Ultrasonic Object Detector

DESIGN DOCUMENT

Sdmay25-36
Client/Advisor:
Professor Jiming Song
Team Members/Roles:
Nate Clarke: ***Project Software Designer***
Brock Dykhuis: ***Circuit Analysis***
Nicholas Jacobs: ***Electronics***
Jonathon Madden: ***UI Design and Software Testing***
sdmay25-36@iastate.edu
https://sdmay25-36.sd.ece.iastate.edu/

Revised: 12/7/2024
Version: 1

Executive Summary

Our goal for this project is to construct an ultrasonic radar that improves on the past two implementations of this project. Ultrasonic radars are important as they have many uses, some of these include military/navy applications, water level detections, surgical applications, and car sensors. Our major improvement is a better display, this will require both more accurate measurements and better phase delay calculations. A second improvement is to communicate between the radar and the display wirelessly over Wi-Fi. To do this, we are using a Raspberry Pi that will host a dedicated web server that will hold the data, this data will be taken by the display and plotted. This device can be a great learning tool for ourselves, and potential purchasers if it was to be put on the market.

So far, we have set up the radar with one transmitter and receiver, this allows for object detection directly in front of our radar. We have made substantial improvements in how the display will look by creating a graph that, over time, dims and eventually removes precious scanned objects that are no longer detectable. We have set up communication between the microcontroller and Raspberry Pi, as well as setting up the web server that will be used. Currently, we have a sample radar display designed in a processing sketch (.pde file).

Our design so far is meeting our client's requirements, as Professor Song wants our design to use a more detailed display. We are also improving the connection between the radar and the display, as a cord will not be needed, instead, all that is needed is for the display to be on the same Wi-Fi network.

Some potential next steps will be to expand our circuit design to include ten transmitters and additional receivers, work on phase delay calculations, and code the Raspberry Pi to upload the scanned data directly to the web server. The display can be improved by experimenting with C++ using OpenGL or SFML to reduce delays. Additionally, panning, zooming, cursor coordinates, and color coding can potentially be added.

Learning Summary

Development Standards & Practices Used

Standard Practices Used

- Circuit Design and Analysis
- Circuit Simulation and Testing
- Iterative software development
- Software testing
- Embedded systems programming
- Networking

Engineering Standards

- IEEE 1471-2000
- IEEE 754-2008
- IEEE 802.11
- IPC 2221

Summary of Requirements

- Construct an ultrasonic radar using transducers as both transmitters and receivers.
- Utilize software to manage phase delay (activating different transmitters at certain times in accordance with the necessary number of wave cycles of difference).
- Utilize software to send short pulses.
- Calculate the distance between the object and the radar using time delay.
- Generate an image using the Euclidean distance and determine the angle of read data as points.
- Display readings in a readable and intuitive display.

Applicable Courses from Iowa State University Curriculum

- ComS 2520 - Linux Operating System Essentials
- ComS 3520 - Introduction to Operating Systems (for concurrency)
- ComS 3270 - Advanced Programming Techniques (C & C++ language)
- CprE 2880 - Embedded Systems
- EE 2300 - Electronic Circuits and Systems
- EE 2240 - Signals and Systems I

- EE 3240 - Signals and Systems II
- NS 3330 - Naval Ship Systems II (Weapons)
- SE 1850 - Problem Solving in Software Engineering
- SE 3170 - Introduction to Software Testing
- SE 3190 - Construction of User Interfaces

New Skills/Knowledge acquired that was not taught in courses

- Communication between the microcontroller and Raspberry Pi using a MQTT server.

- Coding an ESP32 MCU using Arduino.

- The basics of a phased array and how it can be implemented with phase delay.

Table of Contents

List of figures/tables/symbols/definitions

Definitions:

- Phased Array: An array of transmitters that can change a single shape and direction without moving.
- Phase Delay: A time delay for a traveling wave.

1. Introduction

1.1. PROBLEM STATEMENT

The problem being addressed is designing a radar device for object detection using ultrasonic sound waves. The radar will be able to determine an object's distance and direction from the radar. This is done through the use of ultrasonic transmitters lined up side-by-side a fixed distance away from each other, this is known as a phased array, these transmitters send an ultrasonic wave that is then reflected back to the ultrasonic receiver. Distance from the radar and object can be found by using the time delay, from when the signal is sent and then received. The direction of the object can be found using phase delay. The system will provide a learning opportunity for ultrasonic object detection, which can further be used in various applications (collision detection, water level determination, airspace scanning, ect.).

1.2. INTENDED USERS

Client (Professor Song):

Our client is an Electrical Engineering Professor at Iowa State University and has run projects in the past dedicated to designing ultrasonic radars. Our client is likely often busy planning for his classes, but intends to guide us to develop a more effective radar implementation than past projects. Our client needs our team to develop an ultrasonic radar system that can detect object distances and directions, and the implementation must improve upon the accuracy of past implementations. Our client intends to provide a learning experience to our team by giving us experience with embedded systems and determining areas of improvement by viewing past designs.

Theoretical Purchaser:

While there are no plans to release this design after we complete it, there are many commercial uses for a sort range object detector, some of these include: checking blind spots on cars, such as making a beeping noise if there is an object behind you while in reverse, or to check if there is a car to the side of you while driving, this is useful for changing lanes. Another use case could be a security mechanism, this could detect if someone walks through a door or if an object was removed from a pedestal.

Designers (group members):

This user group consists of the student engineers working on the project, each starting with different levels of knowledge in software and hardware integration. One key need is for the project to offer learning opportunities in both the software and hardware aspects. Additionally, the project will present technical challenges as the team integrates both of the aforementioned sectors. The benefits of meeting these needs include gaining valuable hands-on experience, with the skills learned being applicable to future software and hardware development careers.

2. Requirements, Constraints, And Standards

2.1 REQUIREMENTS & CONSTRAINTS

Functional Requirements

- Radar must be able to detect objects that are up to 1 meter away (**constraint**)
- Radar must display sweep/radar image for detected objects
- Use a phased array with phase delay to determine object directions
- Use pulse time delay to determine object distance (the delay between sending and receiving a pulse)
- Use pulses with an ultrasonic frequency of 40 KHZ (**constraint**)

Physical Requirements

- Use 10 ultrasonic transducer transmitters in linear layout (**constraint**)
- Use 1 ultrasonic transducer receiver (**constraint**)
- Transducers must be approximately 10 mm in diameter (**constraint**)

Experimental Requirements

- Provide an Error Log
- Clear component and software testing documentation.

Environmental Requirements

- The sound produced by the radar system must be 40 kHz, ensuring that it is outside the range of human hearing. It should only be heard by a small number of animals (e.g. a bat)

Resource Requirements

- Raspberry Pi 3 used as server for wireless data transmission
- External supply is needed to power the ultrasonic radar system
- The system with use MA40S4S/R (10S, 1R

Aesthetical Requirements

- Display must be clear, readable, and interactive (allowing for zooming and panning)
- Object data points and groupings must be clear and readable
- The radar must precise down to the at least 1 cm, with 1 mm of precision being optimal (Constraint)

2.2 ENGINEERING STANDARDS

Engineering standards help ensure products have a quality status to strive for, ensuring safety, convenience, reliability, and other various factors. Most products around us adhere to these standards, significantly improving their usability.

IEEE 1471-2000

- This standard deals with creating, analyzing, and maintaining software architecture.
- This standard applies since we will have to deal with the maintenance and display of software with the ever changing device specifications.

- This standard will have to be addressed through thorough documentation.

IEEE 754-2008

- This standard deals with storing floating point numbers in computer systems.
- Indirectly applies since we are using floating point operations, which the computer should store, but it may apply directly to how the hardware depicts its data.

IEEE 802.11

- This standard covers transmitting data to a computer wirelessly.
- This standard applies because our project will be using a Raspberry Pi to transmit data wirelessly.

IPC 2221

- This standard establishes design requirements for PCBs.
- This standard applies to the construction of the radar hardware, since it will be using a PCB board as its foundation.

