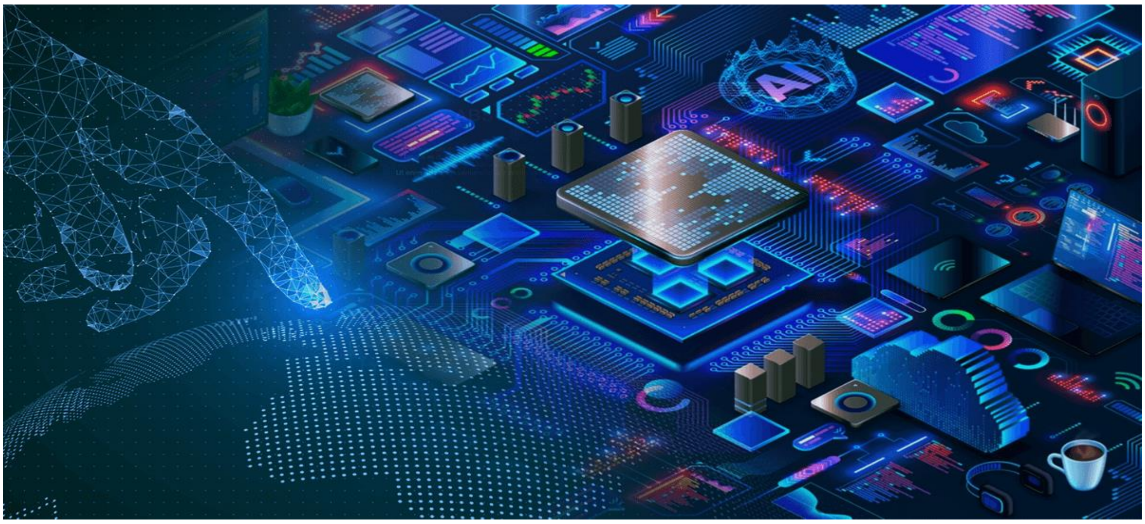
Bachelor Thesis

Master of Computer Science and Engineering 300 hp



Developing Submarine for Monitoring Water Quality through a Cooperative and Innovative Approach

Degree Project in Computer Science and Engineering 15hp

Halmstad 2022-06-05
Mohammad Hammoud

HALMSTAD UNIVERSITY

Abstract

This report is about implementing a real-time monitoring water quality system to measure water quality autonomously in any water environment. The purpose of collecting data is to analyze the results and build a clear picture to move quickly to find solutions in the event of pollution or any other dangerous circumstances. The report explains the submarine system that connects to the winch system, part of the primary system.

Two Raspberry Pi microcontrollers and sensors are used to collect the data. Serial connection is used to build communication between the microcontrollers in order to transfer data between them. The focus area of the whole project is the communication part since the prototype's design is not a part of this thesis. Storing the collected data happens in the central microcomputer in a CSV file. However, it should be stored in a database server to take advantage of this data in the fastest way possible.

Keywords: IoT, microcontrollers and microcomputers, sensors, programming, python, database

Sammanfattning

Den här rapporten handlar om att implementera ett vattenkvalitetssystem för övervakning i realtid för att mäta vattenkvaliteten självständigt i vilken vattenmiljö som helst. Syftet med att samla in data är att analysera resultat och bygga upp en tydlig bild för att snabbt kunna hitta lösningar i händelse av föroreningar eller andra farliga omständigheter. Rapporten förklarar ubåtssystemet som ansluter till vinschsystemet, en del av det primära systemet.

Två Raspberry Pi-mikrokontroller och sensorer används för att samla in data. Seriell anslutning används för att bygga kommunikation mellan mikrokontrollerna för att överföra data mellan dem. Fokusområdet för hela projektet är kommunikationsdelen eftersom design av prototypen inte är en del av detta arbete. Lagring av insamlade data sker i den centrala mikrodatorn i en CSV-fil. Det bör dock sparas i en databasserver för att dra resultat och slutsatser av denna data på snabbast möjliga sätt.

Nyckelord: IoT, mikrokontroller och mikrodatorer, sensorer, programmering, python, databas

Contents

1 Introduction

1.1 Project Overview

Nowadays, autonomous drone systems are widely spread in various fields because of their benefits. These uses can be seen in several domains, such as agricultural engineering or military assignments [1]. This project is to build an autonomous system to increase the flexibility and accuracy of water monitoring while reducing technical and operational complexity and cost. It can obtain real-time data from the ecosystem, with movement capabilities achieving a representative sample in all physical dimensions (X, Y and Z coordinates). X represents the forward and backward directions, Y represents the right and left directions, while Z represents the depth of the water environment.

The project's vision is to prototype and build an autonomous system of underwater drones designed to comprehensively detect, monitor, and map water ecosystem health indicators. The system ultimately aims to reduce the complexity and cost of water quality monitoring using IoT, drone and AI technologies.

The system presented in this report consists of the submarine, an underwater drone capable of sampling and moving in marine and freshwater bodies such as rivers and lakes, and the winch system to descend and ascend the submarine. However, the project could be a part of a larger project, such as connecting this system to another autonomous system as a boat capable of moving to measure water quality.

Getting samples from freshwater is a complex process because water is not homogeneous. The mass varies during the day and throughout the year [2], making it essential to obtain multiple samples during the day to get the best result. This collected sample is representative of the quality of that particular site then. In addition, most representative sampling sites tend to be isolated and far from the coast, which makes it hard to collect without using any technologies.

Several factors can affect the water's quality as anthropogenic, atmospheric, and geological inputs. Each of these factors benefits some organisms and harms others, potentially leading to dramatic changes in ecosystems. In the Earth's crust, minerals and rocks can be leached out by rainwater and natural weathering, leading to surface water contamination and public water systems [3]. For example, high fluoride concentration was highly reported in surface water in areas with fluoride-rich volcanic rocks [4].

1.2 Environmental Problem

Manufactured toxic substances caused by humankind, natural and microorganisms in water bodies can cause harm to wildlife and human health. An example of a symptom that spreads through water is diarrhea, one of the deadliest diseases among children globally [5]. Water pollution negatively affects human health and the environment. According to an analysis published in The Lancet, Water pollution caused 1.8 million deaths in 2015 [6].

Dangerous water is the reason for the illness of about 1 billion people. Many poor communities are most at risk because most homes are usually closest to the most polluting industries. Statistics show that 3.5 million Americans suffer from skin diseases every year due to swimming in polluted water, such as "hot tub rash" (Pseudomonas folliculitis) [7]. Marine ecosystems are highly affected by a chemical alteration of water surfaces [8]. For example, shellfish require need carbonate to build their skeletons and shells. However, when acidity levels rise, the ocean's carbonate levels go down; this threatens threats those animals' lives.

1.3 Project Goals

The project aims to measure water quality autonomously to preserve the environment and to raise awareness among companies' owners and society of the essential of safeguarding the environmental community. The collected data reflects water safety to make the government moves quickly if water pollution occurs. The project will support the authorities' decision-making in Sweden to achieve environmental compliance and help raise public awareness, reinforcing actions taken by society to build better aquatic ecosystems.

The more significant reason is that environmental awareness urges the government to restore the water body to its pristine state. In addition, ecological scientists center on safeguarding ecosystems, protecting endangered species, and improving human health standards. At the same time, continuous monitoring of environmental quality parameters is essential to restore and manage the environment [9].

1.4 Purpose

This thesis will cover two areas of the development process, the submarine part and the winch part.

- The submarine: will cover all required components and their programming part as hardware and software configurations.

