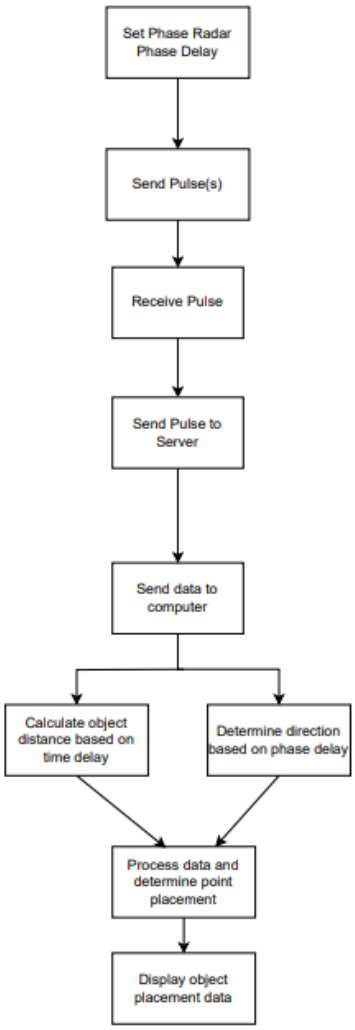
3 Project Plan

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

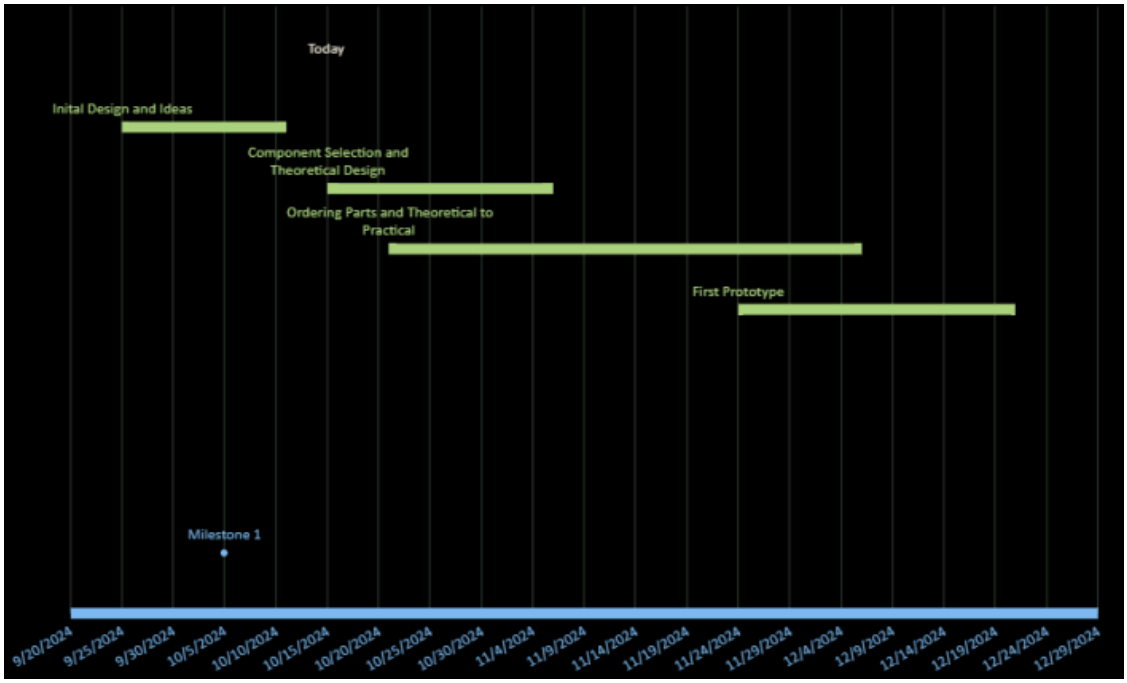
Task 4: Data Processing

- 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



Initial Design and Ideas

- Figuring out which parts to order, also includes initial research into our project to gain an understanding of what we are tasked to do.

Component Selection and Theoretical Design

- We will find the parts and components we plan on using.
- Once we have figured out these parts, we can start doing theoretical design of circuits.

Ordering Parts and Constructing a Design

- We will need to build this theoretical design on a breadboard, this will allow us to determine if our theoretical design works. If not we will need to make changes.
- After we have a working design we can start transmitting the data to try and get a working display.

First Prototype

- We have constructed a first prototype to demo during our faculty panel.

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

Software Resource Requirements

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