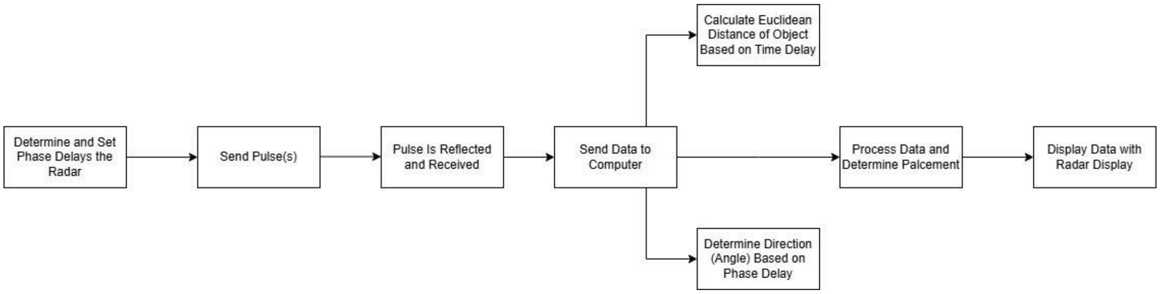
3 Project Plan

3.1 Project Management/Tracking Procedures

For this project, we are using the waterfall approach. The reason this works for our project is that in order for some of our tasks to be completed, we will need previous tasks completed. Such as, initially we have to plan on what components to order. After we determine what parts to order, we need to draft a design. Then, we will need to test this design and start software development for the display. We will then need to build and test our design with the display. Finally, we will need to make any last improvements, such as improving accuracy. In order to keep track of team progress throughout the project, Git will be used to create issues requiring completion, and weekly meetings will be used to check progress.

3.2 Task Decomposition

To create our project, we will have to split it into a series of tasks. The first major step is to set the array to a particular phase delay to determine the direction. After this, the pulses are sent and received, and data is sent to the server. The data is then sent to the computer to process. The data is processed to determine distance and direction. Processed data is displayed on radar sweep.



3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Task 1: Phase setting and initialize array

- Phase delay is calibrated accurately within 5 degrees

Task 2: Pulse transmission and reception

- Pulse is accurately transmitted and received over the range of 1 meter.
- The microcontroller receives accurate data.

Task 3: Transmit Data to the server

- The connection between the server and Raspberry Pi is stable
- Data is transferred in under 50 ms

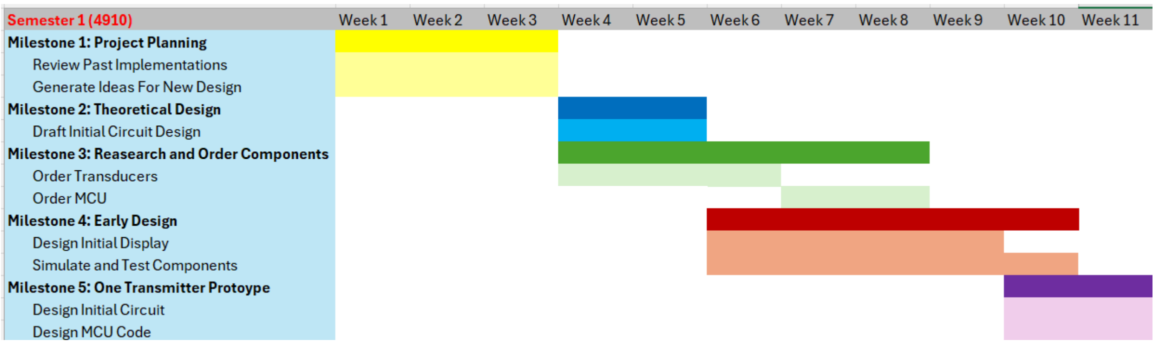
Task 4: Data Processing

- Processed data shows the object's distance and location
- The data displayed is at least 90% accurate in distance and location

Task 5: Radar Display

- Display updates in real-time to show the location of the object with 90% accuracy

3.4 PROJECT TIMELINE/SCHEDULE



Semester 1 (4910)	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
Milestone 1: Project Planning	🟨	🟨	🟨								
Review Past Implementations	🟨	🟨	🟨								
Generate Ideas For New Design	🟨	🟨	🟨								
Milestone 2: Theoretical Design				🟦	🟦						
Draft Initial Circuit Design				🟦	🟦						
Milestone 3: Reasearch and Order Components				🟩	🟩	🟩	🟩	🟩			
Order Transducers				🟩	🟩	🟩					
Order MCU							🟩	🟩			
Milestone 4: Early Design						🟥	🟥	🟥	🟥	🟥	
Design Initial Display						🟧	🟧	🟧	🟧		
Simulate and Test Components						🟧	🟧	🟧	🟧	🟧	
Milestone 5: One Transmitter Protoype										🟪	🟪
Design Initial Circuit										🟪	🟪
Design MCU Code										🟪	🟪

Project Planning

- Review past implementations to gain a better understanding.
- Generate ideas for new designs.

Theoretical Design

- Initial planning to determine the basic layout, and needed components for this project.

Research and Order Components

- Determining which parts to order based on client specifications and additional research.

Early Design

- Early Design stages for the radar display, simulating with pretend data.
- Simulating expected components and testing them once they have been ordered.

One Transmitter Prototype

- Building the theoretical design on a breadboard allows us to determine if our theoretical design works, we will use 1 transmitter in this step.
- Coding MCU for detection with one transmitter

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

1. **Signal Amplification and Processing**
   a. Risks:
      i. Amplification may not reach target gain (probability: 0.6)
      ii. Processing may introduce signal delays, affecting timing accuracy (probability: 0.4)
   b. Mitigation:
      i. Replace the amplification module with a higher-gain alternative if required
      ii. Consider a signal processing library or pre-built DSP to achieve consistent timing
2. **Transducer Testing and Timing**
   a. Risks:
      i. Transducer may vary across environmental conditions (probability: 0.5)
      ii. Precision timing adjustments may not achieve the required/desired detection range (probability: 0.6)
   b. Mitigation:
      i. If precision remains unreliable, evaluate off-the-shelf transducers with stable timing performance or utilize a different timing control algorithm
      ii. Implement additional temperature compensation to counter environmental variability
3. **System Integration**
   a. Risks:
      i. Microcontroller data throughout may not sync correctly with radar pulses (probability 0.5)
   b. Mitigation:
      i. Test alternate microcontroller models with faster processing speeds if needed or switch to an FPGA-based solution for high-speed integration

We'll evaluate the potential for off-the-shelf components or more advanced processing algorithms for high-risk tasks, ensuring an adaptable approach to mitigate risks with high probabilities.

3.6 PERSONNEL EFFORT REQUIREMENTS

Task	**Person-Hours**	**Explanation**
Determine and Order Transducers	6 hours	Need to find transmitters for the 40KHz range with a size of around 10mm. This is quite difficult to find within our price range.
Acquire Raspberry Pi 3b for Wireless Data Transfer	2 hours	Rented from ETG.
Determine the Power Supply Requirements and Order It	6 hours	Need to determine the required voltage to ensure the radar functions as needed.
Initial Circuit Layout Analysis and Transducer Placement	72 hours	Need to Place the transducers in a way to maximize signal to

		noise ratio.
40 KHz Square Wave	4 hours	Creation of a 40 kHz oscillator and amplifier that will take a few hours. It involves setting up a stable signal source and matching the amplifier for sufficient output power.
Determine Phase Delay/Shifts	72 hours	Determine phase delay and shifts needed to scan in a particular direction.
Initial Data Processing	168 hours	This will be one of the major tasks in this project; it will present complexity in determining time delay and phase delay and using those values to determine an object's distance angle incidence.
Object Detection Troubleshooting	72 hours	Determining the accuracy of object detection may prove difficult, and require many small adjustments.
Image Display	72 hours	Develop a simple radar display to display detected objects, which must allow for zooming for easy reading.
Redesign Circuit with Additional Receivers.	72 hours	The radar will be upgraded with additional receivers to improve image quality.
Updated Data Processing	336 hours	Improve data processing to account for additional receivers. This should prove to be complex.
Improved Display	144 hours	Experiment with C++ using OpenGL or SFML to reduce delays.

