2016

Project Poseidon

AN UNDERWATER ACOUSTIC MODEM

SPONSORED BY MECHATRONICS ROBOSUB

PREPARED FOR DR. R. LAL TUMMALA AND PROFESSOR JOHN KENNEDY

ENGINEERS

Blair Chantrill

Davy Phakdy

Emmanuel Palafox

Jeffrey Miller

John Salamat

Leban Ali

Mark Dimaano

Nutdanai Swangpattaraphon

Sebhat Yidelwo

Shideh Naderi

Trevor Blazek

Uros Obradovic

Bao Hoang

Brady Anderson

Cavan Page

Jonah Mariano

SAN DIEGO STATE UNIVERSITY | 5500 CAMPANILE DRIVE SAN DIEGO, CA 92182

CONTENTS

ABSTRACT

SDSU's Robosub team, Mechatronics, excelled last year in the AUVSI Foundation's International Robosub Competition, achieving first place with their AUV (Autonomous Underwater Vehicle) titled; Defiance. To better defend their title as the champs, they wish to expand to multiple AUVs to complete the designated course at a faster time. In order to effectively achieve this, they requested for an underwater communication system, enabling coordination between multiple AUVs. The AUVs will send periodic data packets to update each other on their status and current task. This senior design group will attempt to achieve this task by accomplishing the following. First, acoustic communication signals will be used to deliver the message as RF signals attenuate too much underwater. These signals will be generated using both handmade and off the shelf waterproof transducers. Both FSK (Frequency Shift Keying) and ASK (Amplitude Shift Keying) modulation techniques will be implemented for the purpose to evaluate which scheme is superior with a modem subsequently designed for the superior scheme. Finally, the use of multiple error correction schemes will be implemented in testing to evaluate their usability in the final system. This includes Golay, Hamming, maximum likelihood, Viterbi algorithm, multi-dimensional parity-check, repetition, Reed- Solomon, and low density parity-check.

PRODUCT DESCRIPTION

Two words, underwater communication. The goal of this project is to create an underwater acoustic modem for communication between two AUVs that will be implemented into the design of the 2017 RoboSub. With that said, there are certain hardware requirements that must be fulfilled. Our PCB cannot be bigger than the size of 3.5" x 3.5" x 0.5" with the option of using two, if needed. Our embedded systems portion of the project must receive data packets employing acoustic signals using low cost piezo transducers in order to transmit and receive signals. The modem communication itself will be half-duplex and must transmit and receive at a rate of 200 bps using FSK in addition to ASK modulation schemes.

Both submarines must be able to receive and transmit frequencies in the range of 50 kHz - 400 kHz. With that in mind, the attenuation factor of water will most likely force us to use frequencies on the lower side of the frequency range stated above. The hardware inside each of the submarines must be able to receive signals from the other submarine and filter out unwanted frequencies with the use of a bandpass filter. The bandpass filter will be centered on the carrier frequency of interest. This way if no signals are being transmitted by either submarine the noise present at other frequencies (submarine motors, pinger, and opponent submarine motors) should be filtered out. The bandpass filter that will be used is an active one. Active bandpass filtering will allow us to filter and amplify the signal in one step by using an op amp and the correct low pass and high pass filters at the input and output stage respectively. Once a signal is received we will use digital signal processing or an analog solution in order to find the frequency of the signal. The transmitting, receiving, and transducer blocks of the block diagram (Figure 1) are explained in more detail below. As an aside, we are using the dsPIC33EP64GS502 microcontroller by Microchip Technologies.

The transmitting block of our modem is labeled Tx and can be viewed in Figure 1. The transmitting block that includes a Class AB amplifier, a bandpass filter and a DDS or a DAC. The microcontroller will command the DDS or DAC to generate a sine wave at a certain frequency. Depending upon how complicated it is to program the DDS we may use a DAC. Once the sine wave is generated we will amplify the signal to a specified peak to peak voltage and amplify the current with a class AB amplifier in order to power the transducer to transmit signals up to 30m of maximum range of communication. The amount of peak-to-peak voltage we must amplify to will be very dependent upon the distance and the frequency of communication.

The receiving block of our modem is labeled Rx and can be viewed in Figure 1. We will receive all the signals that we transmit and convert them into bits and store the bits in the microcontroller. The receiving section of the block diagram includes an active bandpass filter and a variable amplifier. We will implement two different methods to convert an analog signal to a digital signal. The first method will use analog solution of two narrow active bandpass filters and an envelope detector. One of the narrow active bandpass filters will detect a 0 and the other will detect a 1. The DSP solution will utilize a wide active bandpass filter and then use IIR filtering techniques such as complex resonator to figure out what frequency was received. Depending upon what frequency is received will correlate with a 0 or a 1. The number of bits that will come in at each time are dependent on the number of tones that we will implement in our design. The more bandwidth capability we have the more specific bit patterns we can implement. For example, 8 tones allows us to send three bits at a time 000… 111 and 4 tones allows us to send 2 bits at a time 00… 11. The receiving block will be able to demodulate FSK and ASK signals. FSK will use a different carrier frequency to represent a pattern of bits. ASK will use a threshold voltage and if the threshold voltage is passed then we will know that a signal is present. Depending on the signal amplitude we will determine whether the signal is a 1 or a 0.

