



Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar

TDK DOLGOZAT

Kiterjesztett valóságban megjelenített, ultrahang alapú helymeghatározó rendszer fejlesztése

Benedek András
Info-Bionika Mérnöki MSc

2022

Témavezetők: Dr. Koller Miklós, Dr. Cserey György Gábor

Content

Abstract	3
1. Introduction	4
2. Literature research	5
2.1. Sound Source Localization.....	5
2.2. Sound localization in human	5
2.3. Virtual reality	6
2.4. Raspberry pi Pico.....	7
2.5. Operational Amplifiers	8
2.5.1. Parameters of an Operational amplifier	9
2.6. IR Track in VR space	10
2.6.1. VR tracking system.....	10
2.6.2. TS4231.....	12
2.6.3. BPW34S Photodiode	13
2.7. Ultrasonic Transducer	14
2.7.1. Single Element Transducers	14
2.7.2. Piezoceramic Element.....	15
2.7.3. Housing	15
2.7.4. Matching Layer	15
2.7.5. Backing Layer	16
2.8. Schmitt trigger	16
3. Planning	18
3.1. Transmitter circuit.....	18
3.2. Receiver circuit	18
3.3. Board for the TS4231 chip.....	19
3.4. US receiver board design.....	19
3.5. The marker device board design	20
3.6. Software.....	21
3.6.1. Point visualization on the screen	21
3.6.2. Visualization in VR space.....	22
4. Implementation	22
4.1. Printed circuit boards.....	22
4.2. Programming the microcontrollers	22
4.2.1. Code for processing the TS4231 Chip's data.....	22
4.2.2. The marker device's code	23

4.2.3. Head mounted device's code.....	24
5. Results	24
6. Summary.....	25
7. Future plans	27
8. References	28

Abstract

This manuscript summarizes the main principles and developed components of an ultrasound (US)-based positioning and localization system for augmented reality environment. The long run goal of this work is to develop a complete system which can be put into operation in maritime search and rescue, to overcome poor visibility conditions.

The system has two parts: a marker device and a head-mounted device (HMD). The marker device has one transmitter and one receiver, while the HMD has one transmitter and three receivers. As a measurement protocol, the HMD's transmitter sends a pulse to the marker device, which responds with another pulse to the HMD. The three receivers of the HMD can then localize the relative position of the marker device with the help of triangulation.

An infrared (IR)-based measurement system is also designed and developed, similarly to the tracking system of Valve's VR: primarily to validate my US-based system, but later on it will become a standalone environment. This IR system utilizes the second-generation lighthouses from the VR kit and the TS4231 chip from Triad.

1. Introduction

Seeing in a bad environment condition could highly increase the performance and success rate of emergency operations. In 2019 there was a boating accident on the flooded Danube. A smaller ship called 'Hableány' collided with a bigger boat and sank immediately. The biggest problem of the rescue team was the zero underwater visibility. On this motivation, I started to develop an ultrasound-based localization system. The system consists of two parts. One is a device that acts as a marker device and a Head-mounted Display with three receivers. My goal was to localize these marker devices with the ultrasound receivers mounted to the HMD and display their relative location in the VR/AR space. I have researched sound localization in both animals and computer science to achieve this result. After this, I started to design the required circuits. I needed a driver circuit for the transmitter and an operational amplifier to preprocess the incoming signal before connecting it to the microcontroller. I used the recently launched Raspberry Pi Pico W board for this purpose and programmed it with the Arduino IDE.

To validate my results and make my system more accurate I started to work with IR based systems as well. This type of localization is already used in the most common VR environments, such as SteamVR. This type of systems consists of one or more IR transmitter stations called 'lighthouse', and multiple IR receiver circuit integrated into the tracked device. The lighthouses send Differential Manchester encoded data via IR which has to be processed on the tracker side [1]. From this data the relative position of a tracking device can be determined. In the tracker system I used the TS4231 chip which is an ASIC for this operation. This type of chip is commonly used in every VR tracking related products. The IC amplify and preprocess the incoming signal from the photodiode. The preprocessed signal then has to be processed with a microcontroller. In the consumer sector these trackers usually use FPGA-s to process the data [1], but instead of an FPGA I used a Raspberry Pi Pico W here as well. To calculate the position in the VR space, the only problem we must solve, is to determine the time difference between the two incoming signal [1].

In this work I first going through the basics of sound localization, and the basic principles of the used systems and components. After the literature research I will describe the planning phase of the project. There I describe the used printed circuit design and circuit diagrams. After the planning I also show the used software in this project. In the implementation phase I will describe the different microcontroller codes I used to solve the problems.

2. Literature research

2.1. Sound Source Localization

The goal of sound or ultrasound localization is to determine the position of an acoustic source using two or more receivers or microphones. In my case, the acoustic source is an ultrasound transducer, and my goal was to extract its 3D location in space. I used 3D localization and Direction of Arrival Estimation (DOAE) methods.

DOAE methods are based on the fact that the two (or more) receivers are at distinct differences from the source. This means that the signal reaches them at different times. Based on the time difference of arrival (TDOA), the direction of the source can be estimated. The estimation of the source can be obtained through the use of the far-field model. This model assumes that the receivers are far enough from the source to allow the spherical wave propagation to be approximated by planes (Figure 1) [2].

Based on the model, we can see, that the direction of the source (θ) is given by this equation, where T is the delay between the two receivers, c is the speed of sound and δ is the distance between the receivers [2]:

$$\theta = \cos^{-1}(\tau / \delta) = \cos^{-1}(cT / \delta)$$

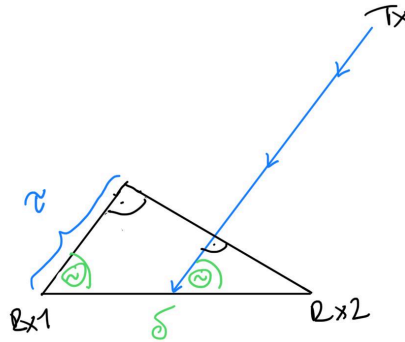


Figure 1: 2D DOAE far-field model: Rx1 and Rx2 stand for the two receivers, delta is the distance between them, Tx is the transmitter, T means the signal delay to the two different receivers, tau is the same difference as distance traveled, theta is the direction of the source

3D localization can be decomposed into two problems of 2D localization [2].

2.2. Sound localization in human

Humans can localize sound based on three primary cues. These are the interaural time difference (ITD), the interaural level difference (ILD), and the Head-related transfer function (HRTF). ITD detection happens in the auditory cortex and at the superior olivary. These regions receive input from both ears. The signal location is coded by the firing pattern of the panoramic

neurons. With this information, we can determine the azimuth angle of the source. To judge the elevation, we can tilt our head [3].

2.3. Virtual reality

Virtual reality (VR) is a computer-generated environment with scenes and objects that appear to be real. This makes the user feel they are immersed in their surroundings. This environment is perceived through a head mounted display (HMD). This is basically one or two small display and a lens system which projects the image into the user's eyes [4]. This technology allows us to immerse ourselves into a virtualized environment such as a sport training or a simulation of a high-risk operation (e.g., heart surgery). With this method it is possible to practice the otherwise dangerous or high-risk situations [5].

Augmented reality (AR) can work with the same hardware described at Virtual Reality, but instead of making a new 'world' that we immerse ourselves with through the HMD, we use the real world as a framework. In AR we enhance the real world by visual elements, sound or other sensory stimuli. The main goal of this technology is to highlight specific features, increase understanding of those features, and derive smart and accessible insight that can be applied to real-world applications. During my project I mainly focused on this type of concept [6] [5]. My goal was to visualize the relative positions of the marker devices through the HMD, and with this information help the user's work in bad environment conditions. Basically, my device extends the world in a way that the user can 'see' in bad visibility as well.

Mixed Reality (MR) is the combination of the previously described realities. This technology makes it possible to see virtual objects in the real world and build an experience in which the physical and the digital are practically indistinguishable. In MR you can interact with and manipulate both physical and virtual items and environments. [5] [7]

During the resource I used a Valve Index VR kit (Figure 2), which consist of two controllers, two second-generation lighthouses and one HMD.



