# Examensarbete

**Elektroingenjörsprogrammet**

# ”Buddy Tracker”, an early warning system for recreational divers

**Martin Hautamäki**
**Elias Persson**

Handledare:
Rikard Ed-Svensson
vt 2010

# ”Buddy Tracker”, an early warning system for recreational divers

**Martin Hautamäki**
**Elias Persson**

**Examensarbete**
***Degree Project***
**Elektroingenjörsprogrammet**

**vt 2010**

**Handledare: Rikard Ed-Svensson**

Detta examensarbete omfattar 22,5 hp och ingår i Elektroingenjörsprogrammet, 180 hp, vid Karlstads universitet.
This 22,5 hp Degree Project is part of the 3 year, 180 point Electrical Engineering course at Karlstad University, Sweden

Denna rapport är skriven som en del av det arbete som krävs för att erhålla Elektroingenjörsexamen/Teknologie kandidatexamen. Allt material i denna rapport som inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

________________________________________

Martin Hautamäki

________________________________________

Elias Persson

-----------------------------------------------------------------------------------------

Rapporten godkänd,

________________________________________

datum Handledare: Rikard Ed-Svensson

________________________________________

datum Examinator: Peter Röjder

## ABSTRACT

In order to improve SCUBA diver safety this report will present the means for constructing a cheap and robust underwater tracking system that will monitor the distance between two SCUBA divers. The electronics is built around a microcontroller that calculates the time it takes for a signal to travel forth and back between the two SCUBA diver units. From this time, a distance is calculated, which then is compared with a preconfigured alarm distance. If the alarm distance is exceeded, then the unit will notify the SCUBA diver of the imminent danger. To achieve this functionality we have to clarify the problems associated with underwater wireless communication, in extremely shallow waters. The main issue is that communication through an underwater communication channel is restricted to ultrasonic signaling due to the high grade of absorption, when talking about electromagnetic signals. Another obstacle is the rich multipath environment that the environment presents. To overcome this we make use of on-off keying (OOK) together with guard slots to transmit the pulse code modulated (PCM) data signals by the use of narrow-band ultrasonic transducers.

# TABLE OF CONTENT

# 1 INTRODUCTION

## 1.1 Background

We, the authors, are both recreational divers. As a recreational diver, you are always diving together with another diver, in a pair [1]. This is for safety reasons, for instance if you run out of air, so that you can get some air from your dive buddy instead, while you start ascending to the surface. But what happens if you run out of air and your "dive buddy" is nowhere to be found? You and your dive buddy always have to stay in reaching distance of each other so you can get immediate help from each other if something happens [1]. This is not always that easy, and inexperienced divers tend to lose sight of each other to often. If you lose sight of your dive buddy while diving, you got one minute to find him/her before you will have to start the ascent to the surface to check for your dive buddy [1]. If nothing bad has happened you will probably find your dive buddy at the surface, but your dive is forfeited for this time, and you will have to plan a new one if you are going to dive more [1]. Both the ascent and that you have to wait and plan a new dive can take much time, and also be a waste of money if you are on a vacation and out with a dive boat. What if you could have prevented losing sight of each other in the first place? This is what we are going to find a solution to in the following parts of this work.

## 1.2 Task

To prevent the problem with losing sight of each other, we are going to try to develop a cheap electronic system that will help to prevent the divers from getting separated from each other. We face two main problems with this. One of them is being able to measure the distance between the two divers, and alert them if they start to get too far away from each other, so that they will notice this before getting out of sight of each other. The other problem is to make the system capable of communicating in the, for signals, hostile environment of water, so that both divers are alerted if they are about to swim away from each other. In addition, the system has to allow several units to co-exist in the vicinity without unnecessary crosstalk between the units.

## 1.3 Delimitations

The project will mainly focus on the development of the technical solution. It will contain information on suggested hardware solutions, software flow charts and other system related specifications. The information will be collected and distilled mainly by studying related articles and literature. We will not be presenting any prototypes due to the short development period of the product, but as mentioned, we will provide a good starting point for further development.

## 1.4 Aim

The aim of the project is to present a viable solution that will assist scuba divers during recreational dives in keeping track of their dive buddies. The system is built around a technique using ultrasonic sound waves that will open up for marine wireless communication between two units. All prompted solutions will be evaluated and chosen according to their implementation cost, and their ability to complete the given task. This in the purpose of keeping the product simple and cheap, but yet robust.

### 1.4.1 Desired system function

The final product will be encapsulated in waterproof aluminum casing. Its user interface will feature an on/off button, selectors for alarm distance, channel settings and a master/slave switch. The electronics are built around a microcontroller that calculates the time it takes for a package to travel forth and back between the transmitter and receiver. This time is then compared with the preconfigured alarm distance. If the alarm distance is exceeded, then the unit will notify the diver by activating an acoustic signal in combination with a flashing light.

## 1.5 Outline

The report is organized so that, it at first will present commercially available communication and tracking systems, which are related to diver safety. It will then give an understanding of the problems related to maritime communication and the properties of underwater acoustics. After this, the different technical block sets like modulation techniques and ultrasonic transducers are discussed. And from all this we will distil and present our suggestion for a working safety system. The hardware is presented by block sets explaining the functionality and purpose of each part of the system. The software is presented by flowcharts and clarifying texts. After presenting the results of the project we will discuss further enhancements to the system that are outside the scope of this project.

# 2 METHOD

## 2.1 Overview

In this section we present how we have worked with the project and how we have obtained the information that lead to our results, that will be shown later on in this report.

## 2.2 Techniques to transmit data in an underwater environment

When we first began this work, we got a tip from our tutor that white noise signals together with GCC (general cross correlation) could be used for distance measurements. We therefore started with researching these methods. During the research of this (and all other things we researched) which took several weeks, we read books, that we lend at the university library, and several pages on the internet to get a proper understanding of how they work and if we could utilize them for our needs. At this time we also looked at the NLMS (Normalized Least Mean Square) method.

Besides the signal to send, we searched for information about sending ultrasonic signals and other types of signals in water and compared them to each other. Then we did some research about different types of ultrasonic transducers and their specific properties, and made some calculations about sound propagation in water as well as oscillation times for the transducers.

After we had gotten a better understanding of the white noise and GCC methods, we continued by searching for, and reading, articles, books and web pages about the digital modulation techniques PCM and PPM. We also read about different digital keying methods such as FSK and OOK.

In addition, we looked for ways to synchronize the signal at the receiver and which data that should be in the signal, and calculated the times for pulses and guard slots.

## 2.3 Choice of microcontroller unit

In our aim with this work, we want to use a MCU (Microcontroller Unit) to measure distances under water. We therefore did some research about different MCU’s, in particular Texas Instruments DSP and Microchip dsPIC processors. Besides different properties and how advanced different MCU’s are, we looked at the price level, where to buy them and how to program them. After that, we made contact with manufacturers and suppliers to see if they were willing to gives us free MCU samples and programming equipment. We ordered some free samples of dsPIC processors from Microchip and asked our university if they could contribute with a programmer.

## 2.4 Development of hardware solutions

When developing the hardware solutions, we studied articles made on related work in combination with internet studies of existing technology. Then we zoomed in on the different block sets of the presented choices. This in order to better understand their functionality, and thereby draw new conclusions upon viable hardware solutions. The pros and cons of the presented choices were also evaluated. When we evaluated the transmitter board we also made calculations upon the needed signal strength of the transmitter. A PCB was also constructed for the programming of the chosen microcontroller. Furthermore, we looked at some possible solutions for the user interface. This section was written so that it would satisfy the aim of this project. At last, we looked at possible solutions on power supplies by surfing the internet, and also some encapsulation techniques that we distilled from looking at existing electronics that are aimed for underwater use.

## 2.5 Software development

When most of the hardware solutions development were finished, we started the development of the software solution. We divided this process in two steps and accounted for our hardware solutions to make it work as good as possible. The first step was to write down what we wanted the system to do, in pseudo code. In the second step, we created flow charts from the code. We then modified these charts quite a lot and divided them in different levels to avoid getting too complex charts. The development of this solution took considerable time as we had to rearrange a lot to make it function as intended.

# 3 UNDERWATER ACOUSTICS AND COMMUNICATION

This chapter will provide necessary information related to underwater communication. First of all we will look at existing products that are available on the market, and aids in diver safety. After this, we perform a brief presentation of related work that has been made in the area of underwater communication. Furthermore, the physics of underwater communication together with communication techniques are treated in the aim of clarifying the difficulties and opportunities that apply.

## 3.1 Existing technology

Commercially, few products on the market target recreational divers when it comes to systems that aid in getting separated dive buddies back together in the water. Most of the available systems are either some form of active sonar based systems or systems utilizing a pinger and a scanner. In Figure 1 we see a system constructed by Desert Star Systems, the "Dive Tracker SCOUT". It consists of two units, a receiver and a transmitter unit. The transmitter is a pinger unit that continuously sends out ultrasonic pulses that are picked up by the receiver. The receiver has a directional transducer that allows the user to locate the direction of the pinger by sweeping it around, searching for the highest signal amplitude. This signal amplitude also gives an estimate on the distance to the transmitter. [6]

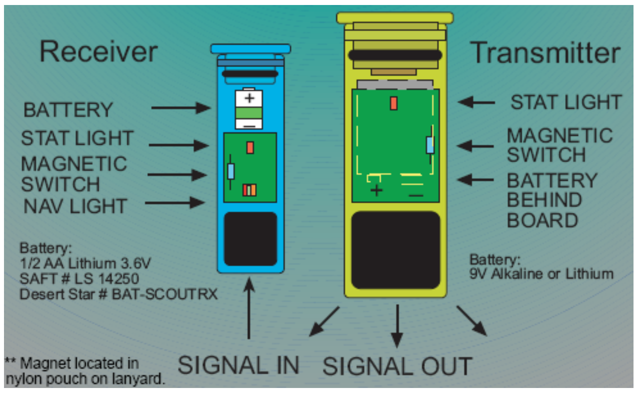

**Figure 1: Desert Stars Dive Tracker SCOUT system. [6]**

As mentioned, there are also some active sonar systems available on the market. For instance RJE-International provides the market with their DLS-1, Diver Locator Sonar. It's a hand held active sonar and pinger locator. It works the way that you point it in a direction, pull the trigger and an acoustic pulse is then sent in that direction. The time it takes for the signal to bounce back indicates the distance to the nearest object. Unlike the pinger and scanner system, here you will not get a confirmation on that the found object actually is your dive buddy, but in return you can use it for other tasks, like for instance depth and distance measurements. [7]

Both the above-mentioned systems aid in getting two separated divers together again in the water. However, they assist in solving the occurred problem, not to prevent it. To do this we need a system that monitors the distance between the two divers, and alerts them in case they are on the way of getting separated. For this we need some sort of communication between the two divers that monitors the distance. If we look at the available products on the market that make use of two-way communication we find underwater wireless voice communicators and acoustic modems. However, these products are not used for the purpose of diver safety, at least not in the same sense as what is aimed for in this project. Almost all these products make use of acoustic communication with ultrasonic transducers and Desert Star Systems "SAM-1 acoustic modem" is no exception, se figure 2 [6].

**Figure 2: Desert Star Systems SAM-1 acoustic modem. [6]**

The SAM-1 is an acoustic modem that is used to transmit sensor data, messages, status and control information underwater. Data is transmitted in serial with a speed of between 5 to 150 bit/s which often is sufficient when sending this types of data. The data is modulated with Pulse Position Modulation and to overcome the multipath effects of underwater acoustic transmission the system utilizes 4-channel frequency hopping in the interval 33.8 kHz to 42 kHz. The transmission power is between 183dB and 189dB depending on supply voltage. This gives the system a transmission range of up to 1000m, under good sea conditions. [6]

A system that actually measures the distance between two units in the water is the "Easytrak" provided by Applied Acoustics, figure 3. The system is an underwater positioning and tracking system that transmits and receives acoustic signals to subsea targets such as diver beacons. From which range, bearing and depth information can be determined. [8]

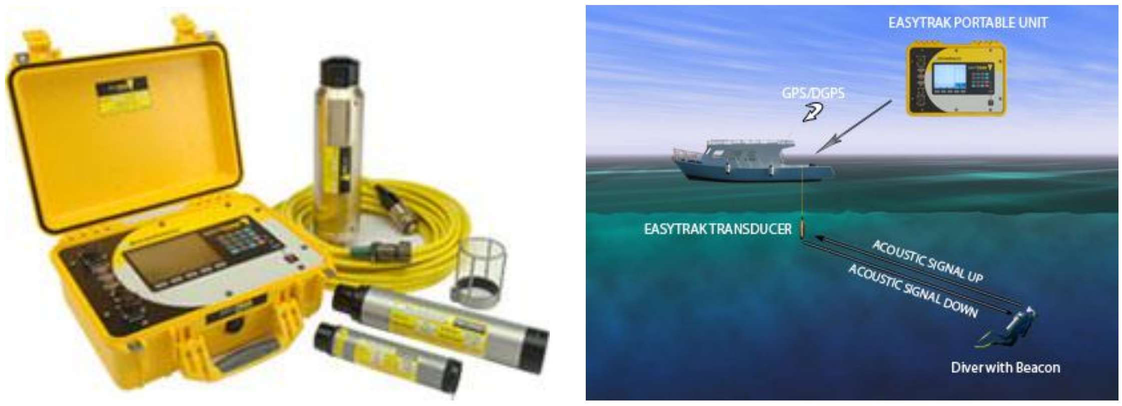

**Figure 3: Applied Acoustics Easytrak System. [8]**

It is the technique of using an Ultra Short Base Line (USBL) system for the transceiver that gives the system the ability to determine the bearing of the incoming acoustic signals. The USBL transceiver head contains three or more transducers separated by a baseline, and by measuring the phase difference, we can calculate the bearing of the subsea target [9]. The system has specifically been designed to track underwater targets that operate out of sight. So the system only gives information to the on board unit although two way signaling is used. The diver is "blind", but of course, this can be solved by adding under water communication devices that can give feedback to the diver. However, this unit is packed with features that aid in accurate tracking of the subsea target, but none of which we will look further upon [8]. Instead, we will take a look on related work that has been done in the area of underwater communication, and especially in the field of biotelemetry and also the above mentioned acoustic modems.

## 3.2 Related work

In this section, we will look further upon related work that has contributed and aided in the development process of our Buddy Tracker system. We will look at telemetry systems and acoustic modems. Telemetry systems are systems that are used to transmit measurements from a remote unit, not necessarily a wireless transmission but in our case that is true [9] [2]. Biotelemetry is a specific type of telemetry systems that are used to send medical data such as vital signs from divers [2]. This type of systems often make use of one way communication so they are not directly applicable on our system but on the other hand they are often constructed to meet stringent size, weight, cost and power constraints that we can take advantage of [10]. The knowledge about two-way underwater communication comes from the acoustic modems such as the "WHOI Micro-Modem", which has been developed by Woods Hole Oceanographic Institute in Woods Hole, USA, and of course, the on market SAM-1 acoustic modem mentioned earlier in this report.