In addition to implementing directional transducers, we will also design and develop our own omnidirectional transducers for consideration within this project for use in the transducer block as shown in Figure 1. The motive behind this route serves us several design purposes. First, via this method we can directly compare test data between both types of transducers and move forward with more favorable findings. Second, we can start our design using the directional transducer as this would greatly simplify this segment of the project. After establishing the basic functionalities necessary to meet the constraints of this project we can then decide on expanding the complexities involved with the design of our transducers via introducing more functions. Lastly, it would be redundant and non-cost effective if we were to simply design and buy both transducers of the same kind. With all this in mind, we have opted on expanding the capabilities of our modem by using piezo ceramic cylindrical material with a frequency of 47 kHz. In order to make our transducers transmit and receive signals through a denser medium, we need to match the density of said medium, in this case, water. This is achieved by gradually increasing the gradient of the potting compound used in our design to match the density of water at approx. 1000 kg/m^3 using the density of air as a base. All of this is enclosed in a polyurethane layer for waterproofing with two leads protruding from the transducer for the appropriate connections. As a last comment, by using lower frequency omnidirectional transducers this will enable us to

broaden the spectrum in which signals can be sent uniformly in all directions. This in turn will allow us to avoid communication link failure between the daughter and the mother sub.

On the master PC/GUI side of the project, we will be using the python scripting language and the PIC microcontroller to handle outgoing and incoming packets. This includes error detection/correction as well as managing the data buffer on the microcontroller. We will use the tkinter library for managing the GUI components and Pyserial for serial communication to the PIC microcontroller. The GUI will help us manage and visualize the data that is being sent and received. For the purposes of testing we will be using a USB to RS232 converter to connect the PC to the PIC microcontroller. We will use a MAX232 chip to convert the RS232 signal into a TTL signal which will go directly into the Tx and Rx pins on the microcontroller. Inside the PIC we will use the built in USART registers and C code to manage the outgoing and incoming signals. The C code will handle how the buffer is stored and from there it will wait to get modulated onto a carrier frequency and sent out for transmission.

DESIGN

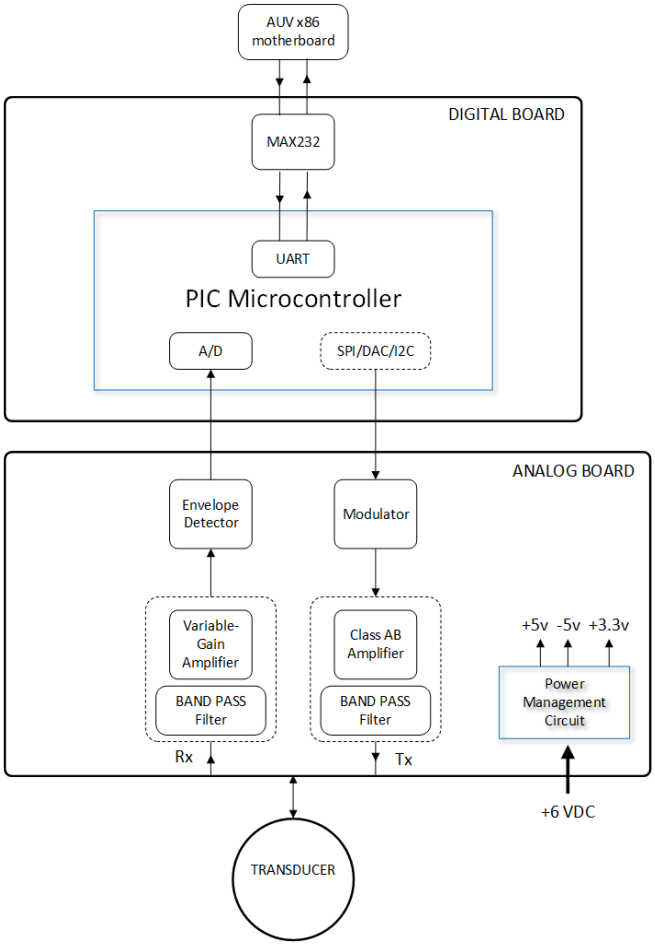


FIGURE 1: HARDWARE BLOCK DIAGRAM

For our initial stage of development, the hardware design methodology is shown in the block diagram above. As you can see, the acoustic modem system consists of the digital board, the analog board and the transducer. The digital board which is responsible for digital signal processing, is populated with the PIC microcontroller IC along with the UART-to-TTL logic conversion circuit utilizing the MAX232 chip. The analog board is function as the acoustic transceiver and is equipped with the power management circuit to regulate clean DC power supplies of +/- 5 volts and +3.3 volts to digital and analog elements on the two boards from a noisy 6 VDC input. The transducer is directly connected to the analog board. Finally, the two boards can be assembled and stacked to constitute the entire modem hardware which can be interconnected with external devices such as a personal computer or to the sub's motherboard via MAX232 UART protocol port.

For packet data sending, the digital board gets information from the PC or the AUV, construct a physical layer frame, package the data, and sends it to the analog board transmitter circuit. At the transmitter circuit, a modulated ASK or FSK signal is generated, amplified and filtered. The amplified signal gets fed to the transducer, transforming the electrical signal to acoustic waves which then propagate through an underwater channel. At the receiving end, when the acoustic waves reach the transducer, they are converted back to an electrical signal. The receiver circuit on the analog board will pass the electrical signal through a band-pass filter, variable amplifier and envelope detector. The packet data are reconstructed at the digital board and transferred the information over to the PC or the AUV's main computer.