Figure 2: Valve index VR kit [8]

2.4. Raspberry pi Pico

Last semester I used the Arduino nano to capture the incoming signals through interrupt-based programming, but I found it instable, so I started to look after alternatives. Last year (2021) Raspberry launched its microcontroller board, the raspberry pi Pico family. This microcontroller is built around the RP2040 silicon platform. The RP2040 is a high-performance, low-cost chip, and easy to use. The chip has two symmetric processor cores, and large on-chip memory. These two features can come handy when we are dealing with interrupt-based programming. This board is fully compatible with the Arduino IDE environment so it can be programmed like an Arduino, but in the end, we get a more stable result [9].

With detailed documentation, a polished Micro Python port, and a UF2 bootloader in ROM, it has the lowest possible barrier to entry for beginner and hobbyist users [9].

RP2040 is manufactured on a modern 40nm process node, delivering high performance, low dynamic power consumption, and low leakage, with a variety of low-power modes to support extended-duration operation on battery power [9].

Raspberry Pi Pico pairs RP2040 with 2MB of Flash memory, and a power supply chip supporting input voltages from 1.8-5.5V. It provides 26 GPIO pins, three of which can function as analogue inputs [9]. The pinout of the board can be seen in Figure 3.

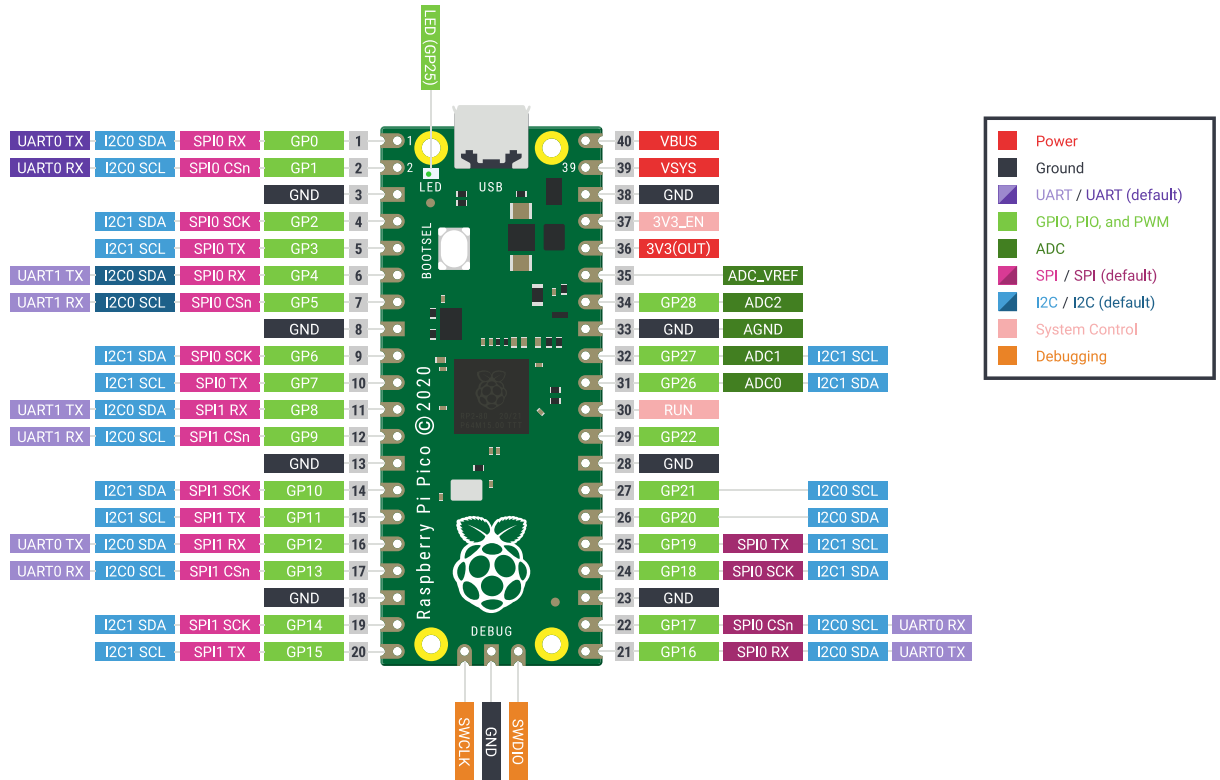


Figure 3: The Raspberry Pi Pico W board pinout [10]

2.5. Operational Amplifiers

Operational amplifiers or op-amp is a very important part of my design. The incoming low voltage signal must be amplified to make it processable to the microcontroller board. On this reason I have done some research on different type of operational amplifiers.

An Operational Amplifier, or op-amp for short, is fundamentally a voltage amplifying device designed to be used with external feedback components such as resistors and capacitors between its output and input terminals. These feedback components determine the resulting function or “operation” of the amplifier and by virtue of the different feedback configurations whether resistive, capacitive or both, the amplifier can perform a variety of different operations, giving rise to its name of “Operational Amplifier” [11].

An Operational Amplifier is a three-terminal device which consists of two high impedance inputs. One of the inputs is called the Inverting Input, marked with a negative sign (–). The other input is called the Non-inverting Input, marked with a positive sign (+) [11]. The circuit diagram of this device can be seen in Figure 4.

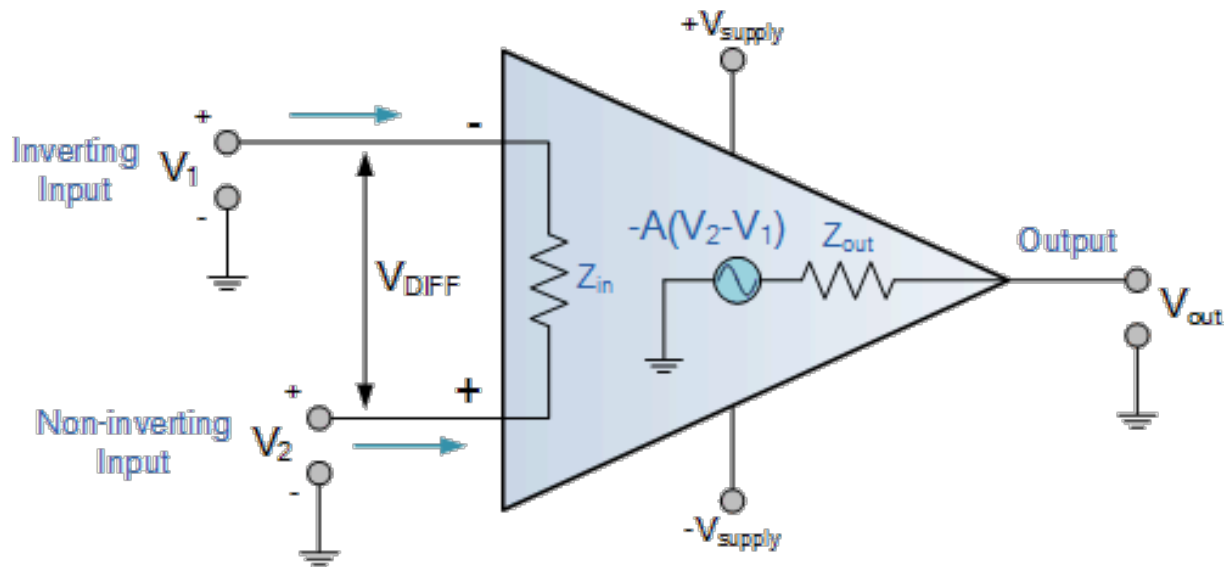


Figure 4: Circuit diagram and equivalent circuit of an ideal operational amplifier [11]

The third terminal represents the operational amplifiers output port which can both sink and source either a voltage or a current. In a linear operational amplifier, the output signal is the amplification factor, known as the amplifiers gain (A). Multiplied by the value of the input signal and depending on the nature of these input and output signals, the operational amplifier gain can be classified into four different groups: Voltage – Voltage in and Voltage out; Current – Current in and Current out; Transconductance – Voltage in and current out; Transresistance – Current in and Voltage out [11].