### 3.2.1 An Ultrasonic Communication System for Biotelemetry in Extremely Shallow Water

At North Carolina State University a group at the department of electrical and computer engineering has developed an ultrasonic communication system for biotelemetry in extremely shallow water. The system is aimed to be used in the study of migration patterns of aquatic life. For this reason the system has to be small, power efficient, low weight and low cost, due to it is meant to be a disposable system. This aim for miniaturization is in line with our work and we will later on pick out details from their work and evaluate if they could be implemented into our system. But first we will provide an overview of their work. [10]

The group has developed a telemetry system that consists of a transmitter board that may be placed on an aquatic animal and a top side receiver board. Microcontrollers control both boards. The receiver board is constructed with less restrictive constrains when it comes to power consumption and size because of its top side placement. The system utilizes one-way communication of telemetry data. Therefore the transmission protocol of the system is a model without hand-shaking. This in combination with the challenges of underwater transmission raises the need for error correction coding (ECC). They tested two types of error correction. One 5-bit ECC, which could detect a one-bit error, and another, more complex algorithm, called convolutional-Viterbi scheme. At a range of 30 meters between the receiver and transmitter and at a baud rate of 42 symbols per second a transmission without error correction would be successful up to about 96%, according to Figure 4. An introduction of the 5-bit ECC would only increase the performance slightly while the use of Viterbi would result in a more or less error free transmission As we can see in Figure 4, if we set up a baud rate greater than 50 then we would get a dramatic decrees in the number of successful transmissions. [10]

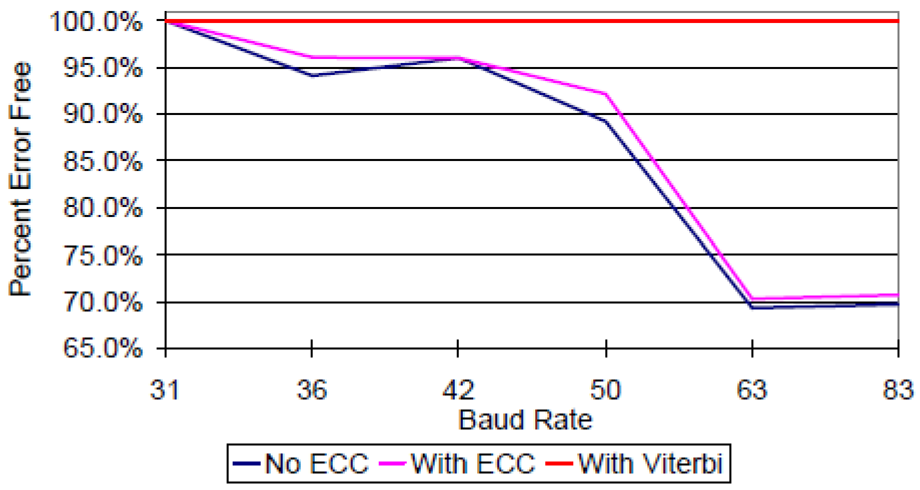

**Figure 4: Percent of Error Free Packets at 30 meters. [10]**

The transmission itself is established by using a ceramic transducer, powered by a transformer in combination with power MOSFET transistors to boost the voltage to 52.8V. All this produces a natural LC oscillator which frequency can be altered by adding capacitors in parallel with the transducer, se Figure 5. To transmit the information through the underwater channel they utilized a type of frequency shift keying, or the abbreviation FSK [10]. You can read more about FSK later on in this report but in short, FSK is a technique to transmit the information through a discrete number of frequencies that each represents a predetermined bit, symbol or sequence. In this case, they make use of four discrete frequencies [9].

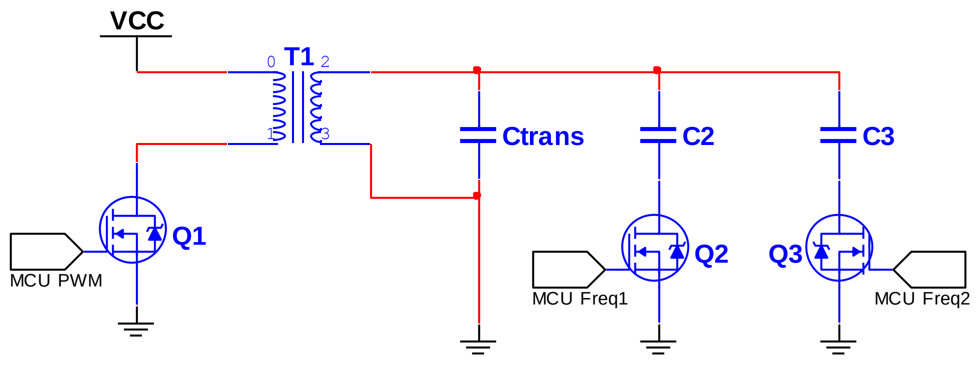

**Figure 5: Schematic of a four-frequency transducer driver.**

The receiver board is constructed so that the computational complexity of the microcontroller will be as low as possible. Therefore they have chosen a hardware solution when it comes to the process of reshaping the distorted signals. They have one pulse shaping unit for each of the four frequencies. These four signals are then feed to the AD-converter. And after that the four bit-streams are combined into one, and sequenced according to the package lengths. Also some error correction is applied as mentioned earlier. The drawback when adding external hardware is the rise in power consumption. However, this can be justifiable if it leads to a more robust system. [10]

The data in Figure 4 was collected from dives in a shallow lake, with a depth less than 5 meters and with no motorized activity in the vicinity. Also the weather conditions were moderate. So the test data is collected under quite good conditions. Conditions that you might not have under normal operating conditions. The practical baud rate might therefore decrease quite a bit for the non-corrected signals. For our work even a 50% reduction of the baud rate would still be sufficient to carry out the signaling tasks between the two units. Their packages are 32-bit long and with a baud rate of 42 it would take 768ms to transmit the package. That gives us 24ms for each bit. [10]

They also tested the system at a distance of 100m. However, for this experiment they boosted the transducer voltage to 87V and lowered the baud rate to 31. This resulted in that the receiver managed to correctly decode 67% of the data without the use of error correction. When switching on the Viterbi the percentage raised to 94%. [10]

The two microcontrollers used in this project were 8-bit AVR:s from Atmel named Atmega168, and Atmega324. Running at 8MHz respectively 20MHz [10]. Both of which accepted quite large supply voltage variations. For instance the Atmega168 accepting between 2,7V and 5,5V as supply voltage [11]. The faster microcontroller operated the receiver unit.

### 3.2.2 Microcontroller-Based Underwater Acoustic ECG Telemetry System

The "microcontroller-based acoustic ECG telemetry system" is built in the purpose of monitoring cardiovascular activity during SCUBA-diving and swimming activities. Swimming pools and extremely shallow waters produce a great amount of multipath signals that distort the transmitted signals. To overcome this problem the authors of this article have chosen burst-mode on-off keying (OOK) with guard slots as transmission schema. This method works for systems with low bit-rate, and if the guard slots are chosen correctly then the system will be more or less immune to multipath propagations of the underwater channel. The electrocardiogram (ECG) that is to be transmitted by the system is encoded with either of two data encoding schemas, pulse code modulation (PCM) or pulse positioning modulation (PPM). They also conduct a comparative performance analysis of the two encoding methods. In this comparison, it is shown that the PPM can provide ECG signals at higher bit rates than the PCM method. Another important fact in this article is the posted time durations for the guard slots. They were chosen between 15 and 20ms. This time was long enough, so that the multipath would have time to decay in the large indoor testing tank, measuring 9x5 meters with a depth of 2 meters. The transfer rate of the system was 50bit/s when transmitting PCM signals in combination with 20ms guard slots. [12]

The microcontroller used in this design was an 8-bit Intel 87C51 processor operation at 24MHz [2]. It had 4k-byte on board memory and 128-byte RAM [13].

### 3.2.3 WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms

"The WHOI micro-modem is a compact, low-power, underwater, acoustic communication and navigation system" [14]. The system is aimed to assist autonomous underwater vehicles (AUV) in multi-vehicle operations in shallow water. The general networking capability and the ability of the system to work with other systems such as REMUS navigation makes the system versatile. It communicates by utilizing four different techniques for transmission. However, we are only interested in the so-called frequency-hopping frequency-shift keying (FH-FSK) transmission technique. FSK is a robust underwater communication technique and when combined with FH it gains better resistance against multipath propagations that exist in shallow water environments. Default data rate of the system is rated at 80bps. Three modules, main board, power amplifier and floating-point coprocessor outline the system. You can see the main board and the underlying power amplifier in Figure 6. The coprocessor is only an expansion card designed to support the main board when using computational complex phase shift keying (PSK). [14]

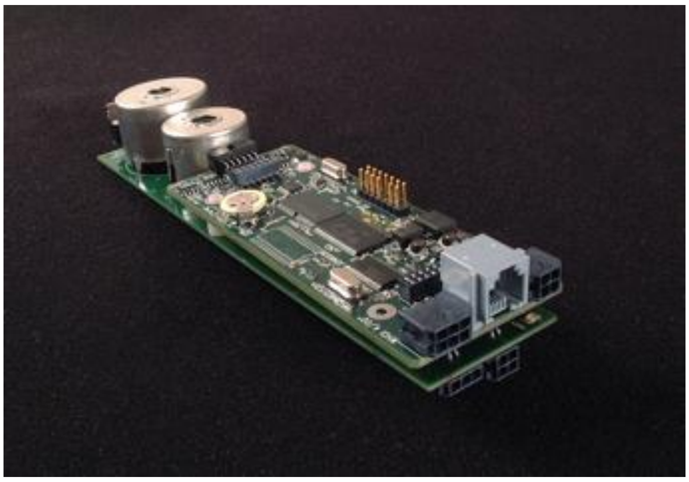

**Figure 6: The WHOI Micro-modem basic board set: power amplifier and main board attached. [14]**

The main board is built around a 16-bit digital signal processor (DSP) from Texas Instrument. This to cope with the high level of complexity in the chosen modulation techniques. The DSP provides up to 160 million instructions per second (MIPS). External 12-bit AD/DA-converters provide analog inputs and outputs to the processor. These are sampled at 80 kHz. Information is transmitted in serial within 3 to 30 kHz. Between the transmissions, the system has the ability to enter hibernation mode to save power. [14]

The power amplifier block is outfitted with a Class-D power amplifier to drive the ceramic transducer. The class-D amplifier was chosen because of its efficiency characteristics and the ease of matching it with different transducers. However, the output power is fixed. The amplifier also includes a transmitter/receiver switching network that allows the use of just one transducer for transmitting and receiving. A power-conditioning unit that provides power to the micro modem is also situated on this board. [14]

The software of the WHOI Micro-modem is extensive to make it compatible and easily reconfigured to fit the intended purpose. Standard frame size is 32-bit. However, the ability of sending 21-bit “Mini-packages” for various system functionalities is possible [14]. For instance, “Cycle Init” that sets up a transmission link between units. The “Ping” function that is used for distance measurements works as follows:

*“The modem sends a ping mini-packet to the specified node, which responds after a fixed turn-around time. The originating node subtracts the turn-around time and reports the one-way travel time to the user [14].”*

With their method they claim to get an accuracy within 125µs which is less than 0,2m in the water [14].

## 3.3 Underwater acoustics

Underwater acoustics presents many obstacles that we have to take in consideration when developing an ultrasonic communication device. Multipath propagation is a big issue when it comes to shallow water communication. Also the signal attenuation due to the chosen frequency needs to be considered. However, when it comes to signal attenuation, it is the geometrical spreading that stands for the most of the signal attenuation. The Doppler Effect together with the speed of sound in water presents even more parameters to consider. Here follows a presentation of the mechanisms that apply to acoustic signaling in water. [2]

### 3.3.1 Signal attenuation

When sending signals through different kinds of mediums, you have to take into account the effect of signal attenuation. To send signals through water, several types of signals could work. The first type that comes to mind is electromagnetic signals, which you usually use when you send signals in the air. Unfortunately, electromagnetic waves are not suitable for transmissions in water because its attenuation in water is much higher than in air[15]. If you were to use electromagnetic waves you would have to ”use very low frequencies (10 to 30 kHz) where attenuation is in the order of 3.5 to 5 dB per meter”[15] which would require very large antennas [15]. There are also two other types of signals that you can use, which are optical and ultrasonic (sound at frequencies higher than the human ear can distinguish) signals. Optical signals are for obvious reasons often not an alternative due to the very low visibility in some waters because of particles that would absorb the light within a few meters. Ultrasonic signals however, do not get affected that hard by the particles and the density of the water, and propagates much better in the medium than the two other signal types mentioned [2]. If you, for instance, would be sending an ultrasonic signal at 20kHz in average sea water and the temperature of the water is 10°C, the attenuation would only be between 2,5-3,4dB/km (depending on which formula you use)[16]. That means that if you were to send an ultrasonic signal at 20kHz in 10m (which is the longest distance we're going to measure) the signal would have only be attenuated by 0,025-0,034dB (at a depth of 10m and with a signal going in a straight direction), [16]. With a higher frequency of 40kHz, the attenuation would be about 0,087-0,12 dB in 10 meters[16].

The so called "geometrical spreading" has the biggest impact on signal attenuation in water. When talking about sound pressure the "inverse distance law" will give us that, a doubling of the distance to a point source will result in a 50% reduction of the sound pressure, i.e. a decrees of 6,02dB. Hence, the sound pressure is inverse proportional to the distance. However, the sound intensity follows the inverse-square law [9].

### 3.3.2 Multipath propagation

When sending signals in water, they will propagate in multiple directions, called multipath propagation. The signals will propagate and reflect on different things in the water like boats, the bottom and at the surface. This will create weaker time-varying reflections of the original signals that will mix with the original, non time-varying signals, causing intersymbol interference (ISI). Since the signals are reflected at both the bottom and the surface, there will be more reflections in shallow than in deep areas, and the interference will therefore be more severe. Because of this, the signals may become distorted, and the data will then be lost. [2]

### 3.3.3 Doppler Effect

The Doppler Effect is an effect that causes frequency variations in signals. The variations occur because of the movement between the source of the waves, and the observer. If the source and the observer move towards each other, the signal will get a higher frequency because the wavelength between the first wave and the second becomes shorter then if the source and the observer would have been stationary. The opposite happens when the source and the observer moves away from each other, the wave gets stretched, which causes a longer wavelength, and in turn a lower frequency. This effect affects all electromagnetic, ultrasonic and light waves. For sound however, the effect can also be caused by the medium that the sound propagates through. If the sound propagates through a non-solid medium and the medium moves relative to the source or the receiver of the signal, the frequency will vary in the same way as described when the observer and the source moved relative to each other. [17]

## 3.4 Ultrasonic transducers

An ultrasonic transducer is a device that converts electrical energy to ultrasonic vibrations and vice versa. It works as a combined microphone and speaker, but at higher frequencies than normal, frequencies that humans cannot hear [18]. Ultrasonic transducers are very useful when you want to both create and detect ultrasonic vibrations, as you can do both with just one transducer as long as you do not need to do both at the same time. An ultrasonic transducer is called a projector when sending and a hydrophone when receiving [19].

### 3.4.1 Different types of transducers

Ultrasonic transducers come in many forms and you usually divide them in two groups, mechanical and electromechanical transducers. Mechanical transducers are transducers that are driven by liquid, by pneumatics or by gas, while electromechanical transducers are of the types magnetostrictive or piezoelectric. The mechanical transducers are limited to low frequencies while electromechanical transducers can handle extremely high frequencies. [20]

### 3.4.2 Piezoelectric transducer

Piezoelectric transducers are built in materials such as quartz crystals and ceramics [18]. These transducers work much like capacitors. When connected to a high voltage they change their physical shape depending on the polarity of the voltage, and thereby stores an amount of energy [9]. When the voltage returns to its former level, the shape of the material will be restored [9]. This property works in both ways, meaning that if the transducers shape transform, the voltage over it will change [9]. If you connect an alternating voltage to a piezoelectric transducer, it will vibrate at the same frequency as the voltage alternates, and thereby emit an ultrasonic vibration with that same frequency [19]. The opposite also applies for the receiver. When the piezoelectric transducer begins to vibrate from ultrasonic vibrations, it then starts to emit an alternating voltage with the same frequency as the received acoustic signal[3][19].

## 3.5 White noise

White noise is a signal that consists of all frequencies from zero to infinity and has an equal amount of energy at all frequencies. True white noise can therefore not exist, as it would have an infinite amount of power. However, band limited white noise exists and consists of all frequencies within a limited frequency band, where the power of the signal is equal at any given frequency. [9] If you want to create a band limited white noise signal, you could do this in MATLAB, as you could then specify many properties of the noise [21].

### 3.5.1 GCC - Generalized Cross Correlation

Generalized Cross Correlation (GCC) is a method that could be use to find a known signal, for instance a white noise signal, masked by noise by comparing the incoming signal with the known signal you want to identify. When comparing the signals, you multiply them with each other and then calculate the integral of the product. The value you get will get higher the more similar the signals are to each other. By sliding one of the signals in the time domain the integral will eventually get a maximum and when that happens, you will have found your signal.[9][3]

## 3.6 Pulse modulation techniques

When you want to send a signal, analog or digital, you may want to, but sometimes also need to, modulate the signal to a different form to be able to send it. Several techniques exist for this purpose and here we present two pulse modulation techniques that we have taken a closer look upon, pulse code modulation (PCM) and pulse position modulation (PPM).