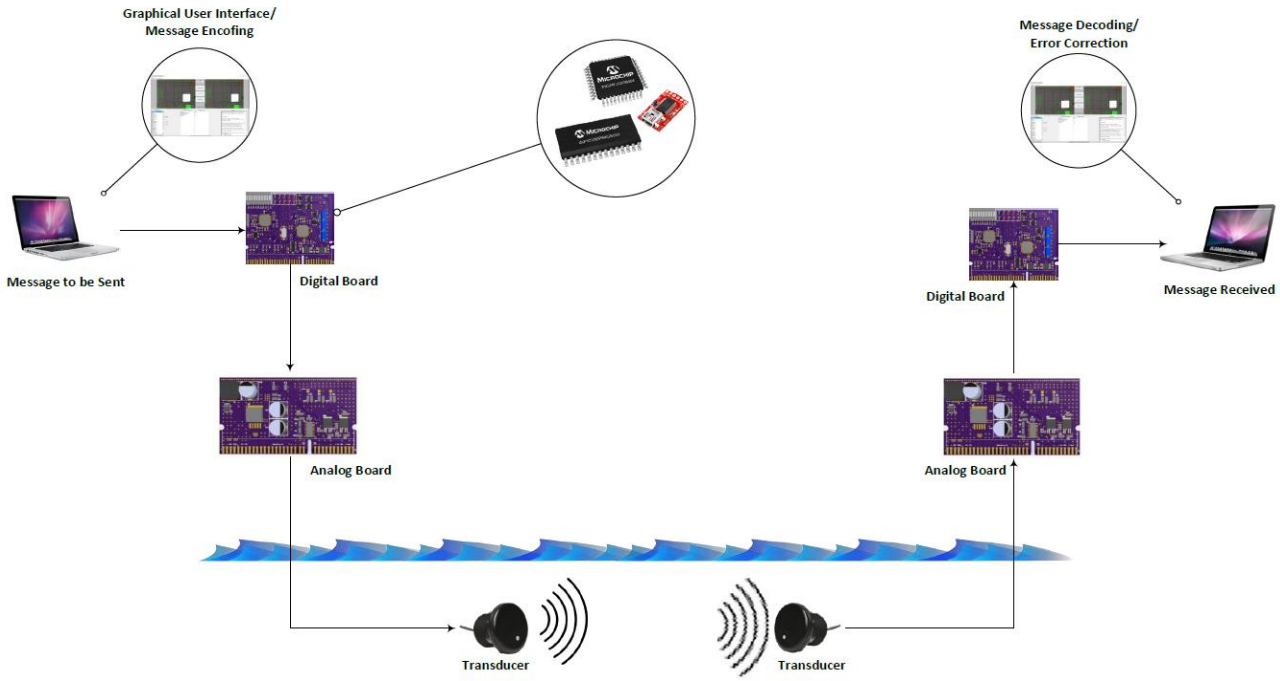


FIGURE 2: SYSTEM ARCHITECTURE

FIGURE 3: SYSTEM FLOW CHART

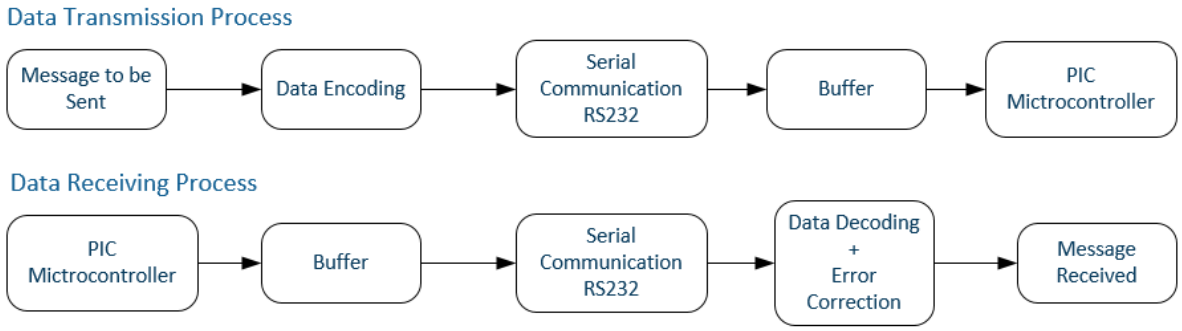


FIGURE 4: SOFTWARE BLOCK DIAGRAM

FIGURE 5: PACKAGING DEPICTIONS

PERFORMANCE REQUIREMENTS

Topic	Description
Power Consumption:	The power to receive and transmit will be different. Transmitting will take much more power than receiving as we have to send signals up to 30m. The average power consumption should be 1-40W transmitting (depending upon distance) and 1 W receiving. The average power consumption should be around 4.5W.