The output voltage signal of an Operational Amplifier depends on the difference between the signals being applied to its two individual inputs. This means that, an op-amps output signal is the difference between the two input signals [11].

2.5.1. Parameters of an Operational amplifier

Open Loop Gain, (A_{vo}): The main function of an operational amplifier is to amplify the input signal and the more open loop gain it has the better. Open-loop gain is the gain of the op-amp without positive or negative feedback and for such an amplifier the gain will be infinite but typical real values range from about 20 to 200 [11].

Input impedance, (Z_{IN}): Input impedance is the ratio of input voltage to input current and is assumed to be infinite to prevent any current flowing from the source supply into the amplifiers input circuitry ($I_{IN} = 0$). Real op-amps have input leakage currents from a few pico-amps to a few milli-amps [11].

Output impedance, (Z_{OUT}): The output impedance of the ideal operational amplifier is assumed to be zero acting as a perfect internal voltage source with no internal resistance so that it can supply as much current as necessary to the load. This internal resistance is effectively in series with the load thereby reducing the output voltage available to the load. Real op-amps have output impedances in the 100Ω - $20k\Omega$ range [11].

Bandwidth, (BW): An ideal operational amplifier has an infinite frequency response and can amplify any frequency signal from DC to the highest AC frequencies, so it is therefore assumed to have an infinite bandwidth. With real op-amps, the bandwidth is limited by the Gain-Bandwidth product (GB), which is equal to the frequency where the amplifiers gain becomes unity [11].

Offset Voltage, (V_{IO}): The amplifiers output will be zero when the voltage difference between the inverting and the non-inverting inputs is zero, the same or when both inputs are grounded. Real op-amps have some amount of output offset voltage [11].

Based on these parameters I choused the MCP6024 Rail-to-Rail operational amplifier. The packaging of the chip contains four op-amps. It has an Open-Loop Gain of 110 dB, a wide 10 MHz bandwidth which is enough to amplify the incoming 40 kHz signal. This chip has a low offset voltage ($500\mu V$ at industrial temperature or $250\mu V$ at extended temperature) [12].

2.6. IR Track in VR space

2.6.1. VR tracking system

To use the tracking system of my VR headset, first I had to understand the working principals of the system. After this I was able to recreate the tracking system implemented by Valve with the second-generation lighthouses (Figure 5).



Figure 5: first-generation lighthouse internals (left) and second-generation lighthouse internals (right) [1]

The older first-generation lighthouses have two rotors with IR emitters on them and a bright, wide angle IR emitter. Trackers could determine their location relative to the lighthouse based on a flash-sweep method. It sends out a bright flash to all the sensors in the area. And then passes a sweep (either horizontal or vertical) over the sensors immediately following the flash. The trackers determine the time between the flash and the corresponding sweep to determine what angle that sensor is in relation to the lighthouse. With two lighthouses (which requires syncing between the two) and some sensor geometry, one can determine a position in the area of covered by the lighthouses [1].

On the other hand, with the second-generation lighthouses, there is only a single rotor, and it doesn't use flash cycle. The sensors have increased accuracy and doesn't require syncing between lighthouses. This means, that tracking can be achieved by a single lighthouse as well. The two major changes that allow this are that data is now encoded into the sweep IR, and the IR emitters that were providing the horizontal or vertical sweeps in the first-generation model are now on the same rotor with a 90-degree offset from each other [1].

In the second-generation lighthouse, a Linear Feedback Shift Register (LFSR) has been implemented to generate a very long, unique, and eventually cyclic bit sequence. LFSR is a shift register whose input bit is a linear function of its previous state [13]. It uses 20 bits tap sequence that represents a polynomial for generating the output bit sequence. The 20 bits tap sequence determines which bits from the active LFSR sequence get XOR-ed to determine the next bit. That bit is fed into the shift register, and the first bit gets popped off and becomes the next bit of the output sequence. The lighthouses have 16 channels, and each channel has 2 polynomials that are used in an alternating fashion for every other full rotor rotation. After the LFSR is determined, the data is encoded with Differential Manchester Encoding. This converts one bit into an edge (or no edge). A bit value of 0 is converted into an edge, and a bit value of 1 is converted into no edge (Figure 6) [14]. There is also always an edge between each bit [1].

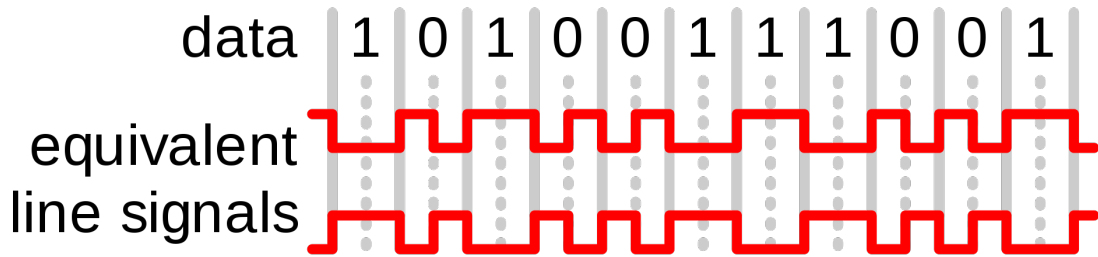


Figure 6: An example of Differential Manchester encoding: Gray vertical lines, full and dotted, represent the two clock ticks per bit period [14].

In the trackers, the IR sensor (BPW34S) [15] is connected to the TS4231 chip by Triad Semiconductors. The TS4231 needs to be reconfigured on every power cycle for it to work with second-generation lighthouses. The chip has two lines: data (D) and envelope (E) [15]. From the data line (D) we can read the differential Manchester encoded data. The envelope line (E) turns low when the IR sensor senses a sweep. Based on this information we can tell if the incoming data is live. This means, that the data coming from the data line (D) is only valid, when the envelope line (E) is low [1].

2.6.2. TS4231

The TS4231 is Triad’s second generation Light-to-Digital Converter IC. The TS4231 contains all the features of the TS3633 while adding a DATA output pin that tracks the modulated light carrier input. The TS4231 also includes a two-wire bus that shares the D/E output of the device to provide for configuration of the device and provides the ability to dynamically put the device in a power savings mode. The TS4231 is fully compatible with Valve Corporation’s SteamVR Tracking System. The basic block diagram of the chip can be seen in Figure 7. [15].

The addition of the DATA output pin means that higher bit-rate digital data can be communicated from an optical source such as a SteamVR Tracking base station to the TS4231. This higher speed digital link opens the capability of using more than two base stations. Multiple base stations installations are useful for digital out of home experiences, arcades, “house-scale” tracking, and non-VR tracking applications such as warehouse-scale or grocery-store-scale tracking [15].