### 3.6.1 PCM – Pulse Code Modulation

Pulse code modulation is a technique for representing analog signals in a digital form. You do this by sampling the analog signal and then store every sample as binary data. You can then send the data by using keying techniques such as frequency shift keying (FSK) and on-off keying (OOK), which we describe later.

When you are sending a sample with PCM, it takes *t* seconds to send one bit, and one whole sample then takes *n* times that of a single bit when a sample consisting of *n* bits. PCM is used as standard for digital audio in computers [9]. Here the environment presents little, or no multipath signals. An example of a 3-bit PCM transmission is seen in Figure 7.

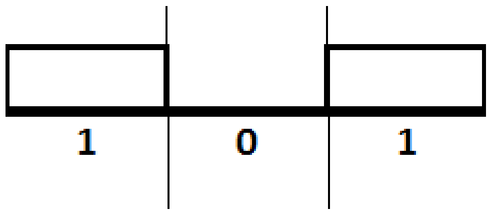

**Figure 7:** A PCM signal consisting of three bits with the value five.

A shortcoming for this modulation technique is that if you expose the signal to multipath propagations, the first bits will interfere with the following bits and cause intersymbol interference (ISI), which could lead to that the communication fails [2][9]. In shallow water, which is a rich multipath environment the PCM would suffer greatly due to the ISI introduced, unless used in combination with guard slots or some other keying technique that will be presented later. For wireless communication with little multipath the PCM has a higher transmission rate than PPM, which we will talk about in the next section. [2]

### 3.6.2 PPM – Pulse Position Modulation

Another pulse modulation technique, much like the PCM, is PCM. In this technique, you have digital bit representations of the data you want to send, and then send the data in a stream just like with PCM. The difference here is that instead of sending the bits one-by-one, you are sending the whole value that the bits represent with one pulse. The position of the pulse in time is what determines the value of it. To accomplish this you will have to use eight timeframes to be able to represent a 3-bit binary value from zero to seven, while this only takes three timeframes in PCM. If you look at Figure 8 you can see a PPM signal with the value five.

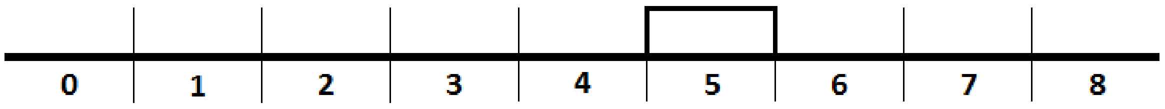

**Figure 8:** A PPM signal with one pulse representing the value five.

If you want to be able to send higher values, you have to add one extra timeframe for every value you want. Because of this, the signal will grow in length very quickly. Multipath propagation just as with PCM also affects this modulation technique, but PPM is less susceptible to this, because when the receiver picks up the first pulse it can ignore any pulse that comes in the remaining time of the frame. [2][4]

## 3.7 Data Keying Techniques

Although most communications today are digital, the signals are still analog when transmitted wirelessly due to the analog channel the signals travel through, such as air or water [5]. To send digital signals through these kinds of channels you can key the digital levels to varying physical properties in the analog signal. When the receiver receives the signal, it can convert the signal back to its original digital representation. There are many ways of doing this, which of we will present two, namely Frequency Shift Keying (FSK) and On-Off Keying (OOK), presented next. [9]

### 3.7.1 FSK – Frequency Shift Keying

Frequency Shift Keying is a method of sending data by keying the logical levels to different frequencies, which the transmitter then jumps between when sending. This means that to implement FSK you need to be able to transmit on at least two different frequencies (for binary signals) and the signal could then look like the one in Figure 9. The more bandwidth you have, the more frequencies you can use. One benefit of being able to use several frequencies is that you can key more levels to frequencies, and the transmission can therefore get faster. One disadvantage with this method is that you need to have enough bandwidth to spread the frequencies apart so that you can distinguish them from each other. [22][9]

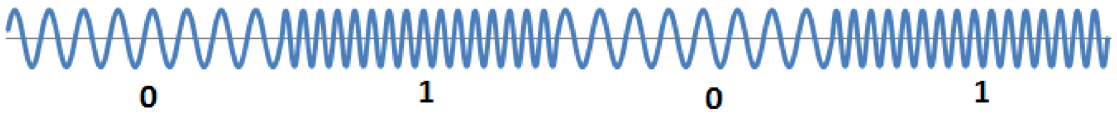

**Figure 9:** Binary data keyed with Frequency Shift Keying.

When data is to be transmitted in water, FSK is frequently used due to its good properties, one of being more resistant to multipath propagations in shallow water than modulation techniques such as PCM and PPM. Another is the increased transmission rate and also the relatively simple implementation when compared to even more complex techniques such as differential phase shift keying (DPSK). [2]

### 3.7.2 OOK – On-Off Keying

OOK is a very simple method that you can use instead of FSK. In this method, you only need one frequency, the carrier frequency, because the presence of the carrier frequency presents a logical one and the lack of the carrier frequency presents a logical zero. This can be seen in Figure 10. [23]

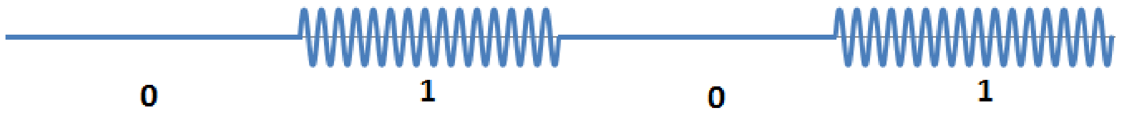

**Figure 10:** Binary data keyed with on-off keying.

Advantages with this method are that it is simple to implement, you only need to have one frequency available, and it is more power efficient than FSK due to idling when sending logical zeros. Disadvantages with the method are that it only can key two logical levels and because the lack of a signal represents one of these levels, it gets harder to detect transmission losses than with FSK where there are signals for all levels. [23]

### 3.7.3 Guard slots

Guard slots is a method you can use together with keying, but it is not limited to the keying techniques described above. The method is very useful when you want to send a signal that does not heavily depend on the transmission speed through a reverberant channel. Guard slots are empty timeslots in the signal, and when a guard slot appears in the signal, the transmitter and the receiver are idling to let reverberations fade away in order to get rid of unwanted ISI, before continuing the transmission. This works well as long as the transmitter and the receiver synchronize with each other properly. [2]

# 4 SELECTION OF MEASUREMENT TECHNIQUE

We begin with calculating the speed of sound in water, which will then be used when we continue with presenting how we want to measure the distance between the two underwater units. We then present why we chose to use ultrasonic communication and what type of transducers that can be used, and why. The data packages that need to be sent between the units will also be explained. All this will be followed by the evaluation of the available transmission techniques and data modulation techniques.

## 4.1 Speed of sound in water

When we measure the distance between the two units we are sending out ultrasonic signals in the water and measure how long time it takes for the signals to travel to and from the receiving unit. When we then calculate the actual distance between the units, we need to know at which speed the sound waves travel in the water.

To calculate the speed of sound in seawater, we used Wilson W. D.s formula, which is accepted by the National Oceanographic Data Center in the USA for computer processing of hydrological information [24]. When we used this formula, the resulting speed depended on three parameters, the temperature, the hydrostatic pressure and the salinity of the water. If any of these parameters would go up or down, so would the speed. Because of this, we decided to calculate three different speeds to compare and see if they differ much, or if the differences are small enough to neglect. The three speeds we decided to calculate and compare was the average, the lowest and the highest speed. To do this we needed to know the average, lowest and highest values for the parameters. The lowest temperature for the water that we used was 4°C as the water at the bottom of lakes almost never goes below that temperature, and if it is that could at the surface you usually do not dive when you are a recreational diver [9][2]. The average temperature of seawater is 17°C[25]. The highest temperature we used were 30°C because the formula is not made for temperatures higher than that, and the water seldom goes above that temperature. When you dive recreational dives you usually dive to at least 5m, but you never go below 30m [2]. If we then assume that the average depth would be that centered between 5m and 30m we get a depth of about 18m, which also is the depth limit when you are not an advanced recreational diver [2].

$$1kg/cm^2 = 0{,}09807MPa\ [6]$$
$$1kg/cm^2 / 9{,}75m\ in\ sea\ water\ [27]$$
$$1kg/cm^2 / 10m\ in\ fresh\ water\ [27]$$
$$9{,}75m + (\frac{10m - 9{,}75m}{2}) = 9{,}875m \quad \Rightarrow \quad 1kg/cm^2 / 9{,}875m\ in\ average\ water$$
$$1kg/cm^2 / 9{,}875m\ in\ average\ water = 0{,}09807MPa / 9{,}875m\ in\ average\ water$$
$$Hydrostatic\ pressure\ at\ different\ depths:$$
$$5m: \quad 5m(0{,}09807MPa / 9{,}875m) \approx 0{,}050MPa$$
$$18m: \quad 18m(0{,}09807MPa / 9{,}875m) \approx 0{,}179MPa$$
$$30m: \quad 30m(0{,}09807MPa / 9{,}875m) \approx 0{,}299MPa$$

**Figure 11:** Calculation of hydrostatic pressure at different depths.

The resulting hydrostatic pressures at the different depths that we calculated are 0,050MPa at 5m, 0,179MPa at 18m and 0,299MPa at 30m (Figure 11). When we had both the temperatures and the hydrostatic pressures, we also needed the salinity levels. The average salinity level in the seas is 35psu (‰)[25]. The lowest level of salinity is in lakes, where we assume that it is 0psu (‰). The highest salinity level that the formula allows is 37psu (‰) so that is what we used for the calculation of the highest speed, even though some waters have a higher salinity level (like the dead sea which have a very high salinity level of 210psu)[26]. The resulting speeds were 1467m/s, 1514m/s and 1546m/s. The difference between these speeds are slight, and to compare them we calculated how long time it would take sound to travel different distances at the different speeds. As seen in Table 1, the time it takes to travel a distance between two and ten meters takes almost the same time at all three speeds. [24]

**Table 1:** The time it takes sound to travel different lengths with various speeds in ms.

| **The time for sound to travel at the different speeds in ms** | | | |
|---|---|---|---|
| Meters to travel | at minimum speed 1466,78m/s | at average speed 1513,67m/s | at maximum speed 1546,15m/s |
| 2 | 1,4 | 1,3 | 1,3 |
| 3 | 2,0 | 2,0 | 1,9 |
| 4 | 2,7 | 2,6 | 2,6 |
| 5 | 3,4 | 3,3 | 3,2 |
| 6 | 4,1 | 4,0 | 3,9 |
| 7 | 4,8 | 4,6 | 4,5 |
| 8 | 5,5 | 5,3 | 5,2 |
| 9 | 6,1 | 5,9 | 5,8 |
| 10 | 6,8 | 6,6 | 6,5 |

## 4.2 Distance evaluation

When we measure the distance, we do this by measuring the time it takes for the sound to travel to the other unit and back again. We subtract the response time from the measured time and divide it by two to get the time it takes for the sound to travel in one direction. When we then have the travel time, we need to decide the distance that the time represents. Because the speed of the sound in water varies with changing parameters, which we saw in chapter 4.1, the time to travel a distance will of course also vary a little, which we saw in Table 1. To be able to decide which distance a measured time equals to, we made a new table, Table 2, which consists of Table 1 and four extra columns for minimum and maximum time and corresponding distance intervals. To decide the minimum and maximum time limits in Table 2, we calculated them by using equations 4-1 and 4-2. When the system then determines the distance between the units, it checks the measured time against these columns and hence obtain the distance. The two columns for the distance intervals show in which interval the real distance is. As you can see in these columns, the values overlap each other slightly. This overlapping occurs because of the variations in the sound speed in water. When the system approximates a distance, it might approximate it to different distances in different environments depending on the temperature, salinity and hydrostatic pressure of the water. This is however not a big problem as long as the users are aware of this behavior and keep it in mind, if they use the equipment in environments where any of these parameters differ much from the average.

$$f(x) = b(x-1) + \frac{a(x) - b(x-1)}{2}$$

**4-1**

$$g(x) = b(x) + \frac{a(x+1) - b(x)}{2} - 0{,}1$$

**4-2**

$$\begin{aligned}
&x = distance\ in\ metres\\
&f(x) = Min.\ time\ limit\ for\ x\ metres\\
&g(x) = Max.\ time\ limit\ for\ x\ metres\\
&a(x) = Min.\ travel\ time\ for\ x\ metres\\
&b(x) = Max.\ travel\ time\ for\ x\ metres
\end{aligned}$$

**Table 2:** Sound speed in water together with time limits and distance intervals.

| **The time for sound to travel at the different speeds in ms** | | | | **Time limits** | | **Distance intervals** | |
|---|---|---|---|---|---|---|---|
| Meters to travel | at minimum speed 1466,78m/s | at average speed 1513,67m/s | at maximum speed 1546,15m/s | Min (ms) | Max (ms) | Min (m) | Max (m) |
| 2 | 1,4 | 1,3 | 1,3 | 1,0 | 1,6 | 1,5 | 2,5 |
| 3 | 2,0 | 2,0 | 1,9 | 1,7 | 2,2 | 2,4 | 3,5 |
| 4 | 2,7 | 2,6 | 2,6 | 2,3 | 2,9 | 3,4 | 4,5 |
| 5 | 3,4 | 3,3 | 3,2 | 3,0 | 3,6 | 4,4 | 5,5 |
| 6 | 4,1 | 4,0 | 3,9 | 3,7 | 4,2 | 5,4 | 6,5 |
| 7 | 4,8 | 4,6 | 4,5 | 4,3 | 4,9 | 6,3 | 7,6 |
| 8 | 5,5 | 5,3 | 5,2 | 5,0 | 5,6 | 7,3 | 8,6 |
| 9 | 6,1 | 5,9 | 5,8 | 5,7 | 6,2 | 8,3 | 9,6 |
| 10 | 6,8 | 6,6 | 6,5 | 6,3 | No limit | 9,3 | No limit |

## 4.3 Acoustic signals as information carrier

When sending signals you can choose between several types such as electromagnetic, sound or optical signals. As stated in chapter 3.3.1 electromagnetic waves are not suitable because of its high attenuation in water, and optical signals would attenuate too fast when there are many particles in the water. We are therefore left with the ultrasonic signal type, which propagates pretty well in water and does not get affected by the particles in the same way as optical signals. Most underwater equipment also use this signal type when communicating, as seen when you read about existing technologies and related work in chapter 3.1 and 3.2.

When sending and receiving ultrasonic signals we choose to use a piezoelectric ultrasonic transducer (see chapter 3.4) which can both transmit and receive ultrasonic signals and is made for high frequency signals. When looking at these kind of transducers we notice that the price gets considerably higher if we chose transducers with higher bandwidths (several kHz and more) than about 1-2 kHz, which are much cheaper [28]. As our aim is to develop a cheap and simple system, we choose to utilize cheap transducers, and therefore we have limited bandwidth.

Nevertheless, we cannot just choose cheap transducers; they have to be waterproof too. There are transducers that are both cheap and waterproof [29] which could fit our needs, but it might also be possible to make non-waterproof transducers waterproof. In [30] they are making simple waterproof hydrophones by rubber molding piezoelectric elements into plastic "eggs". The speed of sound in the rubber they use is 1430m/s. Another method they use is to mount the piezoelectric element into a PVC pipe that they fill with canola oil and then seal is. The sound speed in the oil they use is 1440m/s. The speed of sound in both materials that they have used when building these waterproof hydrophones are very near the speed of sound in water that we calculated in chapter 4.1, but will probably have a very low impact on the sound waves when measuring their travel time. If the impact gets to big though, the difference could be accounted for in the software. If these techniques are viable alternatives to the waterproof transducers we do not know as they only made hydrophones for receiving sound waves and not for the transmission of them. We find these ideas interesting though and would gladly have tested them to try to make our own waterproof transducers if we would have had the time.

Another thing that we have to consider when choosing transducers is the angel of which the transducer can receive and transmit. This is something that varies greatly between different manufacturers. Our goal is to have a transmitter and receiver with an omnidirectional signal pattern. Therefore, if the angle is too small we might be forced to use several transducers in parallel, aimed in different directions.