- The winch: descend the submarine to the level of depth requested, and ascend it back up. It will cover the programming part of it.

The designing part is not included in this thesis. However, the focus area is to make those two areas work as intended when connecting them with the different components, considering the known problems and circumstances that may appear.

The specification can be listed in four main points:

- Program the submarine sensors in order to collect data.
- Program the winch operation in order to ascend and descend it.
- Set up a connection between the two embedded systems in order to transfer the collected data to the primary system.
- Save the collected data.

How to test those areas to ensure achieving the purpose is described in **How the task is specified** section.

1.5 Limitations

The primary constraints for this project are economic issues because this project is my own, which means there are no funds for it. The limitations affect the following:

- Using only a temperature sensor instead of several sensors to measure water qualities. Measuring only the water's temperature is not what any particular entity aspires to. Water temperature does not give enough information about its water environment.
- Instead of a database, the collected data will be saved on a CSV file, including the date and time.
- Using a battery is the usual solution to supply the prototype with electricity. In this case, it will be supplied by direct plugging into the building's electricity.
- In order to choose a suitable motor for the winch, a study should be done considering the submarine's weight and the provided power. Because this thesis is in the range of computer science, the motor

type and the winch design are unnecessary. However, to make it run as it should be is the scope of our interest. Considering the budget, a DC motor is enough to build this prototype.

- It is impossible to provide a real-time test environment to test the prototype because there is no design.

1.6 How the task is specified

Testing the prototype after the software and the hardware setup is essential to ensure the task is specified in the desired way. Testing methods will be on three main areas:

- **Ensure that the sensors work:**
  - Check if the collected data is reasonable. This can be done by checking the temperature measurement if it is reasonable.
  - Ensure the collected data is saved successfully. This can be tested by checking the CSV file in the main system after the measurement process is done of one or more coordinates.
- **Ensure the communication between the sub and primary system:** Testing the communication between the microcomputers is the most significant part of this project. This can be done by testing the start of the winch motor based on the sonar measurements and transferring the collected data between the microcomputers to save it.
- **Ensure the system can handle any hardware errors:**
  Ensure the emergency system works if any error occurs on any sensor. This can be tested by removing the temperature sensor manually, and the motor should ascend after three attempts of running the code.

The final test is to test all these three areas together as one system. These tests are more described in **Analyzing the results** section in **Materials and Methods** chapter**.**

Figure 1 explains how the whole system is connected together.

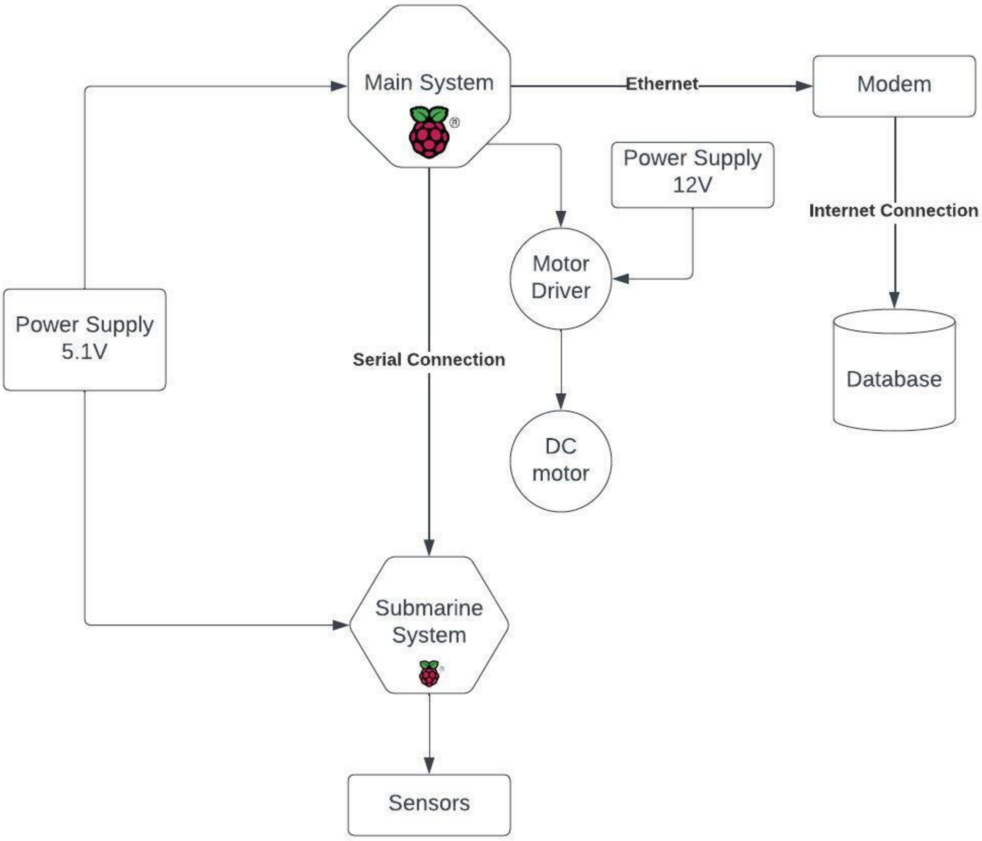


Figure 1: an overview of the whole system.

2 Background

The same idea of making a real-time water quality monitoring system has been done before in many other projects. One example is *CatFish project* [10], a Fish connected to an autonomous boat to measure water quality from different points and depths in water environments. The second example is *Design of Smart Sensors for Real-Time Water Quality Monitoring* [11]. Sensors are connected to microcomputers to collect and transmit data via receiver modules and a radio transmitter. The third example is *IoT Application for Real-Time Monitoring of Solar Home Systems Based on Arduino™ With 3G Connectivity* [12]. It is an IoT system that uses 3G to connect to the internet in order to measure climatic and electric parameters using the Arduino embedded system. The last example is *Weather tracking system using MQTT and SQLite* [13]. This IoT system is controlled by the Raspberry Pi, which transmits messages using MQTT protocol.

2.1 Internet of Things (IoT)

The Internet of Things (IoT) is an emerging paradigm that makes our lives easier by enabling communication between electronic devices and sensors over the internet. To provide innovative solutions, IoT leverages intelligent machines with the internet. Those solutions are based on problems and challenges related to the world communities [14]. IoT is an innovation that brings together various frameworks, intelligent systems, sensors and smart devices. Additionally, it leverages nanotechnology and quantum in storage, acquisition and processing speeds that were previously unimaginable [15]. For example, a monitoring sensor for real-time ammonia (NH3) was developed by Tang et al. in 2019 [16].

2.2 Microcontrollers and Microcomputers

An embedded system is a microprocessor or microcontroller system, which is a combination of computer hardware and software designed to achieve a specific assignment. A fire alarm is an example of an embedded system [17]. *An Arduino* board can read inputs such as a finger on a button or light on a sensor. These inputs convert into an output. Man can control this process by sending instructions to the microcontroller on the board. The Arduino programming language that should be used is C/C++ [18].

Raspberry Pi is another well-known embedded system. It is a low-cost, credit-card-sized computer that enables us to program all the connected components using python programming language. It does everything that a desktop computer would do, such as playing videos/games and browsing the web [19].

2.3 Sensors

Humans today are surrounded by a world full of sensors. All different types of sensors are everywhere [20], such as in our cars, homes and offices, that is because to make our lives easier, such as adjusting the room temperature or detecting fire.

A *sensor* is a device that detects and responds to some input from the physical environment. The specific information can be movement, light, pressure, heat, humidity, or any factor of many other environmental phenomena. It works to sense these phenomena and convert them to human-readable displays [21].