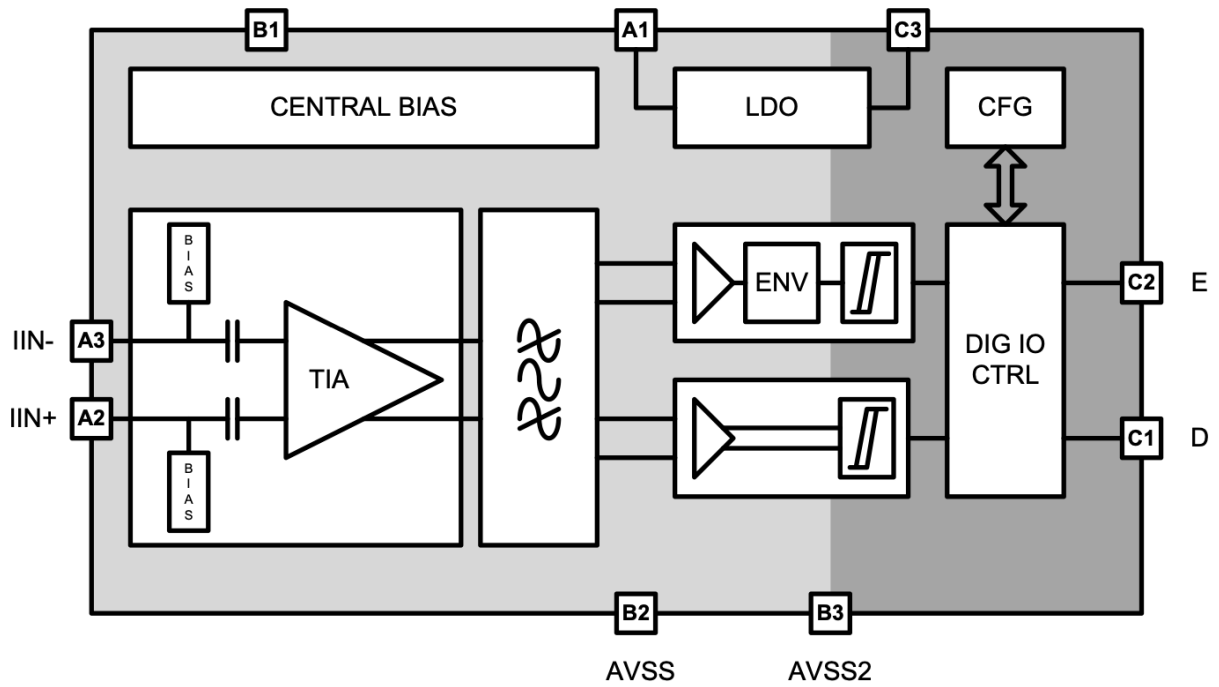


Figure 7: TS4231 Basic Block Diagram [15]

2.6.3. BPW34S Photodiode

BPW34 is a PIN photodiode with high speed and high radiant sensitivity in miniature, flat, top view, clear plastic package. It is sensitive to visible and near infrared radiation. The diode has a 7,5 mm² radiation sensitive area and a $\pm 65^\circ$ angle of half sensitivity (Figure 8) [16]. This type of diode is commonly used in consumer VR products with Triad's TS4231 chip.

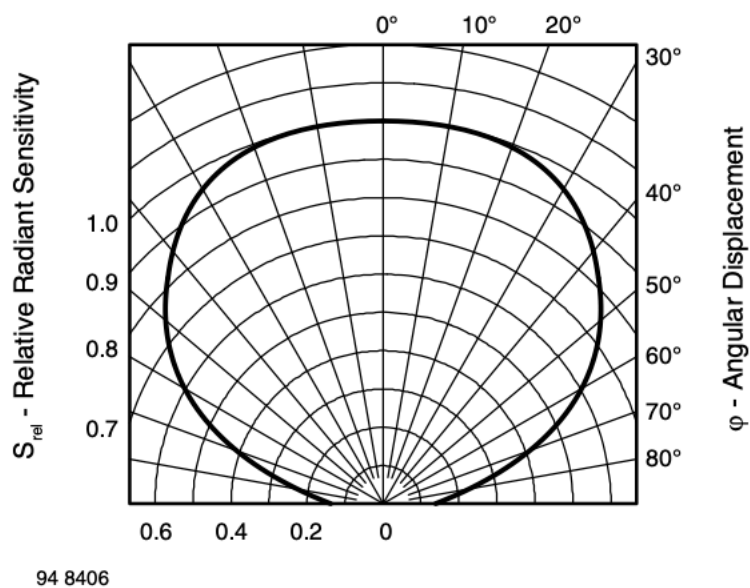


Figure 8: Relative Radiant Sensitivity vs. Angular displacement of the BPW34S photodiode [16]

2.7. Ultrasonic Transducer

Transducers are converting energy form one form to another. I case of a piezoelectric transducer it means an electrical to mechanical or mechanical to electrical energy conversion. This conversion can take place in various medium such as air or water. Based on the medium we will need different transducer design [17].

2.7.1. Single Element Transducers

Single element transducers are often the most cost-effective solution, especially when experimenting with transduction for specific applications. Single element transducers contain one active ceramic element, fewer than the multiple element or pre-stressed stack transducers. This often results in a reduced cost and simplicity in design. They can be easily designed as transmitters, receivers, transceivers, etc. Applications for single element transducers include medical, therapeutic, NDT (non-destructive testing), flow/level sensing, air transduction or distance measurement [17].

Single element transducers are made up of 4 primary components, the piezo-ceramic element, the backing layer, the matching layer, and the housing/cable assembly as seen in Figure 9. [17].

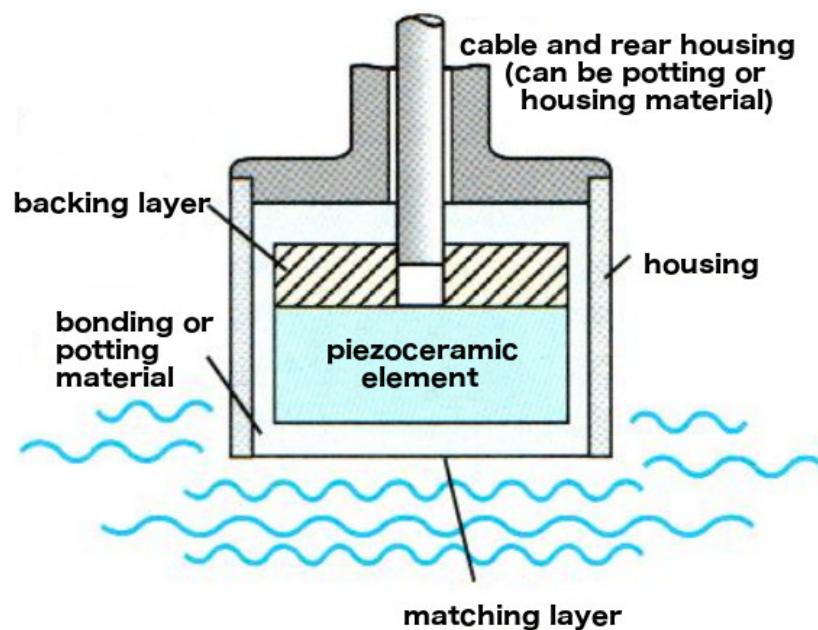


Figure 9: Single element transducer primary components [17]

2.7.2. Piezoceramic Element

In a single element transducer, the ceramic element generates or receives the pulse and often operates at resonance. This simplifies the design and optimizes the performance. The ceramic element can also help determine the physical size of the transducer and its housing. When choosing a ceramic element, start by thinking about what it is you are trying to accomplish. Whether transmitting, receiving, a simple pulse echo, or a complex crack detection, the application dictates the power requirements and beam angle, therefore setting the PZT material composition requirement (hard for high power applications vs. soft for low power and sensing applications) and physical size (physical size sets frequency and beam angle) [17].

2.7.3. Housing

After selecting the piezoceramic element based on the criteria described above, we must choose the housing of the element. Here we must keep in mind the final application of the transducer. It must withstand the environment conditions. For example, corrosion, high/low temperature, or different chemicals. Common housing materials can be various types of aluminums, stainless steels, plastics, or other composite materials [17].

2.7.4. Matching Layer

The purpose of the matching layer is to provide a material “buffer” from the acoustic impedance of the ceramic element to the acoustic impedance of the medium which the generated signal will be travelling through. The acoustic impedance of a material is defined as the density of the material times the speed of sound of the material. If the acoustic impedance mismatch is small enough, a single matching layer can be used. However, if the acoustic impedance mismatch is large, several matching layers may work together to gradually transition from the impedance of the ceramic to the impedance of the target medium. In general, the higher the number of iterative steps in the matching layer and the closer the final match to the target medium, the higher the efficiency. Single matching layer efficiency can be in the range of 10-50%, while multiple layers can increase that efficiency by 25% or more [17].

Also important is the speed of sound of the material used for the matching layer. It is very common to utilize matching layers in $1/4$ wavelength increments to additionally enhance transmission efficiency and promote operation at resonance [17].