Transmission Distance:	The modem should be able to transmit packets up to 30m.
Message Error Percentage:	The amount of bits received in error should be less than 10% in order to allow the software team's error correction code to be effective.
Bit Transmission Rate:	The rate of bits that should be transmitted to our receiving modem should be 200 bps. The modem will be working in half-duplex.
Waterproofing:	Waterproof the transducers in order to test the hardware. Waterproof enclosure to protect electronic hardware.
Error Correction :	Use of multiple error correction code schemes. This includes Golay, Hamming, maximum likelihood, Viterbi algorithm, multi-dimensional parity-check, repetition, Reed-Solomon, and low density parity-check
System Communication:	Encode messages on a computer for sending and be able to receive and decode on another computer.
Microcontroller Communication:	Send and receive packets from microcontroller buffer.
GUI Integration:	Ability to test system by displaying sent and received messages. Also be able to select code scheme for sent and received data. Lastly, graph bit error rate before/after error correcting code for analysis.

HARDWARE TESTING PROCEDURES

ACTIVE BANDPASS FILTER

In order to test our active bandpass filters, our goal is to input frequencies using the function generator within the filter range, outside of it, and on the edge. Doing so will generate additional frequencies and then we can record the signal seen at the output using an oscilloscope. This will help us confirm what frequencies are being allowed to pass through the filters and which frequencies are being denied.

DIGITAL SIGNAL PROCESSING

The digital signal processing will be tested by sending a known analog signal to the DAC inside the microcontroller that will sample the analog signal. A digital signal will be outputted and then frequency will be determined by software. All this is done by using complex resonator IIR filters. This will allow us to tell what frequency is coming into the microcontroller. Once this is determined, the microcontroller will save the bit or bits that correspond with the frequency sent.

The microcontroller will save the bit or bits in memory and we will be able to look at this in memory to determine if the correct bit or bits was saved.

AMPLIFIERS (CLASS AB AND VOLTAGE)

The point of the amplifier is to increase the voltage or current. The class AB amplifier will be tested by measuring the average current running through a load. This will allow us to understand how much current is flowing through the resistor by putting an oscilloscope across the resistor and dividing the voltage on the oscilloscope by the resistor. The average current can also be recorded by using a DMM. For the voltage amplifier all we have to do is put in a test voltage and see how much gain we received by dividing output voltage by input voltage.

TRANSDUCERS

The transducers will be tested by providing a signal to one of the transducers at each side of a fish tank or pool. From there, we will then have an oscilloscope connected to the other transducer and look for the wave we sent from the other side. We will also connect the oscilloscope to the transducer sending the signal to make sure that the same signal is being received. We will also test the transducers at different distances to understand how that affects the signal that is being transmitted/received. This will help us understand how much we need to amplify a signal over various distances. The transducers should be tested for additional attenuation factor of the signal. This can done by putting the two transducers close together at a known distance (so water attenuation is known) and then we can determine the attenuation from the transmitter and receiver. Thus we will know the attenuation caused by the piezoelectric element.

INTEGRATED SYSTEM TESTING

Upon testing all individual parts for accuracy, the next step involves proceeding to assemble all the components of our design into a full design in incremental steps. The first task begins via combining the voltage amplifier and the AB class amplifier to run one of our transducers. The voltage amplifier in conjunction with the AB class amplifier will allow us to boost the analog signal for the purpose of sending out signals to further distances. Once we are able to amplify the signal we will then be able to test in a larger area such as the pool at the Aquaplex. Then we will try to send signals across the pool to determine if we are generating enough power to reach up to 30m. The other transducer should have an active bandpass filter connected to it in order for us to filter out other frequencies while amplifying the signal that was received. Once we are able to effectively send and receive signals then we will move onto integrating the digital part of the system.

To test the accuracy of sending information we must integrate the digital communication to both transducers. Keeping the transducers close we will attempt to send a packet of bits. In order to test the ability of our transducers, bandpass filters, and digital system we will attempt to send information across the fish tank. The amount of bits sent will be increased in increments of 10 bits to 50 bits. Once we are able to send packets of 50 bits in the fish tank then we will increase the distance at which we are sending the packets to 30m. The transducers will still be directional transducers and one will only receive and the other will only transmit.

Once we are able to get the transducers to work as a single transmitter and receiver then we will work on making both transducers able to send and receive. We will then implement the omnidirectional transducers that were built at this time. Thus we will have to have two modems built in order to test the ability of the modem hardware and the omnidirectional transducers. Once we are able to have both modems communicating with one another we will then implement a starting and closing pattern to complete a full bit packet. The receiving circuitry will not start saving bits until the starting pattern is received and won't stop storing until a closing pattern is received. This is the goal of the project to send and receive full packets. Once this is done with the correct speed and accuracy we will want to hopefully integrate the system with some kind of electronic system to show that our communication completes a task of some sort. If we are unable to get this far we will stick with sending AESCI code between the two modems or some other type of code that we may come up with.

SOFTWARE TESTING PROCEDURES

ERROR CORRECTION

To test that the program properly encodes and decodes binary data properly, a series of test vectors with varying corruption will be passed through. The results will let us know if the correction and detection are working properly when compared to what we expected.

SYSTEM COMMUNICATION

To test that our computer to computer communication is working, without our encoding and decoding, we will first send small amounts of data from one computer to the other. The next step involves us sending a simple small message. Following this we will send a file over the same channel by use of multiple packets. Lastly, to test the communication, we will be implementing the encoding and decoding while talking from computer to computer and redoing the test steps with the encoder and decoder active.

MICROCONTROLLER COMMUNICATION

To test the communication to the microcontroller, we will setup a microcontroller between the two computers and do computer to computer communication. We will then follow the same computer to computer communication testing. This will test if our data buffers on both the microcontroller and computer are working properly and using the flags to communicate full and empty.