There are many types of sensors, such as digital, camera, optical, etc. The decided sensors to be used are the following:

- Ultrasonic Distance Sensor - HC-SR04 [22]
- Temperature Sensor - DS18B20 [23]

2.4 Python

The founding of programming languages is that computer programmers can develop scripts and software programs etc. Many languages have some similarities, but each has its syntax. In order to make the computer understand the code, the programmer compiles the code into machine language so that it can understand it. However, Scripting languages use an interpreter to execute the script because they do not require a compiler [24].

Python programming language is an open-source language developed by Guido van Rossum, a Dutch programmer in 1991 [25]. It is considered an easy-to-use and very compact language. Python later became one of the most popular programming languages, including JavaScript and Java [26].

2.5 Power

Power is essential to any similar project to make it work. The energy in this project is necessary to run the microcomputers and the winch motor. The best power supply for an autonomous robot is rechargeable batteries. With the presence of solar panels, the battery would be able to recharge. In this case, the robot would do the task without any human interruption.

In order to supply a *Raspberry Pi 3 Model B+* with power, it needs a 5V, 2.5 A [27]. If power is not enough, it should boot, but it may reset on its own as it does not have enough ability to run correctly. *Raspberry Pi 3 Model B+* consumes about 400 mA of current at 5.0 V (about 1.9-2.1 W) when no USB devices are connected and idle [28].

In order to choose the proper power supply for DC motors, two points should be considered; the first is the maximum current used by the motor and the second is the behavior of the power supply when it is connected to a short circuit [29]. DC motors can operate from 1.5 to 100 Volts or even more. Most robots are battery-powered, making roboticists often use motors that operate on 6, 12, or 24 volts [30], and most batteries are available with these values.

If the power supply was a battery, the cable used to build communication between the sub and primary system (microcontrollers) should power the submarine system besides the communication part. This cable would transform power from the power supply to the submarine as a USB connection. Choosing a USB instead of a GPIO connecter (5V and GND) is more beneficial because there might not be enough free pins connections in the raspberry board if many sensors are connected.

2.5.1 Motor Drivers

Motor drivers are integrated circuit chips used as motor controllers in autonomous robots and integrated circuits [31]. Indeed, a motor driver is something that moves the motor based on given instructions or input (low and high). Microprocessors perform on low-level voltage/current, in contrast to motors. For example, the famous Raspberry Pi or Arduino microcontrollers have an operating voltage of 3.3V or 5V, but a DC motor requires 5V or 12V [32]. We need a high voltage to supply the power to the motor. However, that microprocessor output cannot give enough power from its I/O pin to drive a motor, and this is why it needs a Motor driver between the motor and controller. The one used for this project is *L298N Motor Driver Module* [33].

Figure 2 shows how the motor driver is connected to the power supply and the microcomputer.

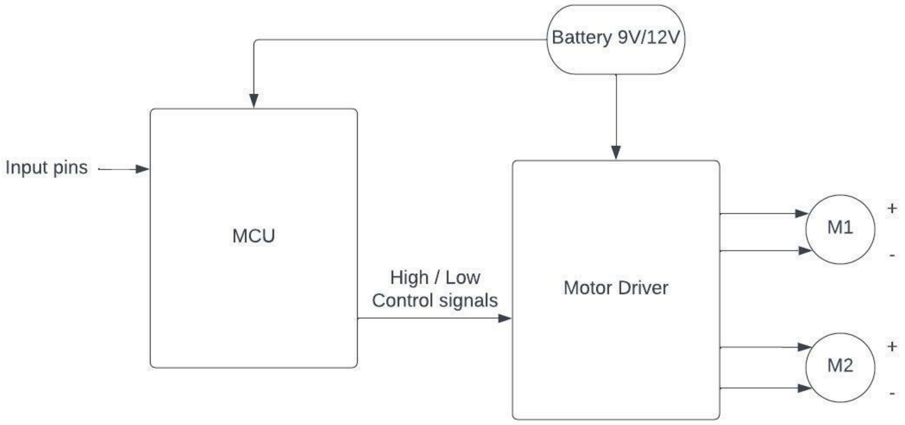


Figure 2: block diagram of motor drive.

2.6 Database

Throughout the ages, books have been the primary references for data. The entry of technology into human life has a considerable impact; for example, books have become online, which means anyone can access their data without having the physical book itself. Libraries are the natural place to contain books. So, what would correspond to libraries on the internet world? To answer that in a word, I would say a cloud. Online services such as email, file storage and bank services use cloud computing. It provides from any device access to servers, documents, etc and it protects business continuity. Examples of Cloud services out on the market are *Google Cloud Platform, Microsoft Azure and Amazon Web Services* [34].

A database is a structured collection of data that makes it easy to access and maintain. Its primary purpose of it is to manage large amounts of information by managing, storing and retrieving data. It was created in such a way that only one set of software programs allows all users to access the data. You can organize and index data into tables, rows, and columns to find the relevant information in an easier way [35]. There are multiple different types of databases; a couple of examples are relational databases and non-relational databases.

2.6.1 SQL Databases

Relational Databases were founded in the 1970s. The data is stored in multiple linked tables. In a table, information is stored in rows and columns. Users can manage it by RDBMS (relational database management system). The most widely used language to manage the database in RDBMS is SQL (Structured Query Language). This type of database is very reliable; it conforms to ACID, which is a bunch of standard properties of the database transactions (Atomicity, Consistency, Isolation, Durability). Examples of software are *Microsoft SQL Server* and *PostgreSQL* [36].

2.6.2 NoSQL Databases

Unlike relational databases, this type is ideal for organizations that want to store unstructured/semi-structured data because the data do not have to conform to a predefined schema. All databases that do not use SQL as their primary language are considered non-relational databases. The most significant advantage of those databases is that changes can be made on the fly without affecting the database applications [36].

2.7 Communication

Communication is necessary between devices to connect them, especially if there is a desire to save data. The embedded systems used in this project should communicate with each other. It can be done by wireless or cables method. Internet access is another essential factor in building similar projects, and this could be done using *WiFi technology*. To access the internet on an embedded system, using *Mobile Broadband technology* is one of the popular solutions today. *TCP/IP protocol* is needed in order to transmit data over the internet.

Create a successful connection between microcomputers, in their turn, a successful connection to the internet to access the database server. When these steps are done successfully, the collected data can be stored in the database and reviewed whenever the user wants it. To understand how this can be done, we have to cover these technologies and protocols in detail.

2.7.1 Wi-Fi and Mobile Broadband Technologies

WiFi "Wireless Fidelity" has spread over the last years fast to become the dominant wireless LAN. Anyone can set up a *WiFi* network because it works in unlicensed frequency bands and covers around 100-500 feet with high-speed wireless access to a LAN and therefore to the internet [37]. It is a wireless Ethernet 802.11b standard for WLANs and it refers to the radio transmission of the data from a wireless internet connection to a host computer [38]. The Internet connection is usually a higher speed than slower dial-up connections such as DSL, satellite or cable. It is essentially a wireless connection between the internet and the user’s computer.

Mobile broadband is also known as a USB modem; mobile broadband refers to an internet connection delivered over a mobile network. The connection process is much faster because mobile broadband is delivered "over the air," and it is a plug-and-play technology [39].