2.7.5. Backing Layer

The backing layer is a highly attenuative layer designed to direct energy out the front of the transducer and eliminate energy loss from the backside of the ceramic. The backing layer can be made of various materials, including air. Backing materials can also be used to control pulse resolution, at the expense of transducer efficiency. When choosing a material for backing a transducer, it is important that the density be high for the purpose of acoustic impedance matching of the ceramic element, but the layer must also be attenuative. When backing with air, the thickness of the air layer is not critical, however if backing with another material the thickness will need to be determined by the material properties, density, and attenuation. A properly selected backing layer with high acoustic impedance can increase the transducers signal quality by absorbing and damping the back-end signal from the ceramic element [17].

If backing with air and potting the transducer, an air substitute must be used as a placeholder and is often made up of very specific foams or cork materials [17].

I used the T40-16 and R40-16 matched ultrasonic transmitter and receiver pairs for the communication. These devices are operated at a 40 kHz center frequency. The transducer utilizes ceramic's piezoelectrical properties, providing high sound pressure and sensitivity. The transmitter has a minimum of 120 dB sound pressure level, and the receiver has -59 dB/v/ubar sensitivity at the center frequency [18].

2.8. Schmitt trigger

The Schmitt Trigger is a logic input type that provides hysteresis or two different threshold voltage levels for rising and falling edge. The Circuit diagram of a Schmitt trigger can be seen in Figure 10 This is useful because it can avoid the errors when we have noisy input signals from which we want to get square wave signals [19].

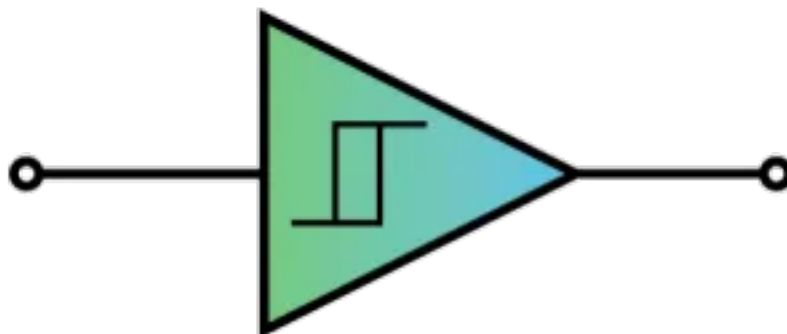


Figure 10: Circuit diagram of a Schmitt trigger [19]

If we have a noisy input signal like this, that is meant to have 2 pulses, a device that has only one set point, or threshold, could get incorrect input and it could register more than two pulses as shown in this illustration. If we use the Schmitt Trigger for the same input signal, we will get a correct input of two pulses because of the two different thresholds. This is the primary function of the Schmitt Trigger, to convert noisy square waves, sine waves or slow edges inputs into clean square waves (Figure 11) [19].

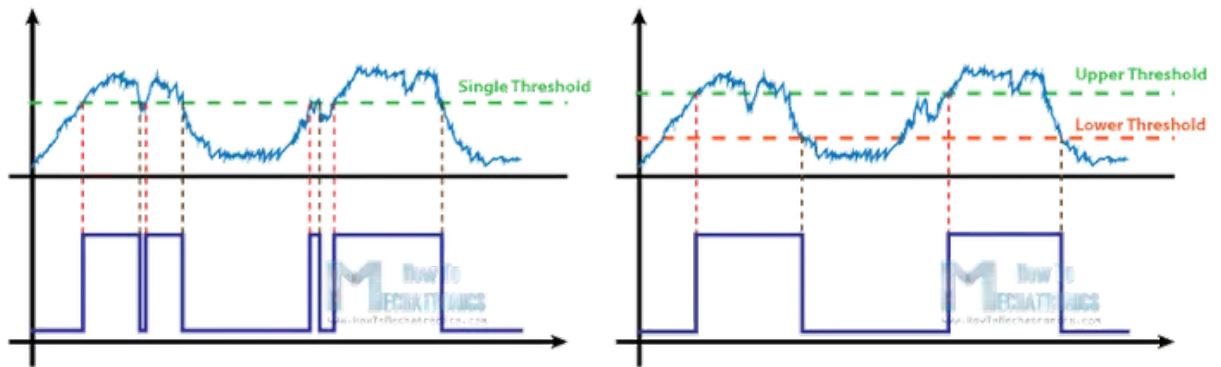


Figure 11: The working principal of a Schmitt trigger. On the top there is the noisy input signal. On the bottom we can see the output square wave signal based on the thresholds. [19]

The 74HCT14 is a high-speed Si-gate CMOS device. It provides six inverting buffers with Schmitt-trigger action. This device converts the input signal into a sharply defined, jitter-free output signal. Suppose the input voltage reaches the threshold of the Schmitt trigger and the output changes to the VCC voltage. Otherwise, it stays on GND. The pin diagram of the integrated circuit can be seen in Figure 12. [20].

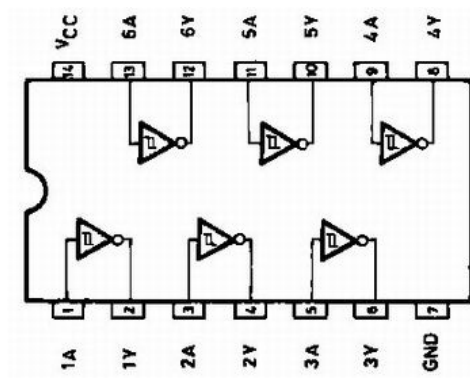


Figure 12: 74HCT14 pin diagram [17]

3. Planning

3.1. Transmitter circuit

To drive the ultrasound transmitter, I used an H bridge circuit. The circuit contains two BC182 NPN transistors and two BS557 PNP transistors. The H bridge is operated with a 40 kHz square wave signal generated by the Arduino nano board. The signal inversion is made by a BC182 transistor-based NOT gate. With this bridge circuit, I can make a 40 kHz 5V peak-to-peak output signal. The circuit diagram of the receiver can be seen in Figure 2

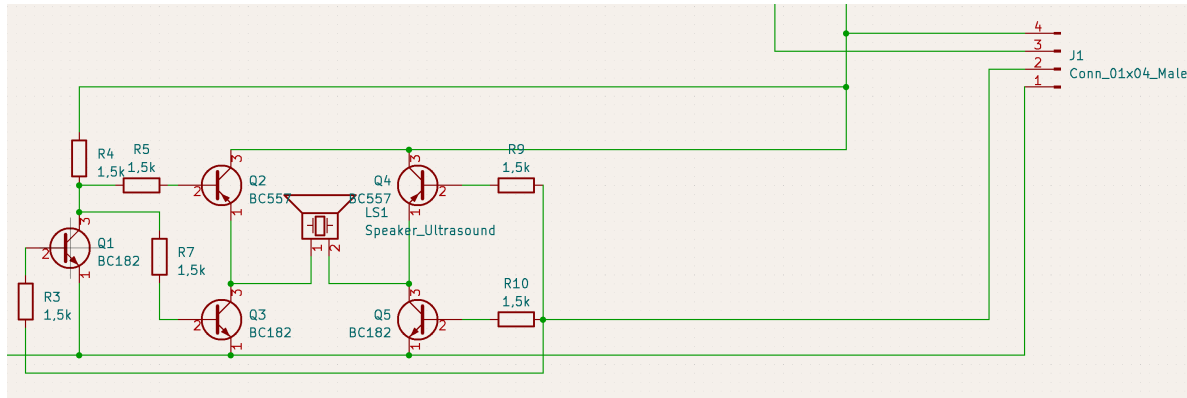


Figure 2: The transmitter circuit's diagram. It is an H bridge controlled with the incoming signal from J1's 2nd pin.

3.2. Receiver circuit

I used an MCP6024 operational amplifier IC and a 74HCT14 Schmitt triggers IC. I used two amplifier circuits for each ultrasound transducer. Each stage has a gain value of 33 (10k resistor on the input, 330k resistor feedback resistor). After the amplification, I converted the analog signal to a digital signal with the help of the Schmitt trigger. I also used two of them in series to get the initial phase of the input signal because the used Schmitt trigger has a built-in NOT gate. In the previous semester I used an Arduino Nano board to process the incoming signal, but it was instable. On this reason I replaced the Arduino with the new Raspberry Pi Pico W board, which has a two-core processor, and a much better interrupt handling, based on my experience. The output of this circuit is connected to one of the GPIO of the Raspberry Pi Pico W microcontroller. The described circuit can be seen in Figure 3 The Raspberry Pi Pico W board input is connected to the 2nd and 3rd pin of the J1 connector. Pin 4 is connected to 5V, and pin 1 to GND.