GUI INTEGRATION

To test the GUI integration, we will use the GUI to control the computer to computer communication and redo the same tests for computer to computer communication. We will

then compile the data and compare the goodput, throughput, bit error rate, and comparing before and after error correction.

TESTING & VERIFICATION

BENCHMARKS

The standard we are setting for our project will be compared to two underwater acoustic modems made by Gangneung-Wonju National University and University of California, San Diego. It is important that we maintain a low power consumption of around 4.5 watts and be able to transmit signals up to 70 meters just as Tae-Hee Won and Sung-Joon Park from Gangneung-Wonju National University were able to accomplish. Furthermore, our underwater acoustic modem needs to manage transmitting 200 bits per second and have a 30 percent or lower bit error upon receiving signal in a pool environment. Lastly, it is also our focus to keep the cost under $600 just as was designed by the UCSD team. A recapitulation of our desired benchmarks is illustrated in the following table.

Desired Benchmarks as Compared with UCSD & National University	
Power Consumption	W (Gangneung-Wonju National University)
Transmission Distance	70m (Gangneung-Wonju National University)
Bit Transmission Rate	200 bps (UCSD)
Message Error Percentage	30% when implemented in the pool (UCSD)

PROJECT PLAN

INDIVIDUAL RESPONSIBILITIES

The design team architecture is divided into two sub teams of Hardware and Software. Software sub-team consists of 5 engineers mainly responsible for the digital aspect of the project.

Trevor Blazek and Blaire Chantrill are in charge of implementing and testing various error correction schemes and comparing their outcomes in order to come up with the best and most robust algorithm for this project.

Cavan Page is in charge of using serial communication to send and receive packets between the master PC via GUI and the microcontroller, which is an essential task in the design process.

Shideh Naderi, is responsible for overseeing project needs, allocating individual responsibilities and monitoring task progress as well as working closely with the Hardware team to ensure data is being buffered properly in the microcontroller during send and receive stage to avoid data overlap.

Jonah Mariano is in charge of creating a Graphical User Interface (GUI). The interface will be used for faster and easier testing procedure. Mariano will implement various tabs for serial communication and multiple error correcting scheme as well as plotting performance of each algorithm for visual purposes.

The hardware sub team is made up of 11 engineers. All of the engineers have been split up into sub groups to work on the different aspects of the hardware design.

Jeffery Miller, Davy Phdaky, Nate Swangpatton, and Bao Hoang are on the microcontroller team. This team will be responsible for generating the sine wave that we want to send out, storing the bits that will make up the packet, and converting analog to digital conversion. Most of the functions will be performed within the microcontroller dsPIC33EP64GS502. The team will be in charge of creating interrupts and timers to perform necessary tasks in an efficient manner.

Sebhat Yidelwo, John Salamat, and Emmanuel Palafox are on the transducer team. This team will be responsible for designing, testing, and building the transducers that will be used by our modems. Additionally, they are responsible for fabricating the fixtures for the transducers as well as procuring a tank for testing purposes.

Uros Obrovic, Mark Dimaano, Brady Anderson, and Leban Ali are on the hardware and DSP team. This team will be responsible for creating bandpass filters for ASK and FSK modulation schemes, amplification circuits, and digital signal processing. The group will compare the use of analog design to digital signal processing design.

MONITORING AND REPORTING

The monitoring is done by each Project Manager in a weekly manner. Tasks are allocated and reports are collected weekly. The tasks are then presented and discussed during each sub-team's meeting. Each team's focus is mainly on delegating tasks which are presentable by the meeting time to ensure that team members are on track with the schedule.

TASK PROGRESS AND SCHEDULE STATUS

As seen below, May 4th, 2016 is a strict deadline for the team to deliver and present the final product. The team has set multiple milestones to meet that are crucial to the success of this project. Tasks related to the software team are divided up between the members and are done in parallel such that design goals can be met faster and separate from one another. After each task is done, members will integrate their parts into the bigger system, which is called *Phase 1 Integration.*

It is expected to face obstacles while trying to integrate individual designs into the main system, there we created the *Enhancement Phase,* in which members will modify their designs to fit in the main system. We will then enter the *Final Integration Phase.* This phase must be in accordance with hardware team but it can also be an independent event in which it will be fully comprised of software capabilities. At this point all individual tasks must be done and system should be ready

for full on *Testing Phase*. This phase should rid the design of any bugs or issues. At the beginning of May the design is fully functional and ready to be presented.

Similar to the Software Team, the Hardware Team will work efficiently by dividing up the work between various teams. This began with the *Specification Definition Phase*, which concludes with this project proposal. In this phase we determined the overall system architecture and the specific requirements of each individual subsystem.

PROJECT MANAGEMENT

BUDGET

For the design of this project the maximum budget was established as $1500. The following charts depicted below illustrate both the relative allocation of funds for our project as well as exact monetary values. As you can see, most of the budget will be employed in the hardware side, more specifically, with the transducers alone totaling roughly one-third of the total expenditure. As for the software side, we incurred little to no expenses due the nature of the design goals.