### 4.3.1 Pulse calculations for piezoelectric transducers

The time, during which you apply the voltage to the transducer, we call $\tau_{burst}$. This time consist of $\tau_{\min}$ and $\tau_{pulse}$. When you apply the voltage to the transducer, it needs some time to build up the amplitude of the transmitted signal [2]. This build-up time we call $\tau_{\min}$ and in order to get correct oscillations it is the minimum time we need to apply the voltage to the transducer. We then need to keep the voltage applied during the time $\tau_{pulse}$, for the transducer to send a whole pulse at its intended strength. This to ensure that the maximum distance will be reached. To calculate these times there are some formulas that you can use which are 4-3, 4-4 and 4-5 where *N* is the number of periods during the pulse, $f_0$ the transmission frequency, $BW_T$ the transducers bandwidth and *Q* the number of periods needed during $\tau_{\min}$ [2].

$$\tau_{\min} = \frac{Q}{f_0}$$

**4-3**

$$\tau_{burst} = \frac{N}{f_0}$$

**4-4**

$$Q = \frac{f_0}{BW_T}$$

**4-5**

### 4.3.2 Narrow band piezoelectric transducer testing

To test the characteristics of a narrow band piezoelectric transducer we conducted a brief test with two EC4016 transducers from Sencera Co. Ltd, that were available at the university. The test set up was made up by one transducer that acted as transmitter and therefore was connected to a signal generator. The signal generator was set to produce a 40kHz sine wave with 10Vpp. The receiving transducer was connected to an oscilloscope in parallel with a 3,7kΩ resistor, similarly to the test setup in the transducer data sheet [29].

The received sine wave signal was greatly reduced, and at a distance of about 30cm the peak to peak voltage level had dropped to the magnitude of millivolts. When changing the signal type to square wave for the transmitter we noticed a slight amplification of the amplitude on the receiver side. However, the oscilloscope was still showing a pure sine wave.

The bandwidth of the transducers was estimated to be somewhere around 1,5kHz.

## 4.4 Transmitted data

When measuring the distance between the units, we will send ultrasonic signals to measure the time that it takes for the signals to travel back and forth between the units. However, these signals will not only be ultrasonic signals, they will contain data that will be essential for the system functionality. The data that the packages will consist of is the ten bits seen in Figure 12. The first four bits (seen from the right) are for synchronization [2], the following four bits are for channel ID, the ninth bit is for the alarm and the final bit is for parity check.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| P | A | $ID_3$ | $ID_2$ | $ID_1$ | $ID_0$ | $S_3$ | $S_2$ | $S_1$ | $S_0$ |

**Figure 12: Signal data.** $S_0$-$S_3$ are synchronization bits, $ID_0$-$ID_4$ are the channel ID, A is for the alarm and P is the parity check bit.

### 4.4.1 Synchronization

When one unit shall receive a signal, it has to be able to identify the beginning of the signal to be able to interpret it correctly. In asynchronous RS232 serial communication between computers where the communication takes place in a cable, the synchronization consists of a single bit that informs the receiving part that data is coming [31]. This method would probably have worked in our communication if we where to use a cable as the medium to communicate through. However this is not the case as we communicate wirelessly through the water, where noise and other signals easily can interfere with our signal and those make the receiving unit miss the synchronization bit and act as if one of the following non-synchronizing bits were the synchronizing bit. To overcome this problem we choose to implement a solution much like the one used in [12] where they send seven synchronization bits, all having the value of one, which the receiver must receive to be able to identify the beginning of the signal and prepare for incoming data. In our solution we also send several synchronization bits, but instead of sending seven bits which all have the value one, we choose to send three bits with the value one followed by one bit with the value zero. The first three bits has the same function as described with the seven bits above, the receiving unit must identify three incoming bits, which all have to be ones, to identify the beginning of the signal. If the receiving unit misses the first one or two synchronization bits, and thinks of the second or third bit as the first, the fourth bit will interrupt the synchronization process and ensure that the unit will not try to synchronize using other data bits that follows the synchronization bits. If the receiving unit would miss all of the synchronization bits and try to synchronize with the following data (which we do not think is likely to occur), the unit will most likely fail to identify three ones and a zero followed by a correct channel ID. When failing, the unit will restart the process of finding a signal and try to synchronize again.

### 4.4.2 Channel ID

The channel ID in the package, is an ID that the receiving unit compares with its own to check if it should react on the package or not, because the package could be destined for another unit nearby, or it could originate from a whole other type of system. The ID consists of four bits, which gives us $2^4$=16 different channels. This gives the possibility to use 16 pairs of units in the same area as long as not all transmit at the same time. This has to be regulated by the software.

### 4.4.3 Alarm

The alarm bit in the package is a bit that the sending unit uses to inform the other unit of its alarm status. A value of one means that the units alarm is on, while a value of zero means that it is off. We will explain more about how this works in chapter 6.2.2.

### 4.4.4 Parity

This bit is for a simple parity control of the signal and it will help to ensure that the received signal is correct. The unit does this by checking if there is an even or odd number of ones in the data received prior to the parity bit, and then compares it with the parity bit. An even number of one's correspond to the parity bit being a zero while an odd number of one's correspond to it being a one. This is called even parity, and it is able to detect an odd number of incorrectly transmitted data bits. [32]

## 4.5 Transmission technique

To transmit our package containing the data mentioned in chapter 4.4, we can use several techniques mentioned in chapters 3.5-3.7. We first considered sending white noise, which we ran some simulations on, but then realized that it would be better to use one of the keying methods instead.

### 4.5.1 White noise signaling

When we considered using band-limited white noise (chapter 3.5) as our signal, we thought about using white noise generated in MATLAB. In this way, we would have a unique signal, which we would know the content of, to transmit and the units could have individual signals so that several pair of units could operate nearby each other, without risking receiving the wrong signal. To identify the signal, we could use general cross correlation described in chapter 3.5.1. There are several disadvantages with this method of using white noise though. First off all, this method will consume more power than other methods that we have looked at. This is because general cross correlation includes some heavy calculations [3]. Second is that we cannot send different data with the white noise as there is no way of adding data to it. If we would want to send data in any form, we would need to have different white noise signals that would have to represent different data. This would force the receiver to be able to identify more than one signal, and the computational demand would thereby be much higher, resulting in higher power consumption. Another disadvantage is that the implementation would be more complex than the methods we will present in the following chapters (4.5.2 & 4.6), considering the software implementation for the MCU. The biggest disadvantage though, is that to send a band-limited white noise signal (which as stated in chapter 3.5, consist of all frequencies within the band), we would need a transducer with a wide bandwidth that could switch between all the frequencies without any delay. As mentioned in chapter 4.3, we intend to use cheap transducers, which have a very limited bandwidth, causing the white noise method to fail. Moreover, considering that the transducer would have to be able to switch between different random frequencies without delay, we do not think this method is possible with any piezoelectric transducer due to the signal buildup time, $\tau_{\min}$ (presented in chapter 4.3.1), of the transducers.

### 4.5.2 Comparison of OOK and FSK

As we did not find the method to send band limited white noise signal a viable solution (chapter 4.5.1), we need another way of sending our signal. OOK and FSK that we presented in chapter 3.7.2 and 3.7.1 are two alternatives that we consider instead, as they might suit our needs. Both methods have their advantages and disadvantages, which we present in Table 3.

**Table 3:** A comparison of the keying methods OOK and FSK.

| | **OOK** | **FSK** |
|---|---|---|
| **Minimum number of frequencies needed** | One | Two |
| **Sensitivity to multipath interference** | More | Less |
| **Difficulty of implementation** | Simple | More complex |
| **Detection of signal loss** | Hard | Easy |
| **Logical levels** | 2 | Limited by bandwidth |

Looking at the table, we see that OOK only needs one frequency while FSK needs at least two. Knowing that the ultrasonic transducer that we will use has a very limited bandwidth OOK looks much more promising for us. If we where to use FSK we would need a more expensive transducer with a wider bandwidth. However, instead of using a transducer with a wide bandwidth, another alternative would be to use several cheap transducers with different frequencies. None of these alternatives is suitable for us as they are both more expensive than using OOK and the second alternative would be too complex to implement. Instead, we will use OOK and try to get around the biggest disadvantage that it has, namely its sensitivity to multipath interference. This can be handled by adding guard slots, mentioned in section 3.7.3, to the OOK and that is the method we will use. The guard slots will then make the system immune to multipath propagations.

## 4.6 Modulation technique

While we were looking into different ways of sending and modulating signals we looked at the two modulation techniques PCM (chapter 3.6.1) and PPM (chapter 3.6.2). PCM, as stated in chapter 3.6.1, is mainly a technique to store samples of analog signals in a simple digital form for easy transmission via digital communications. As our signal is digital we can say that it already is in PCM format. Nevertheless, an alternative to the PCM technique could be to send it with the PPM technique. We will therefore compare the PCM technique with the PPM technique to see if we should convert our signal to the PPM format instead. PPM is much like PCM, but it gets slower as the demands to send higher values arise. When sending data with PCM you send the bits in a stream and the signal would then look like the first signal in Figure 13. When sending the same data with PPM the signal would in this case look like the second signal in the same figure and as you can see, PPM needs five timeframes more to transfer the same data as PCM transfers in three.

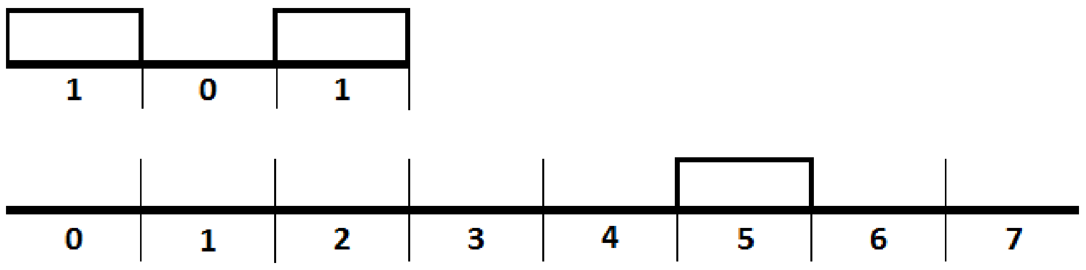

**Figure 13:** Signal with the value five in PCM (the first signal) and PPM (the second signal).

Although the PPM technique is slower, it has the advantage of being less vulnerable to multipath propagation interference than the PCM technique due to the transmission of only one pulse, which makes it possible to ignore any pulse arriving after the first. The PPM technique is however not immune to multipath propagation interference. If the pulse when sending one value comes in one of the last timeframes, its multipath propagation could spread to the timeframe for the next value and interfere with that instead, causing a communication error.

As both PCM and PPM has problems with multipath propagation interference, although PPM has not got that big of a problem as PCM, we need to try to reduce this problem to a minimum. For that reason, we introduce the method of using "guard slots" presented in chapter 3.7.3. By using guard slots in the transmissions, we introduce time for the multipath propagations to fade away before continuing the transmission, making it immune to multipath interference. The signals in Figure 13 would then look as in Figure 14.

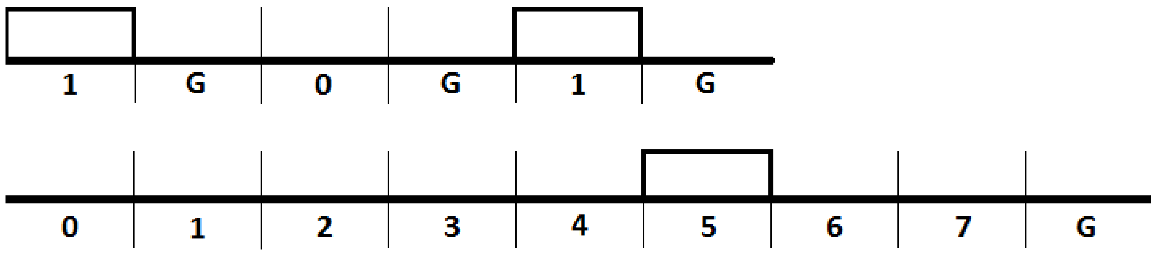

**Figure 14:** Signal with the value five in PCM (the first signal) and PPM (the second signal) together with guard slots represented by the G’s.

As you can see, we need to have a guard slot between every bit in the signal when using PCM while we only need to have one guard slot at the end of the signal when using PPM. Since the PPM technique only needs one guard slot independent of how long the signal is, while the PCM technique needs one guard slot for every bit, the PPM technique could be faster than the PCM if the time that a guard slot takes is long enough. To be able to determine which technique that will be the fastest for us, we need to know how long time the pulses and guard slots takes.

In order to calculate this time, we use the formulas 4-3, 4-4 and 4-5 presented in chapter 4.3.1 to get $\tau_{burst}$ which is the time we seek. In these formulas, we need the transmission frequency and the bandwidth for the transducer. We therefore used the values of the transducer 400EP18A [28] which are $f_0$=40 kHz and $BW_T$=2,0 kHz. Using these values, we got a $Q$ of 20, Figure 15. To get a signal burst with about 20 correct oscillations, which should be enough, we therefore choose a value of $N$=40. $\tau_{burst}$ then became 1ms where the first 0,5 ms is made up by $\tau_{\min}$ and the remaining 0,5 ms is $\tau_{pulse}$.

$$Q = \frac{f_0}{BW_T} = \frac{40kHz}{2kHz} = 20$$

$$\tau_{\min} = \frac{Q}{f_0} = \frac{20}{40kHz} = 0{,}5ms$$

$$\tau_{burst} = \frac{N}{f_0} = \frac{40}{40kHz} = 1ms$$

**Figure 15:** Calculations of $Q$, $\tau_{\min}$ and $\tau_{burst}$.

We now have the time $\tau_{burst}$ = 1ms which we sought. With this time, together with the duration of a guard slot, 20ms [12], we can now calculate the speeds with the formulas 4-6, 4-7 and 4-8 where $\tau_{sync}$ is the time for the synchronization bits and $B$ is the number of data bits.

$$\tau_{sync} = 4(\tau_{burst} + \tau_{guard})$$

**4-6**

$$T_{PCM} = \tau_{sync} + B(\tau_{burst} + \tau_{guard})$$

**4-7**

$$T_{PPM} = \tau_{sync} + 2^B \tau_{burst} + \tau_{guard}$$

**4-8**

Using these formulas, we get the times in Figure 16 to send our package consisting of six data bits.

$$\begin{aligned}
\tau_{sync} &= 4(\tau_{burst} + \tau_{guard}) = 4(1ms + 20ms) = 84ms \\
T_{PCM} &= \tau_{sync} + B(\tau_{burst} + \tau_{guard}) = 84ms + 6(1ms + 20ms) = 210ms \\
T_{PPM} &= \tau_{sync} + 2^B \tau_{burst} + \tau_{guard} = 84ms + 2^6 * 1ms + 20ms = 168ms
\end{aligned}$$

**Figure 16:** Calculations of $\tau_{sync}$, $T_{PCM}$ and $T_{PPM}$.

As you can see, the transmission is 42ms faster with PPM than with PCM. This is a considerable difference in speed, but we do not think that it is huge, and because of that, something that we do not consider important as the system is not that dependent on the speed of the transmission. Beside this difference we cannot see that either one of the methods have any advantage over the other when they are combined with the Guard Slot-method, and as our signal already is in the PCM format we therefore choose to let it stay that way, choosing the PCM method.

# 5 HARDWARE DESIGN

## 5.1 Overview of hardware block sets

To find the best hardware solution to our system we first have to identify the system block sets that outline our system. The aim is to minimize the number of external components in order to minimize power consumption, but at the same time try to keep the system simple and robust [10]. In Figure 17 we see six main blocks that we have identified.