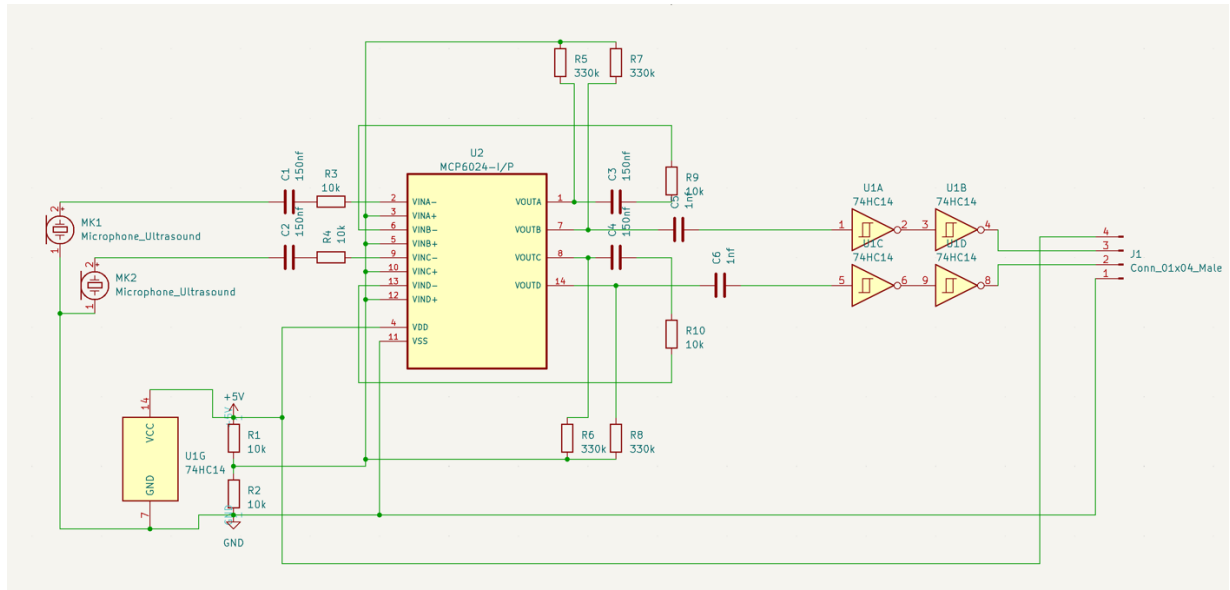


Figure 3: Receiver circuit for two ultrasound receivers

3.3. Board for the TS4231 chip

The TS4231 chip is a Light-to-Digital Converter IC, which is responsible for the integration into the SteamVR Tracking system. In this project I designed a board for this chip and for the IR photodiode based on the reference design provided by the chip manufacturer. Each board has a dimension of 8,5mm x 9,5mm. With this size, the detectors are easily placeable on the device surface. The board has four connectors. From left to right: 3,3V; D (Data output); E (Envelope Output); GND (Figure 15).

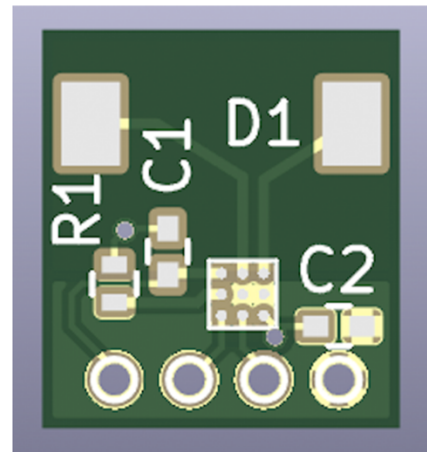
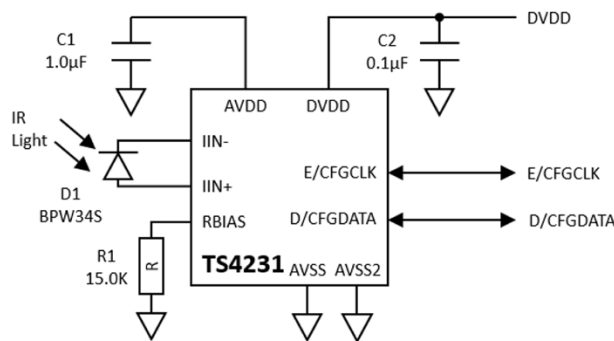


Figure 15: TS4231 reference circuit design from the datasheet (Left) [15]. The designed board for the TS4231 (Right)

3.4. US receiver board design

For the receiver I made a board design which hosts the necessary parts for two US receivers. The reason for this was the fact that the MCP6024 IC has four Operational

amplifiers integrated into it and for one US receiver I needed two. With KiCad 6.99 I designed a single layer printed circuit board for this and used vias on the component side to connect the otherwise non-reachable points of the circuit. I tried to minify the number of vias to reduce every possible noise factor in my design. The unused parts of the PCB are connected to GND. The design can be seen in Figure 16 This board is connected to the Raspberry Pi Pico W microcontroller via the J1 connector.

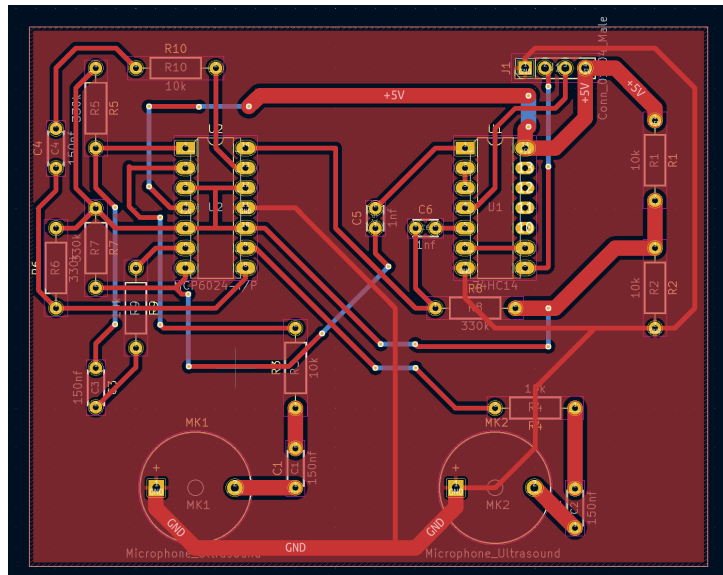


Figure 16: US receiver board design, designed in kiCad 6.99

3.5. The marker device board design

On the marker device I used a transmitter and a receiver circuit. The only functionality of this system is to transmit a signal, if the receiver received one from the HMD side. For this I placed a transistor-based H-bridge and a receiver circuit on this board. The finished board shown in Figure 17 has an US transmitter on the left, and an US receiver on the right. The Raspberry Pi Pico W microcontroller is connected via the connectors on the bottom of the board.

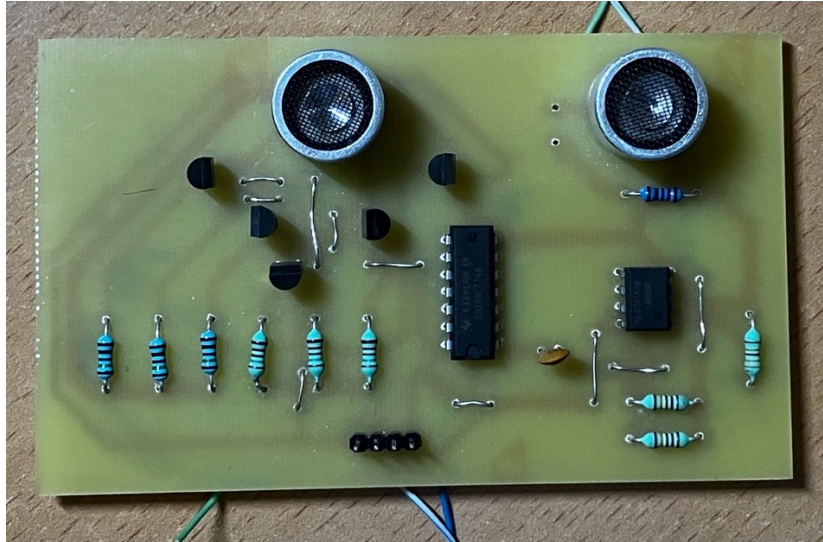


Figure 17: The marker device PCB with the components soldered in it.