2.7.2 Serial Connection

Serial connection is one way to set communication between embedded systems. For example, the Raspberry Pi has a UART serial port on the GPIO 8, RXD GPIO 15 and TXD on GPIO 14 and 10 [40]. To connect multiple devices to it using The UART port with a serial interface to convert the UART voltages to an industry-standard such as RS485 or RS232 or directly using 3.3V UART pins. This mechanism is the same for Arduino as well [41]. By enabling this connection, the collected data will be able to transfer from the secondary microcomputer to the main head one in order to save it later on.

2.7.3 UART Protocol

UART stands for universal asynchronous receiver transmitter and is one of the most popular communication protocols. This protocol can work with different serial protocols that involve receiving and transmitting data. Serial communications need fewer wires and circuitry depending on the system requirements and application, which decreases the cost of implementation [42].

The UART interface transmits data asynchronously, so it does not use a clock signal to synchronize both devices (receiver and transmitter). The receiver uses the internal clock signal to sample the incoming data, while the transmitter generates a bitstream based on its clock signal. The synchronization point is handled using the same baud rate on both devices [43].

2.7.4 TCP/IP Protocol

TCP/IP (Transmission Control Protocol/Internet Protocol) permits information to be transferred among devices on a network. It is designed to interrupt a message into packets of information to make sure the message reaches its final spot correctly as fast as it can. [44]. It uses the client-server communication model in which another server delivers a client service in the network. This protocol is typed as stateless, which means each client request is assumed as a new request; stateless frees up network paths so it can be used continually.

The Internet Protocol (IP) is a manner of transmitting data between devices. Every device has an IP address that allows it to communicate with other devices through the internet and uniquely identifies it. IP is responsible for determining how devices and applications exchange data. IP is the primary protocol within the internet layer of the TCP/IP. Its primary goal is to deliver data packets between the device or source application and its destination using structures and processes that place tags within data packets, such as address information.

2.7.5 MQTT Protocol

This is another way to transfer data between two microcomputers via WiFi (MQTT protocol). MQTT stands for Message Queuing Telemetry Transport, a publication-subscription messaging protocol based on the TCP / IP protocol [45].

Figure 3 shows the MQTT's operating principle. Publishers are machines such as Arduino and Raspberry Pi that capture values such as temperature and transmit them to a Broker server. Clients are linked to this server and request that specific data be transmitted. When the Broker server receives the desired data, it resends it to the subscribed clients of this data flow.

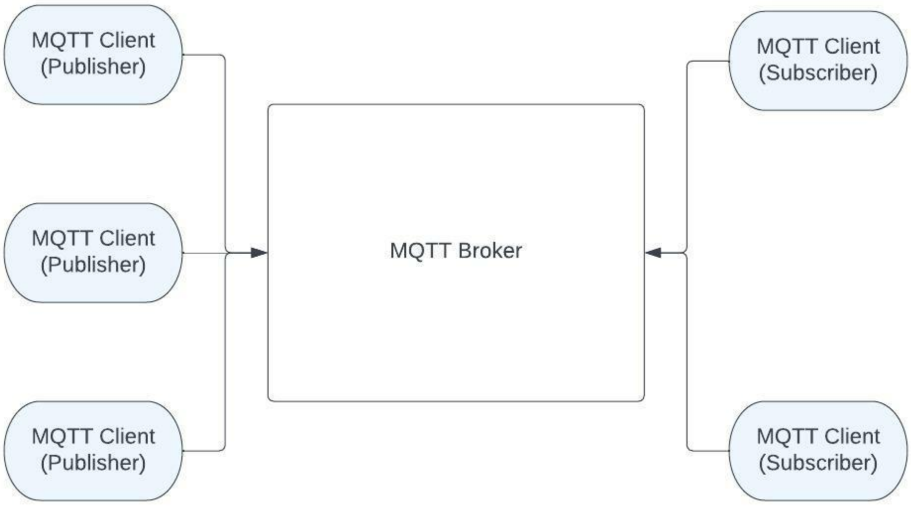


Figure 3: diagram shows the MQTT's operating principle.

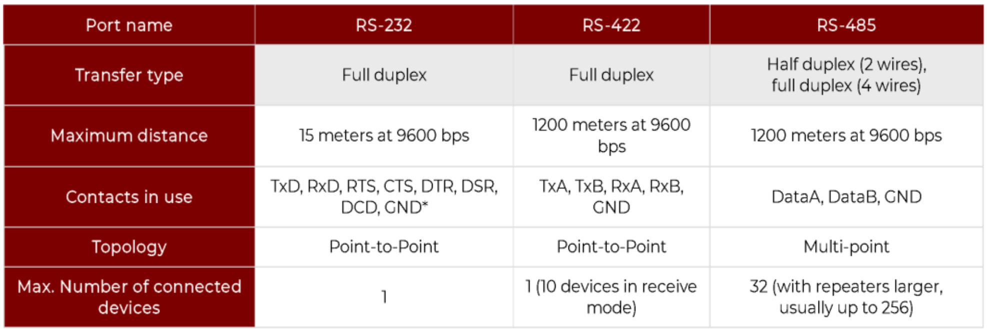
2.7.6 Comparison of RS-232, RS-422 and RS-485

The expansion of faster ports such as USB made RS-232 an old technology, but RS-232 still has its uses because of Its ease of implementation and low cost. RS-232's capacities have been stretched to over 1 Mbps for some applications today. However, RS-232 is intended for short cable runs up to 50 feet [46]. It is created for a unidirectional half-duplex communications mode which means you can transmit from a sender to a receiver over a copper cable. The data moves in one direction over that cable. If both ways of transmission are wanted, two separate communications paths are needed in this case. It operates from a 3 to 5V supply and provides ESD protection on all transmit and receive pins. RS-232 ICs today are more qualified than what was first presented in 1962. On the other hand, if a greater distance, higher data rate, or transmission to multiple receivers is desired, RS-422 becomes the most practical choice.

The primary purpose of designing RS-422 was to overcome the speed and distance limitations of RS-232 and accommodate multiple receivers. The length of data lines can reach 4000 feet with a 100kbps data rate. Unlike RS-232, it can serve up to 10 receivers on the same bus [47]. However, RS-422 is still half-duplex one-way data transmissions over a two-wire line. On the other hand, if the full-duplex operation is desired, RS-485 becomes the best consideration in this case.

RS-485 Standard provides bidirectional half-duplex multi-point data communications over a single two-wire bus. Both RS-485 and RS-422 have the same distance and data rate specifications. It uses differential signaling, allowing multiple drivers on the same bus, up to 32 drivers and 32 receivers on the same bus [48]. Table 1 shows the main differences that have been mentioned in this section between these Standards.

Table 1: differences between RS-232, RS-422 and RS-485.



Port name	RS-232	RS-422	RS-485
Transfer type	Full duplex	Full duplex	Half duplex (2 wires), full duplex (4 wires)
Maximum distance	15 meters at 9600 bps	1200 meters at 9600 bps	1200 meters at 9600 bps
Contacts in use	TxD, RxD, RTS, CTS, DTR, DSR, DCD, GND*	TxA, TxB, RxA, RxB, GND	DataA, DataB, GND
Topology	Point-to-Point	Point-to-Point	Multi-point
Max. Number of connected devices	1	1 (10 devices in receive mode)	32 (with repeaters larger, usually up to 256)

3 Materials and Methods

3.1 Choice of hardware and coding environment

For both embedded systems in the submarine and the primary system, a *Raspberry Pi 3 Model B+* is selected to perform this task. That is because of the different features and advantages it has. This small embedded system has considerable processing power in a compact board, making it possible to build effective and complex systems at a lower price. It has many interfaces (Wi-Fi and Bluetooth, many GPIOs, USB powered, etc.) that would help to build this project, considering the possibility of implementing more sensors is taken into account.