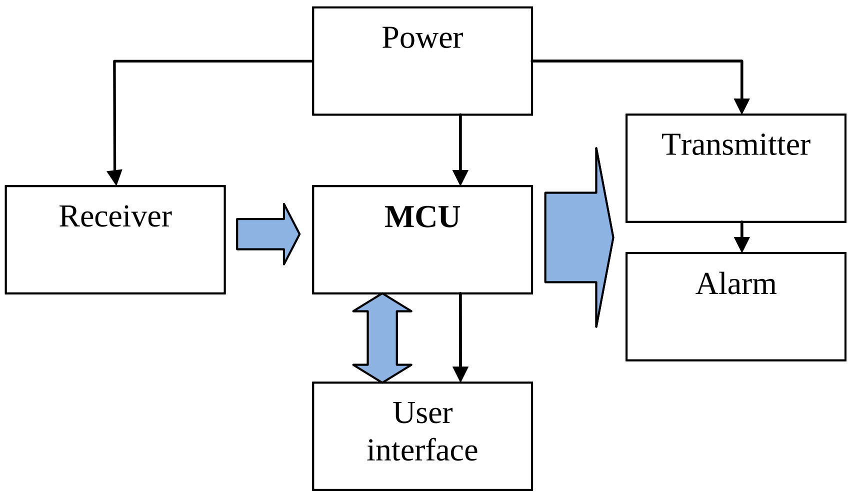

**Figure 17: Overview of hardware block set.**

The microcontroller (MCU) processes and coordinates information to and from the other blocks. The receiver unit transforms and reconditions the acoustic waves into electrical signals before they are sent to the MCU for AD-conversion. After processing the data, the MCU makes use of the DA-converter to pass on information to either the transmitter or alarm block set, depending on the information received, or in case it does not receive any data. The transmitter unit boosts the signal voltage before the transducer sends the ultrasonic, burst mode pulse. In case of an alarm event the alarm block set will notify the user of the current condition. As mentioned earlier, the user interface is made up by three parameters that can be altered by the user. The on/off button, selectors for alarm distance and channel settings.

## 5.2 Transmitter

There are several different approaches, which can be used when implementing the transmitter block set to the Buddy Tracker unit. The main goal here is to get a strong enough signal so that it reaches the receiver at its maximum specified distance but no further. This is done to keep down the amount of multipath signals that otherwise would flood the receiver. [2]

### 5.2.1 Sonar Equations to calculate transmission losses

To calculate the strength of the acoustic signal that is to be sent we can make use of the so called sonar equations. These equations try to quantify all the effects that apply when sending acoustic signals through an underwater transmission channel [33]. There are two types of sonar equations, passive and active [34]. The active sonar equations apply when the acoustic signals are produced by our system, propagated through the underwater channel, and then reflected by the target [34]. This system utilizes the same principles as land based RADAR systems [34]. However, in the case of our intended system, we are only interested in sending the information one-way, so that the receiver can detect and react to the information that has been sent. For this, we can make use of the passive sonar equations in order to calculate the needed source level (SL) in decibel (dB), [2]. However, we have to modify the equations so that they better fit our short range, omnidirectional transducer pattern. Starting with a standard sonar equation 5-1, this equation can be reduced to equation 5-2 where RL represents the receiver sound level in dB and TL the transmission losses in dB due to the underwater channel [2] [33].

$$SL = NL + TL - (DI_R + DI_T) + DT$$

**5-1**

$$SL = RL + TL$$

**5-2**

These reductions are made possible due to that we are using omnidirectional transducers and because of the short distances between the transmitter and receiver [33] [2]. The directivity index of the omnidirectional transmitter ($DI_T$) and receiver ($DI_R$) has a power ratio of one, which corresponds to 0dB and is therefore neglected in the reduced equation [33]. NL represents the total effect of background- and self-noise that is affecting the system [33]. However, the short distance makes it possible to neglect this parameter [2]. The detection threshold (DT) is a user specific parameter, and can therefore be neglected [33].

The reduced equation shows that the source level is equal to the receiver level plus the transmission losses of the transmission channel. The transmission losses increase with distance due to the geometrical spreading, and it is also affected by frequency dependent absorption [2], according to equation 5-3 where r is the range and α is the absorption coefficient.

$$TL = 20 \log r + \alpha r$$

**5-3**

The first part of equation 5-3 refers to the geometrical spreading, which produces the most of the transmission losses [2]. At a range of 20 meters, the losses related to geometrical spreading would be about 26dB. The impact of the waters ability to absorb sound waves can be seen in Figure 18. The data is calculated according to the Francois and Garrison algorithm from 1982 [16].

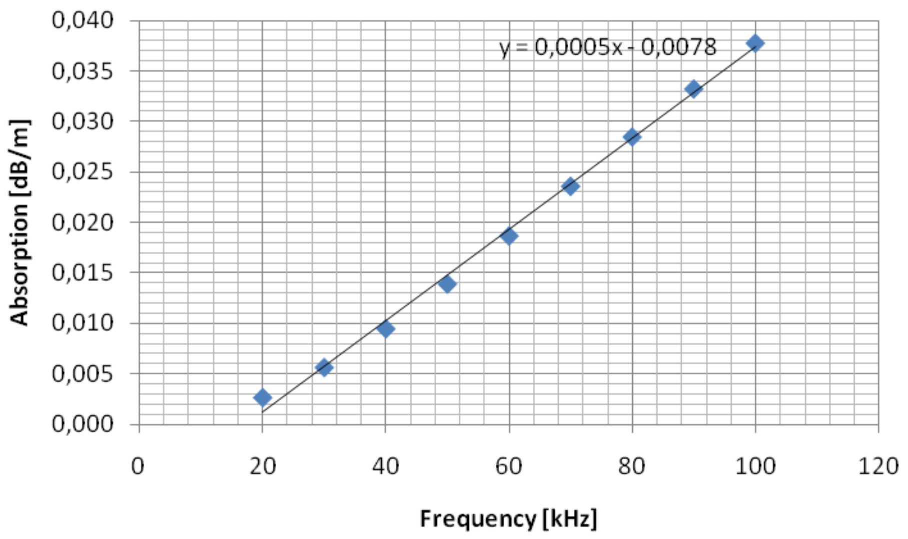

**Figure 18: Sound absorption vs. frequency 20-100 kHz.** Data is calculated under the following conditions: Temperature= 17°C, Depth 18m, Salinity 35ppt, Acidity 8pH.

Here we can see that the absorption varies between 0,003-0,038 dB/m for frequencies of 20-100 kHz [16]. Of course, the absorption coefficient varies based on other parameters such as temperature and salinity. However, the variations are slight and do not change the fact that the geometrical spreading plays the main part of the transmission losses [16]. So for two systems separated by 20 meters and working at 40 kHz the transmission losses would be:

$$TL = 20 \log 20 + 0{,}01 * 20 = 26{,}2dB$$

**5-4**

The source level of the ultrasonic transducer is often specified in the datasheets, for the specific product. The sound pressure level (SPL) in the data sheets often refer to the pressure of 20µPa as the zero decibel level [35]. This is the reference level in air. In water the reference level of 1µPa is often used, and therefore a recalculation of the dB-levels is often necessary [9]. The specified SPL should also be measured at the standard distance of 1m at the center of the acoustic source [2]. However many manufacturers specify the distance of 30cm. If we are to apply this data to our sonar equations we have to calculate the SPL at 1m, and this is done according to [35]:

$$L_2 = L_1 - 20 * log\frac{r_2}{r_1}$$

**5-5**

$L_1$ represents the sound pressure level at the specified distance $r_1$. In order to calculate the SPL at distance $r_2$ we solve $L_2$ for the equation above. For instance if the SPL for a given transducer is 100dB at 30cm, with a reference level of 1µPa, then the equation above would give us a SPL of 89,5dB at the distance of 1m from the source. Once again if we use the reduced sonar equation in combination with the given examples for transmission losses and source levels we would get a receiver level of approximately 63,3dB at a distance of 20m.

### 5.2.2 Hardware implementation

The sonar equations are important tools when it comes to designing the transmitter unit. As mentioned earlier there are many different ways of implementing the hardware. We have looked at methods used in earlier work, for instance the use of transformers to boost the ultrasonic waves, or switched amplifiers [10] [14]. But first we have the form and type of the signal when it leaves the microcontroller. It is quite easy to generate square waves with a micro controller. Normally we would have to convert them into sine waves in order to send them without undesired overtones [9]. However, if a transducer with a small bandwidth is chosen, then it will acts as a band-stop filter (see section 4.3.2), and therefore only transmit the first harmonic (fundamental tone) of the received square wave, given the first harmonic coincides with the transducer frequency [9]. It is a bit confusing to say that the transducer acts as a band-stop filter at its resonance frequency. But if we think about it, we realize that if the transducer is to transmit a signal, we have to apply an alternating voltage with a high amplitude over the transducer to make it oscillate, as the alternating current it needs would otherwise be too low. This is because a band-stop filter has its highest impedance in the stop band, and when applying ohms law, we see that to get a high alternating current, the alternating voltage amplitude needs to be high as well [9].

Another method to construct a sine wave is to utilize the pulse width modulation (PWM) technique that is implemented in many microcontrollers [36]. This in combination with a low pass filter will generate a sine wave [37]. The drawback with this technique is that it requires the PWM-frequency to be quite high. For instance if we want to generate a 40kHz sine wave with 5-bit resolution (32 steps) the PWM-frequency is then given by the following formula [37]:

$$PWM_{freq} = (sine_{freq}) * (nr.of\ steps) * 10 \tag{5-6}$$

This gives us a PWM-frequency of 12.8MHz. As you can see this method requires a high frequency in order to generate a moderate frequency of 40kHz. Of course, the required frequency can be reduced by accepting a lower resolution by changing the number of steps [37]. An advantage with this type of signal generation is that the center frequency easily can be changed by editing the software of the microcontroller. This is a benefit given the variation of center frequency in the transducers.

As mentioned earlier in this section the transducer drive can be built around a transformer [14]. The schematic in Figure 19 shows the transformer and how the secondary winding is connected to the piezoelectric transducer that has been modeled as a capacitor. Due to that the transducer can be modeled as a capacitor, the right hand side of the circuit will then act as a parallel LC-oscillator, producing a band-stop filer [9] [10].

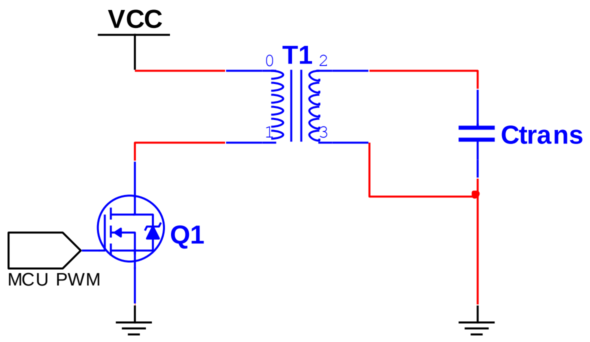

**Figure 19: Schematic for a transducer drive.**

The primary winding of the transformer is connected from positive supply voltage to the source pin of a power MOSFET, which is controlled by a PWM signal, provided by a microcontroller (MCU) [10]. The LC-circuit will be oscillating at its resonance frequency when energy is applied. In addition, by timing the MOSFET switches to the upswing of the LC-oscillator resonance frequency, we will only impart that much energy to the LC-oscillator, so that it replaces the energy losses to the water. Thereby the actual signal generation is left to the LC-oscillator [10]. However, there are both pros and cons when choosing this alternative. The obvious benefit is that the supply voltage of the transducer circuit can be held low. Another is the signal generation speed of the circuit. It can reach full amplitude within a single period of operation [10]. A drawback with this kind of solution is that the inductor and capacitor of the parallel LC-oscillator set the resonance frequency. In order to get the desired frequency we have to add capacitors in parallel with the transducer [10]. This is especially tricky when dealing with narrow bandwidth transducers. Another drawback is that transformers are relatively expensive when comparing to other components.

A more straightforward method, presented by [12] and [14] is to utilize a switching amplifier to boost the transducer voltage. View the block configuration in Figure 20.

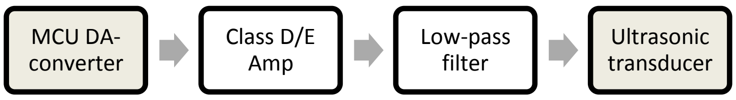

**Figure 20: Transmitter set up for the "Microcontroller-Based Underwater Acoustic ECG Telemetry System"**

To minimize power dissipation we should discard the linear amplifiers and stick to the Class D and Class E switched amplifiers [12] [9]. The drawback with this method when compared with the method of using transformers as the boost circuit is that the switched amplifiers require a supply voltage that is a bit higher than the desired transducer voltage [9]. To solve this we either have to use batteries with higher voltage output or some kind of step-up converter so generate sufficient supply voltage to the switched amplifiers [9]. The low-pass filter or reconstruction filter in Figure 20 is there to eliminate high-frequency switching components from the PWM signal [9] [14].

### 5.2.3 Choice of transmitter unit

The transmitter unit that we have decided to implement is the one utilizing a switched amplifier. The motivation for this is that an implementation of the transformer boosted transmitter circuit would be tricky, when considering the narrow bandwidth transducers used in our system. The chosen transducers would also make the calculations for the resonance frequency of the LC-oscillator both critical and hard. Furthermore, it would be more costly. Thereby we aim for the solution of using switched amplifiers to amplify the square-wave bursts, produced by the MCU. The amplified signal will then be fed directly to the ultrasonic transducer. This is possible due to the band-stop characteristics of the narrow bandwidth transducers. If the bandwidth of the transducer is too wide, we could make use of filters in-between the amplifier and the transducer [37] [9]. The block set for the opted solution is shown in Figure 21.

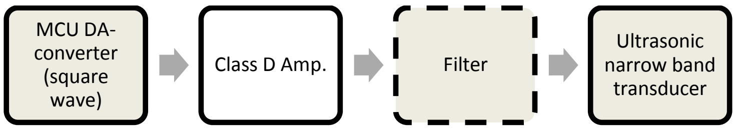
MCU DA-converter (square wave)
Class D Amp.
Filter
Ultrasonic narrow band transducer

**Figure 21: Chosen transmitter block set.**

## 5.3 Receiver

The general configuration of the receiver block set is seen in Figure 22 [14] [10] [12]. We need some kind of pre-amplification of the signal. The problem here is the great variations in signal level caused by varying distances between the two Buddy Tracker units. The preamplifier will be followed by some kind of signal conditioning unit. The extent of this block set highly depends on whether or not the MCU will be conducting the signal processing, or just sampling the already conditioned signal. More about this later on in this section.

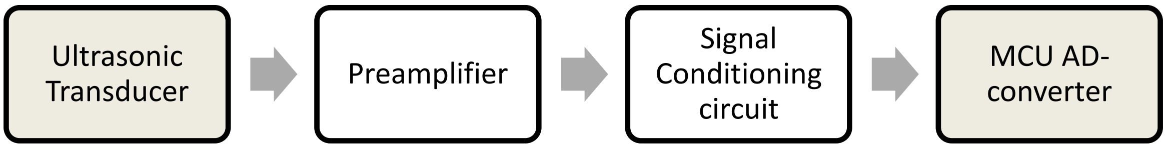

**Figure 22: General receiver block set.**

### 5.3.1 Evaluation of receiver types

In section 3.2.1 they used a software controlled amplifier to boost the signal from the ceramic transducer. This signal was then sent to a tone decoder [10]. The block set of this configuration is seen in Figure 23.

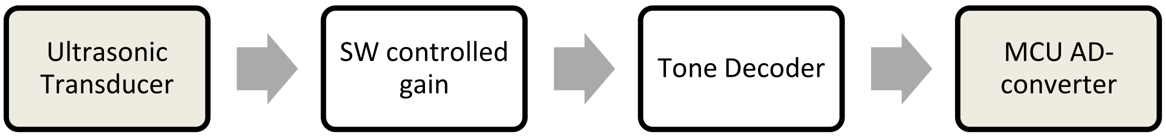

**Figure 23: Receiver set up, "Ultrasonic communication system for biotelemetry in extremely shallow waters".**

The software-controlled gain is a way of assuring that the gain output level of the amplifier is held within the operation level of the tone decoder, which is of type LM567 and accepts voltage levels up to 9,5V [9] [38]. The tone decoder is designed to provide a saturated transistor switch to ground when an input signal is present within the preconfigured pass band [38]. In other words, if a transmission burst is detected within the pass band, say 40kHz, the tone decoder outputs a high output determined by the supply voltage and regardless of the strength of the incoming signal [38]. This signal is then AD-converted by the MCU [10]. This method is easy to implement except for some programming of the software controlled gain. However, when it comes to power consumption, this is not the best choice in hand. The LM567 Tone Decoder has a specified power dissipation of 1100mW according to the data sheet [38]. This can be avoided by passing some of the signal processing onto the MCU and thereby replacing the tone decoder with less power consuming electronics.

