that enable the desired autonomous operation of this type of system. Therefore, the need for a low-power consumption multi-purpose communications platform for these autonomous systems becomes self-evident.

This project emerged from the developments undertaken by the Underwater Systems and Technology Laboratory (LSTS) [1] at Porto University led by Professor João Borges de Sousa, which is an interdisciplinary research laboratory established in 1997 with researchers specialized in the design, construction, and operation of unmanned underwater, surface and air vehicles and on the development of tools and technologies for the deployment of networked vehicle systems.

LSTS believes that the future of marine research will consist of autonomous robotic vehicles connected via inter-operating networks to perform complex tasks in the ocean. The integration and interaction of these systems into new and modern inter-operating networks will only be more manageable and maintainable if the hardware framework that supports these devices is also modular.

With that in perspective, LSTS has been developing an open-source software toolchain that enables this interoperability. This toolchain covers many levels: from the onboard navigation software for the autonomous vehicles, DUNE [2], to a command and control platform for mission planning and execution, Neptus [3], all aggregated by a common inter-module communication protocol, IMC [4]. They even provide a Web-based application for situational awareness and marine data dissemination, Ripples [5].

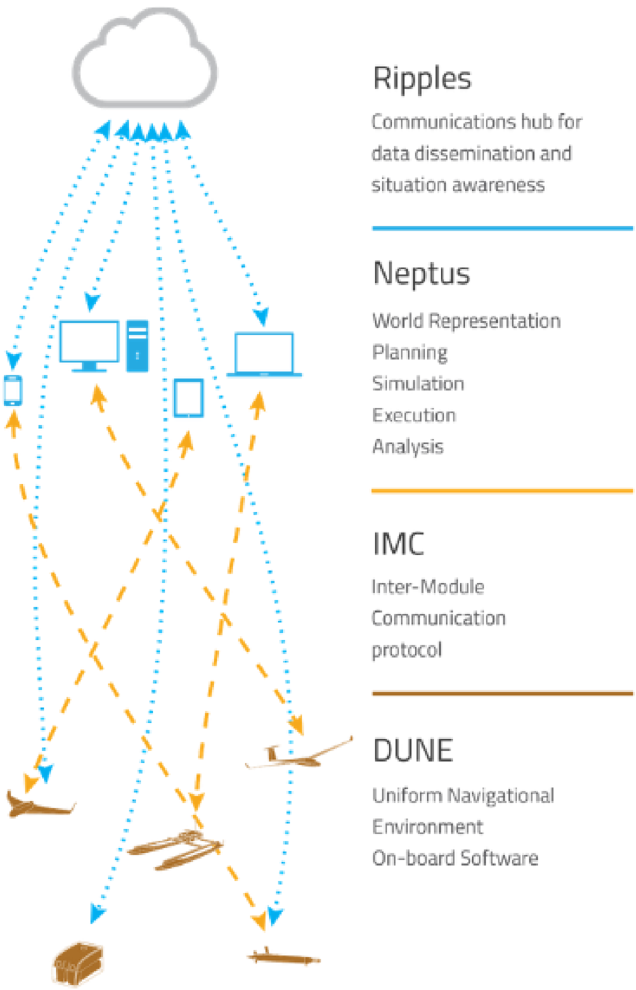

Figure 1.1: LSTS open-source Software Toolchain.

1.2 Fulbright Scholarship

The present work was developed under a Fulbright scholarship with the support of the Luso-American Development Foundation, FLAD. This scholarship funded the travel to California, USA, to be hosted at the Monterey Bay Aquarium Research Institute, MBARI. The MBARI staff gave essential technical guidance for the research of the design of the board and manufacturing of the prototype, which was funded by the David and Lucile Packard Foundation.

1.3 Publications

Alongside this dissertation, an extended abstract titled "Communications Module for a Light Autonomous Underwater Vehicle" was accepted for participation in the 2024 IEEE OES AUV Symposium, and the subsequent paper has been submitted to be presented at the conference.

1.4 Document Structure

The document follows the next outline:

- Chapter 2 provides an overview of the necessary background that substantiates this work. It focuses on the computational architectures of AUVs, exploring in detail the vehicle developed by LSTS, the LAUV.
- In chapter 3, the problem that defines this project is specified.
- Considering the problem definition, chapter 4 points out limitations in previous approaches by exploring new perspectives on how to solve those challenges.
- In chapter 5, the approach is defined, alongside an extensive description of the methodology followed towards the proposed solution.
- With the finished prototype assembled, chapter 6 reports the results of the verification and validation process, with remarks on possible applications of the final product.
- Finally, chapter 7 concludes the document by presenting considerations about the developments done and future work.

Chapter 2

Background

This chapter will cover the essential context to understand the research conducted in this thesis. This background will not only highlight the significance of the problem addressed but also elucidate the theoretical and empirical foundations that inform the approach. It provides a comprehensive overview about the Light Autonomous Underwater Vehicle, which inspired this project, and a brief survey about AUVs hardware architectures, with focus on their computational systems.

2.1 Autonomous Underwater Vehicles

With the rising popularity of underwater research comes the need to find the best and most efficient ways of optimizing the exploration of this medium. In this sense, increasingly autonomous underwater assets, such as ROVs and AUVs, have been developed for various purposes. These applications range from military goals to oceanographic research, broadly impacting this field of study.

Having a system capable of executing and completing tasks autonomously is a great improvement, when compared to systems that require an operator or human intervention. These systems enable the possibility of reaching scenarios and observing phenomena otherwise inaccessible by humans. Although this topic has been getting a rise in popularity in recent years, these autonomous systems have been proving to be very efficient and relevant since the beginning of the century [6], [7].

Nowadays, increasingly complex algorithms are being developed [8], [9], achieving high levels of accuracy for underwater navigation. In this sense, the technology inside these systems needs to evolve in parallel with those theoretical breakthroughs, because they require high computational performance.

In an autonomous system, the computational unit is a pivotal component, enabling it to perform autonomous operations through data processing, control, and communication. Understanding the architecture and functionality of the computational unit is essential for designing efficient and capable AUVs, tailored to meet the specific requirements of various underwater missions. Unlike

ROVs, which require a human operator to control the vehicle, AUVs rely on their own "control system, computing, and sensing capabilities" [10].

Figure 2.1 shows a simplification of a General System Breakdown for an AUV.

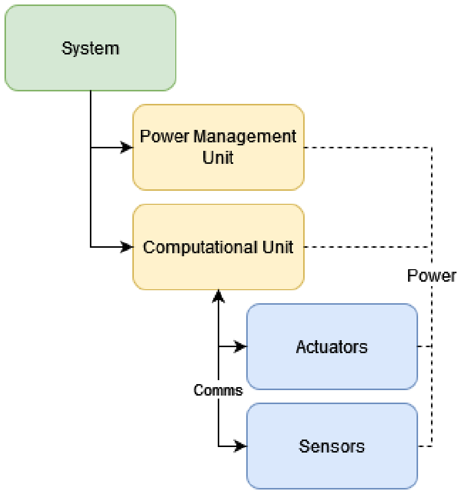

Figure 2.1: General System Breakdown of an AUV

2.2 Towards a Long Endurance Vehicle

Apart from its autonomy, another key aspect of AUVs is their endurance. Long endurance systems are highly relevant as they enable continuous and extensive data collection over longer periods, which is crucial for comprehensive environmental monitoring, scientific research, and resource exploration. These vehicles reduce operational costs and logistical challenges by minimizing the need for frequent deployment and retrieval while enhancing safety by operating autonomously in hazardous or remote areas.

There are currently many long-endurance AUVs, which have been showing notorious results during sea trials worldwide. For example, the Tethys-class AUV developed by Monterey Bay Aquarium Research Institute [11], combining a buoyancy engine and a shifting mass with an optimized propeller, the Autosub "capable of long ranges at low speed" [12], the Pipeshish AUV [13], with its hovering capabilities, the Explorer1000 [14], ranging 517 Km at 2 and 3 knots, and the Sea-whale 1000 AUV, which "owns the maximum measured range of sea trials" [15], with 4196 Km. Moreover, there are similar types of vehicles, such as buoyancy-driven gliders, with even greater ranges than these AUVs. In short, their ability to perform long-term missions supports critical applications in oceanographic data collection, underwater infrastructure inspection, and climate change studies, making them indispensable for modern underwater exploration and operations.

However, to accomplish long endurance and accommodate the required batteries, these systems are typically bigger in size and may require complex logistics for deployment and retrieval.

2.2.1 Light Autonomous Underwater Vehicle

The Light Autonomous Underwater Vehicle developed by LSTS [16], on the other hand, appears with a promising concept that focuses on providing a lightweight vehicle designed to be launched, operated, and recovered by regular users with minimal operational setup and training. Since this project emerged from the developments done by LSTS in their LAUV, a crucial step is to explore and understand its current architecture, identify its main characteristics, and evaluate how significant this project is concerning the overall improvement of this system.

Each LAUV is designed for specific applications, which are achieved by the usage of different sets of payloads. However, regardless of its applications, there are common elements for these systems:

- **Battery pack** - each pack is composed of various parallel sets of 18650 Lithium-Ion batteries arranged in a 7 series/3 parallel format. This arrangement increases the battery life of the vehicle, meaning that the higher the number of battery sets the higher the battery life will be (the number of sets depends on the space available inside the frame of the vehicle).
- **Battery Management System** - this unit is responsible for monitoring and protecting the battery pack.
- **Power Controller** - this unit is responsible for the distribution of power to all the components of the system.
- **Propulsion System** - constituted by one thruster and four rudders. [1]
- **Primary CPU** - the main computational unit responsible for the Navigation system.
- **WiFi modem** - responsible for surface communications with control units via WiFi.
- **GSM modem** - responsible for surface long-distance communications using the cellular network.
- **IRIDIUM modem** - responsible for surface long-distance communications using the Iridium satellite network.
- **Attitude and Heading Reference System** - responsible for providing attitude [2] information about the vehicle.
- **Global Positioning System** - responsible for providing the global position of the vehicle while on the surface.
- **Depth sensor** - responsible for measuring the depth of the vehicle.

[1] The thruster and the rudders are controlled by two independent units that receive high level commands from the CPU through serial communication.

[2] Attitude information includes yaw, pitch and roll, which represent the orientation of the vehicle regarding each axis of the reference frame.

Additionally, other elements are common to any application, even though they are optional:

- **Secundary CPU** - responsible for low power applications.
- **Acoustic modem** - responsible for underwater communications.

These are the essential components that constitute the LAUV. Then, there are two main versions: one version is designed for oceanographic purposes and the other is designed for mapping purposes.

Oceanographic

The Oceanographic version of the LAUV focuses on the collection of data for water quality measurements and biological parameters. Therefore, to achieve these goals, these vehicles may be equipped with the following sensors:

- **CTD sensor** - responsible for measuring essential properties of the water, such as: Conductivity, Temperature and Depth.
- **Turbidity sensor** - responsible for measuring the amount of suspended particles in the water.
- **Chlorophyll sensor** - responsible for measuring the phytoplankton concentration in the water.

Mapping

The Mapping version of the LAUV is a high-end system designed for seafloor mapping. Being equipped with:

- **Camera** - to capture underwater images.
- **Forward Looking Sonar** - responsible for object detection in front of the vehicle.
- **Sidescan Sonar** - responsible for 2D mapping of the seafloor.
- **Multibeam Sonar** - responsible for 3D mapping of the seafloor.

Since these vehicles are used for mapping large areas of the seafloor, they are required to be underwater for longer periods, depending on a precise position prediction, because there is no GPS signal underwater for position correction. Therefore, they are additionally equipped with a high-performance **Inertial Navigation System** and a **Doppler Velocity Logger**, used to measure the velocity of the vehicle relative to the seafloor, which combined allow for more accurate position estimations. For special missions, this type of vehicle can be equipped with a **Magnetometer**, used for the detection of metallic objects underwater, like mines or pipelines.

Figure 2.2 shows a diagram of the general architecture of the LAUV.

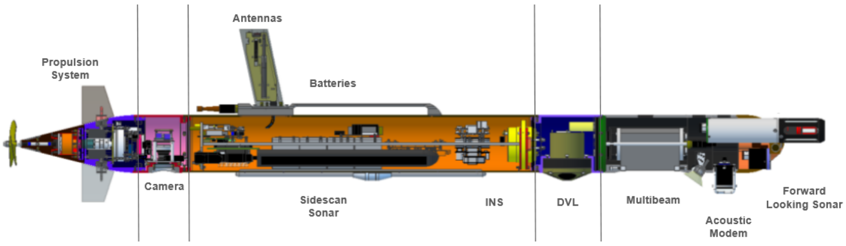

Figure 2.2: LAUV general architecture.