Choosing a programming language is essential in order to be able to program those devices. *Python* is the language of choice for this project and that is because the *Raspberry Pi* Foundation specifically selected Python as the primary language [49]. For the sensors' coding part, this section was done by going back to the open-source libraries developed for those specific sensors.

A ten-meter waterproof pins cable is needed to build communication between the sub and primary system (microcontrollers). The number of pins required is three to make a serial connection (UART).

Having a sub and primary system is that the submarine can now descend 10 meters deep in water without any issues. However, suppose the submarine components and the winch are connected to one microcontroller. In that case, we have to consider all the component's wires in the descent process, which are three wires per sensor.

3.2 Codes Implementation

The programs used in the software are *Raspberry Pi OS* and *Visual Studio Code*.

3.2.1 Code of the submarine sensors

The code of the sensors is structured so that sensors start reading data when a command from the primary system is sent that the submarine is in the right place. After this step, the collected data is transferred to the submarine system and then to the main system. When the transfer process is done successfully, it requests the main system to ascend the winch. Figure 4 shows the structure of the submarine code and how it works.

If something goes wrong, for example, one of the sensors does not work for some reason, the submarine will try to run that specific sensor three times in a time frame of six seconds; if the sensor still does not respond, it sends the ascend request and an error message, so the user knows something wrong with that specific sensor.

In order to know which value is coming from which sensor and which sensor has the error message, the data collecting process would be one by one. If we have three sensors, for example, (Temperature, Dissolved Oxygen and pH value) this means the PH value sensor would not start working until the temperature sensor is done first, then the Dissolved Oxygen.

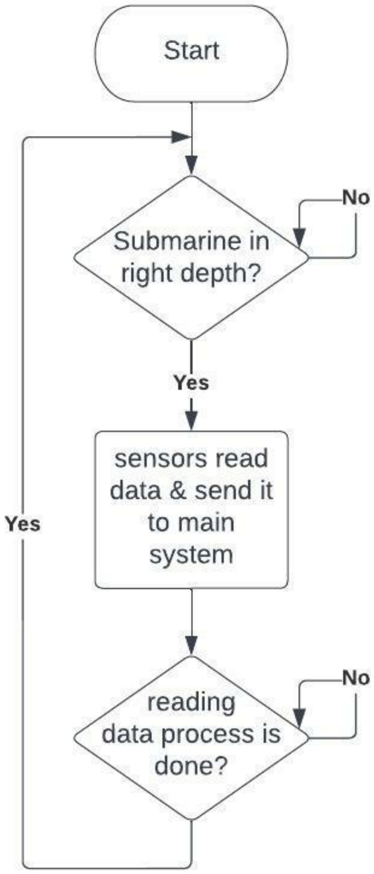

Start
No
Submarine in right depth?
Yes
sensors read data & send it to main system
Yes
No
reading data process is done?

Figure 4: flowchart of the submarine code.

3.2.2 Code of the winch

As seen in Figure 5, the descent process starts when a command from the main system is sent to the motor. This command means the winch motor would start when the robot is at the desired point that we want to take measurements from. The descent process stops when an order is sent from the submarine system to the main one based on the sonar sensor data. This command means stopping the process because 100cm is left to the bottom in order to avoid collisions. The ascent process starts when the collected data is transferred successfully to the main system or an error message if hardware problems cause any errors. In this case, any error regarding the submarine will not affect the winch system. The robot can collect measurements from the following coordinates even if any sensor in the submarine is damaged.

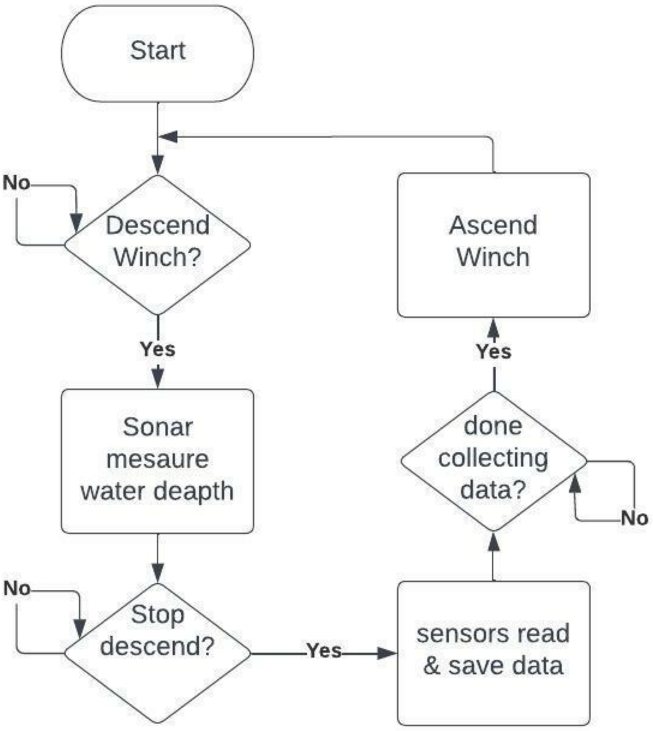


Figure 5: flowchart of the winch code.

3.3 Transfer the collected data to the main system

The method used to build communication between the sub and primary system is serial communication. Utilizing this method, data are sent to the main microcomputer using three PIO pins (GND, TXD0, RDX0). Because of the desired range of this communication (ten meters long), converting the UART to RS-232 Standard is essential to shift up the voltage from 3.3 V to 5

V on the Raspberry Pi. This conversation will avoid problems that disconnect the connection. To do this, a RS-232 Raspberry Pi converter [50] is used.

3.4 Saving the collected data

Creating a database on a server is needed to save the collected data. This section refers to the **Background** chapter's *Wi-Fi and Mobile Broadband* and *Database* sections. To get access to the internet using a 3G or 4G modem to access the database, see Figure 1. Because of the time frame, saving data will be on the main microcomputer system instead of a database. The collected data will be saved on a CSV file, including the date and time. However, creating a database will be covered later in the improvements section in the **Discussion** chapter.

3.5 Analyzing the results

To analyze the results, multiple tests should be performed in the following areas:

Communication tests:

How to test: By sending a message "Hello from the submarine" from the sender (The submarine) to the receiver. the second test is by running the winch motor and facing the sonar to the wall. These tests will be performed using 1, 2-, 3-, 5- and 10-meters long cables.

Expected result: The message should be seen in the main system's terminal, and the motor should stop when the distance between the sonar and the wall is 10 cm or less.

Reasonable data:

How to test: By making a comparison between the collected data from the temperature sensor with the room thermometer device.

Expected result: Both data should be the same value.

Saving data:

How to test: By running the code multiple times, checking if a CSV file has been created in the main system with the measurement values. The next step is to delete the file and rerun the code.

Expected result: The data should be saved with time and date in the same file without losing the old data. after deleting the file, it should create a new file with the same name when rerunning the code.

Sensor error:

How to test: By disconnecting the sonar sensor before starting the operation and disconnecting the temperature sensor during the measurements.

Expected result: The system should continue working without any hindrance from hardware errors regarding the temperature sensor. On the other hand, the system should not start if there is any hardware error regarding the sonar sensor.

4 Results

Figure 7 shows a schematic diagram of how the submarine's sensors are connected to the microcomputer, and Figure 8 shows the same for the main system.