3.6. Software

3.6.1. Point visualization on the screen

First, to visualize the incoming data from the microcontroller I made a simple Python based application to plot the incoming data. This program connects to the device and reads the serial data on a different thread to keep the GUI responsive. The incoming data represents the time difference between the transmission and reception in each receiver. The program converts it first into spherical polar coordinates, then to 3D Cartesian coordinates to visualize it on the 3D plot. The main windows of the software can be seen in Figure 4.

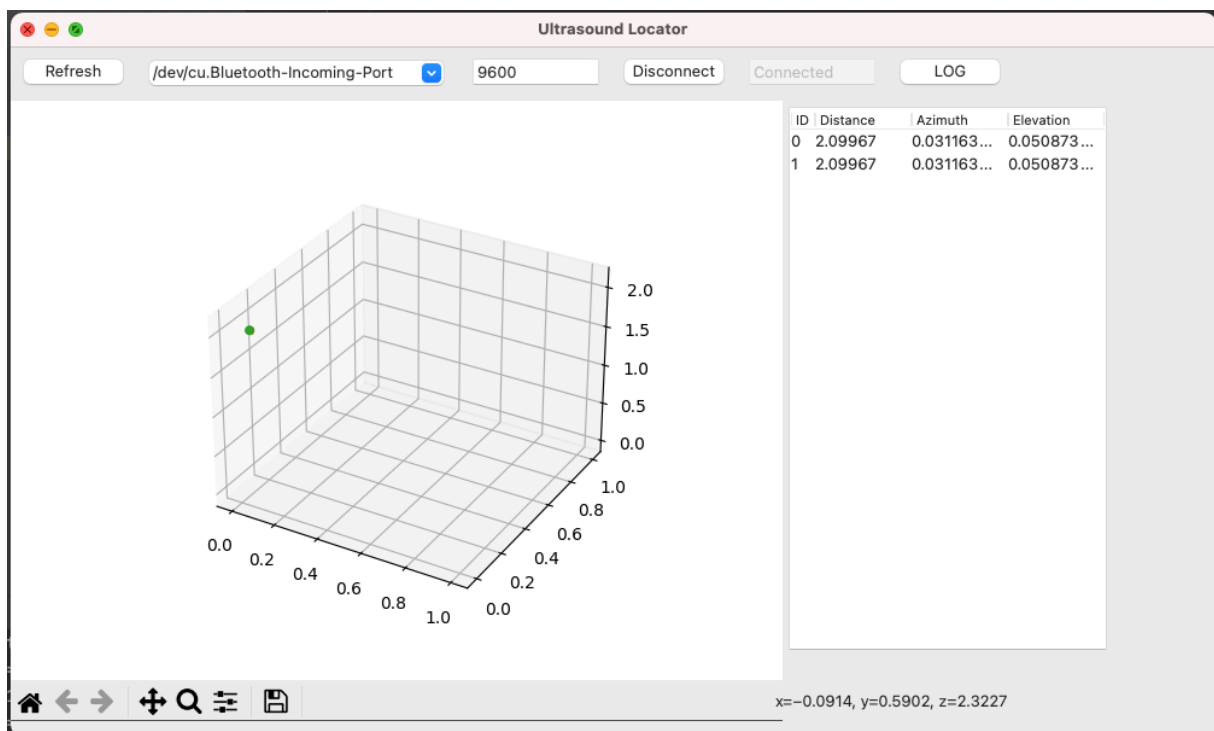


Figure 4: GUI for data visualization

3.6.2. Visualization in VR space

To visualize the point in the VR space, first I developed a custom driver for my device. This driver is based on the [terminal29/Simple-OpenVR-Driver-Tutorial](#) github repository's `TrackingReferenceDevice` example. Here I modified the file to use the incoming data from the serial port as the tracking device's position. Because the incoming coordinates are relative coordinates to the headset, I used the headset coordinates as an offset.

After the driver development I created a project inside Unity, which is a popular game engine, to display the location of my Tracking device connected through my custom driver.

4. Implementation

4.1. Printed circuit boards

Most of the boards are etched by me. The only exception was the board for the TS4231 chip, because of its size and layer count. Single layered boards are easily implementable by hand in low quantity. I used hydrogen peroxide and hydrochloride acid to etch the boards. I transferred the design with coated paper via heat. After the etching phase the ink from the laser printer can be washed by acetone. To protect the copper layer from oxidation I chemically coated it with tin by the TN160 solution.

The board for the TS4231 chip was ordered from a PCB manufacturer company because the previously describe technique has lower accuracy than the pad distance on the design and making double layer PCB with this method is a bit challenging especially in this small form factor.

4.2. Programming the microcontrollers

To program the Raspberry Pi Pico W, I used the Arduino IDE environment. In this way I could reuse my previous code on a much stable microcontroller.

4.2.1. Code for processing the TS4231 Chip's data

The firmware for the tracker needs to capture the data coming from the TS4231. Most of the consumer products uses an FPGA to capture and decode the data coming from the TS4231. Instead of using an FPGA I used a Raspberry Pi Pico W board to decode the incoming data. I used interrupt-based programming to capture the time between pulses. The LFSR on the lighthouse is generating the bit sequence at a rate of 6 MHz, and after encoding the final data

speed being sent over IR is 12 MHz [1]. This means that I have 83 nanoseconds to read and store a bit of the data. I triggered a timer in the Raspberry Pi Pico W with data edges. This counts the time between each edge. Since the differential Manchester encoding has an edge between every bit, it is easy to convert from pulse length to decoded bits [1] [14]. Each bit represented by 166 nanoseconds stretch. If there are no edges in that time period, it's a 'long' (no change, bit value of 1). If there is an edge in that time it will look like a 'short-short' (change, bit value of 0) [1].

After the slow data is captured. There is a large enough time as the rotor spins back to its initial position that the data can be interpreted. First step is to determine what polynomial is being used. This is an intensive process that loops through different known polynomials and uses an LFSR to try and generate a sequence of bits that matches the captured data. Once the polynomial is determined, you can then reference how many iterations into the LFSR it took you to get to that bit sequence in order to determine the angle of the sensor from the lighthouse. From there the process of determining sensor geometry and calculating your position from the lighthouse is the same as it was with first-generation lighthouses [1].

4.2.2. The marker device's code

As I mentioned before, this code is written in the Arduino IDE. The goal of this algorithm is to send out five ultrasound beams if the receiver receives the signal from the head-mounted device's transmitter. To achieve this result, I used interrupt-based programming. In practice, this means that I configured pin 2 of the Raspberry Pi Pico W board to listen for the rising edge of the connected signal. To use interrupt, first I called the `attachInterrupt(digitalPinToInterrupt(2), pinTwo, RISING)` method in the setup function. This line tells the microcontroller to listen on pin 2, and call function `pinTwo` if the rising edge of the signal is detected.

In the `pinTwo` function, I used a for loop to output the five 40 kHz beams. To achieve this frequency, I switched pin 10 ON and off with a delay of 9 ms. After uploading the code to the device, I validated the output signal with an oscilloscope.

4.2.3. Head mounted device's code

This device controls the measurement, and it's connected to the computer via the serial bus. The goal here was to initialize the measurement by sending out an ultrasound beam. After the sending, the receiving is also interrupt-based like in the marker device's code. Here instead of transmitting, the callback function writes the elapsed time from the transmission to the receipt to the serial bus.