Yet another receiver configuration is the one used by the unit in section 3.2.2. The block configuration for this receiver set-up is seen in Figure 24.

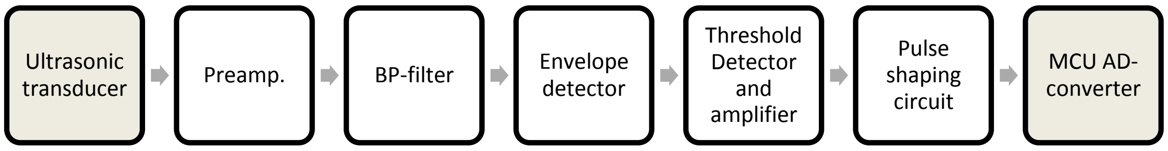

**Figure 24: Receiver set up for the ECG Telemetry system [12].**

First they have a preamplifier that feeds the received signal to a band-pass filter. The need of the band pass filter is determined by the ultrasonic transducer bandwidth (see section 4.3.2). If a narrow bandwidth transducer is used, then the transducer itself will act as a band-pass filter for the received signal. Of course the purpose of the filter could be to eliminate noise that has been amplified by the preamplifier.

The envelope detector is an electrical circuit that converts a high frequency input signal, and outputs the "Envelope" of the input, as can be seen in Figure 25, where the red line on top indicates the envelope [9].

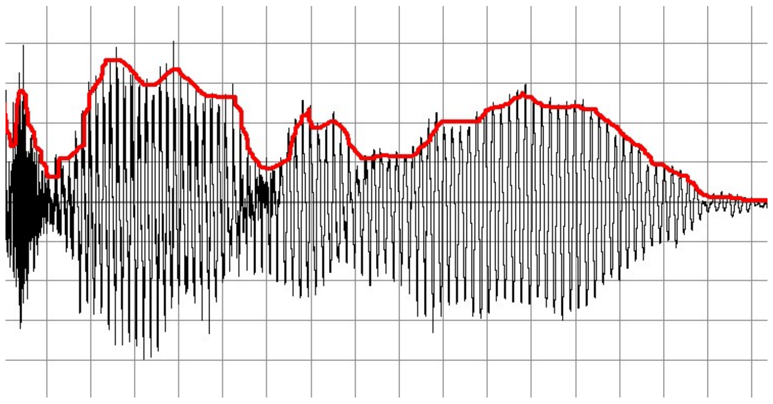

**Figure 25: A signal and its envelope marked with red. [9]**

The envelope detector is often made up by two main parts. First of all the signal is rectified with either a half or full wave rectifier, then the pulsed DC signal is filtered with a low-pass filter to smoothen the pulses [9]. The output pulse will become a bit wider due to the smoothening operation, that will leave a tail after that the last signal pulse has disappeared [9]. To get rid of this, the pulse will later on pass through a pulse shaping circuit, but before that, it will be fed through a Threshold detector [2]. The threshold detector is of hysteresis type, and it will give a saturated output when the input signal passes a hardware configurable threshold [9]. The purpose of this block is to provide a constant output voltage for the subsequent blocks, and possibly, it could be used to wake up a microcontroller from hibernation.

Pulse shaping circuits are often implemented to get rid of inter symbol interference (ISI) in a transmission [9]. In our case the envelope detector generated a "tail" to the processed pulse due to the smoothening process of the low-pass filter. This tail was then sent through the threshold detector that outputted a square pulse, which was a bit wider than the original pulse because of the "tail" mentioned earlier. To reconstruct the original pulse width they made use of a pulse shaper to "cut" away the added pulse width. A commonly used pulse shaper is the raised-cosine filter [9].

After the pulse-shaping procedure, the signal will be ready for the microcontroller.

### 5.3.2 Choice of receiver unit

The receiver type, which we have opted for is configured so that the microprocessor will perform most of the signal processing work. This to minimize power consumption by minimizing the number of external units such as Tone Decoders and threshold detectors that, if implemented would draw considerably more power [38] [9]. The hardware block sets that we still need to implement in our receiver board is seen in Figure 26.

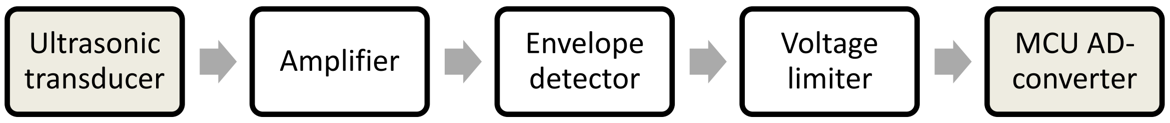

**Figure 26: Chosen receiver block sets.**

First, we have the ultrasonic transducer that should be a narrow band piezoelectric transducer. The acoustic signal, sent from the other unit will have low amplitude and it will also be mixed with noise from the surrounding environment. However, by choosing a narrow band transducer we will get a band pass characteristic on the received signal (see section 4.3.2). This filtering will probably be enough in order to omit the earlier mentioned band-pass filter. The amplifier has a fixed amplification to reduce implementation complexity [14]. The choice of correct gain level will be vital, so that the signal can be amplified enough for the subsequent blocks, but not too strongly. In order to protect the MCU from over voltage we placed a voltage limiter just before the MCU AD-converter [9]. This voltage limiter is made up by a Zener diode to keep the voltage from exceeding MCU maximum ratings. A resistor will be regulating the current that passes through the Zener diode [9].

In-between the amplifier and the voltage limiter we have the Envelope detector which we have chosen to be a full-wave rectifier followed by a low-pass filter [9] [12]. The issue about the envelope detector, and its ability to introduce ISI with the "tail" is not a problem in our case since we will be using pulse code modulation in combination with on-off keying (see section 3.7.2 and 3.7.1). This means that after every successive pulse, there will be a decay time for the multipath, and therefore the introduced ISI will not be distorting any subsequent pulses (see section 3.7.3). The signal that will be feed to the AD-converter of the MCU will be a square-like pulse.

## 5.4 Ultrasonic transducer unit

The ultrasonic transducer unit consists of both the transmitter- and receiver unit. This is made possible with the use of a transmitter/receiver switching network as in the WHOI Micro-modem, mentioned earlier in this report. In practice, we will implement MCU controlled transistor switches to handle the switching between the receiver and transmitter circuitry [9]. By doing this we will be able to use the same ultrasonic transducer for both sending and receiving information. We could also reduce the number of used MCU output/input pins if the chosen MCU supports pin I/O changes during real-time operation [36].

## 5.5 Alarm unit

The alarm unit is activated by the MCU when the SCUBA divers are separated, and the set alarm distance is exceeded, or when no communication signal is received under a predetermined time duration. The unit consists of a buzzer unit and a circuit that flashes a set of high brightness LEDs.

The idea is that the MCU only will need to set or reset an output to the alarm unit. This way the signal generation is left to the alarm unit, and the MCU can continue with the other tasks.

## 5.6 Microcontroller

### 5.6.1 Evaluation of microcontrollers

The two microcontrollers used in 3.2.1 were both from Atmel and belong to their 8-bit series [11]. The microcontroller situated on the receiver side worked at 20MHz compared to the transmitter side, working at 8MHz [10]. This is due to the higher demands on signal processing for the receiver unit, when working with FSK and error correction coding [10]. Otherwise, the two processors were quite similar in their key features as seen in Table 4.

**Table 4: Comparison of microcontroller/DSP features used in related work. [11] [13] [39] [36]**

| **Name** | **Archi-tecture** | **Speed** | **Program memory** | **RAM** | **PWM** | **ADC** | **Wake-up** | **Operation voltage** |
|---|---|---|---|---|---|---|---|---|
| Atmel Atmega168 | 8-bit | Up to 20MHz | 16k | 1k SRAM | 6ch | 8/6ch | Yes | 2.7-5.5V, 20MHz @ 4.5-5.5V |
| Atmel Atmega324 | 8-bit | Up to 20MHz | 32kb | 2k SRAM | 6ch | 8ch | Yes | 1.8-5.5V, 20MHz @ 4.5-5.5V |
| Intel 87C51 | 8-bit | 12/16MHz | 4k | 128b | No | No | ? | Approx. 2.2-5.5V @ 12MHz |
| Texas Instruments TMS320VC5416 (DSP) | 16-bit | 160MHz | 32k | 256k | No | No | Yes | 2.7-3.6V |
| *Microchip dsPIC33FJ128GP802* | *16-bit* | *40MHz* | *128k* | *16k* | *16-bit PWM* | *1-A/D 10x12-bit* | *Yes* | *3.0-3.6V @ 40MIPS* |

The Intel 87C51 microcontroller that was used in 3.2.2 for transmitting and receiving PCM or PPM signals in combination with OOK is running at either 12 or 16MHz, depending on the used microcontroller [13]. However, this indicates that the computational complexity of sending signals in either PCM or PPM with OOK is rather low [12]. This makes the choice of microcontroller less dependent on the processing speed and we can concentrate more on other parameters such as the implementation of AD/DA-converters, PWM-units and power consumption.

Furthermore, the DSP used in the WHOI micro-modem is by far the most competent signal processor in the crowd [39]. As seen in Table 4, it is running at 160MHz with a 16-bit Harvard architecture. The DSP has a high level of parallelism and can perform two read operations and one write operation within a single cycle [39]. This excess of processing power is however overshadowed by the lack of peripheral units such as the above mentioned AD/DA-converter and PWM-units.

### 5.6.2 Chosen microcontroller

The microcontroller we would like to use in our system should have the peripheral units that we need, in order to minimize the external part count and there by minimize power consumption. The above mentioned DSP has great computational power but lacks the peripheral units while the Atmega family has more peripheral units, but not that much overhead for further development [39] [11]. Thereby we sense that we have to find something in between, and with some more peripheral units such as DA-converters.

The Microchip 16-bit dsPIC33 family presents a quite competent microcontroller with DSP functionalities for more complex implementations. The dsPICFJ128GP802 with its speed of 40 MIPS (Million instructions per second) should provide sufficient overhead. It is also fitted with both an AD- and DA-converter for easy input and output configurations. The PWM-unit also provides further development opportunities. For instance the implementation of a PWM driven transmitter unit. [36]

When choosing a microcontroller it is beneficial if it can maintain performance over a wide range of supply voltage variations [10]. This in order to drive the MCU directly from the battery and by that reduce the number of external units, but also lower power consumption. However the chosen dsPIC33 only accepts supply voltage variations within 3,0-3,6V in order to operate at 40MIPS, as seen in Table 4. This is a sacrifice we are willing to take due to the units many peripherals, that will reduce the external part count. For this reason we have to implement a voltage regulator on the power supply unit [9].

## 5.7 User interface

The user interface consists of an on/off switch, a channel selector, an alarm distance selector and a master/slave switch. The set information will be collected by the MCU at system power up in order to determine if the unit will be working as a master or slave in the buddy tracker pair (see section 6 for further information). In addition, the set alarm distance and channel information is collected.

### 5.7.1 Channel selector

The channel selector is to be able to set 16 channels. This is due to that we are using 4-bit identification of the sent packages. A pair of buddy Tracker units will have the same channel set. The easiest way of implementing this would be by assigning a dipswitch with four switches that represents the channels in binary code, which then would occupy four inputs on the MCU. The drawback of this method is that it is not that user friendly. We could apply 16 dipswitches but then we would end up occupying 16 inputs of the MCU. This is not necessary a bad idea because microcontrollers often come with great number of assignable inputs and outputs [36]. However, the problem would be to fit the dipswitches into the casing [29]. A more expensive but yet user friendly approach would be to replace the binary dipswitch with a rotational code switch as that seen in Figure 27.

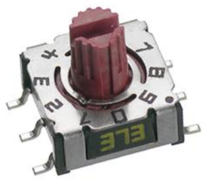

**Figure 27: Coded switch for circuit boards by Hartmann. [29]**

A more elegant but yet more power consuming, and more expensive alternative would be to implement multiplexing of the channels [9]. This approach would only be justified for systems with lot more channels than our 16. A less orthodox way of solving the problem would be to install a potentiometer with fixed steps and connect it to a single AD-converter input of the MCU [29] [36]. The variable voltage level produced by the potentiometer would then be mapped to predetermined channel gaps and then assigned a channel number. However, potentiometers of this type are presumed to be expensive, and therefore the binary approach with the dipswitches seems to be the most cost efficient way of solving the problem and it occupies only four inputs to the MCU.

### 5.7.2 Alarm distance

In section 4.2, "Distance Evaluation" the alarm distances was mapped to ten time intervals that accounted for changing surrounding properties. To represent these distances binary, we will need four bits. Therefore, the same methods as for the channel selector apply. This leads to the implementation of four additional dipswitches.

## 5.8 Power supply

It would be ideal if we could drive the Buddy Tracker unit directly from a battery. At least when considering power consumption. However, many of the MCUs on the market have quite stringent supply voltage requirements. For instance, the dspic33fj128gp802 MCU from Microchip requires 3.0-3.6V to function [36]. For this, we would need some kind of voltage regulator to stabilize the supply voltage [9].

The choice of supply voltage also depends on the method used to boost the voltage to the ultrasonic transducer. If we use the method that makes use of s transformer, then it might be ok to use a low battery voltage, providing that the MCU can operate at the low voltage. However if we go for the alternative of boosting the transducer voltage with a switched amplifier then we have to either provide the system with a higher battery voltage or some kind of "step-up converter"[9]. This is to ensure that the switched amplifier will have enough "swing" to be able to boost the transducer voltage.

## 5.9 Encapsulation

The Buddy Tracker system will be encapsulated in a waterproof aluminum casing. A cylindrical shape with the transducer located in one end, and the user interfaces located in the other end. The user interfaces will be protected inside an o-ring sealed waterproof cap [6]. This will restrict the user from changing settings during the dive. However, by eliminating the chances of accidental system changes we come yet one-step closer to our aim in constructing a reliable early warning system. In addition we might be forced to use an external transducer unit to ensure that there will be no coverage holes in the transmission and so that the "Line of sight" transmissions can be maintained.

# 6 SOFTWARE DESIGN

In the previous chapter, we introduced the hardware solution for the system, and now we will present how we intend the system to operate.

## 6.1 Intended system operation

Before turning the unit on, we will have to set the master/slave switch, the channel and the distance limit. When we then turn on the system, it will save the settings and ignore any changes to them through the user interface. At the same time, the alarm will start sounding due to it always being active at start. By being active at start, we get a confirmation of that the alarm is working and that the battery is not empty. Immediately after that the unit has initialized and the alarm has started, the unit, if set to master, will start the process of measuring the distance to the other unit. The unit that is set to slave will instead start the process of listening after a signal from the other unit, and if successful, respond according to the information received. If everything works and the distance measuring is successful, measuring a distance between the units under the limit set, the units will both deactivate their alarms. If one or both of the units do not deactivate the alarm, this indicates that something is wrong and the system should not be used any more without inspection and testing. If both units work as they should we can now begin our dive while the system continues to monitor the distance between us. As soon as the measure results in a distance that exceeds the limit, the master unit activates the alarm and continues to measure the distance as before, while it at the same time tells the slave unit to activate its alarm. If any of the units does not receive a signal from the other unit in a certain time, they are set to activate the alarm, alerting the divers that something is wrong.

## 6.2 System flow charts

In the last part, we described how the system operates from the user's point of view. In this part, we will do an ingoing presentation of the systems operation by using flow charts that we have made in four levels (shown above the charts), each more specific to a particular task. These levels will consist of several sub processes that we will present in order of appearance.

### 6.2.1 System startup and initialization

We start with the startup process of the system called START, shown in Figure 28. When starting up, the process will call a sub process called INIT, shown in Figure 29, to initialize the system. It will set the system channel, if the unit will act as a master or a slave, and the distance limit, which each are settings that the user made through the user interface before he/she powered on the system.