Categories	Descriptions	Allocation
PCB Fabrication		$100
Components	Integrated Circuits Passive Transducers	$230 $200 $160
Prototyping	Integrated Circuits Breakouts Passive Transducers	$140 $170 $100 $340
Miscellaneous	Testing equipment Extra expenses	$60

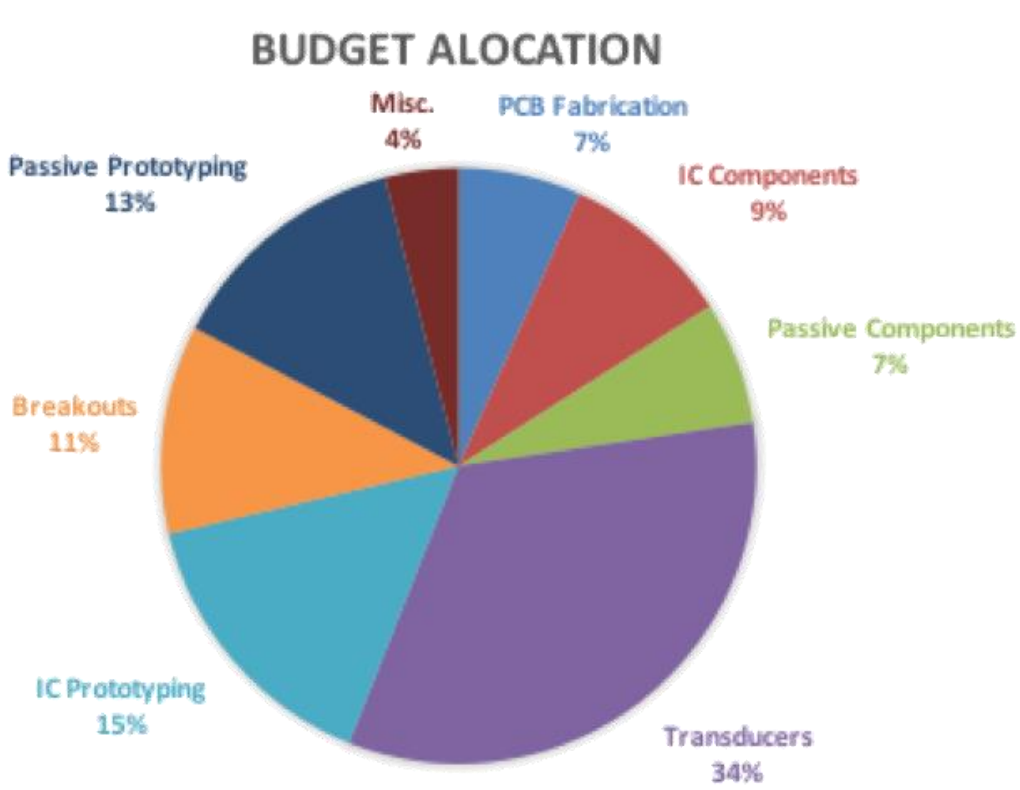


FIGURE 6: BUDGET CHART

MILESTONES

FIGURE 7: HARDWARE GANTT CHART

ID	Task Mode	Task Name	Duration	Start	Finish
1		Senior Design	77 days	Wed 1/20/1	Thu 5/5/16
2		Microcontroller cod	57 days	Sun 2/14/16	Sat 4/30/16
3		Testing and debugging phase	13 days	Tue 4/5/16	Thu 4/21/16
4		Proposal Draft	6 days	Fri 2/19/16	Fri 2/26/16
5		Proposal Complete	4 days	Mon 2/29/1	Thu 3/3/16
6		Obtain 45-50 gallon acrylic fish tank	5 days	Mon 2/22/16	Fri 2/26/16
7		Get assembled directional transducers for	5 days	Mon 2/22/16	Fri 2/26/16
8		Underwater directional transducer testing	6 days	Fri 3/4/16	Fri 3/11/16
9		Finialize design of underwater acoustic	3 days	Fri 3/11/16	Tue 3/15/16
10		Build omnidirectional	4 days	Wed 3/16/16	Mon 3/21/16
11		Acquire live test data via experimentation	1 day	Tue 3/22/16	Tue 3/22/16
12		Build session: proto	2 days	Fri 4/1/16	Mon 4/4/16
13		PCB Assembly	1 day	Fri 4/8/16	Fri 4/8/16
14		PCB troubleshooting/ver	5 days	Tue 4/12/16	Mon 4/18/16
15		DSP Receiving hardv	0 days	Thu 3/31/16	Thu 3/31/16
16		Analog Receiving hardware	0 days	Thu 3/31/16	Thu 3/31/16
17		Generate sine wave with DAC and DDS	0 days	Sun 3/20/16	Sun 3/20/16
18		PWM on dsPIC microcontroller	0 days	Sun 3/20/16	Sun 3/20/16
19		Analog to Digital converter on dsPIC microcontroller	0 days	Sun 3/20/16	Sun 3/20/16
20		Purchase high frequency piezoelectric ceramic materials for production and	0 days	Fri 3/11/16	Fri 3/11/16
21		Determine power amplifier class for intergrated circuit design	0 days	Sun 3/13/16	Sun 3/13/16
22		Schematic for prototype board	0 days	Fri 4/1/16	Fri 4/1/16
23		Schematic for final design	0 days	Fri 4/8/16	Fri 4/8/16
24		Integration with software	0 days	Mon 4/11/16	Mon 4/11/16