The LAUV uses a Rhodeus Single Board Computer (SBC) by Diamond Systems that features the low-power processor Geode LX800 from AMD, offering a fully integrated solution designed to operate in harsh and unattended environments. Although very robust, this SBC is becoming outdated, and some LAUVs have switched to the Raspberry Pi 4B SBC, which is a much more modern and powerful solution. The LAUV needs to adapt in parallel with the constant evolution of technology because as technology progresses, new sensors and payloads are being developed and the interest in integrating those new devices inside these vehicles is self-evident. A system like the LAUV could be upgraded to extend its endurance while maintaining its focus on a smaller logistical setup.

2.3 AUV Hardware Architectures

One of the main challenges concerning the development of this kind of system is related to their computational units. As the central component of the vehicle, the onboard computer will be one of the main constraints of the overall system. Therefore, is important to assess how autonomous underwater vehicles are typically architectured, focusing on their computational units. This way it will be possible to identify the limitations of those architectures and then start to define the requirements for a new proposal.

Considering what was stated before, this section of the chapter will focus on surveying examples of AUVs not only to analyze different techniques used in those scenarios but also to recognize similarities between them, to evaluate the possibility of designing a solution compatible with multiple layouts.

2.3.1 Single Board Computers

Single Board Computers are a popular choice for the computational units for these systems due to their compact form factor, ease of use, and integration of multiple functions on a single board.

An SBC typically integrates several key components onto a single Printed Circuit Board (PCB):

- **Processor** - the central unit responsible for the processing data and the execution of instructions.
- **Memory** - for storage, including data, operating system, and applications.
- **Input/Output Interfaces** - for communication and interaction with other components of the system, which consists of USB ports, general purpose input/output pins, serial interfaces, and network interfaces.
- **Power Suply** - which provides the necessary voltage and current for all the board components and other connected peripherals.
- **Graphics Processing Unit** - which handles rendering and graphical processing tasks.

Based on this, SBCs provide a lot of benefits. By having all the essential components on a single board, the need for additional hardware is reduced and it simplifies the design, making it also generally more affordable, with a wide range of options that comply with different budget constraints and applications. Additionally, SBCs have large user communities, with extensive libraries and support from manufacturers, making them ideal for quick development and testing.

However, this fixed-body architecture induces some limitations. The fixed hardware configurations provided by SBCs can limit the possibility of upgrades or customization of specific components. Moreover, high-performance SBCs consume significant power and high processing loads can lead to overheating, which can be a significant drawback for battery-operated AUVs.

With that in mind, to comply with the requirements for a specific system or autonomous vehicle, it is needed to find a solution that fulfills them. In many cases, this task becomes increasingly difficult, because there is no commercial option available or there is a budget constraint.

SBC examples

Raspberry Pi 5

The Raspberry Pi 5 features a custom-built Broadcom BCM2712 SoC with a quad-core ARM Cortex-A76 CPU and VideoCore VII GPU, offering enhanced performance and graphics. It includes dual-band Wi-Fi 6, Bluetooth 5.0, a 2.5 Gbps Ethernet port, and USB 3.0 for improved connectivity, available with 4 GB and 8 GB RAM. It provides better performance and new features when compared to previous solutions in the Raspberry Pi family while still providing a versatile and affordable choice for a wide range of computing projects.

Figure 2.3 shows the Raspberry Pi 5.

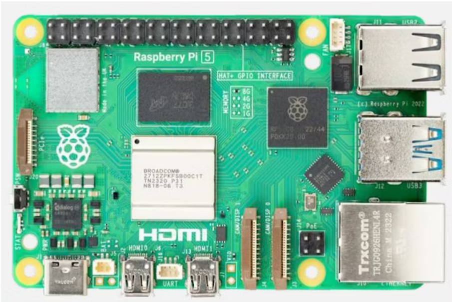

Figure 2.3: Raspberry Pi 5

Versalogic Tetra

The Versalogic Tetra SBC is powered by the NXP i.MX 6 processor, designed with the ARM Cortex-A9 architecture, supports up to 2 GB of RAM and offers flexible storage options, ensuring efficient operation in demanding environments. Its low power consumption makes it ideal for battery-powered or energy-sensitive applications, such as industrial automation, transportation, and medical devices. The Tetra's rugged design, capable of withstanding extreme temperatures and harsh conditions, combined with versatile connectivity and expansion options, makes it a reliable choice for various high-efficiency embedded system needs. The manufacturer provides various expansion modules for specific applications using the mSATA and mini PCIe ports.

Figure 2.4 shows the Tetra SBC.

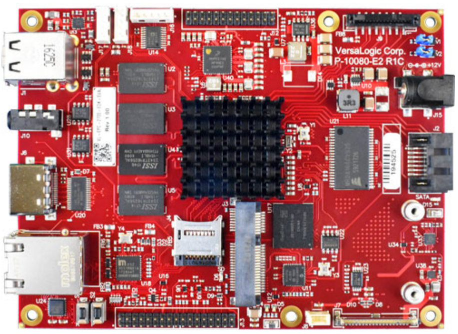

Figure 2.4: Versalogic Tetra

AUV examples using SBCs

LAUV by LSTS

The LAUV typically uses a Rhodeus SBC by Diamond Systems [17]. It features the low-power processor Geode LX800 from AMD, offering a fully integrated solution designed to operate in harsh and unattended environments. This SBC provides a PC/104 expansion bus and supports the Linux operating system. Although very robust and power-efficient, this SBC is becoming outdated, and some LAUVs have switched to the Raspberry Pi 4B SBC [18], a much more modern and powerful solution.

IVER4 900 by L3Harris

The IVER4 900 AUV by L3Harris [19] is a compact and versatile autonomous underwater vehicle designed for various marine applications. It features advanced navigation systems, a modular design for easy deployment, and the ability to carry a range of sensors and payloads. Its autonomous capabilities allow it to perform long-duration missions with minimal human intervention, making it suitable for underwater inspections, environmental monitoring, and scientific research. The IVER4 900 is built to endure harsh underwater conditions, offering reliable performance in deep and challenging environments. It uses an SBC based on the Intel Celeron N2930 SoC Processor running the Microsoft Windows Embedded operating system.

Figure 2.5 shows the IVER4 900 AUV.

Figure 2.5: IVER4 900 by L3Harris

2.3.2 System On Module

System on Modules (SoMs) offer a more flexible and scalable alternative to SBCs in the architecture of AUVs. SoMs integrate the core components of a computer, such as the CPU, memory, and essential peripherals, on a small, replaceable module. Their modular nature allows for easy upgrades and customization, making them suitable for a wide range of applications, from basic control tasks to advanced AI and data processing.

In [20], the author summarizes the interest by developers in this technology in 3 major reasons: "SoMs encapsulate design complexity, are easy to use and provide a high degree of capability in a small form factor". Even though the work mentioned focuses on space hardware development, the list of challenges faced during the development process is shared. The author mentions that from mission to mission, the requirements continuously change with the constant evolution of technology, which means that "some amount of hardware Non-Recurring Engineering costs will always be necessary". Ideally, like in the marine robotics field, the NRE should be reduced as much as possible to mitigate constant upgrades.

Similarly to SBCs, SoMs typically include the **processor**, **memory**, and the **graphics processing unit**. The major difference between these two types of computers relates to their connectivity and how they provide connection to access interfaces and peripherals. SoMs provide a high-density board-to-board connector that interfaces the SoM with a carrier board. This carrier board is responsible for providing additional I/O and power connections.

Compared to the previous approach SoMs allow for easy upgrades and customization because by replacing the module, the system can be updated with newer processors and capabilities without redesigning the entire system, only if the board-to-board connector remains the same. Additionally, this approach opens the door to scalability, since the carrier boards can be designed to meet specific requirements, not limiting it to market available options. Furthermore, many are designed for lower power consumption and are usually manufactured in a small footprint, ideal for more compact and power-efficient AUV designs.

Nevertheless, this increase in modularity and scalability doesn't only translate into benefits. Developing a custom carrier board requires additional design effort and expertise, as well as potentially higher initial costs compared to off-the-shelf SBC solutions. On top of that, the lack for a universal standard for board-to-board connectors means that carrier board must often be customized for specific SoMs, hindering upgrades and cross-compatibility. Even SoMs provided by the same manufacturer may use different connectors and generally those connectors supply different peripherals.

SoM examples

NVIDIA Jetson Nano

The NVIDIA Jetson Nano is a compact, powerful computer designed for AI and robotics applications, featuring a quad-core ARM Cortex-A57 CPU and a 128-core Maxwell GPU. Connectivity includes Gigabit Ethernet and multiple USB ports, with HDMI and DisplayPort outputs. It runs on NVIDIA's JetPack SDK, supporting AI frameworks like TensorFlow and PyTorch. It is ideal for developers and hobbyists to build AI projects like robotics, drones, and smart cameras.

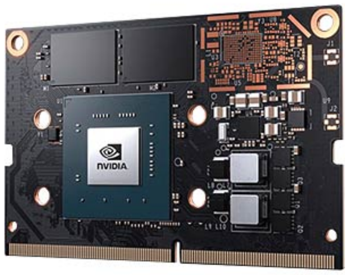

Figure 2.6: NVIDIA Jetson Nano

DART-MX95

The Dart-MX95 by Variscite is a compact SoM powered by the NXP i.MX 95 processor, designed for balanced high-level and real-time processing. It offers extensive connectivity, including Ethernet, Wi-Fi, Bluetooth, and various peripheral interfaces. It is ideal for industrial automation, IoT, medical equipment, and multimedia systems. With efficient power management and support for Linux and Android, the Dart-MX95 is suitable for diverse embedded applications.

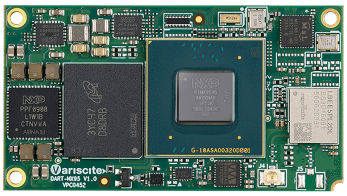

Figure 2.7: DART-MX95

2.3.2.1 Long Range Autonomous Underwater Vehicle by MBARI

As a collaboration with MBARI, this project benefited from having the possibility to observe how they address this problem in their Tethys-class AUV, the LRAUV [21].

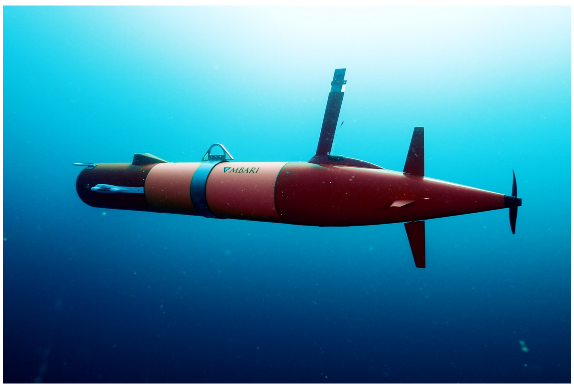

Figure 2.8: LRAUV by MBARI

The computational unit of this system is composed of a power-efficient LPC3250 SoM, compatible with the Linux operating system, consuming a total of 1/3 Watts [22]. Its carrier board provides physical access to the essential peripherals and components and manages the supply of power to the processor. In addition to the main carrier board, this system uses an expansion board responsible for handling serial communication between the CPU and the payload devices. This extra board enables the connection of 24 payload units, each one responsible for managing power distribution for individual sensors and other system devices while handling data exchange. The 24 units are aggregated in groups of 8, where each group shares a common SPI bus and the expansion board interfaces with the main carrier board, through the SPI buses. Additionally, each unit has a chip select pin, necessary for the SPI protocol, being multiplexed using a 3 to 8 decoder, that enables the usage of the bus.

This alternative provides a very sophisticated solution, enabling individual control of 24 power and communication channels. However, this increase in complexity brings problems related to synchronization. Since 8 devices are using the same SPI bus, the computational unit has to precisely synchronize the access of each unit to the bus, otherwise data loss would occur or the main unit could receive data from one device and interpret it as data from another.

On the other hand, tackling the serial communications problem in a device outside the carrier board means that this approach is not exclusive to a specific SoM or processor. By having a standard interface medium, in this case, the SPI bus, virtually any CPU could be expanded by this platform, given that it provides the necessary connection, independently of being an SoM with a carrier board or an SBC.

2.3.3 Serial Communication Expanders

Serial communication expanders are hardware modules that allow a CPU or microcontroller to manage additional serial devices beyond its native capabilities. These expanders are essential in systems where the number of UART interfaces provided by the CPU is insufficient for the required serial communication tasks. They typically use multiplexing techniques to manage multiple UART channels through a single or fewer physical connections, interfacing with the CPU via protocols like USB, PCIe, I2C, or SPI. They are crucial in applications such as embedded systems, industrial automation, and communication systems where multiple UART connections are necessary for interfacing with various peripherals.

Examples of Expanders

PC/104

Numerous SBCs available on the market provide a PC/104 connection. Therefore, there are many serial expanders compatible with this type of connector. The PC/104 bus is a compact and rugged modular computer standard designed for embedded systems. It uses a stackable design, allowing multiple modules to be combined in a compact form. Its design ensures durability in harsh environments, with variants such as PC/104-Plus and PC/104-Express supporting PCI and PCI Express, enhancing functionality with newer peripherals.