5. Results

To validate the system's functionality, first I built a simplified test environment. The setup has two parts. I placed the marker device on the desk and the locator device on a tripod. I put the three receivers and the transmitter in a triangular alignment on the tripod. For this, I used a piece of wood and an Allan key. The two receivers on the wooden part are 20 cm apart from each other; the third receiver is 6 cm higher than the other two (Figure 19). I used shielded audio cables to connect the receivers and the amplifier circuit. Because the Arduino Nano only has two interrupt programable pins, I used two of them to capture data from all three receivers. During the tests, I noticed that my amplifier circuit was too sensitive. Even a tiny movement of the cable triggered the interrupts on the Arduinos. For this reason, I tried to fix the cables with hot glue, which made the system more stable. I made multiple measurements from different relative locations to the marker device. The results are shown in Table 1 The column "Actual distance" is measured by a 3 m tape measure, and the "actual angles" are calculated with the following formula: $Azimuth = \cos^{-1}\left(\frac{center}{Left}\right)$ and $Elevation = \cos^{-1}\left(\frac{center}{right}\right)$.

US measured distance (cm)			Actual distance (cm)			Angles based on UH (degree)		Actual angles (degree)	
Left	Center	Right	Left	Center	Right	Azimuth	Elevation	Azimuth	Elevation
94,1	93,42	95,231	96,5	95,5	98	6,8922	11,191	8,2556	12,969
102,29	99,921	103,64	101,3	100	102	12,355	15,3955	9,189	11,3648
144,2	143,98	145,32	145,5	144,4	147,5	3,1653	7,7868	7,049	11,7675

Table 1: The results measured in the test environment

During the tests, I tried to fool the system to check its limitations. The biggest drawback of my approach occurred when I placed a sheet between the two devices. In this case, the results only represented the distance to the sheet because the receivers only captured the reflected ultrasound beam. This means that the system is only usable if the two devices can see each other without any interruptions.



Figure 5: Ultrasound locator test setup. The two receivers on the wooden part are 20 cm apart from each other; the third receiver is 6 cm higher to the other two. In the center the left one is the receiver; the right is the transmitter.

With the new IR based validation system I didn't do any measurement yet. I only validated if the assembled components are working correctly. The test environment is already planned, but at the time of writing there are no available result from this system.

6. Summary

Seeing in a bad environment condition could highly increase the performance and success rate of emergency operations. On this motivation, I started to develop an ultrasound-based localization system. The system consists of two parts. One is a device that acts as a marker device and a Head-mounted Display with three receivers. My goal was to localize these marker

devices with the ultrasound receivers mounted to the HMD and display their relative location in the VR/AR space. I have researched sound localization in both animals and computer science to achieve this result. After this, I started to design the required circuits. I needed a driver circuit for the transmitter and an operational amplifier to preprocess the incoming signal before connecting it to the microcontroller. I used the recently launched Raspberry Pi Pico W board for this purpose and programmed it with the Arduino IDE.

To validate my results and make my system more accurate I started to work with IR based systems as well. This type of localization is already used in the most common VR environments, such as SteamVR. This type of systems consists of one or more IR transmitter stations called ‘lighthouse’, and multiple IR receiver circuit integrated into the tracked device. The lighthouses send Differential Manchester encoded data via IR which must be processed on the tracker side [1]. From this data the relative position of a tracking device can be determined. In the tracker system I used the TS4231 chip which is an ASIC for this operation. This type of chip is commonly used in every VR tracking related products. The IC amplify and preprocess the incoming signal from the photodiode. The preprocessed signal then had to be processed with a microcontroller. In the consumer sector these trackers usually use FPGA-s to process the data [1], but instead of an FPGA I used a Raspberry Pi Pico W here as well. To calculate the position in the VR space, the only problem we had to solve, is to determine the time difference between the two incoming signal [1].

In this work I first went through the basics of sound localization, and the basic principles of the used systems and components. After the literature research I described the planning phase of the project. There I described the used printed circuit design and circuit diagrams. After the planning I have also shown the used software in this project. In the implementation phase I described the different microcontroller codes I used to solve the problems.

Based on my result, we can see that the developed system can locate the source of a 40 kHz ultrasound signal. The system was unable to handle objects between the two devices. I think this problem is solvable if I not only use the time of arrival but some significant modulation in the signal. This solution can also make it possible to use multiple marker devices simultaneously and differentiate their signals based on the received pattern.

7. Future plans

In the future I would like to implement a fully autonomous IR based system on the principals of the Valve tracking system. To implement this, I have to design a 'lighthouse' like device which will be mounted on the user's head as well as the ultrasound-based system. By combining the strength of the ultrasound and the IR based system we can achieve higher accuracy in the near future.

8. References

- [1] "LH2 tracker," [Online]. Available: <https://bespeland.github.io/portfolio/p3-LH2-Tracker/>. [Accessed 15 08 2022].
- [2] V. Kunin, M. Turqueti, J. Saniie and E. Oruklu, *Direction of Arrival Estimation and Localization Using Acoustic Sensor Arrays*, 3 ed., vol. 1, 2011, pp. 71-80.
- [3] "Sound Localization and the Auditory Scene," [Online]. Available: https://courses.washington.edu/psy333/lecture_pdfs/chapter12_SoundLocalization.pdf. [Accessed 20 05 2022].
- [4] "Head-mounted Displays (HMDs)," [Online]. Available: <https://www.gartner.com/en/information-technology/glossary/head-mounted-displays-hmd>. [Accessed 01 11 2022].
- [5] "Virtual Reality: another world within sight," [Online]. Available: <https://www.iberdrola.com/innovation/virtual-reality>. [Accessed 05 11 2022].
- [6] A. HAYES, "Augmented Reality (AR) Defined, with Examples and Uses," [Online]. Available: <https://www.investopedia.com/terms/a/augmented-reality.asp>. [Accessed 01 11 2022].
- [7] "VR vs. AR vs. MR: What You Need to Know," [Online]. Available: <https://www.intel.com/content/www/us/en/tech-tips-and-tricks/virtual-reality-vs-augmented-reality.html>. [Accessed 01 11 2022].
- [8] "Valve Index," [Online]. Available: <https://www.valvesoftware.com/hu/index>. [Accessed 04 11 2022].
- [9] "Raspberry pi pico documentation," [Online]. Available: <https://datasheets.raspberrypi.com/pico/pico-product-brief.pdf>. [Accessed 20 10 2022].
- [10] "Raspberry pi pico documentation 2," [Online]. Available: <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>. [Accessed 20 10 2022].
- [11] "Operational Amplifiers," [Online]. Available: https://www.electronics-tutorials.ws/opamp/opamp_1.html. [Accessed 05 11 2022].
- [12] "MCP6024 Datasheet," [Online]. Available: <https://ww1.microchip.com/downloads/aemDocuments/documents/APID/Product Documents/DataSheets/20001685E.pdf>. [Accessed 02 11 2022].
- [13] "Linear-feedback shift register," [Online]. Available: https://en.wikipedia.org/wiki/Linear-feedback_shift_register. [Accessed 28 10 2022].
- [14] "Differential Manchester encoding," [Online]. Available: https://en.wikipedia.org/wiki/Differential_Manchester_encoding. [Accessed 28 10 2022].
- [15] "TS4231 Documentation," [Online]. Available: <https://triadsemi.com/product/ts4231/>. [Accessed 21 08 2022].
- [16] "BPW34 Datasheet," [Online]. Available: <https://www.vishay.com/docs/81521/bpw34.pdf>.

- [17] "Anatomy of a Single Element Transducer," [Online]. Available: <https://www.americanpiezo.com/blog/after-manufacturing-part-3-anatomy-of-a-single-element-transducer/>. [Accessed 03 11 2022].
- [18] "TR40-16," [Online]. Available: https://www.hestore.hu/prod_getfile.php?id=10925. [Accessed 15 02 2022].
- [19] "What is Schmitt Trigger," [Online]. Available: <https://howtomechatronics.com/how-it-works/electrical-engineering/schmitt-trigger/>. [Accessed 11 10 2022].
- [20] "74HCT14," [Online]. Available: https://www.hestore.hu/prod_getfile.php?id=4928. [Accessed 20 04 2022].