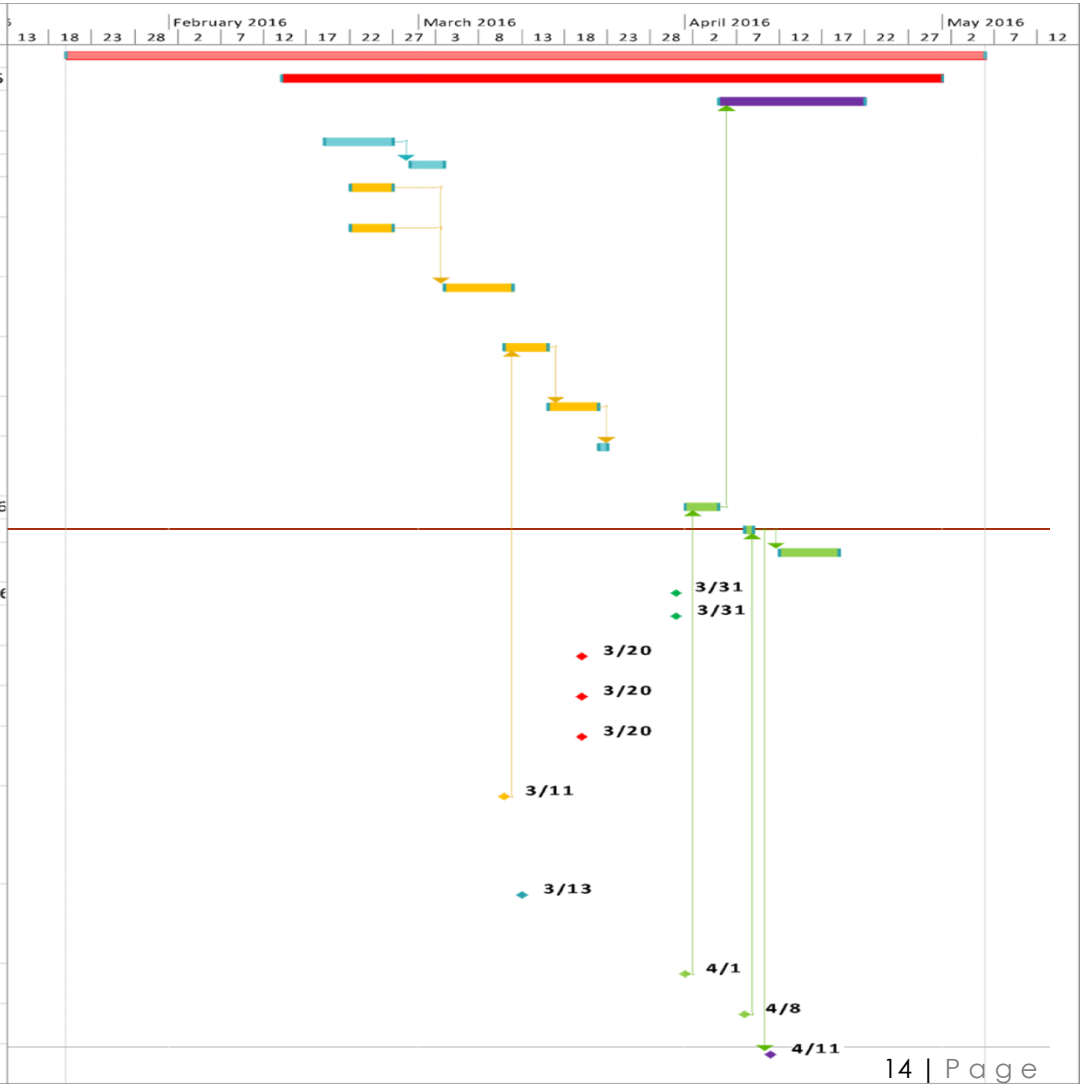
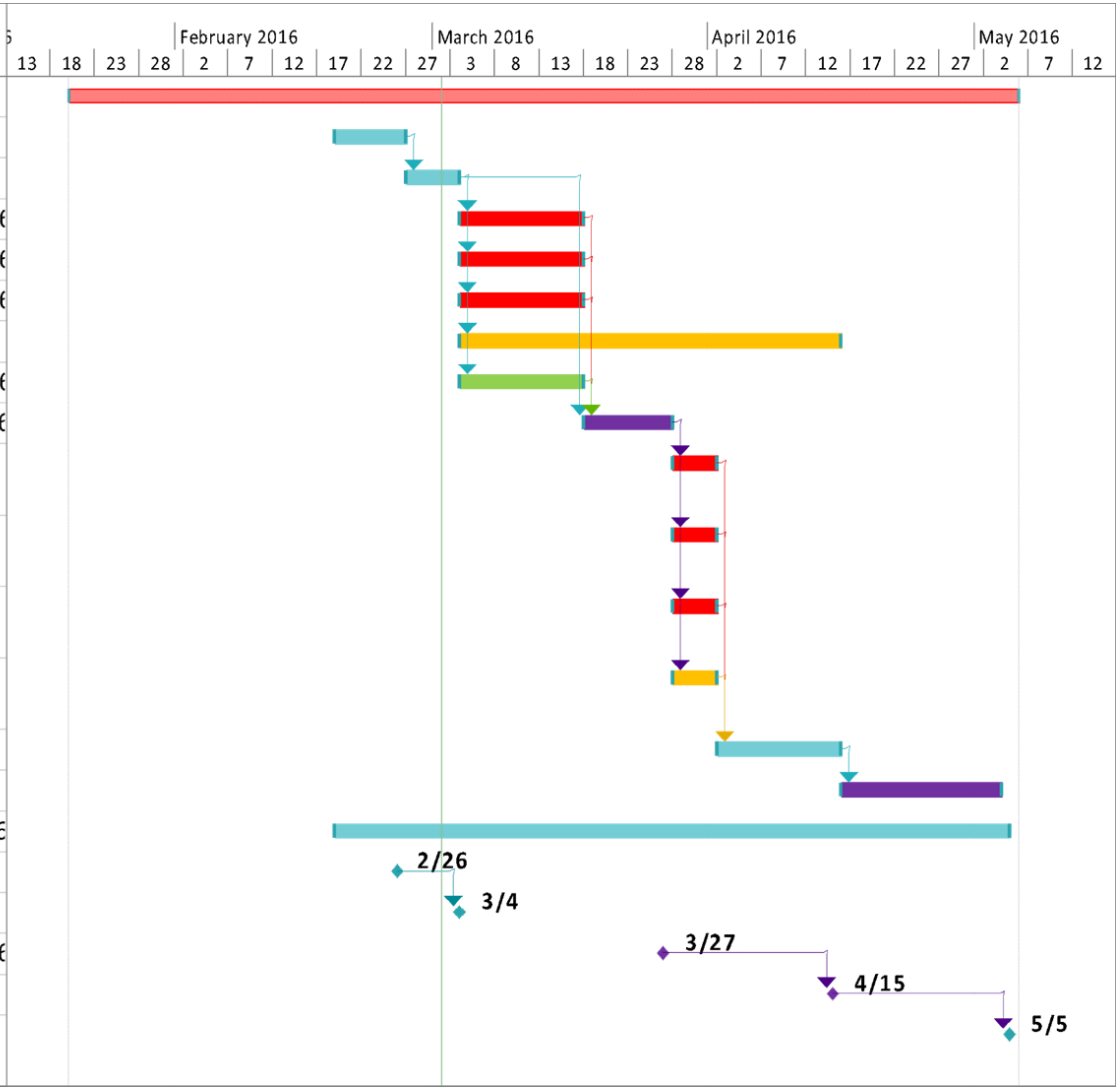