Figure 2.9 shows an example of an expansion board that uses the PC/104 bus, the Xtreme PCI-104 Opto by Connect Tech Inc. [23] that provides access to 12 additional optically isolated multi-protocol serial ports.

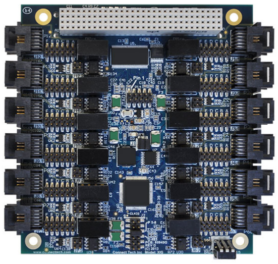

Figure 2.9: Xtreme PCI-104 Opto by Connect Tech Inc.

M.2.

Figure 2.10 shows the MPCIE-UART-KIT01 by IEI Integration Corp. [24], a serial expander that provides 4 additional RS232 serial ports using the M.2. connector with support for PCIe.

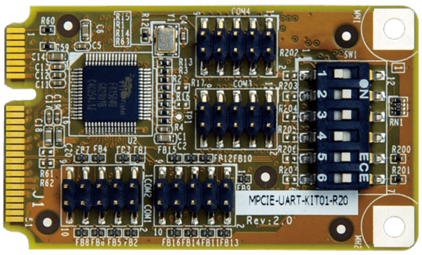

Figure 2.10: MPCIE-UART-KIT01 by IEI Integration Corp.

USB

Figure 2.11 shows the USB-COM232-PLUS4 by FTDI [25] that converts a USB port to 4 independent RS232 serial ports.

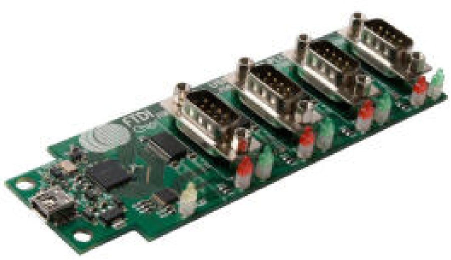

Figure 2.11: USB-COM232-PLUS4 by FTDI

Ethernet

Figure 2.12 shows the ES-279 by Brainboxes [26], an Ethernet to serial adapter that expands connectivity to 8 additional RS232 serial ports, providing a Web Interface for configuration.

Figure 2.12: ES-279 by Brainboxes

Chapter 3

Problem Definition

As shown before, the demand for efficient underwater exploration is growing rapidly. With this goal in mind, technology has been working alongside this aspiration, with increasingly autonomous systems arising. However, the complexity of integrating various payload devices in those systems, maintaining their flexibility and ensuring power efficiency poses significant challenges. Considering the overall study presented in the previous chapter, it was clear that although the typical approaches reach the desired functionality, enabling the systems to be completely autonomous and achieving notorious results, they may face problems in complying with the constant evolution of technology. There are some available options in the industry that could be used to create more distributed architectures, which would, in theory, allow for more pliable systems. Despite that, these solutions are designed for general applications, which may not be optimal for these vehicles. On the other, many of the available solutions may introduce new challenges in integration, considering software compatibilities. This project aims to address these issues by developing a programmable and power-efficient multi-propose communications module specially designed for autonomous systems.

3.1 Payload Integration Challenges

One of the primary challenges in developing autonomous underwater systems is the integration of diverse payload devices, including sensors, communication tools, and actuators. These payloads are essential for achieving the desired autonomous behavior of the system, enabling it to interact with and be aware of its environment, performing designated tasks. These devices generally communicate using standard serial protocols such as RS232, RS485, and RS422. Therefore, the central computational unit must efficiently handle these protocols to ensure proper system functionality.

3.2 System Flexibility

The rapid pace of technological advancement underscores the need for flexible and upgradable system architectures. Then, the goal is to create a scalable and modular hardware structure that can accommodate future technological developments without significant redesigns. By focusing on a modular and distributed approach, the system can be easily upgraded with new modules, ensuring longevity and adaptability.

3.3 Power Efficiency and Endurance

For autonomous underwater systems, power efficiency is crucial for extending operational endurance. Extended missions enable the collection of more comprehensive data and observations of ocean phenomena that would be difficult to analyze otherwise. Therefore, the challenge lies in developing a system that balances power consumption with performance. This power-efficient design is essential for maximizing the autonomy and operational lifespan of the system.

Chapter 4

State of the Art

This chapter complements the background presented in preceding chapters, by focusing on the problem previously defined. First, a project that tried to tackle this challenge will be analyzed to identify its achievements and the possible limitations of its approach. Finally, the focus will be redirected towards microcontroller platforms which have become popular alternatives for data acquisition modules and may provide a new perspective on how to solve this problem.

4.1 BALU motherboard

Regarding the SoM paradigm, is important to mention the first effort made to integrate this technology in the LAUV: BALU. As stated in [27], BALU is "a Main Board capable of yielding 9 expansion modules", designed for the LAUV, which proved that designing a modular motherboard for this system is possible. The main focus of this proposal was to prove the possibility of using off-the-shelf components with standard M.2 connectors, simplifying the cabling inside the vehicle and enabling the upgradability with modern modules. Additionally, this work describes the development of custom-made modules designed with the same M.2. connectors to provide specific features required. This was an ambitious and fruitful project since it proved to be considerably modular, as demonstrated by enabling the utilization of commercial components such as a Solid State Drive and Wi-Fi, Bluetooth and Cellular modems. The BALU board was designed to use the Raspberry Pi Compute Module 4, a SoM developed by Raspberry Pi, with a board-to-board connector that allows integration into the motherboard.

This design only requires one USB port, one PCIe port, one Ethernet port, and two SPI interfaces from the SoM. These general specifications are common features of most CPUs on the market, meaning that this board is compatible with many processors, only requiring the correct interface connection between the carrier board and the SoM.

In a general overview of the project, there are three key features that it uses to achieve modularity: **USB** hubs, **PCIe** switch, and **UART** multiplexers with SPI interface.

By focusing on providing self-expanding peripherals, this work achieves a high level of abstraction from the SoM architecture. This is a significant improvement, especially considering UART connectivity, as the offer for processors with multiple serial ports is very reduced.

However, this means the processor may have relevant peripherals that would not be used. For example, there could exist an SoM with eight native UART ports provided in the board-to-board connector and with this carrier board design, none of them would be used. Although the USB to UART and SPI to UART multiplexers may effectively expand serial ports, it is still unclear how compatible they are with different CPUs and operating systems. The same stands for the USB ports: this design uses USB hubs daisy-chained to provide a USB port in each M.2. connection, which may not be the optimal solution for a specific processor.

In this context, even if this approach solves the connectivity problem and is compatible with a wide range of SoMs, this would not be optimal for many available CPU options.

4.2 Using External Modules for Connectivity Expansion

Considering what was observed in the previous work, it is possible to infer that trying to solve this issue of connectivity directly on the CPU level will eventually result in a solution that may face the same issues as the previous approaches. Even if this work introduces a much more modular technology, the SoMs, it restricts the architecture of the vehicle to that type of CPU, closing the possibility of using any other processor not complying with that design.

On the other hand, recovering the solution used by MBARI in their Tethys-class AUV, it is clear that it is possible to externalize the handling of serial communications in separate modules, interfacing them with the main control unit by a conventional protocol. In their solution, the communications are handled by independent card-like modules, which are aggregated in an interface board that provides access to power and the shared communication bus.

Figure 4.1 shows a close-up picture of the interface board slots and the communication modules for the Tethys-class AUV.

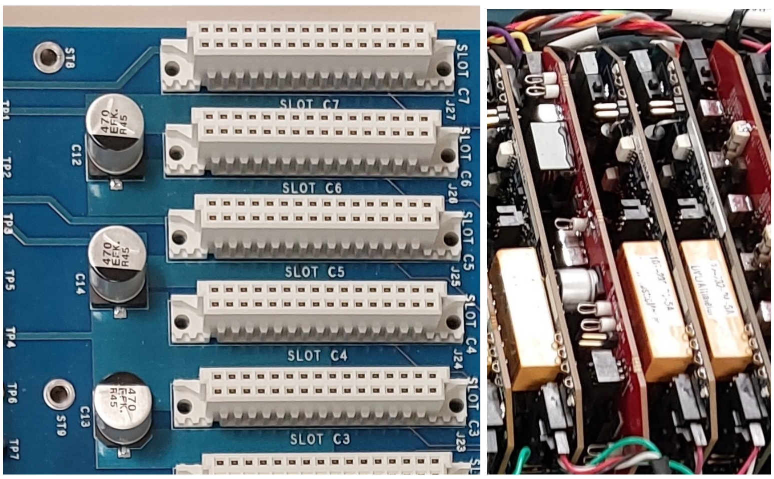

Figure 4.1: Close-up pictures of the interface board slots and the communication modules.

Other works have explored this type of approach. In [28], the authors describe the PXIe system, which consists of a backplane that interfaces three separate modules: the system controller board, with a user interface through an operating system, the system timing board, that handles synchronization between devices, and at least one peripheral module, that provides specific functionalities. All these modules interface in the backplane establishing communication through a PCIe bus. Although this work focuses on the development of electronic test equipment and modular laboratory instruments, not remotely related to autonomous systems, it is a good example of how to achieve modularity in a standardized format. Just like in the MBARI solution, this work focuses on having external modules, the peripheral modules, that implement direct memory access and programmed input/output connected to the host system, in this case, through the PCIe bus.

Another great work that provides a similar solution is [29], which focuses on the development of a multiprotocol asynchronous serial communication module using Mbus. It uses an FPGA chip to handle the serial communication with up to eight RS232 devices or up to four RS485/RS422 devices, interfacing with the host system through the Mbus connecting to a carrier board.

With that in mind, the logical way to advance is to incorporate this concept of external modules and try to integrate this logic in a more comprehensible way. This is actually the approach used in the typical serial expanders commercially available, mentioned before, although they do not offer programmability, having to rely on proprietary software drivers.

In recent years, FPGA technology has been competing with up-and-coming microcontrollers, that provide a more easy-to-use and more affordable option for embedded solutions. Therefore, unlike the last two works which use FPGAs, perhaps using a microcontroller would bring different benefits. Microcontrollers have been used in many different applications regarding data acquisition and data processing: for monitoring the state of biological objects in sound recognition systems [30], for sampling analog/digital and temperature data [31], for data acquisition and classification with Web-based configuration and visualization [32], and for real-time measurements of data to be analyzed and monitored through computer applications like Matlab-Simulink [33] and LabView [34]. Moreover, there are works that describe using microcontrollers with Ethernet interfaces using the LwIP protocol [35], [36], which may be relevant to the proposed solution. Making them a good option for developing a complex module in a more accessible way.

Chapter 5

Methodology

Drawing inspiration from the modular and distributed architecture of the LAUV arises the solution proposed in this work. The main computational unit of the LAUV is responsible for the high-level command and control of the system, and additional modules are responsible for interpreting and implementing the lower-level execution of those commands. Following this paradigm, an additional module could be responsible for expanding the connectivity of the computational unit or acquiring specific payload data.

Similar to the approach followed at MBARI in their LRAUV, bi-directional data exchange is possible by choosing a standard communication interface with the main unit, and the underlying problem of cross-compatibility is solved. Moreover, as this solution would not be integrated into the carrier board for an SoM or adapted to comply with a specific expansion medium for an SBC, it could be compatible with a broader range of processors.

Subsequently, before starting to frame the module's design, choosing the components that will achieve the desired functionalities is crucial. The choice of the microcontroller is decisive for the overall design, as it will determine which and how many peripherals are available. As the central component of the board, the MCU will be the component consuming the most power, which means that the power supply unit should be outlined with that in consideration.

5.1 Approach Definition

With that in mind, the proposition is to develop a PCB centered on a microcontroller that handles data exchange with payload devices via serial communication, ideally using any of the three main protocols, RS232, RS485, and RS422, and interfaces with a main unit through a standard medium. Microcontrollers are generally more power efficient than processors or microprocessors while still being able to perform high-speed communications, making them the best option for designing a low-power solution. In addition, microcontrollers have become increasingly accessible to students. The widespread availability of affordable development boards, comprehensive educational resources, and supportive communities has promoted more straightforward access to these tools, enabling students of different levels to partake in embedded systems and hardware programming. Although this is not one of the main goals of this project, as in [37], it is very important to consider these platforms as educational tools. Therefore, it would be very interesting to develop an efficient communications module that allows students to learn from it.

The approach and its requirements were defined with the help of the LSTS staff, resulting in the set of general requirements shown in table 5.1.

Table 5.1: General requirements for the design.

Identifier	Description
R1	Provide connectivity for at least eight serial devices
R2	Enable interaction with multiple serial protocols (RS232/RS485/RS422)
R3	Choose a standard protocol for interface with main unit (for example, SPI, USB or Ethernet)
R4	Support a wide range of input voltage (from 5V up to 30V)
R5	Consume up to 1 W of power

The schematic was made under the LSTS staff's supervision and revised in a design review at MBARI with field specialists.