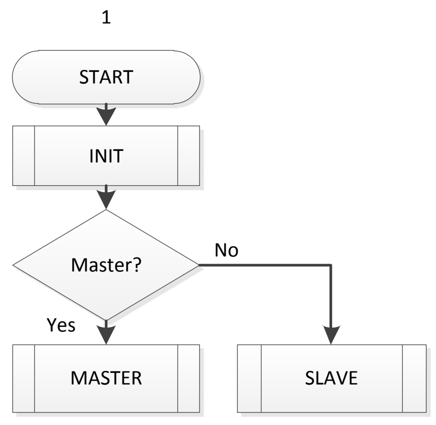

**Figure 28:** Top-level flow chart.

When the initialization is complete, the control returns to the top process, which then checks if the system has been set to operate in master or slave mode and then jumps to one of two sub processes accordingly.

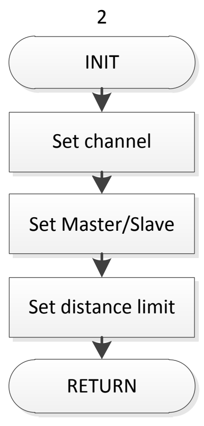

**Figure 29:** Flowchart of the second level process INIT.

### 6.2.2 Master and slave operation processes

Depending on which mode the system is set to operate in, the control will jump to one of the second level sub processes shown in Figure 30 and Figure 31. These processes start by making eventual necessary initializations specific to the mode running. After the extra initialization, there are differences in the operation. The process MASTER will give the control to third level sub process SEND SIGNAL that will start the distance measurement by sending a request to the slave unit.

When the process has sent the signal, it gives the control back to the MASTER process that then continues by running the CHECK FOR SIGNAL sub process, Figure 33. This process checks for a response from the slave unit. Regardless of if the process acquires a response from the slave unit or not, it eventually returns the control to the MASTER process. This process then runs the sub process CHECK DISTANCE, Figure 37, which determines the distance between the units. If the distance exceeds the limit or if the process does not acquire a response in time, it sets the alarm. However, if the process acquires a response in time and determines a distance within the limited range, it instead resets the alarm.

After the sub process CHECK DISTANCE has completed, the MASTER process calls its final sub process ALARM, Figure 34, which activates or deactivates the alarm depending on if the alarm bit is set. When the MASTER process have run through all sub processes that it is supposed to run, it puts the unit to sleep an amount of seconds before the process runs once again, doing a new measurement.

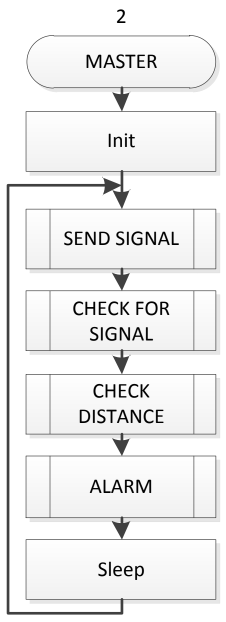

**Figure 30:** Flowchart of the second level process MASTER.

If the unit instead is set to act as a slave, the SLAVE process will continue after the initialization by first doing initializations specific to the slave. The process will then hand over the control to the CHECK FOR SIGNAL sub process, which will check for a request from the master unit. Once done, the SLAVE process then runs the sub process ALARM that checks if the previous process did set the alarm, either by receiving a high alarm bit in a request or because of it not receiving a request in time. The ALARM process then activates or deactivates the alarm according to the status.

When the sub process ALARM has finished and returned the control to the SLAVE process, the SLAVE process checks if it should run its final second level sub process to send a response to the master unit, which it only is supposed to do if it received a request. If it does send a response, it assumes that the master unit receives the response without any problem, as there was not any when the master did send the request. Because of this assumption, there is no need to check for a new request from the master unit for some time and the SLAVE process therefore puts the unit to sleep for a specific amount of time, equal to the time that the master unit sleeps after a successful measurement. If the unit however did not receive a request, the SLAVE process immediately jumps back to its beginning and starts over with the process of checking for a signal.

The advantage of putting the unit to sleep after it has sent a response to the master unit is that it will consume less power, while a continues loop of the program without letting it sleep would drain more power from the power source, and result in a shorter operation time which would inflict on the security. The fact that the slave unit will sleep without knowing if the master unit has received the response also inflicts on the security of the system, but only for a short time, the sleep time. After the slave unit has slept, it will not sleep again until it correctly receives a new request from the master. If it then receives a signal from the master with the alarm status set due to that the master did not receive the response from the slave, the slave will activate its alarm. This leads to a slight delay before the slave activates the alarm, but the slave still activates it and we think that it is much better than that the slave unit drains all of its power, which could lead to the alarm not being able to sound at all.

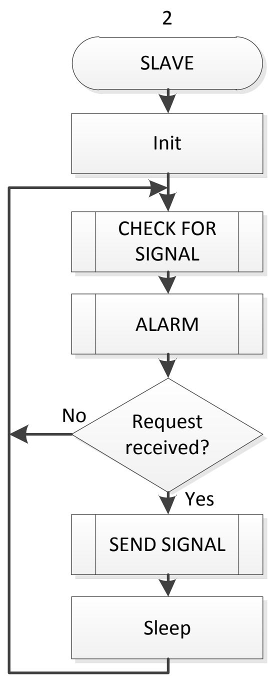

**Figure 31:** Flowchart of the second level process SLAVE.

### 6.2.3 Level three processes

The third level of processes consists of all sub processes called by the processes in level two. These third level processes are SEND SIGNAL, CHECK FOR SIGNAL and ALARM.

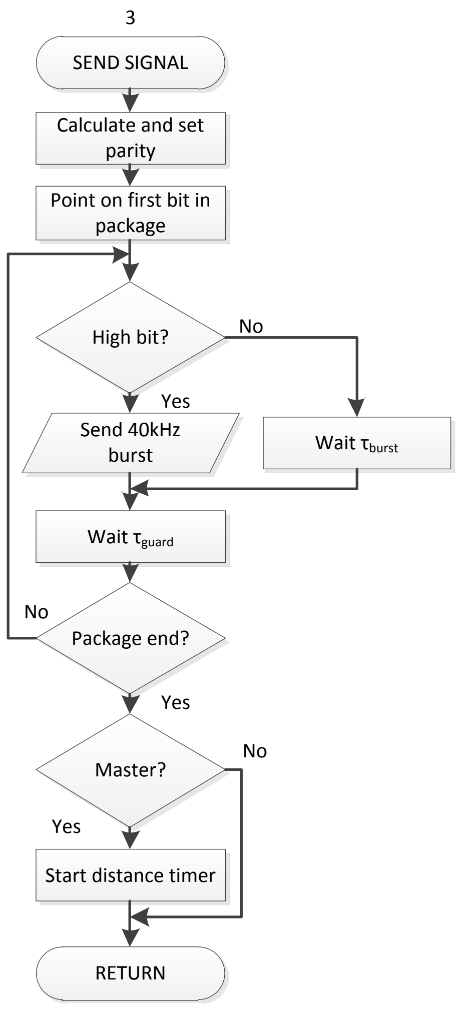

**Figure 32:** Flowchart of the third level process SEND SIGNAL.

The process SEND SIGNAL, shown in Figure 32, is responsible for sending the ultrasonic signal consisting of the four synchronization bits, the channel code, the alarm status and the parity bit. When this process runs, it first starts with calculating the parity and sets that bit in the signal. It then continues by sending the signal, one bit at a time followed by a guard slot, until it has sent the whole signal. After the process has sent the signal, it starts the timer if the unit runs in master mode. It then jumps back to the previous process.

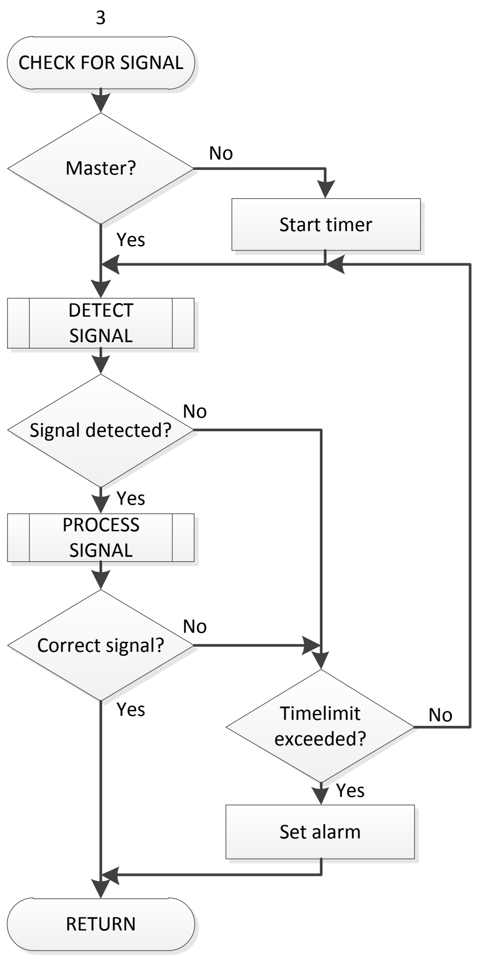

**Figure 33:** Flowchart of the third level process CHECK FOR SIGNAL.

In the process CHECK FOR SIGNAL, Figure 33, the unit, as the name says, will check for a signal. The process begins by starting the timer if the unit runs in slave mode. If run in master mode, the timer should already be running. The process then continues by running one of its own sub processes, DETECT SIGNAL, Figure 35, which will try to detect an incoming signal. If detected, the CHECK FOR SIGNAL process will then start processing the signal by handing over the control to its other sub process, PROCESS SIGNAL, Figure 36, which determines if the unit receives the signal correctly and if the signal contains the right channel code. If the PROCESS SIGNAL process determines that it did not receive the signal correctly, or if the DETECT SIGNAL process did not detect a signal at all, the CHECK FOR SIGNAL process will start over with trying to detect an incoming signal again, as long as it has not taken too much time. If too much time has passed while trying to receive a signal, the process will set the alarm and jump back to the previous process.

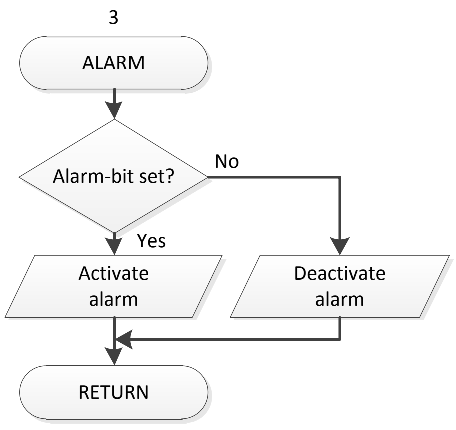

**Figure 34:** Flowchart of the third level process ALARM.

The third and last process of the third level is the ALARM process, Figure 34. This process is very simple. When running, this process activates or deactivates the alarm depending on the alarms status, after which it jumps back to the second level process.

### 6.2.4 Fourth and fifth level processes

In the fourth process level, we have three very important processes that handle the signal reception in the units and provide the system with the basic functionality. These processes are DETECT SIGNAL, PROCESS SIGNAL and CHECK DISTANCE.

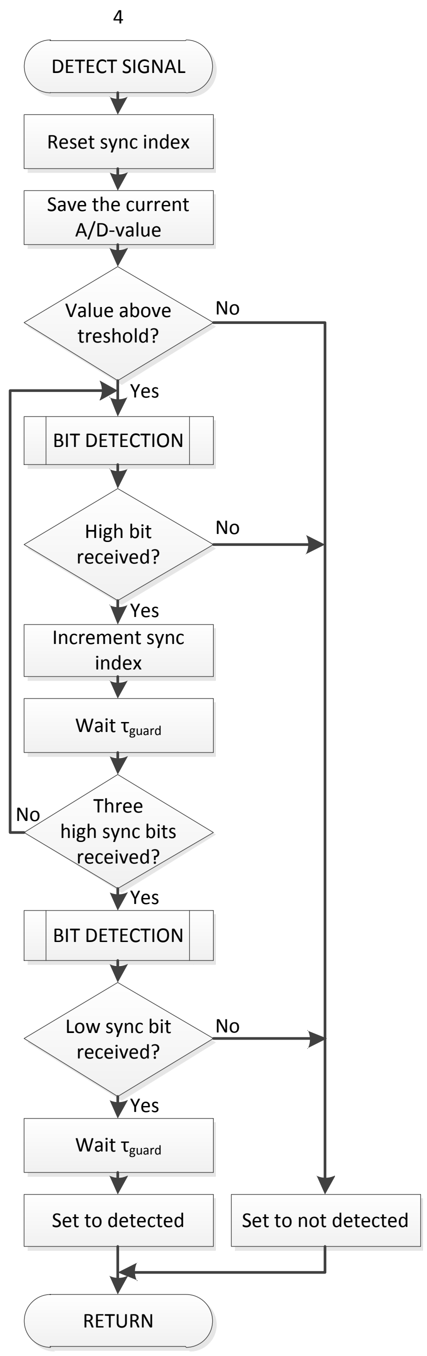

**Figure 35:** Flowchart of the fourth level process DETECT SIGNAL.

DETECT SIGNAL’s, Figure 35, function is to try to identify an incoming signal. When the unit runs this process, the process first tries to detect a potential incoming signal. If detected, the process then continues with trying to synchronize with the signal. If it succeeds, it will set the result to detected, while a failure in any of the steps of the process results in a result set to not detected. Regardless of the outcome, the process then returns control to the previous process.

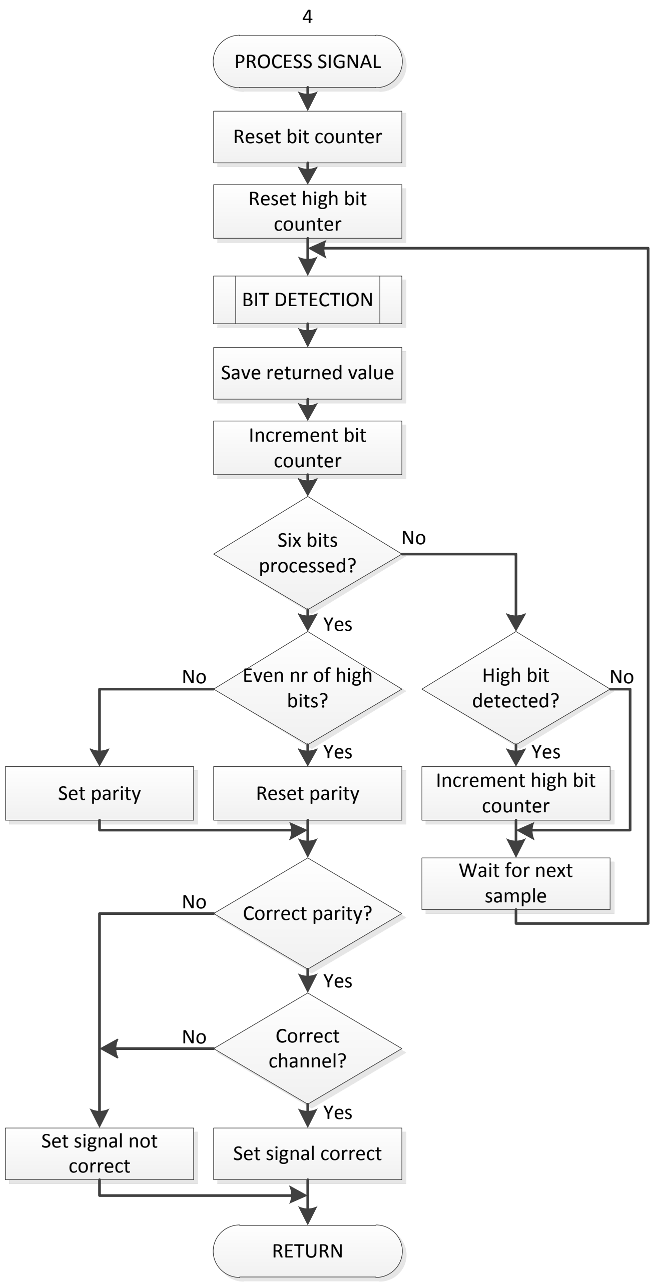

**Figure 36:** Flowchart of the fourth level process PROCESS SIGNAL.

After the DETECT SIGNAL process has synchronized with the signal, the unit needs to receive the rest of the signal that contains the channel code, the alarm status and the parity. To do this, the CHECK FOR SIGNAL process runs the fourth level process PROCESS SIGNAL, Figure 36. In this process the rest of the signal after the synchronization bits, the data, is processed. This process does this by checking each bit (except the parity bit) and then calculates a parity bit that it compares with the received parity bit. If they conform, the process then checks the channel code. If correct, it then sets the signal as correct and returns the control to the previous process. If the parity or the channel code does not conform, the signals will be set as incorrect before returning the control.