3.7 OTHER RESOURCE REQUIREMENTS

Electrical Resource Requirements

- Arduino simulator, used for theoretical simulations
- Electrical components
    - wires, resistors, inductors, capacitors, amplifiers, transducers (MA40S4S/R)
- Raspberry Pi 3B

- Microcontroller ESP32-S3-DevKitC-1-N8R8

Software Resource Requirements
- Web Server, this is hosted using Apache on the Raspberry Pi
- Processing IDE, this is used for the display

4 Design

4.1 DESIGN CONTEXT

4.1.1 Broader Context

Area	**Description**	**Examples**
Public health, safety, and welfare	Our project presents minor safety risks associated with working with a low powered electrical device, and potential concerns involved with human hearing. Otherwise it provides a great learning tool for a variety of users.	The radar has can produce a high sound intensity (120 db) at max voltage which can cause hearing loss without protection The radar is a great tool to learn about phased arrays, and how to use and program a MCU.
Global, cultural, and social	The project should not present any form of intentional bias, potential improvements can be made to benefit users with vision impairment. The overall visual should be easy to read and interpret.	Zooming, panning, cursor labeling, and large text displaying the most recent reading should improve readability and interactiveness. Color coding could potentially need to consider colorblindness.
Environmental	Our project may have a slight effect on a select number of animals with specialized hearing, but it will be used inside and should never cause any disruptions. The project will use electrical components which may require special disposal services.	The project components can be recycled for future implementations decreasing the need for non-recyclable materials. Transducers may be prone to burning out, so prevention steps should be taken to limit this. 40 kHz can be heard by a select number of animals including bats. The device should not be used outside, so this should not present issues.
Economic	Our project is relatively affordable (if it were to be mass produced it would be cheaper per unit), and it can be easily reused for future implementations.	The transducers are very affordable between 5-6 dollars each. The MCU was only 15 dollars as well. The computer would hypothetically be already available (thus not an additional cost), the same can

		potentially be said for the Raspberry Pi. The transducers and MCU can easily be reused for future implements.

4.1.2 Prior Work/Solutions

Past Implementations:

sdmay-23-22 Ultrasonic Radar [1] :

Pros	Cons
The display is easy to read	Only one point is displayed at a time
The code is simple, readable, and well documented	The transmitters used are relatively large, 16mm (reducing field of view)
Well-built with a protective shell and PCB implementation	Only one receiver is used
	Can't distinguish two objects

sdmay24-24 Sonar: Ultrasonic Radar [2] :

Pros	Cons
The radar does a vast amount of sampling	The display is not very readable
	The transmitters used are relatively large, 16mm (reducing field of view)
	Only one receiver is used
	Can't distinguish two objects, and the heat map used is muddy

Planned Improvements:

In order to improve on past projects, we will have to improve the readability of the radar display and potentially add helpful functionalities such as zooming, panning, and marking cursor position. Color coding could potentially be used to distinguish between objects. Opacity could be used to show the age of a data point reading or potentially the confidence level/clarity.

In terms of hardware, a major improvement we hope to complete is using additional receivers to improve reading quality. The current plan is to use 3 receivers placed side-by-side below the transmitters. We have also improved the transducer quality for this project by utilizing the MA40S4S/R transducers. The MA40S4S is the transmitter (in this case, it is referred to as a sender), and it has a diameter of only 10mm.

Another improvement we hope to make is to have the radar wirelessly transmit data to the computer running the display. This will be performed with a Raspberry Pi 3b hosting a server to hold the data for radar readings. If this radar were to be distributed commercially, this would reduce liability associated with a direction connection to a computer.

Research:

Design Basics:

- Sound waves travel at 343.7 m/s in air with a temperature of 20 Celsius and 30% relative humidity [3]. For simplicity, 343 m/s will be used in this project, and it should provide sufficient accuracy.
- Impedance = velocity * density [3].
- Transducers convert signals from an electrical form to an acoustic signal [3].
- The general design is as follows:

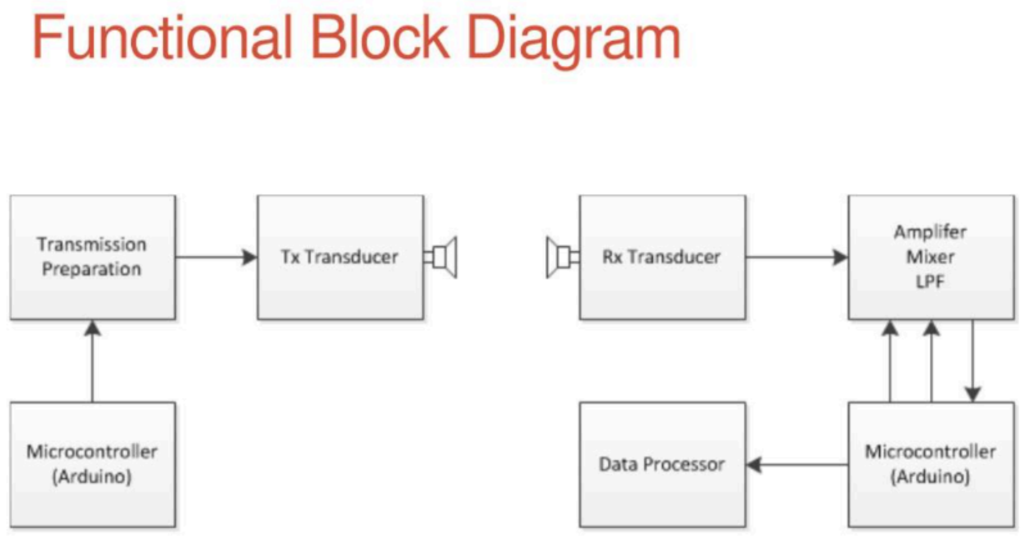
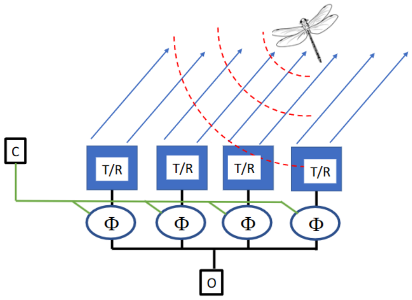


Figure 1: Functional Block Diagram from [3]

- Combining echo signals improves signal-to-noise ratio, improving accuracy [3].

- The basic design of a phased array is as follows:



C is the control circuit to set different phase (Φ) for each transducer (T/R)

O is an oscillator that produces a pulse at an ultrasonic frequency

Figure 2: Phase Array Implementation from [3].

Phased Array Basics:

- A beam has a main lobe and directivity pattern, and a main lobe should be narrow to improve directivity [4].
- Can control steering angle, beam width, and focal depth [4].
- Beams are created with the help of constructive and destructive interference [4].
- Pulses are usually short waves (2-3 cycles) [4].

Time Delay in Phased Arrays

- With a phased array with a half wavelength distance between components, the time delay can be modeled by $\sin(\theta)/2f$, where f is the frequency provided by a transducer[5].
- A larger steering angle will result in a larger time delay [5].
- Aperture is the distance along the width of the array where elements are activated for beam transmission [5].
- Depth of field is modeled by $(7.1\,\lambda F^2)/a$ where F is focal length, a is aperture, and λ is frequency [5].
- Near-field-of-depth can be modeled by $a^2/4.1\,\lambda$ [5].

4.1.3 Technical Complexity

- Phase delay must be calculated and applied to scan in a particular direction/angle. With a high number of transducers, this can become very complex.

- The distance will need to be determined with time delay. This will provide complexity by needing to track the time a pulse is sent and received. This should be a very short time period.

- The calculations needed to determine the distance angle should be very quick to prevent delay in detection processing.

- The ESP32-S3-DevKitC-1-N8R8 MCU should provide ample processing power but will need to be properly programmed.

- Transmitters should be placed in such a way that the signal-to-noise ratio is as high as possible.
- Filtering of pulse inputs is necessary to reduce noise and interference.
- The data will need to be sent over wifi to a Raspberry Pi, which should provide a stronger connection to the computer for final processing. It will also provide complexity in connecting to the Iowa State University's wireless network as we do not manage it.
- Upon receiving the data the display program will need to receive data and parse it continuously.
- The display will need to continuously refresh to prevent a high number of data points and make the current points more relevant based on recency. (Potentially, points could fade based on the time since being detected)
- The display must clearly depict the difference between two close objects and one singular larger object.
- Object groupings should be clear, and the display should be interactive allowing for zooming for improved readability.

4.2 Design Exploration

4.2.1 Design Decisions

One key decision that was made was to use a Raspberry Pi for its cost-effectiveness and functionality. It was also chosen because of its ability to communicate with the microcontroller and processing IDE. Because it is able to process real-time data, our radar display will be more precise. We also wanted a wireless connection for this project, and the Raspberry Pi will allow us to do that.

Deciding which transducer to use was another key decision. Choosing the transducer was one of the most important tasks for this project because they are the parts that are required for sending and receiving ultrasonic pulses. We had to consider the size and range of the transducer as we were looking for transducers that were around 10 mm in diameter and a range of 1 meter. The MA40S4S/R transducers were chosen because they had the criteria requiring 10 mm diameter transducers with a range of at least 1 meter.