5.1.1 Options for the interface between the module and the main unit

The interface between the module and the main computational unit of the system requires a high-speed protocol to achieve the desired functionality of expanding its connectivity to a higher number of devices communicating through UART. Communication protocols such as CAN and I2C have been used in many applications. Yet, they are not ideal for high-speed communications due to their lower data rates and higher latencies. However, in embedded systems and microcontroller applications, communication interfaces like SPI, USB, and Ethernet play crucial roles in high-speed communication, each offering distinct advantages and use cases [38].

The Controller Area Network (CAN) is a robust protocol for automotive and industrial applications, offering reliable communication in noisy environments. Although its error-checking and fault confinement mechanisms may provide high reliability for safety-critical applications, they also impose limitations in data rate, payload data, and efficiency. Additionally, the Inter-Integrated Circuit (I2C) protocol provides simplicity and efficient use of pins, requiring only two lines for data and clock, making it an excellent choice for onboard communication. However, it is generally slower and not suitable for long-distance communication.

On the other hand, the Serial Peripheral Interface (SPI) is a synchronous serial communication protocol known for its simplicity and high-speed data transfer. Compared to more complex protocols, it lacks standardization for long-distance communication and multi-device management, so it is ideal for short-distance communication between microcontrollers and peripherals.

Moreover, the Universal Serial Bus (USB), in contrast, provides a standardized, versatile interface for connecting a wide range of devices, supporting data transfer and power supply. USB's plug-and-play capability and widespread adoption make it a preferred choice for applications requiring easy connectivity and reliable communication with host devices.

Finally, Ethernet, designed for networking, enables high-speed, long-distance communication and supports multiple devices through networking infrastructure. Its robust protocols ensure reliable data transfer in local and wide-area networks. This makes it suitable for applications requiring extensive data exchange and network connectivity.

Ideally, providing the connection for any of these last three protocols, SPI, USB, and Ethernet, would provide more flexibility for the platform. This would mean that it would be possible to choose the protocol that best fits the scenario where it is being used, making it compatible with different applications. Considering its characteristics, they can divided into two groups: short-distance, SPI, and long-distance, USB and Ethernet. As USB and Ethernet fall in the same category, and since Ethernet is more adequate for networking purposes, it was decided that for design simplification, only SPI and Ethernet would be provided.

5.1.2 Choosing the microcontroller

Several microcontroller manufacturers were considered to find the optimal balance of performance, power consumption, and peripheral availability. In addition, other key factors were considered, such as market access and community support, meaning that the search focused on more mainstream options: STM32, ESP32, PIC32, and ATmega/ATxmega.

- **STM32** series by STMicrocontrollers offers high performance, extensive peripheral set, and efficient power management, with a strong development ecosystem, ensuring a versatile and future-proof solution for the PCB design. Despite this, the price of these microcontrollers is typically higher.

- The **ESP32** from Espressif Systems, with its integrated Wi-Fi and Bluetooth capabilities, presents a strong case for IoT applications but may not offer the same processing power and peripheral diversity as other options.

- Microchip Technology's **PIC32** microcontrollers, with options based on the MIPS32 architecture and others using ARM cores, offer robust performance and a decent peripheral set. Yet, their development ecosystem is not as mature as other popular options on the market.

- **ATmega** and **ATxmega** series from Microchip Technology, former Atmel, are popular in educational and hobbyist projects due to their simplicity and ease of use. However, their 8-bit architecture and limited memory capacities make them less suitable for more demanding applications.

Ultimately, the STM32 microcontrollers were selected as the best option as they offer a range of ARM Cortex-M cores, providing high performance and the ability to choose the proper power/performance ratio for different applications. Additionally, the scalability across this microcontroller family makes upgrading or downgrading without significant redesign easy.

The STM32 ecosystem, including STM32CubeMX, HAL libraries, and strong community support, offers robust tools for development, debugging, and code generation, simplifying the development process.

In addition to this versatile development platform, many Arduino libraries have been ported to the STM32 microcontroller family, making this MCU option even more accessible when developing software for the communications module. Besides that, STM32 chips have also been integrated into PlatformIO [39], which is a very powerful and user-friendly tool that has been gaining popularity and has been used in many similar projects [40], [41].

The choice of the MCU was made with the goal of satisfying the general requirements defined previously. In this sense, an extensive search for microcontrollers culminated in the selection of the STM32F767ZIT6 microcontroller, which has proven to be an adequate alternative since the chip uses an ARM Cortex-M7 core at 216 MHz, with 2 Mbytes of Flash memory and 512 Kbytes of SRAM. It features I2C, SPI, and up to eight UART ports, and has support for USB and Ethernet connectivity.

Figure 5.1 shows the STM32F767ZIT6 microcontroller:

Figure 5.1: STM32F767ZIT6 microcontroller.

Just like the generality of the STM32 microcontrollers, the STM32F767ZIT6 provides DMA controllers. DMA stands for Direct Memory Access, which "is a bus master and system peripheral providing high-speed data transfers between peripherals and memory, as well as memory-to-memory" [42]. This allows the microcontroller to move and handle data without using other CPU resources, allowing for non-blocking CPU operation while managing data. The MCU of choice has two DMA controllers, each with eight channels, which can be used with the main peripherals, like UART and SPI. Furthermore, the chip provides dedicated DMA controllers for USB 2.0 and 10/100 Ethernet.

Another important factor that justifies why the STM32 family chips are a good option for this type of project is the fact that STMicroelectronics provides development boards, the Discovery boards [43] and the Nucleo boards [44]. These boards are affordable kits that can be used to test and evaluate the functionalities of many STM32 MCUs. On top of that, the Discovery and Nucleo boards integrate an STLink debugger/programmer. They can even be used as debuggers/programmers for other external boards, eliminating the need for an external probe. In this case, STMicroelectronics offers the Nucleo-144 F767ZI [45], which uses the STM32F767ZIT6 chip.

Figure 5.2 shows the Nucleo-144 F767ZI board:

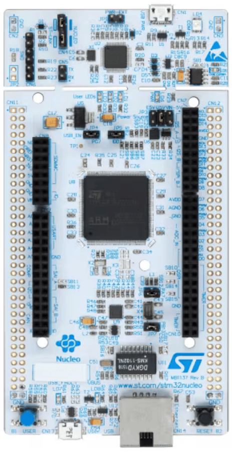

Figure 5.2: Nucleo-144 F767ZI.

By having access to an F767ZI Nucleo board, which LSTS provided, it was possible to start the implementation of software to explore and analyze the capabilities of the MCU before manufacturing the communication module. As a Master's dissertation, considering the restrictions in terms of funding and time, the availability of this evaluation kit proved to be very beneficial for the overall project. Moreover, STMicroelectronics provides open access to datasheets and schematics of the Nucleo boards [46], which served as a guide for developing the schematic of the solution for this project.

5.1.3 Serial Transceivers

From the microcontroller perspective, UART communication is handled in TTL logic levels. These levels are used to represent binary data for digital systems. However, this type of serial communication can not use TTL logic levels for communication between devices. On one hand, TTL logic levels may differ from device to device, which is not supported by many devices. On the other, it would not be reliable for long-distance communication. In this sense, the RS232, RS485, and RS422 standard protocols are typically used for this device-to-device communication. Based on this, these devices need transceivers that convert signal levels between TTL and those standards.

A typical transceiver used in many projects [47], [48] is the MAX232, a transceiver designed for the RS232 interface with support for two communication channels. This transceiver has an integrated charge pump circuitry to operate with a single 5V supply, eliminating the need for a $\pm$12V supply, which is needed in many other drivers.

Although the MAX232 type transceivers are very popular and used in many applications, simplifying the circuitry for the PCB, and can be sourced by many manufacturers, they lack compatibility with the other standard protocols, RS485 and RS422. For this project, having multi-protocol connectivity would give extra flexibility to the module, as it would not be limited to communicating with devices that use a specific protocol.

A possibility to solve the problem of supporting multiple protocols in a single board would be to have designated channels for specific protocols. As the MCU chosen provides eight UART ports, some channels would be dedicated to communicating with RS232, others with RS485, and others with RS422. However, with an extensive search for serial transceivers, it was found that there exist multi-protocol transceivers capable of handling any of the three standard protocols in a configurable way.

With that said, the choice of the transceiver relates to a simple set of parameters: multi-protocol compatibility, power supply scheme, and market availability and price. The transceiver that better fitted these specifications at the time of developing the board was the SP330EEY-L by MaxLinear [49]. This transceiver is designed to support two RS232 channels or one RS485/RS422 channel. It provides four pins for configuration that allow the selection of the protocol and trigger different operation modes: a mode that limits the data rate to minimize electromagnetic interference and a shutdown mode that reduces the chip power consumption when it is not being used. These drivers operate in up to 20 Mbps in RS485/RS422 modes and up to 1 Mbps in RS232 mode and can be powered by a single 3.3V or 5V supply.

Figure 5.3 shows the chosen serial transceiver:

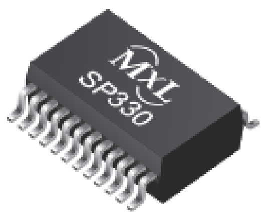

Figure 5.3: SP330EEY-L by MaxLinear.

5.1.4 Ethernet

As the STM32F767ZIT6 provides advanced connectivity for Ethernet compliant with IEEE-802.3-2002 Media Access Controller, choosing an adequate transceiver to convert the digital data to the physical layer is necessary. To simplify the development process, instead of searching for other market-available options, the transceiver of choice was the one used in the Nucleo-144 F767ZI, the LAN8742A-CZ-TR by Microchip [50]. This was only possible because this transceiver was available in the market while developing the board. This helped in terms of design because the Nucleo board would serve as guidance and also helped in terms of software, as the STM32 development platform provides a software driver to interact with this transceiver. The chip uses the Reduced Medium-Independent Interface (RMII), which is compatible with the chosen MCU and is powered by a 3.3V supply.

Figure 5.4 shows the chosen Ethernet transceiver:

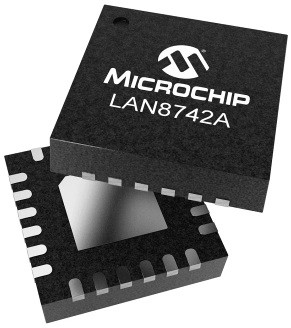

Figure 5.4: LAN8742A-CZ-TR by Microship.

5.1.5 Power Supply

The components that require power supply in the PCB are the microcontroller, the serial transceivers, and the Ethernet transceiver. Considering the options made, all those devices are supplied by 3.3V, simplifying the circuitry needed from the power supply unit perspective. In this context, it is necessary to design an efficient power supply unit that receives an external input voltage and converts it to the desired supply voltage, 3.3V, needed for the components.

For example, in the LAUV by LSTS, the power management system of the vehicle provides controlled power sources of 5V, 12V, and 24V and an uncontrolled power source, the batteries (from 21V up to 29.4V). This is a typical power structure in these kinds of vehicles. Considering this, the board could be designed to be powered by a specific power level. Nevertheless, it is relevant that the board is flexible for different scenarios. Consequently, it must support a wide range of input voltage.

By analyzing the components' datasheets provided by the manufacturers [51], [52], [53], it is possible to determine the maximum supply current that may be required by each one:

- **MCU** - 293 mA.
- **Serial Transceiver** - 4.5 mA.
- **Ethernet Transceiver** - 129 mA.

Since the board will have one microcontroller, one Ethernet transceiver, and eight serial transceivers, it is possible to determine that the power supply unit must provide at least an approximation of $(293 + 129 + 4.5 * 8)$ 458 mA. From an academic perspective, it would be interesting to design a step-down switching regulator to achieve the desired specifications, being the most power-efficient. For example, with an LT8620 step-down converter by Analog Devices [54], achieving an efficiency of up to 92% with a voltage input of 12V would be possible. Notwithstanding the substantial advantages of this alternative, it would bring some complexity to the circuit, as it requires additional passive components besides the main driver, like resistors, capacitors, and inductors.

The option mentioned before comes in a SOIC-type package, which is common in these alternatives, and other alternatives use a QFN-type package.

Other options available on the market, offer selfstanding components that achieve the desired functionality without requiring additional passive components, for example, in a SIP-3 package type, yet they tend to be less power efficient when compared to the previous solutions, as they are designed for general applications.

Figure 5.5 shows the three types of packages mentioned:

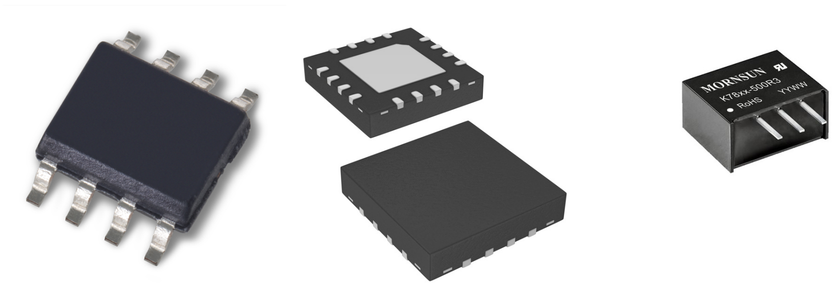

Figure 5.5: SOIC, QFN and SIP-3 package types.