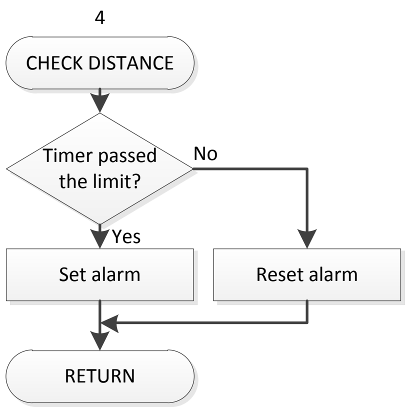

**Figure 37:** Flowchart of the fourth level process CHECK DISTANCE.

The last process of the fourth level is the process that handles the function that the entire system means to accomplish, the distance determination. This process is CHECK DISTANCE, Figure 37, and is very simple. It checks if the timer has passed the limit set by the user or not, and then sets the status of the alarm accordingly.

### 6.2.5 Fifth level processes

This level is the last level and contains only one process, the BIT DETECTION process, Figure 38. The function that this process provides is to be able to detect if an incoming bit is high or low. The process accomplish this by analyzing as many samples as possible during the duration of the bit and counts how many of the samples that is higher than a threshold specified. If the process counts enough samples with a value higher than the threshold, it considers the bit high; else, it considers the bit low. The process then makes the result available for the previous process before returning control to it.

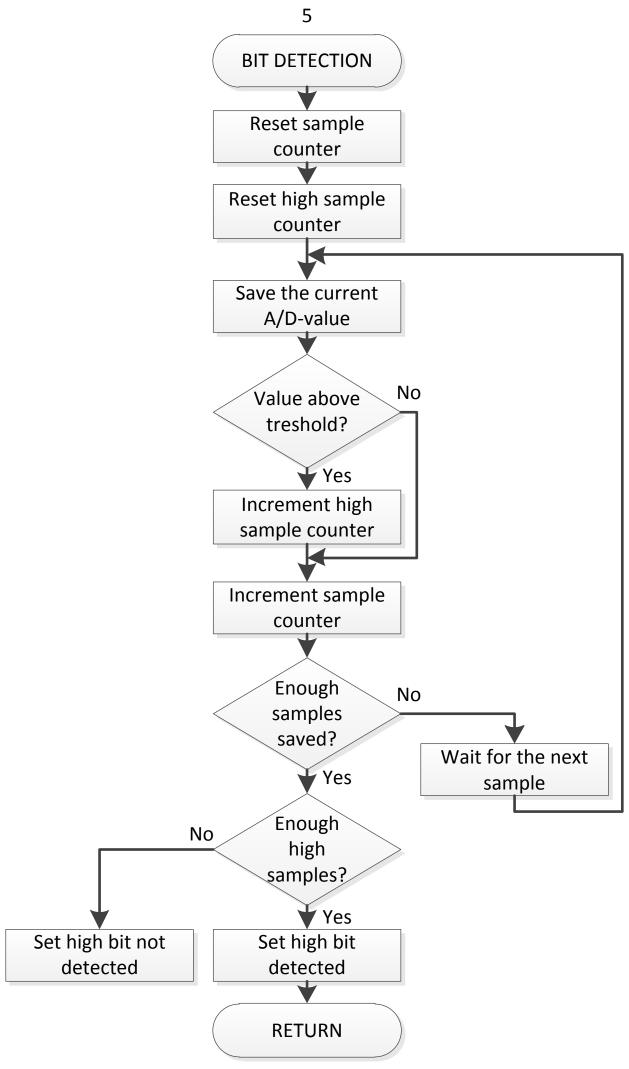

**Figure 38:** Flowchart of the fifth level process BIT DETECTION.

# 7 CONCLUSION

We are confident that the suggested system in this report will provide the means necessary to construct a cheap and simple, but yet robust tracking system to overcome the problem of getting divers separated from each other. The two initial problems of developing the buddy tracker system lay in the method of measuring the distance between the two units, and also how we would overcome the harsh environment of underwater communication.

By utilizing microcontrollers, we gain control over the distance measurement technique. However, in order to tame the harsh environment of underwater acoustics we had to make use of the digital pulse code modulation (PCM) in combination with on-off keying (OOK) to transmit the information in burst mode transmissions. Furthermore, we had to implement guard slot to avoid inter symbol interference (ISI), caused by multipath propagations in extremely shallow waters. This however, severely limits the throughput of the system. Luckily, our demand on throughput is not that high. And a bit rate of approximately 50bit/s, as seen for the PCM system in section 3.2.2, will satisfy our needs.

The aim of making a simple, cost effective, low power system ruled out many possible hardware solutions and left us to choose a microcontroller with many peripheral units in order to minimize the external part count. The most power consuming part of the unit has to be the transmitter side, switched amplifier. When looking at the receiver side, we managed to omit some pulse conditioning block sets by implementing software in order to interpret and decide whether or not the incoming signal was a bit-representation or not.

The flowcharts of this project provide a basis for programming of the buddy tracker system. They are designed to function with multiple units, although problems might occur when the used frequency gets crowded by several buddy tracker units. The solution for this problem is left for further investigation. However, a possible solution would be to introduce a channel-allocation routine that assign units transmission slots based on their current channel settings. It might be possible to tie this function together with the synchronization procedure.

## 7.1 Sugestion for realisation of the buddytracker system

Following is a brief conclusion of our suggested solutions that we have chosen to realize the system with.

Hardware:

- Power – 3,6V Li-ion battery and step-up converter.
- MCU – Microchip 16-bit dsPICFJ128GP802 with A/D & D/A converter. Also possible to configure I/O-pins.
- Transducer – A single ultrasonic narrowband transducer for Tx and Rx.
- Receiver – Fixed gain amplifier, envelope detector, voltage limiter.
- Transmitter – Class-D amplifier and filter.
- Alarm unit – Drive circuit, buzzer and flashing LEDs.
- User interface – Switches for On/Off, channel, distance and master/slave.
- Encapsulation – Waterproof aluminum cylinder.

Other:

- Transmission signal – Pulse code modulated signal with on/off-keying and guard slots.
- Software – Assembly code with total control of the program execution according to the flowcharts in chapter 6.2.
- Package – 10-bits containing synchronization, channel ID, alarm and parity bits (see Figure 12).

## 7.2 Further work

The process of developing the Buddy tracker system is far from over. However, by doing this project we now have a good basis to start from when entering the next phase of the project. This will include system prototyping and testing, so that we can confirm the function of the prompted solutions in this project. The multi channel properties of the buddy tracker unit will need some more attention in order to make system operation more robust. Further development would also aim in minimizing the external part count and thereby transferring even more of the signal processing to the MCU. Implementation of further functionalities such as indicators for low battery, and distance indication is close at hand.

# 8 FINAL WORDS

During our work with this project, we have improved our knowledge in interdisciplinary technologies with the means of connecting them for a common goal, to develop a reliable and cheap system to aid recreational divers in their struggle of not drifting away from each other.

# 9 BIBLIOGRAPHY

## 9.1 Printed

[1] D. Richardson, *" PADI Open Water Diver Manual",* 2006 edition, Books/Cole Publishing Company, 1999, Chap 8,13,14,15,16
[2] R.S.H.Istepanian, M.Stojanovic, *"Underwater Acoustic Digital Signal Processing and Communication Systems"*, Kluwer Academic Publishers, 2002
[3] L.Bengtsson, *"Elektriska mätsystem och mätmetoder"*, 2:3 edition, Studentlitteratur, 2003
[4] P.H.Young, *"Electronic Communication Techniques"*, fifth edition, Pearson Prentice Hall, 2004
[5] J.Geier, *"Wireless System Architecture: How Wireless Works"*, Cisco Press, 2004

## 9.2 Internet

[6] Desert Star Systems, http://www.desertstar.com/, Acc (2010-04-29)
[7] RJE-International, http://www.rjeint.com/, Acc (2010-04-29)
[8] Applied Acoustics, http://www.appliedacoustics.com/, Acc (2010-04-29)
[9] Wikipedia, http://en.wikipedia.org/, Acc (2010-06-04)
[10] An Ultrasonic Communication System for Biotelemetry in Extremely Shallow Waters, http://www.cesr.ncsu.edu/agdean/Papers/wuw1269s-parsons.pdf, Acc (2010-06-04)
[11] Atmel Corporation, http://www.atmel.com/, Acc (2010-06-04)
[12] Microcontroller-Based Underwater Acoustic ECG Telemetry System, http://momed.king.ac.uk/papers/TITBmicro.pdf, Acc (2010-06-04)
[13] Intel 87C51/80C51BH/80C31BH CHMOS SINGLE-CHIP 8-BIT MICROCONTROLLER http://www.datasheetcatalog.org/datasheet/Intel/mXqzvzz.pdf, Acc (2010-06-04)
[14] WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms, http://www.cs.umass.edu/~partan/papers/Micromodem_2005.pdf, Acc (2010-06-04)
[15] Underwater Radio Communication, http://www.qsl.net/vk5br/UwaterComms.htm, Acc (2010-06-04)
[16] Calculation of absorption of sound in seawater, http://resource.npl.co.uk/acoustics/techguides/seaabsorption/, Acc (2010-06-04)
[17] M. Fowler, 2009, *"The Doppler Effect"*, http://galileo.phys.virginia.edu/classes/152.mf1i.spring02/DopplerEffect.htm, Acc (2010-06-04)
[18] A.G.Saremi, *"A Practical Two-Transducer Ultrasonic Flow Meter"*, 2008, http://epubl.ltu.se/1653-0187/2008/102/LTU-PB-EX-08102-SE.pdf, Acc (2010-06-04)
[19] G.McRobbie, P.Marin-Franch, S.Cochrane, *"Beam Characteristics of Ultrasonic Transducers for Underwater Marine Use"*, 2006, http://cds.comsol.com/access/dl/papers/1508/McRobbie.pdf, Acc (2010-05-18)
[20] Encyclopedia Britannica, http://www.britannica.com/, Acc (2010-05-18)

[21] The Mathworks, *"Band-Limited White Noise"*, http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/slref/bandlimitedwhitenoise.html, Acc (2010-06-04)
[22] FSK, http://www.topbits.com/fsk.html, Acc (2010-06-04)
[23] J.Anthes, *"OOK, ASK and FSK Modulation in the Presence of an Interfering signal"*, http://www.rfm.com/corp/appdata/ook.pdf, Acc (2010-04-21)
[24] The speed of sound in sea water, http://www.akin.ru/spravka_eng/s_i_svel_e.htm, Acc (2010-05-03)
[25] Temperature of Ocean Water, http://www.windows2universe.org/earth/Water/temp.html, Acc (2010-05-03)
[26] About the Salton Sea, http://www.saltonsea.ca.gov/thesea.htm, Acc (2010-05-03)
[27] ThinkQuest: Pressure & Buoyancy, http://library.thinkquest.org/28170/32.html, Acc (2010-05-03)
[28] Farnell, http://www.farnell.com/, Acc (2010-06-04)
[29] ELFA, http://www.elfa.se/, Acc (2010-06-04)
[30] H.Matsumoto, S.Nieukirk, M.Fowler, J.Haxel, S.Heimlich, D.K.Mellinger, R.Dziak, C.G.Fox, *"Sound in the Sea: Hands-on Experience with NOAA VENTS Program"*, ftp://ftp.pmel.noaa.gov/newport/mellinger/papers/MatsumotoEtAl03-SoundInTheSea-SatAcademy-Oceans03.pdf, Acc (2010-06-04)
[31] TALtech: Introduction to Serial Communications, http://www.rs232software.com/TALtech_web/resources/intro-sc.html, Acc (2010-06-04)
[32] W.Sandqvist, *"Paritet"*, http://www.ict.kth.se/courses/IL131V/paritet/index.htm, Acc (2010-06-04)
[33] Passive sonar equations, http://www.nadn.navy.mil/Users/physics/ejtuchol/Chapter9.pdf, Acc (2010-06-04)
[34] Active Sonar equations, http://www.nadn.navy.mil/Users/physics/ejtuchol/Chapter16.pdf, Acc (2010-06-04)
[35] Sound pressure level depending on the distance, http://www.sengpielaudio.com/calculator-distance.htm, Acc (2010-06-04)
[36] Microchip Technology Inc., http://www.microchip.com/, Acc (2010-06-04)
[37] Microchip Technology Inc., *"D/A Conversion Using PWM and R-2R Ladders to Generate Sine and DTMF Waveforms"*, http://ww1.microchip.com/downloads/en/AppNotes/00655a.pdf, Acc (2010-06-04)
[38] National Semiconductor Corporation, *"LM567 Tone Decoder"*, http://www.national.com/pf/LM/LM567.html, Acc (2010-06-04)
[39] Texas Instruments, *"TMS320VC5416 Fixed-Point Digital Signal Processor"*, http://focus.ti.com/lit/ds/sprs095p/sprs095p.pdf, Acc (2010-06-04)

## APPENDIX 1: ABSORPTION COEFFICIENTS FOR SOUND WAVES IN WATER

Table A1 shows absorption coefficients for different sound frequencies. The data is calculated according to the Francois and Garrison algorithm from 1982 and with the following conditions applied. Temperature= 17°C, Depth 18m, Salinity 35ppt, Acidity 8pH.

**Table A1: Sound absorption vs. frequency. Data is calculated for Temperature= 17°C, Depth 18m, Salinity 35ppt, Acidity 8pH [http://resource.npl.co.uk/acoustics/techguides/seaabsorption/]**

| Frequency [kHz] | Absorption [dB/m] |
|---:|---:|
| 20 | 0,003 |
| 30 | 0,006 |
| 40 | 0,010 |
| 50 | 0,014 |
| 60 | 0,019 |
| 70 | 0,024 |
| 80 | 0,029 |
| 90 | 0,033 |
| 100 | 0,038 |
| 200 | 0,070 |
| 300 | 0,092 |
| 400 | 0,113 |
| 500 | 0,137 |

In Figure A1 we see how the absorption changes with increasing frequency. The change attenuates with higher frequencies.

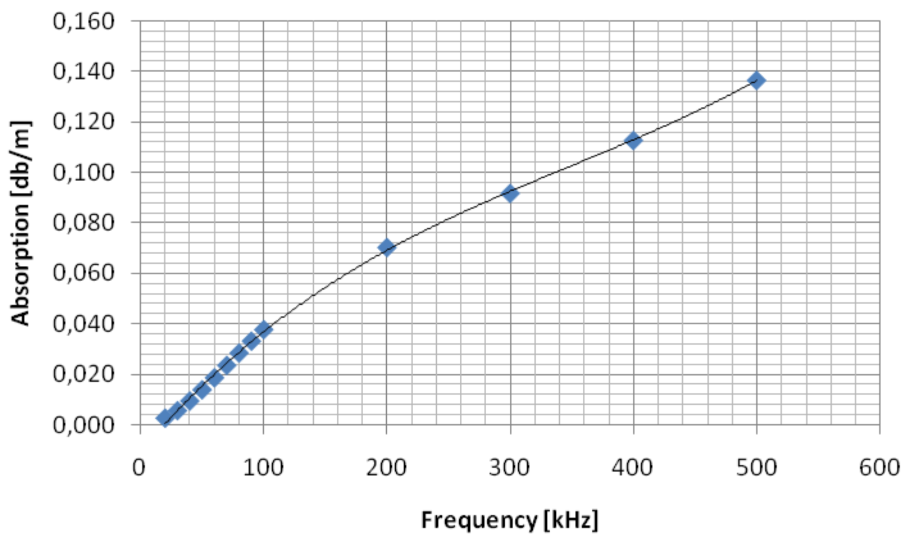

**Figure A1: Sound absorption vs. frequency 20-500 kHz.**