Another important key decision that was made for this project was determining which Microcontroller Unit to use for the radar's data processing. The past implementations used ESP32 MCUs, so this was the general family of products we searched for. The most recent implementation used the ESP32 D1, which is fairly outdated, and they expressed that more processing power could be a good place for improvement. We began looking into the C6 and S3 models and found that the C6 models emphasize power use reduction rather than processing power like the S3. We ended up further exploring the S3 models and found a model with 8 MB of flash and 8 MB of PSRAM. The PSRAM should provide a buffer for calculations, which should prove helpful in our project and should allow for a reduction in delays in displaying data.

4.2.2 Ideation

When choosing our transducers, we had many possibilities. We chose the MA40S4/R, with some of our other considerations included below.

- TCT40-16R/T
    - This transducer is 16mm in diameter; we chose not to pick it because of its size.
- CUSA-T60-150-2400-TH
    - This transducer is 14mm in diameter and costs $2.95; despite its cheap cost, we wanted a smaller transducer.
- MCUSD14A40S09RS
    - This transducer is 14mm in diameter; while it is smaller than the previous option, we still wanted a transducer that was smaller
- 400SR16
    - This transducer has a diameter of 13mm, and we chose not to pick this because of its high cost, being around $8.5
- Eu10POF40H07T/R
    - This transducer has a diameter of 10mm, which is the same as the one we chose, but costs $9.95 each, which is about double the cost of the transducer we chose.

We ultimately picked the MA40S4S/R because of its relatively small diameter of 10mm and cheap cost of $49.30 for 10 transmitters, each costing $4.93 and the receiver costing $5.49.

4.2.3 Decision-Making and Trade-Off

Demonstrate the process you used to identify the pros and cons of trade-offs between each of your ideated options. You may wish to include a weighted decision matrix or other relevant tool. Describe the option you chose and why you chose it.

Transducer Criteria – Chosen: MA40S4S/R

- The transducers should be as small as possible (with 10 mm diameter being the expected size).
- The transducers should have a range of at least one meter.
- The transducers should be the cheapest option, fulfilling all other criteria.
- The transducers should have a 40KHZ signal.
- The transducers should have a wide viewing range.
- The transducers should not have excessive decibel production at far ranges (outside of the detection area of one meter)

Microcontroller Unit Criteria – Chosen: ESP32-S3-DevKitC-1-N8R8

- Provides ample processing power (significantly more powerful than the D1 of past implementations).
- It must have a small form factor to easily fit on the PCB board for the final design.
- It must have a clock rate of at least 240MHZ.
- It must have at least 8MB of flash memory.

Wireless Data Transmission – Chosen Raspberry Pi 3b

1. The device must be within a reasonable price range (Less than $50).

2. The device must allow for a 2.4 GHZ wireless band.
3. The device must allow for easy connection of the computer for display processing.

4.3 PROPOSED DESIGN

4.3.1 Overview

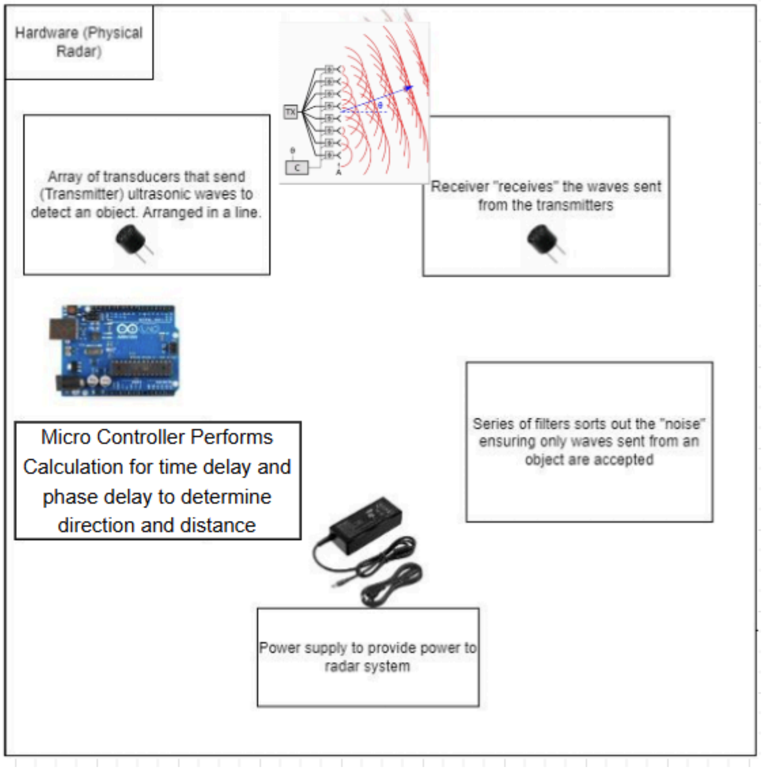
The ultrasonic radar project is a detection system that uses sound waves to pinpoint an object's location and distance from the sensor. It works by sending out ultrasonic waves and then listening for echoes that bounce off objects. We can figure out an object's location by recording the time it takes for the echo to return.

Key Components

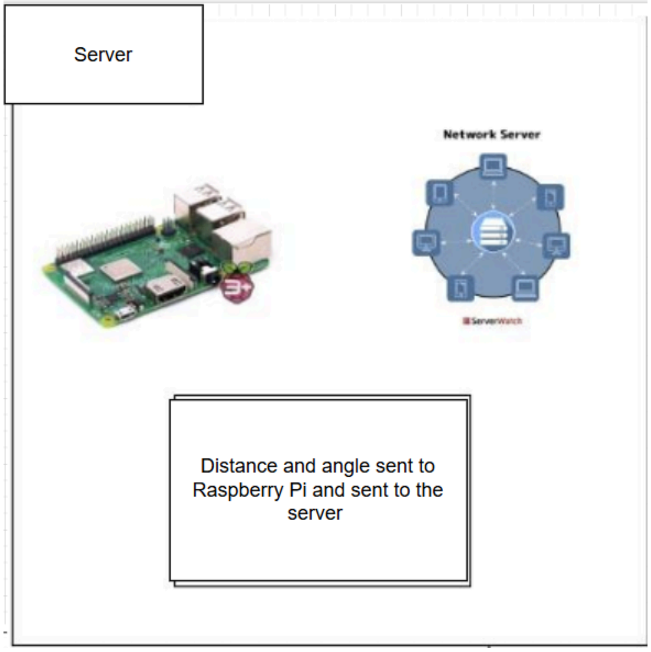
- Raspberry Pi - The Raspberry Pi receives data from the microcontroller and our display is able to receive the same information.
- Transducers (Transmitter and Receiver) are how the radar sends and receives the sound waves. The transducers will emit sound waves, and if the waves reflect and are received, then an object is detected, and the data is relayed to the Raspberry Pi.
- Microcontroller - The microcontroller is a middleman between the transducers and Raspberry Pi. It manages the timing of the signals sent by the transducers. It is also responsible for sending the echo data to the Raspberry Pi.
- Display - Show the data received from the Raspberry Pi and accurately depict objects that were detected. Our display will also have to process the data received from the Raspberry Pi.

4.3.2 Detailed Design and Visual(s)

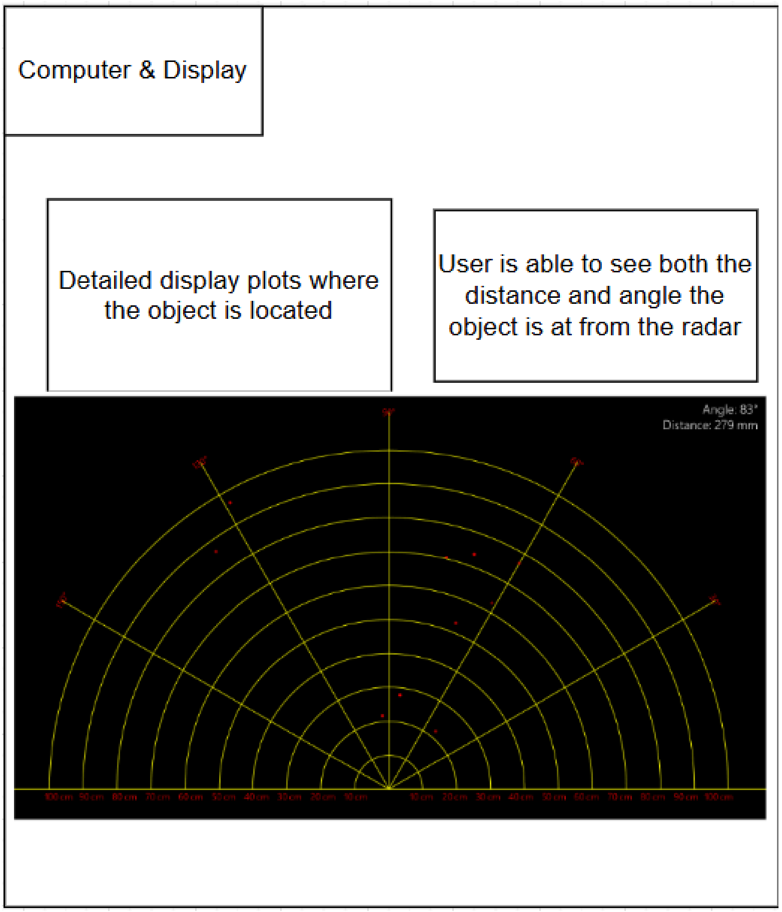
The ultrasonic radar project is an object detection system designed to measure distances and detect small objects using ultrasonic waves with a phased Array. The first major sub-system is the radar hardware, which is equipped with an Esp32S3 microcontroller, MA40S4S/R ultrasonic transducers, and signal amplification and processing circuitry. This is the heart of the system, with the MCU containing code to manage the phased array’s necessary phase delays and calculate distance and direction. The MA40S4/R transducer, optimized for 40 kHz operation, serves as both the emitter and receiver of ultrasonic waves. When triggered by the MCU, a pulse will be sent; if the waves encounter an object, they will be reflected back to the receiver, which then sends the received signal to the MCU. An amplification and filtering circuit amplifies the signal and reduces ambient noise to ensure these return signals are processed accurately. The amplifier circuit, typically an op-amp configured in a non-inverting layout, enhances weak return signals, while low-pass filters remove unwanted high-frequency noise that can interfere with detection accuracy. Finally, the power management circuitry supplies stable voltage across all components, ensuring consistent performance.