The first two package types are suited to be soldered on top of the PCBs. This would be the best alternative for the power supply unit for a future version of this communication module since one of the main goals is to achieve high power efficiency. However, as it would only be possible to manufacture and assemble a single prototype during this development stage, it means that all risks that may affect the overall functionality of the PCB must be avoided.

In this type of project, during the tests of the single PCB, it is common to damage some board components, as the board is being extensively tested. If the power supply unit is damaged, the other components of the board will not work. With a power supply designed with a SOIC or QFN component, it may require additional effort to troubleshoot the problem and fix it. On the other hand, components in SIP-3-type packages offer a solution that only uses three pins, Input Voltage, Ground, and Output Voltage, in a standard and simple footprint. Moreover, since this last package type does not require additional passive components, it means that it is possible to change the power supply unit component without any changes on the PCB. The SOIC or QFN alternatives require specific layouts that change from option to option.

With that said, at the time of the development of this project, the option that better fitted the design was the TSR-1-2433 by Traco Power [55], which comes in a SIP-3 package while still being relatively power efficient, with 91% of efficiency at minimal input voltage. The TSR-1-2433 operates with an input voltage range of 4.75V up to 36V, outputting a nominal voltage of 3.3V. This component has a maximum output current of 1000 mA, which fulfills the power needs of the other components of the board.

Figure 5.6 shows the chosen power supply:

Figure 5.6: TSR-1-2433 by Traco Power.

5.1.6 Connectors for Serial Ports and SPI

Another important aspect to consider is the connector used to interface between the serial ports and the SPI peripheral to external devices. The RS232 and RS485 (half-duplex) protocols use 3 pins, and the SPI and RS422 use 5 pins. Since, a one-by-five connector would require more space to fit in the PCB layout, the decision was to use a two-by-three connector to conserve space. The choice for connector relies on two major aspects: price and robustness. Considering the specifications, at the time of the development of the board, the 430450624 connector by Molex [56] offered the best ratio for cost/efectiveness, by providing a PCB press-fit metal retention clip.

Figure 5.7 shows the chosen connector:

Figure 5.7: 430450624 connector by Molex.

5.1.7 PWM Output Signals

Furthermore, four PWM output channels were added to the design. This additional feature gives the module even more pliability, as it could be used for very different applications other than communications, such as controlling actuators through PWM signals. The STM32F767ZIT6 provides two advanced-control timers that can be used for generating PWM signals. Each timer acts as a PWM generator multiplexed on six different channels, more than enough to provide the four PWM outputs needed.

5.2 Design

All the developments concerning the design of the PCB were done using the software KiCad. Which is an open-source Electronic Design Automation (EDA) software, with the capabilities of Schematic Capture and PCB Layout with Gerber output, used not only by professionals but also suited for nonexperienced users [57]. Therefore, KiCad was chosen for this project as it proved to be the right software to develop advanced projects in a free and easy-to-use package [58].

5.2.1 Schematic

The schematics for the implementation and arrangement of the circuit components will be explored throughout this section.

Power

The power supply unit of the device is composed of the TSR-1-2433 step-down regulator (U1), a one-by-two connector for the input voltage (J1), a fuse (F1), and a TVS diode (D1). The TVS diode protects the circuit against transients and overvoltage threats. And also makes the circuit reverse polarity protected: in the event of connecting the input voltage with inverse polarity, the TVS diode will act as a short circuit, breaking the Fuse, opening the circuit. Additional test points (TP1, TP2, TP3) were added to help with troubleshooting in the future.

Figure 5.8 shows the schematic for the Power Supply part.

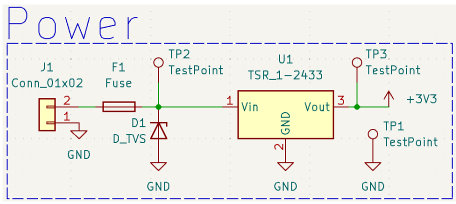

Figure 5.8: Schematic for the Power Supply part.

MCU

The MCU part of the schematic features the STM32F767ZIT6 microcontroller (U2) with all the connections to other components that it interacts with and an 8 MHz external oscillator (Y1), to achieve a more stable and accurate frequency. The power supply scheme for the chip follows the guidelines provided in the datasheet given by the manufacturer [51]. The chip has an internal power supply supervisor, triggered by setting the PDR_ON pin (143) at high, 3.3V, eliminating the need for an external component. The one-by-five connector (J2) provides connectivity for the STLink debugger/programmer using the Serial Wire Debug, SWD, standard. Additionally, eight LEDs were added to provide feedback about the serial transceivers (D2-D9) and a reset switch (SW1) for convenience.

Apart from the power connections and general pinout arrangement configured as specified by the datasheet guidelines, the rest of the pin assignment was done according to the microcontroller peripheral configuration. Using the STM32CubeMX software provided by STMicroelectronics [59] it was possible to configure the MCU, assigning to each pin its functionality.

Figure 5.9 shows the configuration done using the STMCubeMX software.

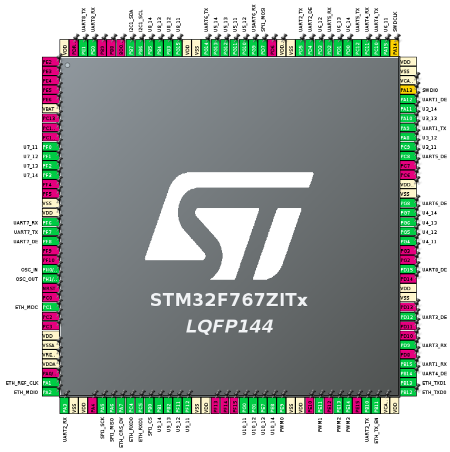

Figure 5.9: Configuration of the MCU using the STMCubeMX software.

Figure 5.10 shows the schematic for the MCU part.

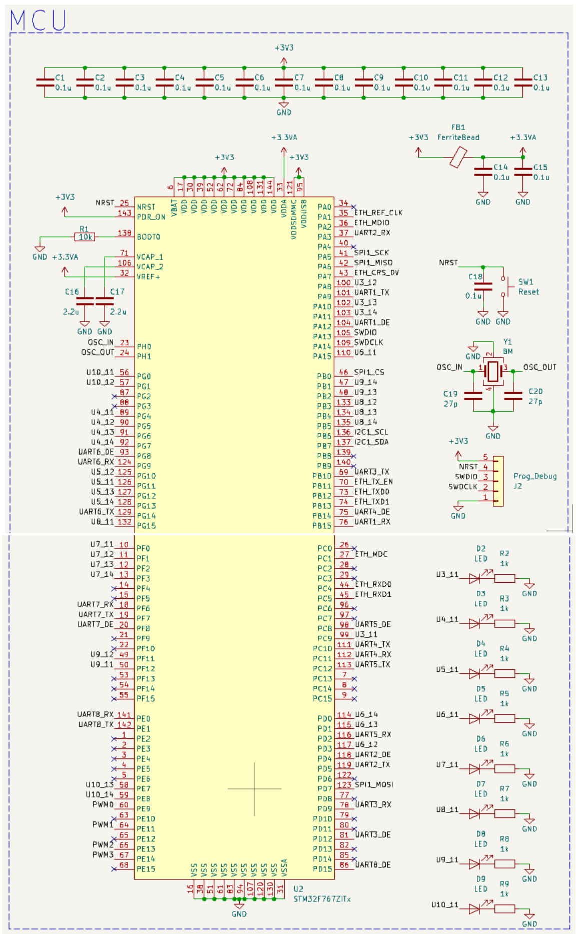

Figure 5.10: Schematic for the MCU part.

Serial

The Serial part of the schematic features the eight SP330EEY-L transceivers (U3-U10) with a power supply scheme according to the manufacturer datasheet [52] and all the connections required for its desired operation. Each driver is connected to a UART channel of the MCU, with three connections: transmission pin (UARTx_TX), reception pin (UARTx_RX) and driver enable pin (UARTx_DE) required for the RS485 protocol. The eight two-by-three connectors (J3-J10) provide connectivity between the transceivers and the external devices. The SP330EEY-L can be configured through the 11, 12, 13, and 14 pins: initially, the idea was to provide physical configuration of the drivers with jumpers that would be set by the user, however, since the MCU provided enough GPIO pins, without compromising other peripherals, the decision was to connect each group of four pins to the MCU so that they can be digitally controlled.

Figure 5.11 shows the schematic for the Serial part.

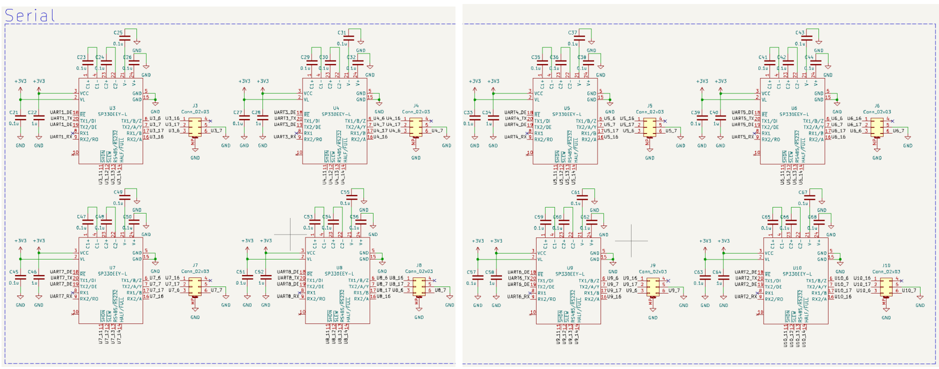

Figure 5.11: Schematic for the Serial part.

Ethernet

The Ethernet part of the schematic features the LAN8742A-CZ-TR transceiver (U11) with a power supply scheme and an external 25 MHz oscillator (Y2) as recommended by the manufacturer [53]. The connections between the MCU and driver follow the implementation of the Nucleo-144 F767ZI board schematic [46] and the manufacturer datasheet. There is also an RJ45 connector with integrated Magnetics (J11) required for interface with external devices and an ESD protection buffer (U12). Since these components can consume a significant amount of power, even in scenarios where they are not being used, it was suggested during the design review at MBARI to add the possibility of cutting off their power supply. With that in mind, the supply of 3.3V to this part is bridged using a jumper (JP1) and in a scenario where Ethernet is not being used, it is possible to physically remove the jumper, reducing the overall power consumption of the module.

Figure 5.12 shows the schematic for the Ethernet part.

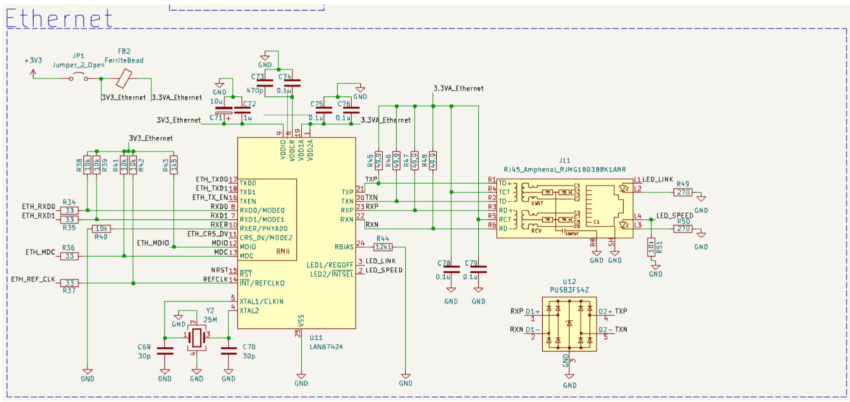

Figure 5.12: Shematic for the Ethernet part.

Additionally, this part of the schematic requires a high level of attention and expertise since it uses high-speed transmission lines with differential pairs. Therefore, it was not guaranteed that this part of the circuit would work as desired, in the final prototype. Isolating the supply of power to this part from the rest of the board, with a physical jumper, means that in the event of a design error, the rest of the board would not be compromised and the other features could still be tested.

PWM

The PWM part of the schematic is composed of four one-by-three connectors (J14-J17) to output the signal provided by the MCU. Supplementary Schmitt-trigger buffers (U13-U16) were added to generate clean PWM signals.

Figure 5.13 shows the schematic for the PWM part.

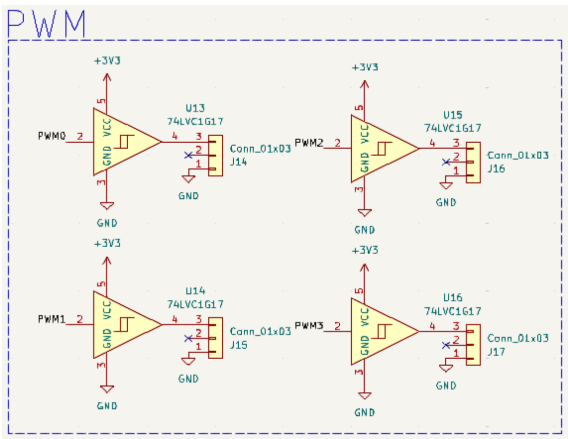

Figure 5.13: Schematic for the PWM part.