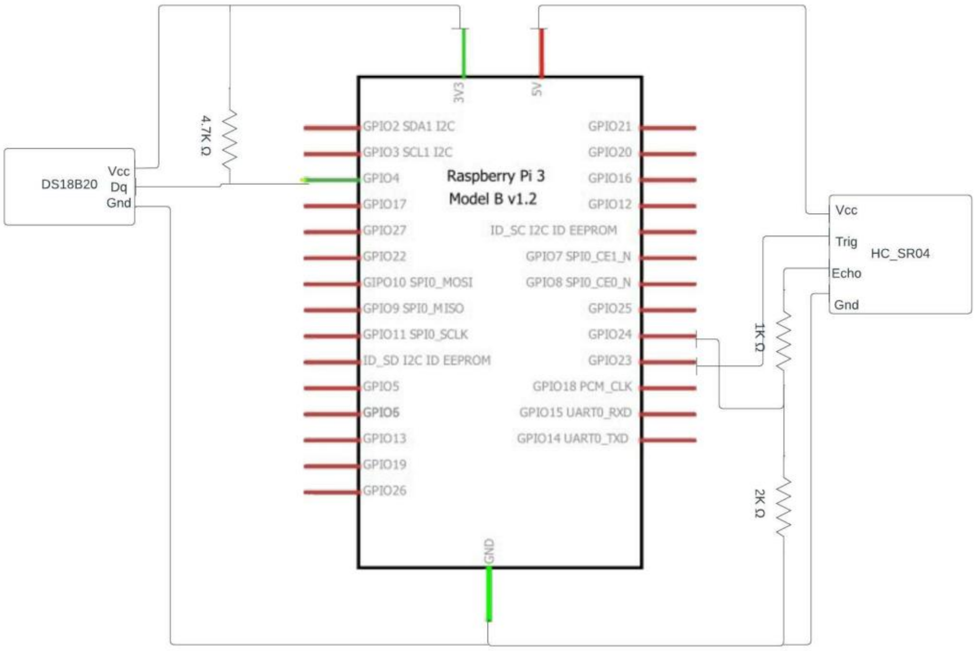


Figure 7: schematic diagram of the submarine's sensors.

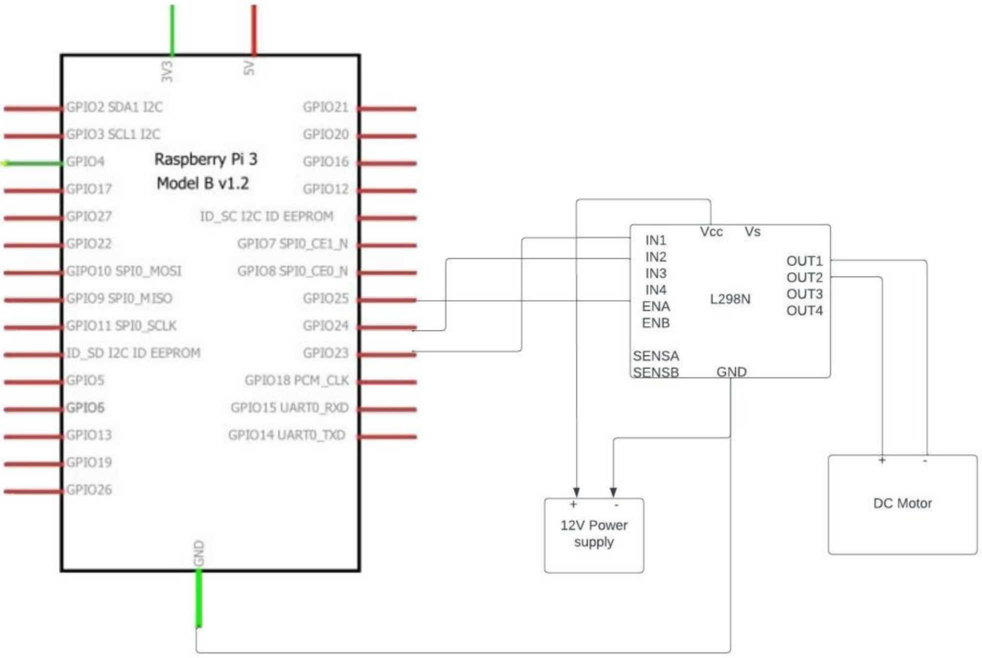


Figure 8: schematic diagram of the winch.

Because the prototype cannot be put in the water now, the collected data are from the sensors in the open air. The following tests areas are related to **Analyzing the results** section:

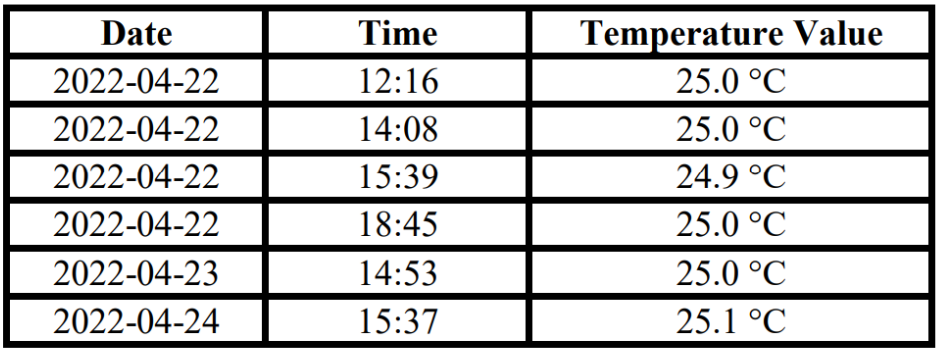
Communication tests: Tests were performed using different cables (1, 2-, 3-, 5- and 10-meters long). The first two cables worked fine with the UART protocol, the message "Hello from the submarine" and the sonar measurements were sent successfully to the main system.

On the other hand, the rest cables did not transfer the data through them, the message did not transmit to the main system, and a massive number of random chars were shown in the terminal instead of the sonar measurements. Converting the UART to RS-232 Standard by connecting the converter to the cable solved the problem, it shifts the voltage to 5V, which makes these longer cables manage to transmit the data through them.

Reasonable data: Tests have been performed on the temperature sensor. The room temperature was collected in standard conditions and the sensor was exposed to a high-temperature object to ensure data validity.

Saving data: This action was performed successfully. a CSV file called "measurements " was created in the main microcomputer with all-temperature measurements with time and date, see Table 2. Running the code multiple times saves the new data in the same file without losing the old ones.

Table 2: shows the stored data in the main system.

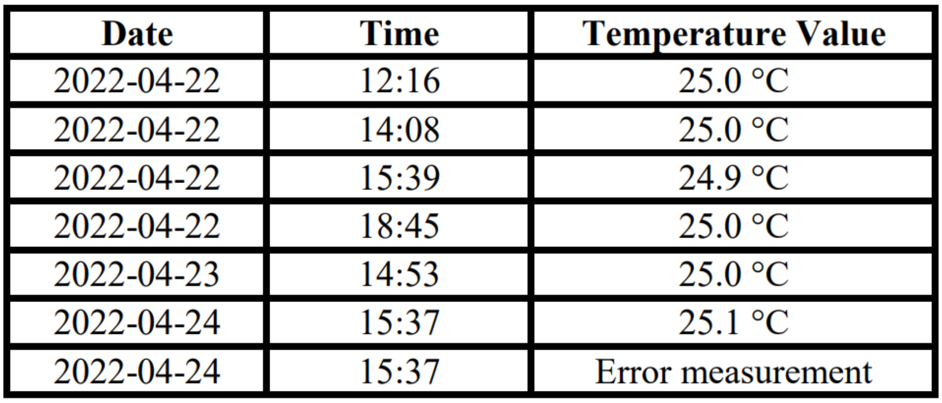


Date	Time	Temperature Value
2022-04-22	12:16	25.0 °C
2022-04-22	14:08	25.0 °C
2022-04-22	15:39	24.9 °C
2022-04-22	18:45	25.0 °C
2022-04-23	14:53	25.0 °C
2022-04-24	15:37	25.1 °C

Sensor error: In order to test if any hardware errors occurred to any sensor, the temperature sensor was removed to check the winch functionality. The system tried to run the code 3 times in a time frame of six seconds. After that, it should write "Error measurement" in the CSV file and move to the next sensor, see Table 3. Because the prototype has only one sensor, the winch motor started after those six seconds in a reverse movement to ascend the submarine.