Another subsystem includes the Raspberry Pi. The Raspberry Pi is responsible for hosting the detected data to be sent to the display. The Raspberry Pi must connect to the MCU to receive new readings. Readings must be made available to the display with a wireless connection.



The last subsystem is the computer used for displaying the data in the form of a radar image. The display code should calculate the x and y distances based on the Euclidean distance and angle associated with each data point. It then uses this data to update the display, having each data point drawn as a point on the radar image in a timely manner.



4.3.3 Functionality

In its real-world context, the ultrasonic sensor system is designed to detect objects within a specific range and provide feedback based on the proximity and size of those objects. The user begins by placing the device in an environment where monitoring is required—such as a lab setup or mounted in a vehicle for obstacle detection. After positioning, the user powers on the device, allowing it to initialize the radar sensor and signal processing circuits. Once active, the user initiates the scanning process by pressing a button on the device or remotely controlling it through a paired interface, such as a mobile app or computer. The sensor system then emits ultrasonic pulses through the MA40S4/R transducer. As these pulses encounter objects, they reflect back, and the device measures the time taken for the echoes to return. This timing data enables the system to calculate the distance and, potentially, the size of any detected objects.

In response to the calculated distance, the system provides feedback to the user. For example, if an object is detected within a critical range, the device can alert the user through audio or visual cues or by sending a notification to the interface. The system continues to scan until it is manually deactivated or a specified object is identified, making it suitable for real-time tracking or monitoring applications. The system automatically adjusts pulse-echo timing based on environmental conditions, such as background noise or clutter, enhancing signal amplification when necessary to optimize range and accuracy. Detected data is stored or transmitted to an external device if required, allowing for distance logging over time for further analysis. A visual representation could illustrate each step: the user activates the system, the device sends a pulse, an object reflects the echo, the system processes the data, and feedback is provided based on the distance detected.

4.3.4 Areas of Concern and Development

The current design of the ultrasonic sensor system is well on its way to meeting the requirements, but several concerns need to be addressed to ensure accuracy and proper functionality in real-world conditions. The system aims to detect objects based on their proximity using ultrasonic pulses, providing feedback to the user based on the detected range. The Raspberry Pi processes data from the MA40S4S/R transducer, which is then amplified and filtered by a custom circuit. However, key issues must be considered in refining the system to fully meet user and client expectations.

1. **Primary Concerns:**

Accuracy: Object detection accuracy depends on the transducer's ability to send and receive pulses effectively. External factors like noise and object shape can cause measurement errors, especially in high-interference environments. Improved calibration and additional filtering or dual sensors may help address this.Signal Processing: The Raspberry Pi may not process signals fast enough for real-time applications, leading to delays or inaccuracies, particularly in dynamic environments. A microcontroller or DSP could improve real-time performance but would increase cost and complexity.

User Interface and Display: The system's effectiveness depends on how clearly the data is presented. Testing different display methods, such as visual or auditory feedback, is necessary to ensure clarity and usability.

Environmental Variability: Environmental factors like temperature and humidity can affect sound wave propagation, leading to accuracy issues. Regular calibration and possible environmental sensors can help improve performance in various conditions.

2. **Immediate Plans to Address Concerns:**

Enhanced Calibration: WE plan to refine the calibration process to better account for environmental factors and ensure accurate readings. I will test the system in different environments to understand how these factors impact performance.

Improving Signal Processing: We are considering integrating a microcontroller or DSP to offload some of the real-time signal processing from the Raspberry Pi. This would improve the responsiveness and accuracy of the system. Alternatively, optimizing the existing Raspberry Pi code to handle processing more efficiently could be a first step.

User Interface Testing: We will design and test different display formats to determine the most intuitive and effective way to present data to the user. This might include graphical displays or more basic feedback, such as LED indicators for proximity warnings.

Environmental Testing: We plan to conduct tests in various environmental conditions (different temperatures, humidity levels, and materials) to determine how sensitive the system is to these variables. This will help identify where adjustments or compensations are needed.

3. **Questions for Clients, TAs, and Faculty Advisors:**

For the Clients: Are there specific environments where this sensor system will be used, and how critical is the system's performance in those environments (e.g., varying temperature, humidity, or interference levels)?

For the TAs: Do you have suggestions for optimizing the Raspberry Pi's signal processing capabilities for real-time applications, or would integrating a microcontroller be a better route?

For Faculty Advisors: Are there alternative sensor technologies or enhancements (such as dual-transducer systems or frequency modulation) that could improve accuracy without significantly increasing cost or complexity? How can we best test and validate the system's accuracy under varying conditions?

4.4 TECHNOLOGY CONSIDERATIONS

MA40S4S/R Transducer:

- Strengths:
    - Reliable and accurate for short/medium-range detection; effective in controlled environments.
- Weaknesses:
    - Limited performance in high-noise environments or with irregularly shaped objects.
- Trade-offs:
    - Cost-effective but limited in adaptability to complex or noisy environments.
- Alternatives:
    - Dual-transducer systems or adaptive frequency modulation sensors, although they would increase cost and power consumption.

Raspberry Pi:

- Strengths:
    - Flexible, supports various programming languages, handles data processing and user interface integration.
- Weaknesses:
    - Limited processing speed and not optimized for real-time signal processing.
- Trade-offs:
    - Offers flexibility in programming but may introduce delays in high-frequency applications.
- Alternatives:
    - Dedicated microcontrollers or digital signal processors (DSPs) for faster real-time processing, but with reduced programming flexibility.

Custom Amplification and Filtering Circuit:

- Strengths:
    - Customizable to specific detection needs; boosts signal range and filters noise.
- Weaknesses:
    - Time-consuming to fine-tune; sensitive to environmental factors like temperature and humidity.
- Trade-offs:
    - Balances sensitivity and noise reduction but may risk false positives.
- Alternatives:
    - Pre-built amplifiers with adaptive filtering capabilities could complicate integration and add system complexity.

4.5 Design Analysis

Based on the testing we have done, all of our parts have worked on their own.

- MA40S4S/R
    - We have tested both the senders and the receiver, which have functioned as expected.
- Raspberry Pi 3B
    - It was able to host the web server locally.
- Display
    - We are using the processing IDE to display the location of the objects; we have tested our display using mock data and have gotten the results to display.

In the future, we will need to start combining our parts and getting everything to work smoothly together. As far as we are aware, the proposed design will work. We feel that this design is feasible in the limited time we have to complete this project.

5 Testing

5.1 Unit Testing

What we are testing:

- Transducers (MA40S4s/R):
    - Verify the transmitters emit 40 kHz ultrasonic pulses
    - Verify the receiver detects returning pulses accurately
- Signal Amplification Circuit:
    - Verify the gain matches theoretical values and filters noise effectively.
- Raspberry Pi Communication:
    - Test the Pi’s ability to receive and process data from the microcontroller.
- Microcontroller:
    - Test GPIO pins for transmitting and receiving pulses.
    - Test the timing of pulse emission and reception.