SPI and I2C

The SPI part is composed of a single two-by-three connector (J12) providing the connection for one SPI peripheral of the MCU. And although it was not a requirement of the project, a one-by-three connector (J13) was added to provide a connection for one I2C peripheral of the MCU, with the required pullup resistors. This last connection enables the board to be connected to external devices which communicate through the I2C protocol.

Figure 5.14 shows the schematic for the SPI and I2C parts.

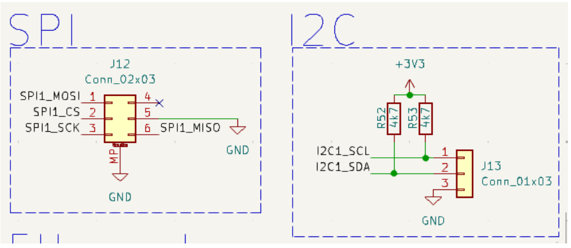

Figure 5.14: Schematic for the SPI and I2C parts.

5.2.2 PCB

The design of the PCB layout followed the general PCB design guidelines [60] and the next set of considerations:

- Use a four-layer setup with a layer configuration of Signal-GND-Power-Signal. Although this board will not need to handle high power, using this type of layer stackup will simplify power connections, while still allowing high-speed signals like Ethernet on the top layer of the board [61].

- Guarantee a 100 Ohm impedance on the Ethernet differential pairs. There are theoretical ways to determine the tracks' specifications that allow for a specific differential impedance [62], with calculators available on the Internet that apply those theoretical guidelines to build the adequate stackup. "The reality is that none of these calculators or guidelines are very useful if you don't have a manufacturable PCB stackup" [63]. In that sense, a common procedure is to request from the PCB manufacturer the correct parameters (track width and length, layer thickness, etc.), that provide the desired functionalities.

- In the internal architecture of the LAUV, the vehicle has two parallel rails where the electronics are mounted. The hole-to-hole distance of this frame is 123mm. To fit inside this setup, the module needs to comply with these dimensions.

- Avoid having adjacent transmission lines, to reduce interference, which may be difficult, due to space limitations.

- Simplify the connection between the UART transceivers and the MCU. As these components are connected to a large number of pins from the MCU, it will be necessary to meticulously define which driver is connected to which UART port and GPIO pins.

- Separate the Ethernet part of the layout from the UART transmission lines, as it is more sensitive to noise caused by other transmission lines.

Figure 5.15 shows the resulting PCB layout:

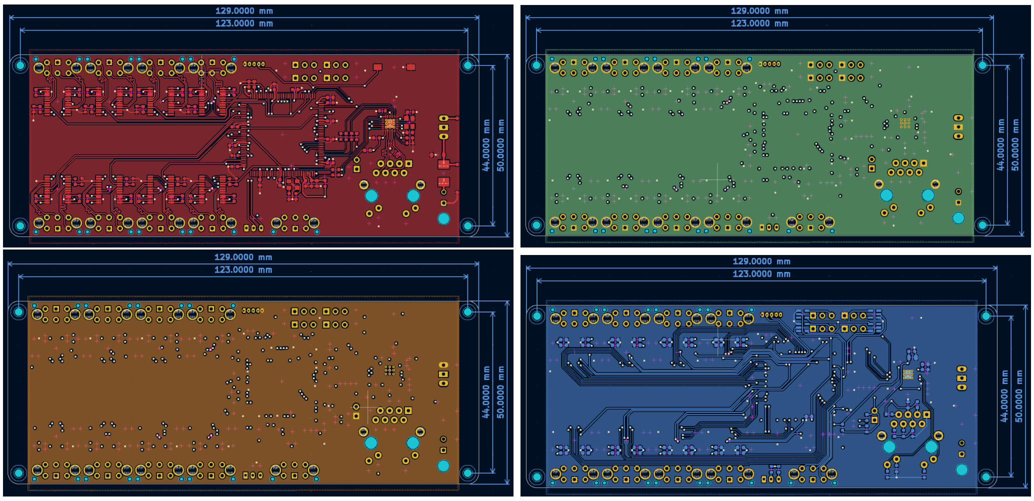

Figure 5.15: PCB layout.

Figure 5.15 shows a 3D render of PCB generated by Kicad:

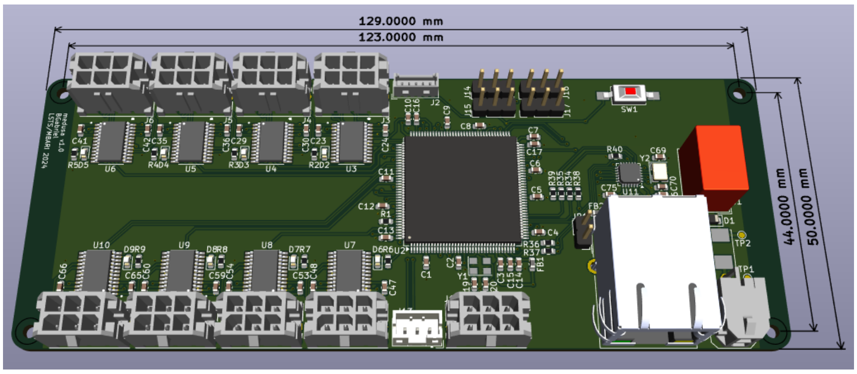

Figure 5.16: 3D render of the PCB.

5.2.3 Assembly of the first prototype

With the design completed, the Build Of Materials (BOM) was defined, the components were ordered, and the PCB was sent to fabrication. MBARI supported the costs of the components and fabrication. Resulting in the assembling of the prototype, shown in figure 5.17.

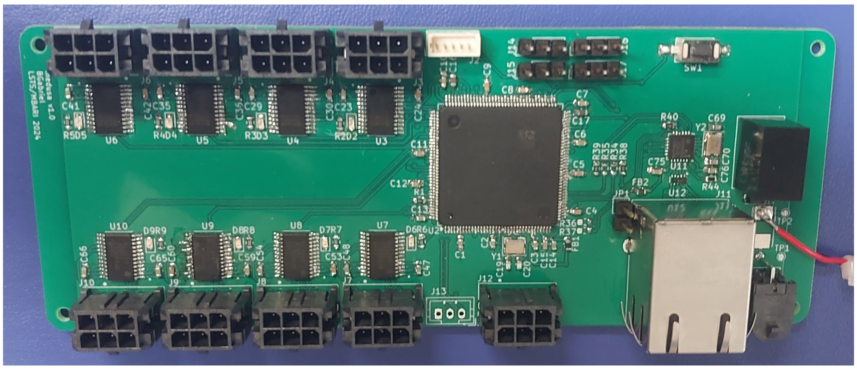

Figure 5.17: Assembeled Prototype.

Overall, there were only two errors in the final layout, discovered during the assembly: the footprints of the Schmitt-trigger buffers for the PWM and of the Fuse did not match the footprint on the PCB layout. In terms of the buffers, it was still possible to fit them in the incorrect footprint, which did not happen for the Fuse. Without the presence of the Fuse, extra care was needed to handle the prototype.

Chapter 6

Results

The present chapter will focus on evaluating the prototype's general performance and its capabilities. With that in mind, a set of tests was defined for each board peripheral to analyze their behavior. Moreover, the board's power consumption was measured for different scenarios, for example, for different values of MCU frequency, for different peripherals enabled, and using different middleware. This chapter concludes with the identification of possible applications for the solution developed.

6.1 Verification and Validation

To test the functionalities of the board, the next set of tests was defined, shown in table 6.1. Tests identified by G are related to the general functionalities of the board, U to the serial ports, E to Ethernet, S with the SPI peripheral, and P with the PWM channels.

Table 6.1: Set of tests for functionalities verification.

Identifier	Description
G1	Power the board and test the microcontroller
G2	Test different values of input voltage
U1	Test the serial ports using RS232, RS485 and RS422
U2	Test the serial ports using different baud rates
E1	Test the Ethernet transceiver connection
E2	Test data exchange through Ethernet
S1	Test data exchange through SPI
P1	Test each PWM channel

6.1.1 General Board Tests

Considering **G1**, the board was powered by 5 V, and using the Nucleo board as a debugger/programmer, the connection to MCU was tested. With the STlink GUI tool provided by STMicroelectronics, it was possible to connect to the chip.

Figure 6.1 shows that the software successfully identified the MCU:

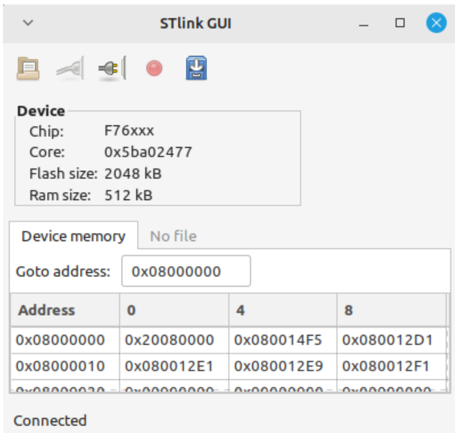

Figure 6.1: STlink GUI connection status.

For **G2**, using a DC voltage power supply, the board was powered with a range of input voltage from 5 V up to 30 V. With a digital multimeter, the voltage on the test points was measured, achieving the desired results. Since the prototype was not assembled with the necessary fuse, no tests were done to verify the protection against inverse polarity.

6.1.2 Serial ports

A simple software was implemented using the Arduino libraries to test the board's serial ports. Firstly, the ports were configured to send a short "Hello World" message every 1 second. Secondly, they were configured to listen for incoming messages, and in the event of the port receiving the message "Test", it would need to reply with the message "OK". In both cases, each port was configured for communicating over the different protocols with a baud rate of 9600 bits/s, to test **U1**, and to test **U2** the ports were configured for RS232 with baud rates of 9600, 19200, 115200 bits/s. The ports were connected to a computer using USB to RS232 and USB to RS485/RS422 converters.

With these tests, it was possible to identify one port that could not send any data and another port that could not receive any data. After troubleshooting both situations, it was spotted that the faulty ports were wrongly soldered to the PCB, resulting in these issues. Apart from that, the other six ports succeeded in all the tests.

6.1.3 Ethernet

To test the Ethernet, the STM32CubeIDE was used for code generation, as it provides software packs that handle the connection to the Ethernet transceiver, using LwIP, and as tools for breakpoint debugging of the software execution. By debugging the code execution, it was possible to identify that the MCU successfully interacted with the transceiver, completing test **E1**.

A more complex setup was needed for test **E2**. Using the STM32CubeIDE, it is possible to generate code configuring the board to use the Ethernet with a static IP to exchange data with an external device.

Figure 6.2 shows the configuration done for the board static IP:

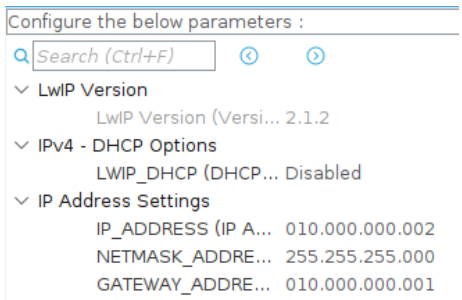

Figure 6.2: Static IP configuration for the module, using STM32CubeIDE.

For this purpose, the board was connected to a computer with an Ethernet cable. The computer, running the Linux Mint operating system, was configured to interact with the board using the defined static IP. Using the Linux **netcat** utility, it was possible to create a TCP socket that the devices could use for communicating. The board successfully connected to the socket and was able to send the message "Hello World". Additionally, using the debugging tool, it was possible to identify that the board also received messages sent by the computer.

6.1.4 SPI

Ideally, to test the SPI peripheral, it would be interesting to use an external device to establish communication. However, it was not possible to have this setup, and the SPI was tested by exchanging data with itself. By connecting the MISO and MOSI pins in the SPI connector, every time the MCU sent a message, it would receive the same message, completing test **S1**.

6.1.5 PWM

The PWM channels were tested using Arduino libraries for ease of implementation. In addition, the board was configured to use one of the serial ports, with RS232, to interpret commands that would control the value of the PWM signal. This test consisted of sending a special command through the computer connected to the serial port using a USB to RS232 converter. Then, the MCU would interpret the command and change the PWM signal accordingly. It was possible to visualize the correct format of the square waves generated by each PWM channel using an oscilloscope, completing test **P1**.

6.2 Power Consumption

Some measurements were done to evaluate the power consumption of the prototype. In this sense, a set of scenarios was defined to evaluate different applications of the board. The results are shown in table 6.2:

Table 6.2: Power consumption measurements.