Any error in the sonar sensor would make the winch motor descent until infinity. This leads the submarine to a collision with the bottom of the water. The software part solved this problem by checking before the descent process happens if the sensor is working. If not, then the motor will not start at all. This result was satisfying as expected.

Table 3: shows the stored data in the main system when hardware error occurs.



Date	Time	Temperature Value
2022-04-22	12:16	25.0 °C
2022-04-22	14:08	25.0 °C
2022-04-22	15:39	24.9 °C
2022-04-22	18:45	25.0 °C
2022-04-23	14:53	25.0 °C
2022-04-24	15:37	25.1 °C
2022-04-24	15:37	Error measurement

An integration test was performed. The operation started by running the motor, and the motor stopped when my hand reached 10cm to the sonar sensor. The temperature sensor reads data and transfers it to the submarine and then to the main system. Whenever the storing process is completed successfully, the motor runs in reverse with the same time frame it took to descend.

The winch motor stops based on the data transmitted from the sonar. The measure sensors in the submarine start based on the motor status. Data would be collected from a specific depth, so the system avoids collecting data all time which avoids any overload problems as well.

The project's task is complete when the central system can complete the whole mission in a real-time environment without human intervention. The results of all previous tests were satisfying as expected.

5 Discussion

The main focus area of this thesis was how to build a successful communication to transmit data between the microcomputers. This was done by serial connection, and the UART protocol had to convert it to RS-232 standard in order to maximize the distance to 10 meters. As mentioned earlier in the **Background** section, other similar projects focused more on how to collect data. Another exciting area besides communication could be how to store this data and visualize it for users and audiences.

In *Design of Smart Sensors for Real-Time Water Quality Monitoring* [11], their focus was on how to make the collected data of the sensors precise. In contrast to this project, the focus area is on how to transmit the collected data instead of measuring accurately. They transmitted information wirelessly to a notification node using two standalone ZigBee specification modules covering 10-70 meters. The microcontroller then converts the data into char variables and then can be transmitted across the wireless modules through serial communication.

The focus area in *Weather tracking system using MQTT and SQLite* [13] is how to implement a framework for publishing data from ESP8266 to SQLite data server hosted on Raspberry Pi. Their used method is MQTT Protocol. An ESP8266 microcontroller acts as MQTT client and sensor data is sent to the MQTT broker running on a Raspberry Pi. Additionally, an SQLite data server is also hosted on the Raspberry Pi module, and the data from the ESP8266 is stored in JSON format.

5.1 Limitations of my solutions

As mentioned in **Sensor error** in the **Result** chapter, the winch system will not start if the sonar does not work. This operation would secure the prototype from the collision with the bottom. On the other hand, there would not be any measurements in this case which is a weak point of this solution.

5.2 Improvements of the submarine system

With that secure operation mentioned above, there would be no data in that case. The sensors should collect data from the water surface instead of the bottom in this specific case to improve it. This improvement would be a solution to collect some data, which still enable us to analyze and take action, if necessary, in required cases.

5.3 Improvements of storing data

The most crucial step to improve in this section is to build a database server instead. The server should be improved in handling, security and connection levels, not just a temporary server. As mentioned earlier, this step raises the project's status as a whole, especially in real life when the robot can reach the required results in a natural environment, non-artificial just for the sake of performing tests. Storing data in the microcomputer is not helpful in real life because the user cannot access it during the processing time. The primary benefit of storing data in a database is that the user can access it anytime during the operation to review it or draw the statistics and results. It reduces the time spent managing and analyzing data in various ways, and then information turns disparate into a valuable resource.

An SQL database would be an easy solution to implement. Choosing a relational database would save time because it does not need any querying processes or complicated structuring. Figure 9 is an ER diagram of a database that would fit this prototype. This design makes it simple to add more sensors.

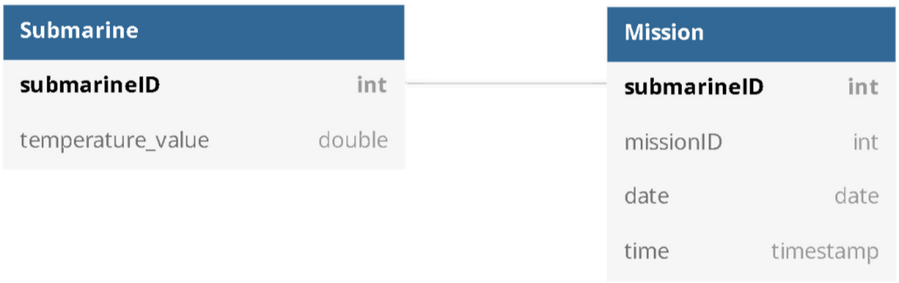


Figure 9: a database ER diagram.

5.4 Social Requirements

This system costs less overall energy compared to full-size computers. Operating it to replace other full-size computers in a more extensive system can significantly decrease the general power consumption and help the environment. Recycling the components is doable for example, Raspberry Pi boards are being recycled.

In addition, this process helps researchers learn and predict natural processes in the environment and determine human impacts on the ecosystem. As mentioned in the Project Goals section, it can also help detect water pollution to stop it before becoming a significant problem. Water pollution can cause aquatic animals to change their locations to find more adaptable water areas. In other cases, the circumstances can kill a whole population of these species leading to extinction.

From an economic view, the lack of clean water limits economic growth by one-third [51]. Deteriorating water quality reduces food production, worsens health conditions, and exacerbates poverty. Different economic sectors need clean water to work and grow, such as farming, tourism and energy production.

6 Conclusion

Clean water is an essential matter that affects the environment, aquatic creatures and human life. A prototype of a submarine measures water quality was implemented in this project. This prototype collects data and stores it in the main microcomputer to analyze it later. The results fulfilled the requirements and have more room for improvements to enhance the project in the future.

Disconnecting any wire of the sensors' wires can happen through the operation. As mentioned before, each sensor has three wires which raises the possibility of disconnecting any one of them. The occurrence of something like this leads to hardware problems, which leads to stopping the whole process. To avoid that, the submarine system is programmed to re-run that specific sensor three times in under six seconds in an attempt to turn it on. If the problem persists, the system will move on to the next sensor and send an error message to inform the user about an error that occurred related to that specific sensor.

The winch operation is based on the sonar sensor readings. Any error in the sonar sensor would make the winch motor descent until infinity. This leads the submarine to a collision with the bottom of the water. To avoid this problem, the motor will not start if no data from the sonar is transmitted to the main system. This solution will stop the whole system before it starts in the first place, and it is better to stop the process than lose the submarine because of the collision.

References

[1] D. Floreano, R. J. Wood, ”Science, technology and the future of small autonomous drones.” Nature, vol. 521, pp. 460-466, 2015.