Tools:

- Test scripts on the Raspberry Pi to verify data reception

5.2 INTERFACE TESTING

Interfaces

1. Microcontroller to Transducers:
    - Verify that the ESP32-S3 correctly triggers the transducers and receives the reflected signal.
2. Microcontroller to Raspberry Pi:
    - Test data transmission to the Raspberry Pi via UART or wireless communication.
3. Raspberry Pi to Display:
    - Ensure processed data from the Pi is displayed correctly on the radar interface.

Tools:

- Python scripts on the Raspberry Pi for interface validation.
- Test environments in the Processing IDE to simulate data rendering.

5.3 INTEGRATION TESTING

Critical Integration Paths:

1. ESP32-S3 and Raspberry Pi:
    - Ensure seamless transmission of distance and direction data.
2. Radar System and Display:
    - Validate that object detection data is rendered correctly on the Processing IDE interface.

Justification for Criticality:

- These paths directly impact the real-time functionality and user experience of the radar system.

Testing Methodology:

- Simulate scenarios with objects at varying distances and angles.
- Verify real-time data transfer speed and accuracy.

5.4 SYSTEM TESTING

System-Level Testing Strategy:

- Test the entire radar system in a controlled environment.
- Place objects at known distances and angles, then compare the detected data against expected values.
- Validate the system's ability to handle varying environmental conditions like temperature and noise.

Tools:

- Ultrasonic calibration environment.
- Logs from Raspberry Pi and ESP32-S3 for debugging.

5.5 REGRESSION TESTING

Testing Plan:

- Maintain a suite of unit tests for all critical components.
- Re-run unit, integration, and system tests after adding new features or making modifications.
- Track key features (e.g., accuracy, real-time responsiveness) to ensure they remain functional.

Tools:

- Automated test scripts on the Raspberry Pi.

5.6 ACCEPTANCE TESTING

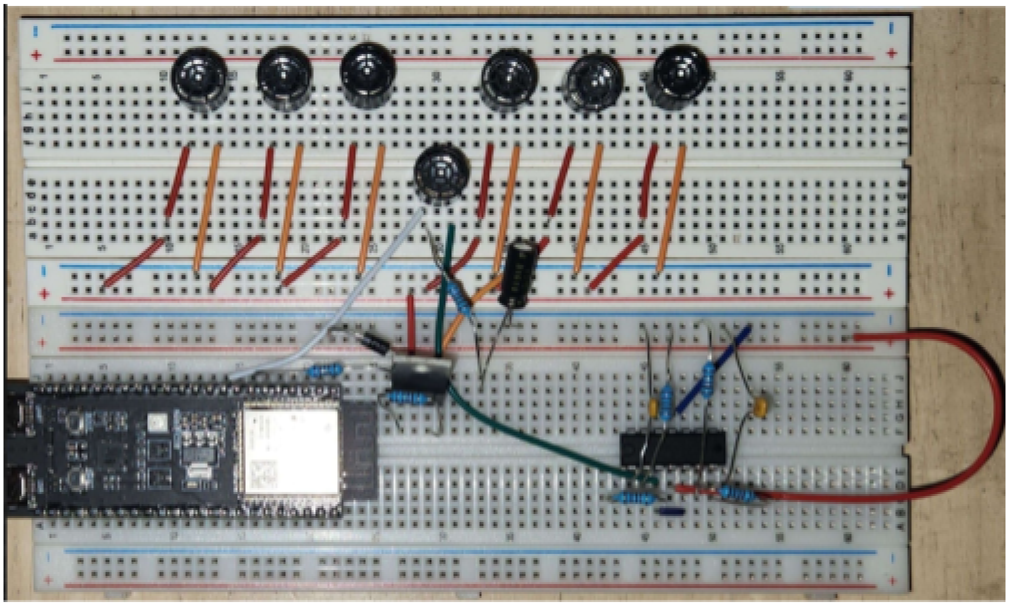
Testing Plan:

- Demonstrate the radar system to the client (Professor Song) in a controlled environment.
- Provide data on:
    - Accuracy of object detection.
    - Real-time performance (data transfer and rendering within 50 ms).
    - Display clarity and usability.
- Obtain client feedback on usability and identify any areas for refinement.

5.7 RESULTS

Our results from testing so far include successful communication between the ESP32 and the Raspberry Pi, as between the Raspberry Pi and the web server. The circuit also successfully sends and receives a signal. The display properly displays data in a readable format but could use some improvements to improve interactivity. The display could benefit from zooming, panning, and cursor tracking. Additionally, color coding could be added to mark different objects. Using C++ with OpenGL or SFML could potentially reduce delays. The next major steps to implement include setting up the phased array functionality, adding additional receivers, and potentially adding a dedicated power supply.

6 Implementation



Prototype #1: Ultrasonic Radar Sensor Circuit

Initial Testing Progress:

1. Hardware:
    - One MA40S4S transmitter and one MA40S4R receiver tested for functionality.
    - The transmitter emits 40 kHz pulses; the receiver detects reflected signals up to 1 meter.
2. Microcontroller (ESP32-S3):
    - Programmed to send 40 kHz pulses and measure time delay.
    - Initial tests confirm accurate distance measurements displayed in the Serial Monitor.
3. Data Display:
    - Mock data visualized on Processing IDE to validate radar display logic.

Next Steps:

1. Amplify Signals: Add a circuit to enhance receiver accuracy so that it does not have as much extra noise.
2. Display Integration: Visualize real-time distance data on the radar display.
3. Scaling: Validate the single-transducer setup before expanding to a phased array.

7 Ethics and Professional Responsibility

7.1 AREAS OF PROFESSIONAL RESPONSIBILITY/CODES OF ETHICS

Area of Responsibility	**Definition**	**Relevant Item from Code of Ethics**	**How it Applies**
Public health, safety, and welfare	Ensuring the system is safe to use and does not harm stakeholders or the environment.	IEEE Code, #1: "To hold paramount the safety, health, and welfare of the public."	Ensured the ultrasonic frequency (40 kHz) is above the human hearing range. Added safety features to minimize environmental impact.
Global, cultural, and social	Respecting the values and practices of affected communities.	IEEE Code, #5: "To improve the understanding of technology, its appropriate application, and potential consequences."	The project design minimizes noise pollution and avoids cultural disruptions
Environmental	Minimizing the environmental impact of the project.	IEEE Code, #8: "To treat all persons fairly and with respect, and to protect the environment."	Used energy-efficient components and avoided hazardous materials
Economic	Ensuring affordability for potential users.	IEEE Code, #3: "To seek, accept, and offer honest criticism of technical work."	Selected cost-effective components like the MA40S4S/R

Performing well:

Economic- we chose components based mostly on price as long as they would work for this project.

Not performing well:

Public health, safety, and welfare - currently, our transducers have a high sound intensity (120 db), but we plan on under volting them to reduce the sound intensity.

.

7.2 Four Principles

	Beneficence	Nonmaleficence	Respect for Autonomy	Justice
Public health, safety, and welfare	Detects objects a meter away, and provides a learning opportunity to a theoretical user.	None	Provides users with reliable data, a user changes the voltage to reduce sound intensity	None
Global, cultural, and social	Contributes to small-scale Applications	Avoids harm to global practices	Purchasing this device is optional, so its use it not forced	None
Environmental	Could be used in environmentally friendly applications	Minimal environmental harm by using low-power components	None	None
Economic	Provides a cost-effective solution for object detection in small-scale systems	The device should be reliable, parts should not be easily damaged	Allows scalability with use in budget-friendly applications	None

Important Pair: Public Health - Beneficence: The radar system enhances safety by providing real-time object detection.

Lacking Pair: Environmental - Nonmaleficence: More testing is needed to confirm minimal environmental impacts.

7.3 Virtues

List and define at least three virtues that are important to your team. Describe what you will do or have done as a team to support these virtues among all team members.

Team Virtues:

- **Communication:** Team members should clearly describe their contributions and make sure everyone is on the same page.
- **Accountability:** Team members should make sure that they complete all of their required tasks in a timely manner or explain their difficulties and seek assistance.

- **Dedication:** Team members should be willing to face challenges and ensure they are producing quality work.

Each team member should also answer the following:

Brock Dykhuis:

- Cooperation - have demonstrated so far
    - This is important because while working as a team, helping each other can help to achieve the common goal we have.
    - I have helped our team download the Arduino IDE and set it up with the ESP 32.
- Reliability - have not demonstrated the best so far
    - This is important as in a team project, some parts will need to be done before others can start their parts.
    - In the future, I need to get my parts done sooner as sometimes it can take a few weeks longer to complete my parts.