Scenario	Voltage (V)	Current (A)	Power (W)
MCU at 216 MHz all peripherals activated	5.0	0.195	0.975
MCU at 108 MHz all peripherals activated	5.0	0.112	0.560
MCU at 108 MHz all peripherals activated except Ethernet	5.0	0.040	0.200
MCU at 54 MHz all peripherals activated except Ethernet	5.0	0.029	0.145

For the first two scenarios, the STM32CubeIDE generated code used LwIP [64] and FreeRTOS [65] software packs recommended for the Ethernet peripheral, and for the last two scenarios, in addition to disabling the Ethernet peripheral, the jumper that connects the transceiver to power was removed.

Although these results prove the module's flexibility and low power consumption, few conclusions can be drawn from them. The power consumption of the final product will also depend on the software the MCU will be using, and the tests were only done at room temperature, which does not translate many possible scenarios. The topic concerning temperature is relevant because the interior of AUVs is isolated, and the temperatures may rise significantly, affecting the module's power consumption.

Nevertheless, the results satisfy the requirement R4, defined for the design, because the module consumed under 1 W of power in the first scenario, even with all the peripherals activated and the MCU running at maximum frequency. Additionally, these tests proved that lowering the MCU frequency would decrease the board's overall power consumption, which is beneficial in less demanding scenarios.

6.3 Example of Applications

This board's flexibility makes it applicable in many different scenarios. Designing an efficient software that handles serial communication using FIFO buffers to redirect serial data from eight serial devices to a host unit and vice versa would achieve the desired goal of expanding any main unit is possible. The design gives the option of choosing either the SPI bus for interface in more power-restricted scenarios or the Ethernet port, which may benefit from its networking characteristics.

Furthermore, just like in other works, where microcontroller platforms have been used as the central control unit for AUVs [66], the high-performance capabilities of the chosen MCU could enable the possibility to lower significantly the consumption of the system since the power consumption of the module is below 1 W, with capabilities of reaching even lower power consumptions.

Moreover, considering LSTS's toolchain, integrating the IMC protocol into this device would allow for more standardized communication between the main unit and the other modules. This board could be used as an interface between CPU and payloads, simplifying their interaction by receiving high-level IMC messages and translating them into low-level commands for different payloads.

6.3.1 Demonstration of the four PWM channels controller over RS232

MBARI's LRAUV team is currently exploring the benefits of integrating hovering capabilities into a long-endurance AUV. Their approach was to attach eight thrusters, four on the front part of the vehicle and four near the tail, aligned to provide heave and sway motions to the vehicle. Each group of four thrusters would act as an external module controlled through RS232.

Figure 6.3 shows the LRAUV with the theorized hovering kit.

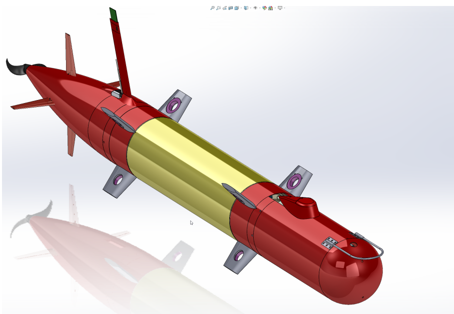

Figure 6.3: 3D render of the LRAUV with the hovering kit.

Before designing the optimal solution for these requirements, the communications module developed in this project was used to evaluate how feasible this setup would be. Although the board provides superfluous features, its flexibility allows for a quick implementation of a setup to test this concept.

The board was mounted inside a waterproof case with four ESCs to actuate the four thrusters, controlled through PWM. The case had a connector that would provide power for the board and thrusters/ESCs and an RS232 connection.

Figure 6.4 shows the assembly, with the board and ESCs inside the case.

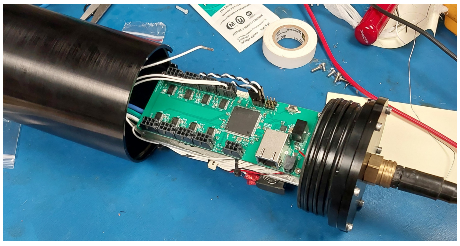

Figure 6.4: The board integrated inside the waterproof case with the 4 ESCs.

Just like in the tests for **P1**, the board would be connected to a computer that sends commands over the serial port to control the PWM channels. The MCU was configured to operate at 0.5 MHz, enough to use the serial port at a baud rate of 9600 bits/s and generate 50 Hz PWM signals. The board was tested at 16 V input voltage in this configuration, consuming 0.007 A, totaling 0.11 W power consumption.

Figure 6.5 shows the LRAUV setup with one hovering module consisting of one case and four thrusters mounted in the central part of the vehicle.

Figure 6.5: LRAUV Setup with the Prototype Hovering Kit.

A Python script was implemented to facilitate testing this setup's hovering capabilities. The script would connect to the serial port and send the control command at a refresh rate of ten messages per second. It provided an interface with a virtual joystick, and the control command was calculated according to its position.

Chapter 7

Conclusion

In conclusion, it is possible to state that this work helped identify the current limitations of the popular solutions for the hardware architectures of AUVs. In this sense, it provides an overview of how these challenges are tackled in different research fields and presents a new perspective on handling them.

With the research done, it was clear that addressing the communication problem in an external module gives more abstraction to the final solution.

However, it is still uncertain how effective this proposal is. This dissertation focused on aggregating good practices from many other works, providing an easy-to-use platform that can be configured for various applications. The module's performance will depend on the software implemented, which was not a goal of this project. In addition to this, since the board was designed for general applications, some could say that it lacks optimization for specific scenarios. Although this is true, this work aimed to define a better approach than the previous ones, and the design optimization will be done according to future implementation.

On the other hand, the setup designed to test the hovering capabilities of the LRAUV proved how easy it is to develop an application using this board. The STM32CubeIDE development environment gives the necessary tools to easily develop the code, with a large community supporting these implementations.

In short, the key achievements of the proposed solution are:

- Enabling the communication with eight serial devices, with compatibility for multiple protocols (RS232, RS485, and RS422).

- The possibility of using a high-speed communication medium, Ethernet.

- It provides a programmable platform for different applications, with support from a great development community.

- It is compatible with any main unit that has SPI or Ethernet connectivity.

7.1 Future Work

There are some considerations for future work:

- Design a more efficient Power Supply Unit. As stated before, many other options allow for more efficient power conversion.
- Now that the design has been tested and validated, make a revision of the board with a more compact layout. The current version of the board does not utilize the space in a very efficient way.
- Implement the software for the board and evaluate its performance. Define the specifications for the intended application and start working on software that better exploits the characteristics of the module.

Bibliography

[1] LSTS, *Highlights | Laboratório de Sistemas e Tecnologia Subaquática*. [Online]. Available: https://www.lsts.pt/.

[2] LSTS, *GitHub - LSTS/dune: DUNE: Unified Navigation Environment*. [Online]. Available: https://github.com/LSTS/dune/ (visited on 07/31/2024).

[3] LSTS, *GitHub - LSTS/neptus: Neptus, Command and Control Infrastructure*. [Online]. Available: https://github.com/LSTS/neptus/ (visited on 07/31/2024).

[4] LSTS, *GitHub - LSTS/imc: IMC: Intermodule Communication API*. [Online]. Available: https://github.com/LSTS/imc/ (visited on 07/31/2024).

[5] LSTS, *Ripples*. [Online]. Available: https://ripples.lsts.pt/ (visited on 07/31/2024).

[6] P. Bandyopadhyay, "Trends in biorobotic autonomous undersea vehicles," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 1, pp. 109–139, Jan. 2005, ISSN: 1558-1691. DOI: 10.1109/JOE.2005.843748. [Online]. Available: https://ieeexplore.ieee.org/document/1435581 (visited on 07/28/2024).

[7] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV Control and Adaptive Sampling in Monterey Bay," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 935–948, Oct. 2006, ISSN: 1558-1691. DOI: 10.1109/JOE.2006.880429. [Online]. Available: https://ieeexplore.ieee.org/document/4089061 (visited on 07/28/2024).

[8] N. Davari, A. P. Aguiar, and J. B. d. Sousa, "An AUV Navigation System Using an Adaptive Error State Kalman Filter Based on Variational Bayesian," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, ISSN: 2377-6536, Nov. 2018, pp. 1–6. DOI: 10.1109/AUV.2018.8729756. [Online]. Available: https://ieeexplore.ieee.org/document/8729756 (visited on 07/15/2024).

[9] J. Guo, D. Li, B. He, and S. S. Ge, "An Intelligent Collaborative System for Robot Dynamics," *IEEE Transactions on Cybernetics*, vol. 53, no. 10, pp. 6211–6221, Oct. 2023, ISSN: 2168-2275. DOI: 10.1109/TCYB.2022.3163440. [Online]. Available: https://ieeexplore.ieee.org/document/9760281 (visited on 07/15/2024).

[10] K. Valavanis, D. Gracanin, M. Matijasevic, R. Kolluru, and G. Demetriou, "Control architectures for autonomous underwater vehicles," *IEEE Control Systems Magazine*, vol. 17, no. 6, pp. 48–64, Dec. 1997, ISSN: 1941-000X. DOI: 10.1109/37.642974. [Online]. Available: https://ieeexplore.ieee.org/document/642974 (visited on 05/21/2024).

[11] M. J. Stanway, B. Kieft, T. Hoover, *et al.*, "White shark strike on a long-range AUV in Monterey Bay," in *OCEANS 2015 - Genova*, May 2015, pp. 1–7. DOI: 10.1109/OCEANS-Genova.2015.7271726. [Online]. Available: https://ieeexplore.ieee.org/document/7271726 (visited on 07/16/2024).

[12] S. McPhail, R. Templeton, M. Pebody, D. Roper, and R. Morrison, "Autosub Long Range AUV Missions Under the Filchner and Ronne Ice Shelves in the Weddell Sea, Antarctica - an Engineering Perspective," in *OCEANS 2019 - Marseille*, Jun. 2019, pp. 1–8. DOI: 10.1109/OCEANSE.2019.8867206. [Online]. Available: https://ieeexplore.ieee.org/document/8867206 (visited on 07/28/2024).

[13] S. M. Schillai, "Pipefish AUV: The flight style AUV delivering small, purpose built, hover capable AUVs," in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, ISSN: 2377-6536, Nov. 2016, pp. 20–27. DOI: 10.1109/AUV.2016.7778714. [Online]. Available: https://ieeexplore.ieee.org/document/7778714 (visited on 07/28/2024).

[14] T. Liu, Z. Jiang, S. Li, and H. Gu, "Explorer1000: A Long Endurance AUV with Variable Ballast Systems," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, May 2018, pp. 1–6. DOI: 10.1109/OCEANSKOBE.2018.8559249. [Online]. Available: https://ieeexplore.ieee.org/document/8559249 (visited on 07/28/2024).

[15] J. Qiao, J. Yu, Y. Huang, J. Cui, B. Wang, and Z. Wang, "Sea-Whale Series AUV-Extending the range to 4000 kilometers," in *OCEANS 2022, Hampton Roads*, ISSN: 0197-7385, Oct. 2022, pp. 1–5. DOI: 10.1109/OCEANS47191.2022.9977249. [Online]. Available: https://ieeexplore.ieee.org/document/9977249 (visited on 07/28/2024).

[16] L. Madureira, A. Sousa, J. Braga, *et al.*, "The light autonomous underwater vehicle: Evolutions and networking," in *2013 MTS/IEEE OCEANS - Bergen*, Jun. 2013, pp. 1–6. DOI: 10.1109/OCEANS-Bergen.2013.6608189. [Online]. Available: https://ieeexplore.ieee.org/document/6608189 (visited on 05/15/2024).

[17] Diamond Systems, *PC/104 Single Board Computer with 500MHz Geode LX800 CPU*. [Online]. Available: https://www.diamondsystems.com/products/rhodeus (visited on 07/29/2024).

[18] Raspberry Pi, *Raspberry Pi 4 Model B specifications – Raspberry Pi*. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/ (visited on 07/29/2024).

[19] L3Harris, *Iver4 900 AUV | L3Harris® Fast. Forward.* en. [Online]. Available: https://www.l3harris.com/all-capabilities/iver4-900-auv (visited on 07/29/2024).

[20] J. Boye, "RADSoM - a System on Module Approach for Reusable Spacecraft Processing Elements," in *2020 IEEE Aerospace Conference*, ISSN: 1095-323X, Mar. 2020, pp. 1–8. DOI: 10.1109/AERO47225.2020.9172642. [Online]. Available: https://ieeexplore.ieee.org/document/9172642 (visited on 07/13/2024).

[21] MBARI, *Long Range AUV (LRAUV) • MBARI*, en-US. [Online]. Available: https://www.mbari.org/technology/long-range-auv-lrauv/ (visited on 07/16/2024).

[22] B. W. Hobson, J. G. Bellingham, B. Kieft, R. McEwen, M. Godin, and Y. Zhang, "Tethys-class long range AUVs - extending the endurance of propeller-driven cruising AUVs from days to weeks," in *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, ISSN: 2377-6536, Sep. 2012, pp. 1–8. DOI: 10.1109/AUV.2012.6380735. [Online]. Available: https://ieeexplore.ieee.org/document/6380735 (visited on 05/15/2024).