[2] Kumar YP, King P. Assessment of Ground Water Quality in Visakhapatnam Area Andhra Pradesh, India. Fundamentals of Limnology. 2005:93.

[3] Lund V. Evaluation of E. coli as an indicator for the presence of Campylobacter jejuni and Yersinia enterocolitica in chlorinated and untreated oligotrophic lake water. Water Research. 1996 Jun 1;30(6):1528-34.

[4] Choi WW, Chen KY. The removal of fluoride from waters by adsorption. Journal-American Water Works Association. 1979 Oct;71(10):562-70.

[5] World Health Organization. Diarrhoea: why children are still dying and what can be done. 2009

[6] Denchak M. Water pollution: Everything you need to know. Nat. Resour. Def. Counc. NY. 2018 May 14.

[7] Yu Y, Cheng AS, Wang L, Dunne WM, Bayliss SJ. Hot tub folliculitis or hot hand–foot syndrome caused by Pseudomonas aeruginosa. Journal of the American Academy of Dermatology. 2007 Oct 1;57(4):596-600.

[8] Inyinbor Adejumoke A, Adebesin Babatunde O, Oluyori Abimbola P, Adelani Akande Tabitha A, Dada Adewumi O, Oreofe Toyin A. Water pollution: effects, prevention, and climatic impact. Water Challenges of an Urbanizing World. 2018 Mar 21;33:33-47.

[9] Wetzel RG. Limnology: lake and river ecosystems. gulf professional publishing; 2001 Apr 20.

[10] Halmstad University, "CATFISH - Project Description," 2020.

[11] Cloete NA, Malekian R, Nair L. Design of smart sensors for real-time water quality monitoring. IEEE access. 2016 Jul 19;4:3975-90.

[12] López-Vargas A, Fuentes M, Vivar M. IoT application for real-time monitoring of solar home systems based on Arduino™ with 3G connectivity. IEEE Sensors Journal. 2018 Oct 17;19(2):679-91.

[13] Kodali RK, Gorantla VS. Weather tracking system using MQTT and SQLite. In2017 3rd International Conference on Applied and Theoretical

Computing and Communication Technology (iCATccT) 2017 Dec 21 (pp. 205-208). IEEE.

[14] Sfar AR, Chtourou Z, Challal Y. A systemic and cognitive vision for IoT security: a case study of military live simulation and security challenges. In2017 International Conference on Smart, Monitored and Controlled Cities (SM2C) 2017 Feb 17 (pp. 101-105). IEEE.

[15] Gatsis K, Pappas GJ. Wireless control for the IOT: Power, spectrum, and security challenges. In2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI) 2017 Apr 18 (pp. 341-342). IEEE.

[16] Ezz El-Din H, Manjaiah DH. Internet of nano things and industrial internet of things. InInternet of Things: Novel Advances and Envisioned Applications 2017 (pp. 109-123). Springer, Cham.

[17] heavy.ai, Embedded System Definition, 2022.

[18] Banzi M, Shiloh M. Getting started with Arduino. Maker Media, Inc.; 2022 Feb 15.

[19] Richardson M, Wallace S. Getting started with raspberry PI. " O'Reilly Media, Inc."; 2012 Dec 10.

[20] Wilson JS. Sensor technology handbook. Elsevier; 2004 Dec 21.

[21] G. M. Smith, "Dewesoft," 09 March 2020.

[22] A. Aqeel, "The Engineering Projects," Introduction to HC-SR04 (Ultrasonic Sensor), 05 October 2018.

[23] Components 101, DS18B20 Temperature Sensor, 07 May 2018.

[24] Iverson KE. A programming language. InProceedings of the May 1-3, 1962, spring joint computer conference 1962 May 1 (pp. 345-351).

[25] Steele Jr GL. The Definition and Implementation of a Computer Programming Language Based on Constraints. MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB; 1980 Aug 1.

[26] Van Rossum G. Python Programming language. InUSENIX annual technical conference 2007 Jun 17 (Vol. 41, No. 1, pp. 1-36).

[27] Fletcher AC, Mura C. Ten quick tips for using a Raspberry Pi. PLoS computational biology. 2019 May 2;15(5):e1006959.

[28] S. Shovon, "linuxhint," Raspberry Pi 3 Power Requirements, 2019.

[29] Power Supplies Australia, POWER SUPPLIES FOR USE WITH DC MOTORS, 26 August 2021.

[30] Deal III WF. Electric Motors Everywhere. Technology and Engineering Teacher. 2004 Sep 1;64(1):20.

[31] components101, Introduction to Motor Driver: HBridge Topology and Direction control, 13 November 2019.

[32] Mepits, "Mepits Innovate Develop Share," DC Motor Driver Circuit, 15 June 2015.

[33] components101, L298N Motor Driver Module, 2021 April.

[34] Miyachi C. What is "Cloud"? It is time to update the NIST definition?. IEEE Cloud computing. 2018 May 1;5(03):6-11.

[35] Revesz P. Introduction to databases. Springer, London, UK; 2010.

[36] Foster I, Pascal H. Databases. InBig Data and Social Science 2020 Nov 17 (pp. 67-99). Chapman and Hall/CRC.

[37] Mohamed A, Mohamed MA, Amer T. The Provision of Wi-Fi Service in Quick Service Restaurants: An Analytical Study from Customers' and Managers' Perspective. Journal of the Faculty of Tourism and Hotels-University of Sadat City. 2019 Jun;3(1).

[38] Al-Alawi AI. WiFi technology: Future market challenges and opportunities. Journal of computer science. 2006 Jan;2(1):13-8.

[39] Astély D, Dahlman E, Furuskär A, Jading Y, Lindström M, Parkvall S. LTE: the evolution of mobile broadband. IEEE Communications magazine. 2009 May 5;47(4):44-51.

[40] Knowledge Base, "AB Electronics," Serial Port setup in Raspberry Pi OS, 19 March 2016.

[41] Dawoud DS, Dawoud P. Serial Communication Protocols and Standards RS232/485, UART/USART, SPI, USB, INSTEON, Wi-Fi and WiMAX. River Publishers; 2020.

[42] Bhadra D, Vij VS, Stevens KS. A low power UART design based on asynchronous techniques. In2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS) 2013 Aug 4 (pp. 21-24). IEEE.

[43] Kanse S, Mayuri B, Abhilasha B, Nilam K. Protocol Converter (UART, I2C, manchester protocols to USB). International Journal of Research in Engineering and Technology. 2015;4(1):88-90.

[44] Forouzan BA. TCP/IP protocol suite. McGraw-Hill Higher Education; 2002 Jul 1.

[45] Soni D, Makwana A. A survey on mqtt: a protocol of internet of things (iot). InInternational Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017) 2017 Apr (Vol. 20, pp. 173-177).

[46] Hazen ME. Understanding some basic recommended standards for serial data communications-a comparison of rs-232, rs-422 and rs-485. URL: http://www. intersil. com/data/wp/WP0585. pdf. 2003.

[47] Buchanan WJ. RS-422, RS-423 and RS-485. InThe Handbook of Data Communications and Networks 2004 (pp. 627-631). Springer, Boston, MA.

[48] Marais H. RS-485/RS-422 circuit implementation guide. AN-960 Analog Devices. 2008.

[49] J. Browning, "engadget," So you got a Raspberry Pi: now what?, 04 September 2012.

[50] Reichelt, *RPI GPIO 2 RS232 Raspberry Pi converter, GPIO to serial.*

[51] Damania R, Desbureaux S, Rodella AS, Russ J. Quality unknown: the invisible water crisis. World Bank Publications; 2019 Sep 26.