Nathaniel Clarke:

- Demonstrated Virtue
    - Communication
        - This is important because a team that works well together produces a much better product, and has an enjoyable experience. This virtue ensures that a team is working efficiently.
        - I demonstrated this by helping teammates understand the setup for the Expressif IDF, and the necessary driver that needed to be installed
- Virtue needing improvement:
    - Documentation
        - This is important as it lets teammates understand what you have completed, helping with future planning.
        - Some weeks, I did most of my work on Wednesday, which led to my Weekly report documentation being somewhat rushed and undetailed.

Jonathon Madden:

- Demonstrate Virtue
    - Communication
        - This is important because it allows for transparency in the project. It allows everyone to be on the same page
        - I have demonstrated it by communicating what I've done during our project group meetings.
- Virtue needing improvement
    - Perseverance

        - This is important because it allows the problems of the project to be solved in a more timely manner.
        - I need to improve on this virtue because instead of figuring out how to use Expressif IDF, I decided to try the Arduino IDE.

Nicholas Jacobs

- Demonstrate Virtue
    - Communictation
        - Communication is essential in a team setting to ensure everyone understands their roles, responsibilities, and how their work contributes to the project. It enables the exchange of ideas, problem-solving, and coordination, which are critical for addressing challenges like design flaws or timing issues. Clear communication also fosters collaboration, reducing misunderstandings and improving the overall efficiency and success of the project.
- Virtue needing improvement
    - Reliability/Time-Management
        - Reliability is important because it builds trust within a team and ensures that tasks are completed on time to keep the project on track. At times, I've struggled with reliability due to conflicts with my NROTC commitments, which have occasionally caused me to miss meetings. To improve, I need to manage my time better by balancing my responsibilities more effectively and planning ahead to avoid schedule conflicts.

8 Closing Material

8.1 CONCLUSION

Our research has focused on understanding the principles of radar technology and identifying the necessary components to construct a radar system. Using these components, we have successfully designed and built a basic radar system featuring one receiver and one transmitter. The collected data has been visualized on our display.

Our project aims to maintain a high level of professionalism in the way we communicate, improve upon previous implementations, and ultimately develop a high-quality ultrasonic radar system. Our goal is to expand the system to include ten transmitters and at least two receivers, which will enhance the accuracy of our scans. This data will be transmitted to a server hosted on a Raspberry Pi and subsequently displayed in a detailed and user-friendly format.

Currently, we have an initial circuit design supporting six of the ten transmitters and have begun testing with a signal transmitter and receiver. We will need to continue to work with the circuit to implement all 10 transmitters and potentially 3 receivers. This will require software implementation of phase delay. These steps will be completed in the Spring semester of 2025.

One major constraint we experienced was getting used to the MCU, it would have been better to order the MCU earlier to allow us to become more familiar with how it should be programmed. Since we only have one MCU, only one person could run tests at a time, so it was more difficult to perform testing. Ordering the MCU earlier would save more time for all team members to gain experience with the MCU, as it would allow other team members to complete other early tasks while waiting to access it.

8.2 References

[1] SDMay24-24 Team, "Sonar: Ultrasonic Radar," Department of Electrical and Computer Engineering, Iowa State University. [Online]. Available: https://sdmay24-24.sd.ece.iastate.edu/. [Accessed: Dec. 6, 2024].

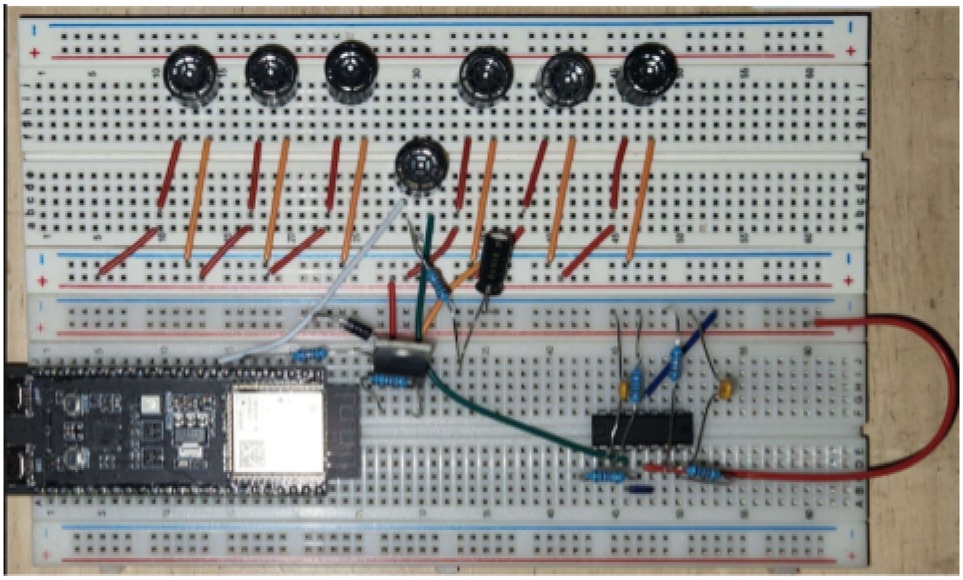
[2] SDMay23-22 Team, "Ultrasonic Radar," Department of Electrical and Computer Engineering, Iowa State University. [Online]. Available: https://sdmay23-22.sd.ece.iastate.edu/. [Accessed: Dec. 6, 2024].

[3] K. McChesney, "Ultrasonic Radar System," presented in "Ultrasonic Radar (Object Detector)," J. Song, Iowa State University, Dept. of Electrical and Computer Engineering, PowerPoint presentation, Sept. 9, 2024.

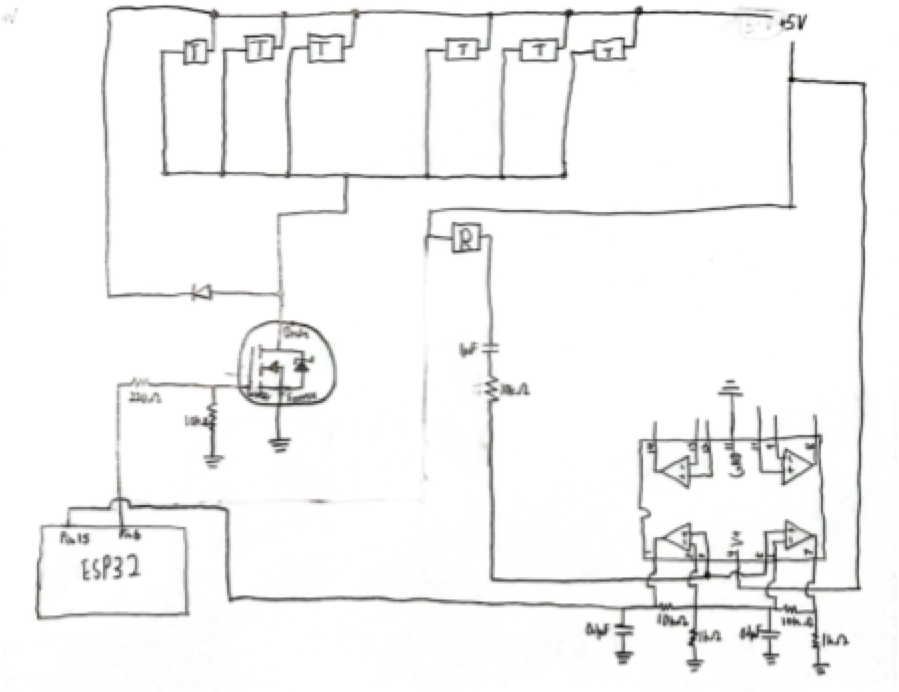
[4] ScienceDirect, “Phased Array,” [Online]. Available: https://www.sciencedirect.com/topics/engineering/phased-array. [Accessed: Dec. 6, 2024]

[5] W. R. Hedrick, D. L. Hykes, “Ultrasonic phased arrays,” *Adaptive Phased Array Technology II*, vol. 12, no. 5, pp. 391–406, Sept. 1996. [Online]. Available: https://journals.sagepub.com/doi/pdf/10.1177/875647939601200502. [Accessed: Dec. 6, 2024].