[23] Connect Tech Inc., *Xtreme/PCI-104 Opto - 12 Port*, en-US. [Online]. Available: https://connecttech.com/product/xtremepci-104-opto-12-port/ (visited on 07/29/2024).

[24] I. I. Corp, *MPCIE-UART-KIT01 :: IEI*, en. [Online]. Available: https://www.ieiworld.com/en/product/model.php?II=147 (visited on 07/29/2024).

[25] FTDI, *USB-COM232-PLUS4 - FTDI*, en-US, Aug. 2020. [Online]. Available: https://ftdichip.com/products/usb-com232-plus4/ (visited on 07/29/2024).

[26] Brainboxes, *ES-279 8 Port RS232 Ethernet to Serial Adapter - Brainboxes*, en-US, Apr. 2024. [Online]. Available: https://www.brainboxes.com/product/ethernet-to-serial/db9/es-279 (visited on 07/29/2024).

[27] R. Falcão, "BALU: A Modular Computational Framework for Maritime Vehicle Systems," English, M.S. thesis, Faculty of Engineering of University of Porto, Porto, Jul. 2022.

[28] A. Ang and R. Dykstra, "An Open Source PXIe Platform for Instrumentation Development," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, ISSN: 2642-2077, May 2019, pp. 1–5. DOI: 10.1109/I2MTC.2019.8827024. [Online]. Available: https://ieeexplore.ieee.org/document/8827024 (visited on 07/30/2024).

[29] L. Zhaoqing, L. Naihai, and P. Xiyuan, "A Design of Multiprotocol Asynchronous Serial Communication Based on M Module," in *2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, Oct. 2011, pp. 377–380. DOI: 10.1109/IMCCC.2011.101. [Online]. Available: https://ieeexplore.ieee.org/document/6154079 (visited on 07/15/2024).

[30] V. K. Abdrakhmanov, R. B. Salikhov, and K. V. Vazhdacv, “Development of a Sound Recognition System Using STM32 Microcontrollers for Monitoring the State of Biological Objects,” in *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, ISSN: 2473-8573, Oct. 2018, pp. 170–173. DOI: 10.1109/APEIE.2018.8545278. [Online]. Available: https://ieeexplore.ieee.org/document/8545278 (visited on 07/17/2024).

[31] H. Wang, Z. jie, X. Nie, and D. Li, “Design of PIC microcontroller-based high-capacity multi-channel data acquisition module,” in *Proceedings of 2012 International Conference on Measurement, Information and Control*, vol. 2, May 2012, pp. 685–688. DOI: 10.1109/MIC.2012.6273385. [Online]. Available: https://ieeexplore.ieee.org/document/6273385 (visited on 07/26/2024).

[32] S. Nuratch, “Design and Implementation of Real-time Embedded Data Acquisition and Classification with Web-based Configuration and Visualization,” in *2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES)*, May 2018, pp. 1–4. DOI: 10.1109/ICESIT-ICICTES.2018.8442071. [Online]. Available: https://ieeexplore.ieee.org/document/8442071 (visited on 07/26/2024).

[33] S. Nuratch, “Firmware design and development of MCU and Matlab-Simulink interfacing for real-time measurement, analysis and control applications,” in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Jun. 2016, pp. 1–6. DOI: 10.1109/ECTICon.2016.7561476. [Online]. Available: https://ieeexplore.ieee.org/document/7561476 (visited on 07/26/2024).

[34] M. John, A. Joseph, and D. Vasanthi, “Design and implementation of low cost multichannel data acquisition hardware for computer aided measurement,” in *2017 Trends in Industrial Measurement and Automation (TIMA)*, Jan. 2017, pp. 1–4. DOI: 10.1109/TIMA.2017.8064805. [Online]. Available: https://ieeexplore.ieee.org/document/8064805 (visited on 07/26/2024).

[35] H. Wang and W. Cai, “Research on Marine Data Collection and Transmission System Based on Lwip Protocol,” in *2012 International Conference on Control Engineering and Communication Technology*, Dec. 2012, pp. 346–349. DOI: 10.1109/ICCECT.2012.203. [Online]. Available: https://ieeexplore.ieee.org/document/6414089 (visited on 03/06/2024).

[36] Z. Yunzhou and W. Chengdong, “Design of a trainning and experimental platform based on multi-protocol communication,” in *2011 International Conference on E-Business and E-Government (ICEE)*, May 2011, pp. 1–4. DOI: 10.1109/ICEBEG.2011.5887065. [Online]. Available: https://ieeexplore.ieee.org/document/5887065 (visited on 03/06/2024).

[37] G. Sun, J. Morgan, J. Porter, and H. Yim, "Development of a novel Modular Integrated Stackable Layer — Analog System Environment (MISL — ASE) platform for embedded systems education," in *2015 IEEE Frontiers in Education Conference (FIE)*, Oct. 2015, pp. 1–5. DOI: 10.1109/FIE.2015.7344262. [Online]. Available: https://ieeexplore.ieee.org/document/7344262 (visited on 07/23/2024).

[38] S. S and K. R.K, "The design of Multiprotocol Interface Device," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, Jul. 2016, pp. 474–476. DOI: 10.1109/ICATCCT.2016.7912046. [Online]. Available: https://ieeexplore.ieee.org/document/7912046 (visited on 07/23/2024).

[39] PlatformIO, *PlatformIO: Your Gateway to Embedded Software Development Excellence*, en. [Online]. Available: https://platformio.org (visited on 07/23/2024).

[40] M. S. Abdul, S. M. Sam, N. Mohamed, N. H. Hassan, A. Azizan, and Y. M. Yusof, "Peer to Peer Communication for the Internet of Things Using ESP32 Microcontroller for Indoor Environments," in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, ISSN: 2162-1241, Oct. 2022, pp. 1–6. DOI: 10.1109/ICTC55196.2022.9952832. [Online]. Available: https://ieeexplore.ieee.org/document/9952832 (visited on 07/23/2024).

[41] N. Krueganta, O. Chieochan, A. Saokaew, *et al.*, "The Low-Cost IoT Mornitoring the Quality of River: A Case Study of Wang River in Lampang Municipality," in *2023 27th International Computer Science and Engineering Conference (ICSEC)*, ISSN: 2768-0592, Sep. 2023, pp. 178–183. DOI: 10.1109/ICSEC59635.2023.10329725. [Online]. Available: https://ieeexplore.ieee.org/document/10329725 (visited on 07/23/2024).

[42] STMicroelectronics, *Getting started with DMA - stm32mcu*, en. [Online]. Available: https://wiki.st.com/stm32mcu/wiki/Getting_started_with_DMA (visited on 07/24/2024).

[43] STMicroelectronics, *STM32 Discovery Boards - STMicroelectronics*. [Online]. Available: https://www.st.com/en/evaluation-tools/stm32-discovery-kits.html (visited on 07/24/2024).

[44] STMicroelectronics, *STM32 Nucleo Boards - STMicroelectronics*. [Online]. Available: https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html (visited on 07/24/2024).

[45] STMicroelectronics, *NUCLEO-F767ZI - STM32 Nucleo-144 development board with STM32F767ZI MCU, supports Arduino, ST Zio and morpho connectivity*. [Online]. Available: https://www.st.com/en/evaluation-tools/nucleo-f767zi.html (visited on 07/24/2024).

[46] STMicroelectronics, *NUCLEO-F767ZI - STM32 Nucleo-144 development board with STM32F767ZI MCU, supports Arduino, ST Zio and morpho connectivity - CAD Resources*. [Online]. Available: https://www.st.com/en/evaluation-tools/nucleo-f767zi.html#cad-resources (visited on 07/24/2024).

[47] P. Ran, B. Wang, and W. Wang, "The design of communication convertor based on CAN bus," in *2008 IEEE International Conference on Industrial Technology*, Apr. 2008, pp. 1–5. DOI: 10.1109/ICIT.2008.4608607. [Online]. Available: https://ieeexplore.ieee.org/document/4608607 (visited on 07/24/2024).

[48] O. V. Biryukova and I. V. Koretskaya, "Using an Ultrasonic Sensor to Monitor Position, Transfer Control Commands and Work Information," in *2019 Systems of Signals Generating and Processing in the Field of on Board Communications*, Mar. 2019, pp. 1–5. DOI: 10.1109/SOSG.2019.8706796. [Online]. Available: https://ieeexplore.ieee.org/document/8706796 (visited on 07/24/2024).

[49] MaxLinear, *SP330 - MaxLinear*. [Online]. Available: https://www.maxlinear.com/product/interface/serial-transceivers/multiprotocol/dual-protocol/sp330 (visited on 07/24/2024).

[50] Microship, *LAN8742A | Microchip Technology*. [Online]. Available: https://www.microchip.com/en-us/product/LAN8742A (visited on 07/24/2024).

[51] STMicroelectronics, *Datasheet - STM32F765xx STM32F767xx STM32F768Ax STM32F769xx*. [Online]. Available: https://eu.mouser.com/datasheet/2/389/stm32f765bi-2956130.pdf (visited on 07/24/2024).

[52] MaxLinear, *Datasheet - SP330E*. [Online]. Available: https://eu.mouser.com/datasheet/2/146/sp330e-1825361.pdf (visited on 07/24/2024).

[53] Microship, *Datasheet - LAN8742A/LAN8742Ai*. [Online]. Available: https://eu.mouser.com/datasheet/2/268/DS_LAN8742_00001989A-3443850.pdf (visited on 07/24/2024).

[54] Analog Devices, *LT8620 Datasheet and Product Info | Analog Devices*. [Online]. Available: https://www.analog.com/en/products/lt8620.html#part-details (visited on 07/24/2024).

[55] Traco Power, *TSR 1-2433 | Traco Power*. [Online]. Available: https://www.tracopower.com/model/tsr-1-2433 (visited on 07/24/2024).

[56] Molex, *PCB Headers and Receptacles Part - 430450624 | Molex*. [Online]. Available: https://www.molex.com/en-us/products/part-detail/430450624 (visited on 07/29/2024).

[57] "Kicad eda - schematic capture & pcb design software." Accessed: 2024-01-05. (), [Online]. Available: https://www.kicad.org/.

[58] A. A. Gautam and V. Laxmi, “Gate Drive for Power Electronic Converters : An Insight into KiCAD’s PCB design !” In *2021 2nd Global Conference for Advancement in Technology (GCAT)*, Oct. 2021, pp. 1–6. DOI: 10.1109/GCAT52182.2021.9587716. [Online]. Available: https://ieeexplore.ieee.org/document/9587716 (visited on 07/16/2024).

[59] STMicroelectronics, *STM32CubeMX - STM32Cube initialization code generator - STMicroelectronics*. [Online]. Available: https://www.st.com/en/development-tools/stm32cubemx.html (visited on 07/26/2024).

[60] V. Anand, V. Singh, and V. K. Ladwal, “Study on PCB Designing Problems and their Solutions,” in *2019 International Conference on Power Electronics, Control and Automation (ICPECA)*, Nov. 2019, pp. 1–5. DOI: 10.1109/ICPECA47973.2019.8975402. [Online]. Available: https://ieeexplore.ieee.org/document/8975402 (visited on 07/16/2024).

[61] Altium, *Two 4 Layer PCB Stackups With 50 Ohms Impedance*, en, Oct. 2021. [Online]. Available: https://resources.altium.com/p/two-alternative-4-layer-pcb-stackups-50-ohms-impedance (visited on 07/25/2024).

[62] Altium, *Impedance Calculations and PCB Stackup Design in Altium Designer*, en, Nov. 2020. [Online]. Available: https://resources.altium.com/p/pcb-stackup-impedance-calculator (visited on 07/25/2024).

[63] Altium, *Designing a 4 Layer Stackup With 50 Ohm Impedance PCB Traces*, en, Jul. 2021. [Online]. Available: https://resources.altium.com/p/designing-4-layer-pcb-stackup-50-ohm-impedance (visited on 07/25/2024).

[64] Analog Devices, *Lightweight TCP/IP (lwIP) Stack | Analog Devices*. [Online]. Available: https://www.analog.com/en/resources/evaluation-hardware-and-software/software/adswp-lwip.html#software-overview (visited on 07/31/2024).

[65] FreeRTOS, *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions*, en-US. [Online]. Available: https://www.freertos.org/index.html (visited on 07/31/2024).

[66] J. Busquets, J. V. Busquets, D. Tudela, *et al.*, “Low-cost AUV based on Arduino open source microcontroller board for oceanographic research applications in a collaborative long term deployment missions and suitable for combining with an USV as autonomous automatic recharging platform,” in *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, ISSN: 2377-6536, Sep. 2012, pp. 1–10. DOI: 10.1109/AUV.2012.6380720. [Online]. Available: https://ieeexplore.ieee.org/document/6380720 (visited on 05/15/2024).