FIGURE 8: SOFTWARE GANTT CHART

ID	Task Mode	Task Name	Duration	Start	Finish
1		Senior Design	77 days	Wed 1/20/1	Thu 5/5/16
2		Proposal Draft	6 days	Fri 2/19/16	Fri 2/26/16
3		Proposal Complete	5 days	Sat 2/27/16	Thu 3/3/16
4		Data Encoding	10 days	Fri 3/4/16	Thu 3/17/16
5		Data Decoding	10 days	Fri 3/4/16	Thu 3/17/16
6		Error Correction	10 days	Fri 3/4/16	Thu 3/17/16
7		GUI/COMM design	31 days	Fri 3/4/16	Fri 4/15/16
8		PIC Programming	10 days	Fri 3/4/16	Thu 3/17/16
9		Phase 1 Full Integra	7 days	Fri 3/18/16	Sun 3/27/16
10		Data Encoding Enhancement	5 days	Mon 3/28/16	Fri 4/1/16
11		Data Decoding Enhancement	5 days	Mon 3/28/16	Fri 4/1/16
12		Error Correction Enhancement	5 days	Mon 3/28/16	Fri 4/1/16
13		GUI/COMM design Enhancement	5 days	Mon 3/28/16	Fri 4/1/16
14		Final Phase Integrat	11 days	Sat 4/2/16	Fri 4/15/16
15		Test Phase	13 days	Sat 4/16/16	Tue 5/3/16
16		Project Managemer	54 days	Fri 2/19/16	Wed 5/4/16
17		Proposal Draft	0 days	Fri 2/26/16	Fri 2/26/16
18		Proposal Complete	0 days	Fri 3/4/16	Fri 3/4/16
19		Phase 1 Full Integra	0 days	Sun 3/27/16	Sun 3/27/16
20		Final Phase Integrat	0 days	Fri 4/15/16	Fri 4/15/16
21		Finished Project/Demo	0 days	Thu 5/5/16	Thu 5/5/16



REFERENCES

[1] Tae-Hee Won and Sung-Joon Park. "Design and Implementation of an Omni-Directional Underwater Acoustic Micro-Modem Based on a Low-Power Micro-Controller Unit." Sensors 12 (2012): 2309 - 2323. Print.

[2] B. Benson, Y. Li, R. Kastner, B.Faunce, K. Domond, D. Kimball, & C. Schurgers. "Design of a Low-Cost, Underwater Acoustic Modem for Short-Range Sensor Networks." Ph.D. Thesis, University of California, San Diego, CA, USA, 2010. Print.

MECHATRONICS

SAN DIEGO STATE UNIVERSITY

Presents

Project Poseidon

An Underwater Acoustic Modem

Jump in and dive with us as we introduce to you a Marvel of Engineering!

SAVE THE DATE!

Wednesday, May 04, 2016

Montezuma Hall

Visit us @ www.poseidon.sdsu.edu