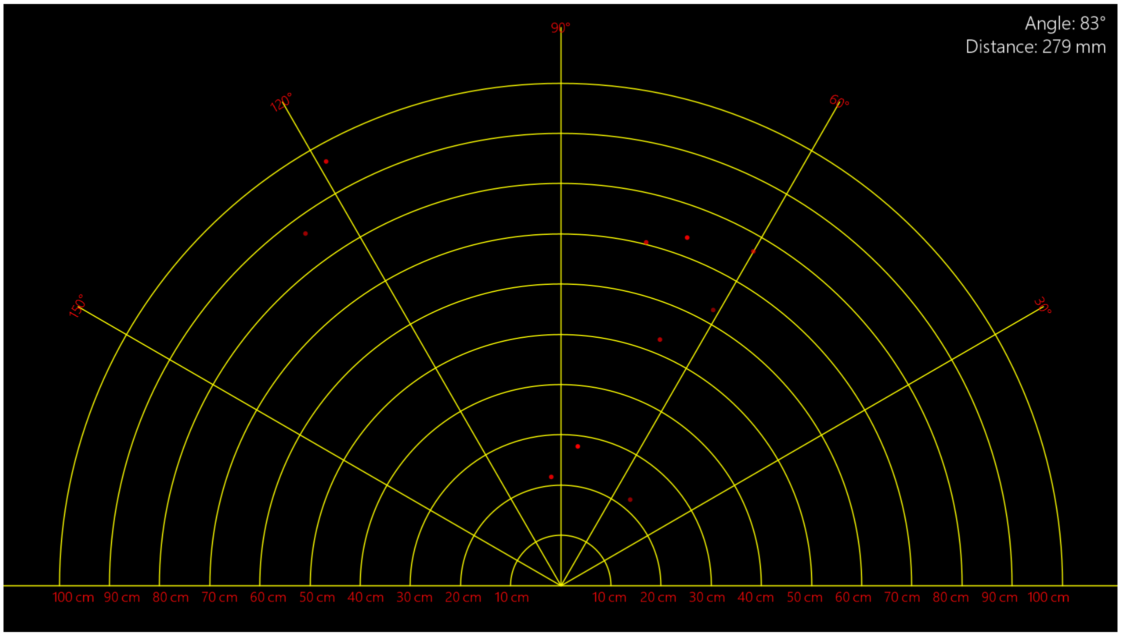
8.3 APPENDICES



Prototype #1: Ultrasonic Radar Sensor Circuit



Theoretical/Schematic of Circuit



Radar Display With Sample Data

9 Team

9.1 Team Members

Major	**Names**
Computer Engineering	Brock Dykhuis
Electrical Engineering	Nicholas Jacobs
Software Engineering	Nathaniel Clarke Jonathon Madden

9.2 Required Skill Sets for Your Project

- Circuit Design
- Circuit Analysis
- Signal Design

- Circuit Troubleshooting
- Web Design
- UI Software Design
- Software Testing
- Embedded Systems Programming
- Networking with Raspberry PI
- Pulse/Phase Control

9.3 SKILL SETS COVERED BY THE TEAM

Circuit Design	Nicholas Jacobs
Circuit Analysis	Nicholas Jacobs
Signal Analysis and Troubleshooting	Nicholas Jacobs
Component Testing	Nicholas Jacobs
Filtering Design/Timing and Synchronization	Nicholas Jacobs
Web Design	Nathaniel Clarke, Jonathon Madden
UI Software Design	Nathaniel Clarke, Jonathon Madden
Software Testing	Nathaniel Clarke, Jonathon Madden
Embedded Systems Programming	Brock Dykhuis
Networking With Raspberry PI	Brock Dykhuis
Pulse/Phase Control	Nathaniel Clarke, Jonathan Madden

9.4 PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM

The waterfall project management style will be adopted by our team.

9.5 INITIAL PROJECT MANAGEMENT ROLES

Team Member	**Role**
Nathaniel Clarke	Project Software Designer
Brock Dykhuis	Circuit Analysis
Jonathon Madden	UI Designer & Software Tester
Nicholas Jacobs	Electronics and Circuit

9.6 TEAM CONTRACT

Team Members:

1) Brock Dykhuis 2) Nathaniel Clarke

3) Nicholas Jacobs 4) Jonathon Madden

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:

   Weekly team meetings will be held at 2 pm on Thursdays in the TLA.

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

   The preferred method of communication for the group is through a group text; most scheduling, reminders, and issues will be addressed this way. The group text should be used to ensure quick responses, if something is missed we will use our secondary form of communication. This would be face-to-face communication, which can primarily be done during meetings, and at the beginning and end of class.

3. Decision-making policy (e.g., consensus, majority vote):

   Our decision making policy will primarily be consensus based. More weight may be given to a team member's input if they have more expertise in a topic or task.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

   Nathaniel Clarke will keep minutes for group meetings. These will be shared/archived in google documents in the group Google Drive folder.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

   All group members are expected to attend all group and client meetings. If someone is unable to attend, they will send a message at least one day before a meeting is scheduled. All group members are expected to arrive on time for meetings, if some one expects to be late they should message the group. During meetings all group members are expected to contribute to the discussion of completed and planned tasks.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

   Contributions to team assignments should be roughly proportional. If some project tasks prove more time consuming, the group may decide to allow a team member to have a smaller contribution to compensate. Timelines and deadlines should be fairly solid, so all work and tasks should be completed prior, with few exceptions.
3. Expected level of communication with other team members:

   Communication should be very frequent and thorough to ensure all team members understand the current progress on the project. Significant updates will be introduced during group meetings. Otherwise, all team members should provide updates at least daily.

4. Expected level of commitment to team decisions and tasks:

   Major decisions should have equal team commitment. Tasks should be equally distributed as much as possible. Tasks will be mainly distributed based on skills and expertise of each individual. Some decisions may be restricted to a team member based on their background.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

   Brock Dykhuis - Team Organization, Server setup, Circuit Analysis

   Nathaniel Clarke - Client Interaction, Project management, Software Design

   Nicholas Jacobs - Circuit Analysis Expert, Circuit Design

   Jonathon Madden - Testing, UI design, Weekly Report

2. Strategies for supporting and guiding the work of all team members:
   To guide project progress, a task/issue board will be used. This will track tasks to complete, tasks in progress, and completed tasks. Tasks can be arranged into weekly or longer term

sprints. During group meetings tasks to be added will be discussed, which will be the expected work to be completed based on their designated completion time.

3. Strategies for recognizing the contributions of all team members:
   The primary method of recognizing team contributions will be showing new contributions at the beginning of each group meeting. This may include a demonstration of how the new changes impact functionality. As mentioned above an task/issue board will be used, which shows which team member completed a particular task.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

   Brock Dykhuis - Has some experience with hardware design, can use LTspice to model circuit design. Experience with C and Java, also has minimal experience with python.

   Nathaniel Clarke - Has expertise in software development, experience with C and C++ will prove to be important for the project. Experience with HTML will also prove helpful in creating the visuals for the system.

   Nicholas Jacobs -

   Jonathon Madden - Has knowledge in software testing and development, using java, C, and C++. Will assist with creating a functional and visually appealing UI.

2. Strategies for encouraging and supporting contributions and ideas from all team members:

   During group meetings the group will insure everyone has contributed to the conversation. The group will discuss any potential challenges someone is dealing with, and determine ways to help if needed. The conversation will also allow the team to bounce ideas off each other, and encourage potential changes to implement.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

   If there are any issues with resolving collaboration or inclusion issues, a team member first tries to address the problem directly. The first step is to message the team respectfully explaining their concerns, all team members should be willing to accept constructive criticism. If it does not succeed, the situation is severe, or a team member is uncomfortable they may discuss the issue with the team advisor or an instructor.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

   - Strive to have consistent and comprehensive communication.
   - Create an effective and professional quality design document.
   - Improve radar efficiency from past projects.

- Create a readable and intuitive UI for the radar.
- Ultimately create a functioning consistent Ultrasonic radar system that is high quality.

2. Strategies for planning and assigning individual and team work:

   An issue/task board will be created as mentioned above. During team meetings we will discuss tasks to add, and who they may be best to assign to.

3. Strategies for keeping on task:

   The team will follow a waterfall-style development cycle, ensuring that progress is made each week.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

   Assuming that an infraction is minor a warning will be issued either through group text or during a group meeting. If the infraction is severe it may be discussed with the team's advisor or a course instructor.

2. What will your team do if the infractions continue?

   If infractions continue the team's advisor and a course instructor should be contacted to observe to ensure that infractions do not continue.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*
1) ___Nathaniel Clarke ________________________________ DATE ____9/18/2024_______
2) ___Nicholas Jacobs_________________________________ DATE ____9/18/2024_______
3) ___Brock Dykhuis__________________________________DATE ____9/18/2024_______
4) ___Jonathon Madden________________________________ DATE ____9/18/2024_______